Universidad de las Ciencias Informáticas

Facultad 3

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

Aplicación Android para la correcta visualización y funcionamiento del módulo estudiantes del Sistema de Gestión Académica de Pregrado en dispositivos móviles.

Autor:

Yoima Valido Cruz

Tutores:

Ing. José Alejandro García Calderón

Ing. Noel Ernesto Enamorado Selema

La Habana, agosto de 2020

“Año 62 de la Revolución”

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 firmo la presente a los ___ días del mes de _____ del año _____.

Autor:

_____

Yoima Valido Cruz

Tutores:

_____

Ing. José Alejandro García Calderón

_____

Ing. Noel Ernesto Enamorado Selema

AGRADECIMIENTOS

A mis padres, Susana Cruz Martínez y Ariel Valido Ulloa por ser mis ejemplos a seguir, por apoyarme incondicionalmente, por hacer de mí unamujer de bien, por siempre brindarme todo lo que estuvo a su alcance, por todos los sacrificios que han echo durante toda la vida para que no me faltara nada. Gracias por todo lo que me dieron y me siguen dando. Gracias. Los amo.

A toda mi familia, mis tías y tíos, mis hermanos, mis sobrinos, mis primas y primos. Gracias a todos por recorrer junto conmigo todos estos años y siempre estar pendientes a mí.

A mi esposo Adonis Benítez Leal, por estar siempre a mi lado en las malas y en las buenas. Gracias por apoyarme y comprenderme. Gracias por formar parte de mi vida. Te amo.

A todos los profesores que han contribuido a mi formación como profesional.

A mis amigos Mauro Sergio Sotolongo Montes, Juan Pablo Gainza García y Liz Yanet Duartes Valdivié, que se convirtieron en mis hermanos. Gracias por estar presentes noche y día, por reírse y también llorar conmigo. Gracias por ser mi mayor apoyo dentro y fuera de la universidad. Siempre los amaré y respetaré.

A mis amigos Danay, Lisandra, Flor María y Cristino Ernesto, gracias a ustedes pasé el año más divertido de la universidad (primer año). Gracias por hacerme reír tanto con sus ocurrencias. Siempre los recordaré como una de las cosas más bonitas que me llevo de la universidad.

A mis tutores José Alejandro y Noel Ernesto.

A mis compañeros de aula y de residencia. Gracias a todos porque de una forma u otra formaron parte del proceso.

DEDICATORIA

A toda mi familia y amigos por su apoyo incondicional, en especial a mis padres.

RESUMEN

Android es un sistema operativo (SO) orientado a dispositivos móviles además de ser un sistema de código abierto, multitarea, que permite a los desarrolladores acceder a las funcionalidades principales del dispositivo mediante aplicaciones, en la actualidad uno de los SO de mayor distribución en el mundo. La presente investigación aborda lo más significativo del desarrollode una aplicación Android del módulo estudiantes que se encuentra integrado al Sistema de Gestión Académica de Pregrado (Akademos) de la Universidad de las Ciencias Informáticas.Dicha aplicacióncorrige los problemas de visualización existentes en el módulo, ya que fue implementada siguiendo las pautas que propone “Mobile First” proporcionándole mayor accesibilidad, eficienciay permitiendo una interacción más amigable desde dispositivos móviles. Para guiar el proceso de desarrollose utilizó como metodología AUP-UCI, Java y XML como lenguajes de programación y Android Studiocomo Entorno de Desarrollo Integrado (IDE en sus siglas en inglés).La metodología de desarrollo utilizada permitió la correcta definición por parte del cliente de las Historias de Usuario (HU), logrando así ver con claridad los requisitos funcionales que hicieron posible el modelado de la solución parcialmente implementada. Lográndose de esta manera el cumplimiento de los requisitos definidos inicialmente y la validación de la solución mediante la completa satisfacción del cliente.

Palabras claves: AplicaciónAndroid,Akademos, plan de estudios, trámites docentes, registro de asignaturas, resumen de evaluaciones.

ABSTRACT

Android is an operating system (OS) oriented to mobile devices as well as being an open source, multitasking system that allows developers to access the main functionalities of the device through applications, currently one of the most widely distributed OS in the world. This research addresses the most significant aspects of the development of an Android application for the student's module that is integrated into the Undergraduate Academic Management System (Akademos) of the University of Computer Sciences. This application corrects the existing display problems in the module, since it was implemented following the guidelines proposed by "Mobile First", providing greater accessibility, efficiency and allowing a more friendly interaction from mobile devices. To guide the development process, AUP-UCI, Java and XML as programming languages and Android Studio as Integrated Development Environment (IDE) were used as methodology. The development methodology used allowed the correct definition by the client of the User Stories (HU), thus achieving a clear view of the functional requirements that made possible the modeling of the partially implemented solution. Achieving in this way the fulfillment of the initially defined requirements and the validation of the solution through complete customer satisfaction.

Keys words: Android application,Akademos, curriculum, teaching procedures, registration of subjects, evaluation summary.

TABLA DE CONTENIDOS

Tabla de contenido

TABLA DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

Introducción

En la actualidad la telefonía celular se ha convertido en una de las tecnologías más usadas a diario por millones de personas alrededor de mundo. Casi treinta años de evolución constante en las comunicaciones han permitido que los dispositivos móviles, se conviertan en una herramienta fundamental para el desarrollo cotidiano de cualquier actividad emprendida por el individuo, convirtiéndolo en el dispositivo electrónico más usado en el mundo.

"*Mobile First*" es un concepto utilizado en el diseño y la concepción de aplicaciones. Basándose en este, se crea una versión optimizada para dispositivos móviles y luego se amplía.La estrategia "*Mobile First*" se enfoca principalmente en lograr aplicaciones más usables para usuarios que interactúan con estas desde un dispositivo móvil, ya que estadísticamente el mayor tráfico que se genera en Internet es desde un Smartphone o Tablet, y no desde computadoras de escritorio(1).

La tendencia "*Bring your own device*" consiste en que el personal de una empresa o institución puede hacer uso de dispositivos personales para realizar sus labores. Permitiendo de esta manera no estar atado a un espacio y lugar, tributando a una mayor productividad(2)(3).

La Universidad de las Ciencias Informáticas (UCI) no queda exenta de estos fenómenos, es notable que la mayoría, tanto trabajadores como estudiantes cuentan con un dispositivo móvil que utilizan como apoyo en el quehacer diario en la universidad.

La UCI cuenta con áreas wifi desde donde con las credenciales del dominio se tiene acceso a los servicios brindados en el campus. Uno de estos servicios es la plataforma para la gestión académica de pregrado, AKADEMOS sistema donde se lleva el registro de la información más relevante del proceso docente en la universidad.

Se ha podido verificar que el sistema Akademos cuando se accede desde dispositivos móviles, no está desarrollado con un diseño completamente adaptativo, no se optimiza el uso del espacio en pantallas de pequeña y mediana resolución. Los componentes usados están enfocados a monitores de computadoras y en su mayoría se deforman en otro tipo de dispositivo, atentando así contra la usabilidad y productividad de los usuarios que requieren su uso desde sus propias terminales.

Las tecnologías de desarrollo sobre las que está soportada actualmente AKADEMOS dificultan una modificación para lograr que el sistema cumpla con pautas para una correcta visualización en resoluciones pequeñas. La librería usada para el desarrollo

de interfaz es jQuery en su versión 1.9, la cual, actualmente ya no cuenta con soporte y al utilizar con un gran número de componentes adaptados a las pautas de diseño de AKADEMOS se hace muy engorroso una actualización de esta y sus dependencias.

Este sistema cuenta con un módulo dedicado a los estudiantes, donde los mismos pueden acceder a la información referente a su perfil docente, plan de estudio, evaluaciones y comportamiento de asistencia. El módulo estudiantehereda los problemas de visualización del sistema Akademos, por lo tanto dificulta la consulta de la información docente del estudiante desde dispositivos móviles. Esto provoca que los estudiantes no accedan a estas funcionalidades desde este tipo de dispositivos.Descuidando el seguimiento de su información docente e impidiendo tomar medidas oportunas que tributen a la corrección de su rendimiento académico.

En la ilustración 1, se muestra el comportamiento visual de algunas funcionalidades de Akademos en dispositivos móviles. Como se evidencia en la primera interfaz, desde que se accede al módulo estudiantes no se obtiene una correcta visualización del contenido y las áreas de acción de los elementos son demasiado pequeñas. En la segunda interfaz, se evidencia que aun haciendo zoom(un acercamiento con el navegador) queda información relevante fuera de la pantalla(en el ejemplo ilustradono se muestran las notas obtenidas en las asignaturas), perdiéndose totalmente la capacidad de interacción con ella.

INTRODUCCIÓN

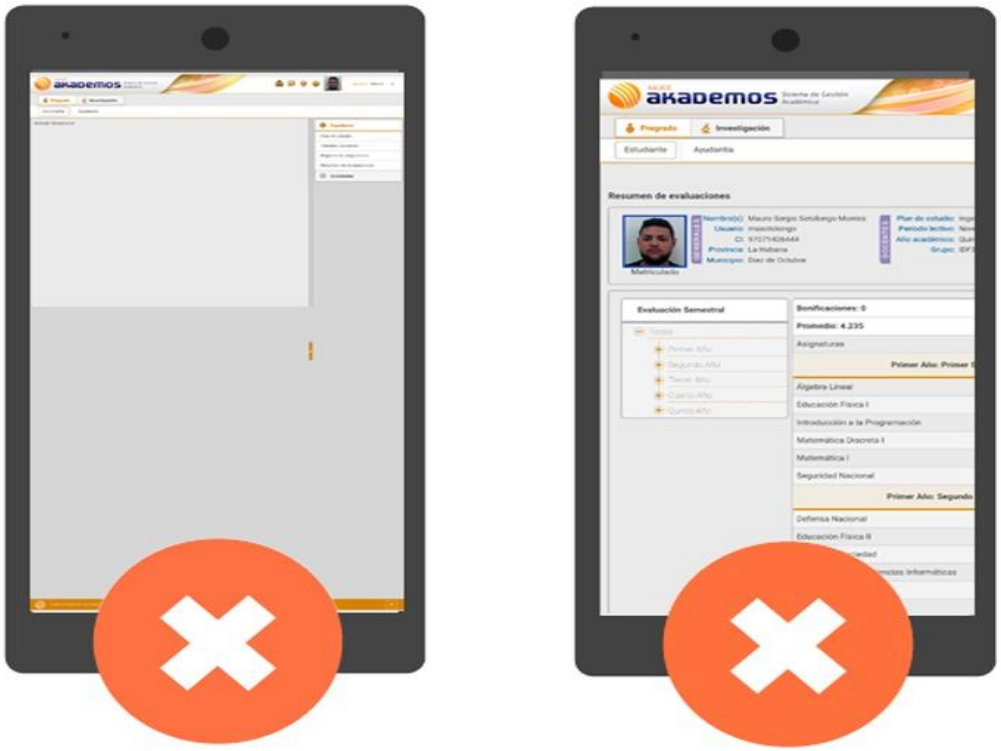

Ilustración 1: Problemas de visualización de Akademos en dispositivos móviles.

Por lo anteriormente expuesto es recomendado el desarrollo de una aplicación para dispositivos móviles que logre mitigar los problemas de visualización del módulo estudiantes, aprovechando las ventajas y potencialidades que las arquitecturas y tendencias actuales proponen para la implementación de aplicaciones móviles.

Teniendo en cuenta estos elementos, surge como **problema a resolver:**¿Cómo mitigar los problemas de visualización de la información del módulo estudiantes de Akademos a través de dispositivos móviles?

Tomando como **objeto de estudio:**Interfaces de usuario para dispositivos móviles.

Enmarcándose en el **campo de acción:** Interfaces de usuario para dispositivos móviles en el ámbito de la gestión académica.

Con el fin de solucionar el problema planteado se define como **objetivo general:**Desarrollar una aplicación Android que permita la correcta visualización del módulo Estudiantes de Akademos.

Posibles resultados: Una aplicación Android con el módulo estudiantes de Akademos completamente funcional. Para dar cumplimiento al objetivo propuesto se han derivado un conjunto de objetivos específicos orientados a proveer los elementos necesarios para la implementación de la solución, los cuales se listan a continuación:

- Elaborar el marco teórico de la investigación para sustentar los principales referentes teóricos en los que se enmarca el desarrollo de la propuesta de solución.
- Fundamentar la selección de la metodología, herramientas y tecnologías a utilizar en el desarrollo del módulo Estudiantes.
- Desarrollar una investigación del estado del arte.
- Definir las funcionalidades a implementar mediante la identificación de los requisitos funcionales.
- Generar principales artefactos de la metodología de desarrollo aplicada para el expediente de proyecto.
- Realizar el análisis, diseño y desarrollo del módulo Estudiantes.
- Efectuar las pruebas necesarias que garanticen la calidad del producto final.

En correspondencia con los objetivos previamente expuestos, se tienen como **tareas de investigación:**

1. Estudio del proceso de desarrollo de software y su metodología.
2. Estudio y análisis del estado del arte.
3. Estudio de la plataforma de desarrollo y de las herramientas utilizadas para diseñar, implementar y probar el sistema.
4. Definición de los requisitos funcionales y no funcionales según el proceso de desarrollo aplicado.
5. Modelar el sistema a través de Historias de Usuario.
6. Estudio y selección de los patrones arquitectónicos y de diseño más factibles para esta propuesta de solución.
7. Realización del Modelo de datos.
8. Realización del Diagrama de despliegue.
9. Desarrollo del módulo.

Para el desarrollo de las tareas de investigación serán empleados los métodos de Investigación. Se pueden clasificar en dos grupos de métodos de investigación: los métodos teóricos y los empíricos. Los primeros son todos aquellos que se basan en la utilización del pensamiento en sus funciones de deducción, análisis y síntesis, mientras que los métodos empíricos, se aproximan al conocimiento del objeto mediante sus conocimientos directos y el uso de la experiencia, entre ellos se encuentran, por ejemplo, la observación y la experimentación(4).Los métodosutilizados para dar solución a la presente investigación son:

INTRODUCCIÓN

Teóricos:

- **Histórico-Lógico:**este método ha sido aplicado en la búsqueda realizada de la documentación sobre diseños actuales y el aprovechamiento del área útil de la pantalla de dispositivos móviles en las aplicaciones para Android.
- **Análisis-Síntesis:**este método ha sido aplicado en el análisis de la documentación identificada, para extraer los elementos que propicien la solución a la problemática planteada, así como la síntesis de los elementos necesarios para la selección de las tecnologías y metodologías adecuadas para el desarrollo de la aplicación Android con la correcta visualización de sus funcionalidades.

Empíricos:

- **Observación:** La observación científica como método consiste en la percepción directa del objeto de investigación. Este método ha sido aplicado en la observación de las interfaces de las aplicaciones y el aprovechamiento del área útil de la pantalla. Es necesario lograr un diseño que permita visualizar correctamente las funcionalidades independientemente del tamaño de la pantalla.

El documento está conformado por tres capítulos, a continuación, se brinda un resumen de los contenidos que abordan los mismos.

Capítulo I. Marco teórico:

En este capítulo se realiza la elaboración del marco teórico donde se exponen los conceptos asociados a la solución de la problemática. También se describen y caracterizan la metodología, herramientas, tecnologías y los sistemas homólogos a utilizar para el desarrollo de la aplicación Android.

Capítulo II. Análisis y diseño:

En este capítulo se realiza un análisis de la propuesta de solución. Aborda elementos relacionados con ladescripción de los requisitos funcionales y no funcionales del sistema a implementar, además del diseño del mismo. Define la arquitectura, así como los patrones arquitectónicos y de diseñoa emplearse.

Capítulo III.Implementación y pruebas:

INTRODUCCIÓN

En el presente capítulo se lleva a cabo la representación del proceso de despliegue del sistema. Se realizará una breve descripción de las funcionalidades representadas en las interfaces de la aplicación. Se dejan plasmadas las pruebas necesarias que se le realizarán a la aplicación una vez concluida su implementación.

Capítulo I. Marco teórico

En el presente capítulo se abordan los conceptos fundamentales para sustentar los principales referentes teóricos en los que se enmarca el desarrollo de la propuesta de solución. Se realiza un estudio del estado del arte referente al problema de investigación, el cual arrojará claridad en las tendencias de diseños actuales y aportará elementos significativos al módulo a desarrollar. Se describen las tecnologíasy herramientas recomendadas para la implementación de aplicaciones móviles para Android y se fundamenta la selección de la metodología empleada para guiarel desarrollo.

1.1Elementos asociados al dominio del problema

BYON

Una de las nuevas tendencias en la utilización de dispositivos móviles como tablets, smartphones o notebooks, se está consolidando en el ámbito estudiantil y laboral. Es un concepto denominado BYOD (Bring Your Own Device o trae tu propio dispositivo) que está revolucionando la forma en que los estudiantes y empleados interactúan con los sistemas de información. Consiste básicamente en que los usuarios de una organización, utilizan sus propios dispositivos móviles para realizar su trabajo cotidiano, teniendo acceso a aplicaciones y datos propios de las instituciones educacionales y de las empresas. El origen del BYOD surge a partir de que los dispositivos móviles como IPhone y los basados en sistema operativo Android mejoraron sus capacidades de procesamiento en los últimos años, además que el número de usuarios propietarios de estos equipos, también se vio incrementado de manera considerable.

Pero una buena iniciativa BYOD en educación no sólo consiste en permitir a los estudiantes llevar al centro sus propios dispositivos y proporcionarles una conexión a Internet, sino que además debe conseguir que esos dispositivos sean usados para la enseñanza y el aprendizaje, tanto dentro como fuera del centro escolar (5).

Mejora de la calidad y la efectividad de la enseñanza y el aprendizaje

- ✓ La disponibilidad de los dispositivos por parte de los alumnos facilita la innovación pedagógica y aumenta las oportunidades para aprender mediante la exploración y la investigación, tanto dentro como fuera del centro escolar.
- ✓ Los dispositivos BYOD incrementan, además, el grado en que los docentes pueden proporcionar actividades de aprendizaje diferenciadas, adaptadas a las necesidades individuales de sus alumnos y a sus estilos de aprendizaje, ayudándolos a:

- mejorar su motivación, sobre todo la de aquellos estudiantes que no están demasiado involucrados y se sienten desmotivados.
- motivar a aquellos estudiantes, sobre todo los de formación profesional, que pueden encontrar aburridos los métodos de enseñanza y los estilos de aprendizaje tradicionales.
- ayudar a los estudiantes con menos habilidades TIC y a aquellos con discapacidades y necesidades educativas especiales.

- ✓ Los dispositivos BYOD permiten a los estudiantes acceder a los libros de textos digitales y a otros recursos de aprendizaje, en diferentes localizaciones.
- ✓ El uso de los dispositivos móviles ofrece a los estudiantes un mayor número de oportunidades para crear sus propios materiales de aprendizaje, además de poder acceder a contenidos educativos creados por otras personas. Pueden explotar las funciones de recopilación de datos integradas en los dispositivos móviles, incluyendo la capacidad de hacer fotografías, grabar vídeo y sonido, introducir texto y almacenar información sobre la localización. Pero también pueden combinar, editar, compartir y añadir datos a los ya recopilados o creados, contribuyendo al aumento de la comunicación, de la colaboración, del aprendizaje entre iguales y del trabajo por proyectos.
- ✓ Que los estudiantes usen sus propios dispositivos para el aprendizaje, les ayuda a desarrollar su competencia digital, más allá del uso social que hacen de ellos, que es sólo una pequeña parte de su potencial (5).

MOBILE FIRST

El mobile first es una filosofía, un conjunto de estrategias para diseñar pensando en los móviles, tomando en cuenta sus ventajas y desventajas para sacar provecho de las mismas (6).

Para medir la importancia de que esta filosofía obtiene en este tiempo donde los dispositivos móviles están cambiando al mundo, estadística realizada por Google: “En EE.UU., el 94% de las personas con smartphones buscan información local desde los teléfonos. Curiosamente, el 77% de las búsquedas en móviles se hacen desde casa o desde el trabajo, donde normalmente hay ordenadores.”

Ventajas de Mobile First:

- La principal ventaja es que el sitio está optimizado para dispositivos móviles.
- El contenido es priorizado sobre todo lo demás, pudiendo aumentar y complementar información, aprovechando la potencia de procesamiento y por supuesto el tamaño de la pantalla refiriéndose a un computador de escritorio.
- Disminuye el tiempo de descarga de los aspectos más importantes del sitio.
- Ofrece una experiencia estandarizada en todas las plataformas (6).

1.2 Estado del arte

Existen varias aplicaciones para dispositivos móviles que permiten la gestión y control docente, a continuación, se describen algunas de las más calificadas:

1.2.1 ClassLink

Este sistema de gestión, distribuido por Crambo, posibilita que los centros unifiquen y faciliten el acceso de alumnos y profesores a las diferentes plataformas web y recursos de software de los que dispongan. Permite que cada usuario acceda desde su perfil personal a todos los programas y portales que utilice en el aula: Google apps, SMART amp, Microsoft, etcétera. Además, genera estadísticas muy precisas relacionadas con el uso que se hace de los programas: tiempo, frecuencia, etcétera(7).

1.2.2 Pupilpro

La versión para docentes permite controlar todos los aspectos relacionados con el día a día: asistencia, unidades formativas, evaluación, diario de clase, información detallada el alumno, notas medias, perfil competencial, programaciones docentes. A través de Pupilpro Padres se puede enviar a las familias información relativa al alumnado, además de disponer de un chat para mantener conversaciones en tiempo real. También hay una versión para alumnos(7).

1.2.3 Gestión aula

El entorno de gestión integral de centros de Grupo Anaya es un proyecto modular que organiza tareas académicas, comunicaciones y la gestión interna de las instituciones educativas. Trabaja en la nube y es operativo con tabletas iOS y Android. Con Gestión Aula-Win, resuelven tareas administrativas y académicas, además de llevar a cabo la gestión de los recursos humanos y económicos. Su uso sebasa en perfiles de usuario con diferentes niveles de acceso y está conectado con la plataforma Greta y el Panel del profesor para favorecer el flujo de datos. Mientras, Gestión Aula-Web ofrece a los

centros una herramienta para crear su propio portal y mantener una comunicación constante con familias y alumnado (7).

El análisis realizado a las herramientas de control docente, permitió elaborar un resumen en forma de tabla teniendo en cuenta los siguientes parámetros:

- Tecnología: Referente a si la herramienta es una aplicación web o es nativa para sistema operativo Android.
- Licencia: Referente a si es libre de costo o es necesaria la compra de la misma.
- Procedencia: Referente a la procedencia del producto. Puede tomar valores Nacional y Extranjera.
- Interfaz adaptativa: Referente a si la herramienta se ajusta y optimiza el espacio en los diferentes tamaños de soluciones principalmente en dispositivos móviles.

Se seleccionan estas características porque se consideran necesarias para dar cumplimiento a los requerimientos del software. Los sistemas estudiados fueron provechosos, pues sirvieron de referencias para la arquitectura de la información, proporcionando buenas prácticas en aprovechamiento del espacio útil de la pantalla de dispositivos móviles.

Sistemas/Aspectos	Tecnología	Licencia	Procedencia	Interfaz Adaptativa
Pupilpro	**Aplicación Android**	**Privativa**	**Extranjera**	**Si**
ClassLink	**Aplicación Android**	**Privativa**	**Extranjera**	**Si**
Gestión aula	**Aplicación Android**	**Privativa**	**Extranjera**	**Si**

Tabla 1: Sistemas Homólogos.

De los sistemas informáticos identificados con el objetivo de mitigar los problemas de visualización, aunque aprovechanal máximo posible el espacio útil de la pantalla en los dispositivos con resoluciones pequeñas, no se pudo seleccionar ninguno para solucionar la problemática planteada porque no contemplan la gestión académica según las leyes vigentes en Cuba para los efectos.Los sistemas Pupilpro, ClassLink y

Gestión aula son de licencia privativa por lo que se imposibilita acceder a su código fuente para poder reutilizarlo y adaptarlo a la situación problemática.Todos estos elementos constituyen un factor importante para decidir apostar por el desarrollo de la aplicación con el módulo Estudiantes para dispositivos móviles.

1.3 Tecnologías y herramientas a utilizar

1.3.1 Sistema Operativo Android

Android es un sistema operativo que constituye una solución completa de software de código libre para teléfonos y dispositivos móviles. Es un paquete que engloba un sistema operativo, un “runtime” de ejecución basado en Java, un conjunto de librerías de bajo y medio nivel y un conjunto inicial de aplicaciones destinadas al usuario final. Android se distribuye bajo la licencia libre permisiva (Apache) que permite la integración con soluciones de código propietario.

Android surge como resultado de la Open Handset Alliance un consocio de 48 empresas distribuidas por todo el mundo con intereses diversos de tecnología móvil y un compromiso de comercializar dispositivos móviles en este sistema operativo. El desarrollo viene avalado principalmente por Google (tras la compra de Android Inc. en 2005) y entre otras compañías, se encuentran compañías de software (Ebay, LivingImage), operadores (Telefónica, Vodafone, T-Mobile), fabricantes de móviles (Motorola, Samsumg, Acer, LG, HTC) o fabricantes de Hardware (nVidia, Intel o Texas Instruments)(8).

Arquitectura de Android

Android presenta una arquitectura basada en 4 niveles como se muestra en la ilustración 2 Arquitectura de Android:

- Un **Kernel Linux** versión 2.6 que sirve como base de la pila de software y se encarga de las funciones más básicas del sistema: gestión de drivers, seguridad, comunicaciones.
- Una **capa de bibliotecas de bajo nivel en C y C++,** como SQLite para persistencia de datos; OpenGL ES para gestión de gráficos 3D, con aceleración 3D opcional y Webkit como navegador web embebido y motor de renderizado HTML.
- Un **framework para el desarrollo de aplicaciones**, dividido en subsistemas para gestión del sistema como el “Administrador de paquetes”, el “Administrador de telefonía” (para la gestión del hardware del teléfono anfitrión) o el acceso a APIs sofisticadas de geolocalización o mensajería

XMPP. Los desarrolladores tienen acceso completo a los mismos APIs del framework usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades ycualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework). Este mecanismo permite que los componentes sean reemplazados por el usuario. También incluye un sistema de vistas para manejar la interfaz de usuario de las aplicaciones, que incluyen la posibilidad de visualización de mapas o renderizado HTML directamente en la interfaz gráfica de la aplicación.

- **Aplicaciones:** Las aplicaciones base incluyen un teléfono, cliente de email, programa de envió SMS, calendario, mapas, navegador, contactos, que pueden a su vez ser usados por otras aplicaciones(8).

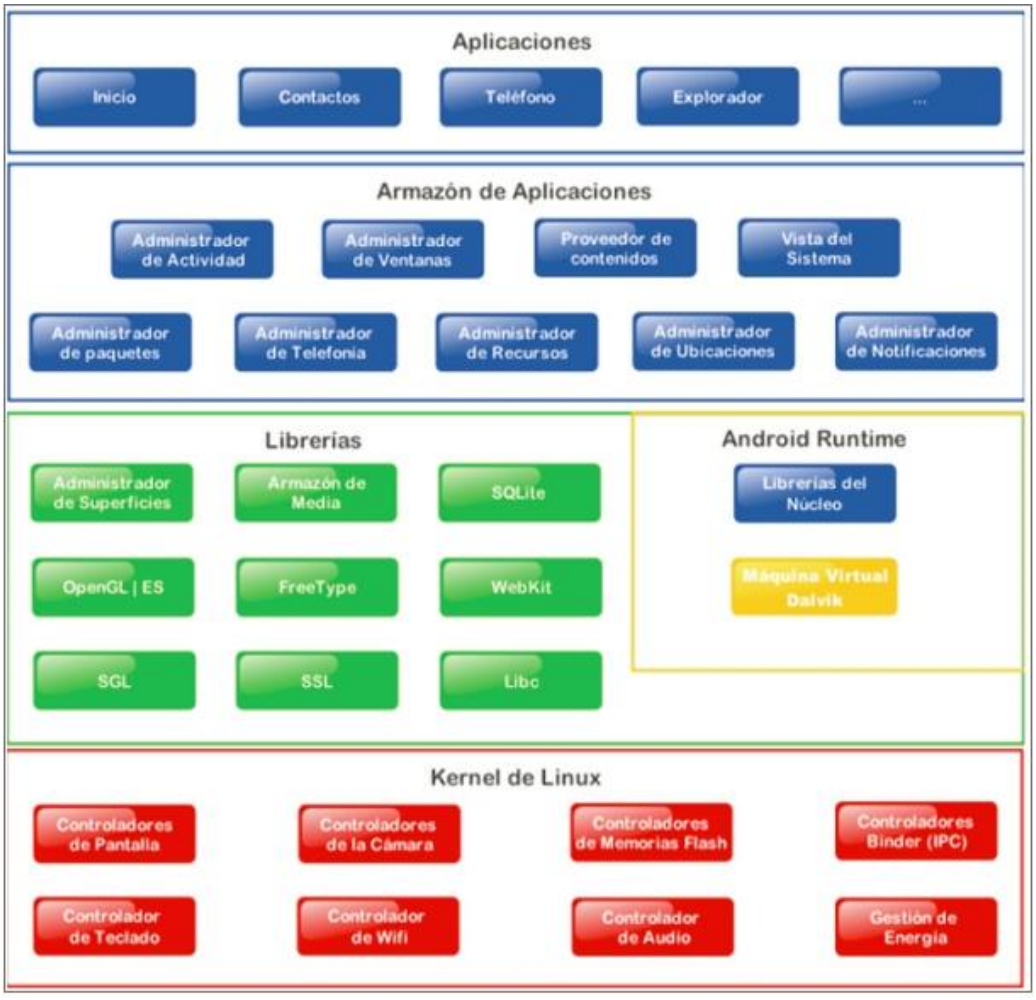

Ilustración 2: Arquitectura de Android.

1.3.2 Android Studio v2.3.3

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android y se basa en IntelliJ IDEA. El potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece aún más funciones que aumentan la productividad durante la compilación de apps para Android como:

- Un sistema de compilación basado en Gradle.
- Un emulador rápido con varias funciones.
- Un entorno unificado en el que se puede realizar desarrollos para los dispositivos Android.
- Instant Run para aplicar cambios mientras la app se ejecuta sin la necesidad de compilar un nuevo APK.
- Integración de plantillas de código y GitHub para ayudar a compilar funciones comunes de las apps e importar ejemplos de código.
- Gran cantidad de herramientas y frameworks de prueba(8).

1.3.3 Gradle v3.4

Gradle es una herramienta de automatización de compilación de código abierto centrada en la flexibilidad y el rendimiento. Los scripts de compilación de Gradle se escriben usando Groovy o Kotlin DSL.

- **Altamente personalizable:** Gradle se modela de una manera que se puede personalizar y ampliar de las formas más fundamentales.
- **Rápido:** Completa las tareas rápidamente reutilizando las salidas de las ejecuciones anteriores, procesando solo las entradas que cambiaron y ejecutando las tareas en paralelo.
- **Potente:** Es la herramienta de compilación oficial para Android y viene con soporte para muchos lenguajes y tecnologías populares(9).

1.3.4 Postman v7.31.1

Postman es una herramienta que se utiliza, sobre todo, para el testing de API REST, aunque también admite otras funcionalidades que se salen de lo que engloba el testing de este tipo de sistemas.

Gracias a esta herramienta, además de testear, consumir y depurar API REST, podremos monitorizarlas, escribir pruebas automatizadas para ellas, documentarlas, mockearlas, simularlas.

Es importante destacar también que, aunque no sea una de las herramientas más famosas para documentar API REST, genera una documentación bastante interesante y bastante atractiva, con ejemplos y fragmentos de código, de forma que hace que sea muy fácil de entender cómo funciona una API determinada(10).

1.3.5 JAVA

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable. Desde portátiles hasta centros de datos, desde consolas para juegos hasta super computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes, que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos diez millones de usuarios reportados(11).

1.3.6 XML

XML (Extensible Markup Language o Lenguaje de Marcado Extensible), se utiliza para representar información estructurada en la web, de modo que esta información pueda ser almacenada, transmitida, procesada, visualizada e impresa, por muy diversos tipos de aplicaciones y dispositivos. Se clasifica como un lenguaje extensible porque permite a sus usuarios definir sus propios elementos. Su objetivo principal es ayudar a los sistemas de información a compartir datos estructurados, particularmente a través de Internet, y se usa tanto para codificar documentos como para serializar datos(12).

1.3.7 pgAdmin3 v1.22.2

Entorno de escritorio visual libre y de código abierto. Instalable en plataformas Linux, FreeBSD, Solaris, Mac OSX y Windows. Permite conectarse a bases de datos PostgreSQL que estén ejecutándose en cualquier plataforma. Está disponible en diferentes idiomas.

Facilita la gestión y administración de bases de datos ya sea mediante instrucciones SQL o con ayuda de un entorno gráfico. Permite acceder a todas las funcionalidades de la base de datos; consulta, manipulación y gestión de datos, incluso opciones avanzadas como manipulación del motor de replicación Slony-I(13).

1.3.8 Pencil v2.0.3

Pencil Project es una herramienta útil de creación de prototipos de GUI que equipa a las personas creativas para diseñar, dibujar, analizar y finalizar sus ideas utilizando una amplia gama de elementos, incluidas formas comunes, elementos web básicos, Sketchy GUI, plantillas y más. Estos se pueden exportar en formato PNG, SVG,

HTML, PDF y ODT para aplicaciones en varios dominios de desarrollo con un plan artístico, pero técnicamente sólido, listo para terminar(14).

1.3.9 Node Js

Es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome.Es el entorno de servidor web de código abierto más popular de la actualidad desarrollado por Ryan Dahl y Joyent Inc (actualmente propiedad de Samsung Electronics). Al ser un entorno de tiempo de ejecución, Node.js ayuda a ejecutar códigos JavaScript fuera de un navegador. Utiliza el motor de Google Chrome para ejecutar JavaScript. En términos simples, ayuda a traducir los códigos de JavaScriptal lenguaje de nivel de máquina, por lo tanto, una máquina puede entender los códigos de JavaScript sin la necesidad de un navegador. Node.js en su conjunto facilita mucho el desarrollo de aplicaciones multiplataforma, del lado del servidor y de red (15).

1.3.10 JSON.

JSON acrónimo de JavaScript Object Notation (Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript, aunque hoy se considera un formato de lenguaje independiente. JSON se emplea habitualmente en entornos donde el tamaño del flujo de datos entre cliente y servidor es de vital importancia. Por lo antes expuesto, se emplea el formato JSON para el desarrollo del módulo en el paso de información entre la APK y el servicio web (16).

1.3.11 JWT

JSON Web Token (JWT) es un estándar abierto (RFC 7519) que define una forma compacta y autónoma de transmitir información de forma segura entre las partes como un objeto JSON. Esta información se puede verificar y confiar porque está firmada digitalmente. Los JWT se pueden firmar usando un secreto (con el algoritmo HMAC) o un par de claves pública / privada usando RSA o ECDSA (17).

1.4 Metodología de desarrollo de software

La revista electrónica International Journal of Computer Applications define una metodología de desarrollo como “un proceso mediante el cual un proyecto de software es completado o desarrollado a través de procesos o etapas bien definidas”. Según el concepto brindado por Pressman: “Se puede definir metodología de desarrollo de software como un conjunto de procedimientos, técnicas, herramientas y un soporte documental para el desarrollo de productos de software”.

1.4.1 Metodología AUP variación para la UCI

A raíz de la utilización de varias metodologías en los proyectos productivos de los centros de desarrollos de la universidad, surge la metodología AUP variación para la UCI, con el objetivo de estandarizar el proceso que guiará el desarrollo productivo para la misma. Esta metodología cuenta con las fases de Inicio, Ejecución y Cierre. Define las disciplinas para la fase de Ejecución: Modelado de negocio, Requisitos, Análisis y diseño, Implementación, Pruebas internas, Pruebas de liberación y Pruebas de aceptación. Para la disciplina de Requisitos define 4 escenarios, de los cuales, la presente investigación se guiará por el Escenario No 4, el cual plantea que los proyectos que no modelen negocio solo pueden modelar el sistema con HU.

El escenario No 4 aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido. El cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Se recomienda en proyectos no muy extensos, ya que una HU no debe poseer demasiada información. Todas las disciplinas antes definidas (desde Modelado de negocio hasta Pruebas de Aceptación) se desarrollan en la Fase de Ejecución, de ahí que en la misma se realicen Iteraciones y se obtengan resultados incrementales. En una iteración se repite el flujo de trabajo de las disciplinas, requisitos, análisis y diseño, implementación y pruebas internas. De esta forma se brinda un resultado más completo para un producto final de manera creciente.Para lograr esto, cada requisito debe tener un completo desarrollo en una única iteración(18).

1.5 Conclusiones

En este capítulo se abordan los principales conceptos de la investigación y el estado del arte que permitieron un mejor entendimiento del negocio y funcionamiento de aplicaciones que implementan buenas prácticas para la correcta visualización en dispositivos móviles.Se definieron las herramientas y tecnologías que actualmente son de las más utilizadas y recomendadas para el desarrollo de aplicaciones móviles.Se definió la metodología de desarrollo AUP en su variante para la UCI que servirá de guía para el proceso de implementación del módulo.Quedando de esta forma conformado el marco teórico conceptual de la investigación.

Capítulo II. Análisis y diseño

En el presente capítulo se lleva a cabo la descripción de la propuesta de solución definida para dar respuesta a la situación problémica existente a partir de las disciplinas de requisitos, análisis y diseño e implementación definidas por la metodología AUP variación para la UCI. Se hace un análisis de la arquitectura del sistema y de los patrones arquitectónicos y de diseño de la aplicación.

2.1 Propuesta de solución

Siguiendo la tendencia MobileFirst, es necesario enfocar el desarrollo es las funcionalidades más críticas para el negocio, aquellas que son las más utilizadas, es necesario la implementación del módulo estudiantes, teniendo como principales funcionalidades: Plan de estudio,Trámites docentes, Registro de asignaturas, Resumen de evaluaciones e Invitados.Como parte de las acciones para el desarrollo, al no contar con acceso a la plataforma de servicios de la universidad, se decide usar tecnologías alternativas, con el objetivo de simular la capa de servicios con la que se comunicará el sistema a implementar. Como una de las opciones seleccionada para enfrentar la limitante tecnológica, fue el uso del módulo de desarrollo de nodeJs denominado json-server este haciendo función como api rest brindará información lo mássimilar posible a la que deberá consumir el módulo una vez integrado con la plataforma de servicios de la universidad. La aplicación hereda los usuarios y permisos que ya existen hoy en Akademos, específicamente tendrán acceso al módulo estudiantes matriculados y profesores que sean graduados UCI.

No se pudo hacer uso de salvas pues la Dirección no cuenta con las bases de datos implicadas en la solución con información de prueba. No se procedió con la instalación de la plataforma de servicios (WSO2) vigente en la universidad por la complejidad de configuración y no poseer los requisitos mínimos de hardware que esta demanda para su optima ejecución.

2.2 Especificación de requisitos

Los requisitos constituyen un punto clave en el desarrollo de aplicaciones informáticas. Un gran número de proyectos de software fracasan debido a una mala definición, especificacióno administración de estos. Los requisitos se enfocan en las especificaciones de lo que se desea desarrollar y tienen dos clasificaciones: requisitos funcionales y no funcionales (19).

2.2.1 Requisitos funcionales

Los requisitos funcionales (RF) son la definición de los servicios que el sistema debe proporcionar, cómo debe reaccionar a una entrada particular y cómo se debe comportar ante situaciones particulares. Deben ser lo más completos, claros y

precisos posible(19). A continuación, se especifican los requisitos funcionales que dan respuesta a la propuesta solución.

No	Nombre	Descripción
RF1	Autenticar usuario	El usuario se autentica en el sistema con sus credenciales, obteniendo acceso a las funcionalidades según su nivel.
RF2	Mostrar Plan de estudio	El usuario visualiza los detalles de su plan de estudio, donde puede ver el conjunto de asignaturas que lo conforman agrupadas por períodos lectivos.
RF3	Listar Trámites docentes	Se listan los trámites docentes e información referente a estos, persona que realiza el trámite y la fecha de registro, esto siempre referente al estudiante autenticado.
RF4	Mostrar Registro de Asignaturas	El usuario visualiza las asignaturas cursadas en las que se le hayan registrado nota final.
RF5	Mostrar Resumen de evaluaciones	El usuario visualiza los detalles de lasevaluaciones obtenidas en las asignaturas.
RF6	Exportar Resumen de evaluaciones	Exporta un resumen de evaluaciones previamente visualizado en el RF5 en formato PDF.
RF7	Listar Reservaciones de invitados	Se listan las reservaciones de los invitados al evento de graduación de la universidad.
RF8	Registrar Invitados	El usuario registra un nuevo invitado en el sistema, la cantidad de registro está limitada a una configuración previa.
RF9	Modificar Invitados	El usuario modifica los datos de un invitado asociado a este.

RF10	Mostrar Invitado	El usuario visualiza los detalles de un invitado previamente seleccionado.
RF11	Listar Invitados	Se listan los invitados del usuario que se encuentra autenticado en el sistema que es desde donde tiene la opción para modificar o eliminar.
RF12	Eliminar Invitados	El usuario elimina del sistema un invitado previamente seleccionado.
RF13	Visualizar Reporte de invitados de la facultad	El usuario con nivel de acceso “vicedecano” visualiza el reporte de los invitados que pertenecen a estudiantes de su facultad.
RF14	Exportar Reporte de invitados de la facultad	El usuario con nivel de acceso “vicedecano” visualiza el reporte de los invitados que pertenecen a estudiantes de su facultad en formato PDF.

Tabla 2: Requisitos Funcionales

2.2.2 Requisitos No Funcionales de la propuesta de solución.

Son propiedades o cualidades que el producto debe tener. No se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes del mismo que hacen al producto atractivo, usable, rápido, confiable, etc. Normalmente son rasgos que se aplican al sistema en su totalidad y no a funciones específicas (19).

A continuación, se enumeran los requisitos no funcionales obtenidos luego de un proceso de levantamiento de requisitos mediante una serie de entrevistas con el cliente:

- **Interfaz: RnF1**El Área de acción de los elementos que permiten la interacción con el usuario con el sistema debe ser como mínimo de 44dp.
- **Interfaz: RnF2** El tamaño de fuente mínimo utilizado en el sistema debe ser de 12dp.
- **Interfaz: RnF3**No debe existir predominio de más de tres colores principales en la aplicación. Naranja, amarillo y blanco serán los colores primarios y secundarios respectivamente.
- **Confiabilidad/ Fiabilidad: RnF4** Garantizar la integridad y confidencialidad de la información mediante mecanismos de seguridad.Esto se garantiza con el

mecanismo de autenticación utilizado (JWT) y los protocolos de comunicación con los servicios a utilizar.

- **Confiabilidad/ Tolerancia al error: RnF5** Mostrar la información necesaria del error.
- **Usabilidad/ Instructibilidad: RnF6**La interfaz debe ser intuitiva, y lo más parecida posible a Akademos siempre que se corrijan los errores de visualización, ya que existe un grupo de usuarios que son profesores de avanzada edad, y estadísticamente este grupo de usuarios tiene un alto rechazo al cambio y una alta dificultad a la hora de interactuar con nuevas tecnologías.
- **Usabilidad/ Instructibilidad: RnF7** Mostrar elementos de apoyo a la navegación.
- **Usabilidad/ Operatividad: RnF8** Mostrar funcionalidades de acuerdo a los privilegios.
- **Disponibilidad: RnF9** Disponibilidad del módulo las 24 horas del día.
- **Software: RnF10** El cliente debe tener sistema operativo Android 5.0 o superior.
- **Soporte: RnF11** El módulo contará con toda la documentación definida en la presente investigación durante el proceso de desarrollo.

Nota: La definición técnica de ¨dp¨ es Density-Independient Pixel (Píxel de Densidad Independiente), hace referencia a la cantidad de pixeles que son necesarios para ocupar un área en la pantalla independientemente de la densidad que esta tenga.

2.3 Historia de Usuario

Las Historias de Usuario (HU) es un artefacto generado por metodologías ágiles como Scrum, XP y la variación de AUP-UCI específicamente en su escenario 4, en el cual se basa este proyecto. Son utilizadas con el fin de especificar los requisitos del software. A través de un conjunto de tablas se describen brevemente las características que desea el cliente. Contienen la información suficiente para que los desarrolladores puedan producir una estimación razonable del esfuerzo necesario para su implementación. Su contenido está estructurado de forma tal que se puedan descomponer en tareas para su posterior desarrollo entre 2 y 4 semanas (20).

Con el objetivo de presentar los requisitos funcionales de una manera más consistente y detallada, se realizaron las historias de usuario (HU) lo que permite hacer una

descripción de los requisitos funcionales de la aplicación. A continuación, se muestra un ejemplo de HU generada en la investigación.

HistoriadeUsuario	
Código:HU1	**NombreHistoriadeUsuario:**Autenticar usuario
ModificacióndeHistoriadeUsuarioNúmero:*ninguna*	
Referencia:RF1	
Programador: Yoima Valido Cruz	**IteraciónAsignada:***Primera*
Prioridad:*Alta*	**Tiempoestimado:***3 días*
RiesgoenDesarrollo:*Medio*	

Descripción: El usuario se autentica en el sistema con sus credenciales, obteniendo acceso a las funcionalidades según su nivel.

Existen dos campos de entrada de texto destinados al formulario y autenticación.

- Campo de texto (usuario): El usuario debe introducir su nombre de usuario correspondiente al sistema LDAP-UCI.
- Campo de texto (contraseña): El usuario debe introducir su contraseña correspondiente al sistema LDAP-UCI.

  El botón Iniciar sesión permite al usuario acceder al sistema.

Observaciones:

1. El usuario debe llenar los campos obligatoriamente, en caso contrario se mostrará un mensaje de error.
2. Si las credenciales son correctas se le dará entrada al usuario al sistema según su nivel de acceso, en caso contrario se mostrará un mensaje de error.

Rol: Usuario

Prototipodeinterfaz:

Tabla 3: Historia de Usuario“Autenticar Usuario”.

2.4 Arquitectura del sistema

La arquitectura de softwarees un conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software, lo que permite a los programadores, analistas y todo el conjunto de desarrolladores compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación. Es considerada el nivel más alto en el diseño de la arquitectura de un sistema puesto que establecen la estructura, funcionamiento e interacción entre las partes del sistema(19).

2.4.1 Arquitectura de Desarrollo de Software Cliente-Servidor a tres capas

Los sistemas cliente/servidor están construidos de tal modo que la base de datos puede residir en un equipo central, llamado servidor y ser compartida entre varios usuarios (19). Los usuarios tienen acceso al servidor a través de una aplicación cliente: En un sistema cliente/servidor de varios componentes, la lógica de la aplicación de cliente se ejecuta en dos capas:

- El cliente reducido se ejecuta en el equipo local del usuario y se encarga de presentar los resultados al usuario.
- La lógica del negocio se encuentra en aplicaciones de servidor que se ejecutan en un servidor. Los clientes reducidos solicitan funciones a la aplicación de servidor, que, a su vez, es una aplicación multiproceso capaz de operar con varios usuarios simultáneos. La aplicación de servidor es la que abre las conexiones con el servidor de la base de datos y se puede ejecutar en el mismo servidor que la base de datos, o se puede conectar a través de la red con otro servidor que opere como servidor de basede datos.

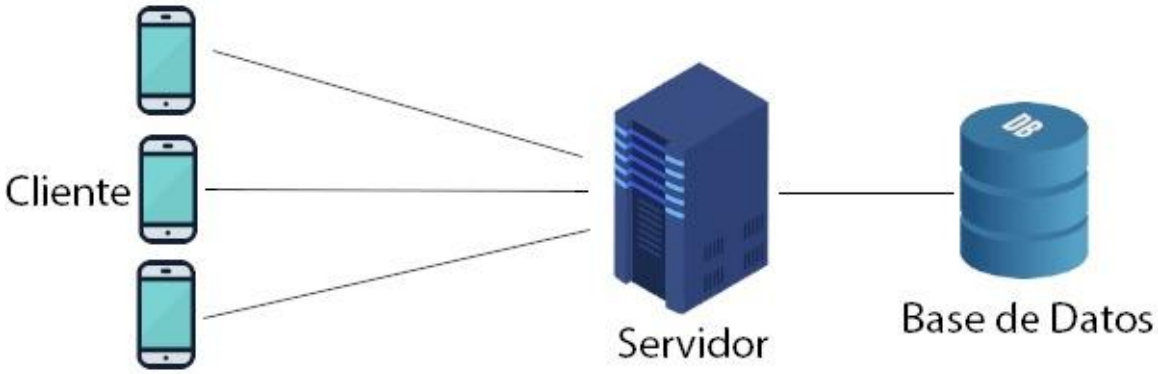

Ilustración 3: Arquitectura Cliente-Servidor.

Ventajas de esta arquitectura para el sistema que se desarrollará:

1. Al estar gestionada la lógica del negocio en el servidor, se consigue que:
   - La aplicación requiera una menor cantidad de espacio al ser instalada.
   - Consuma muy pocos recursos.
2. En cuestiones de seguridad de los datos, en la parte del cliente no existen sentencias SQL que permitan tener noción de que estructura tienen las tablas en la base de datos, y el mismo no tiene interacción directa con los datos, sino que es el proveedor de servicios quien realiza esta función en su logar.
3. Un cambio en la manera de representar los datos en la base de datos no tendría repercusión en la aplicación cliente, por lo que no sería necesario obligar a los usuarios a actualizar la APK para poder usar el sistema(19).

2.5 Patrones

Un patrón es un tipo de tema de sucesos u objetos recurrentes. Puede definirse como aquella serie de variables constantes, identificables dentro de un conjunto mayor de datos los cuáles se repiten de una manera predecible. Una plantilla o modelo puede usarse para generar objetos o partes de ellos, especialmente si los objetos que se

crean tienen lo suficiente en común para que se infiera la estructura del patrón fundamental, en cuyo caso, se dice que los objetos exhiben un único patrón. En la ingeniería del software, un patrón constituye el apoyo para la solución a los problemas más comunes que se presentan durante las diferentes etapas del ciclo de vida de un software (19).

2.5.1 Patrón de Arquitectura del Sistema

El patrón arquitectónico Modelo Vista Controlador (MVC) divide nuestra aplicación en tres niveles distintos, uno que representa a la interfaz gráfica (Vista), otro que representa al tratamiento de datos (Modelo) y otro que se encarga de toda la lógica que se tiene que llevar a cabo por la aplicación (Controlador). Esto se hace para permitir una mayor portabilidad de una aplicación, e incluso facilitar su mantenimiento. Pues si queremos modificar la apariencia de la aplicación sólo prestaremos atención a la Vista, si queremos cambiar de sistema de almacenamiento de datos tendremos que prestar atención a la capa del Modelo, y si lo que queremos es portar la aplicación a otra plataforma lo que haremos será modificar la capa del Controlador, al igual que si queremos modificar el código para optimizar el rendimiento(21).

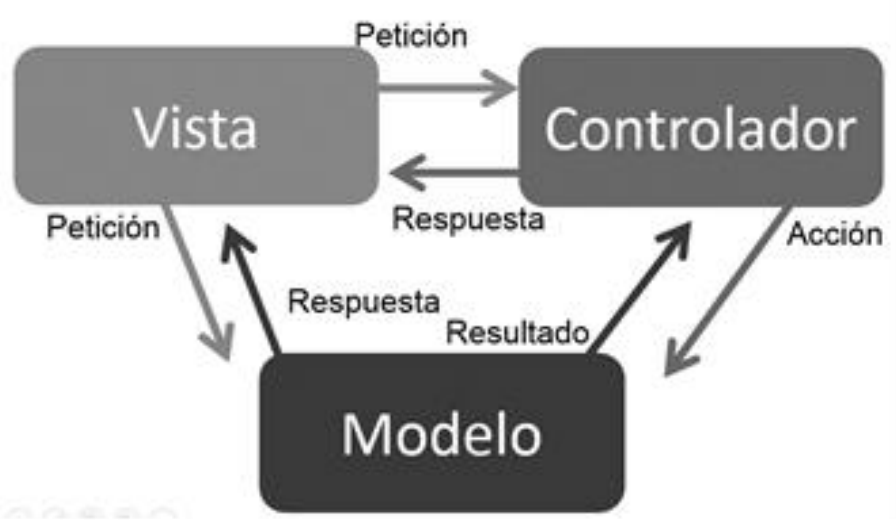

Ilustración 4: Modelo-Vista-Controlador.

- **El Modelo**.

  En la capa Modelo encontraremos siempre una representación de los datos del dominio, es decir, aquellas entidades que nos servirán para almacenar información del sistema que estamos desarrollando. Ejemplos de modelos presentes en el sistema serían las clases Estudiante, Carrera, Plan de estudio.

- **La Vista.**

Los componentes de la Vista son los responsables de generar la interfaz de nuestra aplicación, es decir, de componer las pantallas, páginas, o cualquier tipo de resultado utilizable por el usuario o cliente del sistema. Cuando las vistas componen la interfaz de usuario de una aplicación, deberán contener los elementos de interacción que permitan al usuario enviar información e invocar acciones en el sistema, como botones, cuadros de edición o cualquier otro tipo de elemento, convenientemente adaptados a la tecnología del cliente. En el sistema, son las *activities* en su componente .XML las encargadas de esta función.

- **El Controlador.**

  La misión principal de los componentes incluidos en el Controlador es actuar como intermediarios entre el usuario y el sistema. Serán capaces de capturar las acciones de este sobre la Vista, como puede ser la pulsación de un botón o la selección de una opción de menú, interpretarlas y actuar en función de ellas. Realizarán también tareas de transformación de datos para hacer que los componentes de la Vista y el Modelo se entiendan. Así, traducirán la información enviada desde la interfaz a objetos que puedan ser comprendidos por el Modelo.

  De igual manera que en la Vista, son las *activities* las encargadas de llevar a cabo dichas tareas, solo que en esta ocasión a través del componente .JAVA.

A continuación, en la ilustración 5 se mostrará cómo queda evidenciado el patrón MVC en el desarrollo de la aplicación:

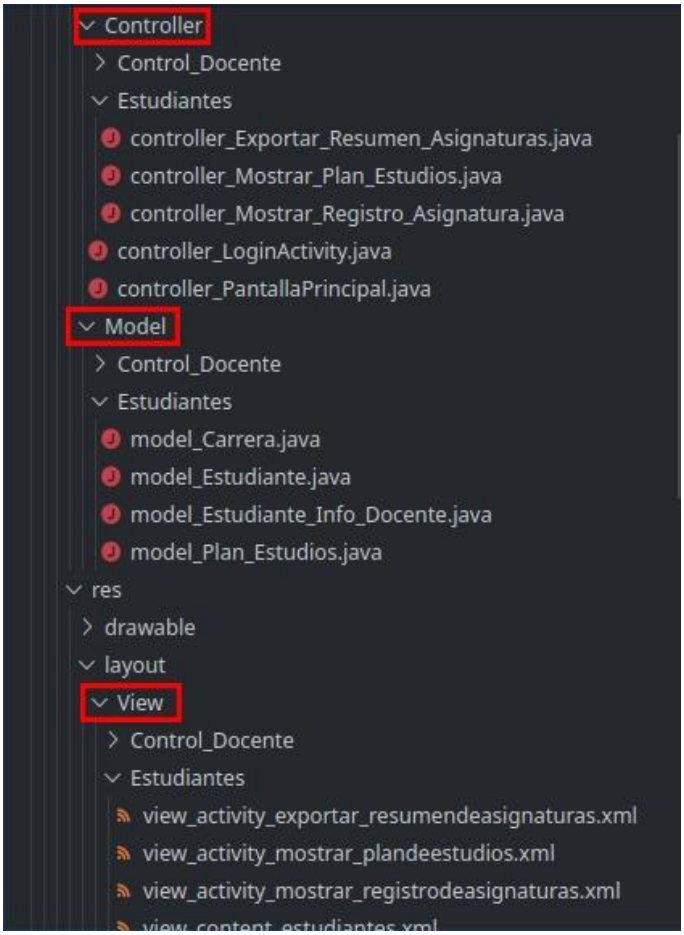

Ilustración 5: Estructura del módulo Estudiantes siguiendo el patrón arquitectónico MVC.

2.5.2 Patrones de diseño

Un patrón de diseño describe una estructura de diseño que resuelve un problema particular del del diseño dentro de un contexto específico y entre “fuerzas” que afectan la manera en la que se aplica y en la que se utiliza dicho patrón.

El objetivo de cada patrón de diseño es proporcionar una descripción que permita a un diseñador 1) si el patrón el aplicable al trabajo en cuestión, 2) si puede volverse a usar (con lo que se ahorra tiempo de diseño) y 3) si sirve como guía para desarrollar un patrón distinto en funciones o estructura(19).

Para el diseño de la propuesta de solución se define por parte del equipo de arquitectura utilizar los Patrones Generales de Software para Asignación de Responsabilidades (GRASP) y los Patrones del grupo de los cuatros: patrones GoF, A continuación, se enumeran los patrones GRASP y GoF utilizados:

2.5.3 Patrones Generales de Software para Asignar Responsabilidades (GRASP)

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones(22).

Controlador: El objetivo de este patrón es asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas, o sea, delega la responsabilidad en otras clases con las que mantiene un modelo de alta cohesión. El flujo del sistema que da paso entre una Activities a otra se delega en la clase Intent.

Bajo acoplamiento*:*Este patrónindica que tan fuerte está conectada una clase con otra, tiene conocimiento de, o influye sobre otra clase. Una clase con bajo acoplamiento no depende de otras clases.

Experto: Este patrón define que la clase experta en información es la que debe llevar la responsabilidad, ya que es la que contiene toda la información para cumplir con la tarea de instanciación. Este patrón se evidencia en la clase RegistrarInvitado.Java.

Creador: Esel encargado de crear las instancias de los objetos que manejan, favoreciendo así a la reutilización y el bajo acoplamiento. Este patrón se evidencia en la clase MostarResumendeEvaluaciones.Java.

2.5.4 Patrones GoF

Los patrones GOF son útiles durante el diseño de objetos y se clasifican en dependencia del propósito para los que hayan sido definidos: creación, estructurales y de comportamiento.

Los patrones de comportamiento plantean la interacción y cooperación entre las clases. Estudian las relaciones entre llamadas entre los diferentes objetos, normalmente ligados con la dimensión temporal(23).

En la solución propuesta se utilizó el patrón de comportamiento **Strategy (Estrategia)** que permite disponer de varios métodos para resolver un problema y elegir cuál utilizar en tiempo de ejecución. En la implementación realizada se utiliza en la GenerarPDF.Java.

2.6 Modelo de datos

El diseño de una base de datos consiste en definir la estructura de los datos que debe tener un sistema de información determinado. Para ello se suelen seguir por regla general unas fases en el proceso de diseño, definiendo para ello el modelo conceptual, el lógico y el físico.

En el modelo relacional las dos capas de diseño conceptual y lógico, se parecen mucho. Generalmente se implementan mediante diagramas de Entidad/Relación (modelo conceptual) y tablas y relaciones entre éstas (modelo lógico). Este es el modelo utilizado por los sistemas gestores de datos más habituales (SQL Server, Oracle, MySQL, etc.) (24).

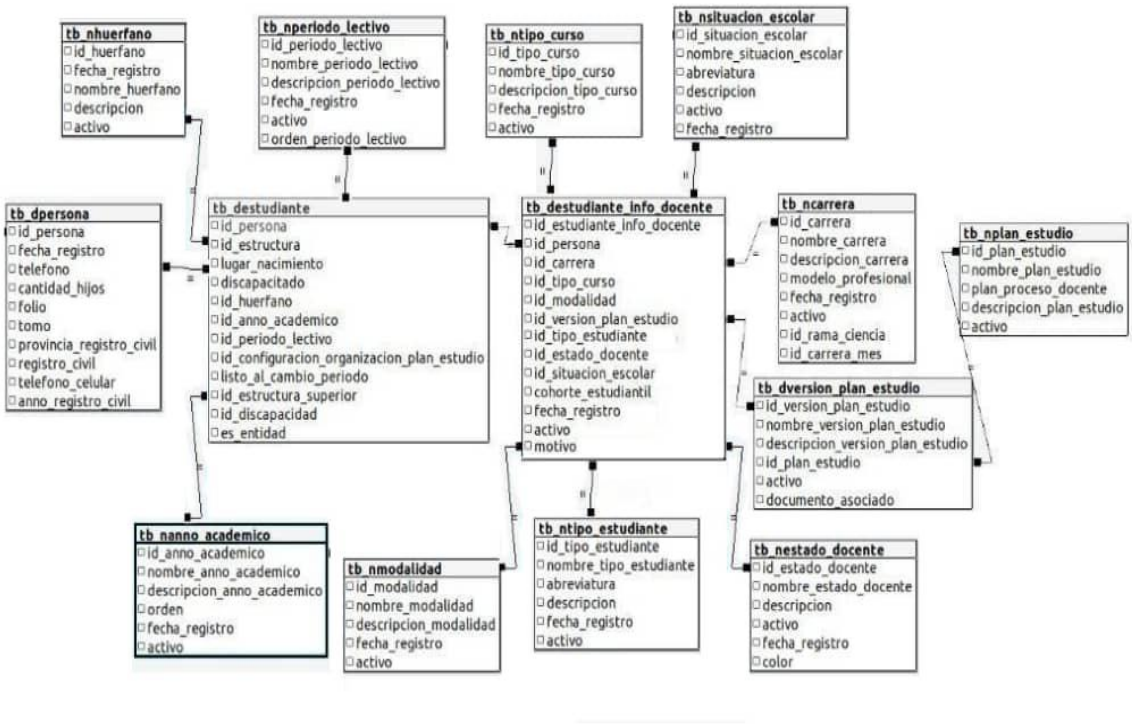

Ilustración 6: *Modelo de datos del módulo Estudiantes.*

2.7 Seguridad del Sistema

La aplicación del módulo estudiantes forma parte de un proyecto que tiene el objetivo de integrar varios módulos del sistema de gestión académica Akademos.

La aplicación del módulo Estudiantes hace uso de la seguridad que es común para todo el proyecto. Existe un Login que es quien accede al servicio de seguridad de la UCI. Este hace uso del servicio JWT para obtener el token de seguridad de autenticación del usuario y este posee información que define el nivel de acceso que tiene el usuario que se está autenticando en el sistema. Luego de realizar un inicio de sesión satisfactorio, el usuario es enviado a una vista denominada Pantalla Principal,

en la cual se presentan todos los módulos que incluye el sistema, solo aparecerán activos los que se encuentra permitidos para su nivel de acceso. Paraanalizar la información devuelta en el token es necesario decodificar el archivo JSON obtenido, para ello es necesario hacer uso de la librería para java JJWT. La información obtenida en este proceso de decodificación es enviada a través de las activities como parámetros del método Intent, para ser usadas por los diferentes módulos cuando lo necesiten.

2.8 Conclusiones

En el presente capítulo se realizó el análisis y diseño de la aplicación.Se hizo una breve descripción de la solución propuesta. La confección de las descripciones de requisitos por procesos, permitieron organizar el proceso de desarrollo del sistema.Además, el análisis de la arquitectura del sistemay utilización de los patrones arquitectónicosy de diseñoseleccionados, sentaron las bases para la disciplina de Implementación.

Capítulo III. Implementación y pruebas

En el presente capítulo se lleva a cabo la representación del proceso de despliegue para la aplicación Android. Quedará documentado el desarrollo parcial de la aplicación mostrando algunas vistas de las funcionalidades principales del módulo implementado y se dejarán plasmadas las pruebas que son necesarias realizarle al software una vez terminado su desarrollo total.

3.1 Modelo de despliegue

El propósito del modelo de despliegue es capturar la configuración de los elementos de procesamiento y las conexiones entre estos elementos en el sistema. Permite el mapeo de procesos dentro de los nodos, asegurando la distribución del comportamiento a través de aquellos que son representados(20).

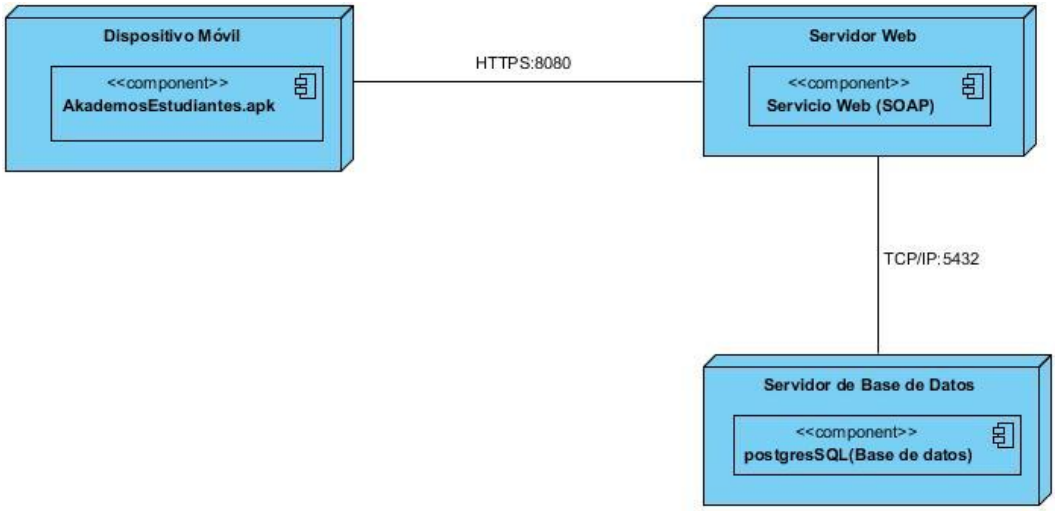

Ilustración 7: Diagrama de Despliegue.

3.1.1 Descripción de componentes

Nodo Dispositivo Móvil: En este nodo (ilustración 8) se instala la aplicación implementada desde la cual se consumirán los servicios webs.

Ilustración 8: Nodo Dispositivo Móvil.

Nodo Servidor Web: En este nodo (ilustración 9) se encuentran hosteados los servicios encargados de llevar a cabo los procesos del negocio y funciona como capa intermedia entre el Nodo Dispositivo Móvil y el Nodo Servidor de Bases de Datos.

Ilustración 9: Nodo Servidor Web.

Nodo Servidor de Base de Datos: En este nodo (ilustración 10) se almacenan todos los datos referentes al Módulo Estudiantes.

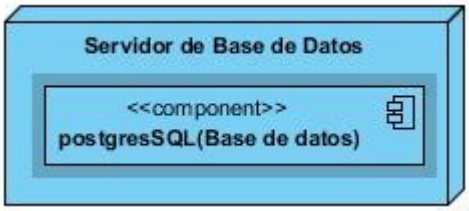

Ilustración 10: Nodo Servidor de Bases de Datos.

3.2 Implementación de la solución:

Una vez concluida la implementación parcial bajo las condiciones anteriormente descrita, se puede verificar que la aplicación está lista para realizar la integración con la plataforma de servicios de la universidad. Donde se recomienda después de un proceso de homologación de modelos de datos, procesamiento de respuestas, la misma vuelva a someterse al proceso de pruebas.

A continuación, se muestran las interfaces de la aplicación:

La interfaz (ilustración 11) muestra la vista Autenticar usuario de la aplicación.

Ilustración 11: Interfaz “Autenticar usuario”.

La interfaz (Ilustración 12) muestra la vista del Menú Lateral de la Pantalla Principal de la aplicación.

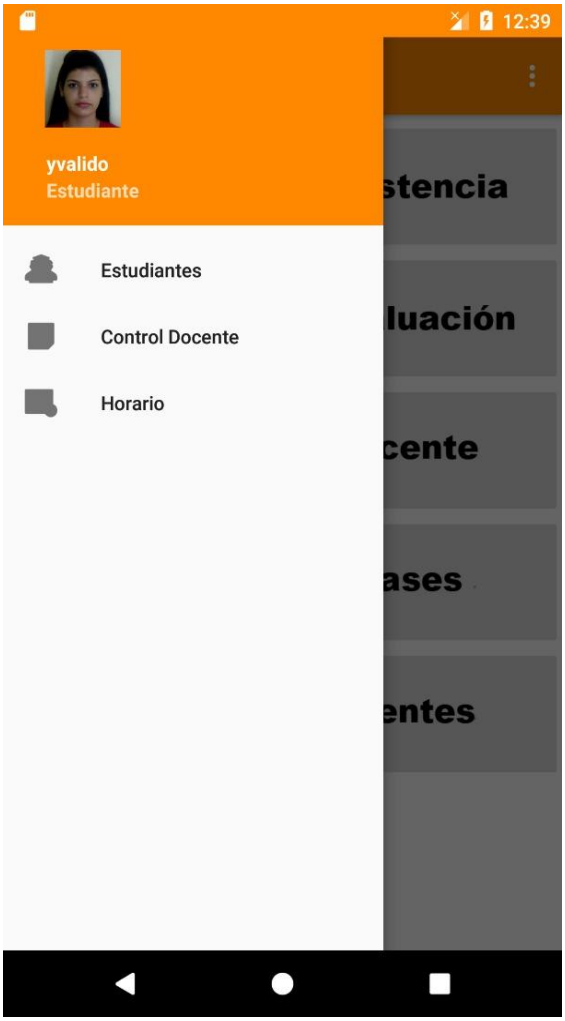

Ilustración 12: Interfaz “Menú lateral de la pantalla principal”.

La interfaz (ilustración 13) muestra la vista de la Pantalla Principal de la aplicación, donde aparecerán todas las funcionalidades a las que el usuario autenticado tenga acceso.

Ilustración 13: Interfaz “Pantalla Principal”.

3.3 Validación de la implementación

Las pruebas y validaciones de software son de vital importancia para corroborar que la aplicación desarrollada no presente problemas de ejecución y se encuentre libre de errores, ya que durante el proceso de desarrollo de software es frecuente que los desarrolladores, al programar las diferentes funcionalidades, obvien algunas posibilidades que pueden variar el resultado esperado durante la ejecución del código; por tanto, estas pruebas y validaciones constituyen la vía a través de la cual se asegura que el producto desarrollado está listo para ser entregado a los usuarios finales. Los errores más frecuentes pueden suceder a causa del mal uso de estructuras de datos, errores lógicos, entre otras (19).

Teniendo en cuenta la metodología de desarrollo que guía el presente trabajo, se deberán aplicar las siguientes pruebas una vez terminado el software:

3.3.1 Prueba de caja blanca

La prueba de caja blanca es una filosofía de diseño de casos de prueba que usa la estructura de control descrita como parte del diseño a nivel de componentes para derivar casos de prueba. Al utilizar el método de prueba de caja blanca, se pueden derivar casos de prueba que garanticen que todas las rutas independientes dentro de un módulo se revisaron al menos una vez. Dichos casos de pruebas pueden revisar todas las decisiones lógicas en sus lados verdaderos y falsos, y ejecutar todos los bucles en sus fronteras y dentro de sus fronteras operativas. Además, pueden revisar estructuras de datos internas para garantizar su validez(19).

En la presente investigación se utilizará la prueba de camino básico, la cual permite al diseñador de casos de prueba derivar una medida de complejidad lógica de un diseño de procedimiento y usarla como guía para definir un conjunto básico de rutas de ejecución. Los casos de prueba derivados para revisar el conjunto básico, tienen garantía para ejecutar todo enunciado en el programa al menos una vez durante la prueba.

Para realizar la prueba de camino básico es necesario conocer el número de caminos independientes de un determinado algoritmo mediante el cálculo de la complejidad ciclomática. Este se realiza de tres formas diferentes:

- El número de regiones del grafo de flujo coincide con la complejidad ciclomática.
- La complejidad ciclomática V(G) de un grafo de flujo G se define como: V(G)= A-N+2
- donde A es el número de aristas del grafo de flujo y N es el número de nodos del mismo.
- V(G) también se calcula como el resultado de P+1 donde P es el número de nodos predicados (nodos de los cuales parten dos o más aristas) que tiene contenido el grafo de flujo G(19).

3.3.2Prueba de caja negra

Las pruebas de caja negra, también denominadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del software. Permiten derivar conjuntos de condiciones de entrada que son las encargadas de revisar por completo todos los requerimientos funcionales para un programa.

Las pruebas de caja negra intentan encontrar errores como: funciones incorrectas o faltantes, errores de interfaz, errores en las estructuras de datos o en el acceso a bases de datos externas, errores de comportamiento o rendimiento, y errores de inicialización y terminación(19).

Para desarrollar la prueba de caja negra existen varias técnicas, entre ellas se encuentran:

Partición de equivalencia: La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar. Una clase de equivalencia representa un conjunto de estados válidos o no válidos para condiciones de entrada. Típicamente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógicas(19).

Pruebas de rendimiento: Las pruebas de rendimiento es un tipo no funcional de prueba para determinar la respuesta del sistema, es decir, velocidad, estabilidad, confiabilidad y escalabilidad. Entre los tipos de prueba de rendimiento se encuentra la prueba de carga. Dicha prueba se lleva a cabo para poner a prueba el comportamiento de la aplicación en diversos niveles de carga dentro de sus límites aceptables. Uno de los principales datos a centrarse durante la prueba de carga es "tiempo de respuesta"(19).

Pruebas de estrés: las pruebas de estrés se realizan para determinar el punto de equilibrio donde el rendimiento de aplicaciones deteriora. Principales parámetros para centrarse en las pruebas de tensión son "tiempo de respuesta" y "rendimiento"(19).

3.4 Estrategias de prueba

Una estrategia de prueba del software integra las técnicas de diseño de casos de prueba en una serie de pasos bien planificados que dan como resultado una correcta construcción del software. Inicialmente, las pruebas se enfocan en cada componente de manera individual hasta alcanzar el sistema en su totalidad(19).

Prueba de seguridad: las pruebas de seguridad intentan verificar que los mecanismos de protección que se construyen en un sistema en realidad lo protegerán de cualquier penetración impropia. Su objetivo es evaluar la confidencialidad, integridad y disponibilidad de los datos por lo que permiten validar que dichos datos o el sistema, solamente sean accedidos por los actores definidos y según sus niveles de acceso (19).

Prueba de integración: las pruebas de integración son una técnica sistemática para construir la arquitectura del software mientras se llevan a cabo pruebas para descubrir errores asociados con la interfaz. El objetivo es tomar los componentes probados de manera individual y construir una estructura de programa que se haya dictado por diseño. Las pruebas de integración verifican el correcto ensamblaje entre los distintos componentes con el fin de comprobar que interactúan correctamente a través de sus interfaces externas e internas, y cumplen con las funcionalidades establecidas(19).

Existen dos tipos de integración, no incremental e incremental. No incremental es cuando se combinan todos los módulos y se prueba todo el programa en su conjunto. La integración incremental es cuando el programa se construye y se prueba en pequeños segmentos del mismo. Existen dos estrategias de integración incremental: ascendente y descendente.

Prueba de rendimiento: las pruebas de rendimiento están diseñadas para probar la resistencia del software en tiempo de ejecución dentro del contexto de un sistema integrado (19).

3.5 Pruebas de aceptación

Es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido(19).

3.6 Conclusiones

En el capítulo se explicó resumidamente a través del diagrama de despliegue como quedaría la comunicación final de la aplicación y su interacción con las fuentes de datos una vez vaya a ser puesta en producción. Se hace un breve compendio de las pantallas que representan algunas de las funcionalidades de la aplicación desarrollada parcialmente y se dejan plasmadas las pruebas de software que se le realizarán a la aplicación una vez concluida su implementación.

Conclusiones Generales

Una vez concluida la investigación se obtuvo como resultado la implementación parcial de la aplicación y todo lo referente al marco teórico de la investigación, lo cual permite arribar a las siguientes conclusiones:

- ✓ A partir del análisis y diseño de la aplicación Android se obtuvieron artefactos necesarios para guiar el desarrollo de la misma.
- ✓ Del estudio de los sistemas homólogos se pudo obtener patrones de diseño que tributaron a la realización de una interfaz amigable y a un óptimo aprovechamiento del espacio útil de la pantalla.
- ✓ La tecnologías y herramientas empleadas, posibilitaron el desarrollo de una aplicación que cumple con los requisitos planteados, obteniéndose así, un producto en correspondencia a los resultados esperados.
- ✓ Los catorce requisitos funcionales que se definieron fueron debidamente implementados y se incluyó en la propuesta de solución las exigencias de todos los requisitos no funcionales detectados.
- ✓ La aplicación de las pautas para la implementación de la tendencia “Mobile First” permitieron darle un enfoque orientado a la óptima visualización al contenido del módulo en dispositivos móviles.

Recomendaciones

- Completar el desarrollo de las funcionalidades que quedaron pendientes.
- Realizar la integración con la plataforma de servicios de la universidad.
- Una vez integrada, realizar pruebas a la aplicación en un entorno donde se tenga acceso a la plataforma de servicios de la UCI.

Referencias Bibliográficas

1. **qualitydevs.** qualitydevs. [En línea] 2017. [Citado el: 22 de 12 de 2019.] https://www.qualitydevs.com/que-es-mobile-first-y-por-que-es-importante/. .

2. **ComputerHo.** Qué es BYOD? ventajas e inconvenientes. [En línea] 18. [Citado el: 25 de 10 de 2019.] https://computerhoy.com/noticias/moviles/que-es-byod-ventajaseinconvenientes-7250..

3. **Espeso, P.** EDUCACIÓN 3.0. «BYOD, trae tu propio dispositivo: el modelo que quiere revolucionar la educación» . [En línea] 2018. [Citado el: 2019 de 10 de 18.] https://www.educaciontrespuntocero.com/noticias/byod-bring-yourowndevice-educacion/32857.html.. .

4. *Métodos "I+D" de la Infórmatica.* **Barchini., Graciela Elisa.** Argentina : © LIEFIUBA, 2015, Vol. Vol. 2.Revista de Informática Educativa y Medios Audiovisuales. . . ISSN 1667-8338..

5. **(INTEF), Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado.** Diseñando el aula del futuro .Bring your own device (BYOD): una guía para directores y docentes . [En línea] 01 de 2016. [Citado el: 28 de 10 de 2019.] http://educalab.es/intef @educaINTEF http://educalab.es/blogs/intef/ .

6. **BURBANO, ANGIE MILENA HERNÁNDEZ.** DESARROLLO DE UNA GUÍA METODOLÓGICA PARA DISEÑO WEB ADAPTATIVO . [En línea] 2018. [Citado el: 23 de 01 de 2020.]

7. **Educaciontrespuntocero.** Educaciontrespuntocero.com. . [En línea] 2018. [Citado el: 18 de 01 de 2019.] https://www.educaciontrespuntocero.com/recursos/plataformasgestionescolar/.

8. **Developer., Android.** Android Developer. [En línea] 2019. [Citado el: 06 de 02 de 2020.] . [En línea] 2016. [Citado el: 06 de 02 de 2020.] http:// developer.android.com.

9. **docs.gradle.org.** docs.gradle.org. . [En línea] 2017. [Citado el: 13 de 02 de 2020.] https://docs.gradle.org/current/userguide/userguide.html.

10. **Cuervo., Víctor.** www.arquitectoit.com. . [En línea] 2017. [Citado el: 23 de 02 de 2020.] http://www.arquitectoit.com/postman/que-es-postman/. .

11. **desarrolloweb.com.** desarrolloweb.com. . [En línea] 2017. [Citado el: 02 de 23 de 2020.] https://desarrolloweb.com/home/java.. .

12. **developer.mozilla.org.** developer.mozilla.org. . [En línea] 2017. [Citado el: 24 de 02 de 2020.] https://developer.mozilla.org/es/docs/Web/XML/Introducci%C3%B3n_a_XML. .

13. **Tools», «pgAdmin - PostgreSQL.** «pgAdmin - PostgreSQL Tools». [En línea] 2018. [Citado el: 24 de 02 de 2020.] https://www.pgadmin.org/...

REFERENCIAS BIBLIOGRÁFICAS

14. **www.campusmvp.es.** www.campusmvp.es.. [En línea] 2016. [Citado el: 27 de 02 de 2020.]. [En línea] 2016. [Citado el: 27 de 02 de 2020.] https://www.campusmvp.es/recursos/post/Herramientas-de-prototipado-deaplicacionesWeb.aspx.. .

15. **nodejs.org.** nodejs.org. [En línea] 2018. [Citado el: 01 de 03 de 2020.] https://nodejs.org/es/...

16. **www.json.org.** www.json.org. . [En línea] 2018. [Citado el: 03 de 03 de 2020.] https://www.json.org/json-es.html.. .

17. **Tokens., jwt.io.** Introduction to JSON Web. . [En línea] 2018. [Citado el: 05 de 03 de 2020.] https://jwt.io/introduction/..

18. **Rodríguez., Tamara Sánchez.***Metodología de desarrollo para la Actividad productiva de la UCI. La Habana. .* La Habana : s.n., 2015.

19. **Maxim., Roger S. Pressman y Bruce R.***Ingeniería del Sortware.Un efoque práctico.Octava edición.* New York : McGraw-Hill Education, 2015. ISBN 978-0-07-802212-8..

20. **Somerville., Ian.***Ingeniería del software.Séptima edición. .* Madrid : PEARSON EDUCACIÓN., 2011.

21. **si.ua.es.** Modelo vista controlador (MVC). [En línea] 2017. [Citado el: 27 de 05 de 2020.] https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vistacontrolador-mvc.html..

22. **Informáticos., Departamento de Lenguajes y Sistemas.***Patrones de Asignación de Responsabilidades (GRASP). .* Sevilla : Universidad de Sevilla : Escuela Técnica Superior de Ingeniería Informática, 2016.

23. **C. A. Guerrero, J. M. Suárez, y L. E. Gutiérrez,.***«Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web ».* 2017.

24. **campusMVP.** Diseñando una base de datos en el modelo relaciona. . [En línea] 2016. [Citado el: 17 de 05 de 2020.] https://www.campusmvp.es/recursos/post/disenando-una-base-de-datos-en-el-modelo-relacional.aspx..