

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL



Facultad de Ingeniería en Electricidad y Computación

“ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA ACADÉMICO SIGUIENDO LOS REQUERIMIENTOS DE LA LOEI PARA EL COLEGIO POLITÉCNICO”

EXAMEN DE GRADO (COMPLEXIVO)

Previa a la obtención del grado de:

INGENIERO EN COMPUTACIÓN ESPECIALIZACIÓN SISTEMAS TECNOLÓGICOS

ENRIQUE GUSTAVO GUEVARA CAIZAPANTA

GUAYAQUIL – ECUADOR

AÑO: 2015

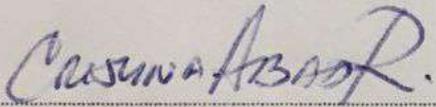
AGRADECIMIENTO

Mis más sinceros agradecimientos a Dios por haber permitido culminar esta etapa, a mi querida madre por el apoyo incondicional en mis estudios y haber soportado largas noches de trabajo, a mi Jefa Msc. Yidda Marcial por haber confiado en mi trabajo y darme la oportunidad de realizar este proyecto académico, a Msc. Dolores Zambrano por su apoyo y de manera especial al CTI – ESPOL por todos los conocimientos adquiridos durante mi formación académica.

DEDICATORIA

El presente proyecto lo dedico a mi madre, mi esposa e hijos quienes han esperado la culminación de esta etapa.

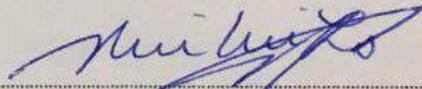
TRIBUNAL DE SUSTENTACIÓN



Nombre del Profesor

Dra. Cristina Abad R.

POR LA SUBDECANA DE LA FIEC



Nombre del Profesor

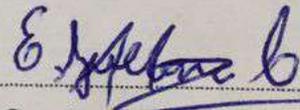
Dra. Katherine Chiluiza G.

POR LA SUBDECANA DE LA FIEC

DECLARACIÓN EXPRESA

"La responsabilidad por los hechos, ideas y doctrinas expuestas en este Informe me corresponde exclusivamente; y, el patrimonio intelectual de la misma, a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL"

(Reglamento de Graduación de la ESPOL).



Enrique Gustavo Guevara Caizapanta

RESUMEN

Debido a los últimos cambios realizados al sistema de educación ecuatoriana, las instituciones públicas y privadas de nivel medio han realizado reestructuraciones en sus políticas y procedimientos. Uno de los cambios radicales se ha enfocado en el cambio del periodo escolar, que actualmente se da por quimestres y anteriormente por trimestres.

La institución donde se realizó este proyecto de graduación ha ejecutado los cambios estipulados por el gobierno a través de un sistema informático moderno e integrado, en línea, para lo cual se ha utilizado herramientas de libre distribución, como modelo se usó el de desarrollo incremental y en la desarrollo del software el modelo en tres capas MVC.

La implementación del mismo se realizó hace dos años, tiempo que ha servido para probar la eficacia, eficiencia y buen funcionamiento del mismo, de tal manera que a futuro tenga opción a ampliar su estructura para cubrir las demandas de crecimiento de la institución y las reformas educativas gubernamentales.

Actualmente el sistema soporta alrededor de cien usuarios, albergando mil estudiantes y cien docentes.

ÍNDICE GENERAL

AGRADECIMIENTO	ii
DEDICATORIA	iii
TRIBUNAL DE SUSTENTACIÓN	iv
DECLARACIÓN EXPRESA	v
RESUMEN.....	vi
ÍNDICE GENERAL	vii
INTRODUCCIÓN.....	viii
CAPÍTULO 1.....	1
1.METODOLOGÍA O SOLUCIÓN TECNOLÓGICA IMPLEMENTADA	1
1.1 Marco Teórico.	1
1.1.1 Modelo de desarrollo incremental	1
1.1.2 Modelo de tres capas MVC	2
1.1.3 Grails	4
1.1.4 Base de datos	6
CAPÍTULO 2.....	9
2.RESULTADOS OBTENIDOS.....	9
2.1 Hardware	9
2.2 Software	10
2.3 Modulo implementados	10
CONCLUSIONES Y RECOMENDACIONES	13
BIBLIOGRAFÍA.....	15
ANEXOS	16

INTRODUCCIÓN

Los cambios que se vienen dando en el sistema educativo ecuatoriano demandan de las instituciones el mejoramiento de la calidad de educación que brinda a la sociedad. Este avance requiere de una adecuada gestión y administración de los procesos de académicos que se desarrollan, siendo imperativo manejarlos en forma automática e integrada con la labor realizada por los docentes.

En la institución donde se desarrolló el sistema, objeto del proyecto de graduación, estos procesos se manejan de forma automática y requieren de actualizaciones constantes de acuerdo a las políticas establecidas por la Ley Orgánica de Educación Intercultural (LOEI) y por las exigencias académicas del Bachillerato Internacional, por tal razón surge la necesidad de actualizar el sistema académico que actualmente soporta dichas actividades en esta institución.

La institución en mención contaba con un sistema automatizado de gestión académica denominado “Visual School”, el mismo que estaba desarrollado mediante el lenguaje de programación Visual Fox Pro. Este permitía administrar los periodos lectivos, pensum académico, gestión de estudiantes, gestión de docentes, manejo de calificaciones parciales y acumulativas entre otros. En el año 2012 el Gobierno del Ecuador decidió reformar las leyes educativas y reestructuró los períodos de clase que eran trimestrales a periodos semestrales, como lo indica el Registro Oficial (RO) N° 754 del 26 de julio de 2012 el título V capítulo I artículo 146:

“Art. 146.- Año lectivo. El año lectivo se debe desarrollar en un régimen escolar de dos (2) quimestres en todas las instituciones educativas públicas, fisco misionales y particulares, y debe tener una duración mínima de doscientos (200) días de

asistencia obligatoria de los estudiantes para el cumplimiento de actividades educativas, contados desde el primer día de clases hasta la finalización de los exámenes del segundo quimestre.”

Las autoridades del colegio Politécnico, en aras de cumplir con la ley, realizó una auditoría de procesos a su sistema informático, encontrando que, aunque cumplía con alguna de las nuevas exigencias, su estructura no le permitía implementar otros procesos relevantes para integrar la gestión administrativa con la gestión académica, de tal manera que se mantenga la información actualizada y compartida con la comunidad copolina. Otra limitante del sistema era la centralización de la información, pues todos los procedimientos se concentran en el Departamento de Secretaría General provocando saturación de trabajo en esta área.

Con estos antecedentes, la institución educativa decidió reemplazar completamente su actual sistema por uno que cumpla con las necesidades especificadas que exige la Ley de Educación. Además, se deseaba que tenga una interfaz web para que los usuarios administrativos y docentes pudieran acceder desde cualquier sitio donde se tenga acceso a internet, para descentralizar los procesos y mantener información confiable y a tiempo.

Realizar estos cambios permitiría a la institución educativa cumplir con lo establecido por la ley, solucionar el problema de centralización de procesos en el área de Secretaría General y disminuir el margen de error producido por digitación. Además, permitiría a los docentes y otras áreas administrativas alimentar el sistema para mantener actualizada la información académica y administrativa.

El proyecto tuvo como objetivo proveer a la institución educativa de un sistema informático que permitiera automatizar sus procesos administrativos y académicos para cada periodo académico. El sistema informático, denominado "SIACAD", se desarrolló usando tecnologías web actuales y debía cumplir con los siguientes objetivos:

- Identificar y analizar los procesos que debe cumplir el Sistema SIACAD con la finalidad de diagnosticar, plantear y documentar procedimientos que permitieran implementarlo.
- Diseñar las interfaces gráficas del Sistema SIACAD junto con sus respectivas entidades y relaciones de la base de datos.
- Establecer los servicios que requerían tanto los usuarios administrativos como los docentes para determinar el nivel de restricciones de operación en el desarrollo del sistema SIACAD.
- Desarrollar un sistema informático basado en la WEB que cumpla con las exigencias de la LOEI y de la institución educativa.
- Ofrecer una herramienta que ayude eficazmente al personal de la institución educativa en el ingreso y visualización de la información.

CAPÍTULO 1

1. METODOLOGÍA O SOLUCIÓN TECNOLÓGICA IMPLEMENTADA

El proyecto profesional utilizó el modelo de desarrollo incremental y para el desarrollo de software se utilizó el modelo de tres capas MVC (Modelo, Vista, Controlador) usando como framework Grails 1.3.7 y base de datos Postgresql 9.0. En algunos módulos específicos, como seguridad, impresión y reportes se, utilizó plugins, entre los cuales tenemos para seguridad: spring security core, reportes: pdf.

1.1 Marco Teórico.

1.1.1 Modelo de desarrollo incremental

Concretamente, el desarrollo incremental se basa en la idea de diseñar una implementación inicial, exponer ésta al comentario del usuario, y

luego desarrollarla en sus diversas versiones hasta producir un sistema adecuado (Figura 1.1). Las actividades de especificación, desarrollo y validación están entrelazadas en vez de separadas, con rápida retroalimentación a través de las actividades [1].

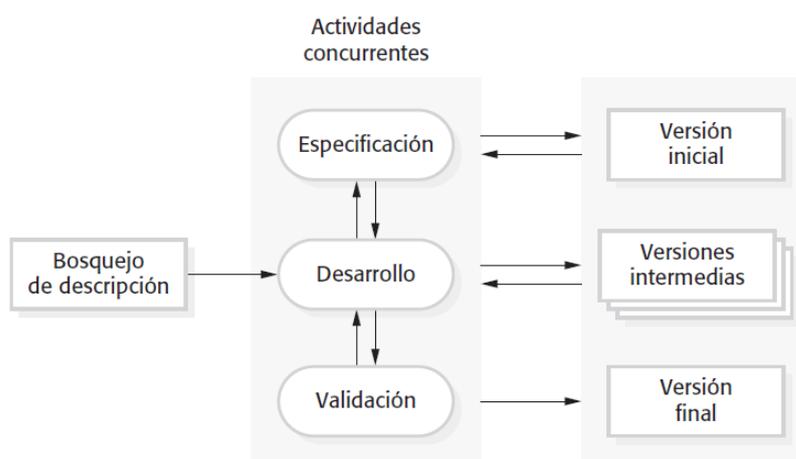


Figura 1.1: Modelo de desarrollo incremental [1].

1.1.2 Modelo de tres capas MVC

Este patrón fue descrito por primera vez por Trygve Reenskaug en 1979, y la implementación original fue realizada en Smalltalk en los laboratorios Xerox. MVC se basa en la separación de la aplicación en tres capas principales: Modelo, Vista y Controlador. Se usa en la gran mayoría de las interfaces de usuario [2], observar Figura 1.2.

Modelo: es la representación específica del dominio de la información sobre la cual funciona la aplicación. El modelo es otra forma de llamar a la capa de dominio, la lógica de dominio añade significado a los datos.

Vista: Se presenta el modelo en un formato adecuado para interactuar, usualmente un elemento de interfaz de usuario.

Controlador: Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

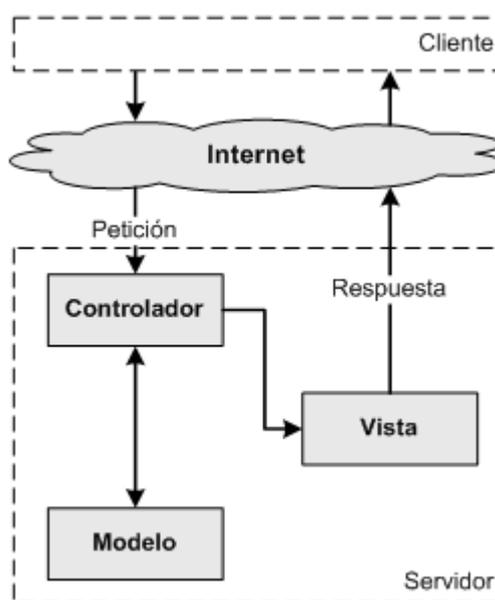


Figura 1.2: Modelo MVC.

A continuación se muestra el flujo de trabajo característico en un esquema MVC.

El usuario realiza una solicitud al sitio web. Generalmente estará desencadenada por acceder a una página del sitio. Esa solicitud le llega al controlador.

El controlador comunica tanto con modelos como con vistas. A los modelos les solicita datos o les manda realizar actualizaciones de los

datos. A las vistas les solicita la salida correspondiente, una vez se hayan realizado las operaciones pertinentes según la lógica del negocio.

Para producir la salida, en ocasiones las vistas pueden solicitar más información a los modelos. En ocasiones, el controlador será el responsable de solicitar todos los datos a los modelos y de enviarlos a las vistas, haciendo de puente entre unos y otros. Sería corriente tanto una cosa como la otra, todo depende de nuestra implementación; por eso esa flecha la hemos coloreado de otro color.

Las vistas envían al usuario la salida. Aunque en ocasiones esa salida puede ir de vuelta al controlador y sería éste el que hace el envío al cliente.

1.1.3 Grails

Grails Framework fue creado por Graeme Rocher en el año 2006 como una respuesta a la necesidad de agilizar, automatizar y simplificar el desarrollo de aplicaciones Web y está basado en el principio “mejoremos la rueda, no la reinventemos” [3].

Grails está basado en los frameworks Hibernate y Spring, que llevan mucho tiempo funcionando y que han sido probadas y se ha demostrado su correcto funcionamiento. Además, se utiliza el lenguaje Groovy por su capacidad de efectuar grandes cantidades de acciones en pocas líneas de código. Grails tiene como característica adoptar las mejores

prácticas de los otros frameworks para formar un marco de trabajo estable, robusto, sencillo de usar y de fácil mantenimiento [4].

Tomando como base los postulados de Smith (2009), se rescatan las siete grandes ideas que distinguen a Grails Framework de otras herramientas, estas son:

- Convención sobre configuración: Posee una estructura especial para la ubicación de cada uno de sus elementos, clases y archivos de configuración. Esto evita que el desarrollador tenga que hacer la configuración mediante XML.
- Filosofía ágil: Permite la reutilización de código para aumentar la productividad; además, es posible hacer cambios y visualizarlos en tiempo real sin necesidad de reiniciar el servidor de aplicaciones.
- Fundamentos sólidos: Se basa en las mejores herramientas de desarrollo existentes, lo que brinda un soporte sólido para el desarrollo de aplicaciones.
- Plantillas y Andamiaje: Maneja el concepto de Andamiaje para la generación automática de código para controladores y vistas basado en una serie de plantillas o templates.
- Integración con Java: Módulos desarrollados en Java pueden ser integrados a Grails de forma natural. Lo anterior, debido a que Groovy es completamente compatible con Java, permitiendo utilizar cualquier clase, API y biblioteca de Java en Grails.

- Wetware: Debido a que tiene muchos seguidores, ha incrementado el número de recursos dedicados al soporte y mantenimiento. Sitios Web, foros de discusión, libros, podcast y plugins son algunas de las herramientas disponibles.

En Grails, un plugin no difiere mucho de una aplicación. El uso de plugins es una de las características principales de Grails Framework y que permiten llevar a cabo la reutilización de códigos.

El repositorio de plugins de Grails cuenta con una amplia variedad de funcionalidades. Cuando se requiere de alguna en específico, lo primero que se debe hacer es consultar dicho repositorio para ver si ya existe algún plugin que cubra los requerimientos. La mayoría de las veces alguien más ya implementó la solución y la comparte con la comunidad de Grails mediante un plugin [5].

1.1.4 Base de datos

Según Camps (2005) es la representación integrada de los conjuntos de entidades instancia correspondientes a las diferentes entidades tipo del SI y de sus interrelaciones. Esta representación informática (o conjunto estructurado de datos) debe poder ser utilizada de forma compartida por muchos usuarios de distintos tipos. En otras palabras, una base de datos es un conjunto estructurado de datos que representa entidades y sus interrelaciones. La representación será única e integrada, a pesar de que debe permitir utilizaciones varias y simultáneas [6].

El contenido de una base de datos engloba a la información concerniente de una organización almacenada en archivos, de tal manera que los datos estén disponibles para los usuarios. Todos los datos se integran con una mínima cantidad de duplicidad, es decir eliminar la redundancia o al menos minimizarla. Los tres componentes principales de un sistema de base de datos son el hardware, el software SGBD y los datos a manejar, así como el personal encargado del manejo del sistema.

Un Sistema de Gestión de Base de Datos (SGBD) es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de una tarea específica. El sistema de gestión de la base de datos, proporcionando acceso controlado a la misma, y su objetivo primordial es facilitar un contorno que sea a la vez conveniente y eficiente para ser utilizado al extraer, almacenar y manipular información de la base de datos. Todas las peticiones de acceso eficaces y rápidas a la base, se manejan centralizadamente por medio del SGBD, por lo que este paquete funciona como interface entre los usuarios y la base de datos.

El sistema SIACAD usará como base de datos PostgreSQL.

PostgreSQL es un potente sistema de bases de datos relacional de código abierto. Tiene 15 años de desarrollo activo y una arquitectura probada que se ha ganado una importante reputación de confiabilidad, integridad de datos y corrección. Corre sobre todos los sistemas

operativos mayoritarios, incluyendo Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows. Cumple completamente con ACID, tiene soporte total para claves foráneas, operaciones de join, vistas, triggers y procedimientos almacenados (en múltiples lenguajes). Incluye la mayoría de tipos de datos de SQL92 and SQL99, incluyendo INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, y TIMESTAMP. Además soporta el almacenamiento de archivos binarios grandes, incluyendo imágenes, sonidos o video. Tiene interfaces de programación nativas para C/C++, Java, Perl, Python, Ruby, Tcl, ODBC, entre otros [7].

CAPÍTULO 2

2. RESULTADOS OBTENIDOS

Se logró diseñar, probar e implementar un Sistema Académico en línea con los requerimientos de la LOEI y la institución educativa.

El sistema automatizado SIACAD requirió de características de hardware y software que se muestran a continuación:

2.1 Hardware

El servidor cuenta con las siguientes características físicas:

- Procesador Intel Xeon 2.4 GHZ de 12 núcleos.
- Memoria RAM de 24 GB DDR3.
- Disco duro de 2 TB SAS.
- Tarjeta de Red Gigabit con 4 puertos LAN.

2.2 Software

El servidor tiene instalado el siguiente software:

- Linux red hat enterprise6.5.
- Servidor de enlace web Apache 2 acoplado con Tomcat 6.0.
- Framework Grails 1.3.7.
- Base de datos: postgresql 9.0.

2.3 Módulos implementados

Los módulos que se implementaron se detallan a continuación. En el Anexo A se presentan las pantallas de algunos de los módulos descritos.

A continuación se listan en categoría, según su ingreso al sistema.

La primera categoría es la gestión administrativa del sistema, los roles, el usuario, la asignación de los roles al usuario.

- Gestión de roles.
- Gestión de usuarios.
- Gestión de asignación de roles al usuario.

La segunda categoría es la gestión académica, donde consta de año lectivo, el nombre de las materias, los cursos con sus respectivos paralelos, la asignación de materias con respecto a curso y paralelo.

- Gestión de año lectivo:
- Gestión de materias.
- Gestión de curso.
- Gestión de paralelos.

- Gestión de asignación de pensum: relacionar paralelos, materias y docentes.

La tercera categoría es la gestión de admisiones y matriculación, donde consta el periodo de admisiones donde se ingresa todos los datos del estudiantes y se lo matricula en el respectivo paralelo para el periodo lectivo.

- Gestión de admisiones.
- Gestión de matriculaciones.
- Gestión de asignación de estudiantes al paralelo.

La cuarta categoría es la gestión de calificaciones, donde el docente ingresa las calificaciones de acuerdo al parcial y quimestre, también ingresa las notas de exámenes quimestrales y los exámenes especiales según corresponda.

- Gestión de notas por parcial y quimestre: existen dos quimestres de tres parciales cada uno.
- Gestión de notas de exámenes quimestrales.
- Gestión de notas de exámenes especiales: mejoramiento, supletorio, remedial y gracia.
- Gestión de ingreso de comportamiento.
- Gestión de apertura y cierre ingreso de notas por parcial, exámenes y exámenes especiales.

Y por último, la categoría de gestión de reportes, donde constan todos los reportes utilizados por docentes, secretaria general y reportes entregados a los representantes.

- Gestión de reportes:
 - Reportes de calificaciones parciales: borrador y oficial.
 - Reportes de sabanas de notas por paralelo, parcial y quimestral.
 - Reporte anual de notas.
 - Boletines parcial quimestral.
 - Boletines finales con resumen de cada parcial y exámenes especiales.
 - Listado de estudiantes por paralelo.
 - Listado general con datos completos de los estudiantes por curso.
 - Cuadro total anual oficial para el ministerio de educación.
 - Acta de promoción.
 - Certificados de matrícula.
 - Certificados de comportamiento.
 - Certificado de asistencia.
 - Certificado de aprovechamiento.
 - Apertura de libro de matrícula para el ministerio.

Al momento el sistema se encuentra operativo y cumple con los objetivos propuestos. El sistema soporta la gestión de cien usuarios aproximadamente y alberga al menos mil estudiantes y cien docentes, manteniendo procesos ágiles y efectivos para la producción de información requerida por la institución.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

1. Todos los objetivos planteados previos a la implementación de la solución se cumplieron.
2. El sistema implementado facilitó la gestión de calificaciones de los docentes descentralizando el departamento de Secretaria General.
3. El sistema implementado es capaz de ser incrementado a futuro con nuevos requerimientos que exija el colegio o la legislación educativa.
4. El sistema implementado ha automatizado muchos procesos que antes se los hacía manualmente, por ejemplo el ingreso de calificaciones era mediante presentación de un acta física entregada a secretaria general, ahora todo es vía Web.

Recomendaciones

1. Módulo para Padres de Familia: Este módulo permitirá la interacción entre el departamento encargado del desempeño académico de los estudiantes y los padres de familia, de tal manera que haya una comunicación inmediata y permita tomar acciones sobre las posibles dificultades que presenten los estudiantes.
2. Módulo para el DATIB: Emitirá información que permita al DATIB tomar decisiones sobre las recomendaciones y acciones que se debe considerar sobre determinados comportamientos de los estudiantes. Esto, a partir de la información del desempeño académico y el comportamiento disciplinar de los estudiantes.
3. También se realizarán las actualizaciones necesarias que demande la institución y la legislación educativa ecuatoriana.

BIBLIOGRAFÍA

- [1] Sommerville Ian, Ingeniería del Software, McGraw - Hill. 9na. Edición, 2011.
- [2] Pilataxi René, Modelo-Vista-Controlador (MVC), <http://es.slideshare.net/renepila/modelo-vista-controlador-26145817>, fecha consulta enero 2015
- [3] Brito Nacho, Manual de Desarrollo Web con Grails JavaEE como siempre debió haber sido, Imaginaworks Software Facto, 2009
- [4] Smith G. & Ledbrook P., Grails in Action, Manning Publications Co., 2009
- [5] Spring Flex S., Plugins Grails, <http://www.springsource.org/spring-flex>, 2011
- [6] Camps Rafael, Introducción a las Bases de Datos, UOC, 2005
- [7] OpenSuse, PostgreSQL, <https://es.opensuse.org/Postgresql>, fecha de consulta enero 2015

ANEXOS

Anexo A. – Pantallas del sistema SIACAD.

COPOL
Colegio Particular Politécnico

COPOL
Colegio Politécnico

Iniciar sesión
VER DOCUMENTOS COPOL
VER CIRCULARES

Iniciar sesión

Usuario:

Contraseña:

No cerrar sesión

Acceder

Pantalla 1: Ingreso al sistema

COPOL
Colegio Particular Politécnico

COPOL
Colegio Politécnico

Hola ENRIQUE GUSTAVO GUEVARA CAIZAPANTA (Salir)
VER DOCUMENTOS COPOL
VER CIRCULARES

Docente

Materias - 2014 - 2015

#	CURSO / PARALELO	MATERIA	I QUIMESTRE	II QUIMESTRE	OTROS
DÉCIMO GRADO DE EDU. GENERAL BÁSICA					
1	ESPOL	INFORMÁTICA	P1 P2 P3 EQ1	P1 P2 P3 EQ2	MEJ SUP REM GRA
2	ESPOL	CLUBES	P1 P2 P3 EQ1	P1 P2 P3 EQ2	MEJ SUP REM GRA
3	HOUSTON	CLUBES	P1 P2 P3 EQ1	P1 P2 P3 EQ2	MEJ SUP REM GRA
4	SIMÓN BOLÍVAR	CLUBES	P1 P2 P3 EQ1	P1 P2 P3 EQ2	MEJ SUP REM GRA
5	S.I.T.	CLUBES	P1 P2 P3 EQ1	P1 P2 P3 EQ2	MEJ SUP REM GRA
6	SORBONA	CLUBES	P1 P2 P3 EQ1	P1 P2 P3 EQ2	MEJ SUP REM GRA
PRIMERO DE BACHILLERATO GENERAL UNIFICADO					
1	CARDIFF	INFORMÁTICA APLICADA A LA EDUCACIÓN	P1 P2 P3 EQ1	P1 P2 P3 EQ2	MEJ SUP REM GRA
2	GEORGIA TECH	INFORMÁTICA APLICADA A LA EDUCACIÓN	P1 P2 P3 EQ1	P1 P2 P3 EQ2	MEJ SUP REM GRA
3	OREGON	INFORMÁTICA APLICADA A LA EDUCACIÓN	P1 P2 P3 EQ1	P1 P2 P3 EQ2	MEJ SUP REM GRA
4	PARÍS VII	INFORMÁTICA APLICADA A LA EDUCACIÓN	P1 P2 P3 EQ1	P1 P2 P3 EQ2	MEJ SUP REM GRA
5	SAN MARCOS	INFORMÁTICA APLICADA A LA EDUCACIÓN	P1 P2 P3 EQ1	P1 P2 P3 EQ2	MEJ SUP REM GRA
TERCERO DE BACHILLERATO GENERAL UNIFICADO					
1	HARVARD	INFORMÁTICA	P1 P2 P3 EQ1	P1 P2 P3 EQ2	MEJ SUP REM GRA
2	OXFORD	INFORMÁTICA	P1 P2 P3 EQ1	P1 P2 P3 EQ2	MEJ SUP REM GRA

Pantalla 2: Vista de curso y paralelo

ESCALA CUALITATIVA	ESCALA CUANTITATIVA	FRECUENCIA	PORCENTAJE
Domina los aprendizajes requeridos	9.00 - 10.00	2	7.41
Alcanza los aprendizajes requeridos	7.00 - 8.99	9	33.33
Está próximo a alcanzar los aprendizajes requeridos	4.01 - 6.99	16	59.26
No alcanza los aprendizajes requeridos	<=4	0	0.00
TOTAL		27	100%
PROMEDIO		6.85	

Pantalla 5: Tabla de gráfico distribución porcentual

ACTA DE SUPLETORIOS TOTAL ANUAL
 AÑO LECTIVO: 2014 - 2015
 CURSO: DÉCIMO GRADO DE EDU. GENERAL BÁSICA
 PARALELO: [REDACTED] ASIGNATURA: INFORMÁTICA

Acta Borrador | Acta Oficial | Acta de Asentamiento

NUM	NOMBRES	PROM. I Q	PROM. II Q	PROM. FINAL	SUPLE/100	SUPLE/10	TOTAL FINAL
1	[REDACTED]	06,54	08,51	07,52			NO SUPLETORIO
2	[REDACTED]	09,66	09,40	09,53			NO SUPLETORIO
3	[REDACTED]	06,93	07,80	07,36			NO SUPLETORIO
4	[REDACTED]	06,19					NO SUPLETORIO
5	[REDACTED]	08,30	08,24	08,27			NO SUPLETORIO
6	[REDACTED]	07,59					NO SUPLETORIO
7	[REDACTED]	05,86	08,16	07,01			NO SUPLETORIO
8	[REDACTED]	08,94	09,78	09,36			NO SUPLETORIO
9	[REDACTED]	06,67	08,66	07,66			NO SUPLETORIO
10	[REDACTED]	05,99	08,29	07,14			NO SUPLETORIO
11	[REDACTED]	08,56	08,46	08,51			NO SUPLETORIO

Pantalla 6: Acta de verificación de supletorios