

Universidad de las Ciencias Informáticas

Facultad 1



Título: “*Versión móvil del Sistema de Gestión Académica de Pregrado v1.0.*”

Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autores:

Anisleidy Pérez Suárez

Hector Luis García Tumbeiro

Tutores:

Ing. Sergio René Vazquez Rodriguez

Ing. Dairo Roberto Gil Martín

La Habana, Junio de 2013

“Año 55 de la Revolución”



"Las ideas no necesitan ni de las armas, en la medida en que sean capaces de conquistar a las grandes masas."

Fidel Castro

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año_____.

Firma del autor

Anisleidy Pérez Suárez

Firma del autor

Hector Luis García Tumbeiro

Firma del tutor

Ing. Sergio René Vazquez Rodriguez

Firma del tutor

Ing. Dairo Roberto Gil Martín

AGRADECIMIENTOS

Anisleidy

A mi papito querido, por ser mi gran amor, por estar siempre pendiente de mí, por darme su apoyo en los momentos difíciles y por siempre creer que yo podía lograrlo.

A mi mamá que me apoya y siempre me da su amor incondicional, por aconsejarme y por estar ahí en el momento que la he necesitado.

A toda mi familia que de una forma u otra han ayudado para que este sueño se haga realidad y que siempre creyeron que yo podía lograrlo.

A mi sobrino que es mi fuente de energía e inspiración.

A mi tía Basy que cuidó de mí desde pequeña y que me ha apoyado en todo momento.

A mis tutores Sergio y Dairo, por creer y confiar en mí, por apoyarme y ayudarme en los momentos más difíciles.

A mi compañero de tesis, por ese hermano durante todo este tiempo, por tenerme paciencia y por confiar en que juntos podíamos alcanzar nuestro sueño.

A todas las personas que me han acompañado y brindado su amistad a lo largo de estos años, en especial a Reiniel, Yenni, Lonna, Yoinel, Damis, Reisel, Jose Alberto, Yainier, Ariel, Juan, Alfonso, Beatriz y Ernesto.

En general agradezco a todas las personas que de una forma u otra aportaron su granito para ayudarme a llegar al final y cumplir mi sueño, en especial al profe Yasmany.

Hector Luis

A la Revolución Cubana y en especial al comandante por darme la posibilidad de formarme y convertirme en el profesional que soy hoy.

A mis padres, por toda la confianza depositada en mí y los consejos brindados, por el empuje que siempre me han dado para seguir avanzando y por todo el cariño y aprecio que me han brindado.

A dos personas muy importantes en mi vida, a Susana y Lorenzo por darme todo el apoyo para que terminara mis estudios, por apoyarme y darme ánimo en todo momento, sin ustedes de veras que no lo habría logrado. A su familia y en especial a Rachely.

A mi abuela Antolina y a mi abuelo Hector por el cariño y el amor que me han brindado toda la vida, por ser mis principales educadores, el motivo de inspiración de llevar todos estos sueños adelante y poder siempre contar con ellos.

A mis hermanos, que me han ayudado mucho en mi carrera y me han dado apoyo para seguir adelante, en especial mi hermana Maira que siempre ha estado pendiente de mí y mis estudios.

A mi compañera de tesis, Anis, gracias por soportarme, por tu amistad y por estar siempre ahí presente, esforzándose porque este trabajo se realizara.

A mis tutores por confiar en mí para realizar este trabajo y por guiarme y enseñarme parte de lo que sé hoy, por estar siempre a mi disposición y servirme de guía en mi vida profesional.

A todas las personas de que de una forma u otra han ayudado a mi formación en estos 5 años de estudios, en especial a la profe Eileen.

A todos los grandes compañeros y amistades que me han acompañado en estos años de estudios, en especial a Ricardo, Tony, Daniel, Jose Alberto, Lonna. A Yenni por soportarme todo este tiempo.

A las primeras personas que me ayudaron en esta Universidad y con los cuales fundé una amistad pura, más que una amistad una hermandad, a ti Yoevis gracias, a ti Yoinel gracias, les estaré eternamente agradecidos.

A todo el que de una forma u otra ha contribuido con la realización de este trabajo, en especial al profe Yasmany.

DEDICATORIA

Anisleidy

Dedico este trabajo a las personas más importantes en mi vida, mi mamá y mi papá, los cuales me han apoyado y creído en mí.

A mis abuelos, a mi tía, a mi hermana, a mi sobrino que es mi vida, a mi primo, a Yaimara y en especial a la personita que me cuidó desde pequeña, y que me enseñó a luchar y a nunca darme por vencida, mi querida tía Basy.

A todas las personas que siempre me ha apoyado y que han ayudado para que este sueño se hiciera realidad.

Hector Luis

A todas las personas que confiaron en mí y depositaron su confianza y esfuerzo para que llegara este momento.

Se la dedico a cada uno de ustedes.

RESUMEN

Las aplicaciones destinadas a teléfonos móviles, y entre ellas las web, deben presentar un diseño gráfico especial y cumplir con diferentes características de implementación que permitan desarrollar un producto acorde a las necesidades de los usuarios. En el presente documento se presentan los resultados de la investigación para el desarrollo de una primera Versión móvil del Sistema de Gestión Académica de Pregrado perteneciente al Departamento de Gestión Universitaria de la Universidad de las Ciencias Informáticas. Se estudian sistemas homólogos y se analizan las características generales que deben cumplir las aplicaciones web para móviles para su correcta visualización. Además se describen las tecnologías utilizadas para el desarrollo del sistema propuesto y se presentan los diagramas y artefactos necesarios para el diseño de la solución. Finalmente se describen elementos sobre la implementación de la propuesta y las pruebas que validan su funcionamiento.

Palabras claves: aplicación web, dispositivos celulares, pregrado, versión móvil.

ABSTRACT

Mobile phones applications, including web ones, should present a special graphic design and meet several implementation features that allow developing a product that satisfies the user's needs. This document presents the results of a first mobile version for the Undergraduate Academic Management System in the University Management Department at the University of the Informatics Sciences. A competitor's analysis is carried out along with an analysis of some general graphic design characteristics of web applications for mobiles. Technologies used to develop the proposed system are also described and diagrams and artifacts required for design of the solution are presented as well. Finally some elements of the implementation and tests phases are described in order to provide architectural background and validate the proposal respectively.

Key words: *mobile devices, mobile version, undergraduate, web application.*

ÍNDICE DE CONTENIDO

Introducción.....	1
Capítulo 1: Fundamentación teórica.....	6
1.1. Introducción.....	6
1.2. Telefonía móvil	6
1.2.1. Navegadores para móviles	7
1.2.2. Aplicaciones web para móviles	8
1.2.2.1. Sistemas similares	9
1.2.2.2. Conclusiones del estudio de sistemas similares.....	11
1.3. Herramientas y tecnologías	12
1.3.1. Proceso de desarrollo.....	12
1.3.2. Lenguaje de modelado	15
1.3.3. Herramienta CASE	15
1.3.4. Herramienta para la creación de prototipos de interfaz	15
1.3.5. Lenguajes	16
1.3.6. <i>Framework</i> de desarrollo	17
1.3.7. Entorno de desarrollo	19
1.3.8. Servidor web	19
1.3.9. Herramienta de control de versiones	20
1.3.10. WURFL 1.4.1	20
1.3.11. <i>Opera Mobile Emulator</i> 12.1.....	20
1.4. Conclusiones del capítulo	21
Capítulo 2: Propuesta de solución: análisis y diseño	22

2.1. Introducción.....	22
2.2. Modelo de dominio	22
2.2.1. Definición de las clases del modelo de dominio	23
2.3. Requisitos funcionales	24
2.4. Propuesta de diseño visual del sistema	24
2.4.1. Pautas del Sistema de Gestión Universitaria	24
2.4.2. Pautas de diseño para móviles	26
2.4.3. Propuesta de diseño visual.	28
2.5. Especificación de requisitos.....	29
2.6. Conclusiones del capítulo	31
Capítulo 3: Propuesta de solución: arquitectura	32
3.1. Introducción.....	32
3.2. Requisitos no funcionales	32
3.3. Arquitectura de <i>software</i>	33
3.4. Descripción de la arquitectura del Sistema de Gestión Universitaria	34
3.4.1. GUUD.....	34
3.4.2. Arquitectura Cliente/Servidor	35
3.4.3. Patrón arquitectónico.....	35
3.4.4. Estructura del Sistema de Gestión Universitaria	36
3.5. Descripción de la arquitectura de la propuesta de solución	38
3.5.1. jQuery Mobile.....	38
3.6. Integración de la solución con el Sistema de Gestión Universitaria	40
3.7. Patrones de diseño	41

3.7.1. Patrones GRASP	41
3.7.2. Patrones GoF.....	43
3.8. Conclusiones del capítulo	44
Capítulo 4: Implementación y pruebas	45
4.1. Introducción.....	45
4.2. Modelo de despliegue	45
4.3. Estándares de codificación.....	45
4.4. Detalles de la implementación de la solución	48
4.5. Pruebas	51
4.5.1. Pruebas de funcionalidad	52
4.5.1.1. Pruebas de caja negra.....	52
4.5.1.2. Diseño de casos de prueba	52
4.5.2. Pruebas de integración.....	55
4.5.3. Pruebas con teléfonos celulares reales y emuladores	55
4.5.4. Pruebas con usuarios finales	55
4.6. Resultados de las pruebas	56
4.7. Conclusiones del capítulo	57
Conclusiones.....	58
Recomendaciones	59
Glosario de términos	60
Bibliografía consultada	62
Bibliografía referenciada	64
Anexos.....	68

ÍNDICE DE FIGURAS

Figura 1: Modelo de dominio.	23
Figura 2: Vista de presentación del SGU.	25
Figura 3: Vista de escritorio del SGU.	25
Figura 4: Vista de gestión de procesos del SGU.	26
Figura 5: Vista de presentación de la versión móvil.	28
Figura 6: Vista de escritorio de la versión móvil.	29
Figura 7: Vista de gestión de procesos de la versión móvil.	29
Figura 8: Flujo de información de GUUD.	35
Figura 9: Arquitectura Cliente/Servidor (SGU).	35
Figura 10: Funcionamiento del patrón Modelo-Vista-Controlador.	36
Figura 11: Directorio <i>base</i> , <i>application</i> y <i>public</i>	37
Figura 12: Sistema Pregrado, módulo carrera.	37
Figura 13: Arquitectura Cliente/Servidor (Pregrado móvil).	39
Figura 14: Implementación del patrón MVC.	39
Figura 15: Vista del módulo “Estudiante móvil”.	40
Figura 16: Modelo de despliegue.	45
Figura 17: Resultados de las pruebas realizadas.	56

ÍNDICE DE TABLAS

Tabla 1: Especificación de requisitos: Buscar persona.	30
Tabla 2: Caso de prueba: Buscar persona.	54
Tabla 3: Resultados del <i>test</i> de usuario.	57

ÍNDICE DE ANEXOS

Anexo 1: Especificación de requisitos: Autenticar usuario.	68
Anexo 2: Especificación de requisitos: Mostrar el resumen de evaluaciones.	69
Anexo 3: Especificación de requisitos: Mostrar registro de asignatura.	70
Anexo 4: Especificación de requisitos: Mostrar plan de estudio.	71
Anexo 5: Especificación de requisitos: Mostrar persona.	71
Anexo 6: Especificación de requisitos: Mostrar el resumen de evaluaciones de persona.	72
Anexo 7: Especificación de requisitos: Registrar asistencia.	73
Anexo 8: Especificación de requisito: Consultar registro de asistencia.	75
Anexo 9: Aplicaciones disponibles para la persona autenticada.	75
Anexo 10: Función <i>acIMovil</i>	78
Anexo 11: Función <i>index</i> de la clase controladora <i>Autenticar</i>	79
Anexo 12: Función <i>index</i> de la clase controladora <i>acceso</i>	80
Anexo 13: Función <i>generarToken</i>	81
Anexo 14: Función <i>obtenerElementoDadoToken</i>	83
Anexo 15: Función <i>obtenerACLDadoToken</i>	84
Anexo 16: Función <i>index</i> de la clase controladora <i>defaultindex</i>	86
Anexo 17: Función <i>portadaMovil</i> de la clase controladora <i>portada</i>	87
Anexo 18: Función <i>index</i> de la clase controladora <i>portada</i>	87
Anexo 19: Atributo “ <i>movil</i> ”	89
Anexo 20: Función <i>obtenerAplicaciones</i>	89
Anexo 21: Función <i>obtenerNodosMoviles</i>	90
Anexo 22: Función <i>obtenerNodosMovil</i>	90

Anexo 23: Listado de funcionalidades adicionadas en el archivo IOC.....	91
Anexo 24: Funciones <i>messageMovil</i> y <i>errorMovil</i>	92
Anexo 25: Imágenes del sistema desde emuladores.	93
Anexo 26: <i>Test</i> de usuario.	95
Anexo 27: Caso de prueba: Autenticar usuario.....	96
Anexo 28: Caso de prueba: Mostrar el resumen de evaluaciones.	97
Anexo 29: Caso de prueba: Mostrar registro de asignaturas.	98
Anexo 30: Caso de prueba: Mostrar plan de estudio.	98
Anexo 31: Caso de prueba: Mostrar persona.	99
Anexo 32: Caso de prueba: Mostrar el resumen de evaluaciones de persona.	99
Anexo 33: Caso de prueba: Registrar asistencia.	100
Anexo 34: Caso de prueba: Consultar registro de asistencia.....	103

INTRODUCCIÓN

En los últimos años se ha apreciado un creciente desarrollo de la telefonía celular y un incremento del uso de Internet en las actividades cotidianas. La telefonía móvil más allá de limitarse solo a llamar o enviar mensajes de texto ha incorporado otras funcionalidades que van desde cámara de fotos, agenda electrónica, reloj despertador, calculadora, hasta poder acceder a la red de redes.

El número de usuarios conectados a Internet desde teléfonos celulares aumenta cada día. Según la consultoría Nielsen, en el año 2011 existían 52 435 000 usuarios conectados desde teléfonos móviles a Internet, mientras que en el año 2012 la cifra era de 95 176 000 (1). A pesar del desarrollo alcanzado por los dispositivos celulares no todos los servicios que se ofrecen en Internet se pueden visualizar correctamente desde ellos. Los teléfonos móviles presentan varias limitaciones entre las que pueden encontrarse: un teclado muy pequeño para la entrada de la información, una pantalla de reducidas dimensiones por lo que el despliegue de la información es limitado y el tiempo de respuesta de los sitios convencionales en este tipo de dispositivo es muy lento. Debido a esto es necesario desarrollar aplicaciones especialmente diseñadas para adaptarse a los teléfonos celulares, estas aplicaciones deben cumplir con una serie de características o buenas prácticas que intentan solucionar los problemas planteados con anterioridad.

En los últimos años Cuba ha acelerado considerablemente la utilización de la telefonía móvil, donde los jóvenes han sido el motor impulsor hacia el acercamiento a estas tecnologías, por ese motivo en las universidades cada vez son más las personas que poseen dispositivos celulares. Al mismo tiempo en las universidades del país se genera una gran cantidad de información relacionada con las áreas existentes en las mismas, debido a esto se hace necesario gestionar toda la información de forma automática.

En la Universidad de las Ciencias Informáticas se encuentra funcionando el Sistema de Gestión Universitaria que integra sistemas que gestionan los procesos sustantivos de la universidad. Para cumplir con su objeto social el sistema cubre las siguientes áreas de proceso: Pregrado, Postgrado, Investigación, Residencia, Biblioteca, Laboratorio, Egreso, Cooperación, Ingreso, Producción, Extensión y Teleformación. El Sistema de Gestión Académica de Pregrado (SGAP) se encarga de gestionar la información relacionada con la vida académica de los estudiantes, siendo este uno de los principales procesos de la institución. Por esta razón y por la importancia que tiene para la comunidad universitaria actualmente es uno de los sistemas de mayor acceso en el Sistema de Gestión Universitaria.

Uno de los problemas que resuelve este sistema es el rápido acceso a la consulta del expediente académico y el plan de estudio de una persona, así como el registro y consulta de la asistencia de estudiantes a clases, eliminando la necesidad de consultar los expedientes físicos (propensos al deterioro y difíciles de analizar cuando son varios grupos o grupos muy grandes). Estas acciones pueden ser realizadas desde una computadora de escritorio o una laptop, equipos complejos de ser trasladados o abiertos en muchos ámbitos sociales. Es por esta razón que se considera que puede ser compleja la ejecución de dichas funcionalidades en cualquier lugar donde se tenga acceso a una red inalámbrica, las que aumentan en número debido a la cantidad (en constante aumento) de usuarios que tienen acceso a dispositivos que pueden acceder a las mismas.

Actualmente el Sistema de Gestión Universitaria tiene pautas de diseño que describen la interfaz visual para el desarrollo de nuevos sistemas y módulos. Estas pautas fueron diseñadas específicamente para sistemas que son utilizados desde computadoras de escritorio, por lo que no cubren el diseño de aplicaciones destinadas a las pantallas pequeñas y la restringida navegabilidad de los teléfonos celulares.

Además, al acceder al actual Sistema de Gestión Académica de Pregrado desde dispositivos móviles algunos de los componentes que presenta se distorsionan o cambian de lugar, ejemplo de esto, es el campo de dominio en la vista de autenticar donde este se sale de la interfaz visual. Por otra parte, el formato original del texto no es legible, y el movimiento horizontal y vertical junto a la posibilidad de ampliar y reducir el tamaño del sistema dificulta la navegación por el sistema.

Teniendo en cuenta estas limitaciones se plantea como **problema de investigación**: ¿Cómo acceder a funcionalidades de consulta y registrar asistencia en el Sistema de Gestión Académica de Pregrado mediante conexión inalámbrica desde teléfonos celulares?

El **objeto de estudio** de la presente investigación es el desarrollo de aplicaciones web orientadas a los teléfonos celulares, enfocando el **campo de acción** en el proceso de desarrollo de la Versión móvil del Sistema de Gestión Académica de Pregrado.

Para darle solución al problema mencionado anteriormente se define como **objetivo general** de la investigación: desarrollar la Versión móvil del Sistema de Gestión Académica de Pregrado que permita acceder a funcionalidades de consulta y registrar asistencia mediante conexión inalámbrica desde teléfonos celulares.

Para lograr el cumplimiento del objetivo propuesto se definen las siguientes **tareas de investigación**:

1. Confección del diseño teórico-metodológico sobre el desarrollo de sistemas web para dispositivos móviles.
2. Fundamentación de las tendencias actuales, tecnologías y conceptos relacionados con el desarrollo de sistemas web para dispositivos móviles.
3. Descripción de las buenas prácticas para el diseño de interfaces dedicadas a los dispositivos móviles.
4. Análisis de los portales desarrollados para ser accedidos desde teléfonos celulares.
5. Análisis de las tecnologías, herramientas y lenguaje de programación para el desarrollo de las funcionalidades.
6. Análisis del proceso de desarrollo de *software* para la guía del desarrollo.
7. Análisis de la propuesta de diseño del Sistema de Gestión Académica de Pregrado.
8. Análisis de los estándares de codificación del centro CENIA¹.
9. Definición de la propuesta de solución teniendo en cuenta los elementos del diseño para móviles.
10. Creación de las interfaces de usuario de la Versión móvil del Sistema de Gestión Académica de Pregrado.
11. Especificación de los requisitos funcionales, no funcionales y del dominio que el sistema debe cumplir.
12. Implementación de los requisitos descritos.
13. Elaboración de los casos de prueba de funcionalidad.
14. Ejecución de las pruebas de funcionalidades del sistema.

¹ Centro de Informatización Universitaria.

Al finalizar la investigación es obtenida una aplicación que posibilita el acceso a funcionalidades de consulta y gestión de información del Sistema de Gestión Académica de Pregrado desde teléfonos celulares, proporcionando el acceso desde cualquier red inalámbrica en cualquier momento. Además son definidas pautas de diseño para el desarrollo de interfaces que faciliten la consulta del expediente académico y del plan de estudio cuando se cuente con un teléfono celular. Finalmente, la versión móvil constituye un valor agregado del Sistema de Gestión Académica de Pregrado, mejorando su competitividad y comercialización en el mercado.

Los **Métodos científicos** utilizados para el desarrollo de la presente investigación son:

Métodos teóricos

Histórico lógico: Este método ayuda a comprender el surgimiento y evolución de la telefonía móvil, así como a realizar un estudio sobre el desarrollo de aplicaciones web para teléfonos móviles a partir de la consulta de la bibliografía especializada en el tema.

Analítico sintético: Este método permite analizar y comprender la documentación relacionada con el desarrollo de aplicaciones web para dispositivos móviles, de forma tal que se puedan extraer los elementos fundamentales relacionados con el objeto de estudio.

El contenido de esta investigación se encuentra desglosado en 4 capítulos, conclusiones, recomendaciones, glosario de términos bibliografía y anexos.

En el **Capítulo 1: Fundamentación teórica**, se abordan diferentes temas relacionados con la telefonía móvil y las comunicaciones inalámbricas. También incluye los resultados del estudio realizado sobre el estado del arte a nivel nacional e internacional sobre diferentes aplicaciones web para dispositivos móviles. Además se realiza una descripción de las herramientas y tecnologías a utilizar en el desarrollo del sistema.

En el **Capítulo 2: Propuesta solución: análisis y diseño**, se realiza la descripción de la propuesta de solución. Además se elabora el modelo de dominio, se definen y especifican los requisitos funcionales del sistema. También se analizan los factores a tener en cuenta para el desarrollo de aplicaciones destinadas a dispositivos móviles.

En el **Capítulo 3: Propuesta solución: arquitectura**, se define la propuesta de la arquitectura del sistema, para ello se describe la arquitectura del Sistema de Gestión Universitaria. Además se incluyen los requisitos no funcionales del sistema y los patrones de diseño.

En el **Capítulo 4: Implementación y pruebas**, se hace referencia a los elementos de implementación, como los estándares de codificación a utilizar y la estructura que va a tener el sistema. También se definen los elementos relacionados con las pruebas efectuadas al sistema para verificar que funcione acorde a los requisitos especificados.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

En este capítulo se describen los principales elementos y conceptos relacionados con las comunicaciones inalámbricas y la telefonía móvil. También incluye los resultados del estudio realizado sobre el estado del arte a nivel nacional e internacional sobre diferentes aplicaciones web destinados especialmente para dispositivos móviles. Además se hace referencia a las herramientas y tecnologías a utilizar en el desarrollo de la solución..

1.2. Telefonía móvil

Guarín (2005) define la telefonía celular como “el servicio privado de telecomunicación que funciona con base en células (celdas de comunicación) conectadas entre sí por medio de señales de radio, cuyas frecuencias son asignadas a los usuarios para que puedan efectuar llamadas nacionales e internacionales, gracias a la interconexión permanente entre las diferentes celdas, y entre éstas y los satélites de comunicación. Cada célula cubre un área geográfica determinada y está controlada por un centro de comunicación, el cual dispone de un conjunto de antenas receptoras-transmisoras de señales” (2).

En 1979 tuvo lugar la aparición de la primera generación (1G) de la telefonía móvil la cual estaba basada en las redes celulares con estaciones de base relativamente cercanas. Estos teléfonos se utilizaban estrictamente para la transmisión de voz y su tamaño era mayor al de hoy en día, debido a que fueron diseñados para ser utilizados en los automóviles (3).

La segunda generación (2G) surge en los años 90 dando paso a dispositivos celulares de pequeños tamaños, eliminando de esta forma a los de la 1G. Los protocolos empleados en esta generación eran más sofisticados, brindando la posibilidad de ofrecer nuevos servicios como los SMS (*Short Message Service*). A causa de una serie de problemas que presentaba la 2G surge la generación 2.5 donde la tecnología más conocida era GPRS (*General Packet Radio Service*) (3).

La tercera generación (3G) tuvo lugar a inicios del siglo XXI pues los teléfonos comenzaron a incorporar una serie de servicios que hasta ese momento no tenían, entre ellos se pueden mencionar: las cámaras fotográficas y los juegos 3D (3). Además incluyen la posibilidad de navegar por Internet mediante

las comunicaciones inalámbricas, motivo por el cual los teléfonos de esta generación incluyen navegadores adaptados a las características de los mismos.

Esta forma de comunicación proporciona movilidad y flexibilidad, siendo complicado de lograr con redes cableadas. Esto hace posible que las limitaciones de espacio y tiempo desaparezcan, lo que significa que se puede establecer comunicación desde cualquier lugar sin tener dependencia del sistema cableado.

“Una comunicación inalámbrica es cuando los medios de unión entre sistemas no son con cables. Sus principales ventajas son que permiten una facilidad de emplazamiento y reubicación, evitando la necesidad de establecer un cableado y rapidez en la instalación” (4).

Para establecer una comunicación inalámbrica entre un dispositivo móvil y la red de redes es necesario contar con redes de conexión inalámbricas, siendo *Wireless Fidelity*² una de la más utilizadas.

1.2.1. Navegadores para móviles

Un navegador es un “programa o aplicación informática que se usa para navegar por las redes informáticas y acceder a documentos, imágenes y demás información” (5).

Un navegador web interpreta archivos y sitios web desarrollados generalmente en código HTML (*HyperText Mark-up Language*) que contiene contenido de hipertexto e información. Existen diferentes tipos de navegadores aunque todos comparten una serie de características que les brinda la posibilidad de mostrar textos e imágenes.

El éxito de los navegadores para móviles radica en mostrar la información de forma rápida y optimizando el contenido debido a la pequeña pantalla de estos dispositivos. Otra de las características que poseen estos navegadores es la visualización de datos, como videos y fotos en resoluciones aptas para móviles, lo que ofrece una experiencia más enriquecida y práctica a la hora de usar el celular (6).

Algunos de los navegadores más conocidos que poseen los teléfonos celulares se describen a continuación:

- **Internet Explorer Mobile:** es un navegador desarrollado por Microsoft para las plataformas *Windows Mobile* y *Windows CE*. Ha sido optimizado para conexiones rápidas o lentas, por lo que si

² Wi-Fi, es la tecnología de comunicación inalámbrica más empleada en los dispositivos inalámbricos hoy en día.

la conexión es rápida puede entrar a páginas con mucho contenido gráfico, y en su última versión tiene un mejor renderizado de las páginas (7).

- **Firefox Mobile:** también conocido como *Fennec*, fue presentado por Mozilla como la versión para teléfonos móviles del navegador *Firefox*. Contiene un buscador automático y se puede mover por la Web utilizando las teclas de posición (8).
- **Opera Mobile:** desarrollado por Opera como la versión para *Windows Mobile Smartphone* del explorador web Opera para Windows. Permite navegar con facilidad en dispositivos con pantalla táctil o teclado, ajustar el texto para facilitar la lectura y además cambia automáticamente entre el modo vertical y horizontal (9; 10).
- **Safari Mobile:** desarrollado por Opera para las plataformas *iPhone OS* e *itouch*. La navegación se centra en la visualización de las páginas de forma rápida y no redimensiona. Además facilita la navegación pues permite abrir varias ventanas a la vez (11).

La navegación web desde teléfonos móviles se convertirá en algo cotidiano pues cada día se busca la forma de ampliar la funcionalidad de los mismos. Una aplicación web no debe desarrollarse para una versión específica, dada la existencia de una gran cantidad de navegadores y dispositivos en el mercado mundial. Por esta razón, llegar a un consenso de interpretación de los sistemas web es la búsqueda final de los navegadores, de forma que la codificación sea transparente para los usuarios finales.

El incremento del uso de los teléfonos celulares para la navegación en Internet ha traído consigo la necesidad de crear sistemas destinados especialmente para ellos, debido a esto los desarrolladores deben tener en cuenta un conjunto de pautas de diseño o buenas prácticas necesarias para el correcto desarrollo de las aplicaciones.

1.2.2. Aplicaciones web para móviles

Las aplicaciones web son un *software* mediante el cual los usuarios pueden acceder a un servidor web utilizando un navegador. Mientras que las aplicaciones web para móviles son aquellas que dan servicios a usuarios de teléfonos celulares con conexión inalámbrica, y han sido diseñadas para visualizarse e interactuar adecuadamente desde dispositivos móviles (12).

1.2.2.1. Sistemas similares

Para lograr que las aplicaciones web cumplan con las características necesarias para su correcta visualización en los dispositivos móviles es esencial que se realice un estudio sobre diferentes sistemas que estén destinados a brindar servicios a estos usuarios.

El análisis se va a centrar en el análisis del objeto de estudio definido en la investigación pues los sistemas móviles de gestión académica encontrados necesitaban de autenticación para poder navegar por los mismos, motivo por el cual no pudieron ser directamente el objeto del análisis.

Para la realización del estudio se lleva a cabo una comparación entre las versiones móviles y de escritorio de diferentes sistemas nacionales e internacionales, teniendo en cuenta las características que estos presentan.

- **Sistemas internacionales**

1. Stanford: sistema de la Universidad de Stanford que muestra toda la información relacionada con la vida y las actividades desarrolladas por los estudiantes de la institución.

La versión de escritorio muestra en la parte superior el nombre del sistema y un buscador. Debajo de estos se encuentra ubicado un menú con las diferentes opciones que ofrece la universidad. Además se muestran gran cantidad de imágenes, posee el área de trabajo dividida en varias columnas, en la parte inferior se encuentra el pie de página y encima de este un grupo de opciones que brinda el sistema. La dirección de acceso a la versión de escritorio del sistema es <http://www.stanford.edu>.

La versión móvil muestra en la parte superior el nombre del sistema, y debajo un menú ordenado de forma vertical con todas las opciones que brinda el mismo. El área de trabajo se encuentra dividida en una sola columna en forma de lista. Está desarrollada con el *framework* jQuery Mobile, y la dirección de acceso es <http://m.stanford.edu>.

2. United Airlines: sistema de la *United Airlines* tiene como objetivo brindar a los usuarios toda la información de la empresa. Además ofrece la posibilidad de realizar reservaciones personalizadas de vuelos, hoteles y autos.

La versión de escritorio muestra en la parte superior el nombre y el logo del sistema, debajo de este se encuentra un menú con varias de las opciones que brinda el mismo. En la parte inferior se encuentra el pie de página y otro grupo de opciones del sistema. El área de trabajo se encuentra dividida en varias columnas, y la dirección de acceso es <http://www.united.com>.

La versión móvil muestra en la parte superior el nombre y el logo del sistema, contiene un menú de forma vertical con varias de las opciones que se ofrecen. Además tiene el área de trabajo estructurada en una sola columna en forma de listas y fue desarrollado haciendo uso del *framework* jQuery Mobile. La dirección de acceso a la versión móvil del sistema es <http://mobile.united.com>.

3. Universidad de Chile: sistema perteneciente a esta universidad donde se muestra la información referente a diferentes áreas dentro de esta institución.

La versión de escritorio muestra en la parte superior el logo y el nombre del sistema, encima y debajo del área de identificación se encuentran ubicados 2 menús con varias opciones. El área de trabajo está estructurada en varias columnas, en la parte inferior se encuentra el pie de página y sobre este hay otro menú. El sistema contiene videos, multimedia, gran cantidad de imágenes y en la parte superior se encuentra ubicado el buscador. La dirección de acceso a la versión de escritorio del sistema es <http://www.uchile.cl/>.

La versión móvil está estructurada en una sola columna y en la parte superior se encuentra el logo y el nombre de la universidad. En la página principal muestra un conjunto de opciones relacionadas con diferentes áreas de la institución y en la parte inferior se encuentra ubicado el buscador. Además posee un menú de navegación en la parte superior y otro en la inferior, donde también se encuentra el pie de página. Siempre se muestra la información de forma breve, abordando solamente los aspectos de interés para los usuarios. La dirección de acceso a la versión móvil del sistema es: <http://m.uchile.cl>.

- **Sistemas locales**

1. Firefoxmanía: sistema cuya finalidad es mostrar los últimos artículos y comentarios publicados sobre Firefox, de forma que se mantenga a la comunidad Mozilla Firefox y a los usuarios del sistema actualizados.

La versión de escritorio muestra en la parte superior una imagen con el nombre y el logo del sistema, debajo aparece un menú con las diferentes secciones del mismo. El área de trabajo se encuentra dividida en varias columnas y en la parte inferior se encuentra ubicado un conjunto de vínculos a diferentes sistemas de la universidad y a otros sitios de interés. Se muestra un resumen e imágenes de la información publicada, dando la opción leer la noticia completa. La dirección de acceso a la versión de escritorio del sistema es <http://firefoxmania.uci.cu>.

La versión móvil tiene el área de trabajo estructurada en una sola columna en forma de listas. En la página principal se encuentra el nombre del sistema en la parte superior, en el centro el logo del mismo y en la parte inferior un menú horizontal formado por palabras. En el resto de las páginas se muestra un resumen de los últimos comentarios y artículos. Además el sistema cuenta con un buscador ubicado en la parte superior de todas las páginas, el cual le facilita la búsqueda a los usuarios. Se utilizó para su desarrollo el *framework* jQuery Mobile, y la dirección de acceso a la versión móvil del sistema es <http://firefoxmania.uci.cu/fxmobile/index.php/app>.

2. Zimbra: es un cliente servidor de correo que permite el acceso en tiempo real al correo electrónico, a la libreta de direcciones y al calendario, entre otras facilidades.

La versión de escritorio en la parte superior muestra el logo del sistema y un buscador junto con algunas opciones (el salir, persona autenticada). Debajo del logo se encuentra ubicado de forma horizontal las opciones de configuración del correo, después se encuentran todas las carpetas del sistema una debajo de otra. Al final de la página en la parte izquierda se encuentra el salir. La dirección de acceso a la versión de escritorio del sistema es <http://correo.estudiantes.uci.cu>.

El área de trabajo en la versión móvil está estructurada en una sola columna en forma de listas y se muestra un resumen del contenido del correo. En la parte superior de la página se encuentra ubicado un menú con íconos de las diferentes secciones del correo, debajo se encuentra otro menú formado por palabras que permite ver el resto de los correos. El sistema reconoce cuando la conexión que viene de un teléfono celular, aunque se puede especificar la versión al autenticarse y el salir del sistema está ubicado el final de la página. La dirección de acceso a la versión móvil del sistema es <http://correo.estudiantes.uci.cu>.

1.2.2.2. Conclusiones del estudio de sistemas similares

Una vez finalizado el estudio realizado a los diferentes sistemas se detectaron un conjunto de características que poseen todos ellos:

- Están estructurados en una sola columna en forma de listas, el ancho está dado por el del navegador, eliminando el desplazamiento horizontal, y el largo de acuerdo con la cantidad de información.
- No se muestra toda la información de los sitios tradicionales, solo está presente aquella que se considera relevante, eliminando la información innecesaria.

- Las imágenes son eliminadas o sustituidas por otras destinadas a los dispositivos móviles.
- Cuentan con un nombre en forma de texto que hace referencia al sistema, utilizándolo como identificador en la parte superior. Además algunos incluyen un logo diseñado especialmente para ellos.
- Los menús de navegación están ubicados en la parte superior e inferior, generalmente en forma de texto. En algunos casos estos menús están compuestos por íconos.
- La dirección de acceso en la mayoría de los casos está dada por un subdominio, aunque también se puede reconocer directamente de que dispositivo viene la conexión.

1.3. Herramientas y tecnologías

1.3.1. Proceso de desarrollo

“Un proceso de desarrollo de *software* es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de *software*” (13). También define quién está haciendo qué, cuándo y cómo alcanzar un determinado objetivo y proporciona normas para el desarrollo eficiente de un *software* de calidad (13).

- **Proceso de desarrollo con un enfoque ágil en los requisitos de CMMI nivel 2**

La Universidad de las Ciencias Informáticas no se encuentra ajena a los problemas que enfrenta la industria del *software*. La universidad actualmente está desplegando un proceso de mejora basado en el modelo CMMI, buscando incrementar la madurez y capacidad de la institución en la producción de *software*. El programa de mejora tiene como objetivo cumplir con las áreas de procesos del nivel 2 de CMMI, garantizando así la reducción de diversos problemas que presenta la producción de *software* en la institución (14).

El enfoque ágil hace referencia a un proceso adaptable que debe realizarse de forma incremental. La Alianza Ágil propone un conjunto de principios para quien quiera alcanzar un desarrollo ágil, a continuación se muestran algunos de ellos (15):

- Entregar con frecuencia *software* en funcionamiento, desde un par de semanas hasta un par de meses, con una preferencia por la escala de tiempo más corta.

- El *software* en funcionamiento es la medida primaria de progreso.
- La mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de *software* valioso.
- El método más eficiente y efectivo de transmitir información hacia y dentro del equipo de desarrollo es la conversación cara a cara.

La agilidad se puede aplicar en cualquier proceso de desarrollo de *software*, aunque para lograrlo es necesario que sea diseñado de forma tal que permita adaptar las tareas, coordinarlas y enfatizar en la entrega de *software* en funcionamiento al cliente tan rápido como sea posible (15).

El proceso de desarrollo va a utilizar las prácticas que proponen las metodologías de desarrollo ágil XP (*eXtreme Programming*) y Scrum. La metodología XP se centra en fomentar las relaciones interpersonales entre las personas que conforman el equipo de trabajo como clave para lograr el éxito en el desarrollo de *software*. Las prácticas que propone fomentan la comunicación entre los integrantes del equipo de trabajo, permitiendo la transferencia de conocimiento entre los desarrolladores y el cliente. Además brinda la posibilidad de mantener un mínimo nivel de documentación y un alto nivel de disciplina de los programadores (12).

Por su parte Scrum define un proceso empírico, iterativo e incremental de desarrollo y busca la forma de aumentar la realimentación sobre el desarrollo del producto, logrando de esta forma mitigar los riesgos tempranamente, además de identificar y corregir los problemas detectados. Esta metodología posee 2 características fundamentales, la primera es que las iteraciones realizadas durante el proceso de desarrollo del *software* tienen una duración de 30 días, obteniéndose como resultado un incremento ejecutable que es mostrado al cliente. La segunda característica es que las reuniones a lo largo del proyecto, especialmente las diarias, son las principales protagonistas en la verificación de la evolución del producto, la coordinación y la integración (16).

CMMI (*Capability Maturity Model Integration*) es un modelo de mejora de los procesos para el desarrollo de productos y de servicios. Este modelo tiene como propósito ayudar a las organizaciones a mejorar sus procesos de desarrollo y de mantenimiento, tanto para los productos como para los servicios. Además es un modelo descriptivo que detalla los atributos esenciales que deberían caracterizar a una organización en un nivel de maduración determinado (17). CMMI tiene 5 niveles de madurez para clasificar a las organizaciones, ellos son: Inicial, Gestionado, Definido, Gestionado Cuantitativamente y Optimizado, para poder alcanzar un nivel superior, se debe cumplir con todas las áreas del nivel inferior.

Para alcanzar el nivel 2 de CMMI es necesario realizar cambios en la forma de trabajar de las empresas, los proyectos son controlados y gestionados durante su desarrollo. La disciplina del proceso reflejada por el nivel de madurez 2 ayuda a asegurar que existen prácticas, y los proyectos son realizados y manejados de acuerdo con los planes documentados. En el nivel 2 el estado de los artefactos y la entrega de los servicios siguen planes definidos, además satisfacen sus descripciones especificadas, estándares y procedimientos.

Este nivel está compuesto por 7 áreas de procesos que se encuentran agrupadas en 3 categorías (18):

1. Gestión de proyectos
 - a. Planificación de proyectos (PP).
 - b. Gestión de la subcontratación (SAM).
 - c. Seguimiento y control de proyectos (PMC).
2. Ingeniería
 - a. Gestión de requisitos (REQM).
3. Soporte
 - a. Aseguramiento de la calidad (PPQA).
 - b. Gestión de configuración (CM).
 - c. Métricas y análisis (MA).

Luego de mencionadas las áreas del nivel 2 de CMMI se explicará la gestión de requisitos:

Gestión de requisitos (REQM): Esta área tiene como propósito la gestión de los requerimientos y de los componentes del producto del proyecto, además de identificar las inconsistencias entre los requerimientos, y los planes y productos de trabajo del proyecto.

El proceso gestiona todos los requisitos recibidos o generados por el proyecto, incluyendo tanto los requisitos técnicos como los no técnicos, y se gestionan las acciones necesarias para asegurar la conformidad con los requisitos por parte de las personas involucradas. También se gestionan los cambios a los requisitos durante su evolución a lo largo del proyecto y se identifica cualquier inconsistencia que existiese entre la planificación y los requisitos. Una parte de la gestión de requisitos es documentar los

cambios realizados a los requisitos y mantener la trazabilidad bidireccional entre la fuente de los requisitos y los productos (19).

1.3.2. Lenguaje de modelado

UML (*Unified Modeling Language*) es un lenguaje que se centra en la representación gráfica de un sistema, proporcionando reglas que permiten la comunicación. Los diseños realizados utilizando este lenguaje se pueden implementar en cualquier lenguaje que soporte las posibilidades de UML, principalmente los lenguajes orientados a objetos.

Además UML es un lenguaje para visualizar, especificar, construir y documentar un sistema de *software*. También ofrece un conjunto de diagramas que permiten visualizar desde varias perspectivas el sistema y sirve para el modelado completo de sistemas complejos con gran cantidad de *software* (20).

1.3.3. Herramienta CASE

Las herramientas CASE (*Computer Aided Software Engineering*) son una serie de programas y ayudas que proporcionan asistencia a los analistas, a los ingenieros de *software* y a los desarrolladores durante todo el proceso de desarrollo de *software*. También proporciona a los analistas más tiempo para el análisis y diseño, y los ayuda a minimizar el tiempo para codificar y probar el sistema (21).

- **Visual Paradigm 8.0**

Visual Paradigm es una herramienta CASE para el modelado que ha sido concebida para soportar el ciclo de vida del proceso de desarrollo de *software* a través de la representación de un conjunto de diagramas. También soporta los principales estándares de modelado, entre ellos: UML y BPMN (*Business Process Modelling Notation*) (22).

La versión 8.0 tiene una serie de nuevas características y mejoras, entre las que se encuentran: genera la base de datos con datos de ejemplo, permite dibujar el diagrama de datos, examinar los diagramas dependientes del proyecto y especificar la duración de una restricción (23).

1.3.4. Herramienta para la creación de prototipos de interfaz

Las herramientas para la creación de prototipos de interfaz de usuario brindan una visión preliminar del futuro sistema y estos deben realizarse en una etapa temprana del desarrollo. La elaboración de los

prototipos del sistema ayuda a estructurar el código de la aplicación, ya que las vistas y sus controladores son fácilmente numerables a partir de ellos.

- **ForeUI 3.006**

ForeUI es una herramienta para confeccionar los prototipos de interfaz de usuario de cualquier aplicación, ya sea, web o de escritorio. Además se puede crear cualquier estructura con fidelidad y se puede modificar su estilo cambiando simplemente el tema de la interfaz de usuario. También se puede diseñar el comportamiento de prototipos mediante la definición de flujo de gráficos intuitivos para controlar eventos específicos y permite exportar en forma de imágenes, documentos PDF o simulación HTML5 (24).

ForeUI ofrece una gran cantidad de componentes que incluyen: botones, menús, pestañas, barras de desplazamiento, *combo box* y otros elementos comunes, además de herramientas gráficas para insertar formas, líneas, textos, imágenes y mucho más. Todos estos factores hacen de ForeUI una herramienta productiva muy útil para el intercambio de ideas y las pruebas de usabilidad (25).

1.3.5. Lenguajes

Lenguaje del lado del servidor:

- **PHP 5.3.x**

PHP (*PHP: Hypertext Preprocessor*) es un lenguaje de código abierto adecuado para el desarrollo web y que puede ser incrustado en HTML. Además es un lenguaje interpretado del lado del servidor que brinda muchas funcionalidades a los desarrolladores web (26).

PHP 5.3.x incluye mejoras de rendimiento, más de 140 correcciones de errores y las interfaces pueden tener constructores. Además incluye una serie de características tales como: el soporte al espacio de nombre, al enlace estático en tiempo de ejecución y a la función *lambda* o anónima, entre otras (27) (28).

Lenguajes del lado del cliente:

- **HTML 5**

HTML (*HyperText Mark-up Language*) es un lenguaje interpretado por el navegador del lado del cliente y se escribe en forma de etiquetas. También puede incluir *script*, los cuales brindan instrucciones específicas a los navegadores que se encargan de procesar el lenguaje. Entre los *scripts* que pueden agregarse, los más conocidos y utilizados son *JavaScript* y PHP (29).

EL HTML se encarga de desarrollar una descripción sobre los contenidos que aparecen como textos y sobre su estructura, complementando dicho texto con diversos objetos. Además cada elemento de un documento HTML consta de una etiqueta de comienzo, un bloque de texto y una etiqueta de fin. El lenguaje HTML es extensible, pues se le pueden añadir características, etiquetas y funciones adicionales para el diseño de páginas web, generando un producto vistoso, rápido y sencillo (29).

- **CSS 3**

CSS (*Cascading Style Sheets*) son un conjunto de reglas en las que se define el formato que va a tener un determinado elemento de una página web, además describen cómo se va a mostrar un documento en pantalla a través de un dispositivo de lectura. La forma de descripción de estilos brinda a los desarrolladores web el control total sobre el estilo y el formato de los documentos (31).

Las hojas de estilo en cascada se pueden escribir internamente en el código HTML de la página web o en un archivo y enlazar la página al mismo. Dentro de las principales características de la programación con CSS se encuentra la de definir todos los estilos de una aplicación web.

- **JavaScript 1.2**

JavaScript es un lenguaje interpretado por lo que no es necesario compilar el sistema para ejecutarlo, por esta razón estos programas son ejecutados en el propio navegador del visitante y no en el servidor. Este lenguaje puede ser insertado dentro del propio documento HTML o puede ser llamado desde el HTML (32).

JavaScript brinda la posibilidad de comprobar los datos introducidos por los usuarios en un formulario, programar páginas dinámicas simples, además de capturar los eventos que fueron generados por el usuario, entre otras facilidades (32).

1.3.6. **Framework de desarrollo**

Un *framework* o marco de trabajo es una estructura de soporte definida que representa una arquitectura de *software* que modela las relaciones generales de las entidades del dominio (33).

- **CodeIgniter 1.7.3**

CodeIgniter es un *framework* PHP utilizado para la creación de aplicaciones web con PHP, siendo un producto de código libre. También define una arquitectura de desarrollo que hace que la programación sea

de forma más ordenada, y contiene diversas herramientas que ayudan a hacer aplicaciones más versátiles, seguras y en poco tiempo.

CodeIgniter tiene como objetivo permitir desarrollar proyectos mucho más rápidos de lo que podría ser si lo escribiesen desde cero, suministrando una interfaz simple y una estructura lógica para acceder a las librerías. Además permite enfocarse en su proyecto minimizando la cantidad de código necesario para una tarea dada. Entre las características de CodeIgniter se puede mencionar que es un sistema basado en el patrón Modelo-Vista-Controlador, posee paginación y encriptación de datos, que es extremadamente liviano y que tiene clases de base de datos llenas de características con soporte para varias plataformas (34).

- **GUUD 1.0**

GUUD (Sistema Integral de Gestión Universitaria, Universidad Digital y Gestión Documental) es un *framework* propuesto por el centro CENIA en la Universidad de las Ciencias Informáticas para el desarrollo de aplicaciones web con PHP, utilizando CodeIgniter 1.7.3 y jQuery 1.3.2, brindándole a los desarrolladores mayor facilidad a la hora de crear los proyectos. Cuenta con las características que ofrece la integración de CodeIgniter y jQuery, incluyendo un conjunto de novedades y adaptaciones, tales como:

- Del lado del servidor fue modificado el manejo de excepciones y mensajes. Además se implementó el IOC (*Inversion of Control*) para lograr la interacción entre los distintos módulos del sistema y fue añadida la característica de modularidad, permitiéndole al sistema dividirse por módulos (35).

Para el desarrollo de la versión móvil se utiliza el *framework* GUUD haciendo uso de las ventajas y novedades que ofrece CodeIgniter dentro del mismo, unido a este se emplea el *framework* jQuery Mobile para el desarrollo de las interfaces destinadas a los dispositivos móviles.

- **jQuery Mobile 1.2.0**

jQuery Mobile no es un *framework* creado desde cero sino que está basado en el *framework* JavaScript jQuery, debido a ello se puede entender como un *plugin* para jQuery. Este *framework* posee un conjunto de herramientas que simplifican el proceso de crear páginas para los dispositivos móviles y el código HTML básico es optimizado (36).

jQuery Mobile 1.2.0 trae una nueva ventana emergente y más de 100 mejoras y correcciones en toda la biblioteca para hacer las cosas mejor y más rápido. Con la incorporación de las ventanas emergentes es

posible crear un elemento flexible, que hace que sea fácil de colocar cualquier contenido en una superposición que flota por encima de la página actual (37; 38).

1.3.7. Entorno de desarrollo

IDE (Entorno de Desarrollo Integrado) es un ambiente de programación que brinda al programador un conjunto de facilidades, entre las que se encuentran: un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Además provee un entorno de programación amigable para varios lenguajes de programación.

- **NetBeans 7.2**

NetBeans es un entorno de desarrollo para que los programadores puedan escribir, compilar, depurar y ejecutar los programas, surgió producto de un proyecto de código abierto y está escrito en el lenguaje de programación Java. Este IDE es un producto libre y gratuito, sin restricciones de uso, y permite crear aplicaciones de varios tipos, tales como: web, de escritorio y móviles (39).

NetBeans soporta varios lenguajes de programación, entre los que se pueden mencionar PHP, Java y C++. También se puede instalar en todos los sistemas operativos que soportan Java, desde Windows a Linux (40).

NetBeans 7.2 ofrece un rendimiento significativamente mejorado y tiene una mayor capacidad de análisis de código estático en el Editor de Java. Además brinda una mejora en la interfaz de usuario, un consumo de memoria reducido, diversas mejoras del depurador y en la finalización del código (41).

1.3.8. Servidor web

Un servidor web es un programa que recibe peticiones realizadas por un cliente a través de la red, las cuales reciben una respuesta en forma de páginas, usualmente código HTML. Además utiliza el protocolo http para enviar páginas web al ordenador de un usuario cuando el usuario las solicita.

- **Apache HTTP Server 2.2.22**

Apache HTTP Server es un servidor web de código abierto que presenta bases de datos de autenticación. Cuando está activo, convierte la máquina en un servidor capaz de enviar contenido a cualquier navegador (42).

Apache HTTP Server tiene como objetivo fundamental proveer un servidor eficaz, seguro y extensible con servicios HTTP en sincronización con los estándares actuales de HTTP. Además se han rediseñado los módulos de paquetes de autenticación y autorización, y se ha simplificado la configuración de diseño por defecto (42).

1.3.9. Herramienta de control de versiones

El control de versiones hace referencia a la capacidad de recordar los cambios que se realizan en los contenidos de los ficheros y en la estructura de los directorios. También brinda la posibilidad de mantener cierto control sobre los cambios que se lleven a cabo cuando varias personas tienen que trabajar sobre los mismos documentos (43).

- **Subversion 1.6.17**

Subversion (SVN) es una herramienta de código abierto utilizada para el control de versiones de ficheros electrónicos, ya sea, *software* o documentación. Se basa en un repositorio central que tiene la capacidad de recordar todos los cambios realizados en los ficheros o estructuras. También permite crear ficheros, renombrarlos, y la creación de una rama de proyecto es tan sencilla como copiar un directorio (43).

1.3.10. WURFL 1.4.1

Hoy en día existe una gran diversidad de teléfonos celulares lo que ha traído consigo una gran polémica entre los desarrolladores de aplicaciones web destinadas a estos dispositivos, para solucionar este problema surge WURFL (*Wireless Universal Resource File*).

WURFL es un Repositorio de Descripción de Dispositivo (DDR), o sea, un componente de *software* que mapea los encabezados de solicitud HTTP al perfil del cliente HTTP (dispositivos móviles) que emitió la solicitud. Consiste en un archivo de configuración XML que contiene la definición de miles de dispositivos, de acuerdo con sus características y capacidades. WURFL está disponible bajo licencia de *software* libre y se actualiza constantemente incorporando las características de los últimos dispositivos salidos al mercado, para lograr que los desarrolladores construyan mejores aplicaciones para los usuarios (44).

1.3.11. Opera Mobile Emulator 12.1

Opera Mobile Emulator, es un emulador desarrollado para ser utilizado en los sistemas operativos Windows, Linux y Mac, y está disponible bajo licencia de *software* libre. Es de fácil instalación y permite a

los programadores estar seguros de lo que se está desarrollando. Este emulador contiene un selector de perfiles que permite generar varias instancias del navegador *Opera Mobile* para probar distintas configuraciones (45).

1.4. Conclusiones del capítulo

En este capítulo se realizó un estudio de los principales conceptos y temáticas relacionados con el diseño teórico planteado en la introducción para un mayor entendimiento del tema. Se analizaron las características de diferentes aplicaciones web desarrolladas en el mundo y en la universidad, permitiendo establecer algunas de las características a tener en cuenta en el desarrollo de la Versión móvil del Sistema de Gestión Académica de Pregrado. Además se efectuó un análisis de las herramientas, lenguajes y tecnologías utilizadas en el desarrollo de aplicaciones, quedando establecidas las que se utilizarán en el desarrollo de la propuesta solución teniendo en cuenta las características que presentan los dispositivos móviles.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN: ANÁLISIS Y DISEÑO

2.1. Introducción

En este capítulo se abordan las características generales del sistema a desarrollar y se realiza un levantamiento de los principales requisitos que debe cumplir. Además se define la propuesta de solución, teniendo en cuenta las pautas de diseño a poner en práctica en el diseño del sistema. También se incluyen el modelo de dominio y las especificaciones de los requisitos definidos.

2.2. Modelo de dominio

En la investigación no se realiza un modelado de proceso debido a que el módulo estudiante del Sistema de Gestión Académica de Pregrado no presenta un proceso bien definido, pues las actividades que realiza un estudiante no representan un proceso en sí. Por otra parte, en el módulo control docente se puede hablar de un proceso pero este en la versión móvil sería ínfimo, pues contaría con 2 o 3 actividades solamente. Por estas razones, y como la versión de escritorio del Sistema de Gestión Académica de Pregrado posee un modelo de proceso del módulo control docente, no se realiza en la versión móvil este modelo. Para lograr una mayor comprensión del negocio se elabora un modelo de dominio, donde se especifican las entidades que están presentes en el negocio y las relaciones entre ellas.

El modelo de dominio captura los objetos más importantes en el contexto del sistema, estos representan los elementos que existen o los eventos que tienen lugar en el entorno de trabajo del sistema. Los modelos de dominio son realizados mediante diagramas UML y le muestran a los clientes, revisores y desarrolladores las relaciones existentes entre las clases de dominio (13).

En la figura 1 se muestran las principales clases del sistema y la relación que existe entre ellas:

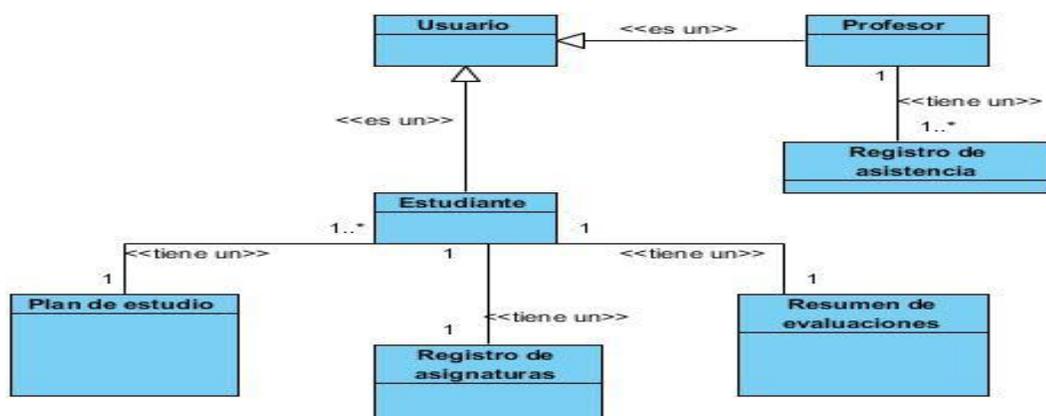


Figura 1: Modelo de dominio.

2.2.1. Definición de las clases del modelo de dominio

Las clases presentes en el modelo de dominio simbolizan los objetos del mundo real que se encuentran involucrados en el negocio del sistema, además muestra las relaciones existentes entre estas clases.

- **Usuario**

El usuario representa todas las personas que se autentican en el sistema y que pueden consultar la información.

- **Estudiante y profesor**

El estudiante y el profesor son usuarios del sistema. El estudiante puede acceder al resumen de evaluaciones para conocer la nota obtenida en las asignaturas cursadas, además puede revisar su plan de estudio y el registro de asignaturas. Por su parte el profesor puede registrar la asistencia a clase de los estudiantes de un grupo docente, en caso de tener asignado alguno.

- **Registro de asignaturas**

El registro de asignaturas contiene el control de las evaluaciones y de la asistencia de un estudiante a cada asignatura en el transcurso de la carrera. Además todo usuario que haya cursado estudio en la universidad puede acceder a este.

- **Plan de estudio**

El plan de estudio contiene todas las asignaturas cursadas en la carrera y puede ser revisado por todo usuario que haya cursado estudios en la universidad.

- **Resumen de evaluaciones**

El resumen de evaluaciones contiene todas las asignaturas con sus notas y el promedio obtenido hasta ese momento, a este puede acceder cualquier usuario que haya estudiado en la universidad.

- **Registro de asistencia**

El registro de asistencia es donde se controla la asistencia de los estudiantes de un grupo docente a un turno de clase determinado, a este solamente puede acceder el profesor que tiene asignado el grupo.

2.3. Requisitos funcionales

Los requisitos funcionales definen el comportamiento del *software*, o sea las funcionalidades que debe poseer el sistema. Luego de un minucioso estudio se definieron los siguientes requisitos:

RF1. Autenticar usuario.

RF2. Mostrar el resumen de evaluaciones.

RF3. Mostrar registro de asignaturas.

RF4. Mostrar plan de estudio.

RF5. Buscar persona.

RF6. Mostrar persona.

RF7. Mostrar el resumen de evaluaciones de un estudiante.

RF8. Consultar registro de asistencia.

RF9. Registrar asistencia.

2.4. Propuesta de diseño visual del sistema

2.4.1. Pautas del Sistema de Gestión Universitaria

La versión de escritorio del Sistema de Gestión Universitaria (SGU) posee una serie de características que deben tenerse en cuenta para la realización de los distintos módulos o aplicaciones que forman parte de este sistema, entre ellas se encuentran:

1. Establecimiento de una diagramación general y las dimensiones estándares para la elaboración de las distintas vistas del sistema.

- **Vista de presentación.**

1. Área de identificación.
2. Área de entrada de datos.



Figura 2: Vista de presentación del SGU.

- **Vista de escritorio.**

1. Área de subprocesos horizontales.
2. Área del nombre de la aplicación.
3. Área del buscador.
4. Área de nombre de usuario.
5. Área de líneas de procesos.
6. Área de servicios horizontales.
7. Área de pie de página.



Figura 3: Vista de escritorio del SGU.

- **Vista de gestión de procesos.**

1. Área de subprocesos horizontales.
2. Área del nombre de la aplicación.
3. Área del buscador.
4. Área de líneas de procesos.
7. Área de traza.
8. Área de línea de progreso.
9. Área de contexto.
10. Área de menú de módulos.

5. Área de módulos.

11. Área de pie de página.

6. Área de nombre de usuario.

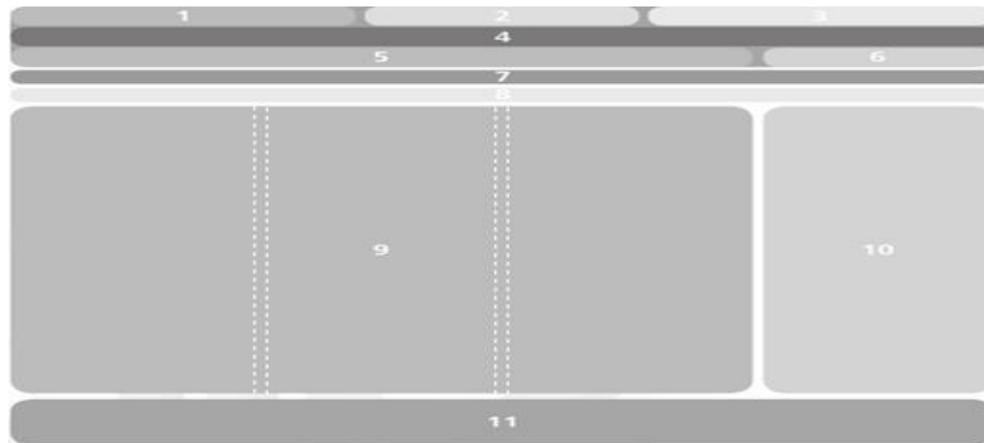


Figura 4: Vista de gestión de procesos del SGU.

2. Establecimiento de las pautas cromáticas a utilizar en el sistema, o sea, se establecen los colores a utilizar en las diferentes secciones y páginas.

- 1- #CCCCCC
- 2- #233865
- 3- #FDFDFD
- 4- #FFFFFF

- 5- #3C3C3C
- 6- #555555
- 7- #6F6F6F
- 8- #E8E8F2

3. Establecimiento de las pautas tipográficas, donde se establecen el tipo de letra y el tamaño. En el sistema se utiliza una sola tipografía en diferentes puntajes para establecer los diferentes niveles de lectura y jerarquías:

- Arial (14) – encabezados
- Arial (12)

2.4.2. Pautas de diseño para móviles

Para diseñar las aplicaciones web que son utilizadas en los teléfonos celulares es necesario considerar algunas características que estas deben cumplir.

Las pautas de diseño a considerar según Ana Milagro Luzardo son las siguientes (46):

1. No utilizar marcos, pues restan espacio de la pantalla.

2. Optimizar los gráficos, utilizando imágenes adecuadas con relación a dimensión y tamaño del dispositivo.
3. Incluir texto alternativo en todas las imágenes, con el objetivo de ofrecer información a los usuarios que han deshabilitado la carga de imágenes en su navegador.
4. La diagramación en una columna es la ideal para móviles, pues evita el desplazamiento frontal que dificulta la lectura.
5. El uso de una tipografía legible en pantalla, preferiblemente de la familia *Sans Serif*, no menor a 9 puntos y un máximo de doce, para destacar títulos y subtítulos.
6. Mantener la barra o menú de navegación presente en todas las pantallas y ubicarlo siempre en el mismo espacio con la misma disposición inicial, permite que el usuario comprenda esa área como zona de comunicación fija.
7. El menor uso posible del desplazamiento horizontal, ya que no favorece la precisión de lectura.
8. Publicar contenidos concisos, o sea, distinguir la información más importante y esencial.
9. Organizar la información cuidadosamente, minimizando la longitud del texto y distribuyendo de forma óptima los vínculos de navegación.

De igual forma la *World Wide Web Consortium (W3C)* define un conjunto de pautas a tener en cuenta en el desarrollo para móviles, dentro de las que se destacan (45):

1. Evitar las ventanas emergentes, y no cambiar de ventana sin informar al usuario.
2. No usar tablas anidadas ni marcos.
3. Concentrar la navegación en la parte superior de la página y reducirla al máximo.
4. Facilitar un equivalente en forma de texto para cada elemento no textual.
5. Asegurar que entre el color del fondo y el del primer plano hay suficiente contraste.
6. Cuando se utilicen imágenes de fondo, asegurarse que el contenido sigue siendo legible en el dispositivo correspondiente.
7. Limitar el desplazamiento a una sola dirección, a no ser que sea imposible evitar un desplazamiento secundario.
8. Siempre que sea posible, evitar la introducción de texto por parte de los usuarios.

9. Siempre que sea posible, establecer valores preseleccionados por defecto.

2.4.3. Propuesta de diseño visual.

Para desarrollar la versión móvil del Sistema de Gestión Académica de Pregrado es necesario tener en cuenta las características definidas para la aplicación de escritorio del Sistema de Gestión Universitaria, pues el sistema de Pregrado se encuentra dentro de este sistema. Además al ser una aplicación de uso en los dispositivos celulares hay que definir y poner en prácticas las pautas de diseño para móviles que se han estudiado.

La versión móvil del Sistema de Gestión Académica de Pregrado debe cumplir con las siguientes características:

1. Diagramación estructurada en una sola columna, evitando el desplazamiento horizontal.
2. Optimizar el uso de imágenes.
3. Utilizar una tipografía legible en pantalla, preferiblemente de la familia Arial, no menor a doce puntos y un máximo de catorce, para resaltar subtítulos.
4. Mantener el menú de navegación en todas las páginas.
5. Publicar solamente la información de importancia para el usuario y organizarla cuidadosamente.
6. Estructurar la interfaz visual de la siguiente forma:

- **Vista de presentación.**

1. Área de identificación.
2. Área de entrada de datos.

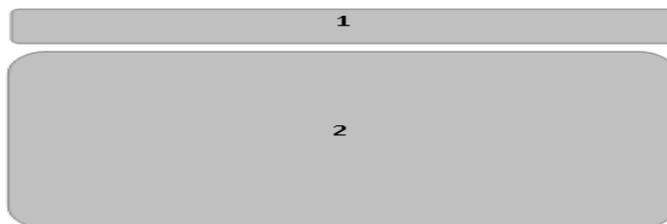


Figura 5: Vista de presentación de la versión móvil.

- **Vista de escritorio.**

1. Área de identificación.
2. Área de servicios.
4. Área del nombre de usuario.
5. Área de pie de página.

3. Área de líneas de procesos.

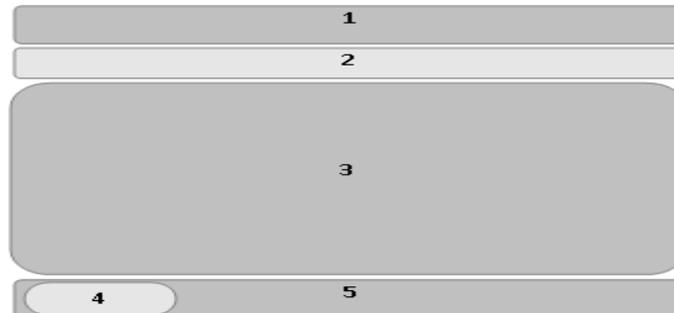


Figura 6: Vista de escritorio de la versión móvil.

- **Vista de gestión de procesos.**

- | | |
|----------------------------|-------------------------------|
| 1. Área de identificación. | 4. Área de nombre de usuario. |
| 2. Área de menú. | 5. Área de pie de página. |
| 3. Área de contexto. | |

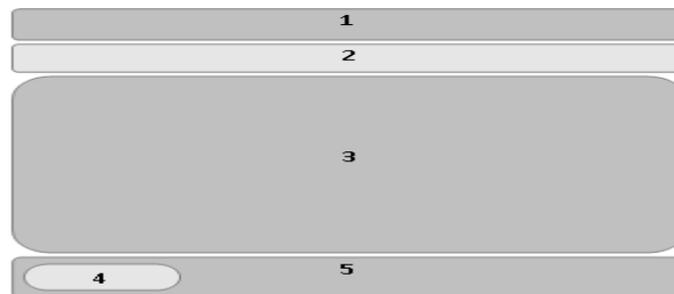


Figura 7: Vista de gestión de procesos de la versión móvil.

7. Las pautas cromáticas a utilizar son las siguientes:

- blanco (#FFFFFF)
- negro (#111111)
- azul (#1E396E)

2.5. Especificación de requisitos

A continuación se muestra la especificación de requisito: “Buscar persona”, a la cual los clientes le dieron prioridad “Alta” siendo de gran significado dentro de la aplicación. El resto de las especificaciones se encuentran en los anexos del 1 al 8.

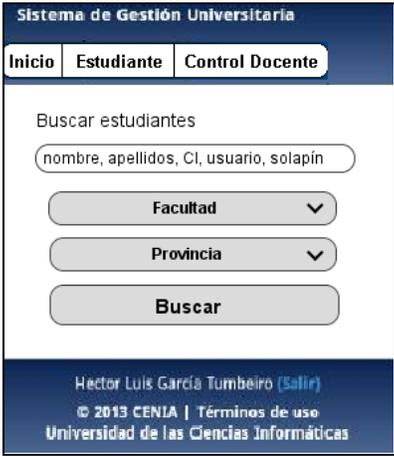
Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF5.	Buscar persona	El usuario selecciona la opción Buscar estudiante, luego introduce el criterio por el que desee buscar, que pueden ser: nombre, apellidos, CI, solapín y usuario. Además puede restringir la búsqueda especificando la facultad y la provincia a la que pertenece la persona.	Alta	Alta
Prototipo				
				
Campos		Tipos de datos	Reglas o restricciones	
• Nombre, apellido, CI, usuario, solapín		• Varchar	• Cualquier letra o número.	
• Facultad		• Varchar	• Seleccionar de la lista.	
• Provincia		• Varchar	• Seleccionar de la lista.	
Observaciones	<ul style="list-style-type: none"> • La búsqueda retorna todas las coincidencias con el valor entrado, siempre de 5 en 5. • La facultad y la provincia el usuario la selecciona de la lista, pues el sistema es el encargado de mostrarlas. 			

Tabla 1: Especificación de requisitos: Buscar persona.

2.6. Conclusiones del capítulo

En este capítulo se explicó la solución que proporciona la Versión móvil del Sistema de Gestión Académica de Pregrado y se analizaron algunas de las características de las aplicaciones web para móviles, permitiendo establecer las pautas a cumplir en el sistema a desarrollar. Además se listaron los requisitos funcionales, se presentaron y explicaron algunos artefactos, como son: el modelo de dominio y las especificaciones de requisitos, quedando especificadas las funcionalidades con que va a contar el sistema.

CAPÍTULO 3: PROPUESTA DE SOLUCIÓN: ARQUITECTURA

3.1. Introducción

En este capítulo se describe la propuesta arquitectónica del sistema a desarrollar, tomando como base la arquitectura existente en el Sistema de Gestión Universitaria. También se enuncian los requisitos no funcionales que deben tenerse presentes en la etapa de desarrollo, además se incluyen la arquitectura y el patrón arquitectónico a utilizar.

3.2. Requisitos no funcionales

Los requisitos o requerimientos no funcionales son propiedades o cualidades que un producto debe cumplir, las cuales posibilitan que sea atractivo, usable, rápido y con la calidad requerida. Además describen aspectos del sistema que son visibles por el usuario, que no incluyen una relación directa con el comportamiento funcional del sistema (47).

- **Requerimientos de *software***

Requerimientos para la máquina servidora.

- Disponer del sistema operativo GNU Linux.
- Tener instalado el Sistema Gestor de Base de Datos PostgreSQL 8.4.1.

Requerimientos para los dispositivos móviles.

- Contar con un navegador web o cliente http.

- **Requerimientos de *hardware***

Requerimientos para la máquina servidora.

- Memoria RAM de 2 GB.
- Disco duro de 80 GB.

- **Requerimientos de diseño e implementación**

- El lenguaje de programación a utilizar es PHP 5.3.x.
- El IDE de desarrollo a utilizar para el desarrollo del sistema es NetBeans 7.2.

- Los *framework* de desarrollo a utilizar son CodeIgniter 1.7.3 y jQuery Mobile 1.2.0.
- La herramienta CASE a utilizar es Visual Paradigm 8.0.
- **Requerimientos de apariencia o interfaz externa**
 - Todos los textos y mensajes se muestran en idioma español.
 - El sistema debe contar con un identificador en la parte superior, ya sea el nombre o el logo del mismo.
 - La aplicación debe visualizarse correctamente en la mayoría de los dispositivos móviles, independientemente de las características y propiedades que estos posean.
 - El sistema utiliza la menor cantidad posible de imágenes.
- **Requerimientos de usabilidad**
 - Todas las páginas deben poseer un menú de navegación.
 - Todos los usuarios de dispositivos móviles que tengan habilitado el servicio de navegación deben poder acceder a la información.
 - Los usuarios pueden ver la información sin la necesidad de utilizar desplazamiento o paginados excesivos.
- **Requerimientos de seguridad**
 - El sistema debe conceder acceso a partir de un nombre de usuario y una contraseña.
 - La seguridad del sistema está dada a nivel de roles, manteniendo la integridad de los datos en función del acceso de cada uno y posibilitando la protección de la información.
- **Requerimientos de interfaz de comunicación**
 - Los teléfonos móviles deben conectarse a la aplicación a través de la red Wi-Fi.

3.3. Arquitectura de *software*

Eugenia Bahit, define “la Arquitectura de *Software* es la forma en la que se organizan los componentes de un sistema, interactúan y se relacionan entre sí y con el contexto, aplicando normas y principios de diseño y calidad, que fortalezcan y fomenten la usabilidad a la vez que dejan preparado el sistema, para

su propia evolución”. Además afirma que la calidad del *software* está dada por el cumplimiento de un conjunto de atributos, dentro los que se destacan (48).

- Integrabilidad de los módulos independientes del sistema.
- Capacidad de interoperar con otros sistemas (interoperabilidad).
- Capacidad de permitir ser modificable a futuro (modificabilidad).
- Ser fácilmente mantenible (mantenibilidad).
- Tener una estructura que facilite la reusabilidad de la misma en futuros sistemas.
- Mantener un diseño arquitectónico escalable que permita su ampliación (escalabilidad).
- Facilidad de ser sometido a pruebas que aseguren que el sistema falla cuando es lo que se espera (testeabilidad).

3.4. Descripción de la arquitectura del Sistema de Gestión Universitaria

3.4.1. GUUD

Es un marco de trabajo utilizado en el desarrollo del Sistema de Gestión Universitaria, que tiene como objetivo permitir el desarrollo de aplicaciones web sin la necesidad de empezar desde cero.

GUUD define un ciclo muy bien estructurado por el cual fluyen los datos, el cual comienza en el controlador frontal *index.php*, en el que se inicializan todos los recursos que necesita el marco de trabajo para ejecutar, el ruteador examina las solicitudes HTTP con el objetivo de determinar qué hacer con estas. Antes de enviarlas a la ejecución normal del sistema verifica que no exista el archivo en la caché, en tal caso se lo envía directo al navegador. Para comenzar la ejecución las solicitudes HTTP son filtradas. Después se carga el controlador encargado de cargar las librerías, las bibliotecas del núcleo, *helpers*, y los recursos necesarios con el fin de procesar las vistas que obtienen información de los *scripts*. En caso de que la caché esté habilitada la vista se cachea para que puedan ser enviadas directas al navegador en próximas peticiones (ver figura 8).

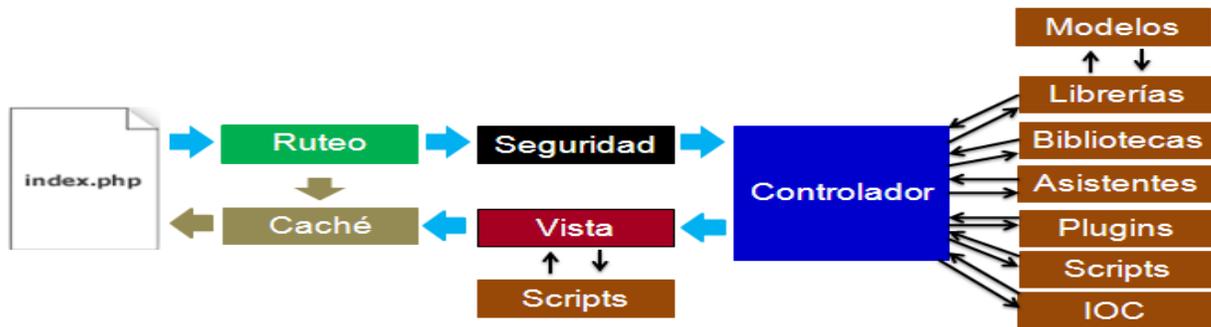


Figura 8: Flujo de información de GUUD.

3.4.2. Arquitectura Cliente/Servidor

La arquitectura utilizada es Cliente/Servidor, donde el cliente es el encargado de realizar las peticiones al servidor, y este a su vez debe darle respuesta a cada una de las peticiones (ver figura 9).

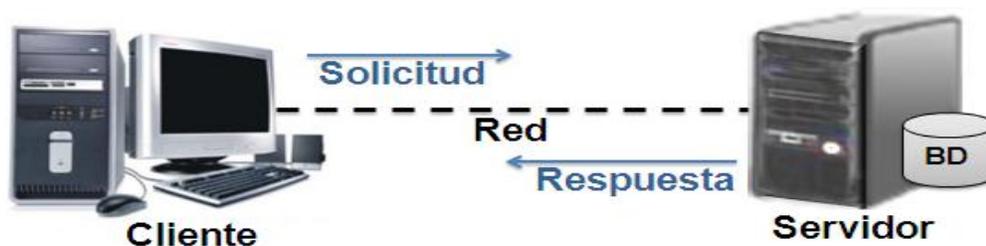


Figura 9: Arquitectura Cliente/Servidor (SGU).

Los clientes y los servidores pueden estar conectados a una red local o red amplia, como la que se puede implementar en una empresa o a una red mundial como Internet. Bajo este modelo cada usuario tiene la libertad de obtener la información que requiera en un momento dado proveniente de una o varias fuentes locales o distantes, y de procesarla según le convenga (49).

3.4.3. Patrón arquitectónico

El sistema está desarrollado utilizando el marco de trabajo CodeIgniter, y este implementa el patrón de arquitectura de *software* Modelo-Vista-Controlador (MVC), que separa la lógica de la aplicación y de la presentación. Esto facilita el trabajo distribuido y la integración de desarrollos independientes sin grandes dificultades, además facilita la reutilización de código y la flexibilidad (ver figura 10).

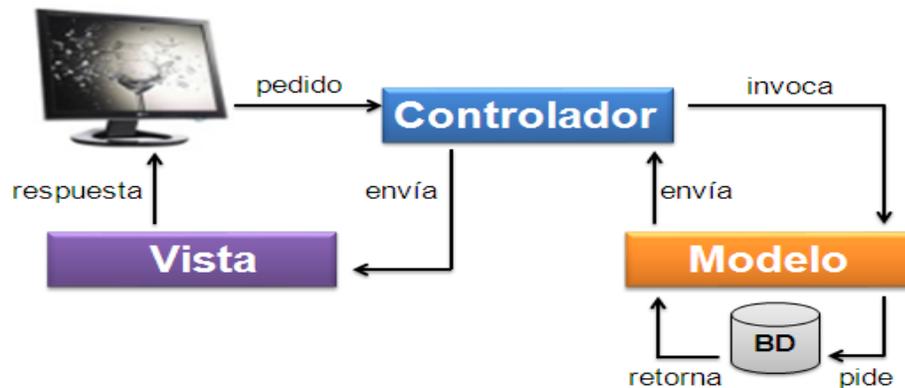


Figura 10: Funcionamiento del patrón Modelo-Vista-Controlador.

Este patrón separa el código en tres grupos:

- **Modelo:** se encuentra todo el código que tiene que ver con el acceso a bases de datos, de forma que actúa como intermediaria.
- **Vista:** es la encargada de codificar y mantener la presentación del sistema de cara al usuario final.
- **Controlador:** hace de enlace entre el modelo, la vista y cualquier otro recurso que se tenga que procesar en el servidor para generar la página web.

3.4.4. Estructura del Sistema de Gestión Universitaria

El directorio base (ver figura 11 a) está formado por otros directorios dentro de los que se destacan:

- **application:** en el cual se encuentra el directorio de todos los sistemas que conforman el Sistema de Gestión Universitaria, además incluye otros directorios (*views*, *config*, *libraries*) (ver figura 11 b).
- **public:** en el cual se incluyen las carpetas para los archivos css, js y las imágenes (ver figura 11 c).

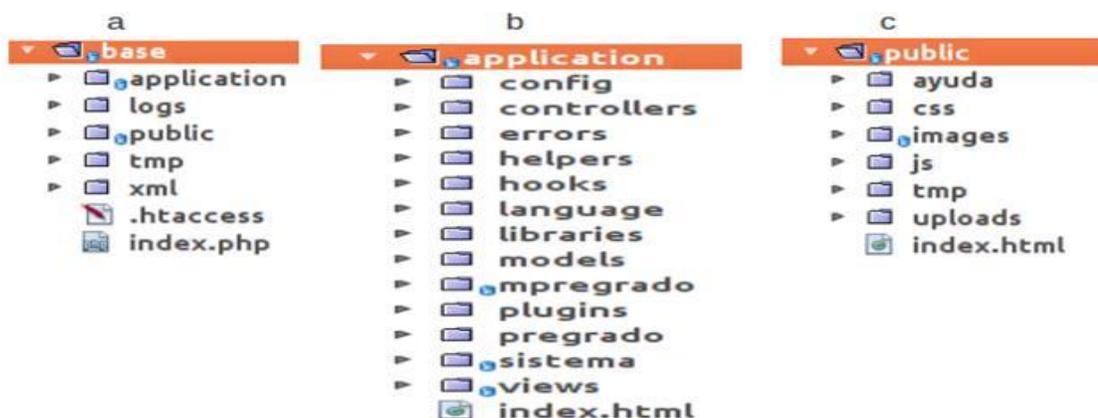


Figura 11: Directorio base, application y public.

Cada carpeta de los sistemas que conforman el SGU está compuesta por carpetas nombradas de acuerdo con el sistema o módulo que representan, respondiendo a la posibilidad de modularidad incorporada al CodeIgniter dentro de GUUD. A su vez, cada sistema contiene 5 carpetas con funciones específicas (ver figura 12).

- En la carpeta *config* se incluye la clase *box*, en la que se especifican las agrupaciones y funcionalidades que contendrá el sistema.
- La carpeta *controllers* contiene las clases controladoras.
- En la carpeta *libraries* se ubican las clases librerías que son las encargadas de la lógica del negocio, cumpliendo la función de intermediarias entre las controladoras y las modelos.
- En la carpeta *models* se concentran las clases encargadas de interactuar directamente con las bases de datos, es decir, los modelos.
- En la carpeta *views* se encuentran las interfaces que son las que interactúan de forma directa con el usuario.

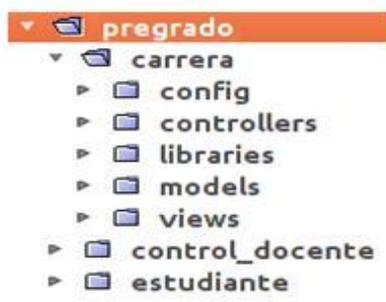


Figura 12: Sistema Pregrado, módulo carrera.

3.5. Descripción de la arquitectura de la propuesta de solución

Con el estudio realizado a la arquitectura actual del SGU se propone insertar la versión móvil del Sistema de Gestión Académica de Pregrado como un sistema del SGU, beneficiándose de las mejoras realizadas al *framework* CodeIgniter en el marco de trabajo GUUD. También aprovechan las características de modularidad, uso de plantillas y se propone sustituir jQuery (versión 1.3.2) por jQuery Mobile (versión 1.2.0).

3.5.1. jQuery Mobile

jQuery Mobile es un *framework* basado en jQuery con el objetivo de permitir desarrollar aplicaciones web compatibles con la mayoría de las plataformas móviles. Con la introducción de los dispositivos móviles se han visto multiplicado el número de navegadores y de plataformas con características muy diversas. jQuery Mobile brinda la posibilidad de no tener que lidiar con cada una de las particularidades de los navegadores. Además permite hacer las cosas más fáciles, de forma que se tenga que escribir menos código (37; 38).

Características de jQuery Mobile

- Está basado en el *framework* de *JavaScript* jQuery, por lo tanto, las personas que conocen de jQuery no tienen que aprender nada nuevo, solo aplicar conocimiento y desarrollar fácilmente aplicaciones para móviles.
- No se trata solo de la creación de código *JavaScript* que funcione en todos los navegadores, sino un conjunto de herramientas que vienen a simplificar el desarrollo de aplicaciones para móviles, desde la escritura del propio código HTML, la maquetación con CSS y la creación de estilos dinámicos con *JavaScript*.
- El *framework* capta si puede hacer conexión *Ajax*³ en lugar de una convencional.
- Permite adaptarse a los dispositivos táctiles.
- Se pueden elegir temas predefinidos y crear nuestros propios temas personalizados.
- El *framework* da soporte a diferentes plataformas compuesto en 3 niveles:
 - Nivel A da soporte a 26 plataformas probadas con varios teléfonos móviles cada una, haciendo uso de transiciones de páginas animadas basadas en *Ajax*.

³ *Asynchronous Javascript And XML*, técnica de desarrollo web para crear aplicaciones interactivas.

- Nivel B utiliza una apariencia mejorada, pero no hace uso de la navegación Ajax siendo compatible con 3 plataformas.
- Nivel C da soporte a 2 plataformas, las cuales solamente interpretan HTML básico.

La Versión móvil del Sistema de Gestión Académica de Pregrado utiliza la arquitectura Cliente/Servidor (ver figura 13).

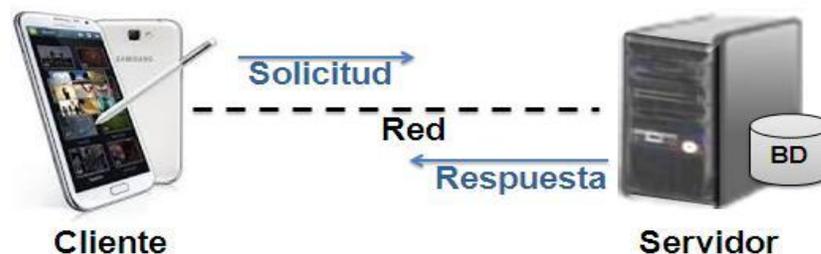


Figura 13: Arquitectura Cliente/Servidor (Pregrado móvil).

No todos los sistemas que utilizan el patrón de arquitectura Modelo-Vista-Controlador lo hacen de la misma forma, esto es posible debido a la flexibilidad que posee el mismo (ver figura 14).

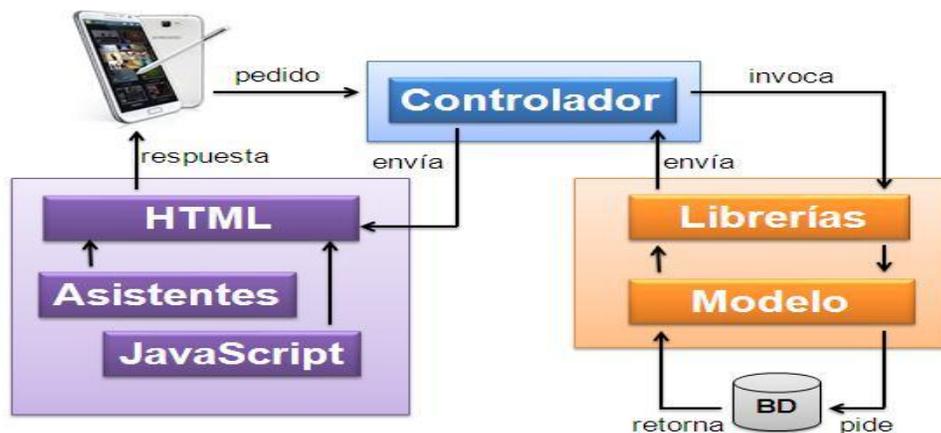


Figura 14: Implementación del patrón MVC.

Cada uno de los controladores de aplicaciones tiene asociada una o varias librerías que son las responsables de procesar los datos, además de implementar la lógica del negocio relacionada con las acciones del controlador. Al ser cargado el controlador de aplicaciones examina la petición para saber si tiene que cargar una vista determinada o si tiene que interactuar con la base de datos, teniendo en este caso que enviar los datos recibidos a las librerías. Estas cargan los modelos mediante los cuales pueden registrar, actualizar y obtener la información solicitada.

Los adaptadores de base de datos generan la sintaxis de la consulta, al obtener los datos se retornan al controlador realizando el proceso inverso al anteriormente descrito. Luego, el controlador carga los datos a archivos escritos en HTML incluyendo llamadas a archivos escritos en *JavaScript* o haciendo uso de asistentes. El resultado de todo este proceso se envía al navegador web como respuesta a la petición inicial.

La versión móvil al igual que la de escritorio, está compuesta por módulos y cada uno de estos por los archivos de configuración, las controladoras, las librerías, los modelos y las vistas (ver figura 15).



Figura 15: Vista del módulo "Estudiante móvil".

3.6. Integración de la solución con el Sistema de Gestión Universitaria

El SGU está integrado por un conjunto de sistemas y estos por varios módulos. Esta integración permite la reutilización de componentes, la eliminación de redundancias en el código y la centralización de archivos e información.

La versión móvil es uno de los sistemas con que cuenta el SGU, y al igual que los demás se relaciona con los componentes del núcleo del sistema. Además se relaciona con el Sistema de Gestión Académica de Pregrado y por tanto con los módulos que lo componen.

En el núcleo del sistema se encuentran un conjunto de componentes que pueden ser utilizados por todos los sistemas, la versión móvil utiliza los componentes que a continuación se describen:

- **Seguridad:** garantiza la seguridad de todo el sistema permitiendo la autenticación de los usuarios, la gestión de roles, usuarios, dominios, grupos de usuarios y los modos de acceso. Además facilita un conjunto de reglas de acceso a las funcionalidades del sistema de acuerdo con el nivel de autorización que posea el usuario.

- **Configuración:** encargado de la configuración del sistema, permite gestionar sistemas o módulos, entidades, activar o desactivar funcionalidades.
- **Trazas:** permite el control de incidencias dentro del sistema, para ello se definen varios tipos de trazas: URL, acción, consultas, interacción y excepción. A estas trazas también se le definen atributos comunes a registrar desde diferentes lugares del sistema entre los que se encuentran: fecha, hora, usuario, IP y URL.
- **Estructura y composición:** posibilita la creación de todas las estructuras de la universidad.

El Sistema de Gestión Académica de Pregrado está compuesto por varios módulos pero la solución propuesta utiliza los que se describen a continuación:

- **Estudiante:** ofrece información sobre los resultados docentes de los estudiantes, ya sean evaluaciones y promedio alcanzado en los distintos años de la carrera. Además muestra el plan de estudio a cumplir para poder terminar la carrera.
- **Control docente:** gestiona el registro de asistencia y evaluaciones de los estudiantes. Además es el encargado de crear los grupos docentes.
- **Personal y Secretaría:** gestiona el personal vinculado con los procesos de pregrado.

La comunicación entre los módulos y sistemas se realiza mediante el IOC (*Inversion of control*) que es un mecanismo que permite que las funcionalidades de un módulo puedan ser llamadas en otros.

3.7. Patrones de diseño

Los patrones de diseño son una descripción de objetos y clases comunicándose entre sí, adaptada para resolver un problema de diseño en determinado contexto. Además proponen una forma de reutilizar la experiencia de los desarrolladores, para ello clasifica y describe formas de solucionar problemas que ocurren de forma frecuente en el desarrollo (50).

3.7.1. Patrones GRASP

Los patrones GRASP (*General Responsibility Assignment Software Patterns*) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. El nombre se eligió para indicar la importancia de captar (*grasping*) estos principios, si se desea diseñar

eficazmente un *software* orientado a objetos. A continuación se explican los patrones GRASP utilizados en el desarrollo del sistema (51):

- **Experto:** se encarga de asignarle una responsabilidad al experto en información, que es la clase que cuenta con la información necesaria para cumplir con la responsabilidad. También brinda beneficios como el soporte a la alta cohesión y al bajo acoplamiento, además conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide (51).

Este patrón se pone de manifiesto en las clases librerías que son las que poseen la información necesaria para poder cumplir las responsabilidades sobre los elementos del negocio (52).

- **Creador:** guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. Este patrón brinda soporte al bajo acoplamiento y tiene como objetivo principal encontrar un creador, el cual debe conectarse con el objeto producido en cualquier evento (51).

El uso de este patrón se evidencia en la clase *loader* que es el objeto *load* de las clases controladoras el cual se encarga de cargar las librerías y los modelos. Además se utiliza en la librería *fabrica_ma_lib*, perteneciente al componente seguridad, que es la encargada de crear los objetos de los modos de autenticación (*ma*) (52).

- **Controlador:** asigna la responsabilidad de manejar un mensaje de un evento del sistema a una clase, la cual representa a la empresa o sistema global. Este patrón debe delegar a otros objetos el trabajo que ha de realizarse mientras se dedica a coordinar la actividad. También ofrece una guía para tomar decisiones apropiadas en la elección de los controladores de eventos (51).

La puesta en práctica de este patrón se pone de manifiesto en las clases controladoras, siendo estas las encargadas de obtener y enviar los datos a las librerías y a las vistas (52).

- **Alta cohesión y bajo acoplamiento:**

La alta cohesión asigna una responsabilidad, de modo que la cohesión siga siendo alta, además caracteriza a las clases con responsabilidades relacionadas que no realicen mucho trabajo. Este patrón mejora la calidad y facilidad del diseño, aunque también promueve la reutilización y genera un bajo acoplamiento (51).

Por su parte el bajo acoplamiento asigna una responsabilidad para mantener bajo acoplamiento. Una clase con bajo acoplamiento no depende de muchas otras, reduciendo el impacto de los cambios, además son más fáciles de reutilizar, aumentando de esta forma la productividad (51).

Estos patrones se evidencian en la implementación del propio marco de trabajo CodeIgniter, el cual permite el uso de forma individual de los componentes, poniendo de manifiesto el bajo acoplamiento, así como la dependencia existente entre ellos o alta cohesión (52).

3.7.2. Patrones GoF

Los patrones GoF (*Gang of Four*) describen las formas comunes en que distintos tipos de objetos pueden ser organizados para trabajar unos con otros. Además permiten crear grupos de objetos para ayudar a realizar tareas complejas, tratan la relación entre clases y la formación de estructuras de mayor complejidad. A continuación se explican los patrones GoF utilizados en el desarrollo del sistema (52):

- **Fábrica abstracta (*Abstract factory*):** brinda la posibilidad de trabajar con objetos de diferentes familias, de manera que ellas no se mezclen entre sí y hace transparente el tipo de familia concreta que se esté usando.

Este patrón se pone de manifiesto en la librería *fabrica_ma_lib* del componente seguridad, la cual se encarga de crear los objetos de los modos de autenticación (ma) que heredan de la clase *autenticacion_lib*, que son ma servicio web, ma base de datos, ma ldap y ma open ldap.

- **Instancia única (*Singleton*):** garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia.

La utilización de este patrón se puede ver en las clases controladoras pues todas ellas son instancias únicas, además se pone de evidencia en la clase IOC, la cual permite la interacción entre los módulos.

- **Mediador (*Mediator*):** permite definir un objeto que coordine la comunicación entre los objetos de diferentes clases, pero que funcionan como un conjunto.

Este patrón se pone manifiesto en las librerías, pues estas funcionan como mediadoras entre las clases controladoras y las de acceso a datos.

- **Observador (*Observer*):** permite definir una dependencia de uno-a-muchos entre objetos, brindando la posibilidad de que cuando un objeto cambie de estado se notifique y actualicen todos los objetos que dependen de él.

La puesta en práctica de este patrón se evidencia en la clase *loader* que es el objeto *load* de las clases controladoras, la cual se encarga de cargar las librerías y los modelos. Además de actualizar la clase controladora instanciada.

3.8. Conclusiones del capítulo

En este capítulo se realizó una descripción de la arquitectura del Sistema de Gestión Universitaria lo que permitió establecer cuál sería la arquitectura a utilizar en el sistema a desarrollar. Además se describen los patrones de diseño y de arquitectura que van a servir de base para la implementación del sistema propuesto.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

4.1. Introducción

En este capítulo se describen los estándares de codificación a utilizar en el desarrollo del sistema y las pruebas utilizadas para comprobar su correcto funcionamiento. Además se incluye el modelo de despliegue y se realiza una descripción de algunos elementos de implementación que presenta el sistema.

4.2. Modelo de despliegue

El modelo de despliegue muestra las relaciones físicas de los distintos nodos que componen un sistema, además se utiliza para modelar el *hardware* utilizado en las implementaciones de sistemas y las relaciones entre sus componentes (ver figura 16).

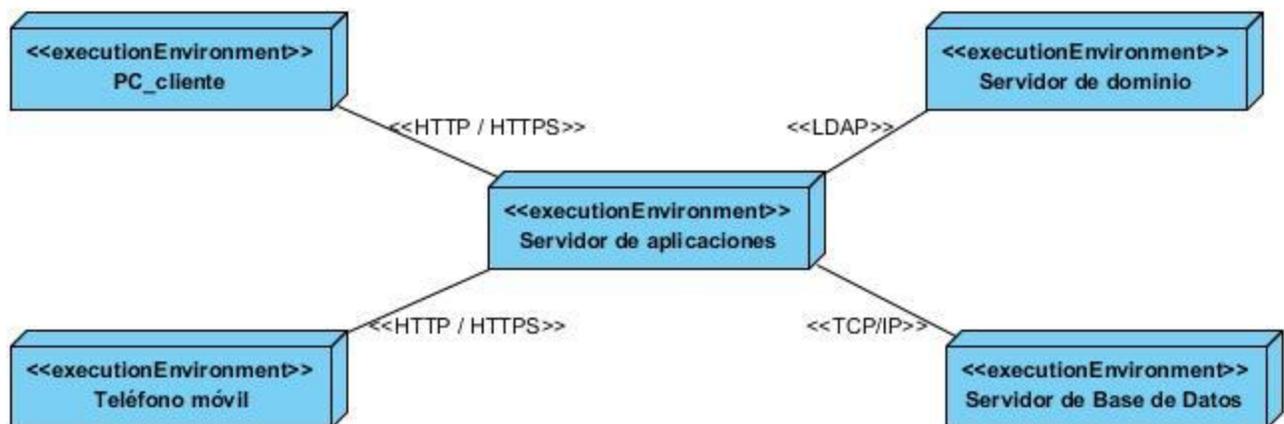


Figura 16: Modelo de despliegue.

4.3. Estándares de codificación

Los estándares de codificación persiguen lograr una homogeneidad entre todos los sistemas, de forma que se logre el desarrollo de códigos fáciles de entender. Estos estándares son reglas definidas que deben ser seguidas por los desarrolladores durante todo el proceso de desarrollo de *software*, permitiendo garantizar la organización y la estructura del código generado.

En el centro CENIA se establecen estándares de codificación con el propósito de desarrollar productos estandarizados en cuanto a código (52).

- **Indentación, llaves de apertura y cierre, y tamaño de las líneas.**

Usar una indentación sin tabulaciones, con un equivalente a 4 espacios, para mantener integridad en las revisiones svn. El uso de las llaves “{ }” será en una nueva línea. La longitud de las líneas de código es aproximadamente de 75-80 caracteres.

Ejemplo:

```
public function obtenerGruposDocentesDeProfesor()
{
    return $this->_ci->ioc->control_docente->obtenerGruposDocentesDeProfesor();
}
```

- **Convención de nomenclatura**

- **Variables:** se rigen por la nomenclatura *camelCase*. Siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula.

Ejemplo:

```
$nombreAnnos = $this->nombreAnnos($annos);
$html = "";
```

- **Clases:** siempre comienzan con mayúscula, en caso de nombre compuesto las palabras se separan con el carácter subrayado “_” y el resto en minúscula.

Ejemplo:

```
class Estudiantes extends MY_Controller
{
    //B
}

class Estudiantes_movil_lib
{
    //B
}
```

- **Funciones:** se rigen por la nomenclatura *camelCase*. Siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula. Los parámetros son separados por espacio luego de la coma que los separa.

Ejemplo:

```
public function __construct()
{
    $this->_ci = & get_instance();
}

public function obtenerGruposDocentesDeProfesor()
{
    return $this->_ci->ioc->control_docente->obtenerGruposDocentesDeProfesor();
}
```

- **Ficheros:** siempre todo en minúscula y en caso de nombres compuestos se usa el carácter subrayado “_”.
 1. **Vistas:** intuitivo y relacionado con el formulario y/o vista que representa.
 2. **Modelos:** con el mismo nombre de la clase que representa que contiene en el nombre el sufijo _mdl.
 3. **Librerías:** con el mismo nombre de la clase que representa que contiene en el nombre el sufijo _lib.
 4. **Controladoras:** con el mismo nombre de la clase que representa.

- **Estructuras de control**

Se incluye *if*, *for*, *foreach*, *while*, *switch*, entre las estructuras de control y los paréntesis debe de existir un espacio. Se recomienda utilizar siempre llaves de apertura y cierre, incluso en situaciones en las que técnicamente son opcionales. Esto aumenta la legibilidad y disminuye la probabilidad de errores lógicos.

Ejemplo:

```
foreach ($pasesEvaluacion as $value)
{
    return $value->nombre_anno_academico;
}

if ($value->id_anno_academico == $idAnnoAcademico)
{
    return $value->nombre_anno_academico;
}
```

```
else
{
    $_SESSION['mensaje'] = 'No se pudo conectar con el servidor.';
    $_SESSION['tipo'] = 'info';
    $this->listarConsultarRegistroAsistencia();
}
```

- **Documentación**

Todos los archivos deben de tener la documentación asociada al mismo. Para esto deben de cumplir con el siguiente bloque al principio de cada clase y función.

Ejemplo:

```
/**
 * Clase controladora registro
 *
 * Esta clase funciona como intermediaria entre la vista
 * y las librerías
 *
 * @package Control Docente
 * @subpackage Controllers
 * @category Controllers
 * @author Hector Luis Garcia Tumbeiro
 */

/**
 * Devuelve la tabla de las evaluaciones de una persona
 * @author Hector Luis Garcia Tumbeiro
 * Anisleidy Perez Suarez
 * @param string idPersona
 * @return html
 */
```

- **Buenas prácticas**

Los valores booleanos y nulos siempre se escriben con mayúscula, para facilitar la legibilidad del código, usar un *enter* antes de las estructuras de control y definición de las funciones.

Ejemplo:

```
$final = TRUE;
$final = FALSE;
```

4.4. Detalles de la implementación de la solución

Para lograr una correcta integración de la versión móvil con el SGU y su correcto funcionamiento, es necesario realizar un análisis de los ficheros del SGU con el objetivo de modificar y agregar ficheros.

Primeramente se estudia la ubicación de los ficheros css y js, decidiéndose ubicar los archivos css

necesarios en el directorio *public/css/movil* y los js en *public/js/jquery.mobile-1.2.0*.

Ficheros CSS:

- *jquery.mobile-1.2.0.min.css*
- *my-estilo.css*

Ficheros JS:

- *jquery.js*
- *jquery.mobile-1.2.0.min.js*
- *movil.js*
- *jquery.validate.js*
- *main.js*

Luego de esto se estudia la posibilidad de utilizar plantillas para lograr el diseño para móviles, por lo que se desarrollan diferentes plantillas:

- Vista de autenticar: */application/views/layouts/layout_empty_movil.php*
- Escritorio: */application/views/layouts/layout_desktop_movil.php*
- Portada de los módulos: */application/views/layouts/layout_portada_movil.php*
- Funcionalidades de cada módulo: */application/views/layouts/layout_proceso_movil.php*

Además de definir las partes de las plantillas:

- */application/views/partials/header_desktop_movil.php*
- */application/views/partials/header_proceso_movil.php*
- */application/views/partials/footer_movil.php*

Para que no se vean afectadas las plantillas de la versión de escritorio es necesario modificar funcionalidades en la clase controladora */application/libraries/MY_Controllers.php* en la cual se cargan las aplicaciones disponibles para la persona autenticada (ver anexo 9). En este controlador se desarrolla la función *acIMovil* con el objetivo de cargar las funcionalidades de cada una de las portadas de los módulos (ver anexo 10).

Para lograr la correcta integración de la versión móvil con el SGU, y su correcto funcionamiento es imprescindible agregar y modificar ficheros en el sistema.

Módulo seguridad

- Agregar la clase controladora *Autenticar_movil* que implementa las funcionalidades registrar y cerrar sesión.
- Agregar la vista *autenticar_movil*.

- Modificar la funcionalidad *index* de la clase controladora *Autenticar* (ver anexo 11).
- Modificar la clase controladora *acceso* (ver anexo 12).
- Agregar la vista *acceso_denegado_movil*.
- Modificar la funcionalidad *generarToken* en la clase librería *autenticar_lib* (ver anexo 13).
- Modificar la funcionalidad *obtenerElementoDadoToken* en la clase librería *autenticar_lib* (ver anexo 14).
- Modificar la funcionalidad *obtenerACLDadoToken* en la clase librería *autenticar_lib* (ver anexo 15).

Módulo común

- En la clase controladora *defaultindex* se modifica la función *index* con el objetivo de detectar el tipo de dispositivo que hace la petición para redireccionar de forma automática de ser móvil (ver anexo 16).
- En la clase controladora *portada* es necesario agregar la funcionalidad *portadaMovil* (ver anexo 17).
- Agregar la vista de escritorio para móvil *portada_movil*.
- Modificar la funcionalidad *index* de la clase controladora *portada* (ver anexo 18).
- Para diferenciar las aplicaciones de escritorio de las móviles, se agrega el atributo “*movil*” con valor “*t*” a las aplicaciones para móviles en la base de datos (ver anexo 19). Debido a esto hay que modificar la clase librería *portada_lib*, modificando la función *obtenerAplicaciones* para agregarle a cada aplicación si es móvil o no (ver anexo 20).

Módulo configuración

- En este módulo se gestiona la configuración del SGU, es aquí donde es necesario implementar las funcionalidades que permiten obtener de la base de datos las aplicaciones registradas como móviles. Para esto se agrega en la clase librería *sistema_lib* la función *obtenerNodosMoviles* (ver anexo 21).
- Para poder acceder a los datos en la clase modelo *tb_rnodo_atributo_valor_mdI* se implementa la función *obtenerNodosMovil* (ver anexo 22).

La comunicación entre los diferentes sistemas y módulos con que se integra la versión móvil se realiza mediante el fichero *ioc.xml*, el cual debe poseer un conjunto de funcionalidades para el correcto funcionamiento del sistema (ver anexo 23).

Debido a las características especiales que presentan los dispositivos móviles se decide que los

errores sean mostrados en la misma página. Para esto se implementan en la clase *MY_Controller* las funcionalidades *messageMovil* y *errorMovil* (ver anexo 24).

Estas funcionalidades son llamadas desde los controladores, especificando en caso de los mensajes el código y en el de los errores una excepción.

```
$this->messageMovil('SYS070');  
$this->listarGrupos();
```

No se encontraron resultados en la búsqueda.

```
$this->errorMovil(new Exception_Error('SYS027'));  
$this->listarGrupos();
```

Su usuario o su contraseña están incorrectos.

En el archivo *messages.xml* se agregan los siguientes mensajes:

```
<SYS0069 message="Debe estar logueado para realizar esta acción." type="L"/>  
<SYS0070 message="No se encontraron resultados en la búsqueda." type="L"/>
```

En el archivo *exceptions.xml* se agrega el siguiente error:

```
<SYS136><es message="La fecha no es válida." type="LP" /></SYS136>
```

4.5. Pruebas

Las pruebas son un grupo de actividades planeadas con anticipación y que se realizan de forma sistemática. Además brindan la posibilidad de evaluar la calidad del *software* y detectar errores en el mismo (15).

La ejecución de pruebas en el caso de la Versión móvil del Sistema de Gestión Académica de Pregrado se realiza para validar tanto el funcionamiento del sistema como el entendimiento que tengan los usuarios finales de sus funcionalidades. Algunos de los métodos aplicados son ejecutados solo a una pequeña muestra del universo real de usuarios finales de la aplicación, pues solamente se completa el desarrollo de la propuesta. No es parte del alcance de este trabajo la ejecución de pruebas generales de calidad

(ejecutadas por grupo de Auditorías y Revisiones del Centro de CALISOFT⁴) o el despliegue del sistema (que permitiría aplicar pruebas de usabilidad a una población mayor de usuarios).

4.5.1. Pruebas de funcionalidad

Las pruebas de funcionalidad examinan si una aplicación cumple con los requisitos funcionales propuestos por el cliente, también tienen como objetivo revelar los problemas y errores que poseen las funcionalidades. Además se debe verificar la validación de los datos y es importante realizarlas en cualquier fase de desarrollo del *software* (53).

Para llevar a cabo las pruebas de funcionalidades se utiliza el método de pruebas de caja negra.

4.5.1.1. Pruebas de caja negra

Las pruebas de caja negra consisten en la realización de pruebas con el objetivo de comprobar que cada función es operativa y que funcione adecuadamente. Además permite demostrar que la entrada es aceptada, que se produce una salida de forma correcta de acuerdo con las especificaciones del cliente y que la integridad de la información externa se mantiene. Al realizar estas pruebas se pretenden encontrar diferentes errores o problemas que puede presentar las funcionalidades desarrolladas, entre los que se pueden mencionar: funciones ausentes o incorrectas, errores de interfaz, de rendimiento, de acceso a la base de datos, entre otros (54).

Las pruebas de caja negra se llevan a cabo mediante la técnica de particiones equivalentes, las cuales dividen el dominio de entrada en un programa en clases de datos y de ellas se derivan los diferentes casos de prueba. Estas clases de equivalencias representan a un grupo de estados, ya sean válidos o inválidos, para las condiciones de entrada (54).

4.5.1.2. Diseño de casos de prueba

Los casos de prueba especifican una forma de comprobar el sistema, incluye un conjunto de datos de entrada, el resultado del sistema y las condiciones bajo las que ha de probarse (13). A continuación se

⁴ El Centro Nacional de Calidad de Software (CALISOFT) es una organización enfocada a contribuir al desarrollo de aplicaciones informáticas en la industria cubana. Es responsable de la evaluación de la conformidad con normas nacionales e internacionales y emite certificación a productos informáticos nacionales o importados. Controla y audita el uso de normativas técnicas, procedimientos, documentos estandarizados y buenas prácticas para el desarrollo de *software* en el país.

muestra el caso de prueba “Buscar persona”, los restantes se pueden encontrar en los anexos del 27 al 34.

Condiciones de ejecución						
El usuario debe estar autenticado en el sistema.						
SC Buscar persona						
Escenario	Descripción	Buscar estudiante	Facultad	Provincia	Respuesta del sistema	Flujo central
EC 1.1 Buscar datos correctamente.	Mediante este escenario se muestra al usuario el listado de personas existentes en el sistema con las características que busca.	V José Antonio	NA	NA	El sistema muestra un listado con todas las personas que tengan coincidencia con el elemento introducido.	-El usuario selecciona el sistema Pregrado. -Luego selecciona la funcionalidad "Buscar estudiante" y llena los campos de búsqueda: buscar estudiante, facultad, provincia. -Después presiona el botón "Buscar".
		NA	V Facultad 3	NA		
		V Pérez	NA	NA		
		V 88113021779	NA	NA		
		V jabo	NA	NA		
		V Yenni Martín Buchillón	V Facultad 1	NA		
		V yrapelop	NA	V Villa Clara		
		V rcrespo	V Facultad 5	V Artemisa		
		NA	NA	V Cienfuegos		
		V (espacio)	NA	NA		

EC 1.2 No existen elementos creados.	Mediante este escenario en caso de que no exista creado ningún elemento se muestra un mensaje.	V pkjgf	NA	NA	El sistema muestra el mensaje "No se encontraron resultados".	-El usuario selecciona el sistema Pregrado y luego la funcionalidad "Buscar estudiante" y presiona el botón "Buscar". -El sistema muestra el mensaje "No se encontraron resultados".
EC 1.3 Dejar vacío el campo Buscar estudiante.	Mediante este escenario en caso de que no se llene ningún campo se muestra un mensaje.	I (vacío)	NA	NA	El sistema muestra el mensaje "No se encontraron resultados".	-El usuario selecciona el sistema Pregrado y luego la funcionalidad "Buscar estudiante" y presiona el botón "Buscar". -El sistema muestra el mensaje "No se encontraron resultados".

Descripción de las variables.

No.	Nombre de campo	Clasificación	Valor nulo	Descripción
1	Buscar estudiante	Campo de texto	Sí	Permite la búsqueda de elementos mediante: nombre, apellidos, usuario, CI, solapín. O cualquier combinación de caracteres alfanuméricos.
2	Facultad	Lista desplegable	Sí	Permite seleccionar una facultad para realizar la búsqueda.
3	Provincia	Lista desplegable	Sí	Permite seleccionar una provincia para realizar la búsqueda.

Tabla 2: Caso de prueba: Buscar persona.

4.5.2. Pruebas de integración

Las pruebas de integración son una técnica sistemática para construir la arquitectura de un *software* al mismo tiempo que se aplican las pruebas para encontrar errores asociados a las interfaces (15).

Además comprueban que los componentes integrados realmente funcionan juntos, que sean llamados correctamente y que transfieran los datos de forma correcta y en el tiempo preciso a través de sus interfaces (55).

En el proceso definido para el desarrollo de aplicaciones en el SGU, se incluyen las pruebas de integración conjuntamente con los casos de prueba de funcionalidades descritos arriba, pues en los mismos es necesario validar el uso de datos, funciones u otros elementos provenientes de otros sistemas o componentes dentro del SGU. Ejemplo: la obtención de las provincias descritas como campos en la Tabla 2, se obtienen del Sistema de Gestión Académica de Pregrado, validándose esta integración con el mismo.

4.5.3. Pruebas con teléfonos celulares reales y emuladores

Las pruebas realizadas con teléfonos móviles reales comprueban que el sistema es accesible mediante conexión inalámbrica y que cumple con las buenas prácticas de diseño para móviles (descritas en el epígrafe 2.4.2). Estas pruebas permiten analizar el correcto funcionamiento de la versión móvil, comprobándose la visualización de las imágenes, el funcionamiento de los vínculos y de los menús.

Para la realización de las pruebas del sistema resulta imposible contar con todos los teléfonos móviles con acceso a red inalámbrica, con todos los sistemas operativos y las versiones de los mismos, tomando en cuenta también la diversidad de navegadores, el tamaño de las pantallas, resoluciones y otras características posibles. Teniendo en cuenta estas razones, para lograr la completitud de las pruebas con celulares reales se utilizan los emuladores, los cuales ofrecen la posibilidad de probar el sistema en otros entornos. Las pruebas son desarrolladas haciendo uso de 5 emuladores: *Opera Mobile Emulator*, *Android Emulator*, *iPhone Emulator*, *The Responsinator* y *iPad Peek* (ver anexo 25).

4.5.4. Pruebas con usuarios finales

Las pruebas efectuadas con herramientas informáticas si bien permiten probar las buenas prácticas de diseño implementadas en el sistema, no aseguran que la experiencia del usuario final sea exitosa. Debido a ello se realizan pruebas de usuario final, las cuales tienen como objetivo detectar otro tipo de problemas

entre los que se encuentran: las opciones no intuitivas, acciones poco claras y dificultades de manejo del sistema desde el teléfono móvil.

Estas pruebas son ejecutadas mediante la aplicación de técnicas o métodos específicos, entre las que se incluye el *test* de usuarios. Este método permite observar y analizar como utilizan el sistema un grupo de usuarios reales, los cuales deben realizar un conjunto de tareas orientadas por el evaluador. En la planificación del *test* de usuario deben incluirse: las tareas que deben realizar los usuarios, las preguntas asociadas a las tareas y la redacción del guión para los participantes, entre otras (ver anexo 26) (56).

Los usuarios son seleccionados en función del nivel de acceso que presentan dentro del sistema desarrollado, por ello se invitó a un grupo de 78 usuarios: 50 estudiantes y 28 profesores de la Universidad.

4.6. Resultados de las pruebas

Las pruebas funcionales y de integración fueron efectuadas en 3 revisiones parciales y una general, las cuales se realizaron siguiendo los diseños de casos de prueba elaborados por cada funcionalidad del sistema. En la primera iteración llevada a cabo se hallaron 3 no conformidades de las cuales se solucionaron todas, en la segunda se encontraron 2 no conformidades que fueron solucionadas completamente, en la tercera se detectaron 3 no conformidades corregidas totalmente y en la última iteración no se obtuvieron no conformidades.

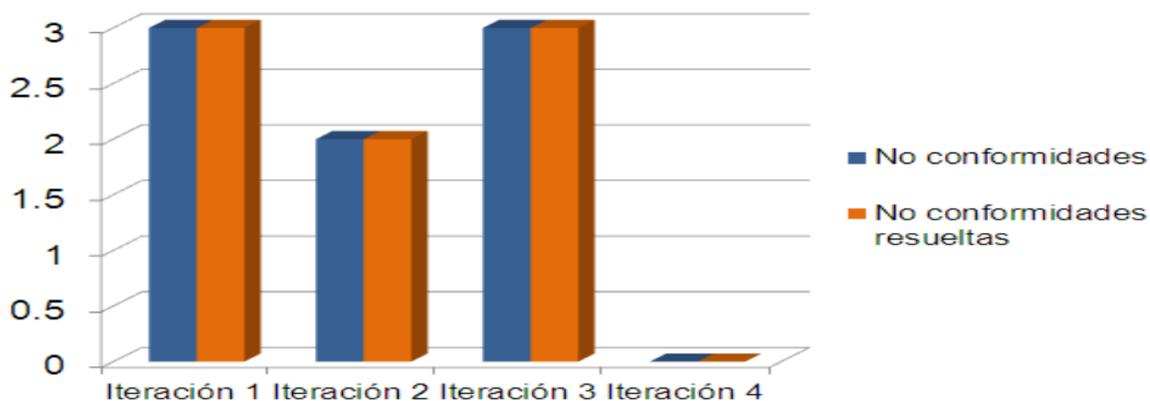


Figura 17: Resultados de las pruebas realizadas.

La aplicación de las pruebas realizadas con teléfonos celulares reales y emuladores de los mismos, validaron que la Versión móvil del Sistema de Gestión Académica de Pregrado reconoce el tipo

de navegador que está accediendo (navegador móvil o navegador de escritorio), mostrando la interfaz correspondiente en cada caso. Por otra parte, en todos los celulares en que se probó el sistema las imágenes se visualizaban de forma correcta.

Además, el sistema permite la navegación sin tener que utilizar desplazamiento horizontal de la página debido a la capacidad de adaptarse a las dimensiones del navegador que posee la versión móvil.

La aplicación del test de usuario, permitió validar que los vínculos, el menú de navegación y la tipografía que posee la versión móvil son entendibles, lográndose que los usuarios naveguen fácilmente dentro de esta. En general los datos de la aplicación del método se resumen en la tabla 3.

	Cantidad	%
Tareas completadas	390	100
Tareas incompletas	0	0
Sugerencias	2	NA
Total	390	

Tabla 3: Resultados del test de usuario.

La información obtenida de los usuarios después de haber realizado el *test* es que lograron cumplir sus objetivos sin presentar problemas de navegación por el sistema. Se mostraron conformes con la fuente y los avisos utilizados, siendo el diseño uno de los elementos más llamativos. Además se obtuvieron las siguientes sugerencias relacionadas con la tarea de registrar asistencia:

- Brindar la posibilidad de regresar del listado de los estudiantes a la de registro de datos generales de la asistencia.
- Diferenciar la opción por defecto de las listas de selección.

4.7. Conclusiones del capítulo

En este capítulo se realizó el artefacto modelo de despliegue, se analizaron y describieron los estándares de codificación utilizados en el desarrollo del sistema. Además se elaboraron los casos de prueba a los cuales fue sometido el sistema, quedando validado el cumplimiento de los requerimientos establecidos.

CONCLUSIONES

Una vez finalizada la investigación y el desarrollo de la propuesta, se obtuvieron resultados que posibilitan a los autores de la misma llegar a las siguientes conclusiones:

- El estudio de sistemas similares permitió establecer elementos comunes tanto en el diseño gráfico como en la ubicación de contenido que luego fueron utilizados en el desarrollo de la Versión móvil del Sistema de Gestión Académica de Pregrado.
- El análisis de las pautas de diseño del Sistema de Gestión Universitaria y las de aplicaciones web para móviles en general, permitieron definir características gráficas de la Versión móvil del Sistema de Gestión Académica de Pregrado.
- El análisis del entorno de desarrollo propuesto por el CENIA para aplicaciones de gestión de información basadas en tecnologías web y el estudio de sistemas similares, permitieron determinar la utilización del marco de trabajo jQuery Mobile para el desarrollo de la Versión móvil del Sistema de Gestión Académica de Pregrado.
- Con el uso de un proceso de desarrollo con enfoque ágil y las herramientas descritas, se implementó la Versión móvil del Sistema de Gestión Académica de Pregrado.
- Las pruebas efectuadas a lo largo y al finalizar la implementación de la Versión móvil del Sistema de Gestión Académica de Pregrado, validaron el cumplimiento de los requerimientos establecidos.

Por lo antes expuesto, se considera que la base teórica, la propuesta de diseño gráfico y las modificaciones para la implementación de la Versión móvil del Sistema de Gestión Académica de Pregrado, así como la aplicación funcional son los principales aportes de este trabajo de diploma. El sistema desarrollado permite acceder a funcionalidades de consulta y registrar asistencia en el Sistema de Gestión Académica de Pregrado mediante conexión inalámbrica desde teléfonos celulares.

RECOMENDACIONES

Al concluir el presente trabajo de diploma se recomienda:

- Actualizar el fichero WURFL para mantener actualizado al sistema sobre los últimos dispositivos móviles existentes en el mercado.
- Continuar desarrollando versiones móviles de los restantes sistemas y módulos del Sistema de Gestión Universitaria.

GLOSARIO DE TÉRMINOS

- **Aplicación o sistema:** en informática, son programas preparados para una utilización específica y que están diseñados para facilitar al usuario la realización de una determinada tarea o trabajo.
- **AJAX:** acrónimo de *Asynchronous JavaScript And XML* (en español, *JavaScript Asíncrono y XML*), es una técnica de desarrollo web para crear aplicaciones interactivas de forma rápida y dinámica. Permite que las páginas web se actualicen de forma asíncrona mediante el intercambio de pequeñas cantidades de datos con el servidor, lo que hace posible actualizar partes de una página web sin volver a cargar la página entera.
- **Emulador:** *software* que permite ejecutar programas de computadora en una plataforma (sistema operativo) diferente de la cual fueron escritos originalmente, y trata de modelar el dispositivo que se está emulando de forma precisa.
- **Hardware:** partes tangibles de un sistema informático, o sea, el conjunto de componentes físicos que integran la parte material de una computadora.
- **HTTP:** acrónimo de *HyperText Transfer Protocol* (en español, *Protocolo de transferencia de hipertexto*), es un método de intercambio de información en la *World Wide Web* mediante el cual se transfiere el contenido de las páginas web a una computadora.
- **Internet:** conjunto descentralizado de redes de comunicación interconectadas con alcance mundial, formada por la conexión directa entre computadoras mediante un protocolo especial de comunicación.
- **iPhone:** teléfono móvil inteligente creado por la compañía Apple Inc., posee pantalla táctil y permite la conexión a Internet por medio de una red inalámbrica.
- **Java:** lenguaje de programación de alto nivel orientado a objetos para el desarrollo de aplicaciones.
- **JavaScript:** lenguaje interpretado que se inserta en documentos HTML, de forma que su código queda reflejado en la propia página y no es llamado o cargado de ninguna fuente externa.
- **LDAP:** acrónimo de *Lightweight Directory Access Protocol* (en español, *Protocolo Ligero de Acceso a Directorios*), es un protocolo estándar que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.
- **Mac OS:** acrónimo de *Macintosh Operating System* (en español, *Sistema Operativo Macintosh*), es un sistema operativo desarrollado por la empresa Apple para su línea de computadoras *Macintosh*.

- **Navegador:** programa que permite visualizar la información que contiene una página web y presentarla de manera legible a los usuarios, permitiéndoles interactuar con el contenido y navegar hacia otros lugares de la red mediante enlaces o hipervínculos.
- **PDF:** acrónimo de *Portable Document Format* (en español, Formato de Documento Portátil), es un formato de almacenamiento de documentos digitales independientemente de plataformas de *software* o *hardware* y fue creado por la empresa *Adobe Systems*.
- **Plugins:** también conocido como *plug-in*, es una aplicación informática que añade funcionalidades específicas a un programa principal, extendiendo de esta forma las capacidades de ese programa.
- **Script:** conjunto de instrucciones que permiten la automatización de tareas y que puede recibir argumentos y devolver un valor. Se utilizan para generalizar código repetido, permitiendo ahorrar espacio y ganar velocidad.
- **Smartphone:** también llamado teléfono inteligente, es un dispositivo electrónico que funciona como un teléfono celular, el cual contiene mayor capacidad para almacenar datos y realizar actividades semejantes a una mini computadora, e igual conectividad que un teléfono móvil convencional.
- **Software:** conjunto de programas, instrucciones y reglas informáticas que brindan la posibilidad de ejecutar distintas tareas en una computadora, es considerado el equipamiento lógico e intangible de un ordenador.
- **Wi-Fi:** acrónimo de *Wireless Fidelity* (en español, Fidelidad Inalámbrica), es un conjunto de estándares para redes inalámbricas basados en las especificaciones IEEE 802.11. Fue creada para ser utilizada en redes locales inalámbricas, aunque actualmente se utiliza para acceder a Internet.
- **XML:** acrónimo de *eXtensible Markup Language* (en español, Lenguaje de Marcas Extensible), es un lenguaje de marcas creado por el *World Wide Web Consortium (W3C)*. Permite definir la gramática de lenguajes específicos, la organización y el etiquetado de documentos.

BIBLIOGRAFÍA CONSULTADA

BATISTA, A. L. S. y ROSARIO, O. M. *Personalización de Drupal para teléfonos móviles*. Trabajo de Diploma, Universidad de las Ciencias Informáticas, 2010.

BRITO, Á. *Diseñando para Móviles (diseño para múltiples tamaños de pantalla)* [Consultado el: 17 de enero de 2013]. Disponible en: <http://www.circulodemaquetadores.com/disenando-para-moviles-diseno-para-multiples-tamanos-de-pantalla/>.

DOMÍNGUEZ, Y. D. y SILVA, M. M. *Teleidentificador Personal. Portal WAP* Trabajo de Diploma, Universidad de las Ciencias Informáticas, 2008.

DOMÍNGUEZ, Y. G. y CRUZ, M. R. *Propuesta metodológica para el desarrollo de aplicaciones móviles utilizando el marco de trabajo PLASER*. Trabajo de Diploma, Universidad de las Ciencias Informáticas, 2008.

ELLISLAB, I. *Codelgniter de un Vistazo* [Consultado el: 4 de febrero de 2013]. Disponible en: http://escodeigniter.com/guia_usuario/overview/at_a_glance.html.

FERNÁNDEZ, A. G.; AMARGÓS, J. D. P., et al. *Estado del arte de las redes inalámbricas*. 3 ed. 2007, [Consultado el: 8 de enero de 2013]. Disponible en: <http://rii.cujae.edu.cu/index.php/revistaind/article/view/84>.

GARCÉS, E. A. y ECHEMENDIA, J. M. *Interfaz para acceso al servidor de correos Zimbra mediante dispositivos móviles*. Trabajo de Diploma, Universidad de las Ciencias Informáticas, 2010.

GARCÍA, E. D. L. V. *Proceso del desarrollo del software ágil, un enfoque hacia la Universidad de las Ciencias Informáticas*. Editado por: Informáticas, U. D. L. C. [Consultado el: 1 de febrero de 2013].

GOODMAN, D.; MORRISON, M., et al. *JavaScript Bible*. 5ta ed. Wiley Publishing, Inc., ISBN 0-7645-5743-2.

MARTÍNEZ, Y. H. *Guía para la Arquitectura de la Información de contenidos web en dispositivos móviles*. Trabajo de Diploma, Universidad de las Ciencias Informáticas, 2011.

MESTRAS, J. P. *Estructura de las Aplicaciones Orientadas a Objetos. El patrón Modelo-Vista-Controlador (MVC)*. Editado por: Madrid, U. C. 2008-2009, [Consultado el: 7 de marzo de 2013]. Disponible en: <http://www.fdi.ucm.es/profesor/jpavon/poo/2.14.mvc.pdf>.

PROGRAMACIÓN., D. D. L. Y. C. D. L. *Tutorial de JavaScript*. [Consultado el: 10 de febrero de 2013]. Disponible en: <http://www.lcc.uma.es/~eat/services/html-js/manual13.html>.

PROJECT MANAGEMENT INSTITUTE, I. *Guía de los Fundamentos de la Dirección de Proyectos*. 3ra ed. Project Management Institute, Inc., 2004, [Consultado el: 5 de febrero de 2013]. Disponible en: http://gio.uniovi.es/documentos/software/GUIA_PMBok.pdf. ISBN 1930699735.

RESIG, J. *Pro JavaScript Techniques*. Apress, 2006. ISBN 978-1-59059-727-9.

SOLDADO, R. M. *Herramientas utilizadas en el proceso de Diseño. Hasta dónde llega su alcance*. [Consultado el: 7 de febrero de 2013]. Disponible en: <http://lsi.ugr.es/rosana/investigacion/papers/rational.pdf> .

TEAM, C. P. *CMMISM for Systems Engineering/Software Engineering/Integrated Product and Process Development/Supplier Sourcing, Version 1.1*. Editado por: Institute, S. E. Pittsburgh: 2002, [Consultado el: 5 de febrero de 2013].

TRABAJO, A. H. y HERNÁNDEZ, P. G. *Portal de servicios con soporte para dispositivos móviles del Sistema Integral para la Atención Primaria de Salud*. Trabajo de Diploma, Universidad de las Ciencias Informáticas, 2011.

VENZANT, M. B. y MARTINI, J. L. R. *Análisis, diseño e implementación del Portal WAP del SIIPOL Móvil*. Trabajo de Diploma, Universidad de las Ciencias Informáticas, 2010.

W3C. *Buenas Prácticas Aplicaciones para la Web Móvil* [Consultado el: 15 de enero de 2013]. Disponible en: <http://www.w3.org/TR/mwabp/>.

W3C. *Buenas Prácticas en Web Móvil* [Consultado el: 14 de enero de 2013]. Disponible en: <http://www.w3.org/Mobile>.

BIBLIOGRAFÍA REFERENCIADA

1. NIELSEN, C. *Nielsen Social Media Report 2012* [Consultado el: 9 de diciembre de 2012]. Disponible en: <http://www.trecebits.com/2012/12/04/nielsen-social-media-report-2012-asi-hemos-usado-las-redes-sociales-este-ano/>.
2. GUARÍN, F. *Los teléfonos celulares*. 2005, [Consultado el: 7 de enero de 2013]. Disponible en: <http://www.funlam.edu.co/revistas/index.php/poesis/article/view/449>. ISBN 1692-0945.
3. INZAURRALDE, M.; ISI, J., *et al. Telefonía celular*. Montevideo, Uruguay: publicado el: 14 de diciembre del 2012 de última actualización: 14 de diciembre del 2012. Disponible en: <http://es.scribd.com/doc/52131154/telefonía-celular>.
4. JORDI MAYNÉ, I. D. A. *Estado actual de las comunicaciones inalámbricas*. 2005, [Consultado el: 16 de diciembre de 2012]. Disponible en: http://www.icee.upc.edu/JCEE2004/pdf/EstadoActual_WirelessRF_SILICA.pdf.
5. WORDREFERENCE.COM. *WordReference.com*. [Consultado el: 12 de febrero de 2013]. Disponible en: <http://www.wordreference.com/definicion/navegador>.
6. LUZARDO, I. *Los mejores navegadores para teléfonos móviles* [Consultado el: 11 de enero de 2013]. Disponible en: <http://www.enter.co/internet/los-mejores-navegadores-para-telefonos-moviles>.
7. CORPORATION, M. *Internet Explorer Mobile* [Consultado el: 8 de enero de 2013]. Disponible en: <http://www.microsoft.com/spain/windowsmobile/5/InstalledFeatures/iemobile.msp>.
8. LÓPEZ, J. M. *Prueba la futura versión de Firefox para móvil en tu PC* [Consultado el: 11 de enero de 2013]. Disponible en: <http://firefox-mobile.softonic.com/>.
9. ASA, O. S. *Opera Mini y Opera Mobile* [Consultado el: 8 de enero de 2013]. Disponible en: <http://www.opera.com/mobile/features/>.
10. BONET, J. *Opera Mobile* [Consultado el: 9 de enero de 2013]. Disponible en: <http://opera-mobile-sp-b3.softonic.com/windowsmobile>.
11. SAINZ, E. *Navegadores para dispositivos móviles* [Consultado el: 10 de enero de 2013]. Disponible en: <http://www.circulodemaquetadores.com/navegadores-para-dispositivos-moviles/>.
12. MORA, S. L. *Programación de aplicaciones web: historia, principios básicos y clientes web*. Universidad de Alicante: [Consultado el: 7 de enero de 2013]. Disponible en: http://rua.ua.es/dspace/bitstream/10045/16995/1/sergio_lujan-programacion_de_aplicaciones_web.pdf ISBN 978-84-8454-206-3.
13. JACOBSON, I.; BOOCH, G., *et al. El Proceso Unificado de Desarrollo de Software*. Editado por: Wesley, A. 2000. ISBN 84-7829-036-2.
14. AGÜERO, D. N.; BLANCO, K. R., *et al. Programa de Mejora en la Universidad de las Ciencias Informáticas*. Editado por: Informáticas, U. D. L. C. [Consultado el: 6 de febrero de 2013].
15. PRESSMAN, R. S. *Ingeniería de Software. Un enfoque práctico*. 6ta ed. 2005. ISBN 970-10-5473-3.

16. BOVENS, A.; MILLS, C., *et al.* *Opera Mobile Emulator for Desktop* [Consultado el: 28 de marzo de 2013]. Disponible en: <http://dev.opera.com/articles/view/opera-mobile-emulator/>.
17. TEAM, C. P. *CMMI® for Development, Version 1.2*. Editado por: Institute, S. E. Pittsburgh: 2006, [Consultado el: 5 de febrero de 2013].
18. GARCÍA, M. P. *Impacto de la aplicación del modelo CMMI Nivel 2 en el ciclo de vida de un proyecto*. Fin de carrera, Facultad de Informática. Universidad Politécnica de Madrid, 2009.
19. GRACIA, J. *CMM - CMMI Nivel 2* [Consultado el: 5 de febrero de 2013]. Disponible en: <http://www.ingenierosoftware.com/calidad/cmm-cmmi-nivel-2.php>.
20. ORALLO, E. H. *El Lenguaje Unificado de Modelado (UML)*. 2002, [Consultado el: 12 de enero de 2013]. Disponible en: <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>.
21. INFORMÁTICA, S.-J. D. *Herramientas CASE*. Editado por: Informática, I. N. D. E. E. 300 Ejemplares ed. 1999, [Consultado el: 31 de enero de 2013]. Disponible en: <http://www.inei.gob.pe/...pub/bancopub/Inf/Lib5103/Libro.pdf>
22. PARADIGM, V. *Visual Paradigm for UML - UML herramienta para el desarrollo de software de aplicación* [Consultado el: 17 de enero de 2013]. Disponible en: <http://www.visual-paradigm.com/product/vpum/>.
23. ---. *Visual Paradigm 8.0* [Consultado el: 27 de enero de 2013]. Disponible en: <http://www.visual-paradigm.com/aboutus/newsreleases/vpum180.jsp>.
24. FOREUI. *About ForeUI* [Consultado el: 8 de febrero de 2013]. Disponible en: <http://www.foreui.com/>.
25. INC, E. S. *ForeUI, Easy-To-Use UI Prototyping Tool* [Consultado el: 8 de febrero de 2013]. Disponible en: <http://foreui.com>.
26. ACHOUR, M.; BETZ, F., *et al.* *Manual de PHP*. 2012, [Consultado el: 26 de enero de 2013].
27. GROUP, T. P. *PHP 5.3.0 Release Announcement* [Consultado el: 7 de febrero de 2013]. Disponible en: http://php.net/releases/5_3_0.php.
28. GROUP, T. P. *Nuevas funcionalidades* [Consultado el: 7 de febrero de 2013]. Disponible en: <http://php.net/manual/es/migration53.new-features.php>.
29. ASTRUP, A.; JACOBSEN, J. C., *et al.* *¿Qué es HTML?* [Consultado el: 15 de enero de 2013 de 2013]. Disponible en: <http://es.html.net/tutorials/html/lesson2.php>.
30. *Definición de html*. [Consultado el: 29 de enero de 2013]. Disponible en: <http://definicion.de/html/>.
31. EDUCATIVAS, I. D. T. *HTLM5 en la educación. Módulo 9: Hojas de estilo*. [Consultado el: 11 de febrero de 2013]. Disponible en: http://www.ite.educacion.es/formacion/materiales/182/cd/pdf/HTML509_mat.pdf.
32. PÉREZ, J. E. *Introducción a JavaScript*. 2009, [Consultado el: 11 de marzo de 2013]. Disponible en: http://sunshine.prod.uci.edu/gridfs/sunshine/books/introduccion_javascript.pdf.
33. COSIO, H. C. y ROQUES, D. D. V. *Desarrollo del Portal WAP para la plataforma de gestión de contenidos Gina*. Trabajo de Diploma, Universidad de las Ciencias Informáticas, 2009.
34. INC, E. *Manual de CodeIgniter*. [Consultado el: 6 de febrero de 2013].

35. PRIETO, J. R. M. *Solución informática para la organización de la planificación docente en el Sistema de Gestión Académica de Pregrado*. Trabajo de diploma, Universidad de las Ciencias Informáticas, 2012.
36. ALVAREZ, M. A. *Introducción a jQuery Mobile* [Consultado el: 22 de enero de 2013]. Disponible en: <http://www.desarrolloweb.com/articulos/introduccion-jquery-mobile.html>.
37. FOUNDATION, T. J. *Overview* [Consultado el: 20 de enero de 2013]. Disponible en: <http://jquerymobile.com/test/docs/about/intro.html>.
38. PARKER, T. *Announcing jQuery Mobile 1.2.0 Final* [Consultado el: 4 de febrero de 2013]. Disponible en: <http://jquerymobile.com/blog/2012/10/02/announcing-jquery-mobile-1-2-0-final/>.
39. NETBEANS. *Bienvenido a NetBeans y www.netbeans.org* [Consultado el: 25 de enero de 2013]. Disponible en: http://netbeans.org/index_es.html.
40. ---. *NetBeans IDE Features* [Consultado el: 9 de enero de 2013]. Disponible en: <http://netbeans.org/features/index.html>.
41. ---. *NetBeans IDE 7.2.1 Información publicada* [Consultado el: 5 de febrero de 2013]. Disponible en: <http://netbeans.org/community/releases/72/>.
42. FOUNDATION, T. A. S. *The Number One HTTP Server On The Internet* [Consultado el: 29 de enero de 2013]. Disponible en: <http://httpd.apache.org/>.
43. SERRADILLA, J. L. *Control de Versiones con Subversion y TortoiseSVN* [Consultado el: 7 de febrero de 2013]. Disponible en: <http://www.um.es/atika/documentos/PREsubversion.pdf>
44. SCIENTIAMOBILE, I. *Welcome to WURFL* [Consultado el: 8 de febrero de 2013]. Disponible en: <http://wurfl.sourceforge.net/>.
45. W3C. *Mobile Web Best Practices* [Consultado el: 20 de enero de 2013]. Disponible en: http://www.w3.org/2007/02/mwbp_flip_cards.
46. ALLIEY, A. M. L. *Diseño de la interfaz gráfica Web en función de los dispositivos móviles*. Maestría de diseño, Universidad de Palermo, 2009.
47. QUIROGA, J. P. *Requerimientos Funcionales y No Funcionales*. Editado por: Andes, U. D. L. [Consultado el: 7 de marzo de 2013].
48. BAHIT, E. *Sunshine. El paradigma de la Programación Orientada a Objetos en PHP y el patrón de arquitectura de Software MVC*. 2011, [Consultado el: 5 de febrero de 2013]. Disponible en: <http://sunshine.prod.uci.cu/book/4eb204eb0571745ff5000007/>.
49. ECURED. *Arquitectura Cliente Servidor* [Consultado el: 7 de marzo de 2013]. Disponible en: http://www.ecured.cu/index.php/Arquitectura_Cliente_Servidor.
50. LARMAN, C. *UML y Patrones*. México: 2002, [Consultado el: 2 de marzo de 2013]. ISBN 0-13-748880-7.
51. VISCONTI, M. y ASTUDILLO, H. *Fundamentos de Ingeniería de Software*. Editado por: María, U. T. F. S. [Consultado el: 13 de marzo de 2013].
52. VIDAL, Y. G. *Arquitectura de Software metodología SXP*. Editado por: Informáticas, U. D. L. C. 2011, [Consultado el: 15 de marzo de 2013].

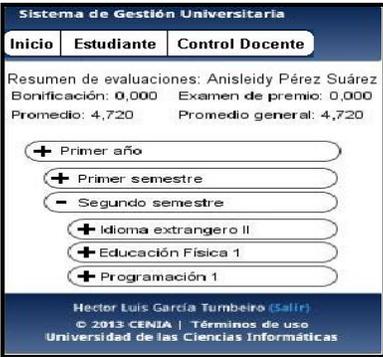
53. INFORMÁTICA, I. T. D. *Testeo de funcionalidad* [Consultado el: 21 de marzo de 2013]. Disponible en: <http://www.iti.es/servicios/servicio/resource/7234/index.html>.
54. GIL, M. T. *Técnicas de prueba*. Editado por: Almerña, U. D. España: [Consultado el: 21 de marzo de 2013]. Disponible en: <http://indalog.ual.es/mtorres/LP/Prueba.pdf>
55. SOMMERVILLE, I. *Ingeniería de Software*. 7ma. ed. 2005. ISBN 84-7829-074-5.
56. HASSAN, Y.; FERNANDEZ, F. J. M., *et al. Diseño web centrado en el usuario: usabilidad y arquitectura de la información* [Consultado el: 14 de mayo de 2013]. Disponible en: http://www.upf.edu/hipertextnet/numero-2/disenio_web.html#5.3.2.

ANEXOS

Anexo 1: Especificación de requisitos: Autenticar usuario.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF1.	Autenticar usuario.	El usuario introduce su usuario, contraseña, selecciona el dominio y la versión a la que desea acceder. Luego da clic en el botón <i>Enviar</i> .	Alta	Alta
Prototipo				
				
Campos		Tipos de datos	Reglas o restricciones	
• usuario		• Varchar	• Debe existir el usuario en el dominio.	
• contraseña		• Varchar	• Debe existir el usuario en el dominio y poseer una contraseña.	
• dominio		• Varchar	• Debe pertenecer a la sede central o a las regionales.	
• versión		• Varchar	• Escoger a cual desea entrar, a la versión de escritorio o a la móvil.	
Observaciones		<ul style="list-style-type: none"> • Si una persona no se puede autenticar en el sistema puede ser por: <ul style="list-style-type: none"> - No poseer un usuario en el dominio. - Introducir el usuario o la contraseña incorrectos. • El dominio a seleccionar es: uci.cu, cav.uci.cu, grm.uci.cu, hab.uci.cu. Además aparece por defecto la opción <i>uci.cu</i>. • La versión a seleccionar debe ser: Móvil o Escritorio, apareciendo por defecto <i>Móvil</i>. 		

Anexo 2: Especificación de requisitos: Mostrar el resumen de evaluaciones.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF2.	Mostrar el resumen de evaluaciones.	El usuario accede a la opción <i>Resumen de evaluaciones</i> y el sistema muestra el nombre de la persona con las bonificaciones, los exámenes de premio, el promedio y el promedio general obtenido. Además muestra todas las asignaturas cursadas por la persona y las notas obtenidas en las mismas, siendo ordenadas por año y período lectivo.	Alta	Alta
Prototipo				
				
Campos		Tipos de datos	Reglas o restricciones	
No procede		No procede	No procede	
Observaciones		<ul style="list-style-type: none"> • Interactúa con esta acción la persona que este o haya cursado estudios en el centro. • Las notas que se muestran son las notas finales obtenida por el estudiante en cada una de las asignaturas cursadas. • El promedio se obtiene a partir de la suma de las notas de todas las asignaturas entre la cantidad de evaluaciones. 		

Anexo 3: Especificación de requisitos: Mostrar registro de asignatura.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF3.	Mostrar registro de asignatura.	El usuario accede a la opción <i>Registro de asignaturas</i> y el sistema muestra el listado de las asignaturas cursadas por la persona, siendo ordenadas por año y período lectivo. Además muestra por cada asignatura el control de asistencia y el de evaluaciones efectuadas a las mismas.	Alta	Alta
Prototipo				
				
Campos	Tipos de datos		Reglas o restricciones	
No procede	No procede		No procede	
Observaciones	<ul style="list-style-type: none"> • Interactúa con esta acción la persona que este o haya cursado estudios en el centro. • Las notas que se muestran son las notas finales obtenida por el estudiante en cada una de las asignaturas cursadas. • El por ciento (%) de asistencia que se muestra, a una determinada asignatura, es general, o sea, de todo el curso. 			

Anexo 4: Especificación de requisitos: Mostrar plan de estudio.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF4.	Mostrar plan de estudio.	El usuario accede a la opción <i>Plan de estudio</i> y el sistema muestra el plan de estudio que está cumpliendo la persona y un listado de las asignaturas que se tienen que cursar, siendo estas ordenadas de acuerdo con el año académico y el período lectivo. Además de las asignaturas se muestra la cantidad de horas clase que tiene y si posee evaluación final.	Alta	Alta
Prototipo				
				
Campos		Tipos de datos	Reglas o restricciones	
No procede		No procede	No procede	
Observaciones		<ul style="list-style-type: none"> Interactúa con esta acción la persona que este o haya cursado estudios en el centro. 		

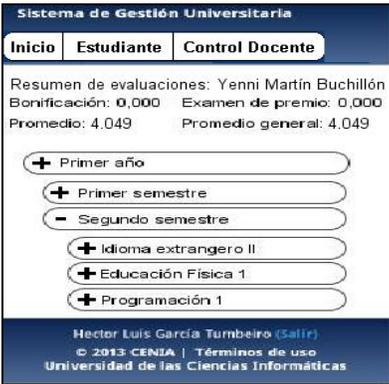
Anexo 5: Especificación de requisitos: Mostrar persona.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF6.	Mostrar persona	El sistema le muestra al usuario un listado de las personas existentes que coincidan con	Media	Alta

		el criterio de búsqueda. De las personas se muestran los siguientes datos: foto, nombre y apellidos, provincia, facultad y grupo. Si el resultado de la búsqueda es mayor de 5 aparecen los botones <i>Anterior</i> y <i>Siguiente</i> .		
Prototipo				
				
	Campos	Tipos de datos	Reglas o restricciones	
	No procede	No procede	No procede	
	Observaciones	<ul style="list-style-type: none"> • Los atributos se ordenan alfabéticamente en la lista por el nombre. • La búsqueda retornará todas las coincidencias con el valor entrado, siempre de 5 en 5. 		

Anexo 6: Especificación de requisitos: Mostrar el resumen de evaluaciones de persona.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF7.	Mostrar el resumen de evaluaciones de persona.	Luego del usuario haber ejecutado los RF5 y RF6, da clic sobre los datos de la persona que desea ver y el sistema muestra el resumen de evaluaciones de dicha persona. El resumen de evaluaciones contiene el nombre de la persona, además de las bonificaciones, los exámenes de	Media	Alta

		premio, el promedio y el promedio general obtenido. También muestra todas las asignaturas cursadas por la persona y las notas obtenidas en las mismas, siendo ordenadas por año y período lectivo.		
Prototipo				
				
	Campos	Tipos de datos	Reglas o restricciones	
	No procede	No procede	No procede	
	Observaciones	<ul style="list-style-type: none"> • Los detalles de las personas se mostrarán al dar clic sobre ella. • Las notas que se muestran son las notas finales obtenidas por el estudiante en cada una de las asignaturas cursadas. • El promedio se obtiene a partir de la suma de las notas de todas las asignaturas entre la cantidad de evaluaciones. 		

Anexo 7: Especificación de requisitos: Registrar asistencia.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF9.	Registrar asistencia.	El profesor selecciona la opción <i>Registrar asistencia</i> y el sistema muestra los grupos docentes que tiene asignado. Después el profesor selecciona un grupo y se muestran los datos generales para el registro (Fig. 1): introduce la fecha y selecciona la hora de	Alta	Alta

clase y la tipología, luego da clic en el botón *Aceptar*. El sistema muestra una interfaz con el listado de los estudiantes del grupo (Fig. 2), el profesor selecciona el estudiante y luego el estado de asistencia y el argumento.

Prototipo



Fig.:1 Datos generales de la asistencia.



Fig.:2 Pase de asistencia por estudiante.

Campos	Tipos de datos	Reglas o restricciones
• fecha	• Varchar	• Debe ser en el siguiente formato DD/MM/AAAA
• Hora clase	• Varchar	• Escoger la hora de clase. (Primera, Segunda, Ambas horas)
• Tipologías	• Varchar	• Escoger el tipo de clase que se impartió.
• Asistencia	• Varchar	• Escoger el estado de asistencia.
• Argumento	• Varchar	• Escoger el argumento en caso distinto del estado Presente.
Observaciones	<ul style="list-style-type: none"> • El profesor debe tener asignado un grupo. • El estado de asistencia del estudiante es Presente por defecto, el profesor debe cambiarlo en cualquier otro caso. • Tener en cuenta que en las tipologías se listan todos los posibles tipos de clase que puede dar el profesor, no están organizadas por forma organizativa. 	

Anexo 8: Especificación de requisito: Consultar registro de asistencia.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF8.	Consultar registro de asistencia.	El profesor accede al módulo <i>Control docente</i> y selecciona la opción <i>Consultar registro de asistencia</i> . El sistema muestra un listado de los grupos docentes que tiene asignado el profesor, el cual selecciona el que desee consultar. Luego se muestra el listado de los estudiantes del grupo seleccionado, donde se muestran los siguientes datos: foto, nombre y apellidos, % presente, % injustificado y % (A + J)/Plan.	Alta	Alta
Prototipo				
				
Campos		Tipos de datos	Reglas o restricciones	
No procede		No procede	No procede	
Observaciones	<ul style="list-style-type: none"> Interactúa con esta acción los profesores que tengan asignado un grupo docente. 			

Anexo 9: Aplicaciones disponibles para la persona autenticada.

```
private final function enabledApplicationVertical($security)
{
```

```
$token = $this->session->userdata('token');
if ($token != null || !empty($token))
{
    $selementos_token = $this->ioc->seguridad->obtenerElementosDadoToken($token);
    $aplicacionesR = $this->ioc->seguridad->obtenerAplicacionPermisoDadoUsuario($security, 'f',
$selementos_token['usuario']);
    $aplicaciones = $this->ioc->seguridad->ordenarActivos($aplicacionesR);
    $sistema = trim ($this->matchbox->fetch_directory(), '/');
$aplicacionesMoviles = $this->ioc->configuracion->obtenerNodosMoviles();
$movil = FALSE;
    foreach ($aplicaciones as $aplicacion)
    {
        if ($aplicacion->permiso == 't')
        {
            $sid_aplicacion = $aplicacion->id_aplicacion;
foreach ($aplicacionesMoviles as $moviles)
            {
                if ($sid_aplicacion == $moviles->id_nodo)
                {
                    $movil = TRUE;
                }
            }
$aplicacion->movil = $movil;
            $modulos = $this->ioc->seguridad->obtenerModuloDadoldAplicacionUsuario($sid_aplicacion,
$security, 'f', $selementos_token['usuario']);
            if (strtolower($aplicacion->nombre_aplicacion) == $sistema)
            {
                $this->template->set_data('__modulesV', $modulos);
                $this->template->set_data('__module', trim($this->matchbox->fetch_module(), '/'));
            }
            $aplicacion->modulos = $this->ioc->seguridad->ordenarActivos($modulos);
```

```

    }
    else
    {
        $this->template->set_data('__modulesV', "");
    }
}
$this->template->set_data('__applicationsV', $aplicaciones);
}
}

```

```

private final function disabledApplicationVertical($security)
{
    $aplicaciones = $this->ioc->seguridad->obtenerAplicacionPermisoDadoUsuario($security, 'f', "");
    $sistema = trim($this->matchbox->fetch_directory(), '/');
    $aplicacionesMoviles = $this->ioc->configuracion->obtenerNodosMoviles();
    $movil = FALSE;
    foreach ($aplicaciones as $aplicacion)
    {
        $id_aplicacion = $aplicacion->id_aplicacion;
        foreach ($aplicacionesMoviles as $moviles)
        {
            if ($id_aplicacion == $moviles->id_nodo)
            {
                $movil = TRUE;
            }
        }
        $aplicacion->movil = $movil;
        $modulos = $this->ioc->seguridad->obtenerModuloDadoldAplicacionUsuario($id_aplicacion,
        $security, 'f', null);
        if (strtolower($aplicacion->nombre_aplicacion) == $sistema)
        {

```

```
$this->template->set_data('__modulesV', $modulos);
$this->template->set_data('__module', trim($this->matchbox->fetch_module(), '/'));
}
$aplicacion->modulos = $modulos;
}
$this->template->set_data('__applicationsV', $aplicaciones);
}
```

Anexo 10: Función *acIMovil*.

```
public final function acIMovil()
{
    $this->applicationVerticalSecurity();
    $aplicacion= trim($this->matchbox->fetch_directory(), '/');
    $modulo= trim($this->matchbox->fetch_module(), '/');
    $this->template->set_data('aplicacion',$aplicacion);
    $this->template->set_data('modulo',$modulo);
    if ($this->xml->weaver->security->enabled[0] == 'true')
    {
        $token = $this->session->userdata('token');
        $module = $this->matchbox->fetch_module();
        $module_name = substr($module, 0, strlen($module) - 1);
        if (!empty($token))
        {
            $this->template->set_data('menu',$this->enabledAcl($token, $module_name));
        }
    }
}
```

```
        else
        {
            $this->template->set_data('menu',array());
        }
    }
    else
    {
        $this->template->set_data('menu', $this->disabledAcl());
    }
}
```

Anexo 11: Función index de la clase controladora Autenticar.

```
public function index()
{
    $dir = dirname(__FILE__);
    $array = explode('application', $dir);
    include_once $array[0] . 'public/wurfl-php-1.4.1/examples/demo/inc/wurfl_config_standard.php';
    $requestingDevice = $wurflManager->getDeviceForHttpRequest($_SERVER);
    if ($requestingDevice->getCapability('is_wireless_device') == 'true')
    {
        $_SESSION['tipoConexion'] = 'movil';
        redirect(base_url() . 'seguridad/autenticar_movil');
    }
    else
```

```
{
    $_SESSION['tipoConexion'] = 'escritorio';
    $this->template->set_structure('empty');
    $dominios = convert_assoc_array($this->usuario_lib->obtenerDominios(), 'id_dominio',
'nombre_dominio');
    $this->template->set_data('id_aplicacion', 1);
    $this->template->set_data('dominios', $dominios);
    $dominio_local = $this->xml->configuraciones->seguridad->dominio_local;
    $this->template->set_data('dominio_local', $dominio_local);
    echo $this->template->render('autenticar/autenticar_view');
}
}
```

Anexo 12: Función *index* de la clase controladora *acceso*.

```
public function index()
{
    $token = $this->session->userdata('token');
    $datosUser = $this->ioc->seguridad->obtenerElementosDadoToken($token);
    if($datosUser['tipoConexion']=='escritorio')
    {
        echo $this->template->render('accesodenegado');
    }
    else
    {
        $this->template->set_structure('desktop_movil');
        echo $this->template->render('acceso_denegado_movil');
    }
}
```

```
}
```

Anexo 13: Función *generarToken*.

```
final private function generarToken($usuario, $id_dominio, $id_aplicacion, $ip = NULL, $poner_hora = TRUE)
{
    $_CI = & get_instance();
    $_CI->load->model('tb_dusuario_md');
    $arreglo_ids = array("nombre_usuario" => $usuario);
    $persona = $_CI->ioc->seguridad->obtenerPersonaDadoUsuarioAutenticado($usuario, $id_dominio);
    if (!empty($persona))
    {
        $arr_relacion = $_CI->ioc->seguridad->obtenerIdRelacion($persona->id_usuario);
        $roles = $_CI->ioc->seguridad->obtenerRolesDadoldUsuario($persona->id_usuario);
        $arr_id_relacion = array();
        if (is_array($arr_relacion))
        {
            foreach ($arr_relacion as $obj)
            {
                array_push($arr_id_relacion, $obj->id_rol_usuario);
            }
        }
        else
        {
            array_push($arr_id_relacion, $arr_relacion->id_rol_usuario);
        }
        $arr_relacion_grupo = $_CI->ioc->seguridad->obtenerIdRelacionGrupo($persona->id_usuario);
        $arr_id_relacion_grupo = array();
        if (is_array($arr_relacion_grupo))
        {
            foreach ($arr_relacion_grupo as $obj)
```

```
{
    array_push($arr_id_relacion_grupo, $obj->id_usuario_grupo_usuario);
}
}
else
{
    array_push($arr_id_relacion_grupo, $arr_relacion_grupo->id_usuario_grupo_usuario);
}
$fecha = date("d-m-Y");
$hora = date("h:m:s");
$cadena = "{$usuario}";
$cadena .= '/' . $persona->id_usuario;
$cadena .= '/' . $persona->primer_nombre;
$cadena .= '/' . $persona->segundo_nombre;
$cadena .= '/' . $persona->primer_apellido;
$cadena .= '/' . $persona->segundo_apellido;
$cadena .= '/' . $persona->id_persona;
$cadena .= '/' . $persona->id_expediente;
$cadena .= '/' . self::definir_filtro_estructura($arr_id_relacion, $arr_id_relacion_grupo);
$cadena .= '/' . json_encode($roles);
$cadena .= "/{$id_aplicacion}";
$cadena .= "/{$id_dominio}";
if (is_null($ip))
{
    $cadena .= "/";
    $poner_hora = FALSE;
}
else
{
    $cadena .= "/{$ip}";
}
}
```

```
$cadena .= ( $poner_hora ? "{/$fecha}/{$hora}" : '//';  
$cadena.=' ' . $_SESSION['tipoConexion'];  
}  
else  
{  
    throw new Exception_Error('SYS001');  
}  
return base64_encode($cadena);  
}
```

Anexo 14: Función *obtenerElementoDadoToken*.

```
final public function obtenerElementosDadoToken($token)  
{  
    $result = array();  
    $token = base64_decode($token);  
    $selementos = explode('/', $token);  
    if ($selementos[0] != "")  
    {  
        $result['usuario'] = $selementos[0];  
        $result['id_usuario'] = $selementos[1];  
        $result['primer_nombre'] = $selementos[2];  
        $result['segundo_nombre'] = $selementos[3];  
        $result['primer_apellido'] = $selementos[4];  
        $result['segundo_apellido'] = $selementos[5];  
        $result['id_persona'] = $selementos[6];  
        $result['id_expediente'] = $selementos[7];  
        $result['filtro_estructura'] = $selementos[8];  
        $result['roles'] = json_decode($selementos[9]);  
        $result['id_aplicacion'] = $selementos[10];  
        $result['id_dominio'] = $selementos[11];  
        $result['ip'] = $selementos[12];  
    }  
}
```

```

$result['fecha'] = $elementos[13];
$result['hora'] = $elementos[14];
$result['tipoConexion'] = $elementos[15];
}
return $result;
}

```

Anexo 15: Función *obtenerACLDadoToken*.

```

final public static function obtenerACLDadoToken($token)
{
    $urls = array();
    $elementos_autenticacion = self::obtenerElementosDadoToken($token);
    $_CI = & get_instance();
    $_CI->load->model('tb_daplicacion_md');
    $funcionalidadesAll=$_CI->tb_daplicacion_md->
acLDadoUsuarioAplicacion($elementos_autenticacion['usuario'],
$elementos_autenticacion['id_aplicacion']);

    $arr_id_estructuras = self::obtenerIdEstructurasActivas();
    $tipoConexion = $elementos_autenticacion['tipoConexion'];
    $modulosMoviles = array();
    $aplicacionesMoviles = $_CI->ioc->configuracion->obtenerNodosMoviles();
    foreach ($aplicacionesMoviles as $valor)
    {
        $modulos = $_CI->ioc->seguridad->obtenerModuloDadoldAplicacionUsuario($valor->id_nodo,
true, 'f', $elementos_autenticacion['usuario']);
        foreach ($modulos as $mod)
        {
            if (!isset($modulosMoviles[$mod->id_modulo]))
            {
                $modulosMoviles[$mod->nombre_modulo] = $mod->id_modulo;
            }
        }
    }
}

```

```
    }
  }
}
foreach ($funcionalidadesAll as $func)
{
  if (in_array($func->id_estructura, $arr_id_estructuras))
  {
    array_push($urls, $func);
  }
}
if (!empty($urls))
{
  $funcionalidades = array();
  foreach ($urls as $url)
  {
    if ($tipoConexion == 'movil' && isset($modulosMoviles[$url->nombre_modulo]))
    {
      if (!isset($funcionalidades[$url->nombre_modulo]))
      {
        $funcionalidad = new stdClass();
        $funcionalidad->urls = array();
        $funcionalidades[$url->nombre_modulo] = $funcionalidad;
      }
      $funcionalidades[$url->nombre_modulo]->urls[$url->url] = $url->url;
    }
    if ($tipoConexion == 'escritorio' && !isset($modulosMoviles[$url->nombre_modulo]))
    {
      if (!isset($funcionalidades[$url->nombre_modulo]))
      {
        $funcionalidad = new stdClass();
        $funcionalidad->urls = array();
```

```
        $funcionalidades[$url->nombre_modulo] = $funcionalidad;
    }
    $funcionalidades[$url->nombre_modulo]->urls[$url->url] = $url->url;
}
}
return $funcionalidades;
}
return FALSE;
}
```

Anexo 16: Función *index* de la clase controladora *defaultindex*

```
function index()
{
    $this->template->set_structure('empty');
    $dir = dirname(__FILE__);
    $array=explode('application', $dir);
    include_once $array[0].'public/wurfl-php-1.4.1/examples/demo/inc/wurfl_config_standard.php';
    $requestingDevice = $wurflManager->getDeviceForHttpRequest($_SERVER);
    if($requestingDevice->getCapability('is_wireless_device')==true)
    {
        $_SESSION['tipoConexion'] = 'movil';
        redirect(base_url().'seguridad/autenticar_movil');
    }
    else
    {
        $_SESSION['tipoConexion'] = 'escritorio';
        $this->template->render('inicio_1');
    }
}
```

Anexo 17: Función *portadaMovil* de la clase controladora *portada*.

```
function portadaMovil()
{
    $this->template->set_structure('desktop_movil');
    if ($this->xml->weaver->security->enabled[0] == 'true')
    {
        $token = $this->session->userdata('token');
        $datosUser = $this->ioc->seguridad->obtenerElementosDadoToken($token);
        if (isset($datosUser['tipoConexion']))
        {
            if ($datosUser['tipoConexion'] == 'movil')
            {
                $this->seguridadHabilitada();
                $this->template->render('portada_movil');
            }
            else
            {
                redirect(base_url().'seguridad/acceso');
            }
        }
    }
    else
    {
        $this->seguridadDesabilitada();
        $this->template->render('portada_movil');
    }
}
```

Anexo 18: Función *index* de la clase controladora *portada*.

```
function index()
```

```
{
    $this->template->set_structure('desktop');
    if ($this->xml->weaver->security->enabled[0] == 'true')
    {
        $token = $this->session->userdata('token');
        $datosUser = $this->ioc->seguridad->obtenerElementosDadoToken($token);
        if (isset($datosUser['tipoConexion']))
        {
            if ($datosUser['tipoConexion'] == 'escritorio')
            {
                $this->seguridadHabilitada();
                $this->template->render('portada');
            }
            else
            {
                redirect(base_url().'seguridad/acceso');
            }
        }
    }
    else
    {
        $this->seguridadDesabilitada();
        $this->template->render('portada');
    }
}
```

}

Anexo 19: Atributo "movil".

Detalles del atributo de configuración

Nombre: movil
 Etiqueta: Movil
 Pertenece a: Tipo de nodo
 Descripción:
 Tipo: selección inclusiva
 Estado: Activo
 dominio: t

Cantidad por página 5

Texto	Valor
movil	t

Página 1 de 0

Cerrar

Detalles de nodo

Tipo de nodo: Aplicación
 Nombre: m_pregrado
 Descripción:
 Etiqueta: M Pregrado
 Estado: Activa
 Movil : movil

Cerrar

Anexo 20: Función *obtenerAplicaciones*.

```
public function obtenerAplicaciones($usuario=NULL)
{
    $security = $this->_ci->xml->weaver->security->enabled[0] == 'true' ? TRUE : FALSE;
    $aplicaciones = $this->_ci->ioc->seguridad->obtenerAplicacionPermisoDadoUsuario($security, 'f',
$usuario);
    $aplicacionesMoviles= $this->_ci->ioc->configuracion->obtenerNodosMoviles();
    $movil=FALSE;
    if($usuario != null)
```

```

    $aplicaciones = $this->_ci->ioc->seguridad->ordenarActivos($aplicaciones);
    foreach ($aplicaciones as $aplicacion)
    {
        $id_aplicacion = $aplicacion->id_aplicacion;
        foreach ($aplicacionesMoviles as $moviles)
        {
            if($id_aplicacion==$moviles->id_nodo)
            {
                $movil=TRUE;
            }
        }
        $aplicacion->movil=$movil;
        $modulos = $this->_ci->ioc->seguridad->obtenerModuloDadoldAplicacionUsuario($id_aplicacion, $security,'f',$usuario);
        if($usuario != null)
            $modulos = $this->_ci->ioc->seguridad->ordenarActivos($modulos);
        $aplicacion->modulos = $modulos;
    }
    return $aplicaciones;
}

```

Anexo 21: Función obtenerNodosMoviles.

```

public function obtenerNodosMoviles()
{
    return $this->_ci->tb_rnodo_atributo_valor_mdll->obtenerNodosMovil();
}

```

Anexo 22: Función obtenerNodosMovil.

```

public function obtenerNodosMovil(){
    $sql="SELECT
        tb_dnodo.id_nodo

```

```
FROM
sq_configuracion_sistema.tb_dnodo,
sq_configuracion_sistema.tb_rnodo_atributo_valor,
sq_configuracion_sistema.tb_natributo_configuracion
WHERE
tb_dnodo.id_nodo = tb_rnodo_atributo_valor.id_nodo AND
tb_natributo_configuracion.id_atributo = tb_rnodo_atributo_valor.id_atributo AND
tb_rnodo_atributo_valor.valor = 't';
return $this->db->query($sql)->result();
}
```

Anexo 23: Listado de funcionalidades adicionadas en el archivo IOC.

Módulo estudiante

- obtenerDatosPlanEstudioDadoldPersona
- obtenerDatosEstudianteDadoldPersona
- obtenerBonificaciones
- obtenerExamenPremio
- obtenerPromedioCarrera
- obtenerCantidadEstudiantesBuscar
- obtenerEstudiantesBuscador
- obtenerNotasParaResumenEvaluacion
- obtenerAnnosPersona
- obtenerPremios
- obtenerRegistroGeneral
- obtenerAbreviaturas

Módulo configuración

- obtenerNodosMoviles

Módulo carrera

- obtenerAsignaturas

Módulo personal secretaría

- fn_obtener_provincia

Módulo control docente

- obtenerPersonaLoggeada
- obtenerFormasOrganizativasDadoldGrupoDocente
- obtenerNombreEstadoAsistencia
- registrarRegistroAsistencia
- obtenerGruposDocentesDeProfesor
- obtenerEstudiantes
- obtenerTipologiasDadoldFormasOrganizativasIdGrupoDocente
- obtenerArgumentos

Módulo seguridad

- obtenerElementosDadoToken
- autenticar
- cerrarSesion

Módulo estructura composición

- obtenerEstructurasDadoldCategorialdEstructuraPadre

Anexo 24: Funciones `messageMovil` y `errorMovil`.

```
final protected function messageMovil($code)
{
    if(!isset($this->messages->$code))
    {
        $this->errorMovil (new Exception_Error('SYS003'));
    }
    else
    {
        $message = $this->messages->$code;
        $_SESSION['mensaje'] = (string) $message['message'];
    }
}
```

```

        $_SESSION['tipo'] = 'info';
    }
}
    
```

```

final protected function errorMovil($error)
{
    $_SESSION['mensaje'] = $error->getMessage();
    $_SESSION['tipo'] = 'error';
}
    
```

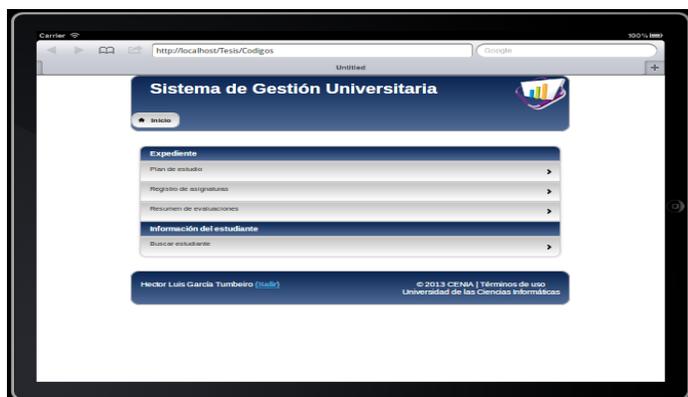
Anexo 25: Imágenes del sistema desde emuladores.



Emulador 1: The Responsinator
[\(<http://responsinator.com>\)](http://responsinator.com).



Emulador 2: iPhone Emulator
[\(<http://iphone-emulator.org>\)](http://iphone-emulator.org).



Emulador 3: iPad Peek (<http://ipadpeek.com>).



Emulador 4: Opera Mobile Emulator.



Emulador 5: Android Emulator (<http://android-emulator.org>).

Anexo 26: Test de usuario.

Objetivo: evaluar la usabilidad de la Versión móvil del Sistema de Gestión Académica de Pregrado.

• **Tareas a llevar a cabos por los usuarios estudiantes.**

1. Autenticar usuario.
2. Seleccionar el sistema de Pregrado.
3. Consultar el resumen de evaluaciones.
4. Buscar un estudiante.
5. Cerrar sesión.

• **Tareas a llevar a cabos por los usuarios profesores.**

1. Autenticar usuario.
2. Seleccionar el sistema de Pregrado.
3. Registrar asistencia.
4. Consultar registro de asistencia.
5. Cerrar sesión.

Las preguntas a realizar para conocer los comentarios de los usuarios para evaluar la usabilidad de la Versión móvil del Sistema de Gestión Académica de Pregrado se listan a continuación:

- ¿Cumplió sus objetivos al visitar el sistema?
- ¿Abandonó alguna tarea? ¿Por qué motivo?
- ¿El sitio puede ser usado sin desplazamiento horizontal?
- ¿Las fuentes son legibles?
- ¿Los avisos al usuario son breves y no ambiguos?
- ¿Detectó algún problema en el sistema? ¿Cuál?

Antes de comenzar el *test* se les pide a los usuarios que realicen todas las tareas enunciadas anteriormente. Una vez finalizado este proceso el encargado de la prueba realiza a los usuarios las preguntas definidas y apunta las respuestas emitidas por ellos.

Anexo 27: Caso de prueba: Autenticar usuario.

SC Autenticar usuario							
Escenario	Descripción	Usuario	Contraseña	Dominio	Versión	Respuesta del sistema	Flujo central
EC 1.1 Introducir datos correctamente.	El usuario introduce usuario, contraseña y selecciona el dominio y la versión del sistema que desee acceder.	V apsuarez	V *****	V uci.cu	V Móvil	El sistema verifica los datos y autentica a la persona, permitiéndole entrar a la vista de escritorio.	-El usuario llena los campos correctamente: usuario, contraseña, dominio, versión. - El sistema muestra la vista de escritorio.
EC 1.2 Introducir datos incorrectos	El usuario introduce de forma incorrecta los siguientes datos: el usuario y la contraseña.	I apsarez	V *****	V uci.cu	V Móvil	El sistema muestra un mensaje "No eres usuario de la aplicación".	-El usuario introduce los datos siguientes incorrectamente: usuario, dominio, contraseña. -Luego presiona el botón "Enviar" y el sistema muestra el mensaje "No eres usuario de la aplicación".
		I ssss	I ****	I grma.cu	V Móvil		
		V apsuarez	I ***	V uci.cu	V Móvil	El sistema muestra un mensaje "Su usuario o su contraseña están incorrectos".	-El usuario introduce la contraseña incorrectamente. -Luego presiona el botón "Enviar" y el sistema muestra el mensaje "Su usuario o su contraseña están incorrectos".

		V apsuarez	V *****	I gma.cu	V Móvil	El sistema muestra un mensaje "Ha ocurrido un error un error contacte al administrador".	-El usuario selecciona el dominio incorrectamente. -Luego presiona el botón "Enviar" y el sistema muestra el mensaje "Ha ocurrido un error contacte al administrador".
EC 1.3 Dejar los campos vacíos.	Mediante este escenario se dejan vacíos los campos: usuario, contraseña.	I (vacío)	I (vacío)	V uci.cu	V Móvil	El sistema muestra un mensaje "Campo requerido".	-El usuario accede al sistema y pulsa el botón "Enviar". - El sistema muestra el mensaje "Campo requerido".

Descripción de las variables.

No.	Nombre de campo	Clasificación	Valor nulo	Descripción
1	usuario	Campo de texto	No	Permite introducir el usuario en el sistema.
2	contraseña	Campo de texto	No	Permite al usuario introducir la contraseña.
3	dominio	Lista desplegable	No	Permite seleccionar el dominio en el que se encuentra el usuario, saliendo por defecto la opción <i>uci.cu</i> .
4	versión	Lista desplegable	No	Permite seleccionar la versión del sistema a la que desea acceder el usuario, saliendo por defecto la opción <i>Móvil</i> .

Anexo 28: Caso de prueba: Mostrar el resumen de evaluaciones.

Condiciones de ejecución

El usuario tiene que estar autenticado en el sistema.

SC Mostrar el resumen de evaluaciones

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Mostrar datos correctamente.	Mediante este escenario se muestra al usuario las evaluaciones obtenidas en cada asignatura cursada.	El sistema muestra el listado de las evaluaciones obtenidas en cada asignatura, organizadas por año académico y período lectivo.	-El usuario selecciona el sistema Pregrado. -Luego selecciona la funcionalidad "Resumen de evaluaciones". -El sistema muestra el listado de las evaluaciones obtenidas en las asignaturas.

Anexo 29: Caso de prueba: Mostrar registro de asignaturas.

Condiciones de ejecución			
El usuario debe estar autenticado en el sistema.			
SC Registro de asignaturas			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Mostrar datos correctamente.	Mediante este escenario se muestra al usuario las asignaturas cursadas junto con el resumen de notas y los datos de la asistencia a la misma.	El sistema muestra el listado de las asignaturas organizadas por año y período lectivo. Además muestra las evaluaciones y los por cientos de asistencias a cada asignatura.	-El usuario selecciona el sistema Pregrado. -Luego selecciona la funcionalidad "Registro de asignatura". -El sistema muestra un listado de las asignaturas ordenadas por año académico y período lectivo.

Anexo 30: Caso de prueba: Mostrar plan de estudio.

Condiciones de ejecución			
El usuario debe estar autenticado en el sistema.			
SC Mostrar plan de estudio			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Mostrar datos	Mediante este escenario se muestra al usuario las asignaturas	El sistema muestra el listado de las asignaturas	-El usuario selecciona el sistema Pregrado.

correctamente.	que tiene que cursar para poder terminar la carrera, separadas por año académico y período lectivo.	a cursar en la carrera, organizadas por año y período lectivo.	-Luego selecciona la funcionalidad "Registro de asignatura". -El sistema muestra el listado de las asignaturas a cursar en la carrera, organizadas por año y período lectivo.
----------------	---	--	--

Anexo 31: Caso de prueba: Mostrar persona.

Condiciones de ejecución			
El usuario debe estar autenticado en el sistema.			
El usuario tiene que realizar el <i>DCP_VM_Buscar</i> personas.			
SC Mostrar persona			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Mostrar datos correctamente.	Mediante este escenario se muestra al usuario el listado de personas existentes en el sistema, que cumplan con las condiciones puestas por el usuario.	El sistema muestra el listado de las personas existentes hasta la fecha, mostrando los datos: *foto *nombre y apellidos *provincia *facultad *grupo	-El sistema muestra un listado de las personas encontradas con los siguientes datos: *foto *nombre y apellidos *provincia *facultad *grupo

Anexo 32: Caso de prueba: Mostrar el resumen de evaluaciones de persona.

Condiciones de ejecución			
El usuario tiene que estar autenticado en el sistema.			
El usuario tiene que haber ejecutado <i>DCP_VM_BuscarPersona</i> y <i>DCP_VM_MostrarPersona</i>			
SC Mostrar el resumen de evaluaciones de persona			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Mostrar datos correctamente.	Mediante este escenario se muestra al usuario las evaluaciones obtenidas por una persona	El sistema muestra el listado las evaluaciones obtenidas por una persona	-El usuario ejecuta las condiciones de ejecución. -El sistema muestra el listado de

	determinada en cada asignatura.	determinada en cada asignatura, organizadas por año académico y período lectivo.	las evaluaciones obtenidas por una persona determinada en cada asignatura.
--	---------------------------------	--	--

Anexo 33: Caso de prueba: Registrar asistencia.

Condiciones de ejecución								
El usuario debe estar autenticado en el sistema.								
SC Registrar asistencia								
Escenario	Descripción	Fecha	Horas clase	Tipología	Asistencia	Argumento	Respuesta del sistema	Flujo central
EC 1.1 Insertar datos correctamente.	Mediante este escenario se registra la asistencia a los estudiantes de forma correcta.	V 02/10/ 2013	V Primera	V Conferencia	V Justificado	V Licencia cultural	El sistema registra el pase de asistencia y muestra el mensaje "La asistencia ha sido registrada satisfactoriamente".	* El usuario selecciona el sistema Pregrado y luego el módulo Control docente. * El usuario selecciona la funcionalidad "Registrar asistencia". * El sistema muestra los grupos asignados al profesor, el cual selecciona uno. * El usuario
		V 02/23/ 2013	V Ambas horas	V Taller	V Presente	V (vacío)		

								<p>llena los datos correctamente: fecha, hora clase, tipologías y pulsa "Aceptar".</p> <p>*El sistema muestra el listado de los estudiantes y el profesor selecciona el estado de asistencia y argumento.</p> <p>* El usuario presiona el botón "Aceptar" y el sistema muestra el mensaje "La asistencia ha sido registrada satisfactoriamente".</p>
EC 1.2 Insertar datos	Mediante este escenario no se introducen todos	V 01/10/ 2013	V	I 1 (vacío)	V Justificado	V Licencia Deportiva	El sistema muestra el mensaje:	* El usuario selecciona el sistema

incompletos.	los datos para registrar la asistencia.	V	V	I	V	V	"Campo requerido".	Pregrado y luego el módulo Control docente. * El usuario selecciona la funcionalidad "Registrar asistencia". * El sistema muestra los grupos asignados al profesor, el cual selecciona uno. * El usuario introduce algunos de los siguientes datos: fecha, hora clase, tipologías y pulsa "Aceptar". *El sistema muestra el listado de los estudiantes y el profesor
		03/16/2013		1 (vacío)	Justificado	Licencia Deportiva		
		V	I	I	V	V		
		04/06/2013	(vacío)	(vacío)	Justificado	Licencia Deportiva		
V	V	V	V	I	V	V		
		03/16/2013	Primera	Taller	Justificado	(vacío)		

								selecciona el estado de asistencia y argumento. * El usuario presiona el botón "Aceptar" y se muestra el mensaje "Campo requerido".
--	--	--	--	--	--	--	--	--

Descripción de las variables.

No	Nombre de campo	Clasificación	Valor nulo	Descripción
1	Fecha	Campo de texto	No	El campo permite introducir la fecha asociada al registro de la asistencia.
2	Horas clase	Campo de selección	No	El campo permite la selección de la hora del pase de asistencia de un turno de clases.
4	Tipología	Campo de selección	No	El campo permite la selección de la tipología correspondiente al pase de asistencia de un turno de clases.
5	Asistencia	Campo de selección	No	El campo permite seleccionar el estado de asistencia del estudiante.
6	Argumento	Campo de selección	Sí	El campo se habilita para el estado de asistencia que lo requiera, permitiendo seleccionar el argumento.

Anexo 34: Caso de prueba: Consultar registro de asistencia.

Condiciones de ejecución

El usuario tiene que estar autenticado en el sistema.

El usuario debe ser un profesor y debe tener asignado un grupo docente.

SC Consultar registro de asistencia

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Mostrar datos correctamente.	Mediante este escenario se muestra al profesor como se comporta la asistencia de los estudiantes de su grupo.	El sistema muestra el listado de los estudiantes pertenecientes al grupo seleccionado, mostrando los siguientes datos: foto, nombre y apellido, % de presente, % de injustificado y % de (A+J)/Plan.	<p>*El usuario selecciona el sistema Pregrado.</p> <p>*Luego selecciona el módulo Control docente y dentro de este la funcionalidad "Consultar registro de asistencia".</p> <p>*El sistema muestra un listado de los grupos docentes que tiene asignado el profesor y este selecciona el que desee consultar.</p> <p>*Después el sistema muestra el listado de los estudiantes del grupo con los siguientes datos: foto, nombre y apellido, % de presente, % de injustificado y % de (A+J)/Plan.</p>