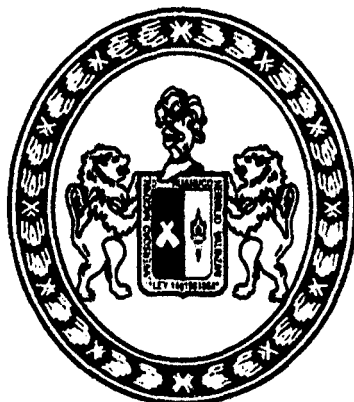


UNIVERSIDAD NACIONAL "HERMILIO VALDIZÁN"

FACULTAD DE INGENIERÍA INDUSTRIAL Y DE SISTEMAS

ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA DE SISTEMAS



**DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN
MÓVIL DE CONSULTAS ACADÉMICAS PARA
ESTUDIANTES DE LA UNHEVAL**

**TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE:
INGENIERO DE SISTEMAS**

TESISTAS:

**HEINZ YURI VARA AROSEMENA
YHON ELADIO PONCIANO TOLENTINO**

ASESOR:

Ing. BERNARDO TELLO, Alcides

HUÁNUCO - PERÚ

2015

UNIVERSIDAD NACIONAL HERMILIO VALDIZÁN
FACULTAD DE INGENIERÍA INDUSTRIAL Y DE SISTEMAS
ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA DE SISTEMAS



**DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL DE
CONSULTAS ACADÉMICAS PARA ESTUDIANTES DE LA UNHEVAL**

TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO DE SISTEMAS

TESISTAS:

**HEINZ YURI VARA AROSEMENA
YHON ELADIO PONCIANO TOLENTINO**

ASESOR:

Ing. BERNARDO TELLO, Alcides

HUÁNUCO - PERÚ

2015

DEDICATORIA

*A nuestros padres y seres queridos, con
todo nuestro esfuerzo y cariño.*

AGRADECIMIENTO

A Dios por habernos iluminado durante toda la carrera y habernos dado fuerzas para seguir siempre adelante.

A nuestros padres y hermanos, que siempre nos brindaron su apoyo y creyeron en nosotros.

A todos nuestros amigos que contribuyeron a que este proyecto concluya satisfactoriamente.

A nuestro asesor Alcides Bernardo por guiarnos y apoyarnos.

RESUMEN

En los últimos años los dispositivos móviles se han convertido en una herramienta esencial para diferentes actividades que los usuarios requieran realizarlas de manera fácil y rápida. Es por ello que se ha tomado en cuenta la necesidad de realizar una aplicación móvil que sea de gran beneficio para los estudiantes de la UNHEVAL.

En la presente tesis, se diseñó, desarrolló e implemento el aplicativo móvil de consulta académica para los estudiantes de la UNHEVAL, pudiendo acceder a los servicios desde sus dispositivos móviles (Smartphone y tablets) con sistema operativo basado en Android.

Luego de realizar reuniones de trabajo de levantamiento de información con los involucrados, se efectuó el análisis de requerimientos y de los componentes de hardware y software. Posteriormente, se efectuó el diseño, programación de la aplicación y el desarrollo de la web service.

Finalmente, se concluyó con la implementación del aplicativo móvil de tecnología Android, alojando los scripts de la web service en el servidor del área de informática de la UNHEVAL.

SUMMARY

In recent years, mobile devices have become an essential tool for different activities requiring users to perform easy and fast. That is why it has taken into account the need for a mobile application that will be of great benefit to students UNHEVAL.

In this thesis, we designed, developed and implemented the mobile application of academic inquiry for students UNHEVAL, being able to access services from mobile devices (smartphones and tablets) with operating system based on Android.

After conducting workshops to gather information with stakeholders, analysis of requirements and the hardware and software was performed. Subsequently, the design and programming of the application and development of the web service was performed.

Finally, it was concluded with the implementation of the Android mobile application of technology, staying the scripts of the web service on the server computer area of the UNHEVAL.

ÍNDICE DE CONTENIDO

CAPITULO I: PLANTEAMIENTO DEL PROBLEMA	10
1.1 DESCRIPCIÓN DEL PROBLEMA.....	10
1.2 DELIMITACIÓN.....	11
1.2.1 DELIMITACIÓN ESPACIAL.....	12
1.3 OBJETIVOS.....	12
1.3.1 OBJETIVO GENERAL.....	12
1.3.2 OBJETIVOS ESPECÍFICOS.....	12
1.4 HIPÓTESIS.....	13
1.5 JUSTIFICACIÓN.....	13
CAPITULO II: MARCO TEÓRICO	15
2.1 FUNDAMENTACIÓN TEÓRICA.....	15
2.1.1 ARQUITECTURAS CLIENTE-SERVIDOR.....	15
2.1.2 APLICACIONES PARA DISPOSITIVOS MÓVILES.....	15
2.1.3 INTRODUCCIÓN A LA TECNOLOGÍAS MÓVILES.....	16
2.1.4 DISPOSITIVOS MÓVILES.....	16
2.1.5 TELÉFONOS INTELIGENTES.....	17
2.1.6 SISTEMA OPERATIVO ANDROID.....	18
2.1.7 ARQUITECTURA ANDROID.....	19
2.1.8 CICLO DE VIDA DE UNA APLICACIÓN ANDROID.....	20
2.1.9 INTERFAZ MÓVIL.....	28
2.1.10 CONSUMO DE DATOS.....	29
2.1.11 CONEXIÓN DE DATOS MÓVILES.....	31
2.1.12 SEGURIDAD EN APLICACIONES MÓVILES.....	32
2.1.13 METODOLOGÍA PARA APLICACIONES MÓVILES.....	33
2.1.14 METODOLOGÍA SCRUM.....	33
2.2 PROPUESTA DE LA SOLUCIÓN.....	34
CAPITULO III: MARCO METODOLÓGICO	35
3.1 TIPO Y DISEÑO DE INVESTIGACIÓN.....	35
3.2 POBLACIÓN Y MUESTRA.....	35

3.3 TÉCNICAS DE RECOJO DE DATOS.....	35
3.4 INSTRUMENTOS DE RECOLECCIÓN DE DATOS, Y VALIDACIÓN DEL INSTRUMENTO	36
3.5 MODALIDAD DE LA INVESTIGACIÓN	36
CAPITULO IV: RESULTADOS Y DISCUSIÓN.....	40
4.1 ANÁLISIS	40
4.1.1 IDENTIFICACIÓN DE REQUERIMIENTOS	40
4.1.2 LISTA DE REQUERIMIENTOS	40
4.1.3 IDENTIFICACIÓN DE HISTORIAS DE USUARIOS.....	42
4.1.4 DESCRIPCIÓN DE LAS HISTORIAS DE USUARIO	43
4.1.5 ANÁLISIS DE LA SOLUCIÓN.....	43
4.1.6 NECESIDADES DEL CLIENTE	44
4.1.7 VIABILIDAD DEL SISTEMA.....	45
4.1.8 ANÁLISIS TÉCNICO - ECONÓMICO	45
4.1.9 ASIGNACIÓN DE FUNCIONES.....	46
4.2 DISEÑO	47
4.2.1 ARQUITECTURA DE LA SOLUCIÓN.....	48
4.2.2 DISEÑO DE LA INTERFAZ GRÁFICA.....	51
4.2.3 MODELO DE VENTANAS	52
4.3 CONSTRUCCIÓN	58
4.3.1 CONSTRUCCIÓN DE LA APLICACIÓN MÓVIL	58
4.3.2 CONSTRUCCIÓN DE LA WEB SERVICE.....	59
4.3.3 PRUEBAS	59
CONCLUSIONES	61
RECOMENDACIONES.....	62
BIBLIOGRAFÍA.....	63
ANEXOS	65

ÍNDICE DE IMÁGENES

FIGURA 2-1: Arquitectura Android.....	19
FIGURA 2-2: Ciclo de Vida de una aplicación Android.....	22
FIGURA 2-3: Métodos del ciclo de Vida de una aplicación Android.....	24
FIGURA 2-4: Servicios Web	30
FIGURA 4-1: Historia de Usuario	42
FIGURA 4-2: Diagrama de arquitectura en 3 capas.....	49
FIGURA 4-3: Navegación de la aplicación móvil.....	52
FIGURA 4-4: Pantalla de Inicio	53
FIGURA 4-5: Pantalla de menú Académico	54
FIGURA 4-6: Pantalla de Notas	55
FIGURA 4-7: Pantalla de Horario.....	56
FIGURA 4-8: Pantalla de Matricula	57

ÍNDICE DE TABLAS

TABLA 4-1: Lista de requerimientos	41
TABLA 4-2: Asignación de funciones.....	47

CAPITULO I: PLANTEAMIENTO DEL PROBLEMA

1.1 Descripción del problema

En la página web de la Universidad Nacional Hermilio Valdizán, se lleva a cabo las consultas académicas de los estudiantes, sin embargo este método es tradicional en relación con la tecnología móvil que actualmente muchas personas utilizan, y que es fomentada por la ONGEI en las entidades del estado.

El sistema web no fue creado con una finalidad de aplicación web móvil, es decir que pudiendo ingresar desde estos dispositivos a la intranet resulta tedioso y desagradable realizar una consulta, por la demora que esta suscita al acceder y descargarse la página completa de la UNHEVAL y en ocasiones estos dispositivos necesitan de unos plugins para poder visualizar los contenidos y poder interactuar con el sistema, a todo esto se suma el mayor consumo en el plan de datos del visitante; es por ello que se desarrolló e implementó de una aplicación móvil nativa Android para consulta académica, donde los estudiantes podrán acceder a sus consultas de manera fácil y rápida, con menor consumo de datos y sin la necesidad de descargarse una extensión.

El uso de dispositivos móviles puede verse como un componente de innovación que a la vez puede contribuir con el mejoramiento de cualquier sector.

Problema General

¿De qué manera optimizamos el servicio de consulta académica mediante el uso de la aplicación móvil de tecnología Android?

Problemas Específicos

- ¿Cuáles son los requerimientos mínimos de hardware y software para el desarrollo e implementación de la aplicación móvil de consulta académica?
- ¿Cómo es el diseño y Codificación del sistema de acceso a las consultas académicas para los estudiantes de la UNHEVAL utilizando plataforma Android?
- ¿Cómo verificar el correcto funcionamiento de consultas académicas para los estudiantes de la UNHEVAL?

1.2 Delimitación

- Área Académica: Software.
- Línea de Investigación: Tecnología y Desarrollo de Software.
- Sub línea de Investigación: Aplicaciones para dispositivos móviles.

1.2.1 Delimitación Espacial

- Universidad Nacional Hermilio Valdizan, Facultad de Ingeniería Industrial y Sistemas.

1.3 Objetivos

1.3.1 Objetivo General

Optimización del servicio de consulta académica para estudiantes de la Universidad Nacional Hermilio Valdizan mediante una aplicación móvil en plataforma Android.

1.3.2 Objetivos Específicos

- Analizar los requerimientos mínimos de hardware y software para el desarrollo e implementación de la aplicación móvil de consulta académica.
- Diseñar y Codificar el sistema de acceso a las consultas académicas para los estudiantes de la UNHEVAL utilizando plataforma Android.
- Implementar la aplicación para verificar el correcto funcionamiento de consultas académicas de los estudiantes de la UNHEVAL.

1.4 Hipótesis

- **Hipótesis:**

Mediante una aplicación móvil implementada en plataforma Android, optimizamos el servicio de consulta académica de estudiantes en la Unheval.

- **Variables:**

- **Variable dependiente:**

Optimización del servicio de consulta académica de estudiantes.

- **Variable independiente:**

Aplicación móvil en plataforma Android.

1.5 Justificación

El desarrollo de esta aplicación móvil es de gran utilidad, debido a que hoy en día el avance tecnológico en dispositivos móviles es muy grande, y con precios cada vez menores.

Esta investigación es de gran importancia debido a que ayudará a extender el servicio que ya presta la UNHEVAL para las consultas académicas de los estudiantes, brindando un adecuado manejo desde cualquier parte y en tiempo real con solo acceder a la aplicación, además se adaptará a cualquier teléfono móvil que soporte la plataforma

android. Esta aplicación se verá novedosa, debido a que muchos usuarios que posean esta tecnología podrán acceder al reporte de sus notas, record, horarios desde su teléfono móvil, con una facilidad y comodidad que ofrece la tecnología.

CAPITULO II: MARCO TEÓRICO

2.1 Fundamentación Teórica

2.1.1 Arquitecturas Cliente-servidor.

Desde el punto de vista funcional, se puede definir la computación Cliente/Servidor como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma.

En el modelo cliente servidor, el cliente envía un mensaje solicitando un determinado servicio a un servidor (hace una petición), y este envía uno o varios mensajes con la respuesta (provee el servicio). En un sistema distribuido cada máquina puede cumplir el rol de servidor para algunas tareas y el rol de cliente para otras. [1].

2.1.2 Aplicaciones para dispositivos Móviles

Una aplicación móvil es un software escrito para dispositivos móviles que realiza una tarea específica, como un juego, un calendario, un reproductor de música, etc.

Se define como nativa aquella aplicación específicamente diseñada para ejecutarse en el sistema operativo de un dispositivo y en el firmware de la máquina, y que, por lo general, tiene que ser

adaptada para distintos dispositivos. Una aplicación Web o una aplicación de navegador, en cambio, es aquella en la cual la totalidad o algunas partes de los programas se descargan de la Web cada vez que se ejecuta. Por lo general, se puede acceder desde todos los dispositivos móviles con capacidad Web. [2].

2.1.3 Introducción a las tecnologías móviles

Durante los últimos años, la tecnología celular ha alcanzado niveles altos en desarrollo de equipos y su evolución continúa siendo extremadamente acelerada. En efecto, la evolución de la tecnología móvil ha permitido llevar al mercado soluciones que brindan rentabilidad y ofrecen una mejor calidad de vida, ya que ponen al servicio del cliente la integración de las comunicaciones con la información. Hoy en día, se pueden observar como el desarrollo de aplicaciones para dispositivos móviles va creciendo en el mercado comercial y se puede apreciar como aplicaciones que antes solo se podían manejar en un computador, funcionan en forma similar en un dispositivo móvil. [3].

2.1.4 Dispositivos Móviles

Los dispositivos móviles constituyen cada vez más una realidad que ofrece al usuario, en un mismo y reducido aparato, funciones de comunicación y procesamiento de datos que van mucho más

allá de las simples llamadas telefónicas o la ejecución de aplicaciones básicas. El gigante de Internet Google ha presentado un nuevo sistema operativo para este tipo de dispositivos, Android, que busca ser una firme alternativa a otros sistemas ya ampliamente extendidos como iPHONE o Windows phone. [4].

2.1.5 Teléfonos Inteligentes

Los Smartphone o teléfonos inteligentes funcionan regidos por un sistema operativo móvil. Tienen el mismo cometido que los sistemas operativos de los computadores. Gestionan y regulan el funcionamiento del aparato, aunque de un modo más simple. Los más conocidos son Android, iOS y Windows Phone.

- Android está basado en Linux y es propiedad de Google. Teléfonos como los HTC(High Tech Computer) o los Samsung funcionan con este sistema operativo.
- iOS es propiedad de la empresa Apple y es exclusivo de los teléfonos iPhone y otros dispositivos de esta marca.
- Windows Phone es propiedad de la empresa Microsoft actualmente distribuido en los dispositivos de la marca Nokia. [5].

2.1.6 Sistema Operativo Android

¿Qué es Android? Android es un conjunto de herramientas de software para teléfonos móviles, creado por Google y la Open Handset Alliance. Está dentro de millones de teléfonos celulares y otros dispositivos móviles, lo que hace Android una plataforma importante para los desarrolladores de aplicaciones. Los primeros pasos con el desarrollo de aplicaciones Android es fácil. No se necesita tener acceso a un teléfono con Android, sólo un equipo en el que se puede instalar el SDK (Software Development Kit) de Android y el emulador de teléfono que viene con él. En cuestión de minutos, "Hello, Android" llevará a la creación de la primera aplicación de trabajo: la versión de Android de "Hello, World". A partir de ahí, se construirá un ejemplo más sustancial: un juego para Android Sudoku. Al añadir gradualmente características para el juego se aprenderá más acerca de muchos aspectos de la programación de Android, incluyendo interfaces de usuario, multimedia, y el ciclo de vida de Android. [6].

Aunque el sistema operativo Android tiene apenas unos años de vida, la librería Java de Android consiste en más de 150 paquetes (APIs(Application Programming Interface)), que contienen miles de clases, métodos, interfaces y constantes. Todas estas clases están documentadas en la página web de Android Developers. La vasta

extensión del software de desarrollo de Android (SDK) puede resultar extremadamente abrumadora para el que se acerca por primera vez a este sistema, especialmente si también es nuevo en el lenguaje de programación JAVA. [7].

2.1.7 Arquitectura Android

El siguiente grafico muestra la arquitectura de Android. Como se puede ver está formada por cuatro capas. Una de las características más importantes es de que todas las capas están basadas en software libre. [8].

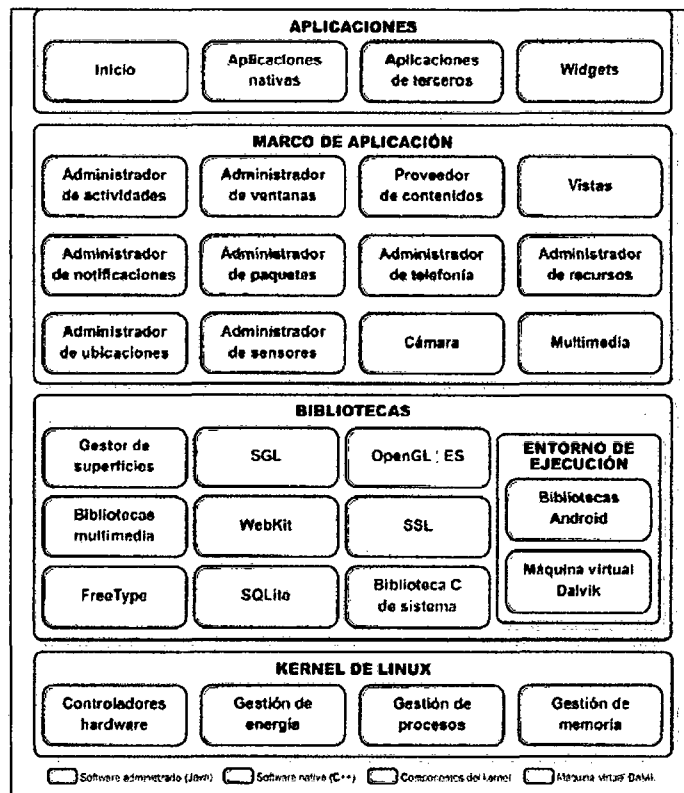


Figura 2-1: Arquitectura Android. [9].

¿Qué hace a Android especial?

1. Plataforma realmente abierta.
2. Portabilidad asegurada.
3. Arquitectura basada en componentes inspirados en Internet.
4. Filosofía de dispositivo siempre conectado a Internet.
5. Gran cantidad de servicios incorporados.
6. Alto nivel de seguridad.
7. Optimización para baja potencia y poca memoria.
8. Alta calidad de gráficos y sonido. [10].

2.1.8 Ciclo de Vida de una Aplicación Android

Creación de una actividad

Las actividades en Android son clases públicas que representan cada una de las pantallas de nuestra aplicación y que heredan de la clase base `android.app.Activity` mediante la palabra reservada `extends`.

Cuando se crea la actividad en el entorno Eclipse, se crea por defecto el método `onCreate()`. Toda actividad sobrescribe este método, que es el punto inicial del ciclo de vida de una actividad, y es invocado cuando la actividad se inicia. Además, la mayoría de las actividades también deben implementar el método `onPause()`,

donde se guardan datos y el estado antes de que la actividad deje de ser visible. El estado de la actividad se almacena en un objeto de la clase Bundle, que contiene el estado anterior de la actividad en caso de que haya sido suspendida.

Según el ciclo de vida de una actividad, después de iniciar una actividad se puede pausar o detener de forma momentánea si el usuario está realizando otras funciones en el teléfono. Si esto ocurre, la actividad deberá iniciarse nuevamente, y es entonces cuando la información de este Bundle es útil. [11].

Ciclo de vida de las actividades

Las aplicaciones Android funcionan con un ciclo de vida controlado por el propio sistema operativo. Este ciclo de vida depende de la situación en que se encuentre en cada momento la aplicación. Desde que se crea hasta que se destruye, sus actividades (Activity) pueden encontrarse en distintos estados. Se considera de interés para cualquier proyecto conocer los ciclos de vida, ya que deberemos programar los métodos necesarios para que nuestra aplicación funcione correctamente.

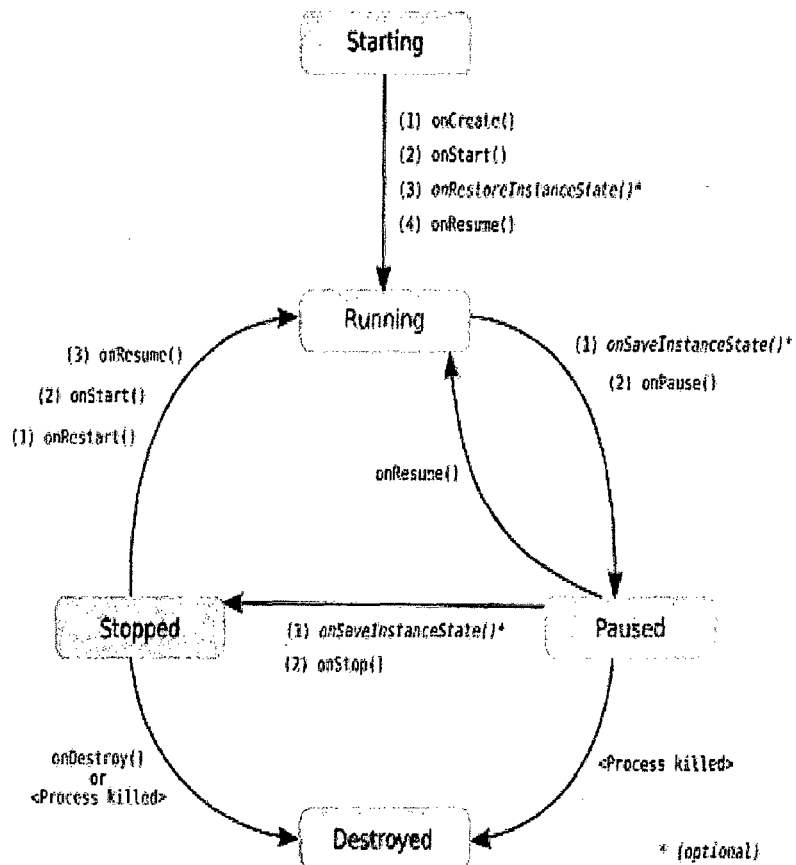


Figura 2-2: Ciclo de Vida de una aplicación android. [12].

- Activa (Running): Está la primera en la pila de ejecución, el usuario ve la actividad y puede interactuar con ella.
- Pausada (Paused): Ha pasado a segundo plano pero aún está visible porque otra actividad se coloca sobre ella pero no la tapa del todo. En este caso la actividad tapada puede ser cerrada por el sistema si necesita liberar recursos para la nueva actividad.

- Parada (Stopped): Ha pasado a segundo plano y está completamente tapada por la nueva actividad, en ese caso el sistema también puede optar por cerrarla si necesita liberar recursos.
- Destruída (Destroyed): Ya no está disponible, se han liberado todos sus recursos y en caso de ser llamada necesitaría comenzar un nuevo ciclo de vida.
- Cuando el estado de una Activity está cambiando es notificado por el sistema operativo, que hace una llamada a los métodos específicos de esa Activity. El siguiente diagrama ilustra los métodos y sus nombres:

Uso del contenedor Extras del objeto Intent para restaurar los datos guardados anteriormente o los datos que se pasa entre los Activity.

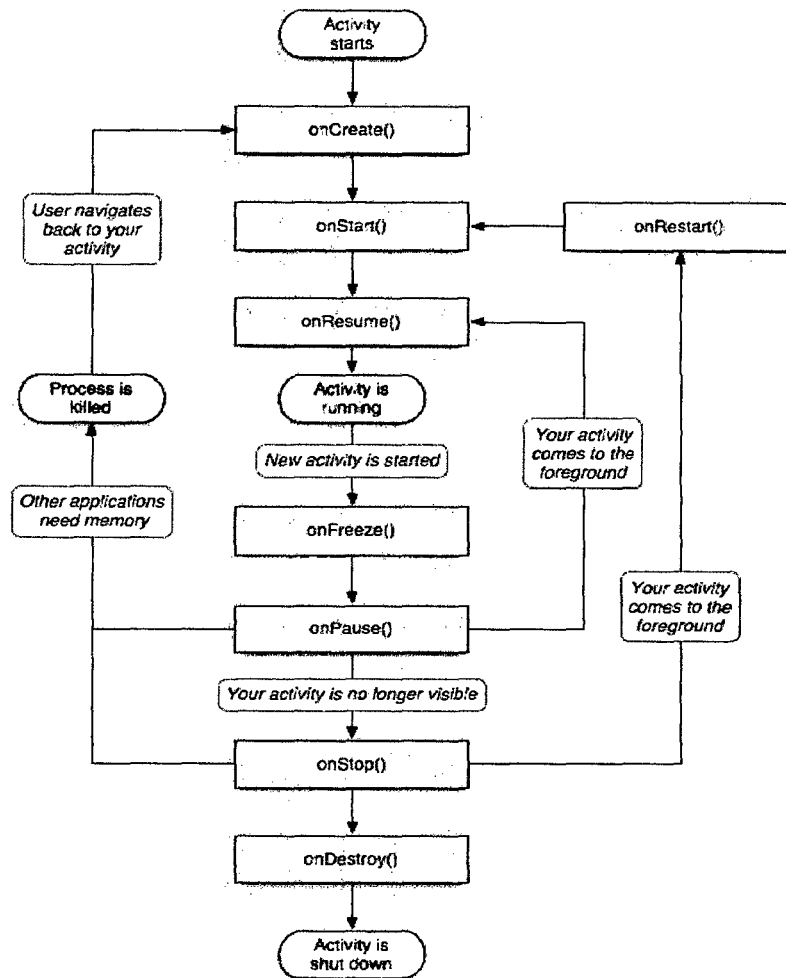


Figura 2-3: Métodos del ciclo de Vida de una aplicación android. [13].

onStart

Este método se llama cuando la Activity está a punto de ser visible para el usuario. Las Activities deben reemplazar este método si necesitan realizar cualquier tarea específica justo antes de que se hagan visibles, como por ejemplo refrescar valores actuales de los

valores de las vistas dentro de la actividad, o de puente para frame rates (una tarea común en la construcción de juegos).

onPause

Este método se llama cuando el sistema está a punto de poner la Activity en background. Las Activities deben reemplazar este método si necesitan confirmar los cambios no guardados a los datos persistentes, destruir o limpiar objetos para liberar los recursos que consumen, etc. Las implementaciones de este método deben finalizar lo antes posible, ya que ninguna Activity sucesiva se reanudará hasta que este método devuelve un valor.

Este método se llama cuando la Activity va a empezar a interactuar con el usuario después de estar en un estado de pausa. Cuando este método se llama, la Activity se está moviendo a la parte superior de la pila y está recibiendo actividad del usuario. Las Activities pueden reemplazar este método si lo necesitan para realizar cualquier tarea después de que la Activity haya aceptado la interacción del usuario.

onStop

Este método se llama cuando la Activity ya no es visible para el usuario, ya que otra Activity se ha reanudado o iniciado y está

eclipsando a esta. Esto puede suceder porque la Activity ha finalizado (el método de finalización fue llamado), porque el sistema está destruyendo a esta instancia de Activity para ahorrar recursos, o porque un cambio de orientación se ha producido en el dispositivo. Se puede distinguir entre estos escenarios mediante el uso de la propiedad `isFinishing`. Un Activity debe reemplazar este método si es necesario para realizar las tareas específicas antes de ser destruido, o si se trata de una interfaz de usuario para iniciar la reconstrucción después de un cambio de orientación.

onRestart

Este método se llama después de que una Activity se haya detenido, antes de que se inicie de nuevo. Este método siempre es seguido por `onStart`. Una Activity debe pasar por `onRestart` si es necesario para realizar las tareas inmediatamente antes de llamar a `onStart`. Por ejemplo, si la actividad previamente ha sido enviada al background y `onStop` ha sido llamada, pero el proceso de la Activity aún no ha sido destruido por el sistema operativo, entonces el método `onRestart` debe ser anulado.

Un buen ejemplo de esto sería cuando el usuario presiona el botón de inicio cuando aún había una Activity en la aplicación. El

onPause, y a continuación el onStop, son llamados, pero la actividad no se destruye. Si el usuario a continuación restaurara la aplicación mediante el administrador de tareas o una aplicación similar, el método onStart de la actividad sería llamado por el sistema operativo, durante la reactivación de actividades.

onDestroy

Este es el último método que se llama en una Activity antes de que sea destruido. Después de llamar a este método, su Activity se matará y será purgado de los grupos de recursos del dispositivo. El sistema operativo se destruirán permanentemente los datos de estado de una Activity después de que este método se haya ejecutado, por lo que una Activity debe reemplazar este método como medio final para salvar los datos del estado.

onSaveInstanceState

Este método es proporcionado por el ciclo de vida de Android para dar a una Activity la oportunidad de guardar los datos cuando se produce un cambio, por ejemplo un cambio de orientación de pantalla.

2.1.9 Interfaz Móvil.

Teléfonos móviles.

La navegación desde dispositivos móviles está en aumento continuamente. El limitado tamaño de la pantalla y la interacción táctil obliga a que se optimice tanto la información que se muestra en cada pantalla, como los mecanismos de interacción del usuario (botones, formularios de datos).

Apps IOS – Iphone

Los interfaces de Apps para iPhone, tanto basados en desarrollo nativo como en HTML5, mantienen pautas de diseño muy definidas y marcadas desde Apple, que consiguen que el usuario no requiera mucho aprendizaje a la hora de enfrentarse a una nueva aplicación.

Tablets Ipad, Android

En las interfaces para tablets se encuentra con un interfaz que en muchas ocasiones no necesita una especial limitación de tamaño para mostrar ciertas informaciones. Muchas interfaces diseñadas para escritorio, pueden proporcionar una experiencia de usuario satisfactoria. [14].

Internet

Internet es un conjunto descentralizado de redes de comunicación interconectadas que utilizan la familia de protocolos TCP/IP, garantizando que las redes físicas heterogéneas que la componen funcionen como una red lógica única, de alcance mundial. Sus orígenes se remontan a 1969, cuando se estableció la primera conexión de computadoras, conocida como ARPANET, entre tres universidades en California y una en Utah, Estados Unidos. [15].

2.1.10 Consumo de Datos

Servicios Web

Existen múltiples definiciones sobre lo que son los Servicios Web, lo que muestra su complejidad a la hora de dar una adecuada definición que englobe todo lo que son e implican. Una posible sería hablar de ellos como un conjunto de aplicaciones o de tecnologías con capacidad para operar en la Web. Estas aplicaciones o tecnologías intercambian datos entre sí con el objetivo de ofrecer unos servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la Web. [16].

¿Para qué sirven?

Estos servicios proporcionan mecanismos de comunicación estándares entre diferentes aplicaciones, que interactúan entre sí para presentar información dinámica al usuario. Para proporcionar interoperabilidad y extensibilidad entre estas aplicaciones, y que al mismo tiempo sea posible su combinación para realizar operaciones complejas, es necesaria una arquitectura de referencia estándar. [16].

¿Cómo funcionan?

El siguiente gráfico muestra cómo interactúa un conjunto de Servicios Web:

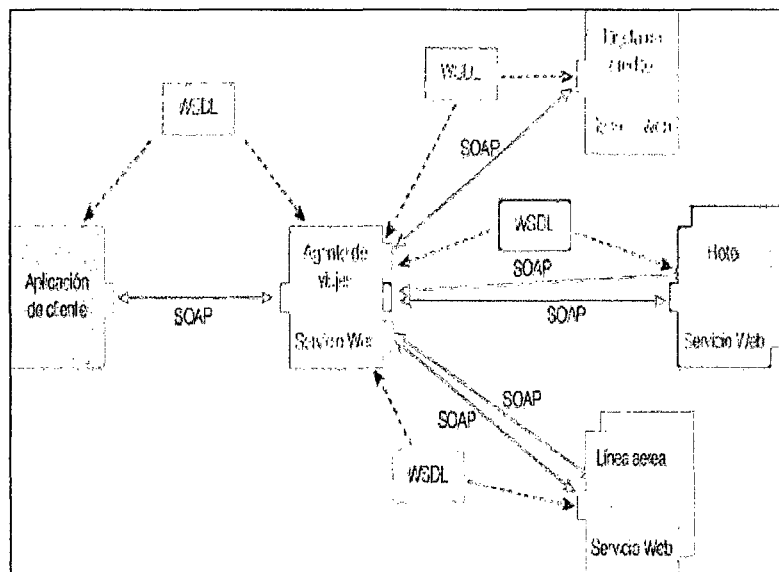


Figura 2-4: Servicios Web. [16].

2.1.11 Conexión de Datos Móviles

Wifi

Wifi es un conjunto de especificaciones para redes locales inalámbricas (WLAN - Wireless Local Area Network) basada en el standard IEEE 802.11.

El nombre Wifi es una abreviación del término inglés "Wireless Fidelity". Es común encontrar el termino Wifi escrito como Wi-Fi, Wi-fi o incluso wifi. Todas esas denominaciones se refieren a la misma tecnología. [17]

Ventajas Wifi

- Permite al usuario utilizar la red en cualquier punto dentro de los límites de alcance de la transmisión.
- Permite la incorporación rápida de otras computadoras y dispositivos en la red.
- Evita que las paredes sean taladradas o adaptadas para pasar los cables necesarios para conectarse a banda ancha por ejemplo. [17].

2.1.12 Seguridad en Aplicaciones Móviles

Los teléfonos móviles inteligentes o comúnmente llamados smartphones son cada día más completos ofreciendo una gran cantidad de características y funcionalidades, y con una tecnología que cada vez más se asemeja a un ordenador personal. Podemos pues, instalar en el dispositivo nuevos programas ampliando sus funcionalidades originales.

Sin embargo, este crecimiento también implica nuevos riesgos. Amenazas como malware en aplicaciones de terceros, o vulnerabilidades en los sistemas operativos pueden originar robo de contraseñas, o facturas telefónicas enormes por envíos ocultos de SMS Premium.

Con todo ello, CSIRT-cv decidió poner en marcha una nueva campaña de concienciación "Seguridad en aplicaciones móviles" - centrada, sobre todo, en aconsejar, a través de consejitos diarios, como evitar este tipo de riesgos. A continuación se resumen dichos consejos:

Cuando se instala una aplicación en un sistema Android, se debe comprobar qué permisos necesita antes de concedérselos. Se debe utilizar el sentido común y desconfiar ante cualquier cosa que resulte extraña. Por ejemplo, un juego no debería tener permisos

para mandar SMS o permisos para modificar la configuración del sistema. [18].

2.1.13 Metodología para aplicaciones móviles

Los teléfonos inteligentes y Smartphone son dispositivos de avanzada tecnología, que poseen una buena conectividad, donde se pueden instalar aplicaciones que son de suma utilidad para el usuario, los teléfonos de hoy en día, son considerados como una computadora ya que se pueden hacer operaciones similares a las de una PC, tales como navegar por internet, revisar documentos de office, revisar correos electrónicos, conectividad de WI-FI además poseen un teclado QWERTY o pantalla táctil y el sistema operativo interactúa con las aplicaciones y el microprocesador para darle un mayor rendimiento al dispositivo. [19].

2.1.14 Metodología Scrum

“Scrum es un marco ágil para el desarrollo de productos”. “The Scrum Manual” se exponen los principales conceptos de la metodología necesarios para entender el desarrollo del proyecto. [20].

2.2 Propuesta de Solución

Para las consultas académicas de los estudiantes de la Universidad Nacional Hermilio Valdizan, se establece una aplicación móvil utilizando la plataforma Android la cual se desarrollará de acuerdo a las normas técnicas que se ha establecido en la página web de la UNHEVAL, la aplicación contendrá los mismos beneficios de este sistema.

CAPITULO III: MARCO METODOLÓGICO

3.1 Tipo y diseño de investigación

El tipo de investigación es tecnológica o aplicada, debido a que se hace uso de teorías existentes para desarrollar el estudio.

El Diseño de la Investigación es No Experimental, ya que el método a aplicar es ensayo – error en la construcción del prototipo.

Método aplicado: Ensayo - error

3.2 Población y muestra

- Población: No aplicable por la naturaleza del estudio.
- Muestra: No aplicable por la naturaleza del estudio.

3.3 Técnicas de recojo de datos

- Fuentes:
 - **Primaria:** Reuniones y entrevistas con los usuarios del aplicativo móvil.
 - **Secundaria:** documentos, reportes.
- **Técnicas:** Análisis documentario, observación directa, etc.

3.4 Instrumentos de recolección de datos, y validación del instrumento

- **Instrumentos:** Entrevistas, modelos, etc.
- **Herramienta:**
 - a) Eclipse es un programa informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma. Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.
 - b) Android SDK (Software Development Kit) de Android, incluye un conjunto de herramientas de desarrollo. Comprende un depurador de código, biblioteca, un simulador de teléfono basado en QEMU.

3.5 Modalidad de la investigación:

Para el desarrollo de esta investigación de tesis se consideró utilizar una metodología en base a las características del proyecto con el fin de que esta nos ayude a gestionar su desarrollo. Se decidió adoptar la metodología Scrum, que es una metodología ágil para el desarrollo de proyectos, debido a que es flexible con respecto a los requisitos que pueden aumentar o cambiar con el tiempo, y al tamaño pequeño del equipo de desarrollo.

A su vez, el ciclo de vida del proyecto estará basado en el modelo incremental puesto que el concepto inicial del software, el análisis de las necesidades y el diseño de la arquitectura se define usando un enfoque de cascada, seguido por las iteraciones de prototipos, los que terminan en la instalación del prototipo final.

Una ventaja al usar este modelo es la resolución del problema en un tiempo corto, ya que sabemos que se tiene un tiempo límite (aproximadamente cuatro meses) para concluir con los requisitos exigidos, otra ventaja es una menor tasa de fallos y una visión de avance en el desarrollo desde las etapas iniciales del desarrollo.

Scrum es un proceso iterativo e incremental, por lo que el proyecto se divide en una serie de sprints, cada uno es de duración fija y durante este tiempo el equipo hace todo lo posible para tomar un pequeño conjunto de características de la idea a la funcionalidad de codificación y prueba. El presente proyecto se desarrolló en ocho sprints de dos semanas cada uno.

La metodología elegida está compuesta por cuatro actividades principales:

- Planeamiento del sprint: se eligen las funcionalidades a desarrollar durante el siguiente sprint.

- Scrum diario: el equipo se reúne todos los días por aproximadamente 15 minutos para compartir lo hecho el día anterior y lo que se hará el mismo día, también se debe compartir los problemas que han ocurrido o pueden ocurrir que pueda poner en riesgo el proyecto.
- Revisión del Sprint: el equipo demuestra al usuario final lo que se ha realizado durante el sprint.
- Retrospectiva del Sprint: el equipo busca la manera de mejorar el producto y el proceso.

Para complementar la metodología Scrum elegida para el desarrollo del proyecto se debe elegir la metodología de desarrollo del producto, que debe ser de la misma manera una metodología ágil. En este caso se ha seleccionado la metodología Extreme Programming (XP) como complemento.

Debido a la metodología seleccionada, se cumplieron las siguientes reglas:

- Escribir las historias de usuarios:
Se definieron las historias de usuarios en el product backlog de acuerdo a las metodologías Scrum y XP.
- Dividir el proyecto en iteraciones:
Se contó con 8 sprints de dos semanas cada uno.

- Realizar pequeños y frecuentes entregables:

Al final de cada sprint se entregó un entregable con las funcionalidades desarrolladas.

- Realizar reuniones breves todos los días:

CAPITULO IV: RESULTADOS Y DISCUSIÓN

En este capítulo se reflejan todos los resultados del proceso de desarrollo del análisis, diseño e implementación del aplicativo nativo de tecnología Android. A continuación se muestran los resultados del desarrollo de la propuesta.

La aplicación se desarrolló mezclando el servicio web y la aplicación móvil.

Uno puede hacer uso de manera óptima los recursos disponibles a nivel de servicio web, así como proporcionar funcionalidades aumentadas, mucho más allá de los disponibles mediante aplicación móvil.

4.1 Análisis

En el presente capítulo se plantea los requisitos funcionales de la solución y se desarrolla el análisis de la solución que consiste en explicar la viabilidad del sistema en términos técnicos y de costo.

4.1.1 Identificación de requerimientos (historias de usuario)

Se identificaron los requerimientos obtenidos luego del trabajo de levantamiento de información a los usuarios internos. Ver Anexo 01

4.1.2 Lista de requerimientos (Product Backlog)

En la tabla 4-1 se observa la lista de requerimientos y sus respectivas tareas.

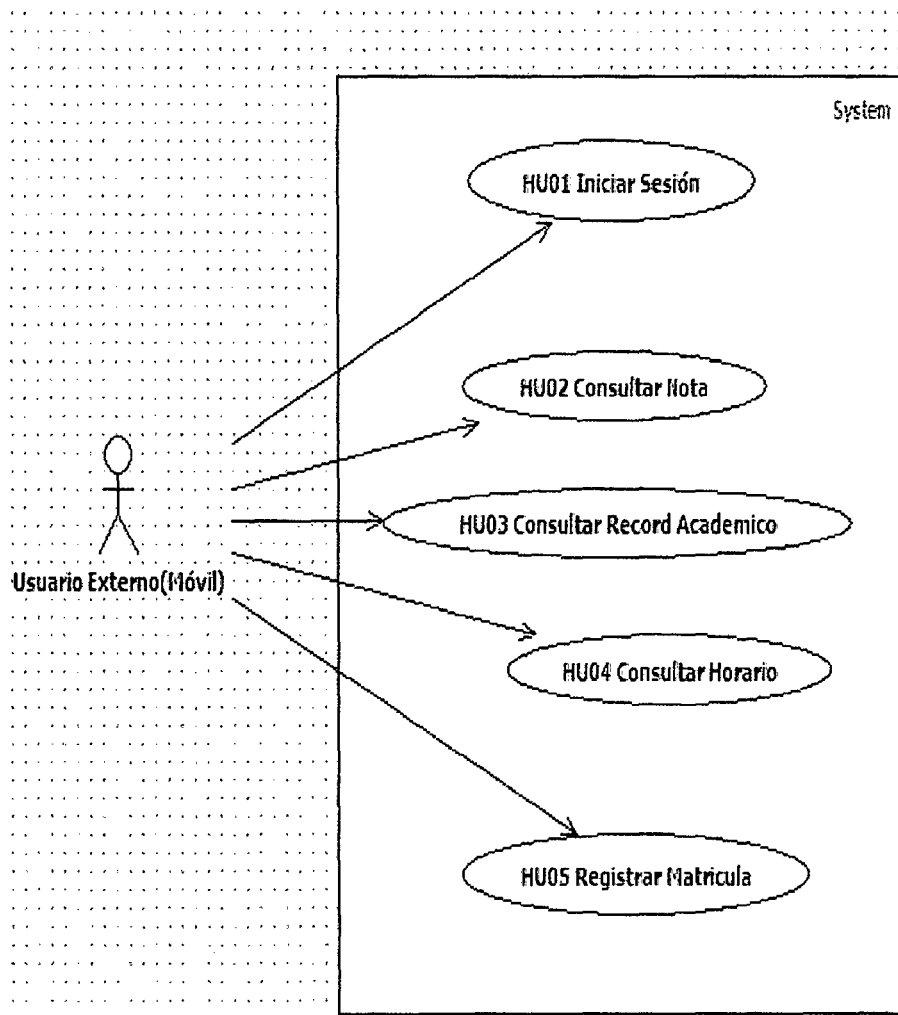
Nº	Descripción	Tareas
	Historia de usuarios	
1	Desarrollar Módulo de Login	Desarrollar Layout Interfaz Grafica
		Desarrollar funcionalidad
		Desarrollar web service Login
2	Desarrollo del Menú Principal	Desarrollar Layout Interfaz Grafica
		Desarrollar funcionalidad
3	Desarrollar Módulo Notas	Desarrollar Layout Interfaz Grafica
		Desarrollar funcionalidad
		Desarrollar web service Notas
		Desarrollar web service Detalle Nota
4	Desarrollar Módulo de Record Académico	Desarrollar Layout Interfaz Grafica
		Desarrollar funcionalidad
		Desarrollar web service Record Académico
5	Desarrollar Módulo Horario	Desarrollar Layout Interfaz Grafica
		Desarrollar funcionalidad
		Desarrollar web service Horario
6	Desarrollar Módulo Matricula	Desarrollar Layout Interfaz Grafica
		Desarrollar funcionalidad
		Desarrollar web service Matricula

Fuente: Elaboración propia

Tabla 4-1: Lista de Requerimientos

4.1.3 Identificación de historias de usuario

En la figura podemos observar los actores que interactúan directamente con el sistema junto con las historias de usuario identificadas.



Fuente: Elaboración propia

Figura 4-1: Historia de Usuario

4.1.4. Descripción de las historias de usuario más importantes

- **HU01 Iniciar Sesión**

Esta funcionalidad permite al usuario externo identificarse con el aplicativo mediante un usuario y contraseña para tener acceso a los módulos.

- **HU02 Consultar Nota**

Esta funcionalidad permite al usuario externo consultar sus notas del semestre en curso.

- **HU03 Record Académico**

Esta funcionalidad permite al usuario externo consultar sus notas de los semestres pasados como un historial de notas.

- **HU04 Consultar Horario.**

Esta funcionalidad permite al usuario externo consultar sus horarios y aulas de los cursos del semestre en curso.

- **HU05 Registrar Matricula**

Esta funcionalidad permite al usuario externo registrarse a los cursos que desea matricularse.

4.1.5 Análisis de la solución

El análisis de la solución se efectuará considerando los siguientes puntos de vista:

- Necesidades del cliente.
- Viabilidad del sistema.
- Análisis técnico – económico.
- Asignación de funciones.
- Restricciones tiempo.

4.1.6. Necesidades del cliente

Luego del análisis e investigación elaborada se ha identificado que el desarrollo de esta aplicación móvil es de gran utilidad, debido a que hoy en día el avance tecnológico en dispositivos móviles es muy grande, y con precios cada vez menores.

Este proyecto es de gran importancia debido a que ayudará a extender el servicio que ya presta la UNHEVAL para las consultas académicas de los estudiantes, brindando un adecuado manejo desde cualquier parte con solo acceder a la aplicación, además se adaptará a cualquier teléfono móvil que soporte la plataforma android. Esta aplicación se verá novedosa, debido a que muchos usuarios que posean esta tecnología podrán acceder al reporte de sus notas desde su teléfono móvil, con una facilidad y comodidad que ofrece la tecnología.

4.1.7. Viabilidad del sistema

Los requerimientos expuestos en líneas anteriores pueden satisfacerse mediante la elaboración de un aplicativo móvil que sea amigable, fácil de manejar e interactúe con una base de datos confiable que pueda mantener la información disponible, íntegra y segura en todo momento.

El sistema planteado refleja la información recogida y procedimientos necesarios para cumplir con los servicios que brindará permitiendo de esta manera hacer una mejora a estos para cumplir con la necesidad de agilizarlos y gestionarlos de una manera más óptima.

4.1.8. Análisis técnico – económico

- **Análisis técnico**

El sistema que se propone en este proyecto de tesis es un aplicativo móvil, ya que se cuenta con la necesidad de extender el servicio mediante los dispositivos inteligentes.

El equipo responsable de la elaboración de esta aplicación está conformado por:

2 Analistas/desarrolladores: Los tesisistas.

1 Consultor: El asesor de tesis.

Los clientes finales: Usuarios internos (trabajadores de informática y la FIIS) y usuarios externos (los estudiantes).

Con respecto a la elaboración se utilizará el IDE Eclipse y el servidor Apache. Asimismo se hará uso del lenguaje de programación PHP y Java.

También será necesario que la institución cuente con servidores relativamente rápidos y con conexión a internet.

- **Análisis económico**

Debido a que el IDE y los lenguajes a utilizar son de libre distribución, para el análisis económico habría que considerar solo el costo que produce el tiempo de la construcción del proyecto y el costo de los recursos. Ver Anexo 04

4.1.9. Asignación de funciones

En la tabla se muestra los requerimientos y funciones asignados a cada elemento involucrado en el sistema.

Elemento	Función
Software	<p>Para la implementación y las Pruebas se necesitara el servidor de aplicaciones web Apache para alojar los scripts PHP.</p> <p>Conexión a internet y una IP publica para la salida y acceso de los servicios del aplicativo.</p>
Hardware	<p>Para el desarrollo del proyecto se necesitara trabajar con una laptop con mínimo 4gb de memoria RAM y conexión a internet.</p>
Recursos	<p>El sistema se desarrolló por dos analistas programadores. Serán encargados del análisis, diseño e implementación del aplicativo.</p>

Fuente: Elaboración propia

Tabla 4-2: Asignación de funciones.

4.2 DISEÑO

Este capítulo describe la arquitectura seleccionada para cumplir con los requisitos planteados del sistema.

Se presenta también en este capítulo el diseño de la interfaz gráfica de la aplicación, definiendo los criterios para la selección del diseño y las principales pantallas de la aplicación con sus respectivas características.

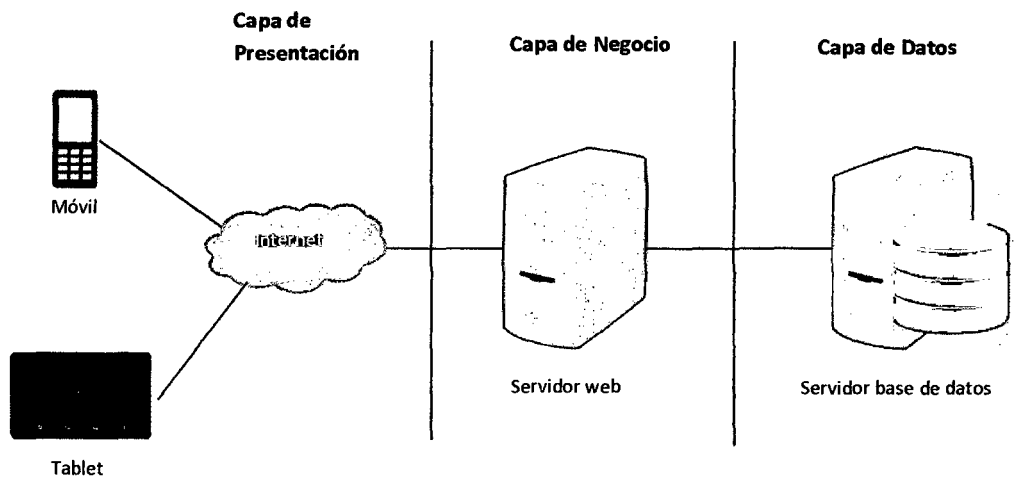
4.2.1 Arquitectura de la solución

La aplicación que se presenta en este trabajo seguirá una arquitectura de 3 capas: capa de datos, capa de negocio y capa de presentación.

Las ventajas de esta arquitectura son la mantenibilidad, escalabilidad y flexibilidad. Cada capa puede ser independiente de otra capa y permite que se puedan hacer cambios sin tener que afectar a toda la aplicación como un todo. Esto también permite por ejemplo dividir el sistema en proyecto y asignarlos a diferentes desarrolladores o equipos de desarrolladores.

La separación de las partes más importantes del sistema brinda flexibilidad a la solución para mejor mantenimiento y posteriores mejoras [21].

En la figura 4.2 se muestra el diagrama de Arquitectura, con la distribución de las capas que han sido contempladas.



Fuente: Elaboración propia

Figura 4-2: Diagrama de arquitectura en 3 Capas

Capa de Presentación

Se denomina también la capa del usuario pues es la que el sistema presenta al usuario, de igual manera el usuario interactúa con el sistema por medio de esta capa, enviando información requerida para la acción del sistema. En el caso del presente trabajo, esta capa está representada por la aplicación nativa en el dispositivo móvil con sistema operativo Android.

Para la comunicación con la capa de negocio, se hará uso de Rest (Representational State Transfer) y JSON (JavaScript Object Notation). De tal manera que si luego por ejemplo se desea hacer

la aplicación para iPhone, se tenga que construir sólo la capa de presentación y no se necesite ninguna mayor modificación.

Capa de Negocio

Esta es la capa más importante del sistema pues aquí es donde se manejan las reglas del negocio y la lógica que tiene el sistema.

Para la implementación se hará uso del lenguaje de programación PHP, los script estarán alojados en el Servidor Web.

La Web Service en esta capa que recibirá peticiones por el formulario del aplicativo usando REST (Representational State Transfer) y que devuelva contenido en JSON para una mejor estandarización.

Capa de Datos

La capa de datos, en donde residen los datos y es encargada de la obtención de los mismos. Para el caso del presente trabajo, esta capa consiste en un Sistema Gestor de Bases de Datos SQLSERVER, el cual tendrá la base de datos relacional.

Beneficios de la arquitectura:

- La lógica de la aplicación reside en la capa media lo que permite tener un cliente más liviano de la aplicación.
- Se puede actualizar o realizar cambios a la base de datos sin tener que afectar a las otras capas.
- Se puede agregar nuevos clientes en la capa de presentación como por ejemplo dispositivos móviles con otros sistemas operativos desarrollando sólo esta capa sin modificar las otras.
- Permite mantener actualizar los componentes de manera más sencilla sin tener que modificar todo el sistema.
- Permite distribuir el trabajo de creación por diferentes capas sólo conociendo la forma de comunicación de las mismas.

4.2.2. Diseño de la interfaz gráfica

En esta sección se presentan los detalles generales del diseño de la interfaz gráfica de la aplicación. Asimismo se presentan las interfaces de las principales pantallas del sistema indicando sus características.

Criterios para el diseño de la interfaz gráfica

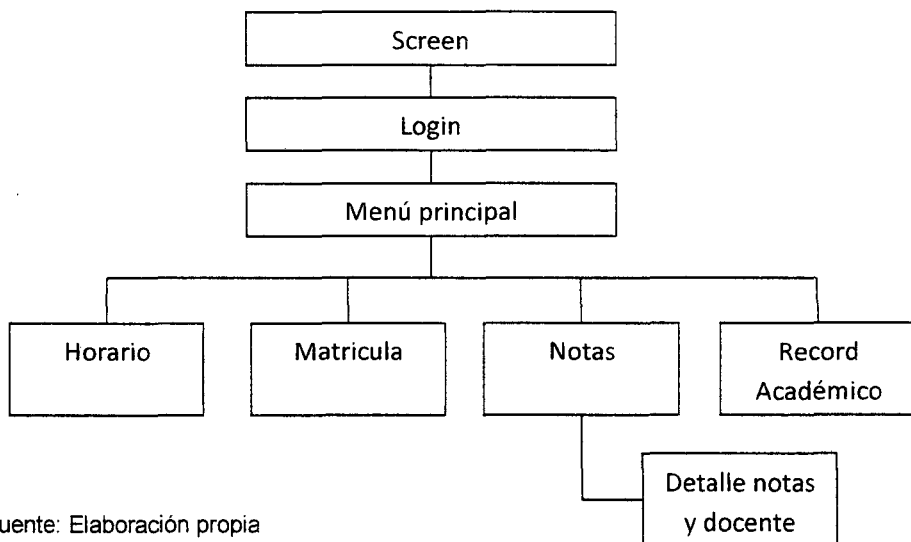
Con el objetivo de uniformizar la interfaz gráfica de la aplicación y hacerlo más intuitivo o fácil de manejar para el usuario final, se

mencionarán las consideraciones que se deben tomar en cuenta en al momento de construir el software. [22].

- Se usará colores que no cansen la vista del usuario.
- Se hará uso de gráficos que reflejen acciones determinadas, en los botones y fondos principales.
- Se presentarán las opciones al usuario de una manera que sea fácil para el usuario usar las herramientas de la aplicación.
- Se presentará una manera fácil para que el usuario pueda salir intuitivamente de la aplicación.

4.2.3 Modelo de Ventanas

En la figura 4.3 se muestra la navegación de la aplicación móvil.



Fuente: Elaboración propia

Figura 4-3: Navegación de la aplicación móvil

En las siguientes figuras de esta sección se muestran algunos ejemplos de los tipos de contenidos de las pantallas que siguen el esquema de trabajo mencionado.

Desde la figura 4.4 hasta la figura 4.8 se presentan las pantallas de la aplicación móvil y las imágenes restantes pertenecen al web service.

Pantalla de Login

En la figura 4.4 se presenta una pantalla de inicio de sesión. Con dos cajas de texto para ingresar el usuario y contraseña.

Escudo de la unheval

Usuario:

Contraseña:

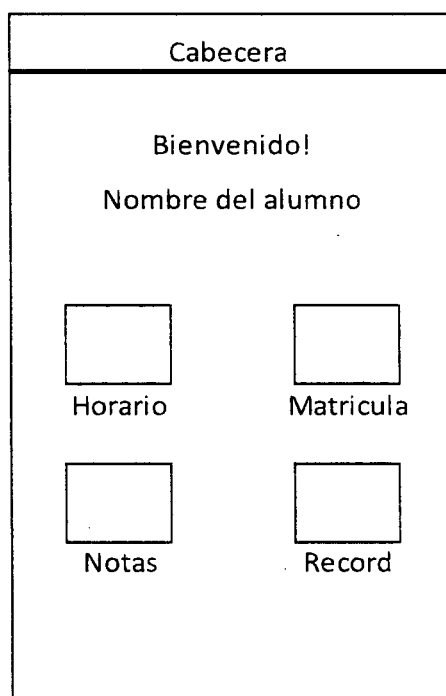
Ingresar

Fuente: Elaboración propia

Figura 4-4: Pantalla de Inicio.

Pantalla de Menú académico

En la figura 4.5 se presenta las opciones del sistema que son: Matricula de cursos, Consulta de Notas, Record Académico y Horario.



Fuente: Elaboración propia

Figura 4-5: Pantalla de menú académico.

Pantalla de Listado de Notas

En la figura 4.6 se presenta un listado con las notas del alumno del ciclo actual al cual se matriculo.

Notas Actuales	
CURSO	NOTA
Curso 1	10
Curso 2	10
Curso 3	10
Curso 4	10
Curso 5	10
Curso 6	10
Curso N	10

Fuente: Elaboración propia

Figura 4-6: Pantalla de Notas.

Pantalla de Horarios

En la figura 4.7 se presenta un listado de los días que el alumno tiene clases el cual contiene el nombre del cursos, hora y salón.

Horario	
Lunes	↓
Martes	↓
Curso 1 9:00 - 11:00 Aula 404	
Miercoles	↓
Jueves	↓
Viernes	↓

Fuente: Elaboración propia

Figura 4-7: Pantalla de Horario.

Pantalla de Matricula

En la figura 4.8 se presenta un listado de cursos para que el alumno pueda matricularse.

Matricula Cursos		
	CURSO	CRED
<input type="checkbox"/>	Curso 1	3
<input type="checkbox"/>	Curso 2	4
<input type="checkbox"/>	Curso 3	3
<input type="checkbox"/>	Curso 4	4
<input type="checkbox"/>	Curso 5	3
<input type="checkbox"/>	Curso 6	4
<input type="button" value="Aceptar"/>		

Fuente: Elaboración propia

Figura 4-8: Pantalla de Matricula.

4.3 CONSTRUCCION

En este capítulo se detalla cómo se utilizaron las tecnologías mencionadas en el primer capítulo para la construcción del producto final.

Adicionalmente se plantea la estrategia de pruebas que se siguió en el producto para la validación y verificación correcta. La estrategia contiene los tipos de pruebas a utilizar y el catálogo de pruebas.

4.3.1 Construcción de la Aplicación Móvil

La aplicación Móvil se desarrolló usando como entorno de desarrollo Eclipse junto con el plugin Android Development Tools diseñado específicamente para el desarrollo de aplicaciones Android.

Se tuvo en cuenta el desarrollo por capas, separando de esta forma la parte del modelo, con la lógica del negocio y el controlador. Además se manejaron servicios REST para el envío y recepción de información hacia el backend. Se usó Json como modelado de datos entre el backend y la aplicación Móvil.

Para el desarrollo de los módulos de consultas Notas, Horario y Record académico y el módulo de matrícula se trabajó con los controladores que brinda el SDK de Android.

4.3.2 Construcción de la Web Service

La Web Service se desarrolló con el lenguaje de programación PHP. Esto nos permite la conexión y comunicación entre el aplicativo móvil y la base de datos. La comunicación es mediante servicios REST. Antes de la construcción del aplicativo móvil es recomendable tener primero el formato de datos que arroja la Web Service.

4.3.3 Pruebas

En esta sección del documento se presenta los tipos de pruebas a realizar, el catálogo de las principales pruebas de la aplicación y el reporte de las pruebas ejecutadas.

Tipos de Pruebas

Los tipos de pruebas que se realizó fueron los siguientes:

Pruebas Unitarias

Consiste en probar cada componente del sistema de manera individual para corroborar su correcto funcionamiento. Estas pruebas se aplicarán por ejemplo a las pantallas de la aplicación móvil para validar campos obligatorios y validaciones que se tengan que tomar en cuenta en el momento del desarrollo.

Pruebas de integración

Las pruebas de integración sirven para determinar si los diferentes componentes integrados en el sistema funcionan de manera correcta. Puede darse el caso de que un componente funcione muy bien por sí sólo pero integrado cause algunos problemas por lo que estas pruebas de integración son importantes para poder determinar la calidad de la solución.

Catálogos de Pruebas

A continuación se presentan los casos de prueba funcionales y de interfaz de usuario más relevantes para el sistema. [23]. Ver Anexo 02.

CONCLUSIONES

Las conclusiones que se pueden obtener del proyecto son las siguientes:

- Se desarrolló e implemento el aplicativo móvil, que brinda a los estudiantes de la UNHEVAL una novedosa e innovadora alternativa para realizar sus consultas académicas.
- Se realizó el análisis de los requerimientos mínimos de hardware y software para el desarrollo e implementación del aplicativo móvil nativa Android, y ofrecer un óptimo servicio a la comunidad valdizana.
- Se realizaron las actividades que son parte de la metodología seleccionada para la gestión del proyecto como por ejemplo las reuniones de sprint, los demo al final de cada sprint y el seguimiento del plan de proyecto.
- Existen varias herramientas con las cuales se pueden desarrollar aplicaciones para dispositivos móviles, sin embargo se ha optado por utilizar el Eclipse IDE siendo esta una herramienta de código libre, además se utilizó para el desarrollo del servicio web PHP y como servidor Apache, es así como se empezó el desarrollo de la aplicación móvil.
- Se realizaron las pruebas necesarias para asegurar que el aplicativo móvil, verdaderamente, cumple el objetivo para el cual fue planteado.

RECOMENDACIONES

Las recomendaciones que se pueden brindar son las siguientes:

- Se recomienda agregar la funcionalidad de notificaciones PUSH, esto permite que cualquier actualización sea comunicado en tiempo real a los dispositivos que tengan instalados el aplicativo, ya que es de código abierto.
- Desarrollar el cliente para móviles iPhone que en la actualidad también posee una gran cantidad de usuarios que necesitaran esta aplicación.
- Se recomienda publicar el aplicativo en Play Store, ya que los estudiantes podrán descargarlo e instalarlo directamente a sus dispositivos móviles, porque todo dispositivo Android tiene instalado por defecto el Play Store.
- Finalmente se propone que se desarrollen aplicaciones similares pero orientado a otros tipos de usuarios; que sean de gran beneficio, tomando en cuenta la factibilidad, adaptabilidad y funcionamiento para la cual este destinada la aplicación, como por ejemplo docente, administrativos, etc.

BIBLIOGRAFÍA

- [1] Tomas Girones, Jesús, El gran libro de Android, Editorial: MARCOMBO, S.A. 2º Edición, mayo de 2012, pp. 322.
- [2] Aplicaciones para dispositivos Móviles. [Online]. Disponible en: http://www.buap.mx/porta1_pprd/wb/cidd/las_aplicaciones_en_dispositivos_moviles.
- [3] Ing. Jorge Nolasco Valenzuela, Desarrollo de aplicaciones móviles Android y J2ME. Macro E.I.R.L: Lima- Perú, 2010, pp. 15.
- [4] Dispositivos móviles. [Online]. Disponible en: <http://e-archivo.uc3m.es:8080/handle/10016/6506>
- [5] Teléfonos Inteligentes. [Online]. Disponible en: <http://www.fotonostra.com/digital/smartphones.htm>
- [6] Android. [Online]. Disponible en: <http://dl.acm.org/citation.cfm?id=1816808>
- [7] José Enrique Amaro Soriano, Android-programacion de dispositivos móviles a través de ejemplos. AlfaOmega S.A de CV: MéxicoD.F, 2011, pp. 11.
- [8] Tomas Girones, Jesús, El gran libro de Android, Editorial: MARCOMBO, S.A. 2º Edición, mayo de 2012, pp. 22.
- [9] Arquitectura Android [Online]. Disponible en : wordpress.com/2011/02/17/arquitectura-de-android/
- [10] Tomas Girones, Jesús, El gran libro de Android, Editorial: MARCOMBO, S.A. 2º Edición, mayo de 2012, pp. 18,19.
- [11] <http://www.proyectosimio.com/es/programacion-android-ciclo-de-vida-de-una-app/>
- [12] <http://www.proyectosimio.com/es/programacion-android-ciclo-de-vida-de-una-app/>

- [13] <http://www.proyectosimio.com/es/programacion-android-ciclo-de-vida-de-una-app/>
- [14] Diseño de Interfaz móvil. [Online]. Disponible en:
<http://hacce.com/servicios/disenio-de-interfaz/>
- [15] Internet [Online]. Disponible en:
<http://aleja11a.wordpress.com/tag/internet-es-un-conjunto-descentralizado-de-redes-de-comunicacion-interconectadas/>
- [16] Servicios Web [Online] Disponible en:
<http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>
- [17] Wifi [Online] Disponible en:
<http://www.informatica-hoy.com.ar/aprender-informatica/Wifi.php>
- [18] Seguridad Aplicaciones Mviles. [Online] Disponible en:
<http://www.csirtcv.gva.es/es/paginas/seguridad-en-aplicaciones-%C3%B3viles.html>
- [19] Metodología para aplicaciones móviles [Online] Disponible en:
<http://histinf.blogs.upv.es/2012/12/03/smartphones/>
- [20] Scrum [Online]. Disponible en:
<http://www.softeng.es/es-es/metodologia-scrum.html>
- [21] Arquitectura de Soluciones
Disponible en: <http://docs.oracle.com/cd/E19528-01/820-0888/6ncjkpnh6/index.html>
- [22] Android User Interface Guidelines
Disponible en:
http://developer.android.com/guide/practices/ui_guidelines/index.html
- [23] Tania Denisse Gonzales Villa (2013), *Análisis e implementación de un sistema móvil para gestión de los servicios de atención que brindan las comisarías a la comunidad*. Tesis para optar grado académico de Ingeniero Informático, PUCP.

ANEXOS

Anexo 01

Número: 001 Usuario: Product Owner	
Nombre historia: GUI login.	
Prioridad en negocio: Alta	Creación – 06/04/2015
Programador responsable: Heinz Vara Arosemena	
Descripción: El acceso debe de ser por medio de usuario (código de estudiante) y contraseña con la que se accede a la intranet de la Universidad	
Observaciones: <i>CONFIRMADO con el Product Owner.</i>	

Tabla 1: Historia de Usuario 001.

Número: 002 Usuario: Product Owner	
Nombre historia: GUI menú principal.	
Prioridad en negocio: Media	Creación – 06/04/2015
Programador responsable: Yhon Ponciano Tolentino	
Descripción: En el menú principal debe contener iconos de acceso a las consultas de notas, record, horario y matrícula	
Observaciones: <i>CONFIRMADO con el Product Owner.</i>	

Tabla 2: Historia de Usuario 002.

Número: 003 Usuario: Product Owner	
Nombre historia: Consulta Notas Actuales.	
Prioridad en negocio: Alta	Creación – 06/04/2015
Programador responsable: Heinz Vara Arosemena / Yhon Ponciano Tolentino	
Descripción: Los estudiantes deben realizar la consulta de notas de los cursos que están llevando en el ciclo actual	
Observaciones: <i>CONFIRMADO con el Product Owner.</i>	

Tabla 3: Historia de Usuario 003.

Número: 004 Usuario: Product Owner	
Nombre historia: Consulta record Academico.	
Prioridad en negocio: Alta	Creación – 06/04/2015
Programador responsable: Heinz Vara Arosemena / Yhon Ponciano Tolentino	
Descripción: Los estudiantes deben revisar los reportes históricos obtenidos durante toda su carrera universitaria o durante el ciclo académico alcanzado hasta el momento	
Observaciones: <i>CONFIRMADO con el Product Owner.</i>	

Tabla 4: Historia de Usuario 004.

Numero: 005 Usuario: Product Owner	
Nombre historia: Visualización de horarios.	
Prioridad en negocio: Alta	Creación - 06/04/2015
Programador responsable: Heinz Vara Arosemena / Yhon Ponciano Tolentino	
Descripción: Los estudiantes deben visualizar los horarios asignados durante presente ciclo o semestre académico.	
Observaciones: <i>CONFIRMADO con el Product Owner.</i>	

Tabla 5: Historia de Usuario 005.

Numero: 006 Usuario: Product Owner	
Nombre historia: Selección de Matricula.	
Prioridad en negocio: Alta	Creación - 06/04/2015
Programador responsable: Heinz Vara Arosemena / Yhon Ponciano Tolentino	
Descripción: Los estudiantes deben visualizar los cursos disponibles del ciclo académico a iniciarse para poder matricularse.	
Observaciones: <i>CONFIRMADO con el Product Owner.</i>	

Tabla 6: Historia de Usuario 006.

Número: 002 Usuario: Product Owner	
Nombre historia: GUI menú principal.	
Prioridad en negocio: Baja	Modificado – 17/04/2015
Programador responsable: Yhon Ponciano Tolentino	
Descripción: En el layout del menú principal debe contener íconos de acceso a las consultas de notas, record, horario, matrícula y además debe visualizarse los nombres y apellidos del estudiante	
Observaciones: <i>CONFIRMADO con el Product Owner.</i>	

Tabla 6: Modificación Historia de Usuario 002

Número: 003 Usuario: Product Owner	
Nombre historia: Consulta Notas Actuales.	
Prioridad en negocio: Alta	Modificado – 12/05/2015
Programador responsable: Heinz Vara Arosemena	
Descripción: Los estudiantes deben realizar la consulta de notas de los cursos que están llevando en el ciclo actual y además se debe visualizar en la parte superior el ciclo actual.	
Observaciones: <i>CONFIRMADO con el Product Owner.</i>	

Tabla 7: Modificación Historia de Usuario 003

Anexo 02

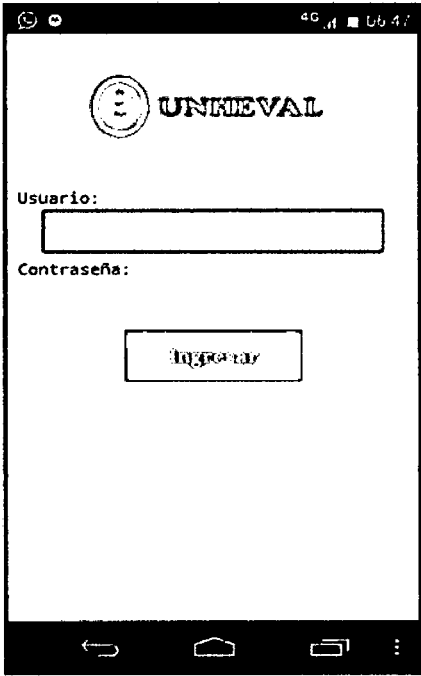
Prueba	01
Descripción	Verificar la interfaz gráfica de la página de inicio sea la correcta
Acciones	1. Tab en el ícono de la aplicación "UnhevalMóvil" del menú principal del dispositivo
Resultado esperado	Después del paso 1, la página de inicio debe mostrar lo siguiente: <ul style="list-style-type: none"> - Usuario - Contraseña - Botón ingresar
Resultado visual	
Fallo	Cuando no se cumpla con la condición anterior

Tabla 1: Caso de prueba 01.

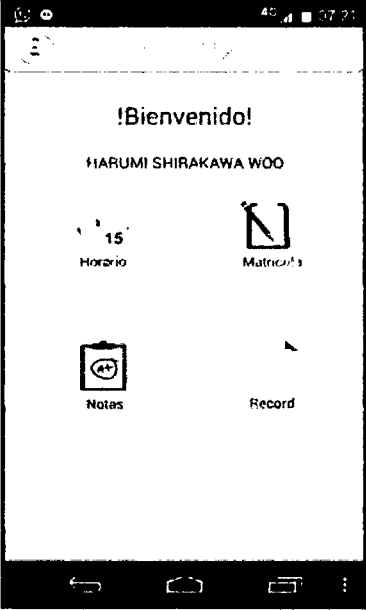
Prueba	02
Descripción	Verificar la interfaz gráfica de la página menú sea la correcta
Acciones	1. Tab en el ícono "UnhevalMóvil". 2. Ingresar datos. 3. Tab en el ícono "Ingresar".
Resultado esperado	Después del paso 3, la pantalla menú debe mostrar lo siguiente: <ul style="list-style-type: none"> - Nombre del alumno - Horario - Matrícula - Notas - Record
Resultado visual	
Fallo	Cuando no se cumpla con las condiciones anteriores

Tabla 2: Caso de prueba 02.

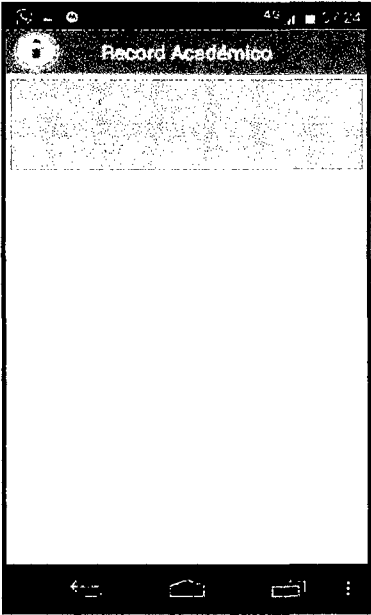
Prueba	03
Descripción	Verificar que se muestre el listado de ciclos cursados cuando el usuario haga tab en el ícono "Record"
Acciones	<ol style="list-style-type: none"> 1. Tab en el ícono "UnhevalMóvil". 2. Ingresar datos. 3. Tab en el ícono "Ingresar". 4. Tab en el ícono "Record".
Resultado esperado	Después del paso 4, se debe mostrar: - Listado de los ciclos académicos cursados.
Resultado visual	
Fallo	Cuando no se cumpla con las condiciones anteriores

Tabla 3: Caso de prueba 03.

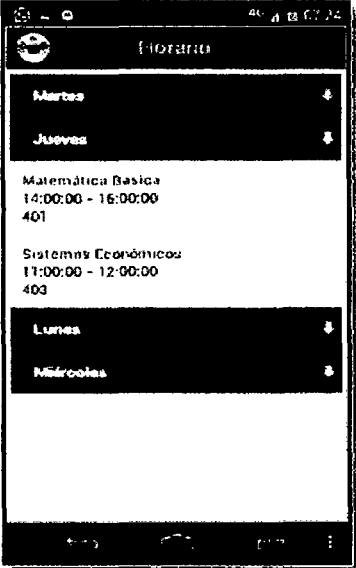
Prueba	04
Descripción	Verificar que se muestre la hora, curso y aula cuando el usuario haga tab en el ícono del día.
Acciones	<ol style="list-style-type: none"> 1. Tab en el ícono "UnhevalMóvil". 2. Ingresar datos. 3. Tab en el ícono "Ingresar". 4. Tab en el ícono "Horario". 5. Tab en el ícono del día que desea ver.
Resultado esperado	<p>Después del paso 5, se muestra lo siguiente:</p> <ul style="list-style-type: none"> - Hora - Curso - Aula
Resultado visual	 <p>The screenshot shows a mobile application interface titled "Horario". It displays a schedule for two days: Tuesday and Thursday. Under Tuesday, there are two entries: "Matemática Basica" with a time of 14:00:00 - 16:00:00 and room 401, and "Sistemas Económicos" with a time of 11:00:00 - 12:00:00 and room 403. Under Thursday, the entry "Matemática Basica" is visible. Each day's section has a dropdown arrow on the right. The interface is in Spanish and appears to be a mobile app for a university.</p>
Fallo	Cuando no se cumpla con las condiciones anteriores

Tabla 4: Caso de prueba 04.

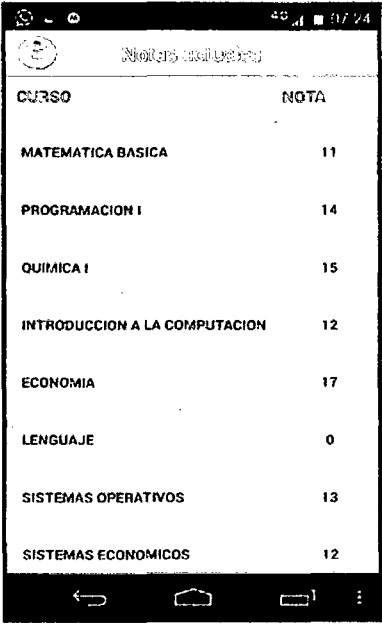
Prueba	05																		
Descripción	Verificar que se muestre el listado de cursos y notas cuando el usuario haga tab en el ícono "Notas"																		
Acciones	<ol style="list-style-type: none"> 1. Tab en el ícono "UnhevalMóvil". 2. Ingresar datos. 3. Tab en el ícono "Ingresar". 4. Tab en el ícono "Notas". 																		
Resultado esperado	Después del paso 4, se muestra el listado de cursos y notas actuales.																		
Resultado visual	 <table border="1"> <thead> <tr> <th>CURSO</th> <th>NOTA</th> </tr> </thead> <tbody> <tr> <td>MATEMATICA BASICA</td> <td>11</td> </tr> <tr> <td>PROGRAMACION I</td> <td>14</td> </tr> <tr> <td>QUIMICA I</td> <td>15</td> </tr> <tr> <td>INTRODUCCION A LA COMPUTACION</td> <td>12</td> </tr> <tr> <td>ECONOMIA</td> <td>17</td> </tr> <tr> <td>LENGUAJE</td> <td>0</td> </tr> <tr> <td>SISTEMAS OPERATIVOS</td> <td>13</td> </tr> <tr> <td>SISTEMAS ECONOMICOS</td> <td>12</td> </tr> </tbody> </table>	CURSO	NOTA	MATEMATICA BASICA	11	PROGRAMACION I	14	QUIMICA I	15	INTRODUCCION A LA COMPUTACION	12	ECONOMIA	17	LENGUAJE	0	SISTEMAS OPERATIVOS	13	SISTEMAS ECONOMICOS	12
CURSO	NOTA																		
MATEMATICA BASICA	11																		
PROGRAMACION I	14																		
QUIMICA I	15																		
INTRODUCCION A LA COMPUTACION	12																		
ECONOMIA	17																		
LENGUAJE	0																		
SISTEMAS OPERATIVOS	13																		
SISTEMAS ECONOMICOS	12																		
Fallo	Cuando no se cumpla con las condiciones anteriores																		

Tabla 5: Caso de prueba 05.

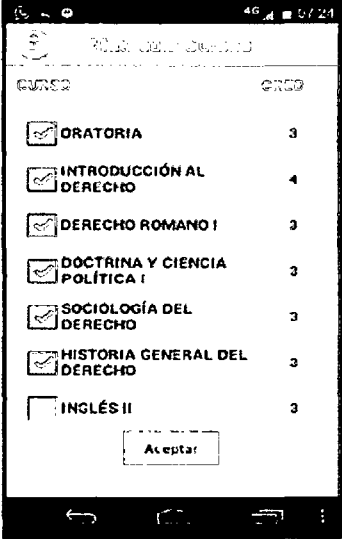
Prueba	06
Descripción	Verificar que se muestre el listado de cursos disponibles cuando el usuario haga tab en el ícono "Matricula"
Acciones	1. Tab en el ícono "UnhevalMóvil". 2. Ingresar datos. 3. Tab en el ícono "Ingresar". 4. Tab en el ícono "Matricula".
Resultado esperado	Después del paso 4, se muestra los cursos disponibles para matricularse.
Resultado visual	
Fallo	Cuando no se cumpla con las condiciones anteriores

Tabla 6: Caso de prueba 06.

Anexo 03

1. Servicios del sistema Back-end

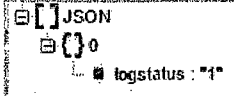
Servicio	SB1 - Login Usuario
Descripción	Servicio que verifica las credenciales de un usuario y retorna información del usuario más las opciones que tiene disponible en el aplicativo móvil.
Método	POST
Entrada (JSON)	<code>{"user":{"usuario":"2006110231","password":"62888b535facf3d5593edd"}}</code>
Salida (JSON)	<code>[{"logstatus":"1"}]</code>
Imagen Salida	

Tabla 1: Servicio Login


Servicio	SB2 - Menú Académico
Descripción	Servicio que muestra los nombres y apellidos del usuario más las opciones del menú
Método	POST
Entrada (JSON)	<code>{"user":{"usuario":"2006110231"}}</code>
Salida (JSON)	<code>[{"Nombre":"YHON ELADIO PONCIANO TOLENTINO"}]</code>
Imagen Salida	

Tabla 2: Servicio Académico

Servicio	SB3 - Lista de Notas
Descripción	Servicio que devuelve una lista de notas de los cursos del alumno
Método	POST
Entrada (JSON)	<code>{"user":{"usuario":"2006110231"}}</code>
Salida (JSON)	<code>[{"Desc_Curso":"QUIMICA I","Nota":"14.00"}, {"Desc_Curso":"CALCULO I","Nota":"7.00"}, {"Desc_Curso":"INGENIERIA GRAFICA","Nota":"12.00"}, {"Desc_Curso":"MATEMATICA BASICA I","Nota":"12.00"}, {"Desc_Curso":"INTRODUCCION A LA COMPUTACION","Nota":"13.00"}, {"Desc_Curso":"METODOLOGIA DEL TRABAJO UNIVERSITARIO","Nota":"15.00"}]</code>
Imagen Salida	<pre> ⊖ [] JSON ⊖ {} 0 ■ Desc_Curso : "QUIMICA I" ■ Nota : "14.00" ⊖ {} 1 ■ Desc_Curso : "CALCULO I" ■ Nota : "7.00" ⊖ {} 2 ■ Desc_Curso : "INGENIERIA GRAFICA" ■ Nota : "12.00" ⊖ {} 3 ■ Desc_Curso : "MATEMATICA BASICA I" ■ Nota : "12.00" ⊖ {} 4 ■ Desc_Curso : "INTRODUCCION A LA COMPUTACION" ■ Nota : "13.00" ⊖ {} 5 ■ Desc_Curso : "METODOLOGIA DEL TRABAJO UNIVERSITARIO" ■ Nota : "15.00" </pre>

Tabla 3: Servicio Lista Nota

Servicio	SB4 - Lista de Horario
Descripción	Servicio que devuelve una lista de los horarios de clases del alumno
Método	POST
Entrada (JSON)	{"user":{"usuario":"2006110231"}}
Salida (JSON)	{ "Jueves": [{"curso": "Ingeniería Gráfica", "hora": "11:00:00 - 12:00:00", "aula": "401"}], "Lunes": [{"curso": "Cálculo I", "hora": "10:00:00 - 12:00:00", "aula": "401"}, {"curso": "Química I", "hora": "18:00:00 - 19:00:00", "aula": "307"}], "Martes": [{"curso": "Química I", "hora": "15:00:00 - 16:00:00", "aula": "302"}, {"curso": "Ingeniería Gráfica", "hora": "11:00:00 - 12:00:00", "aula": "401"}], "Miércoles": [{"curso": "Cálculo I", "hora": "08:00:00 - 09:00:00", "aula": "401"}] }
Imagen Salida	<pre> { "Jueves": [{ "curso": "Ingeniería Gráfica", "hora": "11:00:00 - 12:00:00", "aula": "401" }], "Lunes": [{ "curso": "Cálculo I", "hora": "10:00:00 - 12:00:00", "aula": "401" }, { "curso": "Química I", "hora": "18:00:00 - 19:00:00", "aula": "307" }], "Martes": [{ "curso": "Química I", "hora": "15:00:00 - 16:00:00", "aula": "302" }, { "curso": "Ingeniería Gráfica", "hora": "11:00:00 - 12:00:00", "aula": "401" }], "Miércoles": [{ "curso": "Cálculo I", "hora": "08:00:00 - 09:00:00", "aula": "401" }] } </pre>

Tabla 4: Servicio Lista Horario

Servicio	SB5 - Lista de Record Académico
Descripción	Servicio que devuelve una lista del record académico del alumno
Método	POST
Entrada (JSON)	<code>{"user":{"usuario":"2006110231"}}</code>
Salida (JSON)	<code>{"2014-01":{"curso":"CALCULO I","nota":"7.00"},"curso":"INGENIERIA GRAFICA","nota":"12.00"},"curso":"INTRODUCCION A LA COMPUTACION","nota":"13.00"},"curso":"MATEMATICA BASICA I","nota":"12.00"},"curso":"METODOLOGIA DEL TRABAJO UNIVERSITARIO","nota":"15.00"},"curso":"QUIMICA I","nota":"14.00"},"2014-02":{"curso":"CALCULO I","nota":"12.00"},"curso":"DISEÑO INDUSTRIAL","nota":"4.00"},"curso":"LENGUAJE DE PROGRAMACION I","nota":"14.00"},"curso":"MATEMATICA BASICA II","nota":"9.00"},"curso":"QUIMICA II","nota":"12.00"},"curso":"TEORIA DEL CONOCIMIENTO","nota":"10.00"},"2015-01":{"curso":"CALCULO II","nota":".00"},"curso":"DEFENSA NACIONAL","nota":".00"},"curso":"DEPORTE CALIFICADOS","nota":".00"},"curso":"INFORMES TECNICOS","nota":".00"},"curso":"MICROECONOMIA","nota":".00"},"curso":"QUIMICA ORGANICA","nota":".00"}}</code>

Imagen Salida	<pre> └─ {} JSON └─ [] 2014-01 └─ {} 0 └─ curso : "CALCULO I" └─ nota : "7.00" └─ {} 1 └─ curso : "INGENIERIA GRAFICA" └─ nota : "12.00" └─ {} 2 └─ curso : "INTRODUCCION A LA COMPUTACION" └─ nota : "13.00" └─ {} 3 └─ curso : "MATEMATICA BASICA I" └─ nota : "12.00" └─ {} 4 └─ curso : "METODOLOGIA DEL TRABAJO UNIVERSITARIO" └─ nota : "15.00" └─ {} 5 └─ curso : "QUIMICA I" └─ nota : "14.00" └─ [] 2014-02 └─ {} 0 └─ curso : "CALCULO I" └─ nota : "12.00" └─ {} 1 └─ {} 2 └─ {} 3 └─ {} 4 └─ {} 5 └─ [] 2015-01 </pre>
----------------------	--

Tabla 5: Servicio Lista Record Académico

Anexo 04

Costo del proyecto

Concepto	Costo	Cantidad	Parcial	Total
Desarrollador web service PHP	1200.00	1	1200.00	
Programadores Android	0.00	2	0.00	
Profesional para la revisión de la redacción y ortografía.	150.00	1	150.00	
Laptop	1500.00	2	3000.00	
Impresora	200.00	1	200.00	
Materiales de escritorio, copias, cd's, USB	1030.00		1030.00	
Costo Total				5580.00

Anexo 05

Requerimientos Mínimos de Hardware y Software

1. Requerimientos mínimos de hardware y software para el desarrollo.

- Laptop Procesador I3, de 4 Gb de Ram.
- Dispositivo Móvil (sistema Operativo android).
- IDE ECLIPSE (versión Luna e Indigo).
- SDK de Android.
- S.O. Windows Seven o superior.
- Gestor de base de datos (SQL, My SQL, Oracle, etc.)
- Servidor local XAMPP (APACHE, PHP).

2. Requerimientos mínimos de hardware y software para la implementación.

- Hosting (servidor web con 2 Gb de RAM, 10 Mb de DD).
- Dominio Propio (IP Pública).
- S.O. del servidor web Linux, Debian, Ubuntu.
- Servidor local XAMPP (APACHE, PHP).

3. Requerimientos mínimos de hardware y software para el usuario.

- Smartphone o Tablet con sistema operativo Android y conexión a internet.

Anexo 06

1. Instalación y configuración del entorno Android

Antes de proceder con el desarrollo de cualquier aplicación móvil con Android, necesitamos instalar y configurar un entorno de trabajo, que permitirá al desarrollador depurar el código introducido, y a la vez, probar sus aplicaciones. Para ello, es necesario seguir una serie de pasos, descritos a continuación (cabe destacar que el proceso seguido es para la instalación del entorno en Windows 8.1).

2. JDK (Java Development kit)

Es un software que provee herramientas de desarrollo para la creación de programas en Java. Descargar de la siguiente dirección (a no ser que ya esté instalado en el sistema) el JDK es un complemento para que funcione eclipse:

<http://www.oracle.com/technetwork/java/javase/downloads/java-archive-javase8-2177648.html>

3. Eclipse

Lo primero, es descargar de la siguiente dirección (a no ser que ya esté instalado en el sistema) el IDE con el que se va trabajar, llamado Eclipse:

https://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/luna/SR2/eclipse-jeeluna-SR2-win32-x86_64.zip

4. Plugin ADT

A continuación procedemos a instalar el plugin ADT (Android Development Toolkit) para Eclipse, diseñado para integrar completamente un entorno con el que poder desarrollar aplicaciones Android.

Es tan simple como ir a la barra de menú de Eclipse Help → Install New Software... y añadir la dirección correspondiente que se observa en la siguiente ilustración.

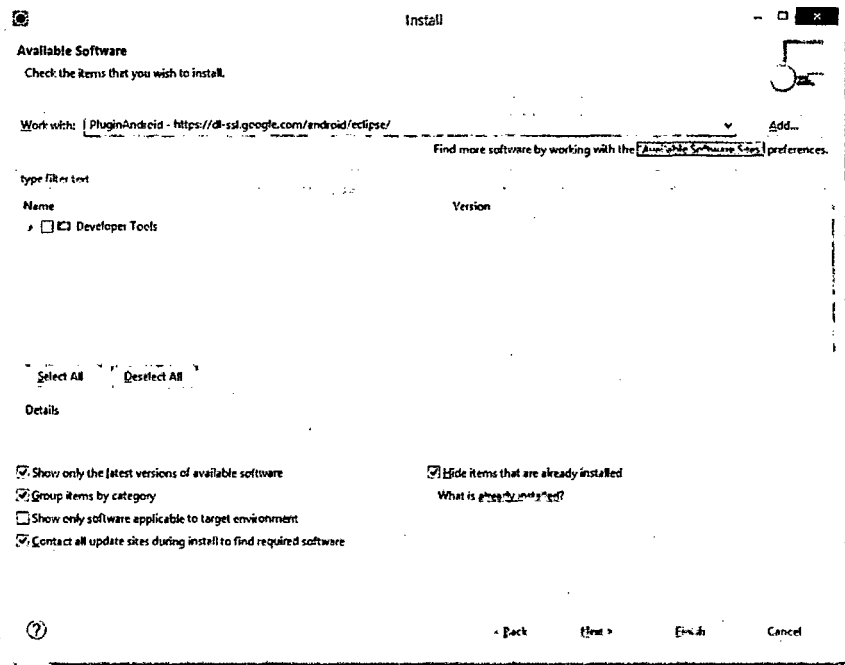


Figura 01: Ventana para la instalación de nuevo software de Android, para Eclipse

5. Android SDK (Software development kit)

También es necesario descargar e instalar el entorno de desarrollo SDK. Para descargarlo hay que acceder a la dirección:

<http://developer.android.com/sdk/index.html>.

Una vez descomprimido el archivo, tenemos decirle a eclipse donde lo puede encontrar, y para ello, están simple como abrir Eclipse, e ir en la barra de menú a Eclipse → Preferencias y clicar en la pestaña Android. Una vez allí, vemos un cuadro de texto, donde hay que buscar el directorio descomprimido del SDK, en nuestro caso, `\Users\Heinz\AppData\Local\Android\android-sdk`.

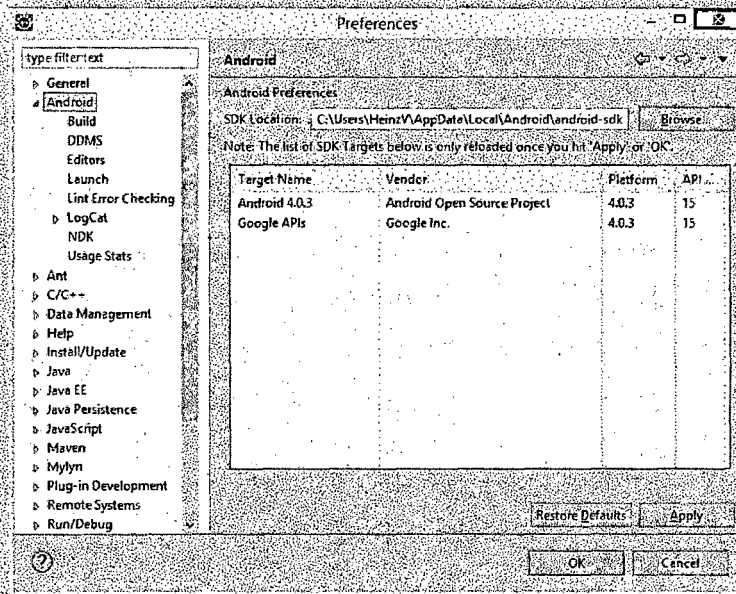


Figura 02: Configuración del SDK de Android

Para instalar nuevas herramientas de desarrollo, simplemente hay que ir a Window → Android SDK and AVD Manager en la barra de menú, y seleccionar de todas ellas, las que sean necesarias.