



UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E INFORMÁTICOS

Tema:

“Sistema académico web utilizando tecnología móvil para el Colegio a Distancia Stephen Hawking de la ciudad de Ambato”.

Trabajo de Graduación. Modalidad: TEMI. Trabajo Estructurado de Manera Independiente, presentado previo la obtención del título de Ingeniero en Sistemas Computacionales e Informáticos.

Sublínea de Investigación: Aplicaciones Web, Aplicaciones para dispositivos móviles.

AUTOR: Tannia Mariela Changoluisa Velasco

TUTOR: Ing. Mg. Kléver Renato Urvina Barrionuevo.

Ambato - Ecuador
Abril/2015

APROBACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de investigación sobre el tema “**Sistema académico web utilizando tecnología móvil para el Colegio a Distancia Stephen Hawking de la ciudad de Ambato.**”, de la señorita Tannia Mariela Changoluisa Velasco, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los tramites y consiguiente aprobación de conformidad el Art. 16 del Capítulo II, del Reglamento de Graduación para obtener el título terminal de tercer nivel de la Universidad Técnica de Ambato.

Ambato, abril 24 del 2015

EL TUTOR

.....

Ing. Mg. Kléver Renato Urvina Barrionuevo

AUTORÍA

El presente trabajo de investigación titulado “**Sistema académico web utilizando tecnología móvil para el Colegio a Distancia Stephen Hawking de la ciudad de Ambato**”. Es absolutamente original, autentico y personal en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, abril 24 del 2015

.....
Tannia Mariela Changoluisa Velasco

C.C.: 1803731148

APROBACIÓN DE LA COMISIÓN CALIFICADORA

La Comisión Calificadora del presente trabajo conformada por los señores docentes Ing. Edison Álvarez e Ing. Franklin Mayorga, revisó y aprobó el Informe Final del trabajo de graduación titulado **“Sistema académico web utilizando tecnología móvil para el Colegio a Distancia Stephen Hawking de la ciudad de Ambato”**., presentado por la señorita Tannia Mariela Changoluisa Velasco de acuerdo al Art. 17 del Reglamento de Graduación para obtener el título Terminal de tercer nivel de la Universidad Técnica de Ambato.

Ing. Vicente Morales L., Mg.
PRESIDENTE DEL TRIBUNAL

Ing. Edison Álvarez, Mg.
DOCENTE CALIFICADOR

Ing. Franklin Mayorga, Mg.
DOCENTE CALIFICADOR

DEDICATORIA

El presente trabajo está dedicado a Dios por haberme dado la vida y permitirme llegar a este momento triunfante de mi carrera, a mis padres José y Rosa por ser mi apoyo incondicional en mi formación académica, por proporcionarme ejemplo de superación y lucha constante, y sobre todo por su amor de padres.

A mis hermanas Karina y Gabriela por estar ahí en los buenos y malos momentos de mi vida académica, por sus consejos y su apoyo absoluto.

A mi esposo Edwin Chicaiza por haber llegado a mi vida en la etapa más importante de mi formación académica, ofreciéndome su apoyo y amor incondicional.

Tannia Mariela Changoluisa Velasco

AGRADECIMIENTO

Extiendo mi sincero agradecimiento a Dios y a mi familia por haber estado a mi lado en el largo trayecto de mi formación profesional.

A mis maestros por compartirme sus conocimientos y orientarme en mi vida profesional, en especial al Ing. Renato Urvina por colaboración total en la culminación de mi proyecto.

También quiero agradecer a todos los que forman parte de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato por su apoyo absoluto.

A mis amigos por su apoyo y amistad en todos estos años compartidos.

Tannia Mariela Changoluisa Velasco

ÍNDICE DE CONTENIDOS

Portada	i
Aprobación del Tutor.....	ii
Autoría	iii
Aprobación de la Comisión Calificadora.....	iv
Dedicatoria.....	iv
Agradecimiento.....	v
Índice de Contenidos	vii
Índice de Figuras.....	x
Índice de Tablas	xii
Resumen Ejecutivo	xiv
Abstract.....	xv
Glosario de Términos y Acrónimos	xvi
Introducción	xix
CAPÍTULO I	1
1. EL PROBLEMA.....	1
1.1. Tema de Investigación.....	1
1.2. Planteamiento del Problema	1
1.3. Delimitación del Problema	2
1.4. Justificación	3
1.5. Objetivos.....	4
1.5.1. Objetivo General:.....	4
1.5.2. Objetivos Específicos:	4
CAPÍTULO II.....	5
2. MARCO TEÓRICO	5
2.1. Antecedentes Investigativos	5
2.2. Fundamentación Teórica	5
2.3. Propuesta de Solución	28
CAPÍTULO III.....	29
3. METODOLOGÍA.....	29
3.1. Modalidad de la Investigación.....	29
3.2. Población y Muestra	29

3.3.	Operacionalización de Variables u Objetivos.....	30
3.4.	Recolección de Información.....	31
3.5.	Procesamiento y Análisis de Datos	31
3.6.	Desarrollo del Proyecto	34
CAPÍTULO IV		35
4.	DESARROLLO DE LA PROPUESTA	35
4.1.	Análisis	35
4.1.1.	Requerimientos del Sistema.....	35
4.1.2.	Diagramas de Casos de Uso.....	37
4.1.3.	Diagramas de Secuencia	46
4.2.	Diseño.....	53
4.2.1.	Diseño de Base de Datos.....	53
4.2.2.	Diccionario de Datos.....	54
4.2.3.	Diseño de la Interfaz de Usuario.....	59
4.3.	Implementación	70
4.3.1.	Implementación de la Interfaz de Grafica de Usuario	70
4.3.2.	Implementación de la Arquitectura de Programación.....	73
4.4.	Pruebas de Funcionamiento.....	78
4.4.1.	Pruebas de Caja Blanca.....	78
4.4.2.	Pruebas de Caja Negra	83
1.1.	Análisis	86
1.1.1.	Requerimientos del Sistema.....	86
1.1.2.	Diagramas de Casos de Uso.....	88
1.1.3.	Diagramas de Secuencia	90
1.2.	Diseño.....	92
1.2.1.	Diseño de Base de Datos.....	92
1.2.2.	Diccionario de Datos.....	93
1.2.3.	Diseño de la Interfaz de Usuario.....	97
4.3.	Implementación	98
4.3.1.	Implementación de la Interfaz de Grafica de Usuario	98
4.3.2.	Implementación de la Arquitectura de Programación.....	100
4.4.	Pruebas de Funcionamiento.....	109
4.4.1.	Pruebas de Caja Blanca.....	109
4.4.2.	Pruebas de Caja Negra	112
CAPÍTULO V.....		117
5.	CONCLUSIONES Y RECOMENDACIONES	117

5.1.	Conclusiones.....	117
5.2.	Recomendaciones	118
	Bibliografía	119
	Anexo 1: Manual de Usuario	123
	Anexo 2: Ficha de Recolección de Notas	155
	Anexo 3: Ficha de Pase de Año de los Estudiantes	156

ÍNDICE DE FIGURAS

Figura. 2.1: Arquitectura MySQL.....	12
Figura. 2.2: Diagrama Relacional MVC	14
Figura. 2.3: Colaboración entre los componentes de un MVC.....	15
Figura. 2.4: Base de Datos y MVC	17
Figura. 2.5: Modelo Iterativo e Incremental para el Ciclo de Vida del Software.....	19
Figura. 2.6: Ciclo de Vida del Modelo de Desarrollo Iterativo e Incremental	20
Figura. 2.7: Etapas Generales del Modelo de Desarrollo Iterativo e Incremental	28
Figura. 3.1: Diagrama de Flujo de la Publicación de Boletines de Notas	32
Figura. 3.2: Diagrama de Flujo del Registro de Notas	33
Figura 4.1: Diagrama de Caso de Uso Administrador.....	38
Figura 4.2: Diagrama de Caso de Uso Administrador – Usuario_Docente.....	43
Figura 4.3: Diagrama de Secuencia MVC: Ingreso al Sistema	46
Figura 4.5: Diagrama de Secuencia MVC: Configuración de Cuentas	47
Figura 4.6: Diagrama de Secuencia MVC: Crear Grupos de Usuarios	48
Figura 4.7: Diagrama de Secuencia MVC: Registrar Usuarios	49
Figura 4.8: Diagrama de Secuencia MVC: Administrar Aplicación	50
Figura 4.9: Diagrama de Secuencia MVC: Administrar Notas	51
Figura 4.4: Diagrama de Secuencia MVC: Salir de la Aplicación	52
Figura 4.10: Diseño de Base de Datos	53
Figura 4.11: Interfaz de Ingreso al Sistema	59
Figura 4.12: Interfaz de la Pagina Maestra	60
Figura 4.13: Interfaz de Ingresos Y Modificación.....	62
Figura 4.14: Interfaz de Eliminación	63
Figura 4.15: Interfaz de Información	64
Figura 4.16: Interfaz de Usuarios y Grupos.....	65
Figura 4.17: Interfaz de Administración de Cursos	66
Figura 4.18: Interfaz de Secciones.....	67
Figura 4.19: Interfaz de Áreas	68
Figura 4.20: Interfaz de Docentes.....	69
Figura 4.21: Pantalla de Inicio de Sesión	70
Figura 4.22: Pantalla Principal.....	71
Figura 4.23: Función para mostrar la Vista de los Cursos	74
Figura 4.24: Función para mostrar la Vista de los Paralelos	74
Figura 4.25: Función para mostrar la Vista de los Paralelos por Curso	75

Figura 4.26: Función para el Método Añadir un Curso	75
Figura 4.27: Función para el Método Editar un Curso	76
Figura 4.28: Función para el Método Eliminar un Curso	77
Figura 4.29: Función para el Trigger Eliminar Alumno	77
Figura 4.30: Diagrama de Flujo para la Validación de los Dígitos de una Cedula	79
Figura 4.31: Grafo de Flujo para la Validación de los Dígitos de una Cedula	82
Figura 4.32: Prueba de Caja Negra – Matricula de un Estudiante.....	85
Figura 4.33: Diagrama de Caso de Uso Administrador - Estudiante.....	88
Figura 4.34: Diagrama de Secuencia MVC: Consultar Datos	90
Figura 4.35: Diagrama de Secuencia MVC: Generar Reportes	91
Figura 4.36: Diseño de Base de Datos	92
Figura 4.37: Interfaz de Estudiantes	97
Figura 4.38: Pantalla Principal.....	98
Figura 4.39: Creación del Trigger para el Control de Notas	101
Figura 4.40: Creación del Trigger para Generar el Código del Matricula de un Alumno	101
Figura 4.41: Creación del Trigger para Generar el Código del Historial de un Alumno	102
Figura 4.42: Creación del Trigger para Quitar un Usuario Estudiante	102
Figura 4.43: Creación del Trigger para Quitar un Usuario Docente	103
Figura 4.44: Creación del Trigger para Crear un Quimestre	103
Figura 4.45: Código de Conexión con el Servidor	103
Figura 4.46: Interfaz de Entrada de Cpanel	104
Figura 4.47: Interfaz de Creación de una Nueva Base de Datos	105
Figura 4.48: Interfaz de Creación de un Subdominio	105
Figura 4.49: Interfaz de Creación de una Cuenta FTP.....	106
Figura 4.50: Interfaz del CuteFTP	106
Figura 4.51: Interfaz de las Carpetas del Server	107
Figura 4.52: Documentos de la Carpeta public_html	107
Figura 4.53: Pantalla de Verificación de Transferencia de Ficheros	108
Figura 4.54: Grafo de Flujo para la Validación de un E-mail	110
Figura 4.55: Prueba de Caja Negra – Representantes.....	114

ÍNDICE DE TABLAS

Tabla 3.1: Operacionalización de Variables u Objetivos	30
Tabla 4.1: Guía de Observación de Información	36
Tabla 4.2: Caso de Uso: Ingresar al Sistema	39
Tabla 4.3: Caso de Uso: Ingresos	39
Tabla 4.4: Caso de Uso: Actualizar	40
Tabla 4.5: Caso de Uso: Modificar	40
Tabla 4.6: Caso de Uso: Eliminar	41
Tabla 4.7: Caso de Uso: Consultar	41
Tabla 4.8: Caso de Uso: Configuración de una Cuenta	42
Tabla 4.9: Caso de Uso: Administrar Notas	44
Tabla 4.10: Caso de Uso: Consultar Datos	45
Tabla 4.11: Descripción MVC Ingreso al Sistema	46
Tabla 4.12: Descripción MVC Configuración de Cuentas	47
Tabla 4.13: Descripción MVC Crear Grupos de Usuarios	48
Tabla 4.14: Descripción MVC Registrar Usuarios	49
Tabla 4.15: Descripción MVC Administrar Aplicación	50
Tabla 4.16: Descripción MVC Administrar Notas	51
Tabla 4.17: Descripción MVC Salir de la Aplicación	52
Tabla 4.18: Descripción de la Tabla Ciclos	54
Tabla 4.19: Descripción de la Tabla Cursos	54
Tabla 4.20: Descripción de la Tabla Especialidades	54
Tabla 4.21: Descripción de la Tabla Secciones	55
Tabla 4.22: Descripción de la Tabla Paralelos	55
Tabla 4.23: Descripción de la Tabla Periodos	55
Tabla 4.24: Descripción de la Tabla Materias	56
Tabla 4.25: Descripción de la Tabla Alumnos.....	56
Tabla 4.26: Descripción de la Tabla Comportamientos.....	56
Tabla 4.27: Descripción de la Tabla Matriculas	57
Tabla 4.28: Descripción de la Tabla Docente	57
Tabla 4.29: Descripción de la Tabla Notas	58
Tabla 4.30: Descripción de la Tabla Quimestres	58
Tabla 4.31: Pruebas de Caja Blanca de la Validación de los Dígitos de una Cedula	80
Tabla 4.32: Pruebas de Caja Negra - Matricula de un Estudiante	84
Tabla 4.33: Guía de Observación de Información	86

Tabla 4.34: Caso de Uso: Consultar Datos	89
Tabla 4.35: Descripción MVC Consultar Datos	90
Tabla 4.36: Descripción MVC Generar Reportes	91
Tabla 4.37: Descripción de la Tabla Secretarias	93
Tabla 4.38: Descripción de la Tabla Representantes	93
Tabla 4.39: Descripción de la Tabla Áreas	94
Tabla 4.40: Descripción de la Tabla NotaGrados	94
Tabla 4.41: Descripción de la Tabla Groups	94
Tabla 4.42: Descripción de la Tabla Users	95
Tabla 4.43: Descripción de la Tabla FotoUsers	95
Tabla 4.44: Descripción de la Tabla NotasDetalles	95
Tabla 4.45: Descripción de la Tabla DocenteParaleloMaterias	96
Tabla 4.46: Prueba de Caja Blanca de la Validación de un E-mail	110
Tabla 4.47 Pruebas de Caja Negra - Representantes	113

RESUMEN EJECUTIVO

El personal administrativo del Colegio a Distancia Stephen Hawking de la Ciudad de Ambato año tras año se encarga del control académico de la información de los estudiantes, con el propósito de mejorar el acceso a la misma.

La demanda de estudiantes que desean inscribirse en la institución va en incremento cada ciclo lectivo, por ende se maneja gran cantidad de información, la misma que es tratada de forma manual y resguardada en archivos físicos, lo que provoca vulnerabilidad en el control de la misma, desaprovechando el uso de las TIC's.

La presente investigación propone el desarrollo de un sistema automatizado para el control de la fiabilidad de la información en la gestión administrativa y académica de la institución, el mismo que permitirá administrar, gestionar y controlar los datos de los estudiantes, agilizando los procesos y mejorado el servicio a la ciudadanía.

Mediante técnicas de observación y diálogos constantes con el personal académico de la institución se ha realizado el levantamiento de requerimientos, los cuales posteriormente han sido transformados en diagramas informáticos para facilitar la interpretación de los procesos. La etapa del desarrollo de la propuesta sigue la metodología Iterativa e Incremental, la cual está enfocada a la realización de varias iteraciones hasta lograr el producto deseado por el cliente.

En base a los diagramas y a la metodología se plantea el diseño del sistema así como los requisitos necesarios para su implementación y funcionamiento. De la misma forma se han seleccionado las herramientas para el desarrollo del mismo.

ABSTRACT

Every year, the administrative staff from the Stephen Hawking distance school from Ambato city has been in charge of the academic management about the student's information having the purpose to improve the access to it.

Every school year, the demand of the students is getting bigger, therefore there is much more information that is manipulated in a manual way and physically filed. This means that the information is vulnerable to any issue and the use of TIC's has been wasted.

The following research proposes the development of an automatic system to control the reliability about the administrative and academic school information, this will help to manage process and control the student's data by making it faster and improving the service.

Through observation and current discussions with the school academic staff, it has been done the adequate studies and information compiles to ease the process interpretation. Then, it has been turned into computer diagrams to interpret the processes. The Iterative and Incremental methodology has been used to develop this proposal, which is focused on several iterations to get the final outcome desired by the client.

Based on the diagrams and methodology, it has been planned the design of the system as well as the needed requirements for the implementation and operation. The tools have been also selected for the running itself.

GLOSARIO DE TÉRMINOS Y ACRÓNIMOS

Framework: Es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar.

HTML: (Lenguaje de Marcación de Hipertexto), es un lenguaje se utiliza comúnmente para establecer la estructura y contenido de un sitio web, tanto de texto, objetos e imágenes.

Base de Datos: Es un “almacén” que nos permite guardar grandes cantidades de datos de forma organizada, para que luego consultarlos y obtener información oportuna y fiable.

MySQL: Es un sistema cliente servidor de administración de bases de datos relacionales diseñado para el trabajo tanto en los sistemas operativos Windows como en los sistemas UNIX/LINUX. Además, determinadas sentencias de MySQL pueden ser embebidas en código PHP Y HTML para diseñar aplicaciones web dinámicas que incorporan la información de las tablas de MySQL a páginas Web. Así mismo, MySQL es compatible con el software más potente de diseño Web, como Dreamweaver MX.

UML: Unified Modeling Language o Lenguaje Unificado de Modelado, es un lenguaje gráfico para especificar, visualizar, construir y documentar los sistemas de software.

Aplicación Web: Una aplicación web (web-based application) es un tipo especial de aplicación cliente/servidor, donde tanto el cliente (el navegador, explorador o visualizador) como el servidor (el servidor web) y el protocolo mediante el que se comunican (HyperText Transfer Protocolo (HTTP)) están estandarizados y no han de ser creados por el programador de aplicaciones.

Servidor Web: Servidor web es un programa que está esperando permanentemente las solicitudes de conexión mediante el protocolo HTTP por parte de los clientes web.

Cliente Web: Cliente web es un programa con el que interacciona el usuario para solicitar a un servidor web el envío de los recursos que desea obtener mediante HTTP2.

Diagramas caso de uso: Es una técnica que resume la funcionalidad completa de un sistema, presentando las respectivas interacciones con los agentes.

Diagramas de secuencia: Muestra la secuencia de los mensajes entre objetos durante un escenario concreto.

PHP: El lenguaje PHP es un lenguaje de programación de estilo clásico, con esto quiero decir que es un lenguaje de programación con variables, sentencias condicionales, bucles, funciones. No es un lenguaje de marcas como podría ser HTML, XML o WML. Está más cercano a JavaScript o a C, para aquellos que conocen estos lenguajes.

Web Móvil: “La expresión Web móvil se refiere al acceso a la Web desde aparatos cuya principal cualidad es la movilidad. Se trata de la misma Web que consultamos desde nuestros ordenadores de sobremesa o portátiles, el único elemento diferenciador es el equipo empleado.

Dispositivos Móviles: Los dispositivos móviles En la Web móvil intervienen varios agentes, desde las redes de telecomunicaciones o los proveedores del servicio (operadores de telefonía móvil) hasta los equipos empleados para la visualización de los contenidos.

Modelo – Vista – Controlador (MVC): El modelo–vista–controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

Modelo de Desarrollo Iterativo e Incremental: Desarrollo iterativo y creciente (o incremental) es un proceso de desarrollo de software creado en respuesta a las debilidades del modelo tradicional de cascada.

Básicamente este modelo de desarrollo, que no es más que un conjunto de tareas agrupadas en pequeñas etapas repetitivas (iteraciones), es uno de los más utilizados en los últimos tiempos ya que, como se relaciona con novedosas estrategias de desarrollo de software y una programación extrema, es empleado en metodologías diversas.

INTRODUCCIÓN

El presente trabajo de tesis cuyo tema es, **“Sistema académico web utilizando tecnología móvil para el Colegio a Distancia Stephen Hawking de la ciudad de Ambato”**, consta de cinco capítulos los cuales se detallan a continuación.

Capítulo I. “El problema”, aquí se define el problema que está aconteciendo en el contexto de la realidad, para lo cual delimitamos sus alcances, justificamos debidamente el problema y planteamos los objetivos, los cuales serán la guía de todo el proyecto.

Capítulo II. “Marco Teórico”, en este capítulo consta el fundamento teórico el cual ayuda a comprender de una forma clara el problema, para posteriormente plantear una propuesta de solución al mismo.

Capítulo III. “Metodología”, se describe la metodología y las modalidades de investigación a seguir para conseguir el resultado inicialmente propuesto, así como también la forma de cómo se recolectará y se procesará la información, además de una descripción global de cómo se desarrolla el proyecto.

Capítulo IV. “Desarrollo de la Propuesta”, en este capítulo se describe el desarrollo de la propuesta de solución, definiendo los requisitos necesarios que se han obtenido del proceso de levantamiento de información y requisitos, documentando el proceso de desarrollo del software a través de casos de uso, diagramas de estado y flujo de datos, diseño de la base de datos, diseño de la interfaz gráfica de usuario, estabilización de la propuesta de solución, así como también el despliegue o implementación de la solución final.

Capítulo V. “Conclusiones y recomendaciones”, se establecen las conclusiones a las que ha llegado el investigador luego del desarrollo del proyecto, de la misma forma se hacen las recomendaciones que el investigador a considerado necesarias.

Finalmente se incluye las referencias citadas en este documento, en los anexos se incluye los instrumentos utilizados para la recolección de la información así como también se incluyen los manuales correspondientes del presente proyecto.

CAPÍTULO I

1. EL PROBLEMA

1.1. Tema de Investigación

Sistema académico web utilizando tecnología móvil para el Colegio a Distancia Stephen Hawking de la ciudad de Ambato.

1.2. Planteamiento del Problema

La tecnología es parte importante dentro de la sociedad y la educación, las instituciones educativas forman parte integral de este proceso. En el ámbito de la información y la comunicación, las instituciones utilizan sistemas y recursos informáticos para el diseño, desarrollo y difusión digitalizada de sus datos.

En Ecuador sobre todo las instituciones de Educación Superior cuentan con páginas web para la gestión de su información académica, las instituciones educativas secundarias en menor cantidad también han aprovechado las ventajas que presenta la tecnología actual; sin embargo los colegios a distancia debido a su naturaleza privada y sistema de educación distinto al tradicional, se han visto también en la necesidad de elaborar o contratar aplicaciones web y móviles, para no quedar rezagadas en el aprovechamiento de las TIC's frente a otras instituciones similares.

En la ciudad de Ambato el control académico de los estudiantes de los colegios a distancia en su mayoría se lo realiza de manera manual, ya que dicho proceso no ha sido

automatizado y la información se la resguarda en archivos físicos, presentando así todo tipo de vulnerabilidad e inseguridad, lo que genera inconvenientes tales como lentitud al realizar búsquedas de información, inadecuado respaldo de datos y errores en los informes, generando así preocupación en el área encargada del proceso académico de la institución.

El Colegio a Distancia Stephen Hawking, actualmente maneja sus datos en unos casos con archivos manuales y en otros con archivos de hojas de cálculo; lo cual ha provocado desorganización, redundancia, duplicidad y vulnerabilidad de la información; ocasionando que la asignación, registro y supervisión de los datos por parte del personal encargado del proceso académico de la institución sea agobiante y necesite de esfuerzo adicional para organizar la información.

1.3. Delimitación del Problema

Área Académica: Software

Línea de Investigación: Desarrollo de Software

Sublínea de Investigación: Aplicaciones Web, Aplicaciones para dispositivos móviles.

Delimitación Espacial: El presente trabajo de investigación se lo realizará en el Colegio a Distancia Stephen Hawking, en la ciudad de Ambato, provincia de Tungurahua. Dirección: Montalvo 05-11 y Sucre (Parque Montalvo).

Delimitación Temporal: El tiempo estimado del proyecto es 6 meses a partir de la aprobación del proyecto.

1.4. Justificación

En la actualidad el uso de internet y tecnologías móviles se ha incrementado rápidamente en las instituciones públicas y privadas, las cuales ayudan a brindar un servicio de calidad a sus clientes/usuarios, facilitando la aceptación de dicha propuesta.

Este proyecto es elemental, porque proporcionará información vía web, facilitará las actividades del registro académico de los docentes, consultas de notas por parte de los estudiantes, además se aprovechará el uso de móviles por parte de los usuarios (Personal Administrativo, Docentes y Estudiantes) del colegio para que estos tengan libre acceso a la información.

El introducir un sistema académico web utilizando tecnología móvil, facilitará la planificación y ejecución de cada uno de los procesos que intervienen en la parte académica de la institución, de esta manera reducirá la vulnerabilidad de la información, reducirá errores en los informes y reportes, que son los principales causantes de pérdida de tiempo en la obtención de información que hasta la actualidad ha tenido que enfrentar la institución.

La utilización de tecnología móvil en este proyecto es muy interesante y novedosa, ya que se empleará de manera útil la tecnología, facilitando el acceso a la información desde cualquier lugar y a cualquier hora, con el solo hecho de tener acceso a internet, además de la aplicación de herramientas de Software Libre las cuales no tienen costo.

Con la introducción de este sistema académico web la institución estará en la capacidad de facilitar información a los usuarios de la institución, controlar de manera automatizada los datos de la institución, generar reportes de forma rápida y exacta, evitando aglomeraciones en las instalaciones, como también se podrá evitar alteraciones de información por parte de terceras personas.

Este proyecto es factible ya que se cuenta con los recursos intelectuales, económicos y de información para realizar el desarrollo del mismo.

1.5. Objetivos

1.5.1. Objetivo General:

- Desarrollar un sistema académico web utilizando tecnología móvil para el Colegio a Distancia Stephen Hawking de la ciudad de Ambato.

1.5.2. Objetivos Específicos:

- Analizar e Identificar los procesos académicos que se llevan a cabo de forma manual en el Colegio a Distancia Stephen Hawking de la ciudad de Ambato.
- Establecer los requerimientos y alcance del Sistema Académico Web.
- Analizar las características del Sistema Académico Web.
- Diseñar un Sistema Académico Web utilizando tecnología móvil.
- Desarrollar un Sistema Académico Web utilizando tecnología móvil.
- Realizar pruebas de funcionamiento al Sistema Académico Web.

CAPÍTULO II

2. MARCO TEÓRICO

2.1. Antecedentes Investigativos

Para la siguiente investigación se han tomado los siguientes antecedentes investigativos:

“Al desarrollar este proyecto se tratará de brindar nuevos servicios con todas las facilidades posibles, que la tecnología permite y que coadyuven a que nuestra universidad alcance día a día la excelencia, como uno de sus objetivos estratégicos.” [1]

“Confiabilidad de los Datos: puesto que los Docentes se convierten en alimentadores directos del sistema y la información se almacena en la base de datos del SIU, está impida la posibilidad de que la información sea modificada o adulteradas por terceras personas.” [2]

“Analizar los requerimientos necesarios para el desarrollo del sistema, utilizando el estándar IEEE 830, de forma que cubra las diferentes consideraciones metodológicas y técnicas.” [3]

2.2. Fundamentación Teórica

Aplicación Web

“Una aplicación web (web-basedapplication) es un tipo especial de aplicación cliente/servidor, donde tanto el cliente (el navegador, explorador o visualizador) como

el servidor (el servidor web) y el protocolo mediante el que se comunican (HyperText Transfer Protocolo (HTTP)) están estandarizados y no han de ser creados por el programador de aplicaciones. El protocolo HTTP forma parte de la familia de protocolos de comunicaciones Transmission Control Protocol/Internet Protocol (TCP/IP), que son los empleados en Internet. Estos protocolos permiten la conexión de sistemas heterogéneos, lo que facilita el intercambio de información entre distintos ordenadores.” [4]

Servidor Web

“Servidor web es un programa que está esperando permanentemente las solicitudes de conexión mediante el protocolo HTTP por parte de los clientes web.

En los sistemas Linux y Unix suele ser un demonio y en los sistemas Microsoft Windows un servicio.

La parte servidor de las aplicaciones web está formada por:

- Páginas estáticas (documentos HTML) que siempre muestran el mismo contenido.
- Recursos adicionales (multimedia, documentos adicionales, etc.) que se pueden emplear dentro de las páginas o estar disponibles para ser descargados y ejecutados (visualizados) en el cliente.
- Programas o scripts que son ejecutados por el servidor web cuando el navegador del cliente solicita algunas páginas. La salida de este script suele ser una página HTML estándar que se envía al navegador del cliente. En algunos casos pueden acceder a bases de datos.” [4]

Cliente Web

“Cliente web es un programa con el que interacciona el usuario para solicitar a un servidor web el envío de los recursos que desea obtener mediante HTTP2. La parte cliente de las aplicaciones web suele estar formada por el código HTML que forma la página web más algo de código ejecutable realizado en lenguaje de script del navegador (Java Script o VBScript) o mediante pequeños programas (applets) realizados en Java.

También se suelen emplear plug-ins³ que permiten visualizar otros contenidos multimedia (como Macromedia Flash⁴), aunque no se encuentran tan extendidos como las tecnologías anteriores y plantean problemas de incompatibilidad entre distintas plataformas. Por tanto, la misión del cliente web es interpretar las páginas HTML y los diferentes recursos que contienen (imágenes, sonidos, etc.). Las tecnologías que se suelen emplear para programar el cliente web son:

- HTML.
- CSS.
- Lenguajes de script: Java Script, VBScript, etc.
- Distintas tecnologías que necesitan la existencia de un plug-in en el navegador:
- Adobe Acrobat Reader, Autodesk MapGuide, Live Picture Photo Vista, Macromedia
- Flash, Macromedia Shockwave, Virtual Reality Modeling Language (VRML), etc.” [5]

Web Móvil:

“La expresión Web móvil se refiere al acceso a la Web desde aparatos cuya principal cualidad es la movilidad. Se trata de la misma Web que consultamos desde nuestros ordenadores de sobremesa o portátiles, el único elemento diferenciador es el equipo empleado.” [6]

Tipos de Aplicaciones Móviles

- “1) Aplicaciones de Mensajería
- 2) Aplicaciones de Contenido.
- 3) Aplicaciones Interactivas.
- 4) Aplicaciones híbridas (p.ej. mensajería + interactiva).” [7]

Dispositivos Móviles

“Los dispositivos móviles En la Web móvil intervienen varios agentes, desde las redes de telecomunicaciones o los proveedores del servicio (operadores de telefonía móvil) hasta los equipos empleados para la visualización de los contenidos.

Estos últimos merecen una atención especial para quienes están interesados en la producción de contenidos, puesto que de sus posibilidades depende la experiencia que el usuario final pueda obtener.” [8]

Programación Orienta a objetos:

“La Programación Orientada a Objetos es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y programas de computadora. Está basado en varias técnicas, incluyendo herencia, modularidad, polimorfismo y encapsulamiento. Su uso se popularizó a principios de la década de 1990.

La programación orientada a objetos expresa un programa como un conjunto de estos objetos, que colaboran entre ellos para realizar tareas. Esto permite hacer los programas y módulos más fáciles de escribir, mantener y reutilizar.” [9]

Lenguaje de programación para Aplicaciones Web:

“Cualquier persona que está familiarizada con el desarrollo de aplicaciones, sabe que el desarrollo de las páginas web no es una tarea simple. Ya que mientras que hay modelos de programación, para aplicaciones de uso común, muy bien establecidos y soportados por un gran número de lenguajes y herramientas de desarrollo, la programación web es una mezcla de varios lenguajes de etiquetas, un gran uso de lenguajes de script y plataformas de servidor. Por todo ello, el conocimiento y habilidad que posee un programador de nivel intermedio en el desarrollo tradicional de aplicaciones tiene muy poco en común con los conocimientos necesarios para desarrollar aplicaciones web.

Los lenguajes de programación más usados en el desarrollo web son principalmente ASP.NET o PHP. Como Asp.Net no es un lenguaje del lado del servidor, este se combina con el code-behind, habitualmente Vb.Net, convirtiendo a estos proyectos en los más potentes y flexibles en lo que a programación orientada a web se refiere.” [10]

Base de Datos

“Una base de datos es un “almacén” que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar

fácilmente. A continuación te presentamos una guía que te explicará el concepto y características de las bases de datos.

Desde el punto de vista informático, la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.

Cada base de datos se compone de una o más tablas que guarda un conjunto de datos. Cada tabla tiene una o más columnas y filas. Las columnas guardan una parte de la información sobre cada elemento que queramos guardar en la tabla, cada fila de la tabla conforma un registro.” [11]

Sistema de Gestión de Base de Datos

“Los Sistemas de Gestión de Base de Datos (en inglés DataBase Management System) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.” [11]

Gestores de Base de Datos

Entre los diferentes tipos de base de datos, podemos encontrar los siguientes:

- MySQL (comunitario y comercial)
- MariaDB (libre)
- PostgreSQL (libre)
- Firebird (libre)
- Oracle (comercial)
- Access (comercial)
- Microsoft SQL Server (comercial) [11]

Tipos de SGBD

“En la actualidad el modelo relacional es el más usado, aunque existen otros como por ejemplo:

- Modelo jerárquico
- Modelo en Red
- Modelo Relacional

Modelo Relacional

Es un modelo de datos desarrollado por Codd. La representación lógica de las entidades y sus relaciones se representan en tablas bidimensionales.

Se llamará registro o tupia a cada fila de la tabla y campo o atributo a cada columna de la tabla. Uno de los requisitos de las tablas será que no puede haber tupias repetidas. Una clave será un atributo o conjunto de atributos que identifique de forma única a una tupia. Las tablas deben de cumplir los siguientes requisitos:

Para manejar estas tablas se utilizan operaciones clásicas de la teoría de conjuntos (Unión, intersección, diferencia y producto cartesiano) así como operaciones específicas del modelo relacional (selección, proyección, reunión y división).

Modelo Jerárquico

Un SGBD jerárquico utiliza árboles, para la representación lógica de los datos. Un SGBD jerárquico posee las siguientes características:

- Los registros están dispuestos en forma de árbol y no pueden existir ciclos.
- Los registros sólo pueden estar relacionados mediante relaciones uno a uno o uno a muchos.
- Cuando se elimina un registro padre se borran todos sus hijos.

Modelo en Red

Los SGBD en red se basan en la utilización de la estructura no lineal red, en la que cada registro hijo puede tener más de un nodo padre, Tiene dos características principales:

- La estructura principal consiste en dos tipos de registros:

- El registro padre se denomina propietario del continuo (en el ejemplo "profesor"), mientras que el registro hijo se llama miembro (en el ejemplo Alumno). Existe un solo propietario y uno o más miembros.
- Un registro se puede asociar con más de un propietario.” [12]

MySQL

“MySQL es un sistema cliente servidor de administración de bases de datos relacionales diseñado para el trabajo tanto en los sistemas operativos Windows como en los sistemas UNIX/LINUX. Además, determinadas sentencias de MySQL pueden ser embebidas en código PHP Y HTML para diseñar aplicaciones web dinámicas que incorporan la información de las tablas de MySQL a páginas Web. Así mismo, MySQL es compatible con el software más potente de diseño Web, como Dreamweaver MX.” [13]

Motores de Almacenamiento de MySQL

“MySQL dispone, actualmente, de los siguientes motores de almacenamiento:

- **MyISAM:** Es el motor por defecto. Es muy rápido pero no transaccional.
- **InnoDB:** Es transaccional, incluyendo integridad referencial.
- **Memory (Heap):** Es una tabla MyISAM, pero almacena en memoria, no en disco. Es todavía más rápida.
- **Archive:** Es una tabla MyISAM, pero comprimida y de solo lectura.
- **MRG_MyISAM:** Es una agregación de tablas MyISAM. Las tablas agregadas deben ser exactamente iguales.
- **CSV:** Es una tabla que se almacena en un fichero de valores separados por comas.
- **FEDERATED:** Se trata de una tabla que, realmente, reside en otro servidor MySQL.
- **Blackhole:** Esta es una base de datos en la que todo lo que metes, desaparece.” [14]

Arquitectura de MySQL

“Podemos imaginar la arquitectura interna de MySQL dividida en tres capas. Se trata de una división lógica, que no coincide necesariamente con la división interna del código.

Las tres capas son:

1. **Capa de Conexión:** En la que reside la funcionalidad que conecta MySQL con otros sistemas y lenguajes (APIs, sockets, ODBC, etc.).
2. **Capa de Lógica:** En la que reside la lógica para procesar consultas SQL (parseo de sentencias, planificación, ejecución, cachés, etc.).
3. **Capa de Almacenamiento:** En la que reside la lógica para almacenar y acceder a los datos de las tablas.

Una característica de MySQL es que puede utilizar distintos motores de almacenamiento. Es decir, que la capa de almacenamiento dispone de varios subniveles, uno por cada tipo de motor soportado.

La siguiente figura nos aclarará un poco más las cosas:

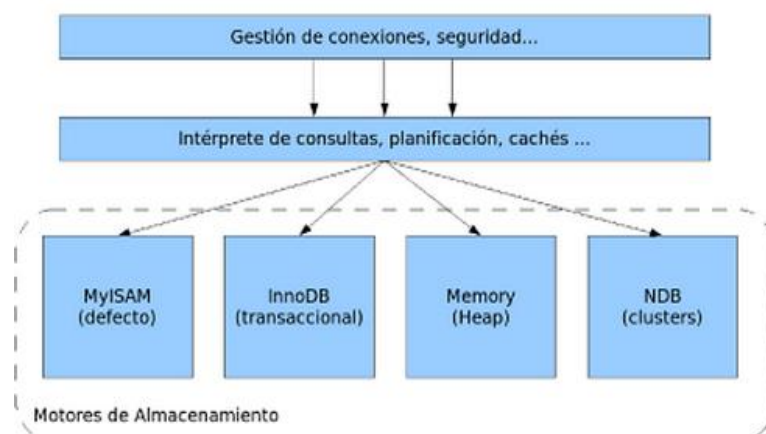


Figura. 2.1: Arquitectura MySQL [14]

Podemos ver que MySQL soporta cuatro motores de almacenamiento básicos. Hay que destacar que una base de datos puede almacenar sus datos utilizando una combinación de distintos motores. Es decir, parte de los datos pueden estar en una tabla MyISAM y otra parte en una tabla InnoDB.” [14]

PHP

“El lenguaje PHP es un lenguaje de programación de estilo clásico, con esto quiero decir que es un lenguaje de programación con variables, sentencias condicionales, bucles, funciones. No es un lenguaje de marcas como podría ser HTML, XML o WML. Está más cercano a JavaScript o a C, para aquellos que conocen estos lenguajes.

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, pero sin embargo para que sus páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP.” [15]

JavaScript

“JavaScript es un lenguaje de script multiplataforma [cross-platform] orientado a objetos.

Es un lenguaje pequeño y ligero; no es útil como un lenguaje independiente, más bien está diseñado para una fácil incrustación en otros productos y aplicaciones, tales como los navegadores Web. Dentro de un entorno anfitrión, JavaScript puede ser conectado a los objetos de su entorno para proveer un control programable sobre éstos.” [16]

CSS

“CSS son las siglas de Cascading Style Sheets - Hojas de Estilo en Cascada - que es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación, es decir, describe cómo se va a mostrar un documento en pantalla, por impresora, por voz (cuando la información es pronunciada a través de un dispositivo de lectura) o en dispositivos táctiles basados en Braille.” [17]

jQuery

“jQuery consiste en un único fichero JavaScript que contiene las funcionalidades Comunes de DOM, eventos, efectos y AJAX. La característica principal de la biblioteca es que permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX. Para ello utiliza las funciones `$()` o `jQuery()`.” [18]

JSON

“(JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo.” [19]

Modelo – Vista – Controlador (MVC)

“El modelo–vista–controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

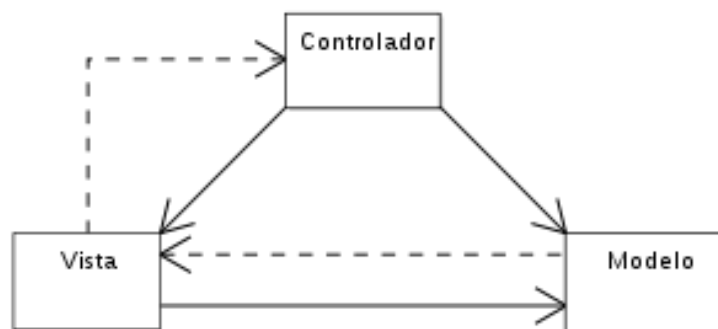


Figura. 2.2: Diagrama Relacional MVC [20]

Descripción del patrón

De manera genérica, los componentes de MVC se podrían definir como sigue:

- **El Modelo:** Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.
- **La Vista:** Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho 'modelo' la información que debe representar como salida.
- **El Controlador:** Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta de 'modelo' (por ejemplo, desplazamiento o scroll por un documento o por los diferentes registros de una base de datos), por tanto se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo' (véase Middleware).

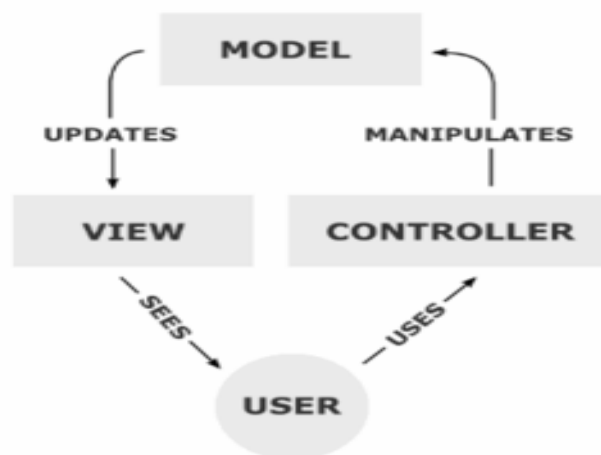


Figura. 2.3: Colaboración entre los componentes de un MVC [20]

Interacción de los componentes

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo de control que se sigue generalmente es el siguiente:

- El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.)
- El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
- El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
- El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se reflejan los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, se podría utilizar el patrón Observador para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. Este uso del patrón Observador no es posible en las aplicaciones Web puesto que las clases de la vista están desconectadas del modelo y del controlador.
- La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

MVC y Bases de Datos

Muchos sistemas informáticos utilizan un Sistema de Gestión de Base de Datos para gestionar los datos que debe utilizar la aplicación; en líneas generales del MVC dicha gestión corresponde al modelo. La unión entre capa de presentación y capa de negocio

conocido en el paradigma de la Programación por capas representaría la integración entre la Vista y su correspondiente Controlador de eventos y acceso a datos, MVC no pretende discriminar entre capa de negocio y capa de presentación pero si pretende separar la capa visual gráfica de su correspondiente programación y acceso a datos, algo que mejora el desarrollo y mantenimiento de la Vista y el Controlador en paralelo, ya que ambos cumplen ciclos de vida muy distintos entre sí.

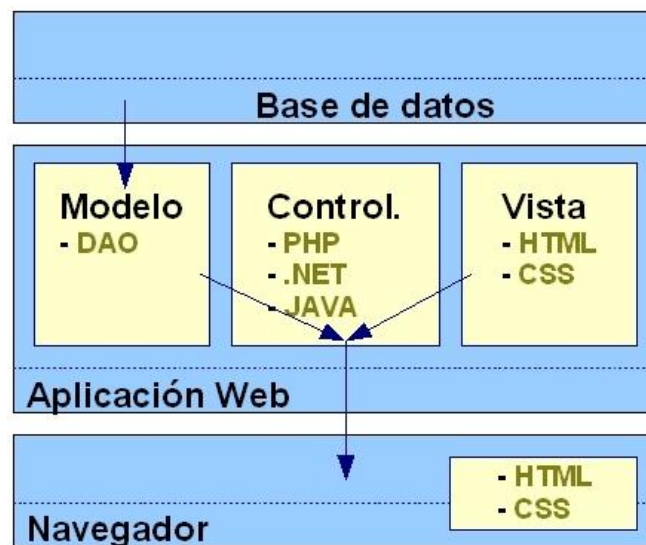


Figura. 2.4: Base de Datos y MVC [20]

Uso en aplicaciones Web

Aunque originalmente MVC fue desarrollado para aplicaciones de escritorio, ha sido ampliamente adaptado como arquitectura para diseñar e implementar aplicaciones web en los principales lenguajes de programación. Se han desarrollado multitud de frameworks, comerciales y no comerciales, que implementan este patrón (ver apartado siguiente "Frameworks MVC"); estos frameworks se diferencian básicamente en la interpretación de como las funciones MVC se dividen entre cliente y servidor.

Los primeros frameworks MVC para desarrollo web planteaban un enfoque de cliente ligero en el que casi todas las funciones, tanto de la vista, el modelo y el controlador recaían en el servidor. En este enfoque, el cliente manda la petición de cualquier hiperenlace o formulario al controlador y después recibe de la vista una página completa y actualizada (u otro documento); tanto el modelo como el controlador (y buena parte de

la vista) están completamente alojados en el servidor. Como las tecnologías web han madurado, ahora existen frameworks como JavaScriptMVC, Backbone o jQuery14 que permiten que ciertos componentes MVC se ejecuten parcial o totalmente en el cliente.” [20]

Frameworks más utilizados en MVC

- “Java Swing
- Java Enterprise Edition (J2EE)
- XForms (Formato XML estándar del W3C para la especificación de un modelo de proceso de datos XML e interfaces de usuario como formularios web)
- GTK+ (escrito en C, toolkit creado por Gnome para construir aplicaciones gráficas, inicialmente para el sistema X Windows)
- ASP.NET MVC Framework (Microsoft)
- PHP
- Google Web Toolkit (GWT, para crear aplicaciones Ajax con Java)
- Apache Struts (framework para aplicaciones web J2EE)
- Ruby on Rails (framework para aplicaciones web con Ruby)
- Etc., etc., etc.” [21]

Modelo de Desarrollo Iterativo e Incremental

“Desarrollo iterativo y creciente (o incremental) es un proceso de desarrollo de software creado en respuesta a las debilidades del modelo tradicional de cascada.

Básicamente este modelo de desarrollo, que no es más que un conjunto de tareas agrupadas en pequeñas etapas repetitivas (iteraciones), es uno de los más utilizados en los últimos tiempos ya que, como se relaciona con novedosas estrategias de desarrollo de software y una programación extrema, es empleado en metodologías diversas.

El modelo consta de diversas etapas de desarrollo en cada incremento, las cuales inician con el análisis y finalizan con la instauración y aprobación del sistema.

Ciclo de Vida

La idea principal detrás de mejoramiento iterativo es desarrollar un sistema de programas de manera incremental, permitiéndole al desarrollador sacar ventaja de lo que se ha aprendido a lo largo del desarrollo anterior, incrementando, versiones entregables del sistema. El aprendizaje viene de dos vertientes: el desarrollo del sistema, y su uso (mientras sea posible).

Los pasos claves en el proceso son comenzar con una implementación simple de los requerimientos del sistema, e iterativamente mejorar la secuencia evolutiva de versiones hasta que el sistema completo esté implementado. En cada iteración, se realizan cambios en el diseño y se agregan nuevas funcionalidades y capacidades al sistema.

El proceso en sí mismo consiste de:

- Etapa de inicialización
- Etapa de iteración
- Lista de control de proyecto

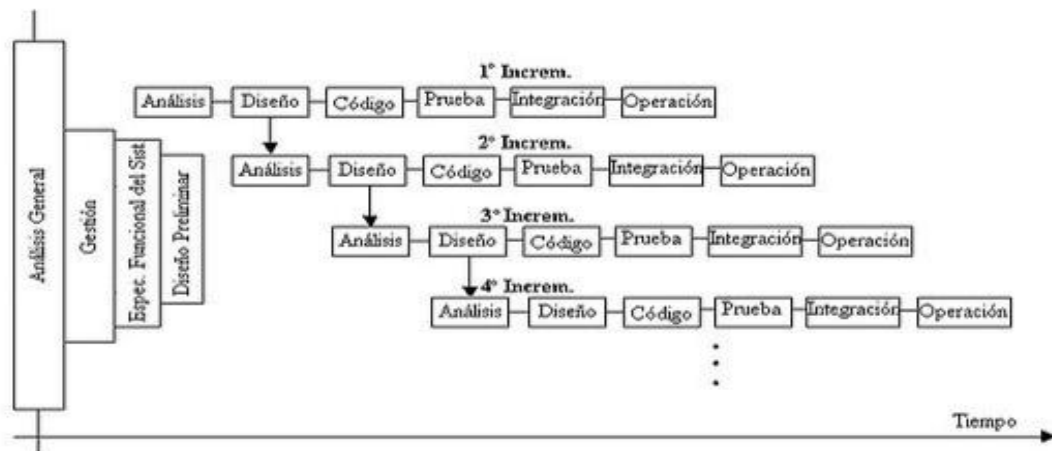


Figura. 2.5: Modelo Iterativo e Incremental para el Ciclo de Vida del Software [22]

Etapa de Inicialización

Se crea una versión del sistema. La meta de esta etapa es crear un producto con el que el usuario pueda interactuar, y por ende retroalimentar el proceso. Debe ofrecer una muestra de los aspectos claves del problema y proveer una solución lo suficientemente simple para ser comprendida e implementada fácilmente. Para guiar el proceso de iteración se crea una lista de control de proyecto, que contiene un historial de todas las tareas que necesitan ser realizadas. Incluye cosas como nuevas funcionalidades para ser

implementadas, y áreas de rediseño de la solución ya existente. Esta lista de control se revisa periódica y constantemente como resultado de la fase de análisis.

Etapa de Iteración

Esta etapa involucra el rediseño e implementación de una tarea de la lista de control de proyecto, y el análisis de la versión más reciente del sistema. La meta del diseño e implementación de cualquier iteración es ser simple, directa y modular, para poder soportar el rediseño de la etapa o como una tarea añadida a la lista de control de proyecto. El código puede, en ciertos casos, representar la mayor fuente de documentación del sistema.

El análisis de una iteración se basa en la retroalimentación del usuario y en el análisis de las funcionalidades disponibles del programa. Involucra el análisis de la estructura, modularidad, usabilidad, confiabilidad, eficiencia y eficacia (alcanzar las metas).

La lista de control del proyecto se modifica y se revisa constantemente como resultado de la fase de análisis.



Figura. 2.6: Ciclo de Vida del Modelo de Desarrollo Iterativo e Incremental [22]

Fases del Proceso Unificado de las Iteraciones

- Inicio
- Elaboración
- Construcción
- Transición

El Inicio identifica el alcance del proyecto, los riesgos y los requisitos (funcionales y no funcionales) a un alto nivel en suficiente detalle para que se pueda estimar el trabajo.

La Elaboración entrega una arquitectura de trabajo que mitiga los riesgos altos y cumple los requisitos no funcionales.

La Construcción reemplaza incrementalmente la arquitectura con código listo para producción del análisis, diseño, implementación y pruebas de los requisitos funcionales.

La Transición entrega el sistema al entorno operativo de producción.

Cada una de las fases puede dividirse en una o más iteraciones, que se agrupan en función de tiempo más que de característica. Los arquitectos y analistas trabajan una iteración por delante de los desarrolladores y testers.

Las guías primarias que guían la implementación y el análisis incluyen:

- Cualquier dificultad en el diseño, codificación y prueba de una modificación debería apuntar a la necesidad de rediseñar o recodificar.
- Las modificaciones deben ajustarse fácilmente a los módulos fáciles de encontrar y a los aislados. Si no es así, entonces se requiere algún grado de rediseño.
- Las modificaciones a las tablas deben ser especialmente fáciles de realizar. Si dicha modificación no ocurre rápidamente, se debe aplicar algo de rediseño.
- Las modificaciones deben ser más fáciles de hacer conforme avanzan las iteraciones. Si no es así, hay un problema primordial usualmente encontrado en un diseño débil o en la proliferación excesiva de parches al sistema.
- Los parches normalmente deben permanecer solo por una o dos iteraciones. Se hacen necesarios para evitar el rediseño durante una fase de implementación.
- La implementación existente debe ser analizada frecuentemente para determinar qué tal se ajusta a las metas del proyecto.

- Las facilidades para analizar el programa deben ser utilizadas cada vez para ayudar en el análisis de implementaciones parciales.
- La opinión del usuario debe ser solicitada y analizada para indicar deficiencias en la implementación referida por él.

Características

Usando análisis y mediciones como guías para el proceso de mejora es una diferencia mayor entre las mejoras iterativas y el desarrollo rápido de aplicaciones, principalmente por dos razones:

- ✓ Provee de soporte para determinar la efectividad de los procesos y de la calidad del producto.
- ✓ Permite estudiar y después mejorar y ajustar el proceso para el ambiente en particular.

Estas mediciones y actividades de análisis pueden ser añadidas a los métodos de desarrollo rápido existentes.

De hecho, el contexto de iteraciones múltiples conlleva ventajas en el uso de mediciones. Las medidas a veces son difíciles de comprender en lo absoluto, aunque en los cambios relativos en las medidas a través de la evolución del sistema puede ser muy informativo porque proveen una base de comparación. Por ejemplo, un vector de medidas m_1, m_2, \dots, m_n puede ser definido para caracterizar varios aspectos del producto en cierto punto, como pueden ser el esfuerzo total realizado, los cambios, los defectos, los atributos lógico, físico y dinámico, consideraciones del entorno, etcétera. Así el observador puede decir como las características del producto como el tamaño, la complejidad, el acoplamiento y la cohesión incrementan o disminuyen en el tiempo. También puede monitorearse el cambio relativo de varios aspectos de un producto o pueden proveer los límites de las medidas para apuntar a problemas potenciales y anomalías.

Ventajas del Desarrollo Incremental

- En este modelo los usuarios no tienen que esperar hasta que el sistema completo se entregue para hacer uso de él. El primer incremento cumple los requerimientos más importantes de tal forma que pueden utilizar el software al instante.
- Los usuarios pueden utilizar los incrementos iniciales como prototipos y obtener experiencia sobre los requerimientos de los incrementos posteriores del sistema.
- Existen muy pocas probabilidades de riesgo en el sistema. Aunque se pueden encontrar problemas en algunos incrementos, lo normal es que el sistema se entregue sin inconvenientes al usuario.
- Ya que los sistemas de más alta prioridad se entregan primero, y los incrementos posteriores se integran entre ellos, es muy poco probable que los sistemas más importantes sean a los que se les hagan más pruebas. Esto quiere decir que es menos probable que los usuarios encuentren fallas de funcionamiento del software en las partes más importantes del sistema.

Ventajas del Desarrollo Iterativo

- En el desarrollo de este modelo se da la retroalimentación muy temprano a los usuarios.
- Permite separar la complejidad del proyecto, gracias a su desarrollo por parte de cada iteración o bloque.
- El producto es consistente y puntual en el desarrollo.
- Los productos desarrollados con este modelo tienen una menor probabilidad de fallar.
- Se obtiene un aprendizaje en cada iteración que es aplicado en el desarrollo del producto y aumenta las experiencias para próximos proyectos.

Debilidades de este Modelo De Desarrollo

- La entrega temprana de los proyectos produce la creación de sistemas demasiados simples que a veces se ven un poco monótonos a los ojos del personal que lo recibe.

- La mayoría de los incrementos se harán en base de las necesidades de los usuarios. Los incrementos en si ya son estipulados desde antes de la entrega del proyecto, sin embargo hay que ver cómo se maneja el producto para ver si necesita otros cambios además de los estipulados antes de la entrega del proyecto. Este problema no se ve frecuentemente ya que la mayoría de las veces los incrementos estipulados suplen satisfactoriamente al usuario.
- Los incrementos no deben constar de muchas líneas de código ya que la idea de los incrementos es agregar accesorios al programa principal (o funcional), para que este tenga una y mil formas de desenvolverse en su tarea; llenar los incrementos de muchas líneas de código provocaría que se perdiera la objetividad o base de lo que se trata el desarrollo incremental.
- Requiere de un cliente involucrado durante todo el curso del proyecto. Hay clientes que simplemente no estarán dispuestos a invertir el tiempo necesario.
- El trato con el cliente debe basarse en principios éticos y colaboración mutua, más que trabajar cada parte independientemente, defendiendo sólo su propio beneficio.
- La entrega de un programa que es parcial pero funcional puede hacer vulnerable al programa debido a la falta de robustez en su sistema, provocando que agentes ajenos puedan interferir con el correcto funcionamiento del programa en sí.
- Sufre fuertes penalizaciones en proyectos en los cuales los requerimientos están previamente definidos, o para proyectos "todo/nada" en los cuales se requiere que se completen en un 100% el producto para ser implementado (por ejemplo, licitaciones) otro punto muy importante es asegurarnos de que el trabajo se pueda cumplir tomando en cuenta los costos que podamos usar en nuestros propios recursos.” [22]

Beneficios del Ciclo de Vida del Software Iterativo e Incremental

- “Se puede gestionar las expectativas del cliente (requisitos desarrollados, velocidad de desarrollo, calidad) de manera regular, puede tomar decisiones en cada iteración. Esto es especialmente interesante cuando:

- El cliente no sabe exactamente qué es lo que necesita, lo va sabiendo conforme va viendo cuales son los resultados del proyecto.
- El cliente necesita hacer cambios a corto plazo (nuevos requisitos o a cambios en los ya realizados) por:
 - Cambios en las condiciones del mercado (por un cambio de necesidades, por un nuevo producto que ha lanzado la competencia, urgencias).
 - La reacción y aceptación del mercado respecto al uso de los primeros resultados del proyecto.
 - Cualquier cambio en el entorno (recursos, etc.), que pueda incluso finalizar el proyecto manteniendo como mínimo los resultados alcanzados hasta ese momento.
- El equipo necesita saber si lo que ha entendido es lo que el cliente espera.
- El cliente puede comenzar el proyecto con requisitos de alto nivel, quizás no del todo completos, de manera que se vayan refinando en sucesivas iteraciones. Sólo es necesario conocer con más detalle los requisitos de las primeras iteraciones, los que más valor aportan. No es necesario realizar una recolección completa y detallada de todos los requisitos antes de empezar el desarrollo del proyecto.
- El cliente puede obtener resultados importantes y usables ya desde las primeras iteraciones.
- El cliente como máximo puede perder los recursos dedicados a una iteración, no los de todo el proyecto.
- La finalización de cada iteración es el lugar natural donde el equipo puede decidir cómo mejorar su proceso de trabajo, en función de la experiencia obtenida. Con esta información ya es posible planificar los cambios necesarios para aumentar la productividad y calidad desde las primeras iteraciones.
- Permite conocer el progreso real del proyecto desde las primeras iteraciones y extrapolar si su finalización es viable en la fecha prevista. El cliente puede decidir re-priorizar los requisitos del proyecto, añadir nuevos equipos, cancelarlo, etc.

- Permite mitigar desde el inicio los riesgos del proyecto. Desde la primera iteración el equipo tiene que gestionar los problemas que pueden aparecer en una entrega del proyecto. Al hacer patentes estos riesgos, es posible iniciar su mitigación de manera anticipada.
- Permite gestionar la complejidad del proyecto.
 - En una iteración sólo se trabaja en los requisitos que aportan más valor en ese momento.
 - Se puede dividir la complejidad para que cada parte sea resuelta en diferentes iteraciones.
- Dado que cada iteración debe dar como resultado requisitos terminados, se minimiza el número de errores que se producen en el desarrollo y se aumenta la calidad.

Restricciones del Ciclo de Vida del Software Iterativo e Incremental

- La disponibilidad del cliente debe ser alta durante todo el proyecto dado que participa de manera continua:
 - El inicio de una iteración, el cliente ha de detallar (o haber detallado previamente) los requisitos que se van a desarrollar.
 - En la finalización de cada iteración, el cliente ha de revisar los requisitos desarrollados.
- La relación con el cliente ha de estar basada en los principios de colaboración y ganar/ganar más que tratarse de una relación contractual en la cual cada parte únicamente defiende su beneficio a corto plazo.
- Cada iteración debe dar como resultado requisitos terminados, de manera que el resultado sea realmente útil para el cliente y no deje tareas pendientes para futuras iteraciones o para la finalización del proyecto.

- Es necesario disponer de técnicas y herramientas que permitan hacer cambios fácilmente en el producto, de manera que pueda crecer en cada iteración de manera incremental sin hacer un gran esfuerzo adicional, manteniendo su complejidad minimizada y su calidad.

Recomendaciones para el Desarrollo del Ciclo de Vida del Software Iterativo e Incremental

- Utilizar iteraciones cortas de 2 a 4 semanas incrementa la productividad del proyecto, dado que el equipo trabaja de forma más eficiente cuando tiene objetivos a corto plazo.
Asimismo, con iteraciones cortas la precisión de las estimaciones aumenta.
- El tamaño depende de:
 - Los condicionantes del proyecto.
 - La necesidad de tener feedback más o menos rápido.
 - Que no se degrade la relación trabajo útil / gestión operativa (por ejemplo reuniones, actividades necesarias que no producen valor directo, etc.).
- Utilizar iteraciones regulares, de manera que todas sean un timebox de la misma duración.
 - El equipo aprende a calcular la velocidad de desarrollo, la cantidad de trabajo que puede hacer en una iteración (sin tener que hacer extrapolaciones si las iteraciones no fuesen regulares).
 - El cliente puede proyectar cuantas iteraciones se necesitan para tener cada entrega, en función de la velocidad de desarrollo del equipo (el trabajo que pudo completar en iteraciones anteriores del mismo tamaño), y tomar decisiones al respecto.
 - Permite gestionar y sincronizar de manera sencilla las necesidades del proyecto con respecto a las de otros proyectos (integración con el trabajo realizado por otros equipos, compartición de personas que son difíciles de asignar a un único equipo).

- o Las iteraciones coincidiendo con meses naturales permiten sincronizar el trabajo del equipo con el de otros departamentos y con el resto de la organización (por ejemplo, la organización puede tener medidas de resultados y objetivos a nivel trimestral o cuatrimestral).” [23]

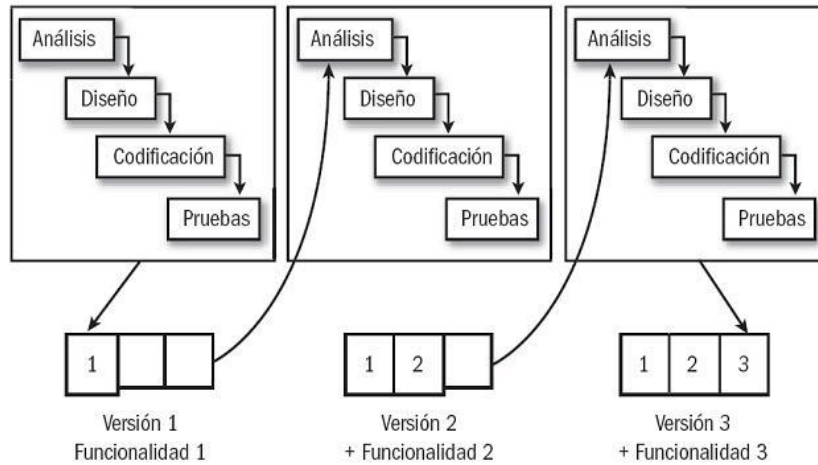


Figura. 2.7: Etapas Generales del Modelo de Desarrollo Iterativo e Incremental [23]

2.3. Propuesta de Solución

El sistema académico web utilizando tecnología móvil para el Colegio a Distancia Stephen Hawking de la ciudad de Ambato buscará solucionar los problemas manuales de los registros de notas, además facilitara la consulta de las mismas a través de un dispositivo móvil sin la necesidad de utilizar una computadora continuamente.

CAPÍTULO III

3. METODOLOGÍA

3.1. Modalidad de la Investigación

Se utilizara dentro de esta investigación varios tipos de modalidades para tener éxito en el proceso de recolección de información sobre los procesos académicos del colegio para lo cual se ha considerado utilizar las siguientes modalidades:

Modalidad Bibliográfica o Documental: Esta modalidad se ha tomado en cuenta para la elaboración del marco teórico, ya que en esta investigación se ha tomado información teórica de libros físicos, libros digitales, internet, etc.

Modalidad de Campo: Se aplicará esta modalidad investigativa porque el investigador para la recolección de información deberá acudir directamente a las oficinas del colegio, utilizando técnicas como la entrevista y el cuestionario, para determinar los requerimientos y generar una idea clara y concisa del proceso.

3.2. Población y Muestra

La población o muestra no es requerida ya que no se ha planteado una hipótesis para el sistema web.

3.3. Operacionalización de Variables u Objetivos

Tabla 3.1: Operacionalización de Variables u Objetivos

Objetivo	Actividades
Analizar e Identificar los procesos académicos que se llevan a cabo de forma manual en el Colegio a Distancia Stephen Hawking de la ciudad de Ambato.	Observar el proceso manual del control académico de los estudiantes dentro de la institución.
Establecer los requerimientos y alcance del Sistema Académico Web utilizando tecnología móvil para el Colegio a Distancia Stephen Hawking de la ciudad de Ambato.	Observar y determinar los requerimientos necesarios para el desarrollo del sistema académico web.
Analizar las características del Sistema Académico Web utilizando tecnología móvil del Colegio a Distancia Stephen Hawking de la ciudad de Ambato.	Selección del SGBD Selección del Lenguaje de Programación.
Diseñar un Sistema Académico Web utilizando tecnología móvil para el Colegio a Distancia Stephen Hawking de la ciudad de Ambato.	Diseño de la Base de Datos. Diseño de las interfaces para los distintos tipos de usuarios. Diseño de la Pagina Web.
Desarrollar un Sistema Académico Web utilizando tecnología móvil para el Colegio a Distancia Stephen Hawking de la ciudad de Ambato.	Creación de la Base de Datos. Codificación de la Pagina Web. Configuración del Servidor Web.
Realizar pruebas de funcionamiento al Sistema Académico Web.	Pruebas de Caja Negra Pruebas de Caja Blanca

3.4. Recolección de Información

Esta investigación está dirigida al personal docente y administrativo del Colegio a Distancia Stephen Hawking de la Ciudad de Ambato.

Para la recolección de la información primaria se emplea la técnica de observación y como instrumento una guía de observación, en la cual el investigador al tener relación directa con el problema que se origina, por el tiempo necesario, llenara la guía en base a las actividades que realizan el personal Administrativo del Colegio a Distancia Stephen Hawking de la ciudad de Ambato, quienes manejan el control académico de la institución, según los ítems que se propone.

Además se requerirá información que se encuentra almacenada/registrada en libros, documentos, revistas técnicas, proyectos de investigación, tesis de grado y en Internet.

3.5. Procesamiento y Análisis de Datos

Una vez recolectada la información se procedió al análisis de los datos obtenidos los cuales fueron importantes para la formulación de la propuesta. Los datos fueron analizados y procesados en relación al problema para poder establecer las respectivas conclusiones asegurando que los datos sean lo más reales posibles.

Como resultado del diálogo realizado directamente con la Ing. Gisela Ramírez, Administradora del Área Informática del Colegio a Distancia Stephen Hawking se ha obtenido la siguiente información:

El Área de Secretaria está encargada de los procesos académicos del colegio sean estos ingresos, modificaciones, actualizaciones y eliminación de notas, así como también la elaboración de los reportes de notas o boletines de notas que serán posteriormente publicados para consulta de los estudiantes.

El Área de Docentes lleva el registro de notas de los estudiantes de manera manual, registro que luego es entregado al Área de Secretaria, para ello utilizan una planilla o formato realizado en Microsoft Excel.

Solo el Área de Sistemas puede realizar cualquier tipo de modificación de información de notas, estudiantes y docentes.

Una vez finalizada la reunión con la Ingeniera a cargo del Área de Sistemas del Colegio a Distancia Stephen Hawking de la Ciudad de Ambato se ha llegado a la conclusión que la Institución Educativa no cuenta con un Sistema Automatizado para llevar a cabo el control del proceso Académico y más aun la utilización del mismo desde dispositivos móviles, es decir se lo realiza de forma manual lo que conlleva a un sinnúmero de infortunios, motivo por el cual precisa un Sistema Académico Web utilizando tecnología móvil que permita el control y el manejo del área académica del colegio.

Análisis del Proceso de Publicación de Boletines de Notas

Diagrama del Proceso

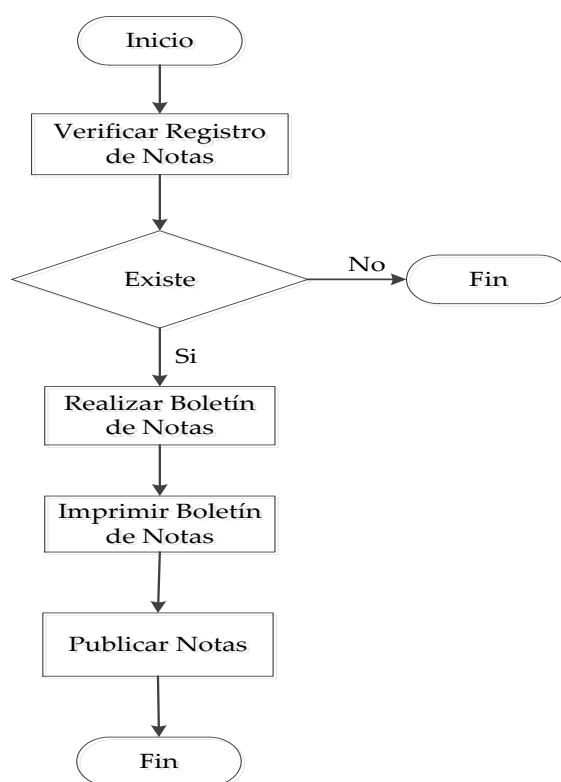


Figura. 3.1: Diagrama de Flujo de la Publicación de Boletines de Notas

Análisis del Proceso de Registro de Notas

Diagrama del Proceso

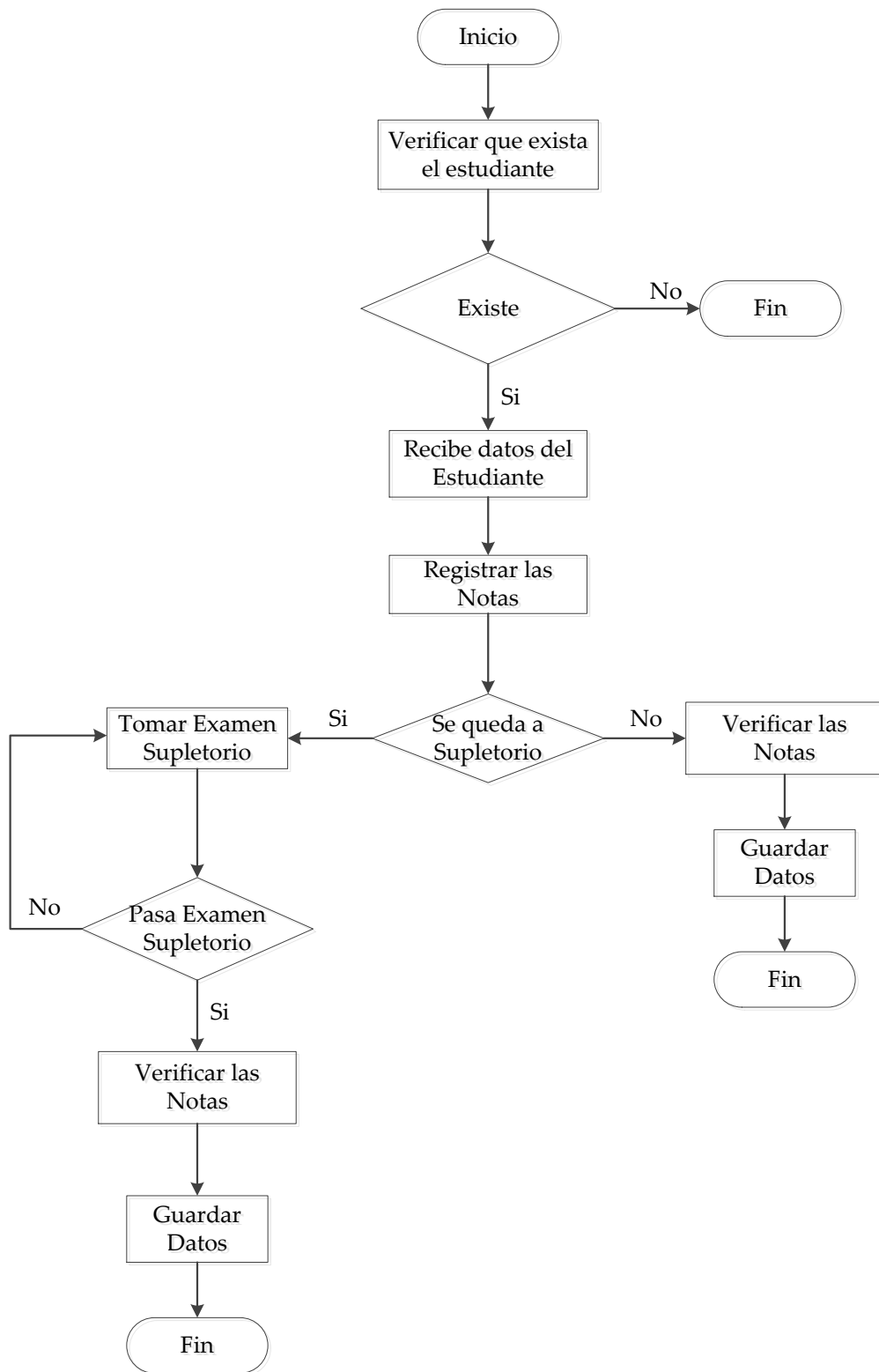


Figura. 3.2: Diagrama de Flujo del Registro de Notas

3.6. Desarrollo del Proyecto

Para cumplir el desarrollo de este proyecto de investigación, a continuación se detallan los métodos, procesos y/o actividades que se realizarán, para cumplir los objetivos que llevaron a la obtención del producto final, teniendo en cuenta que la metodología utilizada es el Modelo Iterativo e Incremental.

Actividades:

Se realiza las iteraciones necesarias hasta lograr el producto deseado por el cliente, utilizando la retroalimentación de procesos requeridos.

- Análisis
 - Requerimientos del Sistema
 - Casos de Uso
 - Diagrama de Secuencia
- Diseño
 - Diseño de la base de datos.
 - Diccionario de datos.
 - Diseño de la interfaz.
- Implementación
 - Codificación y depuración.
- Pruebas
 - Pruebas de caja Blanca
 - Pruebas de caja Negra

CAPÍTULO IV

4. DESARROLLO DE LA PROPUESTA

En el desarrollo del presente proyecto se utilizó el modelo Iterativo e Incremental, mediante el cual el producto se desarrolla por incrementos en el que cada iteración obtiene una versión funcional del producto, de esta forma el sistema se desarrolla poco a poco y obtienen una retroalimentación continua por parte del usuario.

✓ Iteración 1

I.1	A	D	I	P
-----	---	---	---	---

4.1. Análisis

Es el proceso de investigar un problema que se quiere resolver, definir claramente el Problema que se quiere resolver o el sistema que se desea crear, así como también identificar los componentes principales que integrarán el producto.

4.1.1. Requerimientos del Sistema

Utilizando como técnica la observación y como instrumento una guía de observación, se ha recolectado los datos necesarios, mismos que nos permiten conocer los procesos actuales que se realiza en el Colegio a Distancia Stephen Hawking de la Ciudad de Ambato.

➤ Guía de Observación

Tabla 4.1: Guía de Observación de Información

GUÍA DE OBSERVACIÓN	
Objetivo: Recolectar información del Desarrollo de un Proceso de Control Académico, para la automatización del mismo mediante una aplicación Web.	
Tipo de Observación: Observación Directa	
INDICADORES A OBSERVAR	
✓ Medios para la Recolección de la Información	
Ficha de Registro de Notas (Ver Anexo 2)	
Ficha de Pase de Año de los Estudiantes (Ver Anexo 3)	
✓ Medios en que se Almacena la Información Generada	
Hojas de Calculo	Archivos de Textos
Carpetas Físicas	Discos Magnéticos
✓ Documentos Generados en el Procesamiento	
Informe de Calificaciones y Conducta	Informe de Docentes
Informe de Pases de Año	Informe de Estudiantes
✓ Disponibilidad de la Información	
Búsqueda de Archivos Digitales	
Búsqueda de Archivos Físicos en Carpetas	
✓ Elaboración de los Cronogramas	
Creación de Cursos en base a los procesos	Asignación de Materias
Asignación de Cursos	Asignación de Paralelos
Asignación de Especialidad	Asignación de Docentes
✓ Actividades en el Desarrollo de un Proceso Académico	
Registrar Notas de los Estudiantes	Asignar un curso a una especialidad
Registrar Conducta de los Estudiantes	Asignar un paralelo a un curso
Asignar un paralelo a un curso	Asignar un docente a un curso
Asignar un estudiante a un curso	
✓ Seguimiento de Calificaciones	
Evaluación al final de cada Quimestre	Elaboración de un archivo digital
Archivos físicos de las Evaluaciones	

Una vez finalizado el diálogo se ha llegado a la conclusión de que el Colegio a Distancia Stephen Hawking de la Ciudad de Ambato no cuenta con un Sistema Web que permita el apropiado control de los procesos académicos tanto de estudiantes como de docentes y la facilidad de uso de dispositivos móviles, ya que dicha labor se la realiza de manera manual lo que provoca cierta disconformidad por parte del personal del colegio como también en los estudiantes que forman parte del mismo.

Para la elaboración del Sistema Académico Web Utilizando Tecnología Móvil se realizaron reuniones con el personal encargado del manejo, registro y manipulación de los datos, en dichas reuniones se establecieron los requerimientos que debe cumplir el sistema para solucionar los problemas que se presentan a lo largo de los años.

A continuación se detalla los requerimientos funcionales con los que contará el sistema.

➤ **Funcionales:**

- ✓ Ingresar información correcta, manteniendo la consistencia de los datos.
- ✓ Proteger la información, estableciendo tipos de usuarios, los mismos que contarán con un usuario y una contraseña para acceder al sistema. Estos usuarios serán creados por el administrador del sistema.
- ✓ Contar con la información del sistema de forma rápida y precisa.
- ✓ Generar reportes.
- ✓ Interfaz de usuario amigable e intuitivo para el fácil manejo del sistema.

4.1.2. Diagramas de Casos de Uso

Se define un caso de uso como cada interacción supuesta con el sistema a desarrollar, donde se representan los requisitos funcionales. Es decir se está diciendo que es lo que tiene que hacer un sistema y como.

Diagrama de Caso de Uso: Funciones Administrador

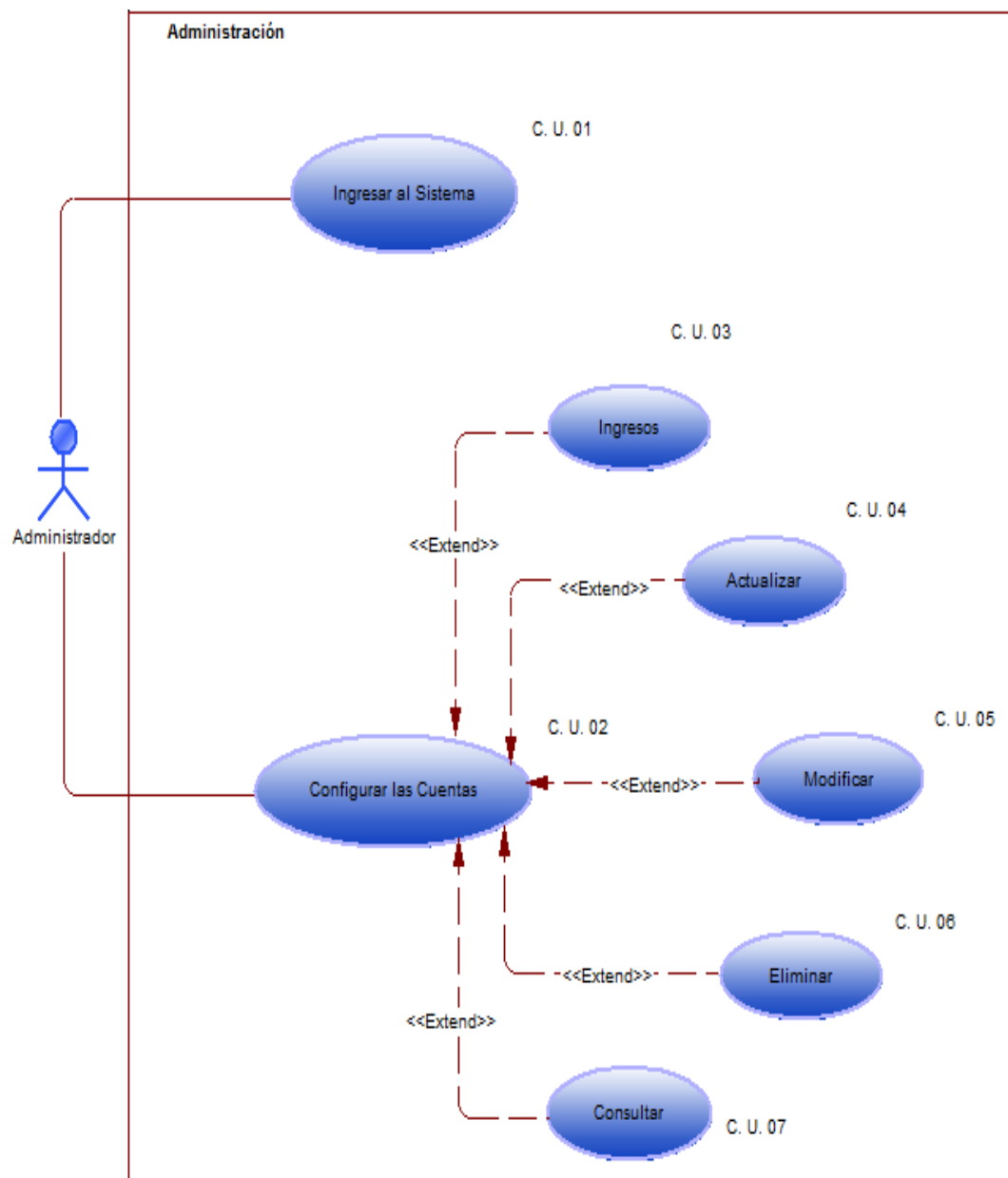


Figura 4.1: Diagrama de Caso de Uso Administrador

Descripción de los Casos de Uso

Tabla 4.2: Caso de Uso: Ingresar al Sistema

Caso de Uso: Ingreso al Sistema (C. U. 01)							
Descripción:	Detalle de cómo ingresar al sistema.						
Actor:	Administrador, Usuario (Usuarios Registrados)						
Precondición:	Los actores deben estar previamente registrados en el sistema como usuarios y tener una sesión activa para poder acceder a la Aplicación.						
Poscondición:	Se presentara la interfaz de la aplicación según el filtrado de lo que puede hacer.						
Presunción:	Base de datos Disponible.						
Flujo Principal:	<table border="1"> <thead> <tr> <th>Evento Actor</th> <th>Evento Sistema</th> </tr> </thead> <tbody> <tr> <td>1. Ingreso de datos necesarios para su autenticación.</td> <td>1. Muestra la pantalla de acceso al Sistema de la Aplicación.</td> </tr> <tr> <td>2. Envío de los datos mediante una petición web.</td> <td>2. Mostrar mensajes de Validación de los Datos.</td> </tr> </tbody> </table>	Evento Actor	Evento Sistema	1. Ingreso de datos necesarios para su autenticación.	1. Muestra la pantalla de acceso al Sistema de la Aplicación.	2. Envío de los datos mediante una petición web.	2. Mostrar mensajes de Validación de los Datos.
	Evento Actor	Evento Sistema					
1. Ingreso de datos necesarios para su autenticación.	1. Muestra la pantalla de acceso al Sistema de la Aplicación.						
2. Envío de los datos mediante una petición web.	2. Mostrar mensajes de Validación de los Datos.						
Flujo Alternativo	1. Verificar la Información recibida. 2. Redireccionar (Aplicación o Nueva Petición)						

Tabla 4.3: Caso de Uso: Ingresos

Caso de Uso: Ingresos (C. U. 03)	
Descripción:	Permite ingresar nuevos registros según la instancia deseada
Actor:	Usuario registrado como Administrador
Poscondición:	El usuario deberá logrearse para ingresar al sistema.
Flujo Normal:	Pulsar botón Nuevo. Ingresar los datos del nuevo registro. Pulsar el botón de Guardar.
Flujo Alternativo:	El sistema comprueba la validez de los datos. Si los datos son correctos guarda el nuevo registro Si son incorrectos envía un mensaje al usuario con el error.

L I A D I P

Tabla 4.4: Caso de Uso: Actualizar

Caso de Uso: Actualizar (C. U. 04)	
Descripción:	Permite actualizar registros según la instancia deseada
Actor:	Usuario registrado como Administrador
Poscondición:	El usuario deberá logrearse para ingresar al sistema.
Flujo Normal:	Seleccionar del menú el ítem deseado Seleccionar los datos a Actualizar.
Flujo Alternativo:	Actualizar: El sistema comprueba la validez de los datos Si los datos son correctos actualiza y muestra los datos actualizados. Si son incorrectos envía un mensaje al usuario con el error.

Tabla 4.5: Caso de Uso: Modificar

Caso de Uso: Modificar (C. U. 05)	
Descripción:	Permite modificar registros según la instancia deseada
Actor:	Usuario registrado como Administrador
Poscondición:	El usuario deberá logrearse para ingresar al sistema.
Flujo Normal:	Seleccionar ítem del menú. Seleccionar registro.
Flujo Alternativo:	Regresarán los datos a los controles en un recuadro flotante. Pulsar botón modificar Modificar los datos Pulsar el botón Guardar.

Tabla 4.6: Caso de Uso: Eliminar

Caso de Uso: Eliminar (C. U. 06)	
Descripción:	Permite eliminar registros según la instancia deseada
Actor:	Usuario registrado como Administrador
Poscondición:	El usuario deberá logrearse para ingresar al sistema.
Flujo Normal:	Seleccionar ítem del menú. Seleccionar registro.
Flujo Alternativo:	Regresarán los datos a los controles en un recuadro flotante. Pulsar botón eliminar Envía un mensaje de Que desea borrar Pulsar el botón Aceptar Se borrar de la base de datos

Tabla 4.7: Caso de Uso: Consultar

Caso de Uso: Consultar (C. U. 07)	
Descripción:	Permite ingresar nuevos registros según la instancia deseada
Actor:	Usuario registrado como Administrador
Poscondicion:	El usuario deberá logrearse para ingresar al sistema.
Flujo Normal:	Seleccionar del menú el ítem deseado Seleccionar el parámetro por el cual se va a consultar. Ingresar criterio de búsqueda y presionar el botón Consultar.
Flujo Alternativo:	Consultar: El sistema comprueba la validez de los datos Si los datos son correctos consulta y muestra los datos. Si son incorrectos envía un mensaje al usuario con el error.

Tabla 4.8: Caso de Uso: Configuración de una Cuenta

Caso de Uso: Configuración de una Cuenta		
Descripción:	Información del actor que utiliza la aplicación.	
Actor:	Administrador, Usuario (Usuarios Registrados)	
Precondición:	Los actores deben tener una sesión activa.	
Poscondición:	Se presenta la información actualizada del actor que inicio sesión.	
Flujo Principal:	Evento Actor	Evento Sistema
	1. Buscar la opción Configurar Cuenta.	1. Muestra en la pantalla el menú en donde está la configuración de la cuenta.
	2. Modificar la información requerida.	2. Cargar la pagina de la configuración de la cuenta del actor.
	3. Enviar datos mediante una petición.	3. Mostrar mensaje de validación de datos.
		4. Verificar la información recibida.
		5. Re direccionar a la página de configuración de cuenta.
Alternativa 1:	1. Buscar opción de inicio.	
	2. Enviar petición web.	2. Re direccionar a la página principal del sistema.
Alternativa 2:	1. Buscar opción actualizar foto.	
	2. Enviar petición web.	2. Abrir el explorador del sistema de búsqueda de archivos.
		3. Cargar el archivo seleccionado.
		4. Actualizar fotografía del actor.

Diagrama de Caso de Uso: Funciones Administrador – Usuario_Docente

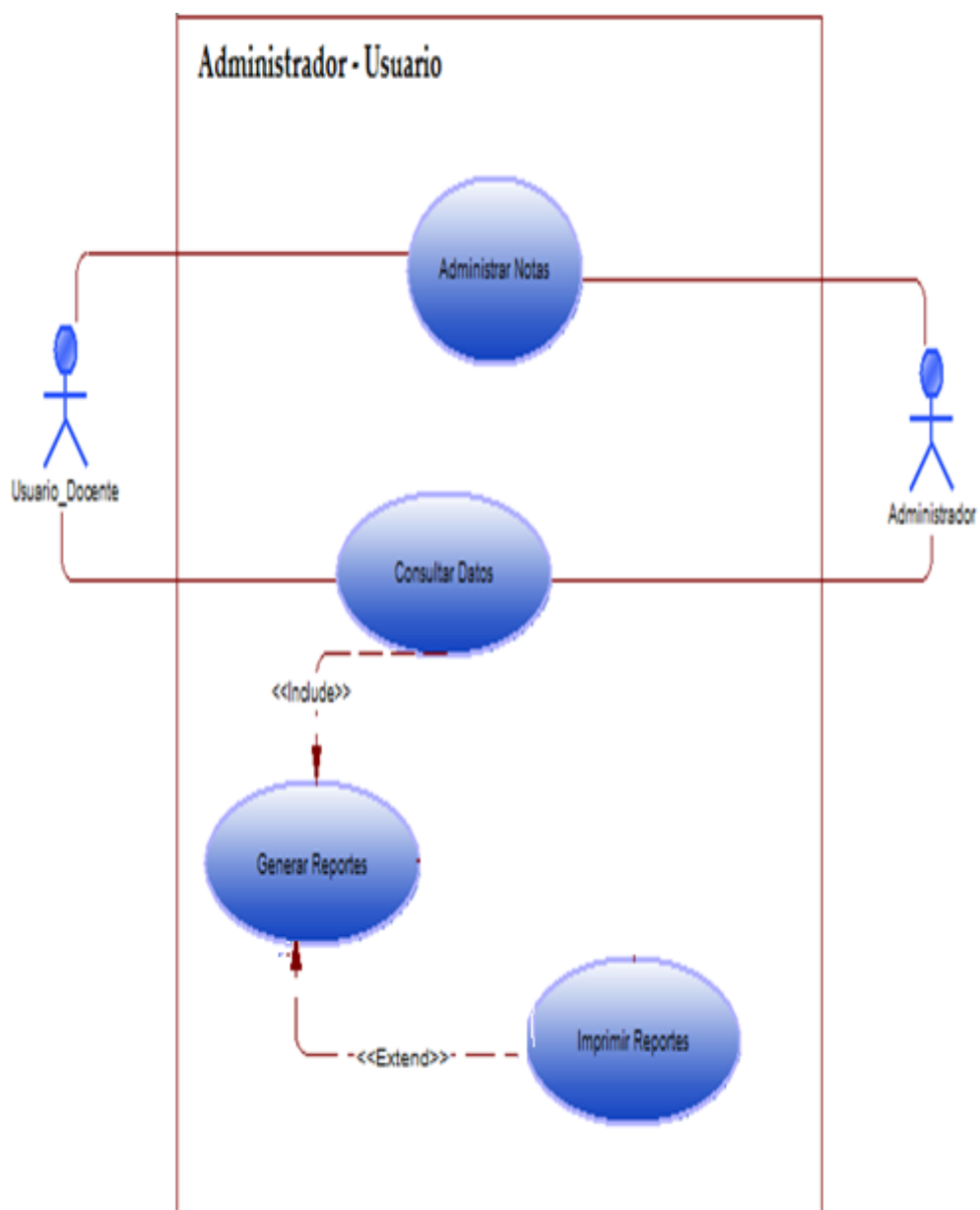


Figura 4.2: Diagrama de Caso de Uso Administrador – Usuario_Docente

Descripción de los Casos de Uso

Tabla 4.9: Caso de Uso: Administrar Notas

Caso de Uso: Administrar Notas													
Descripción:	Permitirá la administración de los procesos relacionados con las notas.												
Actor:	Administrador, Usuario												
Precondición:	El actor debe tener una sesión activa.												
Poscondición:	Se presenta la información actualizada.												
Presunción:	Base de datos Disponible.												
Flujo Principal:	<table border="1"> <thead> <tr> <th>Evento Actor</th> <th>Evento Sistema</th> </tr> </thead> <tbody> <tr> <td>1. Buscar la opción según la operación que se vaya a realizar.</td> <td>1. Muestra en la pantalla el menú de Administración de Notas.</td> </tr> <tr> <td>2. Seleccionar la opción a realizarse.</td> <td>2. Cargar las interfaces de las operaciones a realizar.</td> </tr> <tr> <td>3. Seleccionar actividad a realizar.</td> <td>3. Mostrar las interfaces de las opciones que se pueden realizar en el proceso.</td> </tr> <tr> <td>4. Enviar datos mediante una petición.</td> <td>4. Verificar la información recibida.</td> </tr> <tr> <td></td> <td>5. Actualizar la información modificada.</td> </tr> </tbody> </table>	Evento Actor	Evento Sistema	1. Buscar la opción según la operación que se vaya a realizar.	1. Muestra en la pantalla el menú de Administración de Notas.	2. Seleccionar la opción a realizarse.	2. Cargar las interfaces de las operaciones a realizar.	3. Seleccionar actividad a realizar.	3. Mostrar las interfaces de las opciones que se pueden realizar en el proceso.	4. Enviar datos mediante una petición.	4. Verificar la información recibida.		5. Actualizar la información modificada.
	Evento Actor	Evento Sistema											
	1. Buscar la opción según la operación que se vaya a realizar.	1. Muestra en la pantalla el menú de Administración de Notas.											
	2. Seleccionar la opción a realizarse.	2. Cargar las interfaces de las operaciones a realizar.											
	3. Seleccionar actividad a realizar.	3. Mostrar las interfaces de las opciones que se pueden realizar en el proceso.											
	4. Enviar datos mediante una petición.	4. Verificar la información recibida.											
	5. Actualizar la información modificada.												
Alternativa 1:	1. Buscar opción de inicio.												
	2. Enviar petición web.	2. Re direccionar a la página principal del sistema.											

Tabla 4.10: Caso de Uso: Consultar Datos

Caso de Uso: Consultar Datos		
Descripción:	Permitirá la Consulta de Notas requerida.	
Actor:	Administrador, Usuario	
Precondición:	Los actores deben tener una sesión activa.	
Poscondición:	Se presenta la información requerida por el actor.	
Presunción:	Base de datos Disponible.	
Flujo Principal:	Evento Actor	Evento Sistema
	1. Buscar la opción según la operación que se vaya a realizar.	1. Muestra en la pantalla el menú de Administración de Notas.
	2. Seleccionar la opción a realizarse.	2. Cargar las interfaces de las operaciones a realizar.
	3. Seleccionar actividad a realizar.	3. Mostrar las interfaces de las opciones que se pueden realizar en el proceso.
	4. Enviar datos mediante una petición.	4. Verificar la información recibida.
		5. Mostrar información requerida.
Alternativa 1:	1. Buscar opción de inicio.	
	2. Enviar petición web.	2. Re direccionar a la página principal del sistema.
Alternativa 2:	1. Buscar opción Generar Reporte.	
	2. Enviar petición web.	2. Mostrar el Reporte Generado.
	3. Buscar opción Imprimir Reporte	
	4. Enviar petición web.	4. Imprimir Reporte de Notas.

4.1.3. Diagramas de Secuencia

Se muestra la interacción de los objetos que componen un sistema de forma temporal.

Diagramas de Secuencia MVC: Ingreso al Sistema

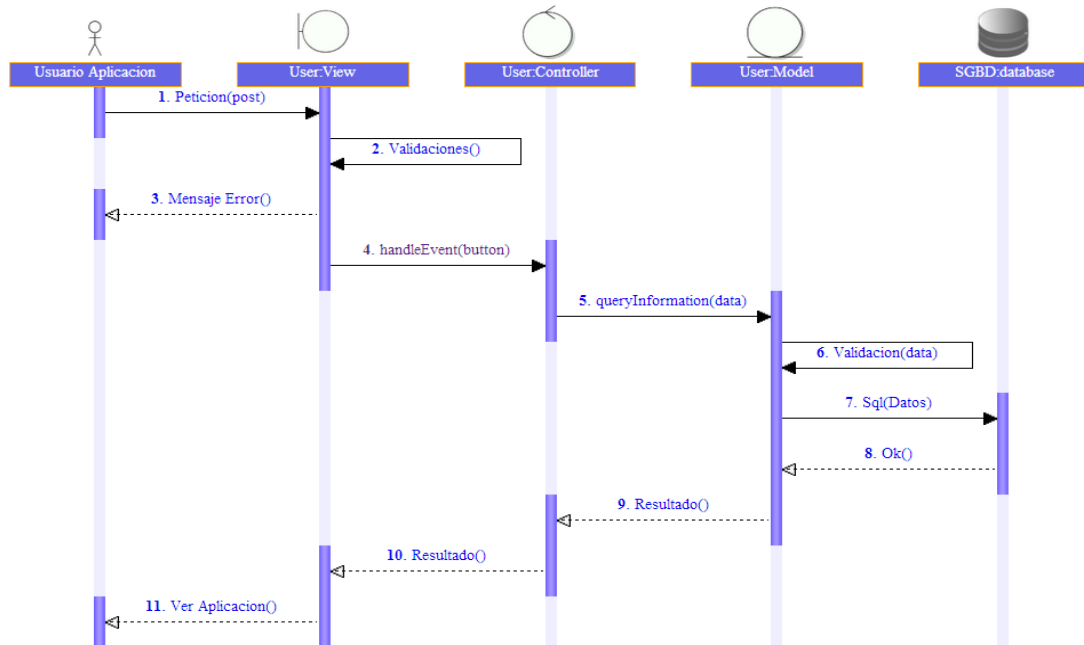


Figura 4.3: Diagrama de Secuencia MVC: Ingreso al Sistema

Descripción: Diagrama de Secuencia: Ingreso al Sistema

Tabla 4.11: Descripción MVC Ingreso al Sistema

Descripción: Diagrama de Secuencia: Ingreso al Sistema
1. Usuario Aplicación call User:View->Petición(post)
2. User: View call User: View->Validaciones()
3. Usuario Aplicación receive from User:View->Mensaje Error()
4. User: View call User: Controller->handleEvent(button)
5. User: Controller call User:Model->queryInformation(data)
6. User:Model call User:Model->Validación(data)
7. User:Model call SGBD:database->Sql(Datos)
8. User:Model receive from SGBD:database->Ok()
9. User: Controller receive from User:Model->Resultado()
10. User: View receive from User: Controller->Resultado()
11. Usuario Aplicación receive from User:View->Ver Aplicación()

Diagramas de Secuencia MVC: Configuración de Cuentas

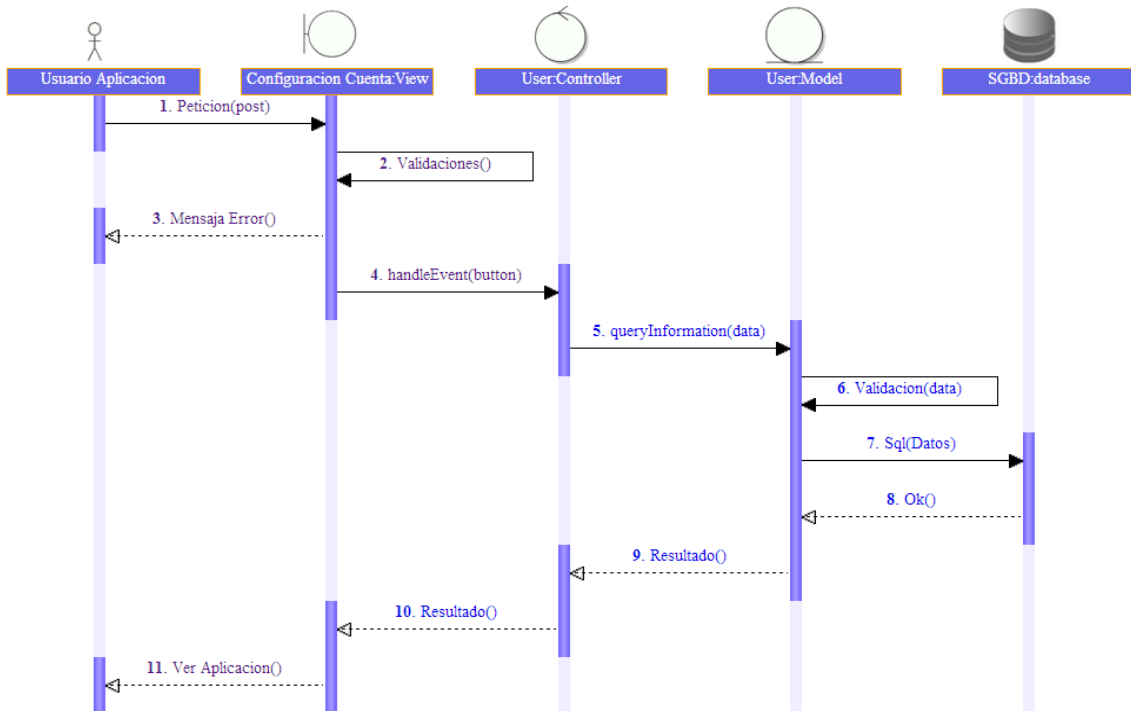


Figura 4.5: Diagrama de Secuencia MVC: Configuración de Cuentas

Descripción: Diagrama de Secuencia: Configuración de Cuentas

Tabla 4.12: Descripción MVC Configuración de Cuentas

Descripción: Diagrama de Secuencia: Configuración de Cuentas
1. Usuario Aplicación call Configuración Cuenta:View->Petición(post)
2. Configuración Cuenta:View call Configuración Cuenta:View->Validaciones()
3. Usuario Aplicación receive from Configuración Cuenta:View->Mensaje Error()
4. Configuración Cuenta:View call User: Controller->handleEvent(button)
5. User: Controller call User:Model->queryInformation(data)
6. User:Model call User:Model->Validación(data)
7. User:Model call SGBD:database->Sql(Datos)
8. User:Model receive from SGBD:database->Ok()
9. User: Controller receive from User:Model->Resultado()
10. Configuración Cuenta:View receive from User:Controller->Resultado()
11. Usuario Aplicación receive from Configuración Cuenta:View->Ver Aplicación()

Diagramas de Secuencia MVC: Crear Grupos de Usuarios

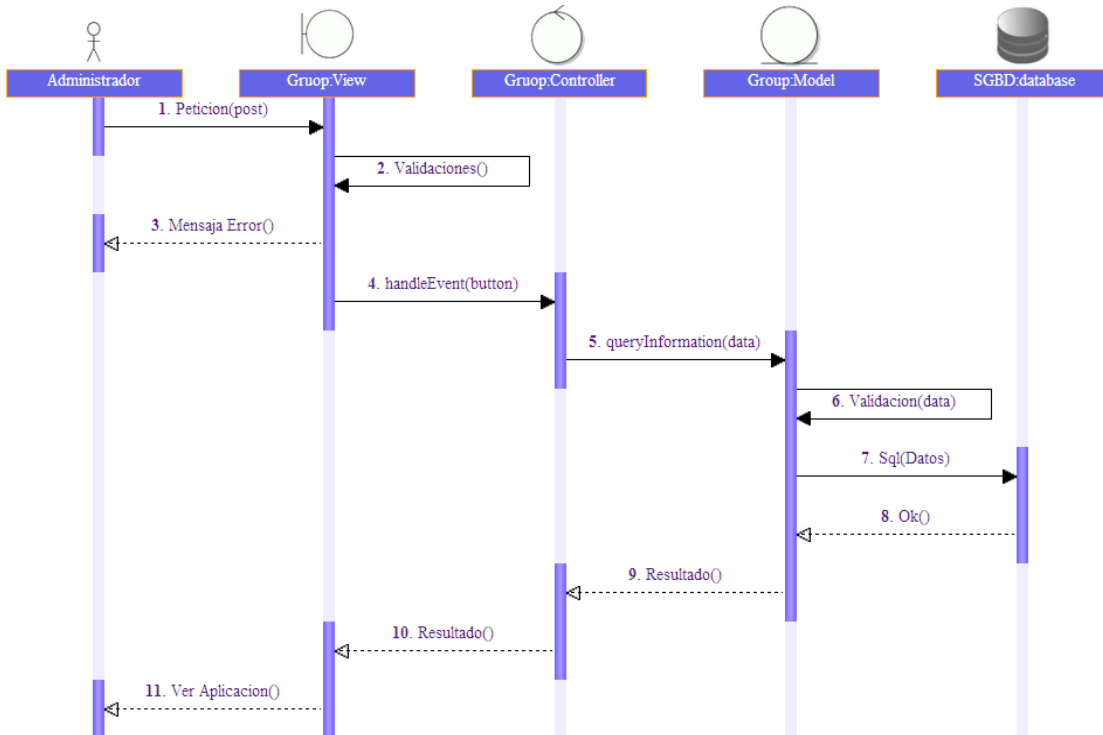


Figura 4.6: Diagrama de Secuencia MVC: Crear Grupos de Usuarios

Descripción: Diagrama de Secuencia: Crear Grupos de Usuarios

Tabla 4.13: Descripción MVC Crear Grupos de Usuarios

Descripción: Diagrama de Secuencia: Crear Grupos de Usuarios
1. Administrador call Grupo:View->Peticion(post)
2. Grupo:View call Grupo:View->Validaciones()
3. Administrador receive from Grupo:View->Mensaje Error()
4. Grupo:View call Grupo:Controller->handleEvent(button)
5. Grupo:Controller call Group:Model->queryInformation(data)
6. Group:Model call Group:Model->Validacion(data)
7. Group:Model call SGBD:database->Sql(Datos)
8. Group:Model receive from SGBD:database->Ok()
9. Grupo:Controller receive from Group:Model->Resultado()
10. Grupo:View receive from Grupo:Controller->Resultado()
11. Administrador receive from Grupo:View->Ver Aplicación()

Diagramas de Secuencia MVC: Registrar Usuarios

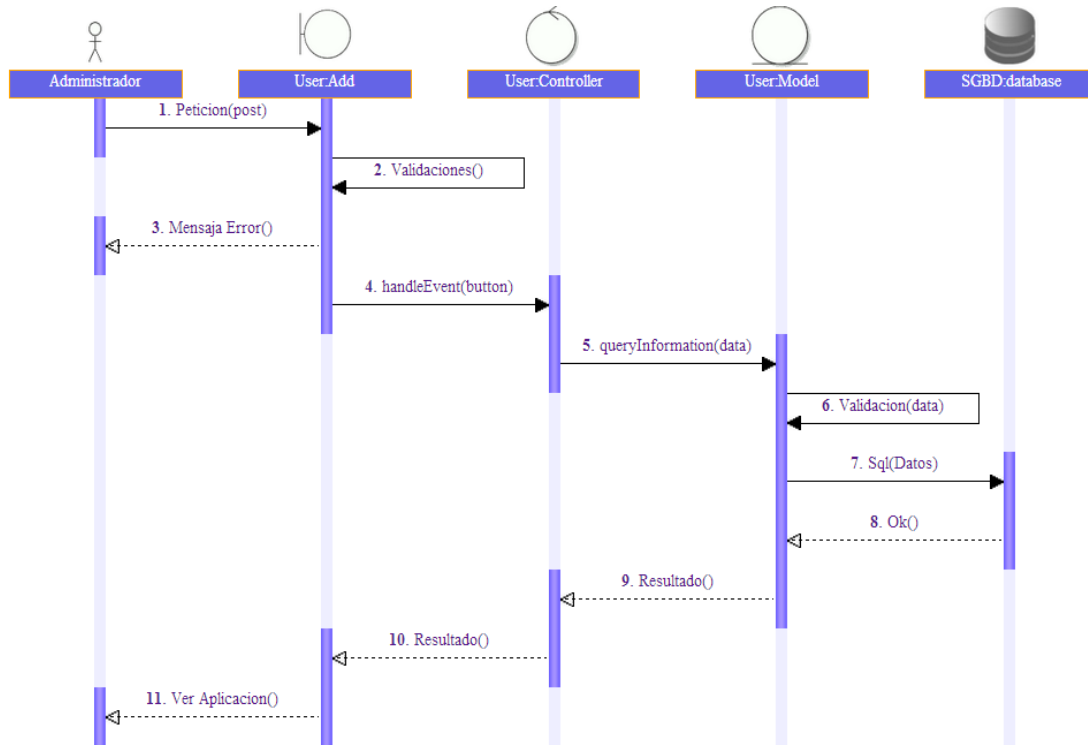


Figura 4.7: Diagrama de Secuencia MVC: Registrar Usuarios

Descripción: Diagrama de Secuencia: Registrar Usuarios

Tabla 4.14: Descripción MVC Registrar Usuarios

Descripción: Diagrama de Secuencia: Registrar Usuarios
1. Administrador call User:Add->Peticion(post)
2. User:Add call User:Add->Validaciones()
3. Administrador receive from User:Add->Mensaje Error()
4. User:Add call User: Controller->handleEvent(button)
5. User: Controller call User:Model->queryInformation(data)
6. User:Model call User:Model->Validación(data)
7. User:Model call SGBD:database->Sql(Datos)
8. User:Model receive from SGBD:database->Ok()
9. User: Controller receive from User:Model->Resultado()
10. User:Add receive from User: Controller->Resultado()
11. Administrador receive from User:Add->Ver Aplicación()

Diagramas de Secuencia MVC: Administrar Aplicación

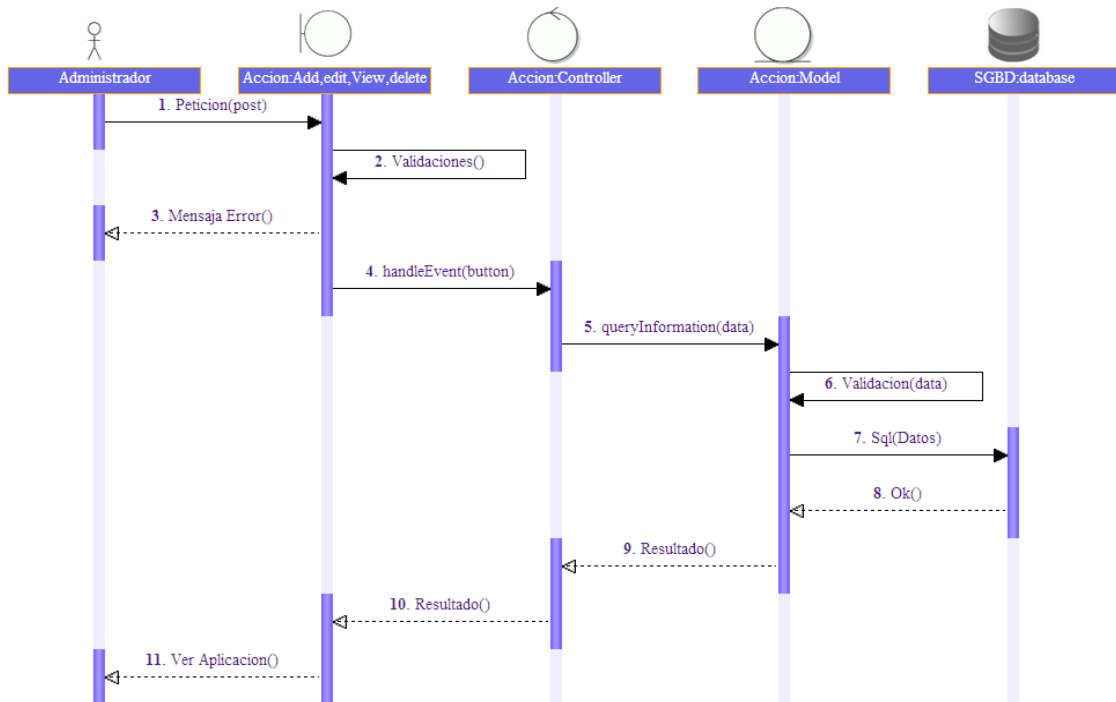


Figura 4.8: Diagrama de Secuencia MVC: Administrar Aplicación

Descripción: Diagrama de Secuencia: Administrar Información de la Aplicación

Tabla 4.15: Descripción MVC Administrar Aplicación

Descripción: Diagrama de Secuencia: Administrar Información de la Aplicación
Administrador call Accion:Add,edit,View,delete->Peticion(post)
Accion:Add,edit,View,delete call Accion:Add,edit,View,delete->Validaciones()
Administrador receive from Accion:Add,edit,View,delete->Mensaje Error()
Accion:Add,edit,View,delete call Accion:Controller->handleEvent(button)
Accion:Controller call Accion:Model->queryInformation(data)
Accion:Model call Accion:Model->Validación(data)
Accion:Model call SGBD:database->Sql(Datos)
Accion:Model receive from SGBD:database->Ok()
Accion:Controller receive from Accion:Model->Resultado()
Accion:Add,edit,View,delete receive from Accion:Controller->Resultado()
Administrador receive from Accion:Add,edit,View,delete->Ver Aplicación()

Diagramas de Secuencia MVC: Administrar Notas

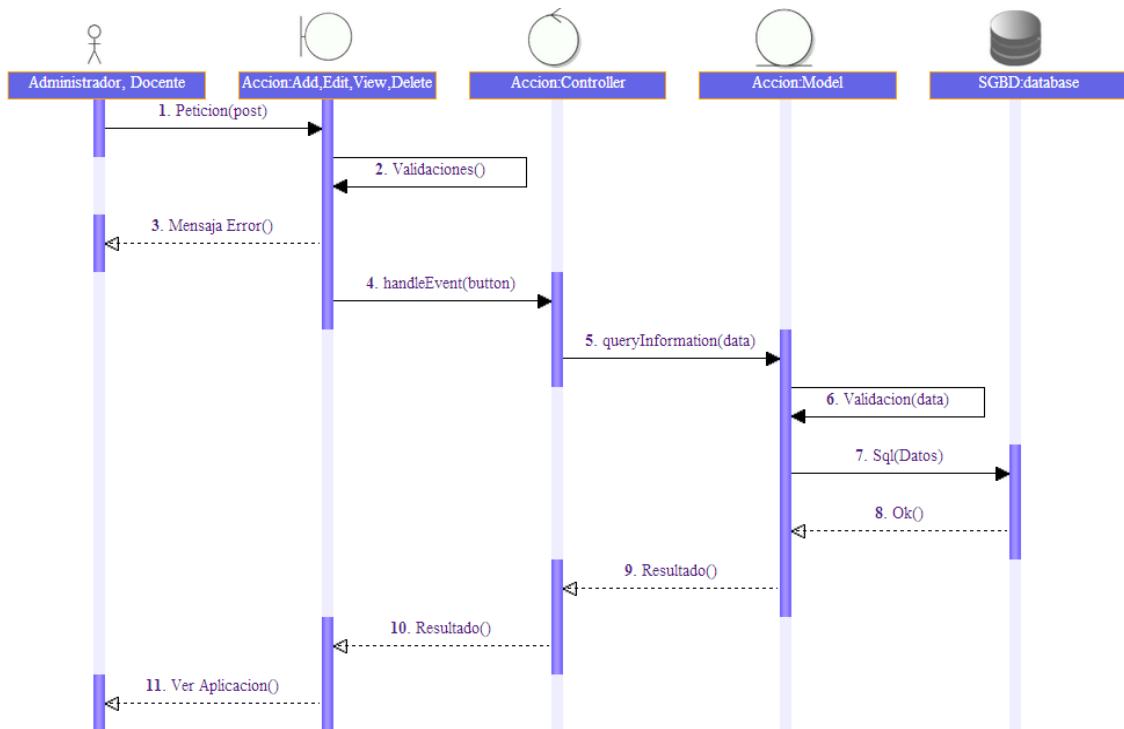


Figura 4.9: Diagrama de Secuencia MVC: Administrar Notas

Descripción: Diagrama de Secuencia: Administrar Notas

Tabla 4.16: Descripción MVC Administrar Notas

Descripción: Diagrama de Secuencia: Administrar Notas
Administrador, Docente call Accion:Add,Edit,View,Delete->Peticion(post)
Accion:Add,Edit,View,Delete call Accion:Add,Edit,View,Delete->Validaciones()
Administrador, Docente receive from Accion:Add,Edit,View,Delete->Mensaje Error()
Accion:Add,Edit,View,Delete call Accion:Controller->handleEvent(button)
Accion:Controller call Accion:Model->queryInformation(data)
Accion:Model call Accion:Model->Validación(data)
Accion:Model call SGBD:database->Sql(Datos)
Accion:Model receive from SGBD:database->Ok()
Accion:Controller receive from Accion:Model->Resultado()
Accion:Add,Edit,View,Delete receive from Accion:Controller->Resultado()
Administrador, Docente receive from Accion:Add,Edit,View,Delete->Ver Aplicación()

Diagramas de Secuencia MVC: Salir de la Aplicación

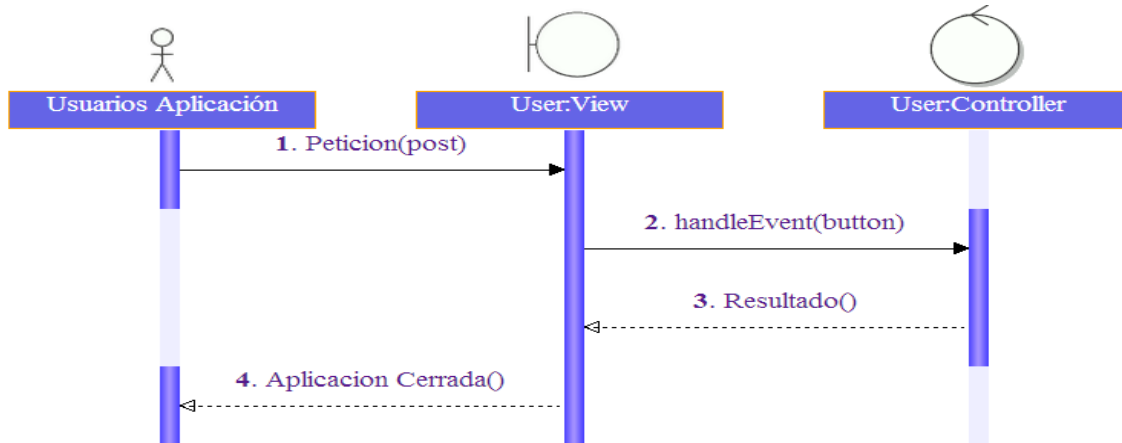


Figura 4.4: Diagrama de Secuencia MVC: Salir de la Aplicación

Descripción: Diagrama de Secuencia: Salir de la Aplicación

Tabla 4.17: Descripción MVC Salir de la Aplicación

Descripción: Diagrama de Secuencia: Salir de la Aplicación
1. Usuarios Aplicación call User:View->Petición(post)
2. User: View call User: Controller->handleEvent(button)
3. User: View receive from User: Controller->Resultado()
4. Usuarios Aplicación receive from User:View->Aplicación Cerrada()

4.2. Diseño

Es el proceso de utilizar la información recolectada en la etapa de análisis al diseño del producto.

4.2.1. Diseño de Base de Datos

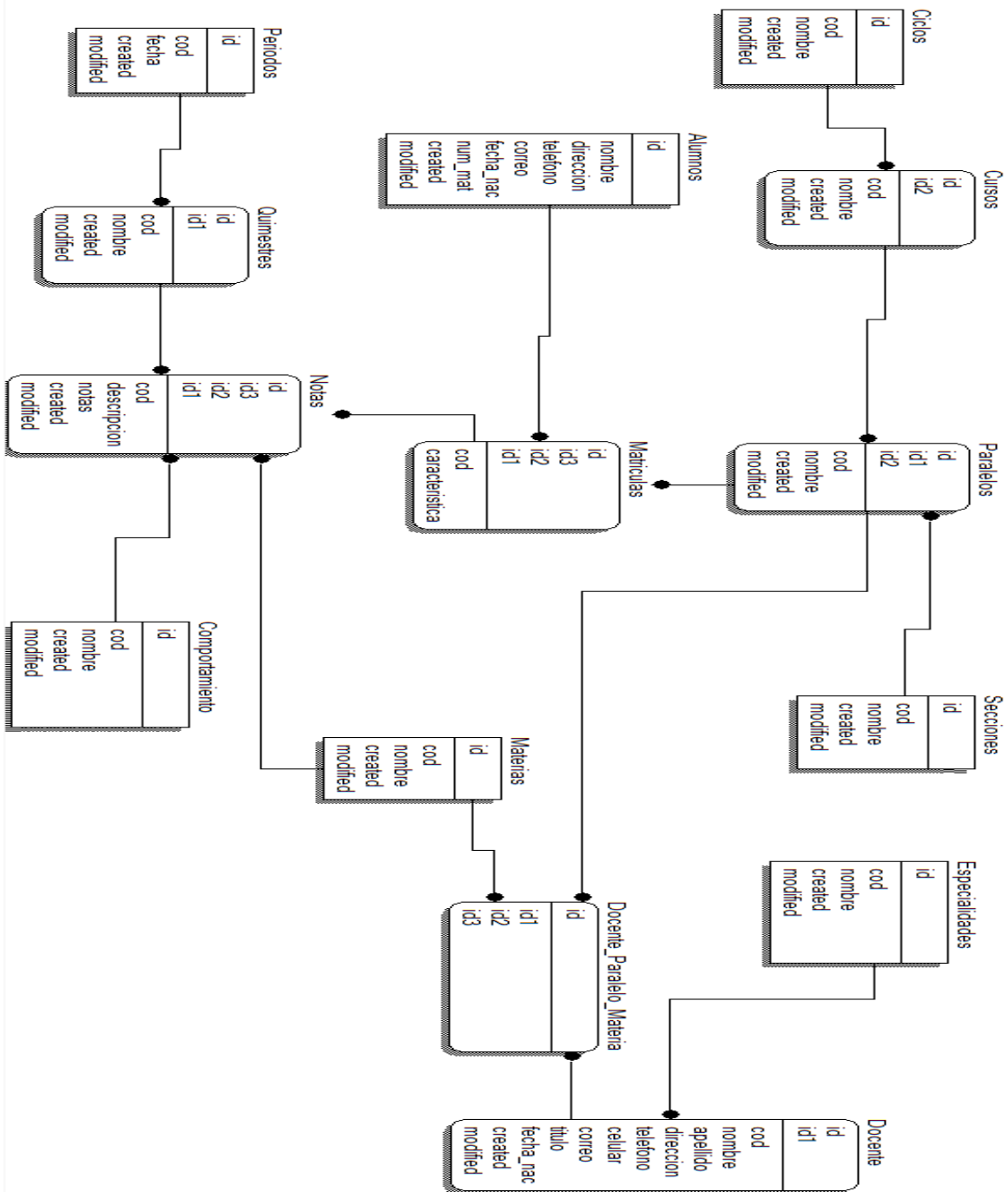


Figura 4.10: Diseño de Base de Datos

4.2.2. Diccionario de Datos

Descripción de cada uno de los campos de las distintas tablas de la Base de Datos utilizada en el Sistema.

Tabla 4.18: Descripción de la Tabla Ciclos

TABLA: Ciclos			
CAMPO	TIPO	DESCRIPCIÓN	CLAVE
Id	int	Id de la tabla Ciclos	PK
Nombre	char(50)	Nombre del Ciclo	
Created	date	Fecha de Creación del Ciclo	
modified	date	Fecha de Modificación del Ciclo	
Estado	int	Estado del Ciclo	

Tabla 4.19: Descripción de la Tabla Cursos

TABLA: Cursos			
CAMPO	TIPO	DESCRIPCIÓN	CLAVE
Id	int	Id de la tabla Cursos	PK
ciclo_id	int	Id de la tabla Cursos_FK	FK
Nombre	char(50)	Nombre del Cursos	
Created	date	Fecha de Creación del Curso	
Modified	date	Fecha de Modificación del Curso	
Estado	int	Estado del Curso	

Tabla 4.20: Descripción de la Tabla Especialidades

TABLA: Especialidades			
CAMPO	TIPO	DESCRIPCIÓN	CLAVE
id	int	Id de la tabla Especialidades	PK
nombre	char(50)	Nombre de la Especialidad	
created	date	Fecha de Creación de la Especialidad	
modified	date	Fecha de Modificación de la Especialidad	
estado	int	Estado de la Especialidad	

Tabla 4.21: Descripción de la Tabla Secciones

TABLA: Secciones			
CAMPO	TIPO	DESCRIPCIÓN	CLAVE
id	int	Id de la tabla Secciones	PK
nombre	char(50)	Nombre de la Sección	
created	date	Fecha de Creación de la Sección	
modified	date	Fecha de Modificación de la Sección	
estado	int	Estado de la Sección	

Tabla 4.22: Descripción de la Tabla Paralelos

TABLA: Paralelos			
CAMPO	TIPO	DESCRIPCIÓN	CLAVE
Id	int	Id de la tabla Paralelos	PK
cursos_id	int	Id del Curso	FK1
secciones_id	int	Id de la Sección	FK2
especialidades_id	int	Id de la Especialidad	FK3
Nombre	char(50)	Nombre del Paralelo	
Cantidad	int	Cantidad de Paralelos	
Created	date	Fecha de Creación de los Paralelos	
Modified	date	Fecha de Modificación de los Paralelos	
Estado	int	Estado del Paralelo	

Tabla 4.23: Descripción de la Tabla Periodos

TABLA: Periodos			
CAMPO	TIPO	DESCRIPCIÓN	CLAVE
id	int	Id de la tabla Periodos	PK
nombre	char(50)	Nombre del Periodo	
fechainicio	date	Fecha de Inicio del Periodo	
fechafin	date	Fecha de Fin del Periodo	
created	date	Fecha de Creación del Periodo	
modified	date	Fecha de Modificación del Periodo	
estado	int	Estado del Periodo	

Tabla 4.24: Descripción de la Tabla Materias

TABLA: Materias			
CAMPO	TIPO	DESCRIPCIÓN	CLAVE
id	Int	Id de la tabla Materias	PK
nombre	char(50)	Nombre de la Materia	
created	date	Fecha de Creación de la Materia	
modified	date	Fecha de Modificación de la Materia	
estado	Int	Estado de la Materia	

Tabla 4.25: Descripción de la Tabla Alumnos

TABLA: Alumnos			
CAMPO	TIPO	DESCRIPCIÓN	CLAVE
Id	int	Id de la tabla Alumnos	PK
Codigohistorial	char(15)	Codigo del Historial	
Nombre	char(50)	Nombre del Alumno	
Apellido	char(50)	Apellido del Alumno	
Dirección	char(50)	Dirección del Alumno	
Celular	char(10)	Celular del Alumno	
Teléfono	char(9)	Teléfono del Alumno	
Correo	char(50)	Correo del Alumno	
Genero	char(50)	Genero del Alumno	
fecha_naci	date	Fecha de Nacimiento del Alumno	
Created	date	Fecha de Creación del Alumno	
Modified	date	Fecha de Modificación del Alumno	
Estado	int	Estado del Alumno	

Tabla 4.26: Descripción de la Tabla Comportamientos

TABLA: Comportamientos			
CAMPO	TIPO	DESCRIPCIÓN	CLAVE
id	Int	Id de la tabla Comportamientos	PK
nombre	char(1)	Nombre del Comportamiento	
descripción	char(50)	Descripción del Comportamiento	
created	date	Fecha de Creación del Comportamiento	
modified	date	Fecha de Modificación del Comportamiento	
estado	Int	Estado del Comportamiento	

Tabla 4.27: Descripción de la Tabla Matriculas

TABLA: Matriculas			
CAMPO	TIPO	DESCRIPCIÓN	CLAVE
Id	int	Id de la tabla Matriculas	PK
Numero	int	Numero de la Matricula	
Código	char(15)	Codigo de la Matricula	
alumno_id	int	Id del Alumno	FK1
paralelo_id	int	Id del Paralelo	FK2
pediido_id	int	Id del Periodo	FK3
representante_id	int	Id del Representante	FK4
Características	char(20)	Características de la Matricula	
Created	date	Fecha de Creación de la Matricula	
Modified	date	Fecha de Modificación de la Matricula	
Estado	int	Estado de la Matricula	

Tabla 4.28: Descripción de la Tabla Docente

TABLA: Docentes			
CAMPO	TIPO	DESCRIPCIÓN	CLAVE
id	int	Id de la tabla Docentes	PK
cedula	char(10)	Cedula del Docente	
nombre	char(50)	Nombre del Docente	
apellido	char(20)	Apellido del Docente	
dirección	char(50)	Dirección del Docente	
celular	char(10)	Celular del Docente	
teléfono	char(10)	Teléfono del Docente	
correo	char(50)	Correo del Docente	
titulo	char(50)	Titulo del Docente	
fecha_naci	date	Fecha de Nacimiento de Docente	
area_id	int	Id del Área	FK1
created	date	Fecha de Creación del Docente	
modified	date	Fecha de Modificación del Docente	

Tabla 4.29: Descripción de la Tabla Notas

TABLA: Notas			
CAMPO	TIPO	DESCRIPCIÓN	CLAVE
Id	int	Id de la tabla Notas	PK
comportamiento_id	int	Id del Comportamiento	FK4
notasdetalles_id	int	Id del Detalle de las Notas	FK2
matricula_id	int	Id de la Matricula	FK3
quimestre_id	int	Id del Quimestres	FK5
docenteparalelomateria_id	int	Id del Detalle entre Docente - Paralelo - Materia	FK1
Notas	decimal	Notas de los Estudiantes	
Created	date	Fecha de Creación de las Notas	
Modified	date	Fecha de Modificación de las Notas	
Estado	int	Estado de las Notas	

Tabla 4.30: Descripción de la Tabla Quimestres

TABLA: Quimestres			
CAMPO	TIPO	DESCRIPCIÓN	CLAVE
id	int	Id de la tabla Quimestres	PK
nombre	char(50)	Nombre del Quimestre	
fechainicio	date	Fecha de Inicio del Quimestre	
fechafin	date	Fecha de Fin del Quimestre	
created	date	Fecha de Creación del Quimestre	
modified	date	Fecha de Modificación del Quimestre	
estado	int	Estado del Quimestre	

4.2.3. Diseño de la Interfaz de Usuario

Con el diseño de la interfaz gráfica de usuario se detalla cómo está situada la información y los controles manejados en cada página de la aplicación web, dando una vista previa de las mismas antes de ser implementadas con una herramienta de desarrollo.

Interfaz de Inicio de Sesión

Autentica a los usuarios que desean ingresar al sistema, ingresando el nombre de usuario y su contraseña, mismos que tendrán acceso a la aplicación dependiendo de los roles proporcionados por el Administrador del Sistema.

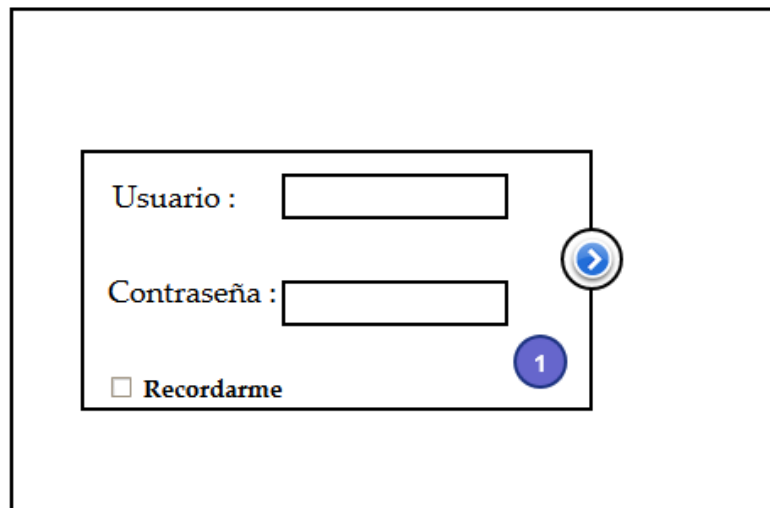


Figura 4.11: Interfaz de Ingreso al Sistema

1. **Área de Inicio de Sesión.-** Permite la autenticación de los usuarios para el acceso a la aplicación web, mediante el ingreso de un usuario y su respectiva contraseña.

Interfaz de la Página Maestra

Se la utiliza como base para las diferentes paginas de la aplicación web, además se muestra los controles en común de todos los módulos del sistema.

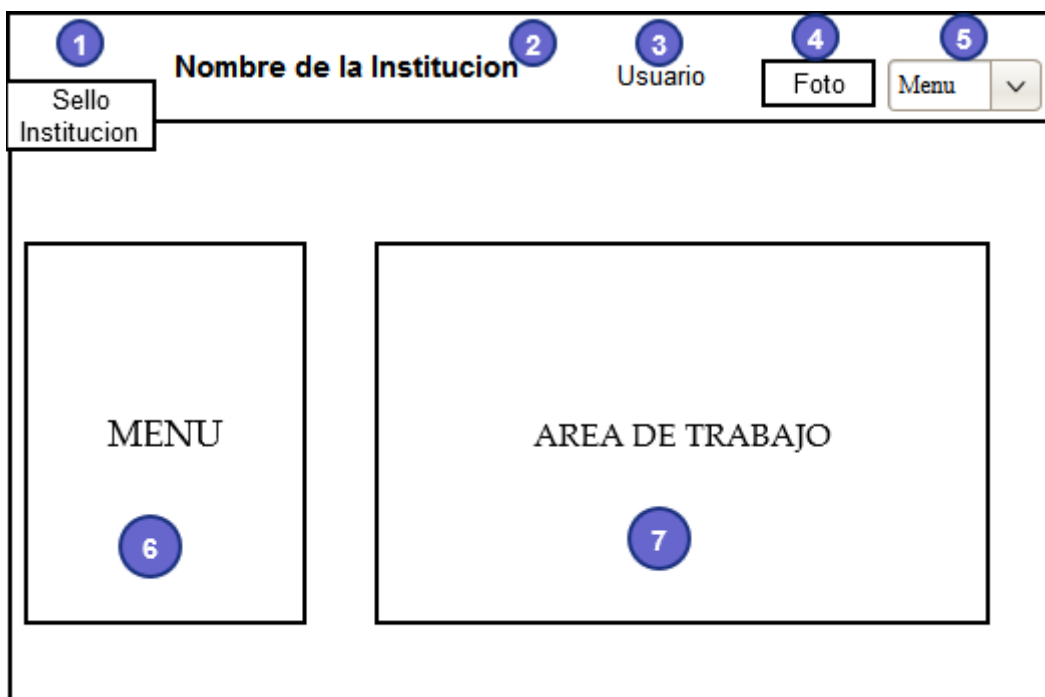


Figura 4.12: Interfaz de la Pagina Maestra

1. **Sello de la Institución.-** Se muestra el respectivo sello de la institución a la que va dirigido el sistema.
2. **Nombre de la Institución.-** Muestra el nombre de la institución.
3. **Usuario.-** Muestra el nombre de usuario activo.
4. **Foto.-** Muestra la respectiva foto del usuario que ha ingresado al sistema.
5. **Menú Configuración.-** Muestra las respectivas opciones del menú configuración de la cuenta del usuario y la opción de salir de la aplicación.
6. **Menú Principal.-** Consta de cinco ítems principales como son el de Usuarios y Grupos, Administración Cursos, Distributivo, Matriculas, Notas.

Cada ítem principal se divide en sub-categorías las mismas que son descritas a continuación.

Usuarios y Grupos:

- Grupos
- Usuarios

Administración Cursos:

- Ciclos
- Secciones
- Especialidades
- Cursos

Distributivo:

- Periodos
- Áreas
- Docentes
- Materias
- Distribución

Matriculas:

- Estudiantes
- Representantes
- Matricula

Notas:

- Quimestres
- Detalle Notas
- Comportamiento
- Registro Notas

7. Área de Trabajo.- Espacio destinado para la gestión de la información. La información mostrada dependerá directamente de las opciones seleccionadas en el menú.

Interfaz de Ingresos y Modificación

Se la utiliza para el ingreso de nueva información así como también para modificar la información existente en la base de datos.

El diagrama muestra una interfaz de usuario con los siguientes elementos:

- 1. Titulo:** Un campo de texto en la parte superior de la ventana.
- 2. Campos de Ingreso:** Tres campos de texto etiquetados como 'Campo1 :', uno debajo del otro.
- 3. Botón Guardar:** Un botón rectangular ubicado debajo del primer campo de texto.
- 4. Botón Cancelar:** Un botón rectangular ubicado debajo del segundo campo de texto.

Figura 4.13: Interfaz de Ingresos Y Modificación

- 1. Titulo.-** Muestra el titulo de la ventana.
- 2. Campos de Ingreso.-** Permite el ingreso de la información requerida mediante el uso del teclado.
- 3. Botón Guardar.-** Guarda la información ingresada en los campos de ingreso, siempre y cuando no haya errores en la información
- 4. Botón Cancelar.-** Cancela el proceso del ingreso de la información.

Interfaz de Eliminación

Se la utiliza para la eliminación de un campo existente en la base de datos seleccionado por el usuario, siempre y cuando tenga los permisos necesarios para realizar esta operación.

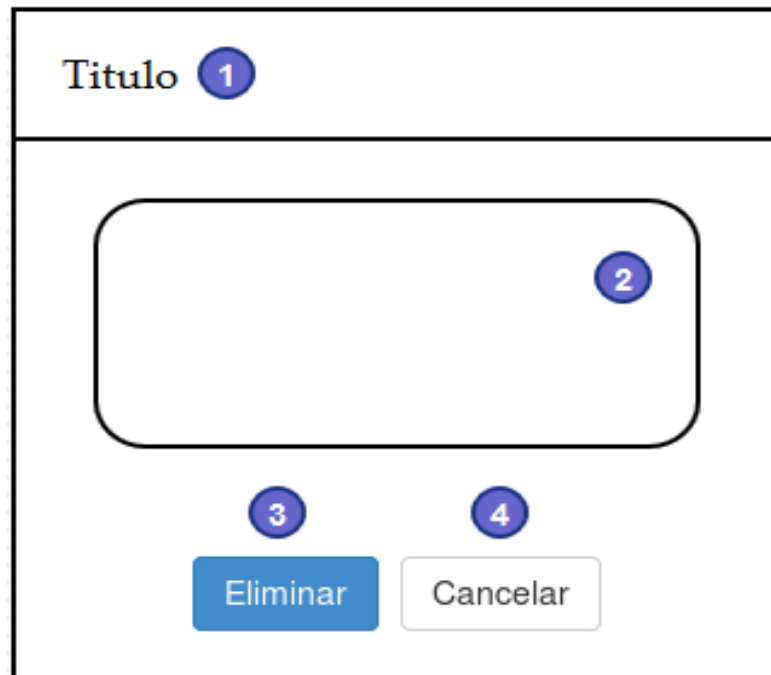


Figura 4.14: Interfaz de Eliminación

1. **Título.-** Muestra el título de la ventana.
2. **Área de Información.-** Muestra la información a eliminarse.
3. **Botón Eliminar.-** Eliminar la información
4. **Botón Cancelar.-** Cancela el proceso de eliminación.

Interfaz de Información

Se la utiliza para la eliminación de un campo existente en la base de datos seleccionado por el usuario, siempre y cuando tenga los permisos necesarios para realizar esta operación.

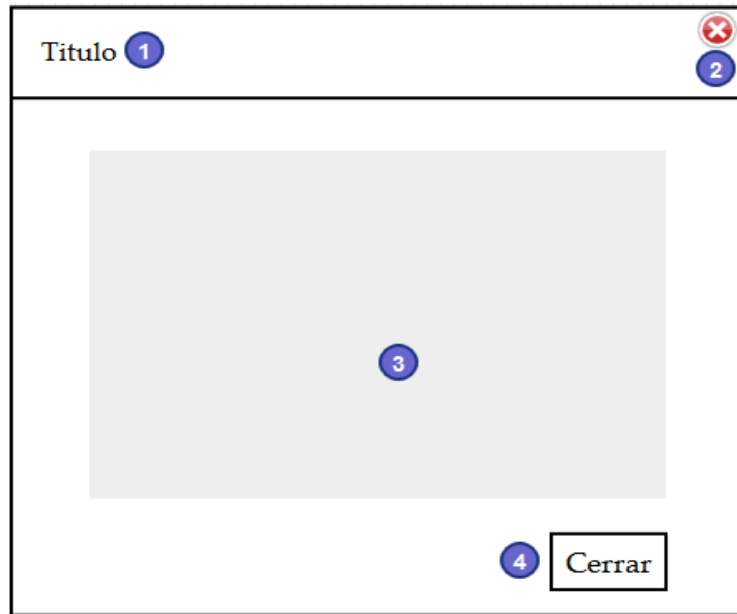


Figura 4.15: Interfaz de Información

1. **Título.-** Título de la ventana.
2. **Opción de Cerrar el Modal.-** Permite cerrar la ventana de información.
3. **Área de Trabajo.-** Muestra la descripción de la información del elemento seleccionado.
4. **Botón Cerrar.-** Cierra la ventana de información.

Interfaz de Grupos

Modal de Administración de Grupos, se mostrara todas las opciones de la misma al usuario que inicio sesión con sus respectivos permisos o restricciones para la manipulación de las operaciones.

Interfaz de Usuarios y Grupos

Modal de Administración de Usuario y Grupos, se mostrara todas las opciones de la misma al usuario que inicio sesión con sus respectivos permisos o restricciones para la manipulación de las operaciones.

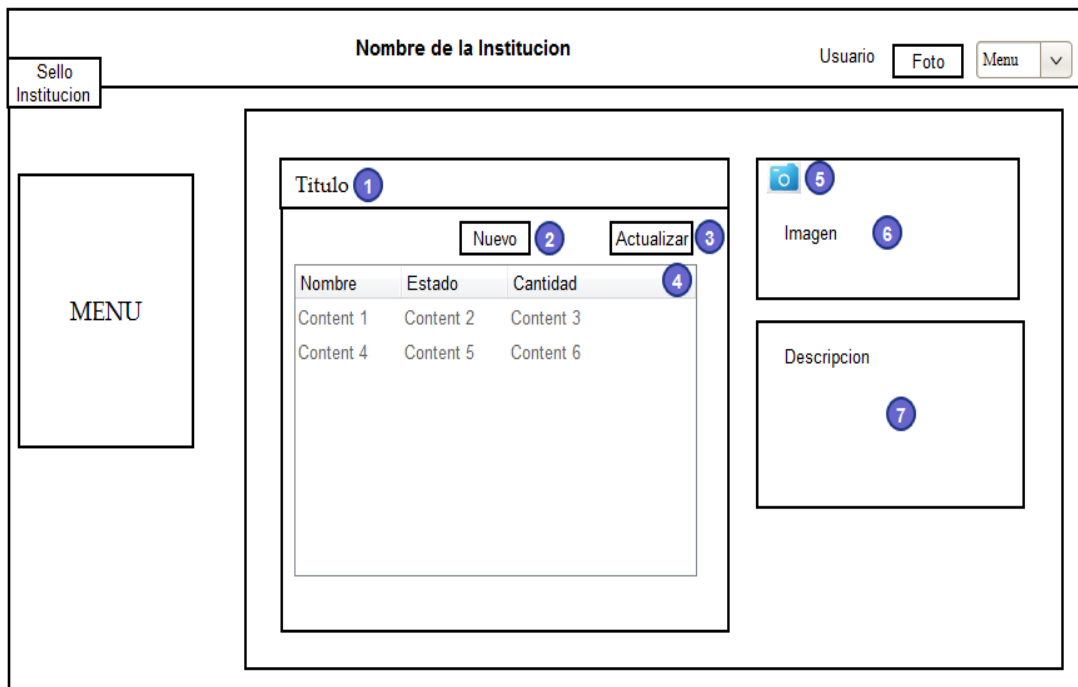


Figura 4.16: Interfaz de Usuarios y Grupos

1. **Título.-** Título de la ventana.
2. **Botón Nuevo.-** Permite agregar un nuevo Grupo.
3. **Botón Actualizar.-** Actualiza el contenido mostrado si se ha realizado algún cambio.
4. **Área de Contenido.-** Muestra el contenido de la tabla Grupos.
5. **Botón de Subir Imagen.-** Permite subir una nueva imagen para el usuario.
6. **Imagen.-** Muestra la imagen del usuario.
7. **Área de Descripción.-** Describe las características del usuario seleccionado.

Interfaz de Cursos

Modal de Administración de Cursos, se mostrara todas las opciones de la misma al usuario que inicio sesión con sus respectivos permisos o restricciones para la manipulación de las operaciones.

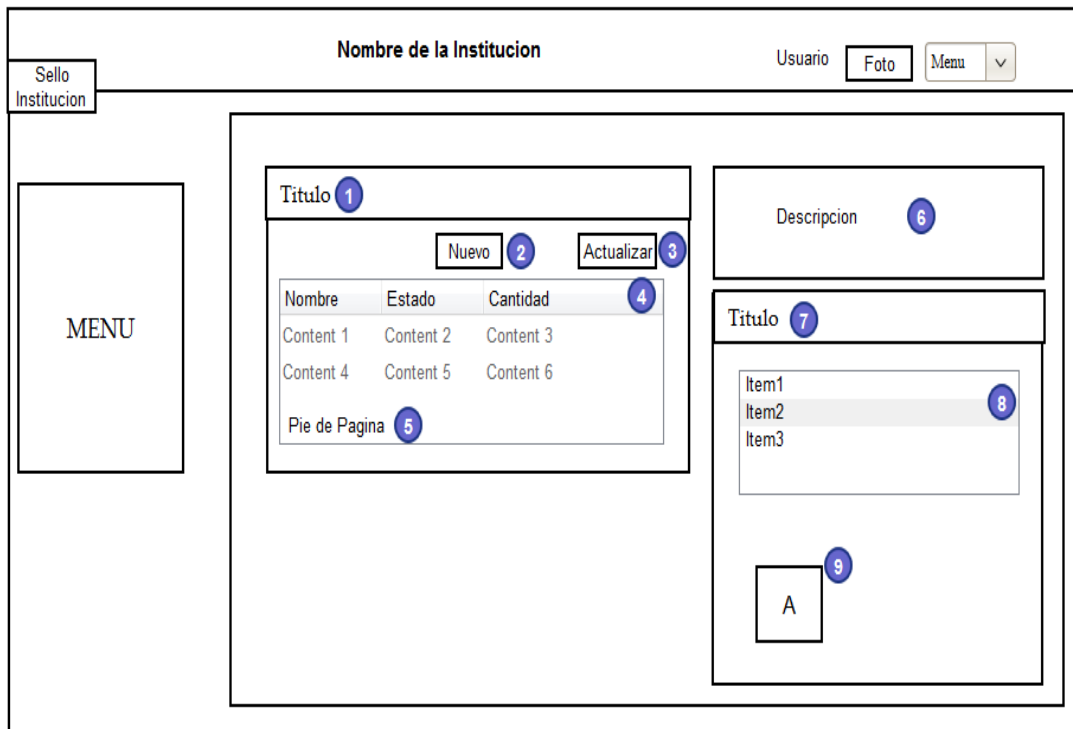


Figura 4.17: Interfaz de Administración de Cursos

1. **Titulo.-** Titulo de la ventana.
2. **Botón Nuevo.-** Permite agregar un nuevo Grupo.
3. **Botón Actualizar.-** Actualiza el contenido mostrado si se ha realizado algún cambio.
4. **Área de Contenido.-** Muestra el contenido de la tabla Cursos.
5. **Pie de Página.-** Muestra información de la cantidad de cursos existentes, además de mostrar el número de páginas.
6. **Área de Descripción.-** Describe las características del curso seleccionado.
7. **Titulo.-** Titulo de la ventana.
8. **Área de Contenido.-** Muestra el contenido de la tabla Paralelos que tiene relación con la tabla Cursos.
9. **Características del Contenido Seleccionado.-** Muestra información del área de contenido seleccionado.

Interfaz de Secciones

Modal de Administración de Secciones, se mostrara todas las opciones de la misma al usuario que inicio sesión con sus respectivos permisos o restricciones para la manipulación de las operaciones.

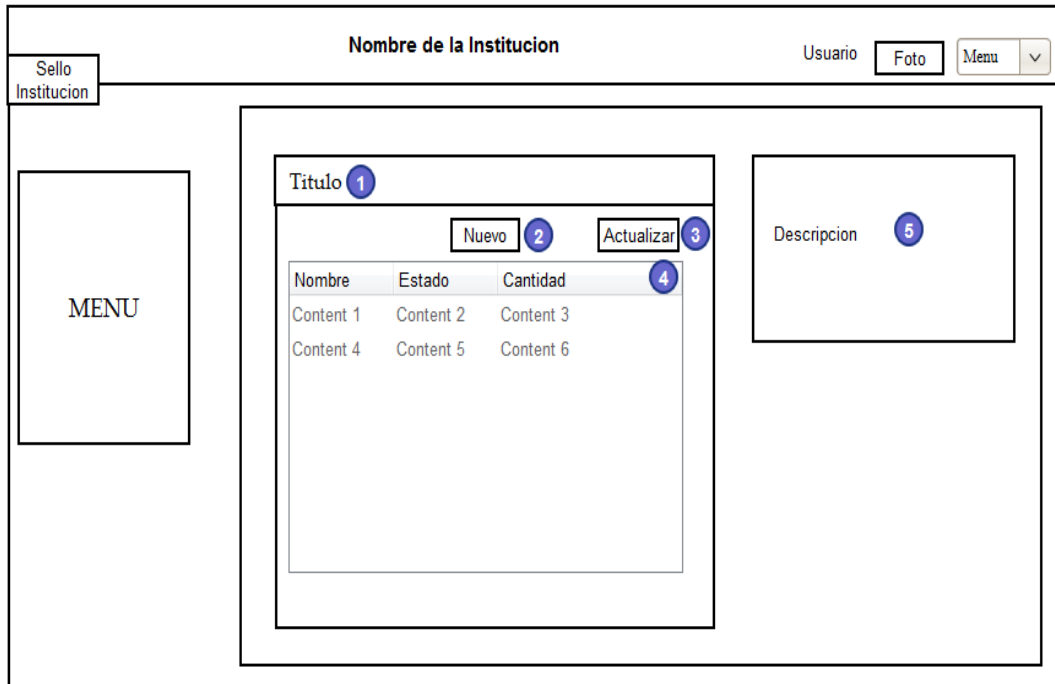


Figura 4.18: Interfaz de Secciones

1. **Titulo.-** Titulo de la ventana.
 2. **Botón Nuevo.-** Permite agregar una nueva Sección.
 3. **Botón Actualizar.-** Actualiza el contenido mostrado si se ha realizado algún cambio.
 4. **Área de Contenido.-** Muestra el contenido de la tabla Secciones.
 5. **Área de Descripción.-** Describe las características de la Sección seleccionada.
- ✓ **El modelo de esta interfaz se utiliza también para los siguientes modales de administración:**
- Ciclos,
 - Periodos,
 - Materias.
 - Especialidades

Interfaz de Áreas

Modal de Administración de Áreas, se mostrara todas las opciones de la misma al usuario que inicio sesión con sus respectivos permisos o restricciones para la manipulación de las operaciones.

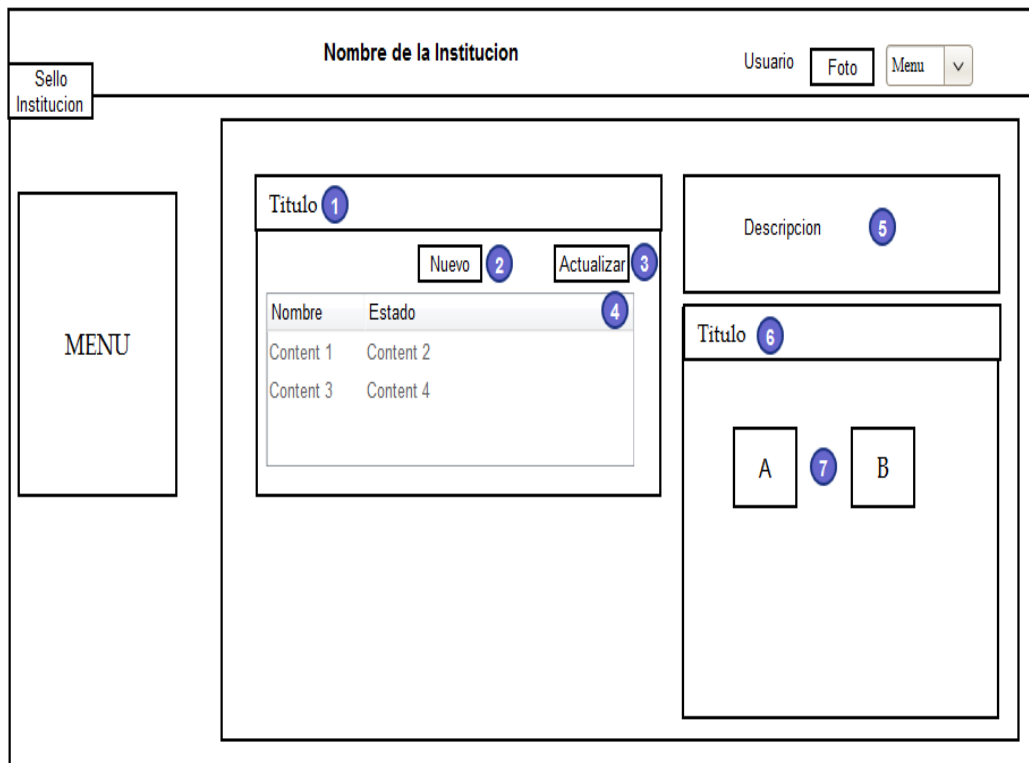


Figura 4.19: Interfaz de Áreas

1. **Título.-** Título de la ventana.
2. **Botón Nuevo.-** Permite agregar una nueva Área.
3. **Botón Actualizar.-** Actualiza el contenido mostrado si se ha realizado algún cambio.
4. **Área de Contenido.-** Muestra el contenido de la tabla Áreas.
5. **Área de Descripción.-** Describe las características del Área seleccionada.
6. **Título.-** Título de la ventana.
7. **Área de Contenido.-** Muestra el contenido de la tabla Docentes que tiene relación con la tabla Áreas.

Interfaz de Docentes

Modal de Administración de Docentes, se mostrara todas las opciones de la misma al usuario que inicio sesión con sus respectivos permisos o restricciones para la manipulación de las operaciones.

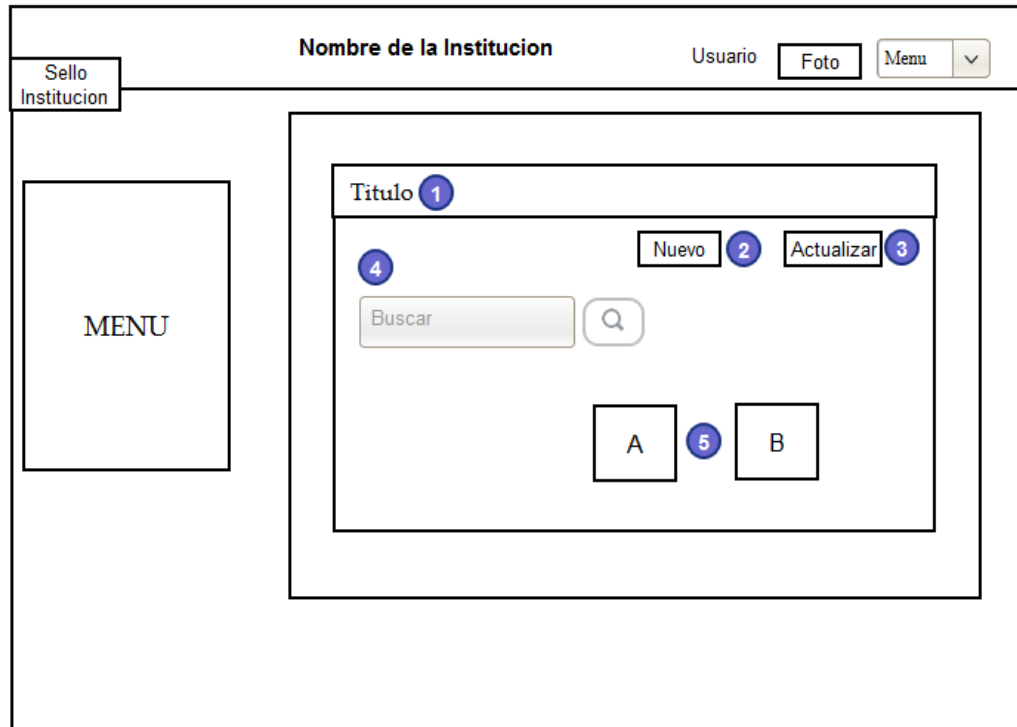


Figura 4.20: Interfaz de Docentes

1. **Título.-** Título de la ventana.
2. **Botón Nuevo.-** Permite agregar un nuevo Docente.
3. **Botón Actualizar.-** Actualiza el contenido mostrado si se ha realizado algún cambio.
4. **Sección de Búsqueda.-** Permite la búsqueda inteligente del docente deseado por el usuario que inicio sesión.
5. **Área de Contenido.-** Muestra el contenido de la tabla Docentes.

4.3. Implementación

Tiene como objetivo producir la aplicación de acuerdo a las especificaciones de diseño arquitectónico y detallado elaboradas en los procesos de diseño.

4.3.1. Implementación de la Interfaz de Grafica de Usuario

Se presenta a continuación algunas de las páginas de la aplicación web implementadas, las mismas que fueron bosquejadas en la etapa de diseño, se especifica también cada sección de dichas páginas. En el manual de usuario en la sección de anexos, se incluye y se explica la totalidad de las páginas así como su respectiva funcionalidad.

Página de Inicio de Sesión

Autentica a los usuarios que desean ingresar al sistema, ingresando el nombre de usuario y su contraseña, mismos que tendrán acceso a la aplicación dependiendo de los roles proporcionados por el Administrador del Sistema.

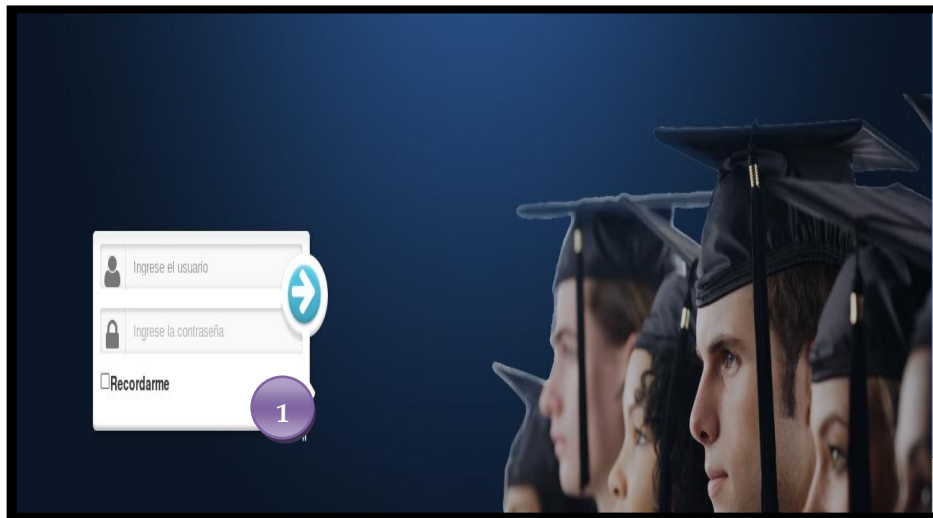


Figura 4.21: Pantalla de Inicio de Sesión

1. **Área de Inicio de Sesión.-** Permite la autenticación de los usuarios para el acceso a la aplicación web, mediante el ingreso de un usuario y su respectiva contraseña.

Página Principal

Se la utiliza como base para las diferentes paginas de la aplicación web, además se muestra los controles en común de todos los módulos del sistema.



Figura 4.22: Pantalla Principal

1. **Sello de la Institución.-** Se muestra el respectivo sello de la institución a la que va dirigido el sistema.
2. **Nombre de la Institución.-** Muestra el nombre de la institución.
3. **Usuario.-** Muestra el nombre de usuario activo.
4. **Foto.-** Muestra la respectiva foto del usuario que ha ingresado al sistema.
5. **Menú Configuración.-** Muestra las respectivas opciones del menú configuración de la cuenta del usuario y la opción de salir de la aplicación.

6. Menú Principal.- Consta de cinco ítems principales como son el de Usuarios y Grupos, Administración Cursos, Distributivo, Matriculas, Notas.

Cada ítem principal se divide en sub-categorías las mismas que son descritas a continuación.

Usuarios y Grupos:

- Grupos
- Usuarios

Administración Cursos:

- Ciclos
- Secciones
- Especialidades
- Cursos

Distributivo:

- Periodos
- Áreas
- Docentes
- Materias
- Distribución

Notas:

- Quimestres
- Detalle Notas
- Comportamiento
- Registro Notas

7. Área de Trabajo.- Espacio destinado para la gestión de la información. La información mostrada dependerá directamente de las opciones seleccionadas en el menú.

4.3.2. Implementación de la Arquitectura de Programación

La arquitectura utilizada en la programación para el desarrollo de la aplicación web sigue el patrón de diseño MVC, el mismo que separa el código en tres capas:

- **El Modelo:** Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.
- **La Vista:** Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho 'modelo' la información que debe representar como salida.
- **El Controlador:** Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta de 'modelo' (por ejemplo, desplazamiento o scroll por un documento o por los diferentes registros de una base de datos), por tanto se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo' (véase Middleware).

A continuación se presenta la codificación de varias funciones del modulo de Administración, en especial lo referente a Cursos, que sirve como referencia para las demás funciones ya desarrolladas.

Función para mostrar la Vista de los Cursos

```
public function view($id = null) {
    if (!$this->Curso->exists($id) {
        throw new NotFoundException(__('Invalid curso'));
    }

    $options = array('conditions' =>
        array('Curso.' . $this->Curso->primaryKey => $id,
        ));
    $this->set('curso', $this->Curso->find('first', $options));

    $docentesCedula=$this->Curso->Paralelo->find('all', $options);

    $cedulas=Set::extract($docentesCedula, '{n}.Paralelo.Especialidades_id');

    $this->loadModel('Especialidades');

    $this->set('Especialidades', $this->Especialidades->find('all',array('conditions' => array(
        array( "Especialidades.id" => $cedulas)
        ))));
}
```

Figura 4.23: Función para mostrar la Vista de los Cursos

Función para mostrar la Vista de los Paralelos

```
public function viewparalelo($id = null) {
    if (!$this->Curso->exists($id) {
        throw new NotFoundException(__('Invalid curso'));
    }

    $options = array('conditions' =>
        array('Curso.' . $this->Curso->primaryKey => $id,
        ));
    $this->set('curso', $this->Curso->find('first', $options));

    $docentesCedula=$this->Curso->Paralelo->find('all', $options);

    $cedulas=Set::extract($docentesCedula, '{n}.Paralelo.Especialidadess_id');

    $this->loadModel('Especialidades');

    $this->set('Especialidades', $this->Especialidadess->find('all',array('conditions' => array(
        array( "Especialidades.id" => $cedulas)
        ))));
}
```

Figura 4.24: Función para mostrar la Vista de los Paralelos

Función para mostrar la Vista de los Paralelos por Curso

```
public function viewparalelosporcursos($id = null) {  
  
    $final=$this->Curso->Paralelo->find('all', array(  
        'conditions' => array(  
            array( "Paralelo.curso_id" => $id)  
        ),  
        'recursive'=>0  
    ));  
  
    $this->set('paralelos', $final);  
}
```

Figura 4.25: Función para mostrar la Vista de los Paralelos por Curso

Función para el Método Añadir un Curso

```
public function add() {  
    $a = array('estado' => FALSE, 'datos'=>null, 'html'=>null );  
  
    if ($this->request->is('post')) {  
        $this->autoRender = FALSE;  
  
        $this->Curso->set($this->data);  
  
        if ($this->Curso->validates() {  
            $this->Curso->create();  
            if ($this->Curso->save($this->request->data)) {  
                $a['estado'] = true;  
                $a['html']=$this->mensajes("Curso ".$this->request->data['Curso']['nombre']." Guardado", "correcto");  
            } else {  
                $a['html'] = $this->mensajes("No se guardo el Curso, intente nuevamente", "error");  
            }  
        }  
        else {  
            $errors = $this->Curso->validationErrors;  
  
            $a['html'] = $this->mensajes($errors["nombre"][0], "advertencia");  
  
            print_r($errors);  
        }  
  
        return json_encode($a);  
    }  
  
    $condiciones=array('conditions' => array('estado' => 1));  
    $ciclos = $this->Curso->Ciclo->find('list', $condiciones);  
    $this->set(compact('ciclos'));
```

Figura 4.26: Función para el Método Añadir un Curso

Función para el Método Editar un Curso

```
public function edit($id = null) {
    $a = array('estado' => FALSE, 'datos'=>null, 'html'=>null );
    if (!$this->Curso->existe($id)) {
        throw new NotFoundException(__('Invalid Curso'));
    }
    if ($this->request->is(array('post', 'put')) {

        $this->Curso->set($this->data);
        $this->autoRender = FALSE;
        if ($this->Curso->validates()) {

            if ($this->Curso->save($this->request->data)) {
                $a['estado'] = true;
                $a['html']=$this->mensajes("Curso ".$this->request->data['Curso']['nombre']." Guardado", "correcto");
            } else {
                $a['html'] = $this->mensajes("No se guardo el Grupo, intente nuevamente", "error");
            }
        }
        else{
            $errors = $this->Curso->validationErrors;
            $a['html'] = $this->mensajes($errors["nombre"][0], "advertencia");
        }

        return json_encode($a);

    } else {
        $options = array('conditions' => array('Curso.' . $this->Curso->primaryKey => $id));
        $this->request->data = $this->Curso->find('first', $options);
        $this->set('curso', $this->request->data);
    }
    $condiciones=array('conditions' => array('estado' => 1));
    $ciclos = $this->Curso->Ciclo->find('list',$condiciones);

    $this->set(compact('ciclos'));
}
```

Figura 4.27: Función para el Método Editar un Curso

Función para el Método Eliminar un Curso

```
public function delete($id = null) {
    $a = array('estado' => FALSE, 'datos'=>null, 'html'=>null );

    $this->Curso->id = $id;
    $options = array('conditions' => array('Curso.' . $this->Curso->primaryKey => $id));
    $this->request->data = $this->Curso->find('first', $options);

    if (!$this->Curso->exists()) {
        throw new NotFoundException(__('Invalid Curso'));
    }
    if($this->request->is(array('post', 'delete'))){

        $this->autoRender = false;

        if(count($this->request->data['Paralelo'])==0)
        {
            if ($this->Curso->delete()) {
                $a['estado'] = true;
                $a['html']=$this->mensajes("Curso ".$this->request->data['Curso']['nombre']. " Eliminado", "correcto");
            } else {
                $a['html'] = $this->mensajes("No se pudo eliminar al Curso ".$this->request->data['Curso']['nombre'], "error");
            }
        }
        else {
            $a['html'] = $this->mensajes("No se pudo eliminar al Curso tiene paralelos", "advertencia");
        }

        return json_encode($a);
    }
}
```

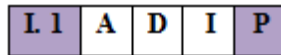
Figura 4.28: Función para el Método Eliminar un Curso

Ejemplo de Triggers Almacenados en la Base de Datos

Función para el Trigger Eliminar Alumno

```
DELIMITER |
CREATE TRIGGER eliminacionDeAlumno BEFORE delete ON alumnos
FOR EACH ROW
BEGIN
    declare idMatricula int;
    Set idMatricula=(SELECT distinct id FROM matriculas where alumno_id=old.id);
    DELETE FROM notas WHERE matricula_id=idMatricula;
    DELETE FROM `shescolastico`.`matriculas` where id=idMatricula;
END
|
```

Figura 4.29: Función para el Trigger Eliminar Alumno



4.4. Pruebas de Funcionamiento

Finalizada la implementación de la aplicación web, se procedió a realizar las pruebas respectivas, con el fin de detectar posibles errores o fallos en la implementación del software. A continuación se describen las diferentes pruebas realizadas a la Aplicación Web implementada.

4.4.1. Pruebas de Caja Blanca

Se basan en el conocimiento de la lógica interna del código del sistema. Las pruebas contemplan los distintos caminos que se pueden generar gracias a las estructuras condicionales, a los distintos estados del mismo, etc.

Las pruebas de caja blanca se derivan de casos de prueba que aseguren que durante la prueba se han ejecutado todas las sentencias del programa al menos una vez y que se ejercitan todas las condiciones lógicas.

Casos de prueba que permita:

- Garantizar que se ejecutan al menos una vez todos los caminos independientes de cada módulo.
- Se utilizan las decisiones en su parte verdadera y en su parte falsa.
- Se ejecuten todos los bucles en sus límites.
- Se utilizan todas las estructuras de datos internas.

Prueba del Camino Básico

Es una técnica de prueba de Caja Blanca que permite obtener una medida de complejidad lógica para generar un conjunto básico de caminos que se ejecutan por lo menos una vez durante la ejecución del programa.

Notación del Grafo de Flujo

El flujo de control de un programa se representa mediante lo que se llama Grafo de Flujo que representa el flujo de control lógico, el mismo que está formado por nodos los mismos que representan una o más sentencias de código fuente o LDP sin bifurcación.

Para realizar las pruebas de caja blanca utilizaremos la técnica del Camino Básico utilizando el código del programa: **Validación de los Dígitos de una Cedula.**

Diagrama de Flujo de la Validación de la Cedula

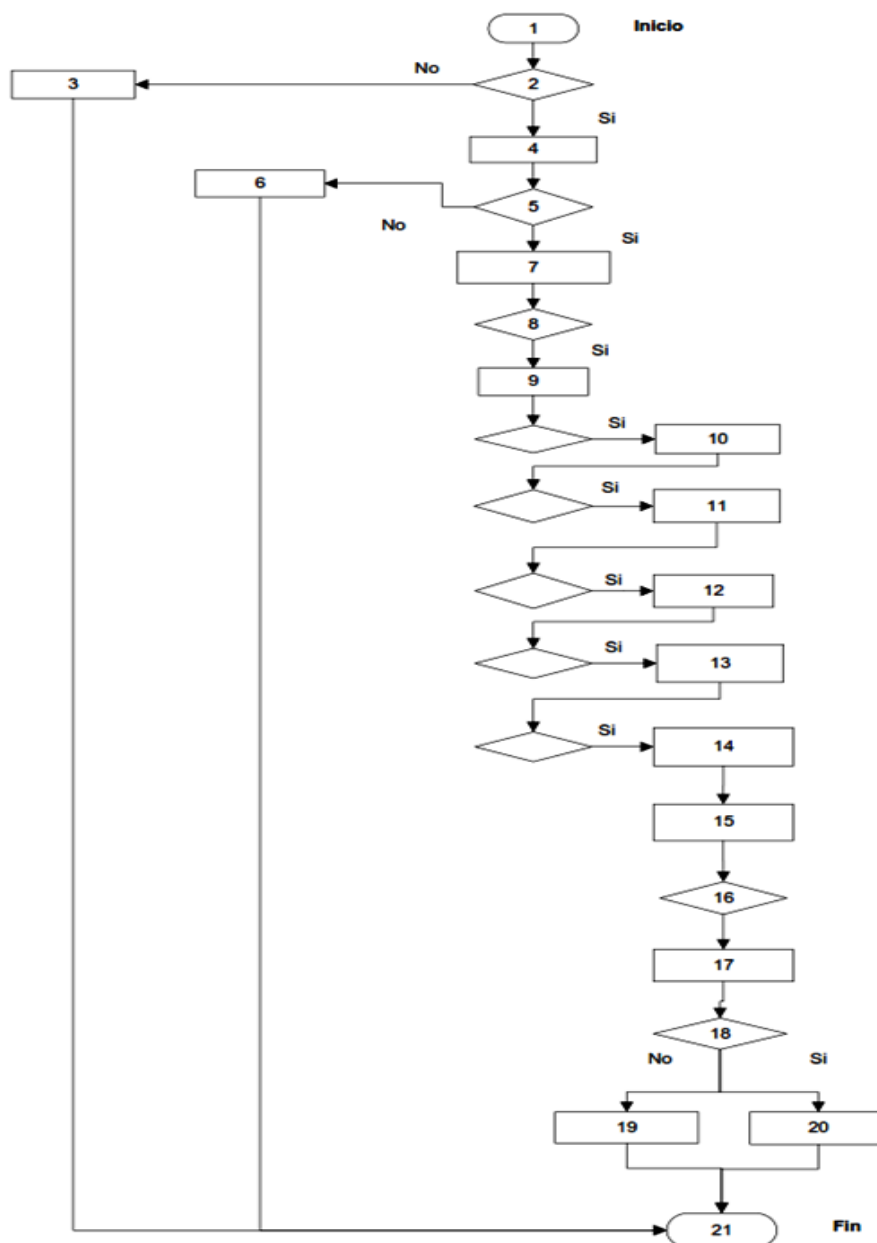


Figura 4.30: Diagrama de Flujo para la Validación de los Dígitos de una Cedula

Tabla 4.31: Pruebas de Caja Blanca de la Validación de los Dígitos de una Cedula

Código de Validación de Cédula	Nodos
function validarCI(\$strCedula) {	1
if(is_numeric(\$strCedula)){	2
se suma el total de caracteres \$total_caracteres=strlen(\$strCedula);	4
compruebo que tenga 10 digitos la cedula if(\$total_caracteres==10){	5
Obtenemos el digito de la región que son los dos primeros dígitos \$nro_region=substr(\$strCedula, 0,2);	7
Compruebo a que region pertenece esta cedula if(\$nro_region>=1 && \$nro_region<=24){ extraigo el ultimo digito de la cedula \$ult_digito=substr(\$strCedula, -1,1); Extraigo los valores pares y los sumo \$valor2=substr(\$strCedula, 1, 1); \$valor4=substr(\$strCedula, 3, 1); \$valor6=substr(\$strCedula, 5, 1); \$valor8=substr(\$strCedula, 7, 1); \$suma_pares=(\$valor2 + \$valor4 + \$valor6 + \$valor8);	8
Extraigo los valores impares \$valor1=substr(\$strCedula, 0, 1); \$valor1=(\$valor1 * 2);	9
if(\$valor1>9){ \$valor1=(\$valor1 - 9); }else{ } \$valor3=substr(\$strCedula, 2, 1); \$valor3=(\$valor3 * 2);	10
if(\$valor3>9){ \$valor3=(\$valor3 - 9); }else{ } \$valor5=substr(\$strCedula, 4, 1); \$valor5=(\$valor5 * 2);	11
if(\$valor5>9){ \$valor5=(\$valor5 - 9); }else{ } \$valor7=substr(\$strCedula, 6, 1); \$valor7=(\$valor7 * 2);	12

Tabla 4.31: Prueba de Caja Blanca de la Validación de los Dígitos de una Cedula (Cont.)

<pre>if(\$valor7>9){ \$valor7=(\$valor7 - 9); }else{ } \$valor9=substr(\$strCedula, 8, 1); \$valor9=(\$valor9 * 2);</pre>	13
<pre>if(\$valor9>9){ \$valor9=(\$valor9 - 9); }else{ }</pre>	14
<pre>\$suma_impares=(\$valor1 + \$valor3 + \$valor5 + \$valor7 + \$valor9);</pre>	15
<pre>Suma Total \$suma=(\$suma_pares + \$suma_impares); Extraigo el primer número de la suma \$dis=substr(\$suma, 0,1); Luego ese número lo multiplico x 10, consiguiendo así la decena inmediata superior \$dis=(\$dis + 1)* 10); Obtenemos la resta de la decena inmediata - la suma total esto nos da el digito validador \$digito=(\$dis - \$suma);</pre>	16
<pre>si la suma nos resulta 10, el decimo digito es cero if(\$digito==10){ \$digito='0'; }else{ }</pre>	17
<pre>Validamos que el digito validador sea igual al de la cedula if (\$digito==\$ult_digito){</pre>	18
<pre>echo "Cedula Correcta";</pre>	20
<pre>}else{ echo "Cedula Incorrecta"; }</pre>	6
<pre>}else{ echo "Esta cédula no pertenece a ninguna Provincia" }</pre>	19
<pre>}else{ echo "Esta cédula tiene menos de 10 Dígitos" }</pre>	3
<pre>} FIN</pre>	21

Grafo de Flujo de la Validación de la Cedula

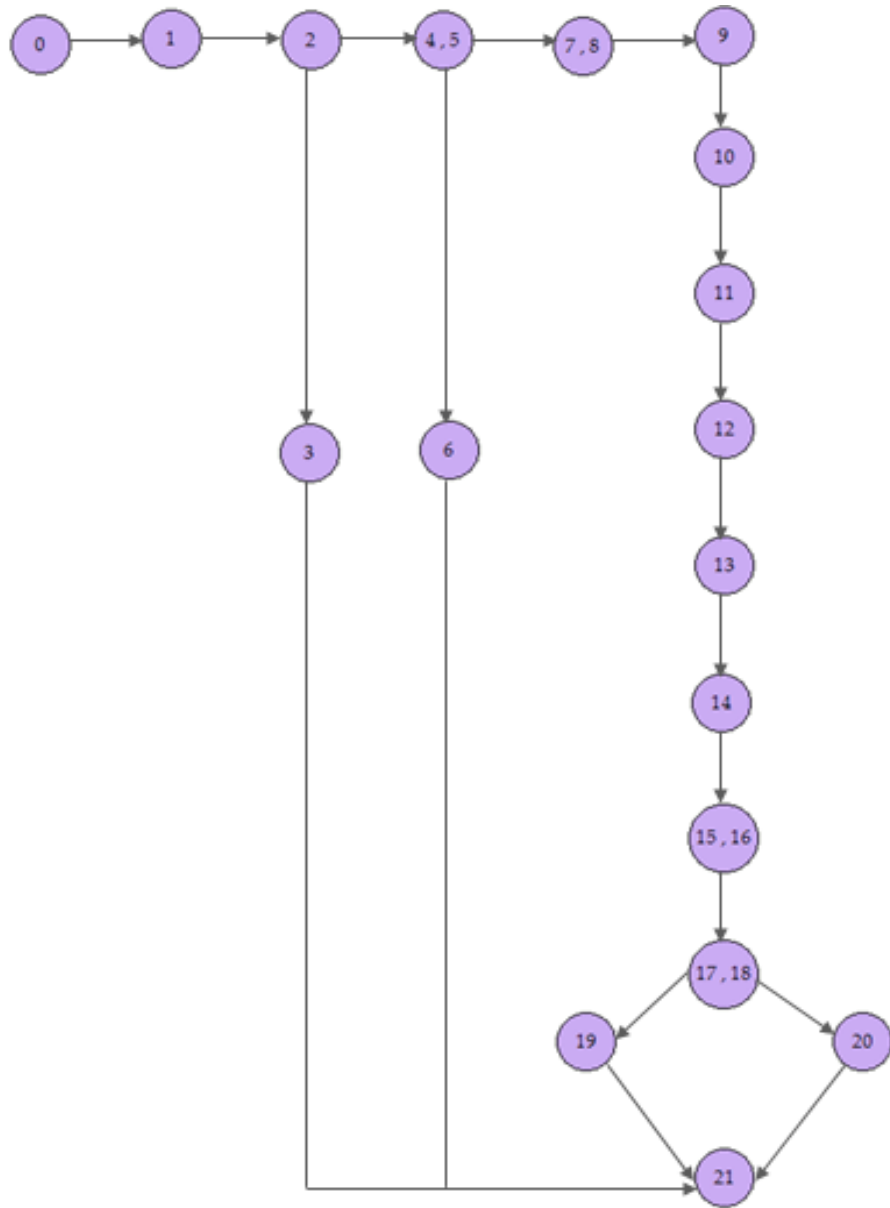


Figura 4.31: Grafo de Flujo para la Validación de los Dígitos de una Cedula

Complejidad Ciclomática

$$V(G) = A \text{ (arista)} - N \text{ (nodos)} + 2$$

$$V(G) = 20 - 18 + 2$$

$$V(G) = 4$$

Caminos Básicos

1: 0-1-2-3-21

2: 0-1-2-4-5-6-21

3: 0-1-2-4-5-7-8-9-10-11-12-13-14-15-16-17-18-19-21

4: 0-1-2-4-5-7-8-9-10-11-12-13-14-15-16-17-18-20-21

Ejemplo:

Ingreso del número de cédula: 180373114

180373114 \neq 10

echo "Esta cédula tiene menos de 10 Dígitos"

FIN.

4.4.2. Pruebas de Caja Negra

El sistema de pruebas de caja negra no considera la codificación dentro de sus parámetros a evaluar, es decir, que no están basadas en el conocimiento del diseño interno del programa.

Estas pruebas se enfocan en los requerimientos establecidos y en la funcionalidad del sistema web en la cual se hizo énfasis en la revisión de los puntos descritos a continuación:

- Entrada y salida de datos por parte de los usuarios.
- Validar las cajas de texto para el ingreso de caracteres específicos.
- Comprobar que las páginas se ajuste a los datos necesitados y requeridos.
- Mostrar la información correcta y entendible para los usuarios.
- No visualizar información innecesaria y secreta en ninguna caso.

Dentro de ingresos, eliminación y actualización las pruebas de caja negra se basan en la verificación de datos que se envían por medio de la interfaz mediante una petición de ingreso de datos necesarios para alterar la información de la base de datos.

En el proceso de selección las pruebas de caja negra se basan en la presentación de la información en los componentes utilizados para mostrar la información de manera fácil e intuitiva para el usuario.

Ejemplo de pruebas de caja negra utilizando un proceso de la aplicación web que es el tratamiento Matricula de un Estudiante de la institución.

Tabla 4.32: Pruebas de Caja Negra - Matricula de un Estudiante

Matricula		
Acciones	Casos Esperado	Resultados Obtenidos
Ingreso	Mensajes de error	Mensajes de error describiendo el motivo del error y la solución.
	Información agregada correctamente.	Mensaje de confirmación de inserción.
Actualización	Mensajes de error	Mensajes de error describiendo el motivo del error y la solución.
	Información actualizada correctamente.	Información actualizada y un mensaje de confirmación de actualización.
Eliminación	Mensajes de error	Mensajes de error describiendo el motivo del error y la solución.
	Datos eliminados correctamente.	Datos eliminados y un mensaje de confirmación de eliminación.
Selección	Despliegue de la información del proceso.	Muestra la información de los procesos requeridos.

Demostración de las pruebas de caja negra según las interfaces de la aplicación web:

El resultado esperado dependerá de los datos ingresados, ya sea de manera correcta o de manera incorrecta, siendo esta: datos incompletos, elementos vacíos, etc., mostrando los mensajes de confirmación o de error respectivamente.

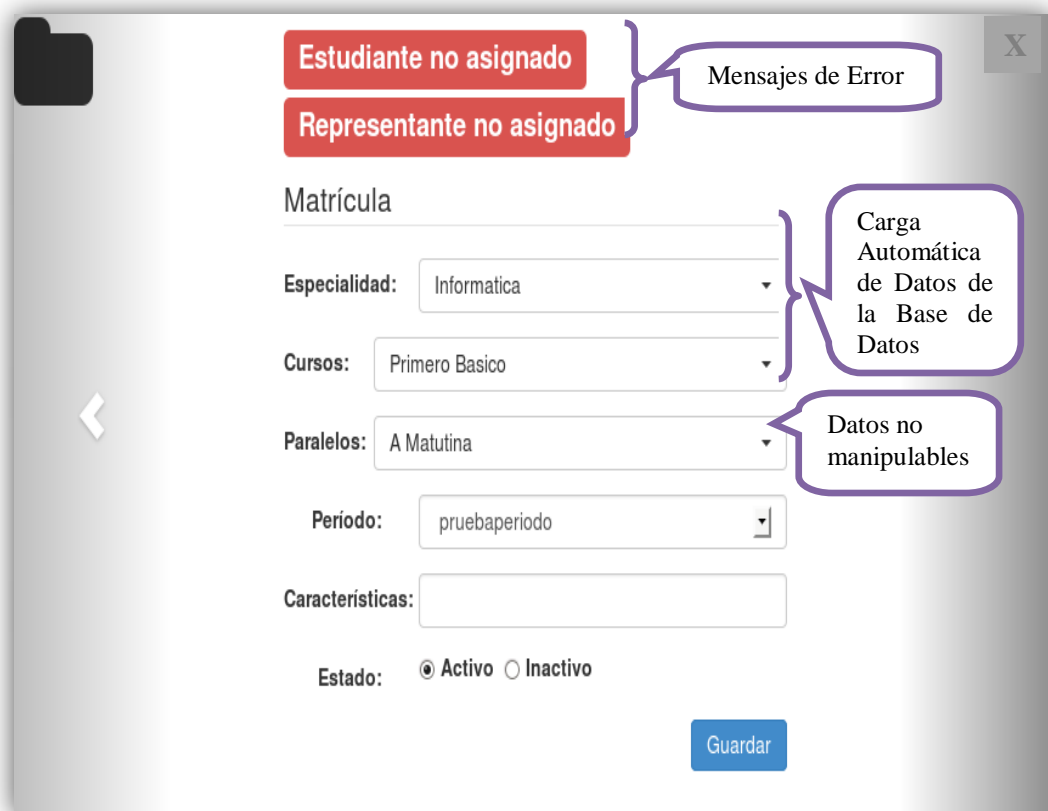


Figura 4.32: Prueba de Caja Negra – Matricula de un Estudiante

Pruebas de Verificación y Validación

A través de las pruebas de verificación y validación de la aplicación, se ha demostrado que el sistema cumple con las especificaciones funcionales que se establecieron al inicio del proyecto además cumple con los objetivos propuestos, estas pruebas se realizaron con la Ingeniera en Sistemas del Colegio a Distancia Stephen Hawking de la ciudad de Ambato.

✓ Iteración 2

I.2	A	D	I	P
-----	---	---	---	---

1.1. Análisis

Es el proceso de investigar un problema que se quiere resolver, definir claramente el Problema que se quiere resolver o el sistema que se desea crear, así como también identificar los componentes principales que integrarán el producto.

1.1.1. Requerimientos del Sistema

Utilizando como técnica la observación y como instrumento una guía de observación, se ha recolectado los datos necesarios, mismos que nos permiten conocer los proceso actuales que se realiza en el Colegio a Distancia Stephen Hawking de la Ciudad de Ambato.

Tabla 4.33: Guía de Observación de Información

GUÍA DE OBSERVACIÓN	
Objetivo: Recolectar información del Desarrollo de un Proceso de Control Académico, para la automatización del mismo mediante una aplicación Web.	
Tipo de Observación: Observación Directa	
INDICADORES A OBSERVAR	
+ Medios para la Recolección de la Información	
Ficha de Registro de Notas Ficha de Pase de Año de los Estudiantes	
+ Medios en que se Almacena la Información Generada	
Hojas de Calculo Carpetas Físicas	Archivos de Textos Discos Magnéticos
+ Documentos Generados en el Procesamiento	
Informe de Calificaciones y Conducta Informe de Pases de Año	Informe de Docentes Informe de Estudiantes
✓ Disponibilidad de la Información	
Búsqueda de Archivos Digitales Búsqueda de Archivos Físicos en Carpetas	

Tabla 4.33: Guía de Observación de Información (Cont.)

✓ Elaboración de los Cronogramas	
Creación de Cursos en base a los procesos	Asignación de Materias
Asignación de Cursos	Asignación de Paralelos
Asignación de Especialidad	Asignación de Docentes
✓ Actividades en el Desarrollo de un Proceso Académico	
Inscripción de Estudiantes	Asignar un estudiante a un curso
Registrar Notas de los Estudiantes	Asignar un curso a una especialidad
Registrar Conducta de los Estudiantes	Asignar un paralelo a un curso
Asignar un estudiante a un paralelo	Asignar un docente a un curso
✓ Seguimiento de Calificaciones	
Evaluación al final de cada Quimestre	Elaboración de un archivo digital
Archivos físicos de las Evaluaciones	
✓ Otorgación de Títulos de Bachiller	
Nota final igual o mayor a 7	Conducta positiva

Una vez finalizado el diálogo se ha llegado a la conclusión de que el Colegio a Distancia Stephen Hawking de la Ciudad de Ambato no cuenta con un Sistema

Web que permita el apropiado control de los procesos académicos tanto de estudiantes como de docentes y la facilidad de uso de dispositivos móviles, ya que dicha labor se la realiza de manera manual lo que provoca cierta disconformidad por parte del personal del colegio como también en los estudiantes que forman parte del mismo.

Para la elaboración del Sistema Académico Web Utilizando Tecnología Móvil se realizaron reuniones con el personal encargado del manejo, registro y manipulación de los datos, en dichas reuniones se establecieron los requerimientos que debe cumplir el sistema para solucionar los problemas que se presentan a lo largo de los años.

1.1.2. Diagramas de Casos de Uso

Funciones Administrador - Estudiante

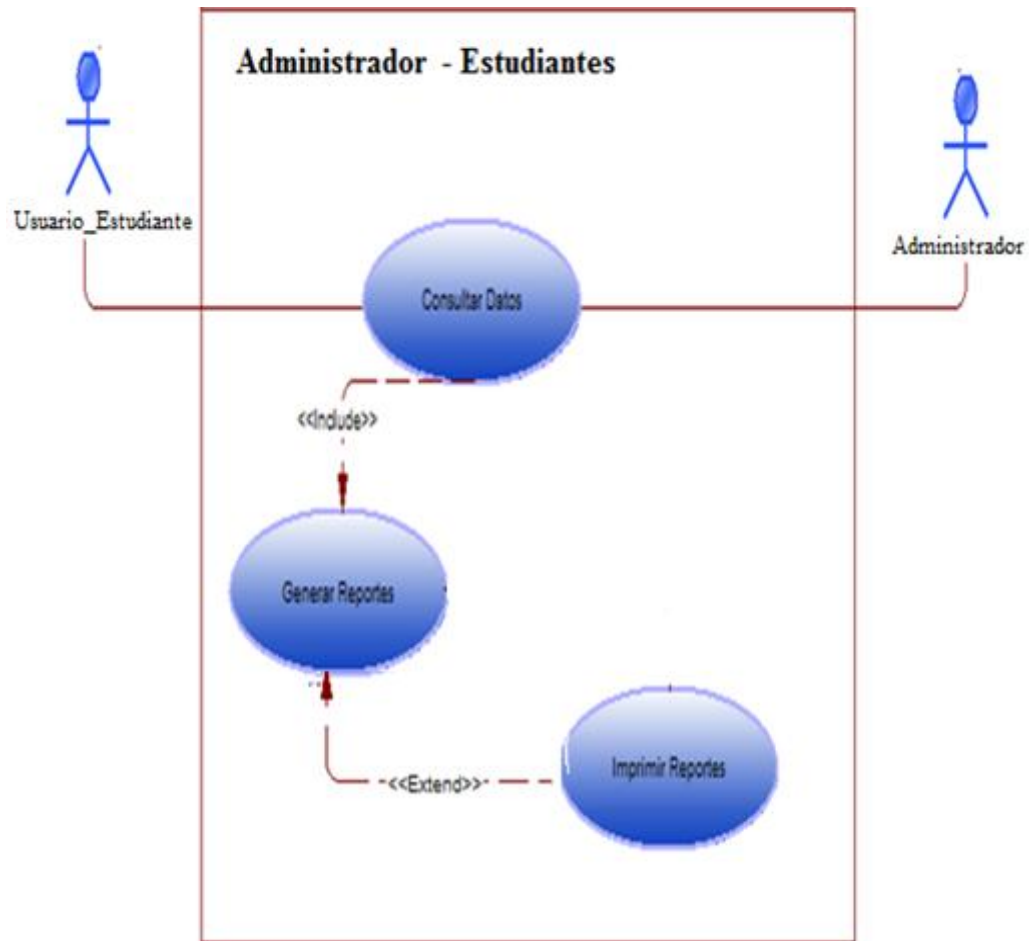


Figura 4.33: Diagrama de Caso de Uso Administrador - Estudiante

Descripción de los Casos de Uso: Consultar Datos

Tabla 4.34: Caso de Uso: Consultar Datos

Caso de Uso: Consultar Datos		
Descripción:	Permitirá la Consulta de Notas requerida.	
Actor:	Administrador, Usuario	
Precondición:	Los actores deben tener una sesión activa.	
Poscondición:	Se presenta la información requerida por el actor.	
Presunción:	Base de datos Disponible.	
Flujo Principal:	Evento Actor	Evento Sistema
	1. Buscar la opción según la operación que se vaya a realizar.	1. Muestra en la pantalla el menú de Administración de Notas.
	2. Seleccionar la opción a realizarse.	2. Cargar las interfaces de las operaciones a realizar.
	3. Seleccionar actividad a realizar.	3. Mostrar las interfaces de las opciones que se pueden realizar en el proceso.
	4. Enviar datos mediante una petición.	4. Verificar la información recibida.
		5. Mostrar información requerida.
Alternativa 1:	1. Buscar opción de inicio.	
	2. Enviar petición web.	2. Re direccionar a la página principal del sistema.
Alternativa 2:	1. Buscar opción Generar Reporte.	
	2. Enviar petición web.	2. Mostrar el Reporte Generado.
	3. Buscar opción Imprimir Reporte	
	4. Enviar petición web.	4. Imprimir Reporte de Notas.

1.1.3. Diagramas de Secuencia

Diagramas de Secuencia MVC: Consultar Datos

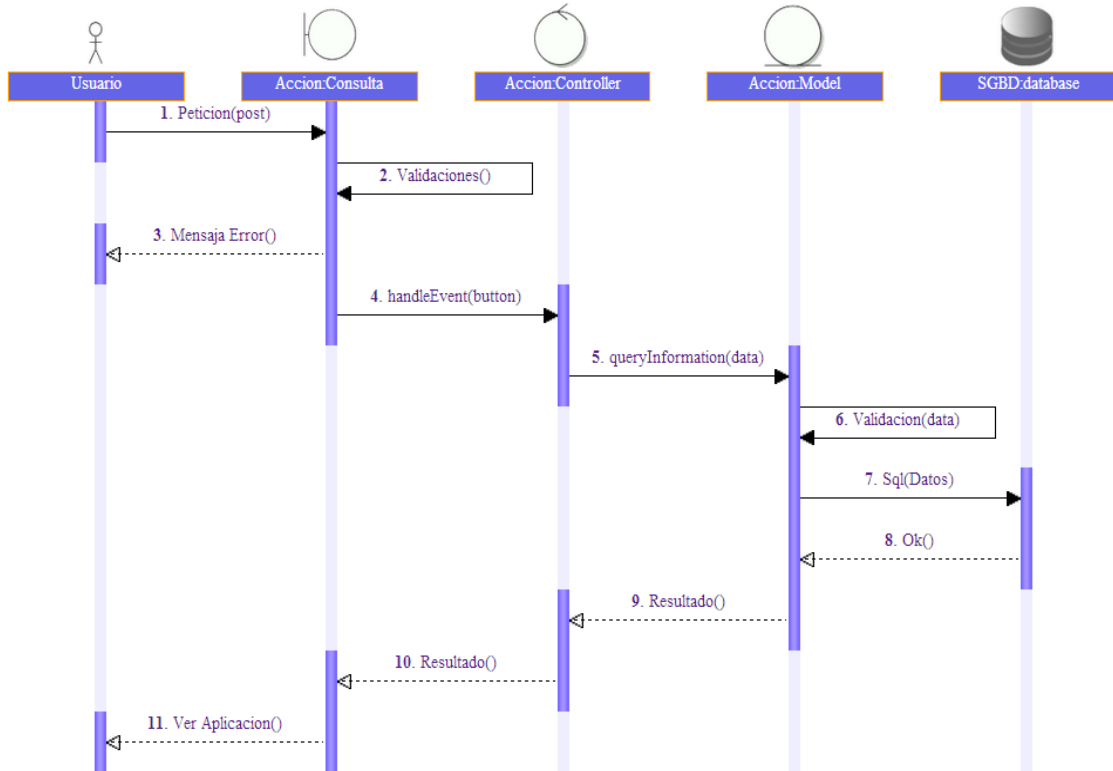


Figura 4.34: Diagrama de Secuencia MVC: Consultar Datos

Descripción: Diagrama de Secuencia: Consultar Datos

Tabla 4.35: Descripción MVC Consultar Datos

Descripción: Diagrama de Secuencia: Consultar Datos
1. Usuario call Accion:Consulta->Petición(post)
2. Accion:Consulta call Accion:Consulta->Validaciones()
3. Usuario receive from Accion:Consulta->Mensaje Error()
4. Accion:Consulta call Accion:Controller->handleEvent(button)
5. Accion:Controller call Accion:Model->queryInformation(data)
6. Accion:Model call Accion:Model->Validación(data)
7. Accion:Model call SGBD:database->Sql(Datos)
8. Accion:Model receive from SGBD:database->Ok()
9. Accion:Controller receive from Accion:Model->Resultado()
10. Accion:Consulta receive from Accion:Controller->Resultado()
11. Usuario receive from Accion:Consulta->Ver Aplicación()

Diagrama de Secuencia MVC: Generar Reportes

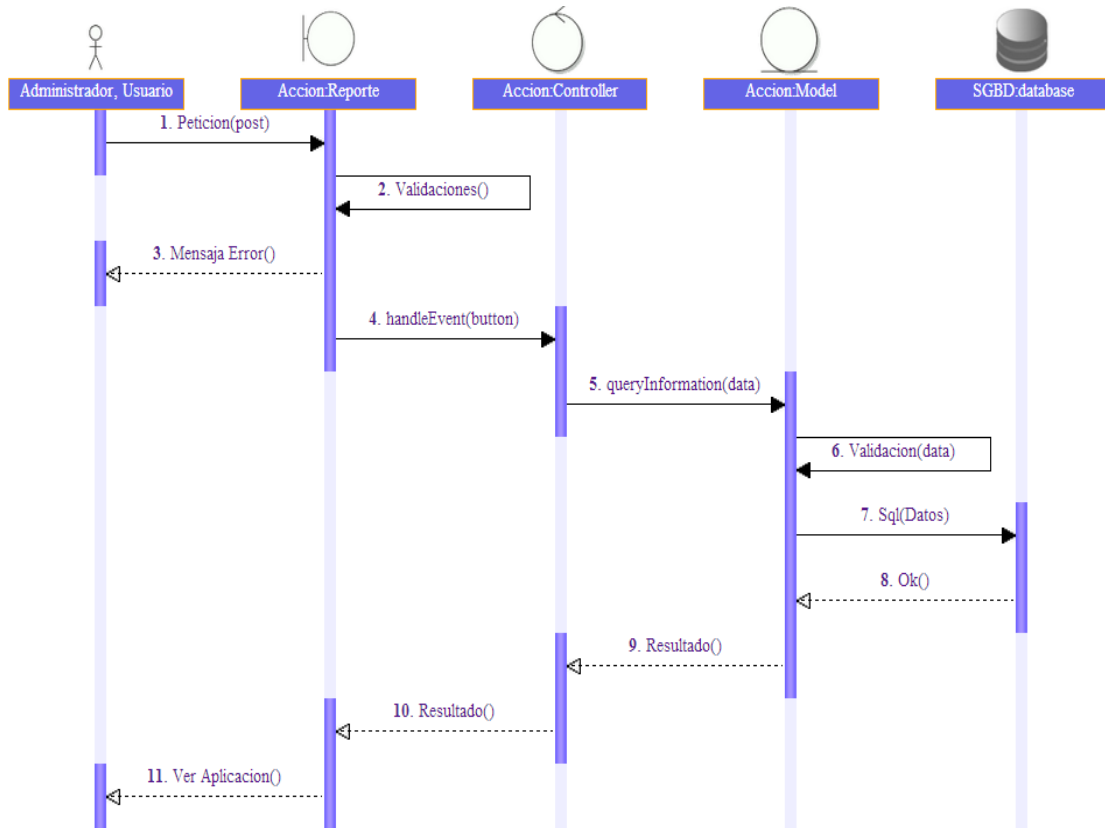


Figura 4.35: Diagrama de Secuencia MVC: Generar Reportes

Descripción: Diagrama de Secuencia: Generar Reportes

Tabla 4.36: Descripción MVC Generar Reportes

Descripción: Diagrama de Secuencia: Generar Reportes
1. Administrador, Usuario call Accion:Reporte->Petición(post)
2. Accion:Reporte call Accion:Reporte->Validaciones()
3. Administrador, Usuario receive from Accion:Reporte->Mensaje Error()
4. Accion:Reporte call Accion:Controller->handleEvent(button)
5. Accion:Controller call Accion:Model->queryInformation(data)
6. Accion:Model call Accion:Model->Validación(data)
7. Accion:Model call SGBD:database->Sql(Datos)
8. Accion:Model receive from SGBD:database->Ok()
9. Accion:Controller receive from Accion:Model->Resultado()
10. Accion:Reporte receive from Accion:Controller->Resultado()
11. Administrador, Usuario receive from Accion:Reporte->Ver Aplicación()

1.2. Diseño

1.2.1. Diseño de Base de Datos

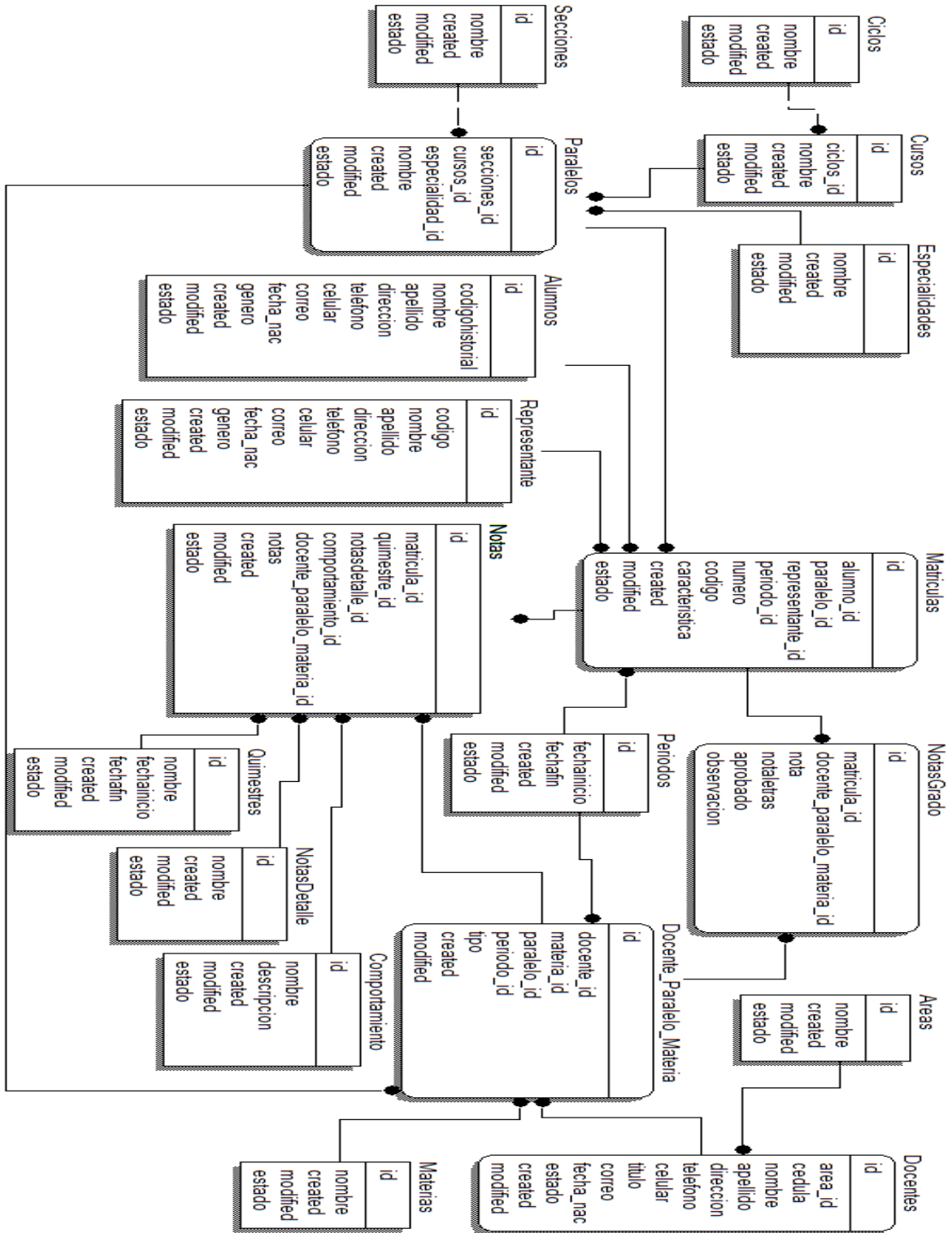


Figura 4.36: Diseño de Base de Datos

1.2.2. Diccionario de Datos

Tabla 4.37: Descripción de la Tabla Secretarias

TABLA: Secretarias			
CAMPO	TIPO	DESCRIPCIÓN	CLAVE
id	int	Id de la tabla Secretarias	PK
cedula	char(10)	Cedula de la Secretaria	
nombre	char(50)	Nombre de la Secretaria	
apellido	char(50)	Apellido de la Secretaria	
dirección	char(50)	Dirección de la Secretaria	
celular	char(10)	Celular de la Secretaria	
teléfono	char(9)	Teléfono de la Secretaria	
correo	char(50)	Correo de la Secretaria	
genero	char(50)	Genero de la Secretaria	
fecha_naci	date	Fecha de Nacimiento de la Secretaria	
created	date	Fecha de Creación de la Secretaria	
modified	date	Fecha de Modificación de la Secretaria	
estado	int	Estado de la Secretaria	

Tabla 4.38: Descripción de la Tabla Representantes

TABLA: Representantes			
CAMPO	TIPO	DESCRIPCIÓN	CLAVE
id	int	Id de la tabla Representantes	PK
código	char(20)	Código del Representante	
nombre	char(50)	Nombre del Representante	
apellido	char(50)	Apellido del Representante	
dirección	char(50)	Dirección del Representante	
celular	char(10)	Celular del Representante	
teléfono	char(9)	Teléfono del Representante	
correo	char(50)	Correo del Representante	
genero	char(50)	Genero del Representante	
fecha_naci	date	Fecha de Nacimiento del Representante	
created	date	Fecha de Creación del Representante	
modified	date	Fecha de Modificación del Representante	
estado	int	Estado del Representante	

Tabla 4.39: Descripción de la Tabla Áreas

TABLA: Áreas			
CAMPO	TIPO	DESCRIPCIÓN	CLAVE
id	int	Id de la tabla Áreas	PK
nombre	char(50)	Nombre del Área	
created	date	Fecha de Creación del Área	
modified	date	Fecha de Modificación del Área	
estado	int	Estado del Área	

Tabla 4.40: Descripción de la Tabla NotaGrados

TABLA: NotaGrados			
CAMPO	TIPO	DESCRIPCIÓN	CLAVE
id	int	Id de la tabla notagrados	PK
Nota	decimal	Nota en numero	
NotaLetras	char(150)	Nota en Letras	
Aprobado	int	Aprobado o no el grado	
Observación	decimal	Observación de la Nota de Grado	
matricula_id	int	Id de la Matricula	FK1
docenteparalelomateria_id	int	Id del DocenteParaleloMateria	FK2

Tabla 4.41: Descripción de la Tabla Groups

TABLA: Groups			
CAMPO	TIPO	DESCRIPCIÓN	CLAVE
id	int	Id de la tabla Grupos	PK
name	varchar(100)	Nombre del Grupo	
created	date	Fecha de Creación del Grupo	
modified	date	Fecha de Modificación del Grupo	
estado	int	Estado del Grupo	

Tabla 4.42: Descripción de la Tabla Users

TABLA: users			
CAMPO	TIPO	DESCRIPCIÓN	CLAVE
id	int	Id de la tabla Usuarios	PK
username	varchar(255)	Nombre del Usuario	
password	char(40)	Contraseña del Usuario	
identificador	varchar(10)	Identificador del Usuario	
group_id	int	Id del Grupo	FK1
created	date	Fecha de Creación del Usuario	
modified	date	Fecha de Modificación del Usuario	
estado	int	Estado del Usuario	

Tabla 4.43: Descripción de la Tabla FotoUsers

TABLA: FotoUsers			
CAMPO	TIPO	DESCRIPCIÓN	CLAVE
id	int	Id de la tabla Foto del Usuario	PK
filename	varchar(255)	Nombre del Archivo	
dir	varchar(255)	Dirección - Ubicación del Archivo	
mimetype	varchar(255)	Tipo de Archivo	
filesize	int	Tamaño del Archivo	
user_id	date	Id del Usuario	FK1
created	date	Fecha de Creación de la Foto de Usuario	
modified	date	Fecha de Modificación de la Foto de Usuario	
estado	int	Estado de la Foto de Usuario	

Tabla 4.44: Descripción de la Tabla NotasDetalles

TABLA: NotasDetalles			
CAMPO	TIPO	DESCRIPCIÓN	CLAVE
id	int	Id de la tabla NotasDetalle	PK
nombre	char(50)	Nombre de la Nota	
created	date	Fecha de Creación de la Notas	
modified	date	Fecha de Modificación de la Nota	
estado	int	Estado de la Nota	

Tabla 4.45: Descripción de la Tabla DocenteParaleloMaterias

TABLA: DocenteParaleloMaterias			
CAMPO	TIPO	DESCRIPCIÓN	CLAVE
id	int	Id de la tabla docenteparalelomaterias	PK
tipo	varchar(20)		
docente_id	int	Id del Docente	FK1
paralelo_id	int	Id del Paralelo	FK2
materia_id	int	Id de la Materia	FK3
created	date	Fecha de Creación del DocenteParaleloMateria	
modified	date	Fecha de Modificación del DocenteParaleloMateria	
estado	int	Estado del DocenteParaleloMateria	

1.2.3. Diseño de la Interfaz de Usuario

Interfaz de Estudiantes

Modal de Administración de Estudiantes, se mostrara todas las opciones de la misma al usuario que inicio sesión con sus respectivos permisos o restricciones para la manipulación de las operaciones.

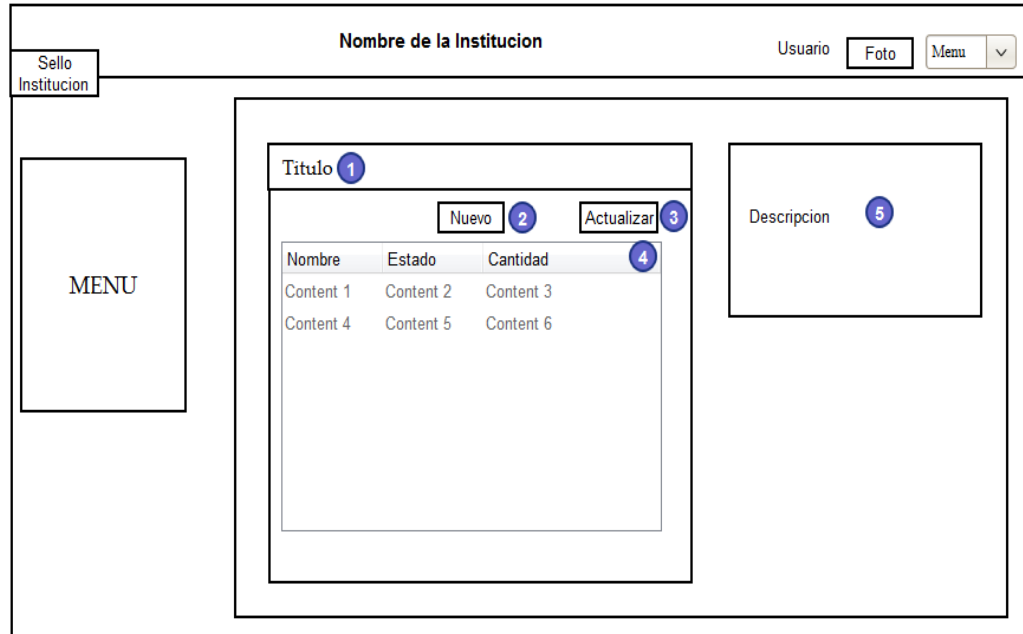


Figura 4.37: Interfaz de Estudiantes

1. **Título.-** Titulo de la ventana.
2. **Botón Nuevo.-** Permite agregar un nuevo Estudiante.
3. **Botón Actualizar.-** Actualiza el contenido mostrado si se ha realizado algún cambio.
4. **Área de Contenido.-** Muestra el contenido de la tabla Estudiantes.
5. **Área de Descripción.-** Describe las características del Estudiante seleccionado.

4.3. Implementación

4.3.1. Implementación de la Interfaz de Grafica de Usuario

Se presenta a continuación algunas de las páginas de la aplicación web implementadas, las mismas que fueron planteadas en la etapa de diseño, se especifica también cada sección de dichas páginas. En el manual de usuario en la sección de anexos, se incluye y se explica la totalidad de las páginas así como su respectiva funcionalidad.

Página Principal

Se la utiliza como base para las diferentes páginas de la aplicación web, además se muestra los controles en común de todos los módulos del sistema.

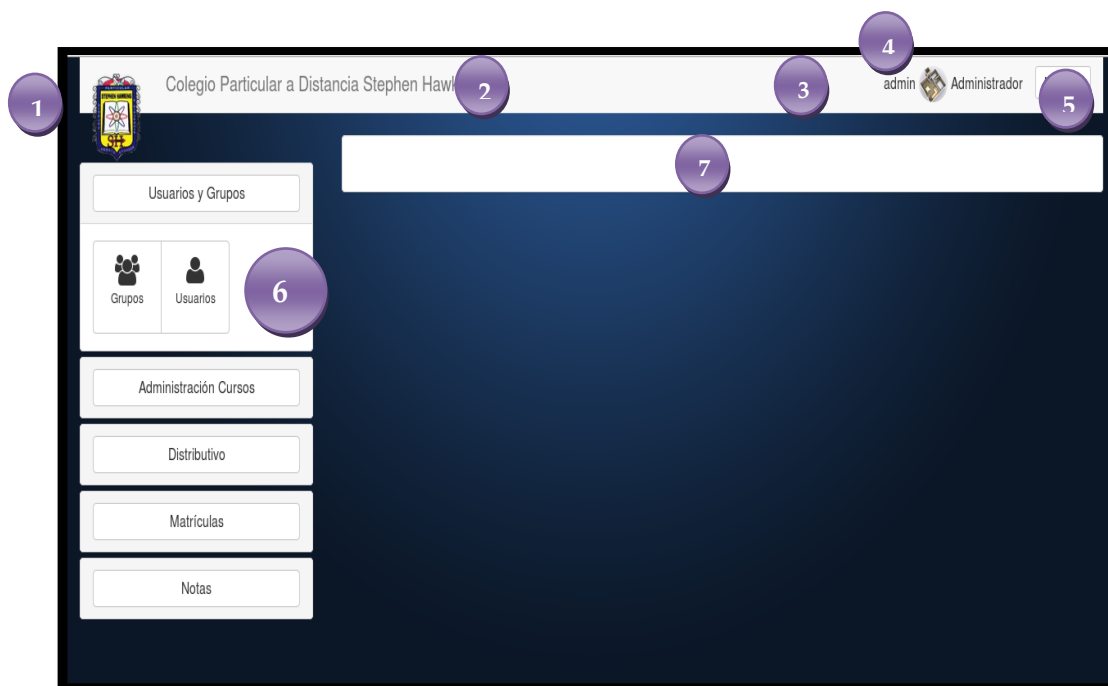


Figura 4.38: Pantalla Principal

1. **Sello de la Institución.-** Se muestra el respectivo sello de la institución a la que va dirigido el sistema.
2. **Nombre de la Institución.-** Muestra el nombre de la institución.

3. **Usuario.-** Muestra el nombre de usuario activo.
4. **Foto.-** Muestra la respectiva foto del usuario que ha ingresado al sistema.
5. **Menú Configuración.-** Muestra las respectivas opciones del menú configuración de la cuenta del usuario y la opción de salir de la aplicación.
6. **Menú Principal.-** Consta de cinco ítems principales como son el de Usuarios y Grupos, Administración Cursos, Distributivo, Matriculas, Notas.

Cada ítem principal se divide en sub-categorías las mismas que son descritas a continuación.

Usuarios y Grupos:

- Grupos
- Usuarios

Administración Cursos:

- Ciclos
- Secciones
- Especialidades
- Cursos

Distributivo:

- Periodos
- Áreas
- Docentes
- Materias
- Distribución

Matriculas:

- Estudiantes
- Representantes
- Matricula

Notas:

- Quimestres
- Detalle Notas
- Comportamiento
- Registro Notas

7. Área de Trabajo.- Espacio destinado para la gestión de la información. La información mostrada dependerá directamente de las opciones seleccionadas en el menú.

4.3.2. Implementación de la Arquitectura de Programación

La arquitectura utilizada en la programación para el desarrollo de la aplicación web sigue el patrón de diseño MVC, el mismo que separa el código en tres capas:

- **El Modelo:** Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.
- **La Vista:** Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho 'modelo' la información que debe representar como salida.
- **El Controlador:** Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta de 'modelo' (por ejemplo, desplazamiento o scroll por un documento o por los diferentes registros de una base de datos), por tanto se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo' (véase Middleware).

A continuación se presenta la codificación de varios Triggers que se utilizaron en el desarrollo y manipulación de la aplicación, los mismos que sirven como referencia para los demás ya desarrollados.

Ejemplo de Triggers Almacenados en la Base de Datos

Creación del Trigger para el Control de las Notas

```
DELIMITER |
CREATE TRIGGER controlDeNotas BEFORE INSERT ON notas
FOR EACH ROW
BEGIN
  declare msg varchar(255);
  if new.notas<0 or new.notas>10 then
    set msg = concat('MyTriggerError: Las notas deben estar entre 0 y 10: ', cast(new.id as char));
    signal sqlstate '45000' set message_text = msg;
  end if;
END
|
```

Figura 4.39: Creación del Trigger para el Control de Notas

Creación del Trigger para Generar el Código de Matricula de un Alumno

```
DELIMITER |
CREATE TRIGGER codigoMatriculaNumero BEFORE insert ON matriculas
FOR EACH ROW
BEGIN
  declare NumeroMatricula int;

  Set NumeroMatricula=( SELECT COALESCE(MAX(numero),0)+1
                        FROM matriculas
                        where periodo_id=new.periodo_id);

  set new.numero=NumeroMatricula;

  set new.codigo=CONCAT('MAE',NumeroMatricula,'PE',new.periodo_id);

END
|
```

Figura 4.40: Creación del Trigger para Generar el Código del Matricula de un Alumno

Creación del Trigger para Generar el Código del Historial de un Alumno

```
DELIMITER |
CREATE TRIGGER generarCodigoHistorial BEFORE INSERT ON alumnos
FOR EACH ROW
BEGIN
Set new.codigohistorial=CONCAT('HAE',(SELECT auto_increment FROM information_schema.tables
WHERE table_schema='shescolastico' and table_name='alumnos'));
END
|
```

Figura 4.41: Creación del Trigger para Generar el Código del Historial de un Alumno

Creación del Trigger para Quitar un Usuario Estudiante

```
delimiter $$
CREATE TRIGGER quitarUsuarioEstudiante after delete ON alumnos
FOR EACH ROW
BEGIN
DELETE FROM users
WHERE identificador=old.codigohistorial;
END
$$
```

Figura 4.42: Creación del Trigger para Quitar un Usuario Estudiante

Creación del Trigger para Quitar un Usuario Docente

```
delimiter $$
CREATE TRIGGER quitarUsuarioDocente after delete ON docentes
FOR EACH ROW
BEGIN
    DELETE FROM users
    WHERE identificador=old.cedula;
END
$$

drop trigger crearUsuarioEstudiante;

delimiter $$
```

Figura 4.43: Creación del Trigger para Quitar un Usuario Docente

Creación del Trigger para Crear un Quimestre

```
delimiter $$
CREATE TRIGGER crearquimestresperiodo after insert ON periodos
FOR EACH ROW
BEGIN
    INSERT INTO quimestres(nombre,created,modified,estado)
    VALUES ('QUIMESTREI',CURDATE(),CURDATE(),4),
           ('QUIMESTREII',CURDATE(),CURDATE(),4);
END
$$
```

Figura 4.44: Creación del Trigger para Crear un Quimestre

Código de Conexión con el Servidor

```
App/Config/database.php

public $default = array(
    'datasource' => 'Database/Mysql',
    'persistent' => false,
    'host' => 'localhost',
    'login' => 'notas',
    'password' => '*****&tb',
    'database' => 'colsteph_shescolastico',
    'prefix' => "",
    'encoding' => 'utf8',
);
```

Figura 4.45: Código de Conexión con el Servidor

➤ Implementación de los Datos de la Aplicación Web

Aquí explicaremos como subir los datos de la aplicación web al servicio de hosting.

Se utilizo el servidor GALAPAGOS.ECUAHOSTING.NET, específicamente en la dirección: <https://galapagos2.ecuahosting.net:2083/>, servidor que ha sido utilizado siempre por el Colegio Stephen Hawking hasta la actualidad, servidor en el cual se maneja el dominio de la Institución.

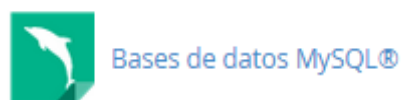
Primero ingresamos a la dirección <https://galapagos2.ecuahosting.net:2083/>, en donde nos aparece el Cpanel.



Figura 4.46: Interfaz de Entrada de Cpanel

En donde ingresamos el nombre de usuario y la contraseña respectivamente para poder acceder al Cpanel.

Una vez ingresado al Cpanel podemos Crear la base de datos requerida para el sistema.



Crear una nueva base de datos

Nueva base de datos:

Crear una base de datos

Modificar las bases de datos

Verificar la base de datos

Verificar la base de datos

Reparar base de datos

Reparar base de datos

Bases de datos actuales

Ir

Base de datos	Tamaño	Usuarios con privilegio	Acciones
colsteph_evash	76.08 MB	colsteph_evash	Cambiar el nombre Eliminar
colsteph_moodle	12.71 MB		Cambiar el nombre Eliminar
colsteph_shescolastico	0.72 MB	colsteph_amw	Cambiar el nombre Eliminar
colsteph_v2	1.35 MB	colsteph_v2	Cambiar el nombre Eliminar

Página: [Primero](#) [1](#) [Último](#)

Por página: [Ir](#)

Figura 4.47: Interfaz de Creación de una Nueva Base de Datos

Después de haber creado la Base de Datos debemos crear un Subdominio

Subdominios

Un subdominio es una subsección de su sitio web que puede existir como un sitio web nuevo sin un nombre de dominio. Use los subdominios para crear URL fáciles de recordar de diferentes áreas de contenido de su sitio. Por ejemplo, puede crear un subdominio para su blog, al que pueda acceder a través de **blog.example.com** y **www.example.com/blog**

Crear un subdominio

Subdominio

Dominio

Directorio raíz

Crear

Modificar un subdominio

Ir

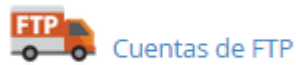
Subdominios.Dominio de raíz	Directorio raíz	Redireccionamiento	Acciones
notas.colstephenhawking.edu.ec	/public_html/notas	not redirected	Eliminar Administrar redireccionamiento

Página: [Primero](#) [1](#) [Último](#)

Por página: [Ir](#)

Figura 4.48: Interfaz de Creación de un Subdominio

Para poder administrar los archivos de nuestro sitio web debemos crear una Cuenta FTP, dentro del Cpanel podemos encontrar esta opción.



Añadir una cuenta de FTP

ATENCIÓN: Uso de 1 de 1 cuentas de FTP disponibles.

Inicio de sesión

@colstephenhawking.edu.ec

Contraseña

Contraseña (nuevamente)

Seguridad

Muy débil (0/100)

Directorio

Cuota

2000 MB

Ilimitado

Figura 4.49: Interfaz de Creación de una Cuenta FTP

Pasos para subir el sitio o aplicación web al Dominio de la Institución

Primero descargamos e instalamos el CuteFTP

Nos aparece esta pantalla, en la cual podemos agregar una nueva conexión.

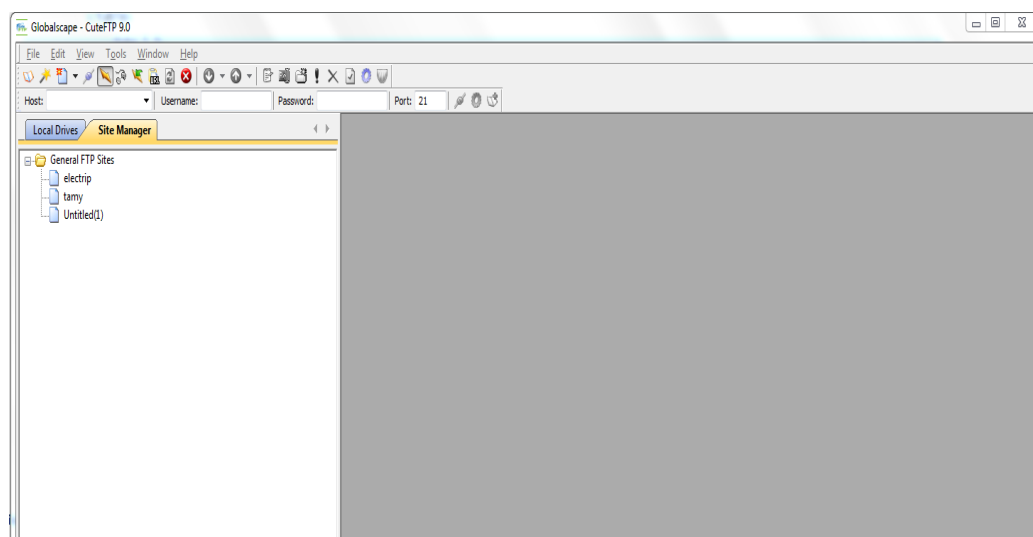


Figura 4.50: Interfaz del CuteFTP

Nos aparecen todas las carpetas del server, de la conexión de hosting creada.

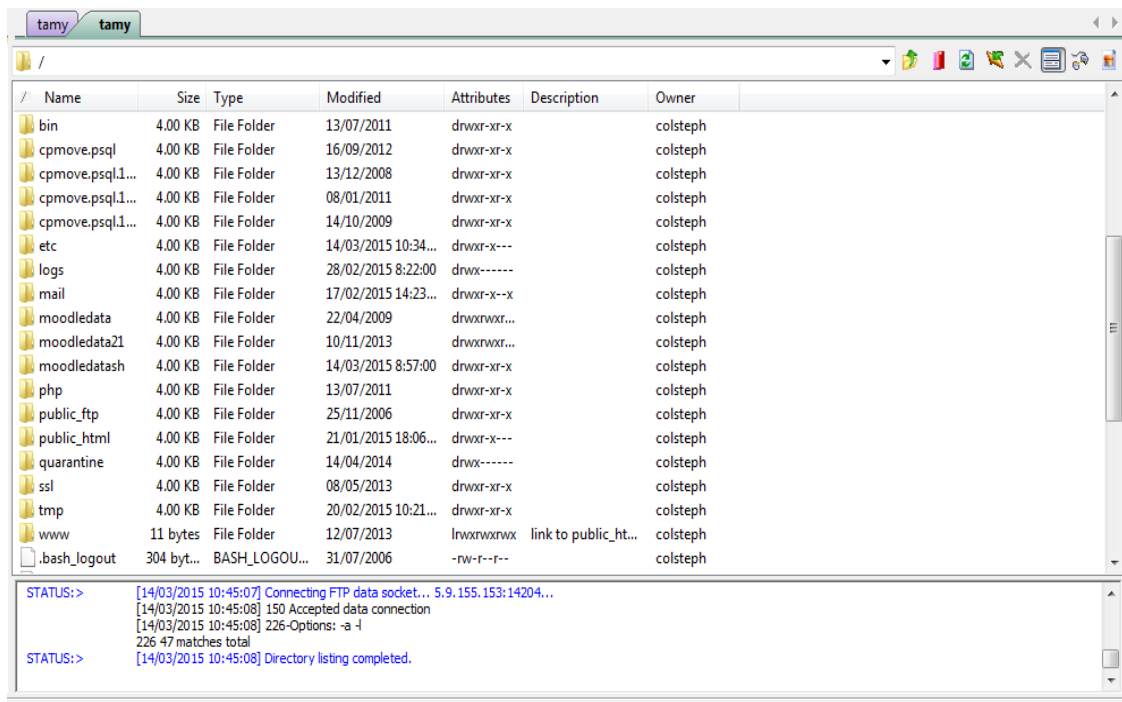


Figura 4.51: Interfaz de las Carpetas del Server

En esta carpeta es en donde debemos agregar la carpeta que contiene la Aplicación Web en sí.

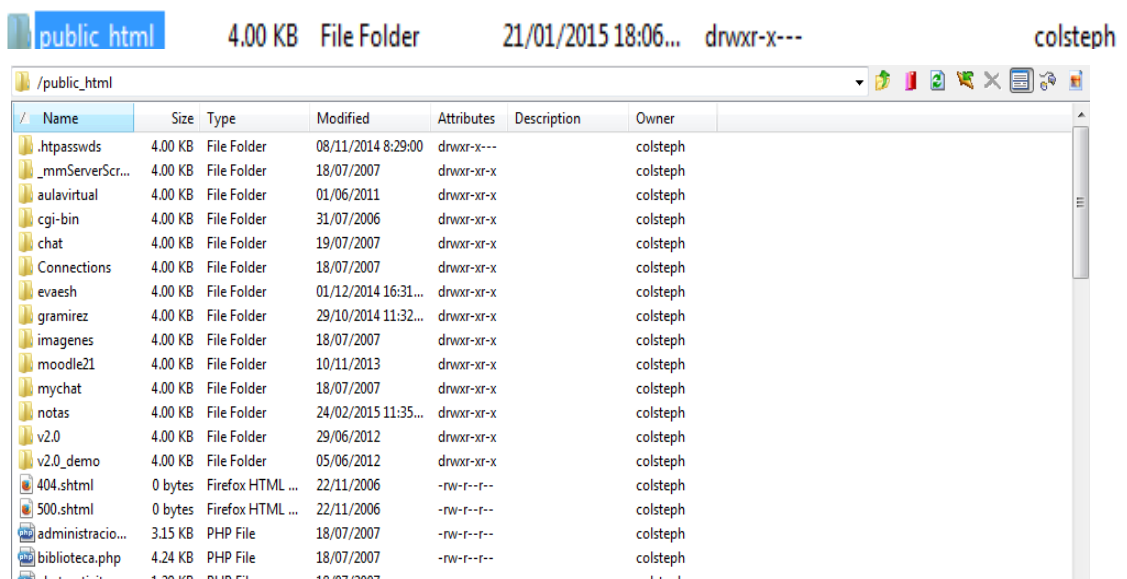
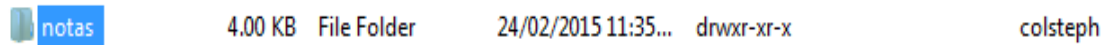
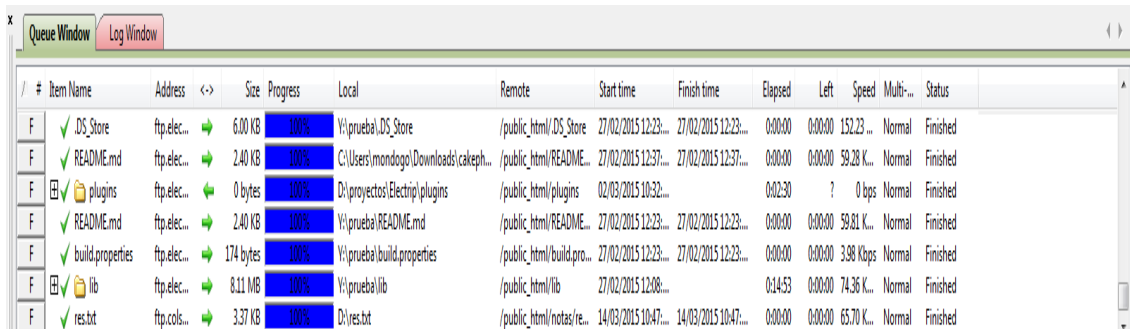


Figura 4.52: Documentos de la Carpeta public_html

En nuestro caso la carpeta nota es la que contiene la Aplicación Web.



Solo cogemos y arrastramos el fichero a transferir, en la parte de abajo especifica si se realizo la transferencia con éxito, o si fallo.



#	Item Name	Address	Size	Progress	Local	Remote	Start time	Finish time	Elapsed	Left	Speed	Multi...	Status
F	✓ .DS_Store	ftp.elec...	6.00 KB	100%	V:\prueba\DS_Store	/public_html/.DS_Store	27/02/2015 12:23:...	27/02/2015 12:23:...	0:00:00	0:00:00	152.23 K...	Normal	Finished
F	✓ README.md	ftp.elec...	2.40 KB	100%	C:\Users\mondogo\Downloads\cakeph...	/public_html/README...	27/02/2015 12:37:...	27/02/2015 12:37:...	0:00:00	0:00:00	59.28 K...	Normal	Finished
F	✓ plugins	ftp.elec...	0 bytes	100%	D:\proyectos\Electripl\plugins	/public_html/plugins	02/03/2015 10:32:...		0:02:30	?	0 bps	Normal	Finished
F	✓ README.md	ftp.elec...	2.40 KB	100%	V:\prueba\README.md	/public_html/README...	27/02/2015 12:23:...	27/02/2015 12:23:...	0:00:00	0:00:00	59.81 K...	Normal	Finished
F	✓ build.properties	ftp.elec...	174 bytes	100%	V:\prueba\build.properties	/public_html/build.pro...	27/02/2015 12:23:...	27/02/2015 12:23:...	0:00:00	0:00:00	3.98 Kbps	Normal	Finished
F	✓ lib	ftp.elec...	8.11 MB	100%	V:\prueba\lib	/public_html/lib	27/02/2015 12:08:...		0:14:53	0:00:00	74.36 K...	Normal	Finished
F	✓ res.txt	ftp.cols...	3.37 KB	100%	D:\res.txt	/public_html/notas/re...	14/03/2015 10:47:...	14/03/2015 10:47:...	0:00:00	0:00:00	65.70 K...	Normal	Finished

Figura 4.53: Pantalla de Verificación de Transferencia de Ficheros

Una vez finalizada la carga de datos al hosting, podemos comprobar su funcionamiento en www.notas.colstephenhawking.edu.ec que es la página del Colegio Stephen Hawking de la ciudad de Ambato.

L	2	A	D	I	P
---	---	---	---	---	---

4.4. Pruebas de Funcionamiento

Finalizada la implementación de la aplicación web, se procedió a realizar las pruebas respectivas, con el fin de detectar posibles errores o fallos en la implementación del software. A continuación se describen las diferentes pruebas realizadas a la Aplicación Web implementada.

4.4.1. Pruebas de Caja Blanca

Se basan en el conocimiento de la lógica interna del código del sistema. Las pruebas contemplan los distintos caminos que se pueden generar gracias a las estructuras condicionales, a los distintos estados del mismo, etc.

Las pruebas de caja blanca se derivan de casos de prueba que aseguren que durante la prueba se han ejecutado todas las sentencias del programa al menos una vez y que se ejercitan todas las condiciones lógicas.

Casos de prueba que permita:

- Garantizar que se ejecutan al menos una vez todos los caminos independientes de cada módulo.
- Se utilizan las decisiones en su parte verdadera y en su parte falsa.
- Se ejecuten todos los bucles en sus límites.
- Se utilizan todas las estructuras de datos internas.

Prueba del Camino Básico

Es una técnica de prueba de Caja Blanca que permite obtener una medida de complejidad lógica para generar un conjunto básico de caminos que se ejecutan por lo menos una vez durante la ejecución del programa.

Notación del Grafo de Flujo

El flujo de control de un programa se representa mediante lo que se llama Grafo de Flujo que representa el flujo de control lógico, el mismo que está formado por nodos los mismos que representan una o más sentencias de código fuente o LDP sin bifurcación.

Para realizar las pruebas de caja blanca utilizaremos la técnica del Camino Básico utilizando el código del programa: **Validación de los Dígitos de una Cedula.**

Tabla 4.46: Prueba de Caja Blanca de la Validación de un E-mail

Código de Validación de un E-mail	Nodos
function valida_email(\$email) {	0
if(ereg("^[_a-z0-9-]+)(\.[_a-z0-9-]+)*@([a-z0-9-]+)(\.[a-z0-9-]+)*(\.[a-z]{2,4})\$", \$email))	1
return true; echo "El mail es válido"; else	2
return false; echo "El mail NO es válido";	3
}	4

Grafo de Flujo de la Validación de un E-mail

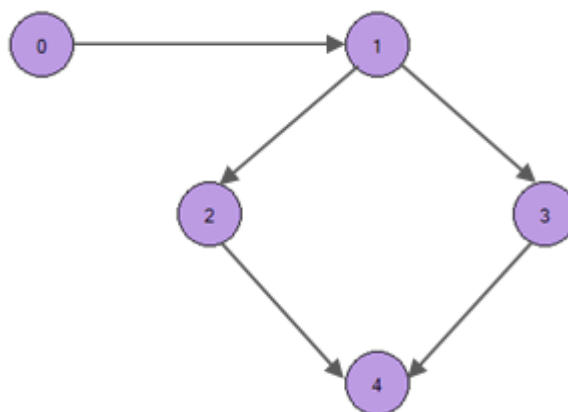


Figura 4.54: Grafo de Flujo para la Validación de un E-mail

Complejidad Ciclomática

$V(G) = A \text{ (arista)} - N \text{ (nodos)} + 2$

$V(G) = 5 - 5 + 2$

$V(G) = 2$

Caminos Básicos

1: 0-1-2-4

2: 0-1-3-4

Ejemplo:

```
$mail = "mail@example.com";  
if(valida_email($mail))  
{  
  echo "El e-mail es válido";  
} else {  
  echo "El e-mail NO es válido";  
}  
FIN
```

4.4.2. Pruebas de Caja Negra

El sistema de pruebas de caja negra no considera la codificación dentro de sus parámetros a evaluar, es decir, que no están basadas en el conocimiento del diseño interno del programa.

Estas pruebas se enfocan en los requerimientos establecidos y en la funcionalidad del sistema web en la cual se hizo énfasis en la revisión de los puntos descritos a continuación:

- Entrada y salida de datos por parte de los usuarios.
- Validar las cajas de texto para el ingreso de caracteres específicos.
- Comprobar que las páginas se ajuste a los datos necesitados y requeridos.
- Mostrar la información correcta y entendible para los usuarios.
- No visualizar información innecesaria y secreta en ninguna caso.

Dentro de ingresos, eliminación y actualización las pruebas de caja negra se basan en la verificación de datos que se envían por medio de la interfaz mediante una petición de ingreso de datos necesarios para alterar la información de la base de datos.

En el proceso de selección las pruebas de caja negra se basan en la presentación de la información en los componentes utilizados para mostrar la información de manera fácil e intuitiva para el usuario.

Ejemplo de pruebas de caja negra utilizando un proceso de la aplicación web que es la consulta de un los Representantes de los Estudiantes de la institución.

Tabla 4.47 Pruebas de Caja Negra - Representantes

Representantes		
Acciones	Casos Esperado	Resultados Obtenidos
Ingreso	Mensajes de error	Mensajes de error describiendo el motivo del error y la solución.
	Información agregada correctamente.	Mensaje de confirmación de inserción.
Actualización	Mensajes de error	Mensajes de error describiendo el motivo del error y la solución.
	Información actualizada correctamente.	Información actualizada y un mensaje de confirmación de actualización.
Eliminación	Mensajes de error	Mensajes de error describiendo el motivo del error y la solución.
	Datos eliminados correctamente.	Datos eliminados y un mensaje de confirmación de eliminación.
Selección	Despliegue de la información del proceso.	Muestra la información de los procesos requeridos.

Demostración de las pruebas de caja negra según las interfaces de la aplicación web:

El resultado esperado dependerá de los datos ingresados, ya sea de manera correcta o de manera incorrecta, siendo esta: datos incompletos, elementos vacíos, etc., mostrando los mensajes de confirmación o de error respectivamente.

Visualización de la información de la Tabla Representantes.

Se muestra información del Estudiante al que pertenece el Representante.

Se muestra la Información del Representante seleccionado

Cédula	Nombres	Apellidos	Celular	Estado
1818181818	pedro	Lopez	3456789345	Activo
1800000000	Luis	Perez	0000000000	Activo
1803994621	wilson	german	0969206562	Activo

Página 1 de 1, mostrados 3 registros de un total de 3 , a partir del registro 1, hasta el registro 3

Representante

Código: 1803994621
 Nombre: wilson
 Apellido: german
 Dirección: pinillo
 Celular: 0969206562
 Teléfono: 1245789632
 Correo: germansss@gmail.com
 Fecha de nacimiento: 1987-04-12
 Genero: Masculino
 Creado: 2014-10-16

Estudiantes

Matricula Actual: 1
 Matriculas Anteriores: 0

Figura 4.55: Prueba de Caja Negra – Representantes

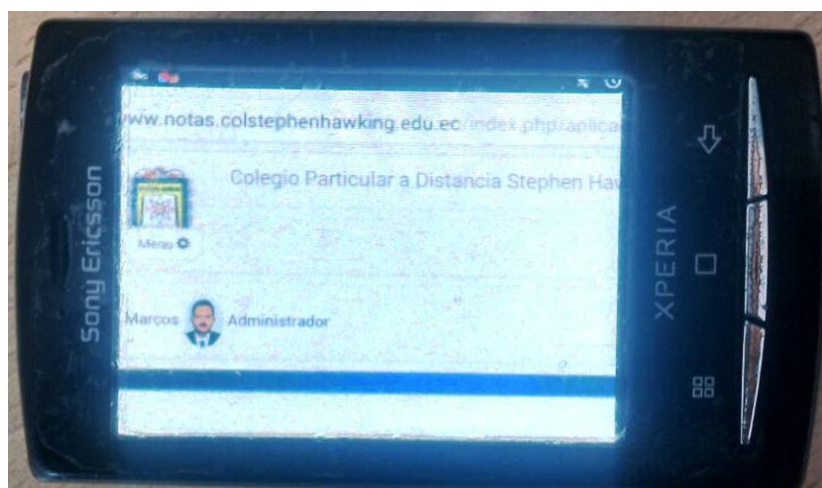
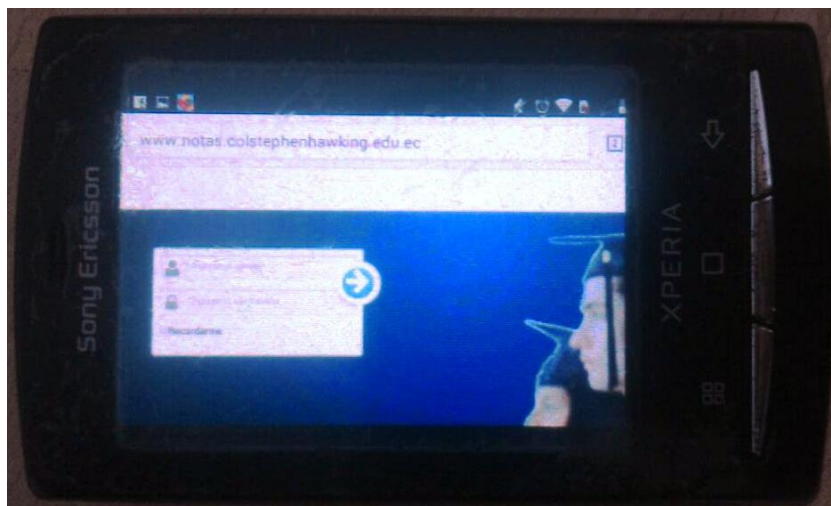
Pruebas de Verificación y Validación

A través de las pruebas de verificación y validación de la aplicación, se ha demostrado que el sistema cumple con las especificaciones funcionales que se establecieron al inicio del proyecto además cumple con los objetivos propuestos, estas pruebas se realizaron con la Ingeniera en Sistemas del Colegio a Distancia Stephen Hawking de la ciudad de Ambato.

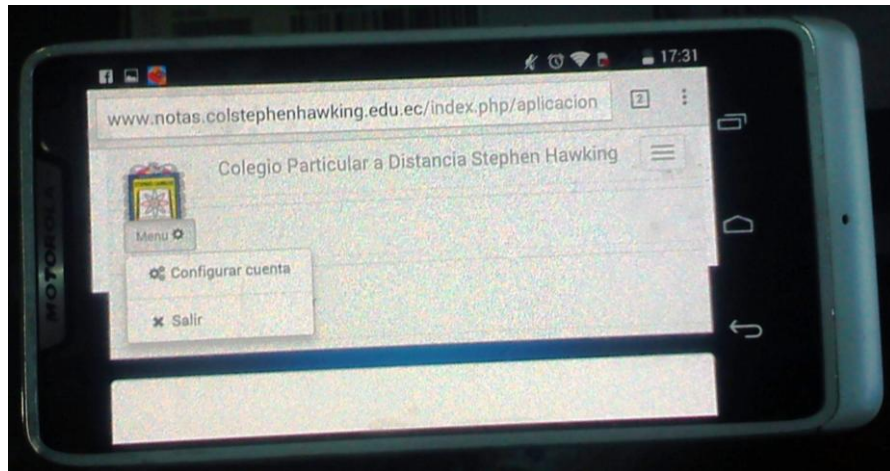
Pruebas en Dispositivos Móviles

Las pruebas se realizaron en diferentes teléfonos móviles, demostrando así la funcionalidad Web Móvil de la aplicación desarrollada.

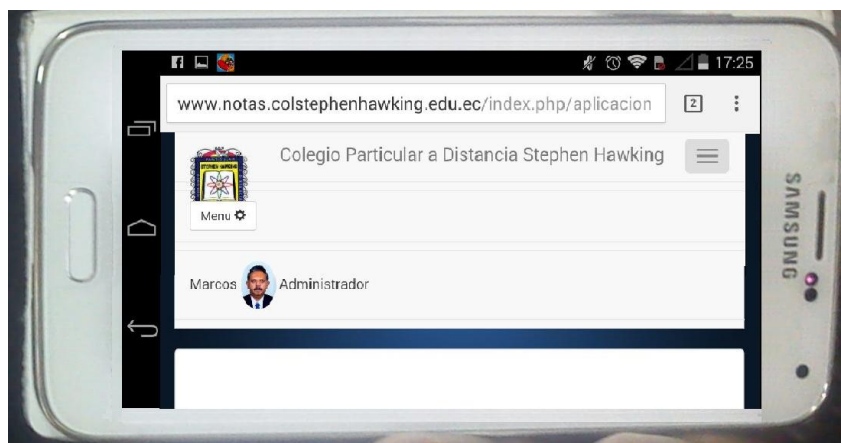
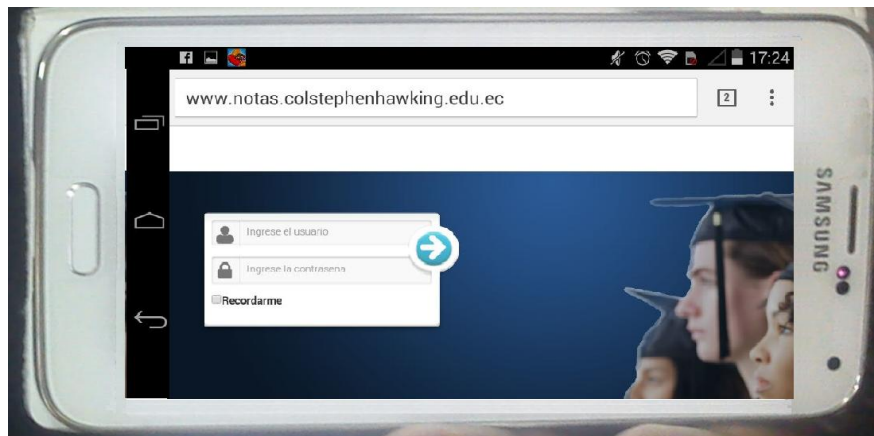
Sony Ericsson Xperia



Motorola



Samsung S4



CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

La forma manual utilizada para la gestión y administración de los datos de estudiantes de la institución generaba un ineficiente procesamiento de la información y no permitía un control adecuado y eficaz de la misma.

Los recursos tecnológicos utilizados hasta antes de la elaboración del sistema para la gestión del área académica de la institución generaban vulnerabilidad en el manejo de la información.

El Sistema académico web utilizando tecnología móvil permitió brindar un mejor servicio en el registro y control de la información, aumentando la confiabilidad en la información obtenida.

La metodología de desarrollo “Iterativo e Incremental” a través de su ciclo de vida permite agrupar las tareas en pequeñas etapas repetitivas (iteraciones), las mismas que facilitan al desarrollador sacar ventaja de lo que se ha aprendido a lo largo del desarrollo anterior, incrementando, versiones entregables del sistema.

El patrón de arquitectura de software MVC facilitó el desarrollo del proyecto, ya que separa los datos y la lógica de negocio de una aplicación, es decir por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.

La realización del sistema fue factible ya que se pudo obtener la información necesaria para el desarrollo del mismo, se contó con el apoyo y la colaboración de todo el personal de la institución beneficiaria los cuales brindaron toda la información del problema, también se tuvo apoyo del personal técnico para el asesoramiento del presente proyecto, esto conllevó a que el proyecto se desarrolle sin ningún problema.

5.2. Recomendaciones

Se recomienda a las autoridades de la institución educativa, designar a una persona especializada en el área de Sistemas o Informática para la administración del Sistema, garantizando así la alta disponibilidad del mismo.

Se recomienda respaldar de la base de datos del Sistema cada cierto tiempo, con el propósito de evitar inconvenientes a futuro, por ejemplo: Pérdida de Información, alteración o manipulación de los datos, virus, etc.

Se recomienda crear un plan de contingencia informática para salvaguardar la información contra los daños producidos por hechos naturales o por el hombre, garantizando así la supervivencia de los datos y entrega de un buen servicio a los usuarios.

Se recomienda realizar una auditoría de seguridad informática periódica dentro de la institución, para conocer la situación exacta de sus activos de información en cuanto a protección, control y medidas de seguridad.

Se recomienda crear un manual sobre políticas de seguridad de la institución para que los usuarios aprendan las reglas a través de sesiones de capacitación y de concienciación.

Bibliografía

[1] Meza G. David, “Análisis, diseño y construcción del sistema de control de calificaciones y servicios virtuales para docentes y estudiantes de estudios presenciales de la ESPE a través del web.”, Tesis de Grado, Escuela Politécnica del Ejercito, Sangolquí, 2005.

[2] Tubay Vergara José Luis, “Desarrollo de una aplicación web para el control de avances académicos y asistencia de docentes.”, Tesis de Grado, Universidad Técnica Estatal de Quevedo, Quevedo, 2010.

[3] Ayala Edison, Enríquez Gabriel, “Análisis, desarrollo e implementación de un sistema orientado a la web para gestión académica.”, Tesis de Grado, Escuela Politécnica del Ejercito, Sangolquí, 2011.

[4] Luján Mora Sergio, “Programación en internet clientes web”, Cottolengo – Alicante (España), Editorial Club Universitario, (2002).

[5] Luján Mora Sergio, “Programación de aplicaciones web: historia, principios básicos y clientes web”, Cottolengo – Alicante (España), Editorial Club Universitario, 2002.

[6] Kroski, Ellyssa. “En el movimiento con la web móvil: bibliotecas y tecnologías móvil”, Editorial Library technology reports, 2008.

[7] Alexander García Dávalos, Tipos de Aplicaciones Móviles, 2001 [online]. Disponible en:

http://augusta.uao.edu.co/moodle/file.php/2896/pres_aplicaciones_moviles_2011.pdf

[8] Natalia Arroyo - Vázquez, Web móvil y bibliotecas, 2009 [online]. Disponible en: http://eprints.rclis.org/13057/1/narroyo_webmovil.pdf

[9] Cesar Villacís, Programación Orienta a Objetos, 2009 [online]. Disponible en: <http://www.slideshare.net/cxvillacis/poo-2515944>

[10] Estanislao García Rey, 2009, Lenguaje de Programación para Aplicaciones Web [online]. Disponible en:

http://www.elinformatico.org/Profesional_informatico_programacion_web.aspx

[11] Damián Pérez Valdés, 2007, Base de datos [online]. Disponible en:
<http://www.maestrosdelweb.com/editorial/%C2%BFque-son-las-bases-de-datos/>

[12] Miguel Ángel Yera, “Diseño y programación de base de datos”, España – Madrid, Editorial Visión Libros, 2009.

[13] C. Pérez, “MySQL para Windows y Linux”, México, Alfaomega Grupo Editor, 2008.

[14] Miguel Jaque Berbero, “Manual de Supervivencia del Administrador de MySQL”, Editorial Autoedición, 2007.

[15] Joaquín Gracia Murugarren, 2009, Conceptos Básicos [online]. Disponible en:
<http://www.webestilo.com/php/php00.phtml>

[16] William Richard Méndez C, 2012, Conceptos de JavaScript [online]. Disponible en:

https://developer.mozilla.org/es/docs/Guía_JavaScript_1.5/Concepto_de_JavaScript

[17] Sonia Rodríguez Ruiz, 2010, Definición de CSS [online]. Disponible en:

<http://www.masadelante.com/faqs/css>

[18] Gabriel Carpio, 2009, Definición de jQuery [online]. Disponible en:

<http://gabrielcarpio.wordpress.com/2009/10/01/definicion-de-jquery/>

[19] Anónimo, 2009, Introducción a JSON [online]. Disponible en:

<http://www.json.org/json-es.html>

[20] Estudiantes de la Universidad Carlos III, 2014, Modelo – Vista – Controlador (MVC) Disponible en:

<http://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador>

[21] Juan Pavón Mestras, 2008, Programación Orientada a Objetos - El patrón Modelo-Vista-Controlador (MVC) [online]. Disponible en:

<https://www.fdi.ucm.es/profesor/jpavon/poo/2.14.MVC.pdf>

[22] Fernando Soriano, 2007, Desarrollo iterativo y creciente [online]. Disponible en:

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

[23] Laura Arnedo, João Gama, Gabriel Prat Masramon, 2010 [online], Disponible en:

<http://www.proyectosagiles.org/desarrollo-iterativo-incremental>

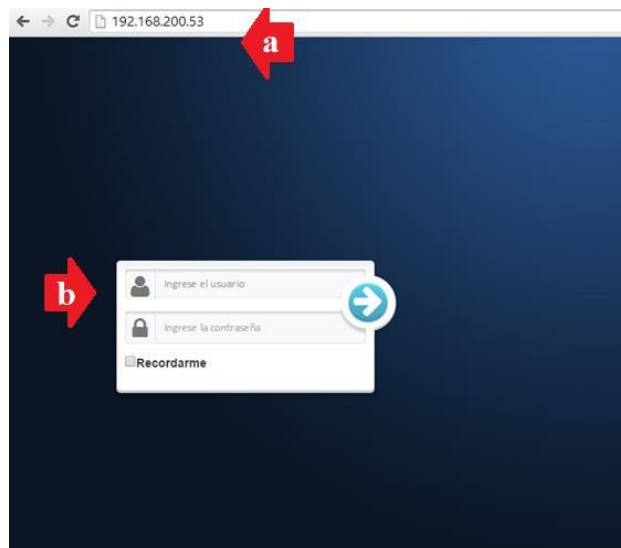
ANEXOS

Anexo 1: Manual de Usuario

MANUAL DE USUARIO

El objetivo del manual es proporcionar a los usuarios una guía para el correcto uso de la aplicación web, explicada de forma sencilla de todas las funcionalidades de cada componente de cada una de las páginas que componen la aplicación.

PANTALLA INICIO DE SESIÓN

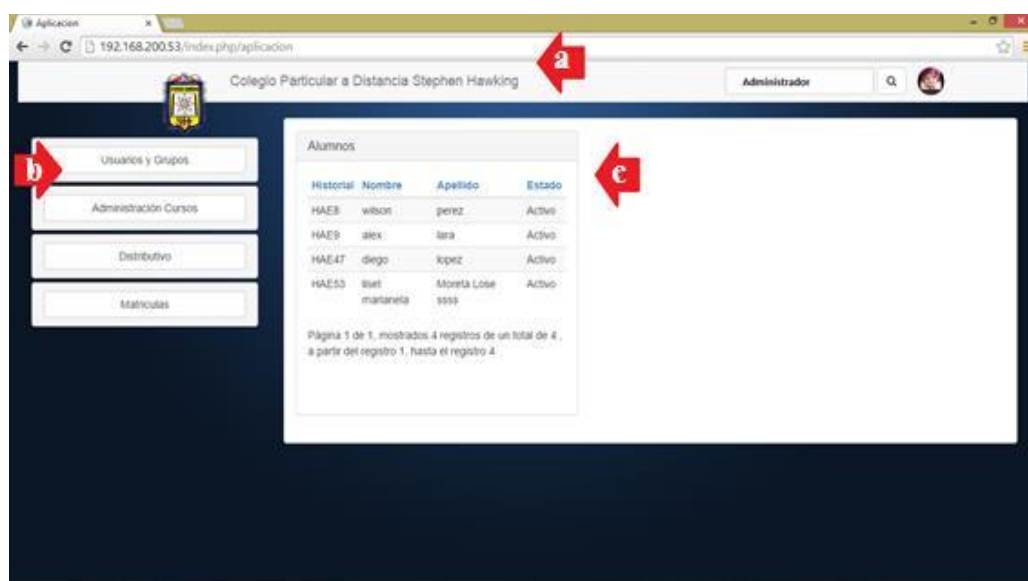


a) Página de Inicio: Ingreso a la página de inicio de sesión de la aplicación de entorno web digitando la url o IP.

b) Inicio de Sesión: Controles de inicio de sesión, ingresar usuario, contraseña y dar clic en botón.

PANEL DE OPCIONES

Una vez ingresado se redirige a la aplicación con el usuario loggeado de acuerdo a los privilegios concedidos y contenido, la cual están divididas en zonas.



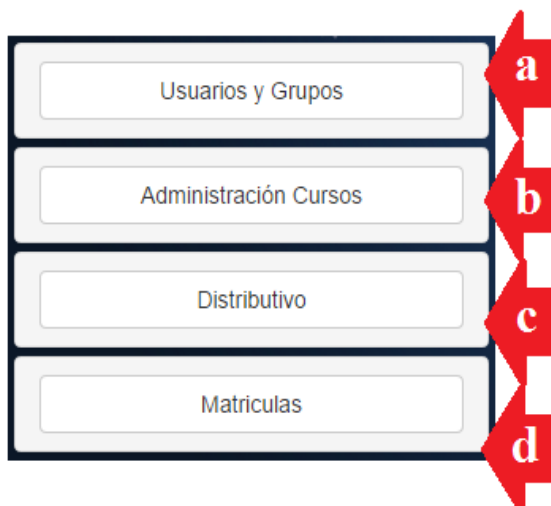
a) Ingreso a la aplicación: Después de haberse logueado se redirige a la página principal e la aplicación y menú de opciones.

b) Panel de control: Esta área contiene las distintas opciones del sistema que nos permitirá manipular la aplicación de acuerdo a las necesidades.

c) Área de contenido: Muestra el contenido de cada opción elegida en el panel de control y que nos permite el ingreso de datos y posterior manipulación.

MENÚ DE CONTROL

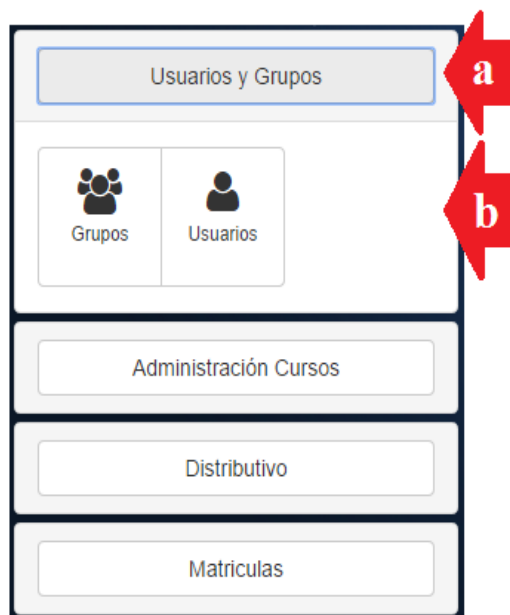
Contiene las distintas acciones a realizar dentro de la aplicación



- a) **Usuarios y Grupos:** Administra a los usuarios que se crean de acuerdo a los privilegios otorgados en cada grupo designado.
- b) **Administración de Cursos:** Administra el tiempo de los ciclos de acuerdo los cursos y sus especialidades.
- c) **Distributivo:** Contiene y administra los distintos Quimestres de acuerdo a la distribución de los Docentes y Materias.
- d) **Matriculas:** Administra a los Estudiantes y sus respectivas matriculas, con sus debidos representantes.

USUARIOS Y GRUPOS

En esta pestaña contiene usuarios y grupos.



- a) **Usuarios y Grupos:** Esta pestaña permite ingresar al submenú donde podemos ingresar los usuarios y ser asignados a los distintos grupos.
- b) **Opciones del Submenú:** Dentro de la pestaña encontramos los botones de Grupos y Usuarios.



Grupos: Permite crear grupos de trabajo para el manejo de la aplicación de acuerdo al rol asignado, y a las funciones que va desempeñar dentro de la opción que trae ese componente.

Nombre	Estado	Cantidad
Administrador	Activo	1
Estudiante	Activo	4
Secretaria	Activo	0
Docente	Activo	3
Anonimo	Activo	0

Página 1 de 1, mostrados 5 registros de un total de 5 , a partir del registro 1, hasta el registro 5

- a) **Ingreso a área de trabajo Grupos:** Dentro del Submenú se despliega el área de trabajo, en el cual visualiza el contenido de los datos almacenados en la base de datos.
- b) **Botones Nuevo-Actualizar:** Permite el ingreso de nuevos grupos por medio de un usuario administrador, de acuerdo al rol creado y/o su actualización.
- c) **Paginación:** Se puede apreciar la página en la que nos encontramos de acuerdo a la cantidad de datos que se encuentra almacenados.

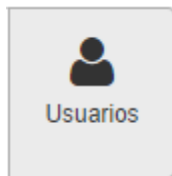
NUEVO GRUPO

Nuevo Grupo







Grupo:

Estado: Activo Inactivo

- a) **Ingreso ventana Grupo:** Se visualiza una nueva ficha después de haber elegido la opción Nuevo Grupo.
- b) **Datos requeridos:** Ingresar de datos en el espacio en blanco y opciones de elección.
- c) **Botones Guardar-Cancelar:** Después de estar seguro que todos los datos necesarios están completos se procede a Guardar la información o caso contrario Cancelar.



Usuarios: Esta opción permite revisar los usuarios creados y almacenados en la base de datos y que se puede visualizar en el área de contenido y a su vez también agregar nuevos usuarios.

a	Usuario	b	Grupo	c	Estado
	admin		Administrador		Activo
	1803994621		Docente		Activo
	1803600970		Docente		Activo
	0503352304		Docente		Activo
			Estudiante		Activo
	german20@ymail.com				
			Estudiante		Activo
	german20a@ymail.com				

- a) **Ingreso al área de trabajo Usuarios:** Se visualiza una ficha de usuarios que se encuentran almacenados en la base de datos.
- b) **Ficha Usuarios:** En esta ficha de puede observar al grupo al que pertenecen y su estado actual.

c) **Botones Nuevo-Actualizar:** Esta opción permite el ingreso de un nuevo o su actualización de acuerdo a la opción elegida.

The image shows a web form titled "Nuevo Usuario" with a close button (X) in the top right corner. The form contains the following fields and controls:

- Usuario:** A text input field.
- Contraseña:** A text input field.
- Grupo:** A dropdown menu with "Administrador" selected.
- Estado:** Radio buttons for "Activo" (selected) and "Inactivo".
- Identificador:** A text input field.
- Buttons:** "Guardar" (blue) and "Cancelar" (white) buttons at the bottom right.

Red arrows with letters point to specific elements: 'a' points to the title bar, 'b' points to the 'Grupo' dropdown, and 'c' points to the 'Identificador' field.

a) **Ingreso ventana Usuario:** Se visualiza una nueva ficha después de haber elegido la opción nuevo Usuario.

b) **Datos requeridos:** Ingresar los datos en los espacios en blanco y elegir en las opciones múltiples.

c) **Botones Guardar-Cancelar:** Después de estar seguro que todos los datos necesarios están completos se procede a Guardar, caso contrario se procede a Cancelar.

ADMINISTRACIÓN CURSOS

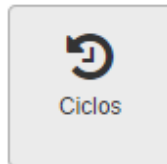
The image shows a vertical menu with the following items:

- Usuarios y Grupos
- Administración Cursos** (highlighted with a blue bar and a red arrow 'a')
- Ciclos
- Secciones
- Especialidades
- Cursos
- Distributivo
- Matriculas

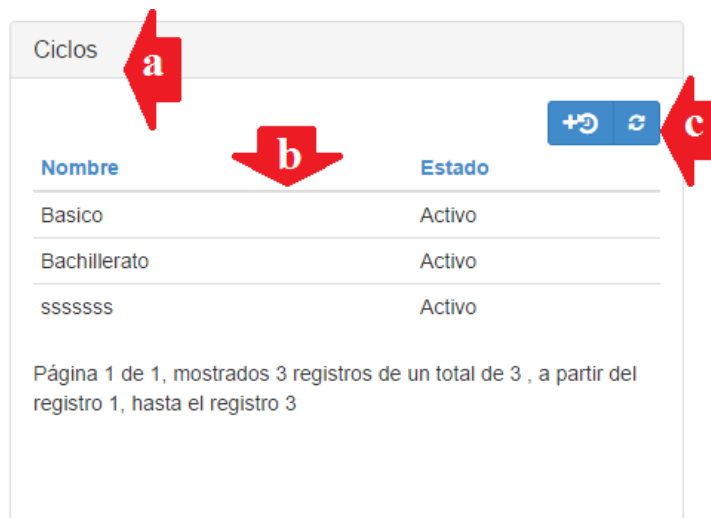
Red arrows with letters point to specific elements: 'a' points to the "Administración Cursos" menu item, and 'b' points to the "Especialidades" menu item.

a) **Administración Cursos:** Esta pestaña permite el ingreso al submenú de las opciones que las componen.

b) **Opciones del Submenú:** Dentro de esta pestaña encontramos los botones de las opciones disponibles.



Ciclos: Este botón permite la manipulación de los distintos ciclos que componen los ciclos educativos, el que permite visualizar los datos almacenados y agregar nuevos ciclos.



Panel de 'Ciclos' que muestra una tabla con los siguientes datos:

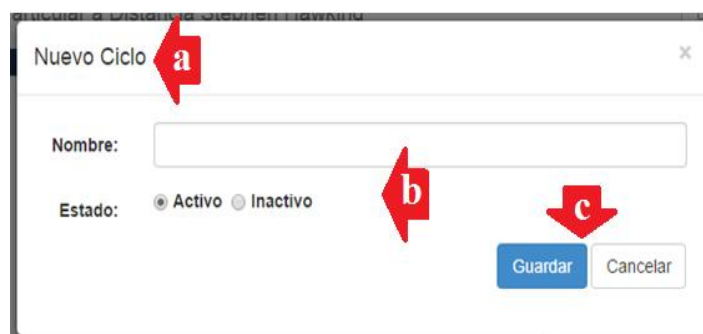
Nombre	Estado
Basico	Activo
Bachillerato	Activo
sssssss	Activo

En la parte inferior de la tabla se muestra: 'Página 1 de 1, mostrados 3 registros de un total de 3 , a partir del registro 1, hasta el registro 3'. En la esquina superior derecha del panel hay botones '+', 'Actualizar' y 'Eliminar'.

a) **Ingreso panel Ciclos:** Permite visualizar la ficha de ciclos, después de haber elegido la opción Ciclos.

b) **Ficha ciclos:** Dentro de la ficha Ciclos se muestra los ciclos y sus estados almacenados den la base de datos.

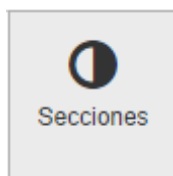
c) **Botones Nuevo-Actualizar:** Permite crear un nuevo ciclo o actualizar los datos.



Formulario 'Nuevo Ciclo' con los siguientes campos:

- Nombre:
- Estado: Activo Inactivo
- Botones: Guardar, Cancelar

- a) **Ingreso ventana Ciclo:** Se aprecia una ficha de nuevo ciclo después de haber elegido la opción de crear nuevo.
- b) **Datos requeridos:** Ingresar los datos en los espacios en blanco y alguna opción de los radio botones.
- d) **Botones Guardar-Cancelar:** Después de estar seguro que todos los datos necesarios están completos se procede a Guardar, caso contrario se procede a Cancelar.



Secciones: Este botón permite la manipulación de las distintas secciones que componen los ciclos educativos, el que permite visualizar los datos almacenados y agregar nuevas secciones.

Secciones	
Nombre	Estado
Nocturna	Activo
Matutina	Activo

Página 1 de 1, mostrados 2 registros de un total de 2 , a partir del registro 1, hasta el registro 2

- a) **Ingreso panel Secciones:** Dentro de esta área de trabajo y después de haber elegido la opción tenemos a Secciones.
- b) **Ficha Secciones:** Permite visualizar los datos almacenados con sus respectivos estados.
- c) **Botones Nuevo-Actualizar:** Esta opción permite el ingreso de un nuevo o su actualización de acuerdo a la opción elegida.

The screenshot shows a form titled "Nueva Sección" with a close button (X) in the top right corner. The form contains the following elements:

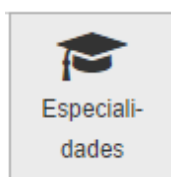
- Nombre:** A text input field.
- Estado:** Two radio buttons labeled "Activo" (selected) and "Inactivo".
- Buttons:** A blue "Guardar" button and a white "Cancelar" button.

Red arrows with letters indicate key features: 'a' points to the title bar, 'b' points to the input fields, and 'c' points to the "Guardar" button.

a) Ingreso ventana Sección: Se aprecia una ficha de nueva sección después de haber elegido la opción de crear nueva sección.

b) Datos requeridos: Ingresar los datos en los espacios en blanco y elegir las demás opciones.

e) Botones Guardar-Cancelar: Después de estar seguro que todos los datos necesarios están completos se procede a Guardar, caso contrario se procede a Cancelar.



Especialidades: Esta opción permite la visualización e ingreso de nuevas especialidades dentro las secciones anteriormente expuestas, dentro un ciclo lectivo.

The screenshot shows a table titled "Especialidades" with a header row and three data rows. A blue button with a graduation cap icon and a refresh icon is located in the top right corner. Red arrows with letters indicate key features: 'a' points to the title bar, 'b' points to the table headers, and 'c' points to the blue button.

Nombre	Estado
Informatica	Activo
Contabilidad	Activo
General	Activo

Página 1 de 1, mostrados 3 registros de un total de 3 , a partir del registro 1, hasta el registro 3

a) **Ingreso panel Especialidades:** Dentro de esta opción y después de haber elegido el botón Especialidades se presenta el área de trabajo Especialidades.

b) **Ficha Especialidades:** Se visualiza los datos ingresados y almacenados en la base de datos.

c) **Botones Nuevo-Actualizar:** Al elegir esta opción nos permite ingresar una nueva Especialidad o en su defecto actualizar según la opción elegida.

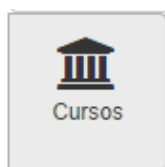


The image shows a screenshot of a web application window titled "Nueva Especialidad". The window contains a form with the following elements: a text input field labeled "Nombre:", a radio button group labeled "Estado:" with options "Activo" (selected) and "Inactivo", and two buttons at the bottom right: "Guardar" (blue) and "Cancelar" (white). Three red arrows with letters 'a', 'b', and 'c' point to the title bar, the "Nombre:" field, and the "Guardar" button, respectively.

a) **Ingreso ventana Especialidad:** Se aprecia una nueva ficha después de haber elegido ingresar nueva Especialidad.

b) **Datos requeridos:** Ingresar datos en los espacios en blanco y las demás opciones requeridas.

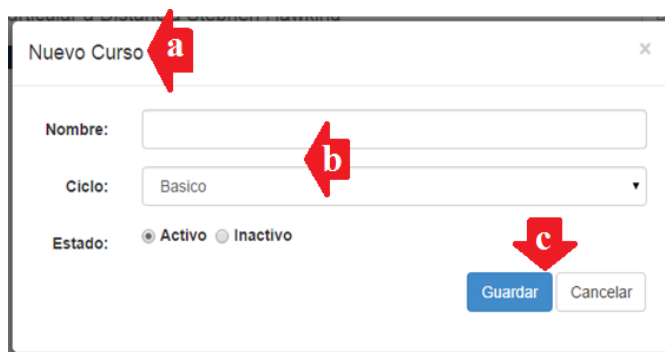
c) **Botones Guardar-Cancelar:** Después de estar seguro que todos los datos necesarios están completos se procede a Guardar, caso contrario se procede a Cancelar.



Cursos: Esta opción permite la visualización de cursos, como también el ingreso de nuevos cursos según las necesidades lo requiera.

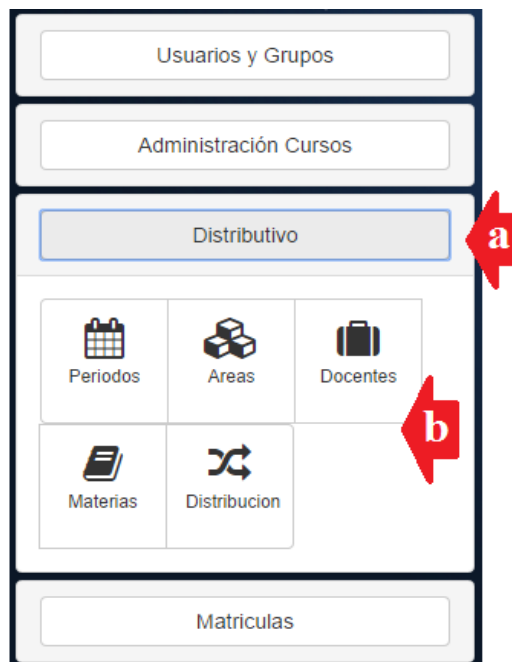


- a) **Ingreso panel Cursos:** Se aprecia en el área de trabajo una nueva ficha de Cursos.
- b) **Ficha Cursos:** Se puede visualizar los cursos ingresados y que se encuentran almacenados en la base de datos.
- c) **Botones Nuevo-Actualizar:** Al elegir esta opción se nos presenta una nueva ventana para ingresar Cursos.



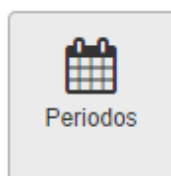
- a) **Ingreso ventana Cursos:** Se presenta una nueva ficha Cursos después de haber elegido la opción ingresar nuevo curso.
- b) **Datos requeridos:** Ingresar los datos en los espacios en blanco y elección de ciclos.
- c) **Botones Guardar-Cancelar:** Después de estar seguro que todos los datos necesarios están completos se procede a Guardar, caso contrario se procede a Cancelar.

DISTRIBUTIVO



a) **Distributivo:** Esta pestaña permite el ingreso al submenú de las opciones que las componen dentro del Distributivo de Trabajo.

b) **Opciones del Submenú:** Dentro de este submenú encontramos las opciones a elegir de acuerdo a las necesidades, entre ellas Quimestres, Áreas y Docentes.



Periodos: Esta opción me permite la visualización del Periodo vigente en el cual se lleva a cabo el ciclo lectivo y sus respectivas opciones.

Nombre	Fecha inicio	Fecha fin	Estado
pruebaperiodo	2014-08-12	2015-08-12	Activo
periodoprueba2	2014-09-02	2014-09-17	Inactivo

Página 1 de 1, mostrados 2 registros de un total de 2 , a partir del registro 1, hasta el registro 2

- a) **Ingreso a panel Periodos:** Se aprecia un área de trabajo Periodos después de haber elegido la opción correspondiente.
- b) **Ficha Periodos:** Se visualiza los datos almacenados en la base de datos con sus respectivas características.
- c) **Botones Nuevo-Actualizar:** Esta opción permite el ingreso de un nuevo periodo o a su vez podemos actualizar nuevos datos.

The image shows a web form titled "Nuevo Periodo" with a close button (X) in the top right corner. The form contains the following fields and controls:

- Nombre:** A text input field.
- Fecha de inicio:** A date selection field with a calendar icon.
- Fecha fin:** A date selection field with a calendar icon.
- Estado:** Radio buttons for "Activo" (selected) and "Inactivo".
- Buttons:** "Guardar" (blue) and "Cancelar" (white) buttons at the bottom right.

Red arrows with letters 'a', 'b', and 'c' point to the title bar, the date fields, and the "Guardar" button, respectively.

- a) **Ingreso ventana Período:** Se visualiza una nueva ficha para ingresar nuevo periodo.
- b) **Datos requeridos:** Ingresar los datos en los espacios en blanco y elegir las fechas requeridas.
- c) **Botones Guardar-Cancelar:** Después de estar seguro que todos los datos necesarios están completos se procede a Guardar, caso contrario se procede a Cancelar.



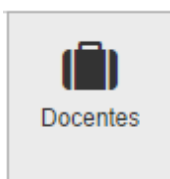
Áreas: Esta opción permite dirigirnos a las áreas que se encuentran registradas e ingresar las que sean necesarias.



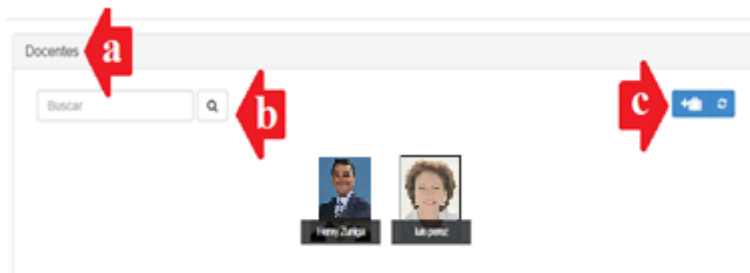
- a) **Ingreso a panel Áreas:** Se aprecia un área de trabajo después de haber elegido a la opción que nos lleva a visualizar su contenido.
- b) **Ficha Áreas:** Se puede visualizar los datos almacenados de las áreas disponibles.
- c) **Botones Nuevo-Actualizar:** Esta opción permite elegir ingresar nuevas áreas que se puede crear o actualizar datos.



- a) **Ingreso ventana Área:** Se presenta una ventana para del ingreso de nuevas áreas.
- b) **Datos requeridos:** Llenar los espacios en blanco y elegir las opciones necesarias.
- c) **Botones Guardar-Cancelar:** Después de estar seguro que todos los datos necesarios están completos se procede a Guardar, caso contrario se procede a Cancelar.

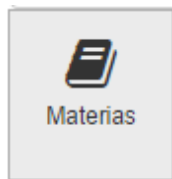


Docentes: Esta opción permite visualizar los docentes disponibles dentro las áreas de cada área asignada y el ingreso de nuevos.



- a) **Ingreso a panel Docentes:** Se visualiza la ficha de Docentes disponibles después de haber accedido a la opción docente.
- b) **Búsqueda Docentes:** Se tiene la opción de búsqueda docentes almacenados en la base de datos.
- c) **Botones Nuevo-Actualizar:** Esta opción permite el ingreso de un nuevo Docente o a su vez podemos actualizar los nuevos datos.

- a) **Ingreso nuevo Docentes:** Al haber elegido nuevo docente se presenta una nueva ficha de ingreso de los datos necesarios para un Docente.
- b) **Datos requeridos:** Llenar los espacios en blanco con los datos necesarios para el registro de un nuevo docente.
- c) **Botones Guardar-Cancelar:** Después de estar seguro que todos los datos necesarios están completos se procede a Guardar, caso contrario se procede a Cancelar.



Materias: Esta opción permite visualizar las materias disponibles o agregar nuevas según las necesidades.

Materias a					
Id	Nombre	Created b	Modified	Estado c	Actions
7	Matematica	2014-08-08	2014-08-08	Activo	+ - ↺ ↻
8	Fisica	2014-08-12	2014-09-26	Activo	
9	Contabilidad	2014-09-30	2014-09-30	Activo	

Página 1 de 1, mostrados 3 registros de un total de 3 , a partir del registro 1, hasta el registro 3

- a) **Ingreso a panel Materias:** Se visualiza una ficha, después de haber elegido la opción Materias.
- b) **Ficha Materias:** Permite visualizar la Materias disponibles en la base de datos.
- c) **Botones Nuevo-Actualizar:** Esta opción permite el ingreso de un nuevo Docente o a su vez podemos actualizar los nuevos datos.

- a) **Ingreso nueva Materias:** Permite visualizar una nueva venta para ingresar nuevas materias.
- b) **Datos requeridos:** Llenar datos en los espacios en blanco y elegir las opciones necesarias.
- c) **Botones Guardar-Cancelar:** Después de estar seguro que todos los datos necesarios están completos se procede a Guardar, caso contrario se procede a Cancelar.



Distribución: Esta opción permite visualizar el distributivo de trabajo del ciclo lectivo con una ficha de presentación.

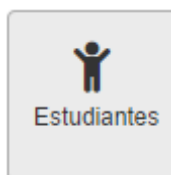
Cursos	
Nombre	Estado
Primer	Activo
Cuarto	Activo

- a) **Ingreso a panel Distributivo de Trabajo:** Como producto de haber elegido esta opción se visualiza el Distributivo en una ficha de presentación.
- b) **Período lectivo:** Dentro de los datos que se visualiza, está el periodo lectivo con los respectivos cursos

MATRICULAS



- a) **Matriculas:** Al elegir esta opción se despliega un submenú opciones para realizar la matricula del estudiante.
- b) **Opciones del Submenú:** Dentro de esta opción, encontramos botones que permiten matricular un estudiante.



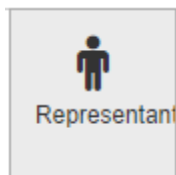
Estudiantes: Ésta botón permite la opción de revisar los estudiantes mediante una lista de estudiantes.

Alumnos			
Historial	Nombre	Apellido	Estado
HAE8	wilson	perez	Activo
HAE9	alex	lara	Activo
HAE47	diego	lopez	Activo
HAE54	liset marianela	Moreta Nose	Activo

Página 1 de 1, mostrados 4 registros de un total de 4 , a partir del registro 1, hasta el registro 4

a) **Ingreso panel Estudiantes:** Se visualiza en el área de trabajo la ficha de estudiantes almenados en la base de datos.

b) **Ficha Estudiantes:** Se detalla la lista de los estudiantes que se encuentran matriculados.



Representantes: Este botón permite el visualizar los representantes registrados en la base de datos y el ingreso de nuevos.

Representantes				
Cédula	Nombres	Apellidos	Célular	Estado
1212121515	WilsonSSSS German	Perez Nata	7777777777	Activo
1803	aaaaaaaa	german	0358784545	Activo
1800000000	asdhjkhdas	ghghjhghj	2345678234	Activo
1803994621	aaaaaaaa	german	0358784545	Activo

Página 1 de 1, mostrados 4 registros de un total de 4 , a partir del registro 1, hasta el registro 4

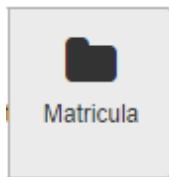
- a) **Ingreso panel Representantes:** En el área de trabajo se visualiza una ficha de Representantes disponibles.
- b) **Ficha Representante:** Al elegir esta opción se detalla los Representantes almacenados en la base de datos.
- c) **Botones Nuevo - Actualizar:** Esta opción permite el ingreso de un nuevo representante o a su vez podemos actualizar los nuevos datos ingresados.

The image shows a web form titled "Nuevo Representante" with a close button (x) in the top right corner. The form is labeled "Representante" and contains the following fields and options:

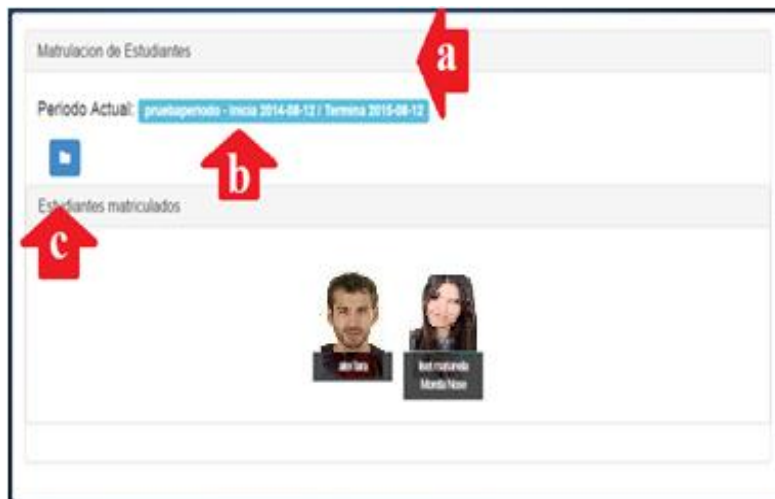
- Cedula:
- Nombres:
- Apellidos:
- Dirección:
- Celular:
- Teléfono:
- E-mail:
- Fecha de nacimiento:
- Género: Masculino Femenino
- Estado: Activo Inactivo

At the bottom right, there are two buttons: "Guardar" (blue) and "Cancelar" (white). Three red arrows with white letters point to specific elements: 'a' points to the title bar, 'b' points to the "Celular" field, and 'c' points to the "Guardar" button.

- a) **Ingreso ventana nuevo Representante:** Al elegir esta opción se presenta una ventana para el ingreso de nuevo representante.
- b) **Datos requeridos:** Llenar los espacios en blanco y demás datos necesarios para completar la información.
- c) **Botones Guardar-Cancelar:** Después de estar seguro que todos los datos necesarios están completos se procede a Guardar, caso contrario se procede a Cancelar.

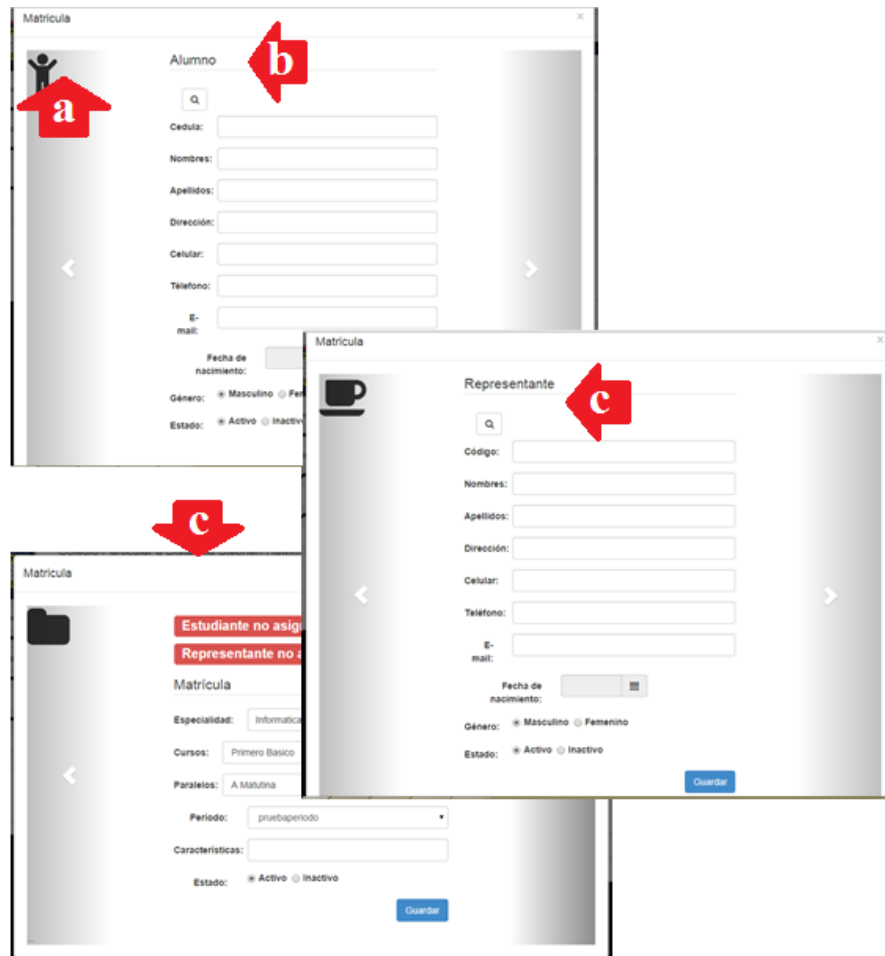


Matrícula: Este botón permite la opción más importante, de matricular a un estudiante con sus datos respectivos



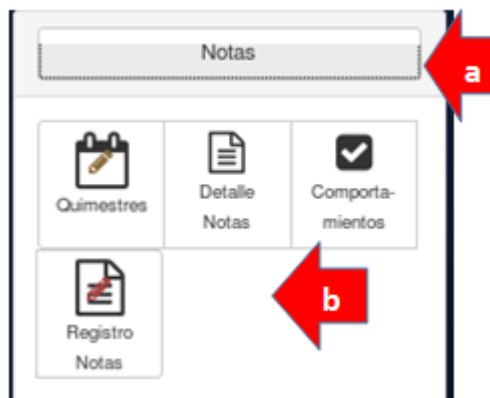
- a) **Ingreso panel Matricula:** Después de haber elegido la opción de matrícula se presenta la matriculación de estudiantes.
- b) **Ficha Matricula:** Se presenta estudiantes registrados en el período actual con opción de nuevas matrículas.
- c) **Botones Nuevo - Actualizar:** Esta opción permite el ingreso de un nuevo representante o a su vez podemos actualizar los nuevos datos ingresados.

La Matricula se procede a realizarse después de tener un sistema con datos ingresados.



- a) **Ingreso ventana nueva Matricula:** Después de elegir el botón Matricula se presenta una nueva ventana de matrícula.
- b) **Ingreso ventana Alumno:** En la nueva venta se presenta el ingreso de un nuevo estudiante o la búsqueda si ya se encuentra previamente registrado.
- c) **Ingreso ventana Representante:** Después de haber llenado los datos necesarios del alumno se presenta la nueva ventana de representantes, los cuales deben llenarse o en su defecto se elige después de la busque de sus datos.
- d) **Ingreso ventana Matricula:** Para poder terminar con el registro de Matricula se procede a matricular al estudiante de acuerdo a su Especialidad, curso y paralelo. Completando con esto, el círculo de información necesario para cumplir con los pasos establecidos.

NOTAS



a) **Notas:** Esta pestaña permite el ingreso al submenú de las opciones que las componen dentro de las Notas.

b) **Opciones del Submenú:** Dentro de este submenú encontramos las opciones a elegir de acuerdo a las necesidades, entre ellas Quimestres, Detalle de notas, Comportamientos y Registro de Notas.



Quimestres: Esta opción me permite la visualización de los Quimestres que se lleva a cabo el ciclo lectivo y sus respectivas opciones.

The image shows the 'Quimestres' panel. At the top is a tab labeled 'Quimestres'. Below it is a table with four columns: 'Nombre', 'Fecha de inicio', 'Fecha de finalización', and 'Estado'. The table contains two rows of data. To the right of the table is a toolbar with two circular icons. Red arrows labeled 'a', 'b', and 'c' point to the 'Quimestres' tab, a row in the table, and the toolbar respectively.

Nombre	Fecha de inicio	Fecha de finalización	Estado
QUIMESTREI	2014-08-06	2014-10-08	Finalizado
QUIMESTREII	2014-10-09	2014-12-09	Activo

Página 1 de 1, mostrados 2 registros de un total de 2 , a partir del registro 1, hasta el registro 2

a) **Ingreso a panel Quimestres:** Se aprecia un área de trabajo Quimestres después de haber elegido la opción correspondiente.

b) **Ficha Quimestres:** Se visualiza los datos almacenados en la base de datos con sus respectivas características.

c) **Botones Nuevo-Actualizar:** Esta opción permite el ingreso de un nuevo quimestre o a su vez podemos actualizar nuevos datos.

The image shows a web form titled "Nuevo Quimestre". It contains the following fields: "Nombre:" with an empty text input; "Fecha de inicio:" with a date picker showing a calendar icon; "Fecha de Fin:" with a date picker showing a calendar icon; and "Estado:" with a dropdown menu currently set to "Activo". At the bottom right, there are two buttons: "Guardar" (blue) and "Cancelar" (white). Three red arrows labeled 'a', 'b', and 'c' point to the title, the date picker icons, and the buttons respectively.

a) **Ingreso ventana Quimestres:** Se visualiza una nueva ficha para ingresar nuevo quimestres.

B) **Datos requeridos:** Ingresar los datos en los espacios en blanco y elegir las fechas requeridas.

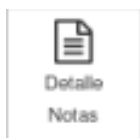
c) **Botones Guardar-Cancelar:** Después de estar seguro que todos los datos necesarios están completos se procede a Guardar, caso contrario se procede a Cancelar.

The image shows a web form titled "Editar Quimestre". It contains the following fields: "Nombre:" with a text input containing "QUIMESTREI"; "Fecha de inicio:" with a date picker showing "2014-08-06" and a calendar icon; "Fecha de finalización:" with a date picker showing "2014-10-08" and a calendar icon; and "Estado:" with a dropdown menu currently set to "Finalizado". At the bottom right, there are two buttons: "Guardar" (blue) and "Cancelar" (white). Three red arrows labeled 'a', 'b', and 'c' point to the title, the date picker icons, and the buttons respectively.

- a) **Área ventana Quimestres Actualizar información:** Se visualiza una nueva ficha para actualizar un quimestre existente.
- b) **Datos requeridos:** Ingresar los datos en los espacios en blanco y elegir las fechas requeridas.
- c) **Botones Guardar-Cancelar:** Después de estar seguro que todos los datos necesarios están completos se procede a Guardar, caso contrario se procede a Cancelar.



- a) **Ventana de Eliminación de Quimestres:** Se visualiza un quimestre existente.
- b) **Datos:** Información del quimestres a eliminarse.
- c) **Botones Guardar-Cancelar:** Después de estar seguro que todos los datos necesarios están completos se procede a Guardar, caso contrario se procede a Cancelar.



Detalle Notas: Esta opción me permite la visualización de las notas que se calificaran al cabo el ciclo lectivo y sus respectivas opciones.

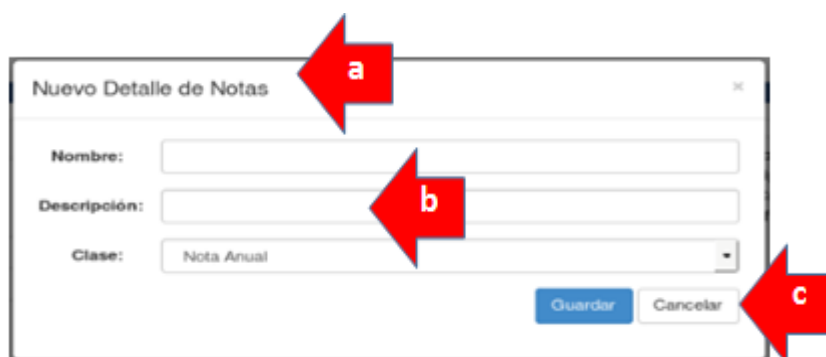


Nombre	Clase
Desarr.Inves.Módulo	Nota Quimestre 80%
Trab.y Taller.Individ	Nota Quimestre 80%
Trab.y Taller Grupal	Nota Quimestre 80%
Lección Oral Escrita	Nota Quimestre 80%
Evalua.Quim.	Nota Quimestre 20%
Examen Supletorio	Nota Anual
Examen Remedial	Nota Anual
Examen Gracia	Nota Anual

Página 1 de 2, mostrados 8 registros de un total de 9 , a partir del registro 1, hasta el registro 8

« 1 2 »

- a) **Ingreso a panel Detalle Notas:** Se aprecia un área de trabajo Detalle Notas después de haber elegido la opción correspondiente.
- b) **Ficha Detalle Notas:** Se visualiza los datos almacenados en la base de datos con sus respectivas características.
- c) **Botones Nuevo-Actualizar:** Esta opción permite el ingreso de un nuevo quimestre o a su vez podemos actualizar nuevos datos.



Nuevo Detalle de Notas

Nombre:

Descripción:

Clase:

- a) **Ingreso ventana Detalle Notas:** Se visualiza una nueva ficha para ingresar nuevo Detalle Notas.
- b) **Datos requeridos:** Ingresar los datos en los espacios en blanco y elegir las fechas requeridas

c) **Botones Guardar-Cancelar:** Después de estar seguro que todos los datos necesarios están completos se procede a Guardar, caso contrario se procede a Cancelar.

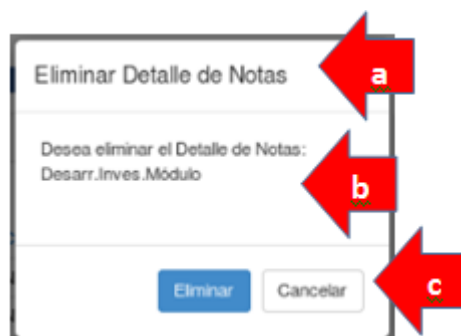


The image shows a web form titled "Editar detalle de notas". It contains three input fields: "Nombre" with the value "Desarr.Inves.Módulo", "Descripción" with the value "Módulo de Investigación", and "Estado" with the value "Nota Quimestre 80%". At the bottom right, there are two buttons: "Guardar" (blue) and "Cancelar" (grey). Three red arrows labeled 'a', 'b', and 'c' point to the title, the description field, and the buttons respectively.

a) **Área ventana Detalle Notas Actualizar información:** Se visualiza una nueva ficha para actualizar un Detalle de Notas existente.

b) **Datos requeridos:** Ingresar los datos en los espacios en blanco y elegir las fechas requeridas.

c) **Botones Guardar-Cancelar:** Después de estar seguro que todos los datos necesarios están completos se procede a Guardar, caso contrario se procede a Cancelar.

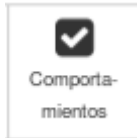


The image shows a dialog box titled "Eliminar Detalle de Notas". It contains the text "Desea eliminar el Detalle de Notas: Desarr.Inves.Módulo". At the bottom, there are two buttons: "Eliminar" (blue) and "Cancelar" (grey). Three red arrows labeled 'a', 'b', and 'c' point to the title, the confirmation text, and the buttons respectively.

a) **Ventana de Eliminación de Detalle Notas:** Se visualiza uno existente.

b) **Datos:** Información del Detalle Notas a eliminarse.

c) **Botones Guardar-Cancelar:** Después de estar seguro que todos los datos necesarios están completos se procede a Guardar, caso contrario se procede a Cancelar.



Comportamientos: Esta opción me permite la visualización de las notas que se calificaron al cabo el ciclo lectivo y sus respectivas opciones.

Nombre	Estado
c	Activo
a	Activo
b	Activo
m	Activo

Página 1 de 1, mostrados 4 registros de un total de 4 , a partir del registro 1, hasta el registro 4

- a) **Ingreso a panel Comportamientos:** Se aprecia un área de trabajo comportamientos después de haber elegido la opción correspondiente.
- b) **Ficha Comportamientos:** Se visualiza los datos almacenados en la base de datos con sus respectivas características.
- c) **Botones Nuevo-Actualizar:** Esta opción permite el ingreso de un nuevo quimestre o a su vez podemos actualizar nuevos datos.

Nuevo Detalle de Notas

Nombre:

Descripción:

Clase:

- a) **Ingreso ventana Comportamientos:** Se visualiza una nueva ficha para ingresar nuevo Comportamientos.
- b) **Datos requeridos:** Ingresar los datos en los espacios en blanco y elegir las fechas requeridas.

c) **Botones Guardar-Cancelar:** Después de estar seguro que todos los datos necesarios están completos se procede a Guardar, caso contrario se procede a Cancelar.

Editar detalle de notas

Nombre: Desarr.Inves.Módulo

Descripción: Módulo de Investigación

Estado: Nota Quimestre 80%

Guardar Cancelar

a) **Área ventana Comportamientos Actualizar información:** Se visualiza una nueva ficha para actualizar un comportamiento existente.

b) **Datos requeridos:** Ingresar los datos en los espacios en blanco y elegir las fechas requeridas.

c) **Botones Guardar-Cancelar:** Después de estar seguro que todos los datos necesarios están completos se procede a Guardar, caso contrario se procede a Cancelar.

Eliminar Detalle de Notas

Desea eliminar el Detalle de Notas:
Desarr.Inves.Módulo

Eliminar Cancelar

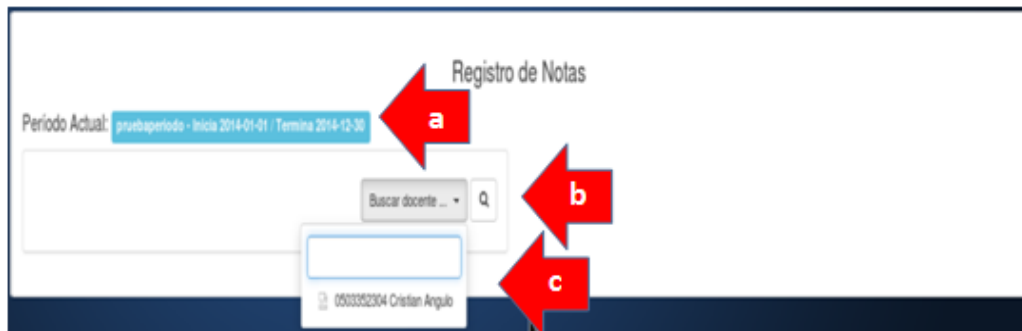
a) **Ventana de Eliminación de Comportamientos:** Se visualiza uno existente.

b) **Datos:** Información del Comportamientos a eliminarse.

c) **Botones Guardar-Cancelar:** Después de estar seguro que todos los datos necesarios están completos se procede a Guardar, caso contrario se procede a Cancelar.



Registro de notas: Esta opción me permite la visualización de las notas que se calificaran al cabo el ciclo lectivo y sus respectivas opciones.



- a) **Periodo actual:** Se aprecia el periodo actual que se está cursando.
- b) **Búsqueda:** Se visualiza los datos almacenados en la base de datos con sus respectivas características del docente y los paralelos que a calificar.
- c) **Docentes:** Se visualiza los docentes existentes en la base de datos.



- d) **Paralelos:** Se visualiza una nueva ficha de con los paralelos de los docentes.

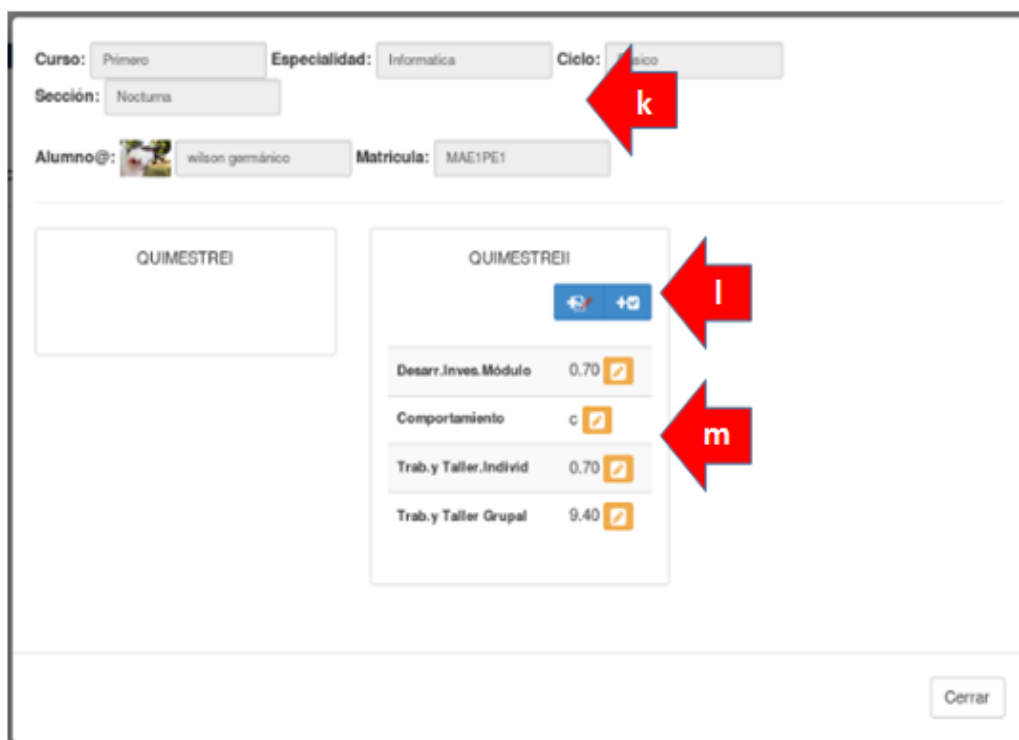


- e) **Especialidad:** Área donde se muestran las especialidades que el docente imparte clases.
- f) **Paralelo:** Nombre del paralelo que a calificar el docente.
- g) **Materia:** Nombre de la materia impartida por el docente.

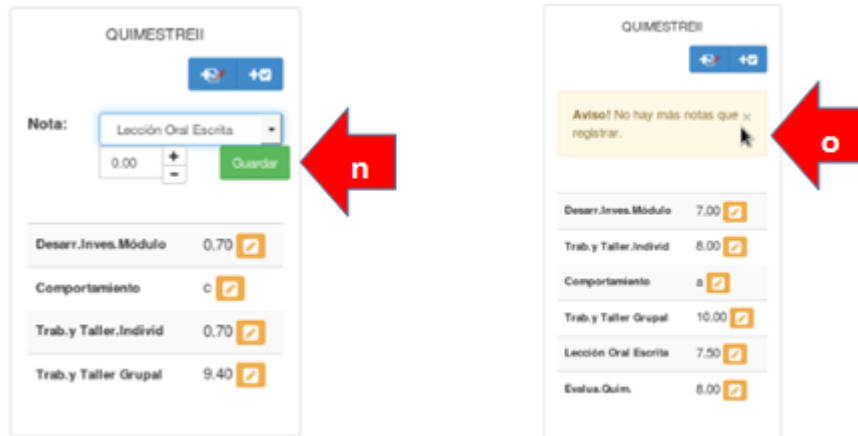
ESTUDIANTES



- h) **Información:** se visualiza una la información de paralelo.
- i) **Datos:** Listado de los estudiantes del paralelo seleccionado.
- j) **Botones:** después de estar seguro que todos los datos necesarios están completos se procede a registrar la nota del estudiante.



- k) **Información:** se visualiza la información.
- l) **Datos:** Botones de registro de notas y comportamiento.
- m) **Notas:** Detalle de las notas y su calificación y puede editar la misma.



- n) **Notas:** Registro de las notas y su calificación.
- o) **Mensaje:** Si no existe más notas que registrar.

Anexo 2: Ficha de Recolección de Notas

ESPECIALIDAD: EDUCACION GENERAL BASICA
CURSO: OCTAVO
AÑO: _____
MES: _____
DIA DE ASIST: _____

UNIDAD EDUCATIVA PCEI TEMPORAL STEPHEN HAWKING
AÑO LECTIVO: SEP-14-JUN15 # 31 C1 **CICLO:** BASICA **SECCION:** MATUTINA
REGISTRO DE CALIFICACIONES, PROMOCIONES Y SUPLETORIOS
PRALELO: A **PROFESOR:** _____
ASIGNATURA: CLUB-INFORMATICA



Nº	NOMINA	Primer Cuatrimestre										Segundo Cuatrimestre										PROMEDIOS Y NOTA FINAL						
		Desarr. Invert. Módulo /10	Trab. y Taller Indiv. /10	Trab. y Taller Grup. /10	Lección Oral Escrita /10	Total Exa. Form. (80%) /10	Exa. Quim. /10	Exa. Sumat. (20%) /10	Nota Final Quim. /10	Comportamiento A B C D E	Desarr. Invert. Módulo /10	Trab. y Taller Indiv. /10	Trab. y Taller Grup. /10	Lección Oral Escrita /10	Total Exa. Quim. /10	Exa. Quim. /10	Exa. Sumat. (20%) /10	Nota Final Quim. /10	Comportamiento A B C D E	Promedio Anual	Examen Selecc. Remedial	Examen Grace	Nota Final Anual	Promedio Comport.				
1	ALLAS YANCHALQUIN NGEL ABEI																											
2	AVACHIRO USULLA NELSON ISRAEL																											
3	BALBECA QUSHIPE EDWIN JOAQUIN																											
4	BERRIO SANCHEZ IMA ANDRES																											
5	CURILLO VIMOS MARIA TORIBA																											
6	GARGES VELASTEGUI ALEX FABIAN																											
7	GUANTULLA CELA JENNY VIVIANA																											
8	KAYAP NANANTAK JUAN MANUEL																											
9	MORETA TORRES MYRIAM DEL ROCIO																											
10	NARANJO PEREZ CHRISTIAN RICARDO																											
11	PEREZ ROMERO ERIKA ELIZABETH																											
12	PULAMUNGA GUANTON JAVIER MIGUEL																											
13	POCAQUIZA PUINNA DARWIN OMAR																											
14	PUINNA IVAN CHRISTIAN ISMAEL																											
15	SENTENO BARRAGAN NATTAL Y SILVANA																											
16	SERRANO NUÑEZ BRYNN PATRICIO																											
17	SOSCO CHANGUANO ROSA HERMINIA																											
18	VILAFUERTE VALDOSPIN WILSON BLADIMIR																											
19	YANSPANTA RAZO WASHINGTON BOLIVAR																											

APROBADO POR EL RECTOR (A)

ELABORADO POR EL TUTOR (A)

Anexo 3: Ficha de Pase de Año de los Estudiantes



UNIDAD EDUCATIVA PCEI STEPHEN HAWKING

Año Lectivo: SEP13-JUN14 #30 C1

INFORME DE
EVALUACION
RESUMEN FINAL

CURSO: OCTAVO
ESPECIALIDAD: CICLO BASICO ALUMN@: CHACHA ALPAPUCHO MYRIAN ALEXANDRA
CICLO: BASICO
SECCION: VESPERTINA **MATRICULA:**

#	ASIGNATURAS	QUIMESTRE		PROMEDIO ANUAL	EXAMEN SUPLETORIO	EXAMEN REMEDIAL	EXAMEN GRACIA	CALIFICACION		RESULTADO
		I	II					CUANTITATIVA	CUALITATIVA	
1	LENGUAJE Y COMUNICACIÓN			0,00				0,00	NAAR	REPROBADO
2	MATEMATICAS			0,00				0,00	NAAR	REPROBADO
3	CIENCIAS NATURALES			0,00				0,00	NAAR	REPROBADO
4	ESTUDIOS SOCIALES			0,00				0,00	NAAR	REPROBADO
5	COMPUTACION			0,00				0,00	NAAR	REPROBADO
6	INGLES	9,40						0,00	NAAR	REPROBADO
TOTAL DE CREDITOS PROMEDIOS										
PROMEDIOS:		9,40		0,00				0,00	NAAR	REPROBADO

EQUIVALENCIAS		
NOTA	EQUIVALENCIA	ABREV
10	SUPERA APRENDIZAJE	SAR
9	DOMINA APRENDIZAJE	DAR
7 - 8	ALCANZA APRENDIZAJE	AAR
5 - 6	PROXIMA ALCANZAR APR.	OAAR
0 - 4	NO ALCANZA APRENDIZAJE	NAAR

ING. PAULINA
BARRIONUEVO

SECRETARIA

REPRESENTANTE