



T
004.68
VQU
V.1

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

“Integración y reimplementación en una plataforma orientada a objetos de un Sistema de Administración de Recursos Académicos y de un Sistema de Comunicación Alumno–Profesor”



D-26931

CIB

TESIS DE GRADO

Previo a la Obtención de los Títulos de:

INGENIERA EN COMPUTACIÓN y

LICENCIADA EN SISTEMAS DE INFORMACIÓN

V.1

Presentada por:

VERÓNICA ISABEL UQUILLAS GÓMEZ

MERCY DEL ROCIO GONZÁLEZ YÉPEZ

GUAYAQUIL – ECUADOR

2001

AGRADECIMIENTO

Agradecemos de manera muy especial al Ing. Guido Caicedo, nuestro Director de Tesis, por su valiosa ayuda en el desarrollo de este proyecto. Además de ser nuestro mentor, se convirtió en un gran amigo para nosotras.

Además agradecemos a las personas que de una u otra forma nos dieron su apoyo: Ing. Carlos Monsalve, Ing. Efrén Villamagua, Sr. Neil Núñez, Srta. Katty Siavichay y a Eduardo Uquillas.

DEDICATORIA

A Dios, por nuestra vida y por su grande amor para nosotras.

A nuestros padres, por habernos inculcado el sentido de lucha para alcanzar nuestras metas y por el apoyo incondicional que nos brindan.

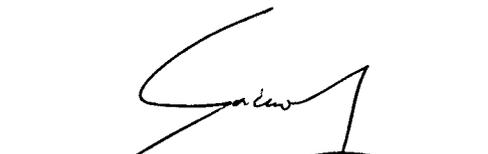
TRIBUNAL DE GRADUACIÓN



ING. LENIN FREIRE
Miembro del Tribunal



ING. MONICA VILLAVICENCIO
Miembro del Tribunal



ING. GUIDO CAICEDO
Director de Tesis



ING. CARLOS MONSALVE
Subdecano de la FIEC

DECLARACIÓN EXPRESA

“La responsabilidad por los hechos, ideas y doctrinas expuestos en esta Tesis de Grado, nos corresponde exclusivamente; y, el patrimonio intelectual de la misma, a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”.

(Reglamento de Graduación de la ESPOL).

Verónica Isabel Uquillas Gómez

Mercy del Rocio González Yépez

RESUMEN

Actualmente, los sistemas de apoyo a la educación utilizando la Internet han tenido un desarrollo considerable en los últimos años, desde que el Word Wide Web (WWW) se volvió una herramienta popular y poderosa para la comunicación a todo nivel.

Cursos Web es un sistema que permite mantener una comunicación constante y confiable entre el profesor y sus estudiantes, de tal manera que el estudiante pueda estar informado de las actividades desarrolladas y de las responsabilidades que debe cumplir como miembro del curso. Por otro lado, el profesor recibirá de manera segura las tareas enviadas por sus alumnos (con lo que se elimina el riesgo de pérdidas de tareas), así como el poder publicar las calificaciones respectivas.

Debido a que en la actualidad la Institución cuenta con una gran cantidad de recursos que no son administrados correctamente (ya que se desconoce quién los posee, administra o utiliza), Cursos Web incluye un componente

que permite que los recursos existentes en las unidades sean administrados de manera eficiente y automática mediante un control de solicitudes y asignaciones. De esta manera se logra minimizar el tiempo necesario para la realización del requerimiento y aprobación del mismo.

No es necesario que la aplicación resida en un lugar específico, puesto que ha sido desarrollada con bases en la arquitectura TCP/IP y utiliza la tecnología Web, lo cual representa una gran ventaja para los usuarios de la aplicación.

En el capítulo 1 de este documento, encontrará detalles sobre los antecedentes y justificaciones para el desarrollo de este proyecto.

El capítulo 2 provee explicaciones sobre todos los componentes necesarios para el desarrollo de aplicaciones sobre la Internet.

En el capítulo 3 se ha definido el alcance del sistema, junto con los detalles de la etapa de análisis del mismo, lo cual ayudará a entender los requerimientos necesarios.

En los capítulos 4 y 5 se encuentran detalles de diseño, implementación y pruebas aplicadas al desarrollo de Cursos Web.

ÍNDICE DE ABREVIATURAS

API	Application Program Interface
ASP	Active Server Pages
CGI	Common Gateway Interface
COM	Component Object Model
DBMS	Data Base Management System
DHTML	Dynamic HTML
DIO	Diagramas de Interacción de Objetos
HTML	Hypertext Markup Language
HTTP	Hypertext Transport Protocol
IE	Internet Explorer
IP	Internert Protocol
JSP	JavaServer Pages
POP3	Postal Office Protocol v3
SAC	Sistema de Administración de Cursos
SAR	Sistema de Asignación de Recursos
SQL	Standard Query Language
SSL	Secure Socket Layer
TCP	Transmisión Control Protocol
UML	Unified Modeling Language
URL	Universal Resource Locator
WWW	World Wide Web

ÍNDICE GENERAL

RESUMEN	5
ÍNDICE DE ABREVIATURAS.....	7
ÍNDICE GENERAL.....	8
ÍNDICE DE FIGURAS	11
ÍNDICE DE TABLAS.....	13
1. INTRODUCCIÓN Y JUSTIFICACIÓN.....	14
1.1. Antecedentes	14
1.1.1. Revisión de prototipo de Aplicación del sistema de manejo de recursos	15
1.1.2. Revisión de prototipo de Aplicación para comunicación Alumno-Profesor.....	16
1.1.3. Limitaciones y problemas de prototipos existentes.....	17
1.2. Justificación	21
1.3. Usuarios del Sistema	25
2. DESARROLLO DE APLICACIONES USANDO EL WWW	28
2.1. Desarrollo de Sitios Web.....	28
2.1.1. El modelo CLIENTE/SERVIDOR	30
2.1.1.1. Cliente Web o Browser	33
2.1.1.2. Servidor Web	35
2.1.2. Generación de páginas HTML	36
2.1.2.1. Páginas estáticas.....	37
2.1.2.2. Páginas dinámicas cliente	38
2.1.2.3. Páginas dinámicas servidor.....	39

2.1.3. El modelo THREE-TIER	42
2.1.4. Bases de Datos y su uso en Internet	45
2.2. Plataformas y herramientas para el desarrollo de Sitios Web.....	46
2.2.1. Servidores.....	47
2.2.2. Common Gateway Interface (CGI).....	48
2.2.2.1. Lenguajes para la implementación de CGIs.....	51
2.2.2.2. ¿Cuándo usar un CGI?.....	52
2.2.3. Bases de Datos	54
2.2.4. Sistemas Operativos	57
2.2.5. Herramientas para el desarrollo de páginas Web.....	59
2.3. Seguridad de Sistemas basados en el WWW	63
2.3.1. Servicios de seguridad	63
2.3.2. Seguridades usando el protocolo SSL.....	64
2.3.3. Autenticación	66
2.3.4. Cookies.....	68
3. ALCANCE Y ANALISIS DEL SISTEMA.....	70
3.1. Especificaciones del Sistema	71
3.1.1. Objetivos.....	72
3.1.2. Descripción y alcance del Sistema.....	74
3.1.2.1. Descripción de Actores.....	76
3.1.2.2. Alcance y Límites del Sistema.....	81
3.2. Especificación de Casos de Uso y Escenarios.....	87
3.2.1. Especificación de Casos de Uso.....	88
3.2.2. Especificación de escenarios	102
3.3. Definición de Clases	132
3.4. Diagrama de interacción de objetos	135
3.5. Diagrama de clases	212
3.6. Análisis de la Interacción Hombre-Máquina	220
3.7. Flujo de ventanas.....	224
3.7.1. Sistema de comunicación Alumno-Profesor.....	225
3.7.2. Sistema de administración	228

4. DISEÑO.....	231
4.1. Diseño del módulo de estandarización de la interfaz.....	231
4.1.1. Archivos de Formato y su Metalenguaje.....	234
4.2. Diseño de las plantillas del Sitio Web.....	243
4.3. Diseño del sistema.....	255
4.4. Diseño de la Base de Datos.....	260
4.5. Diseño del esquema de seguridad.....	280
4.5.1. Autenticación.....	280
4.5.2. Cookies.....	281
4.6. Diagramas de clases de diseño.....	284
4.7. Diagramas de interacción de objetos de diseño.....	292
4.8. Diseño del procedimiento de migración de datos.....	309
4.9. Plan de pruebas.....	312
5. IMPLEMENTACIÓN Y PRUEBAS.....	318
5.1. Estándares de implementación del sistema.....	318
5.2. Instalación del sistema en el servidor de la FIEC.....	321
5.3. Migración de datos de prueba.....	323
5.4. Pruebas y resultados de las pruebas.....	323
5.5. Problemas de Implementación.....	324
CONCLUSIONES Y RECOMENDACIONES.....	327
APÉNDICES.....	331
BIBLIOGRAFÍA.....	525

ÍNDICE DE FIGURAS

Figura 2.1	Modelo Cliente/Servidor del WWW	31
Figura 2.2	Generación de páginas estáticas	37
Figura 2.3	Generación de páginas dinámicas servidor	42
Figura 2.4	Capas del modelo Three-Tier.....	43
Figura 2.5	Modelo Three-Tier.....	45
Figura 3.1	Casos de Uso y Actores.....	75
Figura 3.2	Vista de un escenario.....	136
Figura 3.3	Flujo principal de ventanas para profesores.....	225
Figura 3.4	Flujo de ventanas de la opción “Clases”	226
Figura 3.5	Flujo de ventanas de la opción “Curso”	226
Figura 3.6	Flujo principal de ventanas para estudiantes	227
Figura 3.7	Flujo de ventanas de la opción “Clases”	227
Figura 3.8	Flujo de ventanas de la opción “Curso”.....	228
Figura 3.9	Flujo de ventanas principal para administradores	228
Figura 3.10	Flujo de ventanas de la opción “Parámetros”	229
Figura 3.11	Flujo de ventanas de la opción “Accesos”	229
Figura 3.12	Flujo de ventanas de la opción “Mantenimiento”	230
Figura 3.13	Flujo de ventanas de la opción “Migración”	230
Figura 4.1	Relación entre los diferentes componentes del Metaformato..	233
Figura 4.2	Modelo de páginas del sistema	243
Figura 4.3	Diseño de la página de acceso al Sistema.....	244
Figura 4.4	Diseño de página basada en el modelo combinado	245
Figura 4.5	Componentes de las páginas.....	246
Figura 4.6	Tipos de cabecera horizontal	247

Figura 4.7	Tipos de cabecera Vertical.....	249
Figura 4.8	Ventana de error	253
Figura 4.9	Ventana de confirmación de usuario y contraseña.....	254
Figura 4.10	Arquitectura de Cursos Web	255
Figura 4.11	Estructura de directorios de trabajo de usuarios	257
Figura 4.12	Estructura de los directorios de programas.....	259

ÍNDICE DE TABLAS

Tabla I	Sistemas Operativos	58
Tabla II	Definición de clases.....	135
Tabla III	Directorios de trabajo de usuarios.....	259
Tabla IV	Directorios de programas.....	260
Tabla V	Descripción de Tablas de Base de Datos.....	273
Tabla VI	Scripts de exportación de datos.....	310

CAPÍTULO 1

1. INTRODUCCIÓN Y JUSTIFICACIÓN

Los sistemas de apoyo a la educación utilizando la Internet han tenido un desarrollo considerable en los últimos años especialmente, desde que el Word Wide Web (WWW) se volvió una herramienta popular y poderosa para la comunicación a todo nivel. En la ESPOL ya se han realizado intentos por desarrollar este tipo de aplicaciones, como proyectos de tópicos para la obtención del título de Ingeniero en Sistemas. En la actualidad existen dos prototipos, los mismos que no están en funcionamiento debido a que tienen deficiencias funcionales y no incluyen todo lo necesario para aprovechar los recursos que ofrece la Internet de una manera adecuada y práctica.

1.1. Antecedentes

En esta sección revisaremos los proyectos desarrollados en el Tópico de Graduación “TCP/IP e Internet y Programación Cliente/Servidor” realizado en el año de 1999 y que sirvieron de base para el desarrollo

de esta tesis. Mencionaremos sus funcionalidades principales y detallaremos las limitaciones y problemas que presentan.

1.1.1. Revisión de prototipo de Aplicación del sistema de manejo de recursos

El Sistema de Asignación de Recursos (SAR) es una aplicación Cliente/Servidor desarrollada para usarse en la ESPOL y diseñada para aprovechar el paradigma del Web. Permite que a través de la Internet los profesores reserven recursos tales como aulas, pizarras, proyectores, etc., que necesitan para sus clases y que los alumnos hagan consultas sobre esta información. Además esta aplicación provee un módulo que le permite a una persona administrar las reservaciones realizadas para aprobarlas o rechazarlas y también configurar los tipos y la cantidad de recursos existentes

En el SAR se puede realizar tareas de administración, reservación o consulta de acuerdo al rol del usuario que puede ser de: de administrador, profesor, o estudiante.

En conclusión, SAR es en un sistema que permite automatizar el manejo de los recursos, entendiéndose por recurso todo aquello que

las Unidades Académicas ofrecen a los profesores para el desarrollo de las clases.

1.1.2. Revisión de prototipo de Aplicación para comunicación Alumno-Profesor

El Sistema de Administración de Cursos (SAC), es otra aplicación Cliente/Servidor desarrollada para usarse en la ESPOL y que al igual que SAR, busca aprovechar el paradigma del Web.

SAC provee una interacción virtual entre el profesor y los estudiantes de un curso, permitiéndoles intercambiar información y documentos útiles fuera del aula de clase a través de una aplicación Web.

A través de SAC, el profesor puede publicar información del curso, que incluye: anuncios, políticas, cronogramas, tareas y calificaciones. Por otro lado, el estudiante puede responder las tareas y consultar la información que el profesor haya puesto a disposición de los mismos.

1.1.3. Limitaciones y problemas de prototipos existentes

Tanto SAR como SAC poseen muchas limitaciones y problemas que, de no ser resueltos, impiden una aplicación práctica y satisfactoria para el entorno universitario.

A continuación presentaremos las limitaciones y problemas de los prototipos mencionados:

- Ambos sistemas utilizan la información académica de la ESPOL, proveniente del Sistema Académico¹. Debido a que no existe una comunicación en línea con el Sistema Académico, los datos deben de ser migrados a ambos sistemas. Sin embargo, la migración de los datos se tiene que hacer manualmente, porque ninguno de los sistemas cuenta con un proceso que realice esta operación de manera automática.
- No existe un módulo de Administración para SAR o para SAC, que maneje todo lo referente a la administración, como usuarios, paralelos, parámetros, etc.
- En SAR existe un módulo de mantenimiento de datos, pero este no posee las opciones para ingresar los cursos o paralelos, datos sin

los cuales tanto SAR como SAC no pueden trabajar. Por lo tanto, ambos sistemas dependen del ingreso de los datos de las tablas correspondientes a través de comandos del manejador de base de datos.

Otras limitaciones que presenta específicamente SAC, son:

- Las opciones que posee SAC no permiten un manejo total del curso, ya que solamente abarca los siguientes temas:
 - Anuncios del Profesor
 - Tareas
 - Políticas del Curso
 - Cronogramas (Programa de clases con fechas)
 - Notas de Estudiantes
- SAC no cuenta con opciones para publicar recursos académicos, información general o las clases de cada día.
- Los cronogramas no son manejados por calendarios.

- Las notas de los estudiantes no pueden ser subidas al sitio Web del curso usando archivos existentes en el PC del profesor.
- No existe un mecanismo de intercambio de ideas entre los participantes del curso.
- Los anuncios y tareas se pueden publicar como texto o como un archivo general (documento de Word, Excel, etc.), mientras que las políticas del curso y los cronogramas sólo pueden ser publicadas como texto.
- El manejo de notas se realiza con módulos desarrollados en Java que se basan en la utilización de applets que se comunicaban directamente con la base de datos vía JDBC. Esto presenta la desventaja de tener que utilizar un canal de comunicación alternativo al protocolo HTTP utilizado por el Web que además es lento cuando se utiliza a través de enlaces remotos de baja velocidad. Además este módulo tenía fallas que impedían una operatividad del 100%.
- Cada vez que finaliza un período académico, la información queda obsoleta, ya que el sistema no cuenta con opciones para la reutilización de la misma. Esto causa molestias para los profesores que dictan una misma materia en varios períodos, porque tendrán

que realizar el ingreso de políticas, cronogramas, etc. tantas veces como dicten la materia.

- SAC es un sistema donde el profesor puede ingresar, modificar, consultar y eliminar la información de anuncios, políticas, tareas, cronogramas y calificaciones. Pero, el estudiante, en la mayoría de los casos sólo puede consultar la información y únicamente interactúa con el sistema publicando las respuestas a las tareas que éste envía. Esto representa un nivel de interacción muy bajo para el estudiante y por lo tanto afecta a su motivación para el uso del sistema.

Las limitaciones que presenta SAR, son las siguientes:

- SAR sirve como medio para recopilar las solicitudes de reservación de los recursos. Sin embargo, los recursos no se asignan de una manera automática en base, a ninguna política preestablecida por lo cual es necesaria la constante acción de un administrador que apruebe, postergue o rechace las solicitudes.
- Los profesores son los únicos usuarios que pueden reservar los recursos, lo cual impide que otras personas tales como alumnos o ayudantes académicos puedan en algún momento realizar una reservación.

- Como la aprobación de las solicitudes no es automática, los usuarios tienen que estar pendientes y consultar el sistema hasta que la solicitud sea marcada como aprobada o rechazada. Este mecanismo no ayuda a los usuarios que necesiten conocer inmediatamente si un recurso le fue asignado, porque existen pocas probabilidades de que su solicitud sea prontamente aprobada.

1.2. Justificación

En las distintas unidades académicas de la ESPOL, existe la necesidad de mantener una mayor comunicación entre los participantes de un curso para mejorar el proceso de aprendizaje aprovechando todos los recursos que puedan ser puestos a disposición a través de la Internet como complemento a las clases presenciales.

A continuación mencionaremos varios ejemplos que dan una idea más clara del por qué la comunicación entre los participantes de un curso a través de la Internet es un elemento útil para el proceso de aprendizaje y la administración del curso:

- Si por cualquier motivo intempestivamente se suspenden las clases en la universidad o en un curso específico, no hay facilidades para

informar a los alumnos de tal disposición. Ayudaría el disponer de un mecanismo que permita que los profesores comuniquen a sus alumnos que las clases están canceladas y de esta forma evitar el desperdicio de esfuerzos. En general sería útil poder anunciar cualquier cosa que el profesor considere importante, sin depender del dictado o no de una clase.

- Así mismo, si un alumno no puede asistir a una clase en la cual tiene que entregar alguna tarea o proyecto o recibir algún material, sería conveniente que pueda enviar su tarea o proyecto o recibir documentos de una manera organizada a través de la Internet sin utilizar aplicaciones como el correo electrónico que fue diseñado para propósitos diferentes. Sería bueno además que el profesor tuviera una herramienta que automáticamente controle la fecha de entrega de tareas proyectos y que clasifique aquellas que fueron entregadas fuera de plazo.
- Normalmente, el proceso de entrega de calificaciones en la universidad es lento, y frecuentemente no se puede establecer con certeza la fecha en que las calificaciones se harán públicas. Sería bueno la existencia de un mecanismo que le permita al profesor, comunicar las notas a los estudiantes de una manera directa y detallada a través de la Internet, permitiéndole a cada alumno estar informado de sus calificaciones de exámenes, tareas, lecciones, etc.

- Los alumnos pueden obtener el material proporcionado por el profesor tales como tareas, recursos, etc. únicamente asistiendo a las clases. A través de un mecanismo que aproveche la Internet, los alumnos podrán disponer desde cualquier lugar de todo el material que el profesor a puesto a su disposición.

Los miembros de un curso pueden comunicarse bajo el contexto de un curso únicamente en el aula de clases. Sería bueno disponer de un mecanismo en la Internet que permita a los miembros de un curso, participar en discusiones, aunque estas no sean en línea (el momento en que se lee un comentario no tiene que ser el mismo momento en el que se escribió).

Por otro lado, la asignación y administración de los recursos utilizados para la enseñanza en una unidad (aulas, proyectores, datashows, etc.), se llevan a cabo de manera informal y requiriendo que las personas involucradas participen de manera presencial, lo que impide un manejo eficiente de los mismos.

- La asignación de los recursos no se realiza de manera automática, lo cual, hace lento el procedimiento para la reservación del recurso. La solución a este problema sería a través de un mecanismo que permita realizar la asignación de los recursos de manera automática

o semiautomática y de esta manera los usuarios obtendrían la aprobación o negación inmediata de sus solicitudes.

- La asignación de los recursos sólo se realiza de manera presencial. Si existiera un mecanismo para asignar recursos a través de la Internet, los profesores o estudiantes que desearan reservar algún recurso podrían hacerlo sin necesidad de acudir a la universidad.
- No existe un registro de los recursos asignados ni de las personas que los reservan. Esto impide conocer los recursos que están disponibles en el momento de aprobar una nueva reservación o consultar que personas tienen asignado algún recurso. A través de un mecanismo automatizado que permita a los usuarios involucrados ingresar los datos y al administrador llevar el control de las asignaciones de los recursos se conocerá que recurso ya está reservado, quien lo reservó, etc.
- Permitirle al usuario conocer en cualquier momento el estatus de sus reservaciones, además de poder modificarlas o anularlas, en el caso de que sean reservaciones que aún están vigentes.

Por todo lo expuesto anteriormente es evidente la utilidad que representa un sistema integrado que, aprovechando las facilidades de comunicación a través de la Internet y del paradigma del WWW, permita realizar la asignación y reservación y consulta de los recursos

manejados para asistir la enseñanza, y además mantener una interacción efectiva fuera de clase entre los participantes de un curso.

1.3. Usuarios del Sistema

A los usuarios del sistema se los puede clasificar en cuatro perfiles: profesores, alumnos, ayudantes y administradores. Los alumnos representan un 90%, y el resto (profesores, ayudantes y administradores) representan el 10% de los usuarios.

Los usuarios para poder acceder al sistema tienen que estar previamente registrados como miembros de algún curso (profesor, alumno o ayudante) o como administrador.

De los usuarios potenciales del sistema el 92% (alumnos y ayudantes) están entre los 18 y 25 años, el 4% están entre los 25 y 35 años, y el 4% restante tienen edades mayores a los 35 años.

El porcentaje de usuarios de sexo masculino es del 80%, mientras que los usuarios de sexo femenino representan el 20% restante. Es importante mencionar que el sistema es utilizado en su mayoría por los usuarios con edades comprendidas entre los 18 y 35 años, lo cual

corresponde al 96% de usuarios, mientras que el 4% restante corresponde a los profesores con edades superiores a los 45 años.

A continuación mencionamos la definición de los perfiles que pueden existir en el sistema:

- **Administrador General.** Se encargará de definir los parámetros generales del sistema (tales como las fechas de inicio y fin del período académico), definir quienes serán los administradores de unidad, definir las materias, profesores, alumnos, paralelos, integrantes de los cursos, etc.
- **Administrador de Unidad.** Mantiene actualizada la información de las materias, profesores, alumnos, paralelos, integrantes del curso, pero solo de la unidad a la que esta asignado.
- **Profesor.** Registra toda la información necesaria para sus estudiantes, además de definir políticas de calificación y notas de curso. También puede agregar o eliminar alumnos en su curso.
- **Ayudante.** Persona que podrá realizar las mismas tareas que el profesor excepto las que definen las políticas de calificación, registran las notas y agregan o eliminan alumnos al curso.

- **Alumno.** Puede acceder a toda la información publicada por el profesor o ayudante de su curso. Además, enviar las respuestas a las tareas solicitadas.

El sistema puede ser utilizado por cualquier institución educativa, pero no todos los perfiles de los usuarios existirán en cada una de ellas. Por ejemplo:

- Si el sistema es utilizado por universidades e institutos existirán los perfiles de: profesor, alumno, ayudante, administrador de unidad y administrador general.
- Si el sistema es utilizado por escuelas y colegios existirán los perfiles de: profesor, alumno y administrador general.

CAPÍTULO 2

2. DESARROLLO DE APLICACIONES USANDO EL WWW

El crecimiento de la comunicación y el manejo de información en el Word Wide Web (WWW) o también conocido como el Web es cada vez más sorprendente. Es por este motivo, que el desarrollo de aplicaciones en el Web se ha incrementado en gran manera.

Este capítulo introduce los conceptos, elementos y procedimientos que son indispensables para el desarrollo de estas aplicaciones.

2.1. Desarrollo de Sitios Web

La creación de Sitios Web (Web Sites) ha aumentado por la evolución de la tecnología y de las herramientas para su desarrollo. Esto se debe a que estas aplicaciones son cada vez más competitivas frente a las aplicaciones tradicionales.

Hay que mencionar que se puede desarrollar Sitios Web o Aplicaciones Web (Web Application), pero existe una diferencia entre ellas, generalmente los Sitios Web se utilizan como medios informativos, donde el usuario puede interactuar con el sistema pero no afectar el estado de la información del mismo (por ejemplo no puede modificar la información de la base de datos), mientras que las Aplicaciones Web le permiten al usuario efectuar todo tipo de transacción sobre ella.

Antes de empezar con el desarrollo de un Sitio Web, hay que tener bien definidas las necesidades de los usuarios, ya que dependiendo de éstas, se puede desarrollar un Sitio Web estático o un Sitio Web dinámico. Las diferencias entre estos dos tipos se explicarán más adelante en la sección 2.1.2.

Las consideraciones que se debe tener en cuenta para el desarrollo de un Sitio Web, las detallamos a continuación:

- La plataforma o sistema operativo en el cual va a residir.
- Las herramientas de desarrollo.
- El lenguaje a utilizar.

- La interacción que provee el sitio al usuario. Es decir, si es: presentable, interesante, interactivo, divertido y fácil de usar.

2.1.1. El modelo CLIENTE/SERVIDOR

Cliente/Servidor es un modelo de uso universal que se basa en la existencia de dos elementos: un lado cliente, que es el que por lo general realiza los requerimientos y un lado servidor, que los atiende. Una de las razones de la existencia de esta arquitectura es que contribuye a balancear la carga de procesamiento que antiguamente soportaban los servidores solamente, pasándolas al lado del cliente, esto es, los clientes son un poco más "inteligentes" y capaces de realizar tareas sin necesidad de sobrecargar al servidor.

La principal idea del modelo Cliente/Servidor es definir un sistema como un conjunto de procesos cooperativos (servidores) que ofrecen servicios a los procesos de los usuarios (clientes). En el modelo Cliente/Servidor la comunicación entre procesos se da exclusivamente a través de mensajes.

La arquitectura del Word Wide Web posee todos los requisitos necesarios para ser catalogada como un modelo Cliente/Servidor. Lo cual es mostrado en la Figura 2.1.

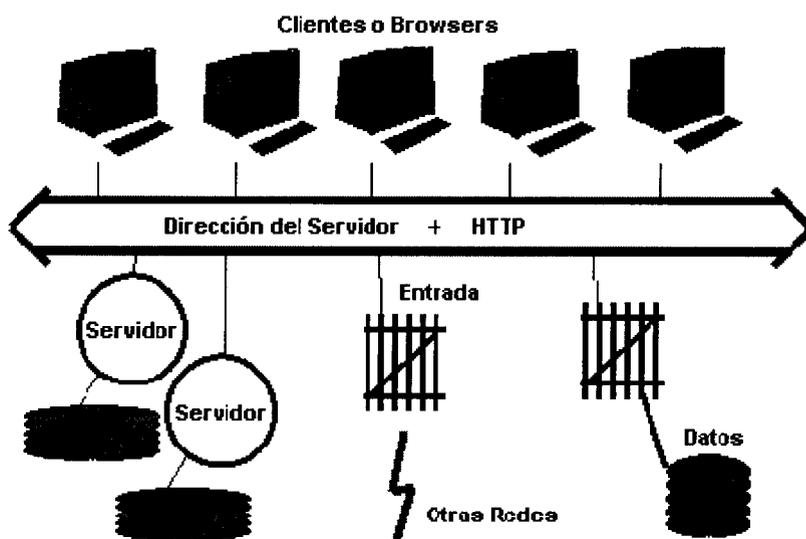


Figura 2.1 Modelo Cliente/Servidor del WWW

El modelo Cliente/Servidor WWW se compone de diferentes partes: por un lado están los usuarios usando los browser o navegadores para acceder a cierta información, y por otro están los programas servidores Web, quienes ofrecen la información en diferentes formatos. Enlazando uno con otro se encuentra el protocolo HTTP,

que es el lenguaje con el que los clientes y servidores se entienden entre sí a través de una o varias redes TCP/IP¹.

El usuario ve una interfaz agradable que es el lado del cliente, y para él es transparente la existencia del protocolo HTTP. Además, sólo conocerá que existe un servidor Web al tener que poner la dirección de tal servidor.

Cuando un browser inicia un requerimiento a un servidor, ocurre lo siguiente: El requerimiento es creado usando el protocolo HTTP. Este requerimiento llegará hasta el servidor (el cual puede estar físicamente en la misma máquina que el cliente o en una máquina diferente), el cual lo interpretará y cumplirá con la acción indicada, que puede ser el enviar de regreso una hoja de hipertexto² o ejecutar un programa CGI³, entre otros.

¹ **TCP/IP (Transmisión Control Protocol / Internet Protocol)** es el protocolo de comunicación utilizado para transmitir datos a través de Internet. El protocolo TCP recoge la información a transmitir y la divide en paquetes TCP, luego el protocolo IP divide los paquetes TCP en subpaquetes IP y les agrega una cabecera con la dirección destino.

² **Hipertexto** es un documento que combina texto con enlaces a otros documentos que pueden ser accedidos en cualquier momento. En Internet el protocolo estándar para hipertexto e hipermedios es el HTML.

³ **CGI (Common Gateway Interface)** es un estándar para el desarrollo de programas que se comunican con el programa servidor Web del lado del servidor.

Generalmente el modelo Cliente/Servidor hace uso de protocolos de comunicación simples de tipo requerimiento/respuesta. A fin de obtener un servicio, un cliente envía un requerimiento al servidor. Este, a su vez, ejecuta las operaciones asociadas al servicio y envía una respuesta al cliente, que contiene datos o un código de error en caso de que el servicio no pueda ser ejecutado.

La mayoría de los programas de aplicación basados en la arquitectura TCP/IP, se basan en el modelo Cliente/Servidor.

2.1.1.1. Cliente Web o Browser

En la arquitectura Cliente/Servidor, los navegadores o browsers son tan importantes como los servidores Web. Para el usuario, el browser será una de las interfaces de entrada al mundo del WWW (también puede acceder a los servidores Web vía correo electrónico) que le permitirá visualizar los homes pages e interactuar con las aplicaciones del servidor.

El cliente Web o Browser, es el programa de aplicación que permite a los usuarios conectarse al sistema. Un cliente Web se comunica con el servidor Web para solicitar la información que necesita. El servidor Web responde al requerimiento del cliente enviándole los datos de la

información solicitada y el cliente Web interpreta ésta información para presentarla al usuario en un formato que pueda entender (página Web).

El cliente Web utiliza un URL (Universal Resource Locator) para enviar los requerimientos al servidor. El URL es un estándar universal que sirve para especificar la ruta de un archivo en una computadora en particular que se desea explorar o ejecutar.

Existen diversos programas de aplicación de clientes Web, llamados browsers. Entre los más conocidos están Netscape Communicator y Microsoft Internet Explorer. También están los browsers Mosaic y Lynx, pero son menos utilizados.

En esencia, todos los navegadores se basan en estándares, pero si existen pequeñas diferencias entre ellos, sea en ciertos detalles de la apariencia de las páginas como en la versión de HTML que soportan.

HTML (HyperText Mark-up Language) es una colección de comandos de formatos para crear documentos hipertextos (páginas Web), los cuales están basados en el uso de tags¹. Cuando

¹ **TAGS (etiquetas)** son las órdenes del lenguaje HTML que se utilizan para el diseño de páginas Web.

accedemos a una página Web, el browser interpreta los comandos HTML embebidos en la página y los usa para crear el despliegue visual del contenido de la misma. Los comandos HTML cubren muchos tipos de formatos y tienen la capacidad de incluir gráficos y otros elementos no-texto.

En pocas palabras, HTML es un lenguaje usado para crear páginas Web, pero los browsers del Web muestran estas páginas de acuerdo a la versión de HTML que poseen. Por este motivo, cuando se implementen aplicaciones en el Web, es importante que sean compatibles con los browsers más utilizados.

2.1.1.2. Servidor Web

Un servidor Web es una computadora que está conectada a Internet en forma permanente. Dicha computadora tiene almacenado en su disco duro las aplicaciones Web o sitios Web.

Los servidores Web a través de programas servidores interpretan código basado en scripts¹, proporcionan acceso a documentos

¹ Conjunto de instrucciones en un lenguaje determinado, los mismos que pueden ser interpretados y ejecutar una función específica.

distribuidos de hipertexto, aplicaciones y base de datos, donde las aplicaciones o bases de datos pueden ser externas al servidor.

Los servidores Web operan bajo un paradigma requerimiento - respuesta, en donde un cliente Web establece una conexión con un servidor y le envía un requerimiento. El servidor interpreta el requerimiento, si no existen problemas envía como respuesta el contenido de una página Web, caso contrario envía un mensaje de error.

El servidor Web más popular actualmente es el Apache Web Server. Pero existen otros servidores Web muy conocidos como: Internet Information Server (IIS), Netscape Application Server (NAS), Microsoft Personal Web Server, Microsoft Frontpage Web Server, Java Web Server e IBM Websphere.

2.1.2. Generación de páginas HTML

Un servidor Web responde a la petición de un explorador de Web devolviendo una página HTML. La página devuelta puede ser de tres tipos: una página HTML estática, una página HTML dinámica cliente o una página dinámica servidor.

2.1.2.1. Páginas estáticas

Las páginas estáticas son páginas HTML fijas, existentes en el servidor. El servidor Web devuelve las páginas HTML al usuario, pero no realiza ninguna acción especial. Para obtener una página de este tipo se realiza lo siguiente:

- a) El usuario pide una página estática escribiendo una dirección URL (<http://www.organización.com/home.htm> en la Figura 2.2) o haciendo click en un vínculo que apunte a una dirección URL.
- b) La petición de dirección URL se envía al servidor y éste responde devolviendo la página HTML estática.

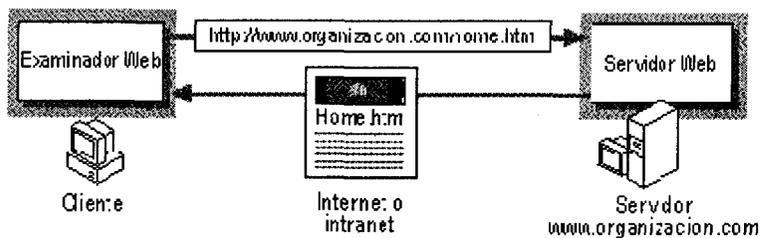


Figura 2.2 Generación de páginas estáticas

2.1.2.2. Páginas dinámicas cliente

Las páginas dinámicas cliente son páginas que no se generan en base, a programas existentes en el servidor, sino que, ya existen en el servidor como páginas HTML, las mismas que en su contenido incluyen: código en lenguajes interpretados, referencias a otros archivos que contienen ese código o referencias a componentes que le dan un comportamiento en especial. Esto le permite a la página interactuar con el usuario en el browser y poseer los siguientes tipos de dinamismo:

- a) Las páginas pueden contener animaciones, como por ejemplo: menús flotantes, mover elementos, desplazar elementos, mover áreas o gráficos animados.
- b) Las páginas también pueden realizar validaciones de ingreso de datos o para realizar operaciones.

Todo esto se consigue a través de lenguajes de programación interpretados como: JavaScript (desarrollado por Netscape, sintaxis similar a la de Java), VBScript (desarrollado por Microsoft, sintaxis similar a la de Visual Basic) o por medio de componentes tales como: Applets (escritos en Java), Component Object Model (COM) (escritos

en Visual Basic, Visual J, Visual C++). Los mismos que se ejecutan del lado del cliente.

2.1.2.3. Páginas dinámicas servidor

Las páginas dinámicas servidor se generan como respuesta a los requerimientos de los usuarios a través de los browsers en tiempo de ejecución (no son páginas HTML existentes en el servidor Web).

Un explorador de Web recopila información presentando una página que incluye sentencias HTML para el manejo de formularios. Estas sentencias permiten contar con cuadros de ingreso de texto, menús y casillas de verificación que el usuario completa o activa. Cuando el usuario hace click en un botón del formulario, los datos procedentes del formulario se envían al servidor Web. El servidor pasa los datos a un archivo de comandos o una aplicación para su procesamiento, los ejecuta y devuelve los resultados al usuario en forma de una página HTML. Para realizar esta operación existen dos maneras:

- 1) Embeber el código HTML en la aplicación o programas que se ejecutan para la generación de la página. Este mecanismo dificulta el desarrollo, implementación y mantenimiento de las páginas HTML.

- 2) Separar la lógica de programación del diseño de la página a través del uso de componentes que son llamados por la propia página en el lado del servidor. Esto facilita y agiliza las tareas involucradas en el desarrollo de una aplicación o sitio Web.

Existen varias tecnologías para el desarrollo de sitios Web o aplicaciones Web dinámicas, las cuales mencionamos a continuación:

- Mediante tecnologías CGI e ISAPI que desarrollan aplicaciones con código interpretado/compilado basado en lenguajes tales como: C, C++, Perl, Python, Delphi, Visual Basic, Power Builder y Java.

CGI (Common Gateway Interface) es un mecanismo que permite a un servidor de Web ejecutar un programa o script sobre el servidor y enviarlo como salida a un Web browser. Este tema trataremos en más detalle en la siguiente sección.

ISAPI (Internet Server Application Programming Interface) es un conjunto de funciones para Servidores de Internet.

- **ASP** (Active Server Pages) son un componente del software del servidor Web de Microsoft que le permite incrustar código de automatización del lado del servidor en páginas Web

(generalmente en VBScript). Evita los temas asociados a la descarga de applets u otros controles del lado del cliente.

- **PHP** (Hypertext Preprocessor) es una tecnología basada en scripts embebidos en páginas HTML y ejecutados en el servidor. La mayoría de los scripts se desarrollan en C, aunque también se utiliza Java y Perl. Es ideal para trabajar sobre la plataforma UNIX.
- **Servlets** son una mejora a los applets, ya que éstos interactúan con el servidor Web o con servidores Web externos. Los servlets son aplicaciones que se ejecutan en el servidor Web para responder a los requerimientos de los clientes.
- **JSP** (JavaServer Pages) combina elementos de lenguaje HTML y elementos creados dinámicamente mediante objetos del lenguaje Java. Los programas JSP llamados páginas JSP se ejecutan como Servlets Java y generan documentos usando datos almacenados en bases de datos y en otros objetos de aplicación.

En la siguiente figura (Figura 2.3) se observa el proceso de generación de las páginas dinámicas servidor.

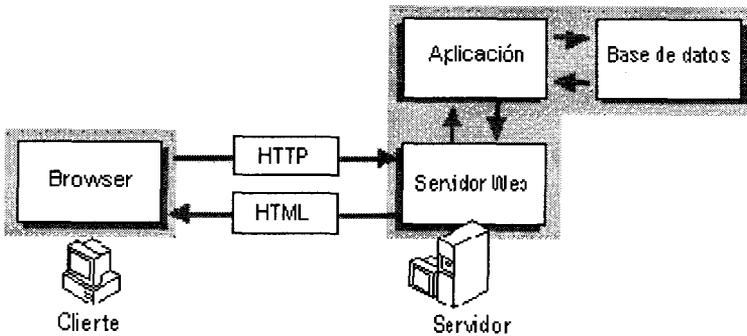


Figura 2.3 Generación de páginas dinámicas servidor

2.1.3. El modelo THREE-TIER

La arquitectura de software Three-Tier (tres capas) emergió en los 90s para superar las limitaciones de la arquitectura Two-Tier¹. La nueva capa (servidor intermedio) está entre la interfaz de usuario (cliente) y el servidor final (generalmente encargado de la administración de los datos). Esta capa intermedia provee la administración de procesos donde la lógica y reglas del negocio son ejecutadas y pueden acomodar a centenares de usuarios al proporcionar funciones de consultas, ejecución de aplicación y presentación de información.

¹ **Two-Tier.** Estas aplicaciones conocidas como aplicaciones Cliente/Servidor, dividen una aplicación en un cliente (solicitador de servicios) y un servidor (proveedor de servicios) estableciendo un middleware que controla las comunicaciones entre ambos. La arquitectura de software Two-Tier es usada principalmente en procesamiento de información no crítica, donde la administración y operaciones del sistema no son complejas.

La arquitectura Three-Tier es usada cuando es necesario que un diseño Cliente/Servidor distribuido provea efectivamente mayor rendimiento, flexibilidad, mantenimiento, reusabilidad y escalabilidad, mientras se esconde la complejidad del procesamiento distribuido. Estas características han convertido a la arquitectura de tres capas en una opción popular para aplicaciones Internet y sistemas de información centrados en la red. El modelo Three-Tier se representa en la Figura 2.4.

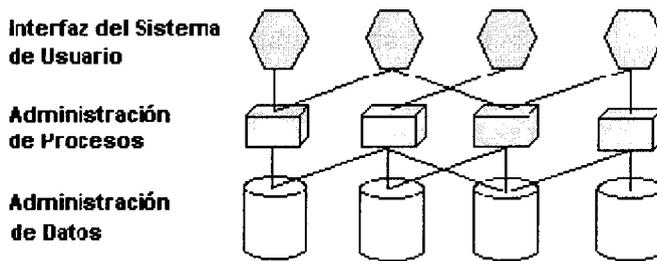


Figura 2.4 Capas del modelo Three-Tier

La capa superior de la arquitectura Three-Tier como se presenta en la Figura 2.4, es la interfaz del sistema de usuario donde los servicios del usuario (tales como la administración de sesión, texto de entrada, cajas de diálogo y despliegue) residen.

La capa inferior provee generalmente una funcionalidad en la administración de los datos y archivos (podría también tener otras

funcionalidades dependiendo del campo de aplicación del sistema). Los componentes de administración de datos garantizan que éstos estén consistentes en el ambiente distribuido a través del control del bloqueo de datos, la consistencia y la replicación. Debe notarse que la conectividad entre las capas puede ser dinámicamente cambiada dependiendo del requerimiento de usuario para los datos y servicios.

La capa intermedia provee servicios de administración de procesos (tales como desarrollo, ajuste, monitoreo y recursos de procesos) que son compartidos por múltiples aplicaciones.

El servidor de la capa media (también referida como el servidor de aplicación) mejora el rendimiento, flexibilidad, mantenimiento, reusabilidad y escalabilidad por la centralización de la lógica de procesos. La lógica de procesos centralizados hace la administración y cambios más fáciles por la localización de la funcionalidad de sistemas de tal manera que los cambios deben sólo ser escritos una vez y localizados en el servidor de la capa intermedia para que estén disponibles a todos los sistemas.

Adicionalmente, la capa de administración de procesos controla las transacciones y colas asincrónicas para asegurar la terminación

confiable de transacciones. La distribución del modelo se muestra en la Figura 2.5.

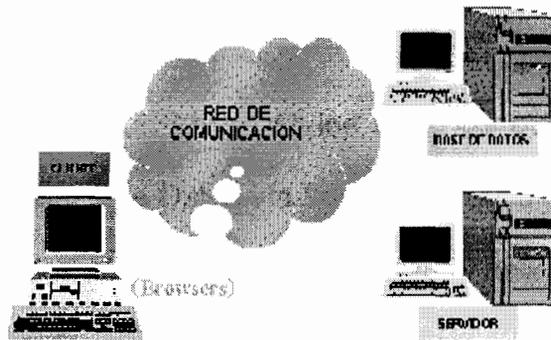


Figura 2.5 Modelo Three-Tier

2.1.4. Bases de Datos y su uso en Internet

En el pasado, las bases de datos sólo podían utilizarse al interior de las instituciones o en redes locales, pero actualmente el Web permite acceder a bases de datos desde cualquier parte del mundo. Estas ofrecen, a través de la red, un manejo dinámico y una gran flexibilidad de los datos, como ventajas que no podrían obtenerse a través de otro medio informativo.

Los usuarios de Internet pueden obtener un medio que puede adecuarse a sus necesidades de información, con un costo, inversión de tiempo, y recursos mínimos. Así mismo, las bases de datos serán

usadas para permitir el acceso y manejo de la variada información que se encuentra a lo largo de la red.

El Web ofrece una gran flexibilidad en el acceso de los datos. Utilizando el modelo Three-Tier los usuarios pueden acceder a los datos de una Base de Datos vía Internet a través de un Servidor Web, para lo cual ellos sólo necesitan un browser y una conexión a Internet.

2.2. Plataformas y herramientas para el desarrollo de Sitios Web

Como mencionamos en el capítulo anterior en la sección Desarrollo de Aplicaciones WWW, hay que tener en cuenta muchos factores para iniciar el desarrollo de una aplicación o sitio Web, los cuales deberían estar basados en los objetivos de la aplicación y en la funcionalidad que se desea tener. En esta sección explicaremos en detalle cada uno de los elementos necesarios para el desarrollo de aplicaciones en el Web, tales como: la plataforma, el lenguaje de desarrollo de la aplicación, el manejador de base de datos y las herramientas de desarrollo a utilizar para la generación de las páginas Web.

2.2.1. Servidores

A continuación se listan los servicios más populares que ofrece la Internet, y se describe cada uno de ellos:

- **Servidor Telnet/WAIS:** Permite al usuario remoto conectarse a un computador y usar los servicios como conexión local.
- **Servidor de Correo:** Sistema de mensajes que permite intercambiar correspondencia electrónica en la red o Internet.
- **Servidor de Listas:** Permite crear y atender listas de correo y responder automáticamente a usuarios, clientes o afiliados.
- **Servidor de Noticias:** Permite establecer foros de discusión y distribuir noticias al grupo.
- **Servidor FTP/Gopher:** Permiten mover archivos en la red o crear un servicio gopher para atender grandes cantidades de información.
- **Servidor de Fax:** Entrega y recibe facsímiles en demanda y por lotes.
- **Servidor Chat:** Permite a los clientes participar en conversaciones en tiempo real usando texto y audio.

- **Servidor Groupware:** Permite trabajar en grupo en una oficina virtual, sin importar la ubicación.
- **Servidor IRC:** Capacidad de participar en tiempo real en discusiones basadas en texto usando un servidor "Internet Relay Chat".
- **Servidor Web:** Hospeda Sitios Web y publica las páginas a quien las requiera. Este es uno de los más importantes en el desarrollo de aplicaciones en el Web y fue explicado en detalle en la sección 2.1.1.2.
- **Servidor Proxy - Firewall:** Permite acceso a la Internet compartiendo un enlace y al mismo tiempo protege los recursos del computador.
- **Servidor de Audio/Video:** Entrega contenido multimedia en protocolo de flujo.

2.2.2. Common Gateway Interface (CGI)

Como ya hemos mencionado anteriormente, una de las maneras para desarrollar aplicaciones dinámicas en el Web, es a través de programas o scripts implementados en lenguajes interpretados, los

cuales se ejecutan en el servidor Web. Esto es más conocido como CGI, que define una manera estándar de intercambiar datos entre el servidor Web y el programa.

Las reglas que rigen este estándar suelen ser dependientes del sistema operativo, por ejemplo: UNIX utiliza variables de ambiente del proceso, Windows NT usa los archivos temporales del proceso de la aplicación, Mac utiliza AppleEvents, etc.

A los scripts o programas que utilizan el estándar CGI se los denomina programas CGI, o simplemente CGIs. También se les llama gateway porque actúan como especie de "compuerta" o intermediario entre el servicio Web y otro servicio especializado, como por ejemplo una base de datos.

El requerimiento de una página HTML al servidor Web, implica que esta página exista de manera estática, es decir que su contenido no cambie en el momento de generar la respuesta al cliente, porque es un archivo que ya está creado y disponible con anterioridad. Mientras que un requerimiento de ejecutar un programa CGI al mismo servidor Web permite crear respuestas dinámicamente en el momento en que son requeridas, pues estos CGIs son ejecutados en tiempo real. Por ejemplo, si se trata de consultar información almacenada en una base

de datos, se pueden crear páginas HTML que contengan información extraída desde la Base de Datos; es decir que la respuesta que se envía al cliente no está almacenada como un archivo estático en alguna parte, sino que se genera en el momento en que se ejecuta el programa.

La información que los programas necesitan recibir del cliente para generar los datos requeridos, usualmente se recolectan a través de los formularios de las páginas HTML.

Antes de que se ejecute un programa CGI, el servidor Web realiza lo siguiente:

1. Determina si el requerimiento es un programa o un archivo estático.
2. Si se trata de un programa o script, el servidor Web lo localiza en el sistema de archivos y averigua si le está permitido ejecutarse.
3. Inicia el ambiente necesario para ejecutar el script y luego le pasa los datos de entrada que recibió desde el cliente. El servidor Web debe leer la salida del script y enviarla al cliente.
4. Debe enviar mensajes de error al cliente Web cuando ocurra un error.

2.2.2.1. Lenguajes para la implementación de CGIs

Antes de comenzar a escribir el programa CGI, se necesita escoger el lenguaje de programación a utilizar. Un programa CGI puede ser escrito en cualquier compilador/interpretador que permita crear programas que puedan leer el dispositivo STDIN (siglas de Standard Input, o entrada de datos estándar), escribir en el dispositivo STDOUT (siglas de Standard Output, o salida de datos estándar) y leer variables del ambiente. Se debe escoger un lenguaje de programación que sea familiar, pero a su vez que sea una buena elección para el sistema en que se correrá, y que pueda ejecutar las operaciones para las cuales fue escrito el programa.

Muchos de los programas CGIs están escritos en AppletScript, C, C++, Perl, Python, TCL, cualquier shell de UNIX o Visual Basic. Estos no son los únicos lenguajes a utilizar, pero son los más comunes. Mientras se pueda escoger cualquier lenguaje que se desee, existen dos buenas razones para considerar alguno de los más comunes.

- Será más rápido elaborar programas CGI con rutinas disponibles. Esto es debido a que muchos programadores del Web han escrito

programas CGI para tareas comunes y sus programas están disponibles gratuitamente a través de la Internet.

- Si se utilizan uno de los lenguajes comunes de programación CGI, es más fácil obtener ayuda para su depuración. Se pueden enviar las preguntas y leer las respuestas en grupos de discusión o listas de correo electrónico. Esto es especialmente válido cuando no hay facilidades ni recursos para aprender estos lenguajes debido a que no existen empresas que den un buen soporte y/o capacitación.

2.2.2.2. ¿Cuándo usar un CGI?

Las aplicaciones basadas en el WWW pueden correr en el servidor Web, en el cliente Web o en ambos. Uno decide si se corre el CGI del lado del servidor o un script del lado del cliente. Se debería encontrar las ventajas y desventajas para cada opción.

- Actualizaciones dinámicas y características de interactividad, tales como validación de la entrada de un dato en un campo, son mejor hechas con un lenguaje de script del lado del cliente como JavaScript, el cual da una entrega inmediata al usuario. Los JavaScript, al igual que los Applets y otras herramientas

soportadas por los browsers, han ayudado a los desarrolladores de aplicaciones basadas en el Web, a enviar junto con los datos contenidos en las hojas HTML, componentes de código que pueden ser ejecutados del lado del cliente, sobre todo para tener un mayor control sobre la interactividad de la aplicación.

- Sin embargo, un script del lado del cliente corre sólo en la máquina del usuario. Ninguna información puede pasar desde el cliente al servidor de Web fuera de la sesión HTTP, lo cual es necesario si se requiere almacenar información en un servidor de bases de datos o ejecutar acciones en el lado del servidor (que generen automáticamente el envío de un mail, por ejemplo).
- La seguridad toma un rol importante en el momento de decidir ejecutar aplicaciones con un script del lado del cliente o un CGI del lado del servidor. Cuando se ejecutan aplicaciones en un Web Browser el servidor está seguro, porque las aplicaciones clientes no pueden acceder a los recursos del servidor.
- Otro aspecto importante es el lugar donde se coloca el CGI en el servidor. Con un CGI del lado del servidor, cada nuevo requerimiento para el programa CGI añade carga de procesamiento al servidor, lo cual puede resultar en tiempos de respuesta lentos si su sitio tiene un alto tráfico. Los scripts del

lado del cliente obviamente no tienen estos problemas pero su uso está fundamentalmente orientado hacia la apariencia de la información presentada.

2.2.3. Bases de Datos

Existen manejadores de bases de datos, tales como: Access, Oracle, DB2, PostgreSQL, Microsoft SQL Server, MySQL, FoxPro, Informix, dBase, Ingres, Paradox, Sybase, etc. En esta sección revisaremos los más comunes que pueden ser utilizados para el desarrollo de aplicaciones en el Web.

Microsoft Access 2000

Access es una base de datos fácil de usar y a bajo costo, pero no es útil como una base de datos de altas prestaciones. Sin embargo, es una solución confiable para pequeñas cantidades de datos y pocos usuarios concurrentes. Access es una buena solución para oficinas que tengan el paquete de Microsoft Office estandarizado, pero los usuarios de Linux o Macintosh no pueden hacer uso de esta base.

MySQL

Es una robusta base de datos Cliente/Servidor que soporta un amplio rango de plataformas, MySQL corre bajo Linux, Solaris, OS/2, HPUX, AIX, y la mayoría de los sistemas UNIX, además de trabajar bajo la plataforma MS Windows. La versión para los sistemas UNIX es gratuita, mientras que la versión para Windows tiene costo. Algunos pueden cuestionar la selección de una base de datos freeware¹, pero MySQL es una excelente opción para proyectos con un pequeño presupuesto. Una desventaja de MySQL es el soporte técnico limitado

IBM DB2

Es otra poderosa solución de bases de datos Cliente/Servidor, DB2 ofrece buen rendimiento, corre en Windows y en la mayoría de sistemas UNIX. Muchas aplicaciones de terceras partes que añaden funcionalidad extra a DB2 están disponibles, y posee una gran comunidad de usuarios que la usan. Pero una desventaja es el costo del manejador, del servicio y soporte técnico que provee IBM.

¹ **Freeware.** Software de distribución y utilización libre.

Microsoft SQL Server 7.07

Es la solución de bases de datos de nivel empresarial de Microsoft más popular, esta base de datos Cliente/Servidor está ganando popularidad rápidamente y ha sido adoptada por algunas grandes compañías como DELL, Nasdaq y Monster.com. Es muy poderosa y fácil de usar. Un inconveniente es que sólo corre sobre MS Windows. Es así, que se debe estar corriendo un sistema operativo Microsoft para poder usarla. SQL Server soporta grandes cantidades de datos por base, además se integra directamente con las cuentas de usuarios y seguridad de Windows NT y Windows 2000.

Oracle 8i

Los gigantes de Internet como Yahoo usan Oracle. Generalmente se la estima como la base de datos más poderosa y segura actualmente disponible. Oracle corre bajo Windows y la mayoría de las variantes de UNIX, y soporta medidas de bases de datos ilimitadas. Pero una de sus mayores desventajas es el elevado costo que posee, además de los costos de capacitación y soporte.

PostgreSQL

PostgreSQL (o simplemente Postgres) funciona bajo los sistemas operativo UNIX, Linux y Windows, y es una alternativa muy utilizada en aplicaciones Web, debido a que es una herramienta que posee el código abierto (open source), y no tiene costo alguno. Es una base de datos robusta y eficiente pero no es muy recomendada en las aplicaciones Web donde las operaciones sobre los datos sean muy complejas. Una ventaja que posee este manejador es que la información y soporte necesario se encuentra en disponible en la Internet.

2.2.4. Sistemas Operativos

Al escoger las herramientas y lenguajes de programación a utilizar, se debe de tomar en consideración la plataforma o sistema operativo sobre el cual correrá la aplicación.

En la siguiente tabla se clasifica a los sistemas operativos en cuatro grandes grupos:

Unix comercial	Unix Source Open	Windows	Mac y otros
A/UX	BSDi	CE	MacOS
Compaq Tru64	FreeBSD	95/98	MacOS X
HP-UX	Linux	Me	AS/400
IBM AIX	OpenBDS	NT 4.0	BeOS
Open VMS	Solaris	2000	OS2
OSF/1		OS/390	XP
QNX		Novell Netware	
SCO OpenServer		Wind River	
Solaris		Vx Works	
SunOs		Lynx	
UnixWare		Milenium	

Tabla I Sistemas Operativos

A continuación definimos en forma breve a tres de los sistemas operativos más utilizados:

WINDOWS: Sistema operativo creado por la compañía Microsoft. Está orientado a las PCs y es el más utilizado en el mundo. Ha pasado por diferentes versiones tales como: 3.1, 3.11, 95, 98, NT, 2000, Milenium.

UNIX: Es un sistema operativo utilizado sobre todo en computadoras grandes. Se caracteriza por ser multiusuario (atiende muchos usuarios al mismo tiempo) y multitarea (realiza muchas tareas al mismo tiempo). La Internet fue desarrollada inicialmente en sistemas UNIX.

LINUX: Es un sistema operativo diseñado originalmente por el finlandés Linus Torvalds. Esencialmente nació como una versión de UNIX para PCs pero ahora se utiliza en muchos otros tipos de computadoras. Actualmente es el sistema operativo de mayor crecimiento en el mundo, y una de sus características principales es que no pertenece a ninguna empresa, sino que es producto de la colaboración de decenas de miles de programadores a través de la Internet.

Aunque existen muchos lenguajes y herramientas comunes que están disponibles para la mayoría de las plataformas para las cuales existen servidores Web, es importante considerar las diferencias que puedan existir entre estas y su integración con funcionalidades específicas del Sistema Operativo.

2.2.5. Herramientas para el desarrollo de páginas Web

Para el desarrollo de páginas Web existe una variedad muy grande de herramientas que facilitan el trabajo a los integrantes de un equipo de desarrollo (programador, diseñador, ingeniero de sistemas, soporte técnico, etc.).

Podemos clasificar las herramientas en varios grupos, de acuerdo a la función que desempeñan:

Editores de páginas Web: Son utilizadas principalmente por los programadores, puesto que generan código HTML conforme se vayan incluyendo partes visuales en una página (tablas, entry text, radio buttons, check boxes, etc.).

- Note Tab Light (<http://www.notetab.com>)
- Freeware, con asistentes para crear rápidamente código HTML, CSS, entre otros
- Home Site (<http://www.allaire.com/products/HomeSite>)
- Editores de Microsoft (<http://www.microsoft.com>)
- Front Page 98, 2000 (<http://www.microsoft.com/frontpage>)
- Visual InterDev 6
- Top Style Lite. Provisto en conjunto con Home Site, permite crear dinámicamente archivos de estilo (CSS)
- HotDog Profesional 4 (<http://www.sausage.com>)
- HotMetal Pro 4 (<http://www.softquad.com>)

- AOLpress 2.0 (<http://aolpress.com>)
- HTMLtool 1.6 (<http://intew.com/home/htmltool>)
- Arachophilia (<http://www.arachnoid.com>)

Herramientas para el manejo gráfico de páginas Web: Permiten manipular y optimizar las imágenes que formarán parte de la página. Estas herramientas son utilizadas principalmente por los diseñadores de páginas Web.

- Microsoft Image Composer
- Adobe Illustrator
- Macromedia Dreamweaver 4
- Macromedia Fireworks 4
- Macromedia UltraDev 4
- Macromedia ShockWave
- CorelDraw
- Flash
- CorelPhotoPaint

- Adobe Photoshop

Herramientas para el manejo de las bases de datos: Proveen facilidades para diseñar las estructuras de las tablas y sus relaciones en una base de datos. A partir de un diseño gráfico, se puede generar archivos con los comandos que generan los diferentes componentes de una base de datos (tablas, integridad referencial, triggers, store procedures, índices, etc.) Estas herramientas son utilizadas principalmente por los administradores de bases de datos.

- Erwin

Herramientas orientadas a objetos: Proveen metodologías que permiten analizar y diseñar aplicaciones orientadas a objetos. Son herramientas CASE que permiten generar código fuente a partir de un diseño de aplicación.

- Rational Rose
- UML

2.3. Seguridad de Sistemas basados en el WWW

El esquema de seguridad que posean las aplicaciones en el Web, es un punto muy importante en su funcionamiento, ya que la aplicación debe garantizar que los usuarios trabajarán sobre un medio seguro y confiable, impidiendo que personas ajenas en la red puedan acceder a la información de forma no autorizada.

2.3.1. Servicios de seguridad

La seguridad en la Internet tiene varios componentes, los cuales se clasifican como servicios de seguridad:

Confidencialidad: requiere que la información sea accesible únicamente por las entidades autorizadas. La confidencialidad de datos se aplica a todos o porciones de los datos intercambiados por las entidades autorizadas.

Autenticación: requiere una identificación correcta del origen del mensaje, asegurando que la entidad no es falsa. Se distinguen dos tipos de entidad, que asegura la identidad de las entidades participantes en la comunicación, mediante biométrica (huellas

dactilares, identificación de iris, etc.), contraseñas, etc. y de origen de información.

Integridad: requiere que la información sólo pueda ser modificada por las entidades autorizadas. La modificación incluye escritura, cambio, borrado, creación y actualización de los mensajes transmitidos.

No repudio: ofrece protección a un usuario frente a que otro usuario niegue posteriormente que en realidad se realizó cierta comunicación.

Disponibilidad: requiere que los recursos del sistema informático estén disponibles a las entidades autorizadas cuando los necesiten.

2.3.2. Seguridad usando el protocolo SSL

Netscape ha propuesto un protocolo para proveer una capa de seguridad de los datos entre el nivel de aplicación y TCP/IP. Este protocolo de seguridad, se llama Secure Socket Layer (SSL), el mismo que brinda encriptación de datos, autenticación de servidor, integridad de mensajes y autenticación opcional de usuario para una conexión TCP/IP; da una manera segura de establecer una comunicación con

usuarios. SSL garantiza la autenticidad del contenido Web mientras verifica la identidad de los usuarios que acceden a sitios restringidos.

SSL conforma una capa entre los protocolos de aplicación, como HTTP, SMTP, TELNET, FTP, Gopher y NNTP, y el protocolo de conexión a Internet, TCP/IP. SSL da seguridad para iniciar una conexión TCP/IP.

SSL encripta la información en el pedido y en la respuesta HTTP, incluyendo la URL que el cliente está demandando, cualquier contenido de formulario aceptado (por ejemplo número de tarjetas de crédito), información sobre autorización de accesos (nombre de usuario y clave) y todos los datos del servidor que se envían al usuario.

SSL está incluido en los paquetes Netscape Secure Server y Netscape Communicator. Actualmente se dispone de varias implementaciones para múltiples plataformas y aplicaciones y es utilizado en diferentes productos, especialmente para la comunicación entre browsers y servidores Web.

En resumen, SSL brinda un método de encriptación y desencriptación de la información que viaja por la Web, protegiéndola de los hackers¹.

2.3.3. Autenticación

Como mencionamos anteriormente, la autenticación es un servicio de seguridad que necesita una identificación del cliente para comprobar que es quien dice ser.

Un mecanismo para realizar la autenticación es POP3 (Postal Office Protocol v3) que permite la comunicación por correo electrónico de forma análoga al sistema. Puede servir para saber si un usuario está registrado en el sistema operativo, lo cual se logra utilizando las funciones para el envío de email al usuario del que se desea saber si es válido en el sistema. Si el resultado de este envío es correcto, significa que el usuario existe en el sistema operativo.

¹ Individuo que siente una gran afición por los aspectos relacionados con la informática y la electrónica y que posee conocimientos profundos sobre el funcionamiento de las computadoras. A menudo son contratados por compañías e instituciones para que evalúen las condiciones de seguridad de sus sistemas.

Ningún sistema de autenticación es totalmente confiable, y las claves de un sistema operativo también pueden ser descubiertas, por esta razón los sistemas de detección de intrusión son el complemento perfecto para una buena autenticación, por ejemplo los firewalls¹.

Existen otras técnicas de autenticación, las cuales mencionamos a continuación:

Kerberos: es un sistema de autenticación diseñado para la utilización sobre redes inseguras (por ejemplo, Internet). El sistema Kerberos fue diseñado con dos propósitos, proveer autenticación y distribuir claves. El sistema Kerberos actúa como autoridad de certificación que garantiza una relación correcta entre claves y personas.

Palabras de paso de uso único (one-time password): Una palabra de paso de uso único actúa exactamente como su nombre indica. Se usa una única vez y no vuelve a ser empleada. Esto proporciona una gran seguridad frente a intrusos que utilicen la repetición ciega de palabras de paso (replay).

¹ **Firewall.** Sistema de seguridad insertado entre Internet y una red local de empresa, que sirve de barrera lógica o filtro para evitar las intrusiones.

2.3.4. Cookies

Las cookies constituyen una potente herramienta empleada por los servidores Web para almacenar y recuperar información acerca de sus visitantes. Dado que el protocolo de Transferencia de HiperTexto (HTTP) es un protocolo sin estados (no almacena el estado de la sesión entre peticiones sucesivas), las cookies proporcionan una manera de conservar información entre peticiones del cliente, extendiendo significativamente las capacidades de las aplicaciones Cliente/Servidor basadas en la Web. Mediante el uso de cookies se permite al servidor Web recordar algunos datos concernientes al usuario, como sus preferencias para la visualización de las páginas de ese servidor, nombre y contraseña, productos que más le interesan, etc.

Una cookie es un fichero de texto que algunos servidores piden al navegador que lo almacene en el disco duro, con información acerca de lo que hemos estado haciendo por sus páginas, logrando de esta manera disminuir la carga sobre el servidor.

Los cookies no pueden extraer información de otros cookies pertenecientes a otros sitios, no pueden interactuar con otros datos en el disco duro del usuario. Sólo pueden grabar o re-llamar información.

Antes que cualquier dato pueda ser almacenado en un cookie, el sitio debe obtener tal información, diciéndole al usuario que llene una forma. Los cookies pueden llegar hasta 4K, pero en la práctica pocos exceden. También poseen fecha de caducidad, por lo cual después de un tiempo dejan de ser operativos.

CAPÍTULO 3

3. ALCANCE Y ANALISIS DEL SISTEMA

Este capítulo cubre la etapa de **análisis**, cuyo propósito es lograr una abstracción precisa y concisa de *qué* es lo que el sistema debe hacer, más no el *cómo* esto será hecho. Se examinarán los requerimientos, se analizarán sus implicaciones y se los reiterarán rigurosamente. Todo esto ayudará a entender los requerimientos y el ambiente del mundo real en el cual el sistema existirá, como una preparación para la etapa de diseño.

Cursos Web ha sido concebido para ser implantado utilizando tecnología orientada a objetos. Esto se decidió por las siguientes razones:

1. Para aplicar los conocimientos del paradigma orientado a objetos aprendidos en la carrera y demostrar su aplicabilidad a sistemas basados en el Web.
2. Para demostrar la utilidad de lenguajes y herramientas orientadas a objetos de dominio público para el desarrollo de sistemas reales basados en el Web.

3. Para favorecer su integración en el futuro con otros sistemas o módulos que también aprovechen este paradigma (cuyo incremento en su uso es una tendencia mundial).

Para la construcción del sistema se utilizará la metodología UML (Unified Modeling Language – Lenguaje de Modelamiento Unificado), que es muy utilizada. UML es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. Permite modelar cosas conceptuales como procesos de negocio y funciones de sistema, además de cosas concretas como escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables. La metodología tiene las siguientes etapas: análisis, diseño del sistema, diseño de objetos e implementación.

3.1. Especificaciones del Sistema

En esta sección se especifican los requerimientos generales del problema que deseamos resolver, los cuales están basados en los objetivos que se describen a continuación y que están alineados al propósito de esta tesis.

3.1.1. Objetivos

- Desarrollar una aplicación integrada para el manejo de los recursos académicos de cualquier unidad académica de la ESPOL (aulas, proyectores, etc.) y la comunicación entre los participantes de un curso de dicha unidad (asignación y entrega de tareas, envío de anuncios a los estudiantes, notificación de calificaciones) que esté completamente desarrollada sobre una plataforma orientada a objetos y que su interfaz con el usuario esté basado en un browser de Web.
- Permitir que el profesor pueda publicar: anuncios, políticas del curso, políticas de calificaciones, tareas, calificaciones, calendarios de las clases con material adjunto de cada clase, información general del curso, recursos (como: programas, artículos, etc.), entre otros. Además de proveer material a sus estudiantes, podrá revisar las respuestas de las tareas enviadas por los mismos. Así también, podrá reutilizar el material de cursos pasados.
- Permitir que el alumno pueda acceder a todo el material suministrado por el profesor, enviar las respuestas a las tareas y publicar recursos (programas, artículos, etc.).

- Permitir que los integrantes de un curso (profesor, alumnos y ayudantes) intercambien ideas, preguntas y material académico a través de un foro de discusión.
- Permitir que los usuarios del sistema puedan reservar los recursos para asistencia a la enseñanza de la institución (proyectores, aulas, etc.). La asignación será automática, es decir, en base a políticas establecidas y a la disponibilidad de horarios de los recursos, las solicitudes de reservaciones serán aprobadas o rechazadas inmediatamente por el sistema.
- Permitir al administrador las tareas de mantenimiento (ingreso, modificación y eliminación de la información necesaria para que el sistema funcione correctamente, tales como: profesores, alumnos, paralelos, materias, etc.), configuración de parámetros, asignación de usuarios y migración de datos.
- Permitir a los usuarios personalizar la apariencia de sus hojas Web a través de plantillas.
- Garantizar el acceso al sistema sólo a personas autorizadas, las mismas que tendrán disponibles solamente las opciones relacionadas al rol que cada una de ellas tiene.

3.1.2. Descripción y alcance del Sistema

El sistema de Comunicación Alumno-Profesor (Cursos Web), es un sistema integrado que permite mantener la administración de los cursos y recursos de apoyo a la enseñanza de la ESPOL a través de la Internet.

Los actores involucrados y su interacción, son mostrados en la Figura 3.1 para luego ser explicados de manera detallada junto con las limitaciones y alcance del sistema.

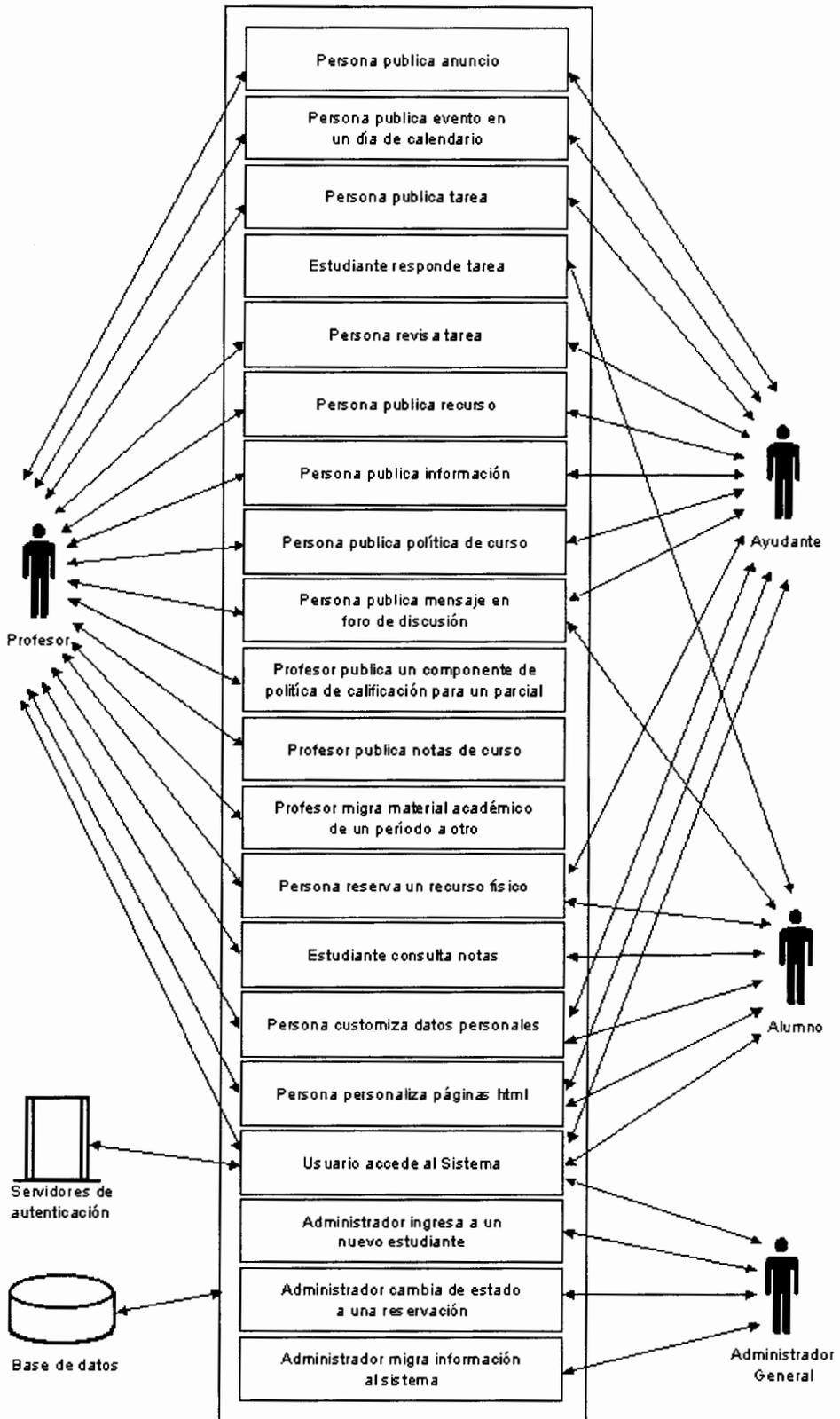


Figura 3.1 Casos de Uso y Actores

3.1.2.1. Descripción de Actores

Un actor es una entidad que interactúa con el sistema como por ejemplo, una persona u otro sistema que requiere o provee de servicios.

Aunque los actores estén fuera de los límites del sistema y no sean objetos del mismo, es importante reconocer sus roles y la manera en que ellos interactúan con el sistema.

Un actor puede jugar múltiples roles; por ejemplo, una misma persona puede estar autorizado a realizar ciertas operaciones como ayudante académico y otras como alumno.

Dentro de determinado contexto pueden haber roles primarios y secundarios: Un actor primario, es aquel que inicia un proceso y para el cual el sistema es construido; y, un actor secundario, es aquel que participa dentro del contexto de forma activa o pasiva para apoyar los objetivos de los actores primarios.

Actores primarios

Profesor. Persona responsable de dictar la materia y de administrar el curso correspondiente. Entre las transacciones que él puede realizar están:

- Publicar información del curso, tal como: anuncios, calendarios, tareas, recursos, información general, políticas del curso, políticas de calificaciones, etc.
- Agregar o eliminar alumnos al curso y definir sus ayudantes académicos.
- Publicar las calificaciones de sus estudiantes a través de archivos existentes o mediante el ingreso directo al sistema. Además tiene la opción para bajar las notas ingresadas a un archivo.
- Interactuar con sus alumnos a través de un foro de discusión.
- Efectuar reservaciones de los recursos académicos.
- Reutilizar la información de otros períodos, tales como: tareas, calendarios, políticas, anuncios, información general y recursos.
- Actualizar su información personal.

- Personalizar la apariencia del sistema a través de las plantillas existentes.

Ayudante. Persona cuya responsabilidad es brindar un soporte tanto al profesor (revisando tareas, tomando lecciones, etc.) como a los alumnos (proporcionando clases o material adicionales, explicando materia, etc.).

En Cursos Web, un ayudante tendrá acceso a realizar a todas las opciones que tiene un profesor, con ciertas excepciones, como: publicar políticas de calificaciones, publicar calificaciones, agregar o eliminar estudiantes y reutilizar la información de cursos anteriores.

Alumno. Persona que debe asistir al curso teniendo la responsabilidad de cumplir con las tareas asignadas por el profesor para aprobar la materia. Entre las operaciones que éste realiza están:

- Consultar toda la información proporcionada por el profesor o ayudante.
- Publicar recursos y las respuestas a las tareas.

- Interactuar con los participantes de su curso a través del foro de discusión.
- Efectuar reservaciones de los recursos académicos.
- Actualizar su información personal.
- Personalizar la apariencia del sistema basándose en plantillas predeterminadas.

Administrador de Unidad. Persona con acceso al sistema de administración, encargada de proporcionar los datos asociados a una unidad académica específica, como por ejemplo, los recursos, materias y paralelos.

Administrador General. Persona con mayores autorizaciones que el administrador de unidad en el sistema de administración (puesto que no tiene restricciones de acceso). Su función principal es la de definir todos los parámetros generales del sistema, así como el mantener los datos necesarios para que el sistema pueda ser utilizado durante el período académico. Tendrá acceso a operaciones como:

- Mantener actualizada la información que utiliza el sistema, tales como: paralelos, materias, profesores, estudiantes, etc.

- Configurar los parámetros del sistema, tales como: año y período vigente, fechas de inicio y fin del semestre, etc.
- Realizar la migración de los datos del Sistema Académico, previo al inicio de un período académico.
- Revisa las reservaciones de recursos las mismas que están aprobadas o pendientes, y según su criterio puede anular las reservaciones aprobadas o aprobar las reservaciones pendientes.

Actores secundarios

Servidores de Autenticación. Servidores de correo electrónico en el que los usuarios poseen una cuenta e-mail. Contra estos servidores se realizará la autenticación POP de los usuarios para garantizar o no el acceso al sistema.

Base de Datos. Repositorio en donde será almacenada y consultada toda la información relacionada a la administración de cursos y recursos del sistema.

3.1.2.2. Alcance y Límites del Sistema

A continuación detallaremos cuál es el alcance y límites del sistema que proponemos con el desarrollo de esta tesis, para poder cubrir las necesidades académicas de la docencia y el alumnado en la Institución.

- Al publicar un anuncio, un profesor o ayudante, estará poniendo a conocimiento de los estudiantes de su curso cualquier novedad existente referente a horarios de clase, extensión para entrega de tareas, etc. Un anuncio podrá ser definido ingresando un texto o especificando un nombre de archivo que lo contenga. Además contará con la facilidad de enviar mails a todos los integrantes del curso informándoles de que existe un nuevo anuncio publicado en el sistema.
- A través de la opción que permitirá publicar un evento en un día de calendario, un profesor o ayudante, podrá poner a conocimiento de sus estudiantes los temas de clases, así como el material relacionado a cada uno de ellos en los días establecidos para dictar la materia. El usuario podrá adjuntar hasta tres archivo por día calendario y tendrá la opción de manejar esta opción a través de una lista o en forma de calendario mes a mes.

- Las especificaciones de una tarea podrán ser publicadas por un profesor o ayudante ingresando un texto explicativo. Adicionalmente se podrá definir un nombre de archivo que contenga información adicional detallada. Con la tarea se deberá definir la fecha de entrega de la misma, con lo cual el sistema controlará si una respuesta de tarea ha sido recibida a tiempo o con retraso. En la publicación de una tarea, el usuario tendrá la opción de publicar un anuncio automático, esto es, el sistema publica un anuncio indicando que existe una nueva tarea. De la misma forma, se podrá enviar un mail automáticamente a todos los integrantes del curso para ponerlos al tanto de la existencia de la nueva tarea.
- Un estudiante podrá responder a una tarea publicada por el profesor o ayudante del curso, para que pueda ser revisada. La respuesta de tarea podrá hacerse en forma de texto o en forma de archivo.
- El profesor o ayudante podrán conocer, a través del sistema, de las tareas que han sido enviadas a tiempo o con retraso. Así también, ambos podrán visualizar el contenido de cada una de las tareas.

- Cualquier integrante del curso podrá publicar recursos académicos. Estos recursos académicos son archivos, artículos, URLs o cualquier otro tipo de información que contribuya a entender el material visto en clases o extender el conocimiento en la materia. Igual que las tareas, la publicación de recursos también permite publicar un anuncio automático indicando que existe un nuevo recurso disponible.
- Al publicar una política de curso, un profesor o ayudante, podrá establecer los lineamientos por ellos exigidos y que los estudiantes deben cumplir, como por ejemplo, asistencia, puntualidad, responsabilidad, consideraciones al tomar lecciones y exámenes, etc. Estas se podrán definir en forma de texto o especificando un nombre de archivo.
- Con la opción de información del curso, se les permitirá al profesor o ayudante, publicar detalles logísticos o reglamentarios que no se consideren como anuncio, recurso o política del curso. Lo podrán hacer ingresando un texto o especificando un nombre de archivo.
- El sistema también permitirá a los integrantes del curso publicar mensajes en un foro de discusión. Este foro de discusión permitirá a los estudiantes, profesores y ayudantes, intercambiar

mensajes relativos al curso a los cuales también podrán adjuntar un archivo de ser necesario. Los mensajes podrán comenzar un hilo de discusión o podrán ser respuestas de un hilo de discusión ya iniciado. La única limitante de los hilos de discusión, es que sólo podrán manejar hasta tres niveles de mensajes. Finalmente sólo la persona que publica un mensaje en el foro, podrá eliminarlo, a excepción del profesor, quien tendrá autorización para eliminar un mensaje publicado por cualquiera de los integrantes del curso.

- El profesor podrá definir la política de calificación de un curso, es decir, cada uno de los componentes de la calificación global tanto de la nota parcial, como de la nota final y mejoramiento. Dentro de cada componente (por ejemplo: deber, examen, proyecto, etc.) se podrá definir la calificación máxima y el porcentaje que éste representa en la calificación total de la nota. Máximo podrán ser definidos hasta cinco componentes en la calificación de una nota.
- El profesor podrá ingresar las calificaciones obtenidas por algunos o todos sus estudiantes en cada uno de los componentes de calificaciones definidas para una nota. Adicionalmente, el profesor podrá ingresar las notas en el

sistema a través de un archivo texto que contenga las identificaciones de sus estudiantes y las notas correspondientes.

- Un estudiante podrá consultar las calificaciones por él obtenidas en cada uno de los componentes de calificación.
- En el inicio de cada período académico, el profesor podrá migrar las tareas, eventos en días de calendario, políticas, anuncios, información y recursos por él publicados en períodos anteriores a su curso en el período actual. Sólo se podrá migrar la información de uno de los períodos anteriores.
- Todos los usuarios del sistema, tendrán la opción de personalizar sus datos personales, tales como: dirección e-mail, dirección domiciliaria, teléfonos, sitio Web de su página personal, aficiones, experiencia laboral, usuario y contraseña de acceso, etc.
- Todos los usuarios podrán personalizar la apariencia de las páginas HTML del sistema, en base a modelos pre-establecidos.
- Estudiantes, profesores y ayudantes podrán hacer reservaciones de recursos a través del sistema, especificando el día y la hora exacta en el que los necesitan. Si un recurso está disponible será otorgado, caso contrario, las solicitudes podrían quedar en lista de espera.

- El acceso al sistema será garantizado, verificando la existencia del nombre de usuario y contraseña en el servidor definido en la información del usuario.
- Existirá un módulo de administración del sistema, a través del cual los administradores podrán:
 - Definir y/o modificar parámetros generales del sistema. Por ejemplo, año y término del período actual, delimitador de campos para migración de datos, tamaños máximos de archivos permitidos en publicaciones realizadas por estudiantes, ayudantes y profesores.
 - Ingresar, modificar y eliminar servidores disponibles para autenticación de usuarios y claves de acceso.
 - Definir cada una de las semanas del período académico actual, esto servirá para reservar recursos.
 - Ingresar, modificar y eliminar estudiantes, profesores, unidades académicas, materias, paralelos, tipos y recursos y recursos disponibles para reservación.
 - Migrar información del sistema académico de la Institución de alumnos, profesores, unidades académicas, materias, paralelos y alumnos por paralelo. Esto se hará utilizando

archivos tipo texto con registros (líneas) que contengan los datos correspondientes, estos registros sólo podrán ser insertados, no será actualizada ninguna información.

- Cada uno de los módulos contará con una ayuda en línea, la misma que detallará qué es lo que el usuario podrá hacer y el cómo deberá hacerlo a través de las diferentes opciones del sistema.

3.2. Especificación de Casos de Uso y Escenarios

Un caso de uso es la secuencia de transacciones¹ en un sistema cuya tarea es producir un resultado con valor medible para un actor del sistema.

Los casos de uso describen los requerimientos funcionales del sistema a alto nivel, desde la perspectiva de los actores típicos. Cada caso de uso describe un comportamiento del sistema observable, el mismo que es una transacción de valor para el actor.

¹ Serie de interacciones entre el sistema y el actor que lo usa.

Los escenarios detallan las variaciones del comportamiento en los casos de uso. Cada escenario describe las suposiciones para cada variación del comportamiento de un caso de uso. Las suposiciones y resultados de cada escenario son descritos en términos del estado de los objetos del dominio del problema y sus relaciones.

3.2.1. Especificación de Casos de Uso

A continuación definimos la lista de los principales casos de uso que han sido considerados en Cursos Web. No son todos, puesto que el sistema es grande, pero son los principales y ayudan a entender la forma en el que el gran todo fue analizado:

1. Persona publica anuncio
2. Persona publica evento en un día de calendario
3. Persona publica tarea
4. Estudiante responde tarea
5. Persona revisa tarea
6. Persona publica recurso
7. Persona publica información

8. Persona publica política de curso
9. Persona publica mensaje en foro de discusión
10. Profesor publica un componente de política de calificación para un parcial.
11. Profesor publica notas de curso
12. Profesor migra material académico de un período a otro
13. Persona reserva un recurso físico
14. Estudiante consulta notas
15. Persona customiza datos personales
16. Persona personaliza páginas HTML
17. Usuario accede al sistema
18. Administrador ingresa a un nuevo estudiante
19. Administrador cambia de estado a una reservación
20. Administrador migra información al sistema

Para cada uno de estos contextos se describe la funcionalidad requerida en el sistema necesaria para cumplir con los objetivos propuestos en esta tesis. Después de realizar un análisis sobre ellos, se analizarán también los escenarios listados en cada caso de uso.

Nombre: 1. Persona publica anuncio

Descripción: Una persona autorizada (profesor o ayudante) ingresa un anuncio que podrá ser consultado por los miembros del curso.

- Notas:**
- Se podrá especificar en forma de texto o en forma de archivo¹.
 - Si se desea se puede enviar un mail a cada uno de los integrantes del curso, informándoles de que existe un nuevo anuncio publicado.
 - El publicar una nueva tarea o un nuevo recurso académico, también permitirá publicar un nuevo anuncio de manera automática.
 - Los anuncios deberán aparecer ordenados por fecha de publicación descendientemente.
 - El último anuncio publicado debe ser más notorio que los demás y se debe mostrar su contenido.

Valor medible: El anuncio es publicado o no.

- Escenarios:**
- 1.1 El anuncio se publica en forma de texto exitosamente.
 - 1.2 El anuncio se publica en forma de archivo exitosamente.
 - 1.3 No se puede publicar el anuncio en forma de archivo porque éste ya existe en el servidor.
 - 1.4 No se puede publicar el anuncio en forma de archivo porque el tamaño de éste sobrepasa al tamaño máximo especificado para el usuario.

Nombre: 2. Persona publica evento en un día de calendario

Descripción: Una persona autorizada (profesor o ayudante) ingresa un evento en un día del calendario para su materia que podrá ser consultado por los miembros de su curso.

¹ Archivo existente en el PC del usuario (documento de Word, Excel, etc.)

- Notas:**
- Además del texto correspondiente al evento, podrán ser adjuntados hasta tres archivos por cada día especificado.
 - Los eventos publicados podrán ser presentados en forma de calendario mes a mes.
 - Podrán aparecer también en forma de lista ordenados por fecha de publicación en forma descendente.

Valor medible: El evento en un día de calendario es publicado o no.

- Escenarios:**
- 2.1 El evento en un día de calendario se publica en forma exitosa.
 - 2.2 No se puede publicar el evento en un día de calendario porque uno de los archivos ya existe en el servidor.
 - 2.3 No se puede publicar el evento en un día de calendario porque el tamaño de uno de los archivos sobrepasa al tamaño máximo especificado para el usuario.

Nombre: **3. Persona publica tarea**

Descripción: Una persona autorizada (profesor o ayudante) ingresa una tarea que podrá ser vista y respondida por cada uno de los estudiantes del curso.

- Notas:**
- Se puede adjuntar un archivo a las especificaciones de la tarea.
 - Si se desea se puede publicar un anuncio, informando que existe una nueva tarea.
 - Si se desea se puede enviar un mail a cada uno de los integrantes del curso, informándoles de que existe una nueva tarea publicada.

Valor medible: La tarea es publicada o no.

- Escenarios:**
- 3.1 La tarea se publica en forma de texto exitosamente.

- 3.2 La tarea se publica con un archivo adjunto exitosamente.
- 3.3 No se puede publicar la tarea porque se desea adjuntar un archivo que ya existe en el servidor.
- 3.4 No se puede publicar la tarea porque se desea adjuntar un archivo cuyo tamaño sobrepasa al tamaño máximo especificado para el usuario.

Nombre: **4. Estudiante responde tarea**

Descripción: Un estudiante ingresa una respuesta de tarea que luego podrá ser revisada por el profesor o ayudante del curso.

Notas: - La respuesta de tarea podrá ingresarse en forma de texto o en forma de archivo.

Valor medible: La respuesta de tarea es enviada o no.

- Escenarios:**
- 4.1 La tarea es respondida a tiempo en forma de texto exitosamente.
 - 4.2 La tarea es respondida con atraso en forma de texto exitosamente.
 - 4.3 La tarea es respondida a tiempo en forma de archivo exitosamente.
 - 4.4 La tarea es respondida con atraso en forma de archivo exitosamente.
 - 4.5 No se puede responder la tarea porque se desea adjuntar un archivo que ya existe en el servidor.
 - 4.6 No se puede responder la tarea porque se desea adjuntar un archivo cuyo tamaño sobrepasa al tamaño máximo especificado para el usuario.

Nombre: 5. Persona revisa tarea**Descripción:** Una persona autorizada (profesor o ayudante) revisa las tareas que han sido registradas por cada uno de los estudiantes del curso.**Notas:****Valor medible:** La respuesta de tarea es revisada.

- Escenarios:**
- 5.1 La respuesta de tarea en forma de texto es revisada exitosamente.
 - 5.2 La respuesta de tarea en forma de archivo es revisada exitosamente.

Nombre: 6. Persona publica recurso**Descripción:** Un usuario ingresa un recurso para que este disponible para los miembros de un curso.

- Notas:**
- Se podrá especificar un enlace (URL), un archivo, un artículo u otros (estos tres últimos realmente permiten ingresar cualquier archivo. La diferencia es simplemente que se categoriza al momento de ingresarlo).
 - Si se desea se puede publicar un anuncio, informando que existe un nuevo recurso.
 - Si se desea se puede enviar un mail a cada uno de los integrantes del curso, informándoles de que existe un nuevo recurso publicado.

Valor medible: El recurso es publicado o no.

- Escenarios:**
- 6.1 El recurso se publica en forma de archivo (cualquiera de las tres categorías) exitosamente.
 - 6.2 El recurso se publica en forma de enlace exitosamente.
 - 6.3 No se puede publicar el recurso porque se desea adjuntar un archivo que ya existe en el servidor.

- 6.4 No se puede publicar el recurso porque se desea adjuntar un archivo cuyo tamaño sobrepasa al tamaño máximo especificado para el usuario.

Nombre: **7. Persona publica información**

Descripción: Una persona autorizada (profesor o ayudante) ingresa alguna información para que este disponible para cualquiera de los miembros de un curso.

Notas: - Se podrá especificar en forma de texto o en forma de archivo.

Valor medible: La información es publicada o no.

- Escenarios:**
- 7.1 La información se publica en forma de texto exitosamente.
 - 7.2 La información se publica en forma de archivo exitosamente.
 - 7.3 No se puede publicar la información porque se desea adjuntar un archivo que ya existe en el servidor.
 - 7.4 No se puede publicar la información porque se desea adjuntar un archivo cuyo tamaño sobrepasa al tamaño máximo especificado para el usuario.

Nombre: **8. Persona publica política de curso**

Descripción: Una persona autorizada (profesor o ayudante) ingresa una política que podrá ser consultada por los miembros del curso.

Notas: - Se podrá especificar en forma de texto o en forma de archivo.

- Valor medible:** La política de curso es publicada o no.
- Escenarios:**
- 8.1 La política de curso se publica en forma de texto exitosamente.
 - 8.2 La política de curso se publica en forma de archivo exitosamente.
 - 8.3 No se puede publicar la política de curso porque se desea adjuntar un archivo que ya existe en el servidor.
 - 8.4 No se puede publicar la política de curso porque se desea adjuntar un archivo cuyo tamaño sobrepasa al tamaño máximo especificado para el usuario.
- Nombre:** **9. Persona publica mensaje en foro de discusión**
- Descripción:** Un usuario ingresa un mensaje en el foro de discusión que podrá ser consultado y respondido por cada uno de los miembros del curso.
- Notas:**
- Sólo podrán ser definidos hasta tres niveles de mensajes en el foro.
 - Se podrá también adjuntar un archivo en el mensaje.
- Valor medible:** El mensaje en el foro de discusión es publicado o no.
- Escenarios:**
- 9.1 El mensaje se publica sin archivo adjunto en forma exitosa.
 - 9.2 El mensaje se publica con un archivo adjunto en forma exitosa.
 - 9.3 El mensaje se publica con texto y archivo adjunto en forma exitosa.
 - 9.4 No se puede publicar el mensaje porque se desea adjuntar un archivo que ya existe en el servidor.
 - 9.5 No se puede publicar el mensaje porque se desea adjuntar un archivo cuyo tamaño

sobrepasa al tamaño máximo especificado para el usuario.

Nombre: **10. Profesor publica un componente de política de calificación para un parcial**

Descripción: El profesor define uno de los componentes de la calificación global de un parcial. El profesor también define la calificación máxima que podrá ser obtenida en este componente (deber, examen, proyecto, etc.) y el porcentaje que este representa en la calificación total del parcial.

Notas: - Máximo podrán ser definidos cinco componentes de la calificación de un parcial.

Valor medible: El componente de la calificación es publicado o no.

Escenarios:

- 10.1 El profesor publica un componente de calificación de manera exitosa.
- 10.2 El profesor no puede publicar el componente de la calificación porque ya existen cinco componentes de la calificación definidos para ese parcial.
- 10.3 El profesor no puede publicar el componente de la calificación porque la suma de los porcentajes de los componentes de calificación del parcial sobrepasa el 100%.

Nombre: **11. Profesor publica notas de curso**

Descripción: El profesor ingresa las calificaciones obtenidas por sus estudiantes en cada uno de los componentes de calificaciones definidas para cada nota.

Notas: - Puede ingresar todas las notas o sólo las de algunos estudiantes.

Valor medible: Las notas del curso son publicadas o no.

Escenarios: 11.1 El profesor publica las calificaciones de los componentes definidos de manera exitosa.
11.2 El profesor no puede ingresar las calificaciones porque no existen componentes de calificaciones definidos para ese parcial.

Nombre: **12. Profesor migra material académico de un período a otro**

Descripción: El profesor selecciona el material que desea tener disponible en el período académico actual de los cursos dictados por él en períodos anteriores.

Notas: - Sólo podrá ser migrado el material del profesor de un período académico anterior por materia.

Valor medible: El material necesario es migrado o no.

Escenarios: 12.1 El profesor migra el material deseado exitosamente.
12.2 El profesor desea migrar el material de una materia que no está dictando en el período actual.
12.3 El profesor desea migrar el material de una materia que ya ha migrado anteriormente.

Nombre: **13. Persona reserva un recurso físico**

Descripción: Un usuario selecciona un recurso y especifica las horas y las semanas del período académico en las que desea reservar el recurso.

- Notas:**
- La solicitud será aprobada siempre y cuando el recurso esté disponible en todos los horarios especificados por el usuario.
 - Si el recurso a reservar no está disponible, el usuario podría decidir colocar su solicitud en lista de espera.

Valor medible: El recurso es reservado o no.

- Escenarios:**
- 13.1 Una persona reserva un recurso físico de manera exitosa.
 - 13.2 Una persona no reserva el recurso físico porque no se encuentra disponible.
 - 13.3 La solicitud de recurso físico queda en lista de espera porque no se encuentra disponible.

Nombre: **14. Estudiante consulta notas**

Descripción: Un estudiante consulta las calificaciones obtenidas en cada uno de los componentes de calificación de una nota. Visualiza también el total obtenido por parcial.

Notas:

Valor medible: Las notas obtenidas por el estudiante en los diferentes componentes de calificación son presentadas o no.

- Escenarios:**
- 14.1 Un estudiante consulta las calificaciones obtenidas en los componentes de calificación.
 - 14.2 Un estudiante no puede consultar las calificaciones obtenidas porque no existen definidos en el sistema los componentes de calificación para ninguna nota.
 - 14.3 Un estudiante no puede consultar las calificaciones obtenidas porque no han sido ingresadas al sistema.

Nombre: **15. Persona customiza datos personales**

Descripción: Un usuario modifica su información personal y el tipo de autenticación que debe realizarse cuando él desee acceder al sistema. También define cuáles datos podrán ser visualizados por todos los integrantes del curso.

Notas:

Valor medible: La información personal del usuario es modificada o no.

Escenarios: 15.1 La información personal de una persona es modificada exitosamente.
15.2 La información personal de una persona no es modificada porque ha definido un servidor de autenticación y un nombre de usuario de acceso que ya existe para otra persona.

Nombre: **16. Persona personaliza páginas HTML**

Descripción: El usuario conectado al sistema (profesor, alumno o ayudante) selecciona el diseño de página de su preferencia para ser aplicado en todas las páginas del sistema.

Notas: - Sólo podrán modificarse los diseños seleccionando plantillas, esto es, no se podrán customizar colores o tipos de letra de manera independiente.

Valor medible: La plantilla seleccionada es aplicada a todas las páginas del sistema para ese usuario.

Escenarios: 16.1 El usuario tiene un directorio de configuración de páginas HTML, selecciona una plantilla y el cambio es realizado exitosamente.
16.2 El usuario no tiene un directorio de configuración de páginas HTML, selecciona una plantilla y el cambio es realizado exitosamente.

Nombre: 17. Usuario accede al Sistema

Descripción: Una persona intenta acceder al sistema introduciendo un nombre de usuario y una clave. Las opciones en el sistema son habilitadas de acuerdo al perfil del usuario identificado.

Notas:

- Considerar que puede existir un mismo nombre de usuario en varios servidores de autenticación.
- El acceso al sistema no puede ser garantizado si los servidores contra los que hay que autenticar (autenticación POP) no responden los requerimientos.

Valor medible: El acceso es otorgado o no.

Escenarios:

- 17.1 Acceso otorgado a persona cuyo nombre de usuario es único, con contraseña correcta y sin conexiones abiertas.
- 17.2 Acceso otorgado a persona cuyo nombre de usuario es único, con contraseña correcta y con conexiones abiertas.
- 17.3 Acceso otorgado a persona cuyo nombre de usuario no es único con contraseña correcta.
- 17.4 Acceso no otorgado a persona cuyo nombre de usuario no existe.
- 17.5 Acceso no otorgado a persona cuyo nombre de usuario tiene contraseña incorrecta.
- 17.6 Acceso no otorgado a persona cuyo nombre de usuario no es único con contraseña incorrecta.

Nombre: 18. Administrador ingresa un nuevo estudiante

Descripción: El administrador especifica la información personal de un nuevo estudiante cuya información no ha sido migrada.

Notas:

Valor medible: Un nuevo estudiante es insertado o no.

- Escenarios:**
- 18.1 Un estudiante es ingresado exitosamente.
 - 18.2 Un estudiante no es ingresado porque el número de identificación pertenece a otro estudiante.
 - 18.3 Un estudiante no es ingresado porque se ha definido un servidor de autenticación y un nombre de usuario de acceso que ya existe para otra persona.

Nombre: **19. Administrador cambia de estado a una reservación**

Descripción: El administrador selecciona una reservación y cambia el estado de la misma. Si está aprobada la coloca en lista de espera; y, si está en lista de espera la aprueba.

Notas: - La solicitud sólo pasará a aprobada siempre y cuando haya disponibilidad del recurso.

Valor medible: La reservación cambia de estado o no.

- Escenarios:**
- 19.1 Una solicitud aprobada es puesta en lista de espera exitosamente.
 - 19.2 Una solicitud en lista de espera es aprobada exitosamente.
 - 19.3 Una solicitud en lista de espera no es aprobada porque el recurso físico no está disponible en todos los horarios de la solicitud.

Nombre: **20. Administrador migra información al sistema**

Descripción: El administrador general especifica los nombres de los archivos que contienen la información del período actual que debe ser insertado en las tablas correspondientes. Además indica si se deben o no generar los directorios de trabajo.

- Notas:**
- No se actualizarán los registros ya existentes en las distintas tablas del sistema.
 - Se podrán migrar los datos de unidades académicas, estudiantes, profesores, materias, paralelos y estudiantes registrados por paralelo.
- Valor medible:** Los datos de los archivos definidos son insertados o no.
- Escenarios:**
- 20.1 Los datos de migración se insertan con éxito y los directorios de trabajo son creados.
 - 20.2 Los datos de migración no son insertados porque estos son inconsistentes¹ con las estructuras de las tablas.

3.2.2. Especificación de escenarios

Caso de Uso 1: Persona publica anuncio

Escenario 1.1: *El anuncio se publica en forma de texto exitosamente.*

Asunciones:

- El directorio de trabajo del paralelo existe.
- Se ha ingresado una descripción y un texto.
- El usuario especifica enviar mails.

Resultados:

- Un nuevo anuncio es registrado.
- Es generado un archivo texto con el contenido del anuncio.
- Un mail es enviado a cada uno de los integrantes del curso informándoles de la publicación de una nueva tarea.
- Un mensaje de éxito es presentado con los datos registrados.

¹ Por ejemplo: Número de columnas diferentes a las definidas en la estructura de la tabla, longitud de datos incorrecta, inconsistencia en tipos de datos, etc.

Escenario 1.2: *El anuncio se publica en forma de archivo exitosamente.*

- Asunciones:**
- El directorio de trabajo del paralelo existe.
 - Se ha ingresado una descripción y un nombre de archivo.
 - El nombre del archivo es válido.
 - El archivo existe en la ruta especificada.
 - El archivo no ha sido definido ningún anuncio previo.
 - El tamaño del archivo no es cero.
 - El tamaño del archivo no es mayor al tamaño máximo permitido para cada uno de los archivos definidos por el usuario.
 - El usuario especifica enviar mails.

- Resultados:**
- Un nuevo anuncio es registrado.
 - El archivo especificado es copiado al servidor.
 - Un mail es enviado a cada uno de los integrantes del curso informándoles de la publicación de una nueva tarea.

Escenario 1.3: *No se puede publicar el anuncio en forma de archivo porque éste ya existe en el servidor.*

- Asunciones:**
- El directorio de trabajo del paralelo existe.
 - Se ha ingresado una descripción y un nombre de archivo.
 - El nombre del archivo es válido.
 - El archivo existe en la ruta especificada.
 - El archivo ha sido definido en un anuncio previo.
 - El tamaño del archivo no es cero.
 - El tamaño del archivo no es mayor al tamaño máximo permitido para cada uno de los archivos definidos por el usuario.

- Resultados:**
- Un mensaje de error es presentado.
 - El anuncio no es registrado.

Escenario 1.4: *No se puede publicar el anuncio en forma de archivo porque el tamaño de éste sobrepasa al tamaño máximo especificado para el usuario.*

- Asunciones:**
- El directorio de trabajo del paralelo existe.
 - Se ha ingresado una descripción y un nombre de archivo.
 - El nombre del archivo es válido.
 - El archivo existe en la ruta especificada.
 - El archivo no ha sido definido ningún anuncio previo.
 - El tamaño del archivo no es cero.
 - El tamaño del archivo es mayor al tamaño máximo permitido para cada uno de los archivos definidos por el usuario.

- Resultados:**
- Un mensaje de error es presentado.
 - El anuncio no es registrado.

<p>Caso de Uso 2: <i>Persona publica evento en un día de calendario</i></p>

Escenario 2.1: *El evento en un día de calendario se publica en forma exitosa.*

- Asunciones:**
- El directorio de trabajo del paralelo existe.
 - Se han especificado nombres de archivos.
 - Los nombres de los archivos son válidos.
 - Los archivos existen en la ruta especificada.
 - Ninguno de los archivos ha sido definido en un evento de día de calendario previo.
 - Ninguno de los archivos tiene tamaño cero.
 - El tamaño de cada uno de los archivos no es mayor al tamaño máximo permitido para cada archivo definido por el usuario.

- Resultados:**
- Un nuevo evento en un día de calendario es registrado.

- Los archivos especificados son copiados al servidor.
- Un mensaje de éxito es presentado con los datos registrados.

Escenario 2.2: *No se puede publicar el evento en un día de calendario porque uno de los archivos ya existe en el servidor.*

- Asunciones:**
- El directorio de trabajo del paralelo existe.
 - Se han especificado nombres de archivos.
 - Los nombres de los archivos son válidos.
 - Los archivos existen en la ruta especificada.
 - Uno de los archivos ha sido definido en un evento de día de calendario previo.
 - Ninguno de los archivos tiene tamaño cero.
 - El tamaño de cada uno de los archivos no es mayor al tamaño máximo permitido para cada archivo definido por el usuario.

- Resultados:**
- El evento en un día de calendario no es registrado.
 - Un mensaje de error es presentado.

Escenario 2.3: *No se puede publicar el evento en un día de calendario porque el tamaño de uno de los archivos sobrepasa al tamaño máximo especificado para el usuario.*

- Asunciones:**
- El directorio de trabajo del paralelo existe.
 - Se han especificado nombres de archivos.
 - Los nombres de los archivos son válidos.
 - Los archivos existen en la ruta especificada.
 - Ninguno de los archivos ha sido definido en un evento de día de calendario previo.
 - Ninguno de los archivos tiene tamaño cero.
 - El tamaño de uno de los archivos es mayor al tamaño máximo permitido para cada archivo definido por el usuario.

- Resultados:**
- El evento en un día de calendario no es registrado.
 - Un mensaje de error es presentado.

Caso de Uso 3: <i>Persona publica tarea</i>

Escenario 3.1: *La tarea se publica en forma de texto exitosamente.*

- Asunciones:**
- El directorio de trabajo del paralelo existe.
 - Se ha ingresado un titulo, una fecha de entrega y un contenido para la tarea.
 - No se adjunta archivo para la tarea.
 - El usuario especifica publicar un anuncio.
 - El usuario especifica enviar mails.

- Resultados:**
- Una nueva tarea es registrada.
 - Es generado un archivo texto con la descripción de la tarea.
 - Un anuncio es publicado automáticamente indicando que existe una nueva tarea.
 - Un mail es enviado a cada uno de los integrantes del curso informándoles de la publicación de una nueva tarea.
 - Un mensaje de éxito es presentado con los datos registrados.

Escenario 3.2: *La tarea se publica con un archivo adjunto exitosamente.*

- Asunciones:**
- El directorio de trabajo del paralelo existe.
 - En la forma de ingreso se especificó un título, una fecha de entrega, un contenido y un nombre de archivo.
 - El nombre del archivo es válido.
 - El archivo existe en la ruta especificada.
 - El archivo no ha sido definido en ninguna tarea previa.

- El tamaño del archivo no es cero.
- El tamaño del archivo no es mayor al tamaño máximo permitido para cada uno de los archivos definidos por el usuario.
- El usuario especifica publicar un anuncio.
- El usuario especifica enviar mails.

- Resultados:**
- Una nueva tarea es registrada.
 - El archivo especificado es copiado al servidor.
 - Un anuncio es publicado automáticamente indicando que existe una nueva tarea.
 - Un mail es enviado a cada uno de los integrantes del curso informándoles de la publicación de una nueva tarea.

Escenario 3.3: *No se puede publicar la tarea porque se desea adjuntar un archivo que ya existe en el servidor.*

- Asunciones:**
- El directorio de trabajo del paralelo existe.
 - En la forma de ingreso se especificó un título, una fecha de entrega, un contenido y un nombre de archivo.
 - El nombre del archivo es válido.
 - El archivo existe en la ruta especificada.
 - El archivo ha sido definido en una tarea previa.
 - El tamaño del archivo no es cero.
 - El tamaño del archivo no es mayor al tamaño máximo permitido para cada uno de los archivos definidos por el usuario.

- Resultados:**
- Un mensaje de error es presentado.
 - La tarea no es registrada.
 - No es publicado ningún anuncio indicando que existe nueva tarea.

Escenario 3.4: *No se puede publicar la tarea porque se desea adjuntar un archivo cuyo tamaño sobrepasa al tamaño máximo especificado para el usuario.*

- Asunciones:**
- El directorio de trabajo del paralelo existe.
 - En la forma de ingreso se especificó un título, una fecha de entrega, un contenido y un nombre de archivo.
 - El nombre del archivo es válido.
 - El archivo existe en la ruta especificada.
 - El archivo no ha sido definido en ninguna tarea previa.
 - El tamaño del archivo no es cero.
 - El tamaño del archivo es mayor al tamaño máximo permitido para cada uno de los archivos definidos por el usuario.
- Resultados:**
- Un mensaje de error es presentado.
 - La tarea no es registrada.
 - No es publicado ningún anuncio indicando que existe nueva tarea.

Caso de Uso 4: <i>Estudiante responde tarea</i>

Escenario 4.1: *La tarea es respondida a tiempo en forma de texto exitosamente.*

- Asunciones:**
- El directorio de trabajo del paralelo existe.
 - El profesor o el ayudante ha publicado una tarea.
 - El estudiante está enviando la tarea dentro del plazo especificado.
 - Se ha ingresado el texto de la respuesta de tarea.
- Resultados:**
- Una nueva respuesta de tarea es registrada.
 - La respuesta de tarea tiene definido el estado de entregada a tiempo.
 - Es generado un archivo texto con el contenido de la respuesta de tarea.
 - Un mensaje de éxito es presentado con los datos registrados.

Escenario 4.2: *La tarea es respondida con atraso en forma de texto exitosamente.*

- Asunciones:**
- El directorio de trabajo del paralelo existe.
 - El profesor o el ayudante ha publicado una tarea.
 - El estudiante está enviando la tarea después del plazo especificado.
 - Se ha ingresado el texto de la respuesta de tarea.

- Resultados:**
- Una nueva respuesta de tarea es registrada.
 - La respuesta de tarea tiene definido el estado de entregada con atraso.
 - Es generado un archivo texto con el contenido de la respuesta de tarea.
 - Un mensaje de éxito es presentado con los datos registrados.

Escenario 4.3: *La tarea es respondida a tiempo en forma de archivo exitosamente.*

- Asunciones:**
- El directorio de trabajo del paralelo existe.
 - El profesor o el ayudante han publicado una tarea.
 - El estudiante está enviando la tarea dentro del plazo especificado.
 - Se ha especificado un nombre de archivo.
 - El nombre del archivo es válido.
 - El archivo existe en la ruta especificada.
 - El archivo no ha sido definido en ninguna respuesta de tarea previamente.
 - El tamaño del archivo no es cero.
 - El tamaño del archivo no es mayor al tamaño máximo permitido para cada uno de los archivos definidos por el usuario.

- Resultados:**
- Una nueva respuesta de tarea es registrada.
 - La respuesta de tarea tiene definido el estado de entregada a tiempo.
 - El archivo especificado es copiado al servidor.

Escenario 4.4: *La tarea es respondida con atraso en forma de archivo exitosamente.*

- Asunciones:**
- El directorio de trabajo del paralelo existe.
 - El profesor o el ayudante han publicado una tarea.
 - El estudiante está enviando la tarea después del plazo especificado.
 - Se ha especificado un nombre de archivo.
 - El nombre del archivo es válido.
 - El archivo existe en la ruta especificada.
 - El archivo no ha sido definido en ninguna respuesta de tarea previamente.
 - El tamaño del archivo no es cero.
 - El tamaño del archivo no es mayor al tamaño máximo permitido para cada uno de los archivos definidos por el usuario.

- Resultados:**
- Una nueva respuesta de tarea es registrada.
 - La respuesta de tarea tiene definido el estado de entregada con atraso.
 - El archivo especificado es copiado al servidor.

Escenario 4.5: *No se puede responder la tarea porque se desea adjuntar un archivo que ya existe en el servidor.*

- Asunciones:**
- El directorio de trabajo del paralelo existe.
 - El profesor o el ayudante han publicado una tarea.
 - Se ha especificado un nombre de archivo.
 - El nombre del archivo es válido.
 - El archivo existe en la ruta especificada.
 - El archivo ha sido definido en una respuesta de tarea previamente.
 - El tamaño del archivo no es cero.
 - El tamaño del archivo no es mayor al tamaño máximo permitido para cada uno de los archivos definidos por el usuario.

- Resultados:**
- Un mensaje de error es presentado.
 - La respuesta de tarea no es registrada.

Escenario 4.6: *No se puede responder la tarea porque se desea adjuntar un archivo cuyo tamaño sobrepasa al tamaño máximo especificado para el usuario.*

Asunciones:

- El directorio de trabajo del paralelo existe.
- El profesor o el ayudante han publicado una tarea.
- Se ha especificado un nombre de archivo.
- El nombre del archivo es válido.
- El archivo existe en la ruta especificada.
- El archivo no ha sido definido en ninguna respuesta de tarea previamente.
- El tamaño del archivo no es cero.
- El tamaño del archivo es mayor al tamaño máximo permitido para cada uno de los archivos definidos por el usuario.

Resultados:

- Un mensaje de error es presentado.
- La respuesta de tarea no es registrada.

Caso de Uso 5: <i>Persona revisa tarea</i>

Escenario 5.1: *La respuesta de tarea en forma de texto es revisada exitosamente.*

Asunciones:

- Existe una tarea publicada.
- La tarea fue respondida en forma de texto.

Resultados:

- El contenido de la respuesta de tarea es presentado.

Escenario 5.2: *La respuesta de tarea en forma de archivo es revisada exitosamente.*

Asunciones:

- Existe una tarea publicada.
- La tarea fue respondida en forma de archivo.

- Resultados:** - Dependiendo del navegador, el contenido de la respuesta de tarea es presentado en el editor correspondiente (por ejemplo: si es un documento.doc se abre Microsoft Word) o se presenta una caja de diálogo para copiar el documento en la PC.

Caso de Uso 6:	Persona publica recurso
-----------------------	--------------------------------

Escenario 6.1: *El recurso se publica en forma de archivo (cualquiera de las tres categorías) exitosamente.*

- Asunciones:** - El directorio de trabajo del paralelo existe.
- Se ha ingresado un nombre de archivo y una descripción.
 - El nombre del archivo es válido.
 - El archivo existe en la ruta especificada.
 - El archivo no ha sido definido ningún recurso previo.
 - El tamaño del archivo no es cero.
 - El tamaño del archivo no es mayor al tamaño máximo permitido para cada uno de los archivos definidos por el usuario.
 - El usuario especifica publicar un anuncio.
 - El usuario especifica enviar mails.

- Resultados:** - Un nuevo recurso es registrado.
- El archivo especificado es copiado al servidor.
 - Un anuncio es publicado automáticamente indicando que existe un nuevo recurso.
 - Un mail es enviado a cada uno de los integrantes del curso informándoles de la publicación de una nueva tarea.
 - Un mensaje de éxito es presentado con los datos registrados.

Escenario 6.2: *El recurso se publica en forma de enlace exitosamente.*

Asunciones: - El directorio de trabajo del paralelo existe.
 - Se ha ingresado un enlace y una descripción para el recurso.
 - El usuario especifica publicar un anuncio.
 - El usuario especifica enviar mails.

Resultados: - Un nuevo recurso es registrado.
 - Un anuncio es publicado automáticamente indicando que existe un nuevo recurso.
 - Un mail es enviado a cada uno de los integrantes del curso informándoles de la publicación de un nuevo recurso.

Escenario 6.3: *No se puede publicar el recurso porque se desea adjuntar un archivo que ya existe en el servidor.*

Asunciones: - El directorio de trabajo del paralelo existe.
 - Se ha ingresado un nombre de archivo y una descripción.
 - El nombre del archivo es válido.
 - El archivo existe en la ruta especificada.
 - El archivo ha sido definido en un recurso previo.
 - El tamaño del archivo no es cero.
 - El tamaño del archivo no es mayor al tamaño máximo permitido para cada uno de los archivos definidos por el usuario.

Resultados: - Un mensaje de error es presentado.
 - El recurso no es registrado.
 - No es publicado ningún anuncio indicando que existe nuevo recurso.

Escenario 6.4: *No se puede publicar el recurso porque se desea adjuntar un archivo cuyo tamaño sobrepasa al tamaño máximo especificado para el usuario.*

- Asunciones:**
- El directorio de trabajo del paralelo existe.
 - Se ha ingresado un nombre de archivo y una descripción.
 - El nombre del archivo es válido.
 - El archivo existe en la ruta especificada.
 - El archivo no ha sido definido ningún recurso previo.
 - El tamaño del archivo no es cero.
 - El tamaño del archivo es mayor al tamaño máximo permitido para cada uno de los archivos definidos por el usuario.
- Resultados:**
- Un mensaje de error es presentado.
 - El recurso no es registrado.
 - No es publicado ningún anuncio indicando que existe nuevo recurso.

Caso de Uso 7:	<i>Persona publica información</i>
-----------------------	-------------------------------------------

Escenario 7.1: *La información se publica en forma de texto exitosamente.*

- Asunciones:**
- El directorio de trabajo del paralelo existe.
 - En la forma de ingreso se especificó un título y un contenido de la información.

- Resultados:**
- Una nueva información es registrada.
 - Es generado un archivo texto con el contenido de la información.
 - Un mensaje de éxito es presentado con los datos registrados.

Escenario 7.2: *La información se publica en forma de archivo exitosamente.*

- Asunciones:**
- El directorio de trabajo del paralelo existe.
 - En la forma de ingreso se especificó un título y un nombre de archivo.

- El nombre del archivo es válido.
- El archivo existe en la ruta especificada.
- El archivo no ha sido definido en ninguna información previa.
- El tamaño del archivo no es cero.
- El tamaño del archivo no es mayor al tamaño máximo permitido para cada uno de los archivos definidos por el usuario.

- Resultados:**
- Una nueva información es registrada.
 - El archivo especificado es copiado al servidor.

Escenario 7.3: *No se puede publicar la información porque se desea adjuntar un archivo que ya existe en el servidor.*

- Asunciones:**
- El directorio de trabajo del paralelo existe.
 - En la forma de ingreso se especificó un título y un nombre de archivo.
 - El nombre del archivo es válido.
 - El archivo existe en la ruta especificada.
 - El archivo ha sido definido en una información previa.
 - El tamaño del archivo no es cero.
 - El tamaño del archivo no es mayor al tamaño máximo permitido para cada uno de los archivos definidos por el usuario.

- Resultados:**
- Un mensaje de error es presentado.
 - La información no es registrada.

Escenario 7.4: *No se puede publicar la información porque se desea adjuntar un archivo cuyo tamaño sobrepasa al tamaño máximo especificado para el usuario.*

- Asunciones:**
- El directorio de trabajo del paralelo existe.
 - En la forma de ingreso se especificó un título y un nombre de archivo.
 - El nombre del archivo es válido.
 - El archivo existe en la ruta especificada.

- El archivo no ha sido definido en ninguna información previa.
- El tamaño del archivo no es cero.
- El tamaño del archivo es mayor al tamaño máximo permitido para cada uno de los archivos definidos por el usuario.

Resultados: - Un mensaje de error es presentado.
- La información no es registrada.

Caso de Uso 8: <i>Persona publica política de curso</i>

Escenario 8.1: *La política de curso se publica en forma de texto exitosamente.*

Asunciones: - El directorio de trabajo del paralelo existe.
- Se ha ingresado un título y un contenido.

Resultados: - Una nueva política es registrada.
- Es generado un archivo texto con el contenido de la política.
- Un mensaje de éxito es presentado con los datos registrados.

Escenario 8.2: *La política de curso se publica en forma de archivo exitosamente.*

Asunciones: - El directorio de trabajo del paralelo existe.
- Se ha ingresado un título y un nombre de archivo.
- El nombre del archivo es válido.
- El archivo existe en la ruta especificada.
- El archivo no ha sido definido en ninguna política de curso previa.
- El tamaño del archivo no es cero.
- El tamaño del archivo no es mayor al tamaño máximo permitido para cada uno de los archivos definidos por el usuario.

- Resultados:** - Una nueva política de curso es registrada.
- El archivo especificado es copiado al servidor.

Escenario 8.3: *No se puede publicar la política de curso porque se desea adjuntar un archivo que ya existe en el servidor.*

- Asunciones:** - El directorio de trabajo del paralelo existe.
- Se ha ingresado un título y un nombre de archivo.
- El nombre del archivo es válido.
- El archivo existe en la ruta especificada.
- El archivo ha sido definido en una política de curso previa.
- El tamaño del archivo no es cero.
- El tamaño del archivo no es mayor al tamaño máximo permitido para cada uno de los archivos definidos por el usuario.

- Resultados:** - Un mensaje de error es presentado.
- La política de curso no es registrada.

Escenario 8.4: *No se puede publicar la política de curso porque se desea adjuntar un archivo cuyo tamaño sobrepasa al tamaño máximo especificado para el usuario.*

- Asunciones:** - El directorio de trabajo del paralelo existe.
- Se ha ingresado un título y un nombre de archivo.
- El nombre del archivo es válido.
- El archivo existe en la ruta especificada.
- El archivo no ha sido definido en ninguna política de curso previa.
- El tamaño del archivo no es cero.
- El tamaño del archivo es mayor al tamaño máximo permitido para cada uno de los archivos definidos por el usuario.

- Resultados:** - Un mensaje de error es presentado.
- La política de curso no es registrada.

Caso de Uso 9:	Persona publica mensaje en foro de discusión
-----------------------	-----------------------------------------------------

Escenario 9.1: *El mensaje se publica sin archivo adjunto en forma exitosa.*

Asunciones: - El directorio de trabajo del paralelo existe.
 - Se ha ingresado un titulo y un contenido para el mensaje.
 - No se adjunta archivo al mensaje.

Resultados: - Un nuevo mensaje es registrado.
 - Es generado un archivo texto con la descripción del mensaje.
 - Un mensaje de éxito es presentado con los datos registrados.

Escenario 9.2: *El mensaje se publica sólo con un archivo adjunto en forma exitosa.*

Asunciones: - El directorio de trabajo del paralelo existe.
 - En la forma de ingreso se especificó un título y un nombre de archivo.
 - El nombre del archivo es válido.
 - El archivo existe en la ruta especificada.
 - El archivo no ha sido definido ningún mensaje previo.
 - El tamaño del archivo no es cero.
 - El tamaño del archivo no es mayor al tamaño máximo permitido para cada uno de los archivos definidos por el usuario.

Resultados: - Un nuevo mensaje es registrado.
 - El archivo especificado es copiado al servidor.

Escenario 9.3: *El mensaje se publica con texto y archivo adjunto en forma exitosa.*

Asunciones: - El directorio de trabajo del paralelo existe.
 - En la forma de ingreso se especificó un título, un contenido y un nombre de archivo.

- El nombre del archivo es válido.
- El archivo existe en la ruta especificada.
- El archivo no ha sido definido ningún mensaje previo.
- El tamaño del archivo no es cero.
- El tamaño del archivo no es mayor al tamaño máximo permitido para cada uno de los archivos definidos por el usuario.

- Resultados:**
- Un nuevo mensaje es registrado.
 - El archivo especificado es copiado al servidor.

Escenario 9.4: *No se puede publicar el mensaje porque se desea adjuntar un archivo que ya existe en el servidor.*

- Asunciones:**
- El directorio de trabajo del paralelo existe.
 - En la forma de ingreso se especificó un título, un contenido y un nombre de archivo.
 - El nombre del archivo es válido.
 - El archivo existe en la ruta especificada.
 - El archivo ha sido definido en un mensaje previo.
 - El tamaño del archivo no es cero.
 - El tamaño del archivo no es mayor al tamaño máximo permitido para cada uno de los archivos definidos por el usuario.

- Resultados:**
- Un mensaje de error es presentado.
 - El mensaje no es registrado.

Escenario 9.5: *No se puede publicar el mensaje porque se desea adjuntar un archivo cuyo tamaño sobrepasa al tamaño máximo especificado para el usuario.*

- Asunciones:**
- El directorio de trabajo del paralelo existe.
 - En la forma de ingreso se especificó un título, un contenido y un nombre de archivo.
 - El nombre del archivo es válido.
 - El archivo existe en la ruta especificada.

- El archivo no ha sido definido ningún mensaje previo.
- El tamaño del archivo no es cero.
- El tamaño del archivo es mayor al tamaño máximo permitido para cada uno de los archivos definidos por el usuario.

Resultados: - Un mensaje de error es presentado.
- El mensaje no es registrado.

Caso de Uso 10: <i>Profesor publica una política de calificación</i>

Escenario 10.1: *El profesor publica un componente de calificación de manera exitosa.*

Asunciones: - Existen menos de cinco componentes de calificación ya definidas por parcial.
- El porcentaje de calificación definido para el nuevo componente, junto con los demás definidos en el parcial, no sobrepasa el 100%.

Resultados: - Un nuevo componente de calificación es registrado.
- Un mensaje de éxito es presentado con los datos registrados.

Escenario 10.2: *El profesor no puede publicar el componente de calificación porque ya existen cinco componentes de calificación definidos para ese parcial.*

Asunciones: - Existen cinco componentes de calificación ya definidos por parcial.
- El porcentaje de calificación definido para el nuevo componente, junto con los demás definidos en el parcial, no sobrepasa el 100%.

Resultados: - El componente de calificación es registrado.

- Un mensaje de error es presentado.

Escenario 10.3: *El profesor no puede publicar el componente de calificación porque la suma de los porcentajes de los componentes de calificación del parcial sobrepasa el 100%.*

Asunciones: - Existen menos de cinco componentes de calificación ya definidos por parcial.
 - El porcentaje de calificación definido para el nuevo componente, junto con los demás definidos en el parcial, sobrepasa el 100%.

Resultados: - El componente de calificación es registrado.
 - Un mensaje de error es presentado.

Caso de Uso 11: <i>Profesor publica notas de curso</i>

Escenario 11.1: *El profesor publica las calificaciones de los componentes definidos de manera exitosa.*

Asunciones: - El parcial seleccionado tiene componentes de calificación.

Resultados: - Las calificaciones son registradas.
 - Un mensaje de éxito es presentado.

Escenario 11.2: *El profesor no puede ingresar las calificaciones porque no existen componentes de calificaciones definidos para ese parcial.*

Asunciones: - El parcial seleccionado no tiene componentes de calificación.

Resultados: - Las calificaciones no son registradas.

- Un mensaje de error es presentado.

Caso de Uso 12: <i>Profesor migra material académico de un período a otro</i>

Escenario 12.1: *El profesor migra el material deseado exitosamente.*

- Asunciones:**
- El profesor está dictando materias en el período académico actual.
 - El directorio de trabajo de la materia de la que el profesor desea migrar el material existe.
 - El tipo de material que se desea migrar no ha sido migrado antes.

- Resultados:**
- La información del material seleccionado es registrada como del período académico actual.
 - Los archivos que son referenciados por dichos materiales, son copiados al directorio de la materia en el año-término actual.
 - Queda registrado cuál es el material que ha sido migrado.
 - Un mensaje de finalización de proceso es presentado.

Escenario 12.2: *El profesor desea migrar el material de una materia que no está dictando en el período actual.*

- Asunciones:**
- El profesor está dictando materias en el período académico actual.
 - El profesor no está dictando la materia de la cual desea migrar el material.
 - El tipo de material que se desea migrar no ha sido migrado antes.

- Resultados:**
- El material de uno de los períodos anteriores no es migrado al período actual.

- Un mensaje de finalización de proceso es presentado.

Escenario 12.3: *El profesor desea migrar el material de una materia que ya ha migrado anteriormente.*

- Asunciones:**
- El profesor está dictando materias en el período académico actual.
 - El directorio de trabajo de la materia de la que el profesor desea migrar el material existe.
 - El tipo de material que se desea migrar ha sido migrado antes.

- Resultados:**
- El material de uno de los períodos anteriores no es migrado al período actual.
 - Un mensaje de finalización de proceso es presentado.

Caso de Uso 13: <i>Persona reserva un recurso físico</i>

Escenario 13.1: *Una persona reserva un recurso físico de manera exitosa.*

- Asunciones:**
- El recurso que se desea solicitar existe.
 - El calendario de actividades del período académico actual existe.
 - El recurso físico se encuentra disponible en las horas y semanas solicitadas.

- Resultados:**
- Una solicitud de recurso físico es creada con estado de aprobada con los detalles correspondientes.
 - Un mensaje de éxito es presentado.

Escenario 13.2: *Una persona no reserva el recurso físico porque no se encuentra disponible.*

Asunciones: - El recurso que se desea solicitar existe.
 - El calendario de actividades del período académico actual existe.
 - El recurso físico no se encuentra disponible en alguna de las horas y/o semanas solicitadas.
 - El usuario no desea generar solicitud para lista de espera.

Resultados: - Ninguno.

Escenario 13.3: *La solicitud de recurso físico queda en lista de espera porque no se encuentra disponible.*

Asunciones: - El recurso que se desea solicitar existe.
 - El calendario de actividades del período académico actual existe.
 - El recurso físico no se encuentra disponible en alguna de las horas y/o semanas solicitadas.
 - El usuario no desea generar solicitud para lista de espera.

Resultados: - Una solicitud de recurso físico es creada con estado de pendiente (en lista de espera) con los detalles correspondientes.
 - Un mensaje de éxito es presentado.

Caso de Uso 14: <i>Estudiante consulta notas</i>

Escenario 14.1: *Un estudiante consulta las calificaciones obtenidas en los componentes de calificación.*

Asunciones: - Los componentes de calificación han sido definidos, al menos para una nota.

- Las calificaciones del estudiante han sido ingresadas.

Resultados: - Las calificaciones obtenidas por el estudiante son presentadas, así como las notas promedio obtenidas por el curso y el total obtenido por el estudiante.

Escenario 14.2: *Un estudiante no puede consultar las calificaciones obtenidas porque no existen definidos en el sistema los componentes de calificación para ninguna nota.*

Asunciones: - Los componentes de calificación no han sido definidos para ninguna nota.
- Las calificaciones del estudiante no han sido ingresadas.

Resultados: - Un mensaje es presentado indicando que no existen componentes de calificación definidos.

Escenario 14.3: *Un estudiante no puede consultar las calificaciones obtenidas porque no han sido ingresadas al sistema.*

Asunciones: - Los componentes de calificación han sido definidos, al menos para una nota.
- Las calificaciones del estudiante no han sido ingresadas.

Resultados: - Las calificaciones obtenidas por el estudiante son presentadas con cero.

Caso de Uso 15: <i>Persona customiza datos personales</i>

Escenario 15.1: *La información personal de una persona es modificada exitosamente.*

Asunciones: - Los datos fueron correctamente ingresados.
 - El nombre de usuario definido no ha sido especificado para ninguna otra persona en el mismo servidor de autenticación.

Resultados: - La información personal del usuario es actualizada.
 - Un mensaje de éxito es presentado.

Escenario 15.2: *La información personal de una persona no es modificada porque ha definido un servidor de autenticación y un nombre de usuario de acceso que ya existe para otra persona.*

Asunciones: - Los datos fueron correctamente ingresados.
 - El nombre de usuario definido ha sido especificado para ninguna otra persona en el mismo servidor de autenticación.

Resultados: - Un mensaje de error es presentado.
 - La información personal del usuario no es actualizada.

Caso de Uso 16: <i>Persona personaliza páginas HTML</i>

Escenario 16.1: *El usuario tiene un directorio de configuración de páginas HTML, selecciona una plantilla y el cambio es realizado exitosamente.*

Asunciones: - Existen plantillas definidas en el sistema.
 - El usuario conectado tiene un directorio de configuración de páginas HTML.

- Resultados:**
- Todos los archivos de configuración del usuario son modificados en base al modelo de la plantilla seleccionada.
 - Un mensaje de éxito es presentado.

Escenario 16.2: *El usuario no tiene un directorio de configuración de páginas HTML, selecciona una plantilla y el cambio es realizado exitosamente.*

- Asunciones:**
- Existen plantillas definidas en el sistema.
 - El usuario conectado no tiene un directorio de configuración de páginas HTML.

- Resultados:**
- Un directorio es creado y los archivos de configuración de páginas HTML copiados en el mismo.
 - Todos los archivos de configuración del usuario son modificados en base al modelo de la plantilla seleccionada.
 - Un mensaje de éxito es presentado.

Caso de Uso 17: <i>Usuario accede al sistema</i>

Escenario 17.1: *Acceso otorgado a persona cuyo nombre de usuario es único, con contraseña correcta y sin conexiones abiertas.*

- Asunciones:**
- El usuario pertenece a un profesor, alumno o ayudante.
 - La contraseña ingresada es correcta.
 - El usuario no tiene registradas conexiones abiertas.

- Resultados:**
- El acceso al sistema es otorgado.
 - Las variables de ambiente son seteadas en base a la información del usuario logeado.
 - Son listadas las materias asociadas al usuario.

Escenario 17.2: *Acceso otorgado a persona cuyo nombre de usuario es único, con contraseña correcta y con conexiones abiertas.*

Asunciones: - El usuario pertenece a un profesor, alumno o ayudante.
- La contraseña ingresada es correcta.
- El usuario tiene registradas conexiones abiertas.

Resultados: - El acceso al sistema es otorgado.
- Las variables de ambiente son seteadas en base a la información del usuario logeado.
- Son listadas las materias asociadas al usuario.

Escenario 17.3: *Acceso otorgado a persona cuyo nombre de usuario no es único con contraseña correcta.*

Asunciones: - El usuario pertenece a varias personas (profesores, alumnos o ayudantes).
- El usuario selecciona el servidor de autenticación correcto contra el cual validar el usuario y la contraseña ingresadas.
- La contraseña ingresada es correcta.
- El usuario no tiene registradas conexiones abiertas.

Resultados: - El acceso al sistema es otorgado.
- Las variables de ambiente son seteadas en base a la información del usuario logeado.
- Son listadas las materias asociadas al usuario.

Escenario 17.4: *Acceso no otorgado a persona cuyo nombre de usuario no existe.*

Asunciones: - El usuario no pertenece a ningún profesor.
- El usuario no pertenece a ningún alumno.

Resultados: - El acceso al sistema no es otorgado.
- Es presentado un mensaje de error de acceso.

Escenario 17.5: *Acceso no otorgado a persona cuyo nombre de usuario tiene contraseña incorrecta.*

Asunciones: - El usuario pertenece sólo a un profesor o alumno.
- La contraseña ingresada es incorrecta.

Resultados: - El acceso al sistema no es otorgado.
- Es presentado un mensaje de error de acceso.

Escenario 17.6: *Acceso no otorgado a persona cuyo nombre de usuario no es único con contraseña incorrecta.*

Asunciones: - El usuario pertenece a varios profesores o alumnos.
- La persona selecciona el servidor correcto contra el cual validar la contraseña ingresada.
- La contraseña es incorrecta.

Resultados: - El acceso al sistema no es otorgado.
- Es presentado un mensaje de error de acceso.

Caso de Uso 18: *Administrador ingresa un nuevo estudiante*

Escenario 18.1: *Un estudiante es ingresado exitosamente.*

Asunciones: - No existe otro estudiante con el número de identificación ingresado.
- Los datos fueron ingresados correctamente.
- El nombre de usuario definido no ha sido especificado para ninguna otra persona en el mismo servidor de autenticación.

Resultados: - El nuevo estudiante es registrado.
- Un mensaje de éxito es presentado.

Escenario 18.2: *Un estudiante no es ingresado porque el número de identificación pertenece a otro estudiante.*

Asunciones: - Existe otro estudiante con el número de identificación ingresado.
 - Los datos fueron ingresados correctamente.
 - El nombre de usuario definido no ha sido especificado para ninguna otra persona en el mismo servidor de autenticación.

Resultados: - El nuevo estudiante no es registrado.
 - Un mensaje de error es presentado.

Escenario 18.3: *Un estudiante no es ingresado porque se ha definido un servidor de autenticación y un nombre de usuario de acceso que ya existe para otra persona.*

Asunciones: - No existe otro estudiante con el número de identificación ingresado.
 - Los datos fueron ingresados correctamente.
 - El nombre de usuario definido ha sido especificado para ninguna otra persona en el mismo servidor de autenticación.

Resultados: - Un mensaje de error es presentado.
 - El nuevo estudiante no es registrado.

Caso de Uso 19: *El administrador cambia de estado una reservación*

Escenario 19.1: *Una solicitud aprobada es puesta en lista de espera exitosamente.*

Asunciones: - La solicitud de recurso existe con estado de aprobada.

- Resultados:** - La solicitud es actualizada como en lista de espera.
- Un mensaje de éxito es presentado.

Escenario 19.2: *Una solicitud en lista de espera es aprobada exitosamente.*

- Asunciones:** - La solicitud de recurso existe en lista de espera.
- El recurso físico de la solicitud se encuentra disponible en las horas y semanas solicitadas.

- Resultados:** - La solicitud es actualizada como aprobada.
- Un mensaje de éxito es presentado.

Escenario 19.3: *Una solicitud en lista de espera no es aprobada porque el recurso físico no está disponible en todos los horarios de la solicitud.*

- Asunciones:** - La solicitud de recurso existe en lista de espera.
- El recurso físico no se encuentra disponible en alguna de las horas y/o semanas solicitadas.

- Resultados:** - La solicitud no es actualizada.
- Un mensaje es presentado indicando que no se pudo actualizar la solicitud.

Caso de Uso 20: <i>Administrador migra información al sistema</i>

Escenario 20.1: *Los datos de migración se insertan con éxito y los directorios de trabajo son creados.*

- Asunciones:** - Los archivos existen en las rutas especificadas.
- Los tipos de datos existentes en los archivos coinciden con los tipos de datos de las columnas definidas en las tablas.

- El número de datos en los archivos es el mismo al número de columnas existentes en las tablas.

- Resultados:**
- Los nuevos registros son insertados en las diferentes tablas.
 - Un mensaje de éxito es presentado.

Escenario 20.3: *Los datos de migración no son insertados porque estos son inconsistentes con las estructuras de las tablas.*

- Asunciones:**
- Los archivos existen en las rutas especificadas.
 - Los tipos de datos existentes en los archivos no coinciden con los tipos de datos de las columnas definidas en las tablas.
 - El número de datos en los archivos no es el mismo al número de columnas existentes en las tablas.

- Resultados:**
- Los nuevos registros no son insertados en las diferentes tablas.
 - Un mensaje de error es presentado.

3.3. Definición de Clases

Una clase describe un grupo de objetos con propiedades similares (atributos), comportamiento común (operaciones), relaciones comunes a otros objetos, y semántica común. Los objetos de una clase tienen las mismas plantillas de atributos y comportamiento. La mayoría de los objetos obtienen su individualidad de las diferencias en sus valores de atributos y relaciones a otros objetos. Sin embargo, es posible tener objetos con valores de atributos y relaciones idénticos.

A continuación, existe una breve descripción de cada una de las clases que se consideran de importancia dentro del dominio del problema de Cursos Web.

Nombre de Clase	Descripción
AdministradorDeArchivosFml	Conoce sobre todos los archivos fml necesarios para customizar la apariencia del sistema.
GeneradorDeArbolParaForo*	Tiene como función el poder construir la estructura del foro de discusión.
BotonDeModeloEnFml*	Mantiene la referencia de los nombres de los prefijos de los botones y etiquetas que serán usados en cada uno de los modelos
GeneradorDeCalendario*	Esta clase se encarga de formar un mes calendario
ConexionABase*	Representa una conexión a la base y se encarga de conectarse o desconectarse de la misma
Customizador	Realiza los cambios necesarios en todos los archivos fml (de formato) de las páginas Web de un usuario, en base a los archivos fml definidos para el modelo pre-definido seleccionado por el usuario.
EspolAlumno	Estudiante registrado en los diferentes cursos.
EspolAlumnoPorParalelo	Mantiene la relación entre un estudiante y un paralelo (materia registrada).
EspolAnuncio	Mantiene la información de un anuncio publicado.
EspolCalificacionDeAlumno	Su función es mantener la nota obtenida por un estudiante en una política de calificación específica (tarea, deber, lección, examen, etc.) y determinar cuál es la nota final de acuerdo al porcentaje que representa de la nota parcial (a través de la política de calificación).
EspolCurriculum	Mantiene información personal de un usuario.
EspolDetalleDeSolicitudDeRecurso	Especifica un momento exacto (semana, día y hora) en el que un recurso es solicitado.
EspolDiaCalendarioDelProfesor	Cada instancia de esta clase contendrá la información necesaria sobre un evento en un día de calendario definido por el

	profesor.
EspolElementoDelPeriodoAcademico	Contiene los atributos comunes que mantienen todas las clases de objetos relacionados a un paralelo.
EspolEmpleado	Representa a un empleado de la institución que puede ser docente o administrativo.
EspolInformacionDelProfesor	Mantiene información publicada por el profesor, no considerada ni anuncio ni política.
EspolMateria	Representa una materia.
EspolMigracionDeMaterial	Controla el material que ha sido migrado por un profesor de un curso de algún período anterior al actual.
EspolNombreDeParcial	Mantiene los parciales considerados en un período académico y el nombre que toman dentro del sistema. Por ejemplo: parcial, final y mejoramiento.
EspolParalelo	Curso que estará formado por un profesor, ayudante(s) y alumnos en una materia y número de paralelo específicos.
EspolParametrosGenerales	Mantiene los datos generales que serán usados en el sistema, ya sea en procesos o para considerar valores por omisión.
EspolPersona	Profesor o trabajador administrativo en la institución.
EspolPoliticaDeCalificacion	Es un componente de calificación en la nota parcial, final o mejoramiento.
EspolPoliticaDelProfesor	Política definida para el curso.
EspolProceso*	Controla las conexiones de cada usuario en el sistema.
EspolRecurso	Recurso físico que podrá ser reservado.
EspolRespuestaDeTarea	Contiene el texto o archivo con los que un estudiante responde a una tarea publicada.
EspolSemanaDeCalendarioDelPeriodo	Identifica a una semana dentro del período académico.
EspolServidor*	Servidor de autenticación, contra el cual se valida un nombre de usuario y contraseña.
EspolSolicitudDeRecurso	Solicitud de un recurso físico.
EspolTarea	Tarea publicada que los estudiantes deben responder.
EspolTemaDiscusion	Mensaje publicado en el foro de discusión.
EspolTipoRecurso	Tipo de recurso que clasifica a cada uno de los recursos físicos existentes.
EspolUnidadAcademica	Unidad académica, que conoce de los recursos físicos que tiene.

EspolVario	Recurso académico, publicado en forma de enlace o archivo.
File*	Utilizado para verificar el tamaño de los archivos que se desean publicar en el sistema.
HtmlAPI	Responde sobre cualquiera de los datos de configuración del sistema, necesarios para la generación de las páginas HTML.
HtmlError	Genera el código HTML de una página de error.
HtmlMetaformato	Genera el código HTML de las páginas del sistema, construidas en base al metaformato.
Mail	Se encarga de los envíos de mails a los integrantes del curso, cuando son publicados anuncios, recursos y tareas.
ObjetoPersistente*	Tiene como función realizar todas las operaciones a la base de datos. Es la clase de la que heredan todos los objetos persistentes.
PartesReusables*	Genera código HTML de información que es presentada en la mayoría de las páginas HTML.
Tools	Clase abstracta que realiza tareas varias.
ToolsCookie	Devuelve información sobre los valores de las variables de sesión del sistema.

* Estas clases fueron creadas en la etapa de diseño o implementación, por lo tanto no aparecen en los modelos de análisis del sistema.

Tabla II Definición de clases

3.4. Diagrama de interacción de objetos

Los diagramas de interacción de objetos (DIOs) definen cómo los objetos en el sistema colaboran para producir el resultado de los escenarios. Cada DIO describe las responsabilidades de los objetos

participantes y las interacciones entre ellos para determinar las salidas de cada escenario.

El escenario provee las especificaciones (como una caja negra) sobre el qué debe hacer el sistema. Una vista en detalle (una caja blanca) de cómo los objetos deben proveer la funcionalidad definida en el escenario puede ser expresada a través de una secuencia de interacción de objetos.

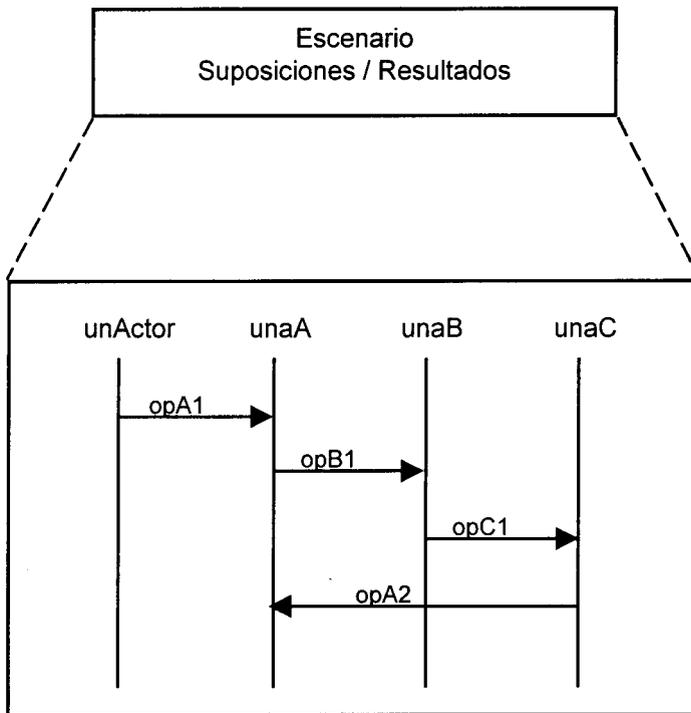


Figura 3.2 Vista de un escenario

La interacción entre objetos es documentada en términos de mensajes enviados, secuencia de mensajes y de control.

En un diagrama de interacción de objetos:

- Se representa el tiempo en forma vertical. Tiempo cero (inicio) está en el tope superior.
- Cada objeto (no clase) que participa está representado por una línea vertical etiquetada.
- Las interacciones están representadas con líneas horizontales (flujo de mensajes).
- Se lee el DIO de arriba hacia abajo para ver la secuencia de las interacciones.
- Se utilizan rectángulos para representar el control.

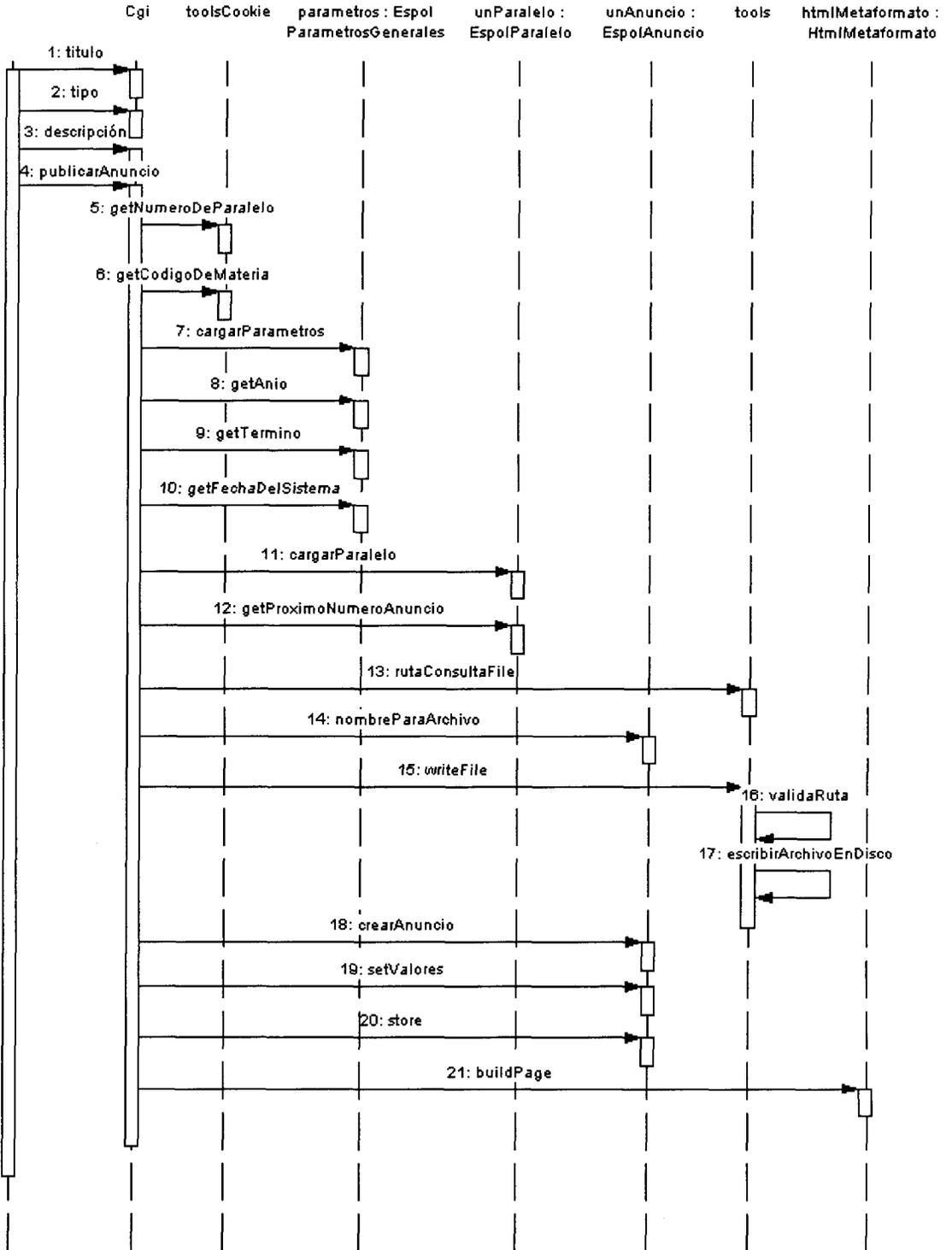
Al hacer un DIO se deben tomar en cuenta las siguientes consideraciones:

- Modelar objetos no clases.
- Definir el alcance correctamente, cada DIO representa el resultado de un escenario.

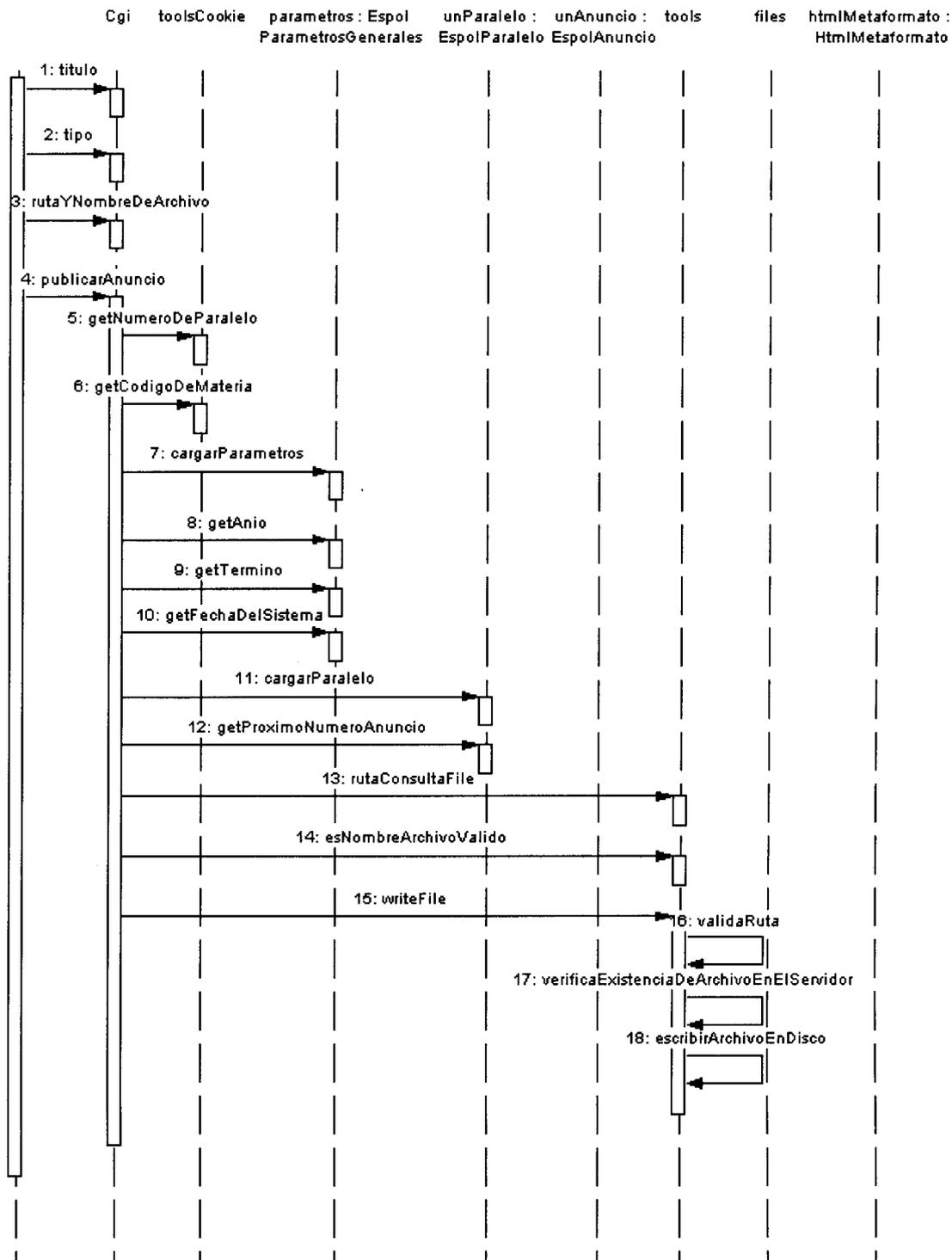
- No incluir todos los objetos conocidos en el sistema, sólo el grupo de objetos relevantes.
- No se preocupe sobre cómo los objetos son creados o destruidos, ni cómo salen o entran en memoria. Si es necesario dibujar una línea para representar un framework de persistencia.
- Las interacciones en el DIO a nivel de análisis representan las responsabilidades de las clases.
- No trate de diseñar todavía.

Los diagramas de interacción de objetos del sistema se encuentran a continuación:

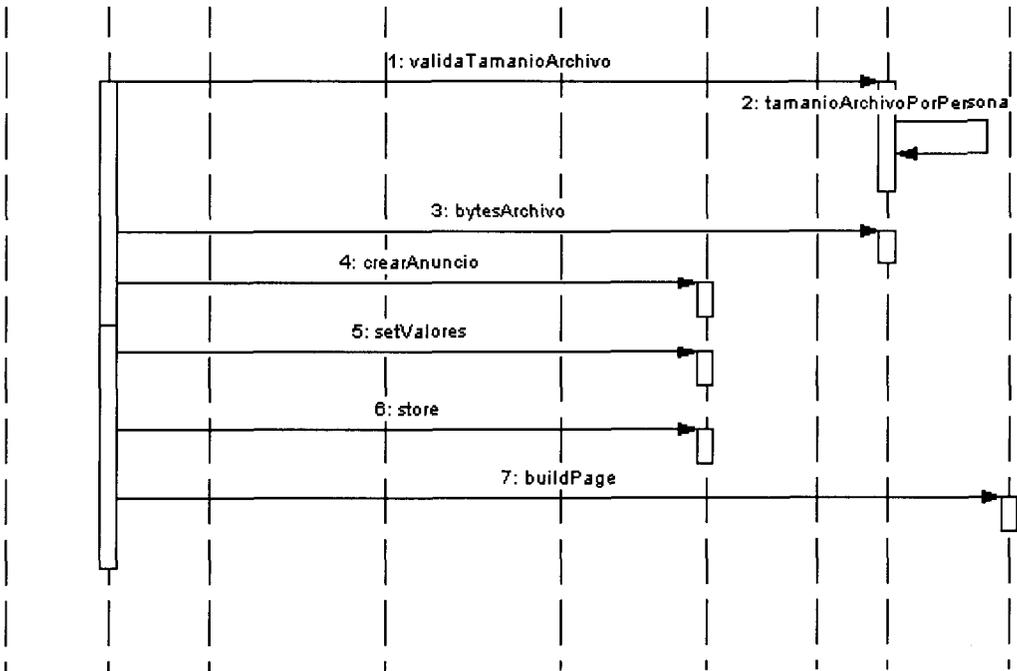
Escenario 1.1: El anuncio se publica en forma de texto exitosamente.



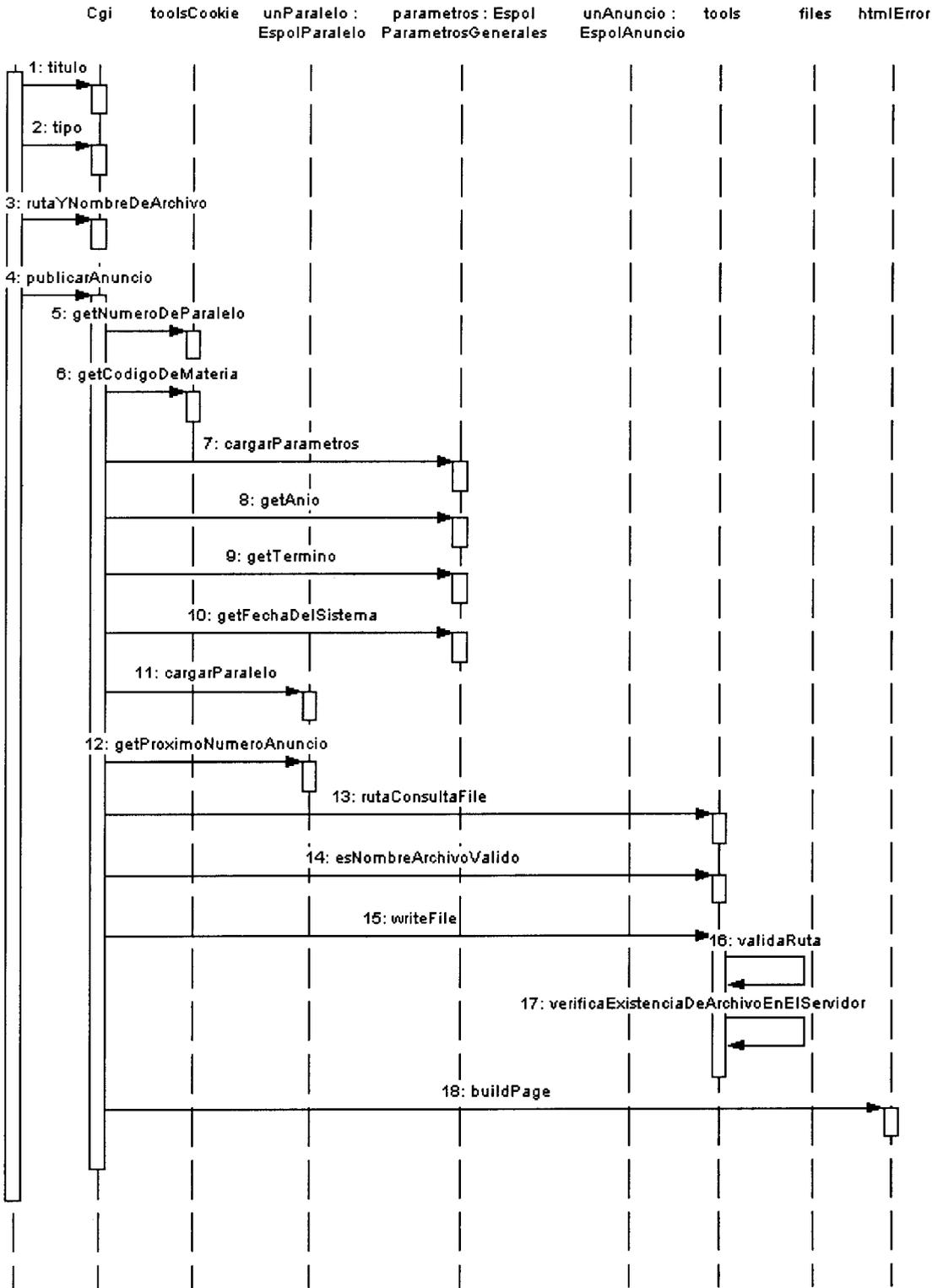
Escenario 1.2: El anuncio se publica en forma de archivo exitosamente.



Cgi toolsCookie parametros : Espol unParalelo : unAnuncio : tools files htmlMetaformato :
 ParametrosGenerales EspolParalelo EspolAnuncio HtmlMetaformato

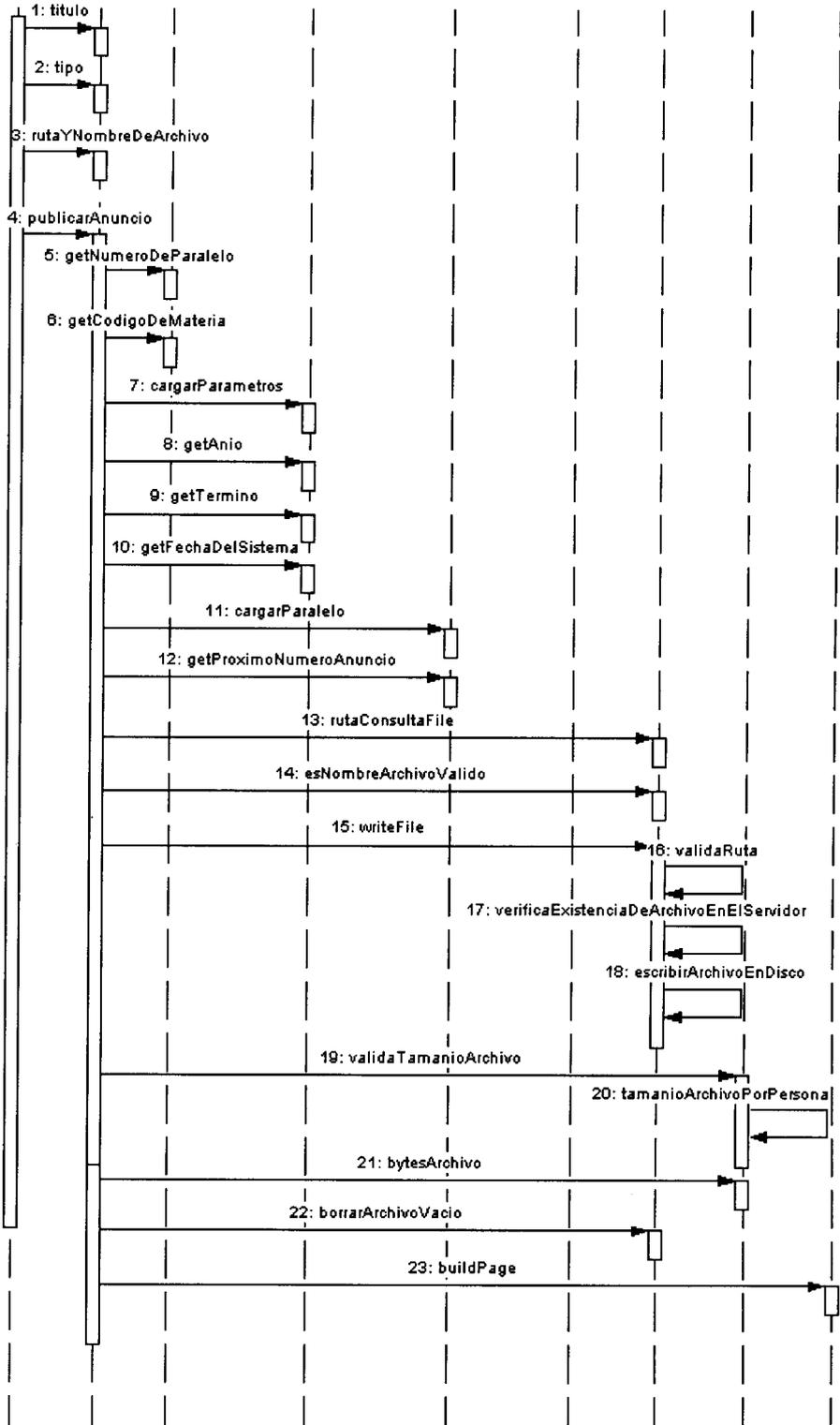


Escenario 1.3 No se puede publicar el anuncio en forma de archivo porque éste ya existe en el servidor.

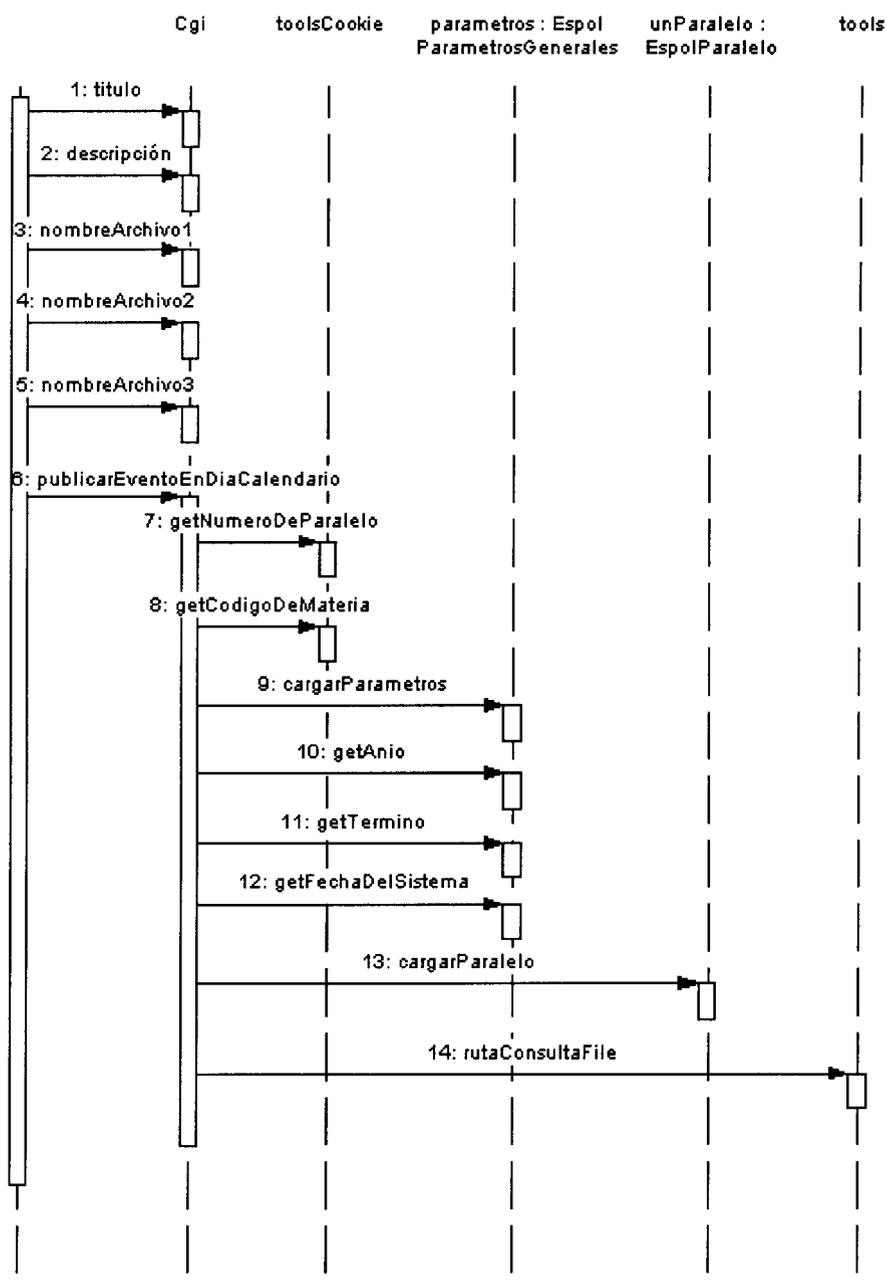


Escenario 1.4 No se puede publicar el anuncio en forma de archivo porque el tamaño de éste sobrepasa al tamaño máximo especificado para el usuario.

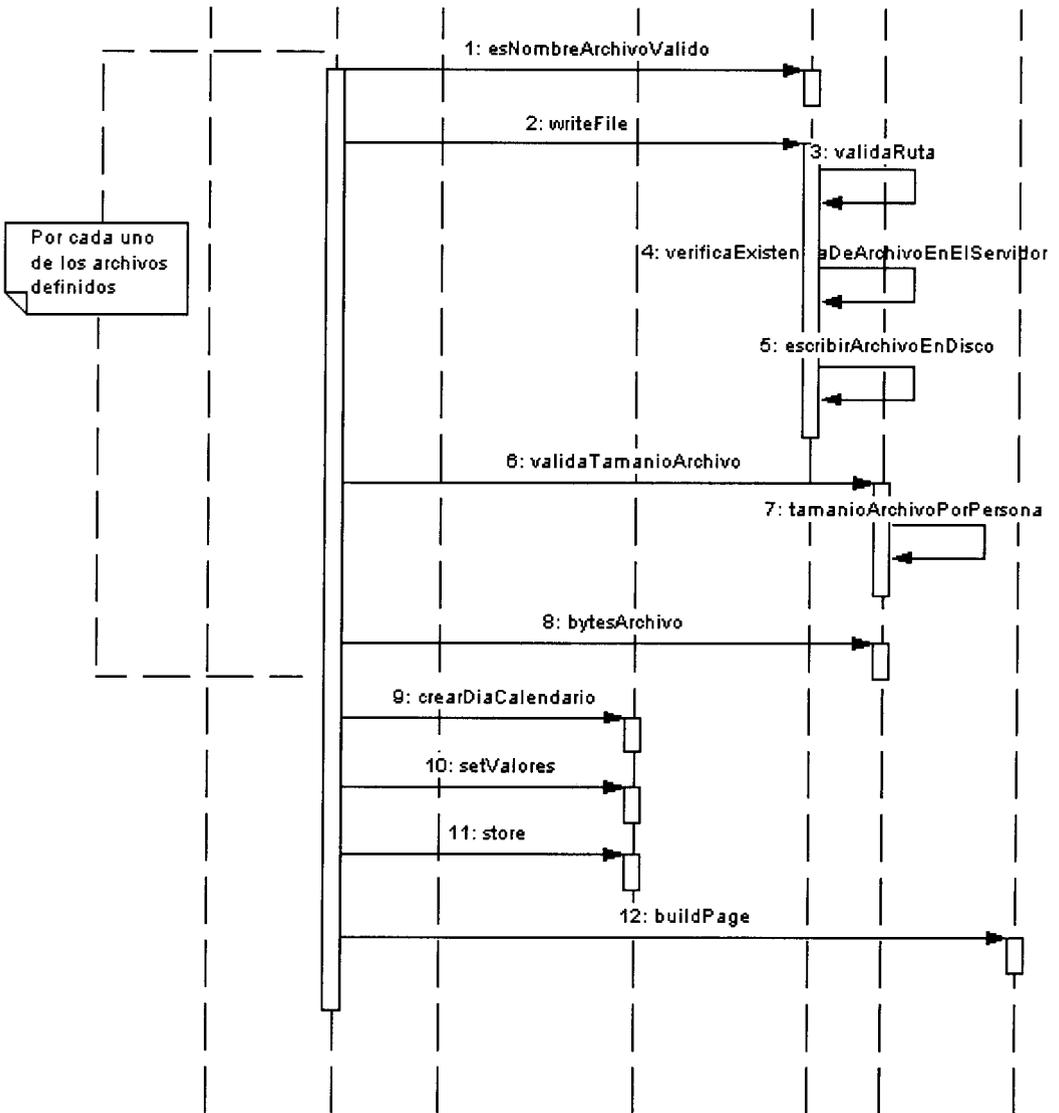
Cgi toolsCookie parametros: Espol unParalelo : unAnuncio : tools files htmlError
ParametrosGenerales EspolParalelo EspolAnuncio



Escenario 2.1: El evento en un día de calendario se publica en forma exitosa.

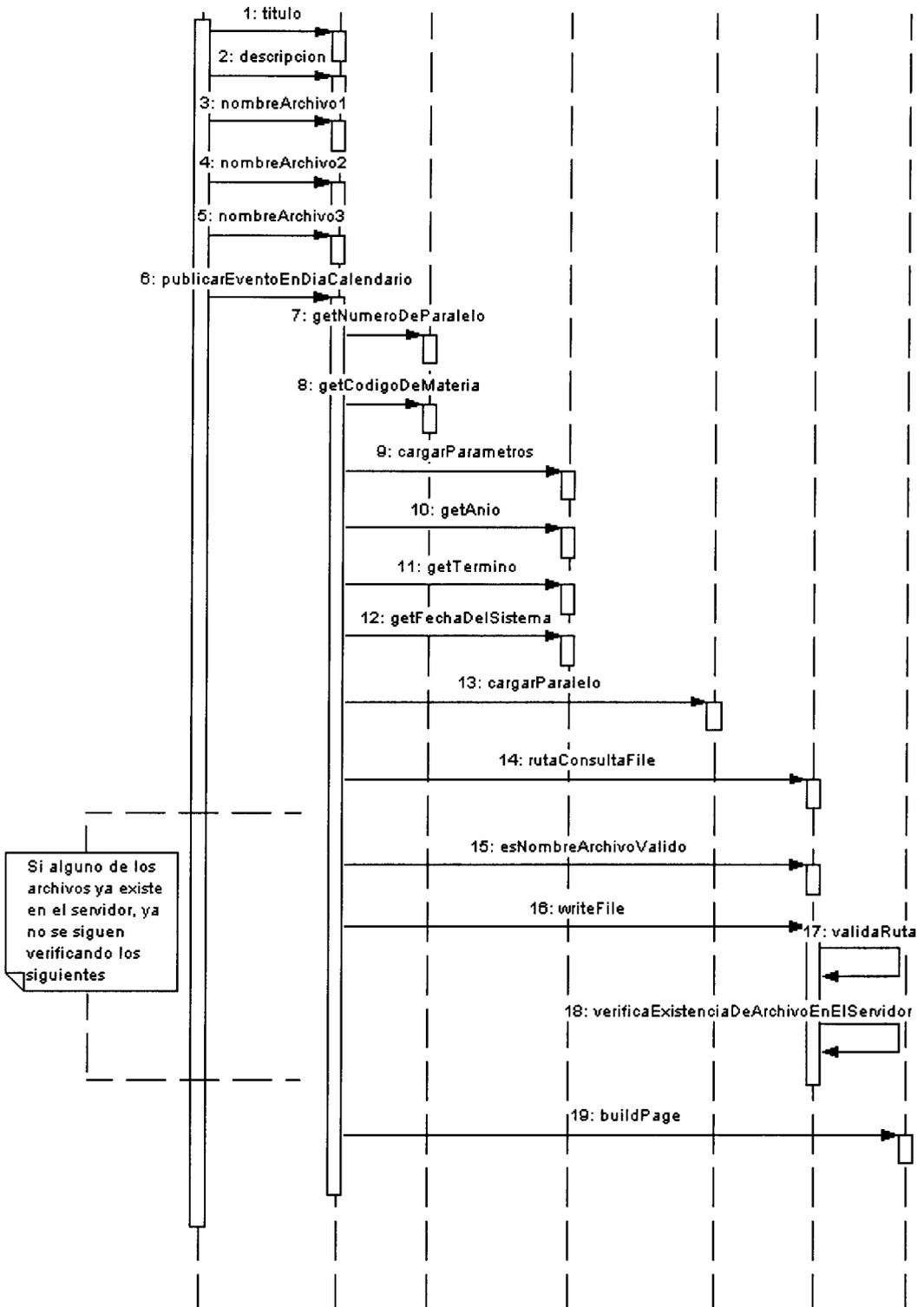


Cgi toolsCookie unDiaCalendario : Espol tools files htmlMetaformato :
 DiaCalendarioDelProfesor HtmlMetaformato

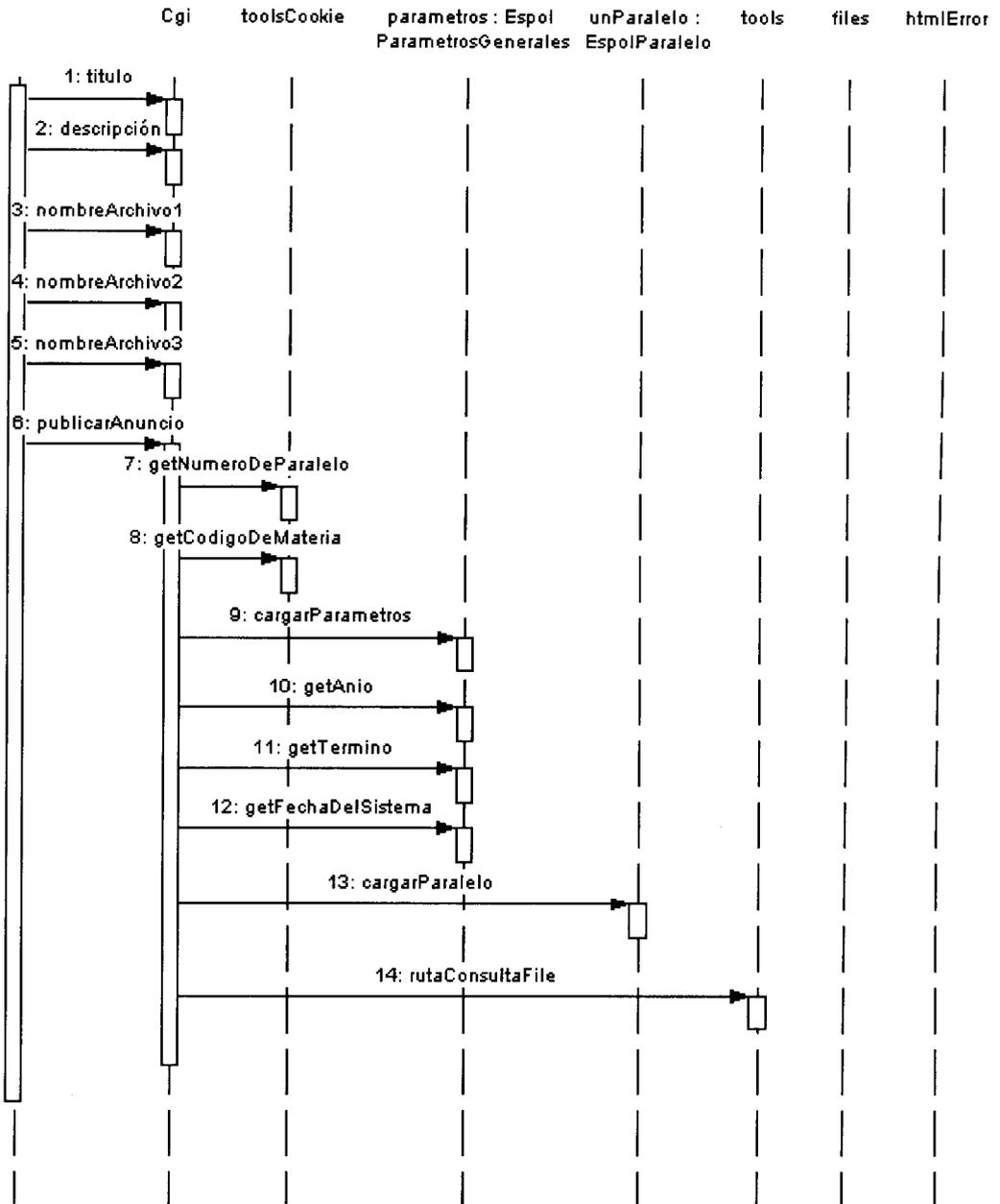


Escenario 2.2 No se puede publicar el evento en un día de calendario porque uno de los archivos ya existe en el servic

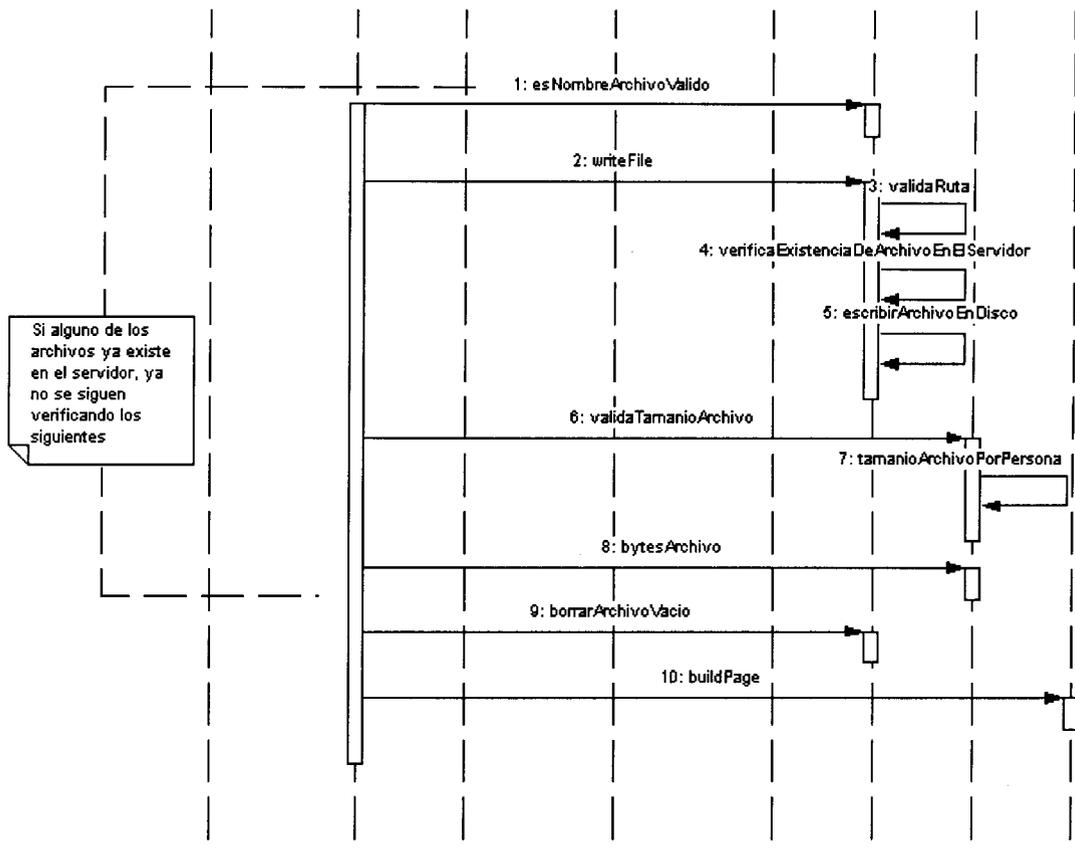
Cgi toolsCookie parametros: Espol unParalelo: tools htmlError
ParametrosGenerales EspolParalelo



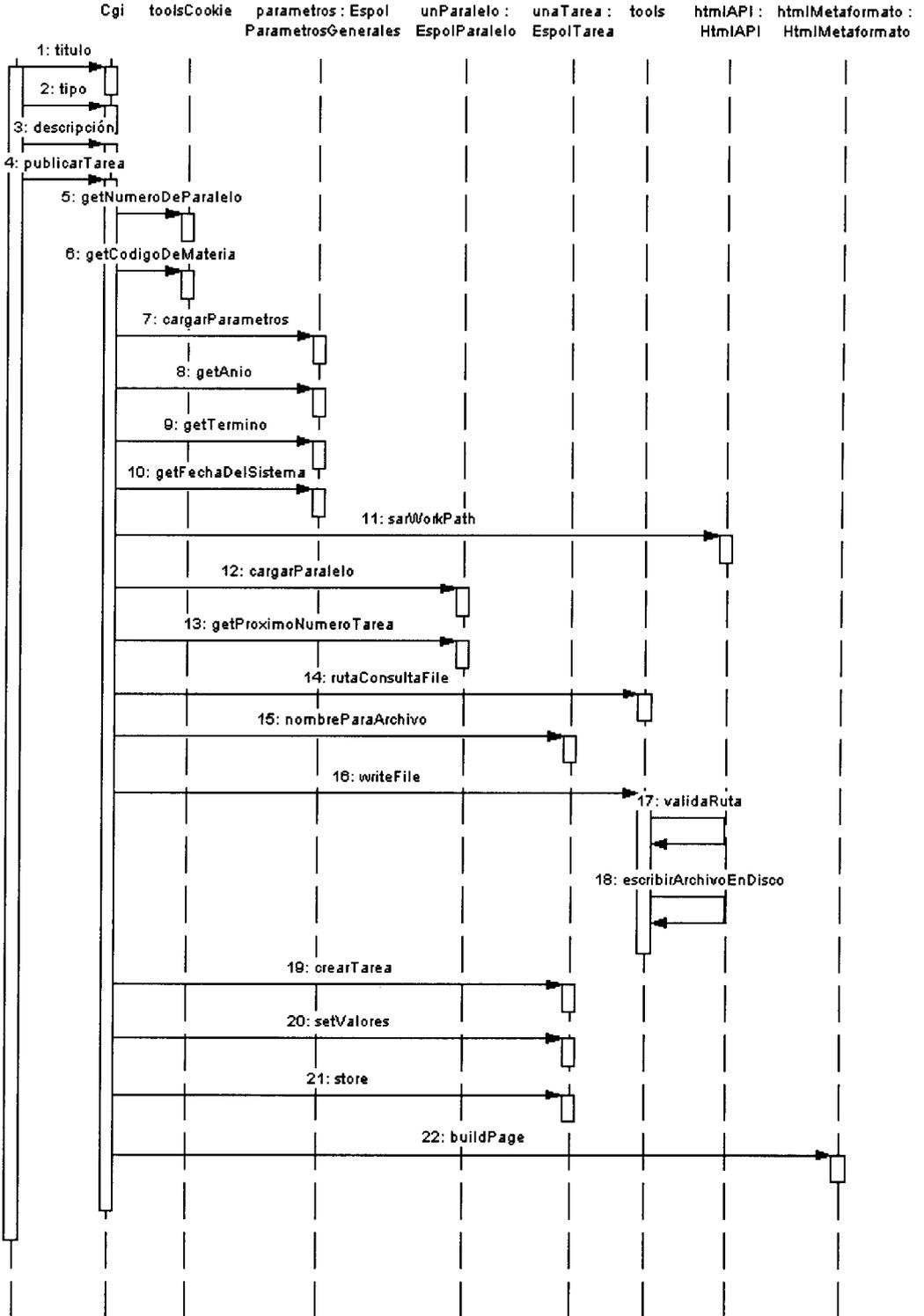
Escenario 2.3 No se puede publicar el evento en un día de calendario porque el tamaño de uno de los archivos sobrepasa al tamaño máximo especificado para el usuario.



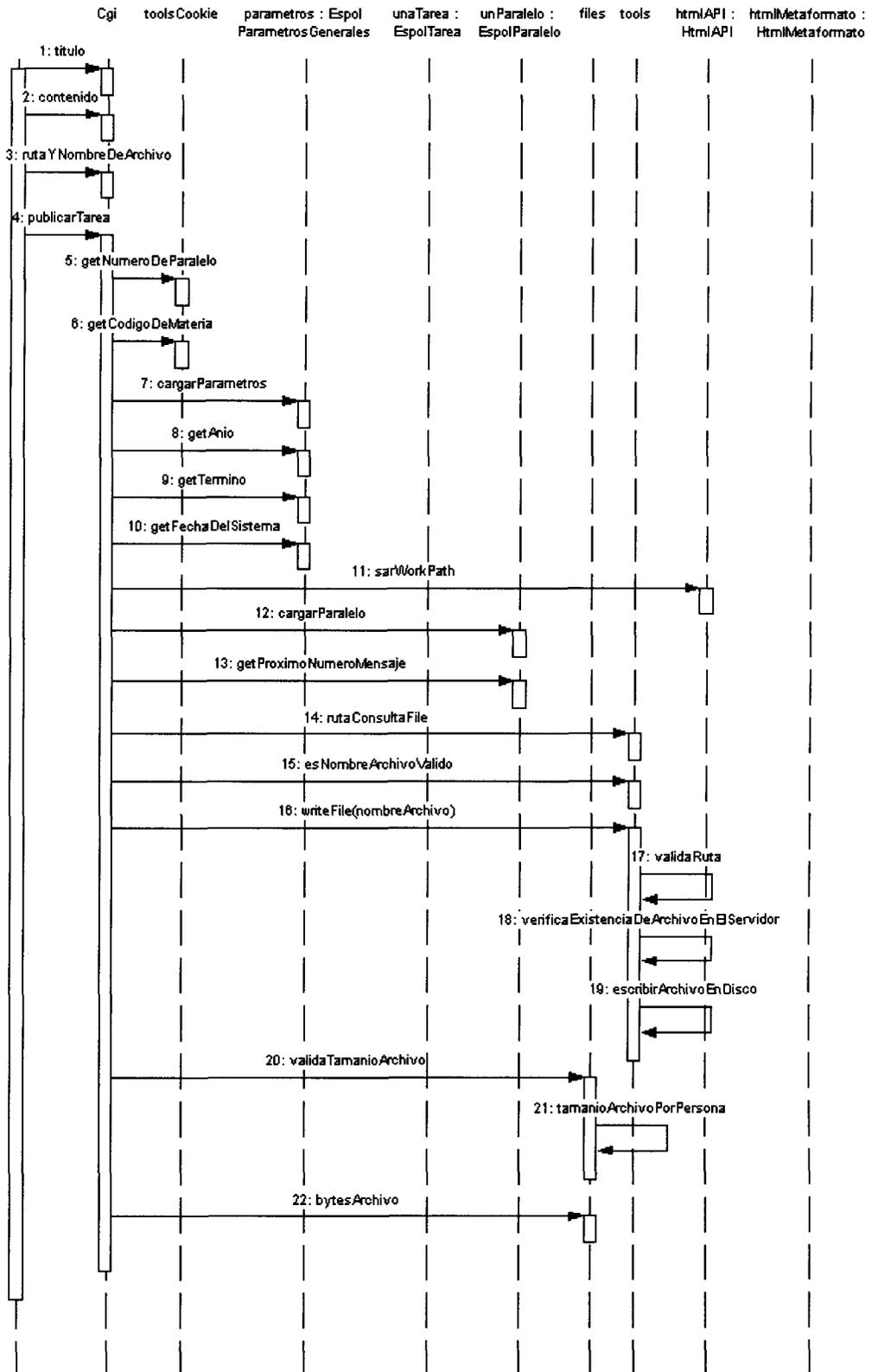
Cgi toolsCookie parametros : Espol unParalelo : tools files htmlError
 Parametros Generales EspolParalelo



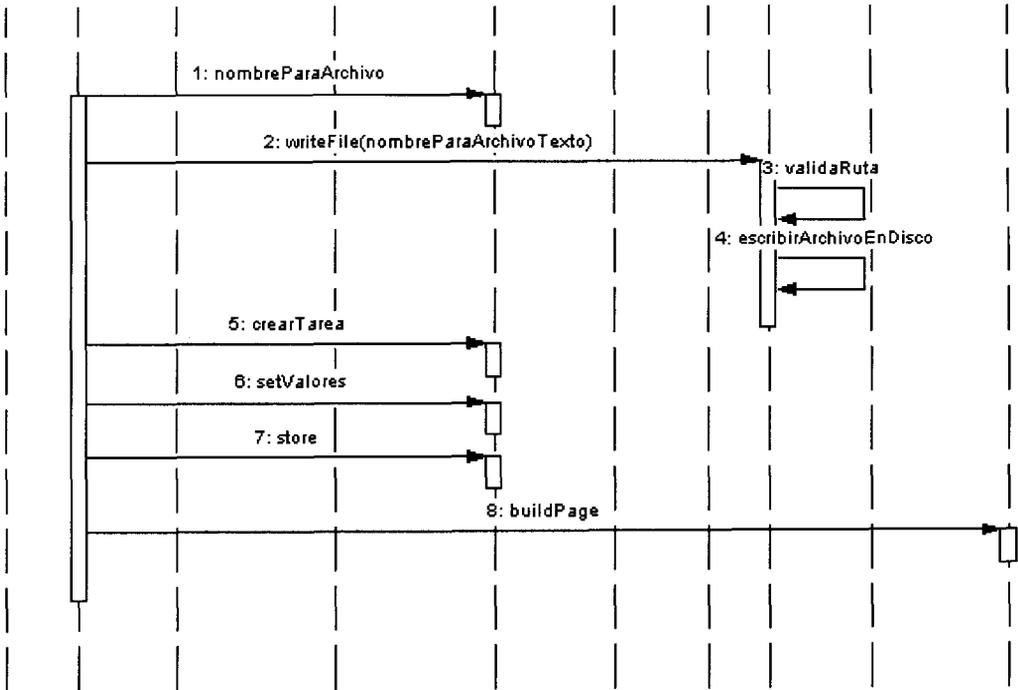
Escenario 3.1: La tarea se publica en forma de texto exitosamente.



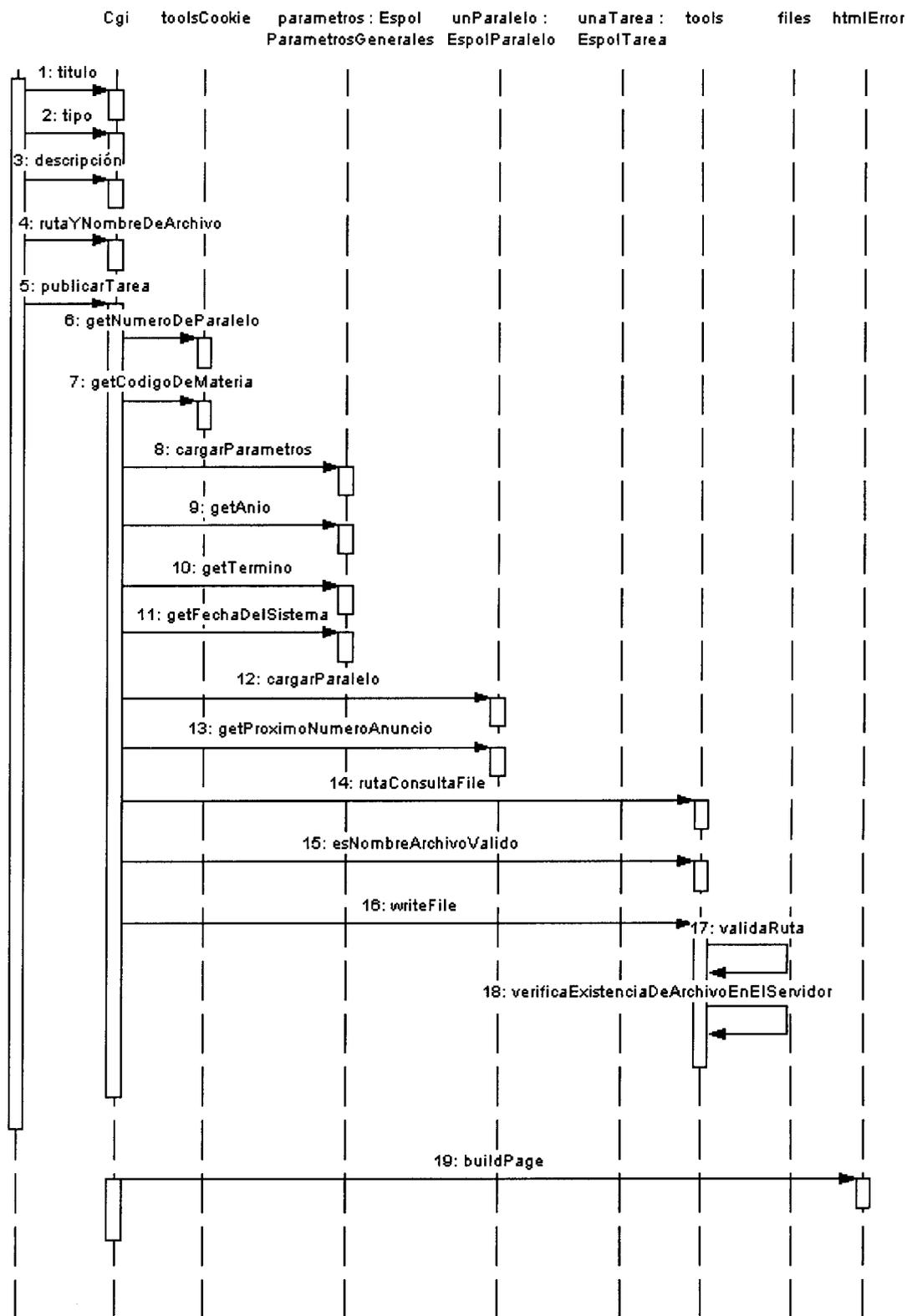
Escenario 3.2: La tarea se publica con un archivo adjunto exitosamente.



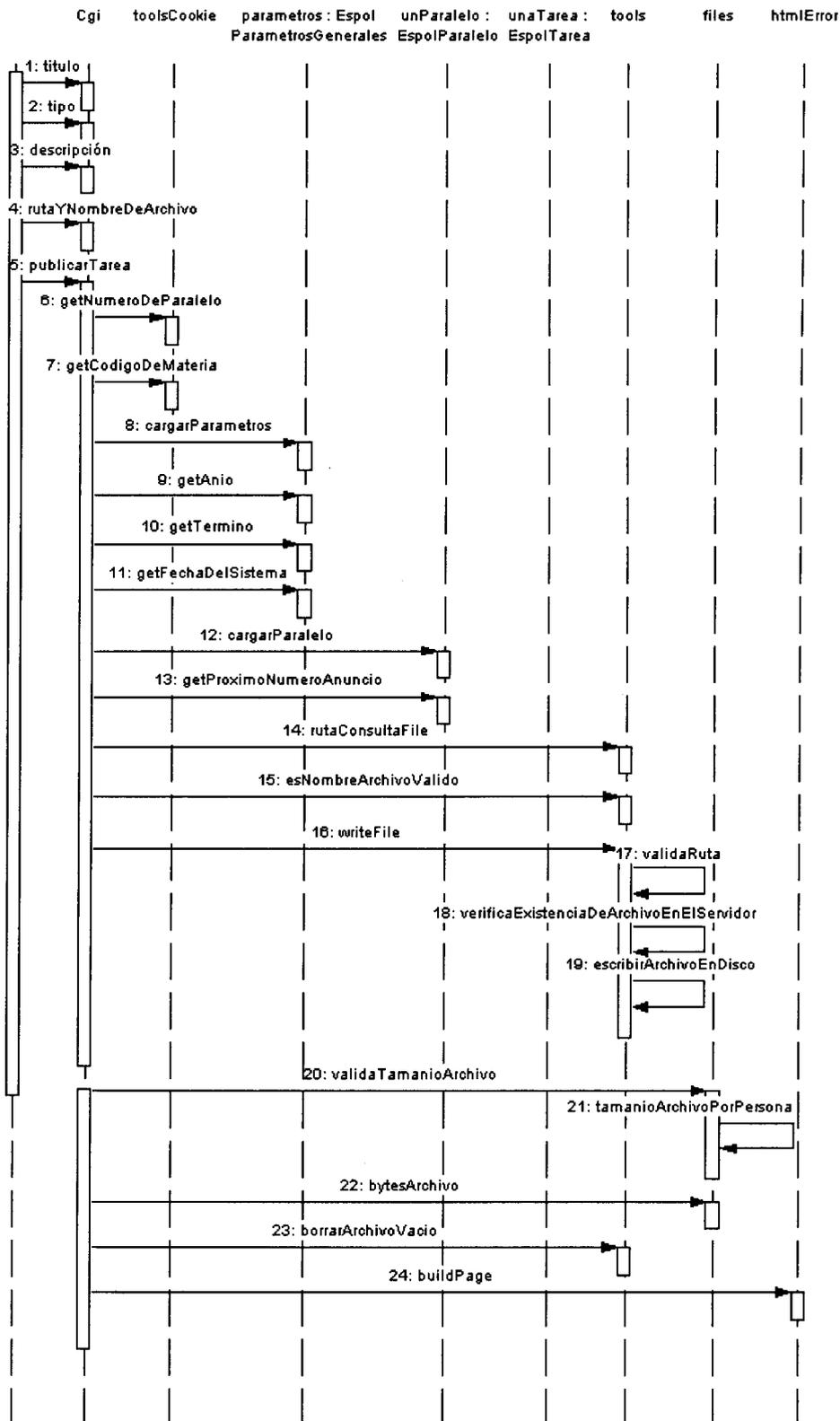
Cgi toolsCookie parametros:Espol unaTarea: unParalelo: files tools htmlAPI: htmlMetaformato:
 ParametrosGenerales EspolTarea EspolParalelo HtmlAPI HtmlMetaformato



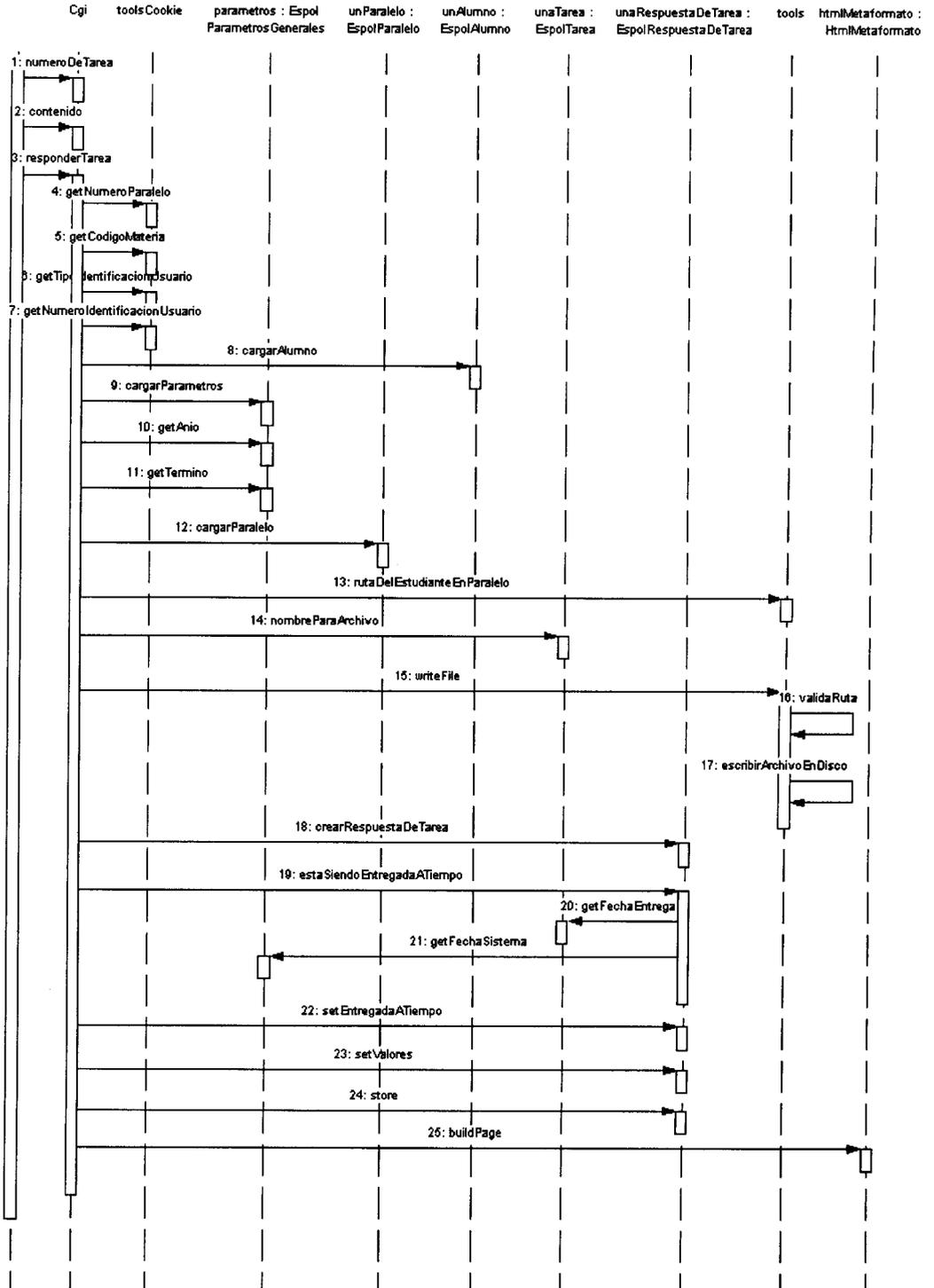
Escenario 3.3 No se puede publicar la tarea porque se desea adjuntar un archivo que ya existe en el servidor.



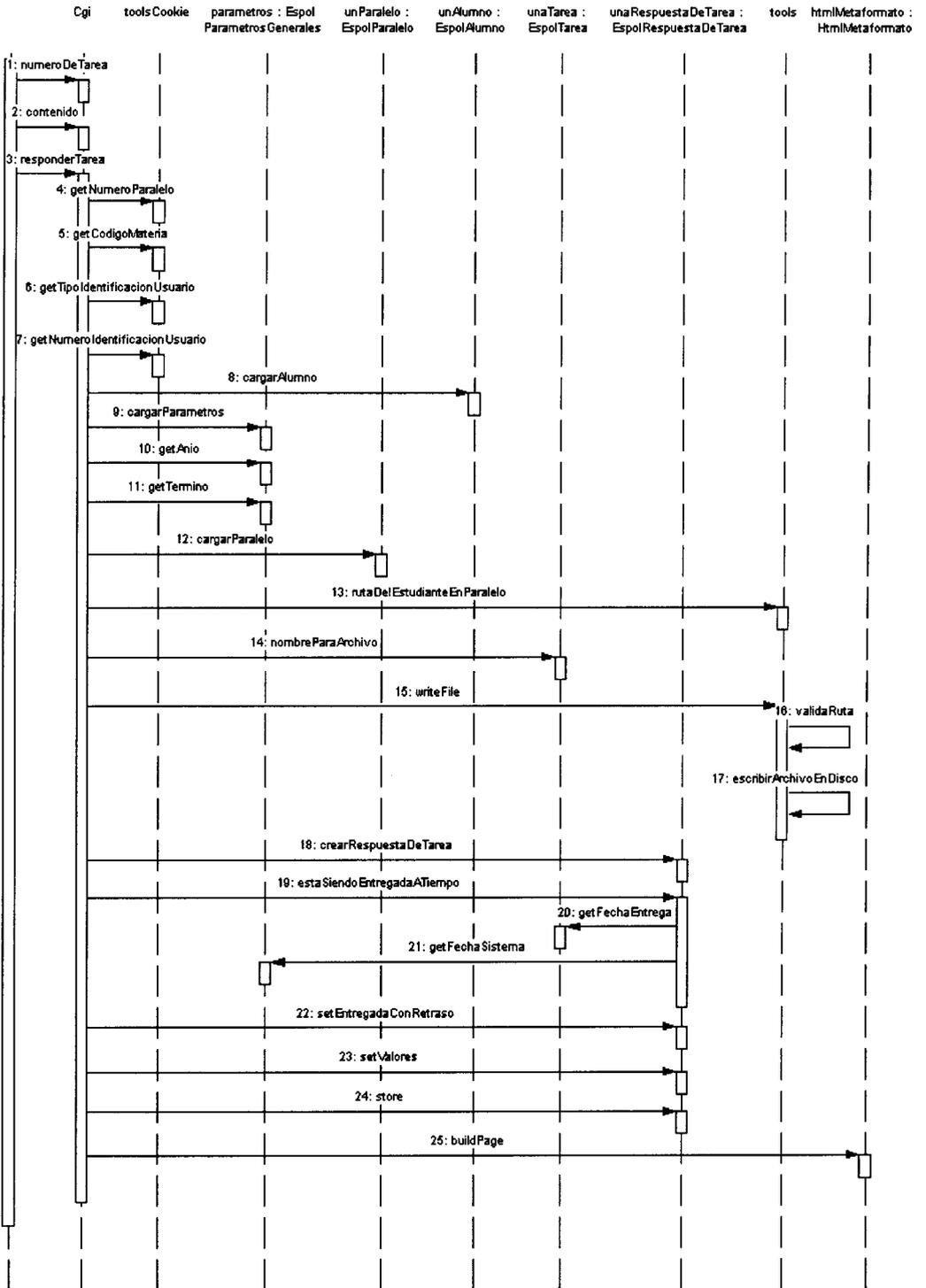
Escenario 3.4 No se puede publicar la tarea porque se desea adjuntar un archivo cuyo tamaño sobrepasa al tamaño máximo especificado para el usuario.



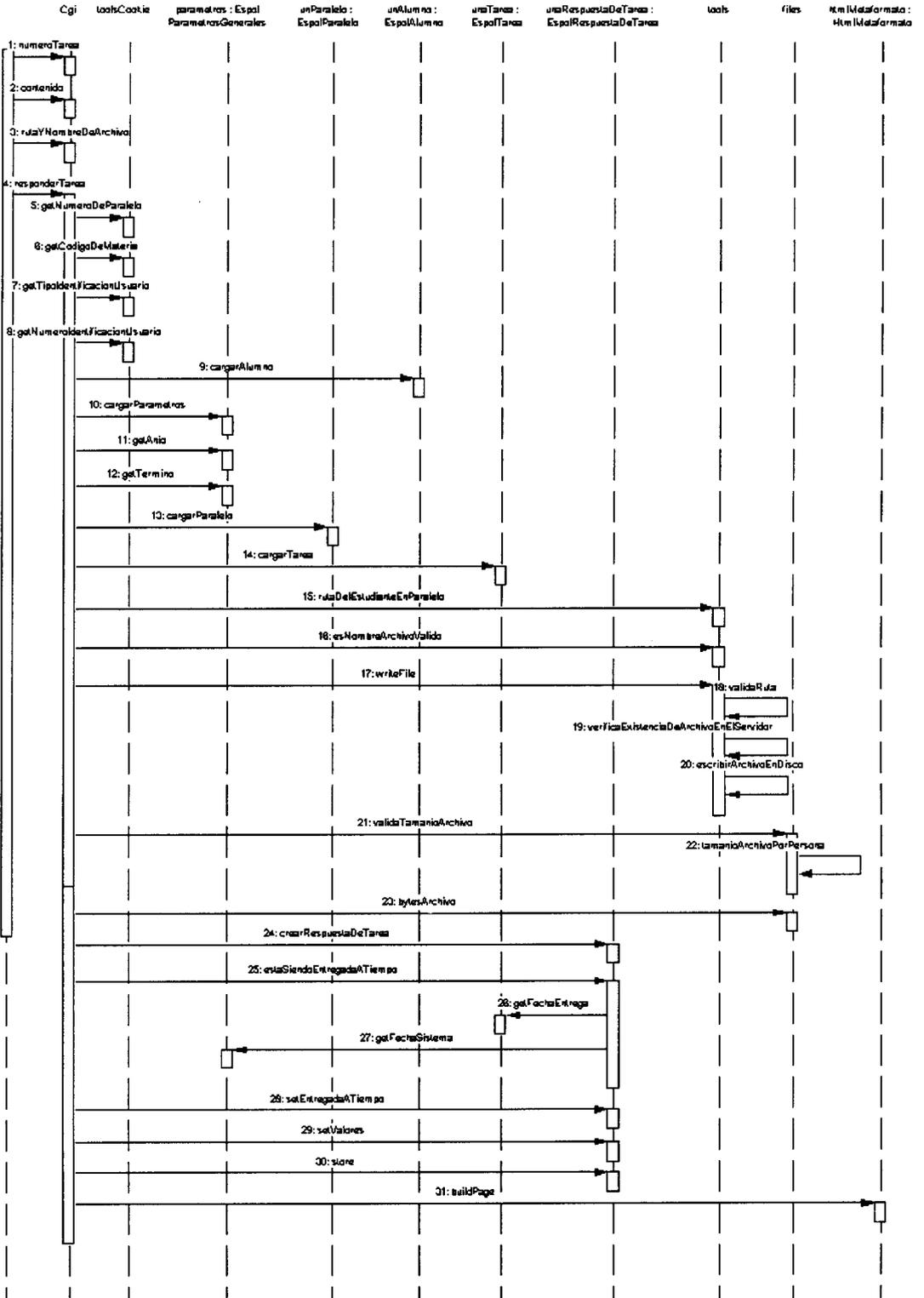
Escenario 4.1: La tarea es respondida a tiempo en forma de texto exitosamente.



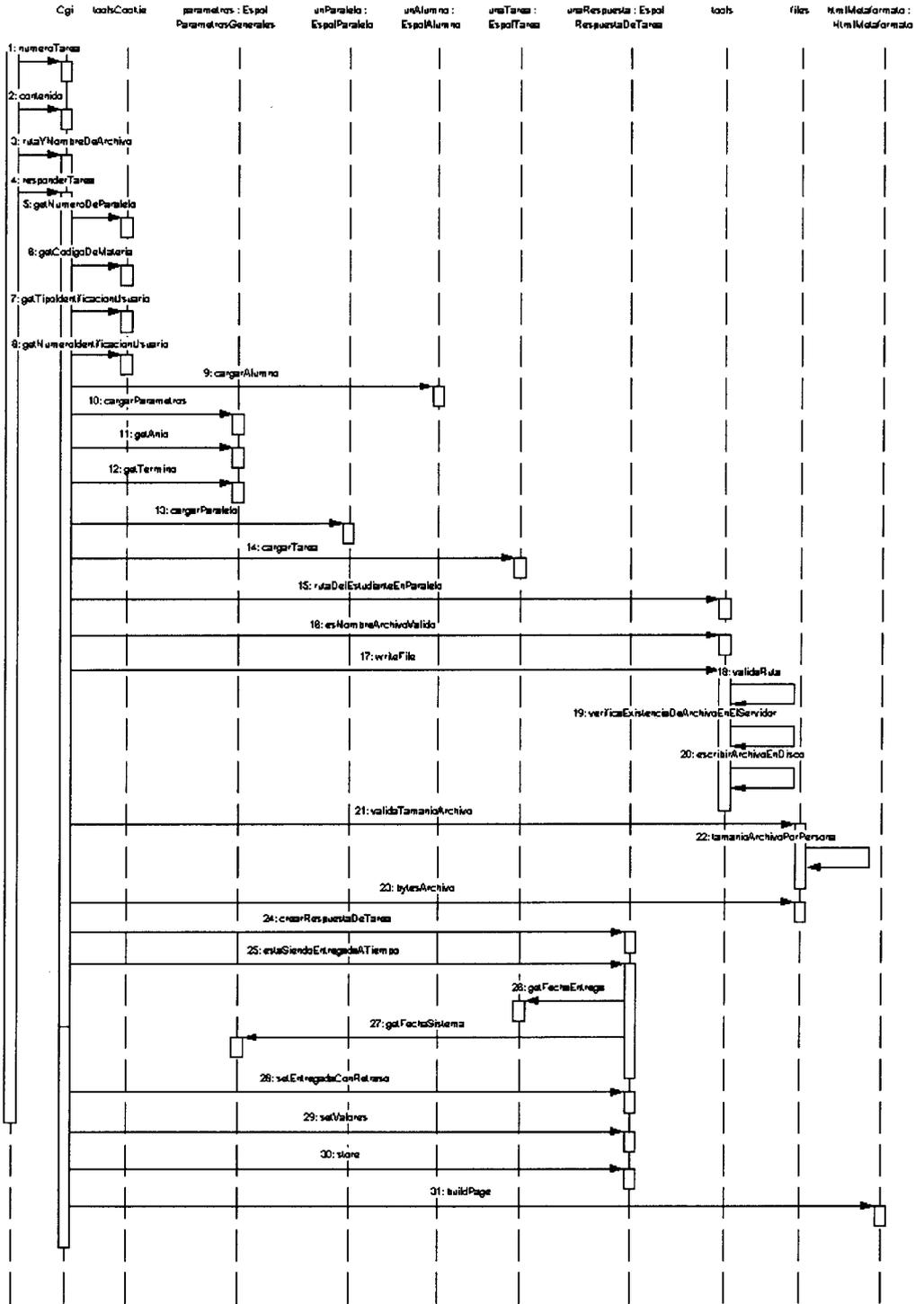
Escenario 4.2: La tarea es respondida con atraso en forma de texto exitosamente.



Escenario 4.0: La tarea es respondida a tiempo en forma de archivo exitosamente.

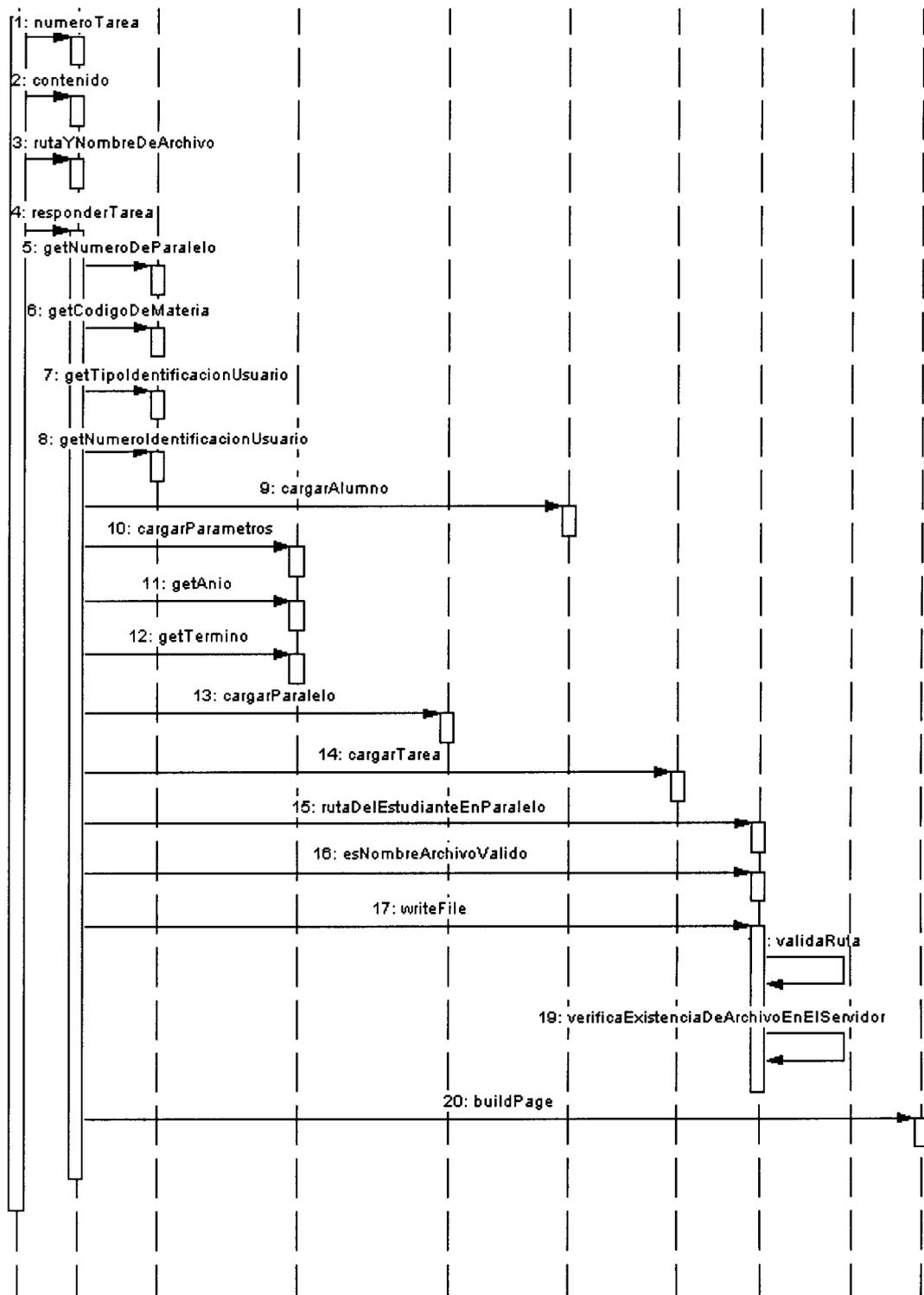


Escenario 4.4: La tarea es respondida con atraso en forma de archivo extensamente.



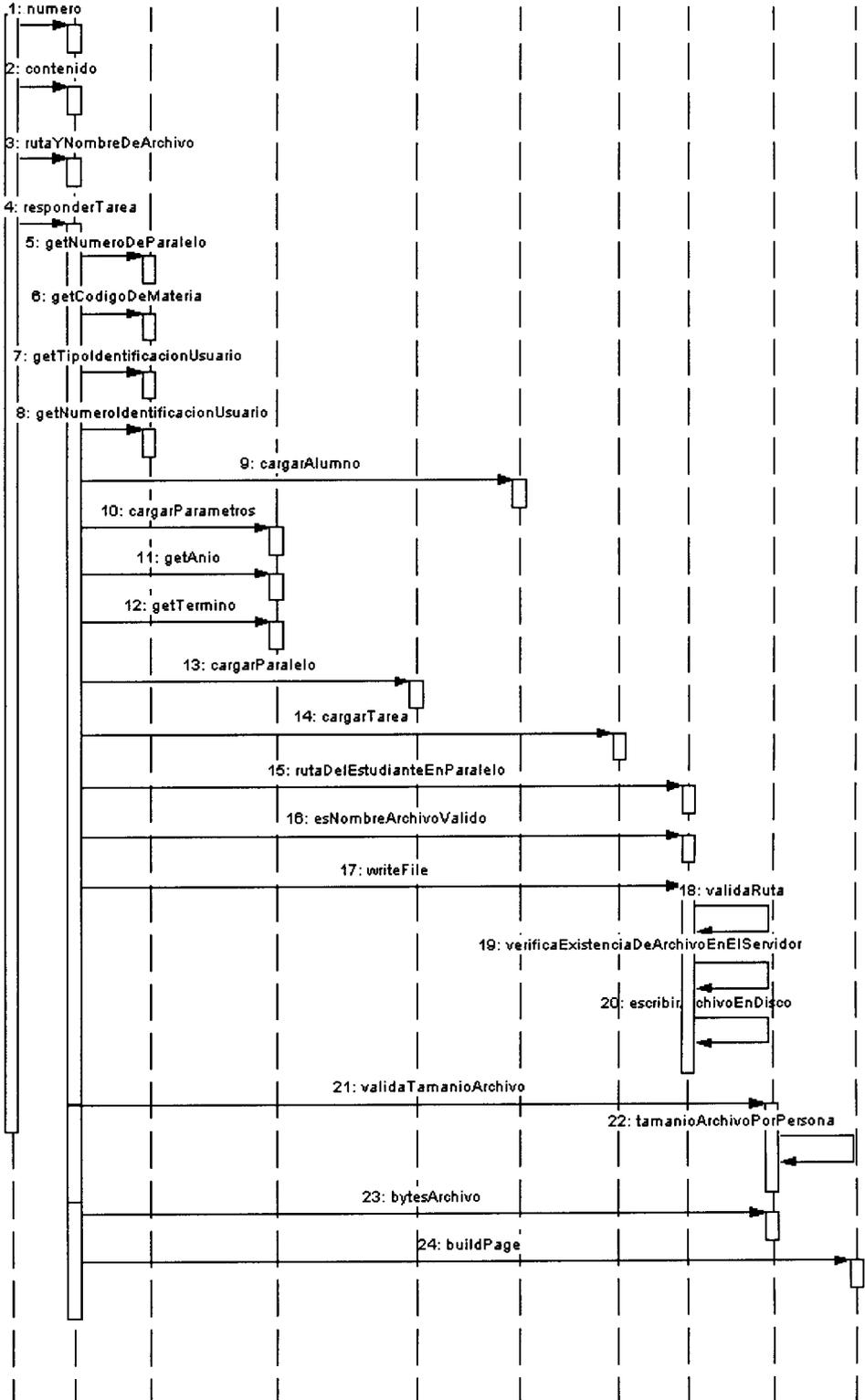
Escenario 4.5: No se puede responder la tarea porque se desea adjuntar un archivo que ya existe en el servidor.

Cgi toolsCookie parametros:Espol unParalelo: unAlumno: unaTarea: tools files htmlError
ParametrosGenerales EspolParalelo EspolAlumno EspolTarea

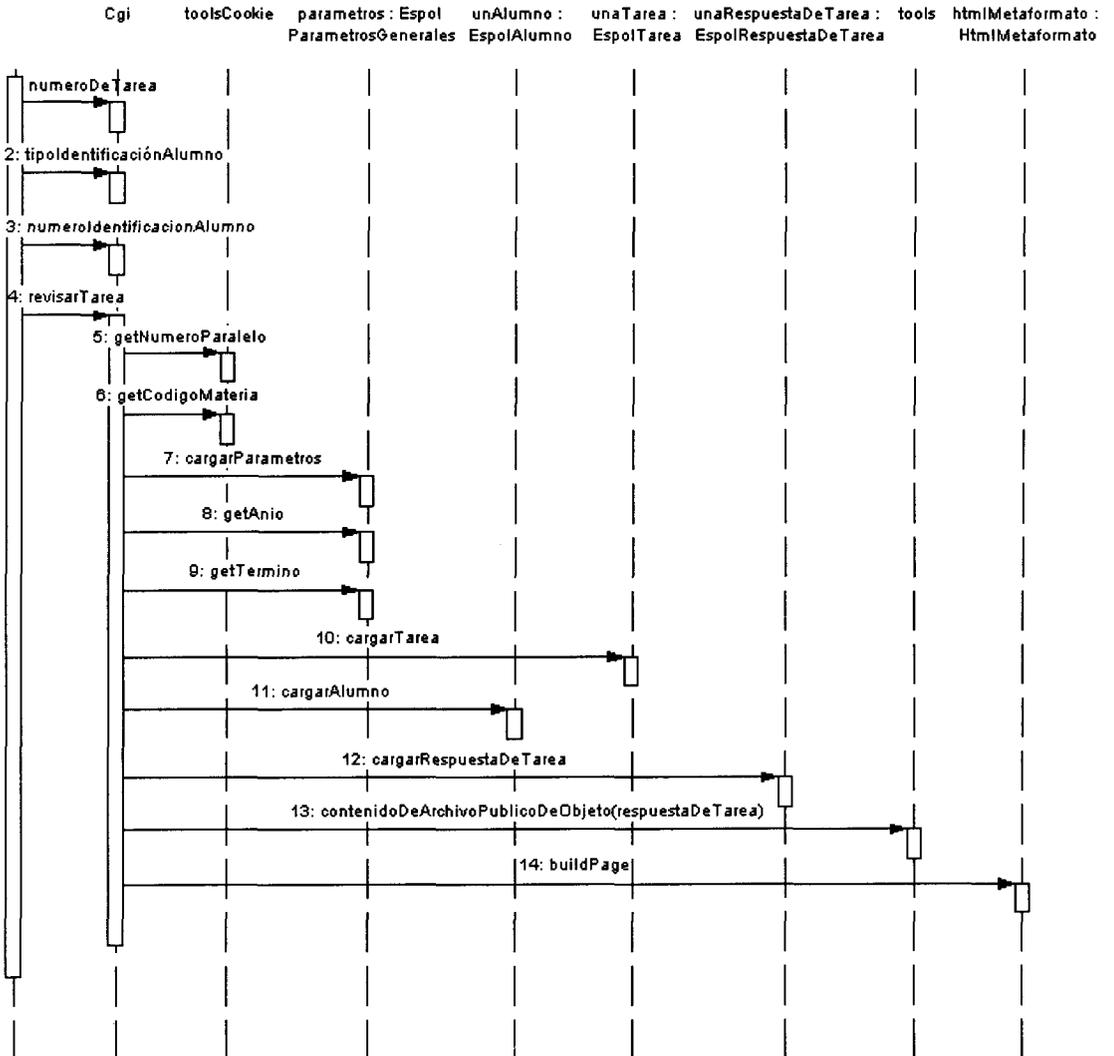


Escenario 4.6: No se puede responder la tarea porque se desea adjuntar un archivo cuyo tamaño sobrepasa al tamaño máximo especificado para el usuario.

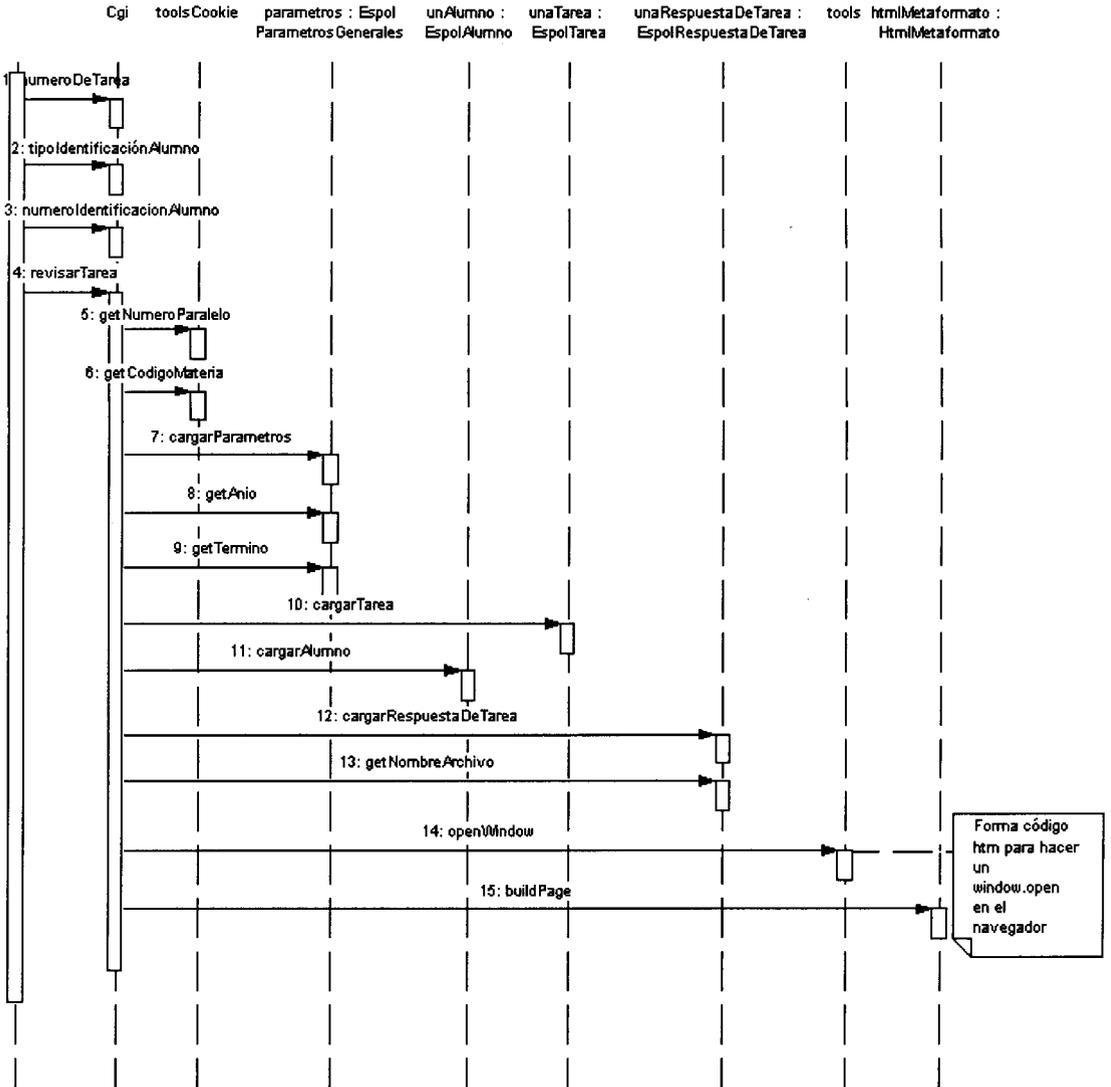
Cgi toolsCookie parametros: Espol unParalelo: unAlumno: unaTarea: tools files htmlError
ParametrosGenerales EspolParalelo EspolAlumno EspolTarea



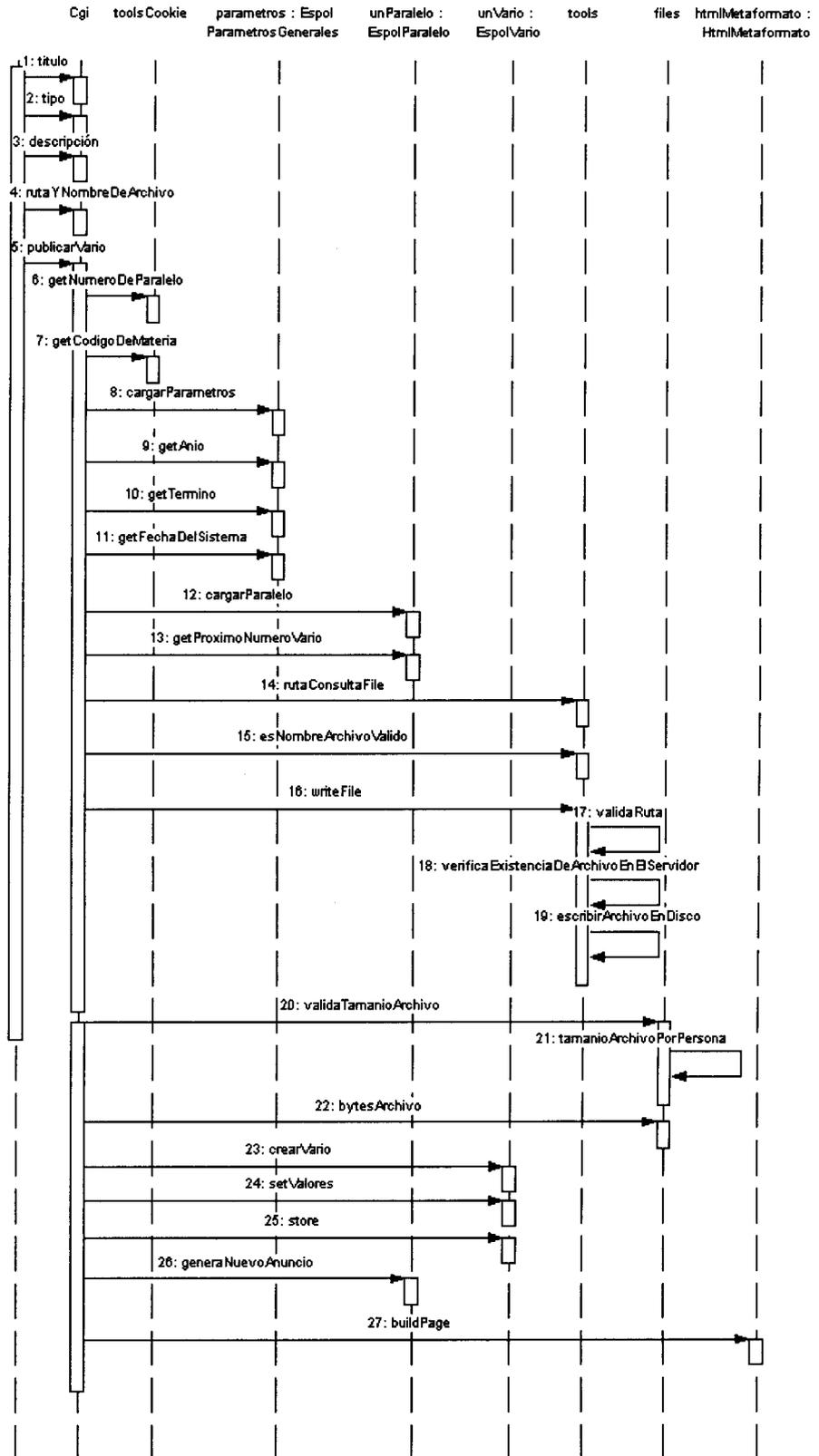
Escenario 5.1: La respuesta de tarea en forma de texto es revisada exitosamente.



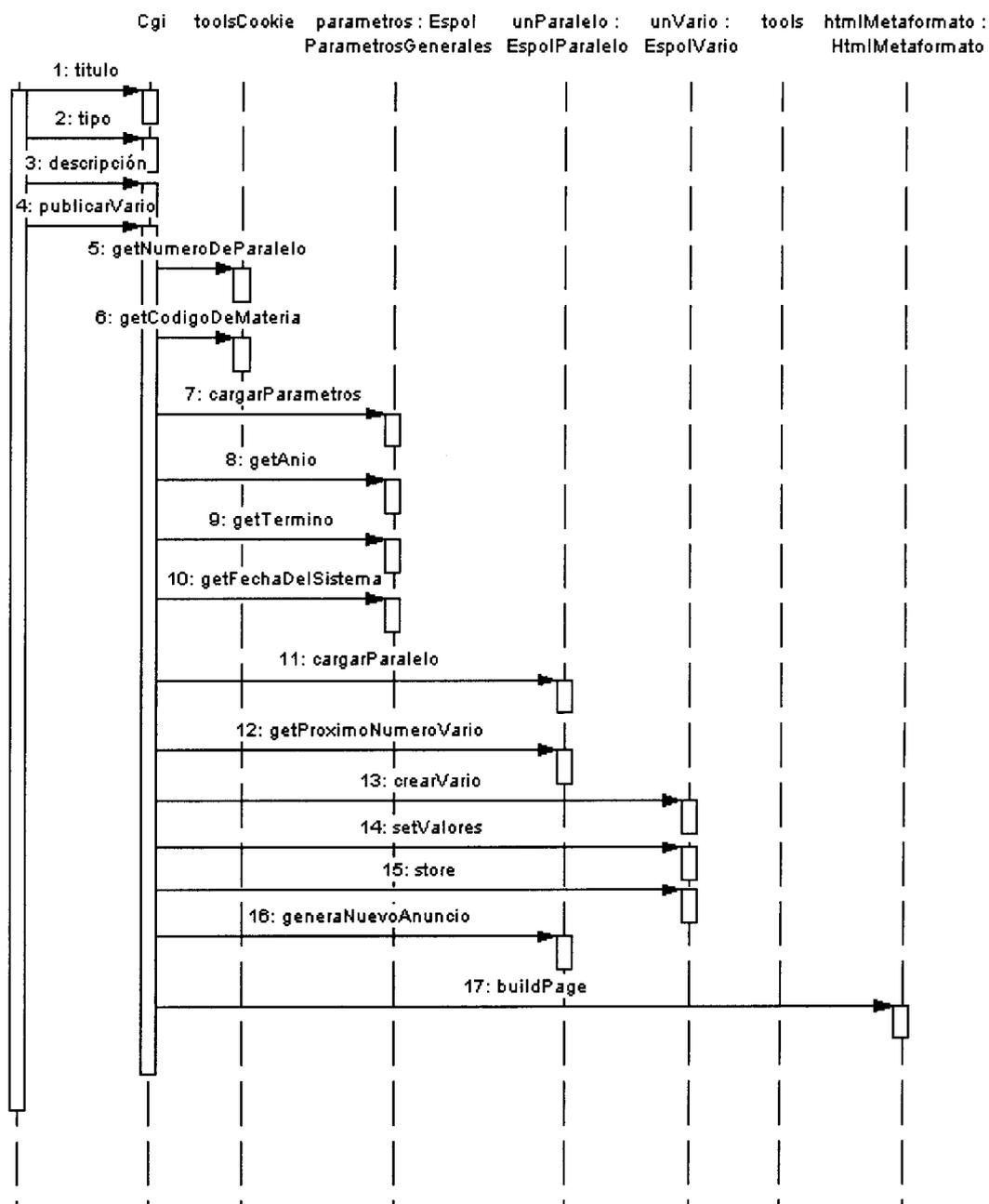
Escenario 5.2: La respuesta de tarea en forma de archivo es revisada exitosamente.



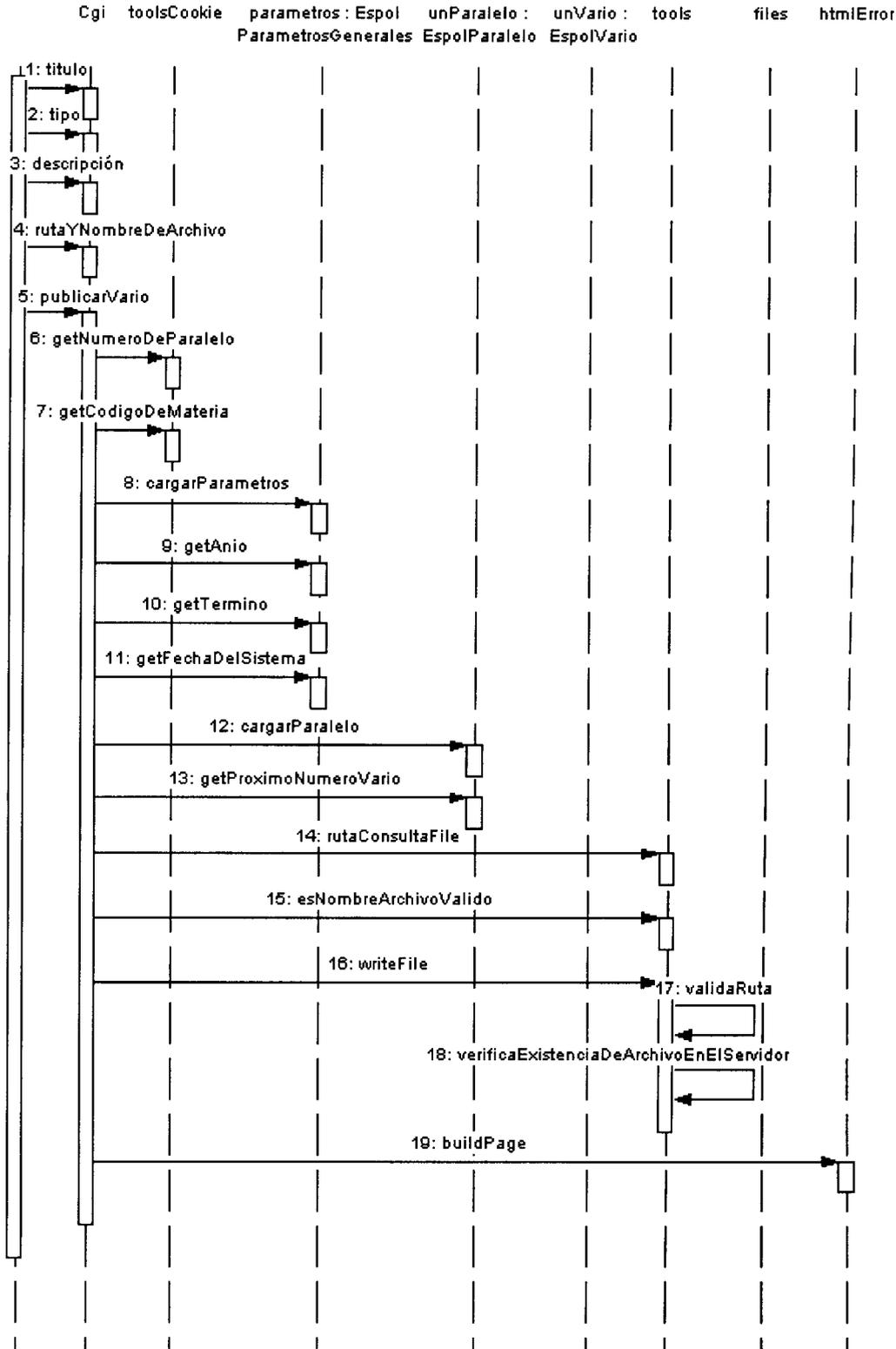
Escenario 6.1: El recurso se publica en forma de archivo (cualquiera de las tres categorías) exitosamente.



Escenario 6.2: El recurso se publica en forma de enlace exitosamente.

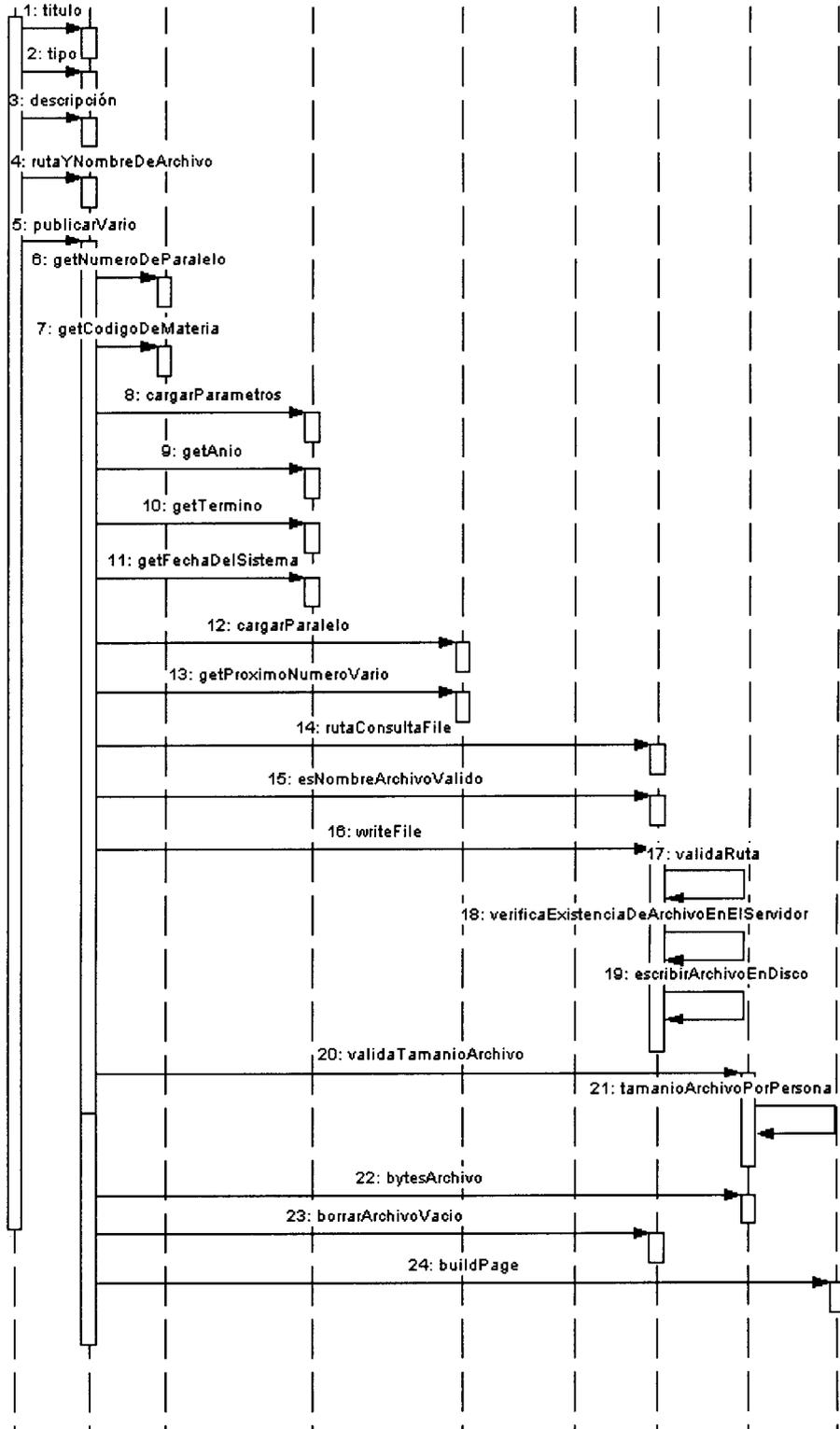


Escenario 6.3 No se puede publicar el recurso porque se desea adjuntar un archivo que ya existe en el servidor

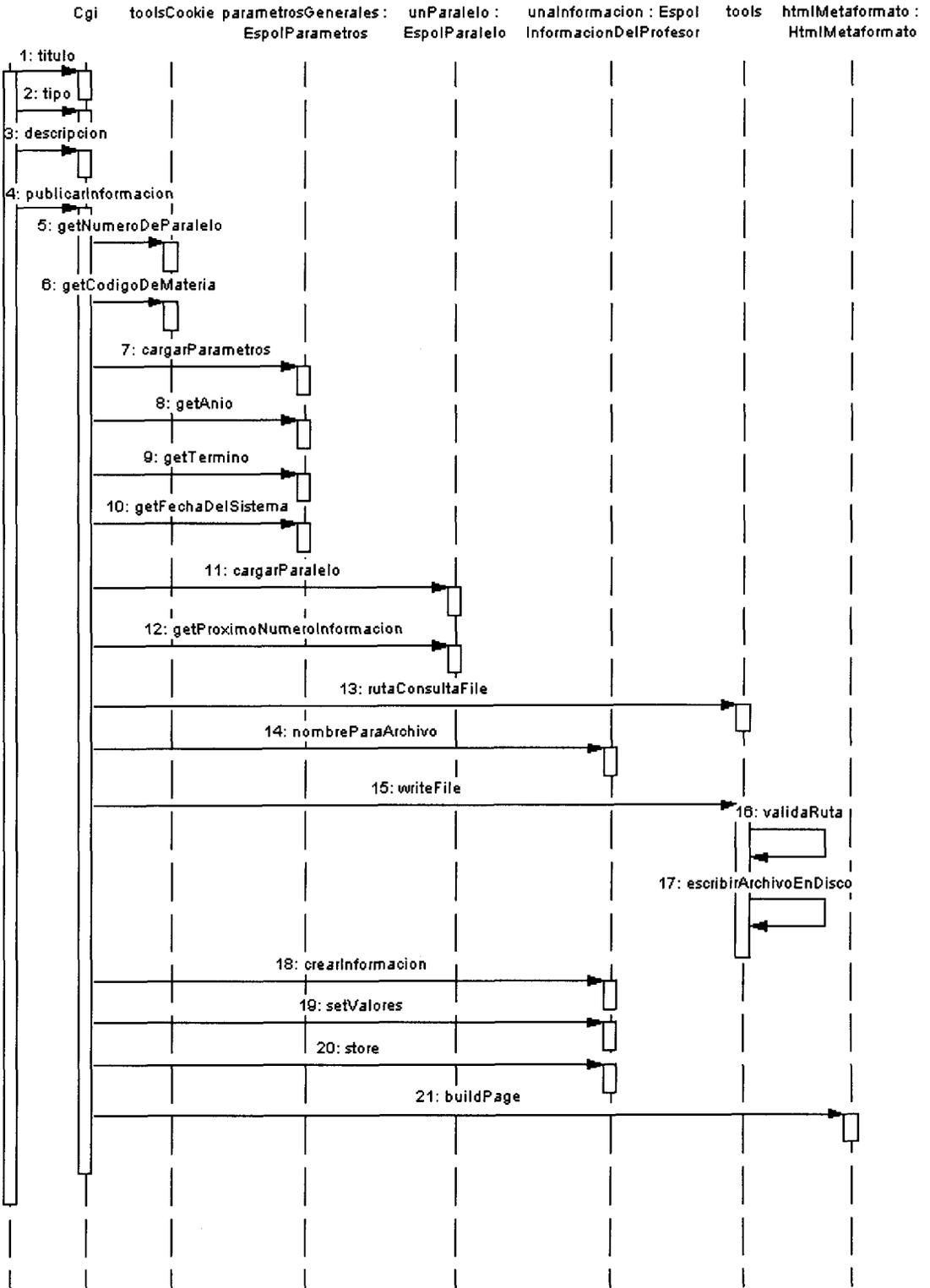


Escenario 6.4 No se puede publicar el recurso porque se desea adjuntar un archivo cuyo tamaño sobrepasa al tamaño máximo especificado para el usuario.

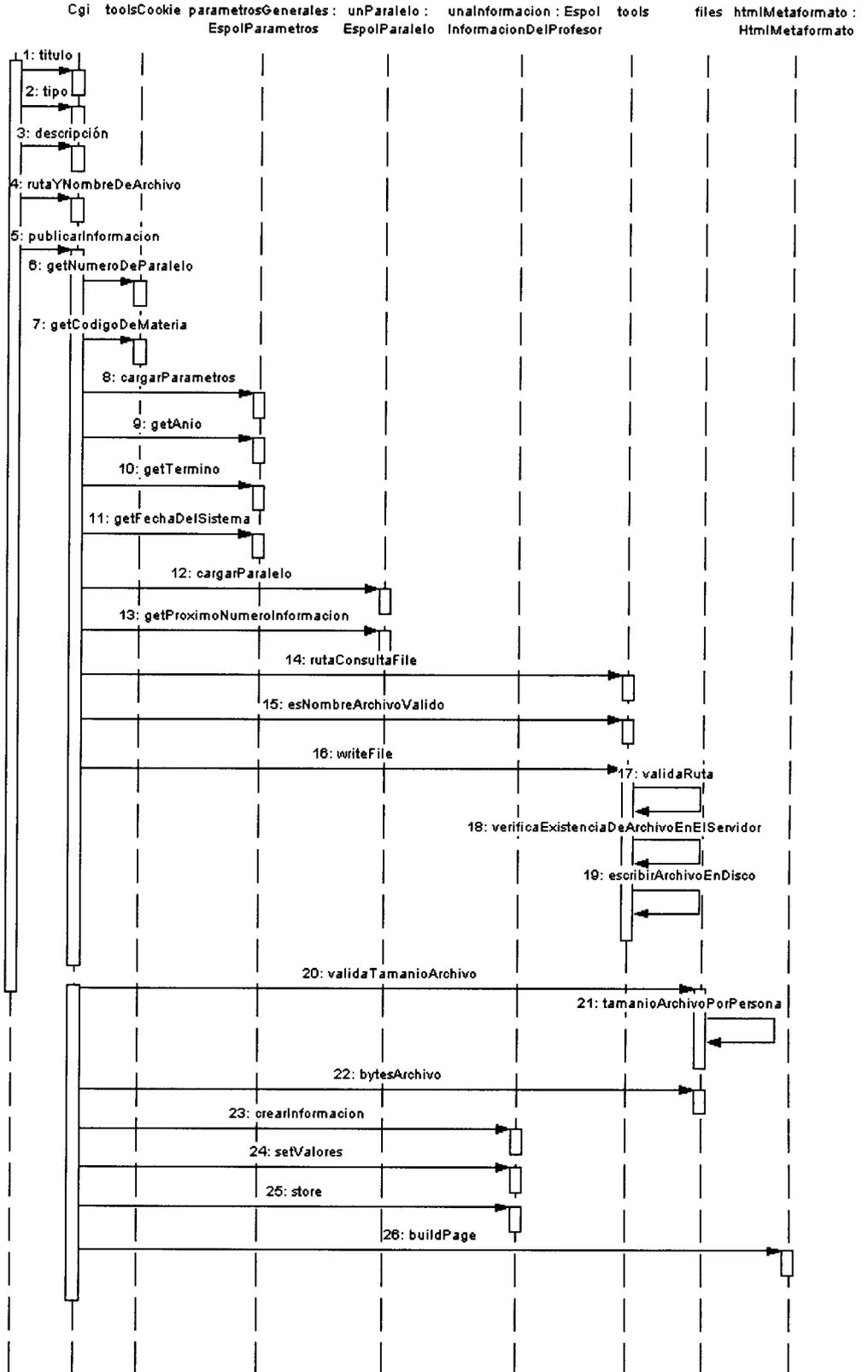
Cgi toolsCookie parametros : Espol unParalelo : unVario : tools files htmlError
ParametrosGenerales EspolParalelo EspolVario



Escenario 7.1: La información se publica en forma de texto exitosamente.

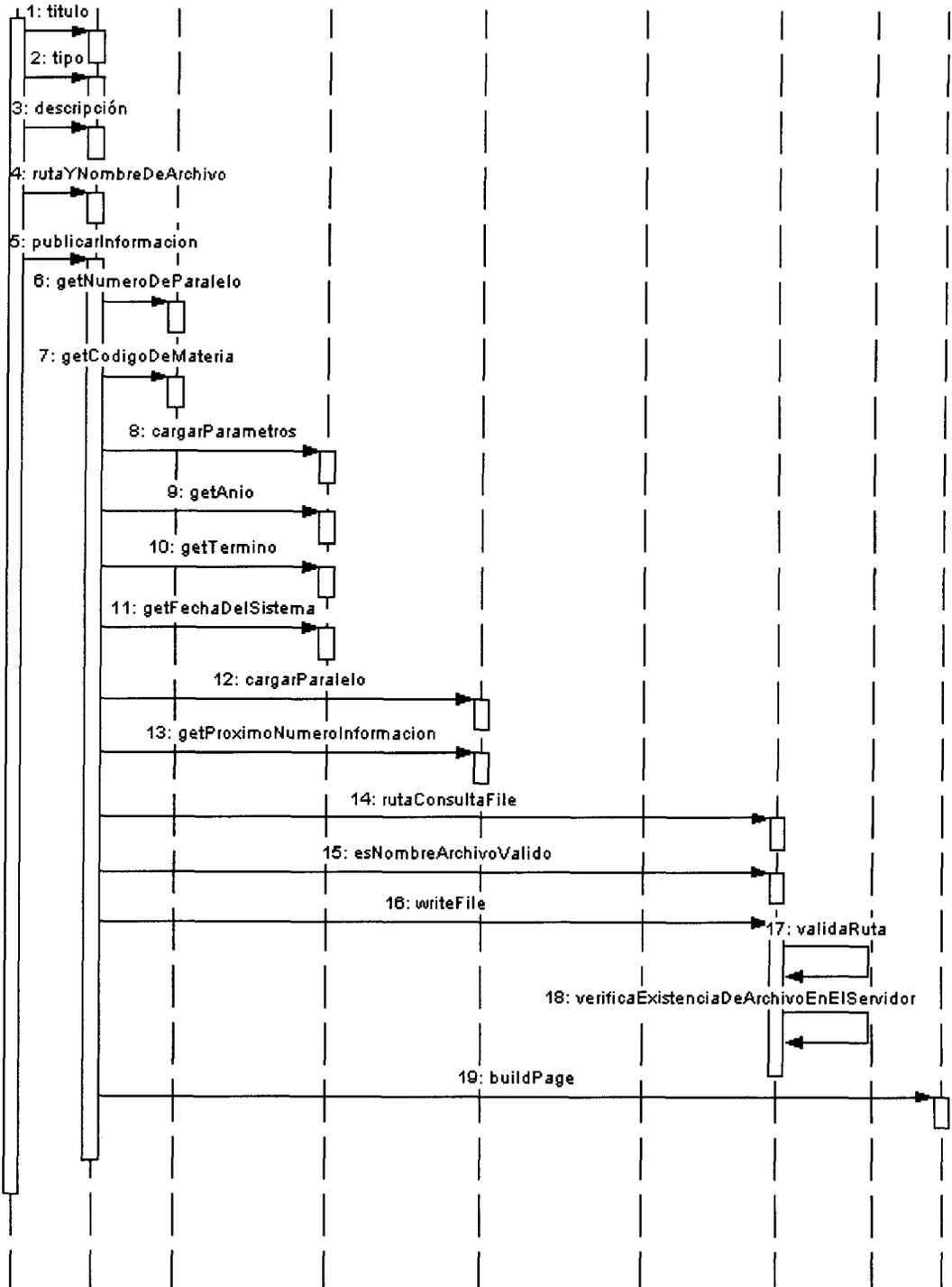


Escenario 7.2: La información se publica en forma de archivo exitosamente.



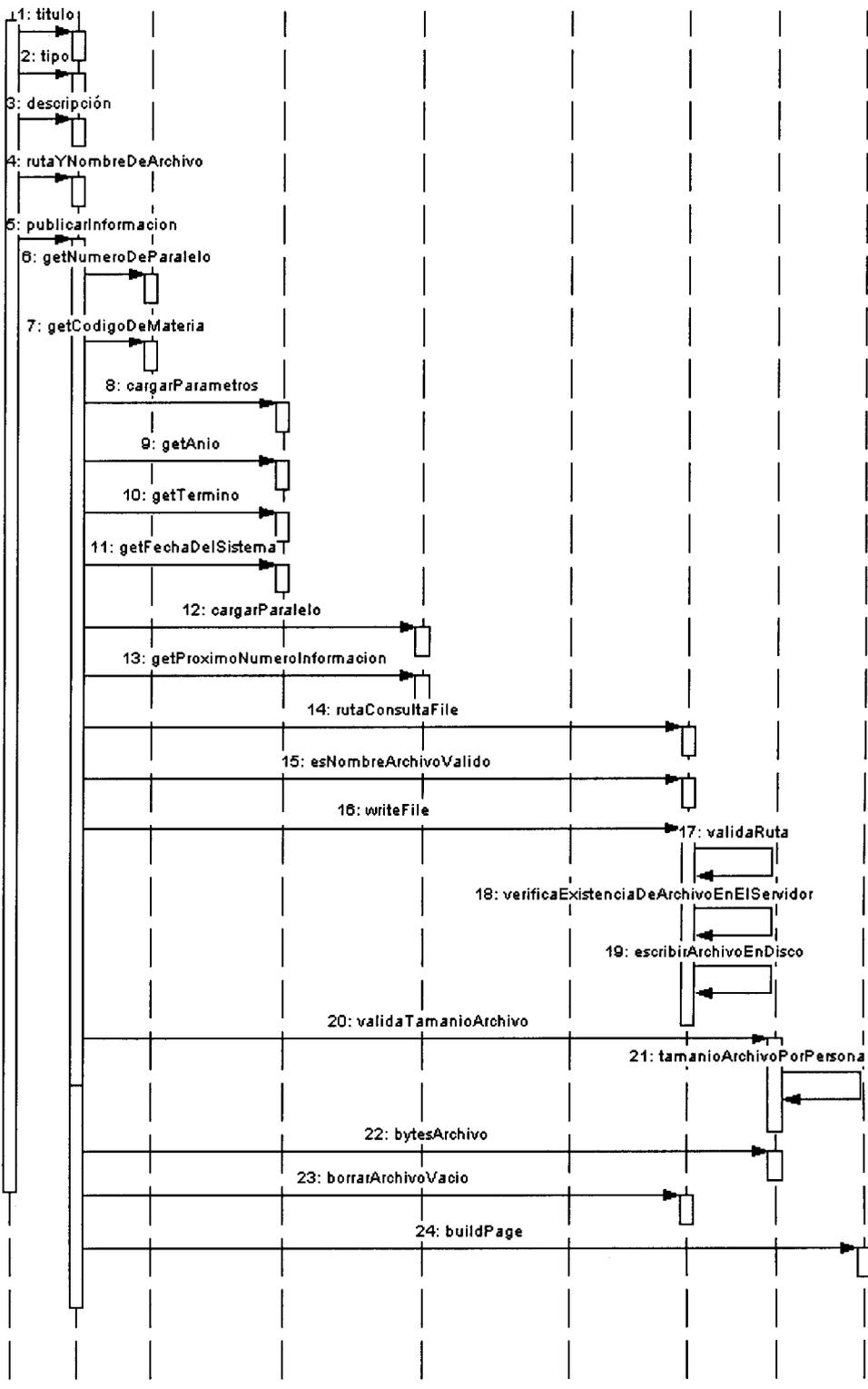
Escenario 7.3 No se puede publicar la información porque se desea adjuntar un archivo que ya existe en el servidor.

Cgi toolsCookie parametrosGenerales : unParalelo : unaInformacion : Espol tools files htmlError
EspolParametros EspolParalelo InformacionDelProfesor

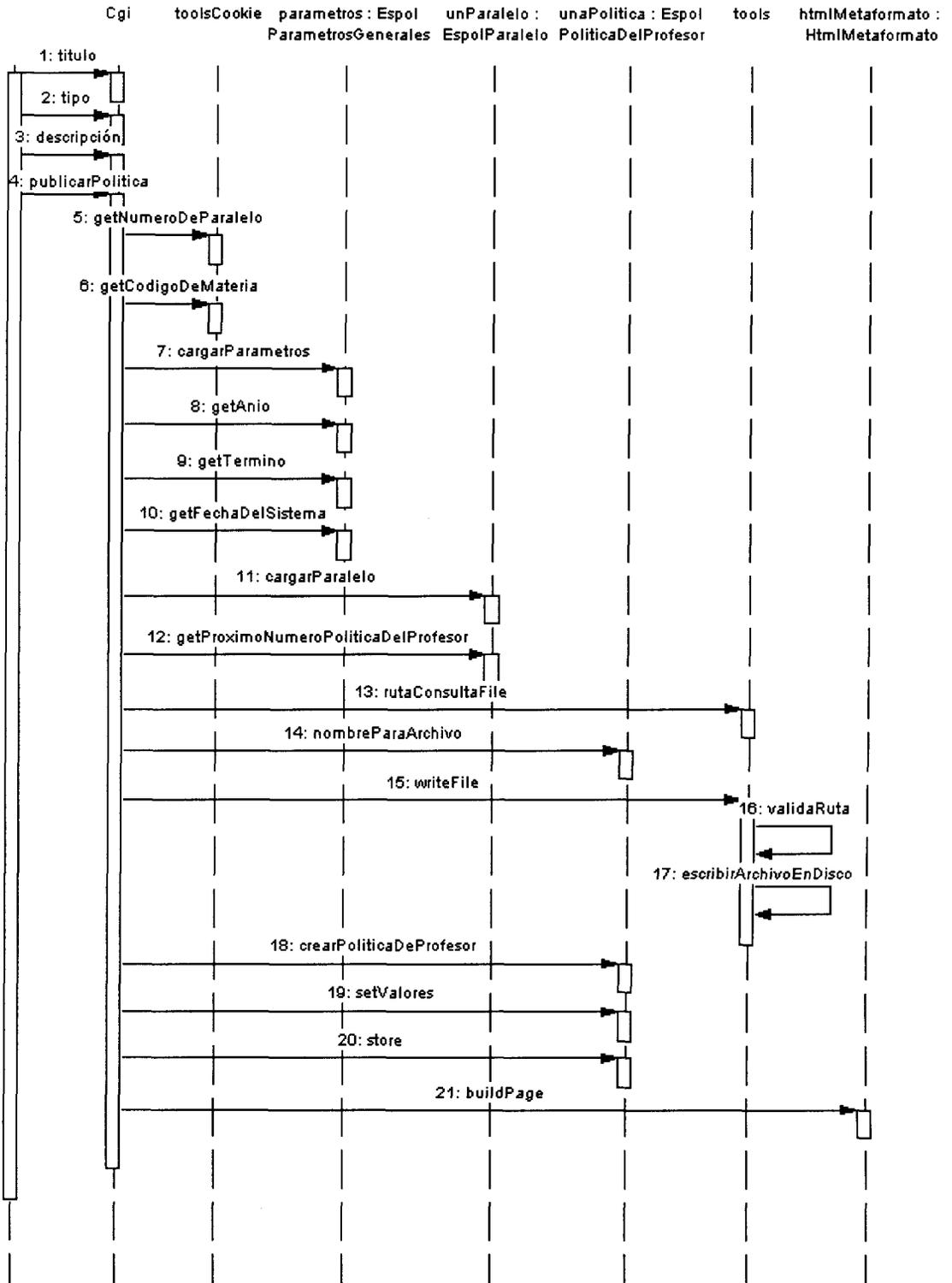


Escenario 7.4 No se puede publicar la información porque se desea adjuntar un archivo cuyo tamaño sobrepasa al tamaño máximo especificado para el usuario.

Cgi toolsCookie parametros: Espol unParalelo: unaInformacion: Espol tools files htmlError
ParametrosGenerales EspolParalelo InformacionDelProfesor

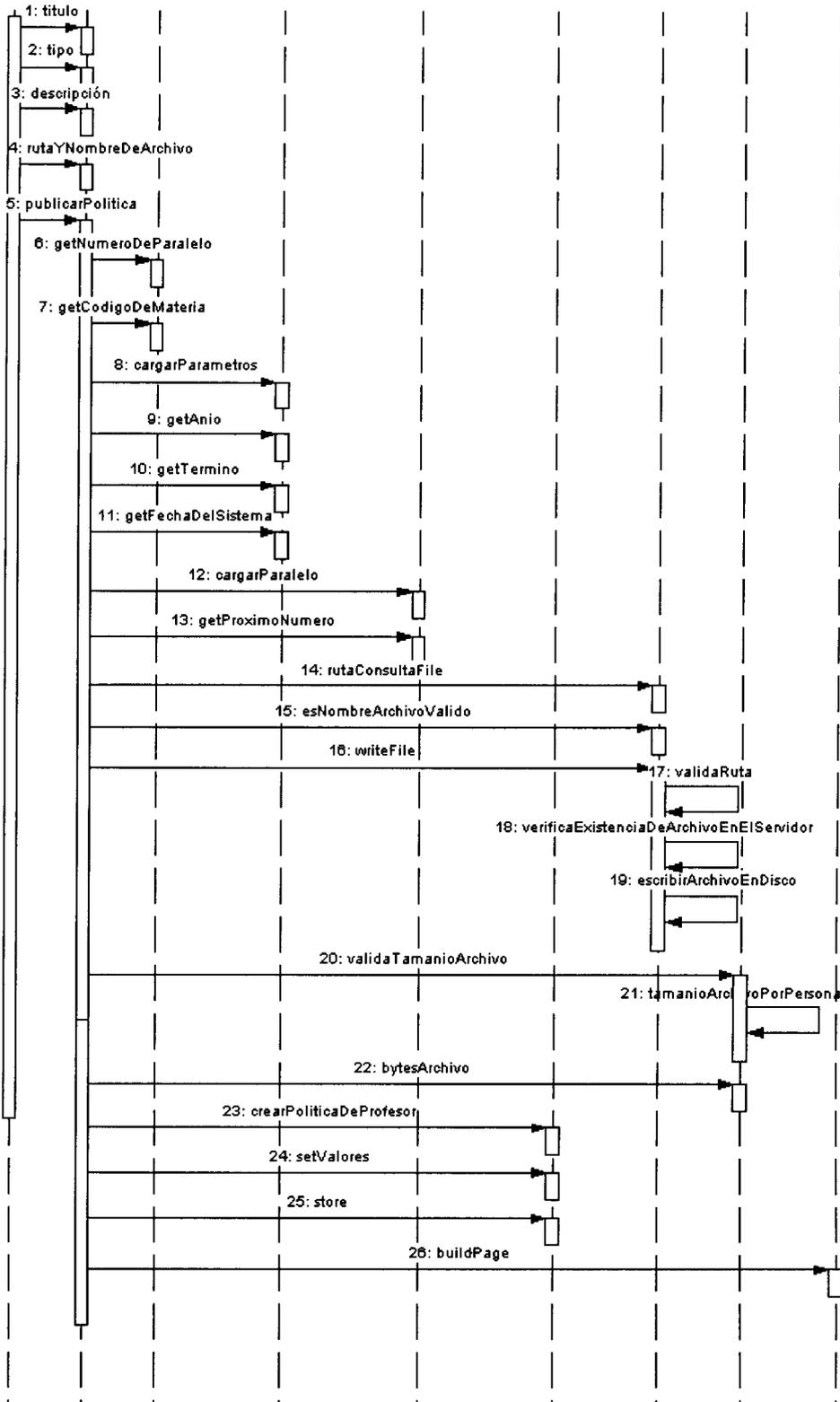


Escenario 8.1: La política de curso se publica en forma de texto exitosamente.



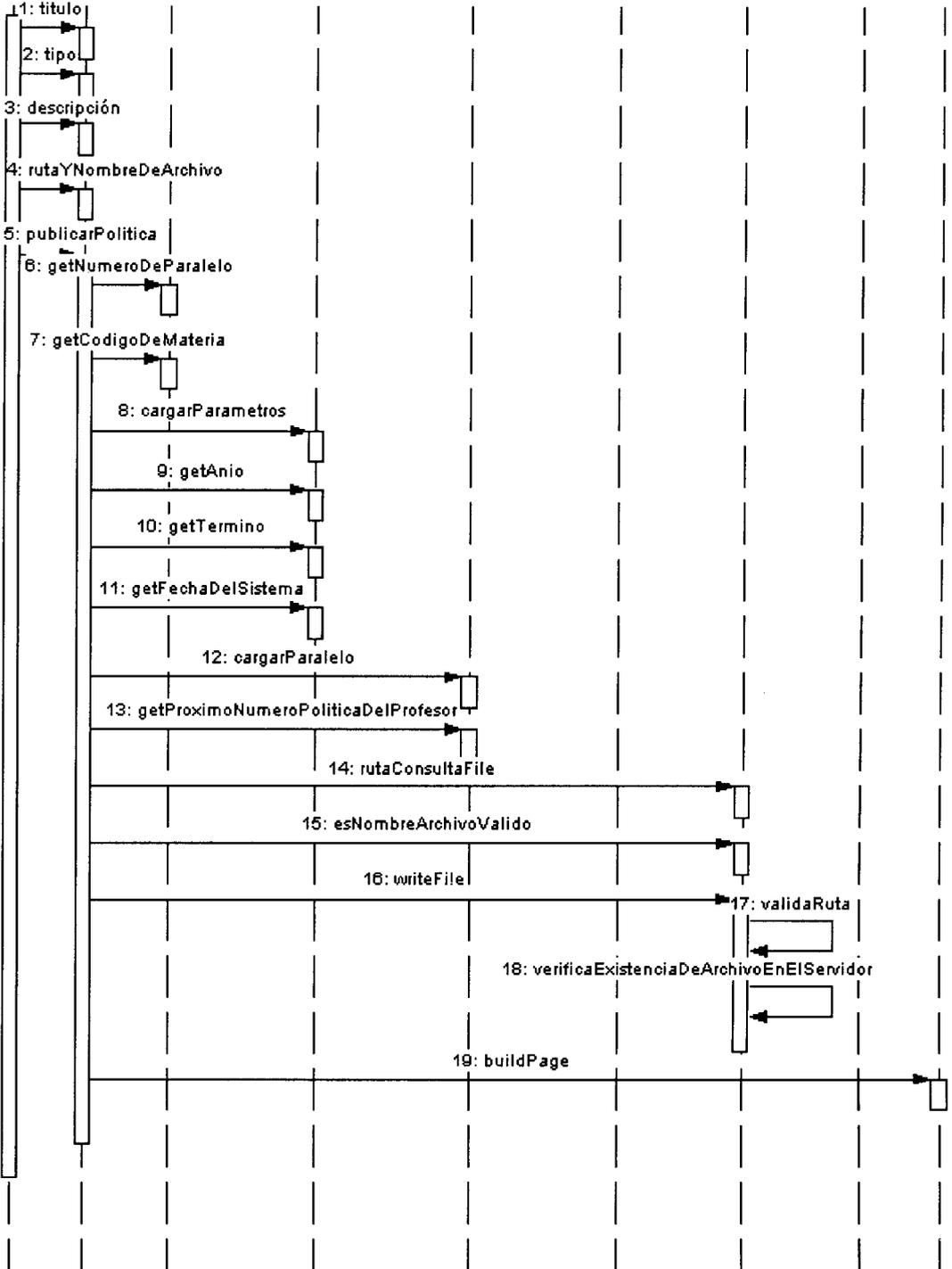
Escenario 8.2: La política de curso se publica en forma de archivo exitosamente.

Cgi toolsCookie parametros : Espol unParalelo : unaPolitica : Espol tools files htmlMetaformato :
 ParametrosGenerales EspolParalelo PoliticaDelProfesor HtmlMetaformato



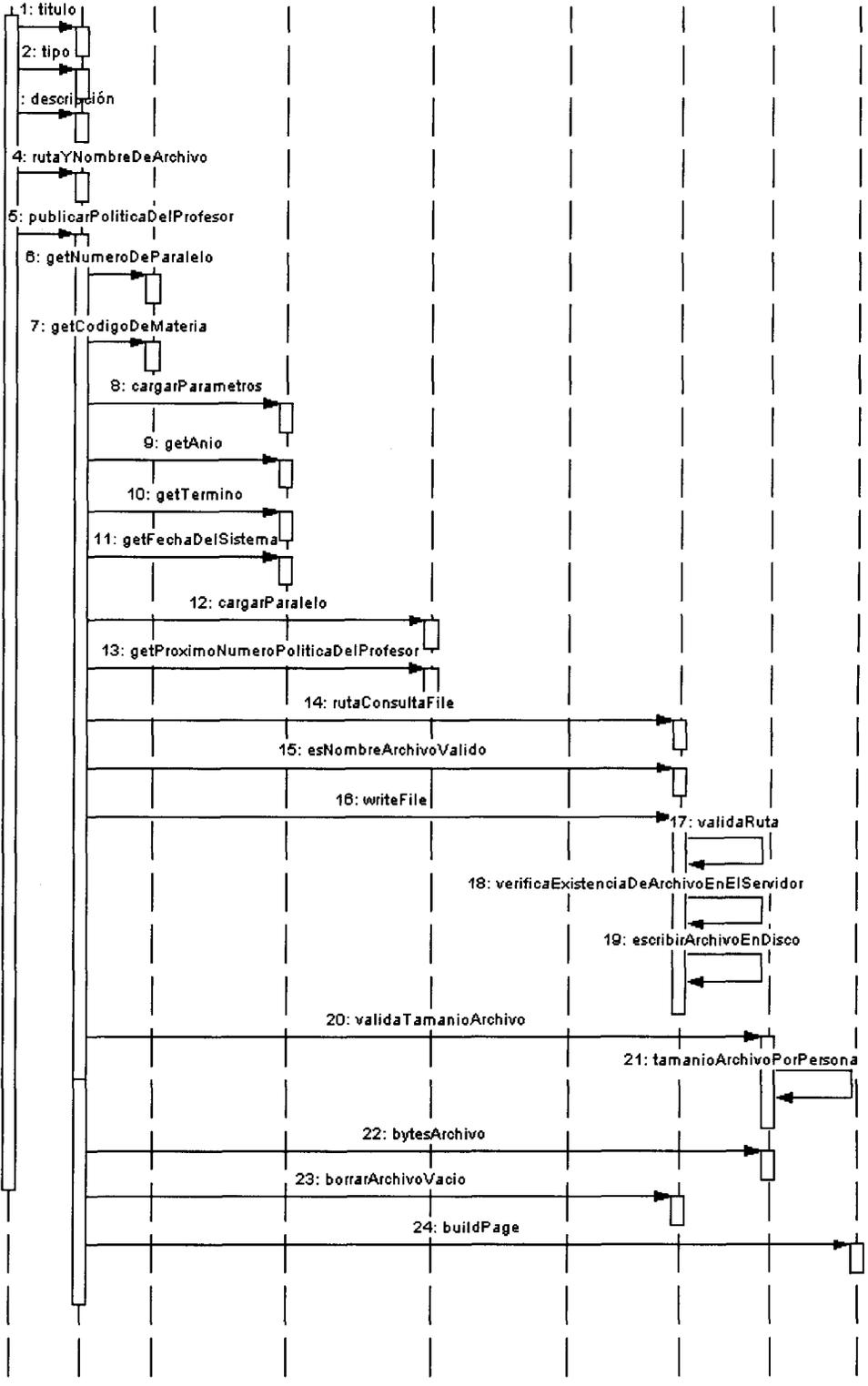
Escenario 8.3 No se puede publicar la política de curso porque se desea adjuntar un archivo que ya existe en el servid

Cgi toolsCookie parametros: Espol unParalelo : unaPolitica : Espol tools files htmlError
 ParametrosGenerales EspolParalelo PoliticaDelProfesor

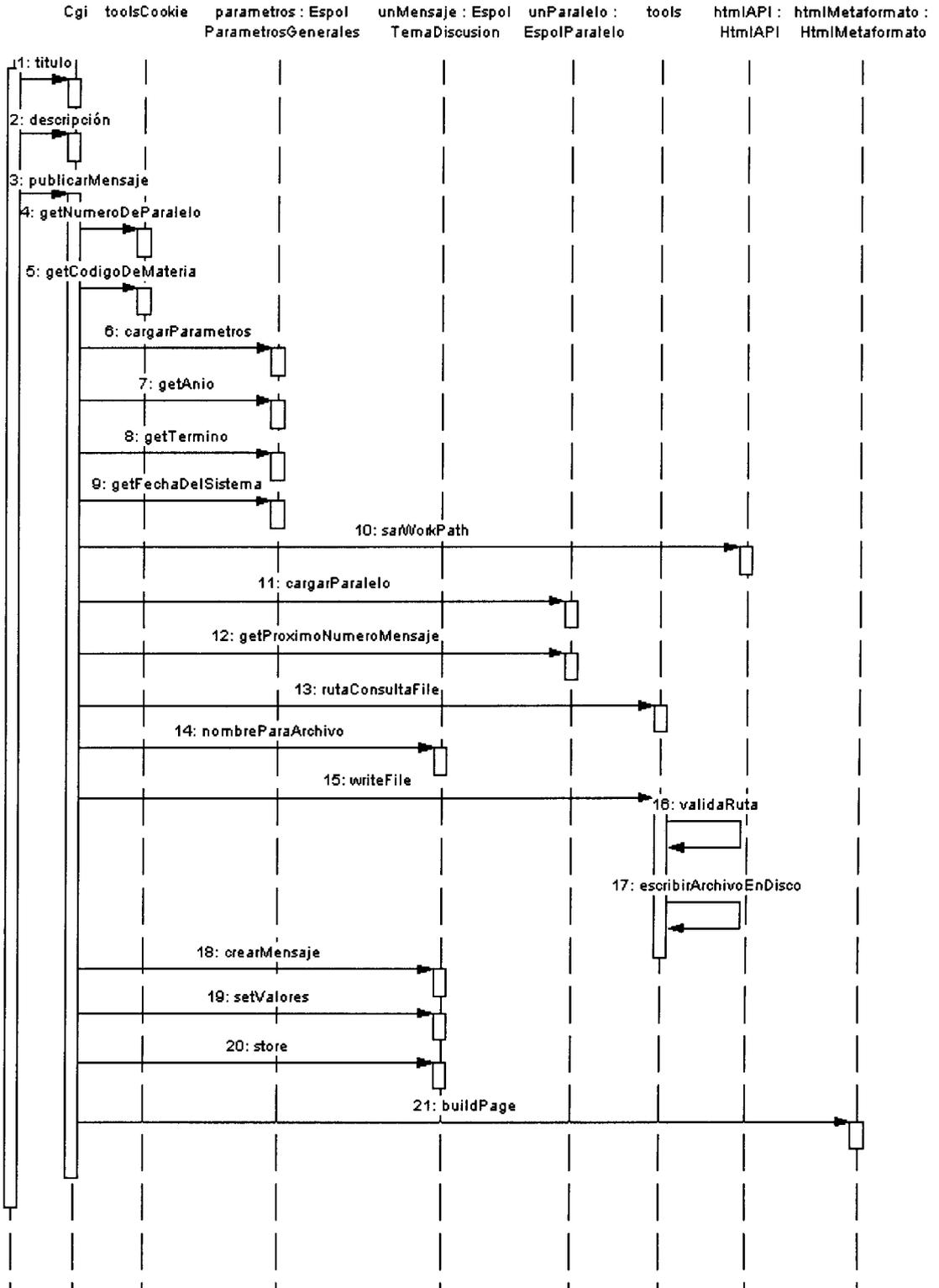


Escenario 8.4 No se puede publicar la política de curso porque se desea adjuntar un archivo cuyo tamaño sobrepasa al tamaño máximo especificado para el usuario.

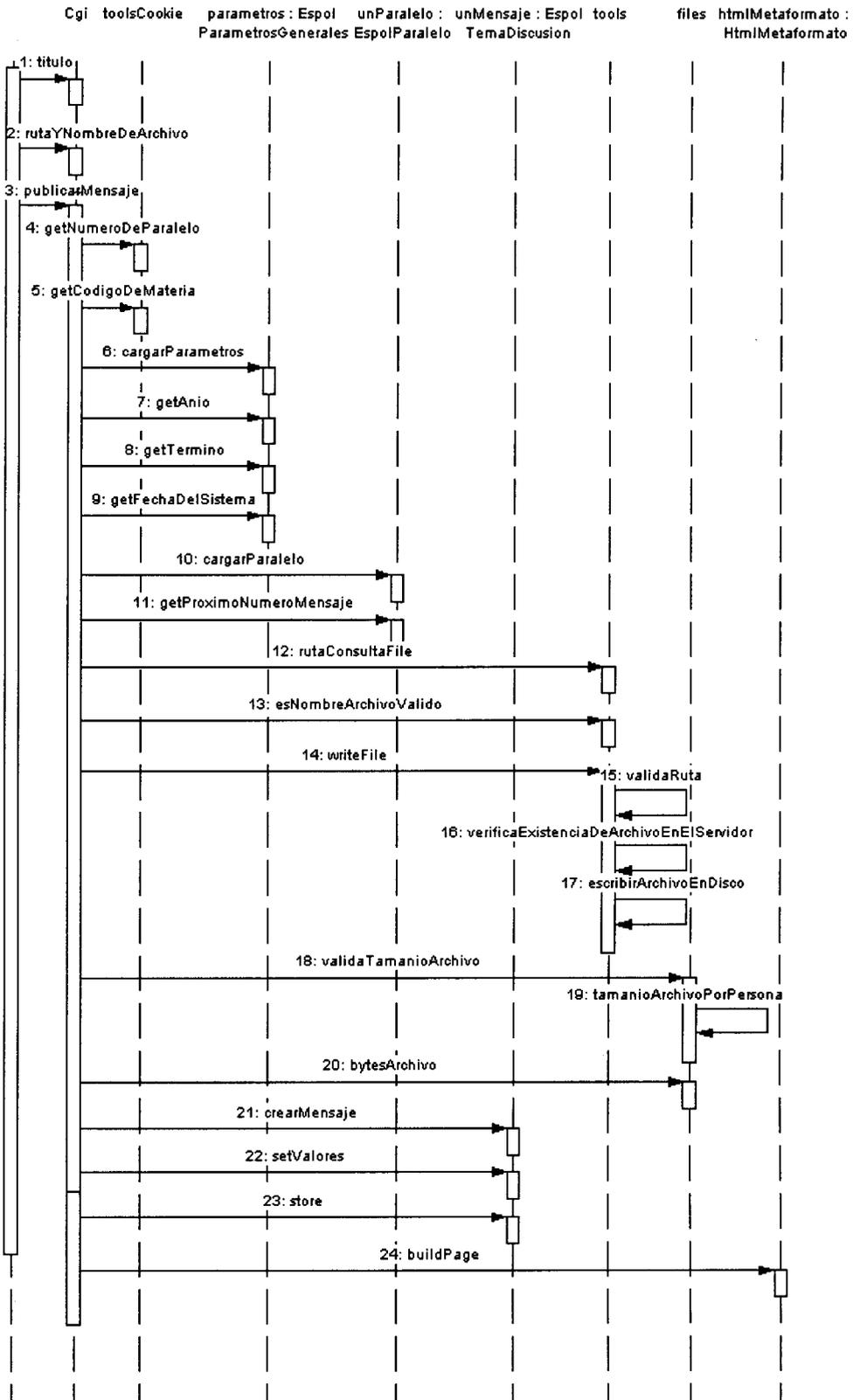
Cgi toolsCookie parametros:Espol unParalelo: unaPolitica:Espol tools files htmlError
ParametrosGenerales EspolParalelo PoliticaDelProfesor



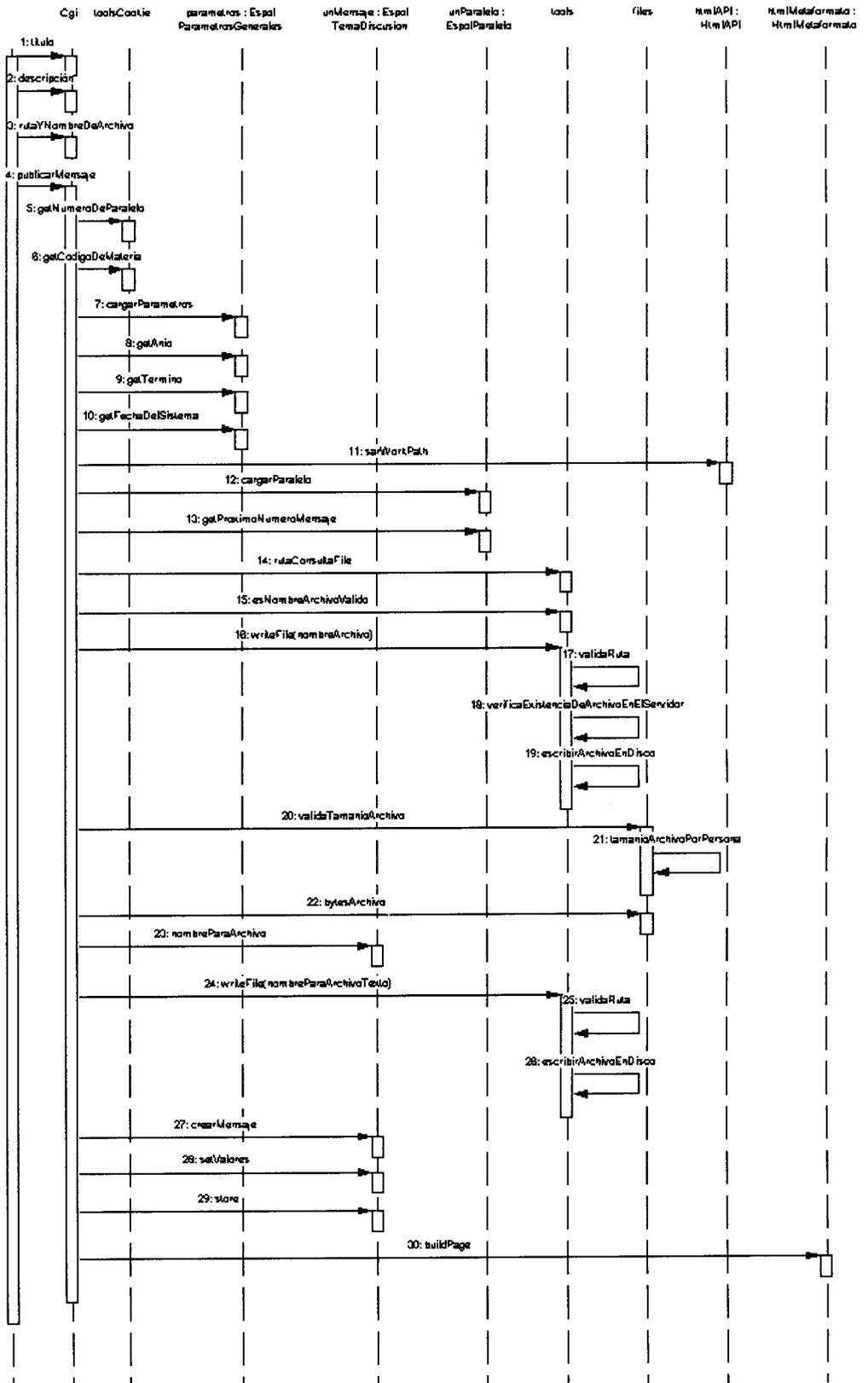
Escenario 9.1: El mensaje se publica sin archivo adjunto en forma exitosa.



Escenario 9.2: El mensaje se publica con un archivo adjunto en forma exitosa.

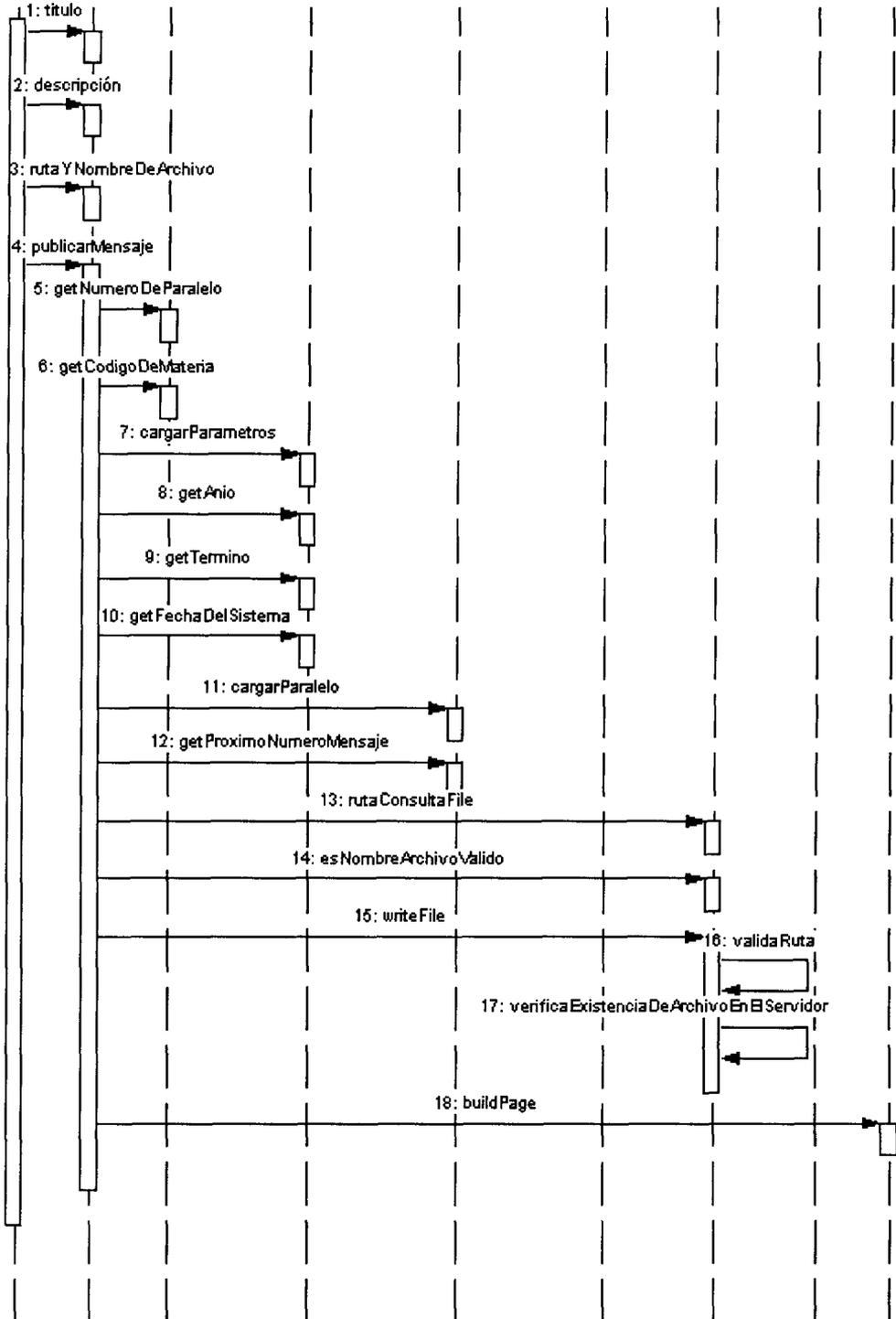


Escenario 9.3: El mensaje se publica con texto y archivo adjunto en forma externa.



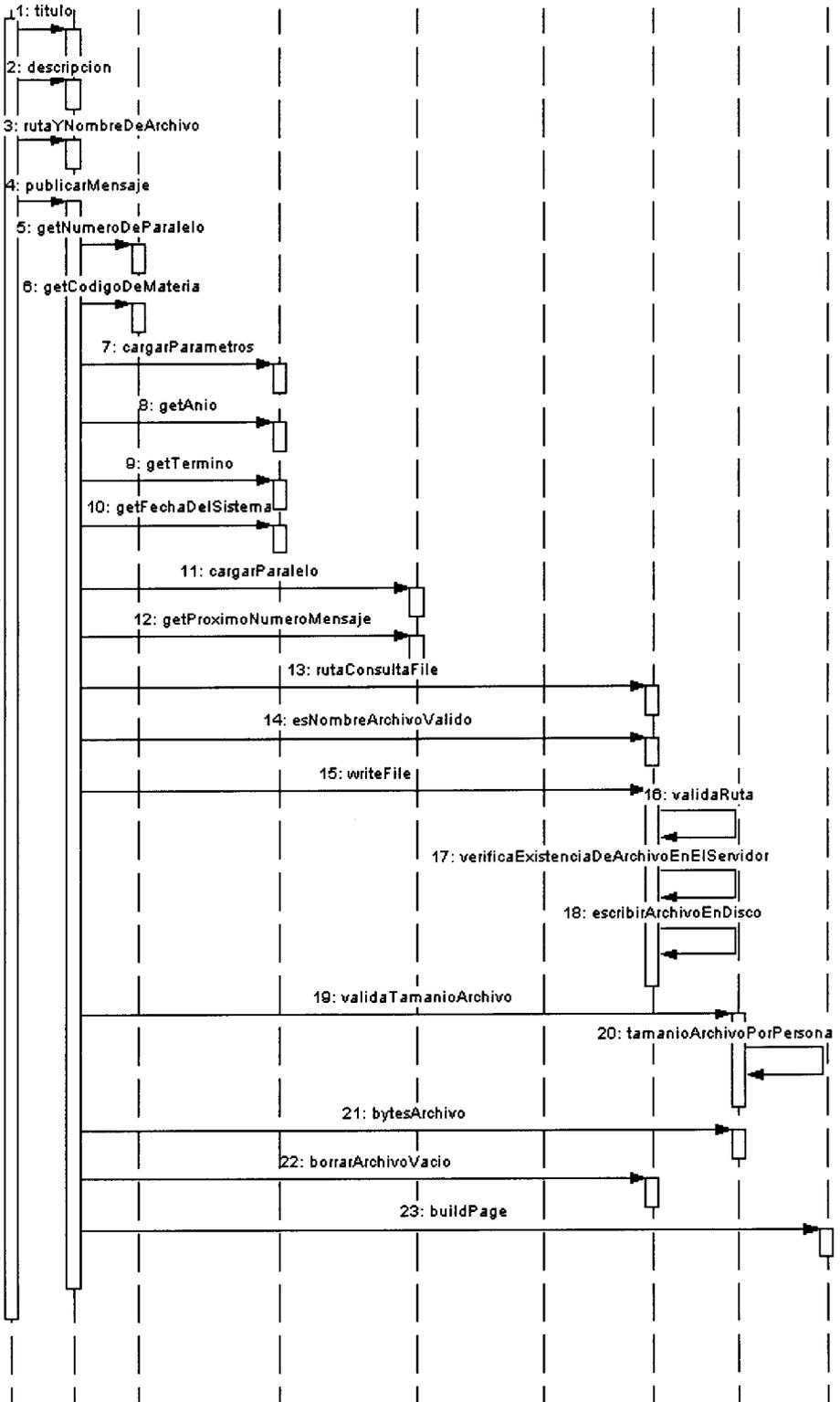
Escenario 9.4: No se puede publicar el mensaje porque se desea adjuntar un archivo que ya existe en el servidor.

Cgi toolsCookie parametros : Espol unParalelo : unMensaje : Espol tools files htmlError
 ParametrosGenerales EspolParalelo TemaDiscusion

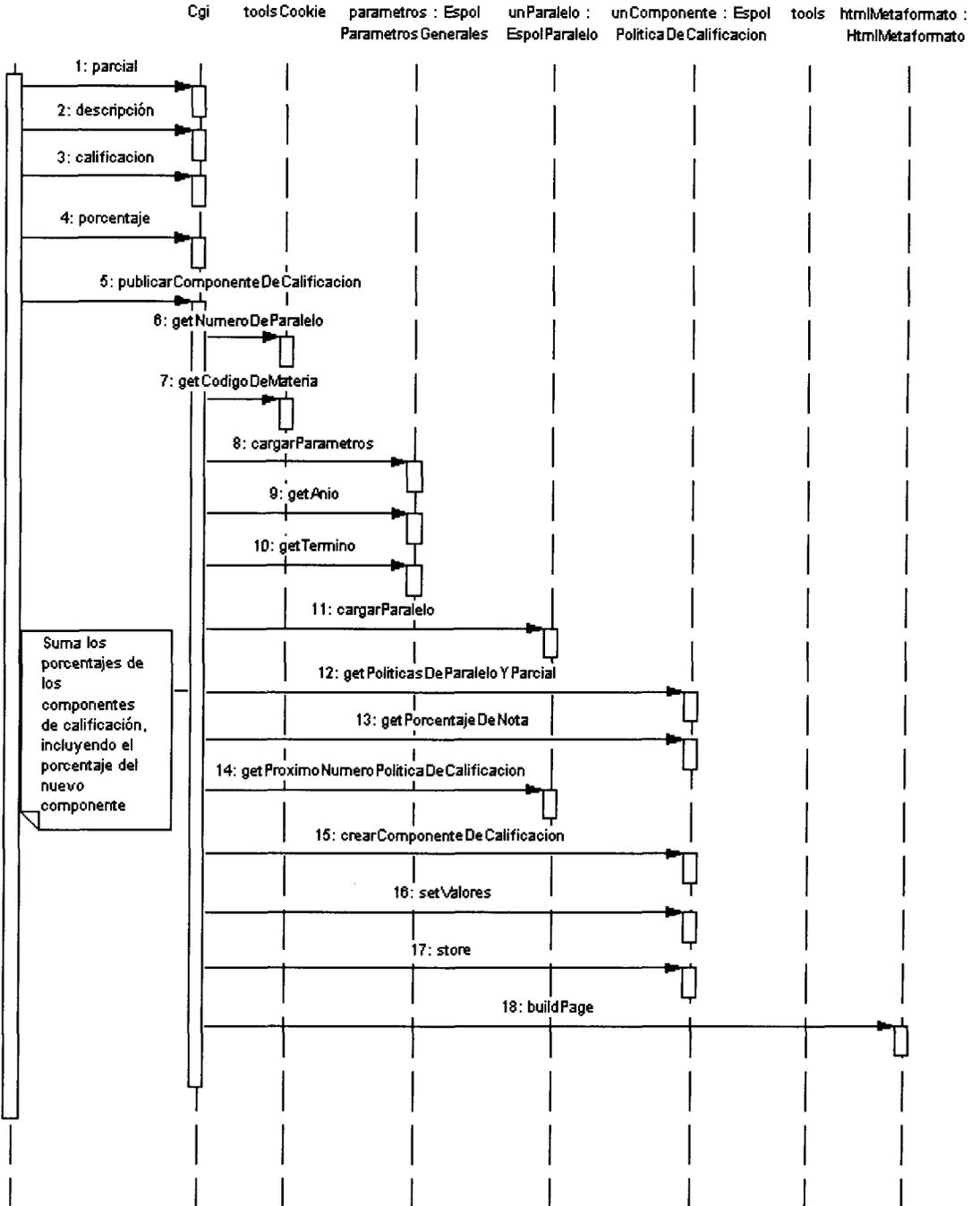


Escenario 9.5: No se puede publicar el mensaje porque se desea adjuntar un archivo cuyo tamaño sobrepasa al tamaño máximo especificado para el usuario.

Cgi toolsCookie parametrosGenerales : unParalelo : unMensaje : Espol tools files htmlError
EspolParametros EspolParalelo TemaDiscusion

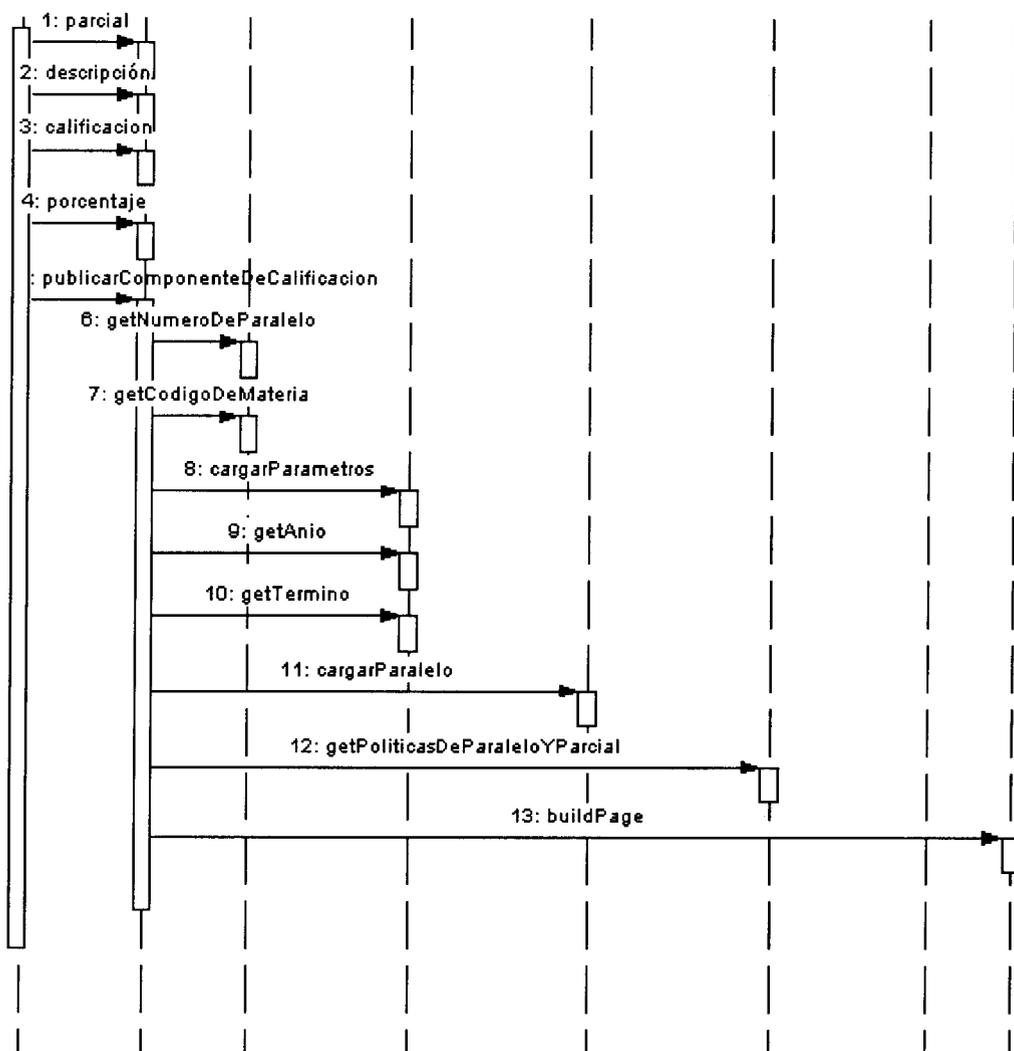


Escenario 10.1: El profesor publica un componente de calificación de manera exitosa.



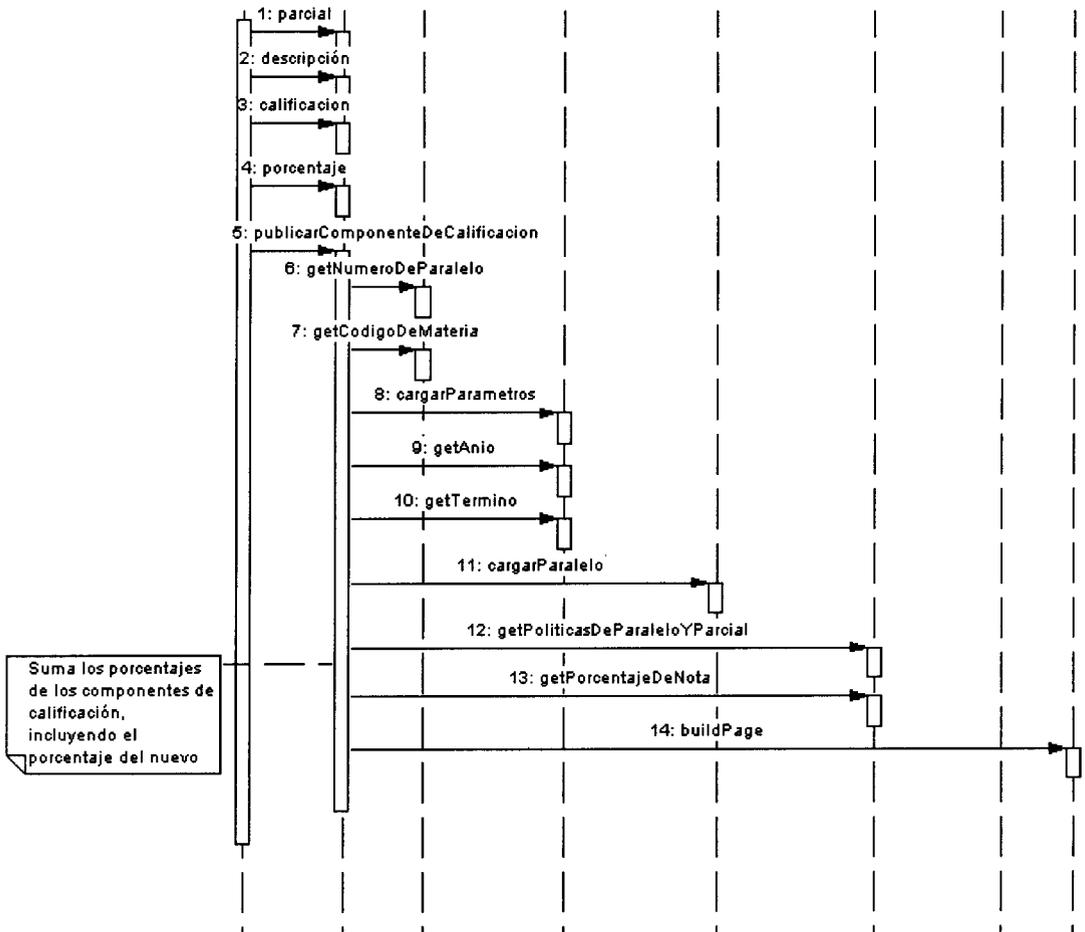
Escenario 10.2: El profesor no puede publicar el componente de la calificación porque ya existen cinco componentes de la calificación definidos para ese parcial.

Cgi toolsCookie parametros:Espol unParalelo: unComponente:Espol tools htmlError
ParametrosGenerales EspolParalelo PoliticaDeCalificacion

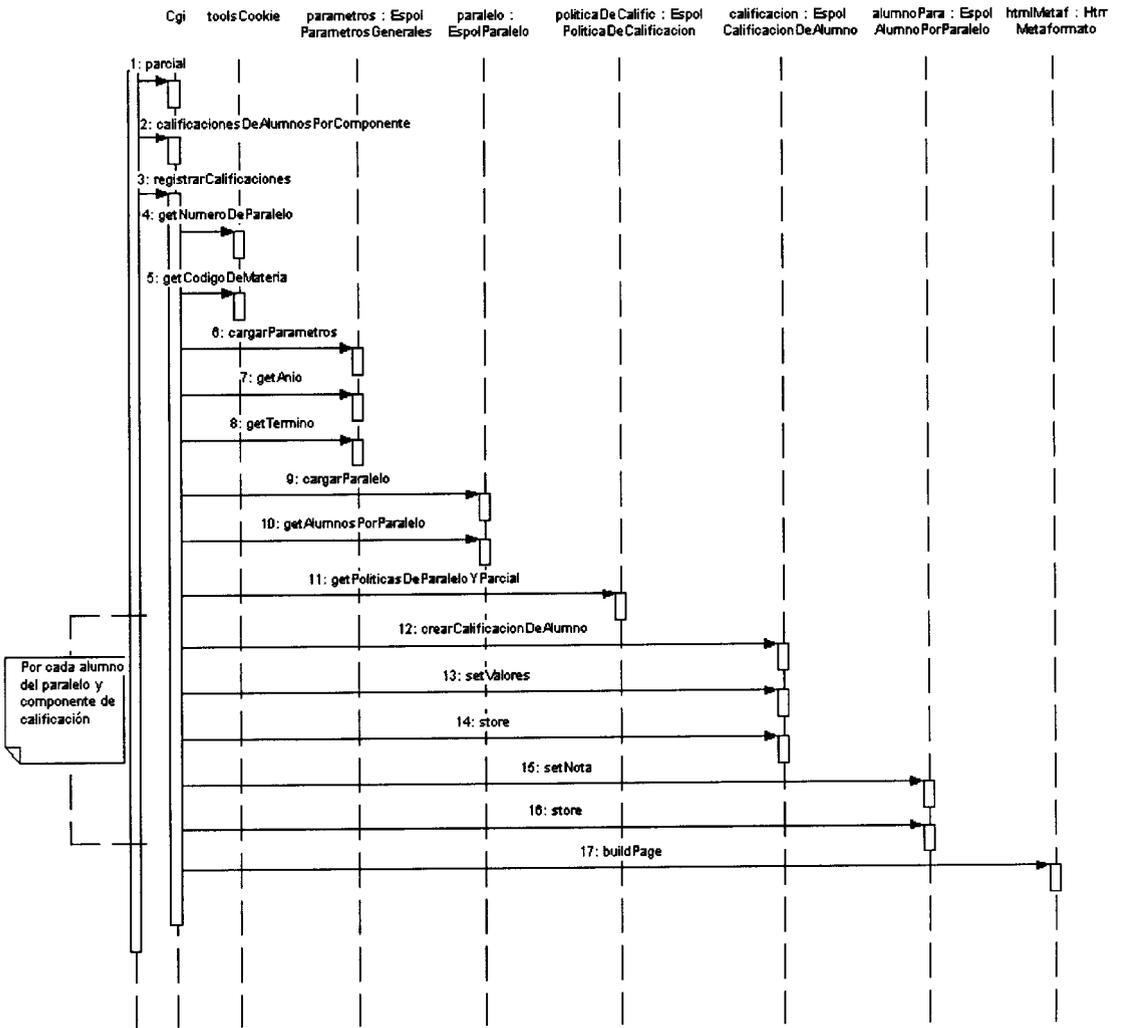


Escenario 10.3: El profesor no puede publicar el componente de la calificación porque la suma de los porcentajes de los componentes de calificación del parcial sobrepasa el 100%.

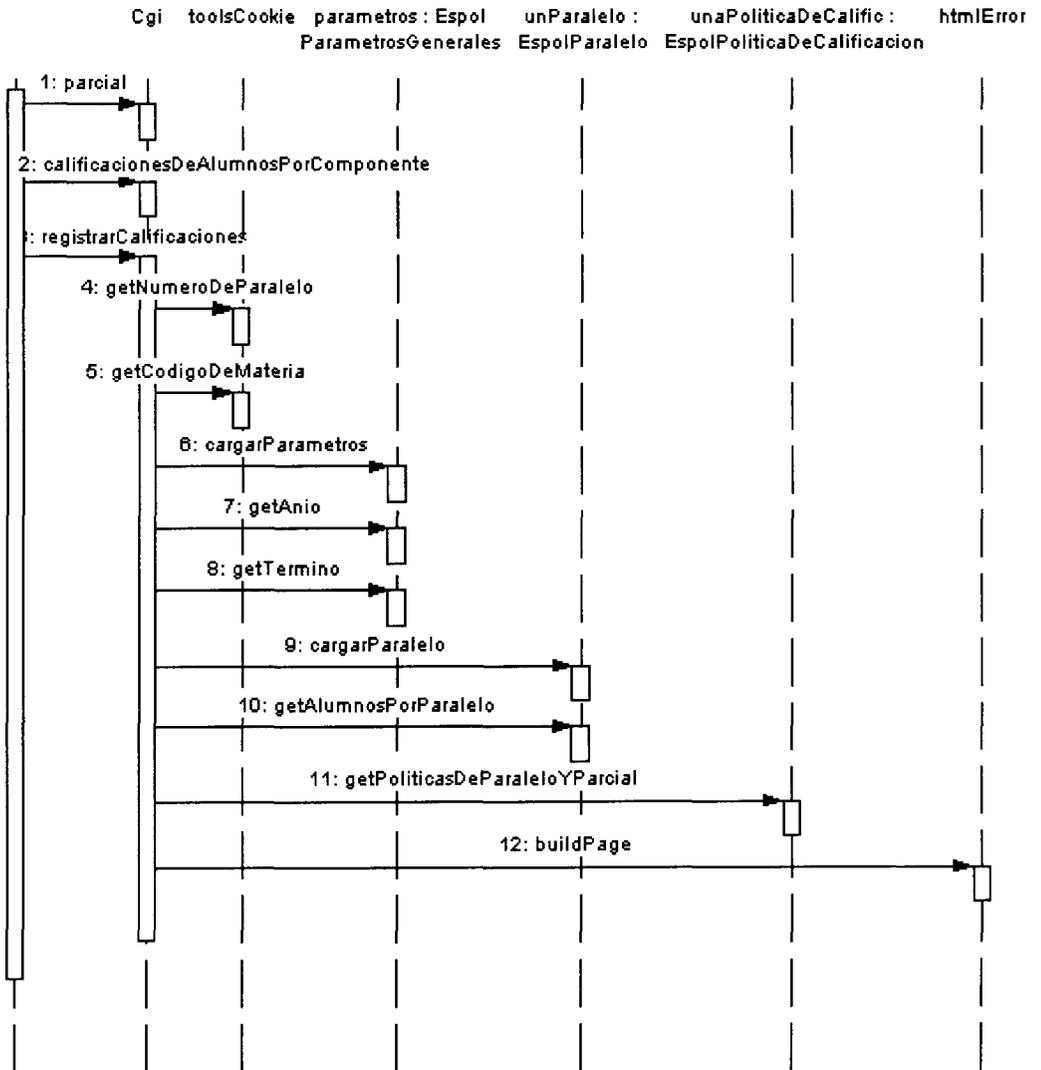
Cgi toolsCookie parametros:Espol unParalelo : unComponente : Espol tools htmlError ParametrosGenerales EspolParalelo PoliticaDeCalificacion



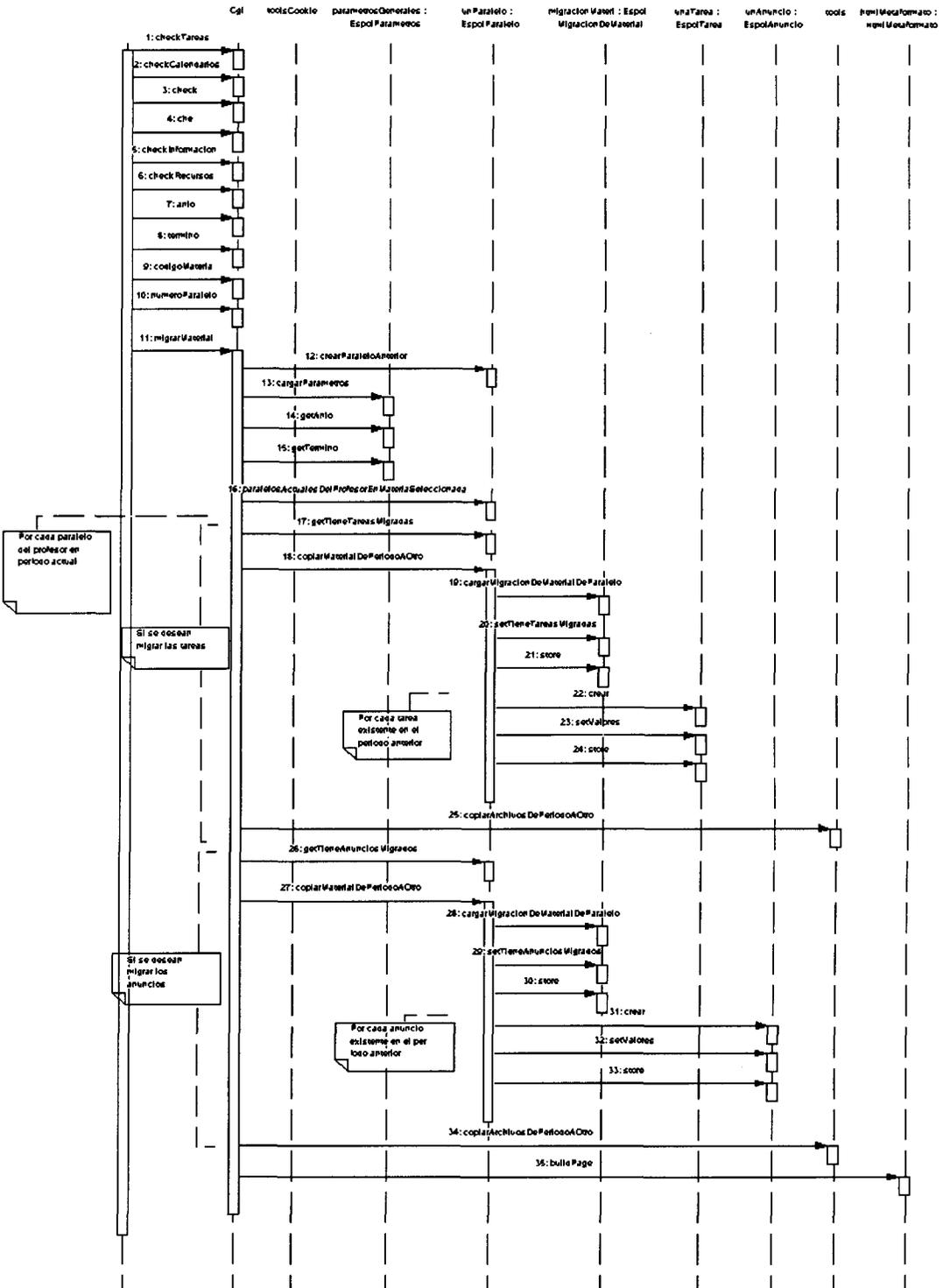
Escenario 11.1: El profesor publica las calificaciones de los componentes definidos de manera exitosa.



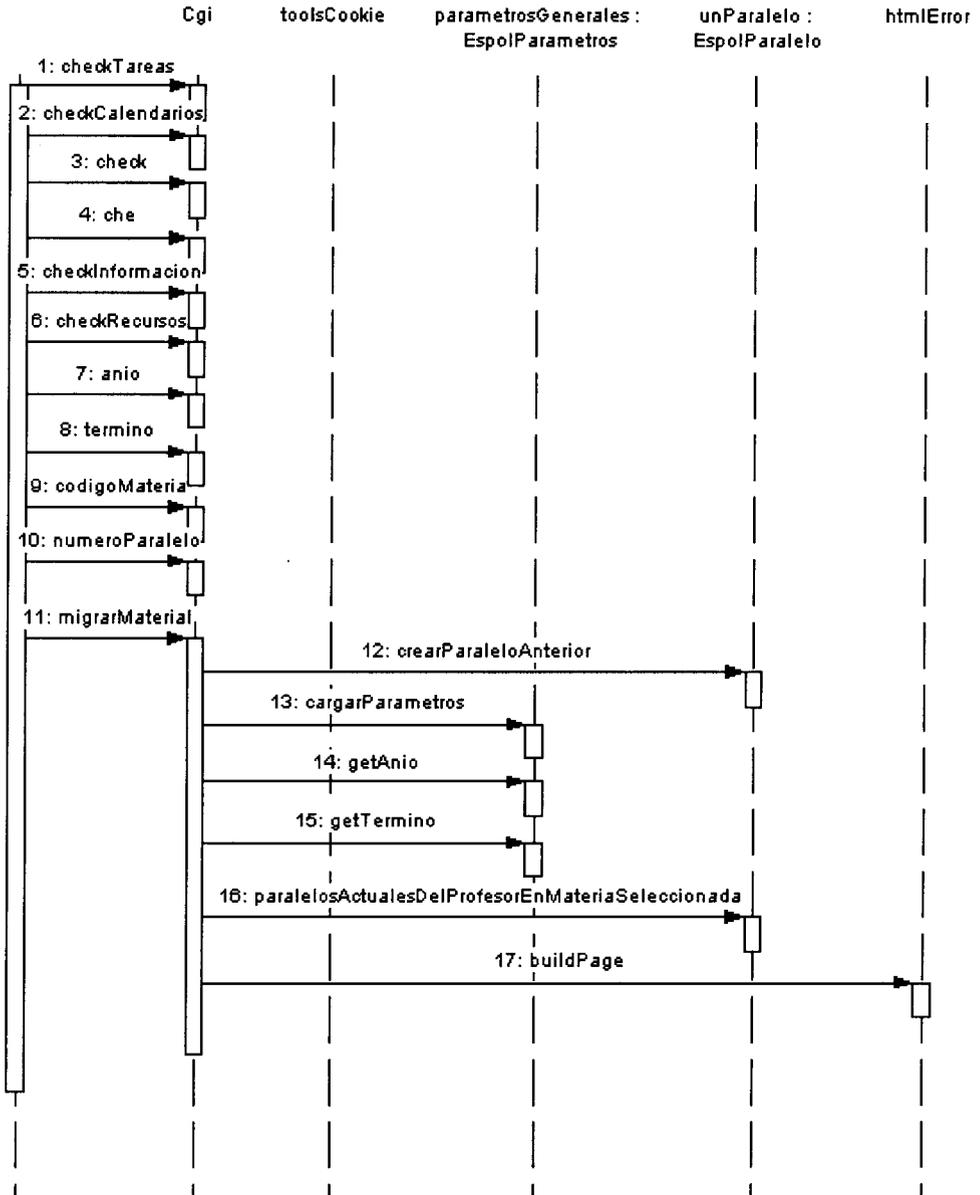
Escenario 11.2: El profesor no puede ingresar las calificaciones porque no existen componentes de calificaciones definidos para ese parcial.



Escenario 12.1: El profesor migra el material existente estocadamente.

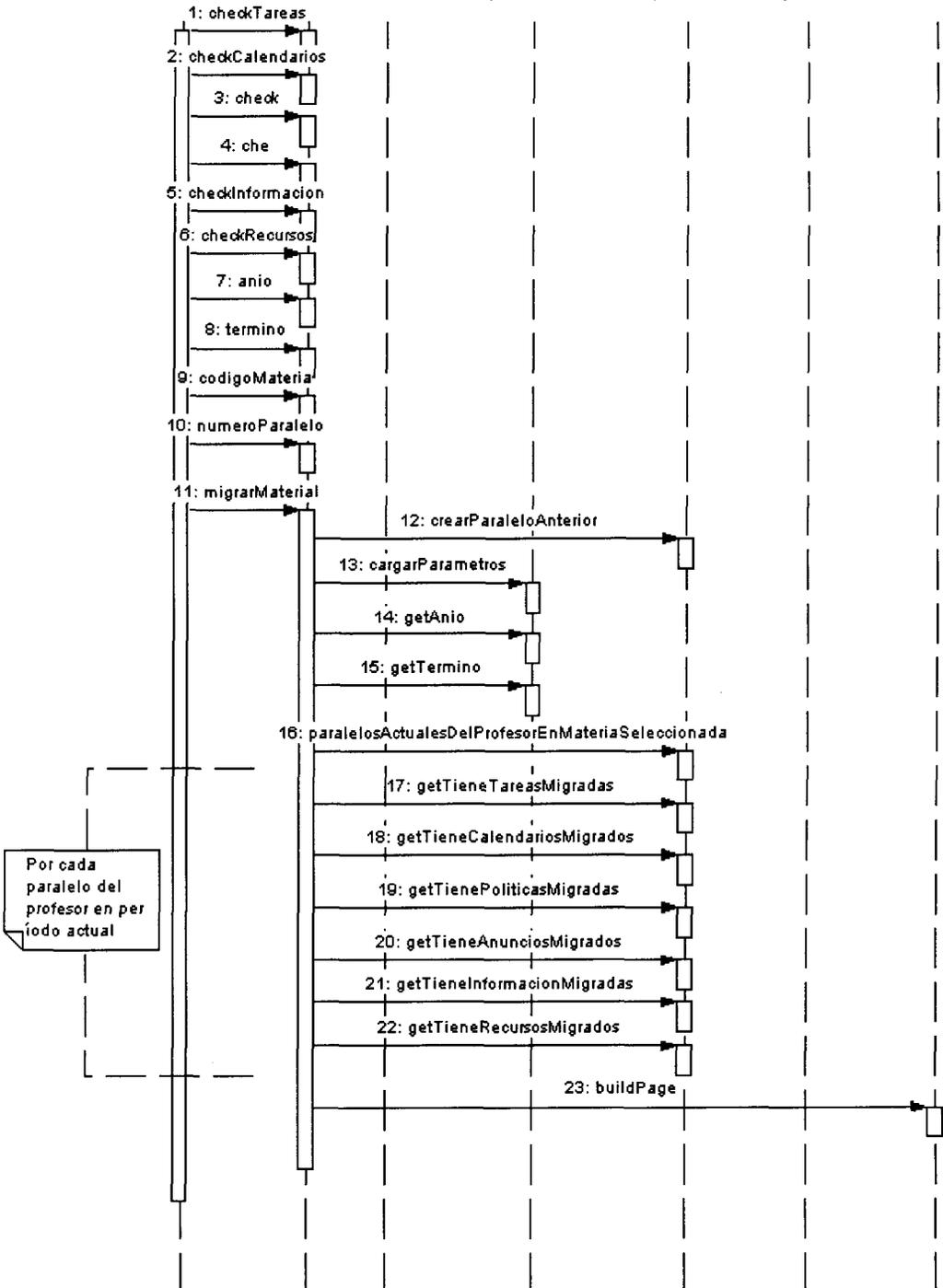


Escenario 12.2: El profesor desea migrar el material de una materia que no está dictando en el período actual.



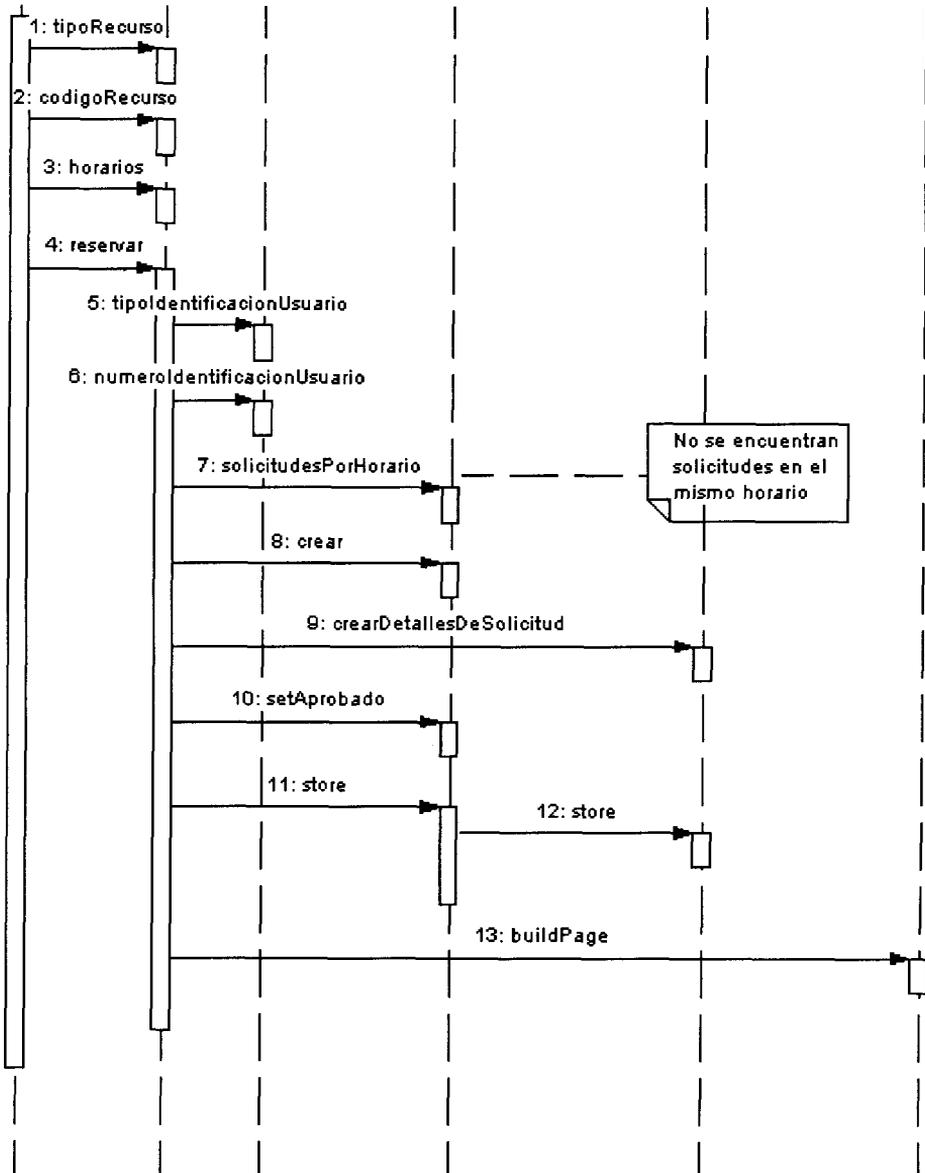
Escenario 12.3: El profesor desea migrar el material de una materia que ya ha migrado anteriormente.

Cgi toolsCookie parametrosGenerales : unParalelo : unaTarea : htmlMetaformato :
EspolParametros EspolParalelo EspolTarea HtmlMetaformato

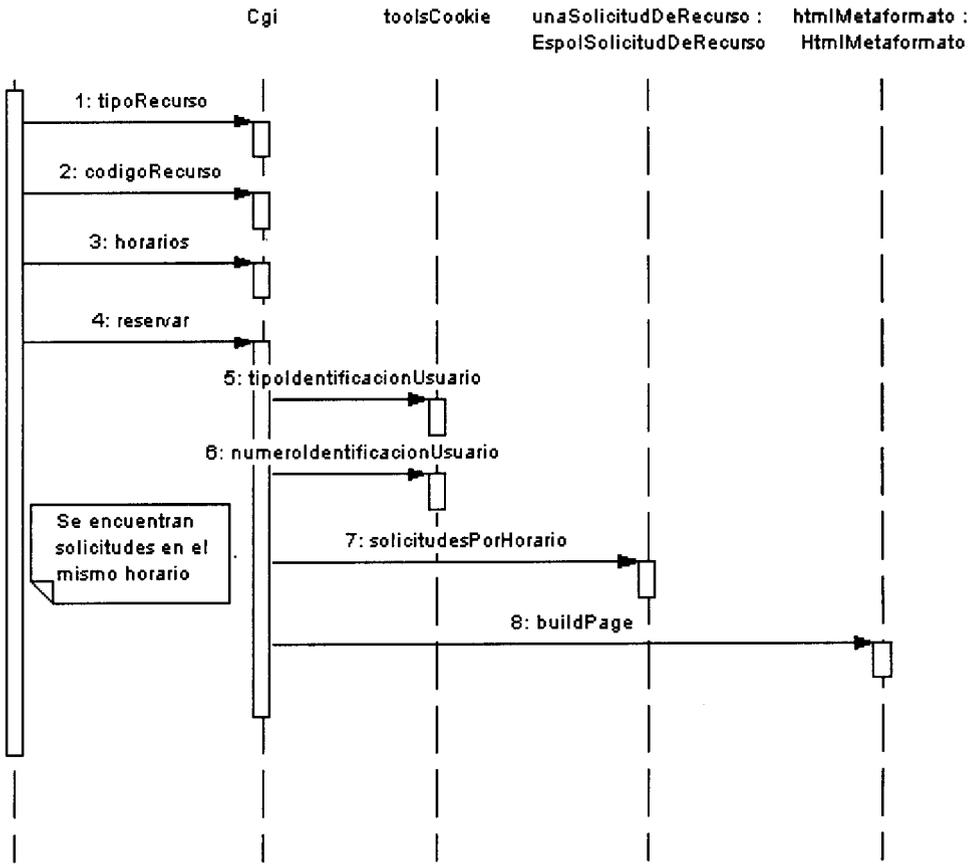


Escenario 13.1: Una persona reserva un recurso físico de manera exitosa.

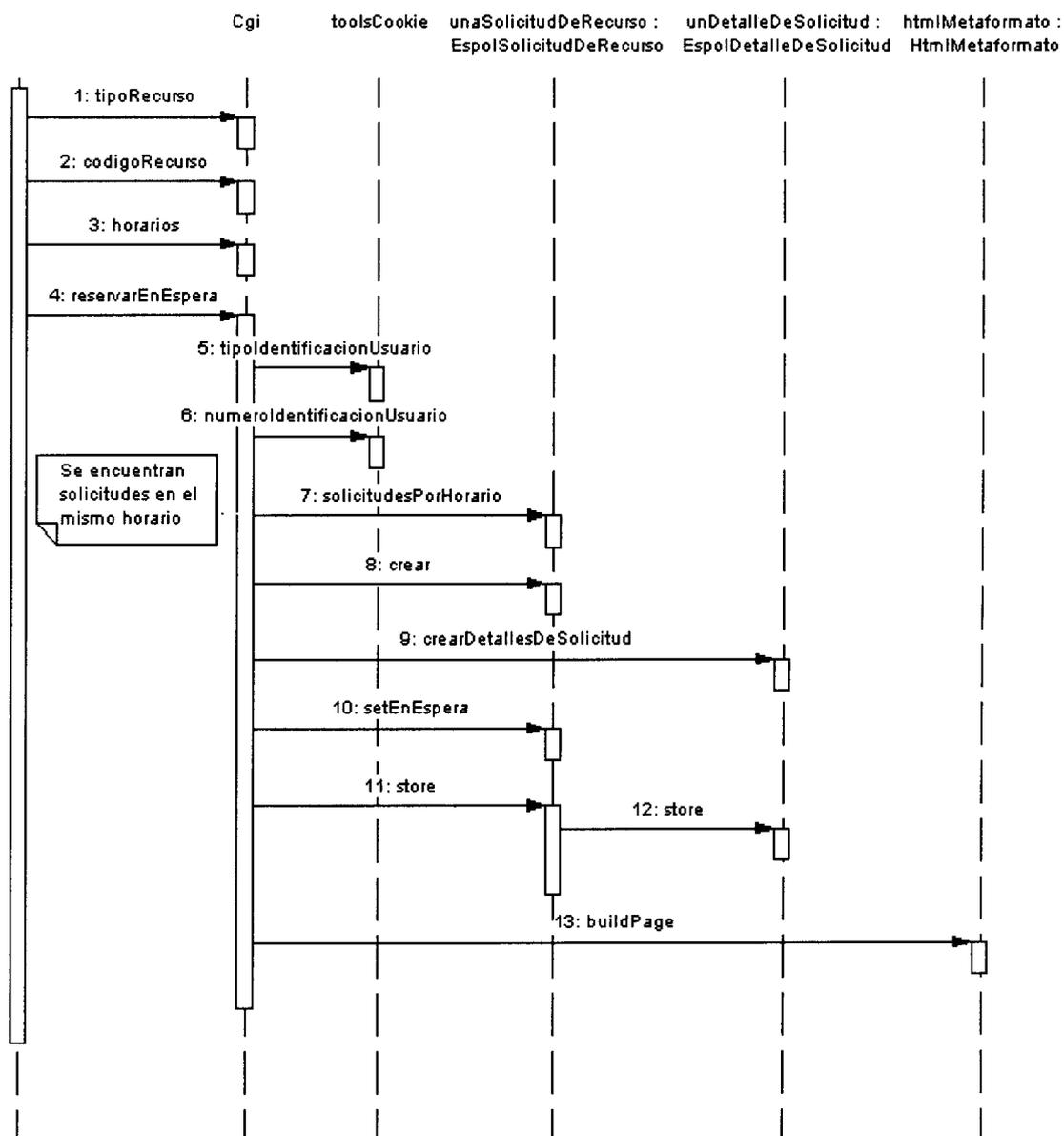
Cgi toolsCookie unaSolicitudDeRecurso : unDetalleDeSolicitud : htmlMetaformato :
EspolSolicitudDeRecurso EspolDetalleDeSolicitud HtmlMetaformato



Escenario 13.2: Una persona no reserva el recurso físico porque no se encuentra disponible.

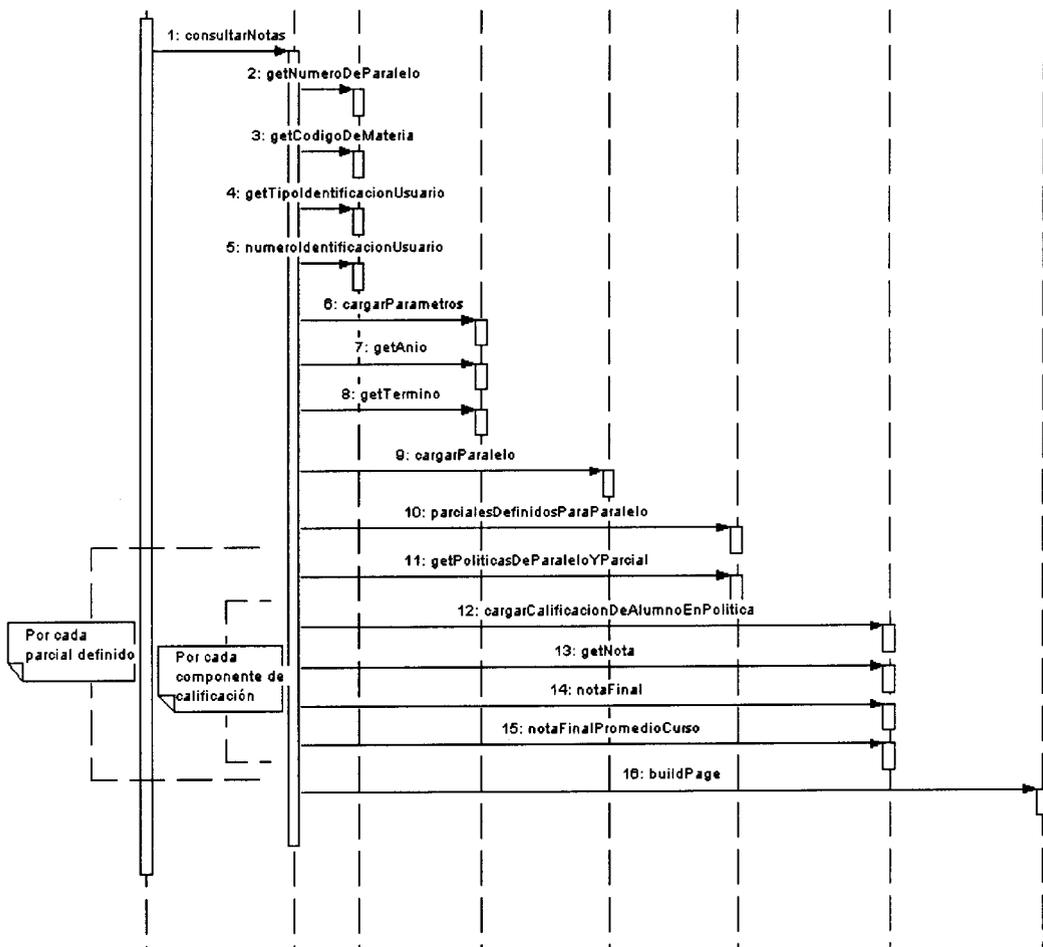


Escenario 13.3: La solicitud de recurso físico queda en lista de espera porque no se encuentra disponible.



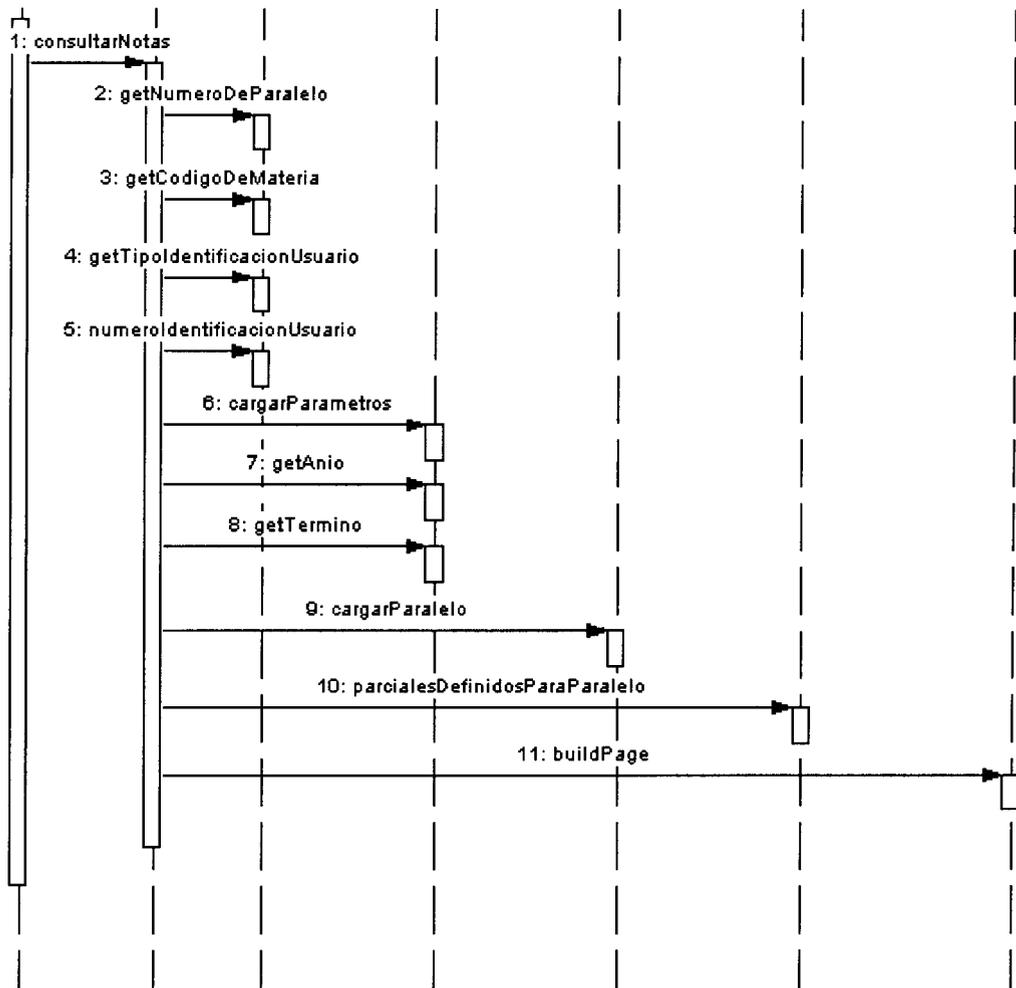
Escenario 14.1: Un estudiante consulta las calificaciones obtenidas en los componentes de calificación.

Cgi toolsCookie parametros:Espol unParalelo: unaPolitica:Espol unaCalific:Espol htmlMetaformato: ParametrosGeneralesEspolParalelo PoliticaDeCalificacionCalificacionDeAlumno HtmlMetaformato



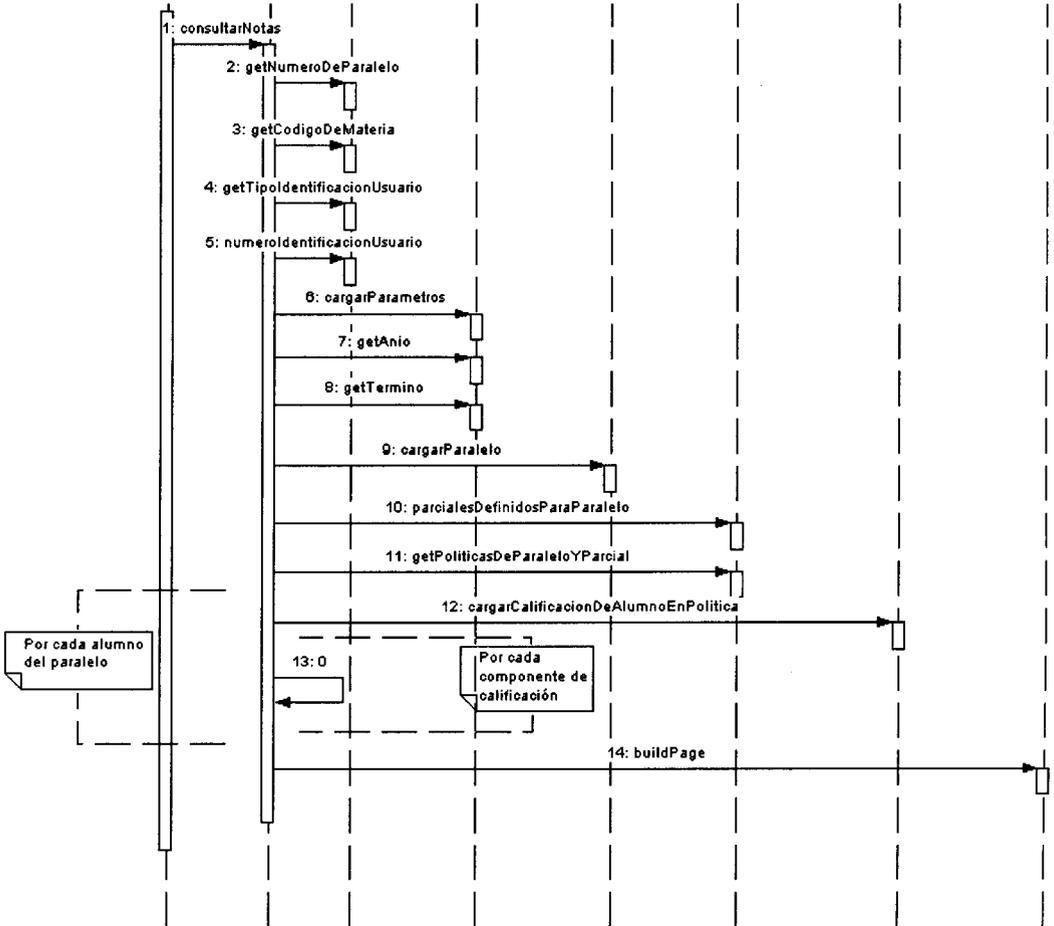
Escenario 14.2: Un estudiante no puede consultar las calificaciones obtenidas porque no existen definidos en el sistema los componentes de calificación para ninguna nota.

Cgi toolsCookie parametros: Espol unParalelo: unaPolitica: Espol htmlMetaformato: ParametrosGenerales EspolParalelo PoliticaDeCalificacion HtmlMetaformato

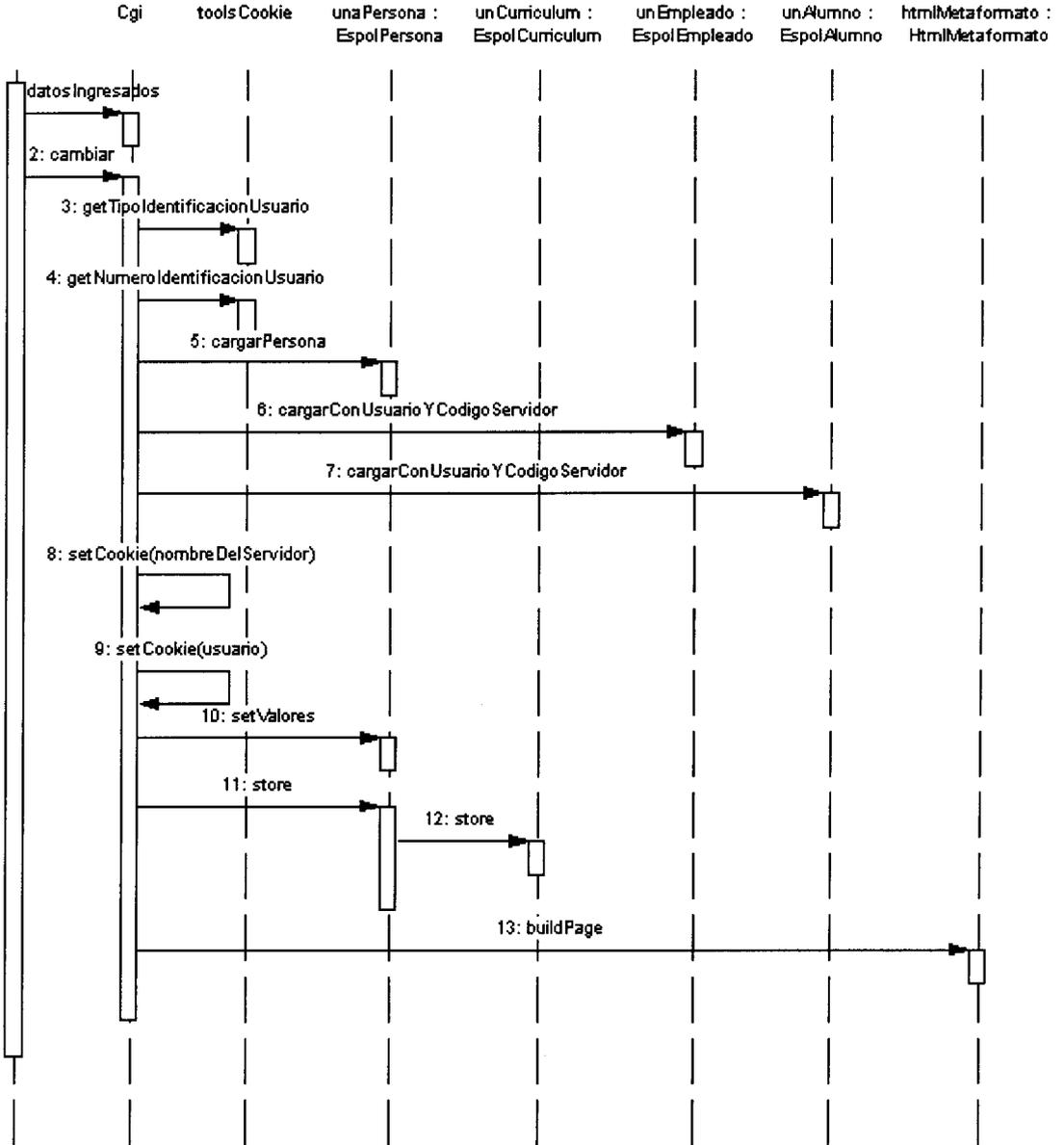


Escenario 14.3: Un estudiante no puede consultar las calificaciones obtenidas porque no han sido ingresadas al sistema.

Cgi toolsCookie parametros : Espol unParalelo : unaPolitica : Espol unaCalificacion : EspolhtmlMetaformato : ParametrosGenerales EspolParaleloPoliticaDeCalificacion CalificacionDeAlumno HtmlMetaformato

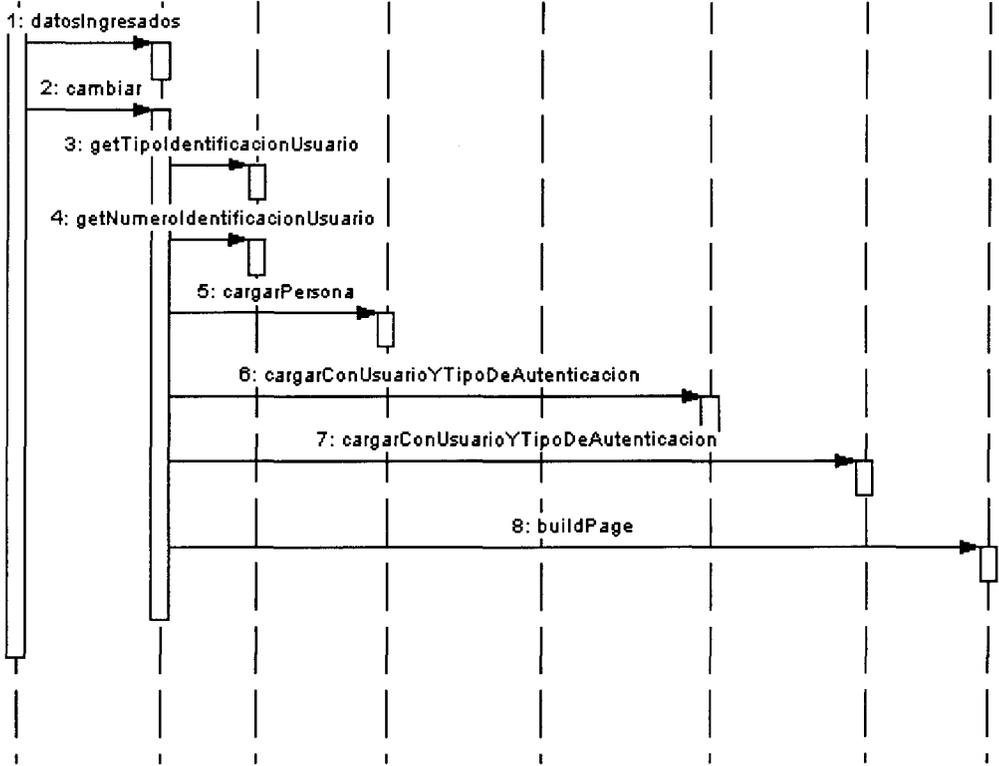


Escenario 15.1: La información personal de una persona es modificada exitosamente.

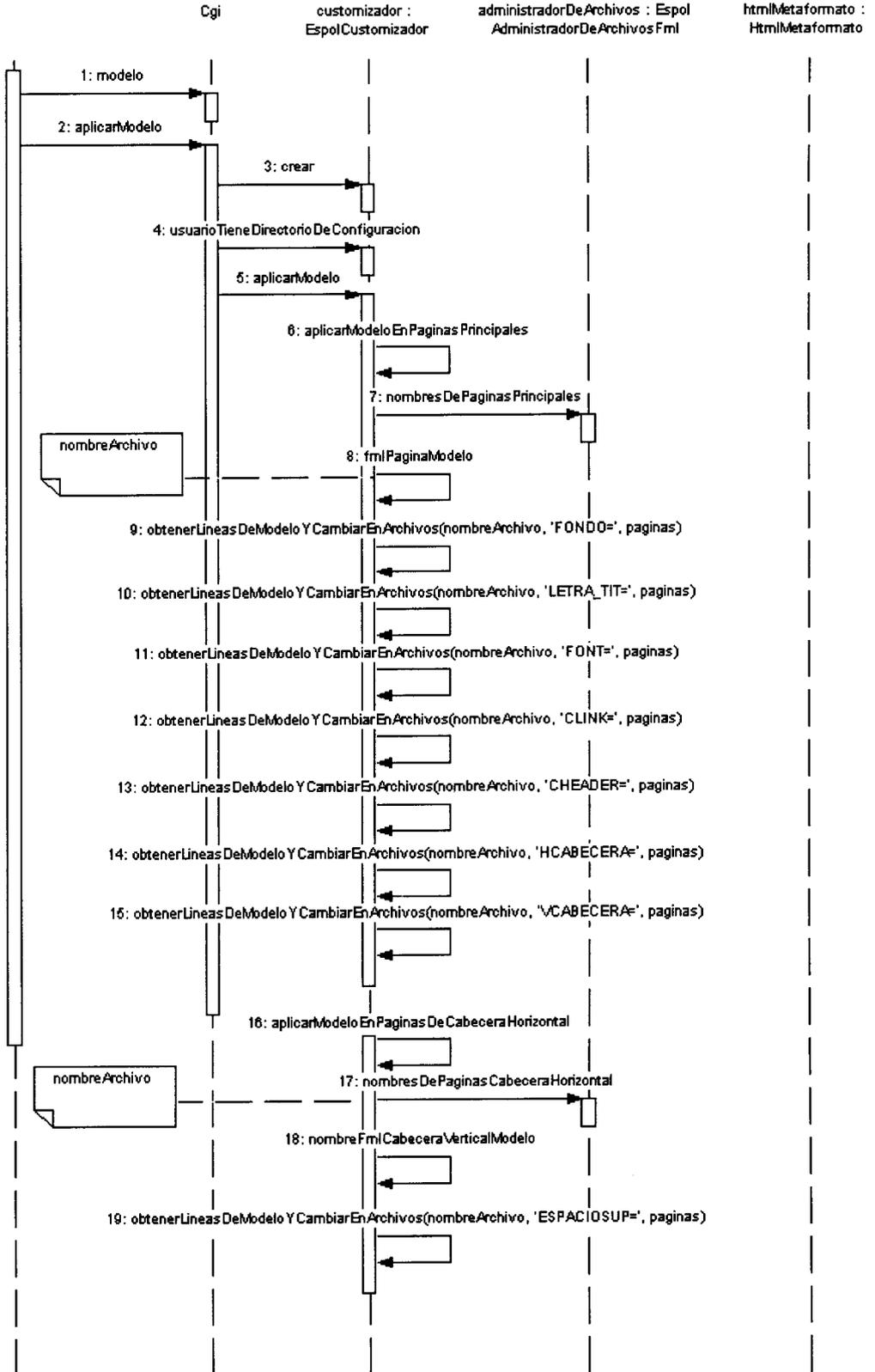


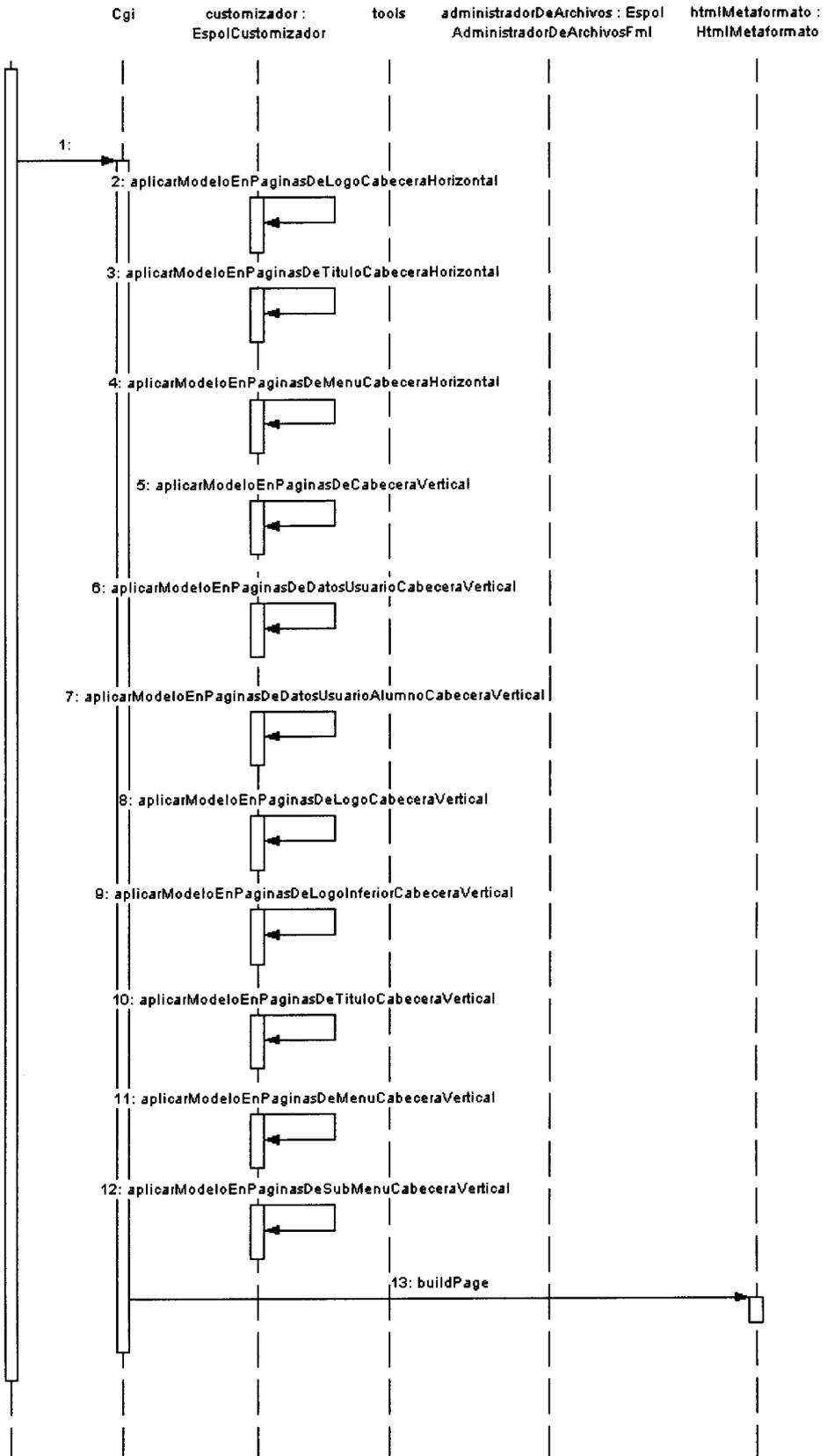
Escenario 15.2: La información personal de una persona no es modificada porque ha definido un servidor de autenticación y un nombre de usuario de acceso que ya existe para otra persona.

Cgi toolsCookie unaPersona : unCurriculum : unEmpleado : unAlumno : htmlError
EspolPersona EspolCurriculum EspolEmpleado EspolAlumno

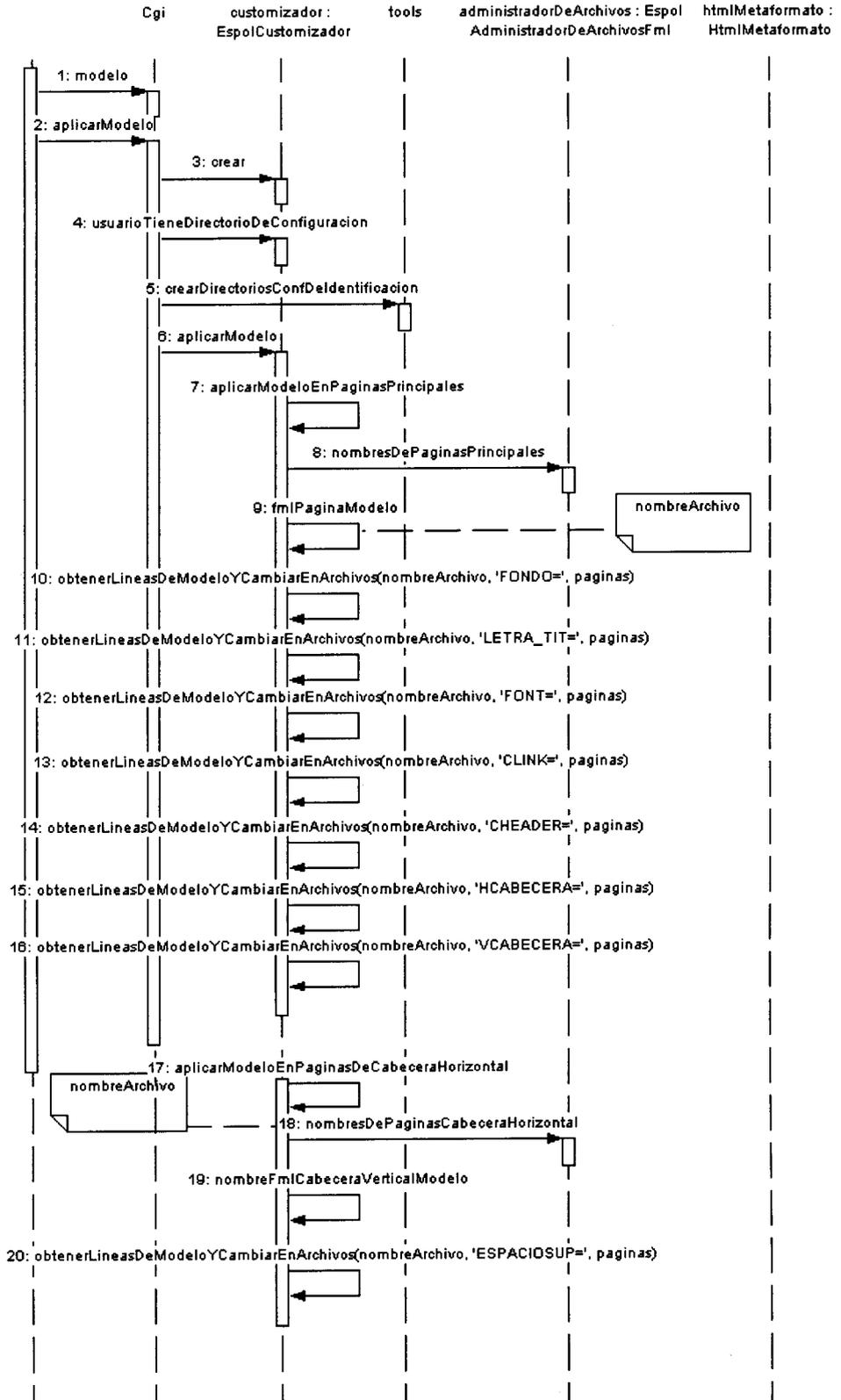


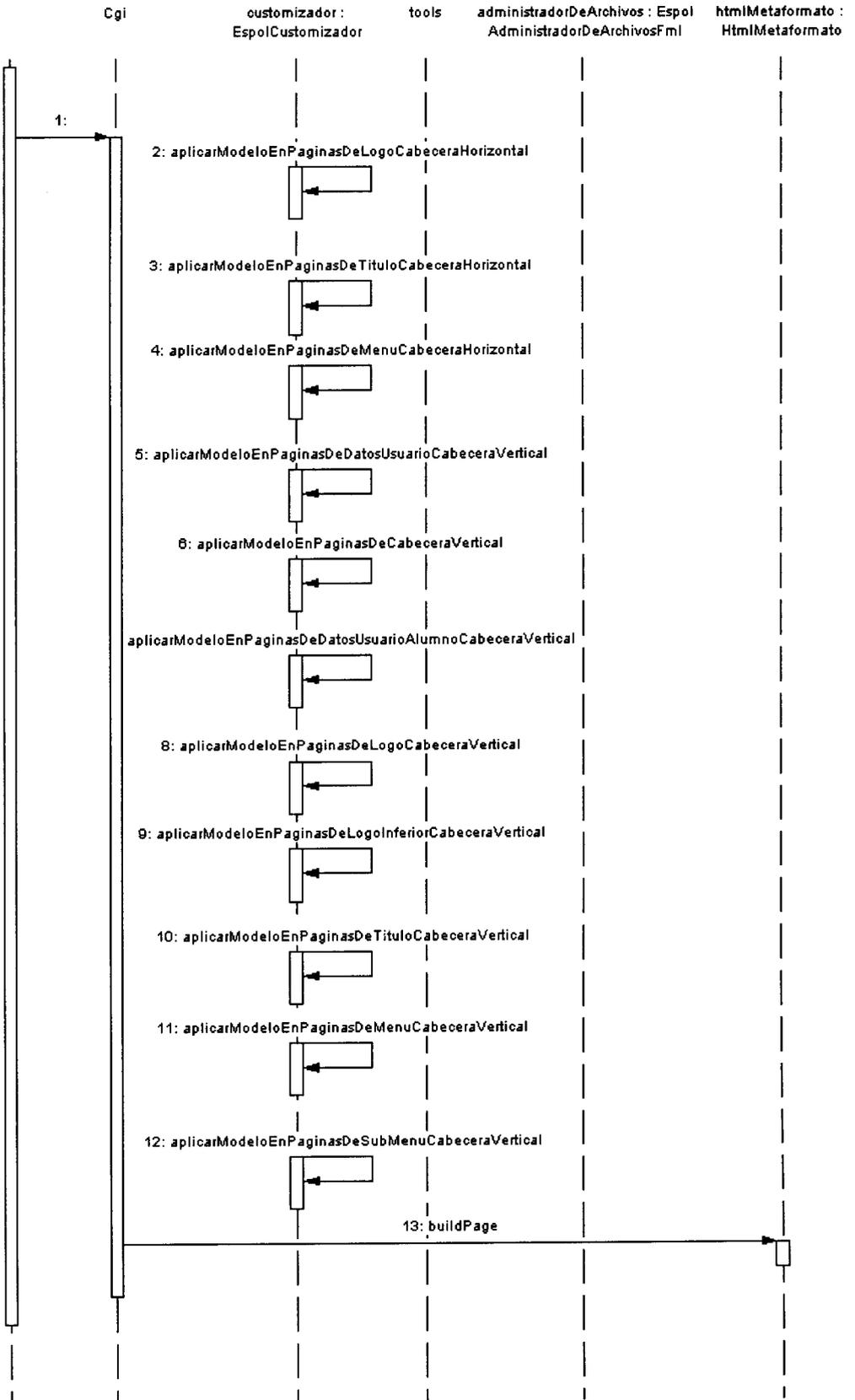
Escenario 16.1: El usuario tiene un directorio de configuración de páginas html, selecciona una plantilla y el cambio es realizado exitosamente.



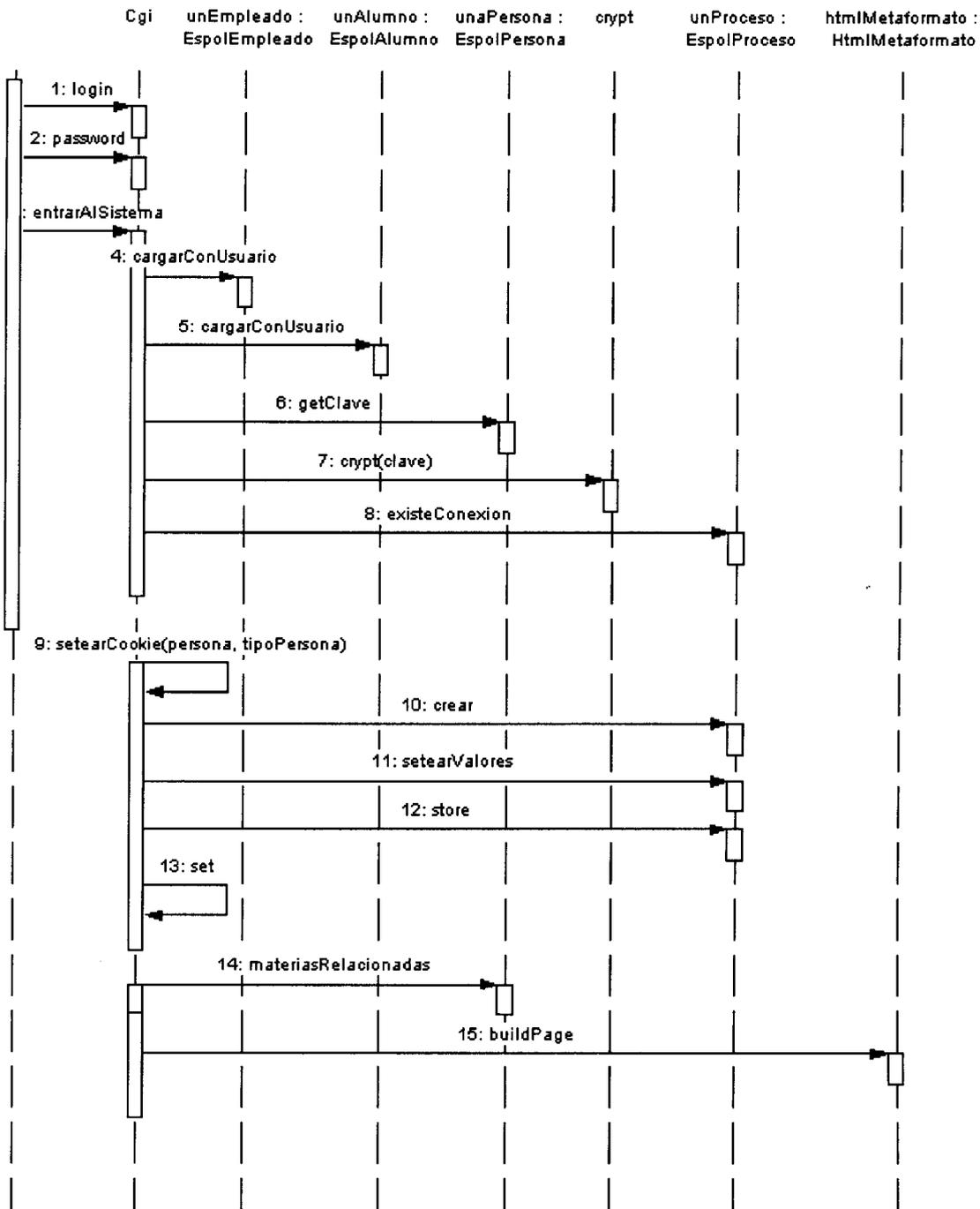


Escenario 16.2: El usuario no tiene un directorio de configuración de páginas html, selecciona una plantilla y el cambio es realizado exitosamente.

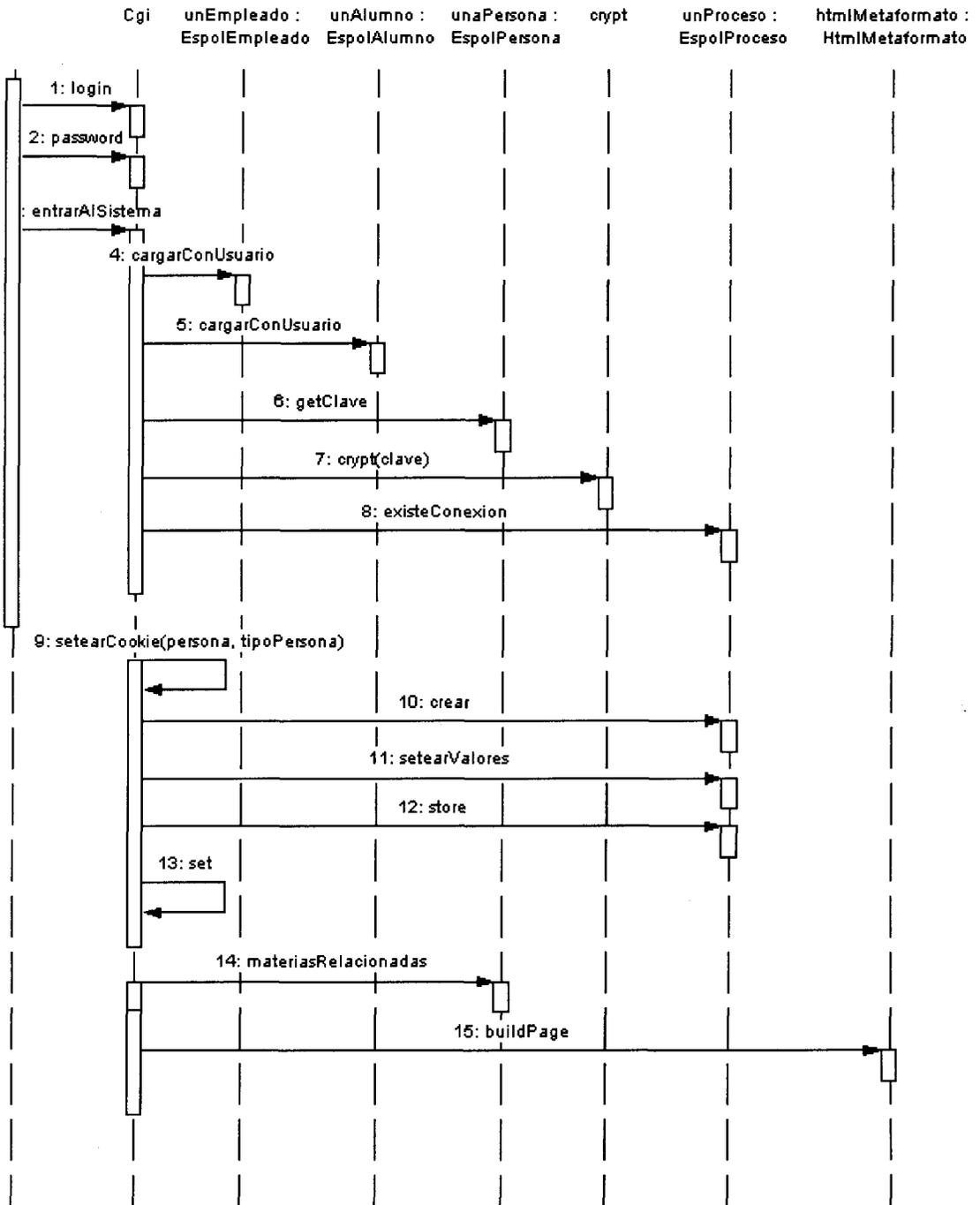




Escenario 17.1: Acceso otorgado a persona cuyo nombre de usuario es único, con contraseña correcta y sin conexiones abiertas.

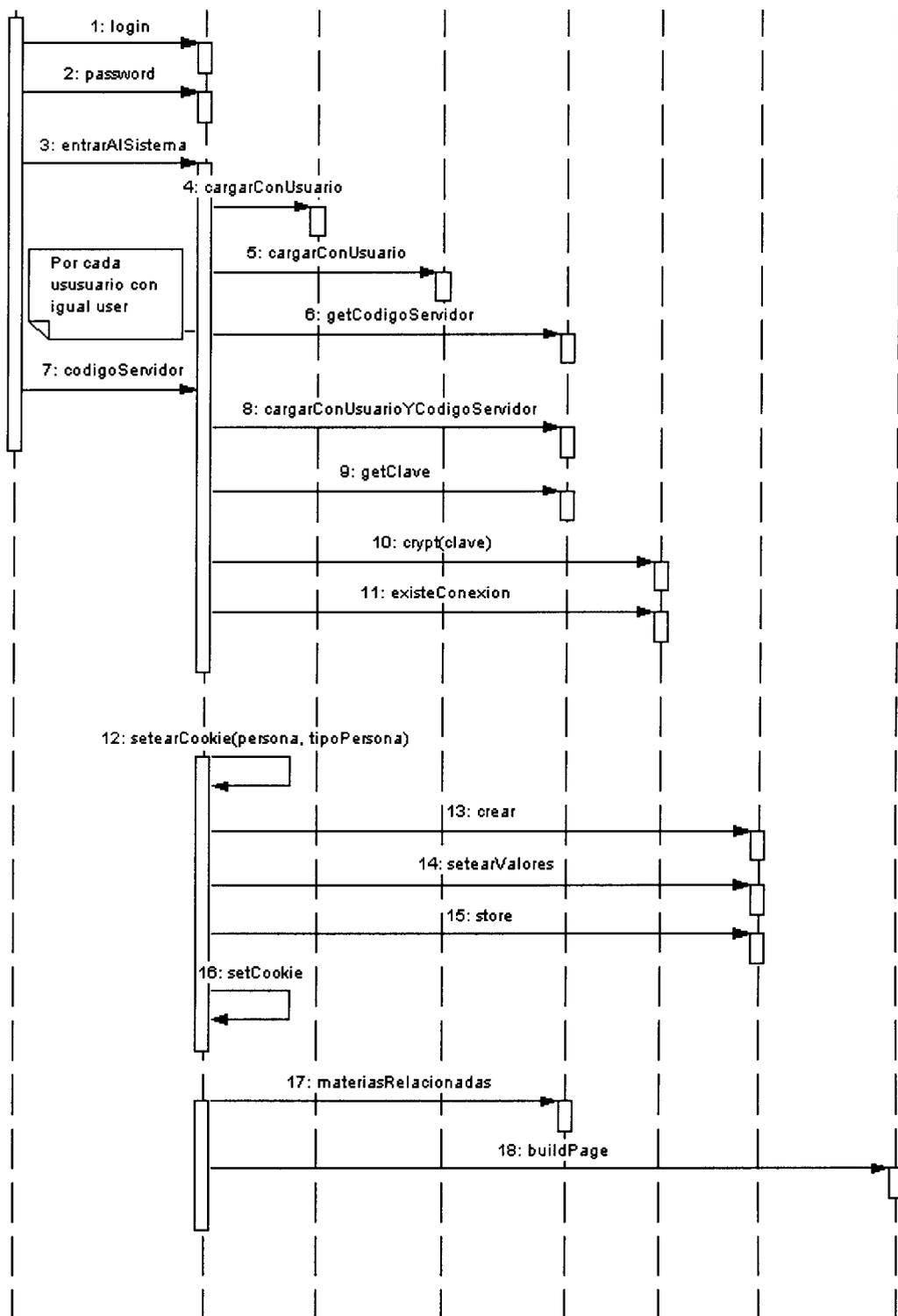


Escenario 17.1: Acceso otorgado a persona cuyo nombre de usuario es único, con contraseña correcta y sin conexiones abiertas.

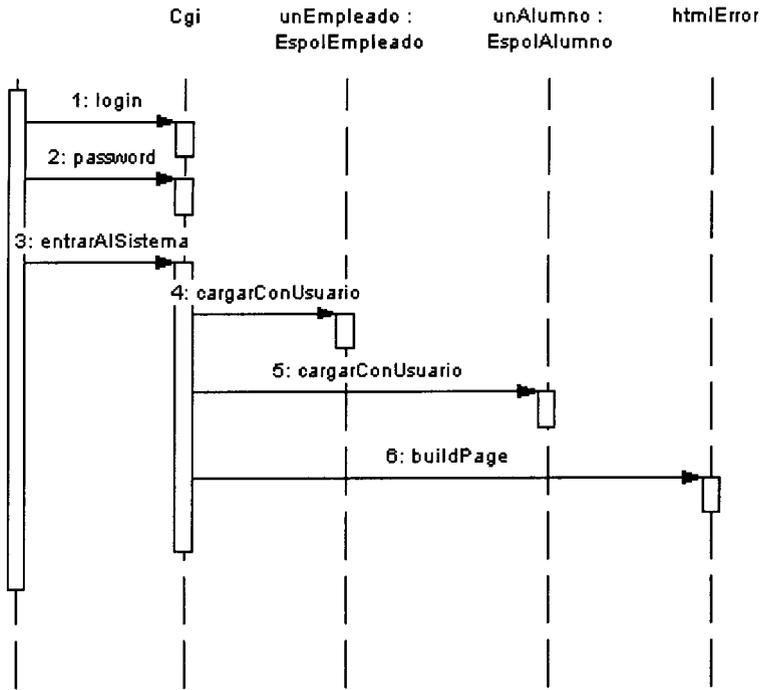


Escenario 17.3: Acceso otorgado a persona cuyo nombre de usuario no es único con contraseña correcta.

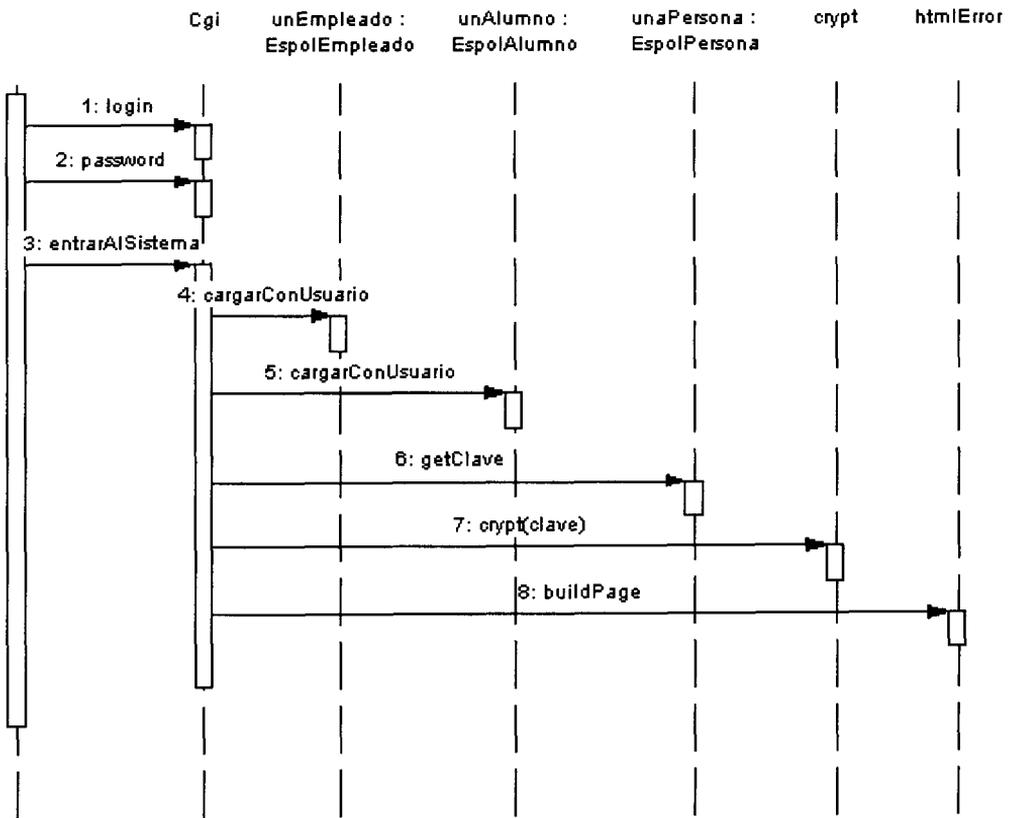
Cgi unEmpleado : unAlumno : unaPersona : crypt unProceso : htmlMetaformato :
EspolEmpleado EspolAlumno EspolPersona EspolProceso HtmlMetaformato



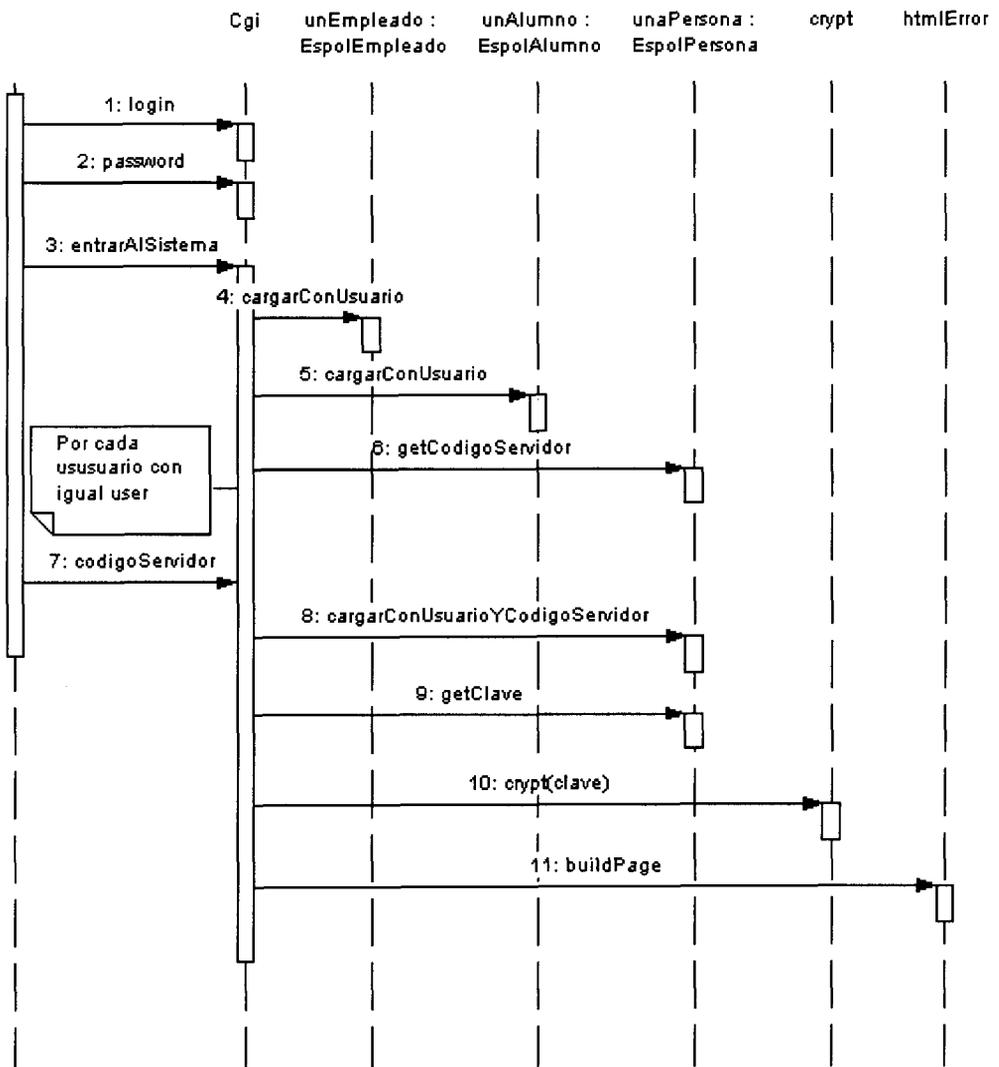
Escenario 17.4: Acceso no otorgado a persona cuyo nombre de usuario no existe



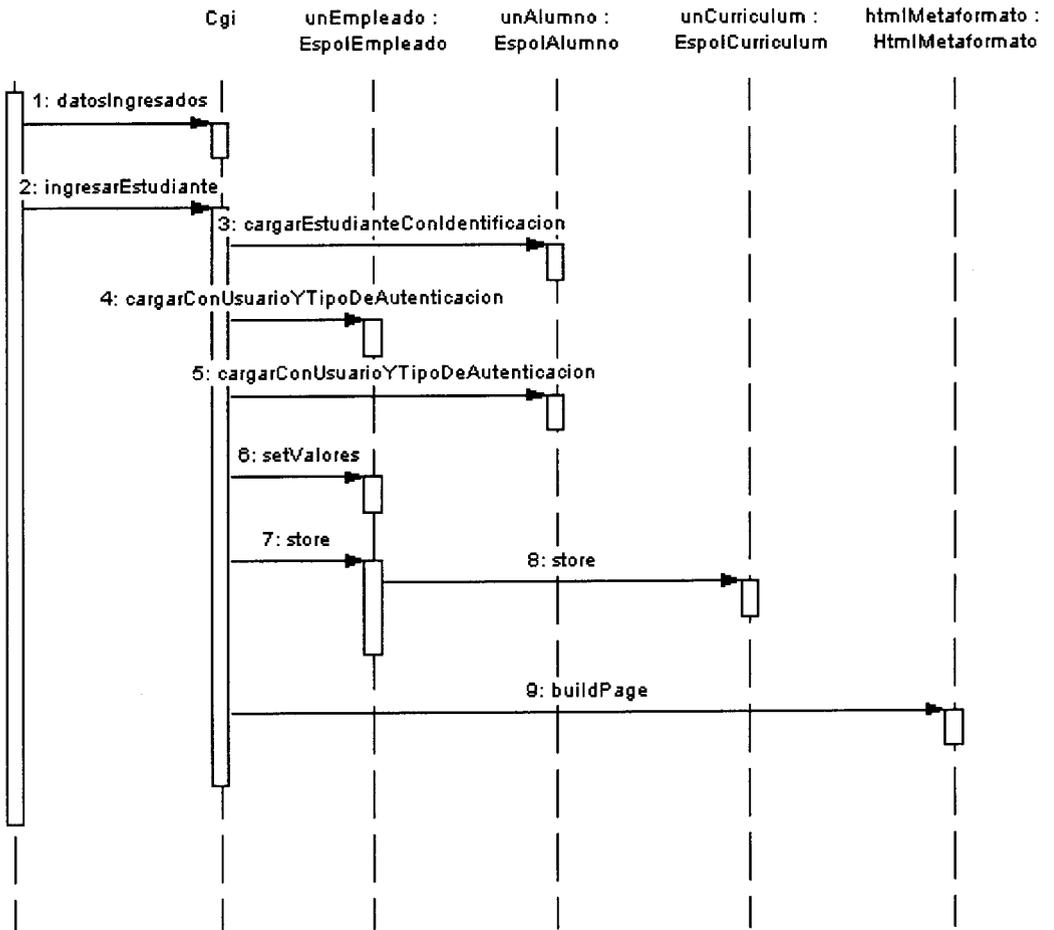
Escenario 17.5: Acceso no otorgado a persona cuyo nombre de usuario tiene contraseña incorrecta.



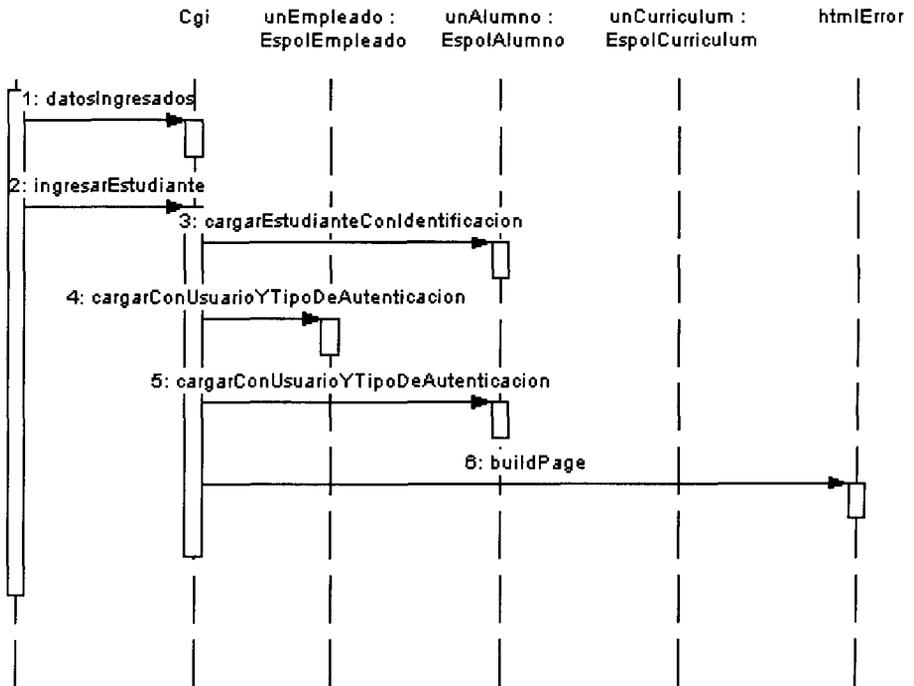
Escenario 17.6: Acceso no otorgado a persona cuyo nombre de usuario no es único con contraseña incorrecta



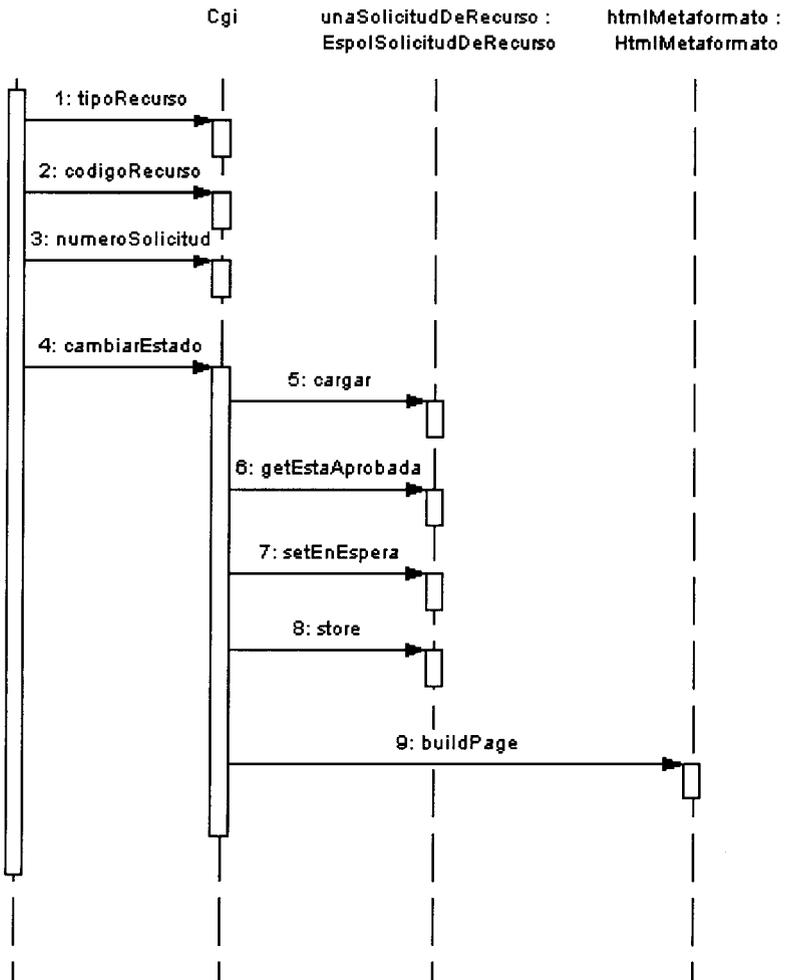
Escenario 18.1: Un estudiante es ingresado exitosamente.



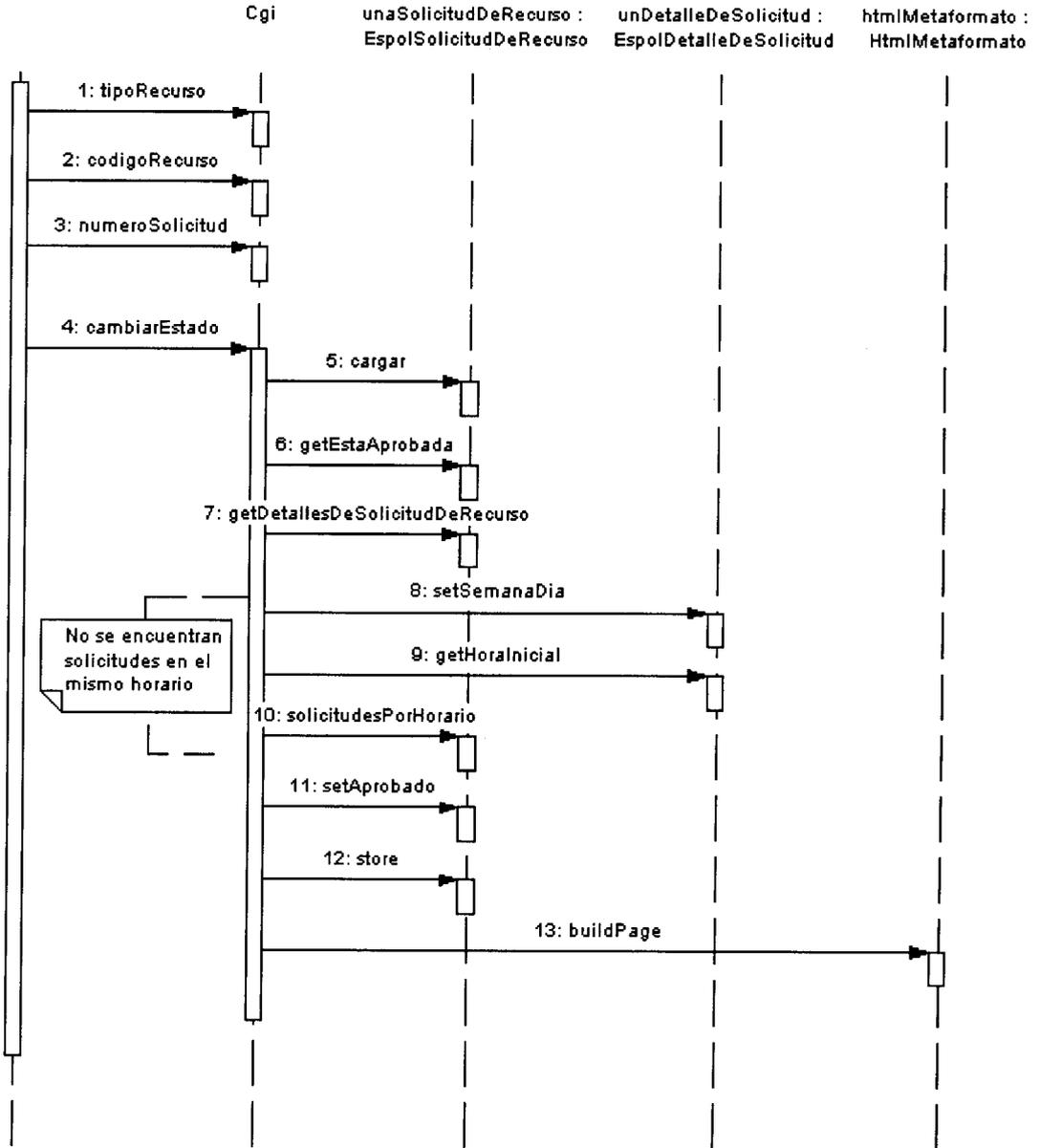
Escenario 18.3: Un estudiante no es ingresado porque se ha definido un servidor de autenticación y un nombre de usuario de acceso que ya existe para otra persona.



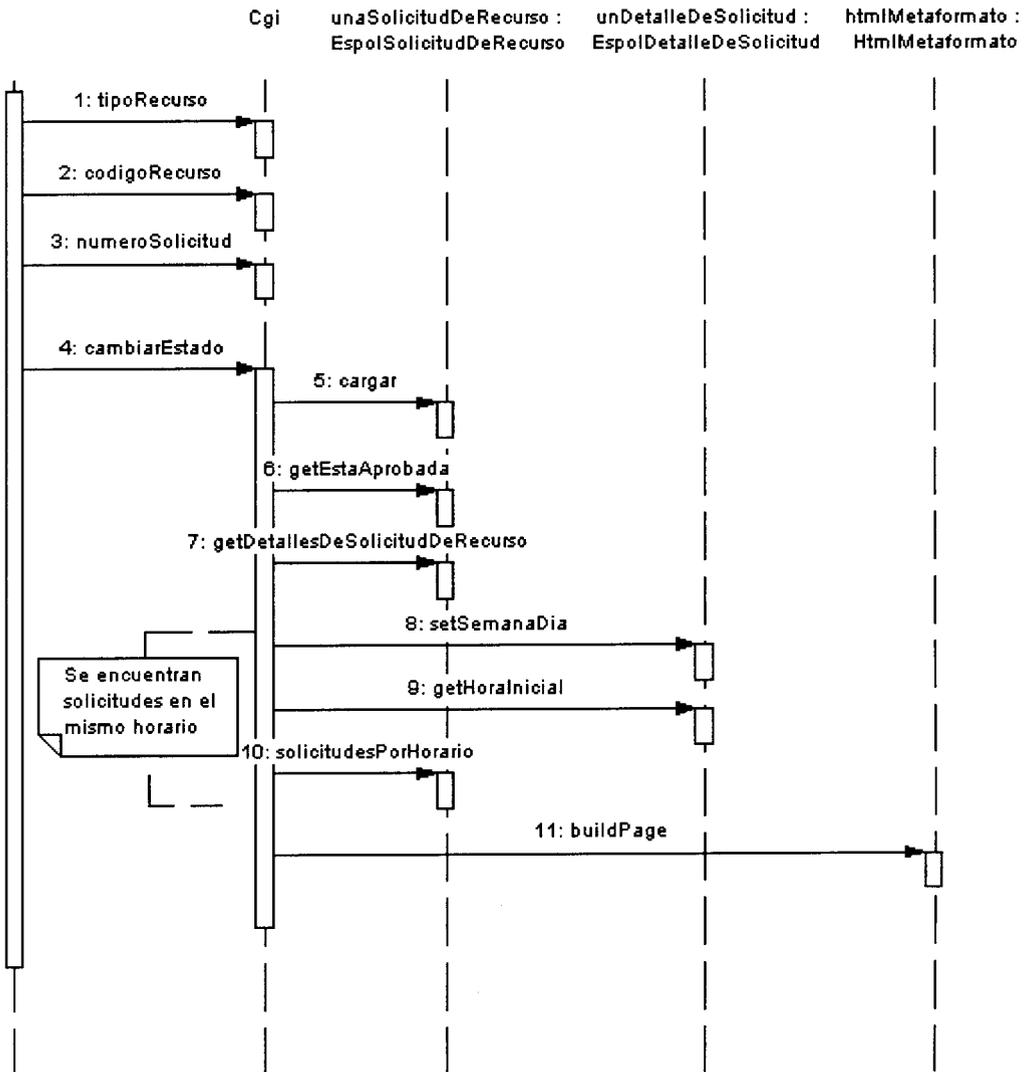
Escenario 19.1: Una solicitud aprobada es puesta en lista de espera exitosamente



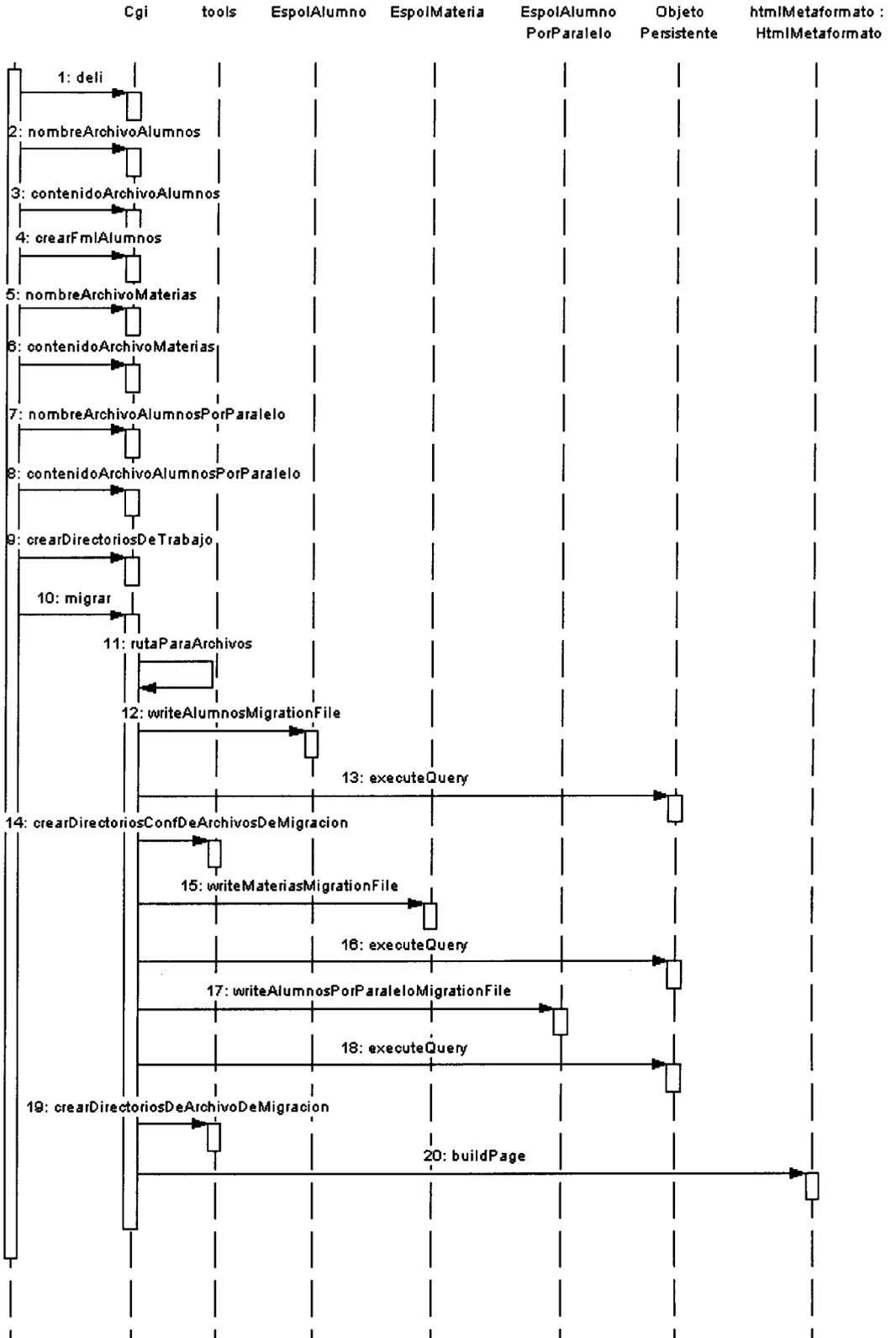
Escenario 19.2: Una solicitud en lista de espera es aprobada exitosamente.



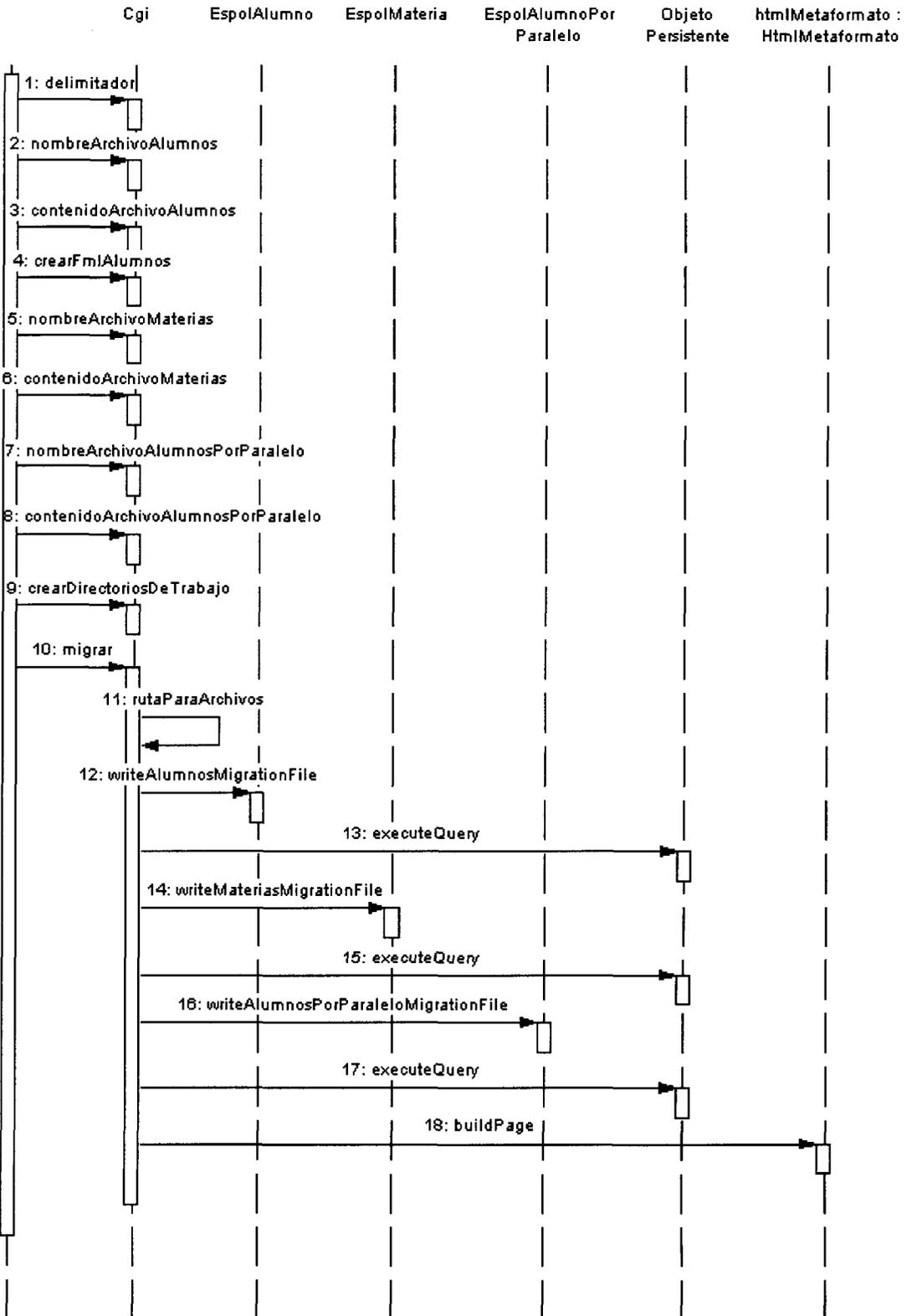
Escenario 19.3: Una solicitud en lista de espera no es aprobada porque el recurso físico no está disponible en todos los horarios de la solicitud.



Escenario 20.1: Los datos se insertan con éxito y los directorios de trabajo son creados.



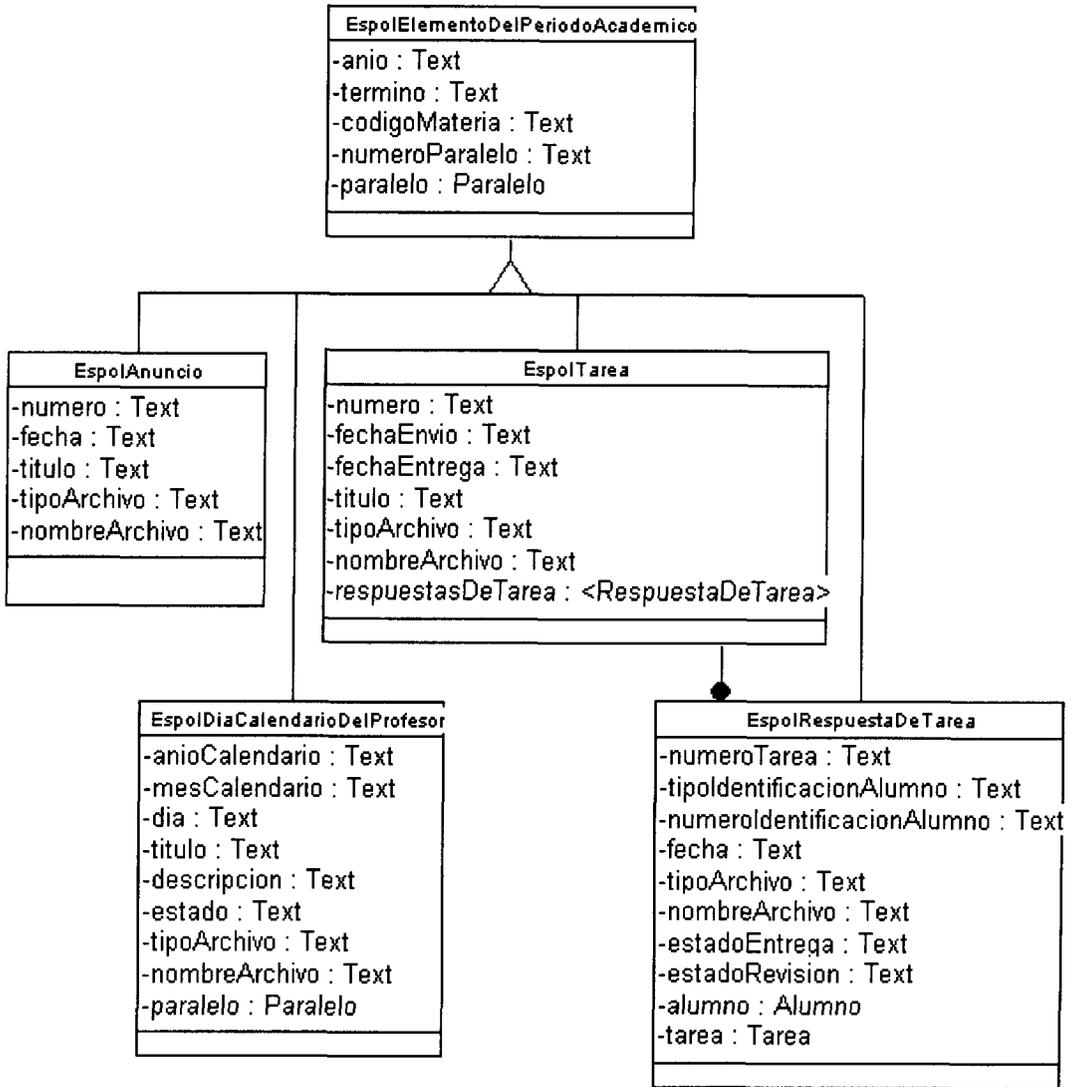
Escenario 20.2: Los datos no son insertados porque estos son inconsistentes con las estructuras de las tablas.

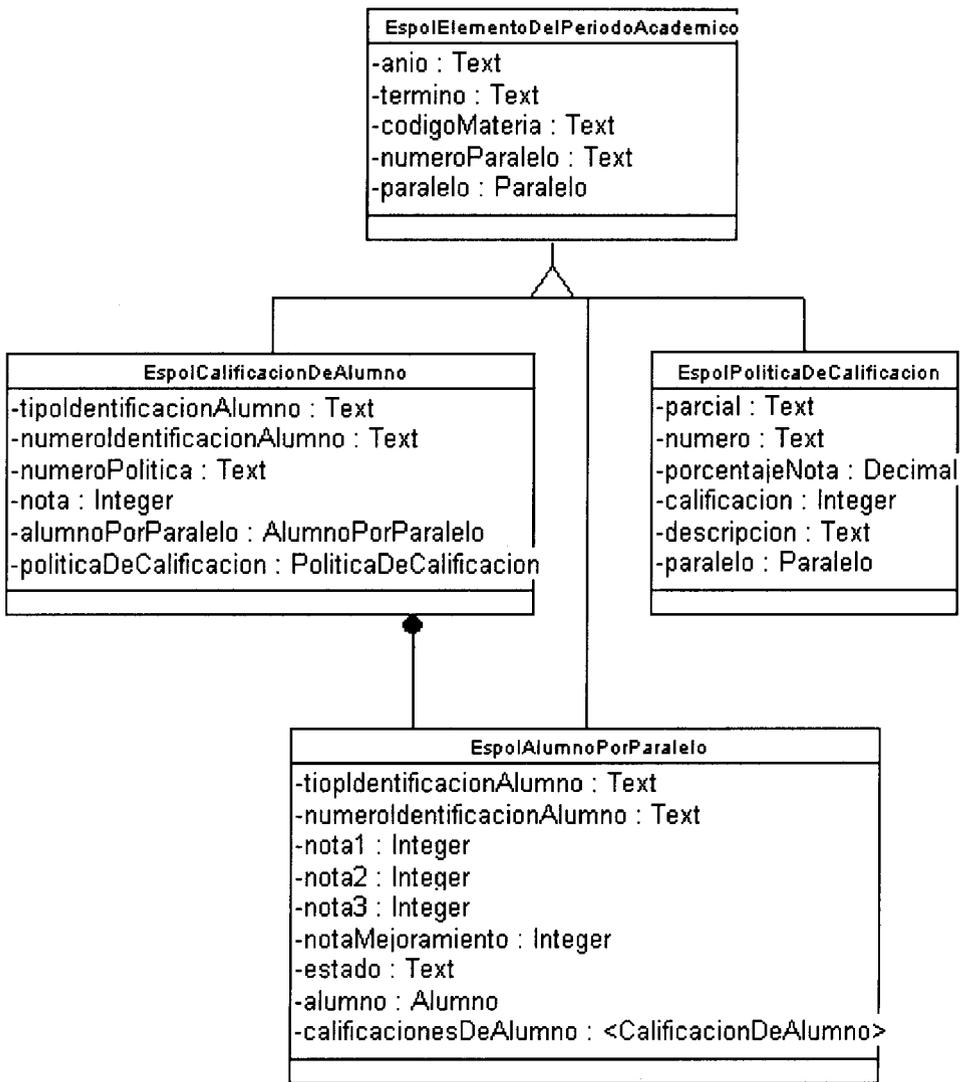


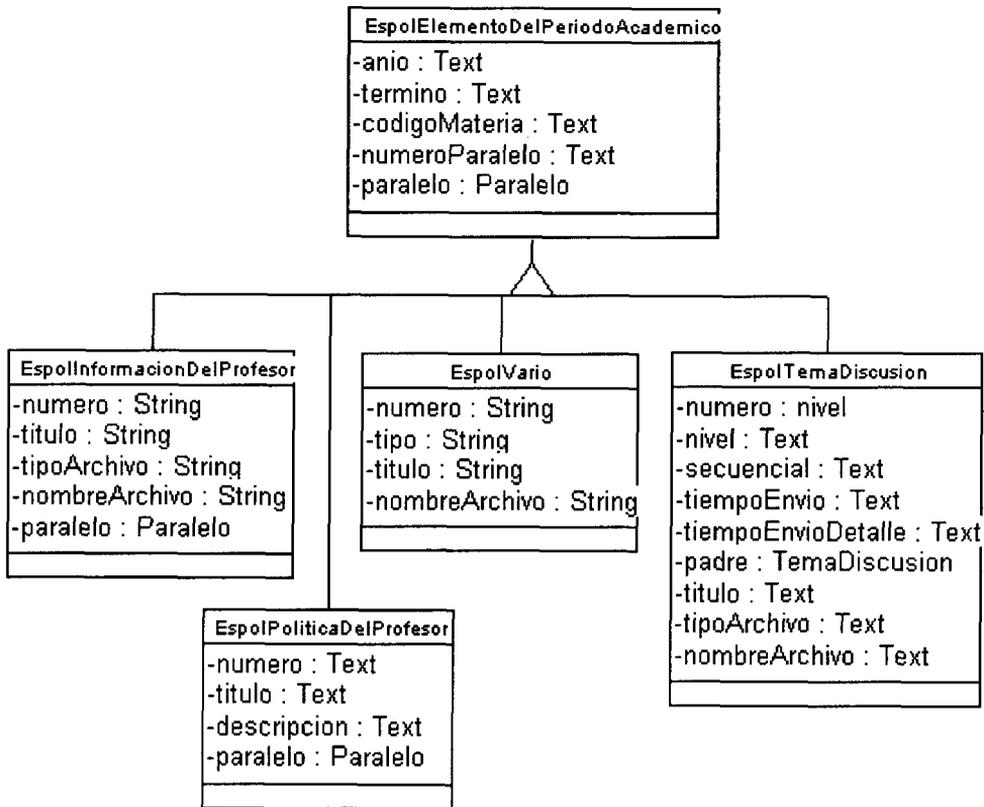
3.5. Diagrama de clases

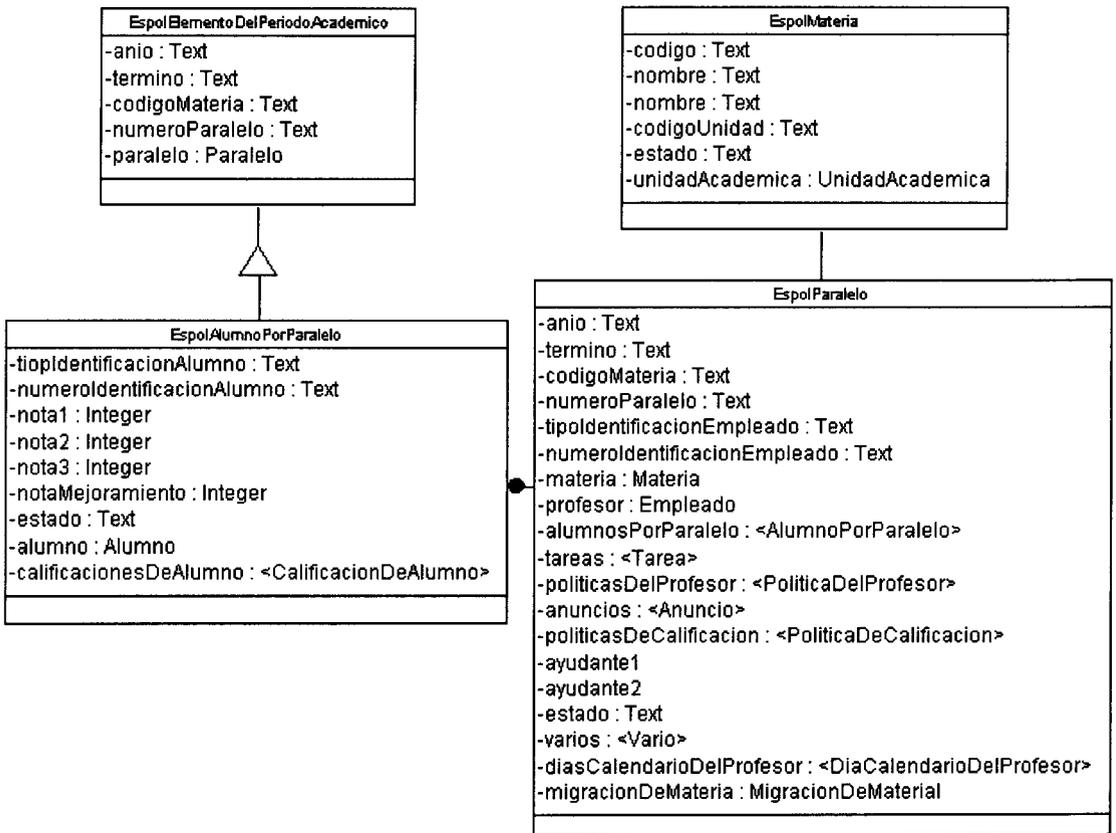
Un diagrama de clases es un esquema o plantilla para describir muchas instancias de datos, describe clases de objetos.

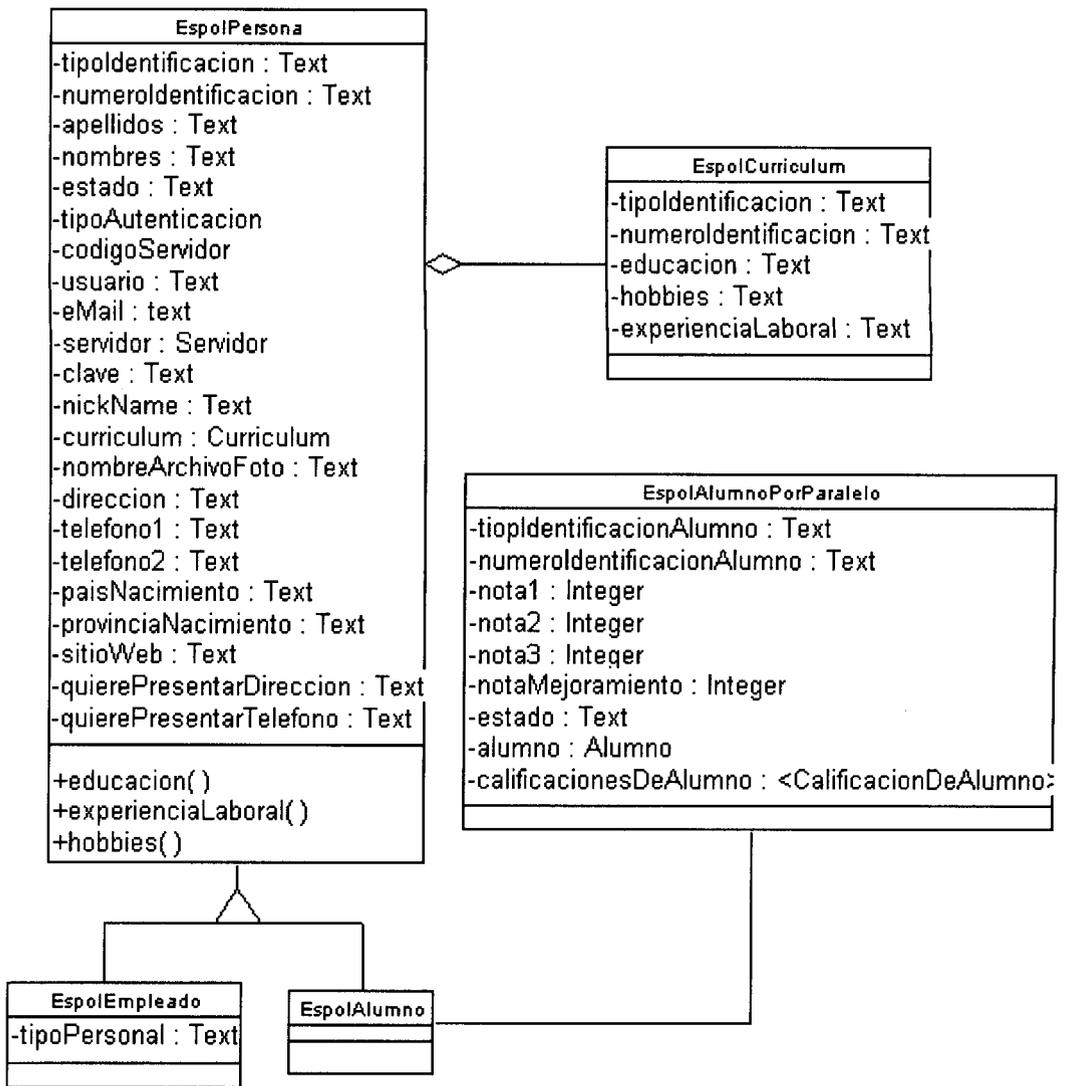
En nuestros diagramas de clases sólo incluiremos atributos, relaciones y los métodos principales que forman parte del comportamiento de cada clase (si se desea el detalle, ver apéndice A) para que estos sean más legibles.

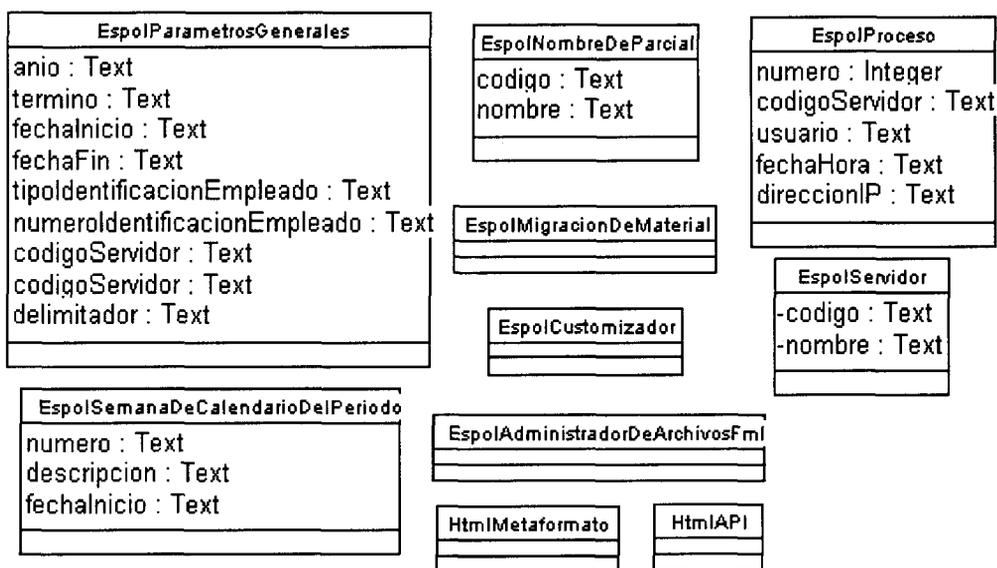


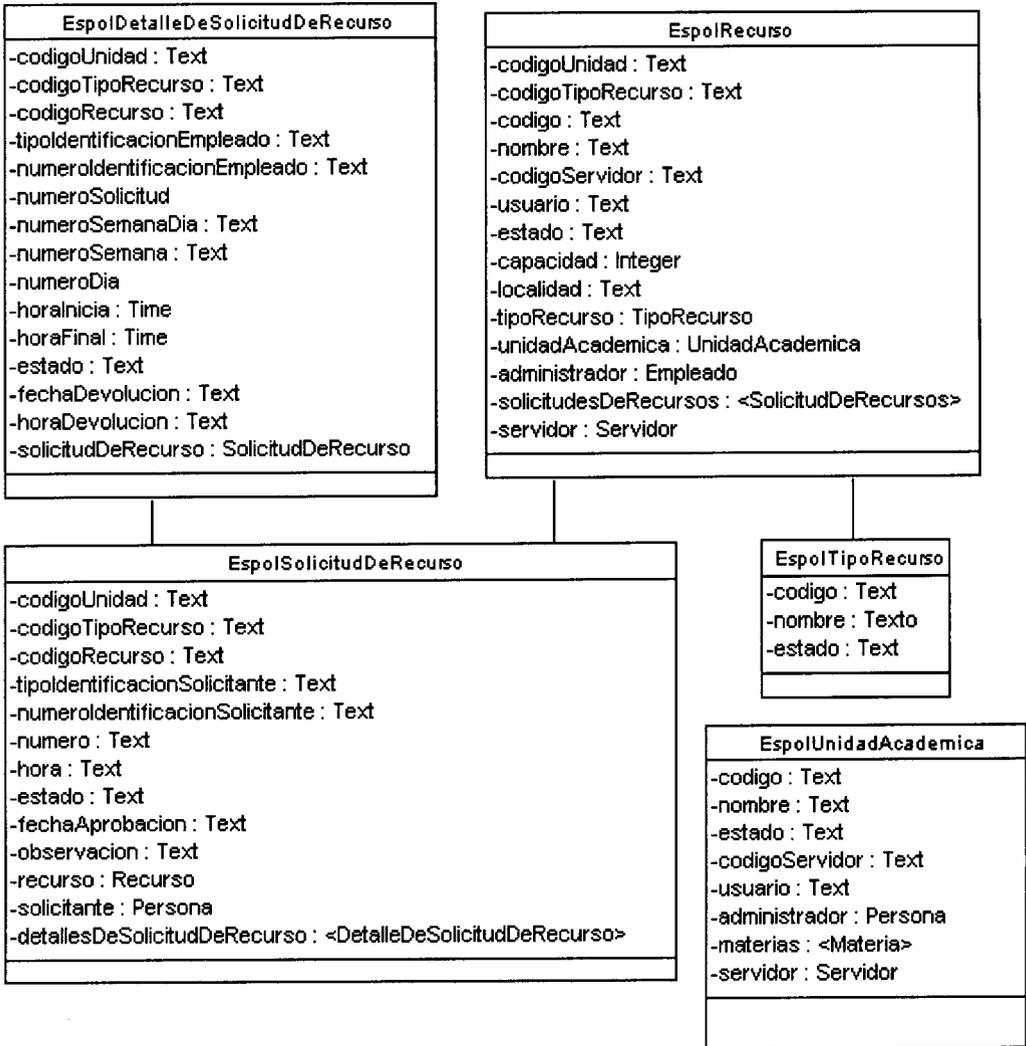












3.6. Análisis de la Interacción Hombre-Máquina

En las últimas décadas, el aumento de la población de usuarios potenciales y reales en los sistemas informáticos y de telecomunicación, así como la creciente complejidad de éstos y de las tareas que realizan, hace que el estudio de la interacción Hombre-Máquina sea muy importante.

El análisis de la interacción Hombre-Máquina esta basado en la usabilidad del sistema, la cual se puede definir como la medida en la cual un producto puede ser usado por usuarios específicos para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso determinado:

- Se entiende como **efectividad** a la precisión y la plenitud con las que los usuarios alcanzan los objetivos determinados.
- La **eficiencia** está dada por la funcionalidad provista por el sistema, ya sea de recursos virtuales o físicos, con la cual el usuario puede alcanzar la efectividad.

- La **satisfacción** se mide por la ausencia de incomodidad y la actitud positiva en el uso del producto. Se trata, pues, de un factor subjetivo.

Existen principios básicos aplicables a la usabilidad en el Web, tales como: accesibilidad, navegabilidad, optimización y productividad. A continuación explicaremos sobre cada uno de ellos y la forma en que son aplicados en Cursos Web:

Accesibilidad

Los usuarios utilizan la Internet a través de diferentes maneras, pudiendo variar servidores, sistemas operativos de las PCs, la resolución de las pantallas de monitores, navegadores e incluso las opciones del navegador (pueden estar o no activadas, por ejemplo: navegar por defecto sin imágenes). Por estos motivos, hay que diseñar aplicaciones que permitan ser accesibles, sin importar las configuraciones de los usuarios.

Cursos Web provee accesibilidad a los usuarios a través del uso de diferentes browsers (Netscape, IE), diferentes resoluciones de la pantalla (640x480, 800x600, 1024x768), diferentes resoluciones de colores (256 colores, 16 bits, 24 bits, 32 bits), y en el caso donde los

usuarios desactiven la opción para cargar imágenes, éstas poseen el texto alternativo que ayuda al usuario a conocer que es lo que representa cada una de las figuras que no serán cargadas.

Navegabilidad:

La navegabilidad de una aplicación o sitio Web esta basada en:

- Facilidad de encontrar la información mediante buscadores o navegación.
- Consistencia de todos los elementos que componen el sitio.
- Facilitar múltiples maneras y vías de encontrar la misma información.

Cursos Web, provee los tres conceptos mencionados anteriormente, mediante:

- Menús principales (en la parte superior e inferior) y submenús (en la parte izquierda) de cada una de las páginas que componen la aplicación.
- Enlaces entre diferentes páginas a través de links, de tal manera que los usuarios estando en una opción pueden consultar

información de otra. Por ejemplo, en los calendarios puede consultar las tareas.

- Todas las páginas de un mismo tipo tienen la misma estructura, de tal manera que el usuario podrá fácilmente identificar cuál es la acción que está realizando.
- El sistema provee diferentes formas en las que el usuario puede obtener una misma información. Por ejemplo: los días del calendario se pueden consultar desde el listado de meses, listado de clases, o del calendario del mes específico.

Optimización:

Esto se refiere al tiempo necesario para descargar una página Web en el browser.

Cursos Web, provee optimización a través de la utilización de imágenes pequeñas (en tamaño de bits), y del uso de los cookies para facilitar información común en cada una de las páginas y de esta forma evitar el constante acceso a la base de datos (por ejemplo: el nombre del usuario).

Productividad:

Consiste en minimizar el trabajo del usuario al realizar las diferentes tareas de la aplicación, evitando de esta manera la generación de errores por parte de quien usa el sistema.

Cursos Web, esta diseñado de tal manera que sus páginas:

- Minimizan el número de tareas necesarias para utilizar una opción.
- Facilitan la utilización de las opciones del sistema a los usuarios, evitando de esta manera la generación de errores.

Esto se consigue gracias a que las páginas del sistema no poseen formatos complejos, mejorando de esta forma la productividad de los usuarios al momento de usar el sistema.

3.7. Flujo de ventanas

Durante la evolución de un modelo de análisis a una construcción de diseño, es necesario considerar no sólo el ambiente de implementación, sino también cómo la información será presentada al usuario.

El flujo de ventanas describe la secuencia de pantallas que el usuario espera ver en el sistema.

Se describe como un gráfico direccionado, donde cada nodo representa ventanas y los arcos representan la dirección de flujo de las ventanas.

3.7.1. Sistema de comunicación Alumno-Profesor.

En Cursos Web, el flujo de ventanas que tendrá disponible un profesor será distinto al que tendrá acceso un estudiante.

Si el usuario es un profesor, el flujo de ventanas es el siguiente:

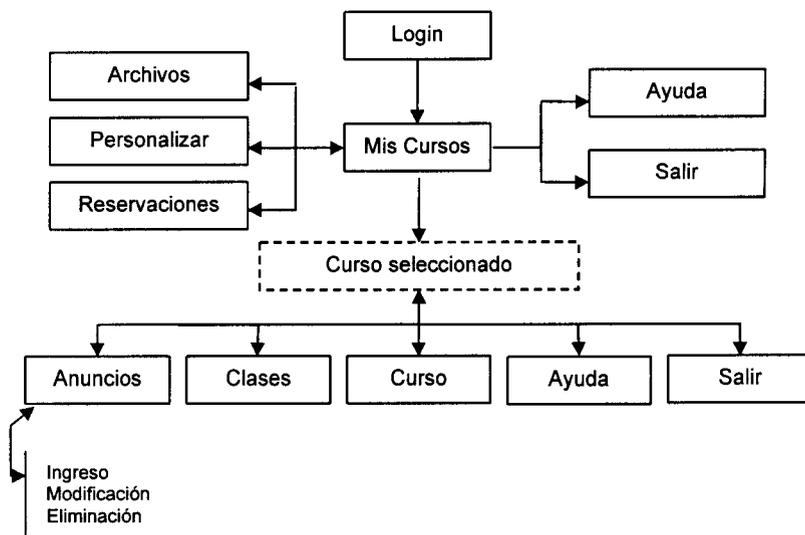


Figura 3.3 Flujo principal de ventanas para profesores

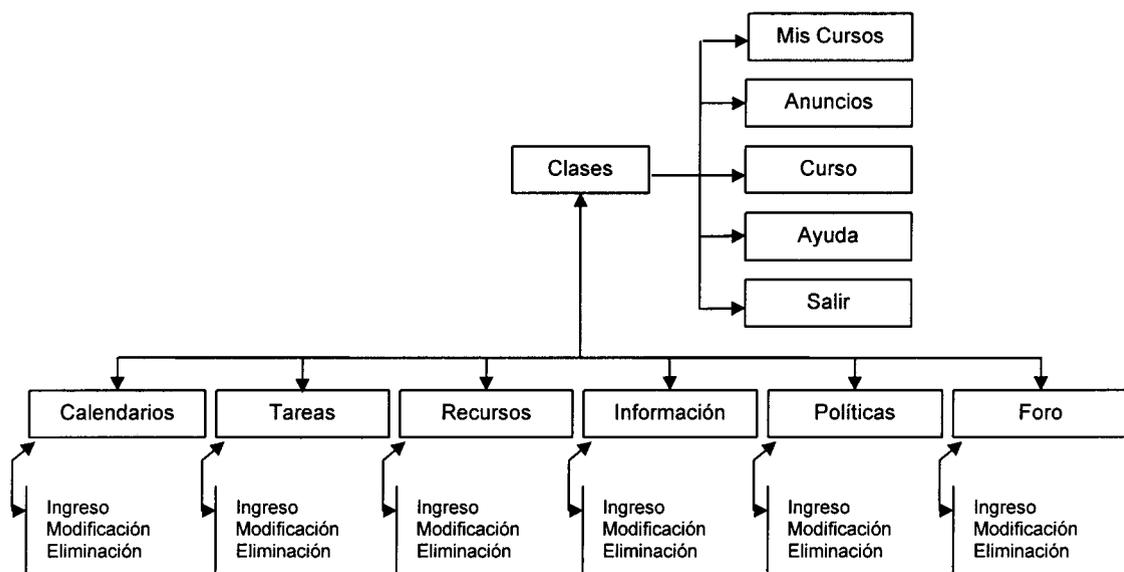


Figura 3.4 Flujo de ventanas de la opción “Clases”

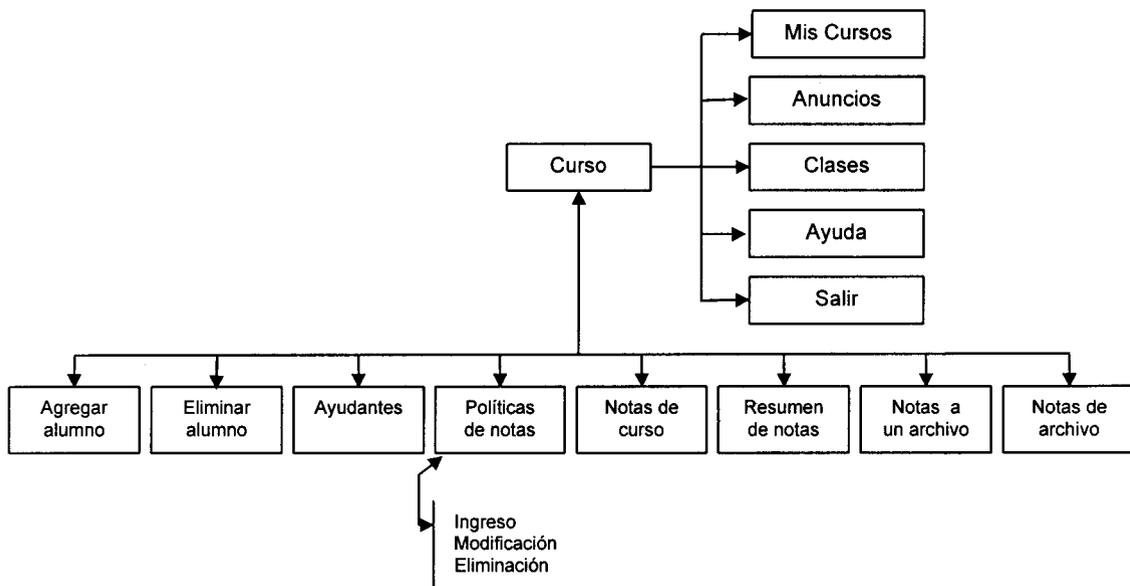


Figura 3.5 Flujo de ventanas de la opción “Curso”

Si el usuario es un alumno, el flujo de ventanas es como sigue:

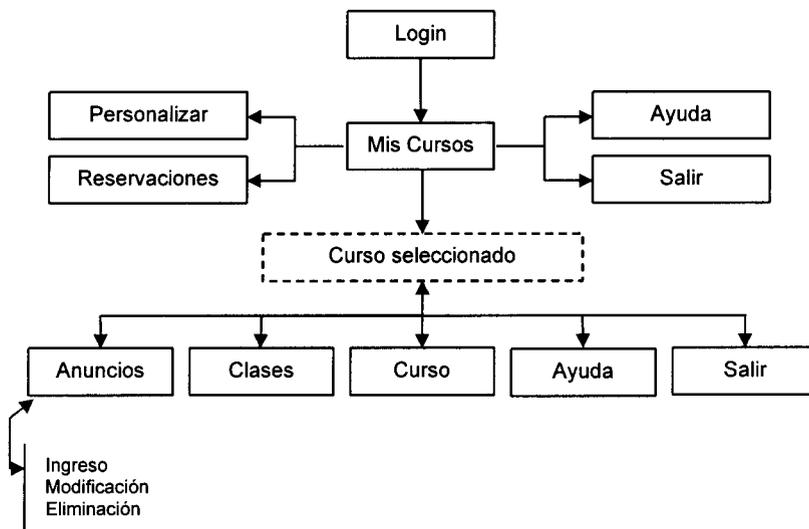


Figura 3.6 Flujo principal de ventanas para estudiantes

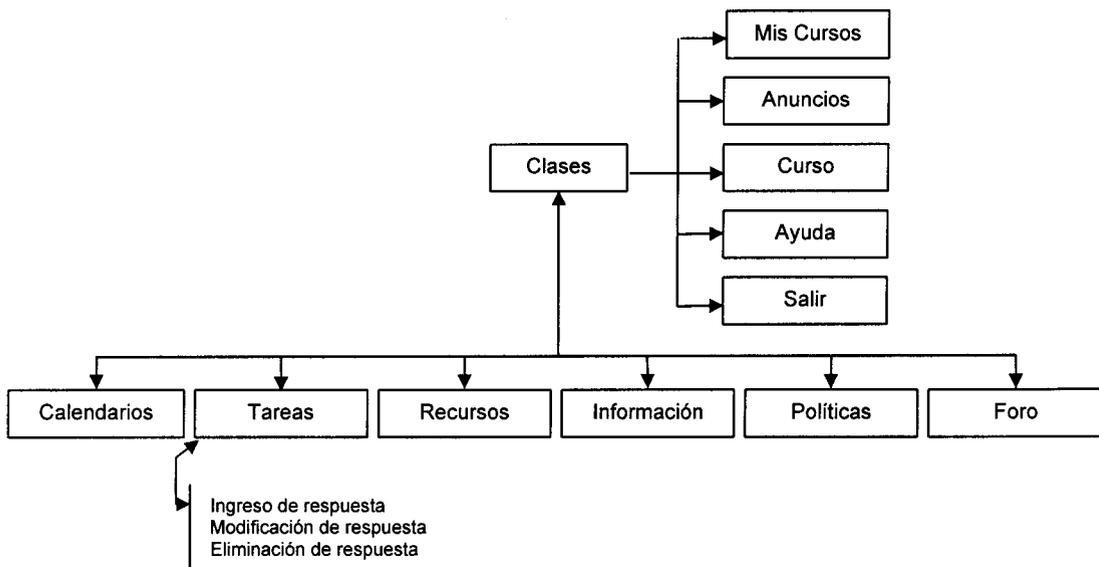


Figura 3.7 Flujo de ventanas de la opción "Clases"

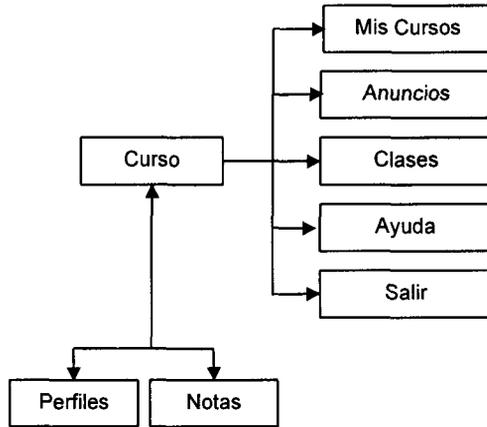


Figura 3.8 Flujo de ventanas de la opción “Curso”

3.7.2. Sistema de administración

El sistema de administración de Cursos Web tendrá el siguiente flujo de ventanas:

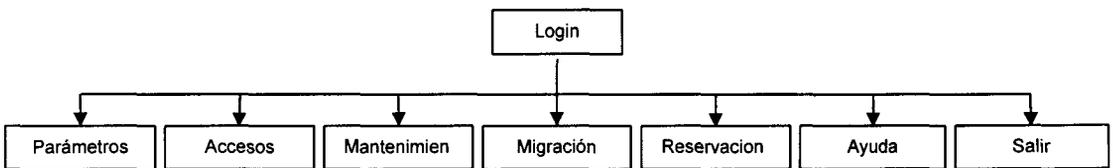


Figura 3.9 Flujo de ventanas principal para administradores

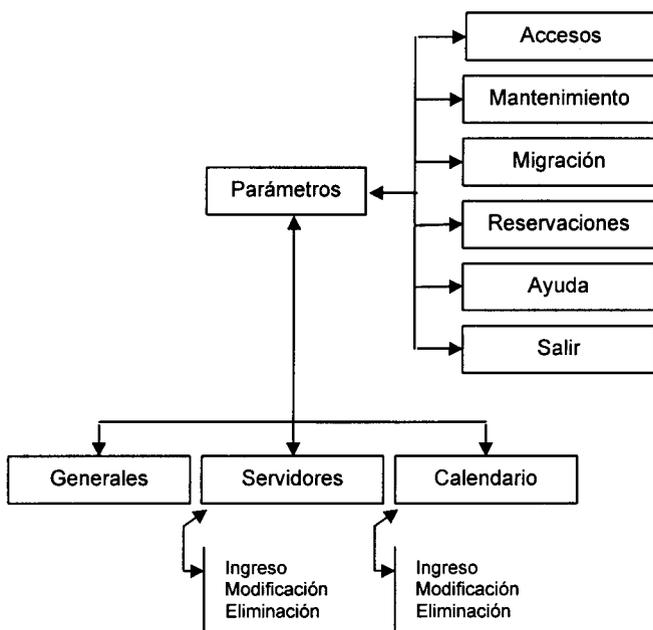


Figura 3.10 Flujo de ventanas de la opción “Parámetros”

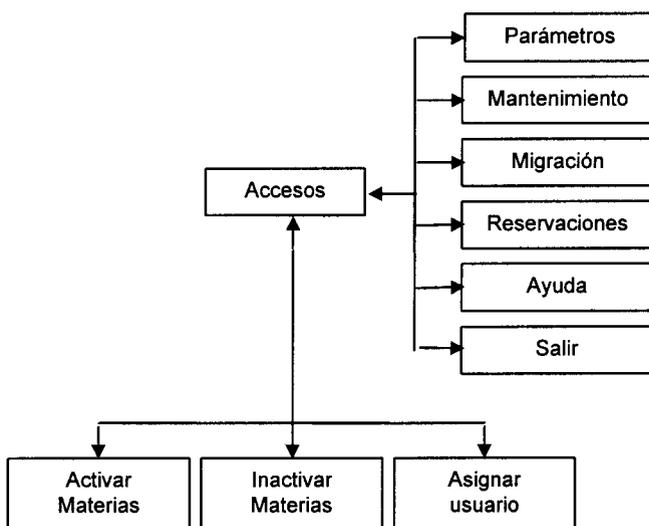


Figura 3.11 Flujo de ventanas de la opción “Accesos”

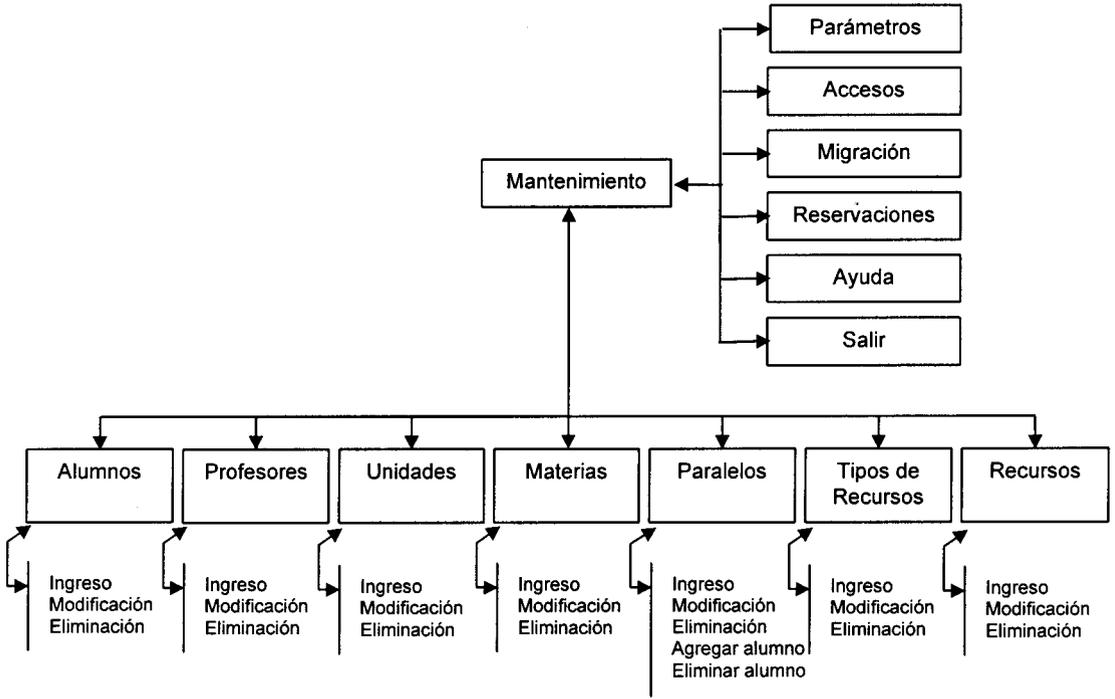


Figura 3.12 Flujo de ventanas de la opción “Mantenimiento”

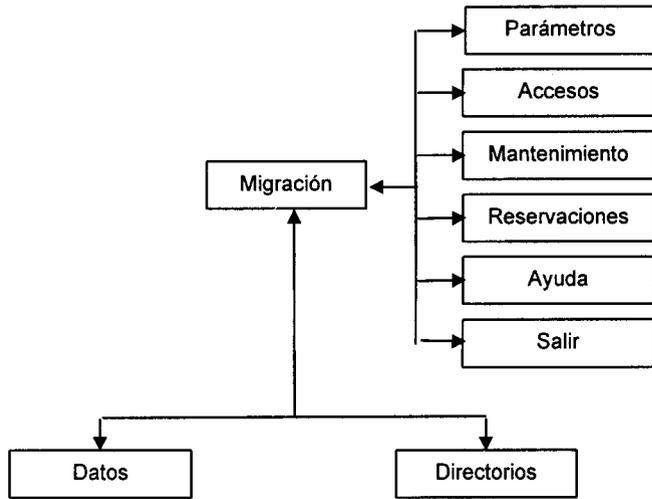


Figura 3.13 Flujo de ventanas de la opción “Migración”

CAPÍTULO 4

4. DISEÑO

En la etapa de análisis de un sistema se determina que es lo que debe hacerse en la implementación, mientras que en la etapa de diseño se determina cuál es el plan de ataque, siendo necesario adicionar objetos internos para la implementación y optimización de procesos.

El objetivo de este capítulo es proveer una base para la implementación de Cursos Web, y para ello seguiremos utilizando la metodología UML, con la cual no hay necesidad de transformar de un modelo a otro, puesto que el paradigma de orientación a objetos se aplica igualmente bien al describir las especificaciones del mundo real y la implementación basada en computadora.

4.1. Diseño del módulo de estandarización de la interfaz

Para el diseño del módulo de estandarización de la interfaz de las páginas HTML de este proyecto, se planteó la utilización del

Metalinguaje propuesto por el Ing. Guido Caicedo y que se explica en detalle en el Apéndice B.

El Metalinguaje o Metaformato es un mecanismo general para dar un formato común a las páginas HTML generadas por los CGIs, el cual esta basado en archivos con parámetros de tags HTML. Estos archivos llamados "Archivos de Formato" no están desarrollados en un lenguaje específico, son simples archivos textos con tags HTML. De tal manera que la tarea de mantenimiento de la apariencia del sistema puede ser realizada por cualquier persona que no conozca sobre la programación de CGIs, ni codificación HTML.

Cada página del sistema se construye partiendo de un archivo de formato base. Este archivo puede hacer referencia a otros archivos de formato, y estos a su vez a otros (dependiendo de los componentes que se deseen definir en la página HTML). Para cambiar la apariencia de un grupo de páginas sólo es necesario modificar uno o varios archivos de formato.

La apariencia de Cursos Web, está basada en un metaformato que está implementado por dos objetos: uno encargado de las tareas generales como enlazar las páginas con su archivo de formato (HtmlAPI), y otro

más especializado que sabe interpretar cada elemento del archivo y traducirlo en un tag HTML (HtmlMetaformato).

La Figura 4.1 muestra la relación entre los diferentes componentes que se encargan de construir las páginas HTML del sistema.

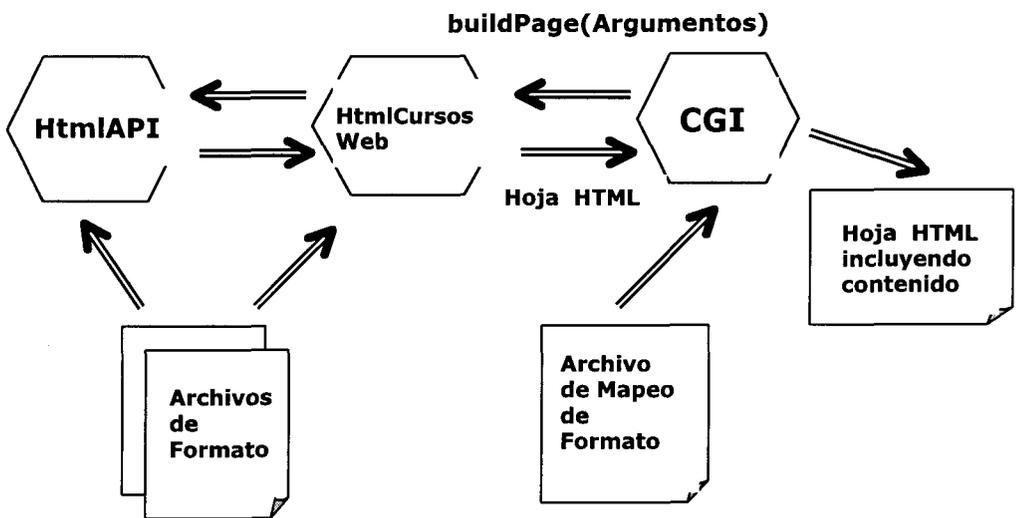


Figura 4.1 Relación entre los diferentes componentes del Metaformato

Los dos componentes del metaformato HtmlAPI y HtmlMetaformato están implementados de tal forma que las funciones de cada uno pueden ser llamadas independientemente. De tal manera que los CGIs también pueden hacer llamadas independientes a las funciones de ambos componentes.

La función `buildPage()`, que se invoca desde el CGI se encarga de construir la página HTML. Esta función recibe los siguientes parámetros:

- Lista de valores a colocar en las variables especificadas en el archivo de formato (sección "Manejo de variables libres" en el Apéndice B).
- Contenido de la página HTML.

4.1.1. Archivos de Formato y su Metalenguaje

Los archivos de formato están definidos en el Apéndice B (sección Archivos de Formato y su Metalenguaje). Existen cambios que fueron aplicados a este esquema, y éstos se detallan a continuación.

Un CGI esta relacionado a un archivo de formato `<PAGINA>` que tiene los parámetros generales de la hoja, este a su vez hace referencia a los archivos de componentes que pueden ser de `<CABECERA>`, `<MENU>`, `<TITULO>`, `<LOGO>` y `<USUARIO>`.

A continuación se define brevemente el metalenguaje a utilizar en cada uno de los diferentes archivos:

PAGINA

La sintaxis del archivo de formato de página es la siguiente:

```
<PAGINA>
  <TITULO= texto en el título de la ventana del browser>
  <FONDO= color  FIG= figura para el fondo>
  <FONT= tipo de caracteres  TAM= tamaño  COLOR= color>
  <CLINK= color  CALINK= color  CVLINK= color>
  <LETRATIT= tipo de letra de títulos  TIPO= negrita, cursiva, normal  TAM= tamaño
  COLOR= color>
  <HEADER= color de los encabezados de las tablas  CBODY= color del contenido
  de las tablas  CCONSULTA= color de las tablas de consulta>
  <PATHGIF= URL donde se encuentran las figuras>
  <HCABECERA= Archivo de formato de cabecera  ALTO= alto de la celda en
  pixeles>
  <VCABECERA= Archivo de formato de cabecera  ANCHO= ancho de la celda en
  pixeles>
  <CABECERA= Archivo de formato de cabecera  ALTO= alto de la celda en pixeles>
  <PIE= Archivo de formato de cabecera  ALTO= alto de la celda en pixeles>
</PAGINA>
```

Este archivo contiene nuevos tags, los mismos que se explican a continuación:

- <LETRATIT> sirve para definir el tipo de letra, estilo (TIPO) y tamaño (TAM) de los títulos en las páginas HTML.
- <HEADER> sirve para definir el color de los encabezados de las tablas de listados en las páginas, además en él se define el color del contenido de estas tablas (CBODY), y también el color del contenido de las tablas que se utilizan para mostrar las consultas (CCONSULTA).

CABECERA

El archivo de formato de cabecera tiene la siguiente sintaxis:

```
<CABECERA>
  <ESPACIOSUP= numero de espacios antes del título o de los datos de los usuarios>
  <LOGOSUP= Archivo de formato del logo superior>
  <ESPACIOINF= numero de espacios después del título o de los datos de los
    usuarios>
  <PROFESOR= Archivo de formato de los datos del profesor>
  <ALUMNO= Archivo de formato de los datos del alumno>
  <TITULO= Archivo de formato de título>
  <MENU= Archivo de formato de menú>
  <LOGOINF= Archivo de formato del logo inferior>
</CABECERA>
```

Las diferencias entre el archivo de cabecera original y el planteado en este metaformato son:

- <ESPACIOSUP> nombre que reemplaza al tag <ESPACIOS>, tiene la misma funcionalidad pero se utiliza solo para definir el número de líneas en blanco en la parte superior de la cabecera.
- <LOGOSUP> nuevo tag que sirve para definir los logos del sistema, van al mismo nivel que el título, pero con la diferencia de que no están definidos en el archivo de formato de título, debido a que tienen una nueva posición (van a la izquierda del título). En este tag se define un nuevo archivo de formato de logo (<LOGO>). En el diseño de las páginas de nuestro sistema este tag es utilizado

únicamente en la cabecera horizontal, pero si puede ser utilizado en la cabecera vertical.

- `<ESPACIOINF>` nuevo tag que define el número de líneas en blanco después del título.
- `<LOGOINF>` nuevo tag que sirve para definir los logos de la institución que utiliza el sistema. Tiene el mismo comportamiento que `<LOGOSUP>` ya que en él se define un archivo de formato de logo (`<LOGO>`). En el diseño de las páginas de nuestro sistema este tag es utilizado únicamente en la cabecera vertical, y no puede ser utilizado en la cabecera horizontal.
- `<PROFESOR>` nuevo tag que define los datos del profesor que dicta la materia del curso accedido por el usuario, en este tag se especifica un nuevo archivo de formato de datos (`<USUARIO>`) y es utilizado únicamente por la cabecera vertical. Esta información se muestra en la parte superior del título.
- `<ALUMNO>` nuevo tag que define los datos del alumno (usuario con de perfil de estudiante) que accede al sistema, en este tag se especifica el archivo de formato de datos (`<USUARIO>`) y es utilizado únicamente por la cabecera vertical. Esta información se muestra en la parte superior del título e inferior de los datos del profesor.

TITULO

La sintaxis del archivo de título es la siguiente:

```
<TITULO>
  <FONDO= color  FIG= figura>
  <CONT= texto del contenido  FIG= nombre de la figura  LINK= URL  ANCHO=
  pixeles>
  <FONT= tipo de caracteres  TAM= tamaño  COLOR= color>
  <BORDE= pixeles  ESPCEL= pixeles  BORCEL= pixeles  ANCHO= pixeles>
  <ALINEA= la tabla va en centro, izquierda o derecha>
</TITULO>
```

Las diferencias existentes entre este archivo y el archivo original de título son:

- No incluye los tags <DICON> e <IICON>, que sirven para definir los iconos de la izquierda y la derecha del título. Estos han sido reemplazados por el tag <LOGOSUP> definido en el archivo de cabecera, y la nueva posición de los iconos es a la izquierda del título.
- El tag <BORDE> incluye una nueva propiedad (ANCHO) que sirve para definir el ancho de la tabla que contiene el título.

LOGO

Es un nuevo archivo de formato, el cual nos permite ubicar dos logos, ya sea en la parte superior o inferior de la página. Si es utilizado con el tag <LOGOSUP> van al mismo nivel que el título, pero si es utilizado con el tag <LOGOINF> van en la parte inferior del menú de la cabecera vertical. Su sintaxis es la siguiente:

```
<LOGO>
  <FONDO= color  FIG= figura>
  <SICON= nombre de la figura  LINK= URL  ANCHO= pixeles  TEXTO= texto
    sobre la figura>
  <IICON= nombre de la figura  LINK= URL ANCHO= pixeles  TEXTO= texto sobre
    la figura>
  <BORDE= pixeles  ESPCEL= pixeles  BORCEL= pixeles  ANCHO= pixeles>
  <ALINEA= la tabla va en centro, izquierda o derecha>
</LOGO>
```

Sus componentes se detallan a continuación:

- <FONDO> define el color del fondo de la tabla que contiene los logos, el atributo (FIG) define una figura para el fondo.
- <SICON> tag que define la figura superior y sus propiedades, tales como LINK, ANCHO y TEXTO, donde texto es el que se muestra cuando se ubica el mouse sobre la figura.

- <IICON> tag que define la figura inferior y sus propiedades, tales como LINK, ANCHO y TEXTO, donde texto es el que se muestra cuando se ubica el mouse sobre la figura.
- <BORDE> tag que esta definido en la mayoría de archivos de formato y aquí tiene el mismo comportamiento. Define las propiedades de la tabla que contiene los logos.
- <ALINEA> define la posición de la tabla que contiene los logos.

USUARIO

Es un nuevo archivo de formato, que permite presentar información del usuario conectado al sistema; y, en el caso de ser un alumno, incluye información del profesor de la materia seleccionada. Este tag es utilizado únicamente en la cabecera vertical y es llamado por los tags <PROFESOR> y/o <ALUMNO>. Su sintaxis es la siguiente:

```
<USUARIO>
  <FONDO= color  FIG= figura>
  <ETIQUETA= título  APELLIDOS= apellidos del usuario  NOMBRES= nombres
    del usuario  EMAIL= email del usuario>
  <FONT= tipo de caracteres  TAM= tamaño  COLORCONT= color contenido
    COLORLINK= color link>
  <BORDE= pixeles  ESPCEL= pixeles  BORCEL= pixeles  ANCHO= pixeles>
  <ALINEA= la tabla va en centro, izquierda o derecha>
</USUARIO>
```

Sus componentes se detallan a continuación:

- <FONDO> define el color del fondo de la tabla que contiene la información, el atributo (FIG) define una figura para el fondo.
- <ETIQUETA> define el título de los datos, por ejemplo: Profesor, Alumno o Usuario. Contiene las propiedades APELLIDOS, NOMBRES y EMAIL.
- define el tipo de letra, contiene otras propiedades como tamaño (TAM), color del texto (COLORCONT) y color del link (COLORLINK) utilizado para el atributo email.
- <BORDE> es un tag que esta definido en la mayoría de archivos de formato y aquí tiene el mismo comportamiento. Define las propiedades de la tabla que contiene los datos.
- <ALINEA> define la posición de la tabla que contiene los datos.

MENU

La sintaxis del archivo de formato para el menú es la siguiente:

<MENU>

- <ESTILO= tipo de menú (horizontal o vertical)>
- <ANCHO= ancho de la tabla en pixeles>
- <ALINEA= la tabla va en el centro, izquierda o derecha>
- <ALINBOT= cada botón va centrado, a la izquierda o a la derecha>
- <BORDE= pixeles ESPCEL= pixeles BORCEL= pixeles>
- <FONDO= color FIG= figura>
- <FONT= tipo de caracteres TAM= tamaño COLOR= color>
- <BOTNOM= nombre de archivo de botones EXT= extensión del archivo>
- <EXTESC= carácter adicional para nombre de botón escogido>
- <EXTDES= carácter adicional para nombre de botones deshabilitados>

```

<LISTABOT= números separados por coma, botones del menú>
<LISTADES= números separados por coma, botones deshabilitados>
<ESCOGIDO>
    <NUMERO= número del botón escogido>
    <FONT= tipo de caracteres TAM= tamaño COLOR= color>
    <SUBMENU= Archivo de formato del submenú
        POSICION= número de posición en el menú>
</ESCOGIDO>
<BOT= archivo de botones>
</MENU>

```

Las diferencias entre el archivo de menú original y el planteado en este metaformato son:

- <ESTILO> nuevo tag que especifica el tipo de menú (horizontal o vertical).
- <EXTDES> nuevo tag que especifica el carácter adicional para identificar los iconos del menú que se mostrarán en modo deshabilitado. Este se utiliza en el caso que las opciones del menú se representen con figuras.
- <LISTADES> nuevo tag para identificar la lista de opciones del menú que se mostrarán en modo deshabilitado.
- <SUBMENU> nuevo tag que especifica un archivo de formato tipo menú para ser utilizado como submenú de una opción del menú. En el atributo POSICION define el número de la opción del menú que tendrá submenú. Este tag es utilizado en la cabecera vertical.

4.2. Diseño de las plantillas del Sitio Web

En esta sección explicaremos el diseño de las páginas Web, las mismas que servirán para la construcción de las páginas con las que va a interactuar el usuario. El diseño de las páginas está basado en el metaformato (explicado en la sección anterior) y contempla los criterios de la Interacción Hombre-Máquina (ver sección 3.6).

En base a los diferentes modelos de páginas que podemos aplicar mediante el metaformato (ver Apéndice B), realizamos el diseño de las mismas y de acuerdo a la funcionalidad del sistema se ha escogido el modelo Combinado, el cual contiene cabecera horizontal (hcabecera), cabecera vertical (vcabecera) y contenido (conformado a su vez por título, cuerpo y pie de página). La única excepción de este modelo es la página de inicio o acceso al sistema, ya que utiliza el modelo Horizontal (no posee cabecera vertical).

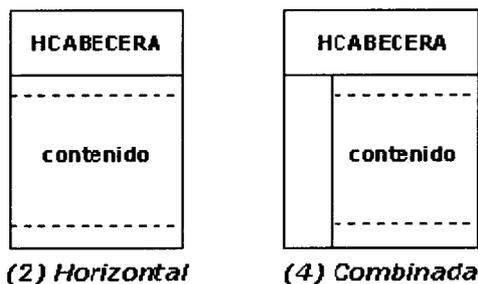


Figura 4.2 Modelo de páginas del sistema

Los modelos utilizados en el sistema se muestran en la Figura 4.2

Empezaremos con la página de acceso al sistema (Figura 4.3) ya que es la única que no sigue el mismo formato de las demás.

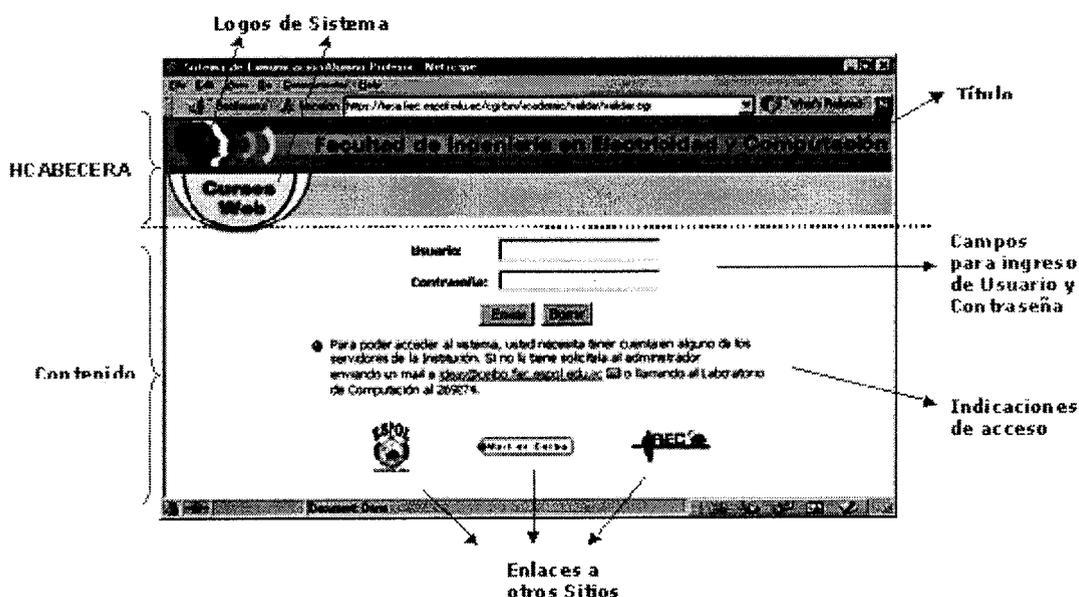


Figura 4.3 Diseño de la página de acceso al Sistema

En el centro de la página se encuentran los campos para realizar el ingreso del usuario y contraseña, además de la información general para poder acceder al sistema, en la parte superior se encuentran los logos y el nombre de la unidad que lo está utilizando. En la parte inferior se encuentran los enlaces a otros sitios de interés para los usuarios.

Como podemos apreciar en la página la visibilidad de sus componentes es adecuada para la interacción con el usuario.

Posterior a la verificación del usuario y contraseña, las páginas del sistema poseen la misma estructura (modelo combinado), lo que podemos observar en la Figura 4.4.

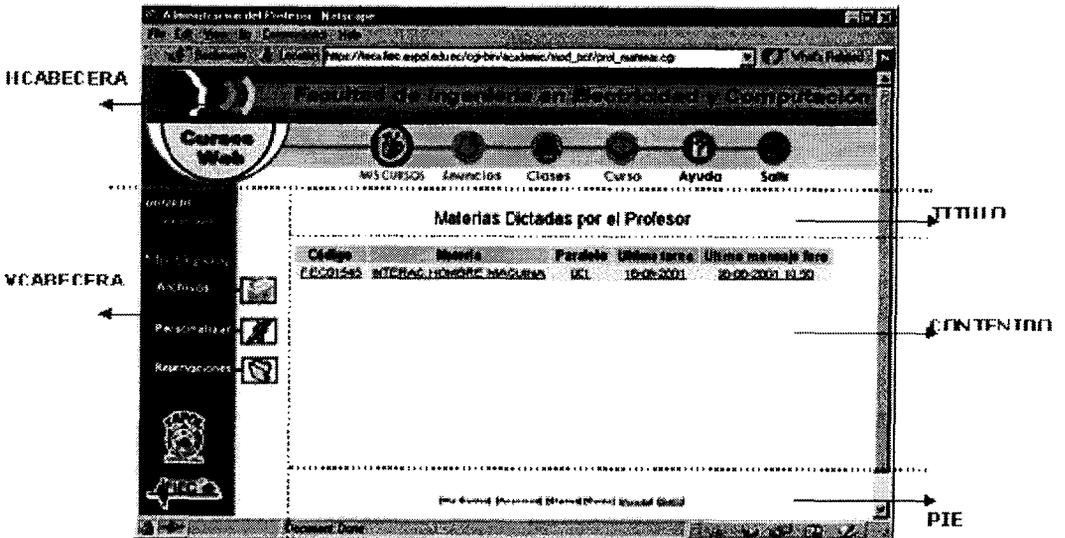


Figura 4.4 Diseño de página basada en el modelo combinado

En esta figura el modelo combinado muestra sus componentes: **hcabecera**, **vcabecera**, **título**, **contenido** y **pie**. Cabe mencionar que estos componentes tendrán contenidos que van a variar de acuerdo al estatus en el que se encuentre el sistema al utilizar las diferentes funcionalidades que provee. Esto se logra gracias a los diferentes

atributos que se puede aplicar en el metaformato para cada uno de los componentes.

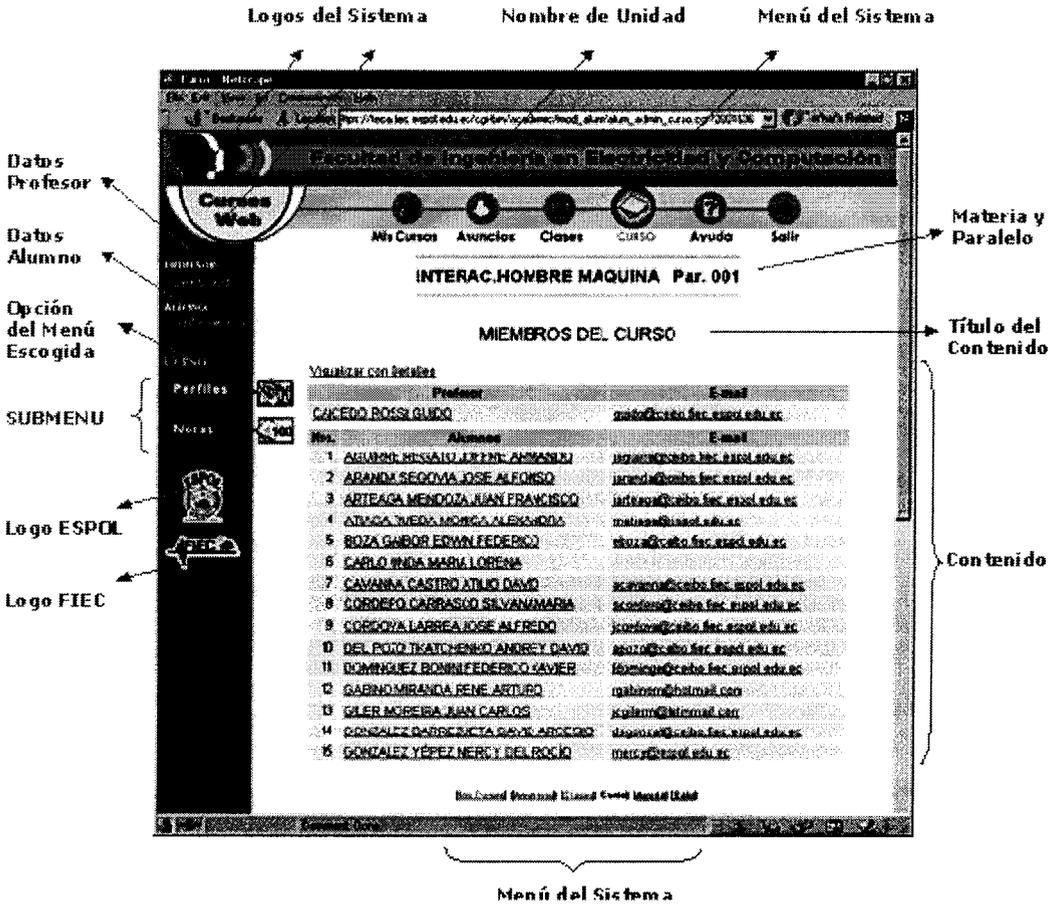


Figura 4.5 Componentes de las páginas

En la Figura 4.5. se muestran los componentes de las páginas HTML del sistema, tales como: logos, menú, submenú, contenido, etc. Como podemos observar el diseño de las páginas considera los principios de la Interacción Hombre-Máquina, haciéndolo más usable. Las páginas proveen menús que le permiten al usuario explorar fácilmente todas las

opciones y navegar en ellas sin la menor dificultad. La distribución de sus partes se diseñó en función a la visibilidad que deben poseer para fijar la atención del usuario.

Tipos de Cabecera Horizontal

Existen tres diferentes tipos de cabecera horizontal, los mismos que son mostrados en la Figura 4.6. Lo que tienen en común son los logos del sistema y el nombre de la unidad que lo utiliza.

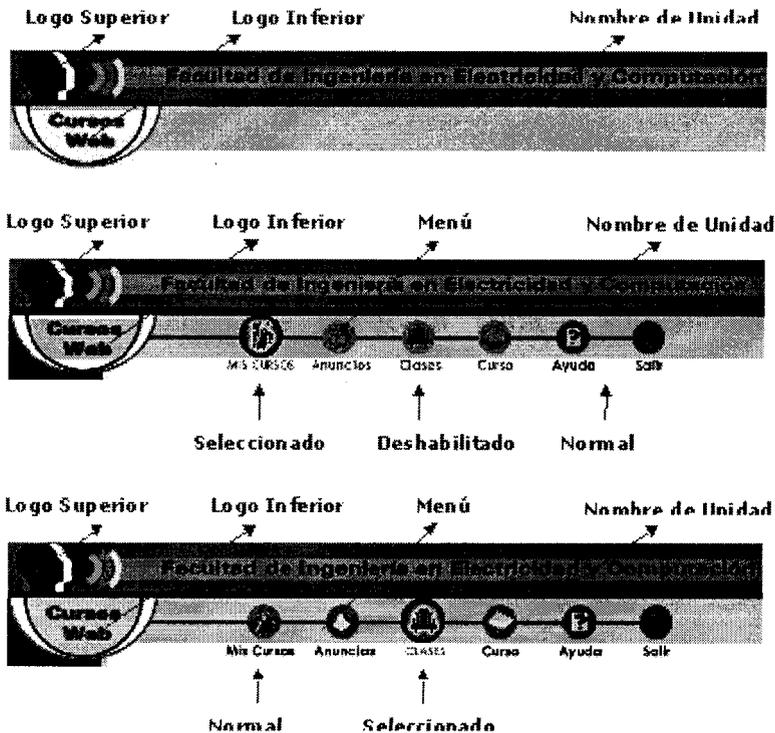


Figura 4.6 Tipos de cabecera horizontal

Las diferencias que se dan en los tres formatos son:

- El primero no posee menú y solo se aplica en la página de acceso al sistema.
- El segundo posee el menú pero contempla la opción de botones habilitados y deshabilitados. Este formato se aplica sólo en la página principal de selección de materias (Mis Cursos) tanto para el profesor como para el alumno.
- El tercero posee el menú que contempla la opción escogida (que es deshabilitada) y las opciones habilitadas. Este formato se aplica en la mayoría de las páginas.

Tipos de Cabecera Vertical

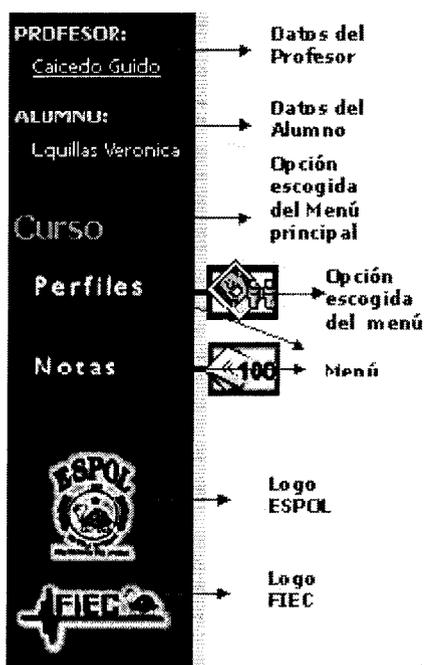
Los diferentes tipos de cabecera vertical son mostrados en las siguientes figuras:



(a)



(b)

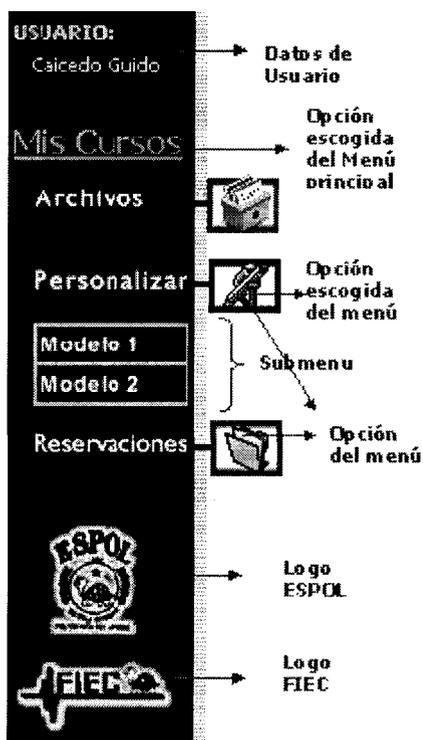


(c)

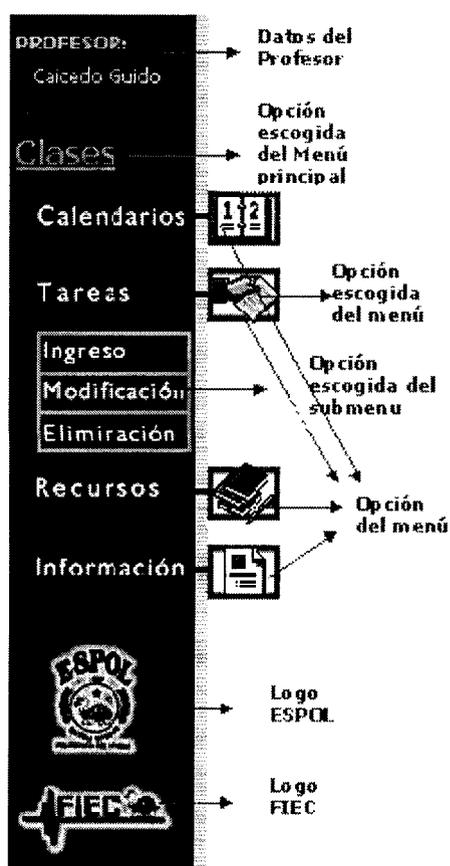


(d)

Figura 4.7 Tipos de cabecera Vertical



(e)



(f)

4.7 Tipos de cabecera Vertical

Como podemos observar en la Figura 4.7, existen diferentes tipos de cabecera vertical, pero algo que tienen todas en común son los logos de la ESPOL y de la FIEC (logo de la unidad que utiliza el sistema). Además todas muestran en la parte superior la información del usuario (profesor y/o alumno).

Los detalles de cada figura y sus diferencias son:

- El tipo **(a)**, mostrado en la Figura 4.7, muestra los datos del usuario que accede al sistema con la etiqueta "**USUARIO**". Se definió esta etiqueta para la página "Mis Cursos", puesto que en ella, la persona conectada aún no se encuentra en ningún curso.
- El tipo **(b)** muestra los datos del usuario que accede al sistema con la etiqueta **ALUMNO** y los datos del profesor que dicta el curso seleccionado con la etiqueta **PROFESOR**. Este modelo es utilizado en la página de los alumnos cuando se encuentran en la opción **ANUNCIOS** del menú principal (en la cabecera horizontal), y es el único que no incluye un menú.
- En la figura del tipo **(c)**, se puede apreciar la opción escogida del menú principal (en este caso Curso), que a su vez posee un menú, el cual aparece con una de sus opciones seleccionadas (en este caso Perfiles).
- El tipo **(d)** es similar al tipo **(c)**, con la diferencia de que la opción escogida del menú principal (Clases) posee un link que le permitirá al usuario regresar a la página de presentación de esta opción.
- El tipo **(e)** a diferencia del tipo **(c)**, posee un submenú en la opción escogida del menú (en este caso Personalizar).
- El tipo **(f)** a diferencia del tipo **(e)**, muestra a una de las opciones del submenú en modo escogido (en este caso Modificación).

Basándonos en los diferentes modelos de cabecera vertical, podemos generalizar lo siguiente:

- Cuando un alumno accede al sistema y se encuentra en un curso específico, la cabecera vertical siempre mostrará los datos del profesor con un link a su dirección email.
- Cuando la opción escogida del menú principal posee un menú, esta se mostrará como título del menú vertical.
- Cuando un usuario accede a una de las opciones del menú vertical se desplegará un submenú en el caso de que lo posea (tipo e), y esta opción se mostrará en modo escogido.
- Cuando un usuario accede a una de las opciones del submenú, ésta se mostrará en modo escogido y la opción que la contiene deja de ser escogida.

Tipos de Contenido en las páginas

El contenido de las páginas en su mayoría contendrá los componentes mostrados en la Figura 4.5, que son:

- Nombre de la materia y paralelo.
- Título del contenido.

- Contenido.
- Menú principal en el pie.

Las excepciones serán las páginas de acceso al sistema, de error, de alerta, de verificación del usuario y la página principal de las materias.

Las páginas de error, alerta y de verificación del usuario no poseen el nombre de la materia, el paralelo y el título.

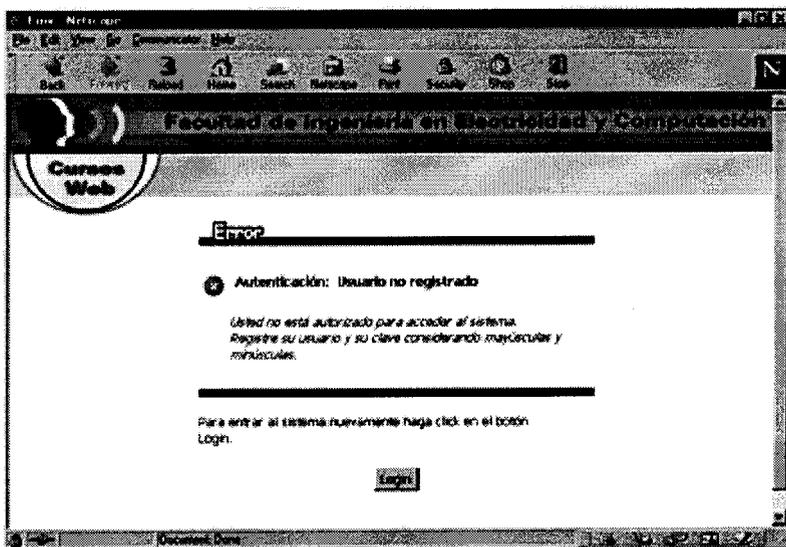


Figura 4.8 Ventana de error

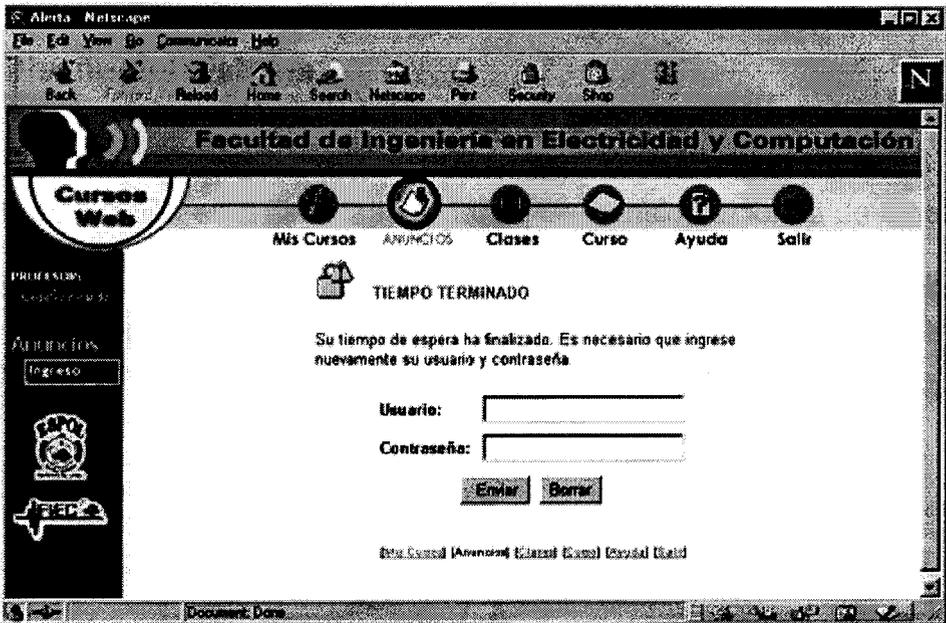


Figura 4.9 Ventana de confirmación de usuario y contraseña

4.3. Diseño del sistema

Esta sección está destinada a definir las consideraciones de diseño del sistema, y mencionar cada uno de los componentes de la arquitectura de Cursos Web, presentados en la Figura 4.10.

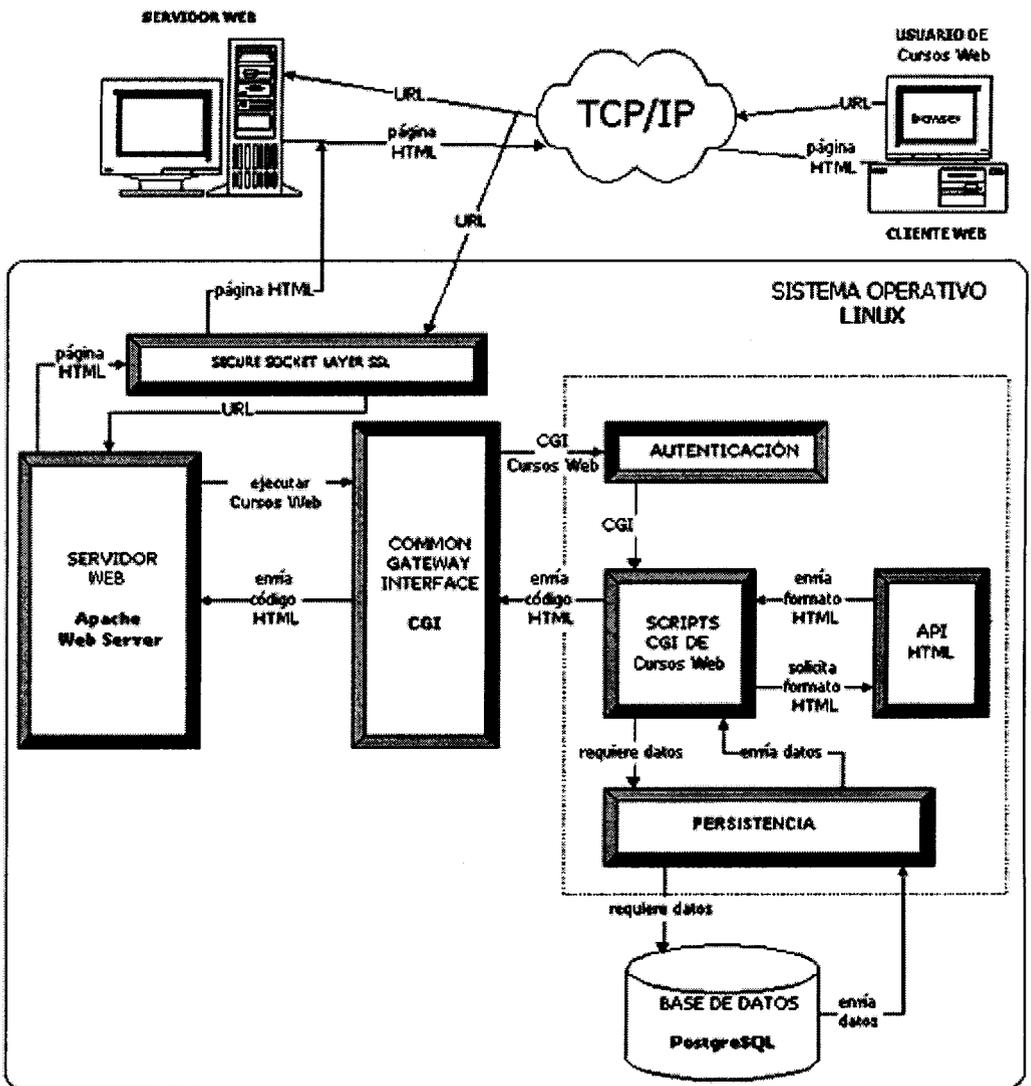


Figura 4.10 Arquitectura de Cursos Web

Como podemos apreciar en la figura anterior, Cursos Web es una aplicación basada en la arquitectura Cliente-Servidor Web, para la cual se utiliza el protocolo TCP/IP.

La máquina servidora cuenta con el sistema operativo Linux, y en éste se encuentra instalado el servidor Web Apache, el cual utiliza el protocolo SSL para proveer una capa de seguridad de los datos entre la aplicación y la comunicación TCP/IP.

Para el desarrollo de los scripts que generan las páginas HTML, se utiliza el estándar CGI (Common Gateway Interface), el cual permite al Servidor Web ejecutarlos y procesar la información que el usuario solicita.

El sistema será desarrollado en base a la metodología de orientación a objetos, utilizando el lenguaje de programación Python. Esto presenta las ventajas de que la aplicación sea reutilizable y fácilmente mantenible.

Para proporcionar el mecanismo que permite almacenar los datos que requieran los usuarios, utilizaremos el manejador de base de datos Postgres.

Adicionalmente al repositorio de datos, se han definido estructuras de directorios en las que se mantendrán los archivos publicados por los usuarios a través de la aplicación Web y los archivos de formato customizable por cada uno de ellos. Se decidió usar estructuras de directorios en lugar de usar la base de datos porque si se utilizara este último, los usuarios sólo podrían publicar información tipo texto (no serían permitidos documentos de Word, Excel, etc.) y de tamaño limitado. Tal estructura se encuentra graficada en la Figura 4.11 y la descripción de cada directorio se explica en la Tabla 4.1.

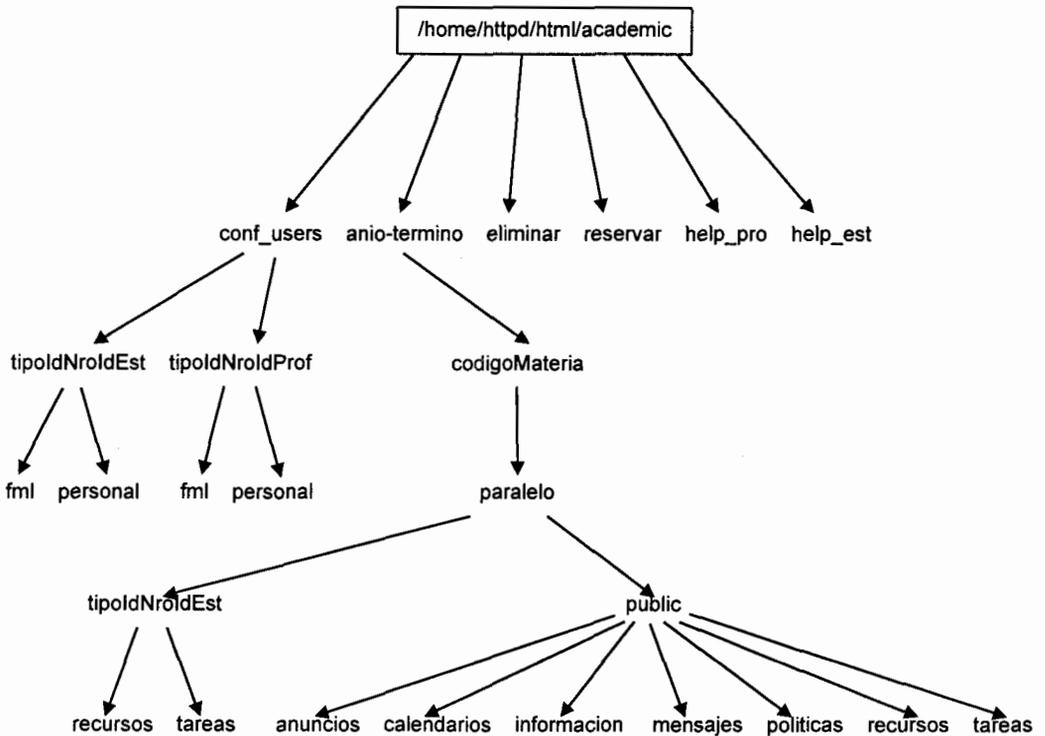


Figura 4.11 Estructura de directorios de trabajo de usuarios

Directorio	Descripción
/conf_users	Directorio raíz para mantener los archivos de formato por cada usuario.
/conf_users/tipoldNroidEst	Existe uno de estos directorios por cada estudiante. El nombre del mismo se construye con la concatenación del tipo y el número de identificación.
/conf_users/tipoldNroidProf	Existe uno de estos directorios por cada profesor. El nombre del mismo se construye con la concatenación del tipo y el número de identificación.
/conf_users/tipoldNroidEst/fml /conf_users/tipoldNroidProf/fml	Contienen los archivos de configuración de cada usuario.
/conf_users/tipoldNroidEst/personal /conf_users/tipoldNroidProf/personal	Contienen archivos personales de cada usuario. Hasta ahora sólo se almacenan las fotos dentro de estos directorios.
/anio-termino	Directorio raíz para la administración de los cursos. Creado por cada año y término en el que es utilizado el sistema.
/anio-termino/codigoMateria	Existe uno de estos directorios por cada materia dictada dentro del año-termino al que pertenecen.
/anio-termino/codigoMateria/paralelo	Creado por cada uno de los paralelos abiertos para cada materia en un año-termino.
/anio-termino/codigoMateria/paralelo/ tipoldNroidEst	Existe uno de estos directorios por cada estudiante en un paralelo determinado. Sirve para mantener los archivos relacionados al curso que cada estudiante publique, tales como respuestas de tarea y recursos.
/anio-termino/codigoMateria/paralelo/ public	Directorio para organizar los archivos que el profesor pone a disposición de sus estudiantes en un paralelo.
/anio-termino/codigoMateria/paralelo/ public /anuncios /calendarios /informacion /mensajes /politicas /recursos /tareas	En cada uno de estos directorios se clasifican los archivos publicados por el profesor del curso. El tipo de información está relacionada al nombre del cada directorio al que pertenecen.
/eliminar	Contiene archivos temporales, utilizados

	cuando un usuario desea eliminar alguno de los objetos publicados.
/reservar	Contiene archivos temporales, utilizados cuando un usuario desea reservar algún recurso.
/help_pro	Contiene los archivos de ayuda del sistema cuando el usuario es un profesor.
/help_est	Contiene los archivos de ayuda del sistema cuando el usuario es un estudiante.

Tabla III Directorios de trabajo de usuarios

Así como se definió una estructura de directorios de trabajo para los usuarios, se creó una estructura de directorios para organizar los programas de la aplicación. Tal estructura es mostrada en la Figura 4.12 y explicada en la Tabla 4.2.

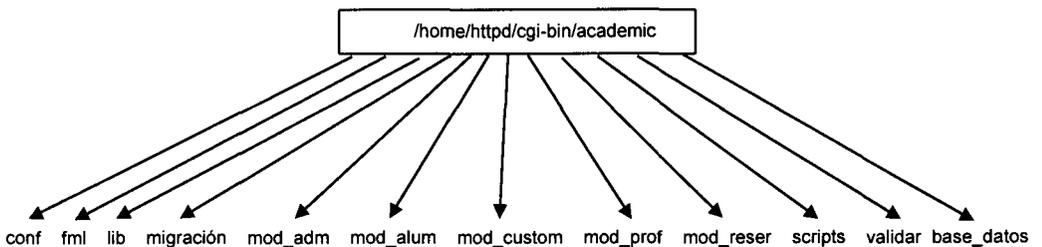


Figura 4.12 Estructura de los directorios de programas

Directorio	Descripción
/conf	Contiene los archivos de configuración del sistema.
/fml	Contiene los archivos de formato que son utilizados cuando un usuario no tiene su propio directorio de archivos de formato.
/lib	Contiene las librerías

/migracion	Contiene los archivos de migración.
/mod_adm	Contiene los CGIs del módulo de administración.
/mod_alum	Contiene los CGIs que ejecutan las opciones disponibles para los estudiantes.
/mod_custom	Contiene los CGIs que ejecutan las opciones para realizar la customización de las páginas.
/mod_prof	Contiene los CGIs que ejecutan las opciones disponibles para los profesores.
/mod_reser	Contiene los CGIs que ejecutan las opciones para la reservación de los recursos físicos.
/scripts	Contiene los JavaScripts que deben ser incluidos en las páginas HTML.
/validar	Contiene los CGIs para realizar las autenticaciones de los usuarios.
/base_datos	Contiene los archivos SQL para exportar los datos del Sistema Académico de la Institución.

Tabla IV Directorios de programas

4.4. Diseño de la Base de Datos

Para Cursos Web será utilizado PostgreSQL como manejador de base de datos relacional y será creada una base de datos llamada **“academic”**.

Será utilizado PostgreSQL por las siguientes razones:

- Permite implementar la base de datos con un diseño relacional.

- Utiliza SQL (Lenguaje Estándar de Consultas), para definir las sentencias necesarias para obtener y administrar la información que almacena. Además PostgreSQL proporciona funciones propias que serán de utilidad en el desarrollo del sistema.
- Es una herramienta que no implica costo alguno, debido a que está incluida en el Sistema Operativo (LINUX) que utilizamos para el desarrollo del proyecto.
- Se puede obtener toda la información necesaria para su aprendizaje y utilización de manera gratuita. Esta información se encuentra disponible en el sitio Web (www.postgres.org).

A continuación encontrará las tablas definidas con una breve explicación, así como las relaciones entre ellas:

Columna	Descripción
Alumno	Mantiene a los estudiantes que se registran en los diferentes cursos, así como a los ayudantes académicos.
apellidos	Apellidos del estudiante.
cod_servidor	Código del servidor contra el que se verifica la autenticación del estudiante, si el tipo de autenticación es Pop3.
clave	Clave del usuario, que es utilizada cuando el tipo de autenticación es Local.
sitio_web	Dirección de la página personal del estudiante en el web.
direccion	Dirección domiciliaria del estudiante.
email	Dirección e-mail del estudiante.
estado_alumno	Estado del estudiante: A - Activo, I - Inactivo.
ultimo_cambio	Indica la fecha y hora del sistema en que fue actualizado el registro. Utilizado para controlar concurrencia en el

	registro de los datos.
present_direc	Indica si la dirección domiciliaria del estudiante podrá ser consultada por los demás integrantes del curso.
present_telef	Indica si los números telefónicos del estudiante podrán ser consultados por los demás integrantes del curso.
matricula	Número de identificación del estudiante.
provincia	Nombre de la provincia de residencia del estudiante.
nick_name	Nombre de pila del estudiante.
foto	Nombre del archivo de la foto del estudiante. Debe ser un .jpg o .gif.
pais	Nombre del país de residencia del estudiante.
usuario	Nombre del usuario asignado a este estudiante.
nombres	Nombres del estudiante.
tam_archivo_alum	Tamaño máximo que puede tener cada archivo que el estudiante publique en un curso. Si es cero, se considerará el tamaño default definido en la tabla de parámetros.
telefono1	Teléfono # 1 del estudiante.
telefono2	Teléfono # 2 del estudiante.
tipo_autentic	Tipo de autenticación a realizarse para este estudiante: L - Local, P - Pop3.
tipo	Tipo de identificación del estudiante: MAT - Matricula, CED - Cédula, PAS - Pasaporte.
Anuncio	Anuncios definidos en los diferentes curso.
anio	Año del período académico.
cod_materia	Código de materia.
fecha	Fecha de publicación del anuncio.
ultimo_cambio	Indica la fecha y hora del sistema en que fue actualizado el registro. Utilizado para controlar concurrencia en el registro de los datos.
nro_anuncio	Número de anuncio.
paralelo	Número de paralelo.
nombre_archivo	Nombre del archivo registrado que contiene la descripción del anuncio.
termino	Término del período académico.
titulo	Título del anuncio.
tipo_archivo	Tipo de archivo registrado: T - Texto, D - Documento.
Calendario	Semanas definidas para el período académico.
descrip_semana	Descripción de la semana calendario.

fecha_inicio	Fecha de inicio de la semana. Consideramos los días lunes como inicio de la semana calendario.
nro_semana	Identifica el número de semana dentro del período.
ultimo_cambio	Indica la fecha y hora del sistema en que fue actualizado el registro. Utilizado para controlar concurrencia en el registro de los datos.
Calific_Alumno	Calificaciones obtenidas por los estudiantes de un curso en cada uno de los componentes de calificación definidos.
anio	Año del período académico.
cod_materia	Código de materia.
ultimo_cambio	Indica la fecha y hora del sistema en que fue actualizado el registro. Utilizado para controlar concurrencia en el registro de los datos.
matricula	Número de identificación del estudiante
paralelo	Número de paralelo.
nro_polit_calif	Número de política de calificación
nota	Nota obtenida por el estudiante en la política de calificación especificada.
termino	Término del período académico.
tipo	Tipo de identificación del estudiante: MAT - Matricula, CED - Cédula, PAS - Pasaporte.
Curriculum	Información adicional de cada estudiante o empleado.
ultimo_cambio	Indica la fecha y hora del sistema en que fue actualizado el registro. Utilizado para controlar concurrencia en el registro de los datos.
educacion	Estudios académicos.
experienc	Experiencia laboral.
hobbies	Aficiones.
nro_identif	Tipo de identificación de la persona.
tipo_identif	Número de identificación de la persona.
Detalle_Solicitud	Especifica cada uno de los días y horas en los que un recurso físico es solicitado.
cod_unidad	Código de la unidad académica en la que es realizada la solicitud del recurso.
cod_recurso	Código de recurso a solicitar.
cod_tipo_recu	Código del tipo de recurso.
estado_detal	Estado del detalle de la solicitud: P - Pendiente, E - Entregado, D - Devuelto.
fecha_devol	Fecha de devolución del recurso.

hora_devol	Hora de devolución del recurso.
hora_final	Hora final en la que será desocupado el recurso en el día.
ultimo_cambio	Indica la fecha y hora del sistema en que fue actualizado el registro. Utilizado para controlar concurrencia en el registro de los datos.
nro_semdia_hi	Número de la semana, día y hora inicial de la solicitud.
nro_solicitud	Número de solicitud a la que pertenece.
Dia_Calendario_Profesor	Eventos en día de calendario de un curso dentro de un período académico.
anio_cal	Año calendario.
anio	Año del período académico.
cod_materia	Código de materia.
dia	Día calendario.
descripcion	Descripción de la clase que será cubierta en el día especificado.
ultimo_cambio	Indica la fecha y hora del sistema en que fue actualizado el registro. Utilizado para controlar concurrencia en el registro de los datos.
migrado	Indica si el evento en día de calendario ha sido migrado o no: S - Si, N - No.
mes_cal	Mes calendario.
paralelo	Número de paralelo.
nombre_archivo	Si el tipo de archivo es texto, este campo tiene el nombre del archivo registrado que contiene la descripción de la clase en el día calendario. Si el tipo de archivo es documento, este campo tiene el nombre del archivo contiene los nombres de los archivos de descripción de la clase en el día calendario.
termino	Término del período académico.
titulo	Título otorgado al día de clase.
tipo_archivo	Tipo de archivo registrado: T - Texto, D - Documento
Empleado	Personas que trabajan para la Institucion (profesores, administradores).
apellidos	Apellidos del empleado.
cod_servidor	Código del servidor contra el que se verifica la autenticación del empleado, si el tipo de autenticación es Pop3.
clave	Clave del usuario, que es utilizada cuando el tipo de autenticación es Local.
sitio_web	Dirección de la página personal del empleado en el web.
direccion	Dirección domiciliaria del empleado.

email	Dirección e-mail del empleado.
estado_trab	Estado del empleado: A - Activo, I - Inactivo
tipo_personal	Identifica si el empleado es docente o administrativo: D - Docente, A - Administrativo
ultimo_cambio	Indica la fecha y hora del sistema en que fue actualizado el registro. Utilizado para controlar concurrencia en el registro de los datos.
present_direc	Indica si la dirección domiciliaria del empleado podrá ser consultada por sus estudiantes.
present_telef	Indica si los teléfonos del empleado podrán ser consultados por sus estudiantes.
nro_identif	Número de identificación del empleado.
provincia	Nombre de la provincia de residencia del empleado.
nick_name	Nombre de pila del empleado.
foto	Nombre del archivo de la foto del empleado. Debe ser un .jpg o .gif.
pais	Nombre del país de residencia del empleado.
usuario	Nombre del usuario asignado a este empleado.
nombres	Nombres del empleado.
tam_archivo	Tamaño máximo que puede tener cada archivo que el empleado publique en un curso. Si es cero, se considerará el tamaño default definido en la tabla de parámetros.
telefono1	Teléfono # 1 del empleado.
telefono2	Teléfono # 2 del empleado.
tipo_autentic	Tipo de autenticación a realizarse para este empleado: L - Local, P - Pop3.
tipo_identif	Tipo de identificación del empleado.
Foro_Discusion	Mensajes publicados en el foro de discusión en cada curso.
anio	Año del período académico.
cod_materia	Código de materia.
padre	Código de mensaje que se está respondiendo.
ts_envio	Fecha y hora en que se publicó el mensaje.
ts_envio_detalle	Fecha y hora para ordenar los hilos de mensajes iniciales en base a las fechas y horas de las respuestas.
ultimo_cambio	Indica la fecha y hora del sistema en que fue actualizado el registro. Utilizado para controlar concurrencia en el registro de los datos.
nro_identif_prop	Número de identificación del usuario que publicó el mensaje.
nro_mensaje	Número de mensaje.

paralelo	Número de paralelo.
nombre_archivo	Nombre de archivo adjunto al mensaje.
secuencial	Secuencia de mensajes dentro de un mismo grupo.
termino	Término del período académico.
titulo	Título del mensaje.
tipo_archivo	Tipo de archivo adjunto: T - Texto, D - Documento, A - Ambos.
tipo_identif_prop	Tipo de identificación del usuario que publicó el mensaje: MAT - Matricula CED - Cédula PAS - Pasaporte.
nivel	Ubicación dentro del árbol de un grupo de mensajes.
Informacion_Profesor	Información adicional definida en los cursos.
anio	Año del período académico.
cod_materia	Código de materia.
ultimo_cambio	Indica la fecha y hora del sistema en que fue actualizado el registro. Utilizado para controlar concurrencia en el registro de los datos.
nro_informacion	Número de información.
paralelo	Número de paralelo.
nombre_archivo	Nombre del archivo registrado que contiene la información proporcionada.
termino	Término del período académico.
titulo	Título de la información proporcionada.
tipo_archivo	Tipo de archivo registrado: T - Texto, D - Documento.
Materia	Materias académicas.
cod_materia	Código de materia.
cod_unidad	Código de unidad académica a la que pertenece la materia.
estado_materia	Estado de materia: A - Activo, I - Inactivo.
ultimo_cambio	Indica la fecha y hora del sistema en que fue actualizado el registro. Utilizado para controlar concurrencia en el registro de los datos.
tiene_mejoram	Indica si la materia tiene examen de mejoramiento: S - Si, N - No.
num_examenes	Número de exámenes, sin considerar mejoramiento, que deben tomarse en esta materia.
nombre_materia	Nombre de materia.
Migracion_Material	Indica cuál es el tipo de material que un profesor ha migrado de un período académico a otro.

anio	Año del período académico.
anio_migra	Año del que se está migrando el material.
cod_materia	Código de materia.
ultimo_cambio	Indica la fecha y hora del sistema en que fue actualizado el registro. Utilizado para controlar concurrencia en el registro de los datos.
informacion	Indica si se ha migrado información de curso: S - Si, N - No.
anuncios	Indica si se han migrado anuncios: S - Si, N - No.
calendarios	Indica si se han migrado eventos en días de calendario: S - Si, N - No.
politicas	Indica si se han migrado políticas de curso: S - Si, N - No.
recursos	Indica si se han migrado recursos académicos: S - Si, N - No.
tareas	Indica si se han migrado tareas: S - Si, N - No.
paralelo	Número de paralelo.
termino	Término del período académico.
termino_migra	Término del que se está migrando el material.
Paralelo_Alumno	Estudiantes registrados en cada uno de los cursos académicos.
anio	Año del período académico.
cod_materia	Código de materia.
estado_mat_tomada	Estado del total de la nota de la materia tomada: AP - Aprobada, RP - Reprobada, PF - Perdió por faltas, EX - Exonerada, CV - Convalidada, AC - Activa.
ultimo_cambio	Indica la fecha y hora del sistema en que fue actualizado el registro. Utilizado para controlar concurrencia en el registro de los datos.
matricula	Número de identificación del estudiante.
paralelo	Número de paralelo.
nota1	Nota obtenida por el estudiante en el primer parcial.
nota2	Nota obtenida por el estudiante en el segundo parcial.
nota3	Nota obtenida por el estudiante en el tercer parcial.
nota_mejoram	Nota obtenida por el estudiante en mejoramiento.
termino	Término del período académico.
tipo	Tipo de identificación del estudiante: MAT - Matricula, CED - Cédula, PAS - Pasaporte.
Paralelo	Representa a un curso.
anio	Año del período académico.

cod_materia	Código de materia.
estado_paralelo	Estado del paralelo: A - Activo, I - Inactivo.
ultimo_cambio	Indica la fecha y hora del sistema en que fue actualizado el registro. Utilizado para controlar concurrencia en el registro de los datos.
horas_pract_seman	Número de horas prácticas semanales.
horas_pract_total	Número de horas prácticas totales (de todo el curso).
horas_teoric_seman	Número de horas teóricas semanales.
horas_teoric_total	Número de horas teóricas totales (de todo el curso).
nro_identif_ay1	Número de identificación del ayudante 1.
nro_identif_ay2	Número de identificación del ayudante 2.
nro_identif_prof	Número de identificación del profesor.
paralelo	Número de paralelo.
termino	Término del período académico.
tipo_identif_ay1	Tipo de identificación del ayudante 1.
tipo_identif_ay2	Tipo de identificación del ayudante 2.
tipo_identif_prof	Tipo de identificación del profesor.
Parametros_Grales	Contiene datos utilizados por el sistema para su correcto funcionamiento.
anio_actual	Año del período académico actual.
codigo	Código único de los parámetros generales.
cod_servidor_def	Código del servidor de autenticación default.
delim_campos	Caracter que será utilizado para delimitar los campos en los archivos para migrar los datos.
fecha_fin	Fecha en la que finaliza el período (año-término) actual.
fecha_inicio	Fecha en la que se da inicio al período (año-término) actual.
ultimo_cambio	Indica la fecha y hora del sistema en que fue actualizado el registro. Utilizado para controlar concurrencia en el registro de los datos.
tiempo_cookie	Minutos permitidos para que el usuario deje de usar el sistema antes de que le pida confirmación de usuario y contraseña.
nro_identif_adm	Número de identificación del administrador general.
nota_maxima	Nota máxima que puede ser otorgada a un estudiante en un parcial.
prom_minimo	Promedio mínimo que debe obtener un estudiante para aprobar una materia.
termino_actual	Término del período académico actual.
tam_archivo_alum	Tamaño máximo default para archivos definidos por los estudiantes.

tam_archivo_prof	Tamaño máximo default para archivos definidos por los profesores.
tipo_identif_adm	Tipo de identificación del administrador general.
Politica_Calific	Componentes de calificación definidos por los profesores en los diferentes cursos.
anio	Año del período académico.
cod_materia	Código de materia.
ultimo_cambio	Indica la fecha y hora del sistema en que fue actualizado el registro. Utilizado para controlar concurrencia en el registro de los datos.
paralelo	Número de paralelo.
parcial_nota	Número de parcial en el que será considerada esta política de calificación: 1 - Primer parcial, 2 - Segundo parcial, 3 - Tercer parcial, 4 - Mejoramiento.
nro_polit_calif	Número de política de calificación.
descripcion	Nombre asociado al concepto que será calificado por el profesor.
calificacion	Nota máxima que podrá ser asignado en el concepto especificado.
porc_nota	Porcentaje sobre la nota final del parcial.
termino	Término del período académico.
Politica_Profesor	Políticas que deben ser cumplidas por los integrantes del curso.
anio	Año del período académico.
cod_materia	Código de materia.
descripcion	Descripción de la política.
ultimo_cambio	Indica la fecha y hora del sistema en que fue actualizado el registro. Utilizado para controlar concurrencia en el registro de los datos.
nro_politica	Número de la política de curso.
paralelo	Número de paralelo.
termino	Término del período académico.
titulo	Título de la política.
Proceso	Mantiene las conexiones al sistema de los distintos usuarios.
cod_servidor	Código de servidor del usuario conectado.
ip	Dirección IP de la máquina desde la cual el usuario está conectado.
estado_conexion	Estado de la conexión: A - Abierta, C - Cerrada.

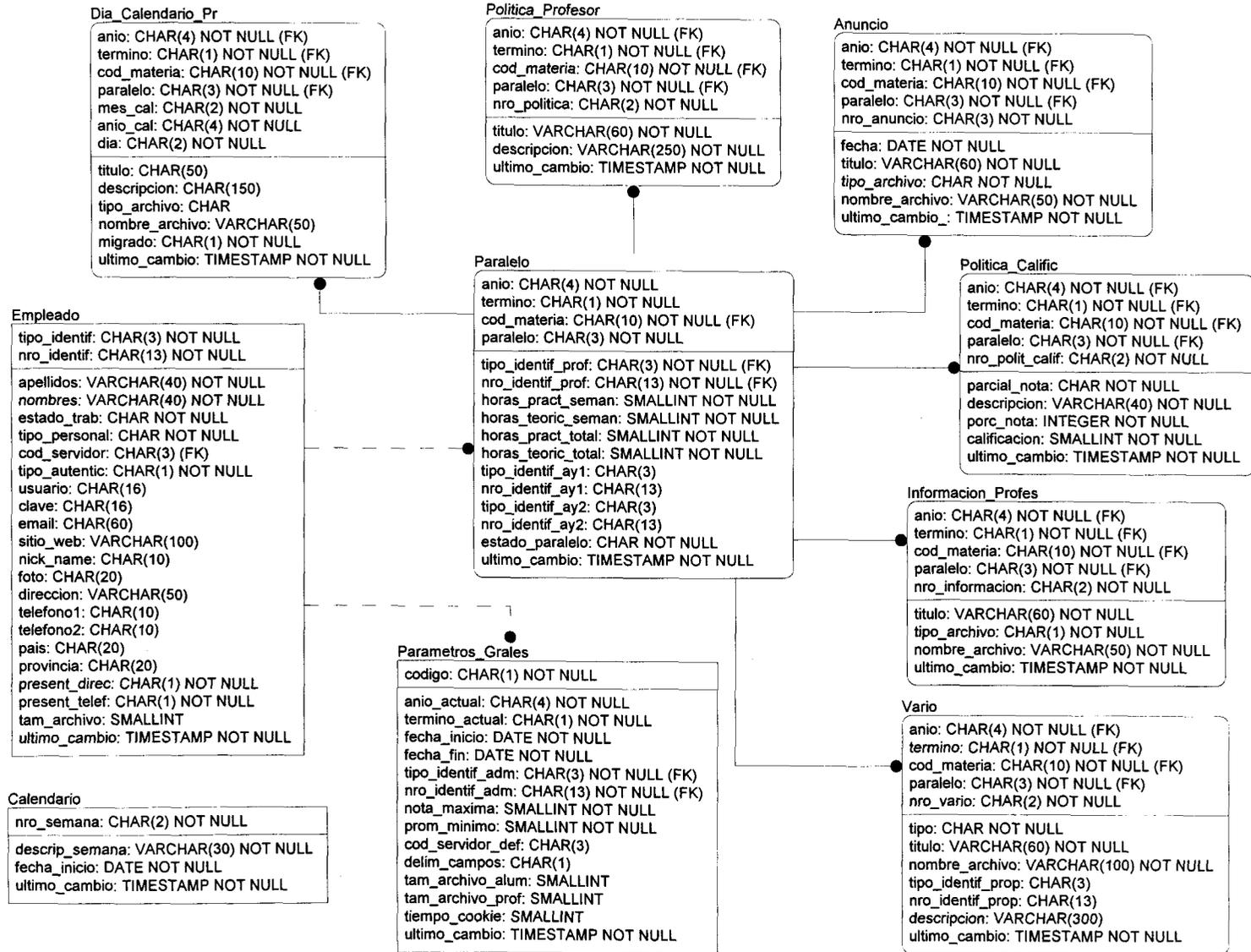
fecha_hora	Fecha y hora de inicio de conexión.
ultimo_cambio	Indica la fecha y hora del sistema en que fue actualizado el registro. Utilizado para controlar concurrencia en el registro de los datos.
nro_proceso	Número de proceso.
usuario	Nombre del usuario conectado.
Recurso	Recursos físicos existentes en cada unidad académica que podrían ser reservados.
cod_servidor_adm	Código de servidor de autenticación del usuario administrador del recurso.
cod_unidad	Código de unidad académica al que pertenece el recurso.
cod_recurso	Código del recurso.
cod_tipo_recu	Código del tipo de recurso.
capacidad	Capacidad del recurso. Por ejemplo número máximo de estudiantes que pueden entrar en un laboratorio.
descrip_recurso	Descripción del recurso.
estado_recurso	Estado del recurso: A - Activo, I - Inactivo.
ultimo_cambio	Indica la fecha y hora del sistema en que fue actualizado el registro. Utilizado para controlar concurrencia en el registro de los datos.
localidad	Localidad donde se encuentra el recurso: P - Peñas, E - Prosperina.
usuario_adm	Nombre de usuario administrador del recurso.
Respuesta_Tarea	Respuestas de los estudiantes a cada una de las tareas enviadas.
anio	Año del periodo académico.
cod_materia	Código de materia.
fecha	Fecha en que fue registrada la respuesta de tarea.
estado_entrega	Identifica si la tarea fue entregada a tiempo o atrasada: A - Atrasada, T - A tiempo.
ultimo_cambio	Indica la fecha y hora del sistema en que fue actualizado el registro. Utilizado para controlar concurrencia en el registro de los datos.
estado_revision	Indica si la tarea ha sido revisada por el profesor: P - Pendiente, R - Revisada.
matricula	Número de identificación del estudiante.
paralelo	Número de paralelo.
nro_tarea	Número de tarea a la pertenece la respuesta.
nombre_archivo	Nombre del archivo registrado, que contiene la descripción de la respuesta de tarea.

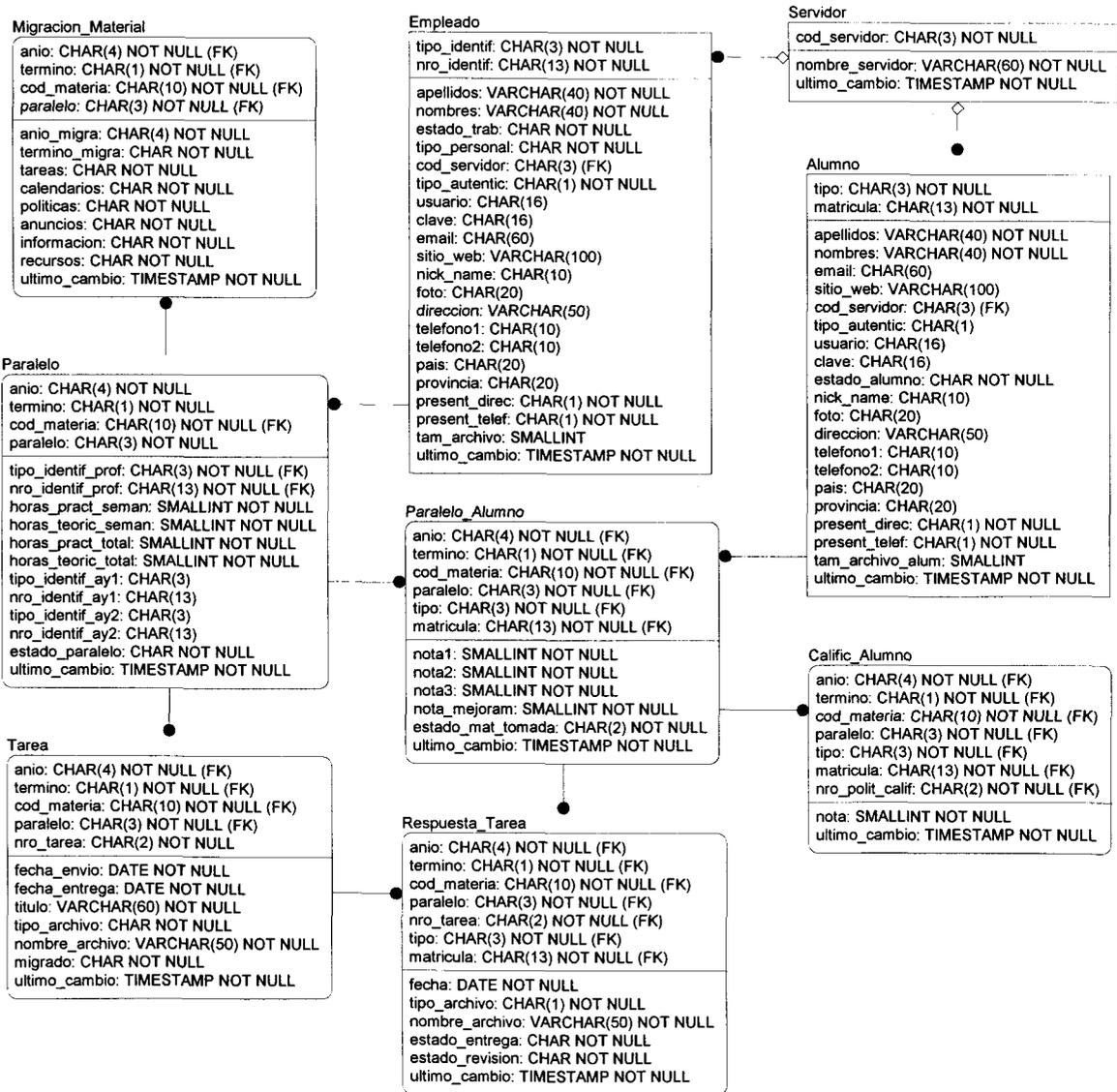
termino	Término del período académico.
tipo_archivo	Tipo de archivo registrado: T - Texto, D - Documento.
tipo	Tipo de identificación del estudiante: MAT - Matricula, CED - Cédula, PAS - Pasaporte.
Servidor	Define a cada uno de los servidores de autenticación.
cod_servidor	Código de servidor, que será utilizado para autenticar usuarios.
ultimo_cambio	Indica la fecha y hora del sistema en que fue actualizado el registro. Utilizado para controlar concurrencia en el registro de los datos.
nombre_servidor	Nombre del servidor
Solicitud_Recurso	Solicitudes de recursos físicos registradas por los distintos usuarios del sistema.
cod_unidad	Código de la unidad académica en la que es realizada la solicitud del recurso.
cod_recurso	Código de recurso a solicitar.
cod_tipo_recu	Código del tipo de recurso.
estado_solic	Estado de la solicitud: E - Espera, A - Aprobado.
fecha_aprob	Fecha de aprobación de la solicitud.
ts_pedido	Fecha y hora en que fue registrada la solicitud.
ultimo_cambio	Indica la fecha y hora del sistema en que fue actualizado el registro. Utilizado para controlar concurrencia en el registro de los datos.
nro_identif	Número de identificación del solicitante.
nro_solicitud	Número de solicitud generada.
observacion	Observación que puede ser definida al hacer la solicitud.
tipo_identif	Tipo de identificación del solicitante: MAT - Matricula, CED - Cédula, PAS - Pasaporte.
Tarea	Tareas enviadas por el profesor a los estudiantes de un curso.
anio	Año del período académico.
cod_materia	Código de materia.
fecha_envio	Fecha en que fue publicada la tarea.
fecha_entrega	Fecha máxima en la que puede ser recibida la tarea para ser considerada entregada a tiempo.
ultimo_cambio	Indica la fecha y hora del sistema en que fue actualizado el registro. Utilizado para controlar concurrencia en el registro de los datos.
paralelo	Número de paralelo.

nro_tarea	Número de tarea asignado.
nombre_archivo	Nombre del archivo en el que se encuentra el detalle de la tarea enviada.
termino	Término del período académico.
titulo	Título de la tarea.
tipo_archivo	Tipo de archivo registrado: T - Texto, D - Sólo documento, A - Ambos (documento y texto).
migrado	Indica si la tarea ha sido migrada o no.
Tipo_Recurso	Tipos de recursos físicos existentes en la Institución.
cod_tipo_recu	Código del tipo de recurso.
descripcion_tipo	Descripción del tipo de recurso.
estado_tipo_recu	Estado del tipo de recurso: A - Activo, I - Inactivo.
ultimo_cambio	Indica la fecha y hora del sistema en que fue actualizado el registro. Utilizado para controlar concurrencia en el registro de los datos.
Unidad_Academica	Distintas áreas académicas dentro de la Institución. Cada una de éstas tendrá definido un administrador.
cod_servidor_adm	Código de servidor de autenticación del usuario administrador de la unidad académica.
cod_unidad	Código de unidad académica.
estado_unidad	Estado de la unidad académica: A - Activo, I - Inactivo.
ultimo_cambio	Indica la fecha y hora del sistema en que fue actualizado el registro. Utilizado para controlar concurrencia en el registro de los datos.
nombre_unidad	Nombre de unidad académica.
usuario_adm	Nombre de usuario administrador de la unidad académica.
Vario	Recursos publicados en un curso.
anio	Año del período académico.
cod_materia	Código de materia.
descripcion	Descripción del recurso publicado.
ultimo_cambio	Indica la fecha y hora del sistema en que fue actualizado el registro. Utilizado para controlar concurrencia en el registro de los datos.
nro_identif_prop	Número de identificación de la persona que publicó el recurso.
paralelo	Número de paralelo.
nro_vario	Número de recurso proporcionado.
nombre_archivo	Nombre del archivo registrado que contiene la información del recurso proporcionado.

termino	Término del período académico.
tipo	Tipo de archivo registrado: T - Texto, D - Documento.
tipo_identif_prop	Tipo de identificación de la persona que publicó el recurso.
titulo	Titulo del recurso.

Tabla V Descripción de Tablas de Base de Datos





Paralelo

anio: CHAR(4) NOT NULL
termino: CHAR(1) NOT NULL
cod_materia: CHAR(10) NOT NULL (FK)
paralelo: CHAR(3) NOT NULL

tipo_identif_prof: CHAR(3) NOT NULL (FK)
nro_identif_prof: CHAR(13) NOT NULL (FK)
horas_pract_seman: SMALLINT NOT NULL
horas_teoric_seman: SMALLINT NOT NULL
horas_pract_total: SMALLINT NOT NULL
horas_teoric_total: SMALLINT NOT NULL
tipo_identif_ay1: CHAR(3)
nro_identif_ay1: CHAR(13)
tipo_identif_ay2: CHAR(3)
nro_identif_ay2: CHAR(13)
estado_paralelo: CHAR NOT NULL
ultimo_cambio: TIMESTAMP NOT NULL

Foro_Discusion

anio: CHAR(4) NOT NULL (FK)
termino: CHAR(1) NOT NULL (FK)
cod_materia: CHAR(10) NOT NULL (FK)
paralelo: CHAR(3) NOT NULL (FK)
nro_mensaje: CHAR(3) NOT NULL
nivel: INTEGER NOT NULL
secuencial: INTEGER NOT NULL

nro_identif_prop: CHAR(13) NOT NULL
tipo_identif_prop: CHAR(3) NOT NULL
ts_envio: TIMESTAMP NOT NULL
ts_envio_detalle: TIMESTAMP
padre: INTEGER
titulo: VARCHAR(60)
tipo_archivo: CHAR(1)
nombre_archivo: VARCHAR(50)
ultimo_cambio: TIMESTAMP NOT NULL

Curriculum

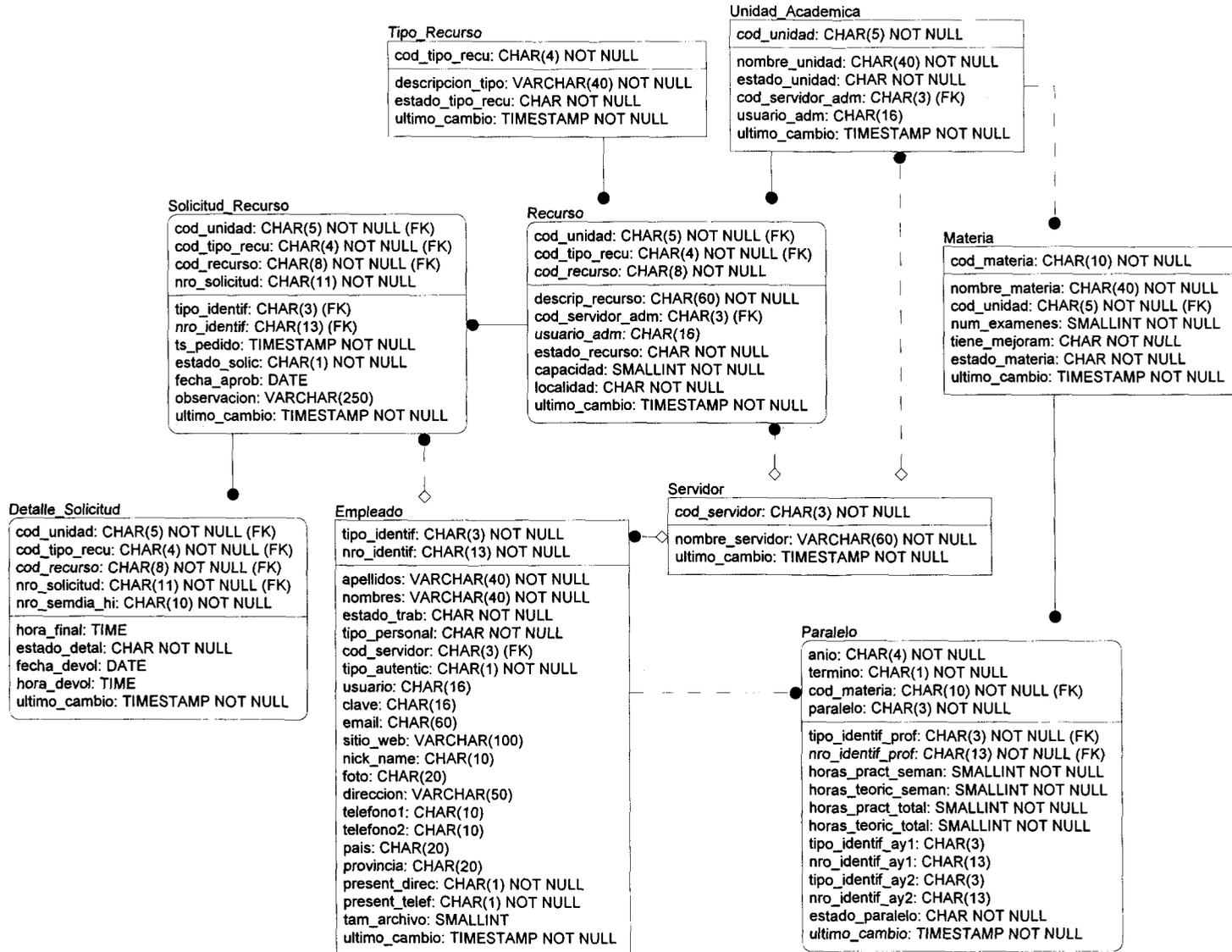
tipo_identif: CHAR(3) NOT NULL
nro_identif: CHAR(13) NOT NULL
educacion: VARCHAR()
hobbies: VARCHAR()
experienc: VARCHAR()
ultimo_cambio: TIMESTAMP NOT NULL

Alumno

tipo: CHAR(3) NOT NULL
matricula: CHAR(13) NOT NULL
apellidos: VARCHAR(40) NOT NULL
nombres: VARCHAR(40) NOT NULL
email: CHAR(60)
sitio_web: VARCHAR(100)
cod_servidor: CHAR(3) (FK)
tipo_autentic: CHAR(1)
usuario: CHAR(16)
clave: CHAR(16)
estado_alumno: CHAR NOT NULL
nick_name: CHAR(10)
foto: CHAR(20)
direccion: VARCHAR(50)
telefono1: CHAR(10)
telefono2: CHAR(10)
pais: CHAR(20)
provincia: CHAR(20)
present_direc: CHAR(1) NOT NULL
present_telef: CHAR(1) NOT NULL
tam_archivo_alum: SMALLINT
ultimo_cambio: TIMESTAMP NOT NULL

Empleado

tipo_identif: CHAR(3) NOT NULL
nro_identif: CHAR(13) NOT NULL
apellidos: VARCHAR(40) NOT NULL
nombres: VARCHAR(40) NOT NULL
estado_trab: CHAR NOT NULL
tipo_personal: CHAR NOT NULL
cod_servidor: CHAR(3) (FK)
tipo_autentic: CHAR(1) NOT NULL
usuario: CHAR(16)
clave: CHAR(16)
email: CHAR(60)
sitio_web: VARCHAR(100)
nick_name: CHAR(10)
foto: CHAR(20)
direccion: VARCHAR(50)
telefono1: CHAR(10)
telefono2: CHAR(10)
pais: CHAR(20)
provincia: CHAR(20)
present_direc: CHAR(1) NOT NULL
present_telef: CHAR(1) NOT NULL
tam_archivo: SMALLINT
ultimo_cambio: TIMESTAMP NOT NULL



Servidor

cod_servidor: CHAR(3) NOT NULL

nombre_servidor: VARCHAR(60) NOT NULL
ultimo_cambio: TIMESTAMP NOT NULL

Calendario

nro_semana: CHAR(2) NOT NULL

descrip_semana: VARCHAR(30) NOT NULL
fecha_inicio: DATE NOT NULL
ultimo_cambio: TIMESTAMP NOT NULL

Proceso

nro_proceso: CHAR(10) NOT NULL

cod_servidor: CHAR(3) NOT NULL
usuario: CHAR(16) NOT NULL
fecha_hora: TIMESTAMP NOT NULL
ip: VARCHAR(16)
estado_conexion: CHAR
ultimo_cambio: TIMESTAMP NOT NULL

Parametros_Grales

codigo: CHAR(1) NOT NULL

anio_actual: CHAR(4) NOT NULL
termino_actual: CHAR(1) NOT NULL
fecha_inicio: DATE NOT NULL
fecha_fin: DATE NOT NULL
tipo_identif_adm: CHAR(3) NOT NULL (FK)
nro_identif_adm: CHAR(13) NOT NULL (FK)
nota_maxima: SMALLINT NOT NULL
prom_minimo: SMALLINT NOT NULL
cod_servidor_def: CHAR(3)
delim_campos: CHAR(1)
tam_archivo_alum: SMALLINT
tam_archivo_prof: SMALLINT
tiempo_cookie: SMALLINT
ultimo_cambio: TIMESTAMP NOT NULL

4.5. Diseño del esquema de seguridad

El esquema de seguridad diseñado para este proyecto de tesis está basado en la autenticación del usuario y en el uso de SSL para manejar la confidencialidad de los datos de la sesión. Una vez que el usuario ha sido autenticado, se verifica la autorización de acceso para cada una de las opciones del sistema de acuerdo al perfil del usuario y el tiempo de existencia del cookie.

4.5.1. Autenticación

Cada usuario del sistema tiene asignado un nombre de usuario, contraseña y el servidor contra el cual se debe autenticar. La autenticación podrá ser hecha contra el servidor que maneja el sistema (autenticación local) utilizando un nombre de usuario y contraseña ingresados por el administrador del sistema en la base de datos. Si el servidor de verificación no es local, se utiliza POP3 (ver sección 2.3.3) para realizar esta función. En el sistema se puede utilizar los diferentes servidores donde los usuarios ya tienen cuentas de email, tales como: goliat.espol.edu.ec, ceibo.fiec.espol.edu.ec en la ESPOL, o servidores como mail.yahoo.com o cualquier otro servidor

que sea definido en el Sistema de Administración y que soporte conexiones usando el protocolo POP3.

4.5.2. Cookies

Los cookies se definieron en la sección 2.3.4, y debido a que son utilizados para mantener variables de ambiente, en Cursos Web, nos ayudan a otorgar o restringir el acceso de los usuarios a cada opción del sistema. En el sistema los cookies se asignan en los siguientes casos:

- Cuando una persona accede al sistema, primero se autentican el nombre de usuario y la contraseña. Después de la verificación exitosa, se crea un cookie con las propiedades del usuario, tales como: identificación, nombre, apellido, nombre de usuario, servidor de autenticación, tipo de usuario (profesor, alumno, ayudante o administrador), fecha y hora de acceso al sistema.
- Cuando un usuario (profesor, alumno o ayudante) accede a uno de sus cursos, se vuelve a verificar que tal usuario se encuentre registrado en dicho curso (esto es necesario ya que cualquier persona podría acceder a otros cursos utilizando las URLs de las opciones), si el resultado de la verificación es exitoso el sistema

actualiza el cookie con información del paralelo, materia, profesor, fecha y hora de acceso.

- Cada vez que un usuario ingresa a cualquier opción en el sistema, se vuelve a grabar en el cookie la fecha y hora de acceso.
- Los valores que fueron almacenados en los cookies se eliminan cuando el usuario sale del sistema a través de la opción SALIR del menú principal o después de una verificación de nombre de usuario y contraseña fallida.

Hay que aclarar que todos los valores almacenados en los cookies no son utilizados para manejar la seguridad del sistema, algunos de ellos se registran porque son comunes en todas las páginas y de esta manera se evita consultarlos en la base de datos constantemente, por ejemplo el nombre del usuario.

Cuando se graba el cookie del usuario al acceder al sistema, también se almacena en la base de datos un registro de conexión, el mismo que contiene el número de sesión, nombre de usuario, fecha y hora de acceso, la dirección IP del PC desde el cual se accede al sistema y el estado de la conexión (Abierta). Al salir del sistema, se actualiza el estado de la conexión de este registro (Cerrada).

Opciones permitidas

Con el tipo de usuario que se registra en el cookie, se controla el acceso de profesores, ayudantes y estudiantes a las opciones del sistema a ellos permitidas. Por ejemplo, un alumno no podrá acceder a las operaciones que le son permitidas sólo al perfil del profesor.

Tiempo de existencia del cookie

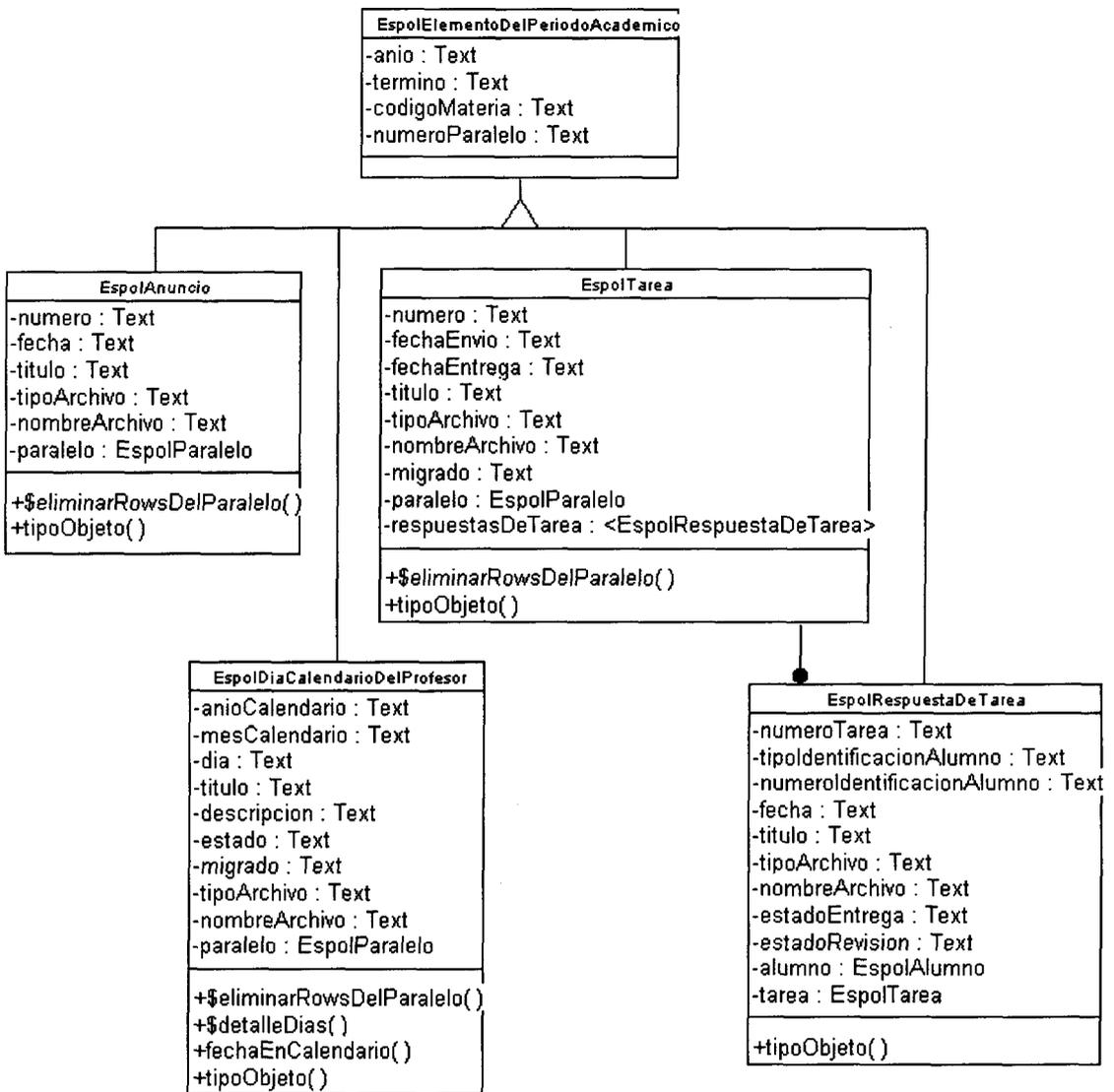
Para acceder a una opción también se verifica el tiempo de existencia del cookie, es decir, si un usuario está mucho tiempo sin navegar en el sistema, se le pide confirmación de usuario y contraseña. Para realizar esto, se obtiene el tiempo de vida del cookie a través de la diferencia entre la fecha y hora del cookie con la fecha y hora del registro de conexión. Si el tiempo es inferior al tiempo de vida del cookie definido como parámetro del sistema, entonces el usuario continúa normalmente. Pero si el tiempo es superior, el sistema le muestra al usuario un mensaje y le pide que ingrese nuevamente su nombre de usuario y contraseña. Luego de la autenticación, el sistema regresa a la última página accedida por el usuario, caso contrario se eliminan todos los cookies y el sistema regresa a la página de inicio.

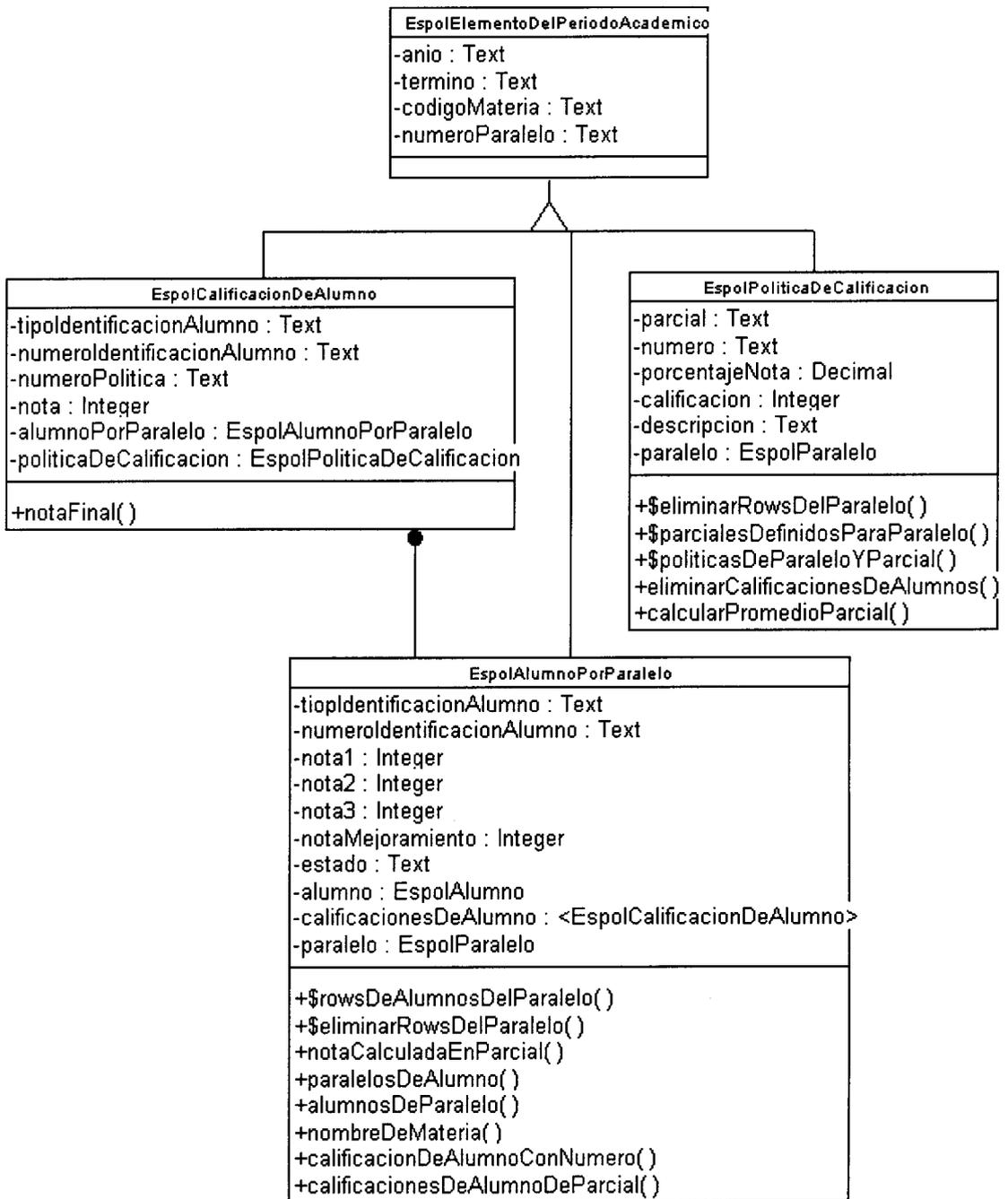
Además de verificar el tiempo de vida del cookie también se valida la dirección IP de la máquina desde la que se está accediendo al sistema. Esto es necesario ya que si una persona tiene una conexión abierta en una PC, y otra intenta acceder con la misma clave, el sistema presenta un mensaje de alerta a ambos usuarios de lo ocurrido. Este control se logra a través del número de sesión que es registrado en la base de datos por usuario (cuando éste entra al sistema), lo cual impide que pueda ser manipulado por alguna otra persona que desee tomar este número de sesión para ingresar con un usuario que no le corresponde.

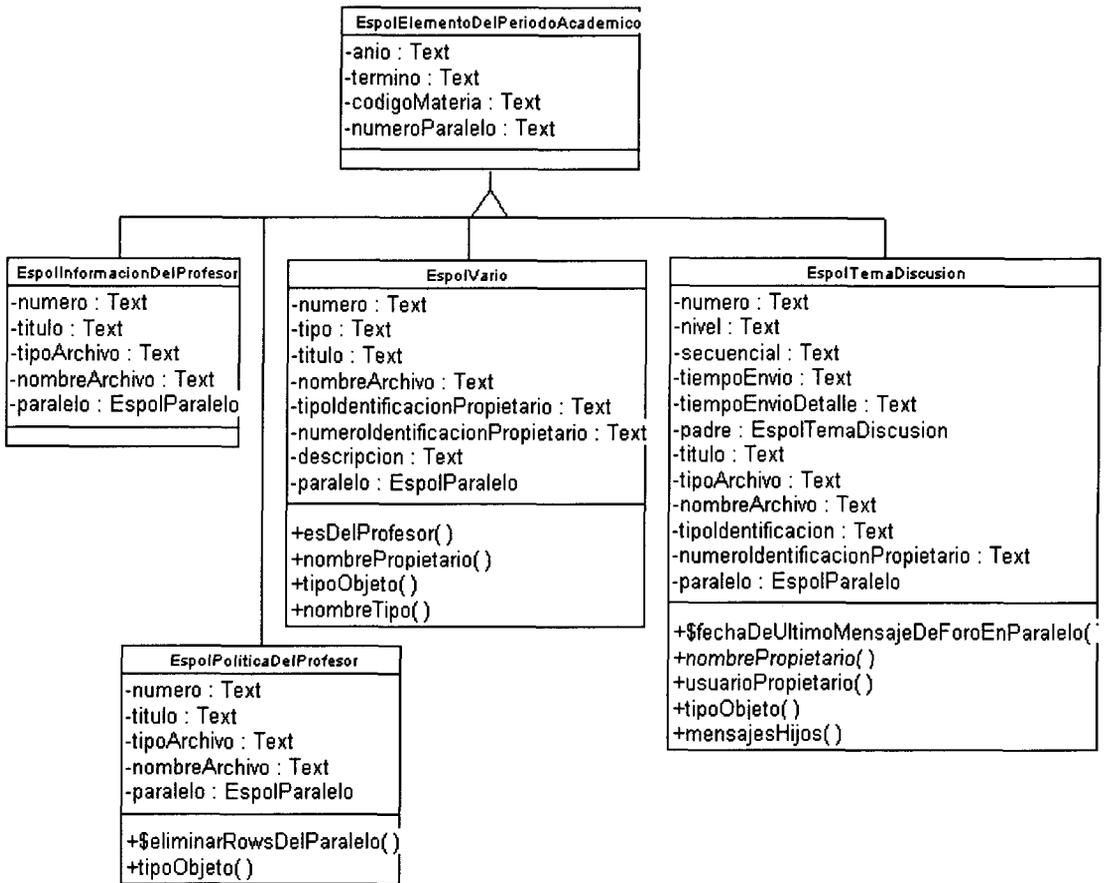
4.6. Diagramas de clases de diseño

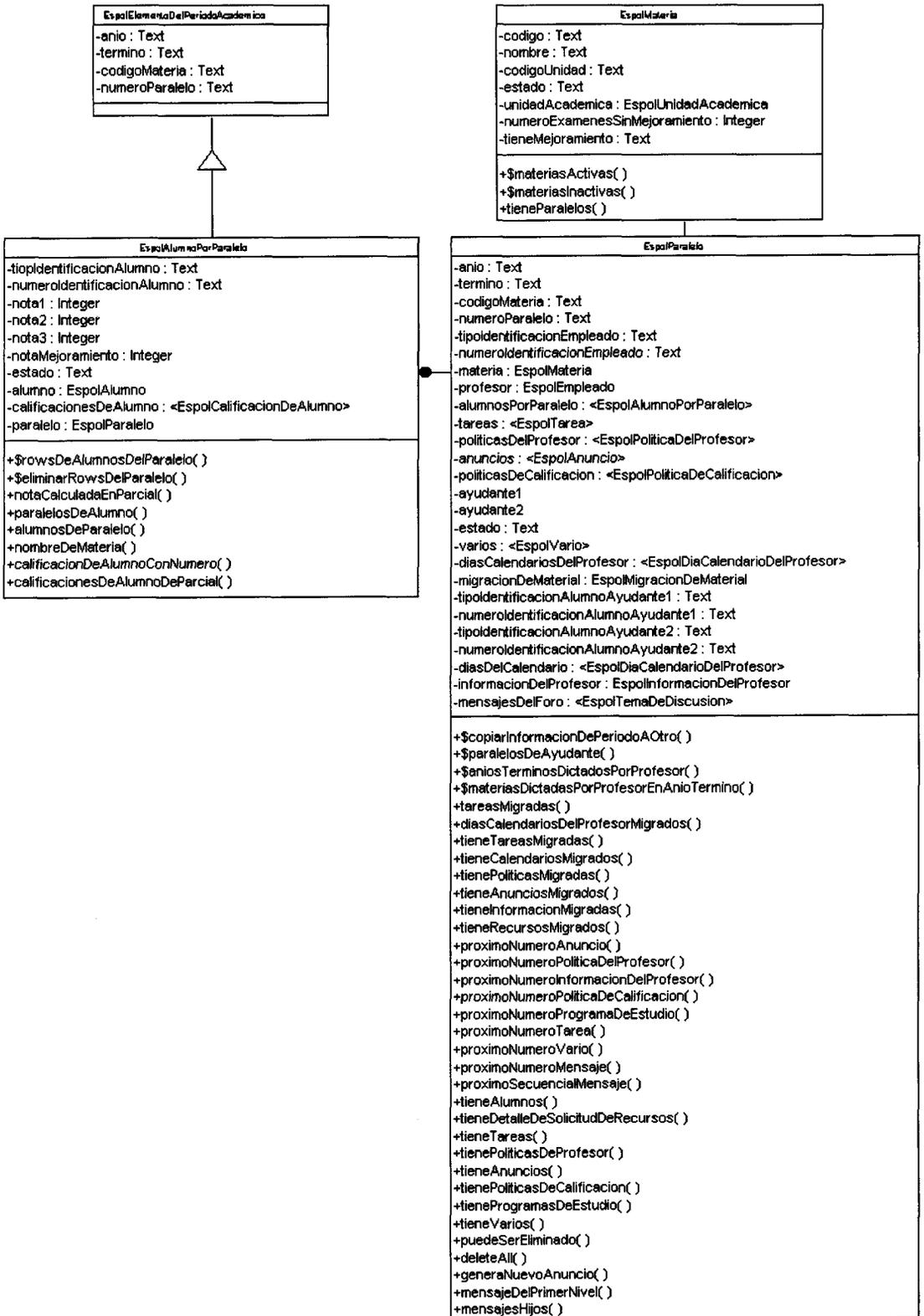
Durante el diseño de clases de objetos, el diseñador toma todos los diagramas de análisis y les da cuerpo a todas las clases que fueron definidas, en base a las necesidades. Los objetos descubiertos durante el análisis sirven como el esqueleto del diseño, pero el diseñador de objetos debe escoger diferentes maneras para implementarlos considerando tiempos de ejecución, memoria y otras medidas de costo.

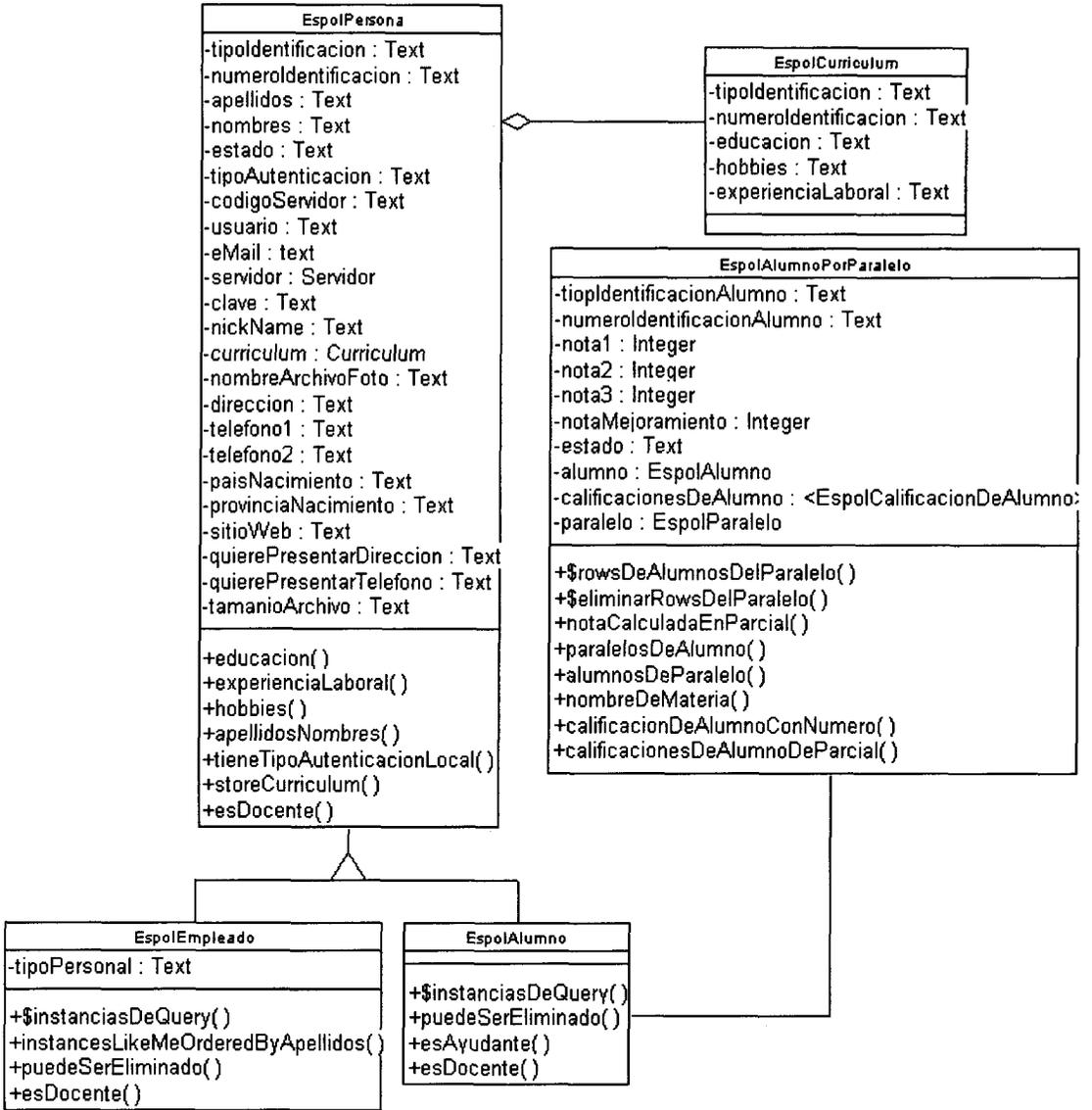
A continuación se encuentran los diagramas de clases que fueron definidos en Cursos Web:











EspolParametrosGenerales
-codigo : Text
-anio : Text
-termino : Text
-fechaInicio : Text
-fechaFin : Text
-tipoidentificacionEmpleado : Text
-numeroidentificacionEmpleado : Text
-codigoServidor : Text
-delimitadorCampos : Text
-notaMaxima : Integer
-promedioMinimo : Integer
-administrador : EspolEmpleado
-servidor : EspolServidor
-tamañoArchivoProfesor : Text
-tamañoArchivoAlumno : Text
-tiempoCookie : Text
+\$parametros()
+\$fechaDelSistema()
+\$mesCalendario()
+\$tsDelSistema()
+\$horaDelSistema()
+\$fechaHoraDeVariable()
+\$fechaHoraCompleta()
+\$semanaActual()
+\$diaActual()
+\$ipMaquina()

EspolMigracionDeMaterial
-anioMigrado : Text
-terminoMigrado : Text
-tarefas : Text
-politicas : Text
-anuncios : Text
-informacion : Text
-recursos : String
-calendarios : Text
-paralelo : EspolParalelo

EspolProceso
-numero : Integer
-codigoServidor : Text
-usuario : Text
-fechaHora : Text
-direccionIP : Text
-estadoConexion : Text
+\$proximoNumeroDeProceso()
+\$eliminaConexiones()
+\$existeConexion()
+\$setConexionAbierta()
+\$setConexionCerrada()

EspolNombreDeParcial
-codigo : Text
-nombre : Text
+\$nombresDeParciales()
+\$nombreDeParcialesParaParalelo()

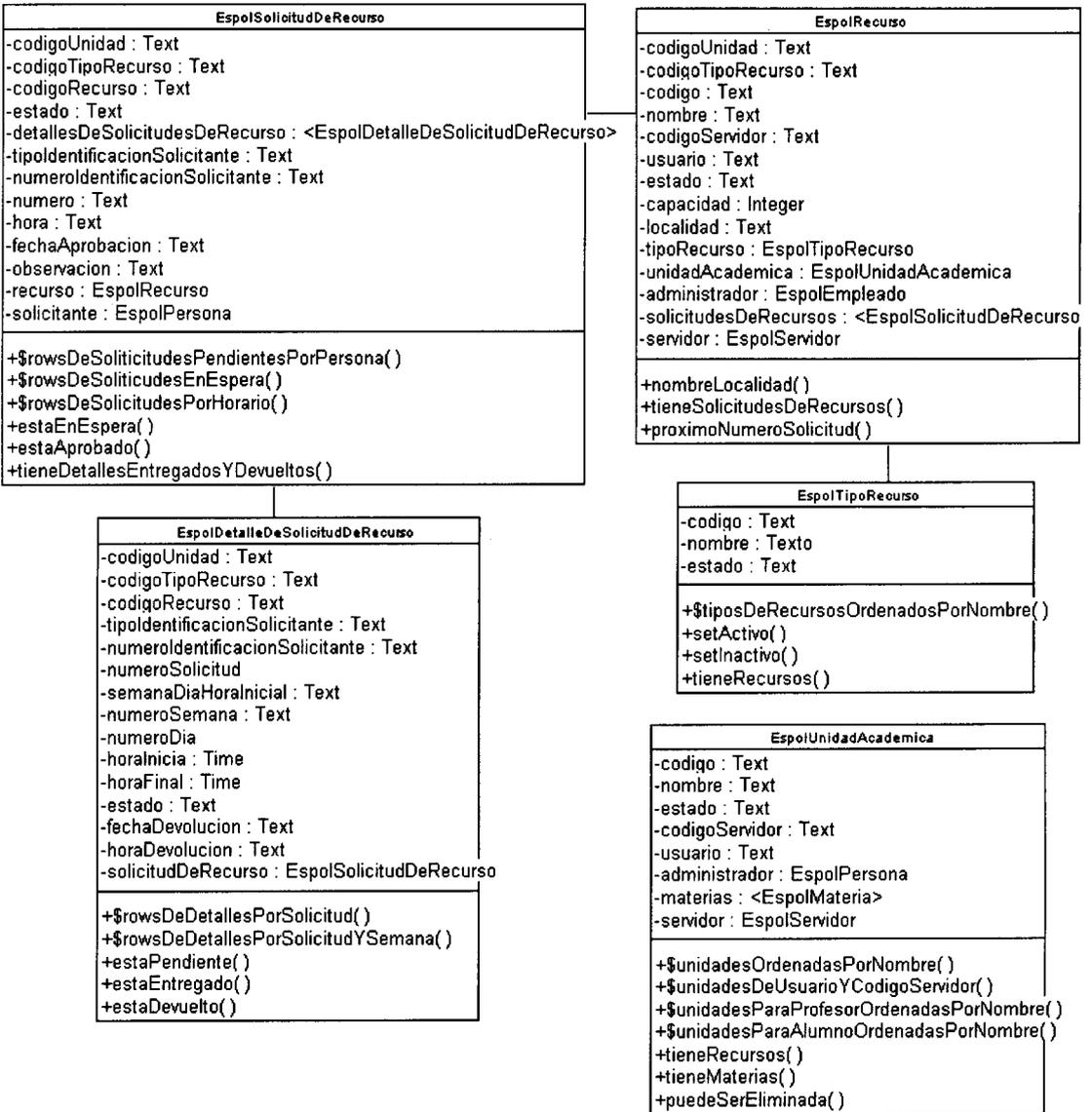
EspolServidor
-codigo : Text
-nombre : Text
+\$servidoresOrdenadosPorNombre()

HtmlError
+\$errorRegistroDuplicado()
+\$errorOperacionNoPermitidaEnMateria()
+\$errorOperacionNoPermitida()
+\$errorAccesoNoPermitido()
+\$errorConexionCerrada()
+\$alertaAcceso()

Customizador
-nombresPaginasPrincipales : <Text>
+\$administradorDeArchivosFml()
+\$puedeCustomizar()
+\$nombreFmlCabeceraHorizontalModelo()
+\$nombreFmlTituloCabeceraHorizontalModelo()
+\$aplicarModelo()
+\$aplicarModeloEnPaginasPrincipales()

HtmlMetaformato
-globalPathFig : Text
-cgiName : Text
-fmlPagina : Text
-tipoLetra : Text
+\$buildPage()
+\$tagInicioTexto()
+\$tagFinTexto()
+\$formaTituloDeContenido()
+\$formaTituloGeneral()
+\$imprimirEncabezado()
+\$imprimirCabecera()
+\$imprimirTitulo()
+\$imprimirDatos()
+\$htmlTextoConReferencias()

HtmlAPI
-dirNameConf : Text
-fileNameConf : Text
+\$findValueInStringLine()
+\$takeValidValue()
+\$linesValid()
+\$nombreFmlPagina()
+\$findValueInFileWithKey()
+\$replaceValueInFileWithKey()
+\$replaceLinesInFileWithKey()
+\$replaceValidValue()
+\$fmlDefault()
+\$workPath()
+\$confPathUser()
+\$dirFmlUser()
+\$dirPersUser()
+\$fileNameTuplas()

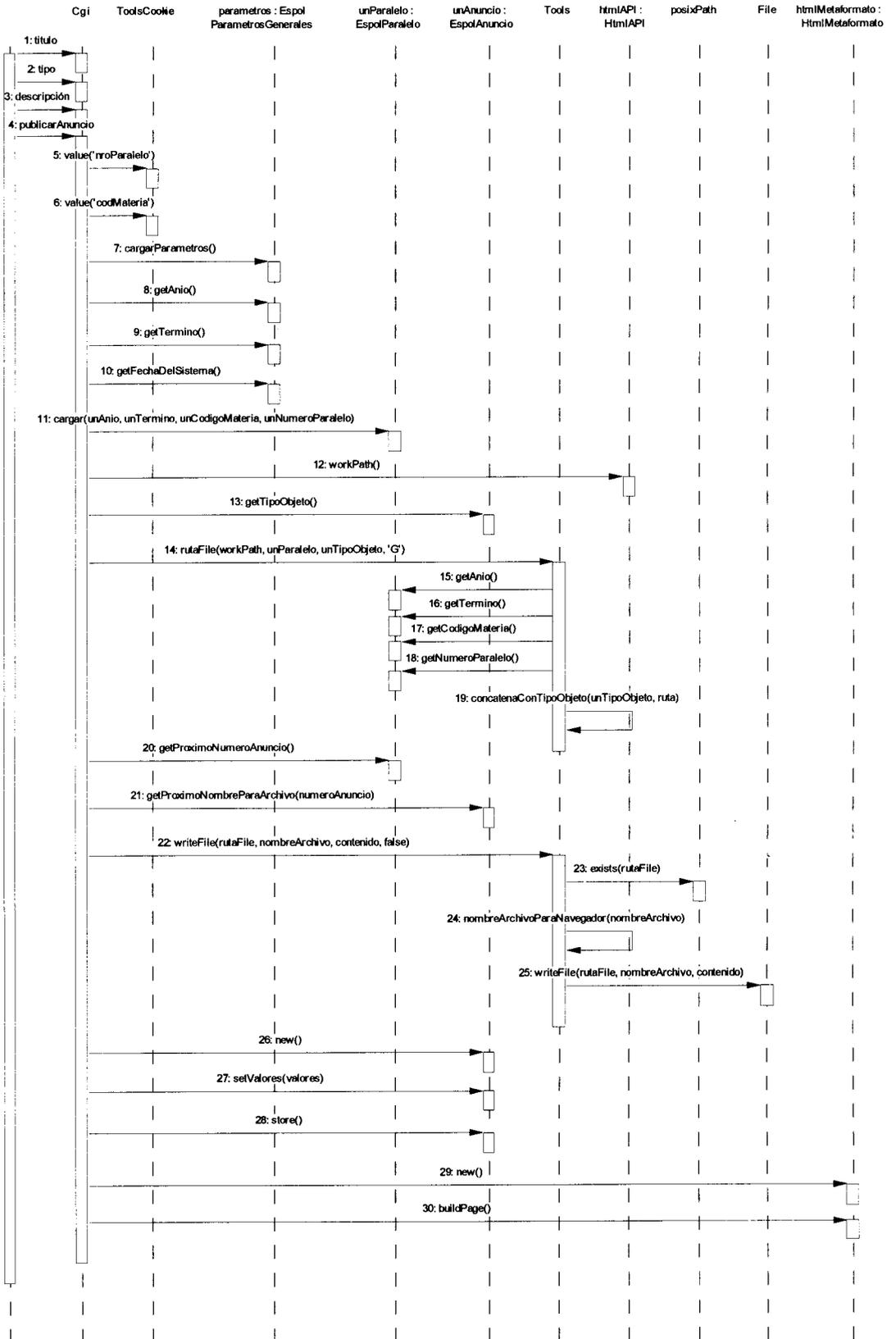


4.7. Diagramas de interacción de objetos de diseño

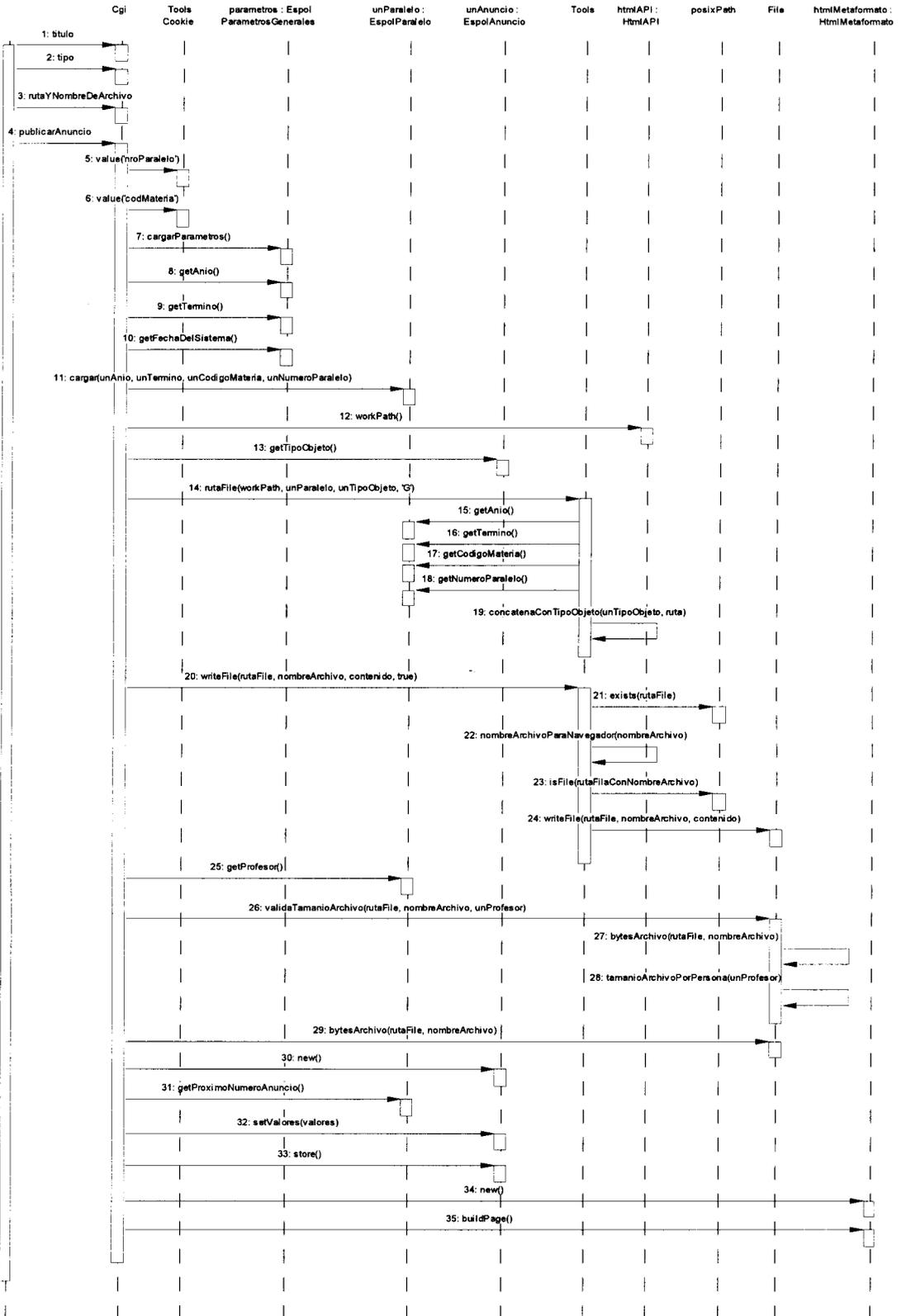
Una vez que se ha definido cuál va a ser el comportamiento que tendrán las clases, así como la forma en que estarán relacionadas unas con otras, se definen los diagramas de interacción de objetos por caso de uso, los mismos que describen el cómo será resuelto un problema. Estos ya incluyen ciertos detalles de implementación, y ayudan a validar todos los cambios aplicados en los diagramas de clases. De tal manera que podemos darnos cuenta cuando de la necesidad de cambios sobre alguna clase en este punto.

Los diagramas de interacción de objetos están definidos a continuación:

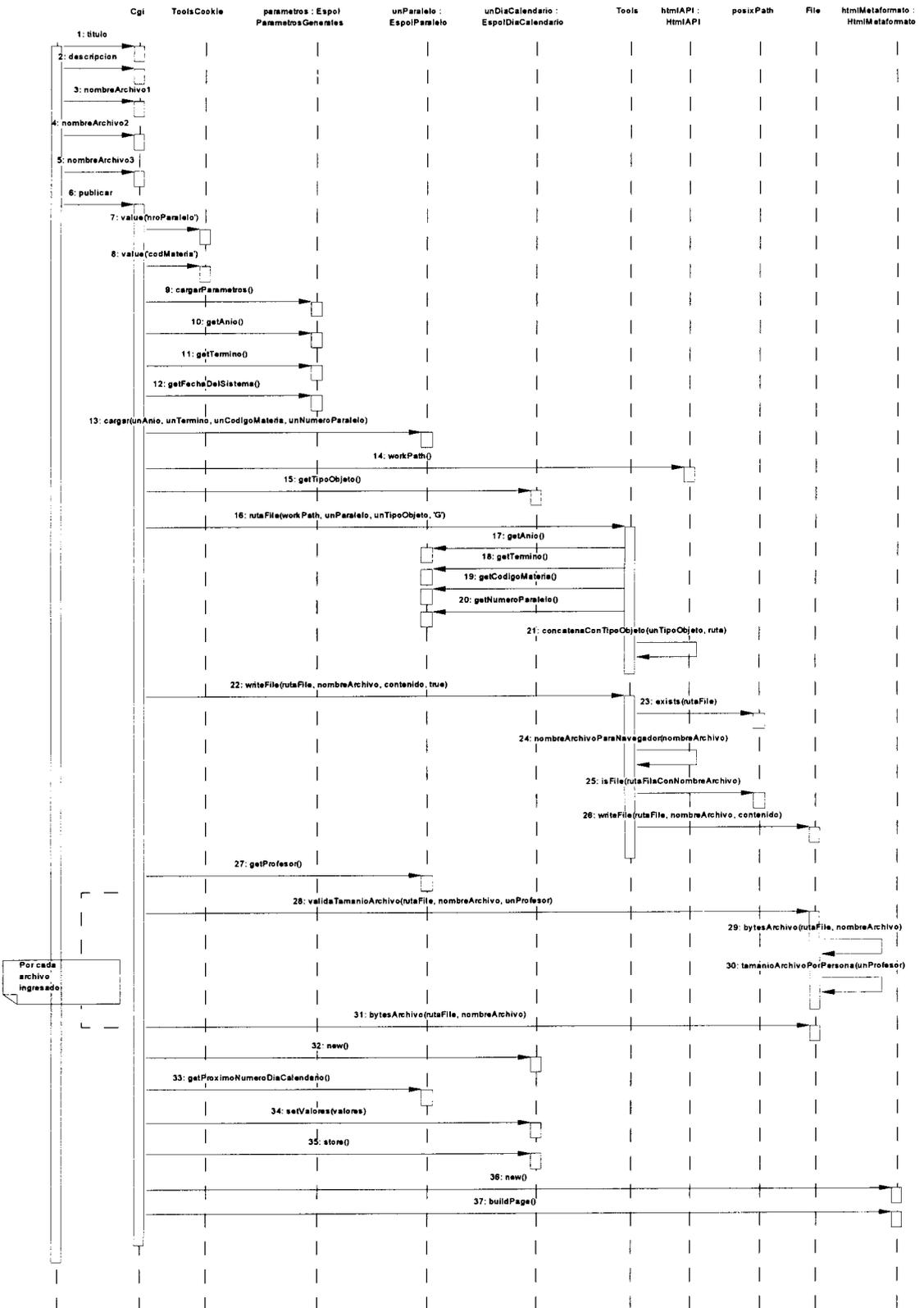
Publicación de un anuncio en forma de texto.



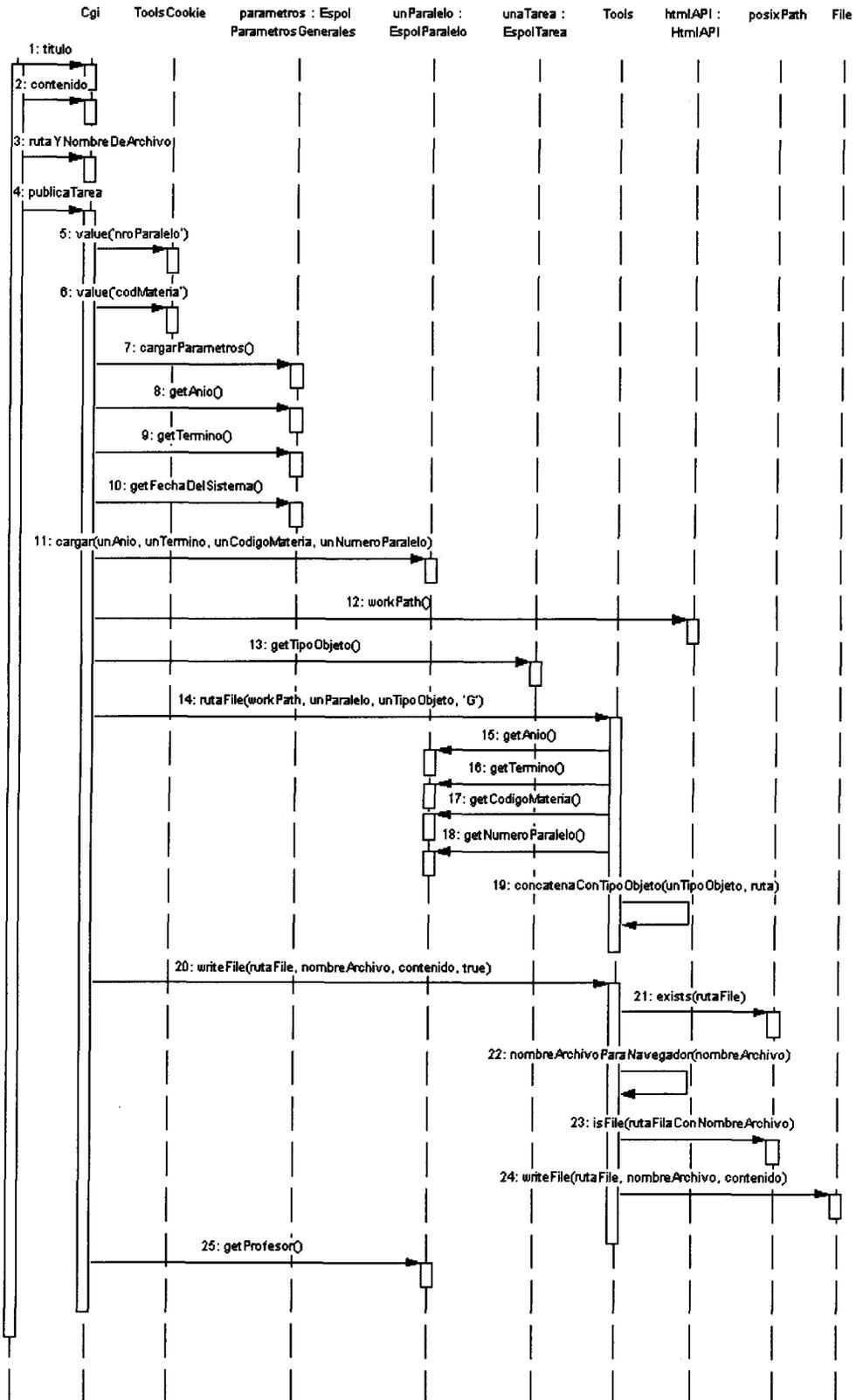
Publicación de un anuncio en forma de archivo.



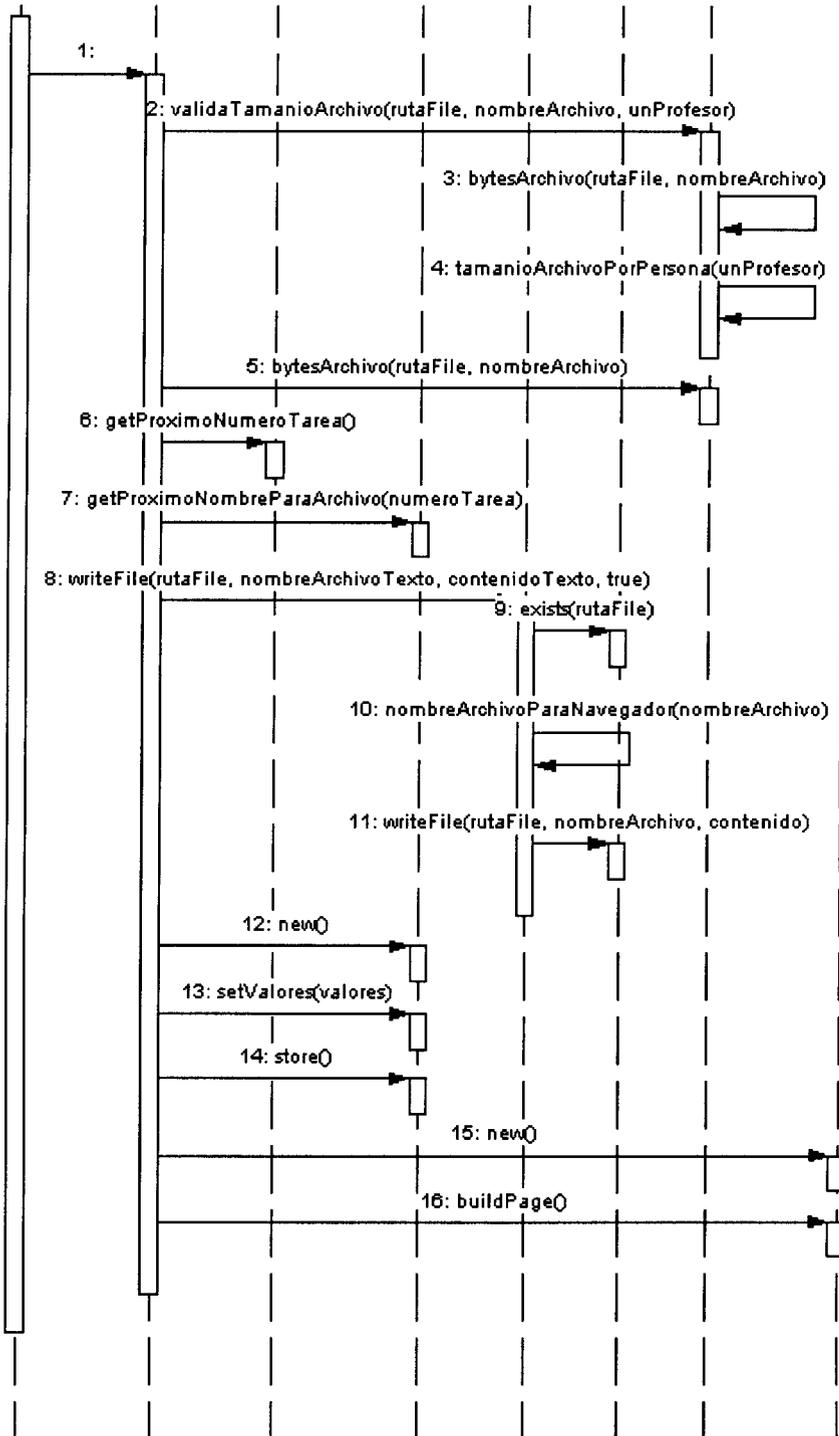
Publicación de un evento en un día de calendario.



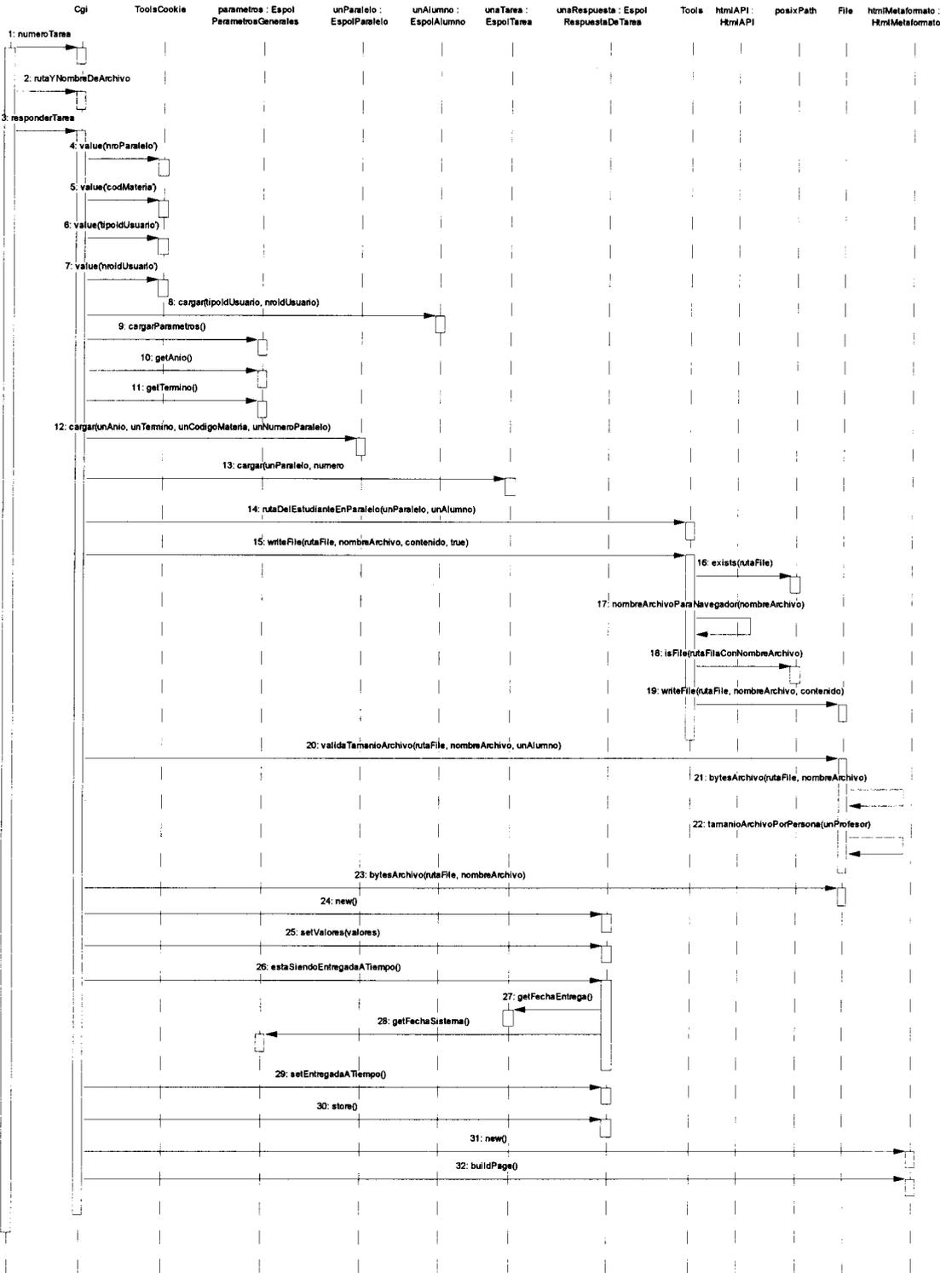
Publicación de una tarea con un archivo adjunto.



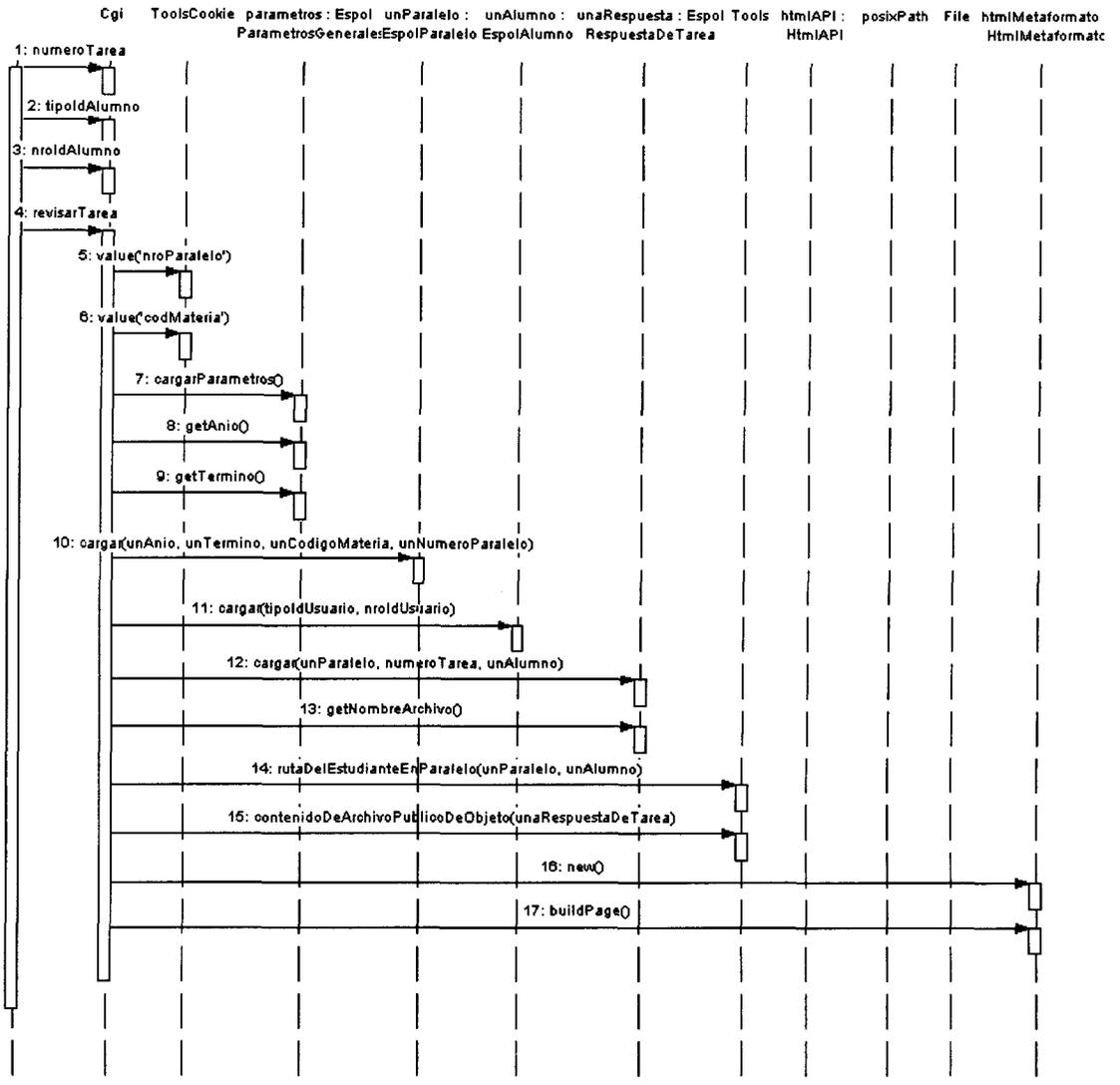
Cgi unParalelo : unaTarea : Tools posixPath File htmlMetaformato :
 EspolParalelo EspolTarea HTMLMetaformato



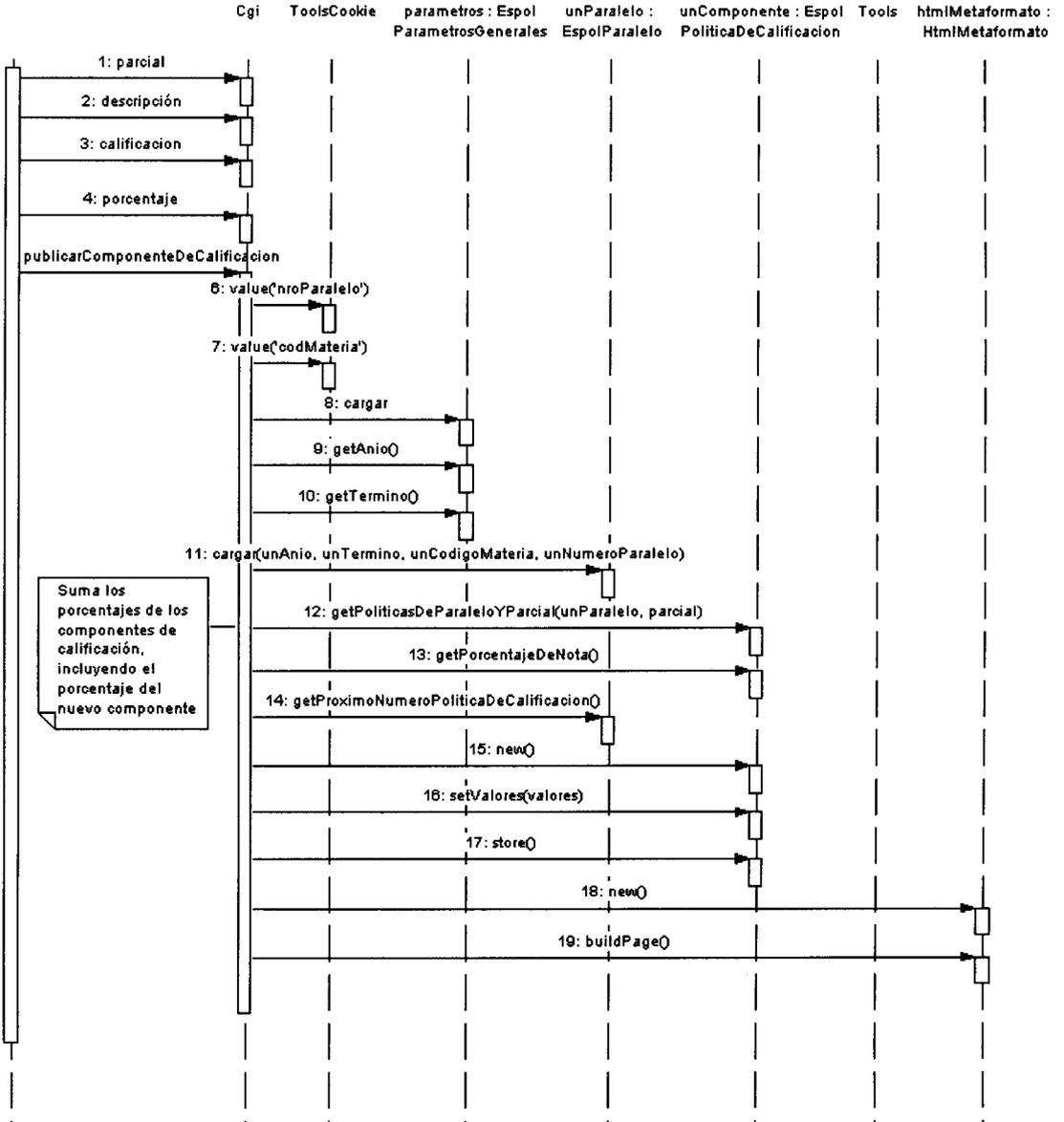
Publicación de respuesta de tareas en forma de archivo.



Revisión de tarea tipo texto.

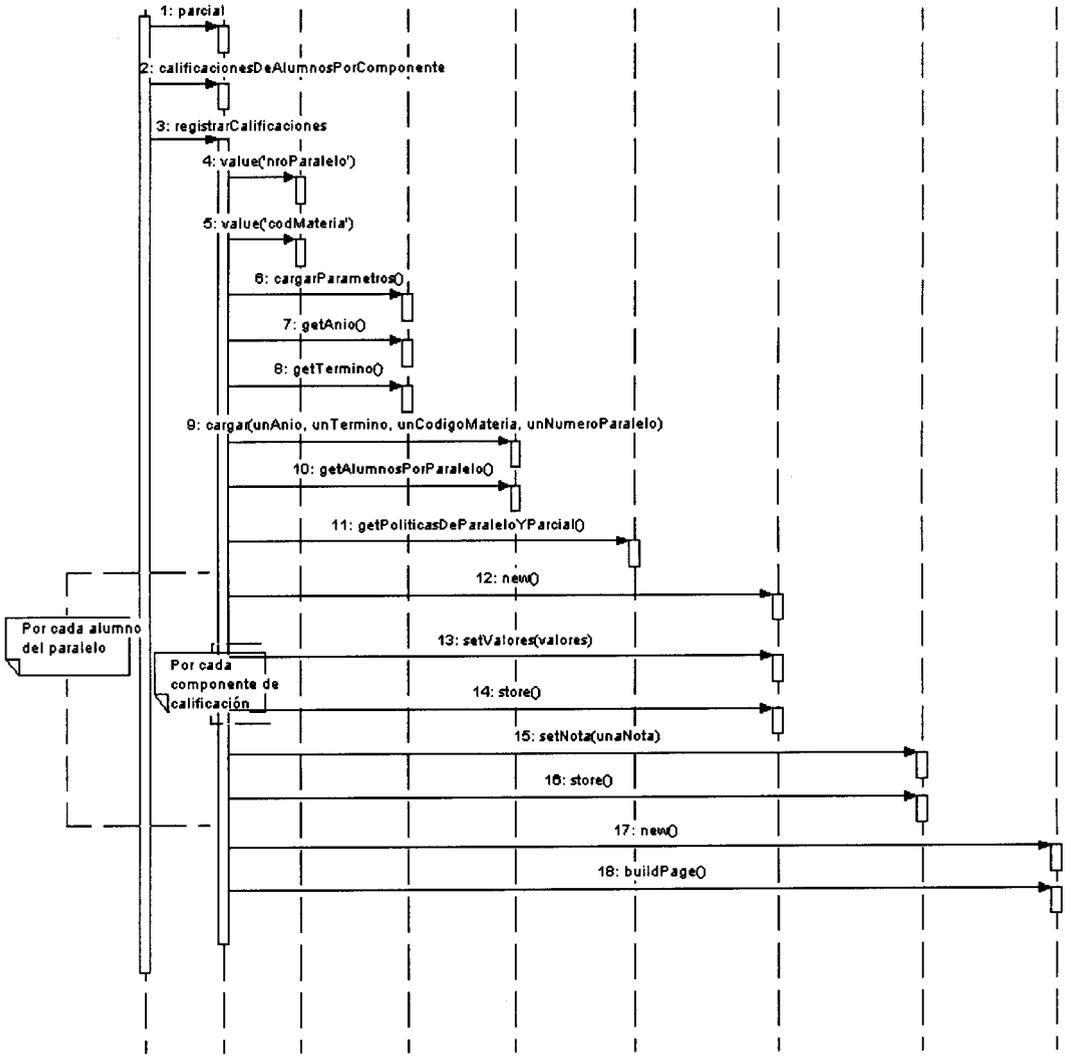


Publicación de un componente de calificación.

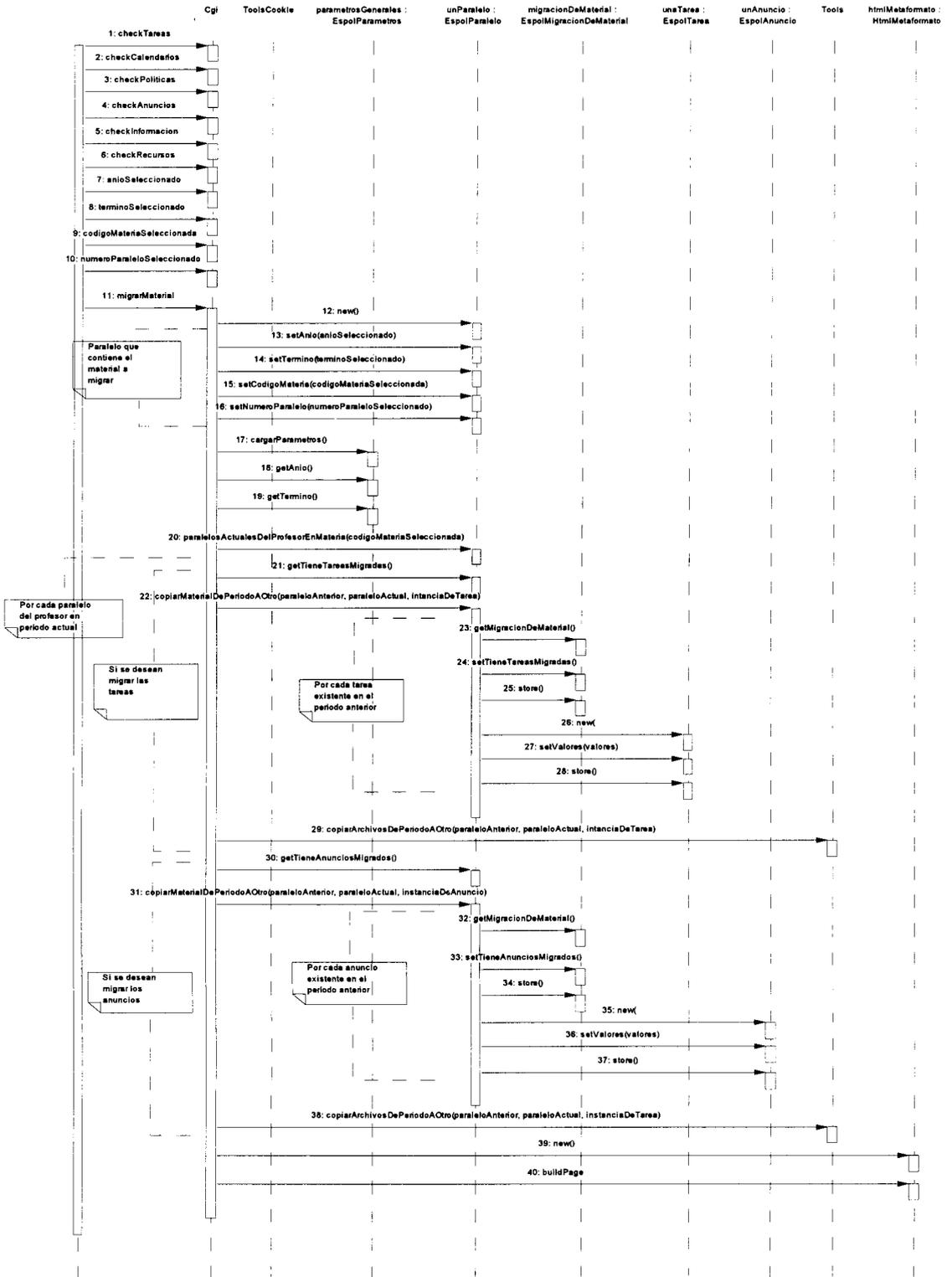


Ingreso de calificaciones.

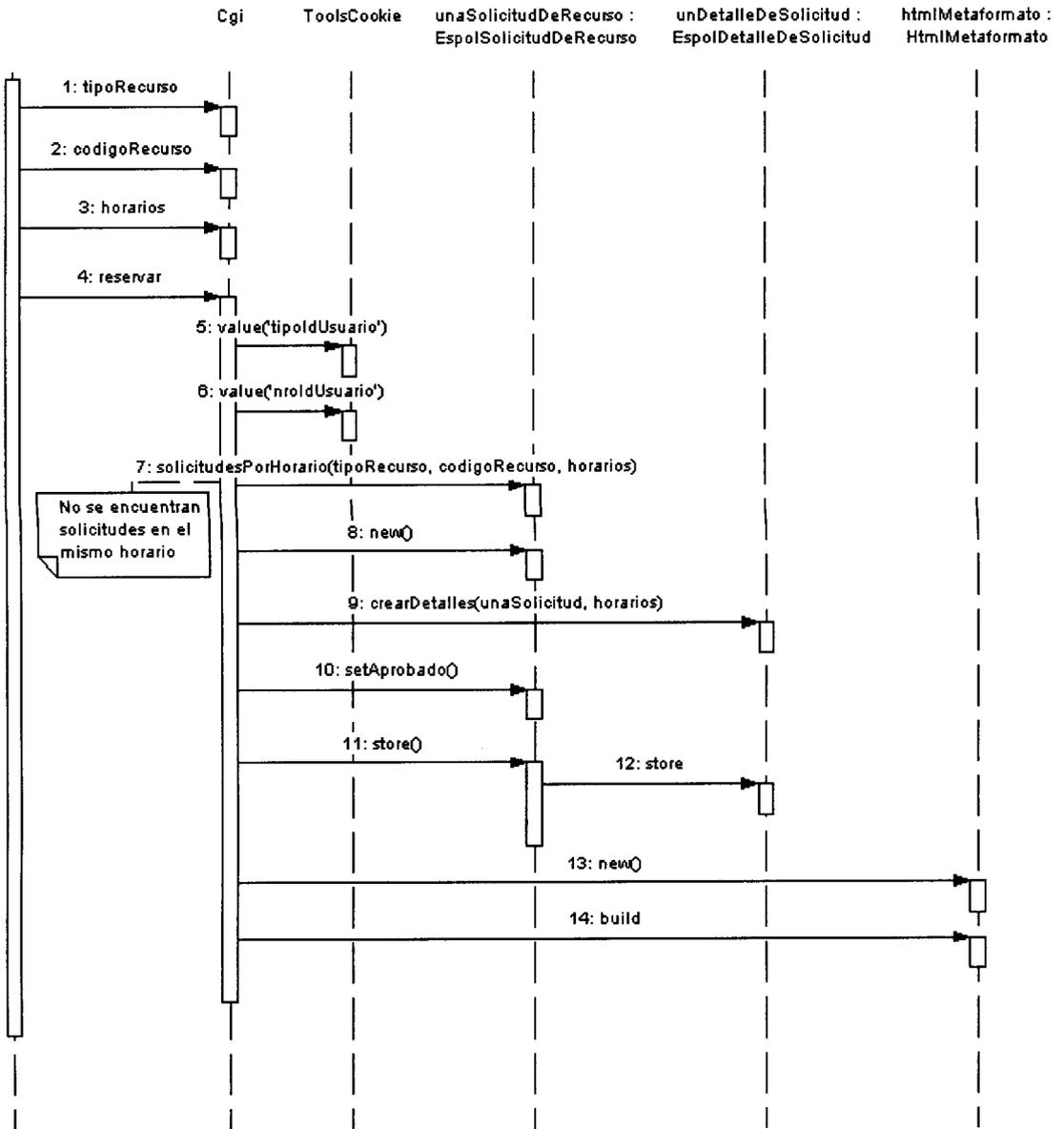
Cgi ToolsCookie parametros : Espo unParalelo : unaPolitica : EspolunaCalificacion : EspunAlumnoPorParalelo : iMetaforma ParametrosGenera EspolParaleloPoliticaDeCalificacionDeAlumEspolAlumnoPorParalelo iMetaform.



Migración del material del profesor.



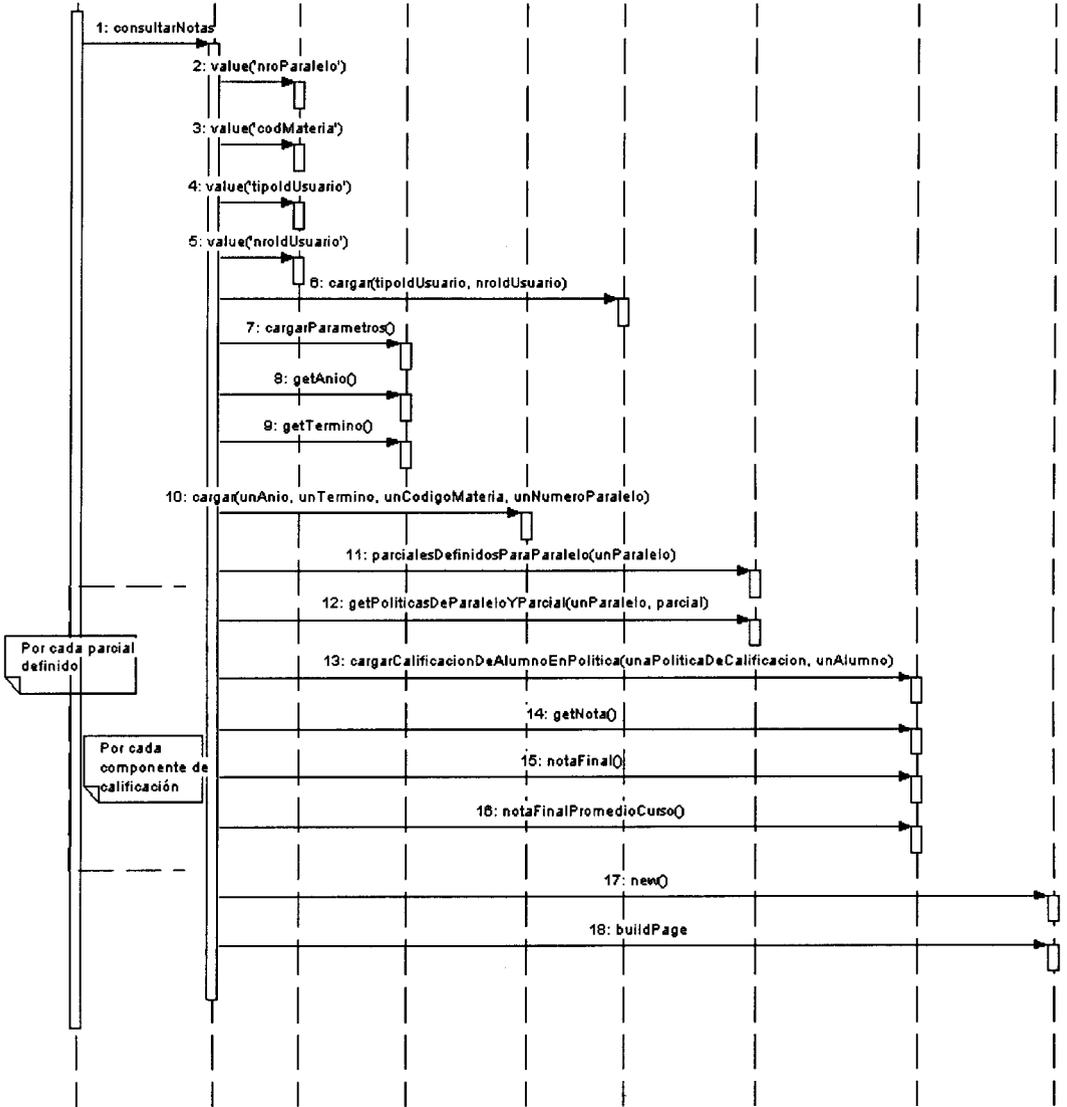
Reservación de recurso físico.



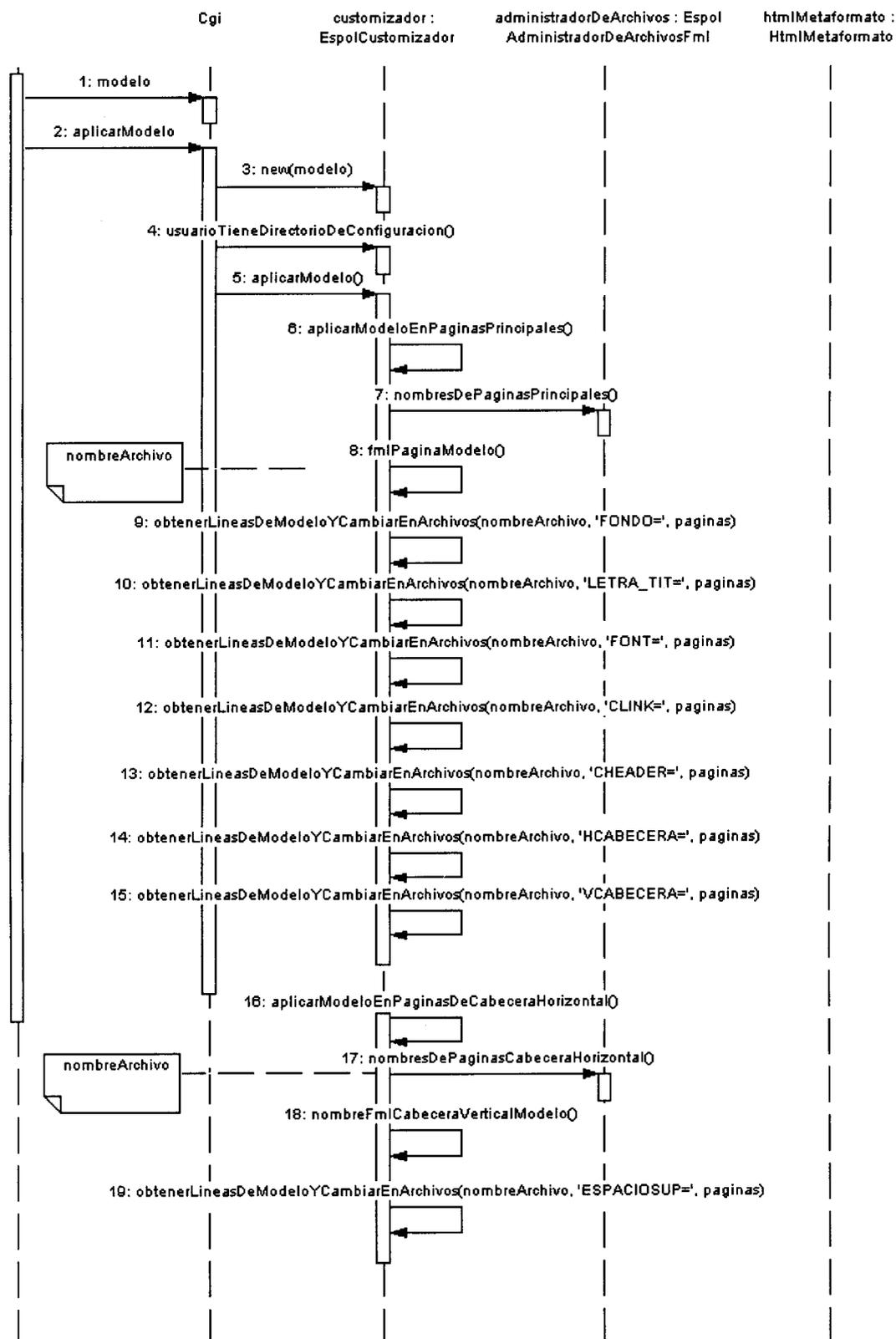
No se encuentran solicitudes en el mismo horario

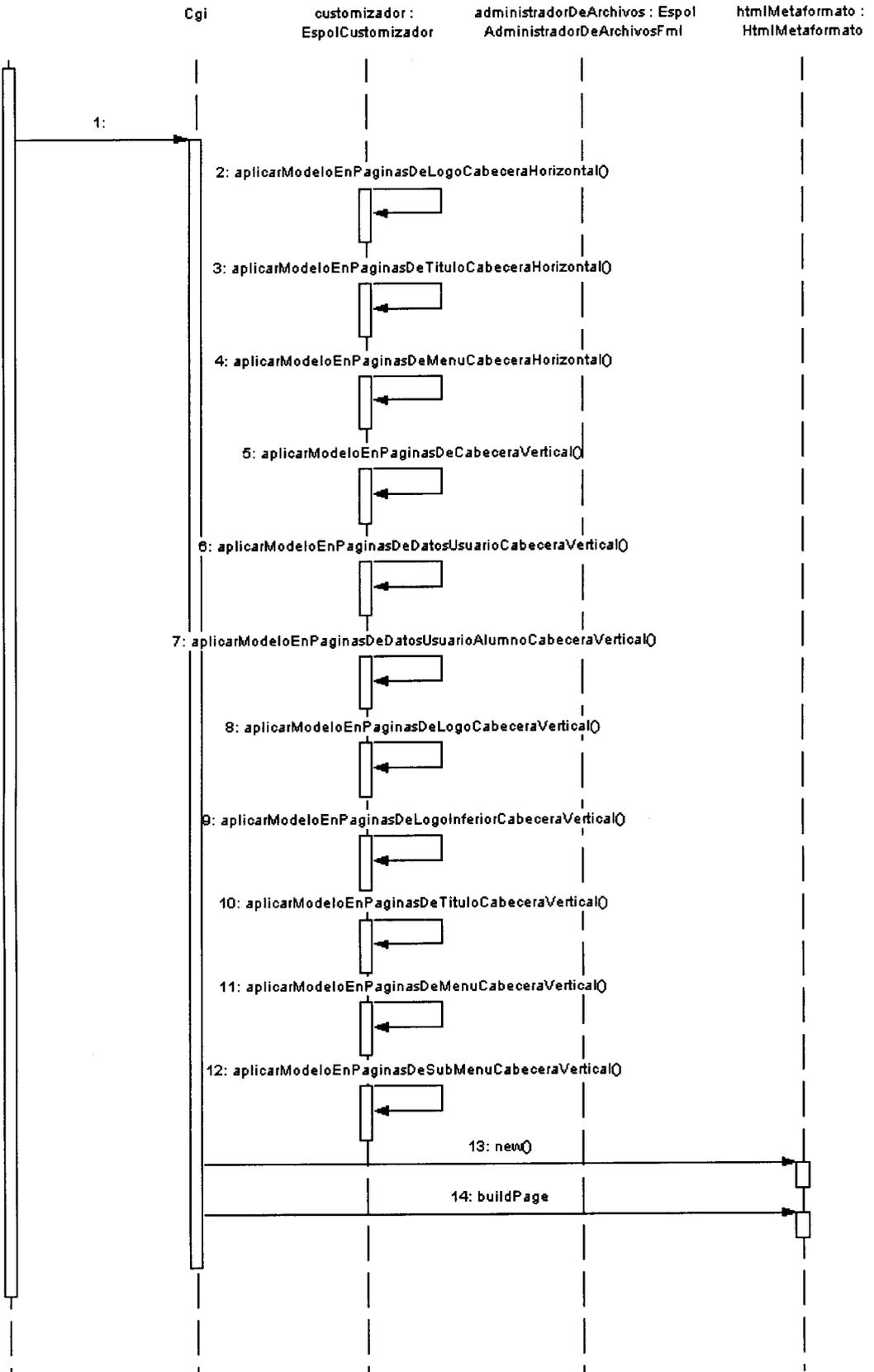
Consulta de calificaciones.

Cgl ToolsCookie parametros : Espol unParalelo : unAlumno : unaPoliticaDeCalificacio unaCalificacion : EsphtmlMetaformat ParametrosGeneraleEspolParalelo EspolAlumnoEspolPoliticaDeCalificaciCalificacionDeAlumno+HtmlMetaforma

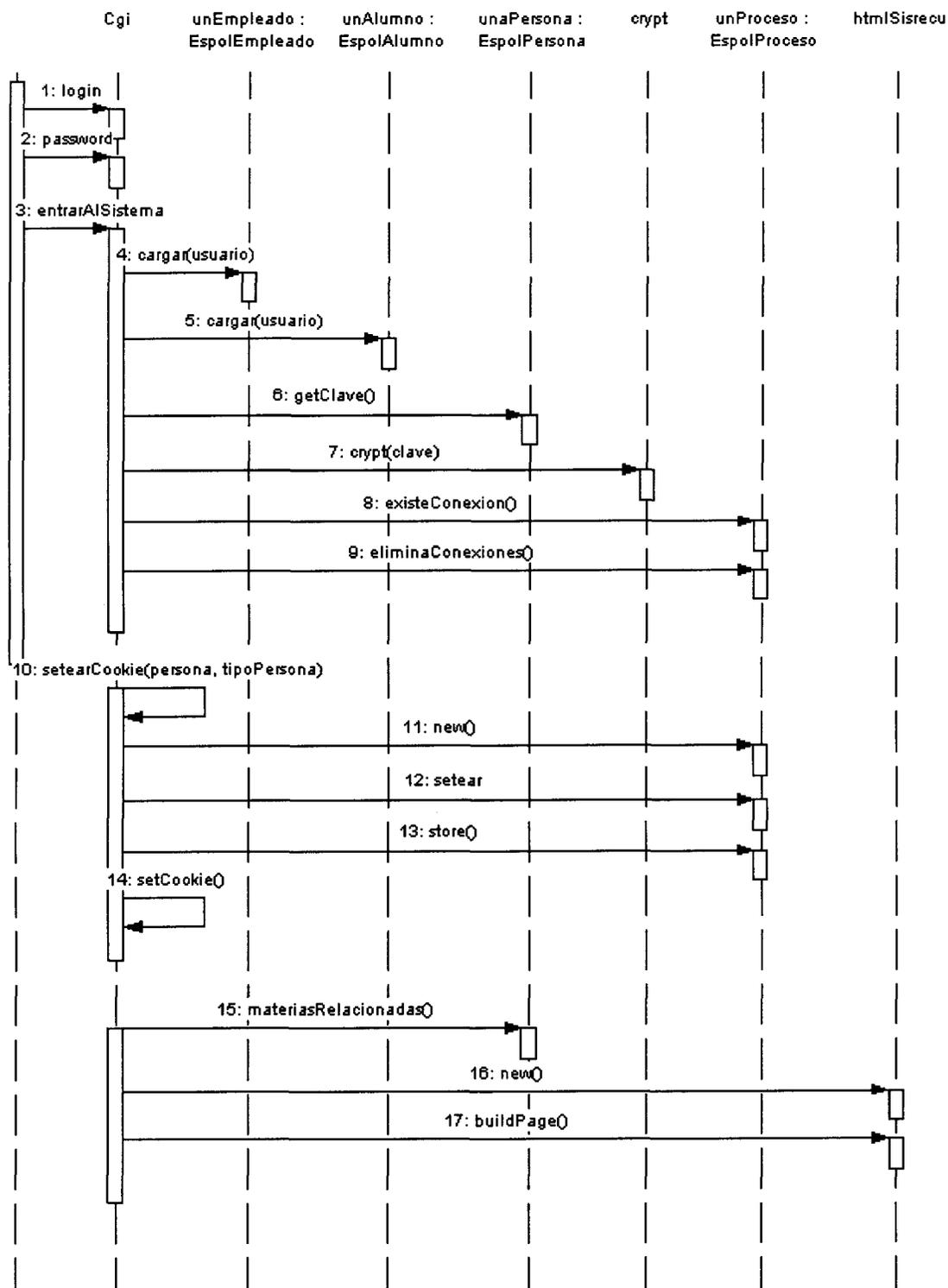


Customización de páginas HTML.

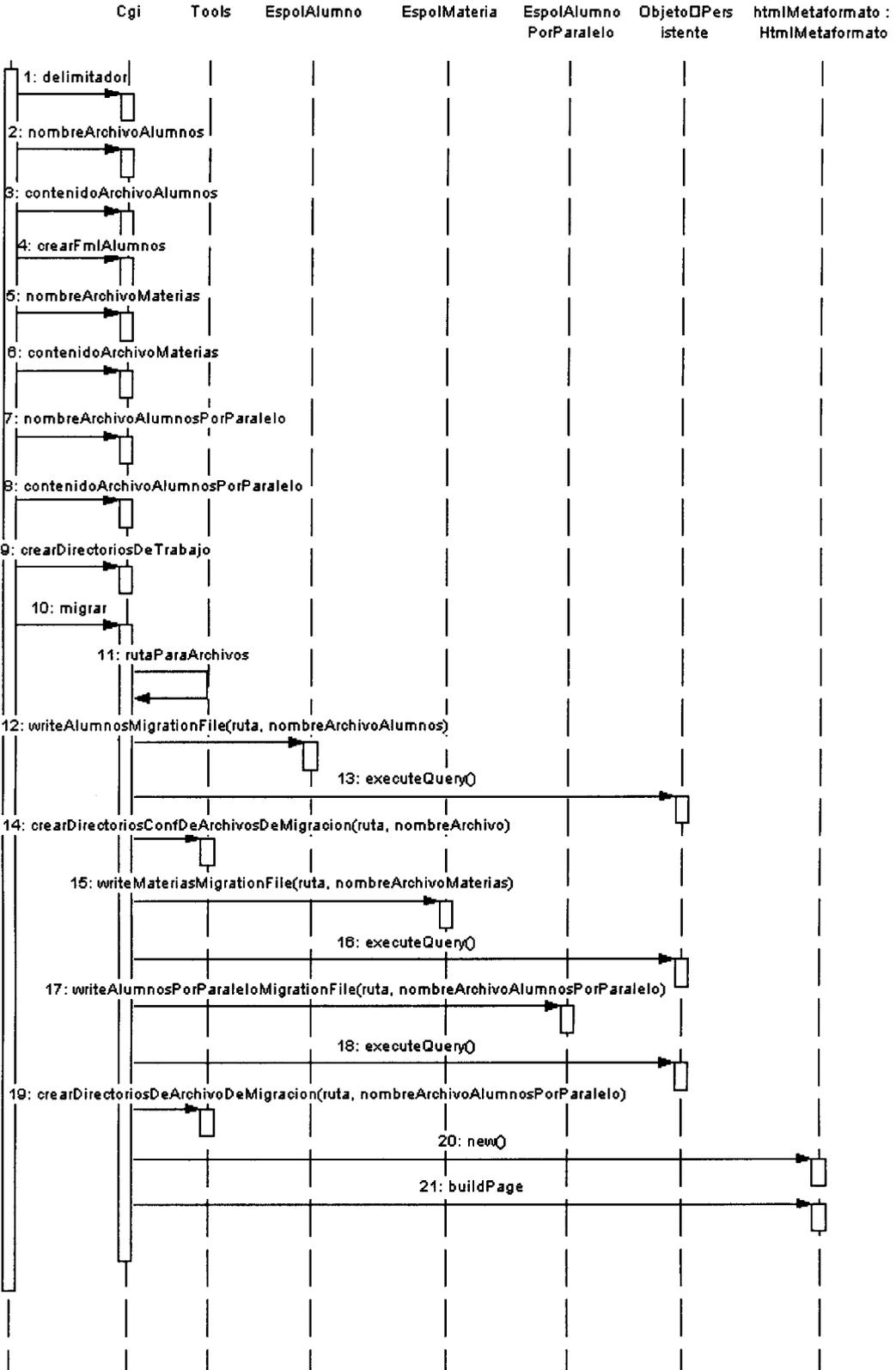




Acceso al sistema.



Migración de datos.



4.8. Diseño del procedimiento de migración de datos

En vista de que Cursos Web es un sistema de apoyo para la educación necesita datos de cada período que son administrados por el Sistema Académico de la Institución, tales como: unidades académicas, estudiantes, profesores, materias, paralelos y estudiantes registrados por paralelo.

En este sistema, la información mencionada anteriormente está distribuida en las tablas: UNIDAD_ACADEMICA, ALUMNO, EMPLEADO, MATERIA, PARALELO, PARALELO_ALUMNO. El detalle de cada una de estas tablas se encuentra en la sección 4.5.

Puesto que estos datos deben obtenerse de una base de datos externa, para facilitar su ingreso a este sistema se ha definido un procedimiento de migración a través del cual, el Administrador del Sistema transfiere los datos de un medio magnético (diskette) a la base de datos de Cursos Web.

El procedimiento de migración de la información es como se describe a continuación:

1. Solicitar al Centro de Servicios Computacionales (CESERCOMP) la información que se necesita.

Para facilitar esta tarea al Dpto. de Sistemas, se han desarrollado scripts que permiten obtener estos datos para cada una de las tablas mencionadas anteriormente generando archivos texto. Dichos scripts ejecutan sentencias SQL en la base del sistema académico. El administrador de Cursos Web podrá encontrar estos archivos en el directorio ***/home/httpd/cgi-bin/academic/base_datos*** del servidor del sistema. Los nombres de los scripts y su descripción se presentan en la siguiente tabla:

Script	Descripción
exportUnidadesAcademicas.sql	Genera el archivo texto unidad_academica.txt que contiene las unidades académicas de la Institución.
exportAlumnos.sql	Genera el archivo texto alumno.txt que contiene la información personal de los estudiantes registrados en el período vigente en la FIEC.
exportEmpleados.sql	Genera el archivo texto empleado.txt que contiene la información personal de los profesores que tienen asignado algún paralelo en el período vigente en la FIEC.
exportMaterias.sql	Genera el archivo texto materia.txt que contiene las materias dictadas en el período vigente en la FIEC.
exportParalelos.sql	Genera el archivo texto paralelo.txt que contiene los cursos abiertos en el período vigente en la FIEC.
exportParaleloAlumnos.sql	Genera el archivo texto paralelo_alumno.txt que contiene los alumnos registrados en cada uno de los cursos abiertos en el período vigente en la FIEC.

Tabla VI Scripts de exportación de datos

2. Cambiar el formato de los archivos texto generados.

Una vez que CESERCOMP entrega los archivos solicitados, el Administrador debe modificar estos archivos, eliminando de su contenido las comillas dobles (") existentes. Esto permitirá que tales archivos puedan ser utilizados en el proceso de migración.

3. Ejecutar opción de migración.

En el módulo de Administración de Cursos Web, el Administrador debe ejecutar la opción **Migración >> Datos**, en donde se especifican los nombres de los archivos a procesar. Puede definir uno o varios archivos a la vez. Si los archivos son demasiado grandes, es recomendable distribuirlos en varios a archivos para procesarlos uno a uno.

Además de especificar los nombres de archivos, el administrador puede definir si se deben crear los directorios de trabajo de estudiantes, profesores, paralelo y alumnos por paralelo.

Cuando se ejecuta esta opción, sólo se procesa aquella información que no genera registros duplicados en el sistema.

La opción de generación de directorios de trabajo, puede o no ejecutarse dentro del proceso de migración. Pero, cabe recalcar, que los usuarios no podrán usar el sistema sino tienen sus directorios de trabajo creados.

4.9. Plan de pruebas

El plan de pruebas de aceptación comprende un conjunto de pruebas que validan al sistema comparando los resultados con la salida esperada. Cada prueba muestra como el sistema satisface algunos de los requerimientos descritos por los casos de uso y/o los requerimientos no funcionales. Deben ser cerradas; esto es, garantizadas de completarse (satisfactoriamente o no) dentro de un período de tiempo finito.

Existen pruebas con granularidad fina o granularidad gruesa, dependiendo del nivel de detalle de las pruebas. En las pruebas con granularidad fina, se definen por lo menos una prueba por cada escenario de un caso de uso. Mientras que, en las pruebas con granularidad gruesa, cada prueba cubre la funcionalidad de muchos casos de uso, aplicables a la funcionalidad definida en los casos de uso involucrados.

Prueba No. 1: Profesor accede al sistema y publica un anuncio**Pre-requisitos:**

- Guido, profesor de la materia “Interacción Hombre-Máquina”, paralelo 001, desea publicar un anuncio, adjuntando un archivo.
- El tiene asignado un usuario y contraseña en el servidor local.

Instrucciones de configuración:

- El curso tiene 20 estudiantes, con toda la información pertinente.
- El archivo tiene 200 K. y el máximo permitido para él es de 1 M.
- El archivo se llama importante.doc y lo tiene en su disco duro, en la ruta C:\documentos.
- Es la primera vez que Guido va a publicar ese archivo.

Instrucciones de la prueba:

- El profesor ingresa al sistema, especificando su nombre de usuario y contraseña.
- El sistema le presenta la lista de materias que él está dictando en el período académico actual.

- El profesor da click sobre la materia “Interacción Hombre-Máquina”.
- El sistema presenta la página con la lista de anuncios publicados.
- El profesor selecciona la opción de ingreso.
- El profesor ingresa la información del nuevo anuncio, seleccionando que el anuncio será de tipo archivo.
- En el nombre de archivo, el profesor especifica C:\documentos\importante.doc. También marca la opción de envío de mail.
- Da click sobre el botón aceptar.

Comportamiento aceptable:

- El sistema garantiza el acceso del profesor, verificando su nombre de usuario y contraseña.
- El sistema verifica que el archivo importante.doc existe en la ruta especificada y que no ha sido publicado anteriormente.
- El sistema registra el nuevo anuncio en la base de datos.
- El sistema envía un mail indicando que existe un nuevo anuncio a todos los integrantes del curso de Guido.

- El sistema presenta una página indicando que la operación fue realizada con éxito.
- Cuando el profesor vuelva a visitar la página de anuncios, encontrará también el nuevo anuncio ingresado.

Prueba No. 2: *Estudiante accede al sistema y responde una tarea publicada*

Pre-requisitos:

- Mercedes, alumna de la materia “Interacción Hombre-Máquina”, paralelo 001, desea responder a una tarea publicada.
- Ella tiene asignado un usuario y contraseña en el servidor goliath.

Instrucciones de configuración:

- El curso tiene 20 estudiantes, con toda la información pertinente.
- La tarea está siendo entregada dentro del plazo especificado.

Instrucciones de la prueba:

- La estudiante ingresa al sistema, especificando su nombre de usuario y contraseña.
- El sistema le presenta la lista de materias en las que ella se encuentra registrada.
- La estudiante da click sobre la materia “Interacción Hombre-Máquina”.
- La estudiante accede a la página de tareas, ingresando a la sección “Clases” y dando click en la opción “Tareas”.
- El sistema presenta la página con la lista de tareas publicadas por el profesor.
- La estudiante da click sobre el ícono en forma de hoja de la columna “Responder” correspondiente a la tarea que se desea responder.
- La estudiante ingresa el contenido de la respuesta de tarea.
- Da click sobre el botón aceptar.

Comportamiento aceptable:

- El sistema garantiza el acceso de la estudiante, verificando su nombre de usuario y contraseña en el servidor de autenticación goliath.
- El sistema registra la respuesta de tarea en la base de datos, con un estado de entrega a tiempo.
- El sistema presenta una página indicando que la operación fue realizada con éxito.
- Cuando el profesor vuelva a visitar la página de tareas, encontrará que el número de respuestas ha incrementado y podrá revisarla.

CAPÍTULO 5

5. IMPLEMENTACIÓN Y PRUEBAS

En este capítulo describiremos detalles de implementación del sistema, tales como estándares establecidos (en codificación, funcionalidad y apariencia de las páginas) y los requerimientos necesarios para que el sistema pueda ser utilizado, tanto a nivel de software como de hardware.

Después de la implementación de un sistema, es importante realizar la etapa de pruebas para dar por terminado el desarrollo del proyecto. El resultado de las pruebas a las que Cursos Web fue se incluyen en este capítulo.

5.1. Estándares de implementación del sistema

Previo a la implementación de los sistemas fueron definidos estándares que nos ayudaron a desarrollar la aplicación de una manera organizada, lo cual permitirá que el mantenimiento no presente mayores inconvenientes.

Los estándares definidos son los siguientes:

- Los nombres de clases comienzan con letras mayúsculas y se definen en singular.
- Todas las clases del dominio del problema tienen el prefijo *Espol*. Por ejemplo: *EspolMateria*.
- Las variables de instancia y métodos comienzan con letras minúsculas. Si es un nombre compuesto la primera letra de cada palabra se define en mayúsculas, excepto la primera.
- Todas las variables de instancia cuentan con métodos para recuperar sus valores (getter) y asignar sus valores (setter). Estos métodos tienen el mismo nombre de la variable anteponiendo la palabra *get* y *set* respectivamente. Por ejemplo: *getCodigo* y *setCodigo* (para la variable *codigo*).
- No se han utilizado abreviaturas para nombrar clases, métodos o variables.
- Los CGIs están organizados dependiendo del módulo en el que son utilizados (ver sección 4.4).
- El código de la aplicación se encuentra debidamente documentado

- El contenido de los CGIs está compuesto por la definición de las librerías necesarias y dos funciones llamadas: `errorPage()` y `principal()`. Las dos invocan al método de construcción de la página, pero la primera con un mensaje de error y la segunda con el contenido normal de la página.
- Los JavaScripts utilizados en cada página son codificados en archivos con el mismo nombre del CGI con la extensión `jsv` (estos se encuentran en el directorio `/home/httpd/cgi-bin/academic/scripts`). Cuando el metaformato construye cada página incluye este código.
- En cuando a la apariencia de las páginas se definió que en las tablas de consulta y de resultado de alguna operación se utilicen tonos celestes. Mientras que en las tablas de listas principales se utilicen tonos naranja. Estos también pueden ser configurados desde los archivos de formato.
- Todas las tablas de listas principales poseen la funcionalidad de *Seleccionar todos*, *Quitar selección* y *Eliminar seleccionados*.

5.2. Instalación del sistema en el servidor de la FIEC

Para realizar las pruebas piloto se instaló el sistema en uno de los servidores de la FIEC, llamado *palma*. Dicho servidor necesito tener incluidas las siguientes características:

Hardware:

- Procesador Intel Pentium 100 MHz.
- 16 MB de memoria RAM.
- 500 MB de espacio libre en disco duro.
- Una tarjeta de red ISA o PCI compatible con Linux (utilizamos una RealTek PCI compatible con NE2000).

Software:

- El sistema operativo RedHat Linux, v5.1.
- El servidor Web Apache-SSL v1.3.9.
- SSLeay v0.9.0b4 (una implementación del protocolo SSL) para Linux.
- Python 1.5.2, con toda su biblioteca estándar de módulos.

- Un servidor POP3.
- El manejador de base de datos PostgreSQL v6.5.2.
- Librerías de interfaces entre Python y PostgreSQL.
- Los programas CGIs de Cursos Web.

Tareas adicionales que fueron realizadas:

1. Se asignaron permisos de lectura y ejecución a todos los directorios dentro de la ruta **/home/httpd/cgi-bin/academic**.
2. Se asignaron permisos de lectura, escritura y ejecución a todos los directorios dentro de la ruta **/home/httpd/html/academic**.
3. En los archivos de configuración se realizaron los siguientes cambios:
4. En el archivo **/etc/httpd/conf/srm.conf** se agregó lo siguiente:

```
SetEnv PYTHONPATH /home/httpd/cgi-bin/academic/lib:
```

```
/usr/lib/python1.5:/usr/lib/python1.5/plat-linux-i386
```

Estas definen los directorios en los que se encuentran las librerías de python y las preferencias que tendrán los directorios para realizar las búsquedas de las librerías.

5. Creación de la base de datos "**academic**".
6. Migración de los datos necesarios (ver la sección 5.3).

5.3. Migración de datos de prueba

Posterior a la instalación del sistema, en el servidor de la FIEC, se procedió con la migración de datos de prueba, los mismos que fueron obtenidos del Sistema Académico de la ESPOL.

Los datos que fueron tomados de muestra correspondieron a los de la Facultad de Ingeniería en Electricidad y Computación del período 2000-2, considerando que ésta sería la unidad piloto en las pruebas del sistema.

Para la migración de estos datos de prueba, seguimos el procedimiento definido en la sección 4.9.

5.4. Pruebas y resultados de las pruebas

Una vez que el sistema estuvo debidamente instalado, tanto profesores como alumnos de la FIEC empezaron a utilizarlo desde el período académico 2000-2. Tiempo desde el cual se reportaron ciertos

problemas que pudieron ser solucionados de manera inmediata. Así también, los usuarios del sistema pudieron dar sus opiniones y sugerencias para realizar mejoras en la funcionalidad y apariencia del sistema. Estas mejoras incluyeron modificaciones y creaciones de nuevas opciones que para ellos serían de mucha utilidad, tales como: un foro de discusión, publicación de recurso, envío automático de mails al publicar anuncios, tareas o recursos, etc.

Podemos concluir que las pruebas resultaron satisfactorias, puesto que el sistema ha sido utilizado en cada uno de los períodos académicos siguientes hasta el presente y cada vez es usado por un número mayor de usuarios. Lo que demuestra que Cursos Web es considerado como una herramienta de apoyo a la educación, cumpliendo así con el objetivo de este proyecto.

5.5. Problemas de Implementación

En vista de que las herramientas de desarrollo utilizadas en este proyecto eran totalmente desconocidas para nosotras, tuvimos ciertos inconvenientes, los mismos que pudieron ser solucionados a medida que fuimos adquiriendo el conocimiento necesario.

Adicionalmente a los problemas que tuvimos durante el desarrollo, se presentaron otros inconvenientes tales como:

Dificultades en migracion de datos

- El proceso de migración de los datos del Sistema Académico se realiza después de la semana de registros, para que profesores y estudiantes puedan empezar a usar Cursos Web. El problema se presenta después de registros, cuando los estudiantes hacen ampliaciones o eliminaciones de materias, puesto que esto produce que la información anteriormente migrada no esté consistente. Para solucionar este problema se deben realizar varias migraciones durante las primeras semanas del período académico.

Problemas administrativos

- El Sistema Académico no cuenta con la dirección de correo electrónico de profesores y estudiantes, por lo que éste tipo de información no se puede migrar. Al no existir esta información en Cursos Web, los usuarios no tienen acceso al sistema, teniendo que acudir al Administrador para que les sea asignado un nombre de usuario y servidor de autenticación en sus datos personales.

Problemas de usabilidad

- Las opciones del menú principal presentadas a un profesor eran nombradas de manera diferente a las opciones presentadas a un estudiante. Esto causaba confusión en el momento en el que un estudiante hacía referencia al profesor sobre alguna opción.

CONCLUSIONES Y RECOMENDACIONES

1. El producto final de este proyecto, Cursos Web, es una aplicación robusta, eficiente y fácil de usar. Es considerada como una herramienta de apoyo a la educación y es un medio de comunicación entre los integrantes de un curso durante los períodos académicos de la Institución. Esto se ha podido comprobar debido a que el sistema ha estado en producción desde el período académico 2000-2 en la Facultad de Ingeniería en Electricidad y Computación, cumpliendo con todos los objetivos propuestos y siendo usado en cada período por un número mayor de usuarios.
2. Puesto que Cursos Web es una aplicación utilizada a través de la Internet, tanto profesores como alumnos podrán disponer de este servicio desde cualquier lugar y en cualquier momento. Evitando de esta manera el desperdicio de tiempo y esfuerzo de profesores y alumnos al tener que trasladarse a la Universidad para mantenerse en comunicación aún fuera de las horas de clase.
3. Como complemento al apoyo académico que brinda Cursos Web, también se provee de un medio que permite agilizar los procedimientos

necesarios para la asignación de los recursos académicos físicos (proyectors, datashows, laboratorios, etc.) a los usuarios que los requieran. Con esto se logra que los recursos sean administrados de manera eficiente y automática mediante un control de solicitudes y asignaciones, minimizando el tiempo necesario para la realización del requerimiento y aprobación del mismo. Puesto que los recursos están disponibles para ser utilizados en horas de clase, se decidió integrar este componente, para brindar a los profesores y estudiantes la facilidad de que los soliciten desde el sistema.

4. El desarrollo de la aplicación está basado en tecnología orientada a objetos, lo que permitirá su integración en el futuro con otros sistemas o módulos, al poder reutilizar las clases con las que cuenta Cursos Web.
5. Cursos Web es una alternativa de bajo costo, ya que las herramientas necesarias para su operación y aquellas utilizadas en el desarrollo están disponibles en la Internet. Linux y Apache son considerados software de alta confiabilidad y además presentan el beneficio de una buena escalabilidad. El sistema completo podría ser migrado fácilmente a otra plataforma de tipo Unix como Solaris o HP ya que tanto el servidor de Web (Apache), como la base de datos (Postgres) y el intérprete de los programas (Python) existen bajo estas plataformas.
6. Es un producto altamente aplicable al entorno educativo ecuatoriano, puesto que, además de su bajo costo, es una herramienta que puede ser

fácilmente adaptable para ser utilizada en colegios, institutos y demás universidades del País.

7. El realizar proyectos de tesis entre estudiantes de diferentes carreras, como lo son Ingeniería en Computación y Licenciatura en Sistemas de Información, es de mucha utilidad puesto que existe una combinación de conocimientos y esfuerzos que generan proyectos muy sólidos y altamente funcionales.
8. Una recomendación para versiones futuras de este proyecto es la de incluir en él, información estadística sobre distintos aspectos involucrados en un curso, de interés para profesores y alumnos. Así, por ejemplo, un profesor podría visualizar de manera gráfica, cuál es el tipo de participación de los estudiantes en el foro de discusión o cuál es el aprovechamiento que sus estudiantes. Los estudiantes podrían observar cuál ha sido su rendimiento académico con relación a sus compañeros de clase.
9. A pesar de que el sistema permite a los usuarios personalizar la apariencia de las páginas en base a modelos predefinidos, se podría incluir una interfaz para que dicha personalización sea más específica. De tal manera que un usuario pueda aplicar cambios sobre cada uno de los componentes que se encuentran en las páginas Web (fondo de página, logos e íconos, colores, tipos y tamaños de letra, etc.).

10. Aunque Cursos Web cuenta con un foro de discusión, sería de mucha utilidad contar con un servicio de chat, para que los integrantes de un curso puedan intercambiar ideas de una manera más directa. Con esto, se lograría que el sistema brinde una facilidad de interacción más dinámica y en tiempo real para la comunicación de mensajes cortos.

11. La autenticación que posee el sistema (cuando esta no es local) se realiza únicamente en base al protocolo POP3. Se recomienda que en versiones futuras se incluyan otros protocolos de autenticación que permita a los usuarios poder usar sus cuentas definidas en servidores que manejen otros tipos de protocolos.

APÉNDICES

APÉNDICE A

DEFINICIÓN DE CLASES

Nombre: AdministradorDeArchivosFml

Descripción:

Conoce sobre todos los archivos fml necesarios para customizar la apariencia del sistema.

Estados:

(Ninguno)

Relaciones:

Superclase: (Ninguna)

Subclases: (Ninguna)

Asociaciones: (Ninguna)

Variables globales:

instAPI
Instancia de HtmlAPI.

Variables de instancia:

nombresPaginasPrincipales
Arreglo con los nombres de los archivos fml principales, excepto los nombres de los archivos fml de las páginas modelo.

Métodos de clase:

(Ninguno)

Métodos de instancia:

nombresDePaginasPrincipales()
Retorna el contenido de la variable nombresDePaginasPrincipales. Si ésta es nulo, analiza el archivo tuplas.conf (que asocia nombres de CGIs con FMLs) para obtener los nombres de los fml principales y asignarlos en la variable de instancia.

nombresDePaginasCabeceraHorizontal()
Retorna un arreglo con los nombres de las paginas fml de cabecera horizontal, analizando cada uno de los archivos fml de páginas principales.

- `nombresDePaginasLogoCabeceraHorizontal()`
Retorna un arreglo con los nombres de las paginas fml de logo de cabecera horizontal, analizando los nombre de páginas de cabecera horizontal.
- `nombresDePaginasTituloCabeceraHorizontal()`
Retorna un arreglo con los nombres de las paginas fml de titulo de cabecera horizontal, analizando los nombres de páginas de cabecera horizontal
- `nombresDePaginasMenuCabeceraHorizontal()`
Retorna un arreglo con los nombres de las paginas fml de menú de cabecera horizontal, analizando los nombres de páginas de cabecera horizontal.
- `nombresDePaginasCabeceraVertical()`
Retorna un arreglo con los nombres de las paginas fml de cabecera vertical, analizando cada uno de los archivos fml de páginas principales.
- `nombresDePaginasLogoCabeceraVertical()`
Retorna un arreglo con los nombres de las paginas fml de logo de cabecera vertical, analizando los nombres de páginas de cabecera vertical.
- `nombresDePaginasLogoInferiorCabeceraVertical()`
Retorna un arreglo con los nombres de las paginas fml de logo inferior de cabecera vertical, analizando los nombres de páginas de cabecera vertical.
- `nombresDePaginasDatosUsuarioCabeceraVertical()`
Retorna un arreglo con los nombres de las paginas fml de datos del usuario de cabecera vertical, analizando los nombres de páginas de cabecera vertical.
- `nombresDePaginasDatosUsuarioAlumnoCabeceraVertical()`
Retorna un arreglo con los nombres de las paginas fml de datos del usuario alumno de cabecera vertical, analizando los nombres de páginas de cabecera vertical.
- `nombresDePaginasTituloCabeceraVertical()`
Retorna un arreglo con los nombres de las paginas fml de titulo de cabecera vertical, analizando los nombres de páginas de cabecera vertical.
- `nombresDePaginasMenuCabeceraVertical()`
Retorna un arreglo con los nombres de las paginas fml de menú de cabecera vertical, analizando los nombres de páginas de cabecera vertical.
- `nombresDePaginasSubMenuCabeceraVertical()`
Retorna un arreglo con los nombres de las paginas fml de menú de cabecera vertical, que sólo son referenciados desde los CGIs.

Notas:

Nombre: GeneradorDeArbolParaForo

Descripción:

Tiene como función el poder construir la estructura del foro de discusión.

Estados:

(Ninguno)

Relaciones:

Superclase: (Ninguna)

Subclases: (Ninguna)

Asociaciones: (Ninguna)

Variables globales:

(Ninguna)

Variables de instancia:

(Ninguna)

Métodos de clase:

construyeFila(mensaje, instMF, nivel, color, tipo, escogido, idPropietario, icoArbol)
 Muestra información relacionada a un mensaje y determina si tiene respuestas asociadas.
 mensaje -> Instancia de EspoTemaDiscusion.
 instMF -> Instancia de HtmlMetaformato.
 color -> Color con el que debe visualizarse la fila.
 tipo -> Identifica si el usuario es un profesor o un alumno (1 - Profesor, 2 - Alumno).
 escogido -> Indica si el mensaje está en modo de consulta.
 idPropietario -> Utilizado para saber si el alumno conectado puede o no eliminar el mensaje.
 icoArbol -> Nombre del ícono utilizado para desplazar o contraer las respuestas del mensaje.

colorFila(instMF, indice)

Retorna un arreglo con color que debe ser utilizado para presentar el mensaje en el índice recibido como parámetro; y, el índice a utilizar en la siguiente línea de mensaje que deba ser presentado.
 instMF -> Instancia de HtmlMetaformato.
 indice -> Indica el tono de color que debe usarse (1 – claro, 2 - oscuro).

construyeArbol(mensaje, instMF, tipo, idPropietario, modoArbol, paralelo)

Retorna el código HTML que construye el arbol de mensajes.

mensaje -> Instancia de EspoTemaDiscusion.

instMF -> Instancia de HtmlMetaformato.

tipo -> Identifica si el usuario es un profesor o un alumno (1 - Profesor, 2 - Alumno).

idPropietario -> Utilizado para saber si el alumno conectado puede o no eliminar el mensaje.

modoArbol -> Tipo de ícono que debe mostrarse para desplazar o contraer las respuestas del mensaje (1 – más, 2 – menos).

paralelo -> Instancia de EspoParalelo.

Métodos de instancia:
(Ninguno)

Notas:
Esta es una clase abstracta.

Nombre: BotonDeModeloEnFml

Descripción:
Mantiene la referencia de los nombres de los prefijos de los botones y etiquetas que serán usados en cada uno de los modelos.

Estados:
(Ninguno)

Relaciones:
Superclase: ObjetoPersistente
Subclases: (Ninguna)
Asociaciones: (Ninguna)

Variables globales:
(Ninguna)

Variables de instancia:
codigo
Identifica el modelo al que pertenece la instancia.

nombreFml
Nombre del archivo fml en el que se hace referencia a este prefijo de botón o etiqueta.

prefijoBoton
Prefijo de los botones o nombre de archivo de etiqueta que deberá ser especificado por este modelo y fml definidos.

Métodos de clase:
(Ninguno)

Métodos de instancia:
__init__()
Inicializa las variables de instancia del objeto en el momento en el que este es creado.

getCodigo()
Retorna el valor del atributo codigo.

setCodigo(unString)
Setea un valor al atributo codigo.

getNombreFml()
Retorna el valor del atributo nombreFml.
setNombreFml(unString)

Setea un valor al atributo nombreFml.

getPrefijoBoton()
Retorna el valor del atributo prefijoBoton.

setPrefijoBoton(unString)
Setea un valor al atributo prefijoBoton.

getNombreEtiqueta()
Retorna el valor del atributo nombreEtiqueta.

setNombreEtiqueta(unString)
Setea un valor al atributo nombreEtiqueta.

Notas:

Nombre: GeneradorDeCalendario

Descripción:
Esta clase se encarga de formar un mes calendario.

Estados:
(Ninguno)

Relaciones:
Superclase: (Ninguna)
Subclases: (Ninguna)
Asociaciones: (Ninguna)

Variables globales:
(Ninguna)

Variables de instancia:
(Ninguna)

Métodos de clase:
parse_events(año, mes, eventos)
Retorna arreglos de títulos y descripciones de los eventos para un año y mes específicos ordenados por día. También retorna el número de días del mes y un arreglo con los números de días hábiles en el mes (de acuerdo al cual estarán ordenadas las otras colecciones).
año -> Año calendario.
mes -> Mes calendario.
eventos -> Rows de días de calendario.

gen_cal(año, mes, títulos, descripciones, maxdd, díasHábiles, instMF, paralelo, pathWeb, tipo)
Construye el código HTML de la estructura del calendario.
año -> Año calendario.
mes -> Mes calendario.
títulos -> Arreglo de los títulos de los eventos de los días de calendario en un año y mes.

descripciones -> Arreglo de las descripciones de los eventos de los días de calendario en un año y mes.
 maxdd -> Número de días del mes.
 diasHabiles -> Arreglo con los números de días hábiles en el mes.
 instMF -> Instancia de HtmlMetaformato.
 paralelo -> Instancia de EspolParalelo.
 pathWeb -> Ruta del directorio en donde se encuentran los íconos.
 tipo -> Identifica si el usuario es un profesor o un alumno (1 - Profesor, 2 - Alumno).

Métodos de instancia:

(Ninguno)

Notas:

Esta clase es abstracta.

Nombre: ConexionABase**Descripción:**

Representa una conexión a la base y se encarga de conectarse o desconectarse de la misma.

Estados:

(Ninguno)

Relaciones:

Superclase: (Ninguna)
 Subclases: (Ninguna)
 Asociaciones: (Ninguna)

Variables globales:

(Ninguna)

Variables de instancia:

__base
 Mantiene la conexión a la base de datos.

Métodos de clase:

(Ninguno)

Métodos de instancia:

databaseName()
 Devuelve el nombre de la base de datos.

connectToDatabase()
 Abre una conexión a la base de datos.

database()
 Devuelve el valor del atributo __base.

close()
Cierra la conexión a la base de datos.

Notas:

Nombre: Customizador

Descripción:

Realiza los cambios necesarios en todos los archivos fml (de formato) de las páginas web de un usuario, en base a los archivos fml definidos para el modelo pre-definido seleccionado por el usuario.

Estados:

(Ninguno)

Relaciones:

Superclase: (Ninguna)
Subclases: (Ninguna)
Asociaciones: (Ninguna)

Variables globales:

administradorDeArchivosFml
Instancia de AdministradorDeArchivosFml.

Variables de instancia:

nombresPaginasPrincipales
Arreglo con los nombres de los archivos fml principales, excepto los nombre de los archivos fml de las páginas modelo.

Métodos de clase:

(Ninguno)

Métodos de instancia:

__init__()
Inicializa las variables de instancia del objeto en el momento en el que este es creado.

puedeCustomizar()
Devuelve true si el usuario tiene directorio de configuracion creado.

nombreFmlCabeceraHorizontalModelo()
Retorna el nombre del archivo fml de la cabecera horizontal del archivo fml modelo.

nombreFmlLogoCabeceraHorizontalModelo()
Retorna el nombre del archivo fml del logo de la cabecera horizontal del archivo fml modelo.

nombreFmlTituloCabeceraHorizontalModelo()

Retorna el nombre del archivo fml del titulo de la cabecera horizontal del archivo fml modelo.

nombreFmlMenuCabeceraHorizontalModelo()

Retorna el nombre del archivo fml del menu de la cabecera horizontal del archivo fml modelo.

nombreFmlCabeceraVerticalModelo()

Retorna el nombre del archivo fml de la cabecera vertical del archivo fml modelo.

nombreFmlLogoCabeceraVerticalModelo()

Retorna el nombre del archivo fml del logo de la cabecera vertical del archivo fml modelo.

nombreFmlLogoInferiorCabeceraVerticalModelo()

Retorna el nombre del archivo fml del logo inferior de la cabecera vertical del archivo fml modelo.

nombreFmlDatosUsuarioCabeceraVerticalModelo()

Retorna el nombre del archivo fml de los datos del usuario de la cabecera vertical del archivo fml modelo.

nombreFmlDatosUsuarioAlumnoCabeceraVerticalModelo()

Retorna el nombre del archivo fml de los datos del usuario alumno de la cabecera vertical del archivo fml modelo.

nombreFmlTituloCabeceraVerticalModelo()

Retorna el nombre del archivo fml del titulo de la cabecera vertical del archivo fml modelo.

nombreFmlMenuCabeceraVerticalModelo()

Retorna el nombre del archivo fml del menu de la cabecera horizontal del archivo fml modelo.

nombreFmlSubMenuCabeceraVerticalModelo()

Retorna el nombre del archivo fml del submenu de la cabecera horizontal del archivo fml modelo.

aplicarModelo()

Realiza el proceso de cambio en todos los archivos de metaformato en base a los archivos modelo.

aplicarModeloEnPaginasPrincipales()

Realiza el proceso de cambio de los archivos principales de metaformato en base al archivo modelo.

aplicarModeloEnPaginasDeCabeceraHorizontal()

Realiza el proceso de cambio de los archivos de cabecera horizontal de metaformato en base al archivo modelo.

aplicarModeloEnPaginasDeLogoCabeceraHorizontal()

Realiza el proceso de cambio de los archivos de logo de cabecera horizontal de metaformato en base al archivo modelo.

- `aplicarModeloEnPaginasDeTituloCabeceraHorizontal()`
Realiza el proceso de cambio de los archivos de titulo de cabecera horizontal de metaformato en base al archivo modelo.
- `aplicarModeloEnPaginasDeMenuCabeceraHorizontal()`
Realiza el proceso de cambio de los archivos de menu de cabecera horizontal de metaformato en base al archivo modelo.
- `aplicarModeloEnPaginasDeCabeceraVertical()`
Realiza el proceso de cambio de los archivos de cabecera vertical de metaformato en base al archivo modelo.
- `aplicarModeloEnPaginasDeLogoCabeceraVertical()`
Realiza el proceso de cambio de los archivos de logo de cabecera vertical de metaformato en base al archivo modelo.
- `aplicarModeloEnPaginasDeLogoInferiorCabeceraVertical()`
Realiza el proceso de cambio de los archivos de logo inferior de cabecera vertical de metaformato en base al archivo modelo.
- `aplicarModeloEnPaginasDeDatosUsuarioCabeceraVertical()`
Realiza el proceso de cambio de los archivos de titulo de cabecera vertical de metaformato en base al archivo modelo.
- `aplicarModeloEnPaginasDeDatosUsuarioAlumnoCabeceraVertical()`
Realiza el proceso de cambio de los archivos de datos de usuario en cabecera vertical de metaformato en base al archivo modelo.
- `aplicarModeloEnPaginasDeTituloCabeceraVertical()`
Realiza el proceso de cambio de los archivos de titulo de cabecera vertical de metaformato en base al archivo modelo.
- `aplicarModeloEnPaginasDeMenuCabeceraVertical()`
Realiza el proceso de cambio de los archivos de menu de cabecera vertical de metaformato en base al archivo modelo.
- `aplicarModeloEnPaginasDeSubMenuCabeceraVertical()`
Realiza el proceso de cambio de los archivos de submenu de cabecera vertical de metaformato en base al archivo modelo
- `obtenerLineasDeModeloYCambiarEnArchivos(archivoModelo, key, paginas)`
Reemplaza en los archivos fml las líneas que contengan la palabra clave, recibida como parámetro, por las líneas del archivo modelo que tengan la misma palabra clave.
archivoModelo -> nombre del archivo modelo a aplicar.
key -> palabra clave a buscar en el archivo modelo para obtener las líneas modelo. Palabra clave a buscar en los archivos fml para reemplazar las líneas por el modelo.
paginas -> colección de nombres de archivos fml que serán modificados con las líneas modelo.
- `obtenerLineasModeloWithKey(archivoModelo, key)`
Devuelve la línea del archivo modelo que tiene la palabra clave recibida como parametro

Nombre: EspolAlumno

Descripción:

Estudiante registrado en los diferentes cursos.

Estados:

(Ninguno)

Relaciones:

Superclase: EspolPersona
Subclases: (Ninguna)
Asociaciones: (Ninguna)

Variables globales:

(Ninguna)

Variables de instancia:

(Ninguna)

Métodos de clase:

instanciasDeQuery(sql)

Retorna las instancias obtenidas a partir de la ejecución del sql recibido como parámetro.

writeAlumnosMigrationFile(ruta, nombreArchivo, contenido, delimitador)

Utilizado para el proceso de migración. Escribe el archivo depurado (sin líneas que produzcan registros duplicados) de migración de alumnos en el servidor.

ruta -> nombre del directorio en el que debe grabarse.

nombreArchivo -> nombre del archivo que debe ser grabado.

contenido -> líneas que contiene el archivo, las mismas que serán analizadas para determinar si provocarían error por registro duplicado en la migración.

delimitador -> caracter separador de campos en cada línea.

Métodos de instancia:

__init__()

Inicializa las variables de instancia del objeto en el momento en el que este es creado.

puedeSerEliminado()

Devuelve true si este alumno no podría estar siendo referenciado por algún otro objeto en el sistema, de tal forma que pueda ser eliminado.

esDocente()

Siempre retorna false

esAyudante()

Devuelve true si este estudiante está registrado como ayudante en alguna materia, caso contrario devuelve false.

Notas:

Nombre: EspolAlumnoPorParalelo

Descripción:

Mantiene la relación entre un estudiante y un paralelo (materia registrada).

Estados:

Activa (AC)
Aprobada (AP)
Reprobada (RP)
Exonerada (EX)
Perdió por faltas (PF)
Convalidada (CV)

Relaciones:

Superclase: EspolElementoDelPeriodoAcademico
Subclases: (Ninguna)
Asociaciones: EspolAlumno
EspolCalificacionDeAlumno
EspolParalelo

Variables globales:

(Ninguna)

Variables de instancia:

tipIdentificacionAlumno
Puede ser 'CED' - cédula, 'PAS' - pasaporte, 'MAT' – matrícula.

numeroIdentificacionAlumno
Número de identificación del alumno.

nota1
Nota obtenida por el alumno en este paralelo en el primer parcial.

nota2
Nota obtenida por el alumno en este paralelo en el segundo parcial.

nota3
Nota obtenida por el alumno en este paralelo en el tercer parcial.

notaMejoramiento
Nota obtenida por el alumno en este paralelo en el examen de mejoramiento.

estado
Estado del estudiante en la materia.

alumno
Instancia de EspolAlumno.

calificacionesDeAlumno
Colección de instancias de EspolCalificacionDeAlumno.

paralelo
Instancia de EspolParalelo.

Métodos de clase:

- `writeAlumnosPorParaleloMigrationFile(ruta, nombreArchivo, contenido, delimitador)`
Utilizado para el proceso de migración. Escribe el archivo depurado (sin líneas que produzcan registros duplicados) de migración de alumnos por paralelo en el servidor.
ruta -> nombre del directorio en el que debe grabarse.
nombreArchivo -> nombre del archivo que debe ser grabado.
contenido -> líneas que contiene el archivo, las mismas que serán analizadas para determinar si provocarían error por registro duplicado en la migración.
delimitador -> caracter separador de campos en cada línea.
- `rowsDeAlumnosDelParalelo(paralelo)`
Retorna una colección de rows de los alumnos registrados en la instancia de paralelo recibida como parámetro.
- `eliminarRowsDelParalelo(paralelo)`
Elimina los registros de alumnos por paralelo en base a la instancia recibida como parámetro.

Métodos de instancia:

- `__init__()`
Inicializa las variables de instancia del objeto en el momento en el que este es creado.
- `getTipoidentificacionAlumno()`
Retorna el valor del atributo `tipoidentificacionAlumno`.
- `setTipoidentificacionAlumno(unString)`
Setea un valor al atributo `tipoidentificacionAlumno`.
- `getNumeroIdentificacionAlumno()`
Retorna el valor del atributo `numeroIdentificacionAlumno`.
- `setNumeroIdentificacionAlumno(unString)`
Setea un valor al atributo `numeroIdentificacionAlumno`.
- `getNota1()`
Retorna el valor del atributo `nota1`.
- `setNota1(unString)`
Setea un valor al atributo `nota1`.
- `getNota2()`
Retorna el valor del atributo `nota2`.
- `setNota2(unString)`
Setea un valor al atributo `nota2`.
- `getNota3()`
Retorna el valor del atributo `nota3`.
- `setNota3(unString)`
Setea un valor al atributo `nota3`.

`getNotaMejoramiento()`
Retorna el valor del atributo `notaMejoramiento`.

`setNotaMejoramiento(unString)`
Setea un valor al atributo `notaMejoramiento`.

`getEstado()`
Retorna el valor del atributo `estado`.

`setEstado(unString)`
Setea un valor al atributo `estado`.

`getAlumno()`
Retorna la instancia del alumno asignado en esta paralelo.

`setAlumno(anEspolAlumno)`
Asigna `anEspolAlumno` al atributo `alumno`.

`getNombreAlumno()`
Devuelve los apellidos y nombres del estudiante.

`getEmailAlumno()`
Devuelve el email del estudiante.

`getCalificacionesDeAlumno()`
Retorna un arreglo de instancias de `EspolCalificacionesDeAlumno` que pertenecen a esta instancia.

`setCalificacionesDeAlumno(aColeccion)`
Asigna `aColeccion` al atributo `calificacionesDeAlumno`.

`getCalificacionesDeAlumnoDeParcial(parcial)`
Retorna un arreglo de instancias de `EspolCalificacionesDeAlumno` que pertenecen a esta instancia y a un `parcial` recibido como parámetro.

`getCalificacionDeAlumnoConNumero(numeroCalificacion)`
Retorna una instancia de `EspolCalificacionDeAlumno` que tiene el `numero` recibido como parámetro.

`setParalelo(anEspolParalelo)`
Asigna `anEspolParalelo` al atributo `paralelo`.

`getParalelo()`
Retorna el valor del atributo `paralelo`.

`getNombreMateria()`
Retorna el nombre de la materia del paralelo.

`getAlumnosDelParalelo()`
Retorna un arreglo de rows de alumnos que se encuentran registrados en este paralelo.

`getParalelosDeUnAlumno()`

Retorna un arreglo de rows de los paralelos en los que este alumno está registrado ordenados por nombre de materia.

`getNotaCalculadaEnParcial(parcial)`

Retorna el puntaje en nota1 calculado en base a las calificaciones obtenidas por el estudiante en el parcial recibido como parámetro.

Notas:

Nombre: EspolAnuncio

Descripción:

Mantiene la información de un anuncio publicado.

Estados:

(Ninguno)

Relaciones:

Superclase: EspolElementoDelPeriodoAcademico

Subclases: (Ninguna)

Asociaciones: EspolParalelo

Variables globales:

(Ninguna)

Variables de instancia:

numero
Identifica al anuncio de otro.

fecha
Fecha de publicación del anuncio.

titulo
Título del anuncio.

tipoArchivo
Tipo de archivo que contiene el detalle del anuncio.
D -> Documento (archivo), T -> Texto

nombreArchivo
Nombre de archivo que contiene el detalle del anuncio.

paralelo
Instancia de EspolParalelo.

Métodos de clase:

`eliminarRowsDelParalelo(paralelo)`
Elimina los registros de anuncios de un paralelo recibido como parámetro.

Métodos de instancia:

- `__init__()`
Inicializa las variables de instancia del objeto en el momento en el que este es creado.

- `getNumero()`
Retorna el valor del atributo numero.

- `setNumero(unString)`
Setea un valor al atributo numero.

- `getFecha()`
Retorna el valor del atributo fecha.
- `setFecha(unString)`
Setea un valor al atributo fecha.

- `getTitulo()`
Retorna el valor del atributo titulo.

- `setTitulo(unString)`
Setea un valor al atributo titulo.

- `getTipoArchivo()`
Retorna el valor del atributo tipoArchivo.

- `setTipoArchivo(unString)`
Setea un valor al atributo tipoArchivo.

- `getNombreArchivo()`
Retorna el valor del atributo nombreArchivo.

- `setNombreArchivo(unString)`
Setea un valor al atributo nombreArchivo.

- `getParalelo()`
Retorna el valor del atributo paralelo.

- `setParalelo(anEspolParalelo)`
Asigna anEspolParalelo al atributo paralelo.

- `tipoObjeto()`
Siempre retorna la letra "A" para indicar que éste objeto es un anuncio.

- `getProximoNombreParaArchivo(unNumero)`
Retorna el nombre que deberá ser asignado al próximo archivo tipo texto a grabar.

Notas:

Nombre: EspolCalificacionDeAlumno

Descripción:

Su función es mantener la nota obtenida por un estudiante en una política de calificación específica (tarea, deber, lección, exámen, etc.) y determinar cuál es la nota final de acuerdo al porcentaje que representa de la nota parcial (a través de la política de calificación).

Estados:

(Ninguno)

Relaciones:

Superclase: EspolElementoDelPeriodoAcademico
 Subclases: (Ninguna)
 Asociaciones: EspolAlumnoPorParalelo
 EspolPoliticaDeCalificacion

Variables globales:

(Ninguna)

Variables de instancia:

tipIdentificacionAlumno

Puede ser 'CED' - cédula, 'PAS' - pasaporte, 'MAT' - matrícula.

numeroIdentificacionAlumno

Número de identificación del alumno al que pertenece esta calificación.

numeroPolitica

Número de política de calificación a la que se está haciendo referencia en esta instancia.

nota

Nota obtenida por el estudiante.

alumnoPorParalelo

Instancia de EspolAlumnoPorParalelo.

politicaDeCalificacion

Instancia de EspolPoliticaDeCalificacion.

Métodos de clase:

notaFinalPromedioCurso(politica, paralelo)

Devuelve la nota final promedio respecto al porcentaje que corresponde de todos los estudiantes de un paralelo.

Métodos de instancia:

__init__()

Inicializa las variables de instancia del objeto en el momento en el que este es creado.

getTipIdentificacionAlumno()

Retorna el valor del atributo tipIdentificacionAlumno.

setTipIdentificacionAlumno(unString)

Setea un valor al atributo tipoidentificacionAlumno.

getNumeroIdentificacionAlumno()

Retorna el valor del atributo numeroIdentificacionAlumno.

setNumeroIdentificacionAlumno(unString)

Setea un valor al atributo numeroIdentificacionAlumno.

getNumeroPolitica()

Retorna el valor del atributo numeroPolitica.

setNumeroPolitica(unString)

Setea un valor al atributo numeroPolitica.

getNota()

Retorna el valor del atributo nota.

setNota(unString)

Setea un valor al atributo nota.

getAlumnoPorParalelo()

Retorna la instancia del alumno asignado.

setAlumnoPorParalelo(anEspolAlumnoPorParalelo)

Asigna anEspolAlumnoPorParalelo al atributo alumnoPorParalelo.

getPoliticaDeCalificacion()

Retorna la instancia de la politica de calificacion asignada.

setPoliticaDeCalificacion(anEspolPoliticaDeCalificacion)

Asigna anEspolPoliticaDeCalificacion al atributo politicaDeCalificacion.

notaFinal()

Devuelve la nota final respecto al porcentaje que corresponde.

Nombre: EspolCurriculum

Descripción:

Mantiene información personal de un usuario.

Estados:

(Ninguno)

Relaciones:

Superclase: ObjetoPersistente

Subclases: (Ninguna)

Asociaciones: (Ninguna)

Variables globales:

(Ninguna)

Variables de instancia:

tipoidentificacion

Puede ser 'CED' - cédula, 'PAS' - pasaporte, 'MAT' - matrícula.

numeroidentificacion

Número de identificación del usuario al que pertenece esta información.

educacion

Texto que contiene una reseña de la educación académica del usuario.

experienciaLaboral

Texto que contiene una reseña de la experiencia laboral del usuario.

hobbies

Texto que contiene información sobre los hobbies practicados por el usuario.

Métodos de clase:

(Ninguno)

Métodos de instancia:

__init__()

Inicializa las variables de instancia del objeto en el momento en el que este es creado.

getTipoidentificacion()

Retorna el valor del atributo tipoidentificacion.

setTipoidentificacion(unString)

Setea un valor al atributo tipoidentificacion.

getNumeroIdentificacion()

Retorna el valor del atributo numeroidentificacion.

setNumeroIdentificacion(unString)

Setea un valor al atributo numeroidentificacion.

getIdentificacion()

Retorna el tipo y numero de identificacion concatenados.

getEducacion()

Retorna el valor del atributo educacion.

setEducacion(unString)

Setea un valor al atributo educacion.

getExperienciaLaboral()

Retorna el valor del atributo experienciaLaboral.

setExperienciaLaboral(unString)

Setea un valor al atributo experienciaLaboral.

getHobbies()

Retorna el valor del atributo hobbies.

setHobbies(unString)
Setea un valor al atributo hobbies.

Notas:

Nombre: EspolDetalleDeSolicitudDeRecurso

Descripción:

Especifica un momento exacto (semana, día y hora) en el que un recurso es solicitado.

Estados:

Pendiente (P)
Entregado (E)
Devuelto (D)

Relaciones:

Superclase: ObjetoPersistente
Subclases: (Ninguna)
Asociaciones: EspolSolicitudDeRecurso

Variables globales:

(Ninguna)

Variables de instancia:

codigoUnidad
Código de la unidad académica a la que pertenece el recurso solicitado.

numeroSolicitud
Número de solicitud al que pertenece este detalle.

semanaDiaHoraInicial
Contiene el número de la semana, día y hora inicial concatenados para el que este recurso se quiere reservar.

numeroSemana
Contiene el número de semana a reservar.

numeroDia
Contiene el número de día a reservar.

horaInicial
Contiene la hora desde la que se desea reservar el recurso.

horaFinal
Contiene la hora hasta la que se desea reservar el recurso.

estado
Contiene el estado del detalle de la solicitud del recurso.

fechaDevolucion
Fecha en el que el recurso fue devuelto.

horaDevolucion
Hora en el que el recurso fue devuelto.

solicitudDeRecurso
Instancia de EspolSolicitudDeRecurso.

Métodos de clase:

rowsDeDetallesPorSolicitud(unaSolicitud)
Retorna una coleccion de los detalles de las solicitudes hechas (pendientes) para una solicitud en especial.

rowsDeDetallesPorSolicitudYSemana(unaSolicitud, unaSemana)
Retorna una coleccion de los detalles de las solicitudes hechas para una solicitud y una semana en especial.

Métodos de instancia:

__init__()
Inicializa las variables de instancia del objeto en el momento en el que este es creado.

getCodigoUnidad()
Retorna el valor del atributo codigoUnidad.

setCodigoUnidad(unString)
Setea un valor al atributo codigoUnidad.

getIdentificacionSolicitante()
Retorna el tipo y numero de identificacion del solicitante concatenados.

getNumeroSolicitud()
Retorna el valor del atributo numeroSolicitud.

setNumeroSolicitud(unString)
Setea un valor al atributo numeroSolicitud.

getSemanaDiaHoraInicial()
Retorna el valor del atributo numeroSemanaDia.

setSemanaDiaHoraInicial(unString)
Setea un valor al atributo numeroSemanaDia.

getNumeroSemana()
Retorna el valor del atributo numeroSemana.

setNumeroSemana(unString)
Setea un valor al atributo numeroSemana.

getNumeroDia()
Retorna el valor del atributo numeroDia.

setNumeroDia(unString)
Setea un valor al atributo numeroDia.

`getSemanaDia()`
Retorna la semana y día.

`getHoraInicial()`
Retorna el valor del atributo `horaInicial`.

`setHoraInicial(unString)`
Setea un valor al atributo `horaInicial`.

`getHoraFinal()`
Retorna el valor del atributo `horaFinal`.

`setHoraFinal(unString)`
Setea un valor al atributo `horaFinal`.

`getEstado()`
Retorna el valor del atributo `estado`.

`setEstado(unString)`
Setea un valor al atributo `estado`.

`getFechaDevolucion()`
Retorna el valor del atributo `fechaDevolucion`.

`setFechaDevolucion(unString)`
Setea un valor al atributo `fechaDevolucion`.

`getHoraDevolucion()`
Retorna el valor del atributo `horaDevolucion`.

`setHoraDevolucion(unString)`
Setea un valor al atributo `horaDevolucion`.

`getEstaPendiente()`
Retorna true si el objeto esta pendiente.

`setPendiente()`
Cambia a estado pendiente al objeto.

`getEstaEntregado()`
Retorna true si el objeto esta entregado.

`setEntregado()`
Cambia a estado entregado al objeto.

`getEstaDevuelto()`
Retorna true si el objeto esta devuelto.

`setDevuelto()`
Cambia a estado devuelto al objeto.

`getSolicitudDeRecurso()`
Retorna la instancia de la solicitud de recurso asignado en esta instancia.

setSolicitudDeRecurso(anEspolSolicitudDeRecurso)
 Asigna anEspolSolicitudDeRecurso al atributo solicitudDeRecurso.

Notas:

Nombre: EspolDiaCalendarioDelProfesor

Descripción:

Cada instancia de esta clase contendrá la información necesaria sobre un evento en un día de calendario definido por el profesor.

Estados:

(Ninguno)

Relaciones:

Superclase: EspolElementoDelPeriodoAcademico
 Subclases: (Ninguna)
 Asociaciones: EspolParalelo

Variables globales:

(Ninguna)

Variables de instancia:

anioCalendario

Año en el que se ha definido el evento.

mesCalendario

Mes en el que se ha definido el evento.

dia

Día en el que se ha definido el evento.

paralelo

Instancia de EspolParalelo.

titulo

Título del evento.

descripcion

Breve descripción del evento.

tipoArchivo

Tipo de archivo que contiene el detalle del anuncio.

D -> Documento (archivo), T -> Texto

nombreArchivo

Nombre de archivo que contiene el detalle del anuncio.

migrado

Indica si el evento del día de calendario ha sido migrado o no.

Métodos de clase:

eliminarRowsDelParalelo(paralelo)

Elimina los registros de días calendario del paralelo recibido como parámetro

detalleDias(paralelo, anio, month)

Devuelve un arreglo de rows de los días de calendario que pertenecen a un paralelo, año y mes.

Métodos de instancia:

__init__()

Inicializa las variables de instancia del objeto en el momento en el que este es creado.

getAnioCalendario()

Retorna el valor del atributo anio del calendario.

setAnioCalendario(unString)

Setea un valor al atributo anio del Calendario.

getMesCalendario()

Retorna el valor del atributo mes del calendario.

setMesCalendario(unString)

Setea un valor al atributo mes del calendario.

getDiaCalendario()

Retorna el valor del atributo mes del calendario.

setDiaCalendario(unString)

Setea un valor al atributo día del calendario.

getNombreDelMes()

Retorna el nombre del mes del calendario.

getTitulo()

Retorna el valor del atributo titulo del calendario.

setTitulo(unString)

Setea un valor al atributo titulo del calendario.

getDescripcion()

Retorna el valor del atributo descripcion.

setDescripcion(unString)

Setea un valor al atributo descripcion del calendario.

getEstado()

Retorna el valor del atributo estado.

setEstado(unString)

Setea un valor al atributo estado.

`getTipoArchivo()`
Retorna el valor del atributo `tipoArchivo`.

`setTipoArchivo(unString)`
Setea un valor al atributo `tipoArchivo`.

`getNombreArchivo()`
Retorna el valor del atributo `nombreArchivo`.

`setNombreArchivo(unString)`
Setea un valor al atributo `nombreArchivo`.

`getFechaEnCalendario()`
Retorna una fecha completa a partir del año, mes y día del día de calendario.

`getMigrado()`
Retorna el valor del atributo `migrado`.

`setMigrado(unString)`
Setea un valor al atributo `migrado`.

`setEsMigrado()`
Setea "S" al atributo `migrado`.

`setNoEsMigrado()`
Setea "N" al atributo `migrado`.

`getParalelo()`
Retorna la instancia del paralelo asignado en esta instancia.

`setParalelo(anEspolParalelo)`
Asigna `anEspolParalelo` al atributo `paralelo`.

`tipoObjeto()`
Siempre retorna la letra "C" para indicar que éste objeto es un evento de día de calendario.

`getProximoNombreParaArchivo(unNumero)`
Retorna el nombre que deberá ser asignado al próximo archivo texto a grabar.

Notas:

Nombre: EspolElementoDelPeriodoAcademico

Descripción:

Contiene los atributos comunes que mantienen todas las clases de objetos relacionados a un paralelo.

Estados:

(Ninguno)

Relaciones:

Superclase: ObjetoPersistente
 Subclases: EspolAlumnoPorParalelo
 EspolAnuncio
 EspolCalificacionDeAlumno
 EspolDiaCalendarioDelProfesor
 EspolInformacionDelProfesor
 EspolMigracionDeMaterial
 EspolPoliticaDeCalificacion
 EspolPoliticaDelProfesor
 EspolRespuestaDeTarea
 EspolTarea
 EspolTemaDiscusion
 EspolVario
 Asociaciones: (Ninguna)

Variables globales:

(Ninguna)

Variables de instancia:

anio
 Año del período académico.

termino
 Término del período académico.

codigoMateria
 Código de la materia.

numeroParalelo
 Número del paralelo.

Métodos de clase:

(Ninguno)

Métodos de instancia:

__init__()
 Inicializa las variables de instancia del objeto en el momento en el que este es creado.

getAnio()
 Retorna el valor del atributo anio.

setAnio(unString)
 Setea un valor al atributo anio.

getTermino()
 Retorna el valor del atributo termino.

setTermino(unString)
 Setea un valor al atributo termino.

getCodigoMateria()

Retorna el valor del atributo codigoMateria.

setCodigoMateria(unString)
Setea un valor al atributo codigoMateria.

getNumeroParalelo()
Retorna el valor del atributo numeroParalelo.

setNumeroParalelo(unString)
Setea un valor al atributo numeroParalelo.

Notas:

Esta clase es abstracta.

Nombre: EspolEmpleado

Descripción:

Representa a un empleado de la institución que puede ser docente o administrativo.

Estados:

(Ninguno)

Relaciones:

Superclase: EspolPersona
Subclases: (Ninguna)
Asociaciones: (Ninguna)

Variables globales:

(Ninguna)

Variables de instancia:

tipoPersonal
Indica si es docente (D) o administrativo (A).

Métodos de clase:

writeEmpleadosMigrationFile(ruta, nombreArchivo, contenido, delimitador)
Utilizado para el proceso de migración. Escribe el archivo depurado (sin líneas que produzcan registros duplicados) de migración de empleados en el servidor.
ruta -> nombre del directorio en el que debe grabarse.
nombreArchivo -> nombre del archivo que debe ser grabado.
contenido -> líneas que contiene el archivo, las mismas que serán analizadas para determinar si provocarían error por registro duplicado en la migración.
delimitador -> caracter separador de campos en cada línea.

instanciasDeQuery(sql)
Retorna las instancias obtenidas a partir de la ejecución del sql recibido como parámetro.

Métodos de instancia:

- `__init__()`
Inicializa las variables de instancia del objeto en el momento en el que este es creado.
- `getTipoPersonal()`
Retorna el valor del atributo `tipoPersonal`.
- `setTipoPersonal(unString)`
Setea un valor al atributo `tipoPersonal`.
- `setDocente()`
Cambia al empleado como docente.
- `esDocente()`
Devuelve `true` si es un docente.
- `instanciasLikeMeOrderedByApellidos()`
Retorna instancias ordenadas por apellidos y nombre.
- `puedeSerEliminado()`
Devuelve `true` si este empleado no podría estar siendo referenciado por algún otro objeto en el sistema, de tal forma que pueda ser eliminado.

Notas:

Nombre: **EspolInformacionDelProfesor**

Descripción:

Mantiene información publicada por el profesor, no considerada ni anuncio ni política..

Estados:

(Ninguno)

Relaciones:

Superclase: EspolElementoDelPeriodoAcademico
Subclases: (Ninguna)
Asociaciones: EspolParalelo

Variables globales:

(Ninguna)

Variables de instancia:

`numero`
Número de información.

`titulo`
Título de la información.

tipoArchivo

Tipo de archivo que contiene el detalle del anuncio.
D -> Documento (archivo), T -> Texto

nombreArchivo

Nombre de archivo que contiene el detalle del anuncio.

paralelo

Instancia de EspolParalelo.

Métodos de clase:

eliminarRowsDelParalelo(paralelo)

Elimina los registros de un paralelo recibido como parámetro.

Métodos de instancia:

__init__()

Inicializa las variables de instancia del objeto en el momento en el que este es creado.

getNumero()

Retorna el valor del atributo numero.

setNumero(unString)

Setea un valor al atributo numero.

getTitulo()

Retorna el valor del atributo titulo.

setTitulo(unString)

Setea un valor al atributo titulo.

getTipoArchivo()

Retorna el valor del atributo tipoArchivo.

setTipoArchivo(unString)

Setea un valor al atributo tipoArchivo.

getNombreArchivo()

Retorna el valor del atributo nombreArchivo.

setNombreArchivo(unString)

Setea un valor al atributo nombreArchivo.

getParalelo()

Retorna la instancia del paralelo asignado en esta instancia.

setParalelo(anEspolParalelo)

Asigna anEspolParalelo al atributo paralelo.

tipoObjeto()

Retorna el carácter "I" indicando que es de tipo Información.

getProximoNombreArchivo()

Retorna el nombre que debe ser asignado al próximo archivo texto a grabar.

Notas:**Nombre:** EspolMateria**Descripción:**

Representa una materia.

Estados:Activa (A)
Inactiva (I)**Relaciones:**Superclase: ObjetoPersistente
Subclases: (Ninguna)
Asociaciones: EspolUnidadAcademica**Variables globales:**

(Ninguna)

Variables de instancia:codigo
Código de la materia.

nombre
Nombre de la materia.

codigoUnidad
Código de la unidad académica a la que pertenece la materia.

estado
Estado de la materia.

unidadAcademica
Instancia de EspolUnidadAcademica.**Métodos de clase:**writeMateriasMigrationFile(ruta, nombreArchivo, contenido, delimitador)
Utilizado para el proceso de migración. Escribe el archivo depurado (sin líneas que produzcan registros duplicados) de migración de materias en el servidor.
ruta -> nombre del directorio en el que debe grabarse.
nombreArchivo -> nombre del archivo que debe ser grabado.
contenido -> líneas que contiene el archivo, las mismas que serán analizadas para determinar si provocarían error por registro duplicado en la migración.
delimitador -> caracter separador de campos en cada línea.materiasActivas()
Retorna una colección de las materias activas ordenadas por nombre.

materiasInactivas()

Retorna una coleccion de las materias inactivas ordenadas por nombre.

Métodos de instancia:

`__init__()`
Inicializa las variables de instancia del objeto en el momento en el que este es creado.

`getCodigo()`
Retorna el valor del atributo codigo.

`setCodigo(unString)`
Setea un valor al atributo codigo.

`getNombre()`
Retorna el valor del atributo nombre.

`setNombre(unString)`
Setea un valor al atributo nombre.

`getCodigoUnidad()`
Retorna el valor del atributo codigoUnidad.

`setCodigoUnidad(unString)`
Setea un valor al atributo codigoUnidad.

`getNumeroExamenesSinMejoramiento()`
Retorna el valor del atributo codigoUnidad.

`setNumeroExamenesSinMejoramiento(unNumero)`
Setea un valor al atributo numeroExamenesSinMejoramiento.

`getTieneMejoramiento()`
Retorna el valor del atributo tieneMejoramiento.

`setTieneMejoramiento(unString)`
Setea un valor al atributo tieneMejoramiento.

`getEstado()`
Retorna el valor del atributo estado.

`setEstado(unString)`
Setea un valor al atributo estado.

`setActiva()`
Cambia el estado de la instancia a activa.

`setInactiva()`
Cambia el estado de la instancia a inactiva.

`getUnidadAcademica()`
Retorna la instancia de EspolUnidadAcademica asignada en esta instancia.

`setUnidadAcademica(anEspolUnidadAcademica)`
Asigna anEspolUnidadAcademica al atributo unidadAcademica.

tieneParalelos()
Devuelve true si la materia tiene paralelos asociados.

Notas:

Nombre: EspolMigracionDeMaterial

Descripción:

Controla el material que ha sido migrado por un profesor de un curso de algún período anterior al actual.

Estados:

(Ninguno)

Relaciones:

Superclase: EspolElementoDelPeriodoAcademico
Subclases: (Ninguna)
Asociaciones: EspolParalelo

Variables globales:

(Ninguna)

Variables de instancia:

anioMigrado
Año del que la información fue migrada.

terminoMigrado
Término del que la información fue migrada.

tareas
Indica si se han migrado las tareas.

politicas
Indica si se han migrado las políticas (del curso y de calificaciones).

anuncios
Indica si se han migrado los anuncios.

informacion
Indica si se han migrado información de curso.

recursos
Indica si se han migrado los recursos académicos.

calendarios
Indica si se han migrado el calendario de actividades.

paralelo

Instancia de EspolParalelo.

Métodos de clase:

(Ninguno)

Métodos de instancia:

__init__()

Inicializa las variables de instancia del objeto en el momento en el que este es creado.

getAnioMigrado()

Retorna el valor del atributo anioMigrado.

setAnioMigrado(unString)

Setea un valor al atributo anioMigrado.

getTerminoMigrado()

Retorna el valor del atributo terminoMigrado.

setTerminoMigrado(unString)

Setea un valor al atributo terminoMigrado.

getTareas()

Retorna el valor del atributo tareas.

setTareas(unString)

Setea un valor al atributo tareas.

getTieneTareasMigradas()

Retorna verdadero si el valor del atributo tareas es S.

setTieneTareasMigradas()

Setea un valor al atributo tareas.

getCalendarios()

Retorna el valor del atributo calendarios.

setCalendarios(unString)

Setea un valor al atributo calendarios.

getTieneCalendariosMigrados()

Retorna verdadero si el valor del atributo calendarios es S.

setTieneCalendariosMigrados()

Setea un valor al atributo calendarios.

getPolíticas()

Retorna el valor del atributo políticas.

setPolíticas(unString)

Setea un valor al atributo políticas.

getTienePolíticasMigradas()

Retorna verdades si el valor del atributo políticas es S.

`setTienePoliticMigradas()`
Setea un valor al atributo politicas.

`getAnuncios()`
Retorna el valor del atributo anuncios.

`setAnuncios(unString)`
Setea un valor al atributo anuncios.

`getTieneAnunciosMigrados()`
Retorna verdadero si el valor del atributo anuncios es S.

`setTieneAnunciosMigrados()`
Setea un valor al atributo anuncios.

`getInformacion()`
Retorna el valor del atributo informacion.

`setInformacion(unString)`
Setea un valor al atributo informacion.

`getTieneInformacionMigradas()`
Retorna verdadero si el valor del atributo informacion es S.

`setTieneInformacionMigradas()`
Setea un valor al atributo informacion.

`getRecursos()`
Retorna el valor del atributo recursos.

`setRecursos(unString)`
Setea un valor al atributo recursos.

`getTieneRecursosMigrados()`
Retorna verdadero si el valor del atributo recursos es S.

`setTieneRecursosMigrados()`
Setea un valor al atributo recursos.

`getParalelo()`
Retorna la instancia del paralelo asignado en esta instancia.

`setParalelo(anEspolParalelo)`
Asigna anEspolParalelo al atributo paralelo.

Notas:

Nombre: EspolNombreDeParcial

Descripción:

Mantiene los parciales considerados en un período académico y el nombre que toman dentro del sistema. Por ejemplo: parcial, final y mejoramiento.

Estados:

(Ninguno)

Relaciones:

Superclase: ObjetoPersistente
Subclases: (Ninguna)
Asociaciones: (Ninguna)

Variables globales:

(Ninguna)

Variables de instancia:

codigo
Código del parcial.

nombre
Nombre del parcial.

Métodos de clase:

nombresDeParciales()
Retorna las instancias de nombres de parciales ordenadas por codigo.

nombresDeParcialesParaParalelo(paralelo)
Retorna las instancias de nombres de parciales disponibles para una materia ordenadas por codigo.

Métodos de instancia:

__init__()
Inicializa las variables de instancia del objeto en el momento en el que este es creado.

getCodigo()
Retorna el valor del atributo codigo.

setCodigo(unString)
Setea un valor al atributo codigo.

getNombre()
Retorna el valor del atributo nombre.

setNombre(unString)
Setea un valor al atributo nombre.

Notas:

Nombre: EspolParalelo

Descripción:

Curso que estará formado por un profesor, ayudante(s) y alumnos en una materia y número de paralelo específicos.

Estados:

(Ninguno)

Relaciones:

Superclase: ObjetoPersistente
 Subclases: (Ninguna)
 Asociaciones: EspolAlumno
 EspolAlumnoPorParalelo
 EspolAnuncio
 EspolDiaCalendarioDelProfesor
 EspolEmpleado
 EspolMateria
 EspolMigracionDeMaterial
 EspolPoliticaDelProfesor
 EspolPoliticaDeCalificacion
 EspolTarea
 EspolVario

Variables globales:

(Ninguna)

Variables de instancia:

anio
 Año al que pertenece el paralelo.

termino
 Término al que pertenece el paralelo.

codigoMateria
 Código de materia.

numeroParalelo
 Número de paralelo.

tipoidentificacionEmpleado
 Tipo de identificación del profesor.

numeroIdentificacionEmpleado
 Número de identificación del profesor.

tipoidentificacionAlumnoAyudante1
 Tipo de identificación del ayudante 1.

numeroIdentificacionAlumnoAyudante1
 Número de identificación del ayudante 1.

tipoidentificacionAlumnoAyudante2
 Tipo de identificación del ayudante 2.

numeroIdentificacionAlumnoAyudante2
Número de identificación del ayudante 2.

materia
Instancia de EspolMateria.

profesor
Instancia de EspolEmpleado.

ayudante1
Instancia de EspolAlumno.

ayudante2
Instancia de EspolAlumno.

alumnosPorParalelo
Arreglo de instancias de EspolAlumnoPorParalelo.

politicasDelProfesor
Arreglo de instancias de EspolPoliticaDelProfesor.

politicasDeCalificacion
Arreglo de instancias de EspolPoliticaDeCalificacion.

anuncios
Arreglo de instancias de EspolAnuncio.

tareas
Arreglo de instancias de EspolTarea.

varios
Arreglo de instancias de EspolVario.

informacionDelProfesor
Arreglo de instancias de EspolInformacionDelProfesor.

diasCalendariosDelProfesor
Arreglo de instancias de EspolDiaCalendarioDelProfesor.

mensajesDelForo
Arreglo de instancias de EspolTemaDiscusión.

migracionDeMaterial
Arreglo de instancias de EspolMigracionDeMaterial.

Métodos de clase:

`writeParalelosMigrationFile(ruta, nombreArchivo, contenido, delimitador)`
Utilizado para el proceso de migración. Escribe el archivo depurado (sin líneas que produzcan registros duplicados) de migración de paralelos en el servidor.
ruta -> nombre del directorio en el que debe grabarse.
nombreArchivo -> nombre del archivo que debe ser grabado.
contenido -> líneas que contiene el archivo, las mismas que serán analizadas para determinar si provocarían error por registro duplicado en la migración.

delimitador -> caracter separador de campos en cada línea.

copiarMaterialDePeriodoAOtro(paraleloAnterior, paraleloActual, instancia)

Copia la información importante de un periodo a otro.

paraleloAnterior -> Paralelo del que se desea copiar la información.

paraleloActual -> Paralelo al que se desea copiar la información.

instancia -> Objeto de clase de la cual se copiará la información.

paralelosDeAyudante(alumno)

Retorna los paralelos de un ayudante ordenados por nombre de la materia.

aniosTerminosDictadosPorProfesor(profesor)

Devuelve un arreglo con los años y terminos en los que un profesor dio alguna materia con excepcion del año termino actual.

materiasDictadasPorProfesorEnAnioTermino(profesor, anio, termino)

Devuelve un arreglo con los codigos y nombres de las materias que un profesor dicto en un año término.

Métodos de instancia:

__init__()

Inicializa las variables de instancia del objeto en el momento en el que este es creado.

getAnio()

Retorna el valor del atributo anio.

setAnio(unString)

Setea un valor al atributo anio.

getTermino()

Retorna el valor del atributo termino.

setTermino(unString)

Setea un valor al atributo termino.

getCodigoMateria()

Retorna el valor del atributo codigoMateria.

setCodigoMateria(unString)

Setea un valor al atributo codigoMateria.

getNumeroParalelo()

Retorna el valor del atributo numeroParalelo.

setNumeroParalelo(unString)

Setea un valor al atributo numeroParalelo.

getTipoidentificacionEmpleado()

Retorna el valor del atributo tipoidentificacionEmpleado.

setTipoidentificacionEmpleado(unString)

Setea un valor al atributo tipoidentificacionEmpleado.

`getNumeroIdentificacionEmpleado()`
Retorna el valor del atributo `numeroIdentificacionEmpleado`.

`setNumeroIdentificacionEmpleado(unString)`
Setea un valor al atributo `numeroIdentificacionEmpleado`.

`getIdentificacionProfesor()`
Devuelve el tipo y numero de identificacion del profesor.

`getTipoIdentificacionAlumnoAyudante1()`
Retorna el valor del atributo.

`setTipoIdentificacionAlumnoAyudante1(unString)`
Setea un valor al atributo `tipoIdentificacionAlumnoAyudante1`.

`getNumeroIdentificacionAlumnoAyudante1()`
Retorna el valor del atributo.

`setNumeroIdentificacionAlumnoAyudante1(unString)`
Setea un valor al atributo.

`getIdentificacionAyudante1()`
Devuelve el tipo y numero de identificacion del ayudante1.

`getTipoIdentificacionAlumnoAyudante2()`
Retorna el valor del atributo.

`setTipoIdentificacionAlumnoAyudante2(unString)`
Setea un valor al atributo `tipoIdentificacionAlumnoAyudante2`.

`getNumeroIdentificacionAlumnoAyudante2()`
Retorna el valor del atributo.

`setNumeroIdentificacionAlumnoAyudante2(unString)`
Setea un valor al atributo `numeroIdentificacionAlumnoAyudante2`.

`getIdentificacionAyudante2()`
Devuelve el tipo y numero de identificacion del ayudante2.

`getEstado()`
Retorna el valor del atributo estado.

`setEstado(unString)`
Setea un valor al atributo estado.

`getMateria()`
Retorna la instancia de materia asignado en este paralelo.

`setMateria(anEspolMateria)`
Asigna `anEspolMateria` al atributo `materia`.

`getNombreMateria()`
Retorna el nombre de la materia del paralelo.

`getProfesor()`

Retorna la instancia del profesor asignado en este paralelo.

setProfesor(anEspolEmpleado)

Asigna anEspolEmpleado al atributo profesor.

getAyudante1()

Retorna la instancia del profesor asignado en este paralelo.

setAyudante1(anEspolAlumno)

Asigna anEspolAlumno al atributo ayudante1.

getAyudante2()

Retorna la instancia del profesor asignado en este paralelo.

setAyudante2(anEspolAlumno)

Asigna anEspolAlumno al atributo ayudante2.

getAlumnosPorParalelo()

Retorna la instancias de alumnosPorParalelo que pertenecen a esta instancia.

setAlumnosPorParalelo(aCollection)

Asigna aCollection al atributo alumnosPorParalelo.

getPolíticasDelProfesor()

Retorna la instancias de políticasDelProfesor que pertenecen a esta instancia.

setPolíticasDelProfesor(aCollection)

Asigna aCollection al atributo políticasDelProfesor.

setInformacionDelProfesor(aCollection)

Asigna aCollection al atributo informacionDelProfesor.

getInformacionDelProfesor()

Retorna la instancias de informacionDelProfesor que pertenecen a esta instancia.

getPolíticasDeCalificación()

Retorna la instancias de políticasDeCalificación que pertenecen a esta instancia.

setPolíticasDeCalificación(aCollection)

Asigna aCollection al atributo políticasDeCalificación.

getAnuncios()

Retorna la instancias de anuncios que pertenecen a esta instancia.

setAnuncios(aCollection)

Asigna aCollection al atributo anuncios.

getTareas()

Retorna la instancias de tareas que pertenecen a esta instancia.

getTareasMigradas()

Retorna la instancias de tareas migradas que pertenecen a esta instancia.

setTareas(aCollection)
Asigna aCollection al atributo tareas.

getVarios()
Retorna la instancias de varios que pertenecen a esta instancia.

setVarios(aCollection)
Asigna aCollection al atributo varios.

getDiasCalendariosDelProfesor()
Retorna la instancias de dias calendarios del profesor que pertenecen a esta instancia.

getDiasCalendariosDelProfesorMigrados()
Retorna la instancias de diasCalendariosDelProfesor migrados que pertenecen a esta instancia.

getMigracionDeMaterial()
Retorna la instancia de migracion de material que pertenece a esta instancia.

getTieneTareasMigradas()
Devuelve true si las tareas de este paralelo ya han sido migradas.

getTieneCalendariosMigrados()
Devuelve true si los calendarios de este paralelo ya han sido migradas.

getTienePoliticasyMigradas()
Devuelve true si las politicas de este paralelo ya han sido migradas.

getTieneAnunciosMigrados()
Devuelve true si las anuncios de este paralelo ya han sido migradas.

getTieneInformacionMigradas()
Devuelve true si la informacion de este paralelo ya han sido migradas.

getTieneRecursosMigrados()
Devuelve true si los recursos de este paralelo ya han sido migradas.

getProximoNumeroAnuncio()
Retorna el numero que debe ser asignado al proximo anuncio creado.

getProximoNumeroPoliticaDelProfesor()
Retorna el numero que debe ser asignado a la proxima politica creada.

getProximoNumeroInformacionDelProfesor()
Retorna el numero que debe ser asignado a la proxima info creada.

getProximoNumeroPoliticaDeCalificacion()
Retorna el numero que debe ser asignado a la proxima politica de calificacion creada.

getProximoNumeroProgramaDeEstudio()

Retorna el numero que debe ser asignado al proximo programa de estudio creado.

getProximoNumeroTarea()

Retorna el numero que debe ser asignado a la proxima tarea creada.

getProximoNumeroVario()

Retorna el numero que debe ser asignado al proximo vario creado.

getProximoNumeroMensaje()

Retorna el numero que debe ser asignado al proximo mensaje creado del primer nivel.

getProximoSecuencialMensaje()

Retorna el secuencial que debe ser asignado al proximo mensaje.

tieneAlumnos()

Devuelve true si el paralelo tiene alumnos asociadas.

tieneDetallesDeSolicitudDeRecurso()

Devuelve true si el paralelo tiene solicitudes de recursos asociadas.

tieneTareas()

Devuelve true si el paralelo tiene tareas asociadas.

tienePolíticasDeProfesor()

Devuelve true si el paralelo tiene políticas de profesor asociadas.

tieneAnuncios()

Devuelve true si el paralelo tiene anuncios asociados.

tienePolíticasDeCalificación()

Devuelve true si el paralelo tiene políticas de calificación asociadas.

tieneProgramasDeEstudio()

Devuelve true si el paralelo tiene programa de estudio asociado.

tieneVarios()

Devuelve true si el paralelo tiene varios asociados.

tieneMensajesDeForo()

Devuelve true si el paralelo tiene mensajes en el foro asociados.

puedeSerEliminado()

Devuelve true si el paralelo puede ser eliminado.

deleteAll()

Elimina el paralelo y toda la información a él relacionada.

getDiasDelCalendario()

Retorna la instancias de los días que pertenecen a esta instancia.

setDiasDelCalendario(aCollection)

Asigna aCollection al atributo diasDelCalendario.

getMensajesDelForo()

Retorna la instancias de los mensajes que pertenecen a esta instancia.

setMensajesDelForo(aCollection)

Asigna aCollection al atributo mensajesDelForo.

generaNuevoAnuncio(unObject, unPath)

Genera un nuevo anuncio tipo texto automaticamente. Esto se hace en ingreso de tareas o recursos, si el usuario lo desea.

mensajesDelPrimerNivel()

Retorna la instancias de los mensajes del primerNivel.

mensajesHijos(objMensaje)

Retorna la instancias de los mensajes hijos.

Notas:

Nombre: EspolParametrosGenerales

Descripción:

Mantiene la datos generales que serán usados en el sistema, ya sea en procesos o para considerar valores por omisión.

Estados:

(Ninguno)

Relaciones:

Superclase: ObjetoPersistente

Subclases: (Ninguna)

Asociaciones: (Ninguna)

Variables globales:

(Ninguna)

Variables de instancia:

__param

Instancia singleton de esta clase.

codigo

Código del pérametro. Es único.

anio

Año del período académico actual.

termino

Término del período académico actual.

fechalnicio

getNumero()
Retorna el valor del atributo numero.

setNumero(unString)
Setea un valor al atributo numero.

getTitulo()
Retorna el valor del atributo titulo.

setTitulo(unString)
Setea un valor al atributo titulo.

getTipoArchivo()
Retorna el valor del atributo tipoArchivo.

setTipoArchivo(unString)
Setea un valor al atributo tipoArchivo.

getNombreArchivo()
Retorna el valor del atributo nombreArchivo.

setNombreArchivo(unString)
Setea un valor al atributo nombreArchivo.

getParalelo()
Retorna la instancia del paralelo asignado en esta instancia.

setParalelo(anEspolParalelo)
Asigna anEspolParalelo al atributo paralelo.

tipoObjeto()
Retorna un char indicando que tipo de objeto es.

getProximoNombreParaArchivo(unNumero)
Retorna el nombre que se debe asignar a la próxima política tipo texto a grabar .

Notas:

Nombre: EspolProceso

Descripción:
Controla las conexiones de cada usuario en el sistema.

Estados:
Activo (A)
Inactivo (I)

Relaciones:
Superclase: ObjetoPersistente

Subclases: (Ninguna)
Asociaciones: EspolServidor

Variables globales:
(Ninguna)

Variables de instancia:

numero
Número del proceso.

codigoServidor
Código de servidor.

usuario
Nombre de usuario.

fechaHora
Fecha y hora de conexión.

direccionIP
Dirección IP de la máquina conectada.

estadoConexion
Estado de conexión.

Métodos de clase:

proximoNumeroDeProceso()
Devuelve el numero que debe ser asignado al proximo proceso.

eliminaConexiones(unaPersona, unaIP, type)
Elimina las conexiones de un usuario.
Si el parametro tipo es nulo eliminamos del de la ip a la que se esta conectando,
sino las de ip diferentes.

existeConexion(unaPersona, unaIP)
Devuelve el numero de conecciones que existen para un usuario.

Métodos de instancia:

__init__()
Inicializa las variables de instancia del objeto en el momento en el que este es creado.

getNumero()
Retorna el valor del atributo numero.

setNumero(unString)
Setea un valor al atributo numero.

getCodigoServidor()
Retorna el valor del atributo codigoServidor.

setCodigoServidor(unString)
Setea un valor al atributo codigoServidor.

`getUsuario()`
Retorna el valor del atributo usuario.

`setUsuario(unString)`
Setea un valor al atributo usuario.

`getFechaHora()`
Retorna el valor del atributo fechaHora.

`setFechaHora(unString)`
Setea un valor al atributo fechaHora.

`getDireccionIP()`
Retorna el valor del atributo direccionIP.

`setDireccionIP(unString)`
Setea un valor al atributo direccionIP.

`getEstadoConexion()`
Retorna el valor del atributo estadoConexion.

`setEstadoConexion(unString)`
Setea un valor al atributo estadoConexion.

`setConexionAbierta()`
Cambia el estado de conexión a abierta.

`setConexionCerrada()`
Cambia el estado de conexión a cerrada.

`getConexionCerrada()`
Devuelve verdadera si la conexión está cerrada.

Notas:

Nombre: EspolRecurso

Descripción:
Recurso físico que podrá ser reservado.

Estados:
Activo (A)
Inactivo (I)

Relaciones:

Superclase:	ObjetoPersistente
Subclases:	(Ninguna)
Asociaciones:	EspolTipoRecurso EspolSolicitudDeRecurso

Variables globales:

(Ninguna)

Variables de instancia:

codigoUnidad

Código de la unidad académica al que pertenece.

codigoTipoRecurso

Código de tipo de recurso.

codigo

Código de recurso.

nombre

Nombre del recurso.

codigoServidor

Código del servidor de autenticación del usuario administrador del recurso.

usuario

Nombre del usuario administrador.

estado

Estado del recurso.

capacidad

Capacidad del recurso. Por ejemplo, número de estudiantes por aula.

localidad

Localidad del recurso.

tipoRecurso

Instancia de EspolTipoRecurso.

unidadAcademica

Instancia de EspolUnidadAcademica.

administrador

Instancia de EspolPersona.

solicitudesDeRecurso

Arreglo de instancias de EspoSolicitudDeRecurso.

servidor

Instancia de EspoServidor.

Métodos de clase:

(Ninguno)

Métodos de instancia:

__init__()

Inicializa las variables de instancia del objeto en el momento en el que este es creado.

`writeAlumnosMigrationFile(ruta, nombreArchivo, contenido, delimitador)`
Utilizado para el proceso de migración. Escribe el archivo depurado (sin líneas que produzcan registros duplicados) de migración de alumnos en el servidor.
ruta -> nombre del directorio en el que debe grabarse.
nombreArchivo -> nombre del archivo que debe ser grabado.
contenido -> líneas que contiene el archivo, las mismas que serán analizadas para determinar si provocarían error por registro duplicado en la migración.
delimitador -> caracter separador de campos en cada línea.

`getCodigoUnidad()`
Retorna el valor del atributo `codigoUnidad`.

`setCodigoUnidad(unString)`
Setea un valor al atributo `codigoUnidad`.

`getCodigoTipoRecurso()`
Retorna el valor del atributo `codigoTipoRecurso`.

`setCodigoTipoRecurso(unString)`
Setea un valor al atributo `codigoTipoRecurso`.

`getCodigo()`
Retorna el valor del atributo `codigo`.

`setCodigo(unString)`
Setea un valor al atributo `codigo`.

`getNombre()`
Retorna el valor del atributo `nombre`.

`setNombre(unString)`
Setea un valor al atributo `nombre`.

`getCodigoServidor()`
Retorna el valor del atributo `codigoServidor`.

`setCodigoServidor(unString)`
Setea un valor al atributo `codigoServidor`.

`getUsuario()`
Retorna el valor del atributo `usuario`.

`setUsuario(unString)`
Setea un valor al atributo `usuario`.

`getEstado()`
Retorna el valor del atributo `estado`.

`setEstado(unString)`
Setea un valor al atributo `estado`.

`setActivo()`
Cambia el estado de la instancia a activo.

`setInactivo()`
Cambia el estado de la instancia a inactivo.

`getCapacidad()`
Retorna el valor del atributo capacidad.

`setCapacidad(unString)`
Setea un valor al atributo capacidad.

`getLocalidad()`
Retorna el valor del atributo localidad.

`setLocalidad(unString)`
Setea un valor al atributo localidad.

`getNombreLocalidad()`
Retorna el nombre de la localidad.

`getTipoRecurso()`
Retorna la instancia del tipoRecurso asignado en esta instancia.

`setTipoRecurso(anEspolTipoRecurso)`
Asigna anEspolTipoRecurso al atributo tipoRecurso.

`getUnidadAcademica()`
Retorna la instancia del unidadAcademica asignado en esta instancia.

`setUnidadAcademica(anEspolUnidadAcademica)`
Asigna anEspolUnidadAcademica al atributo unidadAcademica.

`getAdministrador()`
Retorna la instancia del administrador asignado.

`setAdministrador(anEspolPersona)`
Asigna anEspolPersona al atributo administrador.

`getServidor()`
Retorna la instancia del servidor asignado.

`setServidor(anEspolServidor)`
Asigna anEspolServidor al atributo servidor.

`getSolicitudesDeRecurso()`
Retorna la instancias de solicitudesDeRecurso que pertenecen a esta instancia.

`setSolicitudesDeRecurso(aCollection)`
Asigna aCollection al atributo solicitudesDeRecurso.

`tieneSolicitudesDeRecursos()`
Devuelve true si el recurso tiene solicitudes de recursos asociadas.

`getProximoNumeroSolicitud()`
Retorna el numero que debe ser asignado a la proxima solicitud de ese recurso.

Notas:**Nombre:** EspolRespuestaDeTarea**Descripción:**

Contiene el texto o archivo con los que un estudiante responde a una tarea publicada.

Estados:

(Ninguno)

Relaciones:

Superclase: EspolElementoDelPeriodoAcademico
Subclases: (Ninguna)
Asociaciones: EspolAlumno
EspolTarea

Variables globales:

(Ninguna)

Variables de instancia:

numeroTarea

Número de tarea.

tipoidentificacionAlumno

Tipo de identificación del alumno, dueño de la respuesta de tarea.

numeroIdentificacionAlumno

Número de identificación del alumno, dueño de la respuesta de tarea.

fecha

Fecha en que es respondida la tarea.

titulo

Título de la respuesta de tarea.

tipoArchivo

Tipo de archivo que contiene el detalle del anuncio.
D -> Documento (archivo), T -> Texto

nombreArchivo

Nombre de archivo que contiene el detalle del anuncio.

estadoEntrega

Estado de entrega de la respuesta.

alumno

Instancia de EspolAlumno.

tarea
Instancia de EspoITarea.

Métodos de clase:
(Ninguno)

Métodos de instancia:

__init__()
Inicializa las variables de instancia del objeto en el momento en el que este es creado.

getNumeroTarea()
Retorna el valor del atributo numeroTarea.

setNumeroTarea(unString)
Setea un valor al atributo numeroTarea.

getTipoidentificacionAlumno()
Retorna el valor del atributo tipoidentificacionAlumno.

setTipoidentificacionAlumno(unString)
Setea un valor al atributo tipoidentificacionAlumno.

getNumeroIdentificacionAlumno()
Retorna el valor del atributo numeroIdentificacionAlumno.

setNumeroIdentificacionAlumno(unString)
Setea un valor al atributo numeroIdentificacionAlumno.

getFecha()
Retorna el valor del atributo fecha.

setFecha(unString)
Setea un valor al atributo fecha.

getTipoArchivo()
Retorna el valor del atributo tipoArchivo.

setTipoArchivo(unString)
Setea un valor al atributo tipoArchivo.

getNombreArchivo()
Retorna el valor del atributo nombreArchivo.

setNombreArchivo(unString)
Setea un valor al atributo nombreArchivo.

getEstadoEntrega()
Retorna el valor del atributo estadoEntrega.

setEstadoEntrega(unString)
Setea un valor al atributo estadoEntrega.

getAlumno()
Retorna la instancia del alumno dueño de esta respuesta de tarea.

setAlumno(anEspolAlumno)
 Asigna anEspolAlumno al atributo alumno.

getTarea()
 Retorna la instancia de la tarea de estas respuestas.

setTarea(anEspolTarea)
 Asigna anEspolTarea al atributo tarea.

tipoObjeto()
 Siempre retorna la letra "T" para indicar que éste objeto es una respuesta de tarea.

Notas:

Nombre: **EspolSemanaDeCalendarioDelPeriodo**

Descripción:
 Identifica a una semana dentro del período académico.

Estados:
 (Ninguno)

Relaciones:

Superclase:	ObjetoPersistente
Subclases:	(Ninguna)
Asociaciones:	(Ninguna)

Variables globales:
 (Ninguna)

Variables de instancia:

numero
 Número que identifica a la semana calendario.

descripcion
 Descripción de la semana.

fechaInicio
 Fecha de inicio de la semana.

Métodos de clase:

semanasOrdenadasPorNumero()
 Retorna las instancias de semanas de calendario ordenadas por número.

Métodos de instancia:

__init__()
 Inicializa las variables de instancia del objeto en el momento en el que este es creado.

getNumero()
 Retorna el valor del atributo numero.

setNumero(unString)
 Setea un valor al atributo numero.

getDescripcion()
 Retorna el valor del atributo descripcion.

setDescripcion(unString)
 Setea un valor al atributo descripcion.

getFechaInicio()
 Retorna el valor del atributo fechaInicio.

setFechaInicio(unString)
 Setea un valor al atributo fechaInicio.

tieneDetallesDeSolicitud()
 Devuelve true si la semana tiene detalles de solicitud asociadas.

puedeSerEliminada()
 Devuelve true si la semana puede ser eliminada.

Notas:**Nombre:** EspolServidor**Descripción:**

Servidor de autenticación, contra el cual se valida un nombre de usuario y contraseña.

Estados:

(Ninguno)

Relaciones:

Superclase: ObjetoPersistente
 Subclases: (Ninguna)
 Asociaciones: (Ninguna)

Variables globales:

(Ninguna)

Variables de instancia:

codigo
 Código del servidor.

nombre
 Nombre del servidor.

Métodos de clase:

servidoresOrdenadorPorNombre()

Retorna las instancias de servidores ordenadas por nombre.

servidoresOrdenadosPorNombre()

Retorna las instancias de servidores ordenadas por nombre.

proximoNumeroDeServidor()

Retorna el numero que debe ser asignado al proximo servidor creado.

Métodos de instancia:

__init__()

Inicializa las variables de instancia del objeto en el momento en el que este es creado.

getCodigo()

Retorna el valor del atributo codigo.

setCodigo(unString)

Setea un valor al atributo codigo.

getNombre()

Retorna el valor del atributo nombre.

setNombre(unString)

Setea un valor al atributo nombre.

Notas:

Nombre: EspolSolicitudDeRecurso

Descripción:

Solicitud de un recurso físico.

Estados:

(Ninguno)

Relaciones:

Superclase: ObjetoPersistente

Subclases: (Ninguna)

Asociaciones: EspolEmpleado

 EspolRecurso

 EspolDetalleDeSolicitudDeRecurso

Variables globales:

(Ninguna)

Variables de instancia:

codigoUnidad

Código de Unidad al que pertenece el recurso.

codigoTipoRecurso

Código del tipo de recurso.

codigoRecurso

Código de recurso.

tipoidentificacionSolicitante

Tipo de identificación del solicitante.

numeroIdentificacionSolicitante

Número de identificación del solicitante.

numero

Número de solicitud.

hora

Hora en que fue hecha la solicitud.

estado

Estado de la reservación.

fechaAprobacion

Fecha de aprobación.

observacion

Observación de la solicitud.

recurso

Instancia de EspolRecurso.

solicitante

Instancia de EspolPersona.

detallesDeSolicitudDeRecurso

Arreglo de instancias de EspolDetalleDeSolicitudDeRecurso.

Métodos de clase:

rowsDeSolicitudesDeEmpleadosVigentesPorRecurso(recurso)

Retorna una colección de las solicitudes aprobadas y en espera del recurso, ordenadas por nro de solicitud.

rowsDeSolicitudesDeAlumnosVigentesPorRecurso(recurso)

Retorna una colección de las solicitudes aprobadas y en espera del recurso, ordenadas por nro de solicitud.

rowsDeSolicitudesPendientesPorPersona(persona)

Retorna una colección de las solicitudes que aun tienen detalles pendientes, para una persona, ordenadas por nro de solicitud.

rowsDeSolicitudesEnEspera()

Retorna una colección de las solicitudes que quedarán en estado de espera.

Fecha inicial del período académico actual.

fechaFin

Fecha final del período académico actual.

tipoidentificacionEmpleado

Tipo de identificación del administrador del sistema.

numeroIdentificacionEmpleado

Número de identificación del administrador del sistema.

notaMaxima

Nota máxima que puede ser obtenida en una nota final.

promedioMinimo

Promedio mínimo que debe ser obtenido para aprobar una materia.

administrador

Instancia de EspolEmpleado.

codigoServidor

Código del servidor de autenticación del administrador.

delimitadorCampos

Carácter delimitador de campos utilizados por omisión en la migración de datos.

servidor

Instancia de EspolServidor.

tamanoArchivoProfesor

Tamaño de archivos definido para todos los profesores por omisión.

tamanoArchivoAlumno

Tamaño de archivos definido para todos los estudiantes por omisión.

tiempoCookie

Tiempo en el que el usuario puede dejar de usar el sistema antes de que se le pida verificación de usuario y contraseña.

Métodos de clase:

parametros()

Crea y devuelve una instancia de parametros.

fechaDelSistema()

Fecha actual obtenida del manejador de base de datos.

horaDelSistema()

Hora actual obtenida del manejador de base de datos.

tsDelSistema()

Fecha y hora actuales concatenadas, obtenidas del manejador de base de datos.

Métodos de instancia:

__init__()

Inicializa las variables de instancia del objeto en el momento en el que este es creado.

parametros()

Devuelve el objeto de parametros.

getCodigo()

Retorna el valor del atributo codigo.

setCodigo(unString)

Setea un valor al atributo codigo.

getAnio()

Retorna el valor del atributo anio.

setAnio(unString)

Setea un valor al atributo anio.

getTermino()

Retorna el valor del atributo termino.

setTermino(unString)

Setea un valor al atributo termino.

getFechaInicio()

Retorna el valor del atributo fechaInicio.

setFechaInicio(unString)

Setea un valor al atributo fechaInicio.

getFechaFin()

Retorna el valor del atributo fechaFin.

setFechaFin(unString)

Setea un valor al atributo fechaFin.

getTipoIdentificacionEmpleado()

Retorna el valor del atributo tipoIdentificacionEmpleado.

setTipoIdentificacionEmpleado(unString)

Setea un valor al atributo tipoIdentificacionEmpleado.

getNumeroIdentificacionEmpleado()

Retorna el valor del atributo numeroIdentificacionEmpleado.

getIdentificacionAdministrador()

Retorna el valor del atributo tipo y numero de identificacion concatenados.

setNumeroIdentificacionEmpleado(unString)

Setea un valor al atributo numeroIdentificacionEmpleado.

getAdministrador()

Retorna la instancia del administrador asignado en esta instancia.

`setAdministrador(anEspolEmpleado)`
Asigna anEspolEmpleado al atributo administrador.

`getNotaMaxima()`
Retorna el valor del atributo notaMaxima.

`setNotaMaxima(unString)`
Setea un valor al atributo notaMaxima.

`getPromedioMinimo()`
Retorna el valor del atributo promedioMinimo.

`setPromedioMinimo(unString)`
Setea un valor al atributo promedioMinimo.

`getCodigoServidor()`
Retorna el valor del atributo codigoServidor.

`setCodigoServidor(unString)`
Setea un valor al atributo codigoServidor.

`getServidor()`
Retorna la instancia del servidor asignado.

`setServidor(anEspolServidor)`
Asigna anEspolServidor al atributo servidor.

`getDelimitadorCampos()`
Retorna el valor del atributo delimitadorCampos.

`setDelimitadorCampos(unString)`
Setea un valor al atributo delimitadorCampos.

`getTamanoArchivoProfesor()`
Retorna el valor del atributo tamanoArchivoProfesor.

`setTamanoArchivoProfesor(unString)`
Setea un valor al atributo tamanoArchivoProfesor.

`getTamanoArchivoAlumno()`
Retorna el valor del atributo tamanoArchivoAlumno.

`setTamanoArchivoAlumno(unString)`
Setea un valor al atributo tamanoArchivoAlumno.

`getTiempoCookie()`
Retorna el valor del atributo tiempoCookie.

`setTiempoCookie(unString)`
Setea un valor al atributo tiempoCookie.

Notas:

Nombre: EspolPersona

Descripción:

Profesor o trabajador administrativo en la institución.

Estados:

Activa (A)
Inactiva (I)

Relaciones:

Superclase: ObjetoPersistente
Subclases: EspolAlumno
EspolEmpleado
Asociaciones: EspolCurriculum

Variables globales:

(Ninguna)

Variables de instancia:

tipoidentificacion
Tipo de identificación de la persona.

numeroidentificacion
Número de identificación de la persona.

apellidos
Apellidos de la persona.

nombres
Nombres de la persona.

eMail
Dirección eMail.

sitioWeb
Dirección del sitio web.

codigoServidor
Código del servidor de autenticación.

tipoAutenticacion
Tipo de autenticación. Puede ser: P - POP3 o L - Local.

usuario
Nombre de usuario.

clave
Contraseña de acceso.

estado
Estado de la persona.

servidor
Instancia de EspolServidor.

nickName
Nombre de pila.

nombreArchivoFoto
Nombre del archivo de la foto de la persona.

direccion
Dirección domiciliaria.

telefono1
Teléfono 1.

telefono2
Teléfono 2.

paisNacimiento
Nombre del país de nacimiento.

provinciaNacimiento
Provincia del país de nacimiento.

quierePresentarDireccion
Indica si se desea presentar la dirección dentro de la información personal.

quierePresentarTelefono
Indica si se desea presentar los número telefónicos dentro de la información personal.

curriculum
Instancia de EspolCurriculum.

tamanoArchivo
Tamaño permitido para los archivos.

Métodos de clase:

(Ninguno)

Métodos de instancia:

__init__()
Inicializa las variables de instancia del objeto en el momento en el que este es creado.

getTipIdentificacion()
Retorna el valor del atributo tipIdentificacion.

setTipIdentificacion(unString)
Setea un valor al atributo tipIdentificacion.

getNumeroIdentificacion()
Retorna el valor del atributo numeroIdentificacion.

setNumeroIdentificacion(unString)
Setea un valor al atributo numeroIdentificacion.

`getIdentificacion()`
Retorna el tipo y numero de identificacion concatenados.

`getApellidos()`
Retorna el valor del atributo apellidos.

`setApellidos(unString)`
Setea un valor al atributo matricula.

`getNombres()`
Retorna el valor del atributo nombres.

`getApellidosNombres()`
Retorna el valor de los atributos apellidos y nombres concatenados.

`setNombres(unString)`
Setea un valor al atributo nombres.

`getEmail()`
Retorna el valor del atributo email.

`setEmail(unString)`
Setea un valor al atributo eMail.

`getSitioWeb()`
Retorna el valor del atributo sitioWeb.

`setSitioWeb(unString)`
Setea un valor al atributo sitioWeb.

`getCodigoServidor()`
Retorna el valor del atributo codigoServidor.

`setCodigoServidor(unString)`
Setea un valor al atributo codigoServidor.

`getTipoAutenticacion()`
Retorna el valor del atributo tipoAutenticacion.

`getNombreTipoAutenticacion()`
Retorna el valor del atributo tipoAutenticacion.

`setTipoAutenticacion(unString)`
Setea un valor al atributo tipoAutenticacion.

`setTipoAutenticacionLocal()`
Setea como local el atributo tipoAutenticacion.

`setTipoAutenticacionPop()`
Setea como pop el atributo tipoAutenticacion.

`getTieneTipoAutenticacionLocal()`
Retorna true si el tipo de autenticacion es local.

`getUsuario()`
Retorna el valor del atributo usuario.

`setUsuario(unString)`
Setea un valor al atributo usuario.

`getClave()`
Retorna el valor del atributo clave.

`setClave(unString)`
Setea un valor al atributo clave.

`getEstado()`
Retorna el valor del atributo estado.

`setEstado(unString)`
Setea un valor al atributo estado.

`setActivo()`
Setea activo al atributo estado.

`getServidor()`
Retorna la instancia del servidor asignado.

`setServidor(anEspolServidor)`
Asigna anEspolServidor al atributo servidor.

`getNickName()`
Retorna el valor del atributo nickName.

`setNickName(unString)`
Setea un valor al atributo nickName.

`getNombreArchivoFoto()`
Retorna el valor del atributo nombreArchivoFoto.

`setNombreArchivoFoto(unString)`
Setea un valor al atributo nombreArchivoFoto.

`getDireccion()`
Retorna el valor del atributo direccion.

`setDireccion(unString)`
Setea un valor al atributo direccion.

`getTelefono1()`
Retorna el valor del atributo telefono1.

`setTelefono1(unString)`
Setea un valor al atributo telefono1.

`getTelefono2()`
Retorna el valor del atributo telefono2.

setTelefono2(unString)
Setea un valor al atributo telefono2.

getPaisNacimiento()
Retorna el valor del atributo paisNacimiento.

setPaisNacimiento(unString)
Setea un valor al atributo paisNacimiento.

getProvinciaNacimiento()
Retorna el valor del atributo provinciaNacimiento.

setProvinciaNacimiento(unString)
Setea un valor al atributo provinciaNacimiento.

getQuierePresentarDireccion()
Retorna el valor del atributo quierePresentarDireccion.

setQuierePresentarDireccion(unString)
Setea un valor al atributo quierePresentarDireccion.

getQuierePresentarTelefono()
Retorna el valor del atributo quierePresentarTelefono.

setQuierePresentarTelefono(unString)
Setea un valor al atributo quierePresentarTelefono.

getCurriculum()
Retorna la instancia del curriculum que pertenecen a esta instancia.

setCurriculum(unCurriculum)
Setea un valor al atributo curriculum.

getEducacion()
Devuelve el contenido de educacion del curriculum.

setEducacion(string)
Setea el contenido de educacion del curriculum.

getEducacionParaPresentar()
Devuelve el contenido de educacion el curriculum.

getExperienciaLaboral()
Devuelve el contenido de experiencia laboral del curriculum.

setExperienciaLaboral(string)
Setea el contenido de experiencia laboral del curriculum.

getExperienciaLaboralParaPresentar()
Devuelve el contenido de experiencia laboral del curriculum.

getHobbies()
Devuelve el contenido de hobbies del curriculum.

setHobbies(string)
Setea el contenido de hobbies del curriculum.

getHobbiesParaPresentar()
Devuelve el contenido de hobbies el curriculum.

storeCurriculum()
Graba el curriculum de la persona.

getTamanoArchivo()
Retorna el valor del atributo tamanoArchivo.

setTamanoArchivo(unString)
Setea un valor al atributo tamanoArchivo.

Notas:**Nombre:** EspolPoliticaDeCalificacion**Descripción:**

Es un componente de calificación en la nota parcial, final o mejoramiento.

Estados:

(Ninguno)

Relaciones:

Superclase: EspolElementoDelPeriodoAcademico
Subclases: (Ninguna)
Asociaciones: EspolParalelo

Variables globales:

(Ninguna)

Variables de instancia:

parcial
Código del parcial al que pertenece el componente de calificación.

numero
Número del componente de calificación.

descripcion
Descripción del componente de calificación.

porcentajeNota
Porcentaje que representa en la nota total.

calificacion
Calificación sobre la cual será revisada.

paralelo

Instancia de EspolParalelo.

Métodos de clase:

eliminarRowsDelParalelo(paralelo)

Elimina los registros de un paralelo.

parcialesDefinidosParaParalelo(paralelo)

Devuelve un arreglo con los numeros de parciales en los que se ha definido políticas de calificación en el paralelo recibido como parámetro.

getPolíticasDeParaleloYParcial(paralelo, parcial)

Devuelve un arreglo con los numeros de parciales en los que se ha definido políticas de calificación en el paralelo recibido como parámetro.

Métodos de instancia:

__init__()

Inicializa las variables de instancia del objeto en el momento en el que este es creado.

getParcial()

Retorna el valor del atributo parcial.

setParcial(unString)

Setea un valor al atributo parcial.

getNumero()

Retorna el valor del atributo numero.

setNumero(unString)

Setea un valor al atributo numero.

getDescripcion()

Retorna el valor del atributo descripcion.

setDescripcion(unString)

Setea un valor al atributo descripcion.

getPorcentajeNota()

Retorna el valor del atributo porcentajeNota.

setPorcentajeNota(unString)

Setea un valor al atributo porcentajeNota.

getCalificacion()

Retorna el valor del atributo calificacion.

setCalificacion(unString)

Setea un valor al atributo calificacion.

getParalelo()

Retorna la instancia del paralelo asignado en esta instancia.

setParalelo(anEspolParalelo)

Asigna anEspolParalelo al atributo paralelo.

`eliminarCalificacionesDeAlumnos()`
 Elimina los registros de esta política de calificación.

`calcularPromedioParcial()`
 Elimina las calificaciones de estudiantes ingresadas con esta política y recalcula el promedio del parcial.

Notas:

Nombre: **EspolPoliticaDelProfesor**

Descripción:
 Política definida para el curso.

Estados:
 (Ninguno)

Relaciones:
 Superclase: EspolElementoDelPeriodoAcademico
 Subclases: (Ninguna)
 Asociaciones: EspolParalelo

Variables globales:
 (Ninguna)

Variables de instancia:

`numero`
 Número de la política.

`título`
 Título de la política.

`tipoArchivo`
 Tipo de archivo que contiene el detalle del anuncio.
 D -> Documento (archivo), T -> Texto

`nombreArchivo`
 Nombre de archivo que contiene el detalle del anuncio.

`paralelo`
 Instancia de EspolParalelo.

Métodos de clase:

`eliminarRowsDelParalelo(paralelo)`
 Elimina los registros de un paralelo.

Métodos de instancia:

`__init__()`
 Inicializa las variables de instancia del objeto en el momento en el que este es creado.

Fecha inicial del período académico actual.

fechaFin

Fecha final del período académico actual.

tipoidentificacionEmpleado

Tipo de identificación del administrador del sistema.

numeroIdentificacionEmpleado

Número de identificación del administrador del sistema.

notaMaxima

Nota máxima que puede ser obtenida en una nota final.

promedioMinimo

Promedio mínimo que debe ser obtenido para aprobar una materia.

administrador

Instancia de EspolEmpleado.

codigoServidor

Código del servidor de autenticación del administrador.

delimitadorCampos

Carácter delimitador de campos utilizados por omisión en la migración de datos.

servidor

Instancia de EspolServidor.

tamanoArchivoProfesor

Tamaño de archivos definido para todos los profesores por omisión.

tamanoArchivoAlumno

Tamaño de archivos definido para todos los estudiantes por omisión.

tiempoCookie

Tiempo en el que el usuario puede dejar de usar el sistema antes de que se le pida verificación de usuario y contraseña.

Métodos de clase:

parametros()

Crea y devuelve una instancia de parametros.

fechaDelSistema()

Fecha actual obtenida del manejador de base de datos.

horaDelSistema()

Hora actual obtenida del manejador de base de datos.

tsDelSistema()

Fecha y hora actuales concatenadas, obtenidas del manejador de base de datos.

Métodos de instancia:

__init__()

Inicializa las variables de instancia del objeto en el momento en el que este es creado.

parametros()

Devuelve el objeto de parametros.

getCodigo()

Retorna el valor del atributo codigo.

setCodigo(unString)

Setea un valor al atributo codigo.

getAnio()

Retorna el valor del atributo anio.

setAnio(unString)

Setea un valor al atributo anio.

getTermino()

Retorna el valor del atributo termino.

setTermino(unString)

Setea un valor al atributo termino.

getFechaInicio()

Retorna el valor del atributo fechaInicio.

setFechaInicio(unString)

Setea un valor al atributo fechaInicio.

getFechaFin()

Retorna el valor del atributo fechaFin.

setFechaFin(unString)

Setea un valor al atributo fechaFin.

getTipoIdentificacionEmpleado()

Retorna el valor del atributo tipoIdentificacionEmpleado.

setTipoIdentificacionEmpleado(unString)

Setea un valor al atributo tipoIdentificacionEmpleado.

getNumeroIdentificacionEmpleado()

Retorna el valor del atributo numeroIdentificacionEmpleado.

getIdentificacionAdministrador()

Retorna el valor del atributo tipo y numero de identificacion concatenados.

setNumeroIdentificacionEmpleado(unString)

Setea un valor al atributo numeroIdentificacionEmpleado.

getAdministrador()

Retorna la instancia del administrador asignado en esta instancia.

setAdministrador(anEspolEmpleado)
Asigna anEspolEmpleado al atributo administrador.

getNotaMaxima()
Retorna el valor del atributo notaMaxima.

setNotaMaxima(unString)
Setea un valor al atributo notaMaxima.

getPromedioMinimo()
Retorna el valor del atributo promedioMinimo.

setPromedioMinimo(unString)
Setea un valor al atributo promedioMinimo.

getCodigoServidor()
Retorna el valor del atributo codigoServidor.

setCodigoServidor(unString)
Setea un valor al atributo codigoServidor.

getServidor()
Retorna la instancia del servidor asignado.

setServidor(anEspolServidor)
Asigna anEspolServidor al atributo servidor.

getDelimitadorCampos()
Retorna el valor del atributo delimitadorCampos.

setDelimitadorCampos(unString)
Setea un valor al atributo delimitadorCampos.

getTamanoArchivoProfesor()
Retorna el valor del atributo tamanoArchivoProfesor.

setTamanoArchivoProfesor(unString)
Setea un valor al atributo tamanoArchivoProfesor.

getTamanoArchivoAlumno()
Retorna el valor del atributo tamanoArchivoAlumno.

setTamanoArchivoAlumno(unString)
Setea un valor al atributo tamanoArchivoAlumno.

getTiempoCookie()
Retorna el valor del atributo tiempoCookie.

setTiempoCookie(unString)
Setea un valor al atributo tiempoCookie.

Notas:

Nombre: EspolPersona

Descripción:

Profesor o trabajador administrativo en la institución.

Estados:

Activa (A)
Inactiva (I)

Relaciones:

Superclase: ObjetoPersistente
Subclases: EspolAlumno
EspolEmpleado
Asociaciones: EspolCurriculum

Variables globales:

(Ninguna)

Variables de instancia:

tipoidentificacion
Tipo de identificación de la persona.

numeroidentificacion
Número de identificación de la persona.

apellidos
Apellidos de la persona.

nombres
Nombres de la persona.

eMail
Dirección eMail.

sitioWeb
Dirección del sitio web.

codigoServidor
Código del servidor de autenticación.

tipoAutenticacion
Tipo de autenticación. Puede ser: P - POP3 o L - Local.

usuario
Nombre de usuario.

clave
Contraseña de acceso.

estado
Estado de la persona.

servidor
Instancia de EspolServidor.

nickName
Nombre de pila.

nombreArchivoFoto
Nombre del archivo de la foto de la persona.

direccion
Dirección domiciliaria.

telefono1
Teléfono 1.

telefono2
Teléfono 2.

paisNacimiento
Nombre del país de nacimiento.

provinciaNacimiento
Provincia del país de nacimiento.

quierePresentarDireccion
Indica si se desea presentar la dirección dentro de la información personal.

quierePresentarTelefono
Indica si se desea presentar los número telefónicos dentro de la información personal.

curriculum
Instancia de EspolCurriculum.

tamanoArchivo
Tamaño permitido para los archivos.

Métodos de clase:

(Ninguno)

Métodos de instancia:

__init__()
Inicializa las variables de instancia del objeto en el momento en el que este es creado.

getTipIdentificacion()
Retorna el valor del atributo tipIdentificacion.

setTipIdentificacion(unString)
Setea un valor al atributo tipIdentificacion.

getNumeroIdentificacion()
Retorna el valor del atributo numeroIdentificacion.

setNumeroIdentificacion(unString)
Setea un valor al atributo numeroIdentificacion.

`getIdentificacion()`
Retorna el tipo y numero de identificacion concatenados.

`getApellidos()`
Retorna el valor del atributo apellidos.

`setApellidos(unString)`
Setea un valor al atributo matricula.

`getNombres()`
Retorna el valor del atributo nombres.

`getApellidosNombres()`
Retorna el valor de los atributos apellidos y nombres concatenados.

`setNombres(unString)`
Setea un valor al atributo nombres.

`getEmail()`
Retorna el valor del atributo email.

`setEmail(unString)`
Setea un valor al atributo eMail.

`getSitioWeb()`
Retorna el valor del atributo sitioWeb.

`setSitioWeb(unString)`
Setea un valor al atributo sitioWeb.

`getCodigoServidor()`
Retorna el valor del atributo codigoServidor.

`setCodigoServidor(unString)`
Setea un valor al atributo codigoServidor.

`getTipoAutenticacion()`
Retorna el valor del atributo tipoAutenticacion.

`getNombreTipoAutenticacion()`
Retorna el valor del atributo tipoAutenticacion.

`setTipoAutenticacion(unString)`
Setea un valor al atributo tipoAutenticacion.

`setTipoAutenticacionLocal()`
Setea como local el atributo tipoAutenticacion.

`setTipoAutenticacionPop()`
Setea como pop el atributo tipoAutenticacion.

`getTieneTipoAutenticacionLocal()`
Retorna true si el tipo de autenticacion es local.

`getUsuario()`
Retorna el valor del atributo usuario.

`setUsuario(unString)`
Setea un valor al atributo usuario.

`getClave()`
Retorna el valor del atributo clave.

`setClave(unString)`
Setea un valor al atributo clave.

`getEstado()`
Retorna el valor del atributo estado.

`setEstado(unString)`
Setea un valor al atributo estado.

`setActivo()`
Setea activo al atributo estado.

`getServidor()`
Retorna la instancia del servidor asignado.

`setServidor(anEspolServidor)`
Asigna anEspolServidor al atributo servidor.

`getNickName()`
Retorna el valor del atributo nickName.

`setNickName(unString)`
Setea un valor al atributo nickName.

`getNombreArchivoFoto()`
Retorna el valor del atributo nombreArchivoFoto.

`setNombreArchivoFoto(unString)`
Setea un valor al atributo nombreArchivoFoto.

`getDireccion()`
Retorna el valor del atributo direccion.

`setDireccion(unString)`
Setea un valor al atributo direccion.

`getTelefono1()`
Retorna el valor del atributo telefono1.

`setTelefono1(unString)`
Setea un valor al atributo telefono1.

`getTelefono2()`
Retorna el valor del atributo telefono2.

setTelefono2(unString)
Setea un valor al atributo telefono2.

getPaisNacimiento()
Retorna el valor del atributo paisNacimiento.

setPaisNacimiento(unString)
Setea un valor al atributo paisNacimiento.

getProvinciaNacimiento()
Retorna el valor del atributo provinciaNacimiento.

setProvinciaNacimiento(unString)
Setea un valor al atributo provinciaNacimiento.

getQuierePresentarDireccion()
Retorna el valor del atributo quierePresentarDireccion.

setQuierePresentarDireccion(unString)
Setea un valor al atributo quierePresentarDireccion.

getQuierePresentarTelefono()
Retorna el valor del atributo quierePresentarTelefono.

setQuierePresentarTelefono(unString)
Setea un valor al atributo quierePresentarTelefono.

getCurriculum()
Retorna la instancia del curriculum que pertenecen a esta instancia.

setCurriculum(unCurriculum)
Setea un valor al atributo curriculum.

getEducacion()
Devuelve el contenido de educacion del curriculum.

setEducacion(string)
Setea el contenido de educacion del curriculum.

getEducacionParaPresentar()
Devuelve el contenido de educacion el curriculum.

getExperienciaLaboral()
Devuelve el contenido de experiencia laboral del curriculum.

setExperienciaLaboral(string)
Setea el contenido de experiencia laboral del curriculum.

getExperienciaLaboralParaPresentar()
Devuelve el contenido de experiencia laboral del curriculum.

getHobbies()
Devuelve el contenido de hobbies del curriculum.

setHobbies(string)
Setea el contenido de hobbies del curriculum.

getHobbiesParaPresentar()
Devuelve el contenido de hobbies el curriculum.

storeCurriculum()
Graba el curriculum de la persona.

getTamanoArchivo()
Retorna el valor del atributo tamanioArchivo.

setTamanoArchivo(unString)
Setea un valor al atributo tamanioArchivo.

Notas:**Nombre:** EspolPoliticaDeCalificacion**Descripción:**

Es un componente de calificación en la nota parcial, final o mejoramiento.

Estados:

(Ninguno)

Relaciones:

Superclase: EspolElementoDelPeriodoAcademico
Subclases: (Ninguna)
Asociaciones: EspolParalelo

Variables globales:

(Ninguna)

Variables de instancia:

parcial
Código del parcial al que pertenece el componente de calificación.

numero
Número del componente de calificación.

descripcion
Descripción del componente de calificación.

porcentajeNota
Porcentaje que representa en la nota total.

calificacion
Calificación sobre la cual será revisada.

paralelo

Instancia de EspolParalelo.

Métodos de clase:

eliminarRowsDelParalelo(paralelo)

Elimina los registros de un paralelo.

parcialesDefinidosParaParalelo(paralelo)

Devuelve un arreglo con los numeros de parciales en los que se ha definido políticas de calificación en el paralelo recibido como parámetro.

getPolíticasDeParaleloYParcial(paralelo, parcial)

Devuelve un arreglo con los numeros de parciales en los que se ha definido políticas de calificación en el paralelo recibido como parámetro.

Métodos de instancia:

__init__()

Inicializa las variables de instancia del objeto en el momento en el que este es creado.

getParcial()

Retorna el valor del atributo parcial.

setParcial(unString)

Setea un valor al atributo parcial.

getNumero()

Retorna el valor del atributo numero.

setNumero(unString)

Setea un valor al atributo numero.

getDescripcion()

Retorna el valor del atributo descripcion.

setDescripcion(unString)

Setea un valor al atributo descripcion.

getPorcentajeNota()

Retorna el valor del atributo porcentajeNota.

setPorcentajeNota(unString)

Setea un valor al atributo porcentajeNota.

getCalificacion()

Retorna el valor del atributo calificacion.

setCalificacion(unString)

Setea un valor al atributo calificacion.

getParalelo()

Retorna la instancia del paralelo asignado en esta instancia.

setParalelo(anEspolParalelo)

Asigna anEspolParalelo al atributo paralelo.

`eliminarCalificacionesDeAlumnos()`
 Elimina los registros de esta política de calificación.

`calcularPromedioParcial()`
 Elimina las calificaciones de estudiantes ingresadas con esta política y recalcula el promedio del parcial.

Notas:

Nombre: **EspolPoliticaDelProfesor**

Descripción:
 Política definida para el curso.

Estados:
 (Ninguno)

Relaciones:
 Superclase: EspolElementoDelPeriodoAcademico
 Subclases: (Ninguna)
 Asociaciones: EspolParalelo

Variables globales:
 (Ninguna)

Variables de instancia:

`numero`
 Número de la política.

`título`
 Título de la política.

`tipoArchivo`
 Tipo de archivo que contiene el detalle del anuncio.
 D -> Documento (archivo), T -> Texto

`nombreArchivo`
 Nombre de archivo que contiene el detalle del anuncio.

`paralelo`
 Instancia de EspolParalelo.

Métodos de clase:

`eliminarRowsDelParalelo(paralelo)`
 Elimina los registros de un paralelo.

Métodos de instancia:

`__init__()`
 Inicializa las variables de instancia del objeto en el momento en el que este es creado.

rowsDeSolicitudesPorHorario(unRecurso, unaPersona, semanaDia, horaInicial)
Retorna una colección de las solicitudes que están hechas para un horario específico.

Métodos de instancia:

__init__()
Inicializa las variables de instancia del objeto en el momento en el que este es creado.

getCodigoUnidad()
Retorna el valor del atributo codigoUnidad.

setCodigoUnidad(unString)
Setea un valor al atributo codigoUnidad.

getCodigoTipoRecurso()
Retorna el valor del atributo codigoTipoRecurso.

setCodigoTipoRecurso(unString)
Setea un valor al atributo codigoTipoRecurso.

getCodigoRecurso()
Retorna el valor del atributo codigoRecurso.

setCodigoRecurso(unString)
Setea un valor al atributo codigoRecurso.

getTipoidentificacionSolicitante()
Retorna el valor del atributo tipoidentificacionSolicitante.

setTipoidentificacionSolicitante(unString)
Setea un valor al atributo tipoidentificacionSolicitante.

getNumeroIdentificacionSolicitante()
Retorna el valor del atributo numeroIdentificacionSolicitante.

setNumeroIdentificacionSolicitante(unString)
Setea un valor al atributo numeroIdentificacionSolicitante.

getIdentificacionSolicitante()
Retorna el tipo y número de identificación del solicitante concatenados.

getNumero()
Retorna el valor del atributo numero.

setNumero(unString)
Setea un valor al atributo numero.

getHora()
Retorna el valor del atributo hora.

setHora(unString)
Setea un valor al atributo hora.

`getEstado()`
Retorna el valor del atributo estado.

`setEstado(unString)`
Setea un valor al atributo estado.

`getFechaAprobacion()`
Retorna el valor del atributo fechaAprobacion.

`setFechaAprobacion(unString)`
Setea un valor al atributo fechaAprobacion.

`getObservacion()`
Retorna el valor del atributo observacion.

`setObservacion(unString)`
Setea un valor al atributo observacion.

`getEstaEnEspera()`
Retorna true si el objeto esta en espera.

`setEnEspera()`
Cambia a estado de espera al objeto.

`getEstaAprobado()`
Retorna true si el objeto esta aprobado.

`setAprobado()`
Cambia a estado aprobado al objeto.

`getNombreEstado()`
Retorna el nombre del estado en el que se encuentra la solicitud.

`getRecurso()`
Retorna la instancia del recurso asignado en esta instancia.

`setRecurso(anEspolRecurso)`
Asigna anEspolRecurso al atributo recurso.

`getSolicitante()`
Retorna la instancia del solicitante asignado.

`setSolicitante(anEspolPersona)`
Asigna anEspolPersona al atributo solicitante.

`getDetallesDeSolicitudDeRecurso()`
Retorna la instancias de detallesDeSolicitudDeRecurso que pertenecen a esta instancia.

`setDetallesDeSolicitudDeRecurso(aColection)`
Asigna aColection al atributo detallesDeSolicitudDeRecurso.

`tieneDetallesEntregadosYDevueltos()`

Devuelve true si la solicitud tiene detalles de recursos con estado entregado o devuelto.

Nombre: EspolTarea

Descripción:

Tarea publicada que los estudiantes deben responder.

Estados:

(Ninguno)

Relaciones:

Superclase: EspolElementoDelPeriodoAcademico
 Subclases: (Ninguna)
 Asociaciones: EspolParalelo
 EspolRespuestaDeTarea

Variables globales:

(Ninguna)

Variables de instancia:

numero

Número de la tarea.

fechaEnvio

Fecha de envio de la tarea.

fechaEntrega

Fecha máxima de entrega de las respuestas.

titulo

Título de la tarea.

tipoArchivo

Tipo de archivo que contiene el detalle del anuncio.
 D -> Documento (archivo), T -> Texto

nombreArchivo

Nombre de archivo que contiene el detalle del anuncio.

paralelo

Instancia de EspolParalelo.

migrado

Indica si el evento del día de calendario ha sido migrado o no.

respuestasDeTarea

Arreglo de instancias de EspolRespuestaDeTarea.

Métodos de clase:

fechaDeUltimaTareaEnParalelo(paralelo)

Devuelve la fecha de la última tarea publicada en el paralelo recibido como parámetro.

eliminarRowsDelParalelo(paralelo)

Elimina los registros de un paralelo.

Métodos de instancia:

__init__()

Inicializa las variables de instancia del objeto en el momento en el que este es creado.

getNumero()

Retorna el valor del atributo numero.

setNumero(unString)

Setea un valor al atributo numero.

getFechaEnvio()

Retorna el valor del atributo fechaEnvio.

setFechaEnvio(unString)

Setea un valor al atributo fechaEnvio.

getFechaEntrega()

Retorna el valor del atributo fechaEntrega.

setFechaEntrega(unString)

Setea un valor al atributo fechaEntrega.

getTitulo()

Retorna el valor del atributo titulo.

setTitulo(unString)

Setea un valor al atributo titulo.

getMigrado()

Retorna el valor del atributo migrado.

setMigrado(unString)

Setea un valor al atributo migrado.

setEsMigrado()

Setea un valor al atributo migrado.

setNoEsMigrado()

Setea un valor al atributo migrado.

getTipoArchivo()

Retorna el valor del atributo tipoArchivo.

setTipoArchivo(unString)

Setea un valor al atributo tipoArchivo.

getNombreArchivo()
 Retorna el valor del atributo nombreArchivo.

setNombreArchivo(unString)
 Setea un valor al atributo nombreArchivo.

getRespuestasDeTarea()
 Retorna la instancias de respuestasDeTarea que pertenecen a esta instancia.

setRespuestasDeTarea(aColection)
 Asigna aColection al atributo respuestasDeTareas.

getParalelo()
 Retorna la instancia del paralelo asignado en esta instancia.

setParalelo(anEspolParalelo)
 Asigna anEspolParalelo al atributo paralelo.

tipoObjeto()
 Siempre retorna la letra "T" para indicar que éste objeto es una tarea.

getProximoNombreParaArchivo(unNumero)
 Retorna el nombre que deberá ser asignado para el próximo archivo texto a grabar.

Notas:

Nombre: **EspolTemaDiscusion**

Descripción:
 Mensaje publicado en el foro de discusión.

Estados:
 (Ninguno)

Relaciones:

Superclase:	EspolElementoDelPeriodoAcademico
Subclases:	(Ninguna)
Asociaciones:	EspolParalelo

Variables globales:
 (Ninguna)

Variables de instancia:

numero	Número que identifica al mensaje.
nivel	Nivel al que pertenece.

secuencial
Secuencia dentro del nivel en el que se encuentra este mensaje.

tiempoEnvio
Fecha y hora del momento en el que el mensaje es publicado.

tiempoEnvioDetalle
Fecha y hora de envío de la última respuesta recibida en un hilo de mensaje.

padre
Instancia de EspolTemaDiscusión que contiene a este mensaje.

titulo
Título del mensaje.

tipoArchivo
Tipo de archivo que contiene el detalle del anuncio.
D -> Documento (archivo), T -> Texto, A -> Ambos

nombreArchivo
Nombre de archivo que contiene el detalle del anuncio.

tipoidentificacionPropietario
Tipo de identificación de la persona que genero este mensaje.

numeroidentificacionPropietario
Número de identificación de la persona que genero este mensaje.

paralelo
Instancia de EspolParalelo.

Métodos de clase:

fechaDeUltimoMensajeDeForoEnParalelo(paralelo)
Devuelve la fecha del ultimo mensaje publicado en el foro de discusión del paralelo recibido como parámetro.

Métodos de instancia:

__init__()
Inicializa las variables de instancia del objeto en el momento en el que este es creado.

getNumero()
Retorna el valor del atributo numero.

setNumero(unString)
Setea un valor al atributo numero.

getNivel()
Retorna el valor del atributo nivel.

setNivel(unString)
Setea un valor al atributo nivel.

getSecuencial()
Retorna el valor del atributo secuencial.

setSecuencial(unString)
Setea un valor al atributo secuencial.

getTiempoEnvio()
Retorna el valor del atributo tiempoEnvio.

setTiempoEnvio(unString)
Setea un valor al atributo tiempoEnvio.

getTiempoEnvioDetalle()
Retorna el valor del atributo tiempoEnvioDetalle.

setTiempoEnvioDetalle(unString)
Setea un valor al atributo tiempoEnvioDetalle.

getPadre()
Retorna el valor del atributo padre.

setPadre(unString)
Setea un valor al atributo padre.

getTitulo()
Retorna el valor del atributo titulo.

setTitulo(unString)
Setea un valor al atributo titulo.

getTipoidentificacionPropietario()
Retorna el valor del atributo.

setTipoidentificacionPropietario(unString)
Setea un valor al atributo tipoidentificacionPropietario.

getNumeroIdentificacionPropietario()
Retorna el valor del atributo.

setNumeroIdentificacionPropietario(unString)
Setea un valor al atributo numeroIdentificacionPropietario.

getTipoArchivo()
Retorna el valor del atributo tipoArchivo.

setTipoArchivo(unString)
Setea un valor al atributo tipoArchivo.

getNombreArchivo()
Retorna el valor del atributo nombreArchivo.

setNombreArchivo(unString)
Setea un valor al atributo nombreArchivo.

getParalelo()
Retorna la instancia del paralelo asignado en esta instancia.

`setParalelo(anEspolParalelo)`
Asigna `anEspolParalelo` al atributo `paralelo`.

`getNombrePropietario()`
Retorna la instancia de la persona que lo envió.

`getUsuarioPropietario()`
Retorna el usuario de la persona que envió el mensaje.

`tipoObjeto()`
Siempre retorna la letra "M" para indicar que éste objeto es un mensaje.

`datosMensaje()`
Devuelve un arreglo con la fecha, hora y número de respuestas para ese mensaje.

`datosTiempo()`
Devuelve un arreglo con la fecha, hora de envío.

`mensajesHijos()`
Retorna la instancias de los mensajes hijos.

`getProximoNombreParaArchivo(unNumero)`
Retorna el nombre que deberá ser asignado al próximo archivo texto a grabar.

Notas:**Nombre:** EspolTipoRecurso**Descripción:**
Tipo de recurso que clasifica a cada uno de los recursos físicos existentes.**Estados:**
Activo (A)
Inactivo (I)**Relaciones:**
Superclase: ObjetoPersistente
Subclases: (Ninguna)
Asociaciones: (Ninguna)**Variables globales:**
(Ninguna)**Variables de instancia:**
`codigo`
Código del tipo de recurso.

`nombre`

Nombre del tipo de recurso.

estado

Estado del tipo de recurso.

Métodos de clase:

writeTiposRecursosMigrationFile(ruta, nombreArchivo, contenido, delimitador)

Utilizado para el proceso de migración. Escribe el archivo depurado (sin líneas que produzcan registros duplicados) de migración de los tipos de recursos en el servidor.

ruta -> nombre del directorio en el que debe grabarse.

nombreArchivo -> nombre del archivo que debe ser grabado.

contenido -> líneas que contiene el archivo, las mismas que serán analizadas para determinar si provocarían error por registro duplicado en la migración.

delimitador -> caracter separador de campos en cada línea.

tiposDeRecursosOrdenadosPorNombre()

Retorna las instancias de tipos de recursos ordenadas por nombre.

Métodos de instancia:

__init__()

Inicializa las variables de instancia del objeto en el momento en el que este es creado.

getCodigo()

Retorna el valor del atributo codigo.

setCodigo(unString)

Setea un valor al atributo codigo.

getNombre()

Retorna el valor del atributo nombre.

setNombre(unString)

Setea un valor al atributo nombre.

getEstado()

Retorna el valor del atributo estado.

setEstado(unString)

Setea un valor al atributo estado.

setActivo()

Cambia el estado de la instancia a activo.

setInactivo()

Cambia el estado de la instancia a inactivo.

tieneRecursos()

Devuelve true si el tipo de recurso tiene recursos asociados.

Notas:

Nombre: EspolUnidadAcademica

Descripción:

Unidad académica, que conoce de los recursos físicos que tiene.

Estados:

Activa (A)
Inactiva (I)

Relaciones:

Superclase: ObjetoPersistente
Subclases: (Ninguna)
Asociaciones: EspolMateria
EspolPersona
EspolServidor

Variables globales:

(Ninguna)

Variables de instancia:

codigo
Código que identifica a la unidad académica.

nombre
Nombre de la unidad académica.

estado
Estado de la unidad.

codigoServidor
Código del servidor de autenticación del administrador de la unidad.

usuario
Nombre de usuario del administrador.

administrador
Instancia de EspolEmpleado.

materias
Arreglo de instancias de EspolMateria.

servidor
Instancia de EspolServidor.

Métodos de clase:

writeUnidadesMigrationFile(ruta, nombreArchivo, contenido, delimitador)
Utilizado para el proceso de migración. Escribe el archivo depurado (sin líneas que produzcan registros duplicados) de migración de las unidades académicas en el servidor.
ruta -> nombre del directorio en el que debe grabarse.
nombreArchivo -> nombre del archivo que debe ser grabado.
contenido -> líneas que contiene el archivo, las mismas que serán analizadas para determinar si provocarían error por registro duplicado en la migración.
delimitador -> caracter separador de campos en cada línea.

`instanciasOrdenadasPorNombreDeObjeto(objeto)`
Retorna las instancias de unidades ordenadas por nombre.

`unidadesOrdenadasPorNombre()`
Retorna las instancias de unidades ordenadas por nombre.

`unidadesDeUsuarioYCodigoServidor(unUsuario, unCodigoServidor, todas)`
Retorna las instancias de unidades ordenadas por nombre que tengan como administrador a unUsuario de unCodigoServidor.

`unidadesParaProfesorOrdenadasPorNombre(persona)`
Retorna las rows de unidades para un empleado (profesor-unidades de las materias que dicta), (alumno - unidades de las materias que recibe).

`unidadesParaAlumnoOrdenadasPorNombre(persona)`
Retorna las rows de unidades para un empleado (profesor-unidades de las materias que dicta), (alumno - unidades de las materias que recibe).

Métodos de instancia:

`__init__()`
Inicializa las variables de instancia del objeto en el momento en el que este es creado.

`getCodigo()`
Retorna el valor del atributo codigo.

`setCodigo(unString)`
Setea un valor al atributo codigo.

`getNombre()`
Retorna el valor del atributo nombre.

`setNombre(unString)`
Setea un valor al atributo nombre.

`getEstado()`
Retorna el valor del atributo estado.

`setEstado(unString)`
Setea un valor al atributo estado.

`getCodigoServidor()`
Retorna el valor del atributo codigoServidor.

`setCodigoServidor(unString)`
Setea un valor al atributo codigoServidor.

`getUsuario()`
Retorna el valor del atributo usuario.

`setUsuario(unString)`
Setea un valor al atributo usuario.

`setActivo()`

Cambia a activa la instancia.

setInactivo()

Cambia a inactiva la instancia.

getAdministrador()

Retorna la instancia del administrador asignado.

setAdministrador(anEspolPersona)

Asigna anEspolPersona al atributo administrador.

getMaterias()

Retorna la instancias de materias que pertenecen a esta instancia.

setMaterias(aCollection)

Asigna aCollection al atributo materias.

getServidor()

Retorna la instancia del servidor asignado.

setServidor(anEspolServidor)

Asigna anEspolServidor al atributo servidor.

tieneRecursos()

Devuelve true si la unidad tiene recursos asociados.

tieneMaterias()

Devuelve true si la unidad tiene materias asociadas.

puedeSerEliminada()

Devuelve true si la unidad puede ser eliminada.

Notas:

Nombre: EspolVario

Descripción:

Recurso académico, publicado en forma de enlace o archivo.

Estados:

(Ninguno)

Relaciones:

Superclase: EspolElementoDelPeriodoAcademico

Subclases: (Ninguna)

Asociaciones: EspolParalelo

Variables globales:

(Ninguna)

Variables de instancia:

numero
Número que identifica al recurso.

tipo
Tipo de recurso. Puede ser:
P – Artículo, E – Enlace, S – Archivo, O - Otros

titulo
Título del recurso.

nombreArchivo
Nombre del archivo asociado al recurso.

tipoidentificacionPropietario
Tipo de identificación de la persona que publicó el recurso.

numeroidentificacionPropietario
Número de identificación de la persona que publicó el recurso.

descripcion
Descripción del recurso.

paralelo
Instancia de EspolParalelo, al que pertenece el recurso.

Métodos de clase:

(Ninguno)

Métodos de instancia:

__init__()
Inicializa las variables de instancia del objeto en el momento en el que este es creado.

getNumero()
Retorna el valor del atributo numero.

setNumero(unString)
Setea un valor al atributo numero.

getTipo()
Retorna el valor del atributo tipo.

setTipo(unString)
Setea un valor al atributo tipo.

getTitulo()
Retorna el valor del atributo titulo.

setTitulo(unString)
Setea un valor al atributo titulo.

getNombreArchivo()
Retorna el valor del atributo nombreArchivo.

`setNombreArchivo(unString)`
Setea un valor al atributo `nombreArchivo`.

`getTipoidentificacionPropietario()`
Retorna el valor del atributo.

`setTipoidentificacionPropietario(unString)`
Setea un valor al atributo `tipoidentificacionPropietario`.

`getNumeroIdentificacionPropietario()`
Retorna el valor del atributo `numeroIdentificacionPropietario`.

`setNumeroIdentificacionPropietario(unString)`
Setea un valor al atributo `numeroIdentificacionPropietario`.

`getDescripcion()`
Retorna el valor del atributo `descripcion`.

`setDescripcion(unString)`
Setea un valor al atributo `descripcion`.

`getParalelo()`
Retorna la instancia del paralelo asignado en esta instancia.

`setParalelo(anEspolParalelo)`
Asigna un `anEspolParalelo` al atributo `paralelo`.

`esDelProfesor()`
Retorna `true` si el recurso fue publicado por el profesor.

`getPropietario()`
Retorna el nombre de la persona que publicó el recurso.

`getNombrePropietario()`
Retorna el nombre de la persona que publicó el recurso.

`tipoObjeto()`
Siempre retorna la letra "R" para indicar que éste objeto es un recurso académico.

`nombreTipo()`
Retorna el nombre del tipo de recurso.

`getProximoNombreParaArchivo(unNumero)`
Retorna el nombre que deberá ser asignado al próximo archivo texto a grabar.

Notas:

Nombre: File

Descripción:

Utilizado para verificar el tamaño de los archivos que se desean publicar en el sistema.

Estados:

(Ninguno)

Relaciones:

Superclase: (Ninguna)

Subclases: (Ninguna)

Asociaciones: (Ninguna)

Variables globales:

(Ninguna)

Variables de instancia:

Métodos de clase:

bytesArchivo(ruta, nombre)
Retorna el tamaño de un archivo.

tamanoArchivoPorPersona(persona)
Retorna el tamaño de archivo que tiene asignado una persona.

validaTamanoArchivo(ruta, nombre, persona)
Verifica que el tamaño del archivo no sobrepase al tamaño permitido para la persona recibida como parámetro.

valueTamanoArchivo(value, type)
Retorna el valor a grabar en la base para el tamaño de un archivo, tanto de profesor, como de alumno.

Métodos de instancia:

(Ninguno)

Notas:

Esta es una clase abstracta.

Nombre: HtmlAPI

Descripción:

Responde sobre cualquiera de los datos de configuración del sistema, necesarios para la generación de las páginas HTML.

Estados:

(Ninguno)

Relaciones:

Superclase: (Ninguna)
Subclases: (Ninguna)
Asociaciones: (Ninguna)

Variables globales:

(Ninguna)

Variables de instancia:

dirNameConf = '/home/httpd/cgi-bin/academic/conf'
Dirección en donde se encuentra el archivo de configuración del sistema.
fileNameConf
Nombre del archivo de configuración del sistema.

Métodos de clase:

(Ninguno)

Métodos de instancia:

findValueInStringLine(parameter, stringLine)
Retorna el valor que tiene asignado un parámetro en una línea de texto.

takeValidValue(cadena)
De una tupla clave y valor, retorna el valor.

lineIsValid(line)
Devuelve true si la línea es válida. Esto es, si no es blanco o un comentario (#).

nombreFmlPagina(nombreCGI)
Retorna el nombre del fml asociado a un CGI, con el que se debe contruir la página HTML.

findValueInFileWithKey(parameter, file, key)
Retorna el valor de una palabra existente en un archivo en la línea que contenga el parámetro recibido.

replaceValueInFileWithKey(parameter, file, key, nuevoValor)
Reemplaza el valor de una variable en un archivo por otro

replaceLinesInFileWithKey(key, file, nuevasLineas)
Reemplaza la línea que tiene el key en el file por las nuevasLineas

replaceValidValue(cadena, nuevoValor)
Reemplaza el valor en una tupla clave-valor.
cadena -> Clave y valor separados por igual.
nuevoValor -> nuevo valor que debe ser incluido en la cadena.

fmlDefault()
Retorna el directorio en el que se encuentran todos los archivos de formato originales.

workPath()
Retorna el directorio de trabajo del sistema.

confPathUser()

Retorna el nombre del directorio en el que se encuentran todos los archivos de formato del usuario.

`dirFmlUser()`

Retorna la ruta del directorio en el que se encuentran todos los archivos de formato del usuario.

`dirPersUser()`

Retorna el directorio personal del usuario.

`fileNameTuplas()`

Retorna el nombre del archivo de tuplas, en el cual se definen los nombres de los CGIs vs. archivos de formato.

Notas:

Nombre: HtmlError

Descripción:

Genera el código HTML de una página de error.

Estados:

(Ninguno)

Relaciones:

Superclase: (Ninguna)

Subclases: (Ninguna)

Asociaciones: (Ninguna)

Variables globales:

(Ninguna)

Variables de instancia:

Métodos de clase:

(Ninguno)

Métodos de instancia:

`__init__()`

Inicializa las variables de instancia del objeto en el momento en el que este es creado.

`errorRegistroDuplicado(entidad, entCodigo, entNombre)`

Genera el código con un mensaje de error en los casos en que se intenta insertar un objeto que produce un registro duplicado.

`errorOperacionNoPermitidaEnMateria()`

Genera el código con un mensaje de error en los casos en los que se intenta realizar alguna operación en una materia sobre la cual el usuario no tiene autorización.

`errorOperacionNoPermitida()`

Genera el código con un mensaje de error en los casos en los que el usuario desea acceder a una opción no permitida para su perfil.

`errorAccesoNoPermitido(accion)`

Genera el código con un mensaje de error en los casos en los que la autenticación del usuario no fue válida.

`errorConexionCerrada(accion)`

Genera el código con un mensaje de error en los casos en los que la conexión ha sido cerrada.

`alertaAcceso(accion)`

Genera el código con un mensaje de advertencia, informando que han sido cerradas otras conexiones abiertas por el usuario.

`errorRegistroAsociado(entidad, entCodigo, entNombre)`

Genera el código con un mensaje de error en los casos en los que se desea eliminar un objeto que tiene información relacionada.

`tiempoExpirado()`

Genera el código con un mensaje de advertencia en los casos en los que ha expirado el tiempo de los cookies y es necesario ingresar nuevamente el nombre de usuario y contraseña.

Notas:

Nombre: HtmlMetaformato

Descripción:

Genera el código HTML de las páginas del sistema, construidas en base al metaformato.

Estados:

(Ninguno)

Relaciones:

Superclase: (Ninguna)

Subclases: (Ninguna)

Asociaciones: (Ninguna)

Variables globales:

(Ninguna)

Variables de instancia:

globalPathFig

Nombre del directorio en el que se encuentran todas las figuras e iconos.

cgiName

Nombre del CGI que genera el contenido de la página y que invoca la construcción de la misma.

fmlPagina

Nombre del archivo de formato de la página que será construida con el CGI.

tipoLetra

Tipo de letra a aplicarse en el contenido de la página.

Métodos de clase:

(Ninguno)

Métodos de instancia:

__init__()

Inicializa las variables de instancia del objeto en el momento en el que este es creado.

buildPage(contenido, parametros)

Construye la página HTML a partir de los parametros y del contenido recibidos.

tagInicioTexto(colorLink)

Genera código html del inicio del tag para incluir texto con el formato definido en los fml.

tagFinTexto()

Genera código html para cerrar el tag de texto con las características que fueron definidas al abrirlo.

colorHeader()

Genera código HTML para formar los nombres de las columnas en las tablas.

colorBody()

Genera código HTML para definir el color a utilizarse en el contenido de las tablas.

colorConsulta()

Genera código HTML para definir el color a utilizarse en las tablas de consulta o resultado de la operación de un objeto específico.

colorConsultaForo()

Genera código HTML para definir el color a utilizarse en el foro de discusión para identificar al mensaje seleccionado.

formaTituloDeContenido(titulo)

Genera código HTML para darle formato al título del contenido.

formaTituloGeneral(titulo)

Genera código HTML para darle formato al título general de la página.

- `imprimirEncabezado()`
Genera código HTML que forma el encabezado, considerando los archivos de formato.
- `backGroundSchema()`
Genera código HTML que forma el background (con color o figura), considerando los archivos de formato.
- `imprimirFinal()`
Genera el código para cerrar la codificación del body y la página HTML.
- `imprimirCabecera(cabeceraFileName, cabEstilo, altoAncho)`
Genera código HTML que forma la cabecera vertical, considerando los archivos de formato.
- `imprimirTitulo(file)`
Genera código HTML que forma el título de la página, considerando los archivos de formato.
- `imprimirDatos(file)`
Genera código HTML que presenta la información del usuario conectado en la página, considerando los archivos de formato.
- `imprimirLogo(file, posIcono)`
Genera código HTML que presenta el log, considerando los archivos de formato.
- `imprimirMenu(file, cabeceraEstilo, indicaTab)`
Genera código HTML que forma el menu en la página, considerando los archivos de formato.
- `imprimirNEspacios(nEspacios)`
Genera código HTML par imprime líneas en blanco en la página.
- `htmlFotoDePersona(unaPersona)`
Retorna el código HTML para presentar la foto de una persona.
- `htmlTextoConReferencias(texto)`
Retorna el texto en formato HTML. Esto es, para el caso en el que el texto tenga especificado por <<>>, \$ es el separador de nombre y referencia, \ es el carácter de escape (\\$, \<<, \>>).

Notas:

Nombre: Mail

Descripción:

Se encarga de los envíos de mails a los integrantes del curso, cuando son publicados anuncios, recursos y tareas.

Estados:

(Ninguno)

Relaciones:

Superclase: (Ninguna)
 Subclases: (Ninguna)
 Asociaciones: (Ninguna)

VARIABLES GLOBALES:

(Ninguna)

VARIABLES DE INSTANCIA:**Métodos de clase:**

enviaMails(unObject, unPath, quienEnvia)
 Envía un mail a los estudiantes del paralelo que posean una dirección email.
 unObject -> Instancia del objeto del cual se desea informar.
 unPath -> Ruta del directorio en donde se encuentra el archivo texto del objeto del cual se desea informar.
 quienEnvia -> Nombre de la persona que envía el mensaje.

Métodos de instancia:

(Ninguno)

Notas:

Esta es una clase abstracta.

Nombre: ObjetoPersistente

Descripción:

Tiene como función realizar todas las operaciones a la base de datos. Es la clase de la que heredan todos los objetos persistentes.

Estados:

(Ninguno)

Relaciones:

Superclase: (Ninguna)
 Subclases: BotonDeModeloEnFml
 EspolCurriculum
 EspolDetalleDeSolicitudDeRecurso
 EspolElementoDelPeriodoAcademico
 EspolMateria
 EspolNombreDeParcial
 EspolParalelo
 EspolParametrosGenerales
 EspolPersona
 EspolProceso
 EspolRecurso
 EspolSemanaDeCalendarioDelPeriodo
 EspolServidor
 EspolSolicitudDeRecurso

Asociaciones: EspolTipoRecurso
 EspolUnidadAcademica
 (Ninguna)

Variables globales:
(Ninguna)

Variables de instancia:

Métodos de clase:

fechaDelSistema()
 Retorna la fecha actual, obtenida de la base de datos.

commit()
 Define un punto de consistencia en la base. Cierra la unidad de trabajo.

connection()
 Devuelve la conexion activa a la base de datos.

database()
 Devuelve la base de datos.

databaseManager()
 Devuelve el database manager activo.

databaseManagerClass()
 Devuelve el database manager activo.

databaseName()
 Devuelve el nombre de la base de datos.

disconnectFromDatabase()
 Desconecta la base de datos.

executeSelect(aSqlCommand)
 Retorna un conjunto de registros dada una sentencia select.

executeQuery(aSqlCommand)
 Ejectua una sentencia sql.

loadAll()
 Retorna todas las instancias de un tipo de objeto.

loadAllOrderedBy(anOrderByClause)
 Retorna en un OrderedCollection todas las instancias de la clase ordenadas
 segun la clausula anOrderByClause.
 anOrderByClause = 'ORDER BY UN_CAMPO'

loadFromId(id)
 Devuelve una instancia de esta clase leyendola desde la base de datos de acuer
 al id dado.

loadInstancesWhere(aClauseWhere)

Instancia los registros que cumplen con la sentencia Where enviada modificado para que buildNewInstanceFromRow no tenga que setear ultimoCambio.

rollback()

Cierra la unidad de Trabajo. Cambia el punto de consistencia de la base.

buildNewInstanceFromRow(anAbtRow)

Construye una nueva instancia a partir de la informacion de la row.

instancesFrom(anObject)

Devuelve una coleccion de instancias que corresponden a los atributos comunes entre esta clase y anObject.

instanceFromPK(newInstance, anObject)

Devuelve una instancia cuya clave primaria corresponda a la clave foranea de anObject.

instanceFromPKWithSelectorSuffix(newInstance, anObject, aString)

Devuelve una instancia cuya clave primaria corresponda a la clave foranea de anObject.

instancesFromSelectorSuffix(anObject, aSymbol)

Devuelve una coleccion de instancias que corresponden a los atributos comunes entre esta clase y anObject.

setPKInFrom(anObject, myInstance)

Asigna los valores de la primary key a los atributos correspondientes de anObject.

setPKInFromSelectorSuffix(anObject, myInstance, aSymbol)

Asigna los valores de la primary key a los atributos correspondientes de anObject.

Métodos de instancia:

__init__()

Inicializa las variables de instancia del objeto en el momento en el que este es creado.

getUltimoCambio()

Retorna el valor del atributo ultimoCambio.

setUltimoCambio(unTimestamp)

Setea un valor al atributo estado.

commit()

Define un punto de consistencia en la base. Cierra la unidad de Trabajo.

delete()

Se autoelimina de la base de datos.

instanceFromPK(anObject)

Devuelve una instancia cuya clave primaria corresponda a la clave foranea de anObject.

- `instanceFromPKWithSelectorSuffix(anObject, aString)`
Devuelve una instancia cuya clave primaria corresponda a la clave foranea de `anObject`.
- `instanceLikeMe()`
Devuelve una instancia de la tabla de la clase cuyos clave primaria es la de esta instancia.
- `instancesLikeMe()`
Devuelve un `objectResultTable` conteniendo todas las instancias de la tabla de la clase cuyos atributos coinciden con los atributos de esta instancia que no estan en nulo.
- `instancesLikeMeOrderedBy(anOrderByClause)`
Devuelve un `objectResultTable` conteniendo todas las instancias de la tabla de la clase cuyos atributos coinciden con los atributos de esta instancia que no estan en nulo, ordenados por `anOrderByClause`.
- `modificarEstadoDePersistencia()`
Coloca el estado de persistencia modificado.
- `nombreTabla()`
Devuelve un string con el nombre de la tabla que le corresponde a esta clase.
- `primaryKeys()`
Devuelve un diccionario con relational method elements que forman parte de la clave primaria del objeto.
- `rollback()`
Cierra la unidad de Trabajo. Cambia el punto de consistencia de la base.
- `isValidPK(anObject)`
Devuelve el resultado de la comparacion contra nil de los atributos que conforman la clave foranea en `anObject`.
- `isValidPKWithSelectorSuffix(anObject, aString)`
Devuelve el resultado de la comparacion contra nulo de los atributos que conforman la clave foranea en `anObject`.
- `setPKIn(anObject)`
Setea las `primaryKeys` en `anObject` asumiendo que `anObject` entiende los `foreingSetters` de mis `primaryKeys`. En otras palabras, `anObject` es un hijo de `self`.
- `loadWhere(aClauseWhere)`
Recupera los registros de la tabla de la clase en base a una sentencia `where`.
- `loadWhereForTable(aClauseWhere, aTableNameString)`
Recupera los registros de la tabla de la clase en base a una sentencia `where`.
- `store()`
Guarda el objeto en la base de datos.
- `storeConLeido()`

Guarda el objeto en la base de datos cuando este no existe.

storeConModificado()

Guarda el objeto en la base de datos cuando este ya existe.

insert()

Inserta el objeto en la tabla.

update()

Hace update del objeto en la tabla.

refreshUltimoCambioFor(sqlStatement)

Con el sqlStatement construye un select para actualizar el ultimoCambio. Pide el ultimoCambio asignado, al unitOfWork y lo resetea.

rowLikeMe()

Devuelve una row conteniendo una rows de la tabla de la clase cuyos clave primaria es la de esta instancia.

rowsLikeMe()

Devuelve un resultSet conteniendo todas las rows de la tabla de la clase cuyos atributos coinciden con los atributos de esta instancia que no estan en nulo.

Notas:

Nombre: PartesReusables

Descripción:

Genera código HTML de información que es presentada en la mayoría de las páginas HTML.

Estados:

(Ninguno)

Relaciones:

Superclase: (Ninguna)

Subclases: (Ninguna)

Asociaciones: (Ninguna)

Variables globales:

(Ninguna)

Variables de instancia:

Métodos de clase:

datosProfesorYTituloForm(instMF, cont, paralelo, titulo)

Concatena en la variable cont el código HTML que presenta información del profesor y paralelo con un título recibido como parámetro.

instMF -> Instancia de HtmlMetaformato.
 contenido -> Texto del contenido de la página HTML.
 paralelo -> Instancia de EspolParalelo.
 titulo -> Texto a presentar como titulo.

tituloConSeparadores(instMF, titulo)
 Devuelve el código HTML para presentar el titulo con una línea superior y una inferior.
 instMF -> Instancia de HtmlMetaformato.
 titulo -> Texto a presentar como titulo.

informacionPersonal(unObject)
 Recibe un objeto ya sea de empleado o de alumno, y retorna una cadena con el primer apellido y primer nombre del mismo.

Métodos de instancia:

Notas:

Esta es una clase abstracta.

Nombre: Tools

Descripción:

Clase abstracta que realiza tareas varias.

Estados:

(Ninguno)

Relaciones:

Superclase: (Ninguna)
 Subclases: (Ninguna)
 Asociaciones: (Ninguna)

Variables globales:

(Ninguna)

Variables de instancia:

(Ninguno)

Métodos de clase:

openWindow(ruta, nombreArchivo, gifConPath, contenido)
 Retorna la función para abrir un archivo en otra ventana.

concatenaConTipoObjeto(tipo, cadena)
 Concatena una cadena con el directorio del objeto respectivo.

rutaDelAlumno(pathWeb, paralelo, usuario, modo)
 Retorna la ruta del directorio de trabajo de un estudiante.

rutaFile(pathWeb, paralelo, tipo, modo)
 Retorna la ruta para que un archivo se abra, si modo es nulo, además necesita tipo para indicar si es anuncio, tarea, etc. Retorna una ruta para grabar

informacion, anuncios, tareas, etc. si modo es diferente de nulo, pero tambien necesita el tipo.

rutaDelProfesor(pathWeb, paralelo, raiz)

Retorna la ruta del directorio del profesor si raiz es nulo caso contrario retorna la ruta raiz.

nombreArchivoParaNavegador(nombre)

Retorna el nombre del archivo sin caracteres especiales. Estos son reemplazados por otros caracteres.

validaArchivo(objFile)

Retorna el nombre del archivo y el contenido del archivo en caso de que se haya ingresado un archivo válido.

writeFile(ruta, nombreArchivo, contenido, validaExistencia)

Escribe el contenido de un archivo en el servidor.

contenidoDeArchivoPublicoDeObjeto(rutaInicial, anObject, usuario)

Abrimos archivo tipo texto que contiene informacion relacionada al objeto recibido como parametro.

crearDirectoriosDeArchivoDeMigracion(unArchivo, delimitador, pathWeb)

Crea los directorios en base a los datos migrados de unArchivo.

crearDirectoriosDeTrabajoDelParalelo(pathWeb, anio, termino, mat, para)

Crea los directorios de trabajo en base a la informacion recibida.

crearDirectoriosDeTrabajoDelAlumno(pathWeb, anio, termino, mat, para, tipoidentifAlumno, nroidentifAlumno)

Crea los directorios de trabajo en base a la informacion recibida.

borrarArchivoVacio(ruta, nombreArchivo)

Borra los archivos que se guardaron pero no contienen informacion.

borrarDirectoriosDeTrabajoDelParalelo(pathWeb, anio, termino, mat, para)

Borra los directorios de trabajo de un paralelo.

borrarDirectoriosDeTrabajoDeIdentificacion(pathWeb, anio, termino, mat, para, identif)

Borra los directorios de trabajo de una identificacion.

crearDirectorioDeAnioTermino(pathWeb, anio, termino)

Crea Directorio del anio termino.

crearDirectorioDeMateria(pathWeb, anio, termino, mat)

Crea Directorio de la Materia.

crearDirectorioDeParalelo(pathWeb, anio, termino, mat, para)

Crea Directorio del Paralelo.

crearDirectorioPublicoEnParalelo(pathWeb, anio, termino, mat, para)

Crea Directorio Publico.

crearDirectorioProfesorEnParalelo(pathWeb, anio, termino, mat, para)
Crea Directorio del Profesor.

crearDirectorioDeAlumnoEnParalelo(pathWeb, anio, termino, mat, para, tipoIdentif, nroIdentif)
Crea Directorio del Alumno.

crearDirectoriosConfDeArchivoDeMigracion(unArchivo, delimitador, pathConfDef, pathFmlDef, pathConfUser, dirConfUser, dirFmlUser, dirPersUser)
Crea los directorios de configuracion en base a los datos migrados de unArchivo.

crearDirectoriosConfDeIdentificacion(identificacion, pathConfDef, pathFmlDef, pathConfUser, dirConfUser, dirFmlUser, dirPersUser)
Crea un directorio de configuracion en base a la identificacion.

borrarDirectoriosConfDeIdentificacion(identificacion, pathConfDef, pathFmlDef, pathConfUser, dirConfUser, dirFmlUser, dirPersUser)
Borra un directorio de configuracion en base a la identificacion.

rutaRelativa(rutaACambiar)
Devuelve la ruta relativa de rutaACambiar dependiendo de la ruta del directorio actual.

copiarArchivosDePeriodoAOtro(paraleloAnterior, paraleloActual, pathWeb, instancia)
Crea Directorio Publico.

Métodos de instancia:

(Ninguno)

Notas:

Esta es una clase abstracta.

Nombre: ToolsCookie

Descripción:

Devuelve información sobre los valores de las variables de sesión del sistema.

Estados:

(Ninguno)

Relaciones:

Superclase: (Ninguna)

Subclases: (Ninguna)

Asociaciones: (Ninguna)

Variables globales:

(Ninguna)

Variables de instancia:**Métodos de clase:**

valor(etiqueta)

Retorna el valor de la variable cuya etiqueta se recibe como parámetro.

returnProceso()

Retorna el proceso activo de la conexión.

personaLogoneada()

Devuelve la instancia de la persona que entro al sistema (esto es para uso del módulo de reservaciones).

validaConexion()

Esta funcion valida que un usuario este trabajando en el sistema mientras su conexion inicial este abierta.

calculaSegundos(tiempo)

Retorna el total de segundos para un timestamp enviado.

validaCookie()

Esta funcion valida que la persona que esta accesando sea la que se conecto al sistema. Ademas verifica el tiempo de vida del cookie.

borrarCookie()

Inicializa las variables de sesión.

Métodos de instancia:

(Ninguno)

Notas:

Esta es una clase abstracta.

APÉNDICE B

ESPECIFICACIONES DEL METALENGUAJE Y GUIAS PARA EL API PARA EL FORMATO GENERALIZADO DE PAGINAS HTML

por Guido Caicedo Rossi

Este artículo busca definir un mecanismo general para dar un formato común a las páginas HTML generadas por CGIs. El enfoque del mecanismo es de que este sea manipulable desde el servidor (independiente del browser, modular y completamente configurable).

Para que los CGIs de un sitio Web puedan generar un formato uniforme para todas las hojas existirán dos componentes: Archivos de formato utilizando un metalenguaje especial y un API (conjunto de funciones) a ser llamadas por los CGI para la generación de las hojas HTML basadas en los archivos de formatos.

El modelo que muestra la relación entre los diferentes componentes (incluyendo los archivos de configuración de los CGI) se muestra en la Fig. 1.

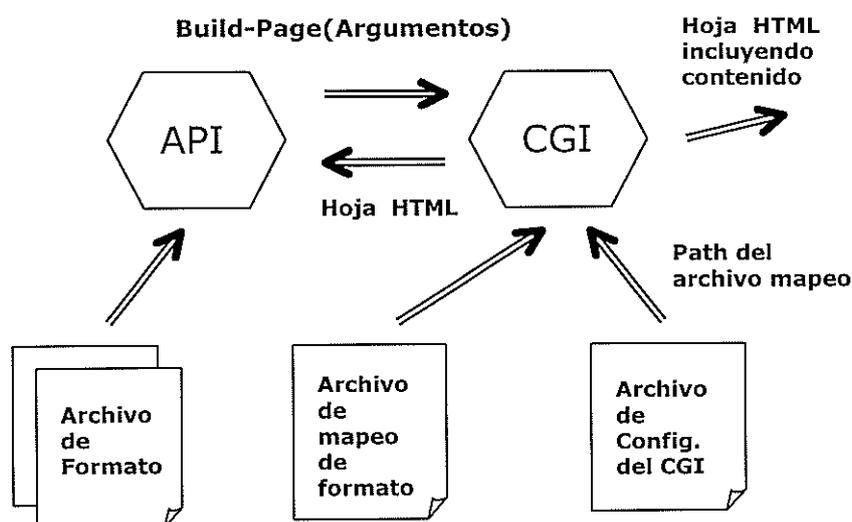


Figura 1. Relación entre los diferentes componentes del metaformato

La función build-page (o el nombre que se le dé en el API), invocada por el CGI, aglutina todas las funciones que construyen la apariencia de la hoja completa (exceptuando la sección con el contenido particular generado por el CGI). Sin embargo, el API debe de ser desarrollado de tal forma que las funciones para construir los diferentes componentes puedan ser llamados independientemente. Esto conlleva el beneficio de que los CGI también puedan hacer llamadas independientes a funciones del API, para aprovecharlas en darle formato al contenido particular. Esta idea se muestra en la Figura 2.

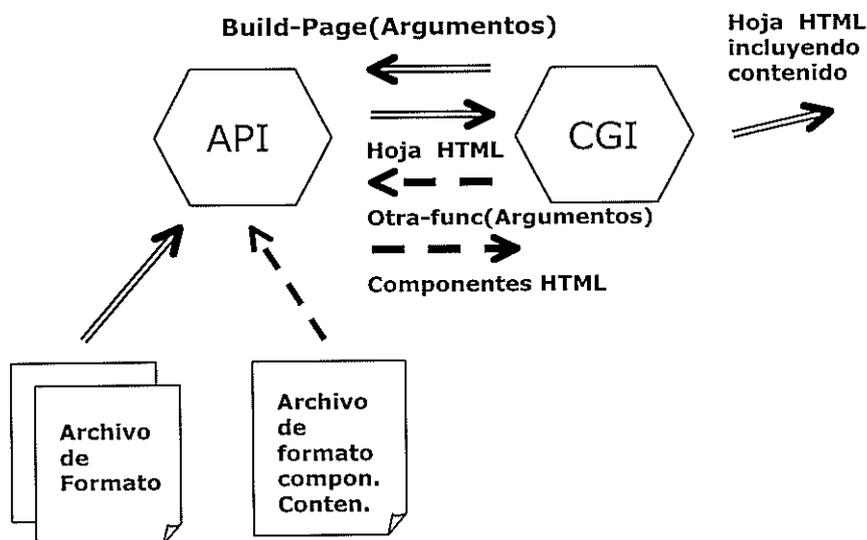


Figura 2. Llamadas al API

Las variables mínimas que deben ser pasadas por el CGI a las funciones del API son las siguientes:

- Path absoluto y nombre del archivo de formato a utilizar (usar la extensión fml para archivos de formato).
- Lista de valores a colocar en variables especificadas en el archivo de formato (ver sección "Manejo de variables libres" más adelante).
- Path absoluto de las figuras. Se especifica el directorio donde se encuentran los gifs o jpgs indicados en los archivos de formato. Si no es nulo, tiene propiedad sobre el path que se encuentre especificado en el archivo de formato.

Archivos de Formato y su Metalenguaje

El archivo de formato para un CGI, en realidad puede estar constituirse de varios archivos. Un archivo principal (<PAGINA>) que tiene los parámetros generales de la hoja, y archivos de componentes que pueden ser de <CABECERA>, <MENU> o <TITULO>. Todos los archivos de formato que pertenezcan a componentes de una PAGINA deben estar ubicados en el mismo directorio que el archivo de formato de la PAGINA.

A continuación se define brevemente el metalenguaje a utilizar en cada uno de los diferentes archivos:

1. PAGINA

Este es el tipo de archivo de formato principal. Básicamente describe la organización general de la hoja y algunos parámetros básicos. Existen 4 organizaciones generales:

Simple, Horizontal, Vertical y Combinada. La primera es una hoja simple HTML. Las otras tres subdividen las regiones de la página utilizando tablas HTML. En la figura 3 se muestra la disposición de cada hoja. Sin embargo, note que la línea de puntos indica que puede ir una cabecera y un pie antes y después del contenido que genera el CGI (lo que implica que aún en la organización simple de la hoja puede utilizarse una tabla de 2 o tres filas). Ninguna sección es obligatoria pero tampoco puede ir repetida.

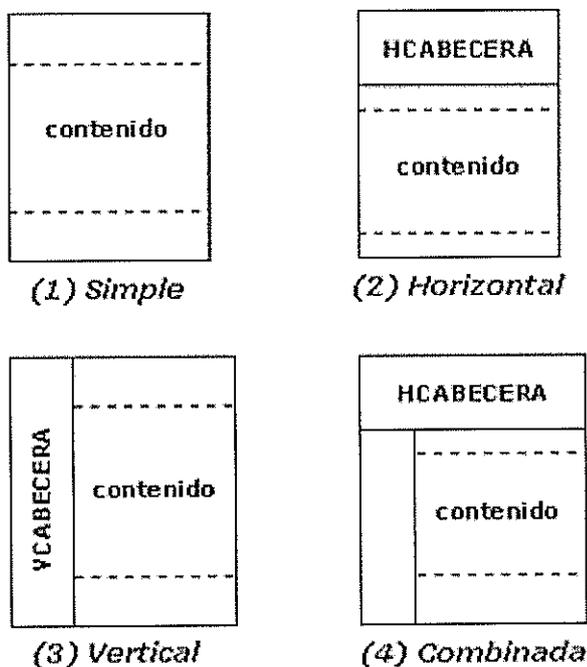


Figura 3. Modelos de páginas usando el metaformato

La sintaxis de un archivo de formato de página es la siguiente:

```

<PAGINA>
  <TITULO= texto en el título de la ventana del browser>
  <FONDO= color  FIG= figura para el fondo>
  <FONT= tipo de caracteres  TAM= tamaño  COLOR= color>
  <CLINK= color  CALINK= color  CVLINK= color>
  <PATHGIF= url donde se encuentran las figuras>
  <HCABECERA= Archivo de formato de cabecera  ALTO= alto de la celda en pixeles>
  <VCABECERA= Archivo de formato de cabecera  ANCHO= ancho de la celda en pixeles>
  <CABECERA= Archivo de formato de cabecera  ALTO= alto de la celda en pixeles>
  <PIE= Archivo de formato de cabecera  ALTO= alto de la celda en pixeles>
</PAGINA>

```

Este archivo de formato de página especifica el formato general de la hoja. El tag <FONDO> especifica parámetros para el fondo de la hoja (donde va el contenido generado por el CGI). El tag especifica el formato general de los caracteres en el contenido generado por el CGI (sección contenido en la figura 3). Sin embargo, esto no significa que el CGI no tiene control sobre el formato del contenido que genera. Simplemente es el valor default que se utiliza a menos que el CGI especifique parámetros diferentes para porciones del texto.

<PATHGIF> es el camino default para buscar todas las figuras. La función principal del API (build_page) pasa como argumento este path a las demás funciones. Sin embargo, los tags que especifican figuras pueden comenzar su propio path empezando con "http://" o simplemente "/" (este último relativo al website de donde cargo la hoja).

2. CABECERA

Tanto para el pie como para los tres tipos de cabeceras indicados en el archivo de formato de página, el archivo de formato de estos componentes se define mediante un archivo de formato de cabecera que va de acuerdo a la siguiente sintaxis:

```

<CABECERA>
  <TITULO= Archivo de formato de título>
  :
  :

```

```

<TITULO= .....>
.
<MENU= Archivo de formato de menú>
<MENU= .....>
.
.      En cualquier parte pueden ir menús o títulos
.
</CABECERA>

```

El formato básicamente especifica el orden en que se suceden los títulos, menús y líneas en blanco.

3. TITULO

Para los títulos especificados en el archivo de cabecera, se pueden usar dos alternativas de archivo de formato:

- Un archivo en HTML sin las directivas de HEAD y/o BODY y con el tag <TPROPIO> al inicio y </TPROPIO> al final (la extensión de este archivo será fml como en los demás).
- Un archivo con tags en el metalenguaje que indican los componentes del título.

Para la segunda alternativa, un título es básicamente una tabla con 1 fila y 3 columnas. En la columna central va el título y en los dos extremos dos iconos.

```

<TITULO>
  <FONDO= color  FIG= figura>
  <DICON= nombre de la figura  LINK= url  ANCHO= pixeles>
  <IICON= nombre de la figura  LINK= url  ANCHO= pixeles>
  <CONT= texto del contenido  FIG= nombre de la figura  LINK= url >
  <FONT= tipo de caracteres  TAM= tamaño  COLOR= color>
  <BORDE= pixeles  ESPCEL= pixeles  BORCEL= pixeles >
  <ALINEA= la tabla va en centro, izquierda o derecha>
</TITULO>

```

4. MENU

Al igual que en los títulos para el archivo de formato de un MENU, se puede usar un archivo de formato HTML (tags <MPROPIO> al inicio y </MPROPIO> al final) o el siguiente archivo de formato usando el metalenguaje:

```
<MENU>
  <ANCHO= ancho de la tabla en pixeles>
  <ALINEA= la tabla va en el centro, izquierda o derecha>
  <ALINBOT= cada botón va centrado, a la izquierda o a la derecha>
  <BORDE= pixeles  ESPCEL= pixeles  BORCEL= pixeles>
  <FONDO= color  FIG= figura>
  <FONT= tipo de caracteres  TAM= tamaño  COLOR= color>
  <BOTNOM= nombre de archivo de botones  EXT= extensión del archivo>
  <EXTESC= carácter adicional para nombre de botón escogido>
  <LISTABOT= números separados por coma, botones del menú>
  <ESCOGIDO>
    <NUMERO= número del botón escogido>
    <FONT= tipo de caracteres  TAM= tamaño  COLOR= color>
  </ESCOGIDO>
  <BOT= archivo de botones>
</MENU>
```

El menú especificado por este archivo de formato es básicamente una tabla donde cada elemento contendrá un botón. La mayoría de los tags especifican características generales de la tabla, los botones y los tags LISTABOT, ESCOGIDO especifican características particulares de la configuración de los botones. El archivo especificado por BOT, contiene tuplas con el siguiente formato:

"Contenido del texto botón", url del botón

La información de este archivo se considera más funcional que de formato, por lo que su contenido es administrado más bien en función del diseño de los CGI. El tag BOT puede especificar el path absoluto del archivo de botones, el cual no necesariamente se encuentra en el mismo directorio de los archivos de formato.

A continuación se aclararan detalles de la semántica de algunos de los tags:

<BOTNOM> y <EXTESC>

Para simplificar el manejo de nombres de archivos en caso de que se desee usar figuras para los botones en un menú, existe el tag <BOTNOM> que especifica el inicio del nombre del archivo de la figura. Para el botón 1 que irá en la primera celda de la tabla del menú se añadirá el número 1 al nombre, para el botón 2 el número 2 y así sucesivamente. El tag <EXTESC> representa caracteres añadidos al final del nombre del archivo de la figura del botón que por algún motivo se quiera mostrar con un archivo diferente (normalmente se usa para mostrar el botón actualmente escogido). Así es que si se usan figuras para los botones, normalmente se tendrán dos juegos de figuras: uno para cuando los botones se muestren en su forma normal y otro para cuando un botón debe ser mostrado en forma diferente.

<ESCOGIDO>

Especifica los parámetros que harán que un botón se vea diferente a los demás. El tag <NUMERO= ...> especifica cual es el botón escogido (normalmente este parámetro es una variable).

<LISTABOT>

Es una lista de números separados por coma (sin espacios) y encerrada entre comillas que indica las opciones del menú que deben ser presentadas y el orden. La idea de este tag es que sea utilizado cuando dinámicamente se quiere manejar que botones del menú serán presentados. Bajo este esquema en el archivo de formato se debe especificar una variable (ver la sección Manejo de Variables) Ej: <LISTABOT= "\$3"> cuyo valor será enviado por el CGI a las funciones del formato conteniendo la lista de botones a mostrar.

Esto trabaja también con el tag <NUMERO= ...> dentro del tag <ESCOGIDO> que le indicaría a las funciones de formato, cual es el botón que debe ser mostrado de manera diferente (si es que así se lo desea).

Cuando el tag <LISTABOT> no existe, se asume que se muestran todos los botones.

Consideraciones para algunos Tags

Todos los tags van entre comilla. Cualquier tag entre los dos principales que definen a un archivo puede no aparecer. En la mayoría, simplemente no se generará el código HTML que depende de ellos, excepto en los siguientes casos:

PATHGIF (dentro de página)

Si no aparece, la función principal del API la tomará del argumento pasado por el CGI. Si el CGI pasa un argumento nulo, se asumirá como path el mismo directorio del archivo de formato.

BOTNOM (dentro de menú)

Si no está esta directiva se asumirá que los botones no llevan figuras.

EXTESC (dentro de menú)

Si no está esta directiva se asumirá que un botón se marcará diferente pero mostrando sólo texto (no figuras).

ESCOGIDO (dentro de menú)

Si esta directiva no existe o está vacía o no existe el tag NUMERO con un valor válido, no se mostrará diferente ningún botón del menú.

LISTABOT (dentro de menú)

Si esta directiva no existe o está vacía o mal construida, se mostrarán todos los botones que estén declarados en el archivo de formato del menú.

BOT1 BOT2 (dentro de menú)

Si no existe un número de botón que esté especificado en LISTABOT, simplemente no saldrá ese botón.

Manejo de Variables Libres

Un CGI tiene la posibilidad de dinámicamente incluir información dentro del contenido de la hoja a generar en cualquier valor indicado por un tag. Para esto el archivo de formato contendrá símbolos que representarán variables y que serán reconocidos por el API que maneja los archivos de formato (funciones que procesan los archivos). Los símbolos estarán conformados por el carácter \$ seguido de un número secuencial. Ej:

```
<CONT= "Usuario: $1">
```

```
<CONT= "Usuario: $1" FIG= "$2" LINK= "/user/$3">
```

En los ejemplos anteriores, el CGI al llamar al API que maneja los formatos, le enviará los valores \$1, \$2 y \$3. Ud. tiene la libertad de definir en su API el mejor mecanismo que considere apropiado para recibir los argumentos del CGI.

Si hay un error en el archivo de formato y no se encuentra una variable necesaria o si hay más variables que valores, simplemente no se mostrarán en el primer caso y se mostrará el nombre de la variable en el segundo.

Si por cualquier motivo existe la necesidad de incluir el carácter \$ en el valor de un tag, se escribirá la secuencia \$\$.

NOTA: Las variables también podrán ser incluidas en secciones con formato propio basadas en código HTML.

Paso de Argumentos

En los menús o en general en cualquier enlace, puede haber llamadas a CGIs. Frecuentemente en esas llamadas es necesario incluir parámetros. Los archivos de formato no incluyen parámetros por lo tanto estos deberán ser colocados por el CGI que construye la hoja. El mecanismo es similar al del uso de variables descrito anteriormente. Simplemente se incluye la variable a continuación del enlace, y su valor será pasado por el CGI al construir la hoja. Al igual que el caso anterior, las variables también pueden ser incluidas en secciones con formato propio.

APÉNDICE D

MANUAL DEL ADMINISTRADOR

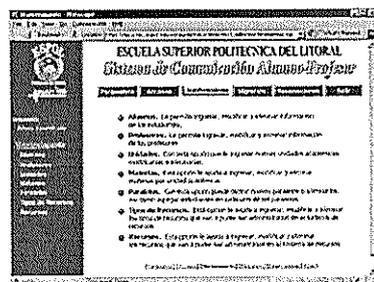
En Cursos Web existe el módulo de Administración, el mismo que será utilizado por los usuarios que tengan el perfil de Administrador General o Administrador de Unidad.

Administrador General. Se encargará de definir los parámetros generales del sistema (tales como las fechas de inicio y fin del período académico), definir quienes serán los administradores de unidad, definir las materias, profesores, alumnos, paralelos, integrantes de los cursos, etc.

Administrador de Unidad. Mantiene actualizada la información de las materias, profesores, alumnos, paralelos, integrantes del curso, pero solo de la unidad a la que esta asignado.

PARA EMPEZAR

Una vez que ha accedido al sistema verá la ventana de Mantenimiento, la cual será similar a la mostrada a continuación:



calificaciones, publicar calificaciones, agregar o eliminar estudiantes y reutilizar la información de cursos anteriores.

Estudiante. Persona que debe asistir al curso teniendo la responsabilidad de cumplir con las tareas asignadas por el profesor para aprobar la materia. Entre las operaciones que éste realiza están:

1. Consultar toda la información proporcionada por el profesor o ayudante.
2. Publicar recursos y las respuestas a las tareas.
3. Interactuar con los participantes de su curso a través del foro de discusión.
4. Efectuar reservaciones de los recursos académicos.
5. Actualizar su información personal.
6. Personalizar la apariencia del sistema basándose en plantillas predeterminadas.

Otras consideraciones que se debe tener en cuenta de acuerdo al perfil del usuario.

Profesor.

- Si en la opción de menú **Curso**, no aparecen todos los estudiantes de su curso usted puede adicionarlos a través de la opción **Agregar Alumno**.
- Si en la opción de menú **Curso**, aparecen listados estudiantes que no pertenecen a su curso usted puede eliminarlos del mismo, con la opción **Eliminar Alumno**.
- En todos los campos descripción usted puede definir enlaces entre el texto, de la siguiente forma:

{ } son los delimitadores del enlace.

\$ Separa al texto que desea que aparezca como enlace de la dirección del enlace. Por ejemplo, podría definir en la descripción de un anuncio:

"Para la próxima clase leer sobre {productos de software\$http://www.productos.com}."

- A través de la opción **Personalizar**, además de poder presentar su foto, usted puede definir otra información personal como: teléfono(s), dirección, email, sitio web, educación, experiencia laboral y hobbies.

Estudiante.

- Al responder una tarea de tipo texto usted puede definir enlaces de la siguiente forma:

{ } son los delimitadores del enlace.

\$ Separa al texto que desea que aparezca como enlace de la dirección del enlace. Por ejemplo, podría definir en la descripción de un anuncio:

"Para la próxima clase leer sobre {productos de software\$http://www.productos.com}."

- A través de la opción **Personalizar**, además de poder presentar su foto, usted puede definir otra información personal como: teléfono(s), dirección, email, sitio web, educación, experiencia laboral y hobbies.

PARA EMPEZAR

Antes de usar el sistema

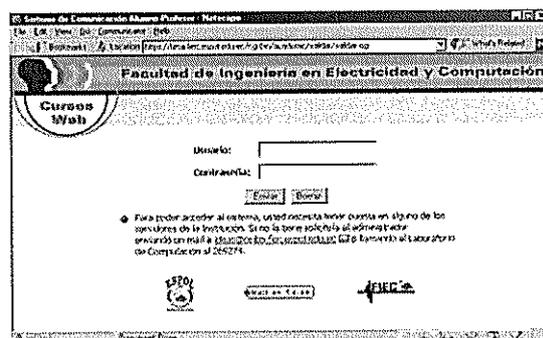
Para poder acceder al sistema, usted necesita tener cuenta en alguno de los servidores de la Institución, es decir un usuario (user) y una contraseña (password) con el cual poder identificarse, caso contrario no podrá ingresar al sistema.

Si todavía no tiene una cuenta

Si no tiene una cuenta, solicítela poniéndose en contacto con el Administrador del sistema

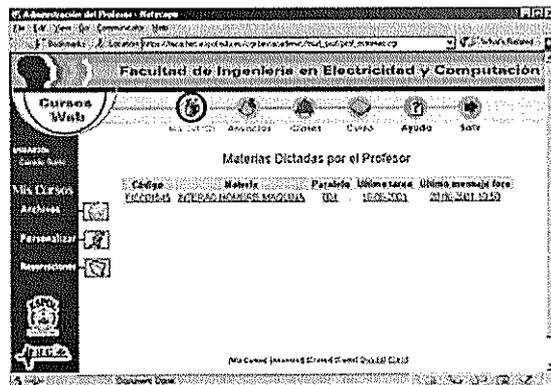
Si ya tiene una cuenta

Si tiene una cuenta en cualquier servidor de la institución, deberá ingresar su usuario y contraseña en la página de inicio de Cursos Web y presionar el botón enviar, luego de lo cual se encontrará en la ventana Principal (Mis Cursos).



SI EL USUARIO ES UN PROFESOR

Una vez que ha accedido al sistema verá la ventana Mis Cursos, la cual será similar a la mostrada a continuación:



En la parte izquierda de la ventana aparece el nombre del profesor que está usando el sistema, esta característica es igual para todas las ventanas del sistema.

En el centro de la página aparece una lista con las materias que el profesor está dictando en el presente período académico y desde donde podrá acceder a cada una de ellas para administrarlas y planificarlas.

En la lista aparece información de la materia, la cual incluye: el código de la materia, el nombre de la materia, el número del paralelo, la fecha de la última tarea, y la fecha del último mensaje del foro de discusión.

Dando click en cualquiera de los atributos mencionados anteriormente, usted accederá a la materia, pero con la diferencia de que:

- Si da click sobre el Código de la Materia, el Nombre de la Materia o el Número del paralelo de la Materia, entonces ingresará a la ventana de Anuncios de la materia.
- Si da click sobre la Fecha de la última Tarea, entonces ingresará a la ventana de Tareas de la materia.
- Si da click sobre la Fecha del último mensaje del Foro de Discusión, entonces ingresará a la ventana del Foro de la materia.

Cuando ingrese en una de las materias encontrará en el menú superior las opciones: **Anuncios** donde podrá administrar los anuncios de esa materia, **Clases**

donde podrá planificar toda la información a utilizar en esa materia, como tareas, políticas, clases, etc., **Curso** donde podrá administrar la información referente al curso tales como: alumnos, calificaciones. Además posee las opciones de **Mis Cursos** para regresar a la ventana principal, **Ayuda** para ver la ayuda del Sistema y **Salir** para salir del sistema y regresar a la ventana de ingreso al Sistema. Estas tres últimas opciones están disponibles en todas las pantallas del sistema.

En la ventana de **Mis Cursos**, en el submenú que se encuentra a la izquierda posee las opciones:

1. **Archivos** que permite reutilizar la información de los cursos que anteriormente ha dictado.
2. **Personalizar** que permite modificar sus Datos Personales y la configuración de las páginas HTML del sistema.
3. **Reservaciones** que permite realizar reservaciones de los recursos académicos, tales como: proyectores, aulas, etc.

➤ REUTILIZAR MATERIAL

La opción **Archivos** de la sección **Mis Cursos** le permite reutilizar la información de sus cursos anteriores, esto es muy útil cuando un profesor dicta una misma materia en varios semestres. La información que puede volver a publicar es básicamente toda la ingresada, tales como: Anuncios, Tareas, Calendarios Políticas, Información y Recursos.

Cuando la información sea reutilizada le aparecerá como material migrado en los casos que necesite realizar ciertas modificaciones, por ejemplo: en las tareas que necesitan la fecha de entrega.

Para utilizar esta opción realice lo siguiente:

1. Seleccione el paralelo que contiene la información que desea reutilizar.

2000-2

Código	Materia	Paralelo
FIE001545	INTERAC.HOMBRE MAQUINA	001

- De click en el botón **Aceptar**.

Ahora debe especificar cuál es la información que desea utilizar en el período actual:

- Tareas
- Calendarios
- Políticas
- Anuncios
- Información
- Recursos

- Especifique la información desea utilizar en el período actual.
- Haga click en el botón **Aceptar** para migrar la información, verá un mensaje de operación exitosa.
- Haga click en el botón **Aceptar** para regresar a a la ventana Principal.

➤ DATOS PERSONALES

La opción **Personalizar** de la sección **Mis Cursos** le permite mantener actualizada su información personal la cual será vista por sus alumnos y les puede ser de mucha utilidad en caso de querer comunicarse con usted directamente.

Su nombre y el tipo de identificación (Número de cédula, Pasaporte) son ingresados por el administrador del sistema.

Los datos que deben ser llenados por usted son:

- País del que proviene.
- Provincia de la que proviene.

3. Dirección domiciliaria (puede optar por mostrarla en la consulta seleccionando la casilla de verificación).

Presentar dirección en consulta

4. Números telefónicos (puede optar por mostrarlos en la consulta seleccionando la casilla de verificación).

5. Dirección e-mail.
6. Sitio web (si lo tiene).
7. Educación.
8. Experiencia Laboral.
9. Hobbies.
10. Fotografía (imagen en formato .jpg o .gif).
11. Servidor de Autenticación (*Sirve para verificar su usuario y su clave de acceso al sistema*).
12. Usuario (*El usuario que posea en el servidor de autenticación. Si escogió Servidor Local, usted define el usuario y contraseña a utilizar*).

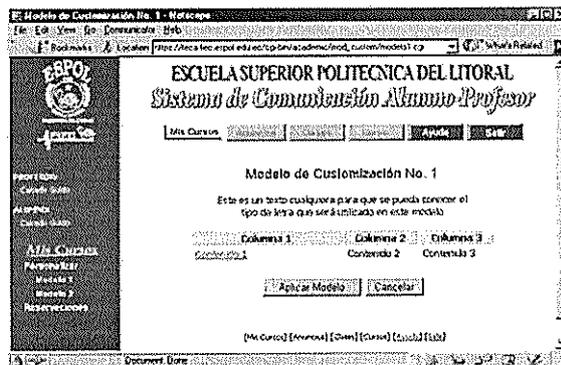
Al hacer click en el botón **Aceptar** se mostrará un resumen de los datos ingresados y luego puede regresar a la ventana Principal.

➤ CONFIGURAR APARIENCIA

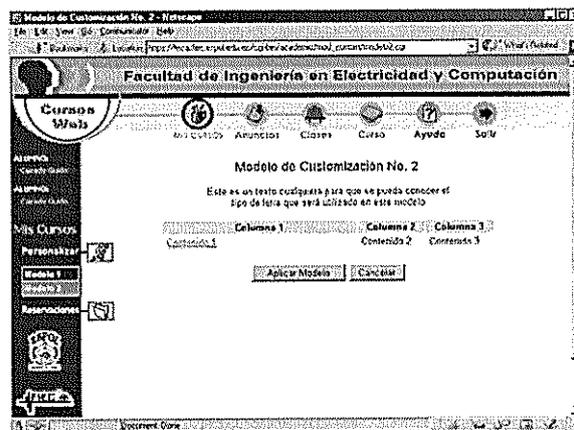
La opción **Personalizar** de la sección **Mis Cursos** además de mantener actualizada su información personal, también le permite configurar la apariencia de las páginas HTML del sistema, basándose en dos plantillas predefinidas.

A continuación se explica como utilizar las plantillas:

1. En el menú izquierdo apareció un listado con las plantillas existentes: **Modelo 1** y **Modelo 2**.
2. Al dar click en **Modelo 1**, le aparecerá la siguiente pantalla, la cual le muestra como quedarán las páginas del sistema.



3. Si desea utilizar esta configuración de click en botón **Aplicar Modelo**.
4. Después de aplicar el modelo en las páginas HTML, el sistema regresa a la página principal de Personalizar.
5. Al dar click en **Modelo 2**, le aparecerá la siguiente pantalla, la cual le muestra como quedarán las páginas del sistema.



6. Si desea utilizar esta configuración de click en botón **Aplicar Modelo**.
7. Después de aplicar el modelo en las páginas HTML, el sistema regresa a la página principal de **Personalizar**.

➤ ADMINISTRAR ANUNCIOS

La ventana de Anuncios es la que se muestra por defecto cuando selecciona una materia de la Lista de Materias de la ventana Principal.

- Si es la primera vez que ingresa, verá un mensaje informándole que no hay anuncios definidos para esa Materia.
- Si anteriormente ha ingresado anuncios los verá en una lista similar a la siguiente:

ADMINISTRACION DE ANUNCIOS

Seleccionar todos Quitar selección		Eliminar Seleccionados
Publicado	Título	
<input type="checkbox"/>	05-10-2000	 Nuevo Recurso disponible
<input type="checkbox"/>	05-10-2000	 Nueva Tarea asignada
<input type="checkbox"/>	04-10-2000	 Nuevo Recurso disponible
<input type="checkbox"/>	04-10-2000	 Nueva Tarea asignada
<input type="checkbox"/>	02-10-2000	 Archivo para el deber

En esta sección usted puede: ingresar, modificar, eliminar y consultar anuncios.

Ingresar un Anuncio

Para ingresar un nuevo anuncio realice lo siguiente:

1. Haga click en la opción **Ingreso** del menú de la izquierda.
2. Seleccione el tipo de anuncio. **Tipo:** Texto Archivo

Si escoge tipo texto deberá llenar el cuadro de texto Contenido de la ventana.

Si escoge tipo archivo deberá adjuntar el mismo.

3. Coloque un título para identificar el anuncio.

Título:

4. Cree el Anuncio.

- Si seleccionó un anuncio tipo texto entonces llene el cuadro de texto Contenido con el mensaje que desea publicar.

Contenido:

- Si seleccionó un anuncio tipo archivo deberá buscar el archivo en su disco duro haciendo click en el botón **Browse...** del campo archivo.

Archivo:

El tamaño del archivo depende del espacio disponible que tenga el usuario.

5. Para que al enviar un anuncio se envíe también un mail a sus estudiantes, marque la opción: **Enviar Mail:**
6. Una vez completo los datos haga click en el botón **Aceptar** para ingresar el anuncio, o haga click en **Limpiar** si desea rehacer el anuncio. Si acepta ingresar el anuncio verá un mensaje de confirmación de Ingreso
7. Para volver a la lista de anuncios haga click en el botón **Ver Lista de Anuncios** y para ingresar un nuevo haga click en el botón **Nuevo Anuncio**.

Modificar un Anuncio

Para modificar el contenido de un anuncio realice lo siguiente:

1. Seleccione el anuncio que desea modificar
2. Haga click en la opción **Modificar** del menú de la izquierda
3. Seleccione el tipo de anuncio **Tipo:** Texto Archivo

Si escoje tipo texto deberá llenar o modificar el cuadro de texto Contenido de la ventana.

Si escoje tipo archivo deberá adjuntar el mismo.

4. Modifique el título del anuncio de ser necesario.

Título:

5. Modifique el Anuncio

- o Si seleccionó un anuncio tipo texto entonces llene o modifique el cuadro de texto Contenido con el mensaje que desea publicar.

Contenido:

- o Si seleccionó un anuncio tipo archivo deberá buscar el archivo en su disco duro haciendo click en el botón **Browse...** del campo archivo.

Archivo:

El tamaño del archivo depende del espacio disponible que tenga el usuario.

6. Una vez modificados los datos haga click en el botón **Aceptar** para actualizar el anuncio, verá un mensaje de confirmación de modificación exitosa y un resumen de los datos modificados.
7. Para volver a la lista de Anuncios haga click en el botón **Ver Lista de Anuncios**.

Eliminar un Anuncio

Hay dos formas de eliminar un anuncio:

- Seleccionando el(los) anuncio(s) en la lista de anuncios y haciendo click en **Eliminar Seleccionados**.
- Cuando se visualiza el contenido del anuncio puede hacer click sobre la opción **Eliminar** del menú de la izquierda.

En los dos casos el sistema pide que confirme la eliminación, si los elimina desde la lista de anuncios se mostrará una lista con los anuncios a eliminar, si elimina un anuncio desde el menú de la izquierda entonces se muestra el número y el título, por ejemplo:

Eliminación de Anuncio

Desea eliminar el anuncio #001, que tiene como título:

Ayudantías

Haga click en el botón **Cancelar** si no desea eliminar el o los Anuncios o haga click en el botón **Eliminar** para borrar definitivamente el o los anuncios, verá un mensaje de eliminación exitosa y podrá regresar a la lista de anuncios.

Consultar un Anuncio

Para consultar un anuncio debe dar click en el título del anuncio, esto lo hace desde la lista de anuncios.

- Si el anuncio es tipo texto se mostrará el contenido del anuncio.

Bibliografía Modificada

Ahora está incluido el libro de Donald Norman

- Si el anuncio es tipo archivo se mostrará una pantalla similar a la anterior, pero con un icono de un archivo. Para abrir el archivo o grabarlo en su PC, debe darle click al icono.

➤ USAR CALENDARIOS

La opción **Calendarios** de la sección **Clases** permite ingresar, modificar y eliminar cualquier evento en un día de calendario específico en la materia, dar a conocer a los alumnos que tema se tratará en un determinado día, las tareas que han sido enviadas, y todo lo que crea conveniente poner en el calendario. Al acceder a la opción calendarios se puede ver el calendario del mes actual, en el cual se puede información en los días en que se va a tener clase y que todavía no hayan sido llenados.

[\[Listado de Meses\]](#) [\[Clase\]](#)

Diciembre 2000

Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
			1	2	
3	4	5	6	7	8
9	10	11	12	13	14
15	16	17	18	19	20
21	22	23	24	25	26

También puede ver la lista de los meses del término académico actual para hacer las modificaciones necesarias en dichos meses. Esto lo consigue dando click en [\[Listado de Meses\]](#) de la parte superior del calendario.

[Visualizar como Listado](#)

Seleccionar todos Quitar selección		Borrar seleccionados
Mes	Año	Calendario
<input type="checkbox"/> Octubre	2000	
<input type="checkbox"/> Noviembre	2000	
<input type="checkbox"/> Diciembre	2000	
<input type="checkbox"/> Enero	2001	
<input type="checkbox"/> Febrero	2001	

Para ver la información de un mes haga click sobre el ícono del mes que desee y verá una ventana similar a la mostrada al principio con la información que haya sido ingresada hasta ese momento.

La opción **Visualizar como Listado** permite ver una lista de todos los días en los que ha realizado una planificación.

[Visualizar como Calendario](#)

[Seleccionar todos](#) | [Quitar selección](#) [Eliminar Seleccionados](#)

Fecha	Título	Clase
<input type="checkbox"/> 14-06-2001	Leccion	

En esta sección usted puede: ingresar, modificar, eliminar una clase y borrar todas las clases de un mes.

Ingresar una Clase

Puede realizar el ingreso de las clases de dos maneras:

- Cuando visualiza el mes en forma de calendario, dando click en el día deseado.
 - Cuando visualiza las clases como listado, dando click en **Ingreso** del menú de la izquierda.
1. Si realiza el ingreso desde el calendario haga click en el día del mes en el que desee poner información. Pero si realiza el ingreso desde el menú de la izquierda debe ingresar la fecha en el que desee poner información:

Fecha: -- (dd-mm-aaaa)

2. Ingrese el título de la clase:

Título:

3. Ingrese el contenido resumido de la clase:

Contenido:

4. Si fuera necesario adjunte archivos para la clase buscándolos en su disco duro, para esto haga click en **Browse...**, de esta forma los alumnos pueden tener con tiempo dichos archivos y pueden revisarlos antes de la clase.

Adjuntar archivo 1:	<input type="text"/>	<input type="button" value="Browse..."/>
Adjuntar archivo 2:	<input type="text"/>	<input type="button" value="Browse..."/>
Adjuntar archivo 3:	<input type="text"/>	<input type="button" value="Browse..."/>

5. Una vez completo los datos haga click en el botón **Aceptar** para ingresar el contenido de la clase, o haga click en **Limpiar** si desea rehacer el contenido. Si acepta ingresar el anuncio verá un mensaje de confirmación de Ingreso y un resumen de los datos ingresados.
6. Para volver a la vista del calendario haga click en el botón **Calendario del Mes** y para ver la lista de meses haga click en **Lista de Meses**.

Eliminar una Clase

Puede eliminar las clases de dos maneras:

- Cuando visualiza el mes en forma de calendario
 1. Haga click en el título de la clase que desea eliminar.
 2. Haga click en **Eliminación** en el menú de la izquierda.
 3. El sistema pide que confirme la eliminación, por ejemplo:
 4. Haga click en el botón **Cancelar** si no quiere eliminar la clase o haga click en el botón **Aceptar** si desea borrar definitivamente el contenido de la clase, verá un mensaje de eliminación exitosa.

Eliminación de Calendarios

Desea eliminar la clase día: 7 Diciembre del 2000

<input type="button" value="Aceptar"/>	<input type="button" value="Cancelar"/>
----------------------------------------	-----------------------------------------

5. Haga click en el botón **Aceptar** para ir a la lista de meses del presente término académico.
- Cuando visualiza las clases como listado.
 1. Seleccione las clases a eliminar.
 2. De click en **Eliminar Seleccionados** de la parte superior de la lista.
 3. Haga click en el botón **Cancelar** para no eliminar las clases o haga click en el botón **Aceptar** para eliminar las clases, verá un mensaje de eliminación exitosa.
 4. Haga click en el botón **Aceptar** para volver al listado de clases.

Borrar las clases de un Mes

Para borrar las clases de un mes meses realice lo siguiente:

1. Seleccione el mes en el que desea borrar las clases definidas.
2. Haga click en **Eliminar Seleccionados**.
3. El sistema pedirá que confirme la eliminación de las clases de los meses listados, por ejemplo:

Eliminación de Clases

Desea eliminar las clases de los meses listados?

Mes	Año
Noviembre	2000

4. Haga click en el botón **Cancelar** para no eliminar las clases o haga click en el botón **Aceptar** para borrar definitivamente las clases del mes o meses del término académico, verá un mensaje de eliminación exitosa.
5. Haga click en el botón **Aceptar** para volver a la lista de meses.

➤ ENVIAR TAREAS

La opción **Tareas** de la sección **Clases** permite ingresar, modificar y eliminar las tareas sobre el material que se está cubriendo en el curso, las tareas son entregadas también en el sistema por lo que pueden ser revisadas en cualquier momento.

Si es la primera vez que ingresa, verá un mensaje informándole que no hay tareas definidas para esa Materia.

Si anteriormente ha ingresado tareas las verá en una lista similar a la siguiente:

ADMINISTRACION DE TAREAS

Seleccionar todos Quitar selección		Eliminar Seleccionados		
Nro.	Título	Publicada	Entrega	Respuestas
<input type="checkbox"/> 1	PRIMER PROYECTO	16-06-2001	07-07-2001	0
<input type="checkbox"/> 2	Resumen del Libro	01-06-2001	13-06-2001	16
<input type="checkbox"/> 3	Conceptos de IHM e IU	26-05-2001	31-05-2001	14

En esta sección usted puede: publicar, modificar, consultar y eliminar tareas, además podrá revisar las respuestas de una tarea.

Publicar una Tarea

Para ingresar una nueva tarea realice lo siguiente:

1. Haga click en la opción **Ingreso** del menú de la izquierda.
2. Coloque un título para la tarea.

Título:

3. Ingrese la fecha de entrega de la tarea.

Fecha de Entrega: -- (dd-mm-aaaa)

4. Coloque la descripción de la tarea en el campo Contenido.

Contenido:

5. Si es necesario usar un archivo para realizar la tarea, búsquelo en su disco duro haciendo click en **Browse...** , y adjúntelo a la tarea que está enviando.

Archivo:

6. Para que al enviar la tarea se envíe también un anuncio sobre la tarea, marque la opción: Enviar Anuncio:
7. Para que al enviar una tarea se envíe también un mail a sus estudiantes, marque la opción: Enviar Mail:
8. Una vez completo los datos haga click en el botón **Aceptar** para ingresar la tarea, o haga click en **Limpiar** si desea rehacer el contenido de la tarea. Si acepta ingresar la tarea, verá un mensaje de confirmación de Ingreso.
9. Para volver a la lista de tareas haga click en el botón **Ver Lista de Tareas** y para ingresar una nueva haga click en el botón **Nueva Tarea**.

Modificar una Tarea

Para modificar una tarea realice lo siguiente:

1. Seleccione la tarea que desea modificar.
2. Haga click en la opción **Modificar** del menú de la izquierda.
3. Modifique el título para la tarea de ser necesario.

Título:

4. Modifique la fecha de entrega de la tarea.

Fecha de Entrega: -- (dd-mm-aaaa)

5. Modifique la descripción de la tarea en el campo Contenido

Contenido:

6. Si es necesario usar un archivo para realizar la tarea, busquelo en su disco duro haciendo click en **Browse...** , y adjuntelo a la tarea que está enviando.

Archivo:

7. Una vez completo los datos haga click en el botón **Aceptar** para actualizar la tarea. Si acepta actualizar la tarea, verá un mensaje de modificación exitosa y un resumen de los datos modificados.
8. Para volver a la lista de tareas haga click en el botón **Ver Lista de Tareas**.

Eliminar una Tarea

Hay dos formas de eliminar una tarea:

- Seleccionando la(s) tarea(s) en la lista de anuncios y haciendo click en **Eliminar Seleccionados**.
- En el momento de modificación puede hacer click sobre la opción **Eliminar** del menú de la izquierda.

En los dos casos el sistema pide que confirme la eliminación, si los elimina desde la lista de tareas se mostrará una lista con las tareas a eliminar, si elimina la tarea desde el menú de la izquierda entonces se muestra el número y el título, por ejemplo:

Eliminación de Tarea

Desea eliminar la tarea #01, que tiene como título:

Problemas

Haga click en el botón **Cancelar** si no desea eliminar la tarea o haga click en el botón **Eliminar** para borrar definitivamente la tarea, verá un mensaje de eliminación exitosa y podrá regresar a la lista de tareas

Consultar una Tarea

Para consultar una tarea debe dar click en el título de la tarea, esto lo hace desde la lista de tareas.

- Si la tarea es tipo texto se mostrará el contenido.
- Si la tarea contiene un archivo se mostrará una pantalla con el contenido de la tarea y con un ícono del archivo. Para abrir el archivo o grabarlo en su PC, debe dar click al ícono.

PRIMER PROYECTO

Contenido: Evaluación y mejoras a Cursos Web
 Archivo: p070701.zip 

Revisar las Respuestas a las Tareas

Para revisar las respuestas realice lo siguiente:

- En la pantalla donde se muestra el listado de las tareas, de click en el número de respuestas (última columna de la lista).
- Le aparecerá un listado con las respuestas que fueron enviadas a tiempo, y otro listado con las tareas que fueron enviadas después de la fecha de entrega. En esta lista aparecerá si la respuesta es tipo texto o tipo archivo.

Respuestas a la tarea #02

Alumnos que han respondido la tarea a tiempo

Matrícula	Alumno	Respondida
199508052	GABINO MIRANDA RENE ARTURO	 13-06-2001
199514837	GILER MOREIRA JUAN CARLOS	 13-06-2001
199709098	CORDERO CARRASCO SILVANAMARIA FRANCHE	 13-06-2001
199710344	DOMINGUEZ BONINI FEDERICO XAVIER	 13-06-2001

Alumnos que han respondido la tarea atrasada

Matrícula	Alumno	Respondida
199614868	LEON ZAMBRANO JAIRO ELIAS	 14-06-2001

- Si es tipo texto al dar click, usted consultará la respuesta, como por ejemplo:

Respuesta de la tarea: Resumen del Libro

Alumno: GILER MOREIRA JUAN CARLOS

Tarea de Juan Giler Moreira (199514837)

1. La historia de la puerta ilustra uno de los mas importantes principios del diseño: VISIBILIDAD
2. Los diseñadores trabajan duro, son realmente cuidadosos, pero nunca habían pensado en intentarlo, o en mirar cualquier persona utilizarlo
3. Las piezas vienen de la habilidad del diseñador para hacer la operación buena, de proyectar una buena imagen de la operación, y tomar ventaja de las cosas que la gente espera

- Si es tipo archivo, al consultarla se abrirá otra pantalla para mostrar el contenido del archivo. O podrá grabar ese archivo en su Pc.

➤ COMPARTIR RECURSOS

La opción **Recursos** de la sección **Clases** permite ingresar, modificar y eliminar los recursos que estarán a disposición de los alumnos del curso, estos pueden ser artículos, direcciones de internet, archivos, etc..

Si es la primera vez que ingresa, verá un mensaje informándole que no hay recursos definidos para la Materia.

Si anteriormente ha ingresado recursos los verá en una lista similar a la siguiente:

ADMINISTRACION DE RECURSOS

[Seleccionar todos](#) | [Quitar selección](#) [Eliminar Seleccionados](#)

PUBLICADOS POR EL PROFESOR		
	Título	Tipo
<input type="checkbox"/>	Estereograma-2	
<input type="checkbox"/>	Estereograma-3	
<input type="checkbox"/>	Test del punto ciego	

PUBLICADOS POR ALUMNOS			
	Título	Publicado por:	Tipo
<input type="checkbox"/>	Ilusiones y efectos visuales	GILER U.	

Aparecen dos listados: uno con los recursos publicados por el profesor y otro con los recursos publicados por los alumnos. El profesor podrá modificar o eliminar todos los recursos del curso, mientras que los estudiantes podrán modificar o eliminar únicamente los recursos que ellos hayan publicado.

En esta sección usted puede: ingresar, modificar, consultar y eliminar recursos.

Ingresar un Recurso

Para agregar un recurso realice lo siguiente:

1. Haga click en la opción **Ingreso** del menú de la izquierda.
2. Ingrese un título al recurso.

Título:

3. Seleccione el tipo de recurso (artículo, enlace, archivo u otro)

Tipo: Artículo Enlace Archivo Otro

Puede colocar un archivo en el campo **Archivo** si el recurso es de tipo artículo, archivo u otro, en caso de ser un recurso de tipo enlace deberá poner la dirección web donde está el recurso. Esta dirección debe comenzar con **http://**

4. Cree el recurso

- Si seleccionó el recurso de tipo artículo, archivo u otro, entonces debe adjuntar el archivo que contiene el recurso, para esto debe buscarlo en su disco duro haciendo click en el botón **Browse...** del campo archivo.

Archivo:

- Si seleccionó el recurso de tipo enlace, ingrese la dirección web que lleva al recurso.

Dirección:

5. Coloque una breve descripción del recurso que está ingresando, en el campo descripción.

Descripción:

6. Para enviar un anuncio sobre el nuevo recurso que está colocando, marque la opción:

Enviar Anuncio:

7. Para que al publicar un recurso se envíe también un mail a sus estudiantes, marque la opción:

Enviar Mail:

8. Una vez completos los datos haga click en el botón **Aceptar** para ingresar el recurso, o haga click en **Limpiar** si desea rehacer el recurso. Si acepta ingresar el recurso verá un mensaje de confirmación de Ingreso.
9. Para volver a la lista de recursos haga click en el botón **Ver Lista de Recursos** y para ingresar uno nuevo haga click en el botón **Nuevo Recurso**.

Modificar un Recurso

Para modificar el contenido de un recurso realice lo siguiente:

1. Seleccione el recurso que desea modificar.
2. Haga click en la opción **Ingreso** del menú de la izquierda.
3. Modifique el título del recurso.

Título:

4. Modifique el tipo de recurso de ser necesario.

Tipo: Artículo Enlace Archivo Otro

Puede colocar un archivo en el campo Archivo si el recurso es de tipo artículo, archivo u otro, en caso de ser un recurso de tipo enlace deberá poner la dirección web donde está el recurso. Esta dirección debe comenzar con **http://**

5. Modifique el contenido del recurso

- Si seleccionó el recurso de tipo artículo, archivo u otro, entonces debe adjuntar el archivo que contiene el recurso, para esto debe buscarlo en su disco duro haciendo click en el botón **Browse...** del campo archivo

Archivo:

- Si seleccionó el recurso de tipo enlace, ingrese o modifique la dirección web que lleva al recurso.

Dirección:

6. Modifique la descripción del recurso en el campo descripción.

Descripción:

7. Una vez completos los datos haga click en el botón **Aceptar** para actualizar el recurso. Si acepta actualizar el recurso verá un mensaje de confirmación de modificación exitosa.
8. Para volver a la lista de recursos haga click en el botón **Ver Lista de Recursos**.

Eliminar un Recurso

Hay dos formas de eliminar un recurso:

1. Seleccionando el(los) recurso(s) en la lista de recursos y haciendo click en **Eliminar Seleccionados**.
2. En el momento de modificación puede hacer click sobre la opción **Eliminar** del menú

En los dos casos el sistema pide que confirme la eliminación, si los elimina desde la lista de recursos se mostrará una lista con los recursos a eliminar, si elimina el recurso desde el menú de la izquierda entonces se muestra el número y el título, por ejemplo:

Eliminación de Recurso (Enlace)

Desea eliminar el Recurso #01, que tiene como título:

Dirección

Haga click en el botón **Cancelar** si no desea eliminar el Recurso o haga click en el botón **Eliminar** para borrar definitivamente el Recurso, verá un mensaje de eliminación exitosa y podrá regresar a la lista de recursos

Consultar un Recurso

Para consultar un recurso debe dar click en el título del recurso, esto lo hace desde la lista de recursos.

- Si el recurso es tipo enlace se abrirá otra ventana y se cargará la página Web correspondiente al enlace.
- Si el recurso es de tipo archivo, artículo, u otro se mostrará una pantalla con un detalle del recurso. Debe dar click en el nombre del archivo para explorarlo o grabarlo en su PC.

Consulta de Recurso (Otros)

Título:	Estereograma-2
Archivo:	stereo2.gif
Tamaño:	140.69 Kb

➤ COMPARTIR INFORMACIÓN

La opción **Información** de la sección **Clases** permite ingresar, modificar y eliminar la información adicional del curso. Ej: horarios de clases, aula, y cualquier otro tipo de información que sea útil para el curso.

Si es la primera vez que ingresa, verá un mensaje comunicando que no hay información disponible para la Materia.

Si anteriormente ha ingresado información la verá en una lista similar a la siguiente:

ADMINISTRACION DE INFORMACION

[Seleccionar todos](#) | [Quitar selección](#) | [Eliminar Seleccionados](#)

Título
<input type="checkbox"/>  Cambio de Aula
<input type="checkbox"/>  Como usar el Programa
<input type="checkbox"/>  Como cambiar el password

En esta sección usted puede: ingresar, modificar, consultar y eliminar información.

Ingresar Información

Para agregar información realice lo siguiente:

1. Haga click en la opción **Ingreso** del menú de la izquierda.
2. Seleccione el tipo de información.

Tipo: Texto Archivo

Si escoje tipo texto deberá llenar el cuadro de texto Contenido de la ventana.
Si escoje tipo archivo deberá adjuntar el mismo.

3. Coloque un título para identificar la Información.

Título:

4. Cree la Información

Si seleccionó información tipo texto entonces llene el cuadro de texto Contenido con el mensaje que desea publicar.

Contenido:

Si seleccionó información tipo archivo deberá buscar el archivo en su disco duro haciendo click en el botón **Browse...** del campo archivo

Archivo:

El tamaño del archivo depende del espacio disponible que tenga el usuario.

5. Una vez completos los datos haga click en el botón **Aceptar** para ingresar la información, o haga click en Limpiar si desea rehacer el contenido de la información. Si acepta ingresar la información verá un mensaje de confirmación de Ingreso.
6. Para volver a la lista de información haga click en el botón **Ver Lista de Información** y para ingresar una nueva haga click en el botón **Nueva Información**.

Modificar Información

Para modificar una información realice lo siguiente:

1. Seleccione la información que desea modificar
2. Haga click en la opción **Modificar** del menú de la izquierda
3. Modifique el tipo de información si es necesario

Tipo: Texto Archivo

Si escoje tipo texto deberá llenar el cuadro de texto Contenido de la ventana. Si escoje tipo archivo deberá adjuntar el mismo.

4. Modifique el título de la Información.

Título:

5. Cambie la Información

- Si seleccionó información tipo texto entonces llene o modifique el cuadro de texto Contenido con el mensaje que desea publicar.

Contenido:

- Si seleccionó información tipo archivo deberá buscar el archivo en su disco duro haciendo click en el botón **Browse...** del campo archivo.

Archivo:

El tamaño del archivo depende del espacio disponible que tenga el usuario.

6. Una vez completos los datos haga click en el botón **Aceptar** para actualizar la información. Si acepta actualizar la información verá un mensaje de modificación exitosa.
7. Para volver a la lista de información haga click en el botón **Ver Lista de Información**.

Eliminar Información

Hay dos formas de eliminar información:

- Seleccionando la(s) información(es) en la lista de información y haciendo click en **Eliminar Seleccionados**
- En el momento de modificación puede hacer click en la opción **Eliminar** del menú del menú de la izquierda

En los dos casos el sistema pide que confirme la eliminación, si los elimina desde la lista de información se mostrará una lista con la información a eliminar, si elimina la información desde el menú de la izquierda entonces se muestra el número y el título, por ejemplo:

Eliminación de Información

Desea eliminar la información #01, que tiene como título:

Descripción de Deber

Aceptar

Cancelar

Haga click en el botón **Cancelar** si no desea eliminar la información o haga click en el botón **Eliminar** para borrar definitivamente la información, verá un mensaje de eliminación exitosa y podrá regresar a la lista de información

Consultar Información

Para consultar información del curso debe dar click en el título de la información, esto lo hace desde la lista de Información.

- Si la información es tipo texto se mostrará el contenido.

PROFESOR

Nombre: Guido Caicedo R. Oficina: Campus Prosperina 16-C-113 (FIEC) Teléfonos: 269-273, 269-300 Email: guido@ceibo.fiec.espol.edu.ec Horas de oficina: Lunes y Miércoles de 15:30 a 17:30 o con cita previa

- Si la información es tipo archivo se mostrará una pantalla similar a la anterior, pero con un ícono de un archivo. Para abrir el archivo o grabarlo en su PC, debe darle click al ícono.

➤ DEFINIR LAS POLÍTICAS

La opción **Políticas** de la sección **Clases** permite ingresar, modificar y eliminar las políticas del curso las cuales regirán durante la duración del período de clases.

Si es la primera vez que ingresa, verá un mensaje informándole que no hay políticas definidas para la Materia.

Si anteriormente ha ingresado políticas las verá en una lista similar a la siguiente:

ADMINISTRACION DE POLITICAS

[Seleccionar todos](#) | [Quitar selección](#) | [Eliminar Seleccionados](#)

	Título
<input type="checkbox"/>	 deberes
<input type="checkbox"/>	 Calificación

En esta sección usted puede: ingresar, modificar, eliminar y consultar políticas.

Ingresar Política

Para agregar una política realice lo siguiente:

1. Haga click en la opción **Ingreso** del menú de la izquierda
2. Seleccione el tipo de política.

Tipo: Texto Archivo

Si escoje tipo texto deberá llenar el cuadro de texto Contenido de la ventana.
Si escoje tipo archivo deberá adjuntar el mismo.

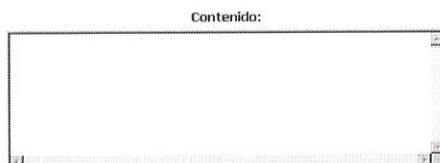
3. Coloque un título para identificar la política.

Título:

4. Cree la política

- Si seleccionó política tipo texto entonces llene el cuadro de texto Contenido con el mensaje que desea publicar.

Contenido:



- Si seleccionó política tipo archivo deberá buscar el archivo en su disco duro haciendo click en el botón **Browse...** del campo archivo.

Archivo:

El tamaño del archivo depende del espacio disponible que tenga el usuario.

5. Una vez completos los datos haga click en el botón **Aceptar** para ingresar la política, o haga click en Limpiar si desea rehacer el contenido de la política. Si acepta ingresar la política verá un mensaje de confirmación de Ingreso
6. Para volver a la lista de políticas haga click en el botón **Ver Lista de Políticas** y para ingresar una nueva haga click en el botón **Nueva Política**.

Modificar Política

Para modificar una política realice lo siguiente:

1. Seleccione la política que desea modificar.
2. Haga click en la opción **Modificar** del menú de la izquierda.
3. Modifique el tipo de política si es necesario.

Tipo: Texto Archivo

Si escoje tipo texto deberá llenar el cuadro de texto Contenido de la ventana.
Si escoje tipo archivo deberá adjuntar el mismo.

4. Modifique el título de la política.

Título:

5. Cambie la política

- Si seleccionó política tipo texto entonces llene o modifique el cuadro de texto Contenido con el mensaje que desea publicar.

Contenido:

- Si seleccionó política tipo archivo deberá buscar el archivo en su disco duro haciendo click en el botón **Browse...** del campo archivo.

Archivo:

El tamaño del archivo depende del espacio disponible que tenga el usuario.

6. Una vez completos los datos haga click en el botón **Aceptar** para actualizar la información. Si acepta actualizar la política verá un mensaje de modificación exitosa.
7. Para volver a la lista de políticas haga click en el botón **Ver Lista de Políticas**.

Eliminar Política

Hay dos formas de eliminar una política:

- Seleccionando la(s) política(s) en la lista de información y haciendo click en **Eliminar Seleccionados**
- En el momento de modificación puede hacer click en la opción **Eliminar** del menú del menú de la izquierda

En los dos casos el sistema pide que confirme la eliminación, si los elimina desde la lista de políticas se mostrará una lista con las políticas a eliminar, si

elimina la política desde el menú de la izquierda entonces se muestra el número y el título, por ejemplo:

Eliminación de Política del Curso

Desea eliminar la política #01, que tiene como título:

Se enviarán 5 deberes

Aceptar

Cancelar

Haga click en el botón **Cancelar** si no desea eliminar la política o haga click en el botón **Eliminar** para borrar definitivamente la política, verá un mensaje de eliminación exitosa y podrá regresar a la lista de políticas.

Consultar una Política

Para consultar una política del curso debe dar click en el título de la política, esto lo hace desde la lista de políticas.

- Si la política es tipo texto se mostrará el contenido.

ASISTENCIA

Debido a que parte de la nota de este curso viene de las discusiones en clase, se espera que los estudiantes asistan a TODAS las clases.

- Si la política es tipo archivo se mostrará una pantalla similar a la anterior, pero con un ícono de un archivo. Para abrir el archivo o grabarlo en su PC, debe darle click al ícono.

➤ DISCUTIR EN EL FORO

La opción **Foro** de la sección **Clases** permite enviar, modificar y eliminar los mensajes del grupo de discusión el cual ayuda a que tanto los alumnos como el profesor puedan analizar los comentarios y respuestas sobre algún tema específico.

En la ventana del Foro se tiene la lista de mensajes del foro y en la parte superior se muestra el contenido del mensaje seleccionado en la lista, generalmente el último mensaje enviado.

Si es la primera vez que ingresa, verá un mensaje informándole que no hay mensajes en el Foro.

Si anteriormente enviado mensajes los verá en una lista similar a la siguiente:

GRUPO DE DISCUSION

[Modificar Mensaje](#) [Incluir Mensaje Original](#) [Responder Mensaje](#)
 Título: Bienvenida
 Publicado por: ricardo Publicado el: 02-10-2000 14:32

Este es un espacio en el que deberán expresar sus comentarios, así como discutir sobre los diferentes temas del curso.

[Seleccionar todos](#) | [Quitar selección](#) [Eliminar Seleccionados](#)

	Título	Publicado por	# Res.	Ultimo Envio
<input type="checkbox"/>	Pregunta	ricardo	0	06-10-2000 09:54
<input type="checkbox"/>	Y cual es su opinión	ricardo	0	06-10-2000 09:42
<input checked="" type="checkbox"/>	* Bienvenida	ricardo	1	06-10-2000 09:07

En esta lista usted puede ver el contenido de un mensaje haciendo click sobre el título correspondiente de al mensaje deseado.

En esta sección usted puede: ingresar, modificar, eliminar y responder mensajes.

Ingresar un nuevo Mensaje

Para agregar un mensaje realice lo siguiente:

1. Haga click en la opción **Ingreso**, en el menú de la izquierda.
2. Coloque un título al nuevo mensaje en el campo Título.

Título:

3. Escriba el mensaje que desea publicar en el campo Contenido.

Contenido:



4. Adjunte un archivo si es necesario, buscándolo en su disco duro, para esto haga click en el botón **Browse...** del campo Archivo.

Archivo:

5. Una vez completos los datos haga click en el botón **Aceptar** para ingresar el mensaje, o haga click en **Limpiar** si desea rehacer el contenido del mensaje. Si acepta ingresar el mensaje verá una de confirmación de ingreso exitoso y una vista del mensaje enviado.
6. Para ver la lista de mensajes del haga click en **Ver Lista de Mensajes del Foro**.

Modificar un Mensaje

Para modificar el contenido de un mensaje realice lo siguiente:

1. Haga click en el mensaje deseado para que se muestre el contenido en la parte superior de la ventana.
2. Haga click sobre **Modificar Mensaje**.
3. Modifique el título del mensaje en el campo Título.

Título:

4. Modifique el mensaje que desea publicar.

Contenido:



5. Adjunte un archivo si es necesario, buscándolo en su disco duro, para esto haga click en el botón **Browse...** del campo Archivo.

Archivo:

6. Una vez completos los datos haga click en el botón **Aceptar** para actualizar el mensaje. Si acepta actualizar el mensaje verá una de confirmación de modificación exitosa y una vista del mensaje modificado.
7. Para ver la lista de mensajes del haga click en **Ver Lista de Mensajes del Foro**.

Eliminar un Mensaje

Para eliminar mensajes del Foro realice lo siguiente:

1. Seleccione el(los) mensaje(s) que desea eliminar de la lista de mensajes.
2. Haga click en **Eliminar Seleccionados**.
3. El sistema pide que confirme la eliminación de los mensajes listados, por ejemplo:

Eliminación de Mensajes

Desea eliminar los Mensajes listados?

Título	Autor	Publicado el
Bienvenidos	ricardo	01-12-2000 16:34

4. Haga click en el botón **Cancelar** si no desea eliminar los mensajes listados o haga click en el botón **Eliminar** para borrar definitivamente los mensajes listados, verá un mensaje de eliminación exitosa y podrá regresar a la lista de mensajes.

Responder un Mensaje

Cuando ingresa al Grupo de Discusión puede seleccionar si al responder se incluye el mensaje original o no, para esto marque la opción **Incluir Mensaje Original**.

Incluir Mensaje Original

Para responder un mensaje realice lo siguiente:

1. Seleccione el mensaje que desea responder para que se muestre en la parte superior de la pantalla.
2. Haga click en **Responder Mensaje**.
3. Escriba su respuesta al mensaje seleccionado.
4. Adjunte un archivo si es necesario, buscandolo en su disco duro, para esto haga click en el botón **Browse...** del campo Archivo.

Publicado por: JUAN CARLOS GILER MOREIRA

Título: RE: Bienvenidos

Contenido:

Gracias por la bienvenida
>> Bienvenidos al curso de Interacción Hombre Máquina

Archivo:

5. Una vez completos los datos haga click en el botón **Aceptar** para ingresar el mensaje, o haga click en **Limpiar** si desea rehacer el contenido del mensaje. Si acepta ingresar el mensaje verá una de confirmación de ingreso exitoso y una vista del mensaje enviado.
6. Para ver la lista de mensajes del haga click en el botón **Listado de Mensajes**.

➤ ADMINISTRAR EL CURSO

La sección **Curso** muestra la lista de las personas que están recibiendo clases en su paralelo, tiene la opción de ver una lista sencilla o con detalles, para acceder a la vista con detalles haga click sobre **Visualizar con detalles** y podrá ver la información que los estudiantes configuran en el sistema (e-mail, web site, foto, etc.).

ADMINISTRACION DEL CURSO

[Visualizar con Detalles](#)

Profesor		E-mail
CEDEÑO RICARDO		rcedeño@ceibo.fiec.espol.edu.ec
Nro.	Alumnos	E-mail
1	GONZALEZ YÉPEZ MERCY DEL ROCÍO	mercy@espol.edu.ec
2	UQUILLAS GÓMEZ VERONICA ISABEL	vuquilla@espol.edu.ec

- Haga click sobre el nombre de un alumno para ver la información detallada del mismo.

- Para enviar un e-mail a un alumno haga click sobre la dirección e-mail de la lista.

En esta sección usted puede: agregar y eliminar estudiantes, así también definir los ayudantes del curso.

Agregar un Alumno

Para agregar un alumno al curso realice lo siguiente:

1. Haga click sobre **Agregar Alumno** en el menú de la izquierda
2. Ingrese el identificador del alumno (matrícula, cédula o pasaporte)

Ingreso de Alumnos

Si desea incluir a un estudiante (que no aparece en la lista de su curso) debe especificar su tipo y número de identificación.

Identificación: MAT

Aceptar Limpiar Cancelar

3. Haga click en el botón **Aceptar** para ingresar al alumno, si el identificador está registrado en el sistema, verá los datos del alumno.
4. Haga click en el botón **Aceptar** para realizar el ingreso, verá un mensaje de ingreso exitoso.
5. Si desea ver la lista de alumnos haga click en **Lista de Alumnos** o si desea ingresar más alumnos haga click en **Agregar otro Alumno**.

Eliminar un Alumno

Para eliminar uno o más alumnos del curso realice lo siguiente:

1. Haga click sobre **Eliminar Alumno** en el menú de la izquierda.
2. Seleccionar uno o más alumnos para ser eliminados de la lista.
3. Para realizar la eliminación haga click en **Eliminar Seleccionados**.
4. El sistema mostrará una lista de los alumnos a ser eliminados.

Eliminación de Alumnos

Alumno	User
FALCONES MONTESDEOCA CRUZ MARIA	cfalcone

Aceptar Cancelar

5. Haga click en **Aceptar** para borrar definitivamente a los alumnos listados de su curso.
6. Se mostrará un mensaje de eliminación exitosa.
7. Haga click en **Aceptar** para volver a la lista de alumnos.

Definir Ayudantes

Para definir los ayudantes de la materia realice lo siguiente:

1. Haga click en **Ayudantes** en el menú de la izquierda.
2. Ingresar el identificador (matrícula, cédula, pasaporte) de los estudiantes responsables de las ayudantías.

Definición de Ayudantes

Identif. Ayudante 1:

Identif. Ayudante 2:

3. Haga click en el botón **Aceptar** para ingresar al alumno, si el identificador está registrado en el sistema, verá los datos del ayudante
4. Haga click en el botón **Aceptar** para realizar el ingreso, verá un mensaje de ingreso exitoso.
5. Haga click en **Aceptar** para regresar a la lista de alumnos.

➤ **POLÍTICAS DE NOTAS**

En la opción **Políticas de Notas** de la sección **Curso** se define la forma en la que serán calificados los exámenes, deberes y lecciones durante el curso. La ventana de administración de políticas de notas depende de su forma de calificar pero será similar a la siguiente:

ADMINISTRACION DE POLITICAS DE CALIFICACION

[Seleccionar todos](#) | [Quitar selección](#)
[Eliminar Seleccionados](#)

Parcial

	Descripción	Calific.	% Nota
<input type="checkbox"/>	examen	70	50
<input type="checkbox"/>	Lecciones	50	50

Final

	Descripción	Calific.	% Nota
<input type="checkbox"/>	Examen Final	100	100

Mejoram.

	Descripción	Calific.	% Nota
<input type="checkbox"/>	Examen	100	100

En esta sección usted puede: ingresar, modificar y eliminar Políticas de Notas.

Ingresar de Políticas de Notas

Para ingresar las políticas del curso realice lo siguiente:

1. Haga click en la opción **Ingreso** en el menú de la izquierda.

Ingreso de Política del Calificación

Nota:

Descripción:

Calificación: % Nota:

2. Seleccione para que examen será definida la política (parcial, el final o el mejoramiento).
3. Ponga una descripción de la política que se está definiendo (examen, deberes, etc).
4. Especifique el valor y el porcentaje de la nota total.
5. Una vez ingresada la información se mostrará un mensaje de ingreso exitoso y un resumen de los datos ingresados.
6. Si desea ver la lista de políticas haga click en el botón **Lista de Políticas** o haga click en el botón **Nueva Política** para ingresar una nueva.

Modificar Políticas de Notas

Para modificar una política de notas realice lo siguiente:

1. Haga click en la descripción de la política que desea modificar.
2. Haga click en la opción **Modificación** en el menú de la izquierda.

Modificación de Política de Calificación

Nota:

Descripción:

Calificación: % Nota:

3. Seleccione para que examen será definida la política (parcial, el final o el mejoramiento).
4. Modifique la descripción de la política que se está definiendo (examen, deberes, etc).
5. Modifique el valor y el porcentaje de la nota total.
6. Haga click en el botón **Aceptar**, verá un mensaje de ingreso exitoso y un resumen de los datos ingresados.
7. Haga click en el botón **Aceptar** para regresar a la lista de políticas.

Eliminar Política de Notas

Hay dos formas de eliminar una política:

- Seleccionando la(s) política(s) en la lista de políticas y haciendo click en **Eliminar Seleccionados**.
- En el momento de modificación puede hacer click sobre la opción **Eliminación** del menú de la izquierda.

En los dos casos el sistema pide que confirme la eliminación, si los elimina desde la lista de políticas se mostrará una lista con las políticas a eliminar, si elimina la política desde el menú de la izquierda entonces se muestra el número y el título, por ejemplo:

Eliminación de Política de Calificación

Desea eliminar la política de calificación que tiene como descripción:

Leccion

Haga click en el botón **Cancelar** si no desea eliminar la política o haga click en el botón **Aceptar** para borrar definitivamente la política, verá un mensaje de eliminación exitosa, haga click en **Aceptar** para regresar a la lista de políticas.

➤ NOTAS DEL CURSO

Aquí puede ingresar las notas de los exámenes de acuerdo con las políticas definidas al inicio de clases.

Ingresar o Modificar notas

Para ingresar o modificar las notas realice lo siguiente:

1. Haga click en **Notas de Curso** en el menú de la izquierda.
2. Escoja el período de calificaciones en el que desea ingresar o modificar las notas y haga click en el botón **Aceptar**.

ADMINISTRACION DE POLITICAS DE CALIFICACION

Debe seleccionar el periodo de calificaciones a ingresar o modificar.

- Parcial
- Final
- Mejoram.

3. Ingrese o modifique las notas de los alumnos de acuerdo con las políticas de notas definidas anteriormente.

Ingreso de Notas de Curso

Parcial

N1	examen	70	50%
N2	Lecciones	50	50%

Nro.	Alumnos	N1	N2	Total
1	Gonzalez Yopez Mercy del Rocio	<input type="text"/>	<input type="text"/>	0
2	Uquillas Gomez Veronica Isabel	<input type="text"/>	<input type="text"/>	0

4. Haga click en el botón **Aceptar** para ingresar las notas, verá un mensaje de Ingreso exitoso.
5. Haga click en el botón **Aceptar** para volver a la venta mostrada en la parte superior.

➤ RESUMEN DE NOTAS

El resumen de notas muestra las notas de los examens parcial, final y de mejoramiento de cada estudiante, esto es muy útil a la hora de llenar las actas de notas que deben entregarse en las unidades académicas.

La ventana con el resumen de las notas del curso será similar a la siguiente:

Resumen de Notas de Curso

Nro.	Alumnos	Parcial	Final	Mejoram.	Total	
1	Gonzalez Yepez Mercy del Rocio	96	89	0	93.0	😊
2	Uquillas Gomez Veronica Isabel	94	88	0	91.0	😊

➤ NOTAS A UN ARCHIVO

La opción **Notas a un Archivo** de la sección **Curso** le permite obtener un archivo con las calificaciones de sus alumnos de un período determinado.

A continuación se explica como utilizar esta opción:

1. Seleccione el período de calificaciones de cual desea obtener las notas.

CREACION DE ARCHIVO CON NOTAS DE CURSO

Debe seleccionar el período de calificaciones del cual desea generar el archivo.

- Parcial
 Final
 Mejoram.

Aceptar

2. Haga click en el botón **Aceptar**.
3. Especifique los componentes de la calificación que desea incluir en el archivo y el separador de los mismos.

Creación de Archivo con Notas de Curso

Debe marcar las columnas que desea sean incluidas en el archivo que será generado.

Parcial

- Identificación de Estudiantes
- Apellidos y Nombres de Estudiantes
- Participación
- Examen
- Proyecto

Delimitador de Campos:

4. Haga click en el botón **Aceptar**.
5. Le aparecerá una pantalla con un mensaje exitoso y con un ícono que muestra el archivo generado **notas.txt**. El mismo que puede abrirlo o grabarlo en su Pc dando click en el nombre del archivo.

Creación de Archivo con Notas de Curso

El archivo con la información seleccionada ha sido generado.

 notas.doc

6. Haga click en el botón **Aceptar** para regresar a la ventana de inicio de esta opción.

➤ NOTAS DESDE UN ARCHIVO

La opción **Notas de Archivo** de la sección **Curso** le permite publicar las notas de los alumnos, sin tener que ingresarlas en el sistema de una en una, sino utilizando un archivo texto que contenga las calificaciones de los estudiantes.

En el archivo cada línea representa a las calificaciones de un estudiante y su formato es:

- La primera columna debe ser **MAT** si el estudiante se identifica por la matrícula o **CED** si el estudiante se identifica por la cédula.
- La segunda columna debe ser la matrícula o cédula respectivamente.
- El resto de columnas representan las notas.
- Las columnas van separadas por el separador que usted defina, y ninguna columna va entre comillas.

A continuación se explica como utilizar esta opción:

1. Seleccione el período de calificaciones donde registrará las notas.

ASIGNACION DE CALIFICACIONES POR ARCHIVO

Debe seleccionar el período del cual se registrarán las calificaciones.

- Parcial
 Final
 Mejoram.

Aceptar

2. Haga click en el botón **Aceptar**.
3. Seleccione las calificaciones que va a publicar.
4. Especifique el archivo que contiene las calificaciones a publicar, busquelo en su disco duro haciendo click en **Browse...**
5. Especifique el separador de las columnas que utiliza en su archivo. Esto lo especifica en **Delimitador de Campos**.

Asignación de Calificaciones por Archivo

Debe marcar las columnas que han sido incluidas en el archivo a procesar.

Parcial

- Participacion
 Examen
 Proyecto

Archivo a Procesar:

Browse...

Delimitador de Campos:

Aceptar Cancelar

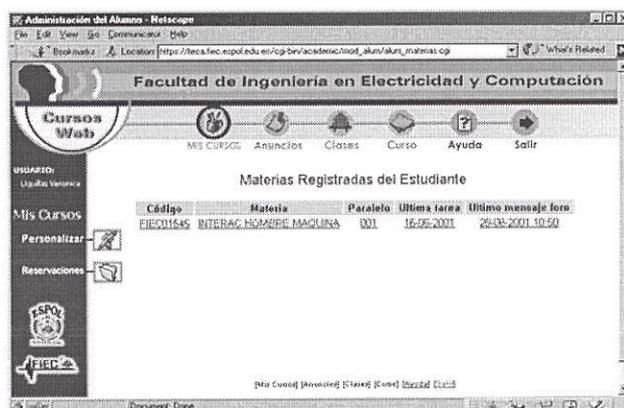
En este caso, cada línea del registro debería ser:

MAT\$19951917\$75\$85\$82. Las dos primeras columnas representan a la identificación del estudiante (MAT **delimitador** matrícula).

6. Haga click en el botón **Aceptar** para publicar las calificaciones, verá un mensaje de operación exitosa.
7. Haga click en el botón **Aceptar** para regresar a la ventana de inicio de esta opción.

SI EL USUARIO ES UN ESTUDIANTE

Una vez que ha accedido al sistema verá la ventana **Mis Cursos**, la cual será similar a la mostrada a continuación:



En la parte izquierda de la ventana aparece el nombre del estudiante que está usando el sistema, esta característica es igual para todas las ventanas del sistema.

En el centro de la página aparece una lista con las materias que el estudiante está tomando en el presente período académico y desde donde podrá acceder a cada una de ellas.

En la lista aparece información de la materia, la cual incluye: el código de la materia, el nombre de la materia, el número del paralelo, la fecha de la última tarea, y la fecha del último mensaje del foro de discusión.

Dando click en cualquiera de los atributos mencionados anteriormente, usted accederá a la materia, pero con la diferencia de que:

- Si da click sobre el Código de la Materia, el Nombre de la Materia o el Número del paralelo de la Materia, entonces ingresará a la ventana de **Anuncios** de la materia.
- Si da click sobre la Fecha de la última Tarea, entonces ingresará a la ventana de **Tareas** de la materia.
- Si da click sobre la Fecha del último mensaje del Foro de Discusión, entonces ingresará a la ventana del **Foro** de la materia.

Cuando ingrese en una de las materias encontrará en el menú superior las opciones: **Anuncios** donde podrá consultar los anuncios de esa materia, **Clases** donde podrá consultar toda la información de la materia, como calendarios, tareas, políticas, clases, etc., **Curso** donde podrá consultar sus notas y los perfiles de los miembros del curso. Además posee las opciones de **Mis Cursos** para regresar a la ventana principal, **Ayuda** para ver la ayuda del Sistema y **Salir** para salir del sistema y regresar a la ventana de ingreso al Sistema. Estas tres últimas opciones están disponibles en todas las pantallas del sistema.

En la ventana de Mis Cursos, en el submenú que se encuentra a la izquierda posee las opciones:

1. **Personalizar** que permite modificar sus Datos Personales y la configuración de las páginas HTML del sistema.
2. **Reservaciones** que permite realizar reservaciones de los recursos académicos, tales como: proyectores, aulas, etc.

➤ DATOS PERSONALES

La opción **Personalizar** de la sección **Mis Cursos** le permite mantener actualizada su información personal la cual será vista por sus alumnos y les puede ser de mucha utilidad en caso de querer comunicarse con usted directamente.

Su nombre y el tipo de identificación (Número de cédula, Pasaporte) son ingresados por el administrador del sistema.

Los datos que deben ser llenados por usted son:

1. País del que proviene.
2. Provincia de la que proviene.
3. Dirección domiciliaria (puede optar por mostrarla en la consulta seleccionando la casilla de verificación).

Presentar dirección en consulta

4. Números telefónicos (puede optar por mostrarlos en la consulta seleccionando la casilla de verificación).

Presentar teléfonos en consulta

5. Dirección e-mail.
6. Sitio web (si lo tiene).
7. Educación.
8. Experiencia Laboral.
9. Hobbies.
10. Fotografía (imagen en formato .jpg o .gif).
11. Servidor de Autenticación (*Sirve para verificar su usuario y su clave de acceso al sistema*).
12. Usuario (*El usuario que posea en el servidor de autenticación. Si escogió Servidor Local, usted define el usuario y contraseña a utilizar*).

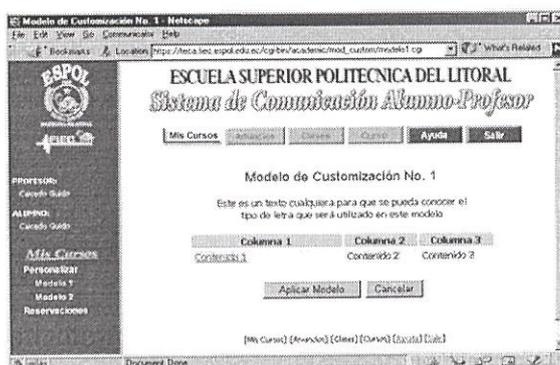
Al hacer click en el botón **Aceptar** se mostrará un resumen de los datos ingresados y luego puede regresar a la ventana Principal.

➤ CONFIGURAR APARIENCIA

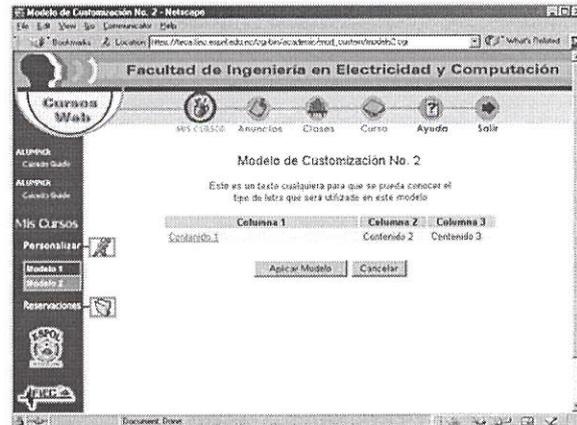
La opción **Personalizar** de la sección **Mis Cursos** además de mantener actualizada su información personal, también le permite configurar la apariencia de las páginas HTML del sistema, basándose en dos plantillas predefinidas.

A continuación se explica como utilizar las plantillas:

1. En el menú izquierdo apareció un listado con las plantillas existentes: **Modelo 1** y **Modelo 2**.
2. Al dar click en **Modelo 1**, le aparecerá la siguiente pantalla, la cual le muestra como quedarán las páginas del sistema.



3. Si desea utilizar esta configuración de click en botón **Aplicar Modelo**.
4. Después de aplicar el modelo en las páginas HTML, el sistema regresa a la página principal de Personalizar.
5. Al dar click en **Modelo 2**, le aparecerá la siguiente pantalla, la cual le muestra como quedarán las páginas del sistema.



6. Si desea utilizar esta configuración de click en botón **Aplicar Modelo**.
7. Después de aplicar el modelo en las páginas HTML, el sistema regresa a la página principal de **Personalizar**.

➤ REVISAR ANUNCIOS

La ventana de Anuncios es la que se muestra por defecto cuando seleccionas una materia de la Lista de Materias de la ventana **Mis Cursos**.

ANUNCIOS

ii Ultimo Anuncio (01-12-2000)!!

NUEVO RECURSO DISPONIBLE
Se ha publicado un ENLACE
con título: DIRECCIÓN

Publicado	Título
01-12-2000	Nuevo Recurso disponible
01-12-2000	Nueva Tarea asignada
01-12-2000	Ayudantías

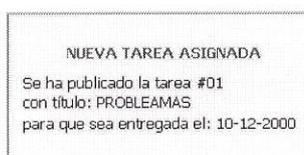
El anuncio más reciente se muestra en detalle sobre la lista de los anuncios que han sido publicados por el profesor, ya sean información específica sobre algún tema, también hay anuncios que te avisan que una nueva tarea ha sido enviada o un nuevo recurso ha sido puesto a disposición del curso.

Leer un Anuncio

Para leer un anuncio debes hacer lo siguiente:

1. Hacer click sobre el título del anuncio que deseas leer.
2. Leer el anuncio, el cual se verá como el siguiente:

CONSULTA DE ANUNCIO



[\[Anuncios del Curso\]](#)

3. Si deseas regresar a la lista de Anuncios debes hacer click en **Anuncios del Curso**.

➤ CONSULTAR CALENDARIOS

La opción **Calendarios** de la sección **Clases** te permite revisar el programa de estudio a seguir en la materia, pudiendo consultar que tema se tratará en un determinado día, las tareas que han sido enviadas, etc.

Al acceder a la opción **Calendarios** puedes ver el calendario con la planificación del mes actual, la cual te ayudará a organizar tu tiempo para estudiar o hacer las tareas de cada una de las materias, la ventana será similar a la siguiente:

[Diciembre 2000] [Mostrar Meses]

Diciembre 2000

Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17
18	19	20	21	22	23
24	25	26	27	28	29

También tienes la opción de ver la lista de meses haciendo click en **Listado de Meses** y verás un listado como el siguiente:

CALENDARIOS

[Visualizar como Listado](#)

Mes	Año	Calendario
Octubre	2000	
Noviembre	2000	
Diciembre	2000	
Enero	2001	
Febrero	2001	

La opción **Visualizar como Listado** muestra todos los días en los que se ha puesto el contenido de una clase.

Si deseas ver el contenido de un mes de la lista solo debes hacer click en el icono del mes correspondiente.

Revisar el contenido de una clase

Para revisar el contenido de una clase debes hacer lo siguiente:

1. Buscar el mes y el día que deseas consultar
2. Hacer click sobre el título de la Clase y verás una ventana como la siguiente:

Día del Calendario

Fecha:	06-12-2000
Título:	Conceptos
Contenido:	Programación

[Calendario del Mes]

3. Para regresar a la vista del Mes debes hacer click en **Calendario del Mes**.

➤ ENTREGAR TAREAS

La opción **Tareas** de la sección **Clases** te permite enviar las respuestas de las tareas que son puestas en el sistema por tu profesor. La ventana de **Tareas** te muestra un listado de las tareas que han sido enviadas, dependiendo de la cantidad de tareas que existan, la ventana será similar a la siguiente:

TAREAS					
Nro.	Título	Publicada	Entrega	Respuesta	Modificar
01	 Problemas	01-12-2000	10-12-2000	 - 	

Leer una Tarea

Para leer una tarea debes hacer lo siguiente:

1. Hacer click sobre el Título de la tarea que quieres revisar
2. Leer el contenido de la tarea, la ventana será como la siguiente

CONSULTA DE TAREA

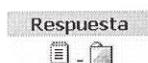
<p>PROBLEMAS</p> <p>Resolver los problemas del capítulo 6 del libro</p> <p>[Tareas del Curso]</p>

3. Para regresar a la lista de tareas debes hacer click en **Tareas del Curso**.

Responder una Tarea

Para responder una tarea debes hacer lo siguiente:

1. Hacer click en uno de los iconos de respuesta:



- Si seleccionas el icono de cuaderno, tu respuesta será de tipo texto y por lo tanto deberas escribir el contenido de la misma en la ventana siguiente:

Respuesta de Tareas

Tarea #01 Título: Probleamas

Si seleccionas el icono de carpeta, tu respuesta será un archivo y debes adjuntar un archivo en la ventana siguiente:

Tarea #01 Título: Probleamas

Para realizar esta operación debe seleccionar el archivo a cargar.

- Hacer click en **Aceptar** para enviar tu respuesta a esa tarea, verás una confirmación de envío exitoso.
- Para regresar a la lista de tareas debes hacer click en **Tareas del Curso**.

Modificar una Tarea

Para modificar una tarea debes hacer lo siguiente:

- Hacer click en uno de los iconos de modificar respuesta:



- Si seleccionas el icono de cuaderno, tu respuesta será de tipo texto y por lo tanto deberas cambiaar el contenido de la misma en la ventana siguiente:

Respuesta de Tareas

Tarea #01 Título: Probleamas

Si seleccionas el icono de carpeta, tu respuesta será un archivo y debes adjuntar un archivo en la ventana siguiente:

Tarea #01 Título: Probleamas

Para realizar esta operación debe seleccionar el archivo a cargar.

3. Hacer click en **Aceptar** para enviar tu respuesta a esa tarea, verás una confirmación de envío exitoso y si estás fuera de tiempo te avisará que tu tarea es entregada tarde.
4. Para regresar a la lista de tareas debes hacer click en **Tareas del Curso**.

➤ COMPARTIR RECURSOS

La opción **Recursos** de la sección **Clases** donde puedes obtener archivos, direcciones web, etc sobre los diferentes temas de una materia y estos recursos pueden ser publicados por el profesor o por tus compañeros de clase, esta opción hace posible que todos los alumnos tengan igualdad de condiciones en lo que a material para estudio se refiere, depende de ustes aprovechar todos los recursos que se ponen a su disposición.

Leer un Recurso

Para leer un recurso debes hacer lo siguiente:

1. Hacer click sobre el título del recurso que deseas leer.
2. Leer el contenido del recurso, el cual se verá como el siguiente:

Consulta de Recurso (Enlace)

Título:	Dirección
Dirección:	http://www.vbcode.com
Descripción:	Dirección sobre Visual Basic

3. Si deseas regresar a la lista de Anuncios debes hacer click en **Recursos del Curso**.

Ingresar un Recurso

Para agregar un recurso realice lo siguiente:

1. Haga click en la opción **Ingreso** del menú de la izquierda.
2. Ingrese un título al recurso.

Título:

3. Seleccione el tipo de recurso (artículo, enlace, archivo u otro)

Tipo: Artículo Enlace Archivo Otro

Puede colocar un archivo en el campo **Archivo** si el recurso es de tipo artículo, archivo u otro, en caso de ser un recurso de tipo enlace deberá poner la dirección web donde está el recurso. Esta dirección debe comenzar con **http://**

4. Cree el recurso
 - Si seleccionó el recurso de tipo artículo, archivo u otro, entonces debe adjuntar el archivo que contiene el recurso, para esto debe buscarlo en su disco duro haciendo click en el botón **Browse...** del campo archivo.

Archivo:

- Si seleccionó el recurso de tipo enlace, ingrese la dirección web que lleva al recurso.

Dirección:

2. Coloque una breve descripción del recurso que está ingresando, en el campo descripción.

Descripción:

3. Para enviar un anuncio sobre el nuevo recurso que está colocando, marque la opción:

Enviar Anuncio:
4. Para que al publicar un recurso se envíe también un mail a sus estudiantes, marque la opción:

Enviar Mail:
5. Una vez completos los datos haga click en el botón **Aceptar** para ingresar el recurso, o haga click en **Limpiar** si desea rehacer el recurso. Si acepta ingresar el recurso verá un mensaje de confirmación de Ingreso.
6. Para volver a la lista de recursos haga click en el botón **Ver Lista de Recursos** y para ingresar uno nuevo haga click en el botón **Nuevo Recurso**.

Modificar un Recurso

Para modificar el contenido de un recurso realice lo siguiente:

1. Seleccione el recurso que desea modificar.
2. Haga click en la opción **Ingreso** del menú de la izquierda.
3. Modifique el título del recurso.

Título:

4. Modifique el tipo de recurso de ser necesario.

Tipo: Artículo Enlace Archivo Otro

Puede colocar un archivo en el campo Archivo si el recurso es de tipo artículo, archivo u otro, en caso de ser un recurso de tipo enlace deberá poner la dirección web donde está el recurso. Esta dirección debe comenzar con **http://**

5. Modifique el contenido del recurso.
 - Si seleccionó el recurso de tipo artículo, archivo u otro, entonces debe adjuntar el archivo que contiene el recurso, para esto debe buscarlo en su

disco duro haciendo click en el botón **Browse...** del campo archivo

Archivo:

- Si seleccionó el recurso de tipo enlace, ingrese o modifique la dirección web que lleva al recurso.

Dirección:

6. Modifique la descripción del recurso en el campo descripción.

Descripción:

7. Una vez completos los datos haga click en el botón **Aceptar** para actualizar el recurso. Si acepta actualizar el recurso verá un mensaje de confirmación de modificación exitosa.
8. Para volver a la lista de recursos haga click en el botón **Ver Lista de Recursos**.

Eliminar un Recurso

Hay dos formas de eliminar un recurso:

1. Seleccionando el(los) recurso(s) en la lista de recursos y haciendo click en **Eliminar Seleccionados**.
2. En el momento de modificación puede hacer click sobre la opción **Eliminar** del menú.

En los dos casos el sistema pide que confirme la eliminación, si los elimina desde la lista de recursos se mostrará una lista con los recursos a eliminar, si elimina el recurso desde el menú de la izquierda entonces se muestra el número y el título, por ejemplo:

Eliminación de Recurso (Enlace)

Desea eliminar el Recurso #01, que tiene como título:

Dirección

Haga click en el botón **Cancelar** si no desea eliminar el Recurso o haga click en el botón **Eliminar** para borrar definitivamente el Recurso, verá un mensaje de eliminación exitosa y podrá regresar a la lista de recursos

➤ REVISAR INFORMACIÓN

La opción **Información** de la sección **Clases** es donde puedes ver cualquier información adicional que tu profesor haya publicado sobre algún tema específico, sobre alguna tarea, o cambios en la planificación de las clases, etc.

INFORMACION DEL CURSO

Título
Descripción de Deber

Leer Información

Para leer una información debes hacer lo siguiente:

1. Hacer click sobre el título de la información que deseas leer
2. Leer la información, la cual se verá como la siguiente:

Descripción de Deber

Se deberán realizar solo lo ejercicios impares

[Información del Curso]

3. Si deseas regresar a la lista de información debes hacer click en **Información del Curso**.

➤ REVISAR POLÍTICAS

La opción **Políticas** de la sección **Clases** es donde puedes ver cuales serán las políticas a seguir durante el curso, esta información es muy importante para los alumnos pues pueden estar enterados de cuanto valor tiene cada examen, deberes, etc.

POLITICAS DEL CURSO

Título
Se enviarán 5 deberes

Leer las Políticas

Para leer una política debes hacer lo siguiente:

1. Hacer click sobre el título de la política que deseas leer
2. Leer la política, la cual se verá como la siguiente:

Se enviarán 5 deberes

Por parcial se entregarán 5 deberes los que equivalen a 20 puntos

[Políticas del Curso]

3. Si deseas regresar a la lista de políticas debes hacer click en **Políticas del Curso**.

➤ VER LAS NOTAS

La opción **Notas** de la sección **Curso** te permite ver cuales han sido tus calificaciones durante el semestre. Las notas que verás estan de acuerdo con las políticas de notas que tu profesor previamente ha definido.

Para consultar tus notas debes hacer click en el link Notas y verás una ventana como la siguiente:

CALIFICACIONES OBTENIDAS

Descripción	Parcial			
	Nota/Max	Calif.	Total	Prom.Curso
Participacion	20/100	90	18.0	15.0
Examen	55/100	95	52.25	41.0
Proyecto	25/100	85	21.25	20.0
TOTAL			91.5	76.0 

➤ PERFILES DEL CURSO

La opción **Perfiles** de la sección **Curso**, permite ver la lista de todos los miembros del curso y haciendo click sobre su nombre podrás ver información mas detallada de cada uno de ellos. Esta lista puede ser vista de forma resumida o de con detalles.

La lista de resumida de los alumnos es la que se muestra por defecto al hacer click en **Perfiles** la cual será similar a la siguiente:

MIEMBROS DEL CURSO

[Visualizar con Detalles](#)

	Profesor	E-mail
	CEDEÑO RICARDO	rcedeño@ceibo.fiec.espol.edu.ec
Nro.	Alumnos	E-mail
1	GONZALEZ YÉPEZ MERCY DEL ROCÍO	mercy@espol.edu.ec
2	UQUILLAS GÓMEZ VERÓNICA ISABEL	vuquilla@espol.edu.ec

Para ver la lista con detalles debes hacer click en **Visualizar con Detalles** y podrás ver una lista que incluye al profesor y al ayudante de la materia , así como a tus demás compañeros de curso, La lista se parecerá a la mostrada a continuación:

Perfiles de Integrantes del Paralelo

[Visualizar en Lista](#)

▼ Profesor	 <p>Cedeño, Ricardo Email: rcedeño@ceibo.fiec.espol.edu.ec ✉ Sitio Web:</p>
▼ Ayudante	 <p>FALCONES MONTESDEOCA, CRUZ MARIA Email: cfalcone@ceibo.fiec.espol.edu.ec ✉ Sitio Web:</p>
▼ Alumnos	 <p>MORENO VELASCO, JUAN POMPILIO Email: jmoreno@ceibo.fiec.espol.edu.ec ✉ Sitio Web:</p>

Para ver la lista de forma resumida debes hacer click en **Visualizar en Lista**.

➤ DISCUTIR EN EL FORO

La opción **Foro** de la sección **Clases** permite enviar, modificar y eliminar los mensajes del grupo de discusión el cual ayuda a que tanto los alumnos como el profesor puedan analizar los comentarios y respuestas sobre algún tema específico.

En la ventana del Foro se tiene la lista de mensajes del foro y en la parte superior se muestra el contenido del mensaje seleccionado en la lista, generalmente el último mensaje enviado.

GRUPO DE DISCUSION

[Modificar Mensaje](#) Incluir Mensaje Original [Responder Mensaje](#)
 Título: Bienvenida
 Publicado por: ricardo Publicado el: 02-10-2000 14:32

Este es un espacio en el que deberán expresar sus comentarios, así como discutir sobre los diferentes temas del curso.

[Seleccionar todos](#) | [Quitar selección](#) [Eliminar Seleccionados](#)

	Título	Publicado por	# Res.	Último Envío
Pregunta		jaranda	0	06-10-2000 09:54
Y cual es su opinión		jaguirre	0	06-10-2000 09:42
<input type="checkbox"/>	* Bienvenida	ricardo	1	06-10-2000 09:07

En esta lista puedes ver el contenido de un mensaje haciendo click sobre el título correspondiente de al mensaje deseado.

En esta sección usted puede: ingresar, modificar, eliminar y responder mensajes.

Ingresar un nuevo Mensaje

Para agregar un mensaje realice lo siguiente:

1. Haga click en la opción **Ingreso**, en el menú de la izquierda.
2. Coloque un título al nuevo mensaje en el campo Título.

Título:

3. Escriba el mensaje que desea publicar en el campo Contenido.

Contenido:

4. Adjunte un archivo si es necesario, buscandolo en su disco duro, para esto haga click en el botón **Browse...** del campo Archivo.

Archivo:

- Una vez completos los datos haga click en el botón **Aceptar** para ingresar el mensaje, o haga click en **Limpiar** si desea rehacer el contenido del mensaje. Si acepta ingresar el mensaje verá una de confirmación de ingreso exitoso y una vista del mensaje enviado.
- Para ver la lista de mensajes del haga click en **Ver Lista de Mensajes del Foro**.

Modificar un Mensaje

Para modificar el contenido de un mensaje realice lo siguiente:

- Haga click en el mensaje deseado para que se muestre el contenido en la parte superior de la ventana.
- Haga click sobre **Modificar Mensaje**.
- Modifique el título del mensaje en el campo Título.

Título:

- Modifique el mensaje que desea publicar.

Contenido:

- Adjunte un archivo si es necesario, buscandolo en su disco duro, para esto haga click en el botón **Browse...** del campo Archivo.

Archivo:

- Una vez completos los datos haga click en el botón **Aceptar** para actualizar el mensaje. Si acepta actualizar el mensaje verá una de confirmación de modificación exitosa y una vista del mensaje modificado.
- Para ver la lista de mensajes del haga click en **Ver Lista de Mensajes del Foro**.

Eliminar un Mensaje

Para eliminar mensajes del Foro realice lo siguiente:

- Seleccione el(los) mensaje(s) que desea eliminar de la lista de mensajes.
- Haga click en **Eliminar Seleccionados**.

3. El sistema pide que confirme la eliminación de los mensajes listados, por ejemplo:

Eliminación de Mensajes

Desea eliminar los Mensajes listados?

Título	Autor	Publicado el
Bienvenidos	ricardo	01-12-2000 16:34

4. Haga click en el botón **Cancelar** si no desea eliminar los mensajes listados o haga click en el botón **Eliminar** para borrar definitivamente los mensajes listados, verá un mensaje de eliminación exitosa y podrá regresar a la lista de mensajes.

Responder un Mensaje

Cuando ingresa al Grupo de Discusión puede seleccionar si al responder se incluye el mensaje original o no, para esto marque la opción **Incluir Mensaje Original**.

Incluir Mensaje Original

Para responder un mensaje realice lo siguiente:

1. Seleccione el mensaje que desea responder para que se muestre en la parte superior de la pantalla.
2. Haga click en **Responder Mensaje**.
3. Escriba su respuesta al mensaje seleccionado.

Publicado por: JUAN CARLOS GILER MOREIRA

Título: RE: Bienvenidos

Contenido:

Gracias por la bienvenida

>> Bienvenidos al curso de Interacción Hombre Máquina

Archivo:

4. Adjunte un archivo si es necesario, buscandolo en su disco duro, para esto haga click en el botón **Browse...** del campo Archivo.
5. Una vez completos los datos haga click en el botón **Aceptar** para ingresar el mensaje, o haga click en **Limpiar** si desea rehacer el contenido del mensaje. Si acepta ingresar el mensaje verá una de confirmación de ingreso exitoso y una vista del mensaje enviado.
6. Para ver la lista de mensajes del haga click en el botón **Listado de Mensajes**.

APÉNDICE D

MANUAL DEL ADMINISTRADOR

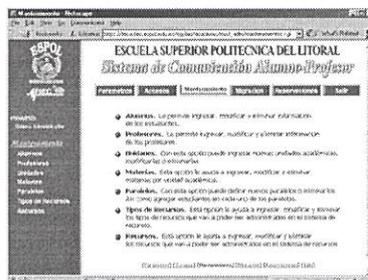
En Cursos Web existe el módulo de Administración, el mismo que será utilizado por los usuarios que tengan el perfil de Administrador General o Administrador de Unidad.

Administrador General. Se encargará de definir los parámetros generales del sistema (tales como las fechas de inicio y fin del período académico), definir quienes serán los administradores de unidad, definir las materias, profesores, alumnos, paralelos, integrantes de los cursos, etc.

Administrador de Unidad. Mantiene actualizada la información de las materias, profesores, alumnos, paralelos, integrantes del curso, pero solo de la unidad a la que esta asignado.

PARA EMPEZAR

Una vez que ha accedido al sistema verá la ventana de Mantenimiento, la cual será similar a la mostrada a continuación:



- En la parte superior izquierda de la ventana aparece el nombre del usuario que está usando el sistema.
- En la parte superior centro aparece el menú principal del módulo de administración, y en la izquierda el submenú de la opción seleccionada del menú principal.
- En el centro de la página principal de cada opción del menú aparece una descripción de todas las operaciones que podrá realizar con ésta opción.

A continuación explicaremos todas las opciones disponibles en el módulo de Administración:

➤ **DEFINIR PARÁMETROS DEL SISTEMA**

La sección **Parámetros** permite que el Administrador General pueda definir todos aquellos datos que el sistema usará como base durante un período académico, ya sea para operaciones internas o para ser usados por defecto en las diferentes opciones.

A continuación describiremos la forma en que estos parámetros se encuentran clasificados:

Generales. Agrupa parámetros que definen el período académico actual, al administrador general, el servidor de autenticación default y tamaños de archivos permitidos.

Servidores. Incluye opciones para ingresar, modificar o eliminar los servidores de autenticación que podrán ser asignados a los distintos usuarios del sistema que tengan tipo de autenticación POP.

Calendario. Permite ingresar, modificar o eliminar semanas en el calendario de actividades de la Institución.

➤ **PARÁMETROS GENERALES**

En la opción **Generales** de la sección **Parámetros**, usted podrá especificar información como:

1. *Año Actual*: Año correspondiente al período académico que el sistema considerará como actual.
2. *Término Actual*: Término correspondiente al período académico que el sistema considerará como actual.
3. *Fecha Inicio Período*: Día, mes y año en el que el período académico actual (correspondiente al año y término definido) comienza.
4. *Fecha Fin Período*: Día, mes y año en el que el período académico actual (correspondiente al año y término definido) termina.
5. *Identific. Administrador*: Tipo y número de identificación de la persona que cumplirá la función de Administrador General del sistema.
6. *Servidor Default*: Servidor de autenticación que aparecerá asignado por default en el ingreso de un nuevo alumno o profesor.
7. *Tamaño Archivo (Profesor)*: Tamaño máximo que pueden tener los archivos publicados por profesores, definido en Kbytes. Este parámetro será considerado para aquellos profesores a los que no se les ha especificado un tamaño para sus archivos.
8. *Tamaño Archivo (Alumno)*: Tamaño máximo que pueden tener los archivos publicados por alumnos, definido en Kbytes. Este parámetro será considerado para aquellos alumnos a los que no se les ha especificado un tamaño para sus archivos.
9. *Delimitador de Campos*: Carácter que será presentado en las opciones de migración de datos, notas a archivo y notas de archivo para ser considerado como separador de columnas.

Al dar click sobre el botón **Aceptar** las modificaciones realizadas sobre estos datos serán registrados, y el sistema considerará estos nuevos parámetros para presentar la información que los usuarios soliciten a partir de ese momento.

➤ ADMINISTRAR SERVIDORES

En la opción **Servidores** de la sección **Parámetros**, usted podrá ingresar, modificar o eliminar servidores contra los cuales se podrán realizar las autenticaciones de usuarios en el acceso al sistema.

- Si es la primera vez que ingresa, verá un mensaje informándole que no hay servidores definidos en el sistema.
- Si existen servidores definidos, los verá en una lista similar a la siguiente:

SERVIDORES

Seleccionar todos Quitar selección		Eliminar Seleccionados
Nombre	Código	
<input type="checkbox"/> ceibo.fiec.espol.edu.ec	02	
<input type="checkbox"/> goliat.espol.edu.ec	03	
<input type="checkbox"/> pop.mall.yahoo.com	04	
<input type="checkbox"/> teca.fiec.espol.edu.ec	01	

Ingresar un Servidor

Para agregar un nuevo servidor de autenticación realice lo siguiente:

1. Haga click en la opción **Ingreso** del menú de la izquierda
2. Ingrese el nombre del servidor. Por ejemplo: goliat.espol.edu.ec

Nombre:

3. Haga click en el botón **Aceptar** para ingresar el servidor, o haga click en **Limpiar** si desea cambiar el nombre del servidor por otro. Si acepta hacer el ingreso verá un mensaje de confirmación.
4. Para volver a la lista de servidores haga click en el botón **Ver Lista de Políticas** y para ingresar una nueva haga click en el botón **Nueva Política**.

Modificar un Servidor

Para modificar un servidor realice lo siguiente:

1. Seleccione el servidor que desea modificar.
2. Haga click en la opción **Modificar** del menú de la izquierda.
3. El único atributo que podrá modificar es el nombre del servidor.

Nombre:

4. Una vez completo los datos haga click en el botón **Aceptar** para actualizar la información. Si acepta actualizar el nombre del servidor verá un mensaje de modificación exitosa.
5. Para volver a la lista de servidores haga click en el botón **Ver Lista de Servidores**.

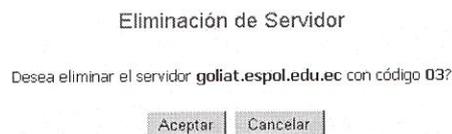
Eliminar Servidor(es)

Hay dos formas de eliminar un servidor:

- Seleccionando el o los servidores en la lista de información y haciendo click en **Eliminar Seleccionados**.

- Cuando se visualiza el contenido del servidor para realizar la modificación puede hacer click en la opción **Eliminar** del menú de la izquierda.

En los dos casos el sistema pide que confirme la eliminación, si los elimina desde la lista de servidores se mostrará una lista con los servidores a eliminar. Si lo elimina desde el menú de la izquierda, entonces se muestra el número y el nombre del servidor a eliminar, por ejemplo:



Haga click en el botón **Cancelar** si no desea eliminar el servidor o haga click en el botón **Eliminar** para borrarlo definitivamente, verá un mensaje de eliminación exitosa y podrá regresar a la lista de servidores.

➤ REGISTRAR CALENDARIOS

En la opción **Calendarios** de la sección **Parámetros**, usted podrá ingresar, modificar o eliminar semanas del calendario, que servirán para definir las fechas en las cuales los usuarios de Cursos Web podrán reservar los recursos académicos.

- Si es la primera vez que ingresa, verá un mensaje informándole que no hay semanas del calendario definidas en el sistema.
- Si existen semanas del calendario definidas, las verá en una lista similar a la siguiente:

CALENDARIO DEL PERIODO ACADEMICO

Seleccionar todos Quitar selección		Eliminar Seleccionadas	
No.	Descripción	Fecha Inicio	
<input type="checkbox"/> 01	Clases	09-10-2000	
<input type="checkbox"/> 02	clases	16-10-2000	
<input type="checkbox"/> 03	clases	23-10-2000	
<input type="checkbox"/> 07	clases	20-11-2000	
<input type="checkbox"/> 08	Exámen Parcial	27-11-2000	

Ingresar una Semana

Para agregar una nueva semana del calendario realice lo siguiente:

1. Haga click en la opción **Ingreso** del menú de la izquierda.
2. Ingrese el número de la semana. El cual sirve para identificarla, generalmente se empieza en 1, y va en orden a las fechas.

No. Semana:

- Ingrese la descripción de la semana, por ejemplo: Exámenes.

Descripción:

- Ingrese la fecha de inicio de la semana (es preferible empezar con el día lunes).

Fecha: -- (dd-mm-aaaa)

- Haga click en el botón **Aceptar** para ingresar la semana del calendario, o haga click en **Limpiar** si desea cambiar los valores ingresados.
- Al dar click en el botón **Aceptar** de la pantalla que muestra el mensaje de operación exitosa, puede ingresar nuevamente una semana, pero si desea volver a la lista de semanas haga click en **Calendario** del menú de la izquierda.

Modificar una Semana

Para modificar una semana del calendario realice lo siguiente:

- Seleccione la semana que desea modificar del listado de semanas.
- Haga click en la opción **Modificar** del menú de la izquierda.
- Modifique la descripción de la semana de ser necesario.

Descripción:

- Modifique la fecha de inicio de la semana de ser necesario.

Fecha: -- (dd-mm-aaaa)

- Haga click en el botón **Aceptar** para actualizar la semana, verá un mensaje de confirmación de modificación exitosa y un resumen de los datos modificados.
- Para volver a la lista de semana haga click en el botón **Aceptar**.

Eliminar Semana(s)

Hay dos formas de eliminar una semana del calendario:

- Seleccionando la o las semanas en la lista del calendario y haciendo click en **Eliminar Seleccionados**.
- Cuando se visualiza el contenido de la semana para realizar la modificación puede hacer click en la opción **Eliminar** del menú de la izquierda.

En los dos casos el sistema pide que confirme la eliminación, si los elimina desde la lista de servidores se mostrará una lista con los servidores a eliminar.

Eliminación de Semana Calendario

Desea eliminar las semanas calendario listadas?

No.	Descripción	Fecha Inicio
07	clases	20-11-2000
08	Exámen Parcial	27-11-2000

Si la elimina desde el menú de la izquierda, entonces se muestra el número y la descripción de la semana a eliminar.

Haga click en el botón **Cancelar** si no desea eliminar la o las semanas o haga click en el botón **Aceptar** para eliminarlas definitivamente, verá un mensaje de eliminación exitosa y podrá regresar a la lista de semanas.

➤ ADMINISTRAR LA INFORMACIÓN DE LOS ALUMNOS

En la opción **Alumnos** de la sección **Mantenimiento**, usted podrá ingresar, modificar o eliminar la información de los usuarios con el perfil de alumnos. Sin esta información los estudiantes no podrán utilizar el sistema.

Al ingresar a ésta opción, usted verá una pantalla que le permite realizar la búsqueda de los estudiantes por apellidos, nombres o identificación.

Búsqueda de Alumnos

Por Apellidos
 Por Nombres
 Por Identificación

Después de ingresar la información para realizar la búsqueda y dar click en el botón **Buscar**, aparecerá una lista con el/los estudiantes que cumplen con los valores ingresados y desde esta pantalla puede realizar nuevamente la búsqueda de otros estudiantes.

Por Apellidos
 Por Nombres
 Por Identificación

Resultado de la búsqueda de Alumnos

[Seleccionar todos](#) | [Quitar selección](#) | [Eliminar Seleccionados](#)

	Nombres	Identific.	Tipo
<input type="checkbox"/>	TAPIA, GEOVANNY	0900000015	GED
<input type="checkbox"/>	TAPIA CARDENAS, MIRIAN ISABEL	200001261	MAT

[\[Criterios de Búsqueda\]](#)

Ingresar un Alumno

Para agregar un nuevo alumno realice lo siguiente:

1. Haga click en la opción **Ingreso** del menú de la izquierda.
2. Ingrese la información del alumno, la misma que puede ser:
 - a. Tipo de identificación (*CED-cédula, MAT-matrícula*)
 - b. Número de identificación (*cédula o matrícula*)
 - c. Apellidos
 - d. Nombres
 - e. País (*opcional*)
 - f. Provincia (*opcional*)
 - g. Dirección domiciliaria (*opcional*)
 - h. Teléfonos (*opcional*)
 - i. Servidor de Autenticación (*Sirve para verificar su usuario y su clave de acceso al sistema*).
 - j. Usuario (*El usuario que posea en el servidor de autenticación. Si escogió Servidor Local, usted define el usuario y contraseña a utilizar*).
 - k. Tamaño máximo de archivos (*Sirve para definir la capacidad de almacenamiento en Kbytes que puede tener el alumno*).
 - l. Dirección Email (*opcional*)
 - m. Sitio Web (*opcional*)
3. Haga click en el botón **Aceptar** para ingresar la información del alumno, o haga click en **Limpiar** si desea cambiar los valores ingresados.
4. Al dar click en el botón **Aceptar** de la pantalla que muestra el mensaje de operación exitosa y los datos ingresados, puede ingresar nuevamente un alumno; pero si desea volver a la pantalla de búsqueda haga click en **Alumnos** del menú de la izquierda.

Modificar un Alumno

Para modificar la información de un alumno realice lo siguiente:

1. Cuando visualice la información de un alumno, haga click en la opción **Modificar** del menú de la izquierda.
2. Modifique la información del alumno, la misma que puede ser:
 - a. Apellidos
 - b. Nombres
 - c. País (*opcional*)
 - d. Provincia (*opcional*)
 - e. Dirección domiciliaria (*opcional*)

- f. Teléfonos (*opcional*)
 - g. Servidor de Autenticación (*Sirve para verificar su usuario y su clave de acceso al sistema*).
 - h. Usuario (*El usuario que posea en el servidor de autenticación. Si escogió Servidor Local, usted define el usuario y contraseña a utilizar*).
 - i. Tamaño máximo de archivos (*Sirve para definir la capacidad de almacenamiento en Kbytes que puede tener el alumno*).
 - j. Dirección Email (*opcional*)
 - k. Sitio Web (*opcional*)
3. Haga click en el botón **Aceptar** para modificar información del alumno, o haga click en **Cancelar** si no desea realizar los cambios.
 4. Aparecerá la información modificada y un mensaje de operación exitosa, y desde aquí puede dar click en el botón **Aceptar** para volver a la pantalla de búsqueda.

Consultar un Alumno

Desde la lista de alumnos que obtuvo en la búsqueda, de click en el que desea consultar y verá la información como se muestra en la figura:

Consulta de Alumno

	Identificación: MAT-200109759 Apellidos: CHÁVEZ JARA Nombres: ROBERTH DARIÓ País: Ecuador Provincia: El Oro Dir. Domicilio: Telefonos: 09138338 Servidor: ceibo.fiec.espol.edu.ec Autenticación: Pop3 Tam. Max. Archivo: 200 Kb Usuario: rchavez Dirección Email: rchavez@ceibo.fiec.espol.edu.ec Sitio Web:
-------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Eliminar Alumno(s)

Hay dos formas de eliminar un alumno del sistema:

- Seleccionando el o los alumnos desde la lista de estudiantes y haciendo click en **Eliminar Seleccionados**.
- Cuando se visualiza la información de un alumno, puede hacer click en la opción **Eliminar** del menú de la izquierda.

En los dos casos el sistema pide que confirme la eliminación, si los elimina desde la lista de alumnos se mostrará una lista con los alumnos a eliminar.

Si lo elimina desde el menú de la izquierda, entonces se muestra información del alumno a eliminar, por ejemplo:

Eliminación de Alumno

Desea eliminar al alumno CHÁVEZ JARA, ROBERTH DARÍO con identificación MAT-200109759?

Haga click en el botón **Cancelar** si no desea eliminar al alumno o haga click en el botón **Eliminar** para borrarlo definitivamente, verá un mensaje de eliminación exitosa y podrá regresar a la pantalla de búsqueda de estudiantes.

➤ ADMINISTRAR LA INFORMACIÓN DE LOS PROFESORES

En la opción **Profesores** de la sección **Mantenimiento**, usted podrá ingresar, modificar, consultar o eliminar la información de los usuarios con el perfil de profesores. Sin esta información los profesores no podrán utilizar el sistema. Al ingresar a ésta opción, usted verá una pantalla que le permite realizar la búsqueda de los profesores por apellidos, nombres o identificación.

Búsqueda de Profesores

Por Apellidos
 Por Nombres
 Por Identificación

Después de ingresar la información para realizar la búsqueda y dar click en el botón **Buscar** aparecerá una lista con el/los profesores que cumplen con los valores ingresados y desde esta pantalla puede realizar nuevamente la búsqueda de otros profesores.

Por Apellidos
 Por Nombres
 Por Identificación

Resultado de la búsqueda de Profesores

[Seleccionar todos](#) | [Quitar selección](#) [Eliminar Seleccionados](#)

	Nombres	Identific.	Tipo
<input type="checkbox"/>	TOPRES, HAYDEE DEL ROCIO MSC	0000000259	CEG
<input type="checkbox"/>	TOPRES ANDRADE, FRANCISCO ING.	1801023225	CEG
<input type="checkbox"/>	TOPRES CAMBA, HAYDEE ING.	0907852867	CEG

[Criterios de Búsqueda]

Ingresar un Profesor

Para agregar un nuevo profesor realice lo siguiente:

1. Haga click en la opción **Ingreso** del menú de la izquierda.
2. Ingrese la información del alumno, la misma que puede ser:
 - a. Tipo de identificación (*CED-cédula, PAS-pasaporte*)
 - b. Número de identificación (*cédula o pasaporte*)
 - c. Apellidos
 - d. Nombres
 - e. País (*opcional*)
 - f. Provincia (*opcional*)
 - g. Dirección domiciliaria (*opcional*)
 - h. Teléfonos (*opcional*)
 - i. Servidor de Autenticación (*Sirve para verificar su usuario y su clave de acceso al sistema*).
 - j. Usuario (*El usuario que posea en el servidor de autenticación. Si escogió Servidor Local, usted define el usuario y contraseña a utilizar*).
 - k. Tamaño máximo de archivos (*Sirve para definir la capacidad de almacenamiento en Kbytes que puede tener el profesor*).
 - l. Dirección Email (*opcional*)
 - m. Sitio Web (*opcional*)
3. Haga click en el botón **Aceptar** para ingresar la información del profesor, o haga click en **Limpiar** si desea cambiar los valores ingresados.
4. Al dar click en el botón **Aceptar** de la pantalla que muestra el mensaje de operación exitosa y los datos ingresados, puede ingresar nuevamente un profesor; pero si desea volver a la pantalla de búsqueda haga click en **Profesores** del menú de la izquierda.

Modificar un Profesor

Para modificar la información de un profesor realice lo siguiente:

1. Cuando visualice la información de un profesor, haga click en la opción **Modificar** del menú de la izquierda.
2. Modifique la información del profesor, la misma que puede ser:
 - a. Apellidos
 - b. Nombres
 - c. País (*opcional*)
 - d. Provincia (*opcional*)
 - e. Dirección domiciliaria (*opcional*)

- f. Teléfonos (*opcional*)
 - g. Servidor de Autenticación (*Sirve para verificar su usuario y su clave de acceso al sistema*).
 - h. Usuario (*El usuario que posea en el servidor de autenticación. Si escogió Servidor Local, usted define el usuario y contraseña a utilizar*).
 - i. Tamaño máximo de archivos (*Sirve para definir la capacidad de almacenamiento en Kbytes que puede tener el profesor*).
 - j. Dirección Email (*opcional*)
 - k. Sitio Web (*opcional*)
3. Haga click en el botón **Aceptar** para modificar la información del profesor, o haga click en **Cancelar** si no desea realizar los cambios.
 4. Aparecerá la información modificada y un mensaje de operación exitosa, y desde aquí puede dar click en el botón **Aceptar** para volver a la pantalla de búsqueda.

Consultar un Profesor

Desde la lista de alumnos que obtuvo en la búsqueda, de click en el que desea consultar y verá la información como se muestra en la figura:

Consulta de Profesor

	Identificación:	DED-0910755503
	Apellidos:	CAICEDO ROSSI
	Nombres:	GUIDO
	País:	ECUADOR
	Provincia:	GUAYAS
	Dir. Domicilio:	
	Telefonos:	269273
	Servidor:	teca.fiec.espol.edu.ec
	Autenticación:	Local
	Tam. Max. Archivo:	4000 Kb
	Usuario:	guido
	Dirección Email:	guido@cebo.fiec.espol.edu.ec
Sitio Web:		

Eliminar Profesor(es)

Hay dos formas de eliminar un profesor del sistema:

- Seleccionando el o los profesores desde la lista de profesores y haciendo click en **Eliminar Seleccionados**.
- Cuando se visualiza la información de un profesor, puede hacer click en la opción **Eliminar** del menú de la izquierda.

En los dos casos el sistema pide que confirme la eliminación, si los elimina desde la lista de profesores se mostrará una lista con los profesores a eliminar.

Si lo elimina desde el menú de la izquierda, entonces se muestra información del profesor a eliminar, por ejemplo:

Eliminación de Profesor

Desea eliminar al profesor **CAICEDO ROSSI, GUIDO** con identificación CED-0910755503?

Haga click en el botón **Cancelar** si no desea eliminar al profesor o haga click en el botón **Eliminar** para borrarlo definitivamente, verá un mensaje de eliminación exitosa y podrá regresar a la pantalla de búsqueda de profesores.

➤ ASIGNAR USUARIOS

En la opción **Asignar Usuario** de la sección **Accesos**, usted podrá asignar el usuario a los profesos o estudiantes que aún no lo tengan registrado.

Al ingresar a esta opción, usted podrá buscar los usuarios por apellidos, nombres o identificación.

Búsqueda de Usuarios

Por Apellidos

Por Nombres

Por Identificación

Después de ingresar los criterios para la búsqueda y dar click en el botón **Buscar**, le aparecerá una lista con los usuarios.

Por Apellidos

Por Nombres

Por Identificación

Resultado de la búsqueda de Usuarios

Nombres	Identific.	Tipo	Usuario	Tipo Aut.	Servidor
CAICEDO ESCOBAR, ANA CRISTINA	199508466	MAT	<input type="text"/>	Pop3	ceibo.fiec.espol.edu.ec
CAICEDO MEJIA, JUAN JOSE	199822297	MAT	<input type="text"/>	Pop3	ceibo.fiec.espol.edu.ec

En éste listado, usted podrá ingresar el usuario, especificar el tipo de autenticación y el servidor de autenticación.

Luego de ingresar esta información, de click en el botón **Aceptar** para registrar los usuarios o de click en el botón **Cancelar** si no desea realizar los cambios.

➤ ADMINISTRAR LAS UNIDADES ACADÈMICAS

En la opción **Unidades** de la sección **Mantenimiento**, usted podrá ingresar, modificar o eliminar las unidades académicas de la ESPOL, que servirán para definir las unidades que podrán utilizar Cursos Web.

- Si es la primera vez que ingresa, verá un mensaje informándole que no hay unidades académicas definidas en el sistema.
- Si existen unidades definidas, las verá en una lista similar a la siguiente:

UNIDADES ACADÈMICAS

Seleccionar todos | Quitar selección Eliminar Seleccionadas

Nombre	Código
<input type="checkbox"/> CELEX	CELEX
<input type="checkbox"/> CICLO BASICO	INCBA
<input type="checkbox"/> FAC. ING. ELECTRICA COMPUTACION	FIEC
<input type="checkbox"/> FAC. ING. EN CIENCIAS DE LA TIERRA	FICT
<input type="checkbox"/> FAC. ING. MARITIMA Y CIENCIAS DEL MAR	FIMAR
<input type="checkbox"/> FAC. ING. MECANICA Y OC. DE LA PRODUCCION	FIMP

Ingresar una Unidad

Para agregar una nueva unidad académica realice lo siguiente:

1. Haga click en la opción **Ingreso** del menú de la izquierda.
2. Ingrese el código de la unidad, que servirá para identificarla.

Código:

3. Ingrese el nombre de la unidad académica, por ejemplo: CELEX.

Nombre:

4. Ingrese el servidor de autenticación de usuarios de la unidad, por ejemplo en la FIEC el servidor es ceibo.fiec.espol.edu.ec.

Servidor de Adm.:

5. Ingrese el nombre de usuario del Administrador de esa unidad.

Usuario Adm.:

6. Haga click en el botón **Aceptar** para ingresar la unidad académica, o haga click en **Limpiar** si desea cambiar los valores ingresados.
7. Al dar click en el botón **Aceptar** de la pantalla que muestra el mensaje de operación exitosa, puede ingresar nuevamente una unidad, pero si desea volver a la lista de unidades haga click en **Unidades** del menú de la izquierda.

Modificar una Unidad

Para modificar una unidad académica realice lo siguiente:

1. Seleccione la unidad que desea modificar del listado de unidades.
2. Haga click en la opción **Modificar** del menú de la izquierda.
3. Modifique el nombre de la unidad de ser necesario.

Nombre:

4. Modifique el servidor de autenticación de usuarios de la unidad.

Servidor de Adm.:

5. Modifique el nombre de usuario del Administrador de la unidad.

Usuario Adm.:

7. Haga click en el botón **Aceptar** para actualizar la información de la unidad, verá un mensaje de confirmación de modificación exitosa y un resumen de los datos modificados.
8. Para volver a la lista de unidades haga click en el botón **Aceptar**.

Consultar una Unidad

Desde la lista de unidades, de click en la que desea consultar y verá la información como se muestra en la figura:

Consulta de Unidad Académica

Código:	FIEC
Nombre:	FAC. ING. ELECTRICA COMPUTACION
Servidor de Adm.:	ceibo.fiec.espol.edu.ec
Usuario Adm.:	stadmin

Eliminar Unidad(es)

Hay dos formas de eliminar una unidad académica:

- Seleccionando la o las unidades en la lista de unidades y haciendo click en **Eliminar Seleccionados**.

- Cuando se visualiza el contenido de la unidad para realizar la consulta puede hacer click en la opción **Eliminar** del menú de la izquierda.

En los dos casos el sistema pide que confirme la eliminación, si las elimina desde la lista de unidades se mostrará una lista con las unidades a eliminar.

Eliminación de Unidades Académicas

Desea eliminar las unidades académicas listadas?

Nombre	Código
CICLO BASICO	INCBA
FAC. ING. ELECTRICA COMPUTACION	FIEC

Si la elimina desde el menú de la izquierda, entonces se muestra la información de la unidad a eliminar.

Haga click en el botón **Cancelar** si no desea eliminar la o las unidades, o haga click en el botón **Aceptar** para eliminarlas definitivamente. Verá un mensaje de eliminación exitosa y podrá regresar a la lista de unidades.

➤ ADMINISTRAR LAS MATERIAS

En la opción **Materias** de la sección **Mantenimiento**, usted podrá ingresar, modificar, consultar o eliminar la información de las materias.

Al ingresar a ésta opción, usted verá una pantalla que le permite realizar la búsqueda de las materias por unidad, nombre, código, etc.

Búsqueda de Materias

Unidad

Por Nombre

Por Código

Todas A - H I - P Q - Z

Después de ingresar la información para realizar la búsqueda y dar click en el botón **Buscar** aparecerá una lista con la/las materias que cumplen con los valores ingresados y desde esta pantalla puede realizar nuevamente la búsqueda de otras materias.

Por Nombre

Por Código

Todas A - H I - P Q - Z

Resultado de la búsqueda de Materias

[Seleccionar todos](#) | [Quitar selección](#) [Eliminar Seleccionadas](#)

Nombre	Código
<input type="checkbox"/> FUNDAMENTO DE PROGRAMACION(IEC)	FIEC04341
<input type="checkbox"/> FUNDAMENTOS COMPUTACION	FIEC04473

[Criterios de Búsqueda]

Ingresar una Materia

Para agregar una nueva materia realice lo siguiente:

1. Haga click en la opción **Ingreso** del menú de la izquierda.
2. Ingrese la unidad a la que pertenece la materia.

Unidad:

3. Ingrese el código de la materia.

Código:

4. Ingrese el nombre de la materia.

Nombre:

5. Ingrese el número de exámenes de la materia.

Número de Exámenes:

6. Especifique si la materia tiene examen de mejoramiento.

Tiene mejoramiento

7. Haga click en el botón **Aceptar** para ingresar la información de la materia, o haga click en **Limpiar** si desea cambiar los valores ingresados.
8. Al dar click en el botón **Aceptar** de la pantalla que muestra el mensaje de operación exitosa y los datos ingresados, puede ingresar nuevamente una materia; pero si desea volver a la pantalla de búsqueda haga click en **Materias** del menú de la izquierda.

Modificar una Materia

Para modificar la información de una materia realice lo siguiente:

1. Cuando visualice la información de un profesor, haga click en la opción **Modificar** del menú de la izquierda.
2. Modifique el nombre de la materia de ser necesario.

Nombre:

3. Modifique el número de exámenes de la materia.

Número de Exámenes:

4. Especifique si la materia tiene examen de mejoramiento.

Tiene mejoramiento

5. Haga click en el botón **Aceptar** para modificar la información de la materia, o haga click en **Cancelar** si no desea realizar los cambios.
6. Aparecerá la información modificada y un mensaje de operación exitosa, y desde aquí puede dar click en el botón **Aceptar** para volver a la pantalla de búsqueda.

Consultar una Materia

Desde la lista de materias que obtuvo en la búsqueda, de click en la que desea consultar y verá la información como se muestra en la figura:

Consulta de Materia

Unidad Académica:	FAC. ING. ELECTRICA COMPUTACION
Código:	FIEC04473
Nombre:	FUNDAMENTOS COMPUTACION
Número de Exámenes:	2 Tiene exámen de Mejoramiento

Eliminar Materia(s)

Hay dos formas de eliminar una materia del sistema:

- Seleccionando la o las materias desde la lista de materias y haciendo click en **Eliminar Seleccionados**.
- Cuando se visualiza la información de una materia, puede hacer click en la opción **Eliminar** del menú de la izquierda.

En los dos casos el sistema pide que confirme la eliminación, si las elimina desde la lista de materias se mostrará una lista con las materias a eliminar.

Si la elimina desde el menú de la izquierda, entonces se muestra información de la materia a eliminar, por ejemplo:

Eliminación de Materia

Desea eliminar la materia FUNDAMENTOS COMPUTACION con código FIEC04473?

Haga click en el botón **Cancelar** si no desea eliminar la materia o haga click en el botón **Eliminar** para borrarla definitivamente, verá un mensaje de eliminación exitosa y podrá regresar a la pantalla de búsqueda de materias.

➤ REGISTRAR PARALELOS

En la opción **Paralelos** de la sección **Mantenimiento**, usted podrá ingresar, modificar, consultar o eliminar los paralelos para un período académico.

Al ingresar a ésta opción, usted verá una pantalla que le permite realizar la búsqueda de los paralelos por nombre de la materia, código de la materia, etc.

Búsqueda de Paralelos

Por Nombre de Materia

Por Código de Materia

Todas A - H I - P Q - Z

Modificar un Tipo de Recurso

Para modificar una unidad académica realice lo siguiente:

1. Seleccione el tipo de recurso que desea modificar del listado de tipos de recursos.
2. Haga click en la opción **Modificar** del menú de la izquierda.
3. Modifique el nombre del tipo de recurso de ser necesario.

Nombre:

4. Haga click en el botón **Aceptar** para actualizar la información del tipo de recurso, verá un mensaje de confirmación de modificación exitosa y un resumen de los datos modificados.
5. Para volver a la lista de tipos de recursos haga click en el botón **Aceptar**

Consultar un Tipo de Recurso

Desde la lista de tipos de recursos, de click en el que desea consultar y verá la información como se muestra en la figura:

Consulta de Tipo de Recurso

Código: PRO1
Nombre: PROYECTOR INFOCUS 800x600 (1024x768)

Aceptar

Eliminar Tipo de Recurso(s)

Hay dos formas de eliminar un tipo de recurso:

- Seleccionando el o los tipos de recursos en la lista de tipos de recursos y haciendo click en **Eliminar Seleccionados**.
- Cuando se visualiza el contenido del tipo de recurso para realizar la consulta puede hacer click en la opción **Eliminar** del menú de la izquierda.

En los dos casos el sistema pide que confirme la eliminación, si lo elimina desde la lista de tipos de recursos se mostrará una lista con los tipos de recursos a eliminar.

Si lo elimina desde el menú de la izquierda, entonces se muestra información del tipo de recurso a eliminar, por ejemplo:

Eliminación de Tipo de Recurso

Desea eliminar el tipo de recurso PROYECTOR INFOCUS 800x600 (1024x768) con código PRO1?

Aceptar Cancelar

Haga click en el botón **Cancelar** si no desea eliminar al tipo de recurso o haga click en el botón **Eliminar** para borrarlo definitivamente, verá un mensaje de eliminación exitosa y podrá regresar a la lista de tipos de recursos.

➤ ADMINISTRAR LOS RECURSOS

En la opción **Recursos** de la sección **Mantenimiento**, usted podrá ingresar, modificar, consultar o eliminar la información de los recursos.

Al ingresar a ésta opción, usted verá una pantalla que le permite realizar la búsqueda de los recursos por unidad, tipo de recurso, nombre, código, etc.

Búsqueda de Recursos

Unidad: FAC. ING. ELECTRICA COMPUTACION

Tipo de Recurso: PROYECTOR INFOCUS 800x600 (1024x768)

Por Nombre:

Por Código:

Todos: A - H I - P Q - Z

Después de ingresar la información para realizar la búsqueda y dar click en el botón **Buscar** aparecerá una lista con el/los recursos que cumplen con los valores ingresados y desde esta pantalla puede realizar nuevamente la búsqueda de otros recursos.

Por Nombre:

Por Código:

Todos: A - H I - P Q - Z

Resultado de la búsqueda de Recursos

≤≤ - I - L - K - L - M - Q - R - ≥≥

Seleccionar todos | Quitar selección | Eliminar Seleccionados

Descripción	Código
<input type="checkbox"/> Proyector Infocus con mouse inalámbrico	LP340-02
<input type="checkbox"/> Proyector INFOCUS con mouse inalámbrico	LP340-01
<input type="checkbox"/> Proyector tipo Cañón de Computación	BB-2#01

[Criterios de Búsqueda]

Ingresar un Recurso

Para agregar un nuevo recurso realice lo siguiente:

1. Haga click en la opción **Ingreso** del menú de la izquierda.
2. Ingrese la unidad a la que pertenece la materia.

Unidad: FAC. ING. ELECTRICA COMPUTACION

3. Ingrese el tipo de recurso al que pertenece.

Tipo de Recurso: PROYECTOR INFOCUS 800x600 (1024x768)

4. Ingrese el código del recurso.

Código:

- Ingrese el nombre del recurso.

Nombre:

- Ingrese el servidor de autenticación del usuario que administrará el recurso.

Servidor:

- Ingrese el nombre de usuario de la persona que administrará el recurso.

Usuario:

- Ingrese la localidad del recurso.

Localidad:

- Ingrese la capacidad del recurso (esto sirve cuando el recurso es un paralelo y se especifica la capacidad de estudiantes).

Capacidad:

- Haga click en el botón **Aceptar** para ingresar la información del recurso, o haga click en **Limpiar** si desea cambiar los valores ingresados.
- Al dar click en el botón **Aceptar** de la pantalla que muestra el mensaje de operación exitosa y los datos ingresados, puede ingresar nuevamente un recurso; pero si desea volver a la pantalla de búsqueda haga click en **Recursos** del menú de la izquierda.

Modificar un Recurso

Para modificar la información de una materia realice lo siguiente:

- Cuando visualice la información del recurso, haga click en la opción **Modificar** del menú de la izquierda.

- Modifique el nombre del recurso de ser necesario.

Nombre:

- Modifique el servidor de autenticación del usuario que administrará el recurso.

Servidor:

- Modifique el nombre de usuario de la persona que administrará el recurso.

Usuario:

- Modifique la localidad del recurso de ser necesario.

Localidad:

- Modifique la capacidad del recurso (en el caso de ser un paralelo).

Capacidad:

- Haga click en el botón **Aceptar** para modificar la información del recurso, o haga click en **Cancelar** si no desea realizar los cambios.

8. Aparecerá la información modificada y un mensaje de operación exitosa, y desde aquí puede dar click en el botón **Aceptar** para volver a la pantalla de búsqueda.

Consultar un Recurso

Desde la lista de recursos que obtuvo en la búsqueda, de click en el que desea consultar y verá la información como se muestra en la figura:

Consulta de Recurso

Unidad Académica:	FAC. ING. ELECTRICA COMPUTACION
Tipo de Recurso:	PROYECTOR INFOCUS 800x600 (1024x768)
Código:	LP340-01
Nombre:	Proyector INFOCUS con mouse inalámbrico
Servidor:	ceibo.fiec.espol.edu.ec
Usuario:	guido
Localidad:	PROSPERINA
Capacidad:	0

Eliminar Recurso(s)

Hay dos formas de eliminar un recurso del sistema:

- Seleccionando el o los recursos desde la lista de recursos y haciendo click en **Eliminar Seleccionados**.
- Cuando se visualiza la información de un recurso, puede hacer click en la opción **Eliminar** del menú de la izquierda.

En los dos casos el sistema pide que confirme la eliminación, si los elimina desde la lista de recursos se mostrará una lista con los recursos a eliminar. Si lo elimina desde el menú de la izquierda, entonces se muestra información del recurso a eliminar, por ejemplo:

Eliminación de Recurso

Desea eliminar el recurso **Proyector INFOCUS con mouse inalámbrico** de la unidad **FAC. ING. ELECTRICA COMPUTACION?**

Haga click en el botón **Cancelar** si no desea eliminar al recurso o haga click en el botón **Eliminar** para borrarlo definitivamente, verá un mensaje de eliminación exitosa y podrá regresar a la lista de recursos.

➤ MIGRAR LA INFORMACIÓN

En la opción **Datos** de la sección **Migración**, usted puede integrar datos al sistema o migrar la información que posea en archivos textos (la mayoría contiene la información del Sistema Académico). En estos archivos la información esta separada por un delimitador que usted especifica.

Cuando ingrese a esta opción usted verá la siguiente pantalla:

Para migrar la información debe especificar el tipo de separador a usar y los nombres de los archivos con la información relacionada.

Delimitador de Campos:	<input type="text" value="\$"/>	
Archivo Alumnos:	<input type="text"/>	<input type="button" value="Browse..."/>
	<input type="checkbox"/> Crear Directorios de Configuración de Alumnos	
Archivo Profesores:	<input type="text"/>	<input type="button" value="Browse..."/>
	<input type="checkbox"/> Crear Directorios de Configuración de Profesores	
Archivo Unidades:	<input type="text"/>	<input type="button" value="Browse..."/>
Archivo Materias:	<input type="text"/>	<input type="button" value="Browse..."/>
Archivo Paralelos:	<input type="text"/>	<input type="button" value="Browse..."/>
Archivo Alumnos por Paralelo:	<input type="text"/>	<input type="button" value="Browse..."/>
	<input type="checkbox"/> Crear Directorios	
Archivo Tipos de Recursos:	<input type="text"/>	<input type="button" value="Browse..."/>
Archivo Recursos:	<input type="text"/>	<input type="button" value="Browse..."/>

1. Especifique el delimitador de campos.
2. Ingrese los archivos para la migración, éstos pueden contener información de:
 - a. Alumnos
 - b. Profesores
 - c. Unidades académicas
 - d. Materias
 - e. Paralelos
 - f. Alumnos por paralelo
 - g. Tipos de recursos
 - h. Recursos

Especifique el nombre de los archivos donde le corresponda o de click en los botones **Browse...** para buscarlos.

3. En esta opción usted también podrá crear los directorios de configuración de Alumnos y Profesores, los mismos que servirán para que ellos puedan cambiar la apariencia de las páginas HTML del sistema mediante las

plantillas pre-definidas. Además podrá crear los directorios de trabajo de los estudiantes registrados en un paralelo.

Para crear estos directorios debe especificar donde dice **Crear Directorios de Configuración** o **Crear Directorios**.

4. Después haga click en el botón **Aceptar** para realizar el proceso de migración, o si desea ingresar nuevamente los nombres de archivos haga click en el botón **Limpiar**.
5. Después verá un mensaje de que el proceso se realizó con éxito.

Es recomendable que realice el proceso en partes, ya que aquí se genera grandes cantidades de información. Por lo tanto, si existen archivos muy extensos es bueno dividirlos en archivos más pequeños.

➤ CREAR LOS DIRECTORIOS DE TRABAJO

En la opción **Directorios** de la sección **Migración**, usted puede crear los directorios de trabajo o configuración de los usuarios, esta tarea también se puede realizar desde la migración de datos.

Cuando ingrese a ésta opción, verá la siguiente pantalla:

Para crear los directorios de trabajo y/o configuración debe especificar el tipo de separador a usar y los nombres de los archivos con la información relacionada.

Delimitador de Campos:	<input type="text" value="\$"/>	
Archivo Alumnos:	<input type="text"/>	<input type="button" value="Browse..."/>
Archivo Profesores:	<input type="text"/>	<input type="button" value="Browse..."/>
Archivo Alumnos por Paralelo:	<input type="text"/>	<input type="button" value="Browse..."/>
<input type="button" value="Aceptar"/> <input type="button" value="Limpiar"/> <input type="button" value="Cancelar"/>		

1. Especifique el delimitador de campos.
2. Ingrese los archivos para la creación de directorios, éstos pueden contener información de:
 - a. Alumnos
 - b. Profesores
 - c. Alumnos por Paralelo

Especifique el nombre de los archivos donde le corresponda o de click en los botones **Browse...** para buscarlos.

3. Después haga click en el botón **Aceptar** para realizar la creación de directorios, o si desea ingresar nuevamente los nombres de archivos haga click en el botón **Limpiar**.
4. Después verá un mensaje de que el proceso se realizó con éxito.

BIBLIOGRAFÍA

1. Eric Ladd, Jim O'Donnell. Using HTML 4, Java 1.1, and JavaScript 1.2. QUE, Segunda Edición. 1998.
2. Guido van Rossum. Python Tutorial. 1999.
3. James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, William Lorensen. Object-Oriented Modeling and Design. 1991.
4. Grady Booch. Análisis y Diseño Orientado a Objetos. Segunda Edición, 1994.
5. Jenny Preece. Human Computer Interaction. 1994
6. www.python.org
7. www.postres.org
8. www.apache-ssl.org
9. www.linux.org



A.F. 142668