ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

INFORME DE PROYECTO DE GRADUACIÓN

“MÓDULO WEB PARA LA CALIFICACIÓN AUTOMATIZADA DE TAREAS DE PROGRAMACIÓN”

Previa a la obtención del titulo de:

INGENIERO EN COMPUTACIÓN ESPECIALIZACIÓN SISTEMAS TECNOLÓGICOS

INGENIERO EN COMPUTACIÓN ESPECIALIZACIÓN SISTEMAS DE INFORMACIÓN

PRESENTADA POR:

DANNY OMAR PINOS SOLANO
WALTER IVÁN UZCA SORNOZA

GUAYAQUIL - ECUADOR

2010

AGRADECIMIENTO

A Dios por darme fuerzas día a día, a mis padres y hermanos por su apoyo incondicional, a Sofía y a Lissette por su gran ayuda y especialmente a la Ing. Cristina Abad por su invaluable aporte.

DEDICATORIA

NUESTROS PADRES:
HECTOR, MERCEDES,
OLMEDO, LIBERTAD

NUESTROS HERMANOS:
HECTOR, CECILIA, KAREN,
WALTER, JOHANNA, CINTHYA

TRIBUNAL DE SUSTENTACIÓN

PRESIDENTE

Ing. Jorge Aragundi

DIRECTOR DE TESIS

Msc. Cristina Abad

VOCAL PRINCIPAL

MBA. Ana Teresa Tapia

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta Tesis de Grado, nos corresponden exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL"

(Reglamento de Graduación de la ESPOL).

Walter Uzca Sornoza

Danny Pinos Solano

RESUMEN

El presente trabajo investiga una forma de automatizar el proceso de calificación de tareas de programación. Hemos tomado la Facultad de Ingeniería en Electricidad y Computación por tener materias que implican tareas de programación tales como Fundamentos de Programación, Estructura de Datos, entre otras. Se realizó un estudio y se implementaron soluciones para los tres lenguajes de programación más utilizados por alumnos y profesores de esta facultad, estos son: C#, C++ y JAVA. A su vez, se implementó un formato de calificación para que se puedan desarrollar en el futuro nuevas soluciones en distintos lenguajes y así poder añadirlas al sistema. La solución se la va implementar utilizando las tecnologías del Web 2.0 por las facilidades que brinda al usuario final, siendo AJAX (Asincronía JavaScript con XML) la principal y JAVA el lenguaje base de programación con su tecnología JSF que esta basado en el modelo MVC. Una vez implementada la solución se procederá a evaluarla para finalmente detallar las conclusiones y recomendaciones que resultaron del trabajo investigativo.

ÍNDICE GENERAL

ABREVIATURAS

BD: Base de datos.
DIO: Diagrama de interacción de objetos.
MVC: Modelo Vista Controlador.
JSF: Java Server Faces.
JSP: Java Server Pages.
SCAT: Sistema de Calificación automática de tareas de programación.
XML: Extensible Markup Language.
XSLT: Extensible Stylesheet Language Transformations.
XUL: XML User Interface Language.
DOM: Document Object Model.
JSON: JavaScript Object Notation.
CSS: Cascading Style Sheets.
AJAX: Asynchronous JavaScript And XML.
HTTP: Hypertext Transfer Protocol.
HTML: Hypertext Markup Language.
XHTML: Extensible Hypertext Markup Language.
URL: Uniform Resource Locator.
SQL: Structured Query Language.
API: Application Programming Interface.

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

En la actualidad se trata de automatizar la mayoría de los procesos realizados por personas a través de soluciones de software. Hay varias tareas que se pueden automatizar para ahorrar tiempo y evitar errores en la manipulación de datos, la tarea de hacer que los procesos se automaticen varía de acuerdo a la complejidad de lo que se trata de realizar; ya sea por el proceso, su lógica o las herramientas y tecnologías que se usen para implementarla.

A veces se puede plantear posibles soluciones pero no son implementadas, porque involucran mucho tiempo o algún tipo de investigación adicional. En las instituciones educativas, por ejemplo, existen sistemas que se encargan de llevar la relación alumno-profesor de forma virtual, pero no ofrecen la característica de calificar las tareas de forma automatizada. Pensando en esta carencia se realizó una investigación la cual tenía como objetivo principal automatizar el proceso de calificación de tareas, para ofrecer de esta forma un sistema que maximice el tiempo de asignación de calificaciones y su publicación.

CAPÍTULO 1

1. PLANTEAMIENTO DEL PROBLEMA Y ANÁLISIS CONTEXTUAL

El presente capítulo describe el problema que dio origen a nuestra investigación acerca del proceso de calificación de tareas de programación, además de los objetivos y diseño de la investigación.

1.1. Justificación de la investigación

Los Sistemas Web que se utilizan en la actualidad, en ESPOL, carecen de métodos sistematizados para calificar tareas de programación básica. Hemos pensado que una posible solución seria crear una aplicación complementaria para realizar estas funciones. Este complemento puede ser integrado a sistemas de comunicación alumno-profesor como SidWeb y Metis. También puede ser administrado de forma independiente como un Sistema Web de calificación de tareas de programación.

La Aplicación Web a desarrollar seria de gran ayuda para el profesor y alumno, pues proporciona de manera automatizada el proceso de revisión y calificación de tareas de programación básica.

Esta aplicación disminuye el tiempo y esfuerzo empleado por el docente al momento de calificar las tareas y además reduce el tiempo de publicación de calificaciones. A su vez, el alumno podrá recibir retroalimentación instantánea en caso de existir inconsistencias o errores de sintaxis al subir su tarea.

1.2. Objetivos

1.2.1. Objetivos Generales

Automatizar el proceso de revisión y calificación de tareas de programación básica mediante la implementación de una Aplicación Web.

1.2.2. Objetivos Específicos

- Analizar las tareas de programación de los estudiantes y asignar automáticamente la calificación correspondiente.

- Brindar retroalimentación en línea al estudiante sobre errores de programación en sus tareas.

- Optimizar el tiempo de publicación de calificaciones de las tareas de programación.

1.3. Definición del problema

La metodología para la revisión de las tareas de programación que se aplica en algunas de las instituciones académicas que tienen una aplicación Web, consiste en que el profesor descargue los archivos del estudiante, los revisa y si es necesario los compila y prueba. Luego se asigna la calificación correspondiente. Todo el proceso debe realizarlo por cada alumno del curso a su cargo.

El problema que existe en la metodología de calificación, para este tipo de tareas, es el número de estudiantes en cada paralelo. Si el número de estudiantes aumenta, el tiempo y el esfuerzo empleado por el docente también.

1.4. Metodología para el desarrollo

La metodología de implementación utilizada en el desarrollo del sistema fue el “Modelo de Desarrollo Evolutivo” [1]. Se implementó de esta forma porque en un inicio solo se conocían los requerimientos básicos y se fue desarrollando de a poco la posible solución a sabiendas que podrían aparecer nuevos obstáculos. Se empezó diseñando una solución para los lenguajes orientados a objetos y luego se vería la forma de desarrollar soluciones para otros tipos de lenguaje.

El desarrollo comenzó con JAVA con la idea de usar polimorfismo subtipado y usar una interfaz que tenga la declaración de métodos necesarios para calificar y obtener la calificación de la tarea. Cuando se inició la implementación en C# surgió el problema de que el objeto de interfaz servía para Java y no se podía hacer que un objeto en C# sea entendido por uno de Java, así que el paso siguiente fue usar lo mismo para C#, lo cual era crear una interfaz e implementar los métodos de calificación, y así sucesivamente para los demás lenguajes.

Para poder ejecutar el proceso de compilación y calificación de las tareas en sus respectivos lenguajes se diseñó un conjunto de archivos por lotes los cuales se ejecutan de forma ordenada para que se cumpla el proceso; de esta forma cada lenguaje tiene la opción de realizar este proceso sin importar el tipo de paradigma utilizado. Con este tipo de solución establecida el programa principal solo se encarga de ejecutar este conjunto de archivos por lotes para luego leer las respuestas generadas en un archivo por cada lenguaje. En la figura 1.1 se muestra el esquema del desarrollo evolutivo.

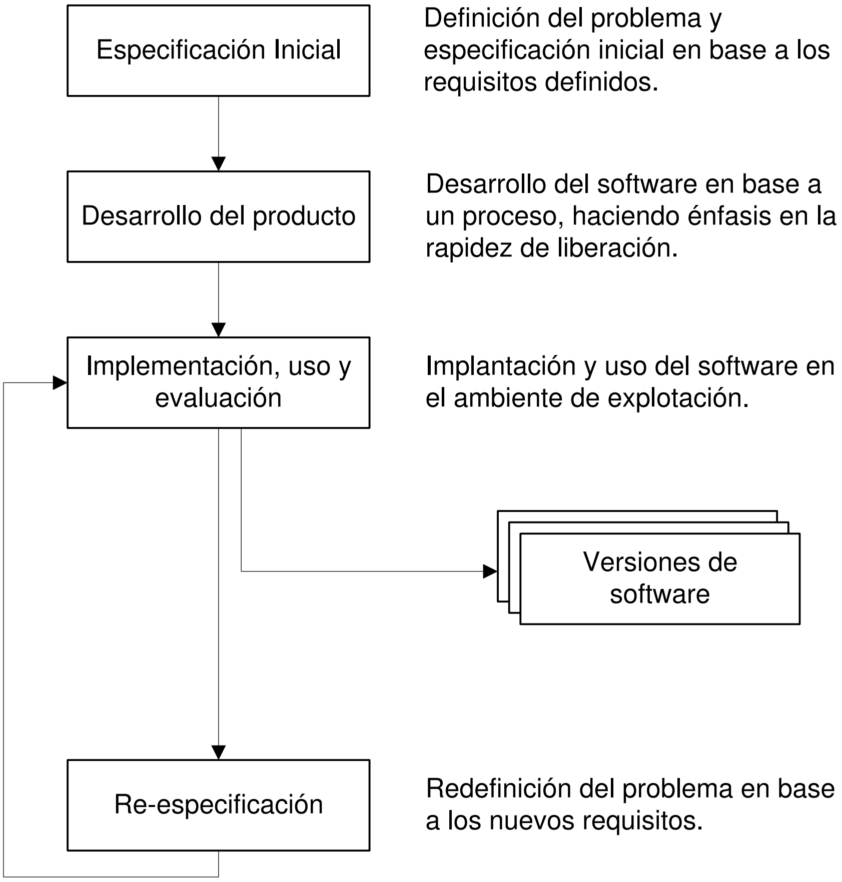


Figura 1.1: Modelo de desarrollo evolutivo.

1.5. Guías de diseño propuestas

A continuación, se describen las guías de diseño para el Sistema de Calificación de Tareas

1.5.1. Área de Operabilidad.

- Presentar el menú de opciones en una estructura de árbol jerárquico bien definida a la vista de los usuarios. La idea aquí es que el usuario, a primera vista, pueda fácilmente tener una buena idea de cuales son las opciones que el puede acceder dentro del sistema.

- Proveer la posibilidad de presentar reportes estadísticos como: número de tareas, respuestas por tarea, fechas de respuestas y porcentaje de tareas correctas, entre otros.

- Dar a los usuarios el tiempo suficiente para publicar la tarea y subir las respuestas, evitando que sufran de cierres de sesión repentinos porque el sistema no detecta interacción. Esto se hace debido a que el proceso de publicación y calificación de tareas toma cierto tiempo para completarse.

1.5.2. Área Diseño de Interfaz

- Acompañar cada nodo del árbol por un ícono que represente la operación o función que va a realizar. Cada ícono se puede seleccionar de un conjunto de imágenes preestablecidas en el sistema. Para los usuarios es mucho más fácil reconocer una operación por medio de un ícono que lo represente.

Este conjunto de guías de diseño presentadas son la base del diseño del Sistema de Calificación que implementamos. En los siguientes capítulos se describirá el área tecnológica de nuestra aplicación.

CAPÍTULO 2

2. MODELO DE EVALUACIÓN DE LAS TAREAS

El presente capítulo describe el problema que dio origen a nuestra investigación acerca del proceso de publicación y calificación de tareas de programación, además de los objetivos y diseño de la investigación. Aquí se explica cada una de las transacciones que se realizan para que estos dos procesos culminen de forma exitosa.

2.1. Publicación de tarea

2.1.1. Descripción

La publicación de la tarea consiste en que un profesor asigne una tarea a un grupo de alumnos que están inscritos en una materia y paralelo específico.

Para explicar de qué se trata la compilación del archivo de ejemplo vamos a definir el archivo de ejemplo (Ver definición en Apéndice A).

La publicación se realiza mediante la ejecución de un proceso el cual a su vez agrupa a un conjunto de tareas que se deben realizar correctamente para finalizar con esta operación.

2.1.2. Procesos

Se debe de cumplir una secuencia de pasos que incluyen validaciones para evitar inconvenientes al cumplir la transacción de publicación. Para esto, se generan archivos por lotes que ejecutan ciertas tareas específicas; las cuales se encargan de verificar que todos los requerimientos necesarios para publicar la tarea estén correctamente configurados; estas tareas las hemos dividido en cinco procesos principales:

- Compilación del archivo de ejemplo.
- Generación de plantilla del archivo de prueba.
- Compilación del archivo de casos de prueba.
- Validación del archivo de casos de prueba.
- Calificación del archivo de ejemplo.

2.1.2.1. Compilación del archivo de ejemplo

Se manda a ejecutar un archivo por lotes que invoca a SCAT_OperacionesLenguaje(Ver definición en apéndice A), este a su vez crea un archivo .bat, el cual compila el “archivo de ejemplo” para verificar si existen errores. Al realizar este proceso se generan documentos de texto donde se guardan mensajes de bandera que especifican si existen errores en compilación, de existir errores, estos se guardarán en otra ruta para mostrarlos en pantalla y luego servirán de retroalimentación al usuario.

2.1.2.2. Generación de plantilla del archivo de casos de prueba.

Una vez que se compiló el "archivo de ejemplo" sin errores, se manda a cargar el objeto resultante de la compilación (.class, .dll, .obj) para obtener los métodos y funciones y así generar un archivo de muestra de casos de prueba, el cual servirá como guía para que el usuario pueda implementar el "archivo de casos de prueba" que calificará al "archivo de ejemplo" subido en el paso anterior y por consiguiente calificará a la respuesta que será registrada por el estudiante.

2.1.2.3. Compilación del archivo de casos de prueba

Se manda a ejecutar un archivo por lotes que invoca nuevamente a SCAT_OperacionesLenguaje, este a su vez crea un archivo .bat, el cual copia a

InterfaceSCAT(Ver definición en apéndice A) en el directorio del “archivo de casos de prueba” (Ver apéndice A), establece la ruta o classpath hacia la ubicación del “archivo de ejemplo” y lo compila para verificar si existen errores. Al realizar este proceso se generan documentos de texto donde se guardan mensajes de bandera que especifican si existen errores en compilación, de existir, se guardará en un archivo de texto de errores que servirá de retroalimentación al usuario.

2.1.2.4. Validación del archivo de casos de prueba

Se manda a ejecutar un archivo por lotes que invoca nuevamente a SCAT_OperacionesLenguaje, este a su vez crea un archivo .bat, el cual carga el objeto resultante de la compilación (.class, .dll, .obj) del “archivo de casos de pruebas”.

Se lo compila junto con nuestro archivo de interfaz, se comprueba que la implemente y que se sobrescriban los tres métodos necesarios para poder continuar con la calificación. De existir errores en la compilación se muestran los errores en pantalla.

2.1.2.5. Calificación del archivo ejemplo contra el archivo de casos de prueba

Se manda a ejecutar un archivo por lotes que invoca nuevamente a SCAT_OperacionesLenguaje, este a su vez crea un archivo .bat, el cual ejecuta a Calificar<Lenguaje> el cual carga el “archivo de casos de pruebas” para que pueda calificar al “archivo de ejemplo” registrado por el profesor en el paso 1.

De existir errores en tiempo de ejecución se crea un documento con mensajes de errores, que se muestran en pantalla. Si no existen errores de ejecución se mostrará la calificación de la tarea, en este caso la calificación del “archivo de ejemplo”, con el resultado de las pruebas al que fue sometido y se publica la tarea para el(los) paralelo(s) seleccionado(s).

2.2. Respuesta de tarea

2.2.1. Descripción

La respuesta de tarea implica que el estudiante registre su “archivo de respuesta” (Ver apéndice A), luego ejecuta el “archivo de casos de prueba” del profesor tomando como base la respuesta del alumno, realizar las operaciones de calificación y que el resultado se imprima en un documento que después será leído para mostrar la calificación al estudiante. Si el estudiante no obtuvo la nota máxima y se encuentra dentro del tiempo de respuesta puede volver a responder la tarea para ver si obtiene una mejor calificación.

2.2.2. Procesos

Se debe cumplir una secuencia de pasos que incluyen validaciones para evitar inconvenientes al cumplir la transacción. Para esto se generan archivos por lotes que ejecutan ciertas tareas específicas; las cuales se encargan de verificar que todos los requerimientos necesarios para culminar con éxito esta operación estén configurados correctamente; estas tareas las hemos dividido en tres procesos principales:

- Compilación del archivo de respuesta.
- Compilación del archivo de casos de prueba.
- Calificación del archivo de respuesta.

2.2.2.1. Compilación del archivo de respuesta

Se manda a ejecutar un archivo por lotes que invoca a SCAT_OperacionesLenguaje, este a su vez crea un archivo .bat, el cual compila el “archivo de respuesta” para verificar si existen errores. Al realizar el proceso se generan documentos de texto donde se guardan mensajes de bandera que especifican si existen errores en compilación, de existir errores, se guardarán en otra ruta para mostrarlos en pantalla y servirán de retroalimentación al usuario.

2.2.2.2. Compilación del archivo de casos de prueba

Se manda a ejecutar un archivo por lotes que invoca nuevamente a SCAT_OperacionesLenguaje, este a su vez crea un archivo .bat, el cual copia a InterfaceSCAT en el directorio del “archivo de casos de prueba”, establece la ruta o classpath hacia la ubicación del “archivo de respuesta” registrado por el estudiante y lo compila para verificar si existen errores. Al realizar el proceso se generan documentos de texto donde se guardan mensajes de bandera que especifican si existen errores en compilación, de existir, se guardará en un archivo de texto de errores que servirá de retroalimentación al usuario.

2.2.2.3. Calificación del archivo de respuesta contra el archivo de casos de prueba

Se manda a ejecutar un archivo por lotes que invoca nuevamente a SCAT_OperacionesLenguaje, éste a su vez crea un archivo .bat, el cual ejecuta a Calificar<Lenguaje> el cual carga el “archivo de casos de pruebas” para que pueda calificar al “archivo de respuesta” registrado por el estudiante en el paso 1.

De existir errores en tiempo de ejecución se crea un documento con mensajes de errores, que se muestran en pantalla. Sino existen errores de ejecución se mostrará la calificación de la tarea, en este caso la calificación del “archivo de respuesta”, con el resultado de las pruebas al que fue sometido y se sube la respuesta de la tarea.

Los procesos de publicación de tareas y respuesta de tarea son los pasos básicos principales del sistema, cada uno ejecuta tareas para completar la operación de forma exitosa. Se ha explicado de forma teórica como se realizan estas acciones, a continuación vamos a describir con gráficos como se realizan cada una de estas tareas.

En la figura 2.1 mostramos un diagrama de flujo el cual enfoca el proceso de publicación de tarea.

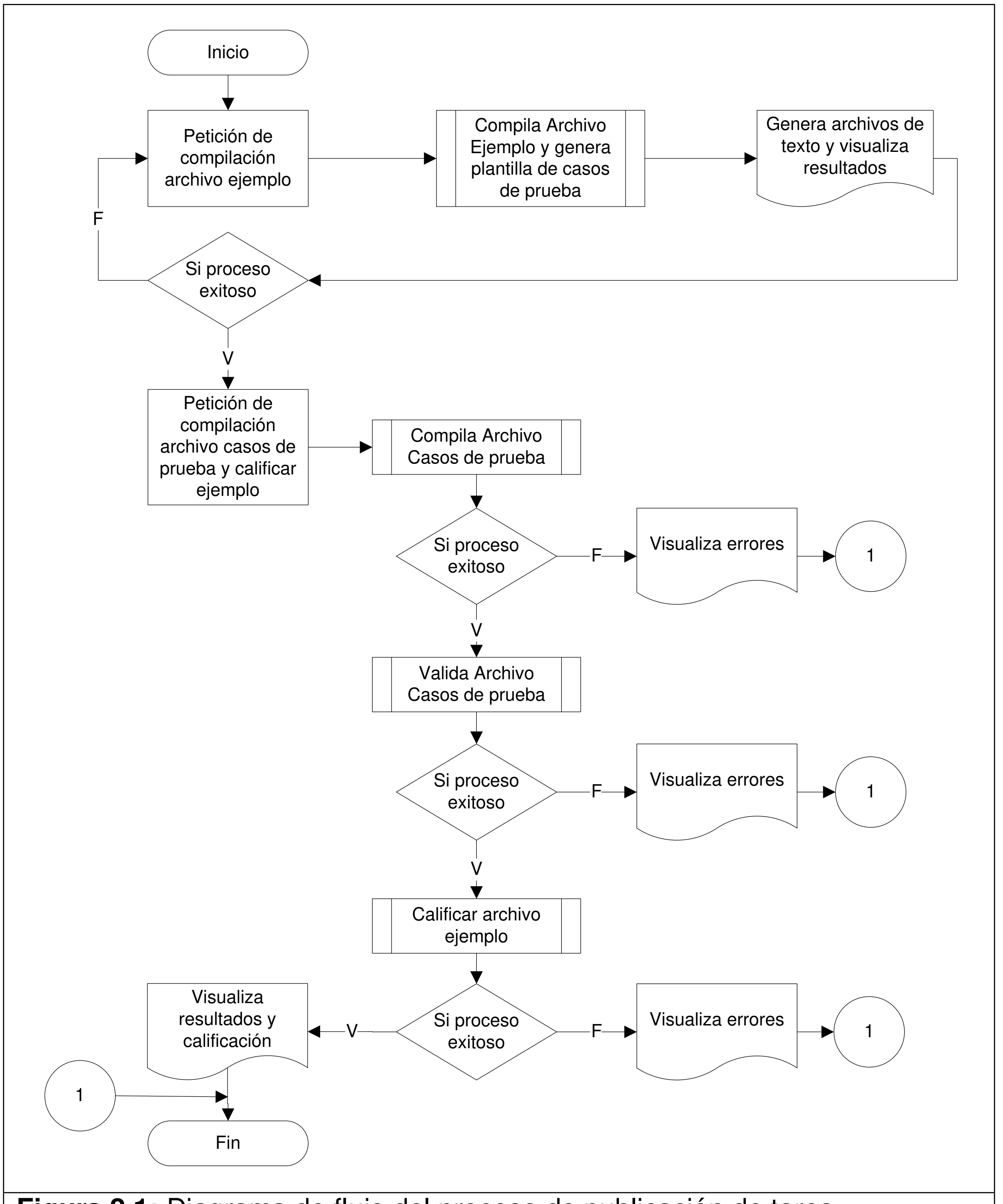


Figura 2.1: Diagrama de flujo del proceso de publicación de tarea.

En la figura 2.2 mostramos un diagrama de flujo el cual enfoca el proceso de responder una tarea.

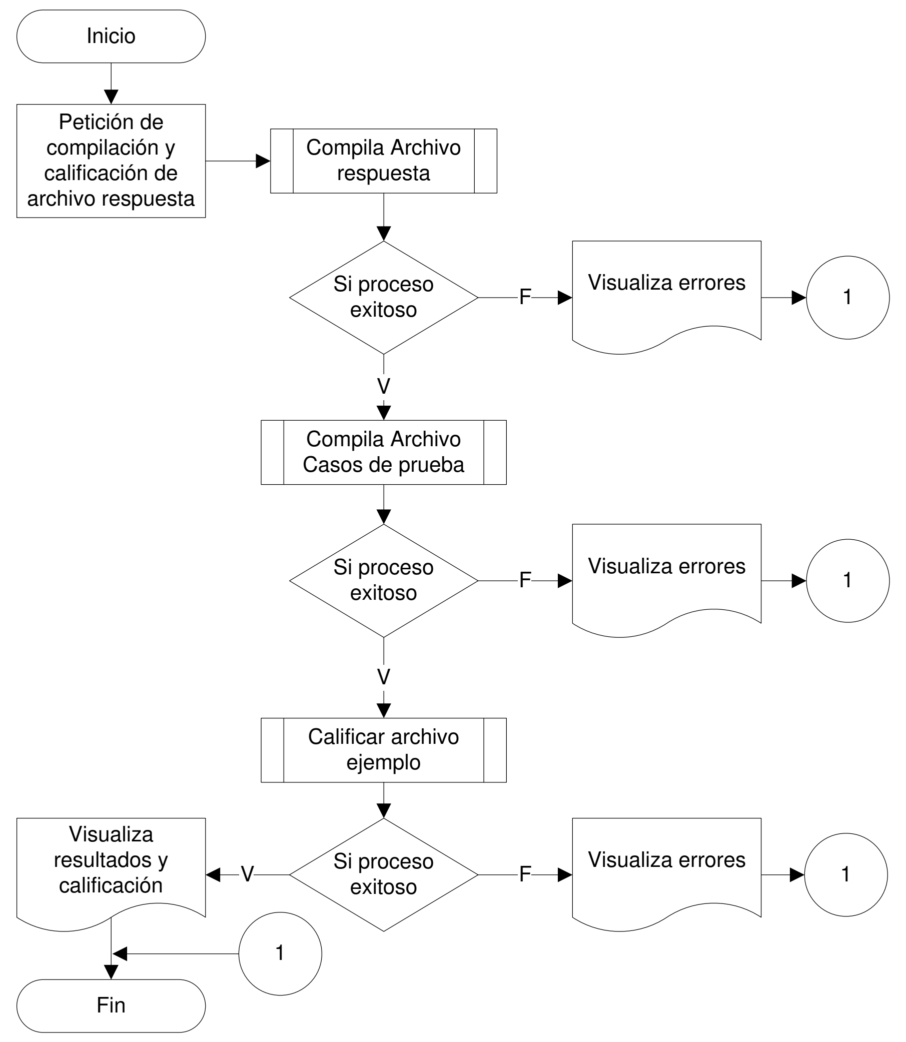


Figura 2.2: Diagrama de flujo del proceso de respuesta de una tarea.

Se puede observar que en estos dos procesos existen tareas tales como compilar y calificar archivos. A continuación mostramos un conjunto de figuras que describen el esquema de cada una de estas tareas.

En la figura 2.3 vemos el esquema de compilación del archivo ejemplo subido por el profesor ó el archivo de respuesta subido por el estudiante.

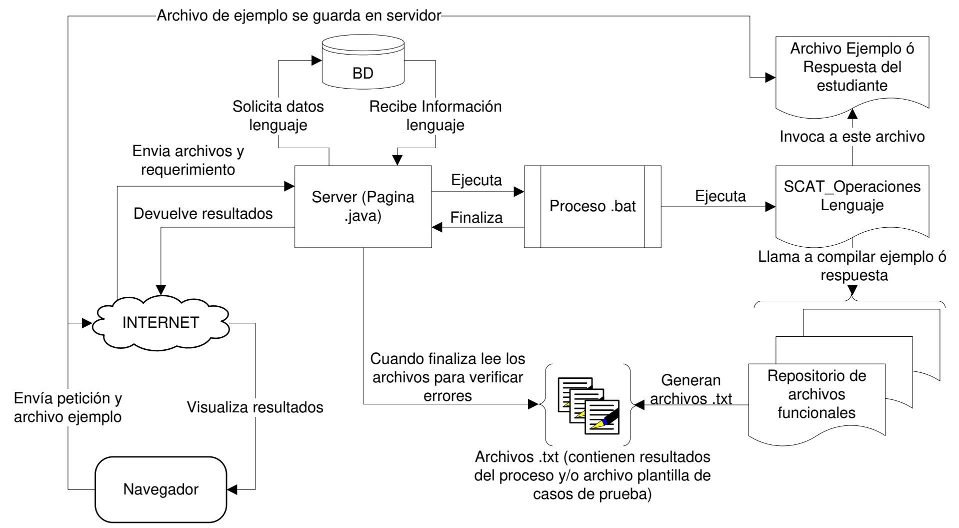


Figura 2.3: Esquema de compilación de archivo ejemplo ó archivo de respuesta y generación de plantilla de casos de prueba.

En la figura 2.4 vemos el esquema de compilación de archivo de casos de prueba subido por el profesor.

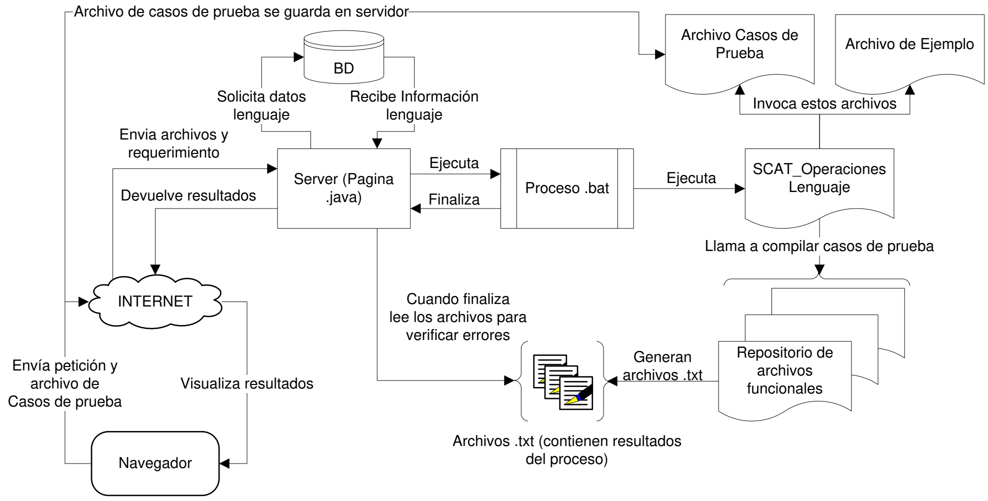


Figura 2.4: Esquema de compilación del archivo de casos de prueba.

En la figura 2.5 vemos el esquema de validación del archivo de casos de prueba.

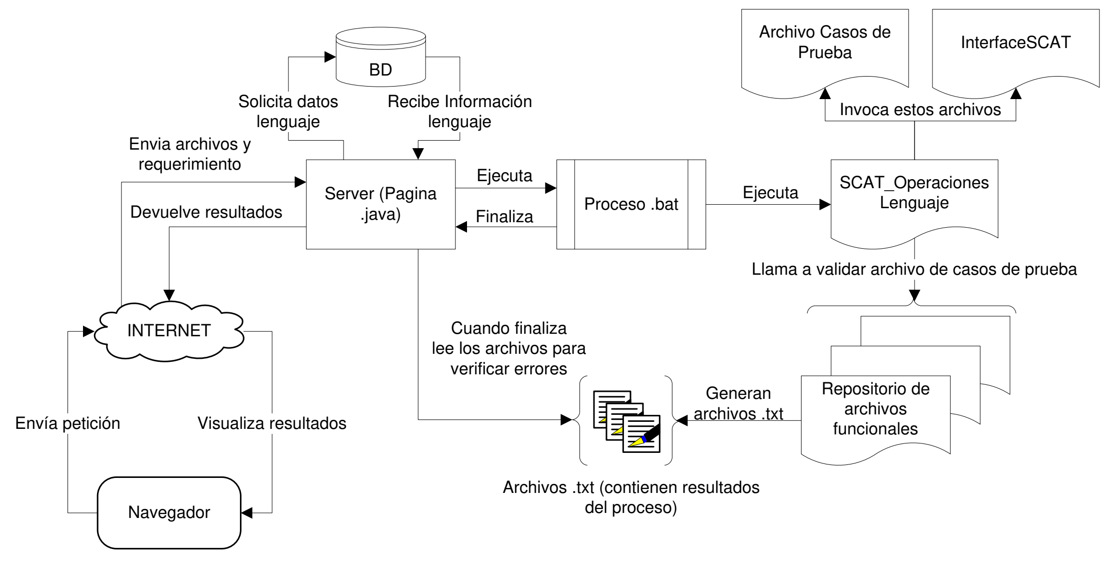


Figura 2.5: Esquema de validación del archivo de casos de prueba.

En la figura 2.6 vemos el esquema de calificación del archivo ejemplo subido por el profesor ó la respuesta subida por el estudiante.

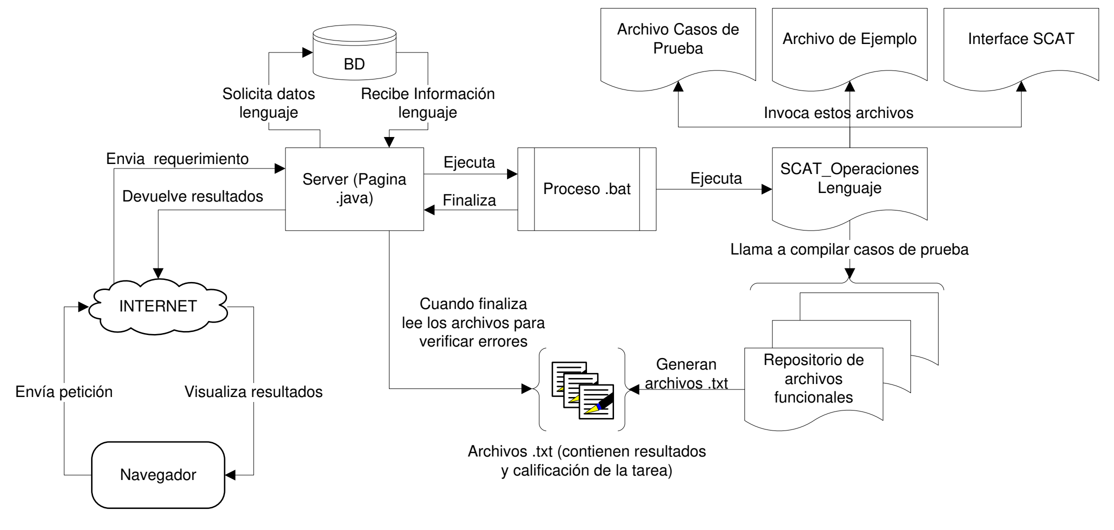


Figura 2.6: Esquema de calificación de archivo de ejemplo ó archivo de respuesta.

CAPÍTULO 3

3. IMPLEMENTACIÓN DE LA APLICACIÓN

En el presente capítulo se describe las tecnologías utilizadas en el desarrollo del Sistema Web. Se presenta el diseño detallado de la aplicación y la implementación de la base de datos.

3.1. Herramientas de desarrollo

A continuación se describen las herramientas y tecnologías utilizadas para la implementación de la aplicación y los motivos por los cuales fueron seleccionadas para el desarrollo del Sistema Web.

El sistema fue analizado para ser desarrollado como una aplicación Web por lo cual se utilizaron tecnologías basadas en la Web 2.0.

Web 2.0 es la representación de la evolución de las aplicaciones tradicionales hacia aplicaciones Web enfocadas al usuario final, no es precisamente una tecnología, sino es la actitud con la que debemos trabajar para desarrollar en Internet. La Web 2.0 busca construir aplicaciones livianas, sencillas y específicas que se pueden acceder desde cualquier computadora por medio de un navegador, fáciles de usar y con un propósito bien definido [2].

Las siguientes técnicas de la Web 2.0 fueron utilizadas para la implementación de la aplicación Web:

AJAX.- Acrónimo de Asynchronous JavaScript And XML (JavaScript y XML asíncronos, donde XML es un acrónimo de eXtensible Markup Language), es una técnica de desarrollo Web para crear aplicaciones interactivas [3]. AJAX no constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas que trabajan conjuntamente.

Las tecnologías que forman AJAX son:

- XHTML Y CSS para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la información.

- XML, XSLT Y JSON, para el intercambio y la manipulación de la información.
- XMLHttpRequest, para el intercambio asíncrono de la información.
- JavaScript, para unir todas las demás tecnologías.

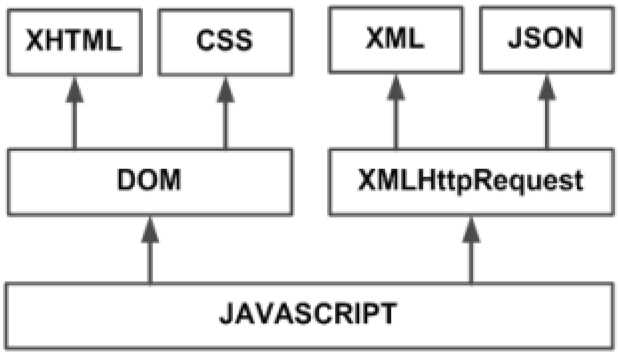


Figura 3.1: Tecnologías agrupadas bajo el concepto de AJAX. [3]

El modelo clásico de aplicaciones Web funciona de esta forma: La mayoría de las acciones del usuario en la interfaz disparan un requerimiento HTTP al servidor Web. El servidor efectúa un proceso (recopila información, procesa números, interactuando con varios sistemas propietarios), y le devuelve una pagina HTML al cliente.

Una aplicación AJAX elimina la naturaleza arrancar-frenar- arrancar-frenar de la interacción en la Web introduciendo un intermediario -un motor AJAX- entre el usuario y el servidor, de esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

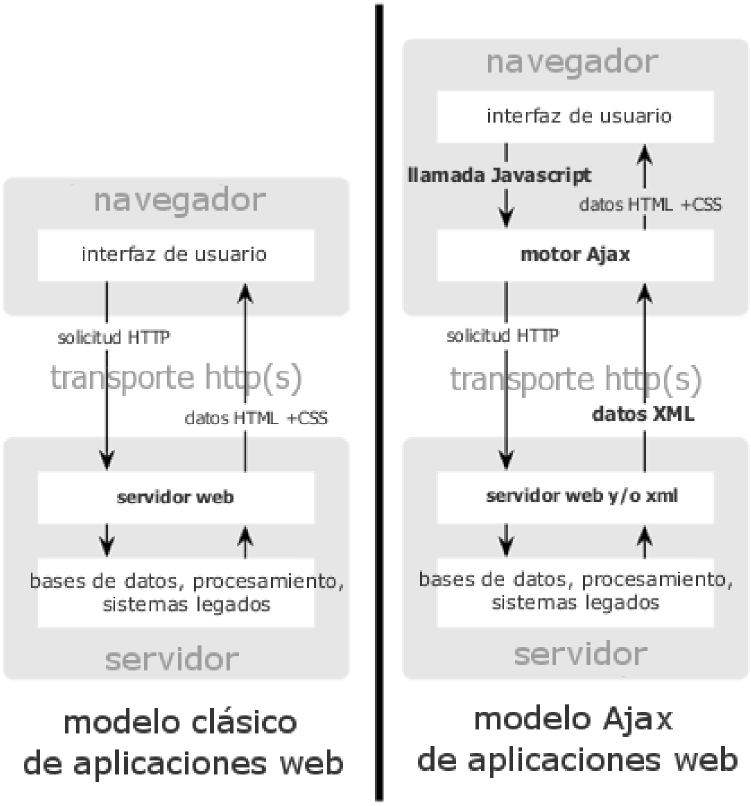


Figura 3.2: Modelo tradicional para las aplicaciones Web (izquierda) comparado con el modelo AJAX (derecha). [4]

Cada acción de un usuario, que normalmente generaría un requerimiento HTTP, toma la forma de un llamado JavaScript al motor AJAX en vez de ese requerimiento. Cualquier respuesta a una acción del usuario que no requiera un viaje de vuelta al servidor (como una simple validación de datos, edición de datos en memoria, incluso algo de navegación) es manejada por su cuenta. Si el motor necesita algo del servidor para responder (sea enviando datos para procesar, cargar código adicional, o recuperando nuevos datos) hace esos pedidos asincrónicamente, usualmente usando XML, sin frenar la interacción del usuario con la aplicación. En la figura 3.3 podemos ver la comparación entre el patrón síncrono de una aplicación Web tradicional y el patrón asíncrono de una aplicación AJAX.

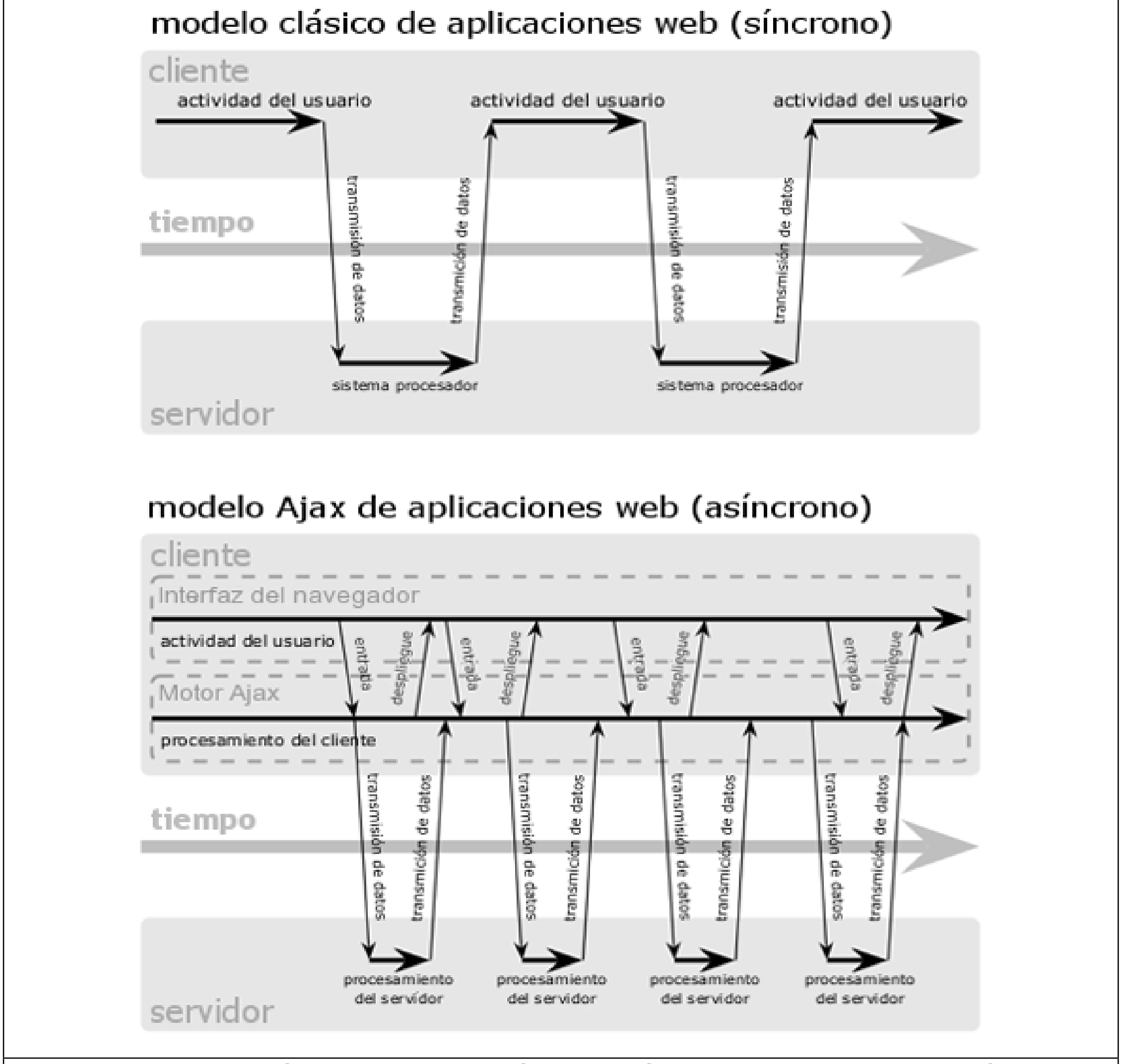


Figura 3.3: Patrón de interacción sincrónico de una aplicación Web tradicional contra el patrón asincrónico de una aplicación AJAX. [4]

JSF.- Java Server Faces es una tecnología y framework para aplicaciones Java basadas en Web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa Java Server Pages (JSP) como la tecnología que permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías como XUL [5].

JSF nos ofrece un marco de trabajo que facilita el desarrollo de aplicaciones, separando las diferentes capas de una arquitectura: presentación, reglas y entidades de negocio. Las principales características que nos ofrece JSF son:

- Un modelo de trabajo basado en componentes UI (User Interface), definidos por medio de etiquetas y XML.
- Una arquitectura basada en el patrón MVC.
- Asocia (de forma modular) cada componente gráfico con los datos (beans de respaldo).
- Incluye la capa de control, definida de forma declarativa en archivos XML. Lo que implica control de eventos y errores.
- Un modelo de eventos en el lado del servidor.
- Validación en cliente y en servidor.
- Control de mensajes y roles.

JSF dispone de varias implementaciones diferentes, incluyendo un conjunto de etiquetas y APIs estándar que forman el núcleo del framework [6].

CSS.- (Cascading Style Sheets) es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas Web complejas [7].

Separar la definición de los contenidos y la definición de su aspecto presenta muchas ventajas, debido a que permite crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "documentos semánticos"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

Al crear una página Web, se utiliza en primer lugar el lenguaje HTML/XHTML para marcar con etiquetas los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, entre otros.

Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, entre otros.

Las tecnologías que se mencionaron fueron las que se utilizaron para el desarrollo de la aplicación Web. A continuación vamos a presentar el listado de las herramientas de software que se utilizaron para la implementación de la solución del sistema Web:

- *Servidor HTTP Apache 6.0* – Servidor de aplicaciones [8].
- *Apache Tomcat 6.0.18* – Servidor Web [9].
- *MySQL Server 5.0.37* – Servidor de base de datos.
- *NetBeans IDE 6.5.1* – Interfaz de desarrollo Java.
- *Java DK 6.0 y Java EE 5.0* – Plataformas de programación.
- *JavaServer Faces (JSF) 1.2* – Plataforma de desarrollo Web.
- *Visual Web JavaServer Faces* – Framework de desarrollo JSF.
- *Dynamic Faces Components (0.2)* – Librería para desarrollo de AJAX.
- *iReport 3.5.2* – Herramienta de código abierto de Java para la elaboración de reportes [10].
- *Librería jasperreports-3.5.2* – Utilizada para la creación y manipulación de datos planos en los reportes.

- *Librería jfreechart-1.0.12* – Utilizada para la creación y manipulación de gráficos estadísticos.
- *Librería iText-2.1.0* – Utilizada para la exportación de reportes a formato pdf.
- *Microsoft Visual Studio .Net 2005* – Entorno de desarrollo.
- *Microsoft Visual Studio 6.0* – Entorno de desarrollo.

3.2. Arquitectura

La aplicación Web fue implementada utilizando programación por capas, el cual es un estilo de programación en el que el objetivo primordial es la separación de la lógica del negocio de la lógica del diseño [11]. La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y en caso de que se necesite realizar alguna modificación solo se lo realiza sobre el nivel requerido.

El diseño utilizado en la implementación de la aplicación Web fue el de programación en 3 capas (Ver figura 3.4), se lo utilizó por las ventajas mencionadas en el párrafo anterior. Las capas de este modelo son:

- Capa de presentación.
- Capa de negocio.
- Capa de datos

A continuación explicaremos brevemente la implementación de cada capa en nuestro sistema:

- ***Capa de presentación:*** Esta capa es la que presenta el sistema al usuario. También es conocida como interfaz gráfica. En el sistema la interfaz tiene las siguientes características: *el menú de opciones es una estructura de árbol bien definida*, esto ayuda a una fácil navegación del usuario en el sistema. *Cada nodo del árbol y cada hipervínculo del sistema esta acompañado de un tooltip y un icono que representa la operación que va a realizar,* es más fácil para el usuario identificar las acciones que ofrece el sistema. *La distribución de la información esta apropiadamente ubicada*, esto ayuda a una mejor lectura del contenido del sistema.

- ***Capa de negocio:*** Esta capa es donde se establecen todas las reglas que deben cumplirse en la aplicación, es decir se plasma toda lógica del negocio. En el sistema todas las operaciones que realizan los botones e hipervínculos están distribuidas en distintas clases que están organizadas de acuerdo a cada modulo del sistema. De esta forma el mantenimiento de la aplicación es más fácil de realizar y no afecta a la interfaz de la aplicación.

- ***Capa de datos:*** Esta capa es la encargada de almacenar los datos y es la que puede acceder a los mismos. La implementación la base de datos en el sistema esta conformada por 29 tablas debidamente normalizadas y 106 stored procedures que brindan las operaciones necesarias para administrar los registros de nuestra base de datos.

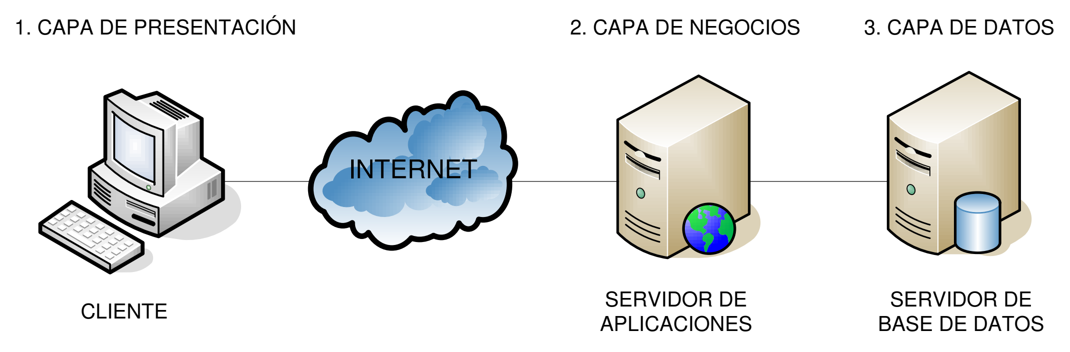


Figura 3.4: Diseño de programación en 3 capas.

En forma sintetizada las herramientas y tecnologías que deben estar instaladas y debidamente configuradas son:

Servidor	
Software	**Descripción**
Windows Server 2003 ó Windows XP	Sistema Operativo
Java DK 6.0 y Java EE 5.0	Plataforma Operativa de Java
Servidor HTTP Apache 6.0	Servidor de Aplicaciones
Apache Tomcat 6.0.18	Servidor Web (Contenedor Web)
MySQL Server 5.0.37	Servidor de Base de Datos
Framework 2.0	Plataforma Operativa de .Net
Visual C++ 6.0	Plataforma Operativa Visual C++ 6.0

Tabla 1: Software utilizado para el funcionamiento de la aplicación en el servidor.

Cliente	
Software	**Descripción**
Mozilla Firefox 3.5.1 o superior / Internet Explorer 7 o superior	Navegador Web
Java RE 6.0	Plataforma Operativa de Java

Tabla 2: Software utilizado para el funcionamiento de la aplicación en el cliente.

En resumen podemos decir que para la implementación de la aplicación se utilizó un diseño de programación de 3 capas, usando elementos la tecnología Web 2.0 tales como JSF, AJAX y CSS, también se utilizan varias herramientas de software para el buen funcionamiento de la aplicación, estos son: MYSQL, Apache 6.0, Tomcat 6.0.18, JEE 5.0, JDK 6.0, Framework 2.0, C++ 6.0.

3.2.1. Diseño detallado

El diseño detallado permite describir como el sistema va a satisfacer los requerimientos solicitados. Se realiza para tener toda la funcionalidad clara antes de implementar una aplicación. Una especificación adecuada de diseño detallado minimiza el número de sorpresas durante la implementación.

3.2.2. Casos de uso y diagrama de casos de uso

En esta sección se van a describir los principales casos de uso que fueron implementados en el sistema. Para ver todos los casos de uso del sistema refiérase al apéndice B:

Caso de Uso:	Publicar tarea.
Descripción:	Permite crear una tarea en el sistema.
Actores: • Profesor.	
Precondiciones: • El actor debió ingresar previamente al sistema. • El actor debió seleccionar algún paralelo específico para poder empezar el proceso.	
Flujo Normal: • El actor debe seguir un wizard para la creación de la tarea. Este wizard consta de 3 pasos. • Paso 1: consta en llenar los datos básicos como: nombre de tarea, descripción, fecha de publicación, fecha de entrega, lenguaje en el cual se va a implementar la tarea, y si la tarea es grupal o individual, si es grupal se especifica el máximo y mínimo número de integrantes por grupo. • Paso 2: consta un subir una posible respuesta y el archivo de pruebas, el cual se va a usar para calificar las tareas registradas por los estudiantes. • Paso 3: finalizar la publicación de la tarea. También puede configurar para que la misma tarea se publique en otros paralelos y para que los grupos, si existiesen, se puedan conformar por alumnos de diferentes paralelos. • En cada paso el sistema verifica la información ingresada.	
Flujo Alternativo: • El sistema verifica que la secuencia de pasos en el wizard sea ejecutada en el orden adecuado. • El sistema verifica en cada paso que la información necesaria esté bien ingresada, de no ser así muestra mensajes con el(los) inconveniente(s) presentado(s).	
Poscondiciones: • La tarea ha sido almacenada y publicada en el(los) paralelo(s) configurado(s).	

Caso de Uso:	Modificar tarea.
Descripción:	Permite modificar los datos de una tarea.
Actores:	• Profesor.
Precondiciones:	• El actor debió ingresar previamente al sistema. • La tarea a modificar debe existir en el sistema
Flujo Normal:	• El actor selecciona la tarea y presiona el enlace de modificación. • El sistema muestra el wizard para que el actor pueda editar los datos de la tarea. • El sistema verifica la información editada.
Flujo Alternativo:	• El sistema verifica que la secuencia de pasos en el wizard sea ejecutada en el orden adecuado. • El sistema verifica en cada paso que la información necesaria esté bien ingresada, de no ser así muestra mensajes con el(los) inconveniente(s) presentado(s).
Poscondiciones:	• La tarea ha sido modificada en el sistema.

Caso de Uso:	Importar tarea.
Descripción:	Permite importar una tarea para volverla a publicar. Puede ser una tarea anterior o una tarea que haya sido compartida por otro profesor.
Actores:	• Profesor.
Precondiciones:	• El actor debió ingresar previamente al sistema.
Flujo Normal:	• El actor selecciona la opción de importar tarea en el paralelo donde piensa publicar la tarea. • El actor consulta todas las tareas anteriores o las tareas compartidas y escoge cuál desea importar indicando la fecha de habilitación y la fecha de entrega.
Flujo Alternativo:	• Ninguno.
Poscondiciones:	• La tarea anterior queda guardada en el sistema para el paralelo escogido para luego ser publicada según la fecha de habilitación.

Caso de Uso:	Consultar tarea.
Descripción:	Permite visualizar un resumen de la tarea publicada por el profesor.
Actores: • Profesor, Alumno.	
Precondiciones: • El actor debió ingresar previamente al sistema.	
Flujo Normal: • El actor debe dar clic en el enlace para visualizar los detalles de la tarea.	
Flujo Alternativo: • Ninguno.	
Poscondiciones: • El sistema muestra toda la información necesaria y requerida por el profesor o el alumno.	

Caso de Uso:	Compartir tareas.
Descripción:	Permite compartir tareas a otros profesores.
Actores: • Profesor.	
Precondiciones: • El actor debió ingresar previamente al sistema. • El actor debió publicar la tarea a compartir en algún paralelo	
Flujo Normal: • El actor selecciona compartir tareas del menú. • El actor consulta la tarea a compartir, la selecciona y escoge el o los profesores a los cuales desea compartir la tarea.	
Flujo Alternativo: • Ninguno.	
Poscondiciones: • La tarea ha sido compartida el(los) profesor(es) seleccionado(s).	

Caso de Uso:	Consulta tareas compartidas.
Descripción:	Permite consultar cuáles tareas han sido compartidas a los profesores.
Actores: • Profesor.	
Precondiciones: • El actor debió ingresar previamente al sistema.	
Flujo Normal: • El actor selecciona compartir tareas del menú. • El actor selecciona la tarea y se marcarán los profesores a los cuales se les ha compartido la tarea.	
Flujo Alternativo: • Ninguno.	
Poscondiciones: • El sistema mostrará los profesores a los cuales se les compartió la tarea seleccionada.	

Caso de Uso:	Consultar respuestas.
Descripción:	Permite visualizar un resumen de la respuesta registrada por el estudiante.
Actores: • Profesor, Alumno.	
Precondiciones: • El actor debió ingresar previamente al sistema.	
Flujo Normal: • El actor debe dar clic en el enlace para visualizar los detalles de la respuesta de tarea subida por el estudiante.	
Flujo Alternativo: • Ninguno.	
Poscondiciones: • El sistema muestra la información de la respuesta registrada por el estudiante.	

Caso de Uso:	Ver estadísticas de tarea.
Descripción:	Permite visualizar una estadística de las respuestas registradas por los estudiantes.
Actores: • Profesor.	
Precondiciones: • El actor debió ingresar previamente al sistema.	
Flujo Normal: • El actor debe dar clic en el enlace para visualizar la estadística de las respuestas registradas por los estudiantes. • Si la tarea fue compartida para varios paralelos puede configurar para ver una estadística respecto a todos los paralelos publicados	
Flujo Alternativo: • Ninguno.	
Poscondiciones: • El sistema muestra un reporte en PDF con la estadística solicitada.	

Caso de Uso:	Responder tarea.
Descripción:	Permite registrar una respuesta a una tarea publicada.
Actores: • Alumno.	
Precondiciones: • El actor debió ingresar previamente al sistema. • Si la tarea es grupal el estudiante debió registrar su grupo previamente.	
Flujo Normal: • El actor debe dar clic en la tarea que desea responder. • El actor sube el archivo de respuesta al sistema	
Flujo Alternativo: • El sistema verifica que los datos registrados estén correctos.	
Poscondiciones: • El sistema registra la respuesta de él(los) estudiante(s), muestra la calificación obtenida y los mensajes de error, en caso de que estos existan.	

Caso de Uso:	Modificar Respuesta.
Descripción:	Permite modificar la respuesta subida al sistema.
Actores: • Alumno.	
Precondiciones: • El actor debió ingresar previamente al sistema.	
Flujo Normal: • El actor debe dar clic en la tarea que desea modificar. • El actor sube el nuevo archivo de respuesta al sistema.	
Flujo Alternativo: • El sistema verifica que los datos registrados estén correctos.	
Poscondiciones: • El sistema registra la respuesta de él(los) estudiante(s), muestra la nueva calificación obtenida y los mensajes de error, en caso de que estos existan.	

Caso de Uso:	Crear grupo.
Descripción:	Permite crear un grupo para una tarea.
Actores: • Alumno.	
Precondiciones: • El actor debió ingresar previamente al sistema.	
Flujo Normal: • El actor debe dar clic en el enlace para formar el grupo. • El sistema muestra una pantalla con la lista de estudiantes disponibles de él(los) paralelo(s) asignados a la tarea. • El actor selecciona a los estudiantes que serán parte de su grupo.	
Flujo Alternativo: • El sistema verifica la cantidad máxima y mínima de los alumnos en el grupo, si la cantidad de alumnos no es apropiada muestra un mensaje describiendo el inconveniente.	
Poscondiciones: • El grupo ha sido almacenado en el sistema.	

Caso de Uso:	Modificar grupo.
Descripción:	Permite modificar las integrantes de un grupo formado previamente.
Actores: • Alumno.	
Precondiciones: • El actor debió ingresar previamente al sistema. • La tarea no debe estar respondida.	
Flujo Normal: • El actor debe dar clic en el enlace para modificar el grupo. • El sistema muestra una pantalla con la lista de los integrantes del grupo y los estudiantes disponibles. • El actor selecciona a los estudiantes que serán parte de su grupo.	
Flujo Alternativo: • El sistema verifica la cantidad máxima y mínima de los alumnos en el grupo, si la cantidad de alumnos no es apropiada muestra un mensaje describiendo el inconveniente.	
Poscondiciones: • El grupo ha sido modificado y almacenado en el sistema.	

Caso de Uso:	Eliminar grupo.
Descripción:	Permite eliminar un grupo formado.
Actores: • Alumno.	
Precondiciones: • El actor debió ingresar previamente al sistema. • La tarea no debe estar respondida.	
Flujo Normal: • El actor debe dar clic en el enlace para eliminar el grupo.	
Flujo Alternativo: • Ninguno.	
Poscondiciones: • El grupo ha sido eliminado del sistema.	

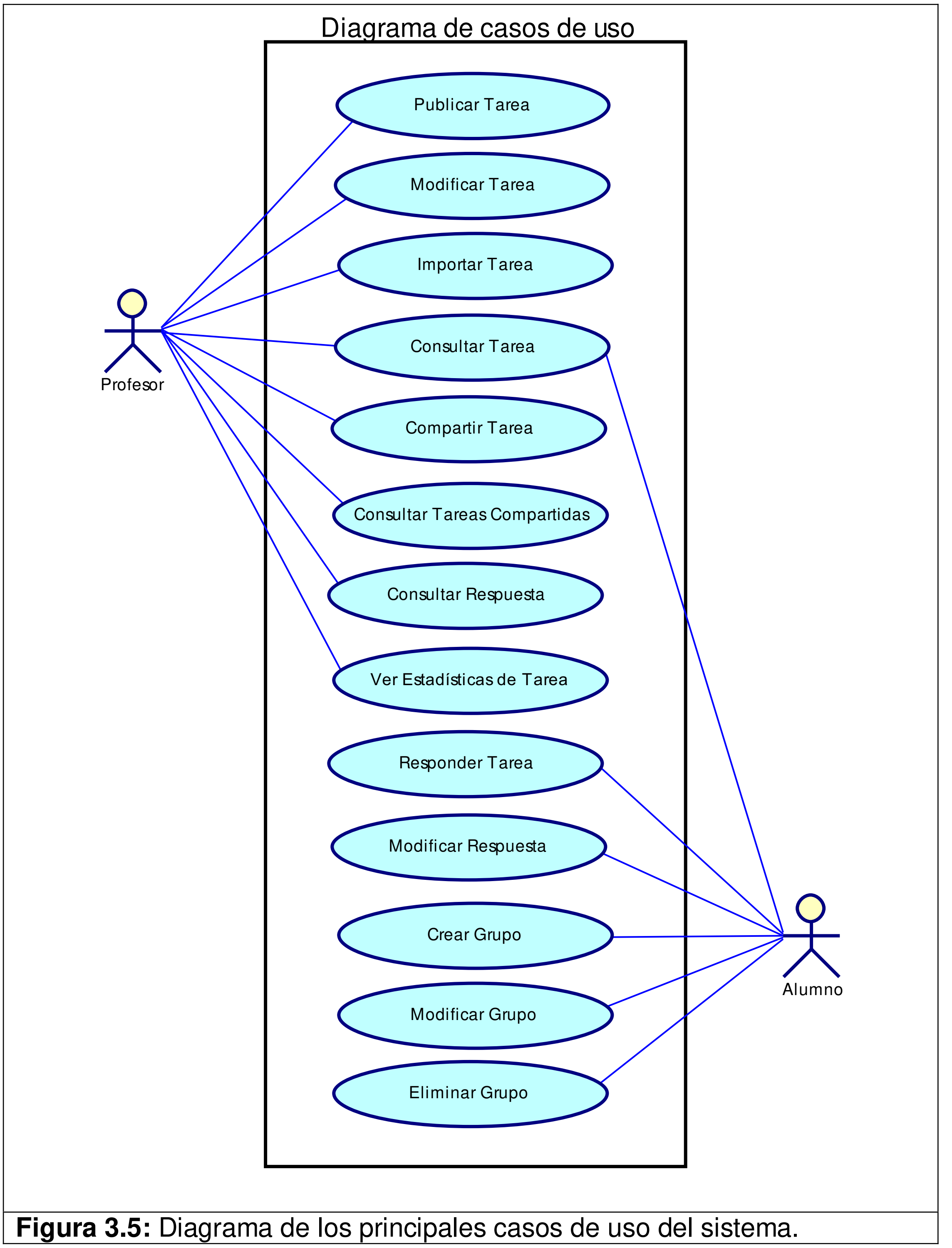


Figura 3.5: Diagrama de los principales casos de uso del sistema.

3.2.3. Diagramas de clases

A continuación el diagrama de clases del sistema. El diagrama de clases es una de las notaciones que tiene el UML, que se realiza en la etapa de diseño y describe de forma gráfica las clases de una aplicación y las asociaciones entre estas.

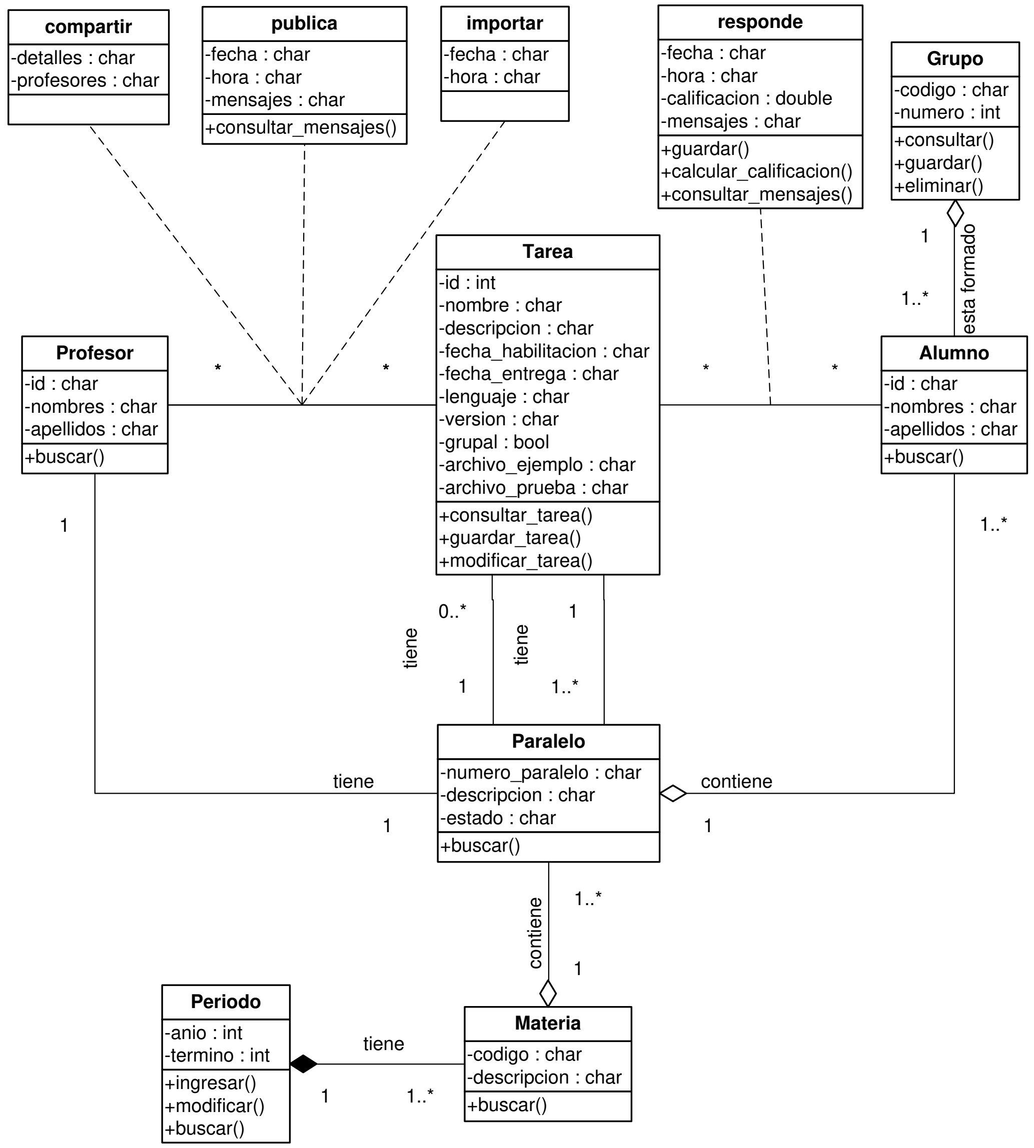


Figura 3.6: Diagrama de clases del sistema SCAT.

3.2.4. Diagramas de Secuencia

A continuación se muestran los principales diagramas de secuencia del sistema. Para ver todos los diagramas de secuencia refiérase al apéndice C:

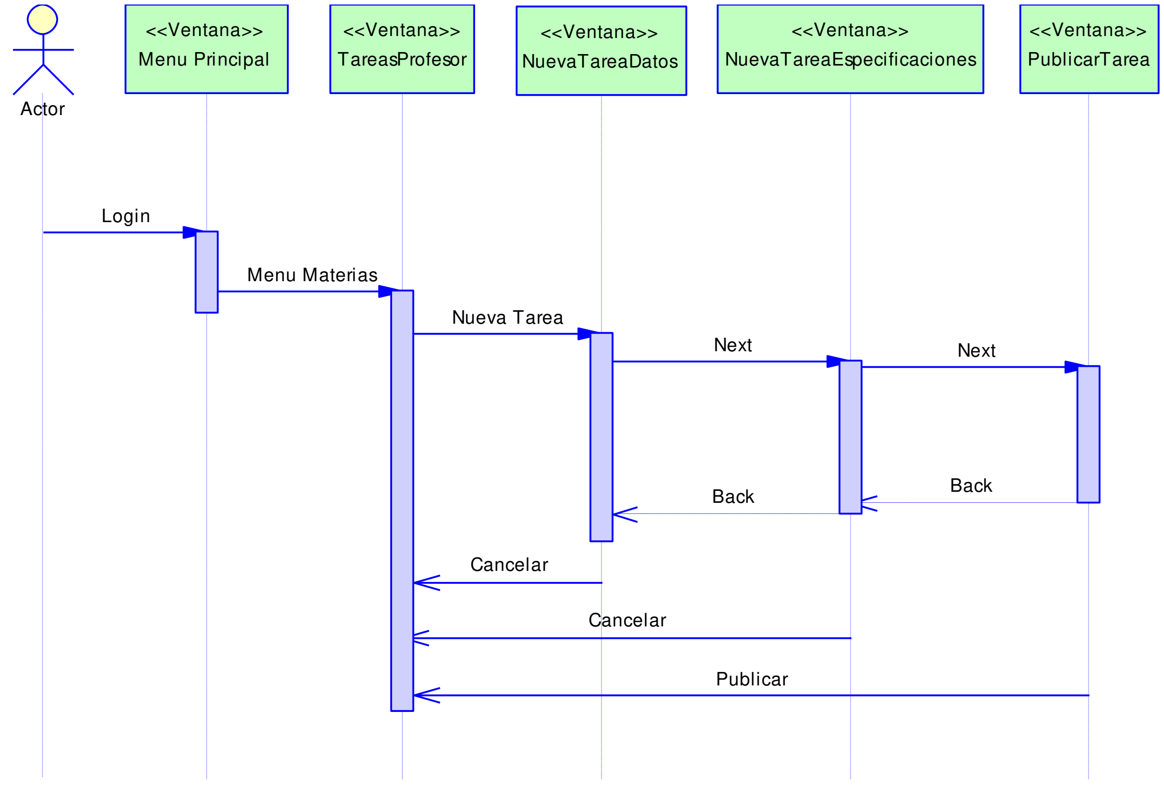


Figura 3.7: Publicar tarea.

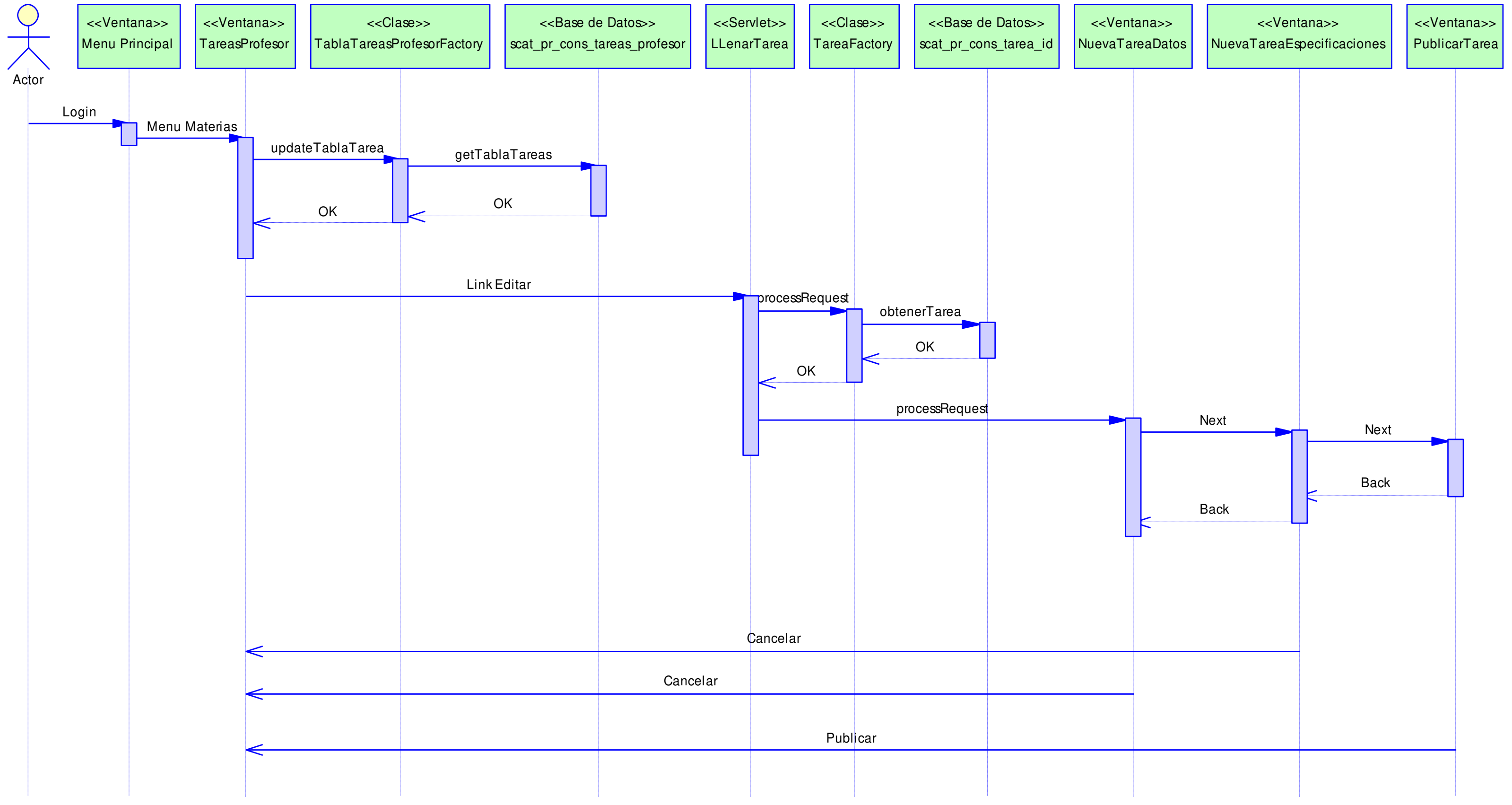


Figura 3.8: Modificar tarea.

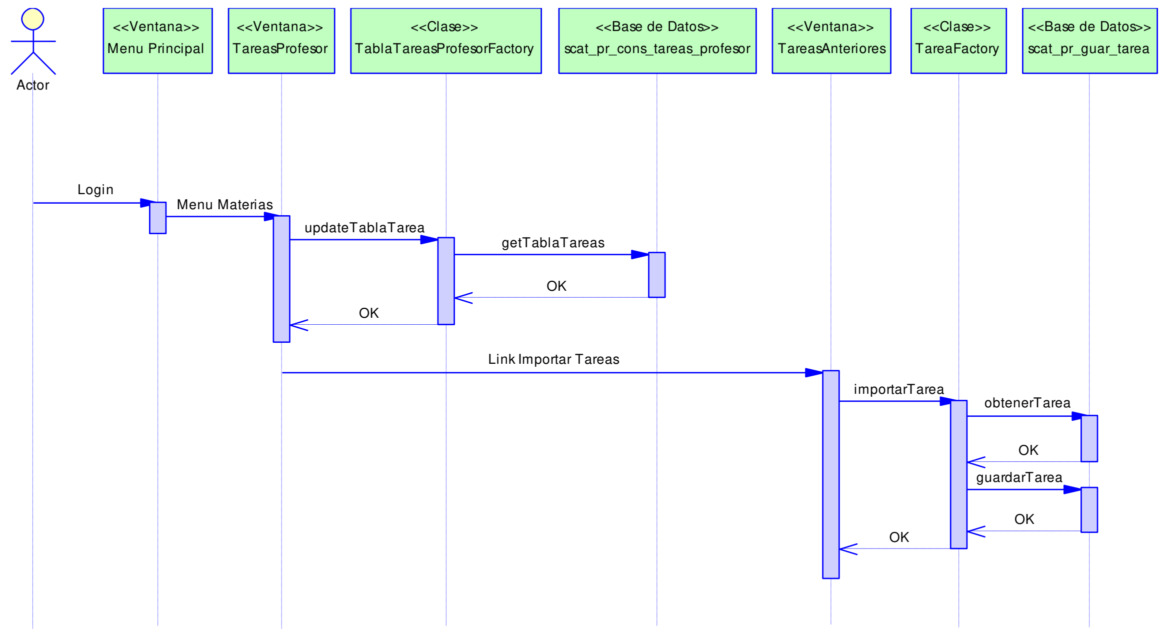


Figura 3.9: Importar tarea.

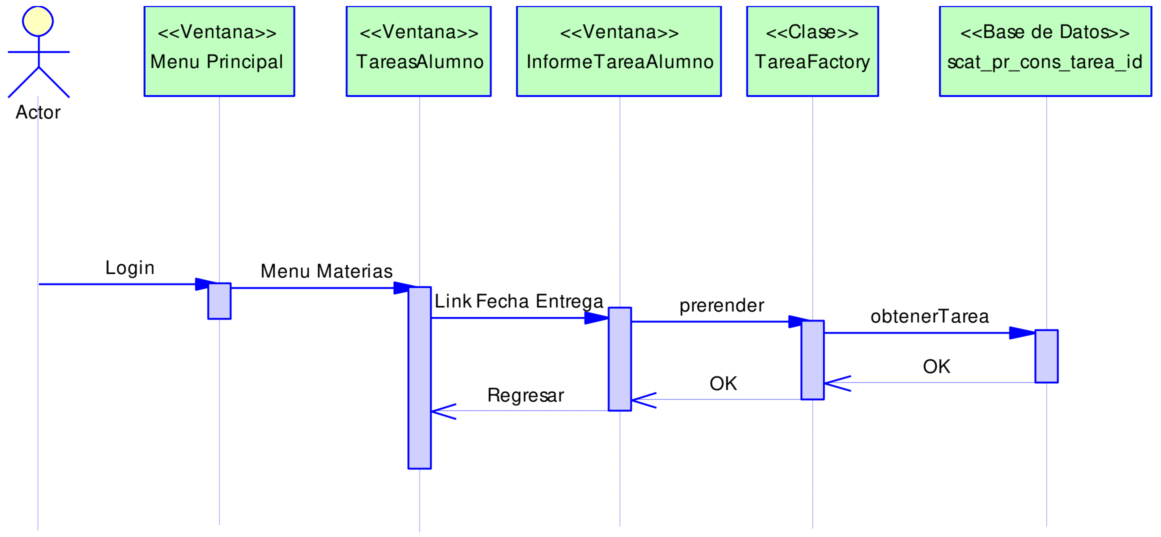


Figura 3.10: Consultar tarea - alumno

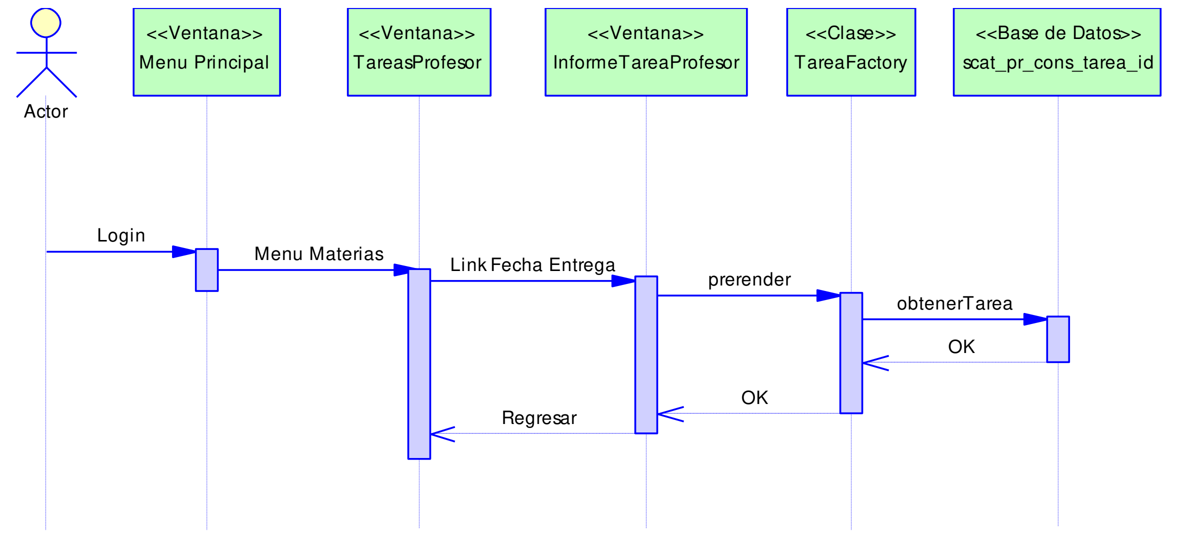


Figura 3.11: Consultar tarea - profesor.

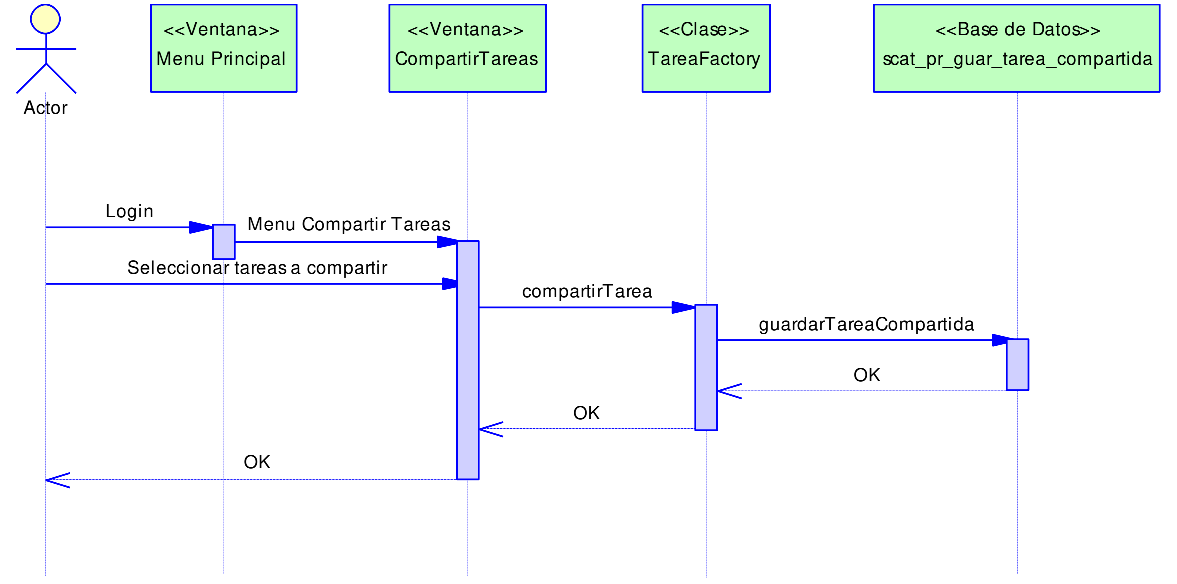


Figura 3.12: Compartir tarea.

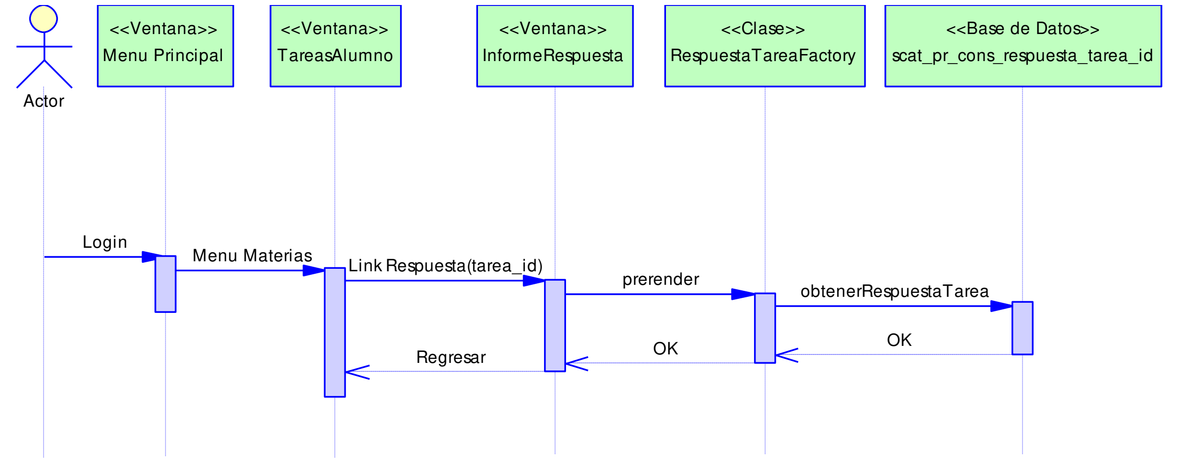


Figura 3.13: Consultar respuesta - alumno.

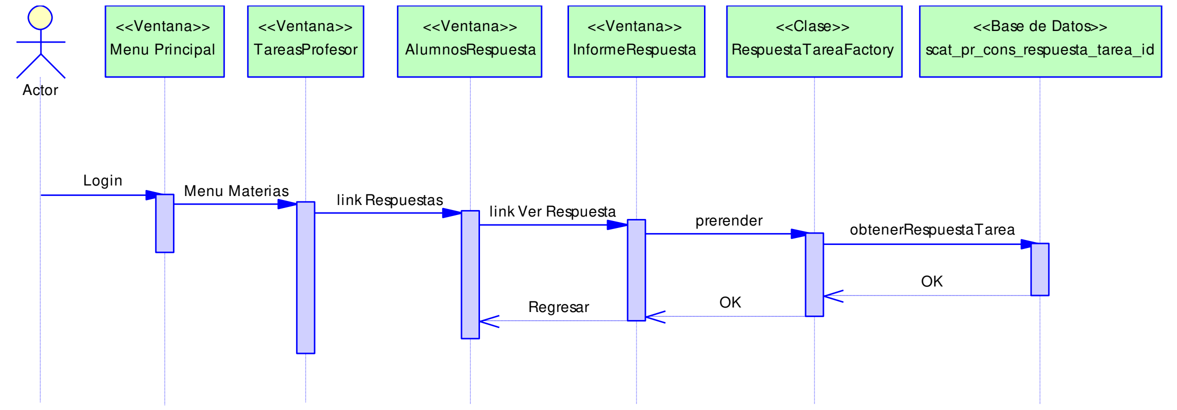


Figura 3.14: Consultar respuesta - profesor.

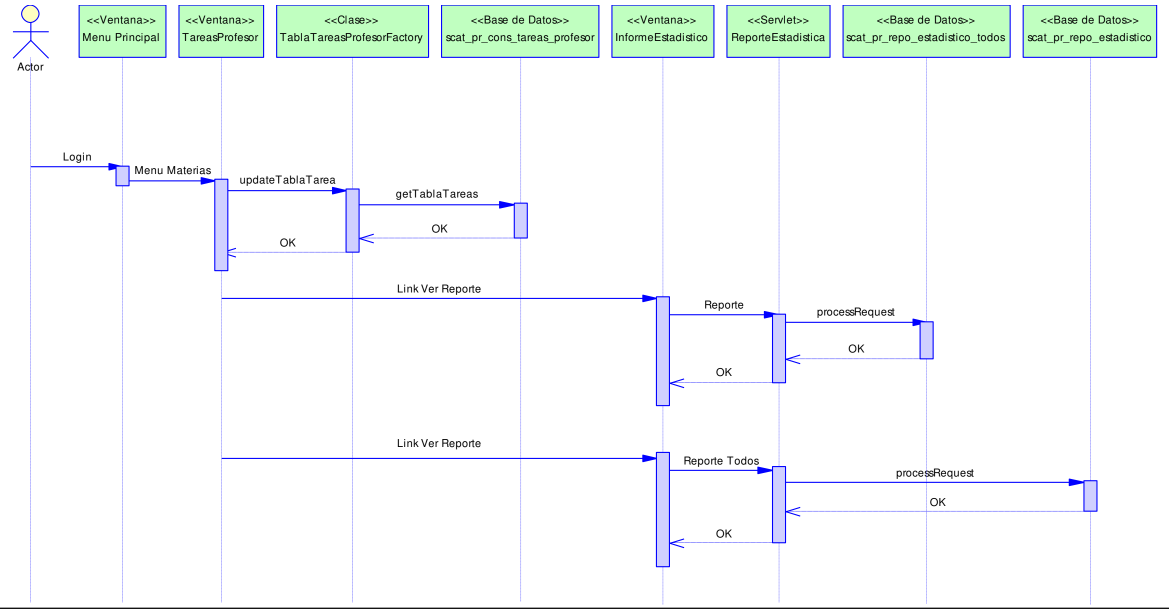


Figura 3.15: Ver estadísticas de tarea.

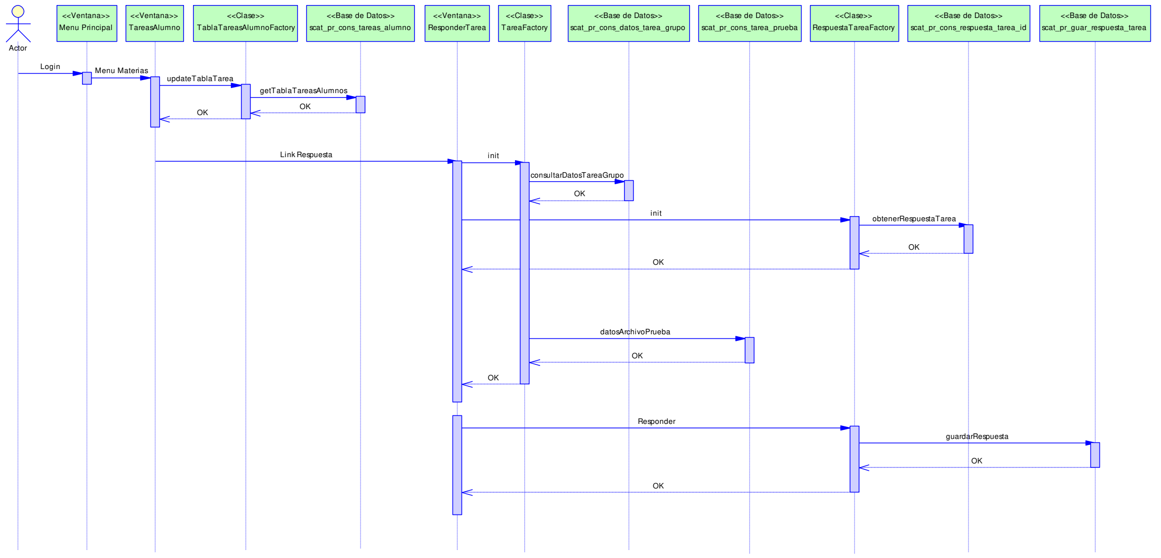


Figura 3.16: Responder tarea.

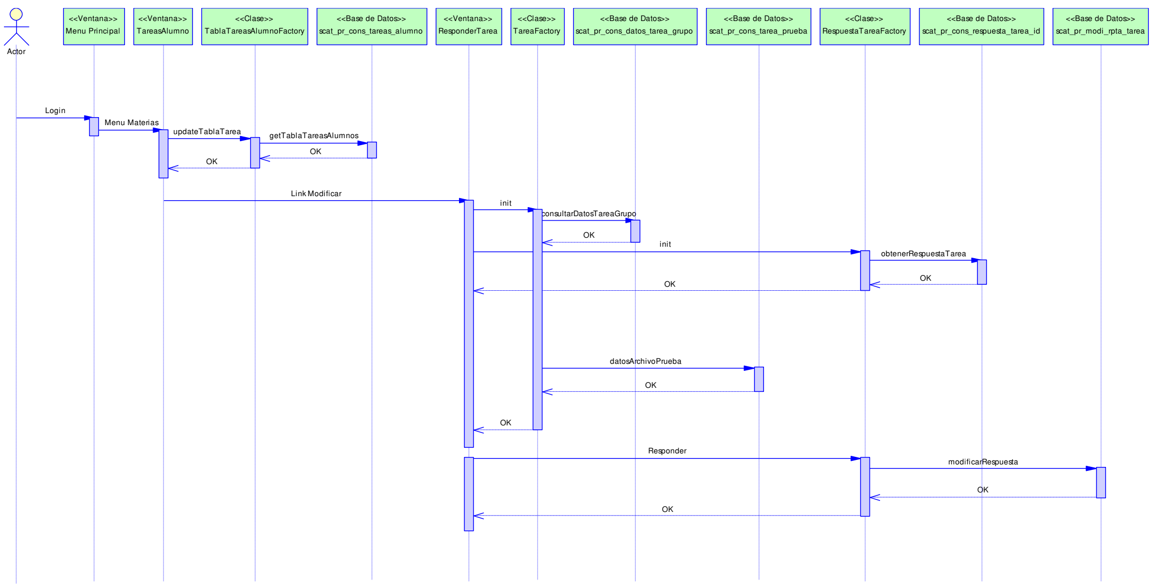


Figura 3.17: Modificar respuesta.

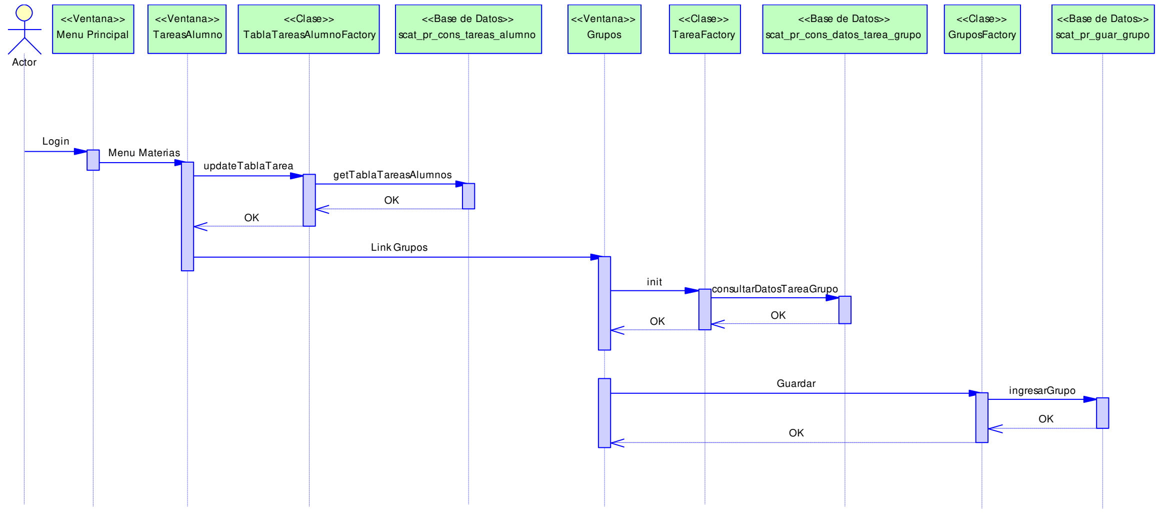


Figura 3.18: Crear grupo.

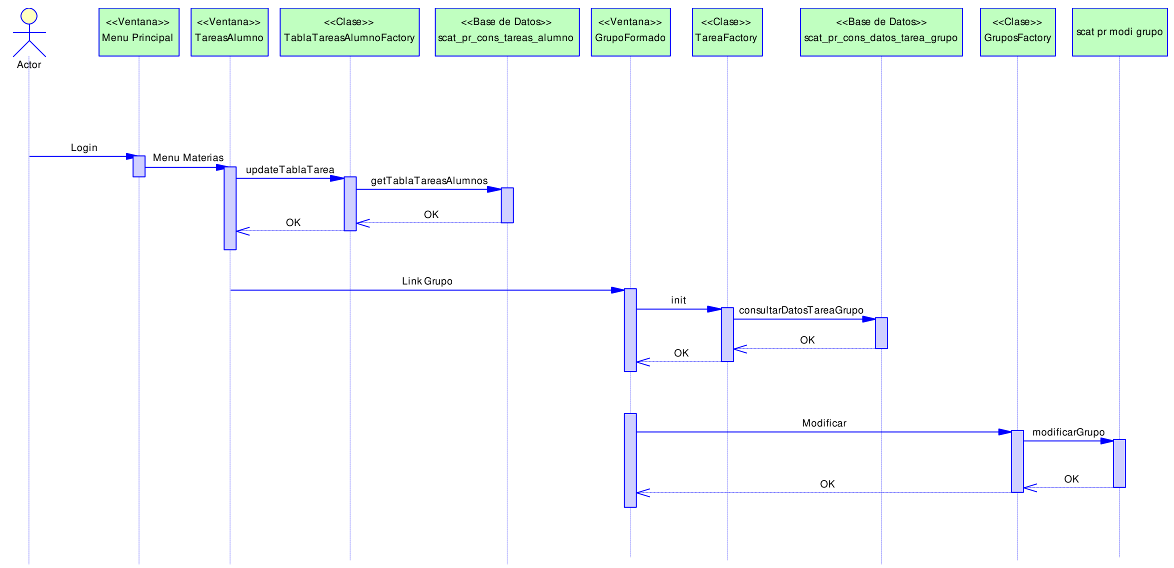


Figura 3.19: Modificar grupo.

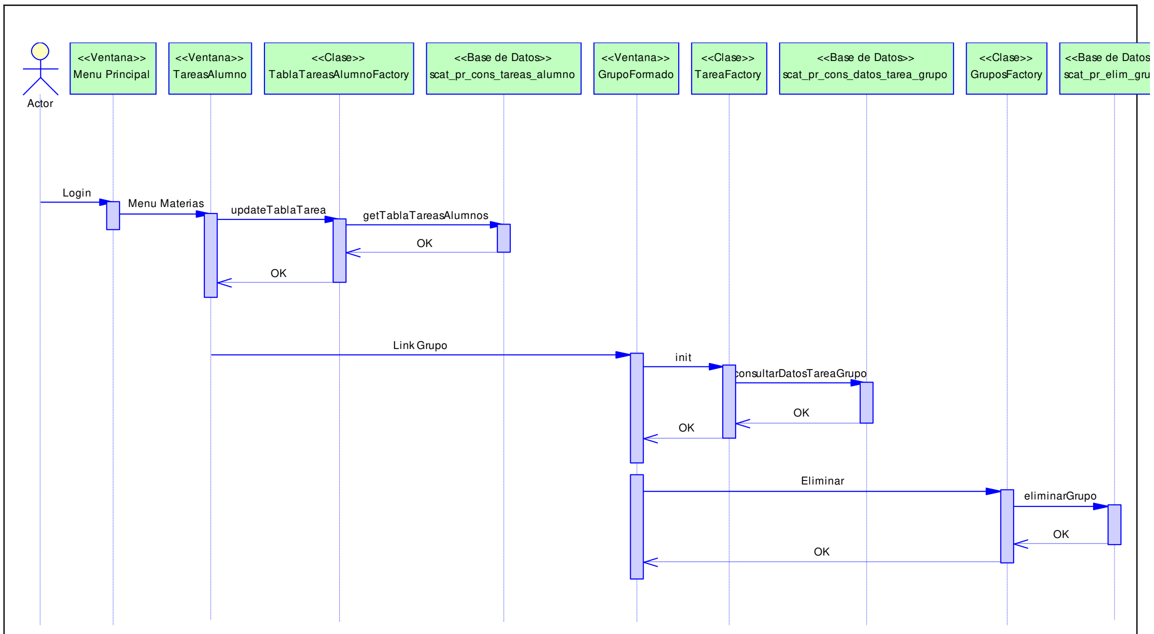


Figura 3.20: Eliminar grupo.

3.3. Diseño de la base de datos

Se escogió MySQL como base de datos del sistema, los motivos por lo cual fue seleccionada son:

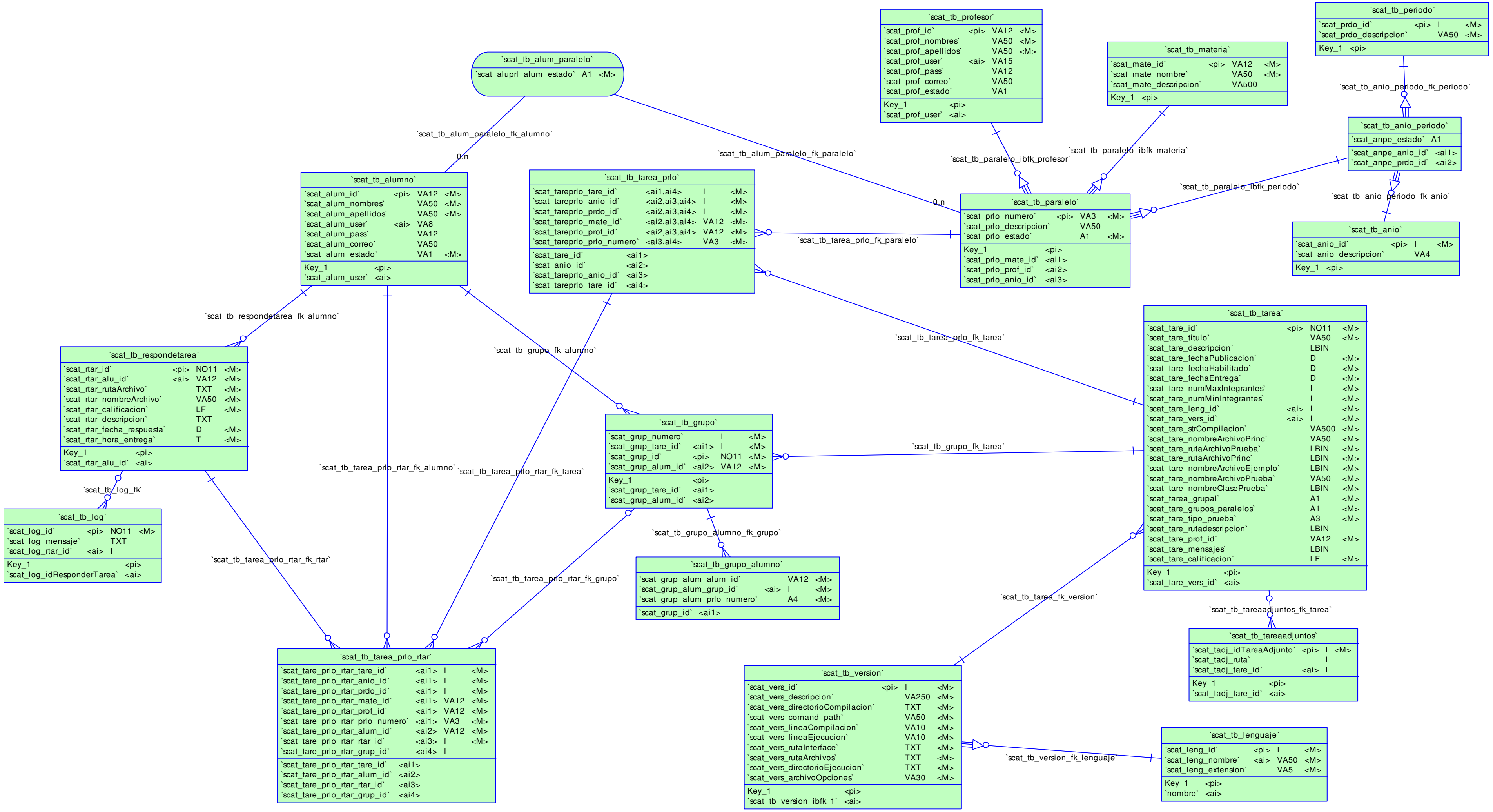
- Portabilidad.
- Escalabilidad.
- Seguridad.
- Sentencias y funciones.
- Gratuita.

A continuación presentamos el modelo físico y lógico de la base de datos. El modelo consta de 29 tablas, respectivamente relacionadas, siendo las más importantes las de publicación y respuestas de tareas.

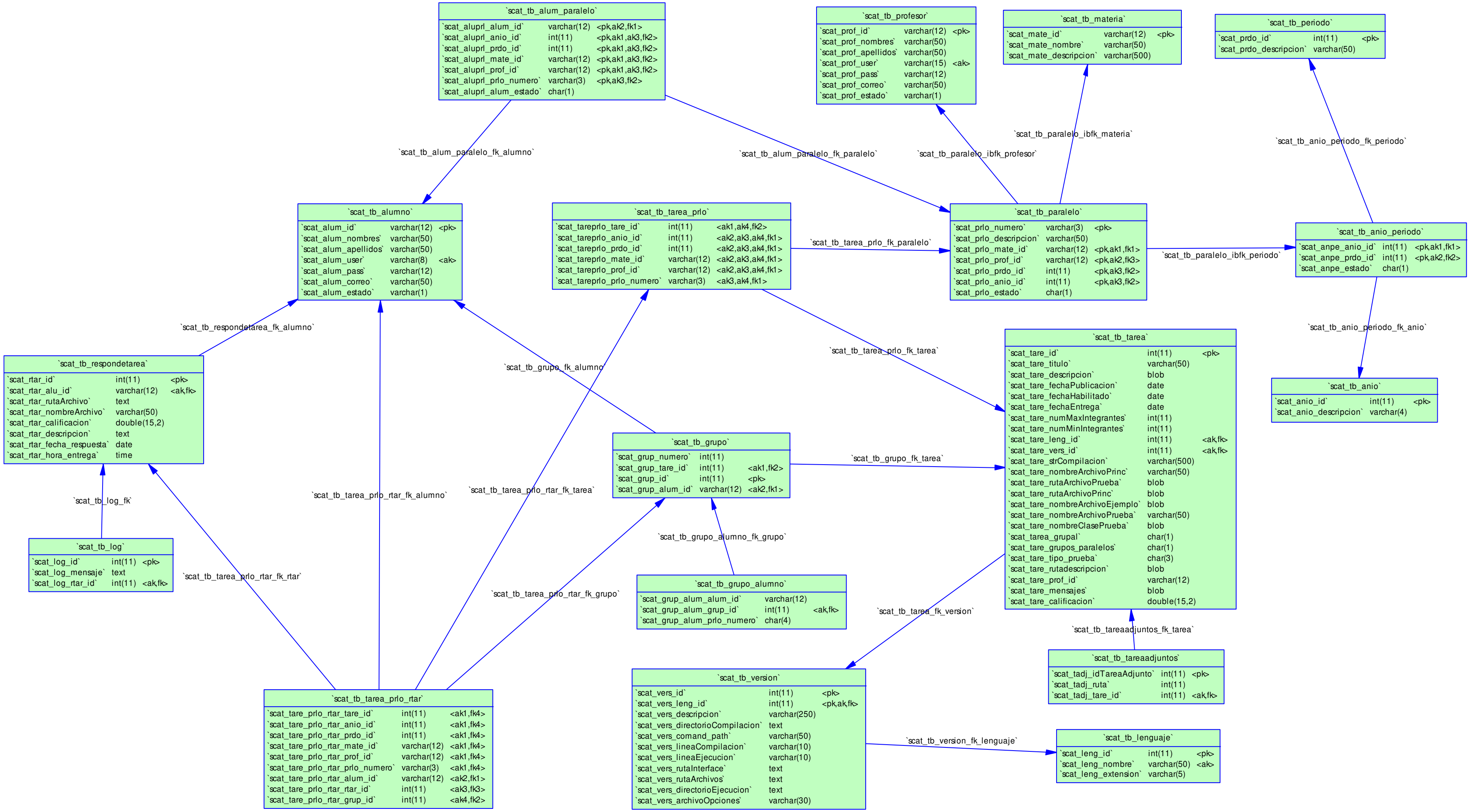
Para ver el modelo físico completo refiérase al apéndice D.

Para ver el modelo lógico completo refiérase al apéndice E.

3.3.1. Modelo físico



3.3.2. Modelo lógico



`scat_tb_alum_paralelo`		
`scat_aluprl_alum_id`	varchar(12)	<pk,ak2,fk1>
`scat_aluprl_anio_id`	int(11)	<pk,ak1,ak3,fk2>
`scat_aluprl_prdo_id`	int(11)	<pk,ak1,ak3,fk2>
`scat_aluprl_mate_id`	varchar(12)	<pk,ak1,ak3,fk2>
`scat_aluprl_prof_id`	varchar(12)	<pk,ak1,ak3,fk2>
`scat_aluprl_prlo_numero`	varchar(3)	<pk,ak3,fk2>
`scat_aluprl_alum_estado`	char(1)	

`scat_tb_profesor`		
`scat_prof_id`	varchar(12)	<pk>
`scat_prof_nombres`	varchar(50)	
`scat_prof_apellidos`	varchar(50)	
`scat_prof_user`	varchar(15)	<ak>
`scat_prof_pass`	varchar(12)	
`scat_prof_correo`	varchar(50)	
`scat_prof_estado`	varchar(1)	

`scat_tb_materia`		
`scat_mate_id`	varchar(12)	<pk>
`scat_mate_nombre`	varchar(50)	
`scat_mate_descripcion`	varchar(500)	

`scat_tb_periodo`		
`scat_prdo_id`	int(11)	<pk>
`scat_prdo_descripcion`	varchar(50)	

`scat_tb_alumno`		
`scat_alum_id`	varchar(12)	<pk>
`scat_alum_nombres`	varchar(50)	
`scat_alum_apellidos`	varchar(50)	
`scat_alum_user`	varchar(8)	<ak>
`scat_alum_pass`	varchar(12)	
`scat_alum_correo`	varchar(50)	
`scat_alum_estado`	varchar(1)	

`scat_tb_tarea_prlo`		
`scat_tareprlo_tare_id`	int(11)	<ak1,ak4,fk2>
`scat_tareprlo_anio_id`	int(11)	<ak2,ak3,ak4,fk1>
`scat_tareprlo_prdo_id`	int(11)	<ak2,ak3,ak4,fk1>
`scat_tareprlo_mate_id`	varchar(12)	<ak2,ak3,ak4,fk1>
`scat_tareprlo_prof_id`	varchar(12)	<ak2,ak3,ak4,fk1>
`scat_tareprlo_prlo_numero`	varchar(3)	<ak3,ak4,fk1>

`scat_tb_paralelo`		
`scat_prlo_numero`	varchar(3)	<pk>
`scat_prlo_descripcion`	varchar(50)	
`scat_prlo_mate_id`	varchar(12)	<pk,ak1,fk1>
`scat_prlo_prof_id`	varchar(12)	<pk,ak2,fk3>
`scat_prlo_prdo_id`	int(11)	<pk,ak3,fk2>
`scat_prlo_anio_id`	int(11)	<pk,ak3,fk2>
`scat_prlo_estado`	char(1)	

`scat_tb_anio_periodo`		
`scat_anpe_anio_id`	int(11)	<pk,ak1,fk1>
`scat_anpe_prdo_id`	int(11)	<pk,ak2,fk2>
`scat_anpe_estado`	char(1)	

`scat_tb_respondetarea`		
`scat_rtar_id`	int(11)	<pk>
`scat_rtar_alu_id`	varchar(12)	<ak,fk>
`scat_rtar_rutaArchivo`	text	
`scat_rtar_nombreArchivo`	varchar(50)	
`scat_rtar_calificacion`	double(15,2)	
`scat_rtar_descripcion`	text	
`scat_rtar_fecha_respuesta`	date	
`scat_rtar_hora_entrega`	time	

`scat_tb_tarea`		
`scat_tare_id`	int(11)	<pk>
`scat_tare_titulo`	varchar(50)	
`scat_tare_descripcion`	blob	
`scat_tare_fechaPublicacion`	date	
`scat_tare_fechaHabilitado`	date	
`scat_tare_fechaEntrega`	date	
`scat_tare_numMaxIntegrantes`	int(11)	
`scat_tare_numMinIntegrantes`	int(11)	
`scat_tare_leng_id`	int(11)	<ak,fk>
`scat_tare_vers_id`	int(11)	<ak,fk>
`scat_tare_strCompilacion`	varchar(500)	
`scat_tare_nombreArchivoPrinc`	varchar(50)	
`scat_tare_rutaArchivoPrueba`	blob	
`scat_tare_rutaArchivoPrinc`	blob	
`scat_tare_nombreArchivoEjemplo`	blob	
`scat_tare_nombreArchivoPrueba`	varchar(50)	
`scat_tare_nombreClasePrueba`	blob	
`scat_tarea_grupal`	char(1)	
`scat_tare_grupos_paralelos`	char(1)	
`scat_tare_tipo_prueba`	char(3)	
`scat_tare_rutadescripcion`	blob	
`scat_tare_prof_id`	varchar(12)	
`scat_tare_mensajes`	blob	
`scat_tare_calificacion`	double(15,2)	

`scat_tb_anio`		
`scat_anio_id`	int(11)	<pk>
`scat_anio_descripcion`	varchar(4)	

`scat_tb_grupo`		
`scat_grup_numero`	int(11)	
`scat_grup_tare_id`	int(11)	<ak1,fk2>
`scat_grup_id`	int(11)	<pk>
`scat_grup_alum_id`	varchar(12)	<ak2,fk1>

`scat_tb_log`		
`scat_log_id`	int(11)	<pk>
`scat_log_mensaje`	text	
`scat_log_rtar_id`	int(11)	<ak,fk>

`scat_tb_grupo_alumno`		
`scat_grup_alum_alum_id`	varchar(12)	
`scat_grup_alum_grup_id`	int(11)	<ak,fk>
`scat_grup_alum_prlo_numero`	char(4)	

`scat_tb_tareaadjuntos`		
`scat_tadj_idTareaAdjunto`	int(11)	<pk>
`scat_tadj_ruta`	int(11)	
`scat_tadj_tare_id`	int(11)	<ak,fk>

`scat_tb_tarea_prlo_rtar`		
`scat_tare_prlo_rtar_tare_id`	int(11)	<ak1,fk4>
`scat_tare_prlo_rtar_anio_id`	int(11)	<ak1,fk4>
`scat_tare_prlo_rtar_prdo_id`	int(11)	<ak1,fk4>
`scat_tare_prlo_rtar_mate_id`	varchar(12)	<ak1,fk4>
`scat_tare_prlo_rtar_prof_id`	varchar(12)	<ak1,fk4>
`scat_tare_prlo_rtar_prlo_numero`	varchar(3)	<ak1,fk4>
`scat_tare_prlo_rtar_alum_id`	varchar(12)	<ak2,fk1>
`scat_tare_prlo_rtar_rtar_id`	int(11)	<ak3,fk3>
`scat_tare_prlo_rtar_grup_id`	int(11)	<ak4,fk2>

`scat_tb_version`		
`scat_vers_id`	int(11)	<pk>
`scat_vers_leng_id`	int(11)	<pk,ak,fk>
`scat_vers_descripcion`	varchar(250)	
`scat_vers_directorioCompilacion`	text	
`scat_vers_comand_path`	varchar(50)	
`scat_vers_lineaCompilacion`	varchar(10)	
`scat_vers_lineaEjecucion`	varchar(10)	
`scat_vers_rutaInterface`	text	
`scat_vers_rutaArchivos`	text	
`scat_vers_directorioEjecucion`	text	
`scat_vers_archivoOpciones`	varchar(30)	

`scat_tb_lenguaje`		
`scat_leng_id`	int(11)	<pk>
`scat_leng_nombre`	varchar(50)	<ak>
`scat_leng_extension`	varchar(5)	

CAPÍTULO 4

4. PRUEBAS

El presente capítulo describe las pruebas efectuadas al sistema para verificar su funcionamiento y usabilidad. Además se muestra un análisis de los resultados de las pruebas del sistema.

4.1. Objetivos

A continuación se detallan los objetivos propuestos para la evaluación de nuestra aplicación:

General

- Apreciar la usabilidad de nuestro sistema de calificación automática de tareas de programación.

Específicos

- Valorar la operabilidad de nuestra aplicación.
- Valorar el control de usuario de nuestra aplicación.
- Valorar el diseño de la interfaz de nuestra aplicación.

4.2. Aplicación de pruebas

Las pruebas se realizaron con 10 estudiantes egresados de diferentes universidades pero que siguieron carreras afines a sistemas como son: Análisis de Sistemas, Licenciatura en Sistemas, Ingeniería en Sistemas, entre otras. Se realizaron encuestas a las personas para recopilar información que pueda mejorar nuestra aplicación. El cuestionario consta de 12 preguntas de opción múltiple enfocadas a evaluar los atributos de usabilidad los cuales son: Operabilidad, Control de Usuario y Diseño de Interfaz. En el apéndice F se puede observar el cuestionario.

Procedimiento de Pruebas:

Se accedió con un usuario que tiene el perfil de Profesor para publicar una tarea, esto se hizo con el objetivo de que los encuestados puedan responder la tarea.

El escenario que se planteó a los estudiantes fue:

"Su profesor de la materia XYZ le ha enviado una tarea ABC, usted debe conectarse al sistema SCAT, resolver la tarea y subir su respuesta al sistema". En el apéndice G se puede observar la prueba realizada.

4.3. Resultados obtenidos

A continuación detallamos los resultados obtenidos de cada una de las 12 preguntas realizadas. Los resultados han sido agrupados en relación al atributo de usabilidad que evalúan.

Existen cuatro preguntas que evalúan la operabilidad del sistema, cuatro preguntas que evalúan el control de usuario y cuatro preguntas que evalúan el diseño de la interfaz de la aplicación.

Operabilidad de la aplicación:

Evalúe el nivel de dificultad en encontrar la tarea a responder		
Respuesta	**Cantidad**	**Porcentaje**
Muy Fácil	5	50.00%
Fácil	4	40.00%
Difícil	0	0.00%
Muy difícil	1	10.00%

Evalúe el nivel de dificultad para responder la tarea una vez encontrada		
Respuesta	**Cantidad**	**Porcentaje**
Muy Fácil	5	50.00%
Fácil	4	40.00%
Difícil	0	0.00%
Muy difícil	1	10.00%

¿Tuvo algún error al responder una tarea?		
Respuesta	**Cantidad**	**Porcentaje**
Si	10	100.00 %
No	0	0.00%

Si la contestación fue afirmativa a la pregunta anterior, la ayuda que te ofrecieron los mensajes de retroalimentación del sistema fue:		
Respuesta	**Cantidad**	**Porcentaje**
Excelente	5	21.62%
Buena	4	45.95%
Mala	0	0.00%
Pésima	1	32.43%

Discusión: En el área de operabilidad se puede apreciar un gran porcentaje que considera que fue fácil usar la aplicación, al momento de realizar acciones como crear grupo, responder tarea, modificar respuesta y navegar dentro de la aplicación en si. Revisando el log de la base se puede apreciar que varios alumnos han modificado su respuesta muchas veces hasta conseguir la nota máxima, lo que nos dice que la operabilidad del sistema es bastante sencilla.

Control de Usuario de la aplicación:

Los resultados obtenidos al usar los controles (íconos / enlaces / botones / hipervínculos) que provee el sistema fueron los esperados.		
Respuesta	**Cantidad**	**Porcentaje**
Completamente de acuerdo	4	40.00%
Parcial acuerdo	5	50.00%
Parcial desacuerdo	0	0.00%
Completamente desacuerdo	1	10.00%

¿Al querer realizar una acción determinada, usted siempre sabe donde esta ubicada la opción o enlace adecuado?		
Respuesta	**Cantidad**	**Porcentaje**
Completamente de acuerdo	5	50.00%
Parcial acuerdo	3	30.00%
Parcial desacuerdo	1	10.00 %
Completamente desacuerdo	1	10.00%

¿El sistema le da la oportunidad a usted de revisar los datos de la respuesta de la tarea?		
Respuesta	**Cantidad**	**Porcentaje**
Completamente de acuerdo	6	60.00%
Parcial acuerdo	2	20.00%
Parcial desacuerdo	2	20.00%
Completamente desacuerdo	0	0.00%

¿En general qué tan buena ha sido la interfaz de navegación que ofrece el sistema?		
Respuesta	**Cantidad**	**Porcentaje**
Excelente	3	30.00%
Buena	5	50.00%
Mala	1	10.00%
Pésima	1	10.00%

Discusión: En el área de control de usuario se puede observar que la mayoría de los usuarios encontraron fácil de usar el sistema en línea, y que la interfaz de navegación fue la adecuada pero no la óptima.

Diseño de la Interfaz de la aplicación:

¿La combinación de colores que se presenta es agradable?		
Respuesta	**Cantidad**	**Porcentaje**
Completamente de acuerdo	7	70.00%
Parcial acuerdo	1	10.00%
Parcial desacuerdo	1	10.00%
Completamente desacuerdo	1	10.00%

¿Los íconos son lo suficientemente representativos para indicar su funcionalidad?		
Respuesta	**Cantidad**	**Porcentaje**
Completamente de acuerdo	5	50.00%
Parcial acuerdo	4	40.00%
Parcial desacuerdo	1	10.00%
Completamente desacuerdo	0	0.00%

¿La presentación o distribución de la información es clara?		
Respuesta	**Cantidad**	**Porcentaje**
Completamente de acuerdo	7	70.00%
Parcial acuerdo	2	20.00%
Parcial desacuerdo	1	10.00%
Completamente desacuerdo	0	0.00%

¿Los colores de fondo ofrecidos por el sistema son los adecuados?		
Respuesta	**Cantidad**	**Porcentaje**
Completamente de acuerdo	7	70.00%
Parcial acuerdo	1	10.00%
Parcial desacuerdo	1	10.00%
Completamente desacuerdo	1	10.00%

Discusión: En el área del diseño de interfaz se observa alrededor de un 70% de aceptabilidad en relación a los colores, los íconos utilizados, la distribución de las áreas, así como también la distribución de la información.

Comentarios y sugerencias dados a la aplicación

Los comentarios que más se repitieron fueron los siguientes:

- Fácil de usar.
- Interesante.
- Rápido.
- Positivo.
- Injusto.
- Comercial.
- Intuitivo.

Y las sugerencias fueron:

- Cambiar el logo del sistema.
- Íconos más representativos.
- Permitir que la fecha de publicación y la fecha de entrega contemplen horas y minutos.
- Validar que funcione en otros navegadores como Google Chrome además del Internet Explorer y el Mozilla Firefox.

Dentro de las limitantes sugirieron que se pueda modificar la tarea un limitado número de veces con el objetivo de que no se copien las tareas que tienen la mayor nota. Así mismo que el profesor tenga la capacidad de poder modificar la calificación de cierta tarea porque puede ser por ejemplo que un método dependa de otro y si el primer método está mal no importa que la lógica del segundo este bien la prueba del segundo método estaría mal puesto que el primer método está erróneo. Por ejemplo, en la tarea que se envió para hacer la encuesta, muchos tenían que la varianza era el cuadrado de la desviación estándar y es correcta la lógica, sin embargo, si el método de la desviación estándar tenía errores por ende había errores en la prueba de la varianza, los encuestados mencionaron que había injusticia en este caso.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

1) El Sistema de Calificación Automática de Tareas de Programación SCAT puede ser utilizado como un sistema de interacción entre el alumno y el profesor, para publicar tareas de programación y que sean calificadas en línea cuando el estudiante conteste su tarea. El resultado se ha obtenido después de realizar pruebas en el sistema de varias tareas de programación básica e intermedia en los diferentes lenguajes que se plantearon en un inicio; Java, C# y C++.

2) Utilizando los beneficios que nos ofrecen hasta ahora las aplicaciones distribuidas y también gracias a la ayuda de los conceptos que se aplican en los lenguajes de programación de alto nivel, se ha podido lograr automatizar el proceso de calificación de tareas de programación por medio de una aplicación Web.

3) Podemos ver que el concepto de polimorfismo que aplican los lenguajes de alto nivel fue la clave para poder avanzar en el desarrollo del sistema puesto que se tenía que lograr una forma estándar de calificación para cualquier tipo de tarea de programación.

4) Conociendo esto se utilizó el polimorfismo, aplicado a una interfaz que contendría tres métodos (setObjeto, calificar y getCalificacion) que invocaríamos para poder obtener la nota de la tarea

5) El uso de archivos por lotes es un recurso muy útil para la ejecución de procesos, y en nuestro caso lo fue para los procesos de compilación y calificación. Aplicamos esta ventaja de tal forma que los archivos por lotes se creaban de forma dinámica y se procesaban en tiempo real para poder compilar las respuestas verificando así errores de sintaxis y luego ejecutando el archivo de casos de prueba para poder realizar la calificación de la tarea.

6) La solución que implementa el sistema para la calificación de tareas es fácil; pero como sabemos no puede aplicarse a todos los lenguajes. Solamente es un camino que puede aplicarse sin problemas a los lenguajes que cumplan un concepto de polimorfismo. Puede seguir incrementándose el número de lenguajes que puede calificar el sistema mientras cumpla con la condición sin mucho cambio ni esfuerzo, pues los conceptos quedan claros después de implementar tres lenguajes, sólo basta conocer las instrucciones para la compilación y ejecución del nuevo lenguaje.

Recomendaciones

1) Como el núcleo de la tesis es el proceso de calificación y aprovechando que se trata de un sistema cliente servidor de interacción entre el alumno y profesor. Se podría aplicar en colegios, a los alumnos de informática o en una universidad para las materias de programación.

2) Mantener el concepto de cliente de servidor para utilizar las ventajas de fácil mantenimiento y menos costos al momento de hacer una actualización a la aplicación.

3) Seguir explotando los conceptos que nos ofrecen los lenguajes de alto nivel para seguir automatizando procesos y liberar carga a los actores de los diferentes sistemas creados y a los nuevos que puedan aparecer. La verdad nunca pensamos que esto se podía lograr sin usar algún tipo de lenguaje que use inteligencia artificial como premisa. Pero no solamente usar los conceptos de los lenguajes, sino combinarlos con otras herramientas tales como procesos batch, tareas programadas, entre otros, para seguir automatizando los procesos.

4) Tomando en cuenta la solución que se tomó para implementar la calificación en línea de las tareas de programación, se recomienda ir integrando un módulo de forma independiente que se encargue de solucionar otros tipos de lenguajes pero sin tener que cambiar la operativa, de tal forma que el nuevo módulo sea transparente para el usuario final.

APENDICE A: GLOSARIO

- **Archivo de Ejemplo:** Es el archivo registrado por el profesor, el cual representa la posible solución de la tarea a publicar. Este archivo debe especificar claramente los métodos porque estos serán implementados por el estudiante al momento de responder la tarea.

- **Archivo de Casos de Prueba:** Es el archivo registrado por el profesor, este contiene los diferentes casos de prueba al cual serán sometidos las posibles soluciones registradas por los estudiantes. Este archivo debe seguir un formato específico y debe contener ciertos métodos necesarios para que los procesos de calificación se realicen de forma exitosa.

- **Archivo de Respuesta:** Es el archivo registrado por el estudiante, el cual contiene la posible solución del problema establecido, este archivo debe contener los métodos definidos por el profesor.

- **SCAT_OperacionesLenguajes:** Este archivo es el que manda a ejecutar las operaciones necesarias para que se lleven a cabo los procesos de publicación y respuesta de las tareas.

- **InterfaceSCAT:** Este archivo contiene los métodos que obligatoriamente debe implementar el “archivo de casos de prueba”.

- **Polimorfismo subtipado:** Es la capacidad de redefinir un método en las clases que heredan de una clase base [12].

- **XML:** Extensible Markup Language (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas [13].

- **AJAX:** Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. Es una técnica de desarrollo Web para crear aplicaciones interactivas [14].

- **JSF:** Java Server Faces es una tecnología y Framework para aplicaciones Java basadas en Web que simplifica el desarrollo de interfaces de usuario [15].

- **MVC:** Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos [16].

- **Classpath:** Se entiende por classpath una opción que indica a la maquina virtual de Java (JVM) donde buscar paquetes y clases definidas por el usuario a la hora de ejecutar programas [17].

APÉNDICE B: CASOS DE USO DEL SISTEMA

Caso de Uso:	Crear catálogo
Descripción:	Permite crear tabla de catálogos que son usados para no crear tablas innecesarias en la base de datos.
Actores: • Administrador	
Precondiciones: • El actor debió ingresar previamente al sistema.	
Flujo Normal: • El actor pulsa un botón para crear un nuevo catálogo. • El sistema habilita cajas de texto para introducir la información necesaria para crear el catálogo. • El actor introduce los datos necesarios del catálogo.	
Flujo Alternativo: • El sistema comprueba que el código del catálogo no exista en la base de datos, si ya existe muestra un mensaje notificándole del error.	
Poscondiciones: • El catálogo ha sido almacenado en el sistema.	

Caso de Uso:	Modificar catálogo
Descripción:	Permite modificar los datos de las tablas de catálogos que existen en el sistema.
Actores: • Administrador	
Precondiciones: • El actor debió ingresar previamente al sistema. • El catálogo a modificar debe existir en el sistema.	
Flujo Normal: • El actor busca el catálogo y carga sus datos en las cajas de texto. • El actor edita los datos del catálogo a modificar. • El sistema comprueba la validez de los datos del catálogo.	
Flujo Alternativo: • El sistema comprueba la validez de los datos, si existe error muestra un mensaje mostrando el inconveniente.	
Poscondiciones: • El catálogo ha sido modificado en el sistema.	

Caso de Uso:	Crear definición de catálogo
Descripción:	Permite crear los registros dentro de una tabla de catálogos específica.
Actores: • Administrador	
Precondiciones: • El actor debió ingresar previamente al sistema. • La tabla de catálogo a la cual se le va a crear el registro debe existir en el sistema.	
Flujo Normal: • El actor pulsa un botón para crear una nueva definición de catálogo. • El sistema habilita cajas de texto para introducir la información de la definición de catálogo a una tabla de catálogos específica. • El sistema comprueba la validez del código de definición de catálogo para poder almacenarlo.	
Flujo Alternativo: • El sistema comprueba que el código de la definición de catálogo no exista en la base de datos, si ya existe muestra un mensaje notificándole del error.	
Poscondiciones: • La definición de catálogo ha sido almacenado en el sistema.	

Caso de Uso:	Modificar módulo
Descripción:	Permite modificar los datos de los módulos que existen en el sistema.
Actores: • Administrador	
Precondiciones: • El actor debió ingresar previamente al sistema. • El módulo a modificar debe existir en el sistema.	
Flujo Normal: • El actor busca el módulo y carga sus datos en las cajas de texto. • El actor edita los datos del modulo a modificar. • El sistema comprueba la validez de los datos del modulo.	
Flujo Alternativo: • El sistema comprueba la validez de los datos, si existe error muestra un mensaje mostrando el inconveniente.	
Poscondiciones: • El módulo ha sido modificado en el sistema.	

Caso de Uso:	Crear transacciones
Descripción:	Permite crear las transacciones del sistema. Las transacciones son las diferentes opciones que accedan los usuarios.
Actores: • Administrador	
Precondiciones: • El actor debió ingresar previamente al sistema. • El módulo al cual se le va a crear la transacción debe existir en el sistema.	
Flujo Normal: • El actor pulsa un botón para crear una nueva transacción. • El sistema habilita cajas de texto para introducir la información necesaria de la transacción. • El sistema comprueba la validez del código y de los datos de la transacción para poder almacenarlo.	
Flujo Alternativo: • El sistema comprueba que el código de la transacción no exista en la base de datos, si ya existe muestra un mensaje notificándole del error.	
Poscondiciones: • La transacción ha sido almacenado en el sistema.	

Caso de Uso:	Modificar transacciones
Descripción:	Permite modificar las transacciones que existen en el sistema.
Actores: • Administrador	
Precondiciones: • El actor debió ingresar previamente al sistema. • La transacción a modificar debe existir en el sistema.	
Flujo Normal: • El actor busca la transacción y carga sus datos en cajas de texto. • El actor edita los datos de la transacción. • El sistema comprueba la validez de los datos de la transacción.	
Flujo Alternativo: • El sistema comprueba la validez de los datos, si existe error muestra un mensaje mostrando el inconveniente.	
Poscondiciones: • La transacción ha sido modificada en el sistema.	

<table>
<tr><td>Caso de Uso:</td><td>Crear perfiles de usuario</td></tr>
<tr><td>Descripción:</td><td>Permite crear los perfiles de usuarios. Los perfiles son las plantillas de opciones que puede tener cada usuario para trabajar en el sistema.</td></tr>
<tr><td colspan="2">Actores:
• Administrador</td></tr>
<tr><td colspan="2">Precondiciones:
• El actor debió ingresar previamente al sistema.</td></tr>
<tr><td colspan="2">Flujo Normal:
• El actor pulsa un botón para crear un nuevo perfil.
• El sistema habilita cajas de texto para introducir la información necesaria del perfil de usuario.
• El sistema comprueba la validez del código y de los datos del perfil de usuario para poder almacenarlo.</td></tr>
<tr><td colspan="2">Flujo Alternativo:
• El sistema comprueba que el código del perfil de usuario no exista en la base de datos, si ya existe muestra un mensaje notificándole del error.</td></tr>
<tr><td colspan="2">Poscondiciones:
• El perfil de usuario ha sido almacenado en el sistema.</td></tr>
</table>

<table>
<tr><td>Caso de Uso:</td><td>Modificar perfiles de usuario</td></tr>
<tr><td>Descripción:</td><td>Permite modificar los perfiles de usuario que existen en el sistema.</td></tr>
<tr><td colspan="2">Actores:
• Administrador</td></tr>
<tr><td colspan="2">Precondiciones:
• El actor debió ingresar previamente al sistema.
• El perfil a modificar debe existir en el sistema.</td></tr>
<tr><td colspan="2">Flujo Normal:
• El actor busca el perfil de usuario y carga sus datos en cajas de texto.
• El actor edita los datos del perfil de usuario.
• El sistema comprueba la validez de los datos del perfil de usuario.</td></tr>
<tr><td colspan="2">Flujo Alternativo:
• El sistema comprueba la validez de los datos, si existe error muestra un mensaje mostrando el inconveniente.</td></tr>
<tr><td colspan="2">Poscondiciones:
• El perfil de usuario ha sido modificado en el sistema.</td></tr>
</table>

Caso de Uso:	Asignar transacciones a perfiles
Descripción:	Permite asignar las diferentes transacciones a los perfiles de usuarios del sistema.
Actores:	• Administrador
Precondiciones:	• El actor debió ingresar previamente al sistema.
Flujo Normal:	• El actor selecciona el modulo y el perfil de usuario para consultar las transacciones. • El actor seleccionar las transacciones que desea se asignen al perfil seleccionado.
Flujo Alternativo:	• Ninguno.
Poscondiciones:	• La asignación de la transacción ha sido almacenado en el sistema.

Caso de Uso:	Migrar datos.
Descripción:	Permite migrar archivos de texto a las tablas de migración en la base de datos.
Actores:	• Administrador, supervisor.
Precondiciones:	• El actor debió ingresar previamente al sistema.
Flujo Normal:	• El actor selecciona el sistema, la tabla, el script a ejecutar y el archivo que desea migrar. El actor puede seleccionar si sube un archivo para una tabla o para todas las tablas. • El sistema verifica si escoge una tabla y migra el archivo según el formato del script escogido. Si escoge todas las tablas se debe seleccionar a que archivo corresponde cada tabla con su respectivo script para migrar. • El sistema muestra el número de registros migrados a cada tabla.
Flujo Alternativo:	• El sistema migra el archivo según el script escogido a la base de datos, si existe un error muestra un mensaje mostrando el inconveniente.
Poscondiciones:	• Los archivos con los datos son migrados a la base del sistema.

Caso de Uso:	Ingresar usuario.
Descripción:	Permite crear usuarios en el sistema.
Actores: • Administrador, supervisor.	
Precondiciones: • El actor debió ingresar previamente al sistema.	
Flujo Normal: • El actor presiona un botón para crear un nuevo usuario. • El sistema habilita campos de texto para que ingrese los datos necesarios del usuario. • El sistema comprueba la validez del código de usuario y de la información necesaria.	
Flujo Alternativo: • El sistema comprueba la validez del código de usuario ingresado, si ya existe envía un mensaje de error notificando el inconveniente.	
Poscondiciones: • El nuevo usuario ha sido almacenado en el sistema.	

Caso de Uso:	Modificar usuario.
Descripción:	Permite modificar los datos de los usuarios existentes en el sistema.
Actores: • Administrador, supervisor.	
Precondiciones: • El actor debió ingresar previamente al sistema. • El usuario a modificar debe existir en el sistema.	
Flujo Normal: • El actor busca el usuario y carga sus datos en la cajas de texto. • El actor edita los datos del usuario. • El sistema comprueba la validez de los datos ingresados.	
Flujo Alternativo: • El sistema comprueba la validez de los datos, si existe error muestra un mensaje mostrando el inconveniente.	
Poscondiciones: • El usuario ha sido modificado en el sistema.	

Caso de Uso:	Homologar datos.
Descripción:	Permite migrar de las tablas de migración a las tablas de trabajo del sistema.
Actores: • Administrador, supervisor.	
Precondiciones: • El actor debió ingresar previamente al sistema. • Los archivos de datos de migración deben registrarse previamente en el sistema.	
Flujo Normal: • El actor debe seleccionar la tabla que desea procesar y junto a ellos debe seleccionar el año y el término. • El sistema migra los datos de las tablas de migración a las tablas de trabajo del sistema en un proceso asincrónico y envía un correo al actor con los resultados.	
Flujo Alternativo: • Ninguno.	
Poscondiciones: • Las tablas del sistema se encuentran con los datos homologados.	

Caso de Uso:	Agregar y quitar alumno-paralelo.
Descripción:	Permite agregar o quitar alumnos de algún paralelo en un año y termino específico.
Actores: • Administrador, supervisor.	
Precondiciones: • El actor debió ingresar previamente al sistema.	
Flujo Normal: • El actor busca un alumno y lo agrega a un paralelo. También puede cambiar el estado de un alumno en un paralelo específico. • El sistema verifica, en el caso de ser una asignación a un paralelo, que el alumno no exista en el paralelo.	
Flujo Alternativo: • El sistema verifica que el usuario no exista en un paralelo cuando se esta haciendo una asignación de alumno-paralelo. Si existe algún error muestra un mensaje mostrando e inconveniente.	
Poscondiciones: • El alumno ha sido asignado o eliminado del paralelo.	

Caso de Uso:	Ingresar Periodos.
Descripción:	Permite crear períodos en el sistema.
Actores: • Administrador, supervisor.	
Precondiciones: • El actor debió ingresar previamente al sistema.	
Flujo Normal: • El actor presiona un botón para crear un nuevo período. • El sistema habilita campos de texto para que ingrese los datos necesarios del periodo. • El sistema comprueba la validez del año y del término a ingresar.	
Flujo Alternativo: • El sistema verifica que el año y el término no existan en el sistema, si existen muestra un mensaje mostrando el inconveniente.	
Poscondiciones: • El periodo ha sido almacenado en el sistema.	

Caso de Uso:	Modificar Periodos.
Descripción:	Permite modificar el estado de un período en el sistema.
Actores: • Administrador, supervisor.	
Precondiciones: • El actor debió ingresar previamente al sistema.	
Flujo Normal: • El actor busca el periodo y carga sus datos en la caja de texto. • El actor modifica el estado del periodo. • El sistema verifica el nuevo estado del periodo.	
Flujo Alternativo: • En el caso de activar un periodo el sistema verifica que no exista otro periodo activo, de ser así muestra un mensaje mostrando el inconveniente.	
Poscondiciones: • El estado del periodo ha sido modificado en el sistema.	

Caso de Uso:	Activar y desactivar paralelo.
Descripción:	Permite modificar el estado de un paralelo en el sistema.
Actores:	• Administrador, supervisor.
Precondiciones:	• El actor debió ingresar previamente al sistema.
Flujo Normal:	• El actor busca el selecciona el año, termino y la materia en la cual desea cambiar el estado de algún paralelo. • El actor modifica el estado del paralelo.
Flujo Alternativo:	• Ninguno.
Poscondiciones:	• El estado del paralelo ha sido modificado en el sistema.

Caso de Uso:	Crear Configurar migración.
Descripción:	Permite crear configuraciones para la migración de archivos de texto a las tablas de migración en la base.
Actores:	• Administrador, supervisor.
Precondiciones:	• El actor debió ingresar previamente al sistema.
Flujo Normal:	• El actor presiona un botón para crear una nueva configuración. • El sistema habilita campos de texto para que ingrese los datos necesarios para la configuración. • El sistema comprueba la validez del código y de los datos de la configuración para poder almacenarlo.
Flujo Alternativo:	• El sistema comprueba que el código de la configuración no exista en la base de datos, si ya existe muestra un mensaje notificándole del error.
Poscondiciones:	• La configuración de migración ha sido almacenada en el sistema.

Caso de Uso:	Modificar Configurar migración.
Descripción:	Permite modificar configuraciones para la migración de archivos a la base.
Actores: • Administrador, supervisor.	
Precondiciones: • El actor debió ingresar previamente al sistema. • La configuración a modificar debe existir en el sistema	
Flujo Normal: • El actor busca la configuración y carga sus datos en cajas de texto. • El actor edita los datos de la configuración. • El sistema comprueba la validez de los datos de la configuración.	
Flujo Alternativo: • El sistema comprueba la validez de los datos, si existe error muestra un mensaje mostrando el inconveniente.	
Poscondiciones: • La configuración ha sido modificada en el sistema.	

APÉNDICE C: DIAGRAMA DE SECUENCIAS

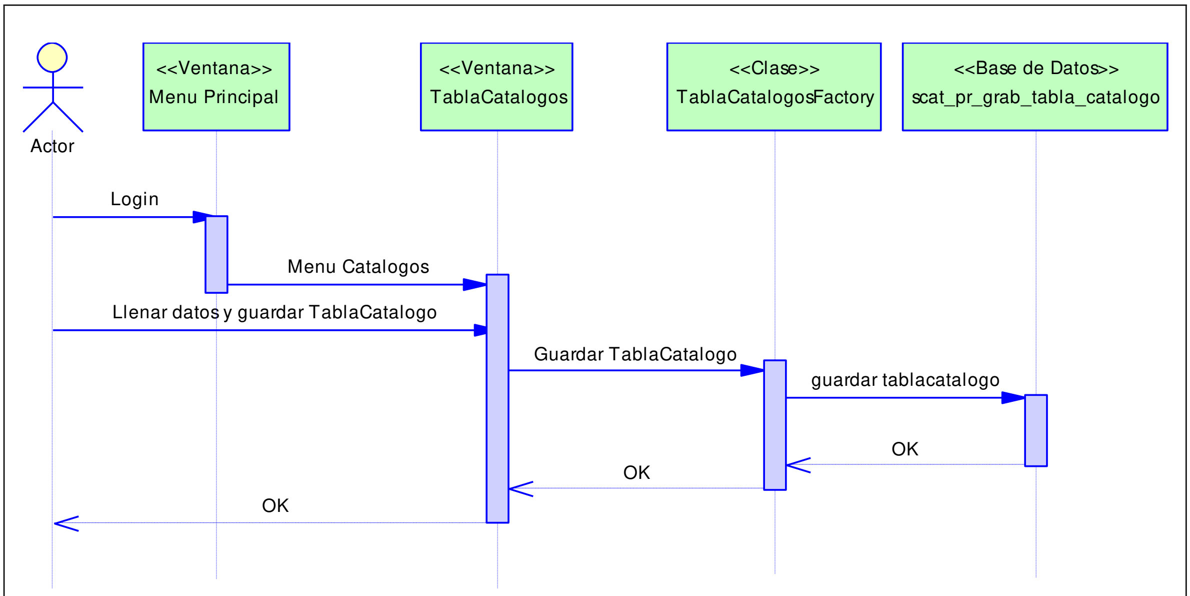


Figura C.1: Mantenimiento Tabla catálogos.

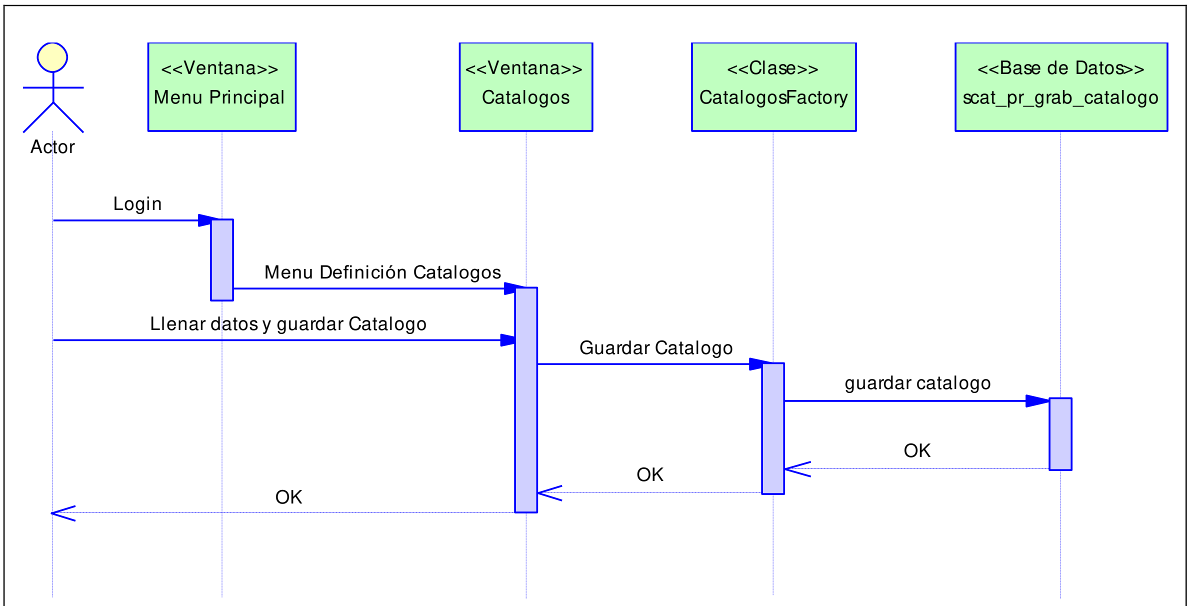


Figura C.2: Mantenimiento de catálogos.

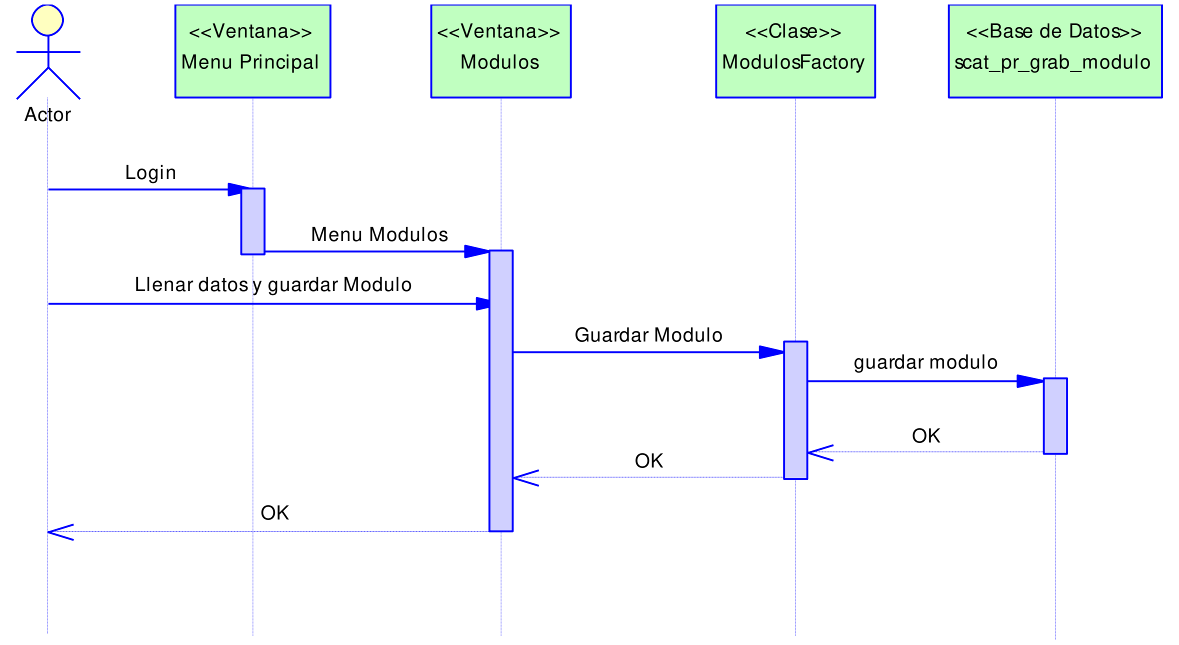


Figura C.3: Mantenimiento módulos.

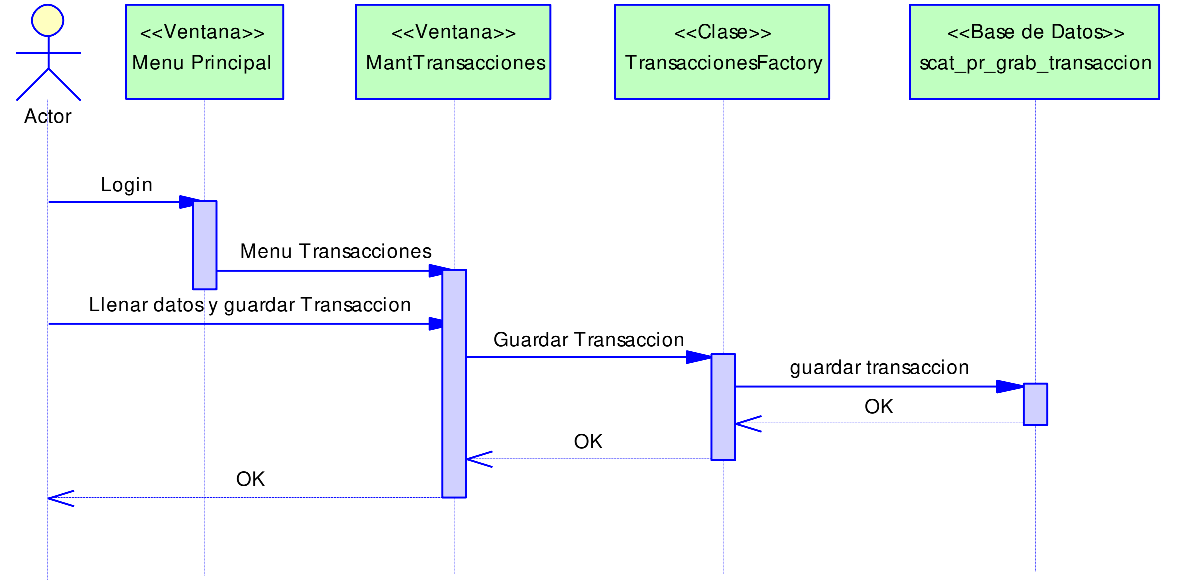


Figura C.4: Mantenimiento transacciones.

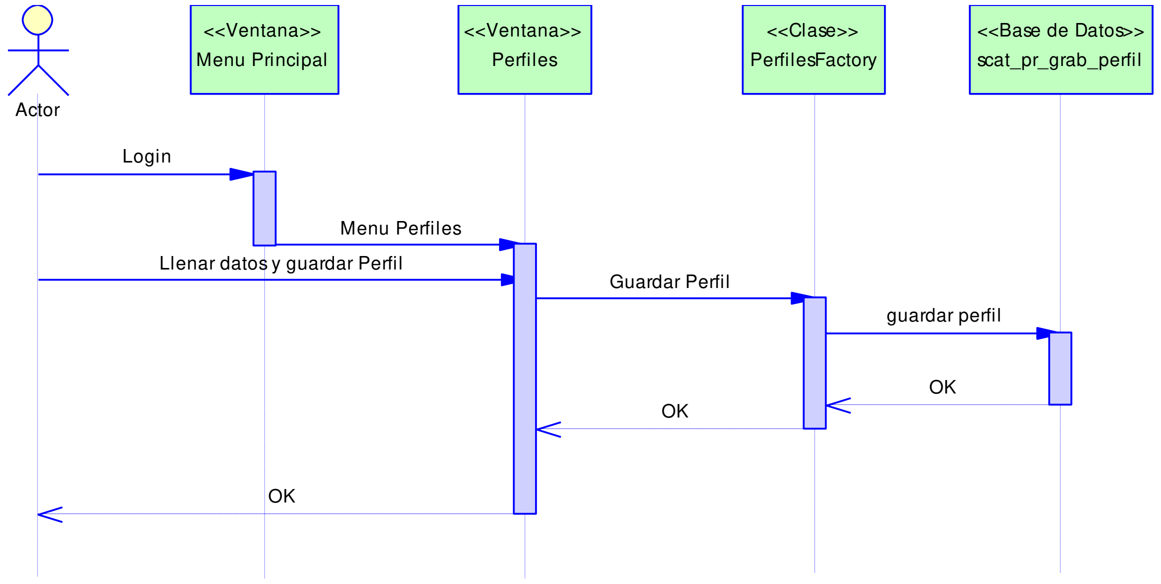


Figura C.5: Mantenimiento de perfil.

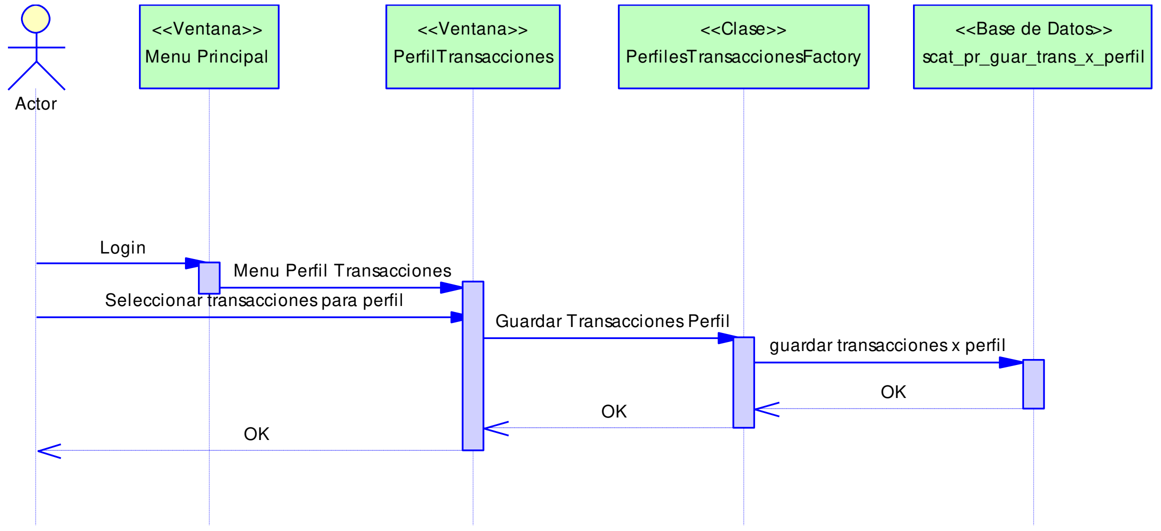


Figura C.6: Asignación de transacciones a perfil.

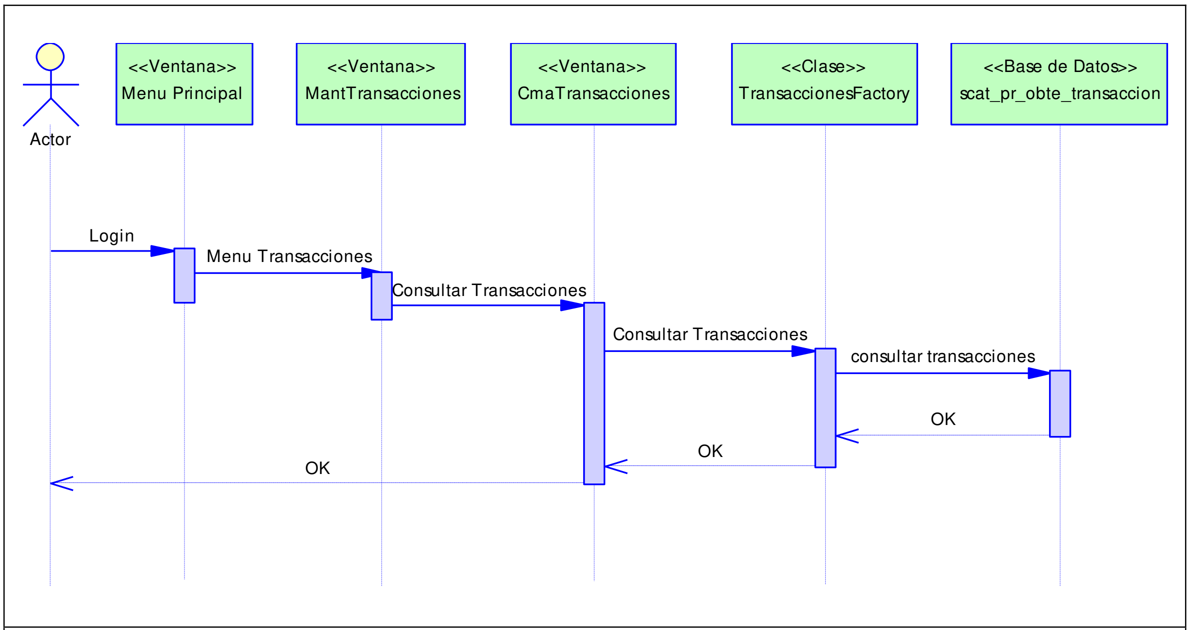


Figura C.7: Consulta transacciones.

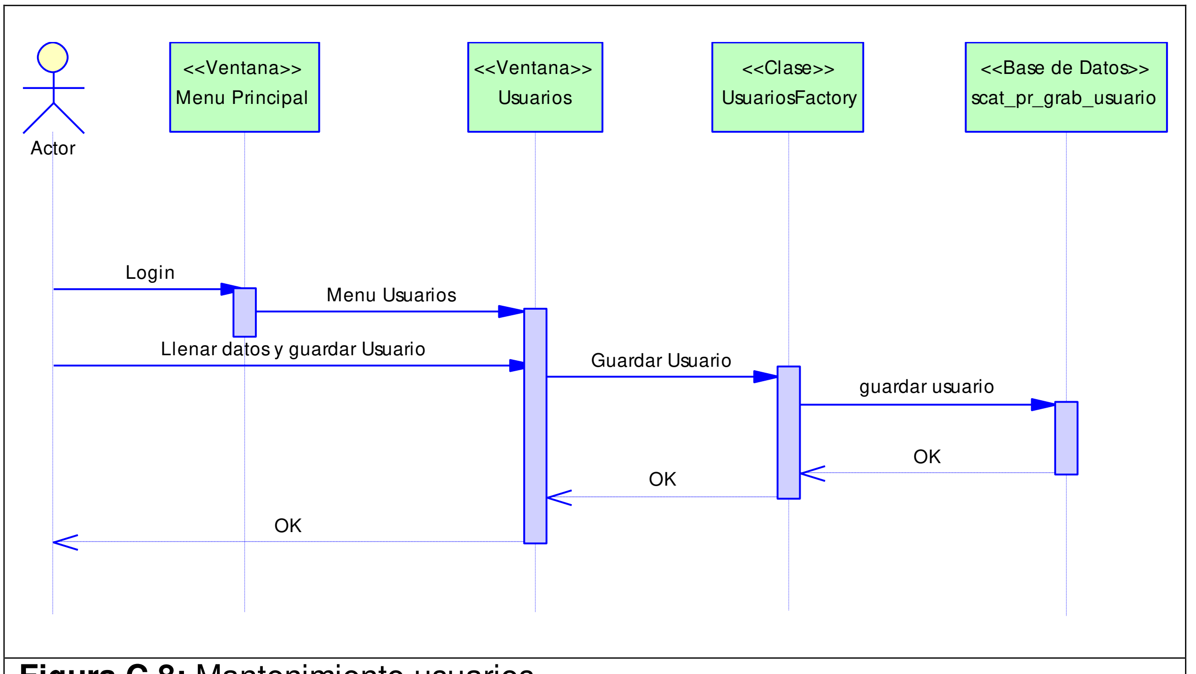


Figura C.8: Mantenimiento usuarios.

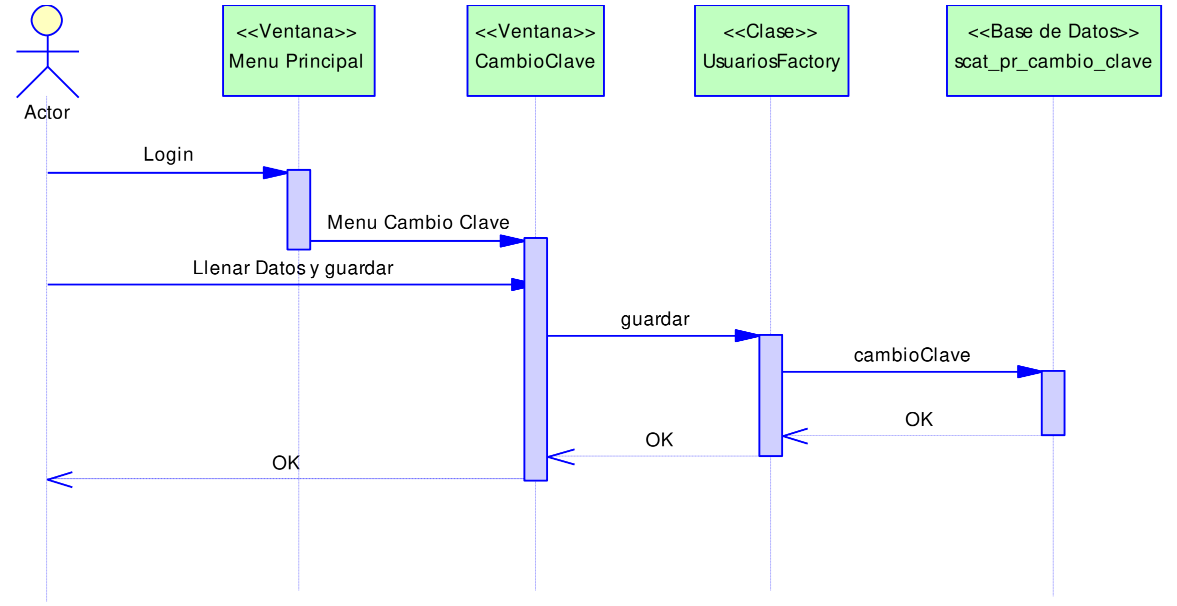


Figura C.9: Cambio de clave.

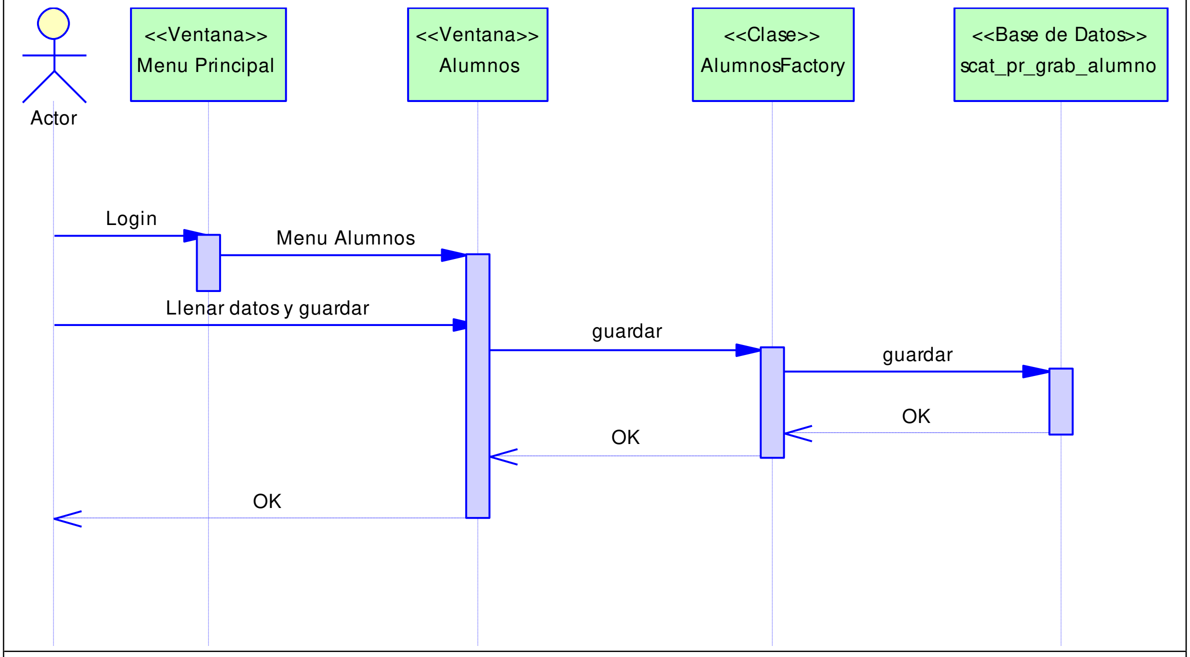


Figura C.10: Mantenimiento alumnos.

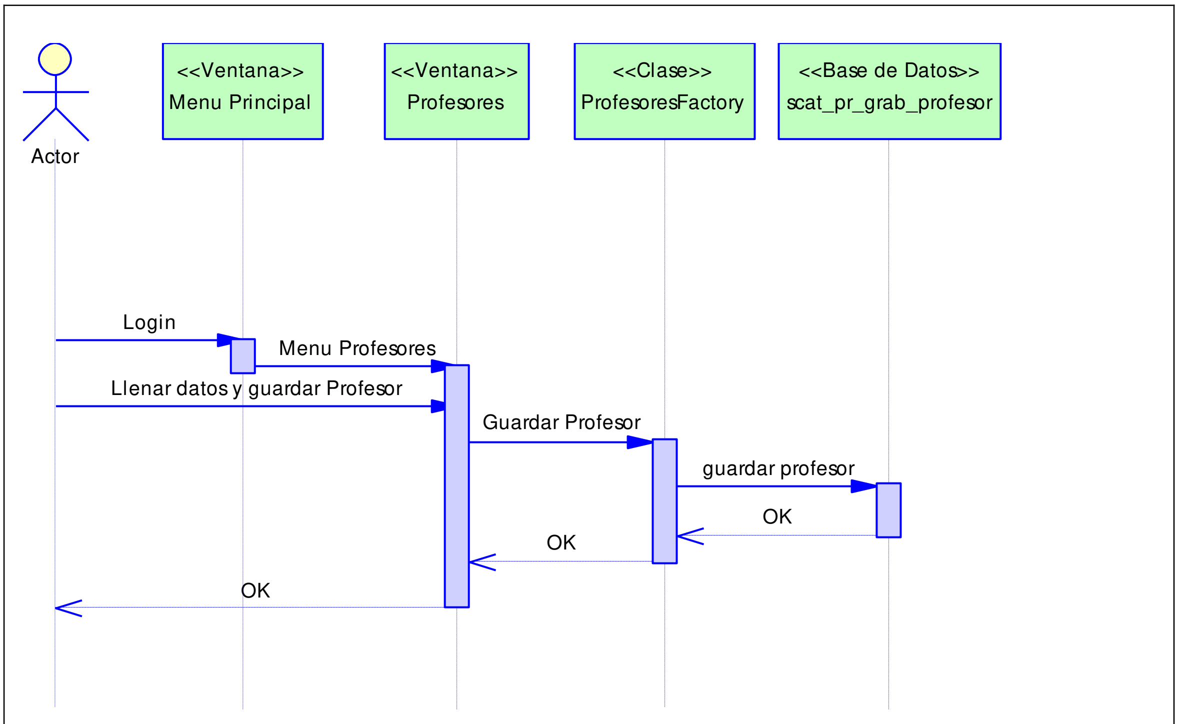


Figura C.11: Administración de profesores.

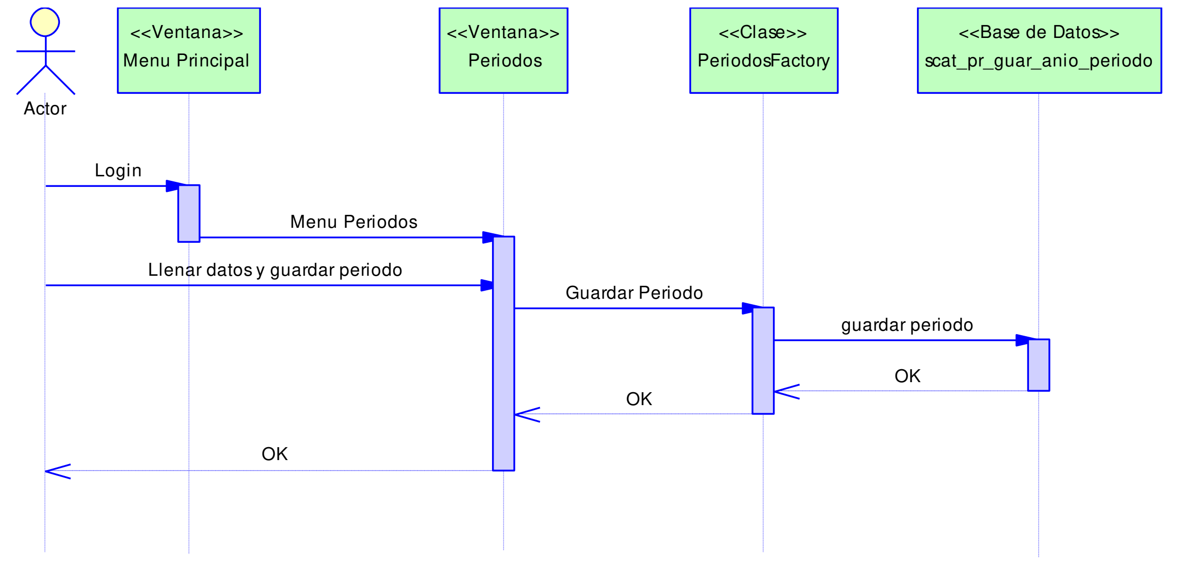


Figura C.12: Mantenimiento de periodos.

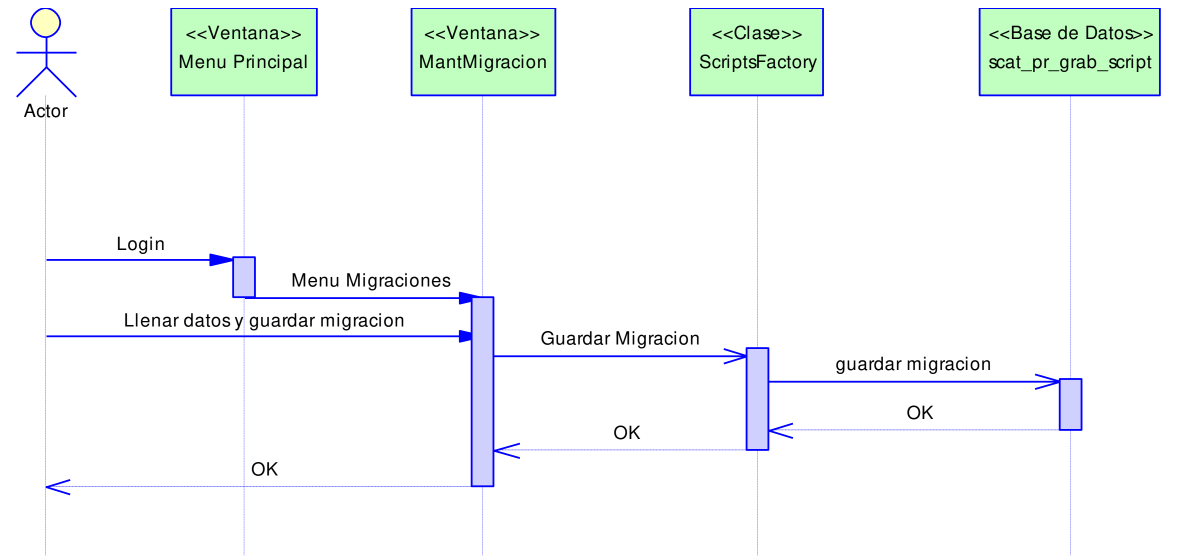


Figura C.13: Mantenimiento de Parámetros de migración.

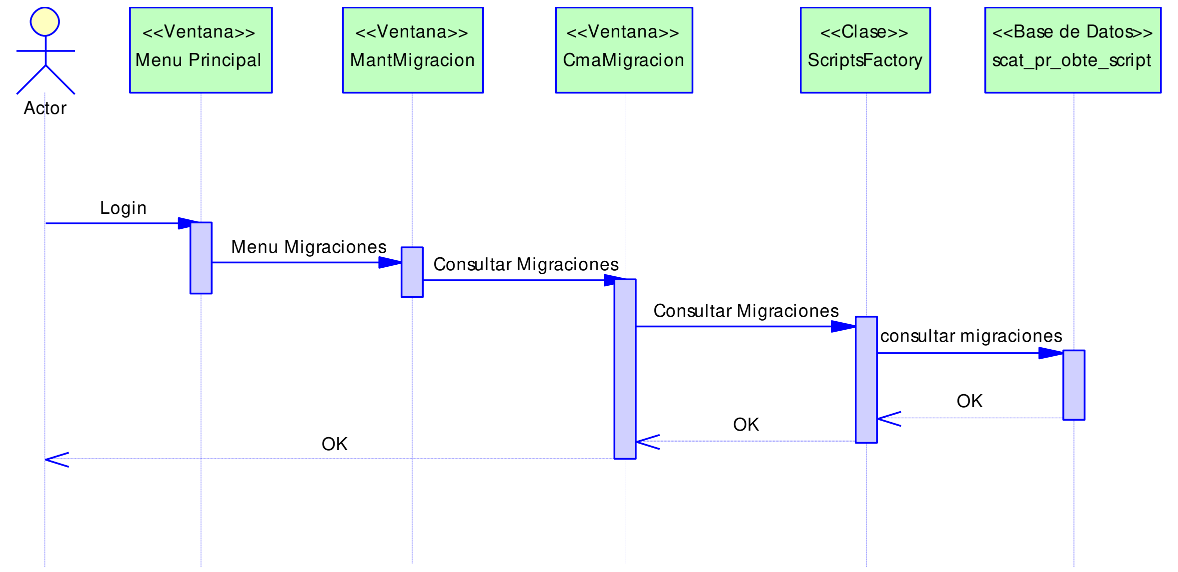


Figura C.14: Consulta de Parámetros de migración.

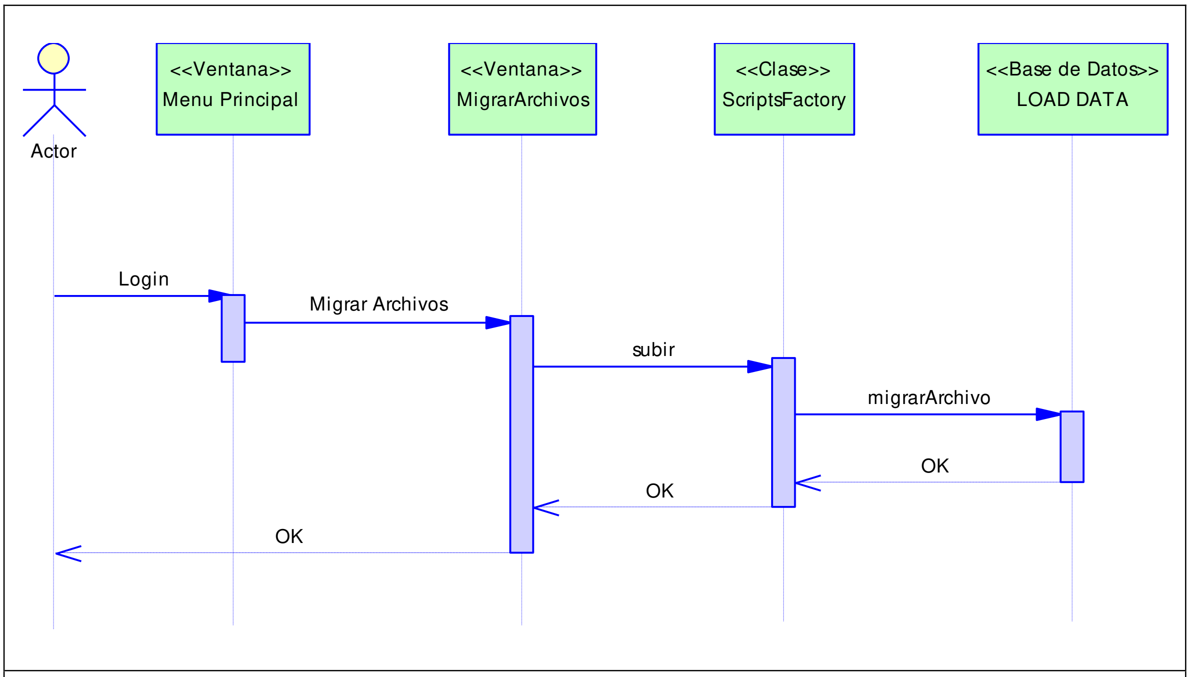


Figura C.15: Migrar archivos.

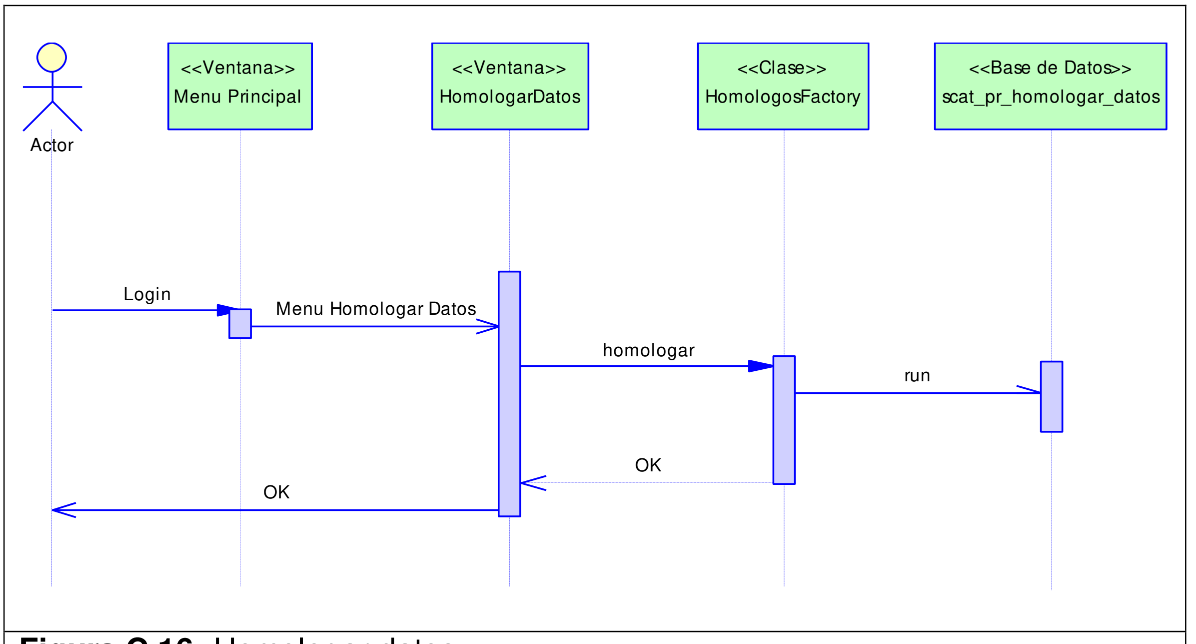


Figura C.16: Homologar datos.

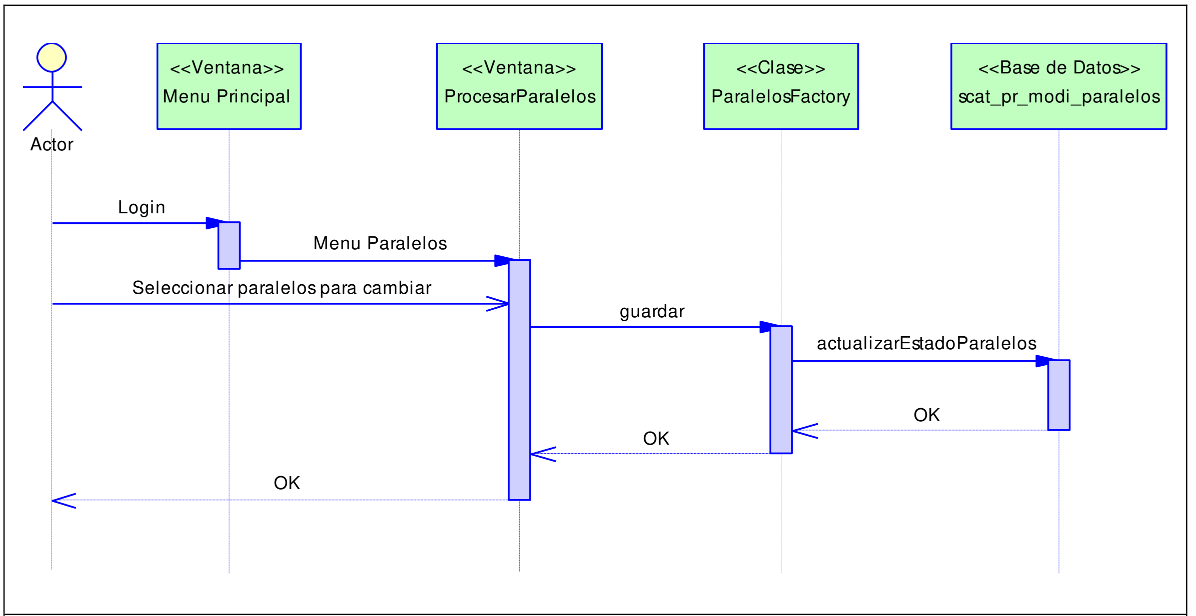


Figura C.17: Activar y desactivar paralelos.

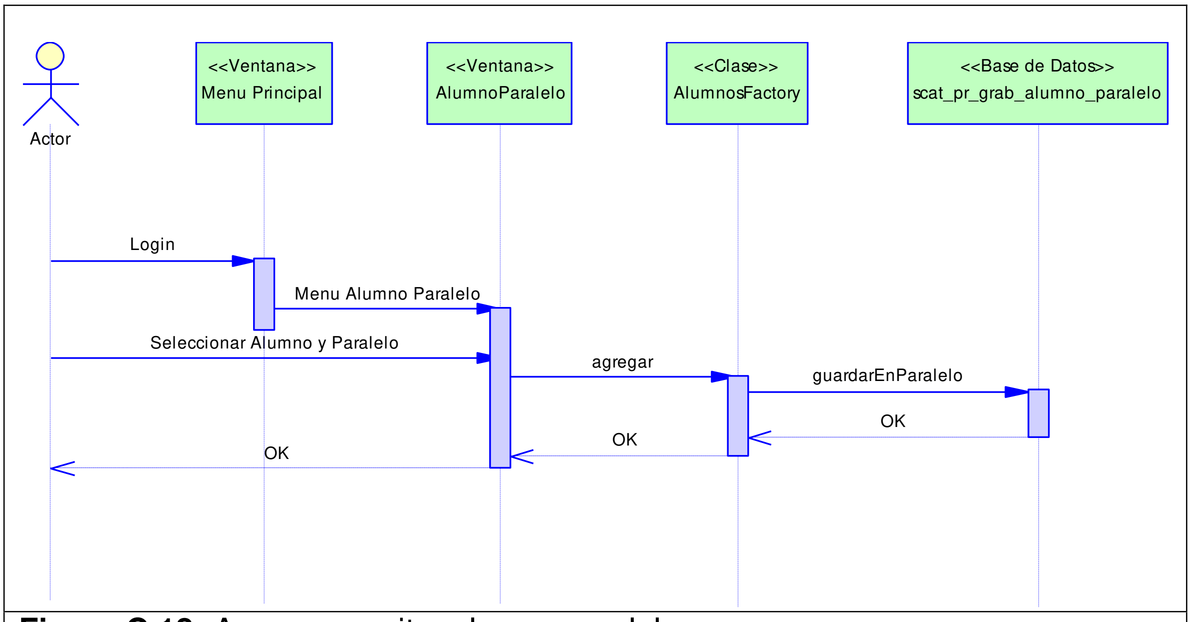


Figura C.18: Agregar – quitar alumno paralelo.

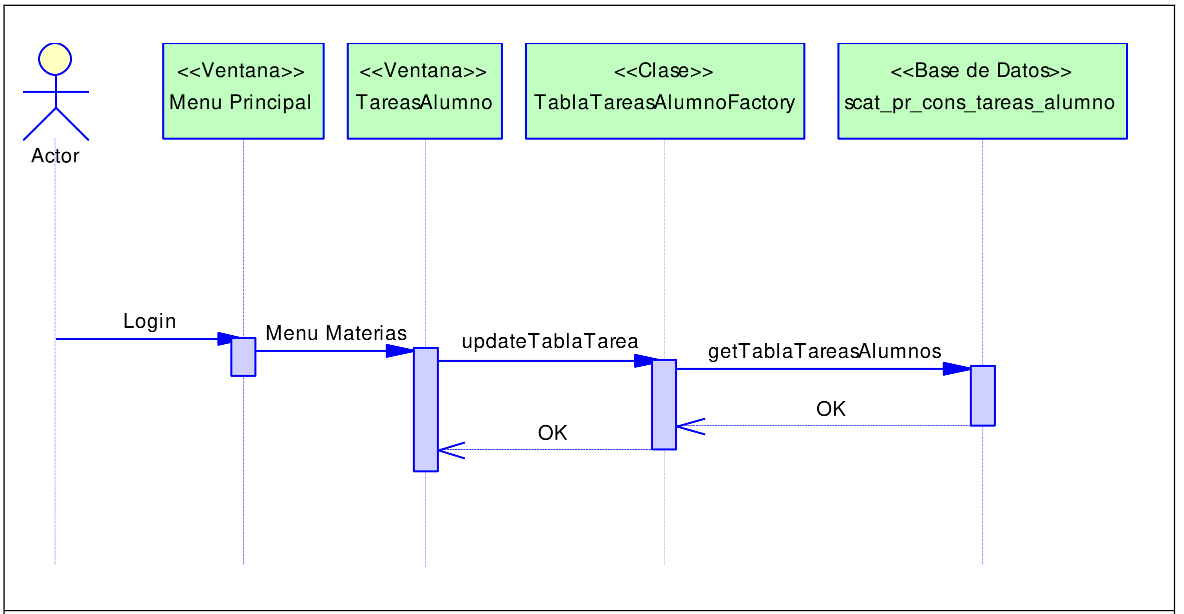


Figura C.19: Listar tareas alumno.

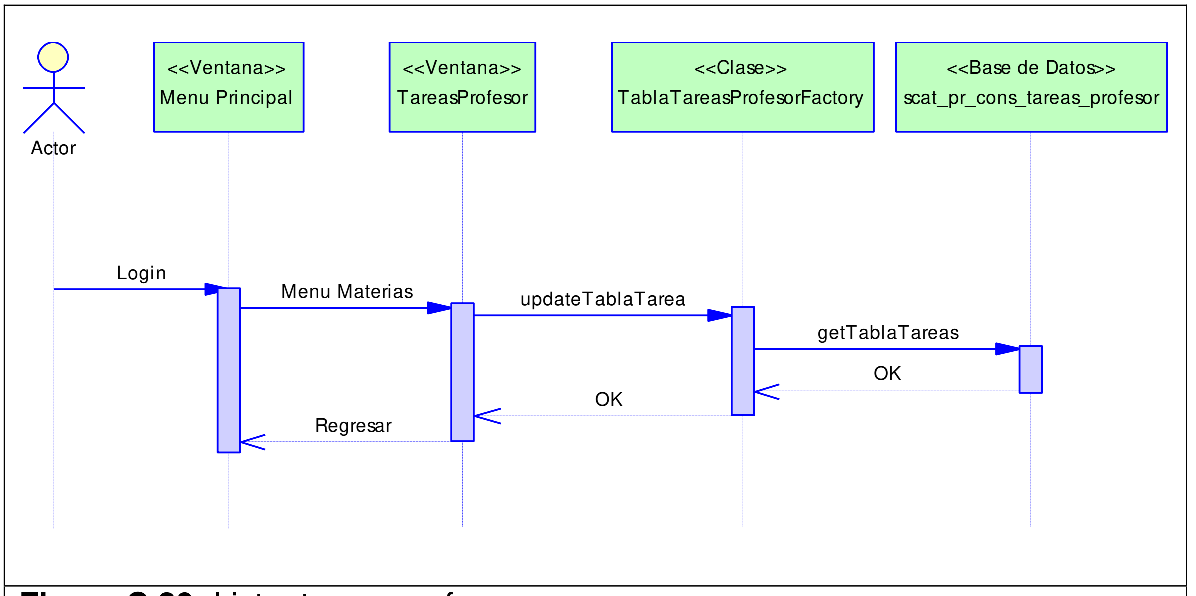
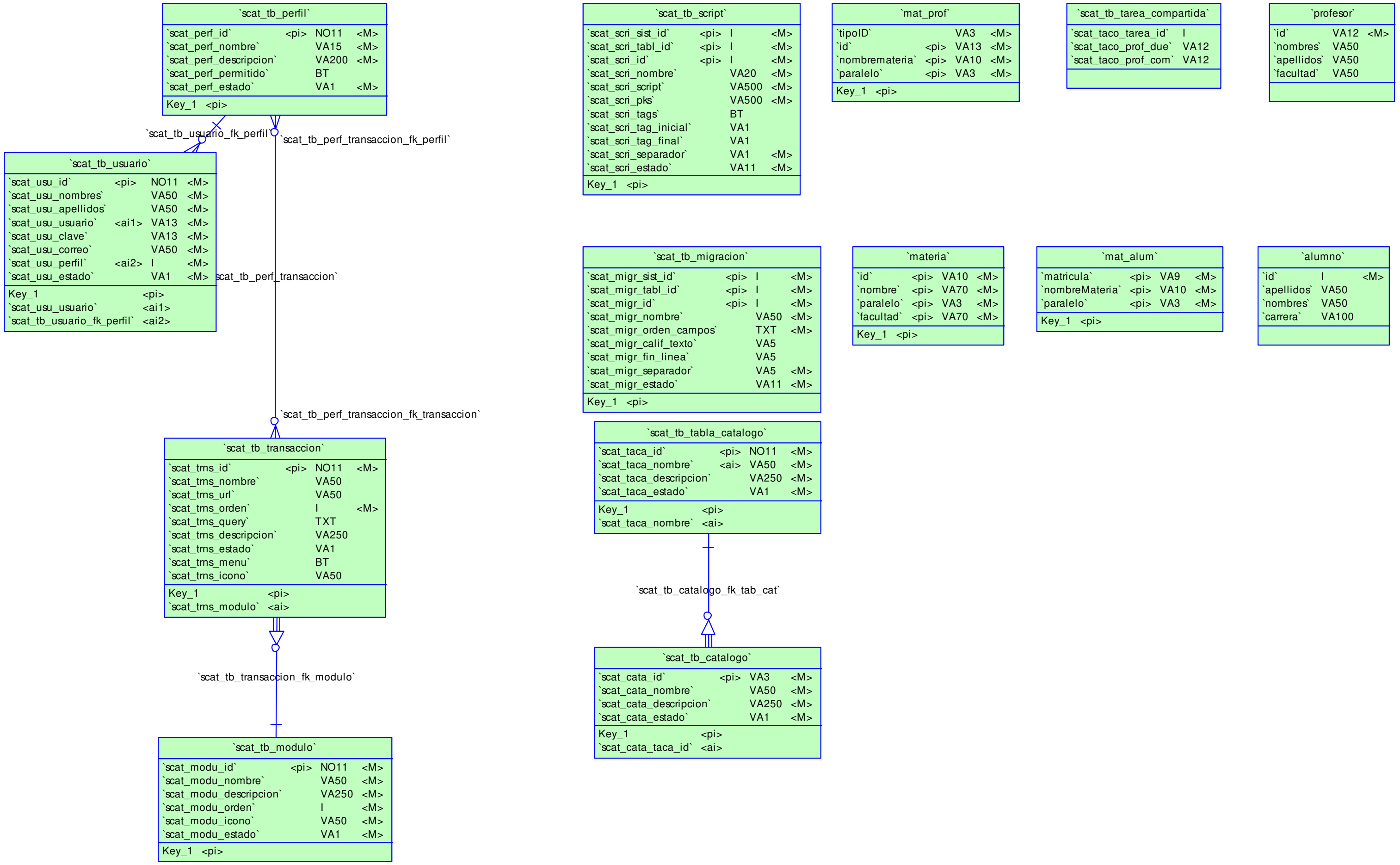
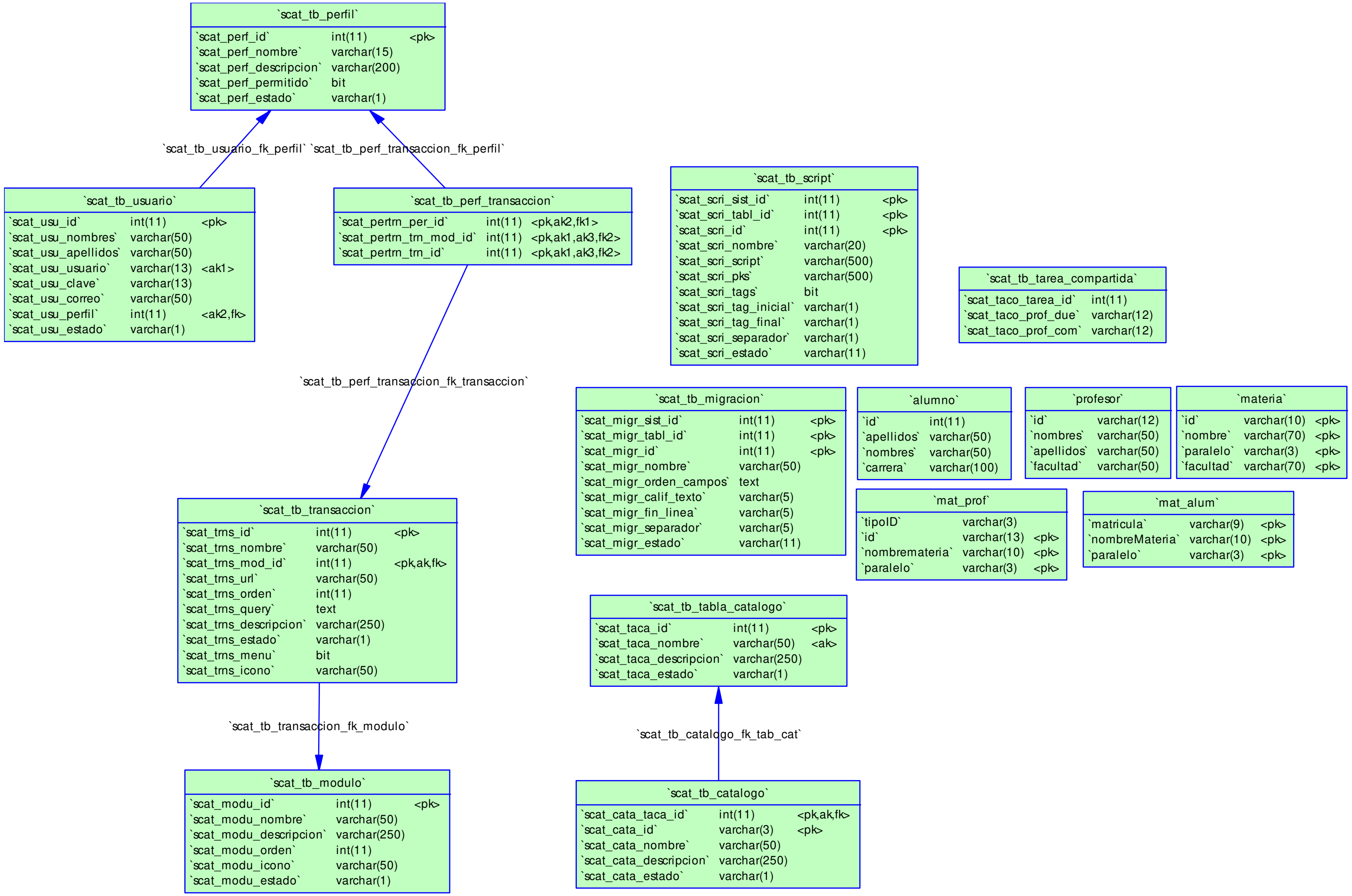


Figura C.20: Listar tareas profesor.

APÉNDICE D: MODELO FÍSICO DEL SISTEMA



APÉNDICE E: MODELO LÓGICO DEL SISTEMA



`scat_tb_perfil`		
`scat_perf_id`	int(11)	<pk>
`scat_perf_nombre`	varchar(15)	
`scat_perf_descripcion`	varchar(200)	
`scat_perf_permitido`	bit	
`scat_perf_estado`	varchar(1)	

`scat_tb_usuario`		
`scat_usu_id`	int(11)	<pk>
`scat_usu_nombres`	varchar(50)	
`scat_usu_apellidos`	varchar(50)	
`scat_usu_usuario`	varchar(13)	<ak1>
`scat_usu_clave`	varchar(13)	
`scat_usu_correo`	varchar(50)	
`scat_usu_perfil`	int(11)	<ak2,fk>
`scat_usu_estado`	varchar(1)	

`scat_tb_perf_transaccion`		
`scat_pertrn_per_id`	int(11)	<pk,ak2,fk1>
`scat_pertrn_trn_mod_id`	int(11)	<pk,ak1,ak3,fk2>
`scat_pertrn_trn_id`	int(11)	<pk,ak1,ak3,fk2>

`scat_tb_script`		
`scat_scri_sist_id`	int(11)	<pk>
`scat_scri_tabl_id`	int(11)	<pk>
`scat_scri_id`	int(11)	<pk>
`scat_scri_nombre`	varchar(20)	
`scat_scri_script`	varchar(500)	
`scat_scri_pks`	varchar(500)	
`scat_scri_tags`	bit	
`scat_scri_tag_inicial`	varchar(1)	
`scat_scri_tag_final`	varchar(1)	
`scat_scri_separador`	varchar(1)	
`scat_scri_estado`	varchar(11)	

`scat_tb_tarea_compartida`	
`scat_taco_tarea_id`	int(11)
`scat_taco_prof_due`	varchar(12)
`scat_taco_prof_com`	varchar(12)

`scat_tb_transaccion`		
`scat_trns_id`	int(11)	<pk>
`scat_trns_nombre`	varchar(50)	
`scat_trns_mod_id`	int(11)	<pk,ak,fk>
`scat_trns_url`	varchar(50)	
`scat_trns_orden`	int(11)	
`scat_trns_query`	text	
`scat_trns_descripcion`	varchar(250)	
`scat_trns_estado`	varchar(1)	
`scat_trns_menu`	bit	
`scat_trns_icono`	varchar(50)	

`scat_tb_migracion`		
`scat_migr_sist_id`	int(11)	<pk>
`scat_migr_tabl_id`	int(11)	<pk>
`scat_migr_id`	int(11)	<pk>
`scat_migr_nombre`	varchar(50)	
`scat_migr_orden_campos`	text	
`scat_migr_calif_texto`	varchar(5)	
`scat_migr_fin_linea`	varchar(5)	
`scat_migr_separador`	varchar(5)	
`scat_migr_estado`	varchar(11)	

`alumno`	
`id`	int(11)
`apellidos`	varchar(50)
`nombres`	varchar(50)
`carrera`	varchar(100)

`profesor`	
`id`	varchar(12)
`nombres`	varchar(50)
`apellidos`	varchar(50)
`facultad`	varchar(50)

`materia`		
`id`	varchar(10)	<pk>
`nombre`	varchar(70)	<pk>
`paralelo`	varchar(3)	<pk>
`facultad`	varchar(70)	<pk>

`mat_prof`		
`tipoID`	varchar(3)	
`id`	varchar(13)	<pk>
`nombremateria`	varchar(10)	<pk>
`paralelo`	varchar(3)	<pk>

`mat_alum`		
`matricula`	varchar(9)	<pk>
`nombreMateria`	varchar(10)	<pk>
`paralelo`	varchar(3)	<pk>

`scat_tb_tabla_catalogo`		
`scat_taca_id`	int(11)	<pk>
`scat_taca_nombre`	varchar(50)	<ak>
`scat_taca_descripcion`	varchar(250)	
`scat_taca_estado`	varchar(1)	

`scat_tb_modulo`		
`scat_modu_id`	int(11)	<pk>
`scat_modu_nombre`	varchar(50)	
`scat_modu_descripcion`	varchar(250)	
`scat_modu_orden`	int(11)	
`scat_modu_icono`	varchar(50)	
`scat_modu_estado`	varchar(1)	

`scat_tb_catalogo`		
`scat_cata_taca_id`	int(11)	<pk,ak,fk>
`scat_cata_id`	varchar(3)	<pk>
`scat_cata_nombre`	varchar(50)	
`scat_cata_descripcion`	varchar(250)	
`scat_cata_estado`	varchar(1)	

APÉNDICE F: ENCUESTA

ENCUESTA PARA EVALUAR EL SISTEMA DE CALIFICACIÓN AUTOMÁTICA DE TAREAS DE PROGRAMACIÓN

La siguiente encuesta está dirigida a evaluar la funcionalidad del Sistema SCAT.

Escoja la opción que mejor describa su opinión y tómese el tiempo necesario para responder estar preguntas

1: Evalúe el nivel de dificultad en encontrar la tarea a responder

○ Muy Fácil

○ Fácil

○ Difícil

○ Muy difícil

2: Evalúe el nivel de dificultad para responder la tarea una vez encontrada

○ Muy Fácil

○ Fácil

○ Difícil

○ Muy difícil

3: ¿Tuvo algún error al responder una tarea?

○ Si

○ No

4: Si la contestación fue afirmativa a la pregunta anterior, la ayuda que te ofrecieron los mensajes de retroalimentación del sistema fue:

○ Excelente

○ Buena

○ Mala

○ Pésima

5: Los resultados obtenidos al usar los controles (íconos / enlaces / botones / hipervínculos) que provee el sistema fueron los esperados.

○ Completamente de acuerdo
○ Parcial acuerdo
○ Parcial desacuerdo
○ Completamente desacuerdo

6: ¿Al querer realizar una acción determinada, usted siempre sabe donde esta ubicada la opción o enlace adecuado?

○ Completamente de acuerdo
○ Parcial acuerdo
○ Parcial desacuerdo
○ Completamente desacuerdo

7: ¿El sistema le da la oportunidad a usted de revisar los datos de la respuesta de la tarea?

○ Completamente de acuerdo
○ Parcial acuerdo
○ Parcial desacuerdo
○ Completamente desacuerdo

8: ¿En general qué tan buena ha sido la interfaz de navegación que ofrece el sistema?

○ Excelente
○ Buena
○ Mala
○ Pésima

9: ¿La combinación de colores que se presenta es agradable?

○ Completamente de acuerdo
○ Parcial acuerdo
○ Parcial desacuerdo
○ Completamente desacuerdo

10: ¿Los íconos son lo suficientemente representativos para indicar su funcionalidad?

○ Completamente de acuerdo
○ Parcial acuerdo
○ Parcial desacuerdo
○ Completamente desacuerdo

11: ¿La presentación o distribución de la información es clara?

○ Completamente de acuerdo
○ Parcial acuerdo
○ Parcial desacuerdo
○ Completamente desacuerdo

12: ¿Los colores de fondo ofrecidos por el sistema son los adecuados?

○ Completamente de acuerdo
○ Parcial acuerdo
○ Parcial desacuerdo
○ Completamente desacuerdo

¡GRACIAS POR SU TIEMPO!!!

APÉNDICE G: EJEMPLO DE PRUEBA REALIZADA

A continuación se describe la prueba que fue realizada a los 10 estudiantes egresados de diferentes universidades:

“La tarea consiste en que dada una muestra en un arreglo de enteros, se pueda obtener la media aritmética, la desviación estándar, la varianza y la frecuencia de un dato específico.

La clase debe llamarse 'Muestra' y debe tener una variable privada, la cual es un arreglo y de tipo entero, esta variable es la que se debe usar para realizar los siguientes métodos públicos:

- *setData, el cual recibe un arreglo de enteros para setear la variable antes mencionada.*
- *getMediaAritmetica, el cual retorna un double y no recibe parámetros.*
- *getVarianza, el cual retorna un double y no recibe parámetros.*
- *getFrecuencia, el cual retorna un entero y recibe un parámetro de tipo entero.*
- *Debe tener un constructor por defecto sin parámetros.”*

Otros atributos de la tarea son:

- Grupos conformados entre 2 y 4 estudiantes.
- Lenguaje utilizado para la resolución: JAVA versión JDK 1.6.0_04.

APÉNDICE H: MANUAL DE USUARIO

En este apéndice se incluirá el manual de usuario del Sistema SCAT.

Para poder ingresar al Sistema de Calificación Automática de Tareas de Programación, debe escribir en su navegador Web la dirección correspondiente. Una vez que haya hecho esto, se abrirá la siguiente ventana (Ver figura H.1):


Sistema de Calificación Automática de Tareas de Programación (SCAT)

Figura H.1: Pantalla de ingreso del sistema SCAT.

En esta pantalla debe ingresar su usuario y su contraseña de acceso. Si usted es un alumno o profesor ingrese su usuario y contraseña de su cuenta de correo de la ESPOL. Si es un supervisor o administrador del sistema por favor ingrese poniendo su usuario y su contraseña que le debieron ser otorgados por el administrador del sistema.

Sistema de Calificación Automática de Tareas de Programación (SCAT)

Transacción fallida

Su login es incorrecto. Por favor intente nuevamente

Usuario: * wuzca

Contraseña: *

Figura H.2: Mensaje de error de login incorrecto.

Si usted es un alumno o profesor y su login es correcto pero no se encuentra registrado en el sistema SCAT, se le mostrará el mensaje de no registrado. (Ver figura H.3)

Sistema de Calificación Automática de Tareas de Programación (SCAT)

Transacción fallida

No se encuentra registrado en el sistema SCAT

Usuario: * wuzca

Contraseña: *

Figura H.3: Mensaje de error no registrado en sistema

Si usted ingresa un usuario y contraseña correcta, visualizará la siguiente pantalla. (Ver figura H.4).

Bienvenido(a) WALTER IVAN UZCA SORNOZA

Usuario: walter

Sesion: Administrador

Bienvenidos al Sistema de Calificación Automática de Tareas de Programación (SCAT)

Este sistema Web le permitirá calificar las tareas de programación de una forma fácil y automatizada.
En el presente usted podrá accesar a diferentes opciones que el sistema brinda con la finalidad de mantenerlo informado de todos las actividades que deben cumplir los estudiantes.

Figura H.4: Pantalla de Bienvenida

En la parte superior de la pantalla inicial (Ver figura H.4) se puede apreciar que se da la bienvenida, seguido del nombre de usuario y más a la derecha el tipo de sesión con el perfil correspondiente. En la esquina superior derecha están los botones de Inicio y Cerrar Sesión. El botón Inicio sirve para volver a la Pantalla de Bienvenida y el botón Cerrar Sesión sirve para cerrar la sesión del usuario y volver a la pantalla de ingreso al Sistema SCAT (Ver figura H.1). En el lado izquierdo se muestra el menú principal que varía de acuerdo al perfil del usuario, puesto que es configurable por el administrador del sistema.

Existen cuatro tipos de perfiles en el sistema SCAT, que son: Administrador, Supervisor, Profesor y Alumno. Cada perfil tiene asignado un menú que le permite realizar sus transacciones correspondientes.

El menú del Administrador tiene las siguientes opciones:
1. Administración
2. Seguridad

El menú de Administración tiene las siguientes opciones:

1. Catálogos
2. Definición de Catálogos
3. Perfiles
4. Módulos
5. Transacciones
6. Perfil Transacciones
7. Usuarios
8. Alumnos
9. Profesores
10. Parámetros de Migración
11. Migrar Archivos
12. Homologar Datos
13. Administración Alumno Paralelo
14. Periodos
15. Paralelos

El menú de Seguridad tiene la siguiente opción:

1. Cambio de Clave

El menú del Supervisor tiene las siguientes opciones:

1. Administración
2. Seguridad

El menú de Administración tiene las siguientes opciones:

1. Migrar Archivos
2. Homologar Datos
3. Administración Alumno Paralelo

El menú Seguridad tiene la siguiente opción:

1. Cambio de Clave

El menú del Profesor tiene las siguientes opciones:

1. Administración
2. Materias

El menú Administración tiene la siguiente opción:

1. Compartir Tareas

El menú Materias muestra como opciones los diferentes paralelos que el profesor tiene a su cargo en el período actual. Si el profesor dicta una materia en más de un paralelo aparecerán dos opciones, una correspondiente a cada paralelo.

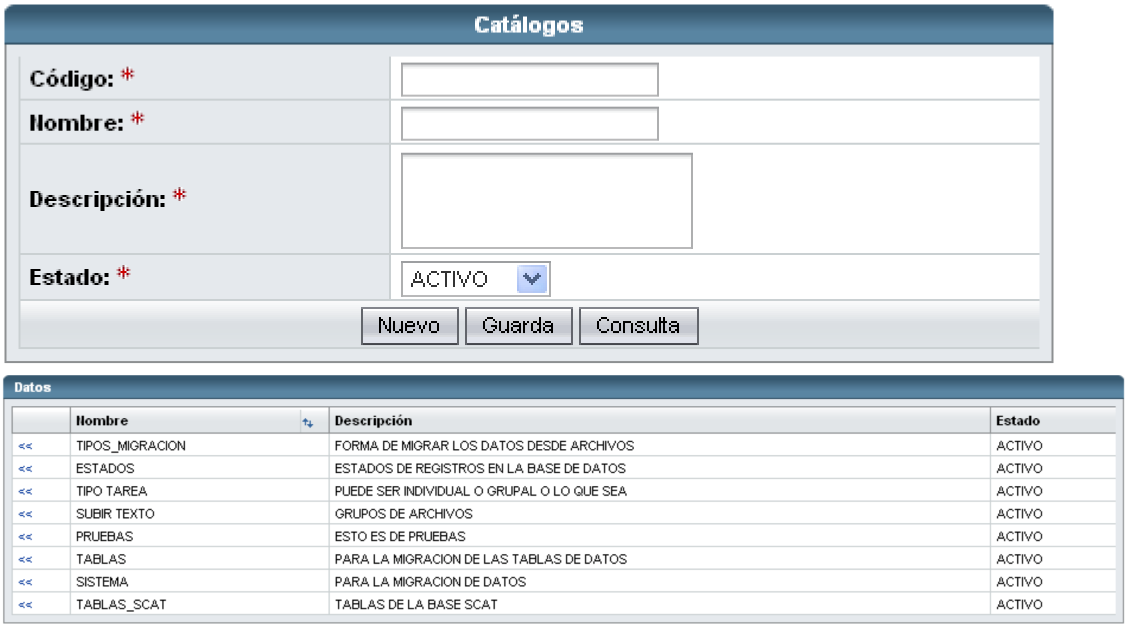
El menú del Alumno tiene la siguiente opción:

1. Materias

El menú Materias muestra como opciones las diferentes materias que el alumno está tomando en el período actual.

A continuación se revisarán todas las transacciones que tiene el sistema.

1. Registro y Mantenimiento de Catálogos.



Código: *

Nombre: *

Descripción: *

Estado: * ACTIVO

	Nombre	Descripción	Estado
<<	TIPOS_MIGRACION	FORMA DE MIGRAR LOS DATOS DESDE ARCHIVOS	ACTIVO
<<	ESTADOS	ESTADOS DE REGISTROS EN LA BASE DE DATOS	ACTIVO
<<	TIPO TAREA	PUEDE SER INDIVIDUAL O GRUPAL O LO QUE SEA	ACTIVO
<<	SUBIR TEXTO	GRUPOS DE ARCHIVOS	ACTIVO
<<	PRUEBAS	ESTO ES DE PRUEBAS	ACTIVO
<<	TABLAS	PARA LA MIGRACION DE LAS TABLAS DE DATOS	ACTIVO
<<	SISTEMA	PARA LA MIGRACION DE DATOS	ACTIVO
<<	TABLAS_SCAT	TABLAS DE LA BASE SCAT	ACTIVO

Figura H.5: Crear y Modificar Catálogos.

En esta opción usted podrá registrar nuevos catálogos y modificar catálogos existentes (Ver figura H.5). La eliminación es lógica y no física, se maneja con el atributo denominado estado que determina si se encuentra activo o inactivo.

Para registrar un nuevo catálogo debo presionar el botón Nuevo, llenar los campos necesarios y luego presionar el botón Guardar. Los campos necesarios son:

- Código.
- Nombre.
- Descripción.
- Estado.

Si va a registrar un nuevo catálogo y no conoce el código que sigue puede escribir el número cero para que se registre con la secuencia siguiente.

Se puede realizar una consulta masiva dando clic en el botón Consulta, lo que mostrará todos los catálogos en una tabla en la parte inferior (Ver figura F.5). La consulta puede ser filtrada por los campos nombre y estado.

Para realizar una consulta específica puede dar clic sobre el hipervínculo “<<” en la primera columna de la tabla presentada y se cargarán los datos del catálogo en los campos correspondientes del formulario. Otra forma de hacer una consulta específica es escribir el código a consultar y perder el foco del campo código para que realice una consulta asincrónica y se carguen los demás campos en el formulario.

Para modificar un catálogo se debe primero realizar una consulta para que se carguen los datos del catálogo, modificar los campos que se desean cambiar y luego dar clic en el botón Guardar.

2. Registro y Mantenimiento de Definición de Catálogos.

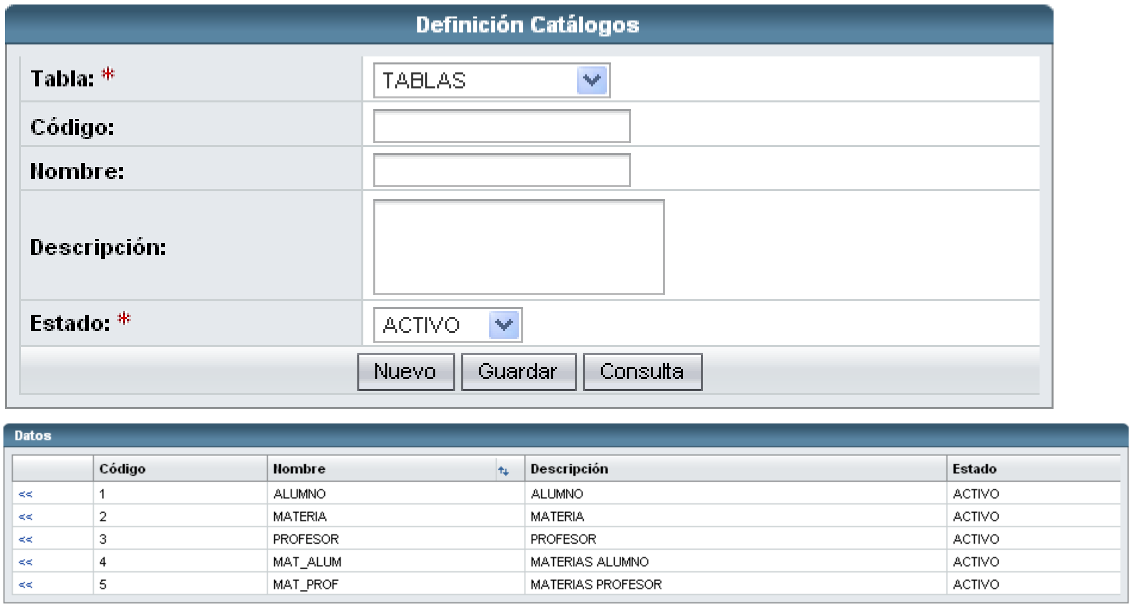


Tabla: *	TABLAS
Código:	
Nombre:	
Descripción:	
Estado: *	ACTIVO

	Código	Nombre	Descripción	Estado
<<	1	ALUMNO	ALUMNO	ACTIVO
<<	2	MATERIA	MATERIA	ACTIVO
<<	3	PROFESOR	PROFESOR	ACTIVO
<<	4	MAT_ALUM	MATERIAS ALUMNO	ACTIVO
<<	5	MAT_PROF	MATERIAS PROFESOR	ACTIVO

Figura H.6: Crear y Modificar Definición de Catálogos.

En esta opción usted podrá registrar nuevas definiciones de catálogos y modificar definiciones de catálogos existentes (Ver figura H.6). La eliminación es lógica y no física, se maneja con el atributo denominado estado que determina si se encuentra activo o inactivo.

Para registrar una nueva definición de catálogo debo presionar el botón Nuevo, llenar los campos necesarios y luego presionar el botón Guardar. Los campos necesarios son:

- Tabla.
- Código.
- Nombre.
- Descripción.
- Estado.

Si va a registrar una nueva definición de catálogo y no conoce si existe el código, solamente escríbalo y si se cargan los campos del formulario quiere decir que existe, o en su defecto puede realizar una consulta masiva y ver todos los códigos que se usan en la definición de catálogo.

Se puede realizar una consulta masiva dando clic en el botón Consulta, lo que mostrará todas las definiciones de catálogos en una tabla en la parte inferior (Ver figura H.6). La consulta puede ser filtrada por los campos tabla, nombre y estado.

Para realizar una consulta específica puede dar clic sobre el hipervínculo "<<" en la primera columna de la tabla presentada y se cargarán los datos de la definición de catálogo en los campos correspondientes del formulario. Otra forma de hacer una consulta específica es escribir el código a consultar y perder el foco del campo código para que realice una consulta asincrónica y se carguen los demás campos en el formulario.

Para modificar una definición de catálogo se debe primero realizar una consulta para que se carguen los datos de la definición de catálogo, modificar los campos que se desean cambiar y luego dar clic en el botón Guardar.

3. Registro y Mantenimiento de Perfiles.

Código: *

Nombre: *

Descripción: *

Permitido: ☐ Perfil permitido

Estado: * ACTIVO

	Nombre	Descripción	Permitido	Estado
<<	Administrador	El que administra el sistema	☑	ACTIVO
<<	Profesor	El man pone las tareas	☐	ACTIVO
<<	Alumno	El man resuelve las tareas	☐	ACTIVO
<<	Supervisor	El que supervisa el sistema	☑	ACTIVO

Figura H.7: Crear y Modificar Perfiles.

En esta opción usted podrá registrar nuevos perfiles y modificar perfiles existentes (Ver figura H.7). La eliminación es lógica y no física, se maneja con el atributo denominado estado que determina si se encuentra activo o inactivo.

Para registrar un nuevo perfil debo presionar el botón Nuevo, llenar los campos necesarios y luego presionar el botón Guardar. Los campos necesarios son:

- Código.
- Nombre.
- Descripción.
- Permitido. Si es permitido aparece en el mantenimiento de usuarios. Solo deben ser permitidos los perfiles Administrador y Supervisor
- Estado.

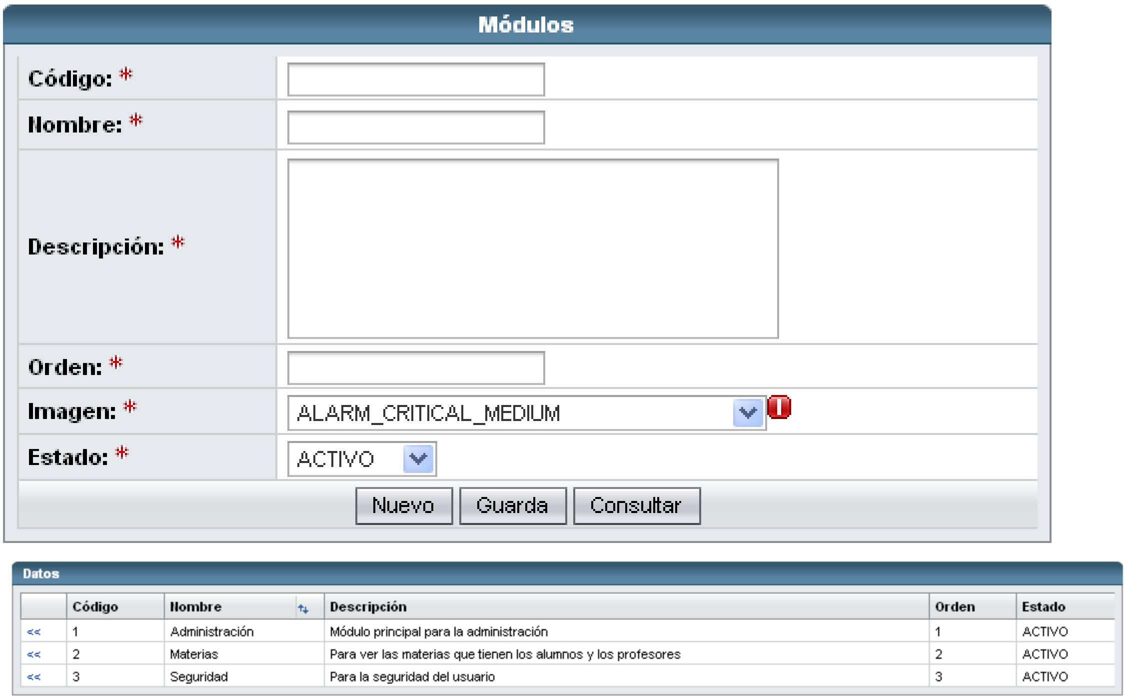
Si va a registrar un nuevo perfil y no conoce el código que sigue puede escribir el número cero para que se registre con la secuencia siguiente.

Se puede realizar una consulta masiva dando clic en el botón Consulta, lo que mostrará todos los perfiles en una tabla en la parte inferior (Ver figura H.7). La consulta puede ser filtrada por los campos nombre y estado.

Para realizar una consulta específica puede dar clic sobre el hipervínculo “<<” en la primera columna de la tabla presentada y se cargarán los datos del perfil en los campos correspondientes del formulario. Otra forma de hacer una consulta específica es escribir el código a consultar y perder el foco del campo código para que realice una consulta asincrónica y se carguen los demás campos en el formulario.

Para modificar un perfil se debe primero realizar una consulta para que se carguen los datos del perfil, modificar los campos que se desean cambiar y luego dar clic en el botón Guardar.

4. Registro y Mantenimiento de Módulos.



Código: *

Nombre: *

Descripción: *

Orden: *

Imagen: * ALARM_CRITICAL_MEDIUM

Estado: * ACTIVO

	Código	Nombre	Descripción	Orden	Estado
<<	1	Administración	Módulo principal para la administración	1	ACTIVO
<<	2	Materias	Para ver las materias que tienen los alumnos y los profesores	2	ACTIVO
<<	3	Seguridad	Para la seguridad del usuario	3	ACTIVO

Figura H.8: Crear y Modificar Módulos.

En esta opción usted podrá registrar nuevos módulos y modificar módulos existentes (Ver figura H.8). La eliminación es lógica y no física, se maneja con el atributo denominado estado que determina si se encuentra activo o inactivo.

Cada módulo representa a un menú que se muestra en la parte izquierda de la pantalla de bienvenida. (Ver figura H.4).

Para registrar un nuevo módulo debo presionar el botón Nuevo, llenar los campos necesarios y luego presionar el botón Guardar. Los campos necesarios son:

- Código.
- Nombre.
- Descripción.
- Orden.
- Imagen.
- Estado.

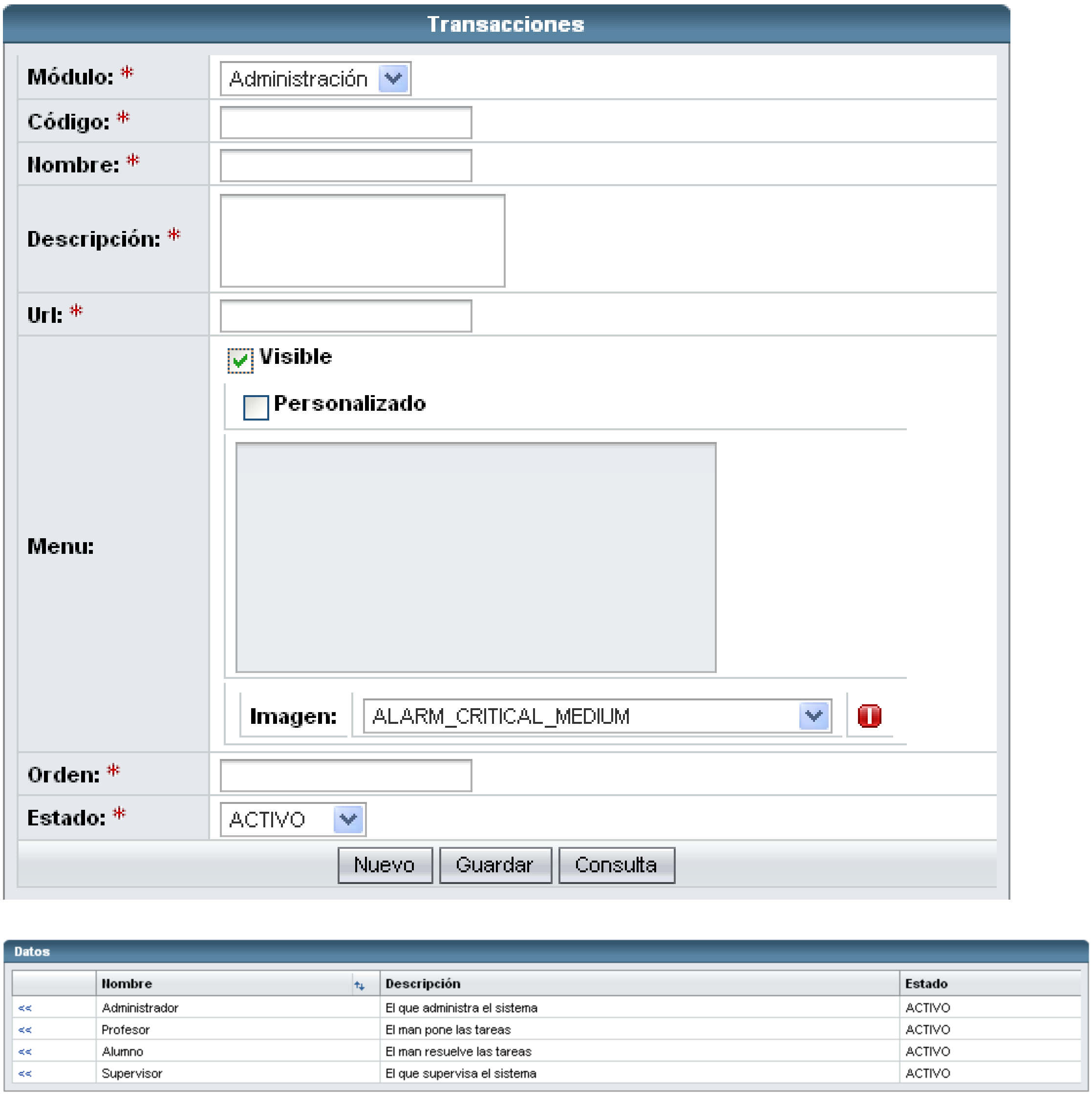
Si va a registrar un nuevo módulo y no conoce el código que sigue puede escribir el número cero para que se registre con la secuencia siguiente.

Se puede realizar una consulta masiva dando clic en el botón Consulta, lo que mostrará todos los módulos en una tabla en la parte inferior (Ver figura H.5). La consulta puede ser filtrada por los campos nombre y estado.

Para realizar una consulta específica puede dar clic sobre el hipervínculo “<<” en la primera columna de la tabla presentada y se cargarán los datos del módulo en los campos correspondientes del formulario. Otra forma de hacer una consulta específica es escribir el código a consultar y perder el foco del campo código para que realice una consulta asincrónica y se carguen los demás campos en el formulario.

Para modificar un módulo se debe primero realizar una consulta para que se carguen los datos del módulo, modificar los campos que se desean cambiar y luego dar clic en el botón Guardar.

5. Registrar y Modificar Transacciones.



Transacciones

Módulo: *	Administración
Código: *	
Nombre: *	
Descripción: *	
Url: *	
Menu:	☑ Visible ☐ Personalizado Imagen: ALARM_CRITICAL_MEDIUM
Orden: *	
Estado: *	ACTIVO

Datos

	Nombre	Descripción	Estado
<<	Administrador	El que administra el sistema	ACTIVO
<<	Profesor	El man pone las tareas	ACTIVO
<<	Alumno	El man resuelve las tareas	ACTIVO
<<	Supervisor	El que supervisa el sistema	ACTIVO

Figura H.9: Registrar y Modificar Transacciones.

En esta opción usted podrá registrar nuevas transacciones y modificar transacciones existentes (Ver figura H.9). La eliminación es lógica y no física, se maneja con el atributo denominado estado que determina si se encuentra activo o inactivo.

Cada transacción representa una opción dentro del menú de la pantalla de bienvenida (Ver figura H.4)

Para registrar una nueva transacción debo presionar el botón Nuevo, llenar los campos necesarios y luego presionar el botón Guardar. Los campos necesarios son:

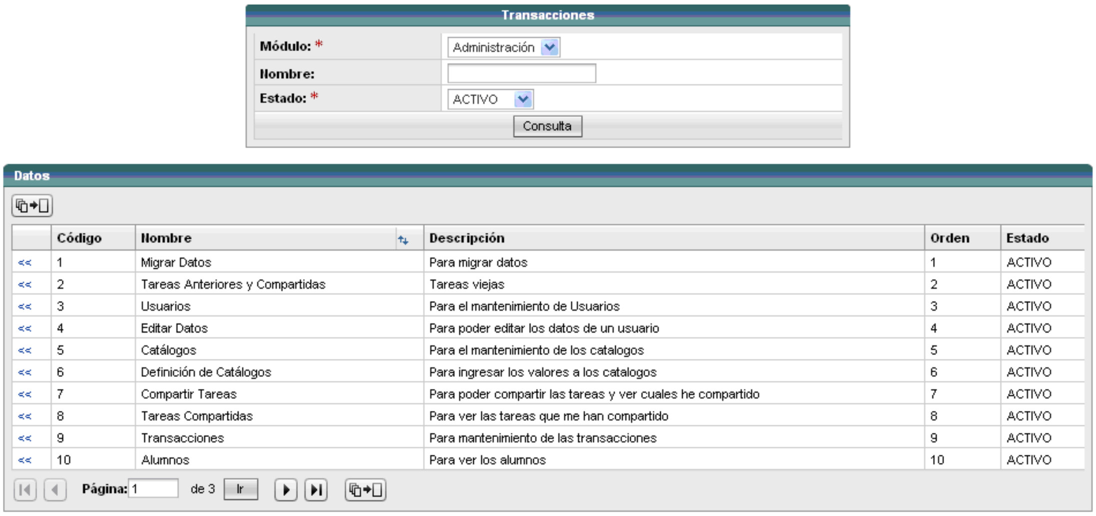
- Módulo.
- Código
- Nombre.
- Descripción.
- Url. Este campo denota la URL que debe mostrar al momento de dar clic sobre la opción del menú.
- Menu. Si este campo es seleccionado se muestra como opción dentro del menú que corresponde al módulo escogido.
- Personalizado. Si este campo es seleccionado se habilita la caja de texto para permitir ingresar una sentencia SQL que permitirá cargar las opciones del menú.
- Imagen. Este campo corresponde a la imagen que se mostrará en la opción del menú.
- Estado.

Si va a registrar una nueva transacción y no conoce el código que sigue puede escribir el número cero para que se registre con la secuencia siguiente.

Para hacer una consulta específica se debe es escribir el código a consultar y perder el foco del campo código para que realice una consulta asincrónica y se carguen los demás campos en el formulario.

Para modificar una transacción se debe primero realizar una consulta para que se carguen los datos de la transacción, modificar los campos que se desean cambiar y luego dar clic en el botón Guardar.

6. Consulta de Transacciones.



Módulo: * Administración
Nombre:
Estado: * ACTIVO

	Código	Nombre	Descripción	Orden	Estado
<<	1	Migrar Datos	Para migrar datos	1	ACTIVO
<<	2	Tareas Anteriores y Compartidas	Tareas viejas	2	ACTIVO
<<	3	Usuarios	Para el mantenimiento de Usuarios	3	ACTIVO
<<	4	Editar Datos	Para poder editar los datos de un usuario	4	ACTIVO
<<	5	Catálogos	Para el mantenimiento de los catalogos	5	ACTIVO
<<	6	Definición de Catálogos	Para ingresar los valores a los catalogos	6	ACTIVO
<<	7	Compartir Tareas	Para poder compartir las tareas y ver cuales he compartido	7	ACTIVO
<<	8	Tareas Compartidas	Para ver las tareas que me han compartido	8	ACTIVO
<<	9	Transacciones	Para mantenimiento de las transacciones	9	ACTIVO
<<	10	Alumnos	Para ver los alumnos	10	ACTIVO

Figura H.10: Consultar Transacciones.

En esta opción usted podrá consultar transacciones existentes ya sea de forma masiva o individual (Ver figura H.10).

Se puede realizar una consulta masiva dando clic en el botón Consulta, lo que mostrará todas las transacciones en una tabla en la parte inferior (Ver figura H.10). La consulta puede ser filtrada por los campos módulo, nombre y estado.

Para realizar una consulta específica puede dar clic sobre el hipervínculo “<<” en la primera columna de la tabla presentada y se enviará a la pantalla de Registrar y Modificar Transacciones (Ver figura H.9), luego se cargarán los datos de la transacción en los campos correspondientes del formulario.

7. Asignación de Transacciones a Perfil

Módulo: * Materias

Perfil: * Profesor

	Código	Nombre	Descripción	Estado
☐	1	Ver Tareas	Permite ver las tareas	ACTIVO
☑	2	Ver Paralelos	Permite ver los paralelos	ACTIVO
☑	3	Informe Tarea Profesor	Para ver el detalle de la tarea	ACTIVO
☑	4	Informe Respuesta	ver respuesta del alumno	ACTIVO
☐	5	Informe Tarea Alumno	Para ver el informe de la tarea del alumno	ACTIVO
☑	6	Alumnos Respuesta	Para ver respuesta de los alumnos	ACTIVO
☑	7	Nueva Tarea - Paso 1	Datos basicos	ACTIVO
☑	8	Nueva Tarea - Paso 2	Datos especificos	ACTIVO
☑	9	Nueva Tarea - Paso 3	Publicar datos	ACTIVO
☐	10	Responder Tarea	Subir la respuesta de una tarea	ACTIVO
☐	11	Grupos	Pantalla para formar los grupos	ACTIVO
☐	12	Grupo Formado	Visulaiza los integrantes del grupo	ACTIVO
☐	13	Respuesta Resultado	Resumen instantaneo de la respuesta subida del estudiante	ACTIVO

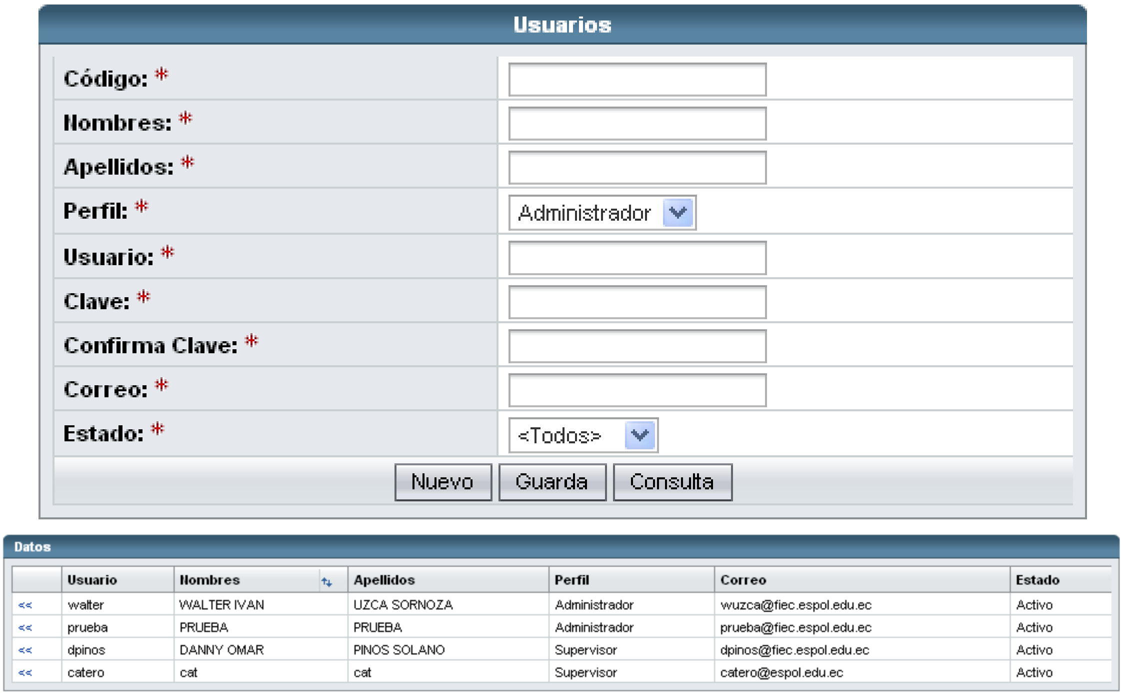
Figura H.11: Asignar transacciones a perfil.

En esta opción usted podrá asignar a un perfil las transacciones que desee. Esta tarea permite hacer configurable el menú para cada uno de los perfiles. Como todas las transacciones fueron registradas para un módulo específico, la consulta de transacciones se hace por módulo, para que luego sean asignadas al perfil seleccionado.

Para asignar transacciones a un perfil específico primero se escoge el módulo y el perfil con el que se desea trabajar, se da clic sobre el botón Consulta para que muestren las transacciones que pertenecen al módulo y automáticamente se marcarán las transacciones que están escogidas para el perfil. Usted puede desmarcar o marcar nuevas transacciones para luego dar clic sobre el botón Guardar dentro de la tabla que muestra la transacciones

(Ver figura H.11). La pantalla es tanto para consultar las asignaciones realizadas como para realizar nuevas asignaciones.

8. Registro y Mantenimiento de Usuarios.



Código: *

Nombres: *

Apellidos: *

Perfil: * Administrador

Usuario: *

Clave: *

Confirma Clave: *

Correo: *

Estado: * <Todos>

	Usuario	Nombres	Apellidos	Perfil	Correo	Estado
<<	walter	WALTER IVAN	UZCA SORNOZA	Administrador	wuzca@fiec.espol.edu.ec	Activo
<<	prueba	PRUEBA	PRUEBA	Administrador	prueba@fiec.espol.edu.ec	Activo
<<	dpinos	DANNY OMAR	PINOS SOLANO	Supervisor	dpinos@fiec.espol.edu.ec	Activo
<<	catero	cat	cat	Supervisor	catero@espol.edu.ec	Activo

Figura H.12: Crear y Modificar Usuarios.

En esta opción usted podrá registrar nuevos usuarios y modificar usuarios existentes (Ver figura H.8). La eliminación es lógica y no física, se maneja con el atributo denominado estado que determina si se encuentra activo o inactivo.

Para registrar un nuevo usuario debo presionar el botón Nuevo, llenar los campos necesarios y luego presionar el botón Guardar. Los campos necesarios son:

- Código.
- Nombres.
- Apellidos.

- Perfil.
- Usuario.
- Clave.
- Confirma Clave.
- Correo.
- Estado.

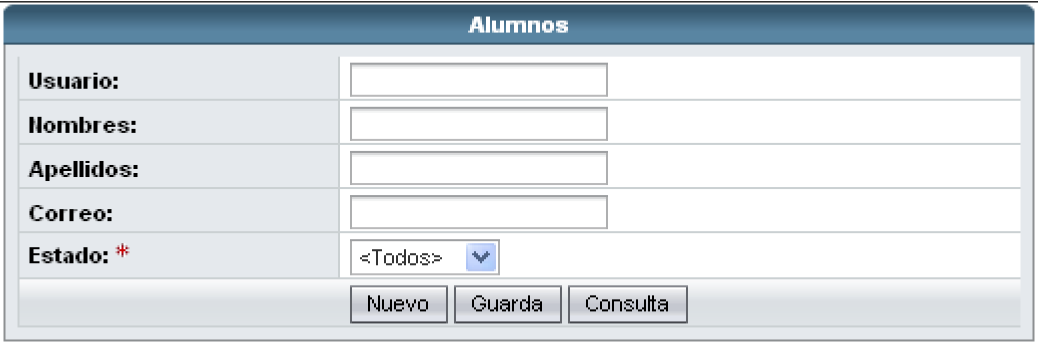
Si va a registrar un nuevo usuario y no conoce el código que sigue puede escribir el número cero para que se registre con la secuencia siguiente.

Se puede realizar una consulta masiva dando clic en el botón Consulta, lo que mostrará todos los usuarios en una tabla en la parte inferior (Ver figura H.12). La consulta puede ser filtrada por los campos nombres, apellidos y estado.

Para realizar una consulta específica puede dar clic sobre el hipervínculo “<<” en la primera columna de la tabla presentada y se cargarán los datos del usuario en los campos correspondientes del formulario. Otra forma de hacer una consulta específica es escribir el código a consultar y perder el foco del campo código para que realice una consulta asincrónica y se carguen los demás campos en el formulario.

Para modificar un usuario se debe primero realizar una consulta para que se carguen los datos del usuario, modificar los campos que se desean cambiar y luego dar clic en el botón Guardar.

9. Consulta y Modificación de Alumnos.



Alumnos

Usuario:	
Nombres:	
Apellidos:	
Correo:	
Estado: *	<Todos>

Figura H.13: Consultar y Modificar Alumnos.

En esta opción usted podrá consultar y modificar alumnos existentes (Ver figura H.13). La razón por la que no se puede ingresar nuevos alumnos es debido a que los archivos de migración contienen el listado de los alumnos que se usan en el sistema. Se puede realizar una consulta masiva dando clic en el botón Consulta, lo que mostrará todos los alumnos en una tabla en la parte inferior (Ver figura H.13). La consulta puede ser filtrada por los campos nombres, apellidos y estado.

Para realizar una consulta específica puede dar clic sobre el hipervínculo "<<" en la primera columna de la tabla presentada y se cargarán los datos del alumno en los campos correspondientes del formulario. Otra forma de hacer una consulta específica es escribir el código a consultar y perder el foco del campo código para que realice una consulta asincrónica y se carguen los demás campos en el formulario.

Para modificar un alumno se debe primero realizar una consulta para que se carguen los datos del alumno, modificar los campos que se desean cambiar y luego dar clic en el botón Guardar.

10. Consulta y Modificación de Profesores.

Profesores

Usuario:
Nombres:
Apellidos:
Correo:
Estado: * ACTIVO

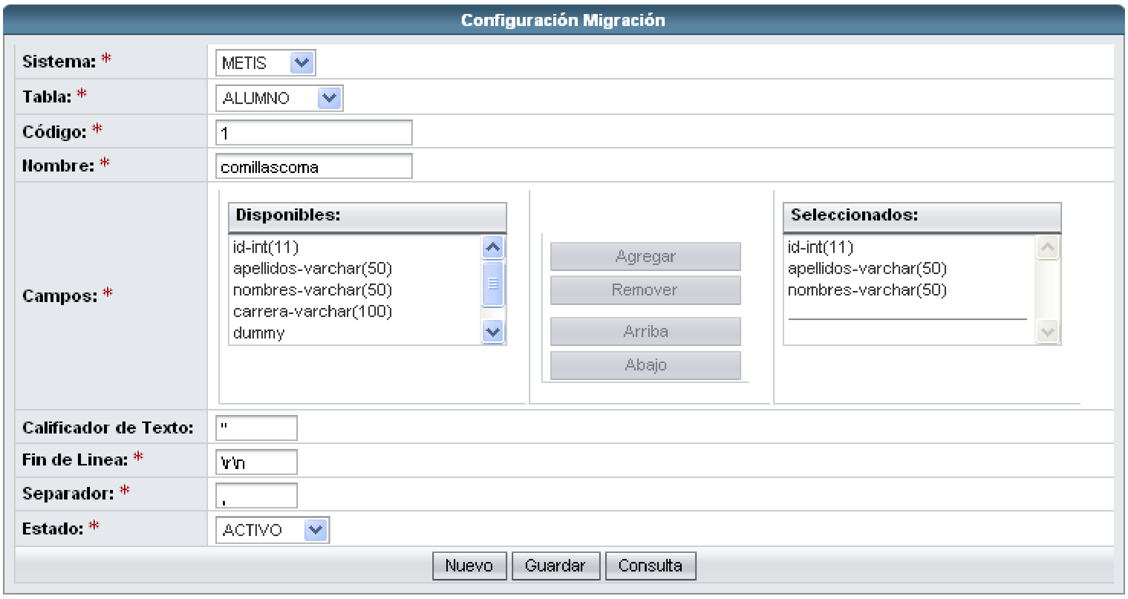
Figura H.14: Consultar y Modificar Profesores.

En esta opción usted podrá consultar y modificar profesores existentes (Ver figura H.14). La razón por la que no se puede ingresar nuevos profesores es debido a que los archivos de migración contienen el listado de los profesores que se usan en el sistema. Se puede realizar una consulta masiva dando clic en el botón Consulta, lo que mostrará todos los profesores en una tabla en la parte inferior (Ver figura H.14). La consulta puede ser filtrada por los campos nombres, apellidos y estado.

Para realizar una consulta específica puede dar clic sobre el hipervínculo “<<” en la primera columna de la tabla presentada y se cargarán los datos del profesor en los campos correspondientes del formulario. Otra forma de hacer una consulta específica es escribir el código a consultar y perder el foco del campo código para que realice una consulta asincrónica y se carguen los demás campos en el formulario.

Para modificar un profesor se debe primero realizar una consulta para que se carguen los datos del alumno, modificar los campos que se desean cambiar y luego dar clic en el botón Guardar.

11. Registrar y Modificar Parámetros de Migración.



Sistema: * METIS
Tabla: * ALUMNO
Código: * 1
Nombre: * comillascoma
Campos: *
Disponibles:
id-int(11)
apellidos-varchar(50)
nombres-varchar(50)
carrera-varchar(100)
dummy
Seleccionados:
id-int(11)
apellidos-varchar(50)
nombres-varchar(50)
Calificador de Texto: "
Fin de Linea: * \r\n
Separador: * ,
Estado: * ACTIVO

Figura H.15: Registrar y Modificar Parámetros de Migración.

En esta opción usted podrá registrar nuevos parámetros de migración y modificar parámetros de migración existentes (Ver figura H.15). La eliminación es lógica y no física, se maneja con el atributo denominado estado que determina si se encuentra activo o inactivo.

Para registrar un nuevo parámetro de migración debo presionar el botón Nuevo, llenar los campos necesarios y luego presionar el botón Guardar. Los campos necesarios son:

- Sistema. Este campo corresponde al Catálogo Sistema
- Tabla. Este campo corresponde al Catálogo Tablas
- Código.
- Nombre.
- Campos. La lista izquierda representa los campos correspondientes a la tabla que se encuentra en la base. Mientras que en la lista de la derecha está los campos que vienen en el archivo en su correspondiente orden. Si existe un columna en el archivo que no es tomado en cuenta para un campo de la base entonces se utilizará el

campo “dummy”, éste es el único campo que se puede seleccionar más de una vez, todos los demás campos se pueden seleccionar solamente una vez.

- Calificador de Texto.
- Fin de Línea.
- Separador.
- Estado.

Si va a registrar un nuevo parámetro de migración y no conoce el código que sigue puede escribir el número cero para que se registre con la secuencia siguiente.

Para hacer una consulta específica se debe es escribir el código a consultar y perder el foco del campo código para que realice una consulta asincrónica y se carguen los demás campos en el formulario.

Para modificar un parámetro de migración se debe primero realizar una consulta para que se carguen los datos del parámetro de migración, modificar los campos que se desean cambiar y luego dar clic en el botón Guardar

12. Consultar Parámetros de Migración.

Sistema: * METIS
Tabla: * ALUMNO
Estado: * ACTIVO

	Nombre	Campos	Separador	Calificador Texto	Fin Línea	Estado
<<	comillascoma	id,apellidos,nombres,dummy	,	"	\r\n	ACTIVO
<<	comillasptocoma	id,apellidos,nombres,dummy	;	"	\r\n	ACTIVO

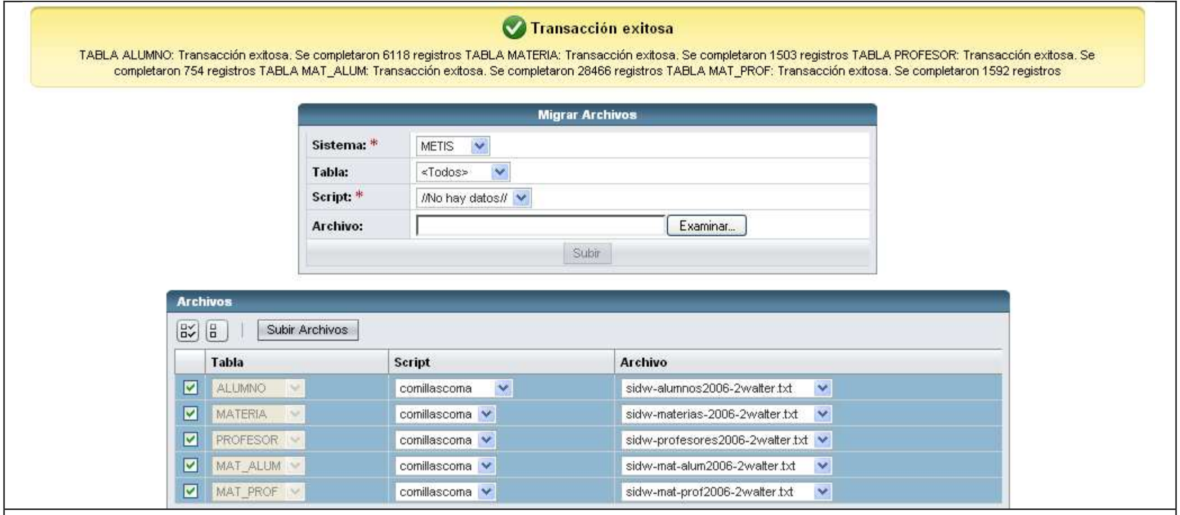
Figura H.16: Consultar Parámetros de Migración.

En esta opción usted podrá consultar parámetros de migración existentes ya sea de forma masiva o individual (Ver figura H.16).

Se puede realizar una consulta masiva dando clic en el botón Consulta, lo que mostrará todas los parámetros de migración en una tabla en la parte inferior (Ver figura H.16). La consulta puede ser filtrada por los campos sistema, tabla y estado.

Para realizar una consulta específica puede dar clic sobre el hipervínculo “<<” en la primera columna de la tabla presentada y se enviará a la pantalla de Registrar y Modificar Parámetros de Migración (Ver figura H.15), luego se cargarán los datos del parámetro de migración en los campos correspondientes del formulario

13. Migrar Archivos.



Transacción exitosa

TABLA ALUMNO: Transacción exitosa. Se completaron 6118 registros TABLA MATERIA: Transacción exitosa. Se completaron 1503 registros TABLA PROFESOR: Transacción exitosa. Se completaron 754 registros TABLA MAT_ALUM: Transacción exitosa. Se completaron 28466 registros TABLA MAT_PROF: Transacción exitosa. Se completaron 1592 registros

Migrar Archivos

Sistema: *	METIS
Tabla:	<Todos>
Script: *	//No hay datos//
Archivo:	

Archivos

	Tabla	Script	Archivo
☑	ALUMNO	comillascoma	sidw-alumnos2006-2walter.txt
☑	MATERIA	comillascoma	sidw-materias-2006-2walter.txt
☑	PROFESOR	comillascoma	sidw-profesores2006-2walter.txt
☑	MAT_ALUM	comillascoma	sidw-mat-alum2006-2walter.txt
☑	MAT_PROF	comillascoma	sidw-mat-prof2006-2walter.txt

Figura H.17: Migrar Archivos.

En esta opción usted podrá migrar los archivos que tendrán el formato especificado en los parámetros de migración para subir los datos a las tablas de migración de la base del sistema SCAT (Ver figura H.17).

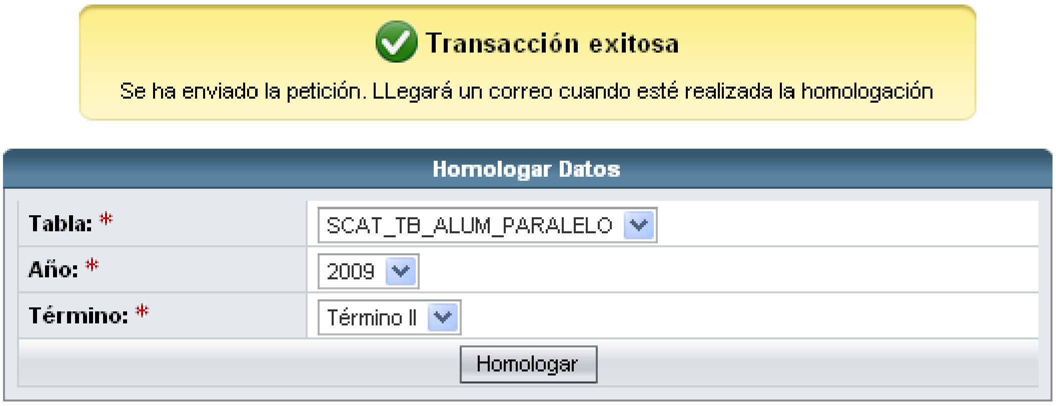
Para poder realizar la migración se deben ingresar los siguientes datos:

- Sistema. Este campo corresponde al Catálogo Sistema.
- Tabla. Este campo corresponde al Catálogo Tablas adicionado la opción de “<Todos>”.

- Script. Este campo me especifica el tipo de parámetro de migración que debió ser configurado en la pantalla de Parámetro de Configuración (Ver figura H.15)
- Archivo. Este campo permite seleccionar el archivo a migrar.

Si se escoge "<Todos>" en Tabla se espera recibir un archivo de extensión ZIP, si escoge una Tabla en particular se esperará un archivo de texto. Si escoge una Tabla y da clic en botón Subir se migrará el archivo y se mostrará los resultados. Si escoge "<Todos>" y da clic en el botón Subir se cargará una tabla para que pueda seleccionar que archivo corresponde a cada Tabla y deberá escoger "Subir Archivos" para realizar la migración masiva, luego se mostrarán los resultados. Solamente se migrarán los archivos que fueron seleccionados.

14. Homologar Datos.



Transacción exitosa

Se ha enviado la petición. LLegará un correo cuando esté realizada la homologación

Homologar Datos

Tabla: *	SCAT_TB_ALUM_PARALELO
Año: *	2009
Término: *	Término II

Figura H.18: Homologar Datos.

En esta opción usted podrá homologar datos de las tablas de migración a las tablas de trabajo en la base SCAT. Esto se trata de hacer las inserciones en las tablas pertinentes a la aplicación desde las tablas a las cuales les fueron migrados los archivos.

Para realizar la homologación se deben ingresar los siguientes datos:

- Tabla.

- Año.
- Término

Luego de dar clic en el botón “Homologar” se mostrará los resultados. Como la transacción toma tiempo, una vez que esté homologado la tabla escogida se enviará un correo con los resultados en un documento de texto adjunto.

15. Administración Alumno Paralelo

Materia: Fundamentos de Programación

Paralelo: 001

Alumno:

	Matricula	Apellidos	Nombres	Estado
☐	198003691	RIERA PERALTA	PATRICIA CECILIA	ACTIVO
☐	198006330	ZAMORA ZU?IGA	SEGUNDO ANGEL	ACTIVO
☐	198104713	VACA MENDIETA	MARITZA MONICA	ACTIVO
☐	198701567	LARA GARCIA	MANUEL ANTONIO	ACTIVO
☐	199102526	LARRETA TORRES	ERWIN WILFRIDO	ACTIVO
☐	200109692	BARCOS SINCHE	CARLOS FERNANDO	ACTIVO
☐	200304939	APUPALO DELROSARIO	LISSETTE VERONICA	ACTIVO

Figura H.19: Agregar – quitar alumnos de un paralelo.

En esta opción usted podrá agregar o quitar alumnos de un paralelo. Todos los alumnos tendrán el atributo estado, el cual sirve para determinar si el alumno se encuentra activo o inactivo en el paralelo.

Para poder agregar un alumno a un paralelo primero se debe consultar el alumno que se agregará. Existen dos formas para la búsqueda; la primera es escogiendo Buscar para que aparezca la ventana de búsqueda de alumnos para hacer consultas específicas (Ver figura H.19), una vez encontrado el alumno lo selecciona para que se carguen los datos en la pantalla principal.

La segunda es escribiendo el número de matrícula en la primera caja de texto; de tal modo que se mostrará el nombre en la siguiente caja de texto para verificar que es alumno indicado. Una vez seleccionado el alumno se da clic en el botón Agregar para poder vincularlo al paralelo.

Nombre: carlos

Matrícula:

	Matricula	Apellidos	Nombres
<<	199004110	YANCE CARVAJAL	CARLOS LEONIDAS
<<	199700113	MONTOYA ZURITA	CARLOS FERNANDO
<<	199714320	ITURRALDE MARURI	CARLOS FERNANDO
<<	199800210	JARA VALDIVIEZO	CARLOS XAVIER
<<	199805003	VALLE CHIRIBOGA	CARLOS ALEX
<<	199819277	VILLACIS INFANTE	CARLOS OMAR
<<	199820846	ROBALINO VITERI	CARLOS SAMUEL
<<	199902792	ANDRADE ACOSTA	CARLOS LUIS
<<	199905894	GUERRERO VALAREZO	CARLOS LEOPOLDO
<<	199906769	PINILLO LOPEZ	CARLOS ALBERTO

Página: 1 de 19

Figura H.20: Búsqueda de alumnos.

Una vez agregado el alumno ya no se puede reversar la transacción, lo que se puede hacer es cambiarle el estado a inactivo y será como si el alumno ya no esté en el paralelo. Esto puede servir para los alumnos que se retiran o se cambian de paralelo.

16. Registro y Mantenimiento de Períodos.

Año: *	
Término: *	
Estado: *	ACTIVO

	Año	Termino	Estado
<<	2006	1	INACTIVO
<<	2006	2	INACTIVO
<<	2006	3	INACTIVO
<<	2007	1	INACTIVO
<<	2007	2	INACTIVO
<<	2007	3	INACTIVO
<<	2008	1	ACTIVO
<<	2008	2	INACTIVO
<<	2008	3	INACTIVO
<<	2009	1	INACTIVO

Página: 1 de 2 Ir

Figura H.21: Crear y Modificar períodos.

En esta opción usted podrá registrar nuevos períodos y modificar períodos existentes (Ver figura H.21). La eliminación es lógica y no física, se maneja con el atributo denominado estado que determina si se encuentra activo o inactivo. Solamente debe existir un período activo.

Para registrar un nuevo período debo presionar el botón Nuevo, llenar los campos necesarios y luego presionar el botón Guardar. Los campos necesarios son:

- Año.
- Término.
- Estado.

Se puede realizar una consulta masiva dando clic en el botón Consulta, lo que mostrará todos los períodos en una tabla en la parte inferior (Ver figura H.21). La consulta puede ser filtrada por los campos año, término y estado.

Para realizar una consulta específica puede dar clic sobre el hipervínculo “<<” en la primera columna de la tabla presentada y se cargarán los datos del período en los campos correspondientes del formulario.

Para modificar un período se debe primero realizar una consulta para que se carguen los datos del período, modificar los campos que se desean cambiar y luego dar clic en el botón Guardar.

Si desea activar otro nuevo período debe primero cambiar el estado del período activo a inactivo para que le permita registrar como activo el nuevo período.

17. Activar y desactivar Paralelos.

Datos para mostrar paralelos

Año: *	2008
Término: *	Termino 1
Materia: *	Fundamentos de Programación

Paralelos

	Paralelo	Profesor	Estado
☐	001	ABAD ROBALINO CRISTINA	ACTIVO
☐	005	ABAD ROBALINO CRISTINA	ACTIVO

Figura H.22: Activar y desactivar paralelos.

En esta opción usted podrá activar y desactivar paralelos del período actual. La razón por la que no se puede ingresar nuevos paralelotes debido a que los archivos de migración contienen el listado de los paralelos que se usan en el sistema.

Para poder activar o desactivar paralelos primero debemos realizar una consulta masiva dando clic en el botón Consulta, lo que mostrará todos los paralelos en una tabla en la parte inferior (Ver figura H.22). La consulta puede ser filtrada por los campos año, término y materia. En la tabla podemos cambiar el estado de uno o varios paralelos a la vez previamente habiéndolo chequeado. Si no es chequeado no es tomado en cuenta. Cuando haya realizado los cambios debe escoger Guardar para que los cambios tomen efecto.

18. Cambio de Clave.

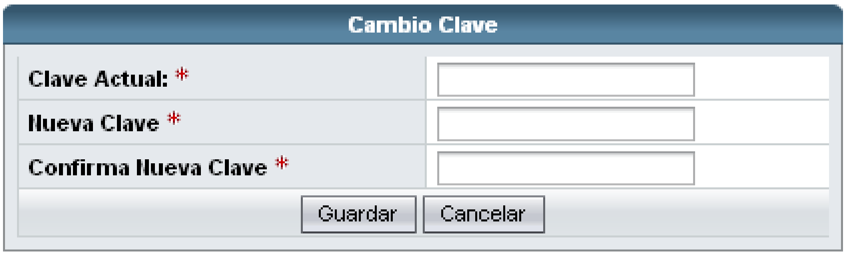
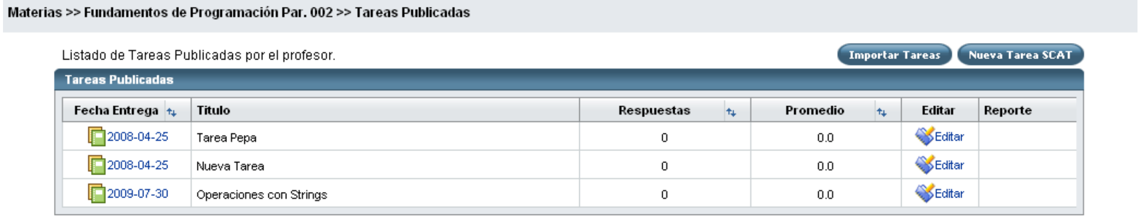


Figura H.23: Cambiar Clave.

En esta opción usted podrá cambiar su clave por una nueva. Para poder cambiar clave debe de estar dentro del sistema. Debe de ingresar su clave actual para verificar que se trata de usuario actual y debe de ingresar la nueva clave dos veces para evitar que se haya confundido al momento de haber ingresado la nueva clave. (Ver figura F.23)

Un usuario no puede cambiar la clave de otro. Solamente puede cambiar su clave. El administrador puede ayudarle sino se acuerda de su clave, puesto este el único que lo tiene permitido.

19. Publicar Tarea



Listado de Tareas Publicadas por el profesor.

Tareas Publicadas

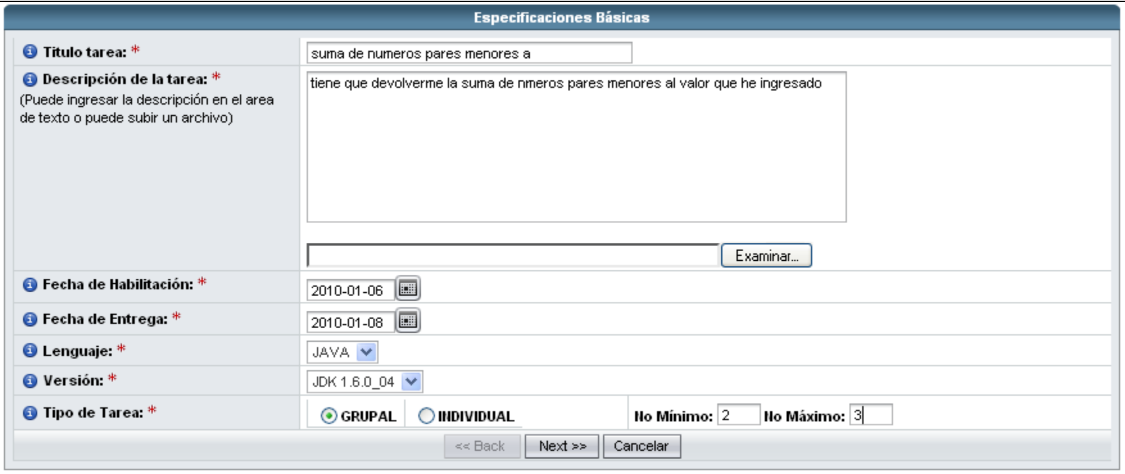
Fecha Entrega	Titulo	Respuestas	Promedio	Editar	Reporte
2008-04-25	Tarea Pepa	0	0.0	Editar	
2008-04-25	Nueva Tarea	0	0.0	Editar	
2009-07-30	Operaciones con Strings	0	0.0	Editar	

Figura H.24: Publicar Tarea.

En esta opción usted podrá publicar una tarea a un paralelo específico. Si usted dicta la misma materia en más de un paralelo tiene la opción de publicarla en ambos. Para publicar la tarea debe de escoger el paralelo al cual desea publicar la tarea en el menú de Materias en la parte izquierda de la pantalla. Luego se muestra el listado de tareas publicadas hasta la fecha en el paralelo (Ver figura H.24). Debemos dar clic sobre el botón Nueva Tarea SCAT para empezar la secuencia de pasos previos a la publicación.

Paso 1. Registro de especificaciones básicas (Ver figura H.25) como:

- Título tarea
- Descripción de la tarea
- Fecha de Habilitación (mayor o igual a la actual)
- Fecha de Entrega (mayor a la de habilitación)
- Lenguaje
- Versión
- Tipo de Tarea [GRUPAL|INDIVIDUAL]



Especificaciones Básicas

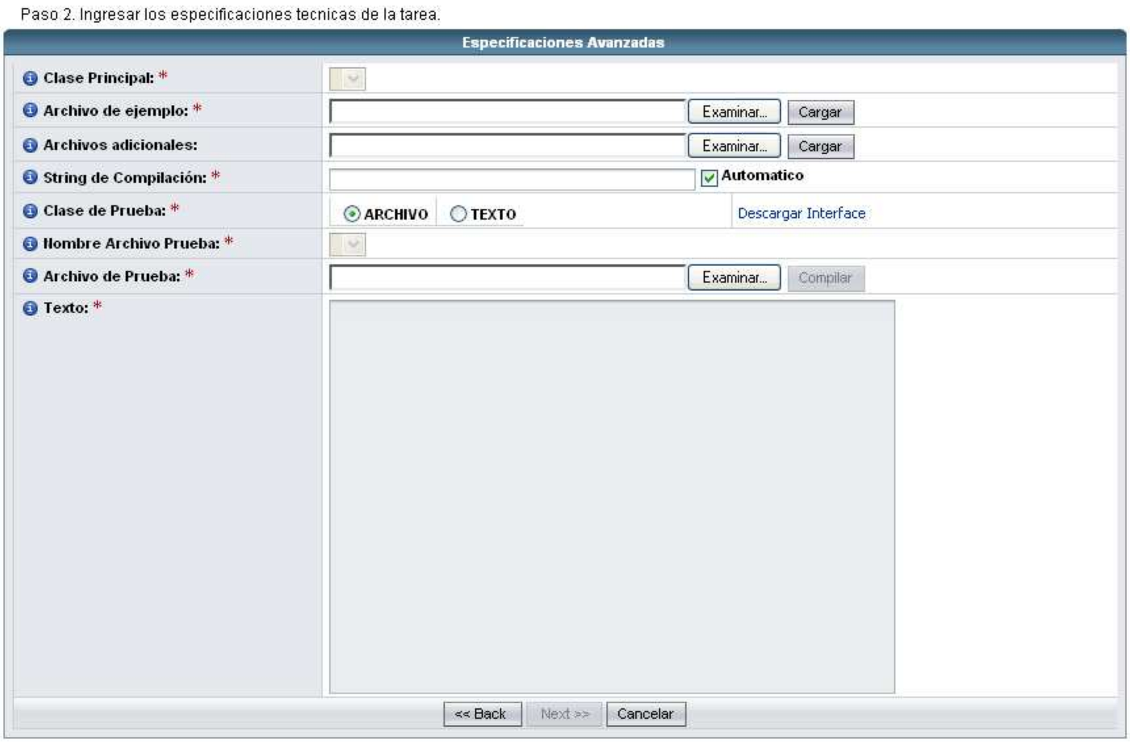
Titulo tarea: *	suma de numeros pares menores a
Descripción de la tarea: * (Puede ingresar la descripción en el area de texto o puede subir un archivo)	tiene que devolverme la suma de nmeros pares menores al valor que he ingresado
Fecha de Habilitación: *	2010-01-06
Fecha de Entrega: *	2010-01-08
Lenguaje: *	JAVA
Versión: *	JDK 1.6.0_04
Tipo de Tarea: *	● GRUPAL ○ INDIVIDUAL **No Mínimo:** 2 **No Máximo:** 3

Figura H.25: Publicar Tarea – Especificaciones Básicas.

Paso 2. Se continúa con el cuerpo de la tarea. Se debe ingresar los siguientes datos:

- Archivo de Ejemplo
- Archivos Adicionales
- String de Compilación
- Clase de Prueba
- Nombre Archivo Prueba
- Archivo de Prueba | TEXTO

Se cargará primero el Archivo de Ejemplo una vez que lo haya seleccionado, debemos presionar el botón Cargar para cargar la clase y ver si tienes errores de compilación. (Ver figura H.26)

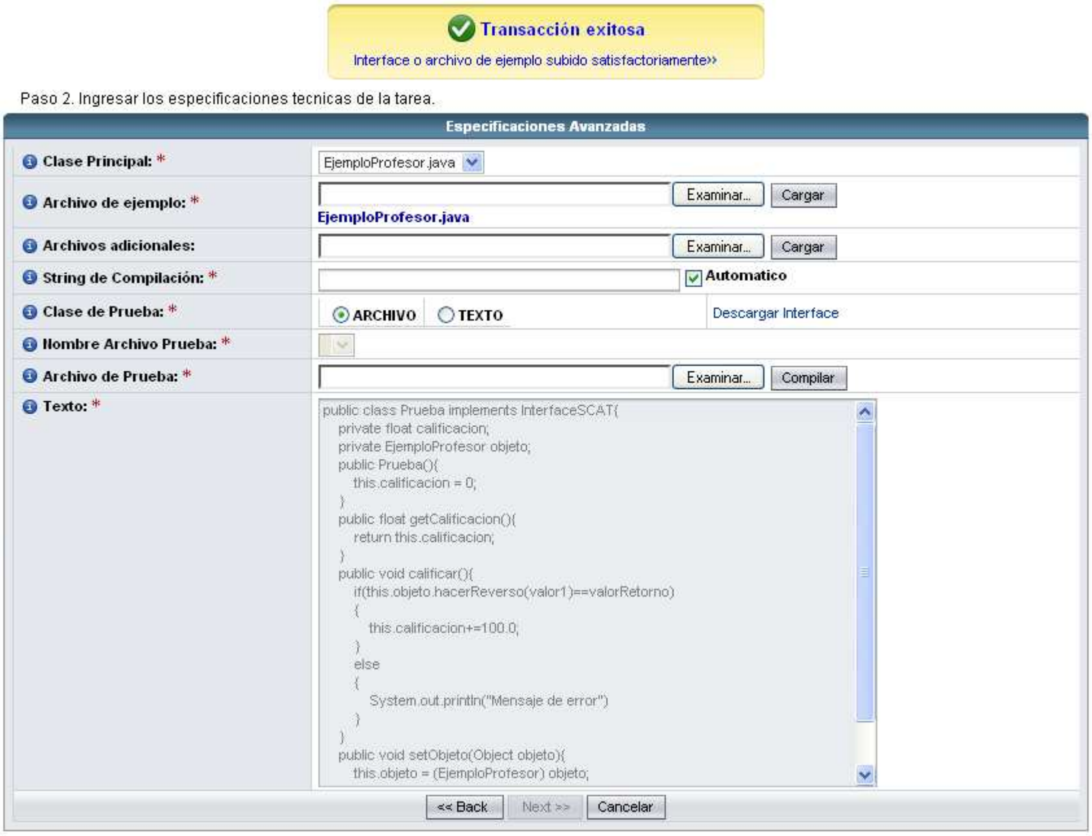

Paso 2. Ingresar los especificaciones tecnicas de la tarea.

Especificaciones Avanzadas

Clase Principal: *
Archivo de ejemplo: *
Archivos adicionales:
String de Compilación: * ☑ Automatico
Clase de Prueba: * ● ARCHIVO ○ TEXTO
Nombre Archivo Prueba: *
Archivo de Prueba: *
Texto: *

Figura H.26: Publicar Tarea – Especificaciones Avanzadas.

Si no tiene errores se mostrará que la clase fue cargada exitosamente. Adicionalmente a esto se cargará un esqueleto en el área de texto correspondiente al Archivo de Prueba (Ver figura H.27) para usarlo si es deseo del usuario.



Transacción exitosa

Interface o archivo de ejemplo subido satisfactoriamente»

Paso 2. Ingresar los especificaciones tecnicas de la tarea.

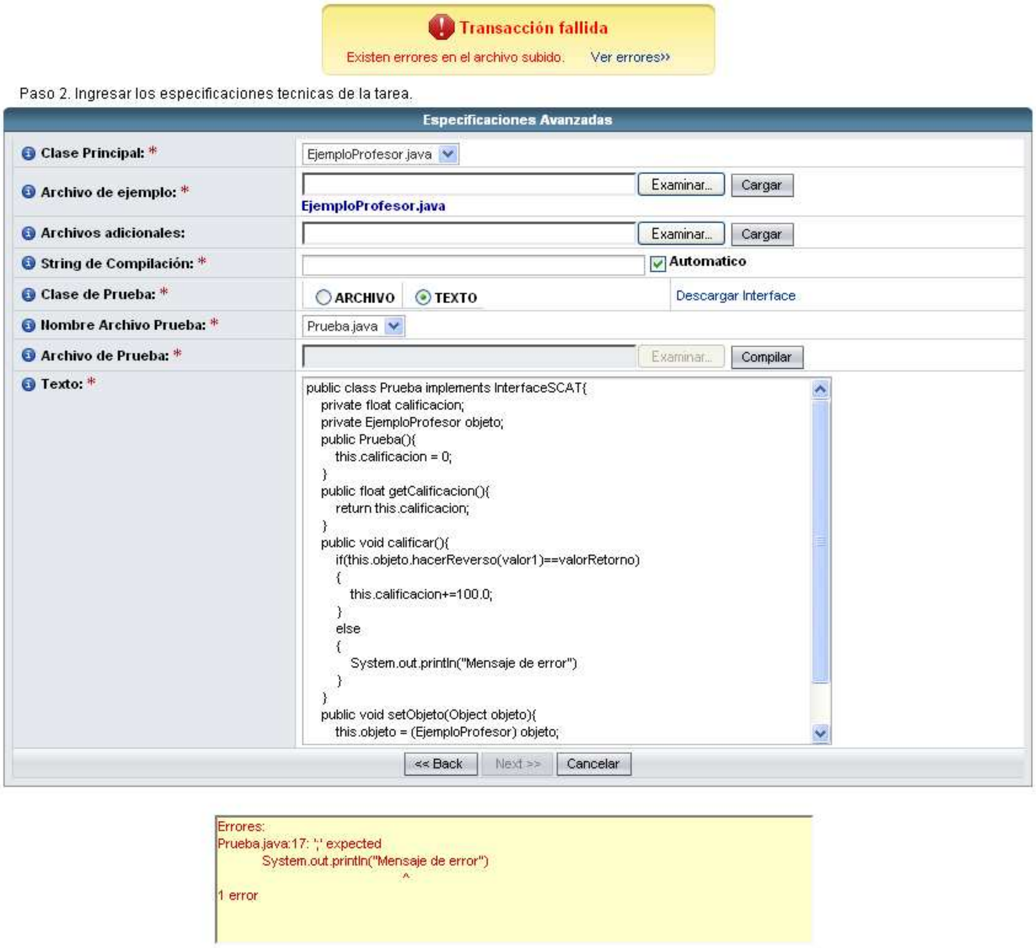
Especificaciones Avanzadas

Clase Principal: *	EjemploProfesor.java
Archivo de ejemplo: *	**EjemploProfesor.java**
Archivos adicionales:	
String de Compilación: *	☑ **Automatico**
Clase de Prueba: *	● **ARCHIVO** ○ **TEXTO**
Nombre Archivo Prueba: *	
Archivo de Prueba: *	
Texto: *	

```
public class Prueba implements InterfaceSCAT{
    private float calificacion;
    private EjemploProfesor objeto;
    public Prueba(){
        this.calificacion = 0;
    }
    public float getCalificacion(){
        return this.calificacion;
    }
    public void calificar(){
        if(this.objeto.hacerReverso(valor1)==valorRetorno)
        {
            this.calificacion+=100.0;
        }
        else
        {
            System.out.println("Mensaje de error")
        }
    }
    public void setObjeto(Object objeto){
        this.objeto = (EjemploProfesor) objeto;
```

Figura H.27: Publicar Tarea – Especificaciones Avanzadas – Archivo Ejemplo.

Luego de seleccionar Clase de Prueba (ARCHIVO|TEXTO) se debe bien seleccionar el ARCHIVO o ingresar TEXTO para poder luego presionar Cargar para cargar la clase y ver si tiene errores de compilación. Si tiene errores se mostrará en una lista los errores. El usuario tendrá que corregir los errores para poder compilar nuevamente y así poder avanzar con la publicación de la tarea. (Ver figura H.28)
]



Existen errores en el archivo subido.

Paso 2. Ingresar los especificaciones tecnicas de la tarea.

Clase Principal: * EjemploProfesor.java

Archivo de ejemplo: * EjemploProfesor.java

Archivos adicionales:

String de Compilación: * ☑ Automatico

Clase de Prueba: * ○ ARCHIVO ● TEXTO

Nombre Archivo Prueba: * Prueba.java

Archivo de Prueba: *

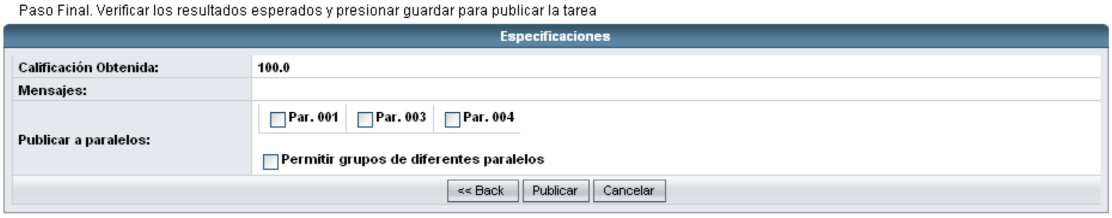
Texto: *

```
public class Prueba implements InterfaceSCAT{
   private float calificacion;
   private EjemploProfesor objeto;
   public Prueba(){
      this.calificacion = 0;
   }
   public float getCalificacion(){
      return this.calificacion;
   }
   public void calificar(){
      if(this.objeto.hacerReverso(valor1)==valorRetorno)
      {
         this.calificacion+=100.0;
      }
      else
      {
         System.out.println("Mensaje de error")
      }
   }
   public void setObjeto(Object objeto){
      this.objeto = (EjemploProfesor) objeto;
```

```
Errores:
Prueba.java:17: ';' expected
          System.out.println("Mensaje de error")
                                                ^
1 error
```

Figura H.28: Publicar Tarea – Especificaciones Avanzadas – Archivo Prueba.

Si no existe errores se procederá a calificar la tarea que intenta publicar y se mostrará el resultado. También el profesor tiene la opción de publicar la misma tarea para otros paralelos y poder permitir grupos de paralelos diferentes. (Ver figura H.29)



Paso Final. Verificar los resultados esperados y presionar guardar para publicar la tarea

Calificación Obtenida: 100.0

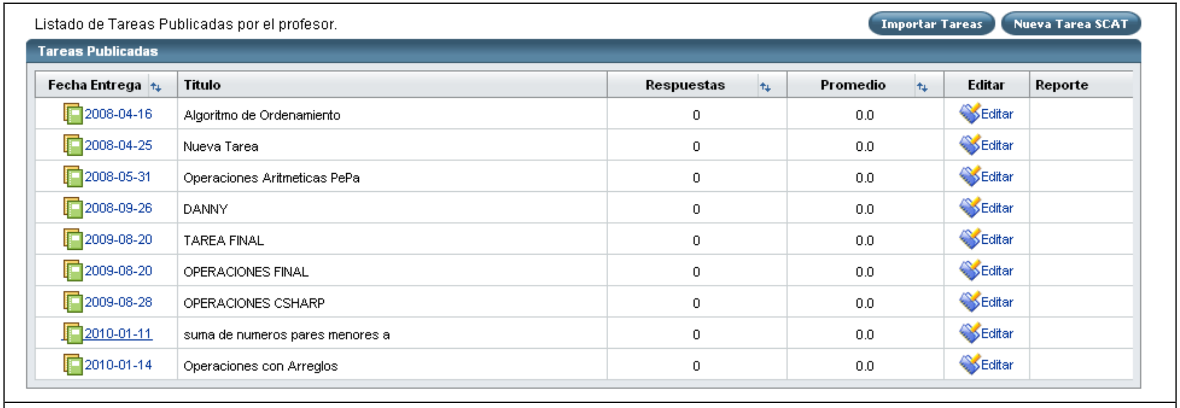
Mensajes:

Publicar a paralelos: ☐ Par. 001 ☐ Par. 003 ☐ Par. 004

☐ Permitir grupos de diferentes paralelos

Figura H.29: Publicar Tarea – Publicar.

Luego de presionar en Publicar aparecerá la tarea de dentro del listado de Tareas del profesor pero no aparecerá publicada hasta la fecha en que el haya escogido que se publique. (Ver figura H.30)



Listado de Tareas Publicadas por el profesor.

Tareas Publicadas

Fecha Entrega	Titulo	Respuestas	Promedio	Editar	Reporte
2008-04-16	Algoritmo de Ordenamiento	0	0.0	Editar	
2008-04-25	Nueva Tarea	0	0.0	Editar	
2008-05-31	Operaciones Aritmeticas PePa	0	0.0	Editar	
2008-09-26	DANNY	0	0.0	Editar	
2009-08-20	TAREA FINAL	0	0.0	Editar	
2009-08-20	OPERACIONES FINAL	0	0.0	Editar	
2009-08-28	OPERACIONES CSHARP	0	0.0	Editar	
2010-01-11	suma de numeros pares menores a	0	0.0	Editar	
2010-01-14	Operaciones con Arreglos	0	0.0	Editar	

Figura H.30: Publicar Tarea – Listar Tareas Publicadas.

20. Mantenimiento de Grupo

Listado de Tareas Publicadas y respuestas subidas.

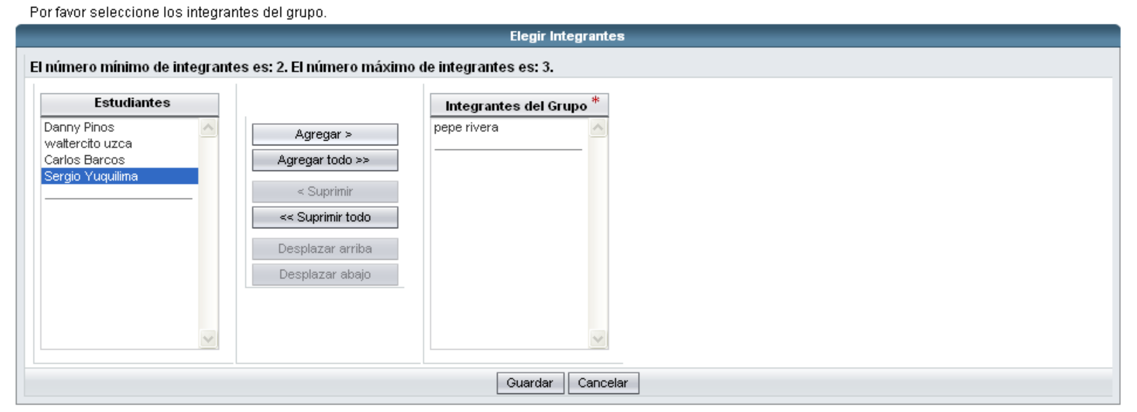
Tareas Publicadas

Fecha Entrega	Titulo Tarea	Grupo	Respuesta	Calificacion	Modificar
2009-08-28	OPERACIONES CSHARP	Grupos	2009-08-05	75.00	Modificar
2010-01-14	Operaciones con Arreglos	Grupos	Responder	---	---
2010-01-08	Operaciones con Arreglos	---	Responder	---	---
2010-01-11	suma de numeros pares menores a	Grupos	Responder	---	---

Figura H.31: Crear Grupo – Listado de Tareas.

En esta opción usted podrá crear un grupo para poder responder la tarea que lo exige. No todas las tareas exigen grupos, es cuestión de cómo el profesor la haya publicado. El estudiante tiene que poder crear un grupo mientras la tarea exija que sea obligación formar grupos.

Para poder crear un grupo debe primero seleccionar el paralelo del menú Materias de la parte izquierda de la pantalla. Se muestra el listado de tareas. (Ver figura F.31) Solamente si aparece el icono representativo de Grupos quiere decir que la tarea es grupal y que primero debe de crear el grupo antes de poder responderla. Debemos entonces dar clic sobre el icono de Grupo, luego se muestra la pantalla de mantenimiento de Grupo (Ver figura H.32) que consiste des dos listados, el de la izquierda muestra los alumnos que pueden formar parte de un grupo y el listado de la derecha muestra los alumnos que forman parten del grupo hasta el momento. Se pueden agregar o suprimir con los botones de acciones que se encuentran en medio de cada listado. En la parte superior me indica el número mínimo y máximo de integrantes que espera el grupo a conformar.

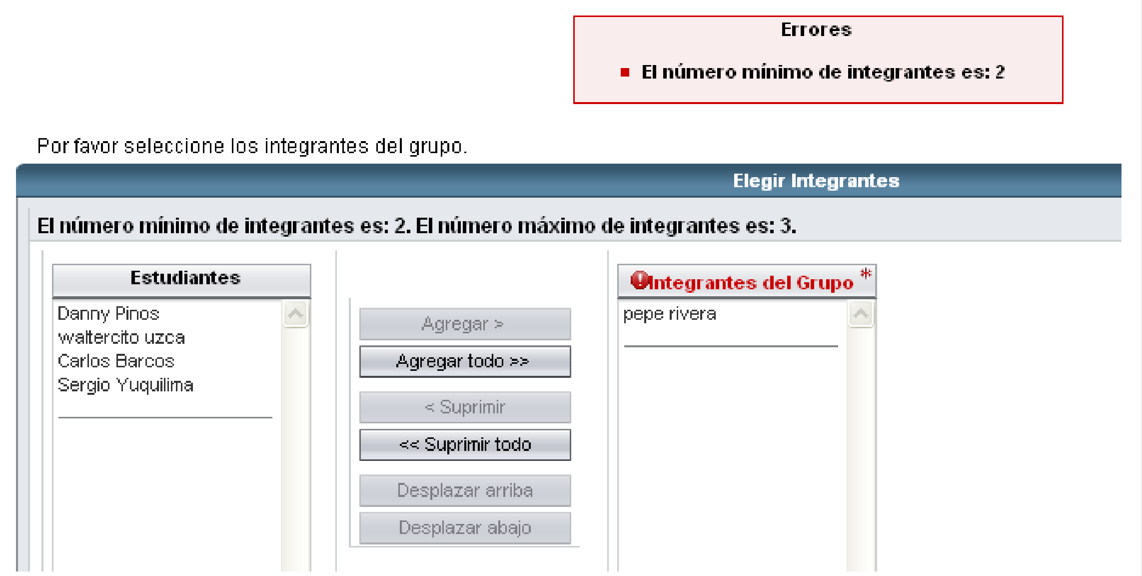


Por favor seleccione los integrantes del grupo.

El número mínimo de integrantes es: 2. El número máximo de integrantes es: 3.

Figura H.32: Crear y Modificar Grupo – Mantenimiento Grupo.

Si la tarea es una tarea grupal y no tiene permitido formar grupos de un integrante, no podrá responder la tarea mientras no hay formado el grupo previamente. Hay que respetar el número mínimo y máximo de integrantes al formar el grupo o no se lo dejará formar (Ver figura H.33)



Errores

- El número mínimo de integrantes es: 2

Por favor seleccione los integrantes del grupo.

Elegir Integrantes

El número mínimo de integrantes es: 2. El número máximo de integrantes es: 3.

Estudiantes

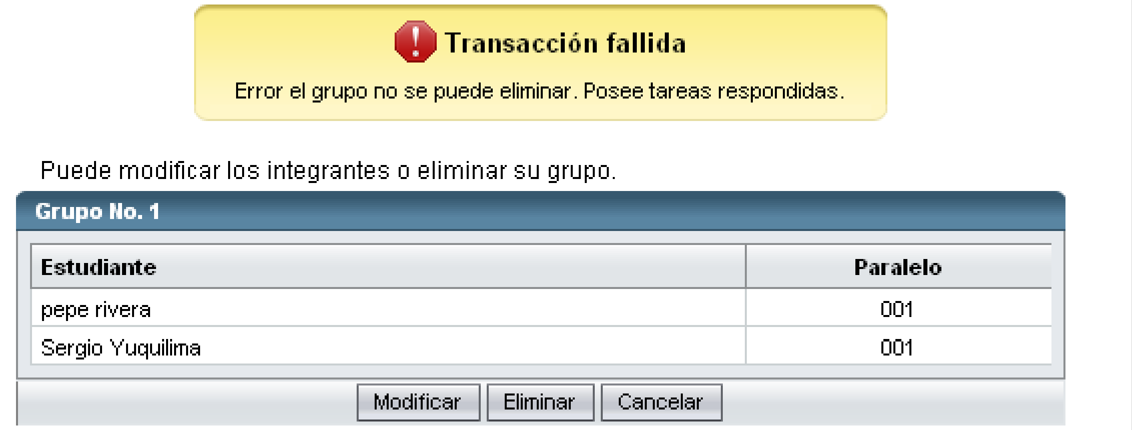
Danny Pinos
waltercito uzca
Carlos Barcos
Sergio Yuquilima

Integrantes del Grupo *

pepe rivera

Figura H.33: Crear y Modificar Grupo – Mensaje de Error.

Mientras no haya respondido la tarea el alumno puede crear, modificar o eliminar el grupo. Una vez que se hay respondido la tarea ya no se puede modificar el grupo, mucho menos eliminarlo. (Ver figura H.34)



Transacción fallida

Error el grupo no se puede eliminar. Posee tareas respondidas.

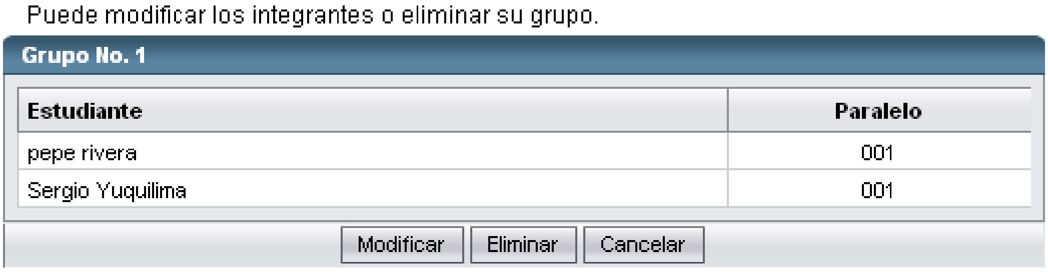
Puede modificar los integrantes o eliminar su grupo.

Grupo No. 1

Estudiante	Paralelo
pepe rivera	001
Sergio Yuquilima	001

Figura H.34: Consultar Grupo – Mensaje de Error.

Si el grupo es formado pero la tarea no es respondida, y uno de los integrantes quiere cambiarse de grupo, el grupo se puede modificar para sacar al estudiante de un grupo y así pueda pertenecer a otro; o en su defecto eliminar el grupo y crear uno nuevo. Una vez que haya dado clic en el botón Guardar su grupo será creado y se mostrará la pantalla con el número de grupo creado y los integrantes escogidos. (Ver figura H.35)



Puede modificar los integrantes o eliminar su grupo.

Grupo No. 1

Estudiante	Paralelo
pepe rivera	001
Sergio Yuquilima	001

Figura H.35: Consultar Grupo – Mantenimiento Grupo.

En ésta página de mantenimiento el Estudiante puede modificar los integrantes o en su caso eliminar el grupo. Si presiona Cancelar iremos nuevamente al listado de tareas.

21. Responder Tarea

Listado de Tareas Publicadas y respuestas subidas.

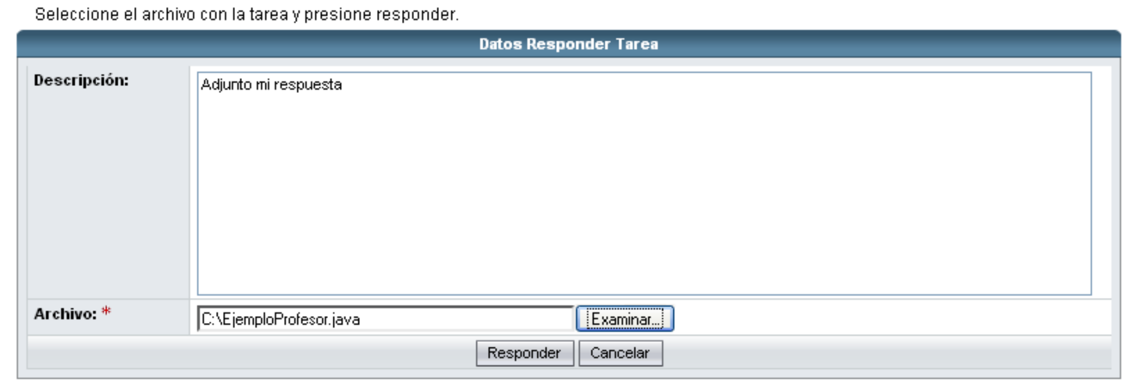
Tareas Publicadas

Fecha Entrega	Titulo Tarea	Grupo	Respuesta	Calificacion	Modificar
2009-08-28	OPERACIONES CSHARP	Grupos	2009-08-05	75.00	Modificar
2010-01-14	Operaciones con Arreglos	Grupos	Responder	---	---
2010-01-08	Operaciones con Arreglos	---	Responder	---	---
2010-01-11	suma de numeros pares menores a	Grupos	Responder	---	---

Página: 2 de 2

Figura H.36: Responder Tarea – Listado de Tareas.

Para poder responder una tarea el estudiante primero debe seleccionar del menú Materias la materia de la cual piensa responder la tarea. Se muestra su listado de tareas. En la columna de Respuesta aparece el hipervínculo “Responder” al cual le damos clic para poder contestar la tarea. Si aparece el hipervínculo “Responder” quiere decir que no se ha respondido la tarea. Si la tarea es una tarea grupal y no tiene permitido formar grupos de un integrante no podrá responder la tarea mientras no hay formado el grupo previamente. El hipervínculo “Responder” será habilitado cuando no existan problemas de grupo y se encuentre dentro del tope máximo de entrega. Le damos clic al enlace para que se muestre la pantalla de responder tarea. (Ver figura H.37)



Seleccione el archivo con la tarea y presione responder.

Datos Responder Tarea

Descripción:	Adjunto mi respuesta
Archivo: *	C:\EjemploProfesor.java

Figura H.37: Responder Tarea – Datos Responder Tarea.

Para responde la tarea debe seleccionar el archivo de respuesta y dar clic en el botón Responder. Adicionalmente pude escribir una pequeña descripción pero que no es obligatoria (Ver figura H.37). Una vez que hay presionado Responder el archivo es llevado al servidor y es calificado por la clase de Prueba del Profesor para luego mostrar los resultados obtenidos. Si existen errores de compilación estos se mostrarán para que puedan ser corregidos.

Resultados de la respuesta

Resultados de la respuesta

Fecha Entrega:	2010-01-10
Hora Entrega:	16:12:10
Fecha Máxima Entrega:	2010-01-11
Nombre Archivo:	EjemploProfesor.java
Descripción:	La respuesta de la tarea
Integrantes:	- pepe rivera - Sergio Yuquilima
Calificación:	100.0
Mensajes:	

Figura H.38: Responder Tarea – Resultados de la respuesta.

Si damos clic en Ver más Tareas (Ver figura H.38), vamos al listado de Tareas donde veremos que la tarea que acabamos de responder ya aparece con su respectiva nota (Ver figura H.39). La respuesta la podemos consultar o modificar mientras nos encontremos en el lapso de tiempo permitido.

Listado de Tareas Publicadas y respuestas subidas.

Tareas Publicadas

Fecha Entrega	Titulo Tarea	Grupo	Respuesta	Calificacion	Modificar
2009-08-28	OPERACIONES CSHARP	Grupos	2009-08-05	75.00	Modificar
2010-01-14	Operaciones con Arreglos	Grupos	Responder	---	---
2010-01-08	Operaciones con Arreglos	---	Responder	---	---
2010-01-11	suma de numeros pares menores a	Grupos	2010-01-10	100.00	Modificar

Página: 2 de 2

Figura H.39: Responder Tarea – Listado de Tareas.

La opción de poderla modificar es para que el alumno pueda mejorar su nota. No importa si ya tiene la calificación, igualmente el sistema permitirá modificar la respuesta.

22. Consulta de Respuesta - Estudiante

En esta opción el estudiante puede consultar la respuesta de una tarea que haya respondido. Se mostrará la pantalla con el resultado obtenido y un hipervínculo para poder descargar la respuesta (Ver figura H.40).

Datos principales de la Tarea (Respuesta).

Datos de la respuesta

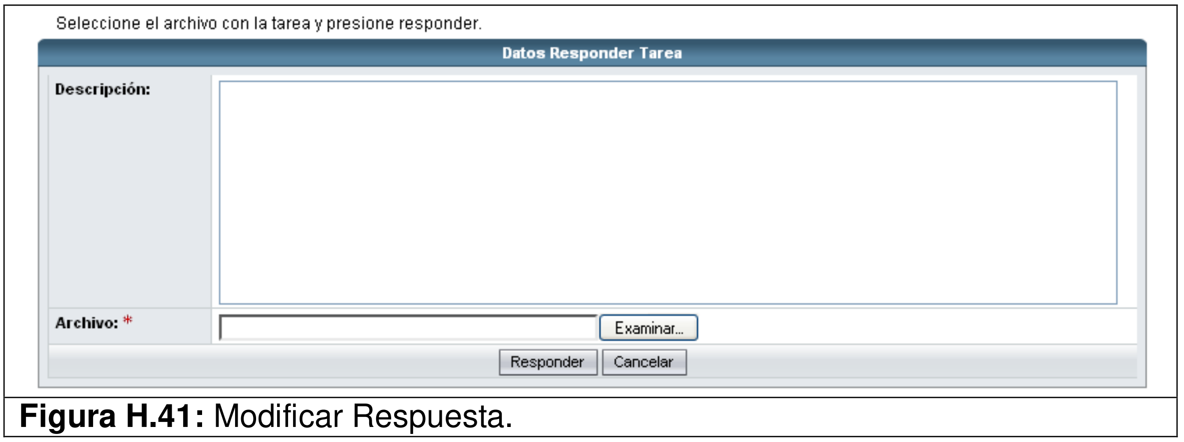
Fecha Respuesta:	2010-01-10
Hora Respuesta:	16:12:10
Fecha Máxima entrega:	2010-01-11
Nombre Archivo:	EjemploProfesor.java
Descripción:	La respuesta de la tarea
Integrantes:	- pepe rivera Par.001 - Sergio Yuquilima Par.001
Calificación:	100.00
Mensajes:	

Figura H.40: Consultar Respuesta.

Si la tarea es de carácter grupal, aparecerán los integrantes del grupo.
Si presionamos Regresar estaremos de vuelta en el Listado de Tareas.

23. Modificar Respuesta – Estudiante

Para poder modificar la respuesta de una tarea, del listado de tareas se debe escoger el hipervínculo de columna Modificar para modificar la respuesta. Lo que hace que se cargué la misma pantalla de Respuesta Tarea. (Ver figura H.41)



Seleccione el archivo con la tarea y presione responder.

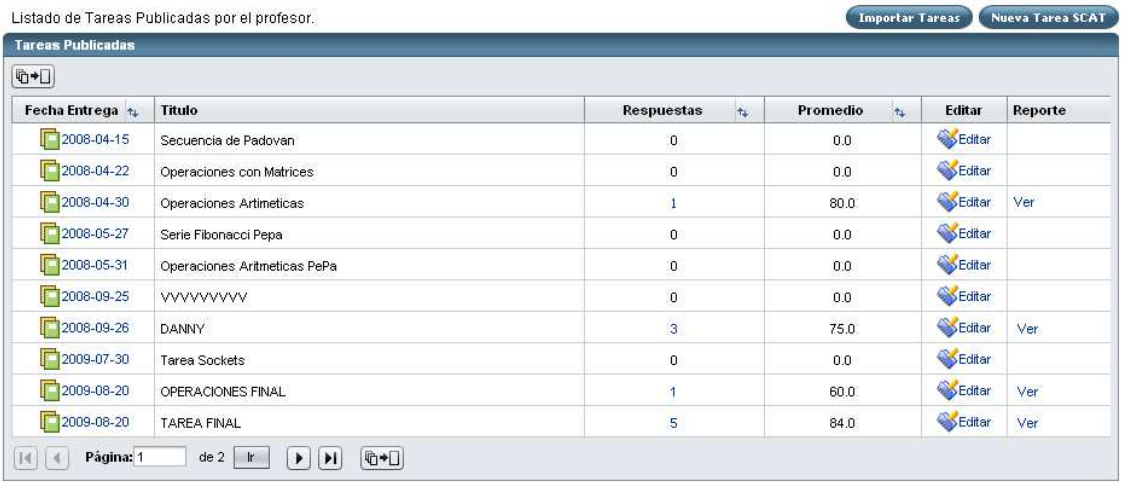
Figura H.41: Modificar Respuesta.

El link estará habilitado si ha respondido la tarea con anterioridad y se encuentra dentro la fecha tope de entrega para poder modificar la respuesta y mejorar la nota.

La modificación se la realiza de la misma forma en que se respondió la tarea por primera vez. Se mostrará luego los nuevos resultados para ver si hubo una mejoría o un empeoramiento. Una vez que se haya modificado la respuesta ya no se podrá volver a la respuesta anterior, es decir la respuesta original es totalmente sobrescrita por la nueva respuesta que se acaba de subir.

24. Consulta de Respuestas – Profesor

El profesor puede consultar la respuesta de sus alumnos. Para esto debe escoger el paralelo al cual piensa revisar las notas para que aparezca el Listado de Tareas. Si existe alguna respuesta se mostrará en la columna Respuestas un hipervínculo con el número de respuestas obtenidas hasta el momento. (Ver figura H.42)



Listado de Tareas Publicadas por el profesor.

Fecha Entrega	Titulo	Respuestas	Promedio	Editar	Reporte
2008-04-15	Secuencia de Padovan	0	0.0	Editar	
2008-04-22	Operaciones con Matrices	0	0.0	Editar	
2008-04-30	Operaciones Artimeticas	1	80.0	Editar	Ver
2008-05-27	Serie Fibonacci Pepa	0	0.0	Editar	
2008-05-31	Operaciones Aritmeticas PePa	0	0.0	Editar	
2008-09-25	VVVVVVVVV	0	0.0	Editar	
2008-09-26	DANNY	3	75.0	Editar	Ver
2009-07-30	Tarea Sockets	0	0.0	Editar	
2009-08-20	OPERACIONES FINAL	1	60.0	Editar	Ver
2009-08-20	TAREA FINAL	5	84.0	Editar	Ver

Figura H.42: Consulta de Respuestas – Listado de Tareas.

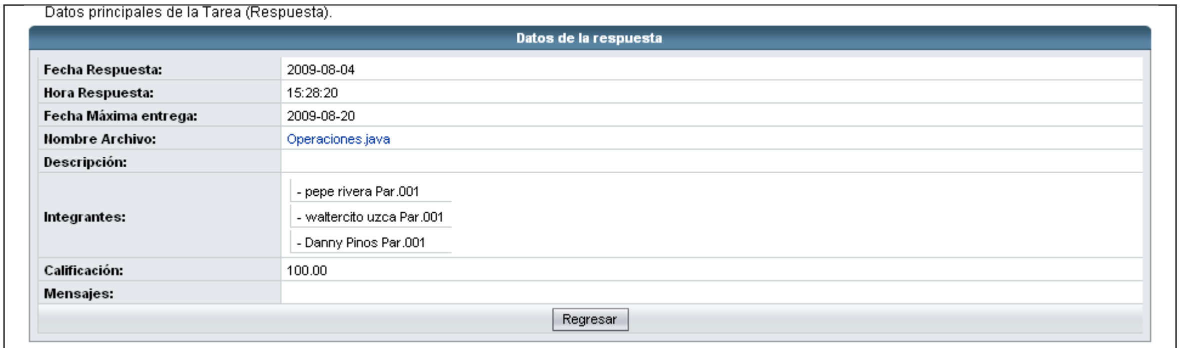
Si damos clic en el link podemos ver los alumnos que han respondido con su respectiva calificación. Además un hipervínculo en la columna de Detalle para poder revisar la respuesta del alumno.

Tarea: TAREA FINAL

Alumno	Calificacion	Detalle
pepe rivera	100.00	Ver Respuesta
Danny Pinos	100.00	Ver Respuesta
waltercito uzca	100.00	Ver Respuesta
Carlos Barcos	60.00	Ver Respuesta
Sergio Yuquilima	60.00	Ver Respuesta

Figura H.43: Consulta de Respuestas – Respuesta de Alumnos.

Si damos clic en Ver Respuesta. Podremos ver el detalle de la respuesta y además saber si es una respuesta grupal o individual.



Datos principales de la Tarea (Respuesta).

Datos de la respuesta

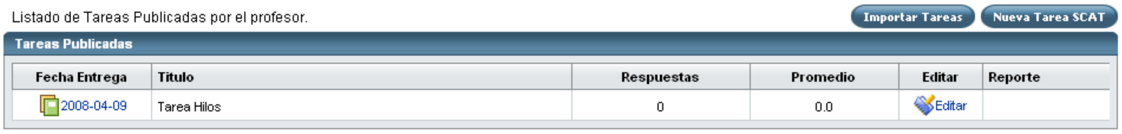
Fecha Respuesta:	2009-08-04
Hora Respuesta:	15:28:20
Fecha Máxima entrega:	2009-08-20
Nombre Archivo:	Operaciones.java
Descripción:	
Integrantes:	- pepe rivera Par.001 - waltercito uzca Par.001 - Danny Pinos Par.001
Calificación:	100.00
Mensajes:	

Figura H.44: Consulta de Respuestas – Detalle de Respuesta.

Aquí el profesor puede ver los integrantes del grupo si son del mismo paralelo o paralelos diferentes. Así mismo puede descargar el archivo con la respuesta del estudiante y verificar el código.

25. Importar Tareas – Profesor

El profesor puede importar tareas, ya sean tareas anteriores que el haya publicado o tareas que otros profesores le hayan compartido. Para esto debe escoger el paralelo al cual piensa publicar la tarea importada para que aparezca el Listado de Tareas. Debemos luego dar clic sobre el botón de Importar Tarea (Ver figura H.45).



Listado de Tareas Publicadas por el profesor.

Tareas Publicadas

Fecha Entrega	Titulo	Respuestas	Promedio	Editar	Reporte
2008-04-09	Tarea Hilos	0	0.0	Editar	

Figura H.45: Importar Tarea.

Se mostrará un listado de las tareas anteriores, con un cuadro de búsqueda para consultar tareas de forma más específica. Así mismo al final de la página podemos dar clic sobre el botón de Tareas Compartidas para poder importar las tareas que fueron compartidas por otros profesores. Para importarla solamente seleccionamos la tarea que deseamos importar y definimos la fecha de habilitación y la fecha de entrega para que se habilite el botón de Importar (Ver figura H.46).

Año: * 2006
Término: * Término II
Materia: * Sistemas Distribuidos
Paralelo: * 006

Fecha de Habilitación: *
Fecha de Entrega: *

	Tarea	Materia	Paralelo	Año	Término
○	Tarea Sockets	Sistemas Distribuidos	006	2006	2
○	Operaciones con Arreglos	Sistemas Distribuidos	006	2006	2

Figura H.46: Importar Tarea – Tareas Anteriores.

Si damos clic en el botón Tareas Compartidas se muestra otra pantalla con todas las tareas que me han compartido los demás colegas profesores. Para importarla solamente seleccionamos la tarea que deseamos importar y definimos la fecha de habilitación y la fecha de entrega para que se habilite el botón de Importar

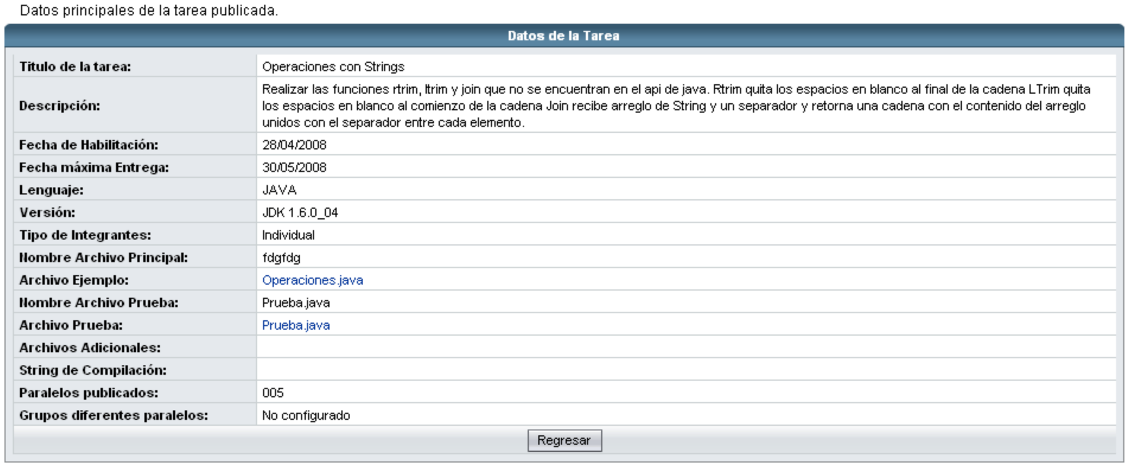
Año: * 2007
Término: * Término II
Materia: * Paradigmas Orientado a Objetos
Paralelo: * 005

Fecha de Habilitación: *
Fecha de Entrega: *

	Tarea	profesor Dueño	Materia	Paralelo	Año	Término
○	Operaciones con Strings	VERÓNICA UQUILLAS	Paradigmas Orientado a Objetos	005	2007	2

Figura H.47: Importar Tarea – Tareas Compartidas.

Se puede escoger el link de la columna tarea para ver el detalle de la tarea (Ver figura H.48).



Datos principales de la tarea publicada.

Datos de la Tarea

Titulo de la tarea:	Operaciones con Strings
Descripción:	Realizar las funciones rtrim, ltrim y join que no se encuentran en el api de java. Rtrim quita los espacios en blanco al final de la cadena LTrim quita los espacios en blanco al comienzo de la cadena Join recibe arreglo de String y un separador y retorna una cadena con el contenido del arreglo unidos con el separador entre cada elemento.
Fecha de Habilitación:	28/04/2008
Fecha máxima Entrega:	30/05/2008
Lenguaje:	JAVA
Versión:	JDK 1.6.0_04
Tipo de Integrantes:	Individual
Nombre Archivo Principal:	fdgfdg
Archivo Ejemplo:	Operaciones.java
Nombre Archivo Prueba:	Prueba.java
Archivo Prueba:	Prueba.java
Archivos Adicionales:	
String de Compilación:	
Paralelos publicados:	005
Grupos diferentes paralelos:	No configurado

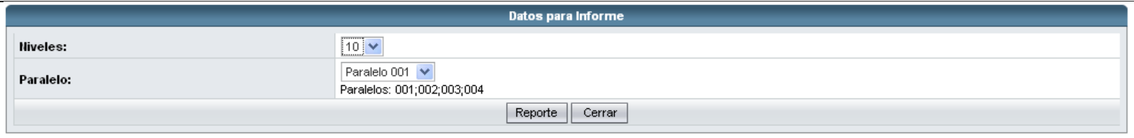
Figura H.48: Importar Tarea – Consultar Tarea.

Luego de escoger Importar, la tarea aparecerá dentro del listado de tareas.

26. Ver Reporte o Estadística

Para poder ver el reporte primero debemos seleccionar el paralelo del menú Materias. Luego se muestra el listado de tareas; debemos escoger el hipervínculo de Ver en la columna Reporte para poder visualizar un informe estadístico.

Se abrirá otra ventana especificando los niveles y si se quiere tomar en cuenta solamente el paralelo de consulta o todos los paralelos a los cuales se les ha puesto la misma tarea (Ver figura H.49)



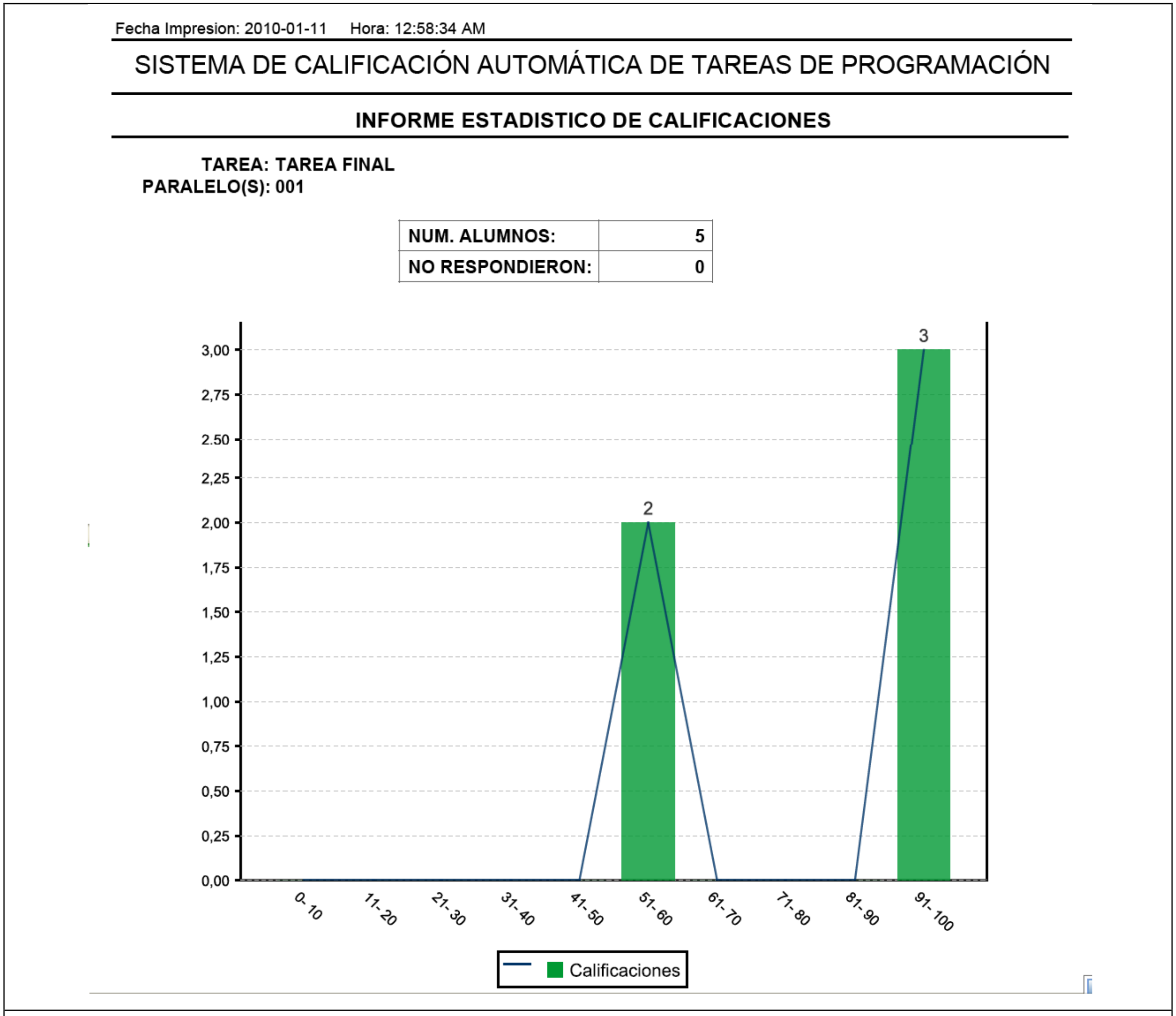
Datos para Informe

Niveles:	10
Paralelo:	Paralelo 001 Paralelos: 001;002;003;004

Figura H.49: Ver Reporte Estadístico - Consultar.

Seleccionamos las opciones de Niveles y Paralelo y damos clic en el botón Reporte para visualizar el reporte (Ver figura H.50). Si se arrepiente puede dar clic en el botón Cerrar para cancelar la consulta.

Para poder ver el reporte hay que tener instalado la JRE o no se podrá ver. El reporte será mostrado en formato pdf para que pueda ser guardado desde su visor de documentos pdf.



Fecha Impresion: 2010-01-11 Hora: 12:58:34 AM

SISTEMA DE CALIFICACIÓN AUTOMÁTICA DE TAREAS DE PROGRAMACIÓN

INFORME ESTADISTICO DE CALIFICACIONES

TAREA: TAREA FINAL
PARALELO(S): 001

NUM. ALUMNOS:	**5**
NO RESPONDIERON:	**0**

Figura H.50: Ver Reporte Estadístico

El objetivo de este reporte es darle una idea al profesor que tan bien están desarrollando las tareas sus alumnos.

1 REFERENCIAS BIBLIOGRÁFICAS

2

---

[1] Gustavo Donoso – Modelo de desarrollo Evolutivo.
http://html.rincondelvago.com/modelos-de-desarrollo-de-software.html
Fecha de último acceso: 6 de febrero de 2010.

[2] Darie, C., Chereches-Tosa, F., Brinzarea, B., Bucica, M. (2006).
Building Responsive Web Applications. Packt Publishing.

[3] Libros Web – Introducción a AJAX.
http://librosweb.es/ajax/capitulo1.html
Fecha de último acceso: 6 de febrero de 2010.

[4] Garret J. – Ajax un nuevo acercamiento a Aplicaciones Web.
http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php
Fecha de último acceso: 6 de febrero de 2010.

[5] Wikipedia la enciclopedia libre – JSF.
http://es.wikipedia.org/wiki/JavaServer_Faces
Fecha de último acceso: 5 de febrero de 2010.

[6] JavaServer Faces.
http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=IntroduccionJSFJava.
Fecha de último acceso: 6 de febrero de 2010.

[7] Wikipedia la enciclopedia libre – CSS.
http://es.wikipedia.org/wiki/Hojas_de_estilo_en_cascada
Fecha de último acceso: 6 de febrero de 2010.

[8] Wikipedia la enciclopedia libre – Servidor Apache.
http://es.wikipedia.org/wiki/Servidor_HTTP_Apache
Fecha de último acceso: 6 de febrero de 2010.

---

[9] Wikipedia la enciclopedia libre – Apache Tomcat.

http://es.wikipedia.org/wiki/Tomcat

Fecha de último acceso: 6 de febrero de 2010.

[10] Cristóbal Vásquez - MyGnet – Introducción a JasperReports e Ireport.

http://www.mygnet.net/articulos/java/301/

Fecha de último acceso: 6 de febrero de 2010.

[11] Wikipedia la enciclopedia libre – Programación por capas.

http://es.wikipedia.org/wiki/Programación_por_capas

Fecha de último acceso: 6 de febrero de 2010.

[12] OOP – Polimorfismo.

http://es.kioskea.net/contents/poo/polymorp.php3

Fecha de último acceso: 6 de febrero de 2010.

[13] Carlos Andrés Arévalo – slideshare Conceptos – XML.

http://www.slideshare.net/mallita/conceptos-presentation

Fecha de último acceso: 6 de febrero de 2010.

[14] Wikipedia la enciclopedia libre – AJAX.

http://es.wikipedia.org/wiki/AJAX

Fecha de último acceso: 6 de febrero de 2010.

[15] Ramiro Lago - Introducción a JSF.

http://www.proactiva-calidad.com/java/jsf/introduccion.html

Fecha de último acceso: 6 de febrero de 2010.

[16] Wikipedia la enciclopedia libre – MVC.

http://es.wikipedia.org/wiki/Modelo_Vista_Controlador.

Fecha de último acceso: 6 de febrero de 2010.

[17] Wikipedia la enciclopedia libre – Classpath.

http://es.wikipedia.org/wiki/Classpath_(Java)

Fecha de último acceso: 6 de febrero de 2010.