**UNIVERSIDAD POLITÉCNICA SALESIANA**

**SEDE QUITO**

**CARRERA:**

**INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de:**

**Ingeniero de Sistemas**

**TEMA:**

**Desarrollo de una aplicación móvil para dispositivos Android, para la reserva de laboratorios, materiales y la gestión de notificaciones de mantenimiento de equipos del CECASIS**

**Autor:**

**Fabricio Ricardo Maldonado Taipe**

**TUTOR:**

**GUSTAVO ERNESTO NAVAS RUILOVA**

**Quito, marzo del 2018**

# CESIÓN DE DERECHOS DE AUTOR

Yo, FABRICIO RICARDO MALDONADO TAIPE, con documento de identificación Nº 1725269151, manifestamos mi voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación con el tema: “Desarrollo de una aplicación móvil para dispositivos Android, para la reserva de laboratorios, materiales y la gestión de notificaciones de mantenimiento de equipos del CECASIS”, mismo que ha sido desarrollado para optar por el título de INGENIERO DE SISTEMAS en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

…………………………..

FABRICIO RICARDO

MALDONADO TAIPE

1725269151

Quito, Junio del 2017

# DECLARATORIA DE COAUTORÍA DEL TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, con el tema: “Desarrollo de una aplicación móvil para dispositivos Android, para la reserva de laboratorios, materiales y la gestión de notificaciones de mantenimiento de equipos del CECASIS” realizado por el estudiante FABRICIO RICARDO MALDONADO TAIPE, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito junio del 2018

…………………………………………

GUSTAVO ERNESTO NAVAS RUILOVA

# DEDICATORIA

A mis padres, los cuales me han enseñado con su gran sabiduría a nunca rendirme, con su gran paciencia a ser mejor ser humano, con su gran carisma a sonreír toda mi vida, con su gran ejemplo a superarme cada día más y en especial con su gran amor que lo demuestran cada día de su vida.

A mi hermana ejemplar que me motiva a superarme mis sueños, que me obliga a perder los miedos y cumplir mis metas. “La extraño muchísimo”.

A todos mis tíos y mis tías, mis primos y mis primas que siempre están pendiente de nuestro bienestar y nuestro existo.

Y en especial a Dios el cual me escucha y me apoya cada día de mi vida.AGRADECIMIENTO

Yo agradezco a la Universidad Politécnica Salesiana que ha contribuido en

nuestra formación profesional, al grupo del CECASIS el cual no ha brindado las

facilidades necesarias para realizar el proyecto de titulación.

Fabricio Ricardo Maldonado Taipe

Contenido

[CESIÓN DE DERECHOS DE AUTOR 2](#_Toc518851918)

[DECLARATORIA DE COAUTORÍA DEL TUTOR 3](#_Toc518851919)

[DEDICATORIA 4](#_Toc518851920)

[RESUMEN 8](#_Toc518851921)

[INTRODUCCIÓN 9](#_Toc518851922)

[ANTECEDENTES 9](#_Toc518851923)

[PROBLEMA 10](#_Toc518851924)

[JUSTIFICACIÓN 10](#_Toc518851925)

[OBJETIVO GENERAL 11](#_Toc518851926)

[OBJETIVOS ESPECÍFICOS 11](#_Toc518851927)

[ALCANCE DEL PROYECTO 11](#_Toc518851928)

[CAPÍTULO 1 13](#_Toc518851929)

[1 MARCO TEÓRICO 13](#_Toc518851930)

[1.2. HERRAMIENTAS DE DESARROLLO 13](#_Toc518851931)

[1.2.1. PLATAFORMAS DE DESARROLLO 13](#_Toc518851932)

[1.2.2. BASE DE DATOS 19](#_Toc518851933)

[1.2.2.1. POSTGRESQL 19](#_Toc518851934)

[1.2.3. DEPENDENCIAS Y LIBRERÍAS 20](#_Toc518851935)

[1.2.3.1. PDO 20](#_Toc518851936)

[1.2.3.2. JSON\_ENCODE 20](#_Toc518851937)

[1.2.3.3. QUERY 20](#_Toc518851938)

[1.2.4. SERVICIO WEB 22](#_Toc518851939)

[1.2.5. METODOLOGIA 23](#_Toc518851940)

[1.2.5.1. SPRING 23](#_Toc518851941)

[CAPÍTULO 2 24](#_Toc518851942)

[2. ANALISIS Y DISEÑO 24](#_Toc518851943)

[2.1. REQUERIMIENTOS INICIALES. 24](#_Toc518851944)

[2.1.1. REQUERIMIENTOS DE HARDWARE 24](#_Toc518851945)

[2.1.2. ESPECIFICACION DE REQUERIMIENTOS DE SOFWARE 25](#_Toc518851946)

[2.2.1. REQUERIMIENTOS ESPECÍFICOS 27](#_Toc518851947)

[2.2.1.2. PROCESOS TO BE 29](#_Toc518851948)

[2.1.3. REQUERIMIENTOS NO FUNCIONALES 33](#_Toc518851949)

[2.1.4. REQUERIMIENTOS FUNCIONALES 33](#_Toc518851950)

[2.2. DISEÑO 43](#_Toc518851951)

[2.2.1. DISEÑO DE INTERFACES. 44](#_Toc518851952)

[2.2.3. DISEÑO DE INTERFAZ DE RESERVA DE LABORATORIOS 45](#_Toc518851953)

[2.2.4. DISEÑO DE INTERFAZ DE DISPONIBILIDAD DE MATERIALES 46](#_Toc518851954)

[2.2.5. DISEÑO DE INTERFAZ DE GESTIÓN DE INCIDENTES. 47](#_Toc518851955)

[2.2.2. DISEÑO DE LA BASE DE DATOS 49](#_Toc518851956)

[CAPÍTULO 3 61](#_Toc518851957)

[3. CONSTRUCCIÓN 61](#_Toc518851958)

[3.1. CONSTRUCCIÓN DE LA APLICACIÓN MÓVIL 61](#_Toc518851959)

[3.2. CONSTRUCCIÓN DE INTERFAZ Y FUNCIONALIDAD 62](#_Toc518851960)

[4. PRUEBAS DEL SOFTWARE 74](#_Toc518851961)

[4.1. SOFTWARE Y HARDWARE 74](#_Toc518851962)

[4.2. PRUEBA DE RENDIMIENTO DE LA BASE DE DATOS 75](#_Toc518851963)

[4.3. PRUEBAS DE ACEPTACIÓN 77](#_Toc518851964)

[4.3.1. PERFIL DE USUARIOS 77](#_Toc518851965)

[4.3.2. PRUEBA DE ACEPTACIÓN: LOGIN 78](#_Toc518851966)

[4.3.7. PRUEBAS DE ACEPTACIÓN: VISUALIZAR LOS MATERIALES DISPONIBLES 81](#_Toc518851967)

[CONCLUSIONES 84](#_Toc518851968)

[RECOMENDACIONES 85](#_Toc518851969)

[BIBLIOGRAFIA 86](#_Toc518851970)

# INDICE DE FIGURAS

[Ilustración 1: PBS de la aplicación Móvil 14](#_Toc518851971)

[Ilustración 2: Procesos AS IS Reserva de laboratorios 29](#_Toc518851972)

[Ilustración 3:Procesos AS IS Consulta de Materiales 30](#_Toc518851973)

[Ilustración 4: Procesos AS IS informe de Mantenimiento 31](#_Toc518851974)

[Ilustración 5:Procesos To Be Consulta y Reserva de Laboratorios 33](#_Toc518851975)

[Ilustración 6: Proceso TO BE de Consulta de Materiales 34](#_Toc518851976)

[Ilustración 7: Procesos TO BE de Gestion de Mantenimiento 34](#_Toc518851977)

[Ilustración 8:Diagrama de casos de uso: Autenticación e Ingreso 37](#_Toc518851978)

[Ilustración 9: Diagrama de casos de Uso: Reserva de Laboratorio 39](#_Toc518851979)

[Ilustración 10: Diagrama de casos de uso: Consulta de Materiales 41](#_Toc518851980)

[Ilustración 11: Diagrama de Casos de Uso: Consulta de Materiales 43](#_Toc518851981)

[Ilustración 12: Diagrama de Caso de Uso: Consulta de Materiales 45](#_Toc518851982)

[Ilustración 13:Prototipo de Autenticación e Ingreso 46](#_Toc518851983)

[Ilustración 14: Prototipo de Noticias Actuales 47](#_Toc518851984)

[Ilustración 15:Prototipo de Reserva de Laboratorios 48](#_Toc518851985)

[Ilustración 16: Prototipo de Disponibilidad de Materiales 48](#_Toc518851986)

[Ilustración 17: Prototipo de Gestión de Incidentes 49](#_Toc518851987)

[Ilustración 18: Base de Datos del Sistema Web 51](#_Toc518851988)

[Ilustración 19: Tablas de Autenticación e Ingreso 52](#_Toc518851989)

[Ilustración 20: Tablas de Reserva de Laboratorios 53](#_Toc518851990)

[Ilustración 21: Tablas de Disponibilidad de Materiales 54](#_Toc518851991)

[Ilustración 22: Tablas para Gestión de Mantenimiento 55](#_Toc518851992)

[Ilustración 23:Servicio web para la Autenticación e Ingreso 56](#_Toc518851993)

[Ilustración 24: Servicio Web Consulta de Laboratorios 57](#_Toc518851994)

[Ilustración 25: Diagrama de Clases Autenticación e Ingreso 58](#_Toc518851995)

[Ilustración 26: Diagrama de Clases de Reserva de laboratorios 59](#_Toc518851996)

[Ilustración 27: Diagrama de clases de Materiales 60](#_Toc518851997)

[Ilustración 28: Diagrama de Clases de Gestion de Mantenimiento 61](#_Toc518851998)

[Ilustración 29: Diagrama : Arquitectura de la APP Móvil 63](#_Toc518851999)

[Ilustración 30: Componentes utilizados en Autenticación e Ingreso 64](#_Toc518852000)

[Ilustración 31: Interfaz de Autenticación e Ingreso 65](#_Toc518852001)

[Ilustración 32:Código de Inicio de Sesión 65](#_Toc518852002)

[Ilustración 33Código Guardar estado de Inicio de Sesión 66](#_Toc518852003)

[Ilustración 34: Código Guardado en variables de memoria interna 66](#_Toc518852004)

[Ilustración 35: Código Guardado en variables de memoria interna 66](#_Toc518852005)

[Ilustración 36:Componenetes utilizados en Autenticación e Ingreso 67](#_Toc518852006)

[Ilustración 37:Interfaz de Noticias 67](#_Toc518852007)

[Ilustración 38:Llamada Asincrónica a el servicio web noticias. 68](#_Toc518852008)

[Ilustración 39:Llamada Asincrónica a las imágenes en el servidor web 68](#_Toc518852009)

[Ilustración 40:Interfaz de Reserva de Laboratorios 69](#_Toc518852010)

[Ilustración 41:Llamada a DatePicker de Android 69](#_Toc518852011)

[Ilustración 42:Interfaz de ver laboratorios 70](#_Toc518852012)

[Ilustración 43:Componentes utilizados en ver Laboratorios 71](#_Toc518852013)

[Ilustración 44:Código Declaración de los Arrays para exponer en el ListView 71](#_Toc518852014)

[Ilustración 45:Interfaz de Consulta de Materiales 72](#_Toc518852015)

[Ilustración 46:Componentes utilizados en ver Laboratorios 72](#_Toc518852016)

[Ilustración 47: Código búsqueda de Materiales desde un ListView 73](#_Toc518852017)

[Ilustración 48:Interfaz de Ingreso de Incidente 74](#_Toc518852018)

[Ilustración 49:Componentes utilizados en el ingreso de incidentes 75](#_Toc518852019)

[Ilustración 50:Código de AlertDialog para guardar el incidente. 75](#_Toc518852020)

**INDICE DE TABLAS**

[Tabla 2:Caso de Uso Autenticación e Ingreso 35](#_Toc518852027)

[Tabla 3: Diagrama de Caso de Uso: Consulta de Materiales 39](#_Toc518852028)

[Tabla 4: Diagrama de Casos de Uso: Gestión de Mantenimiento 41](#_Toc518852029)

[Tabla 5:Diagrama de caos de Uso: Noticias 43](#_Toc518852030)

[Tabla 6: Valores de la prueba de carga y rendimiento de la BDD 77](#_Toc518852031)

[Tabla 7:Perfiles de Usuario 79](#_Toc518852032)

[Tabla 8:Prueba de Aceptación: Login 80](#_Toc518852033)

[Tabla 9: Pruebas de Aceptación ; Visualizar Noticias 80](#_Toc518852034)

[Tabla 10: Pruebas de Aceptacion: Ver detalles de Noticias 81](#_Toc518852035)

[Tabla 11: Pruebas de Aceptación: Login recordando Contraseña 81](#_Toc518852036)

[Tabla 12: Prueba de aceptación: Buscar y Visualizar Laboratorio disponibles 82](#_Toc518852037)

[Tabla 13: Prueba de Aceptación: Visualizar los materiales Disponibles 83](#_Toc518852038)

[Tabla 14: Prueba de Aceptación: Ingresar Incidencia 83](#_Toc518852039)

[Tabla 15: Pruebas de Aceptación: Ver incidencias 84](#_Toc518852040)

[Tabla 16: Prueba de Aceptacion: Ver Asignacion de Inicidencias 84](#_Toc518852041)

# RESUMEN

El presente proyecto está orientado al desarrollo de una aplicación móvil, la cual será implementada en la plataforma Android, dedicada a los estudiantes, profesores y personal de CECASIS la cual gestione la reserva de laboratorios, consulta de materiales y reporte de incidentes.

Para el desarrollo de esta aplicación se utilizó la metodología Scrum, ya que esta es una metodología ágil que permitió llevar el control de cada avance y entregable durante el desarrollo del software.

# INTRODUCCIÓN

## ANTECEDENTES

Debido al avance de la tecnología, en especial de la tecnología móvil, cada día las personas desean aumentar la agilidad en los procesos y reducir tiempos muertos que producen inconformidad; por esto, la mayoría de las empresas alrededor del mundo, privadas y públicas, crean aplicaciones móviles para mejorar la experiencia del usuario final y lograr una mayor satisfacción del cliente.

El centro de capacitaciones y servicios informáticos de la Universidad Politécnica Salesiana Campus Sur (CECASIS) ofrece soporte preventivo y correctivo a las computadoras situadas en los laboratorios.

Servicios del CECASIS

* Servicios de préstamo de hardware para estudiantes internos de la Universidad Politécnica Salesiana
* Servicio de capacitación para estudiantes internos y externos a la Universidad Politécnica Salesiana
* Mantenimiento preventivo como correctivo de las computadoras ubicadas los laboratorios de la Universidad Politécnica Salesiana.
* Soporte de incidentes y problemas a las computadoras de los laboratorios ubicadas en el CECASIS.

## PROBLEMA

Los estudiantes y profesores de la Universidad Politécnica Salesiana notan la existencia de una pérdida de tiempo de los estudiantes y profesores al momento de realizar la reserva de un laboratorio o la petición de un material en el CECASIS ya que se necesita estar personalmente para realizar esta tarea. Con una aplicación móvil esto sería mucho más sencillo, eficiente, rápido y fácil de gestionar.

Existe una pérdida de recursos por parte del personal de CECASIS ya que lleva un registro manual lo cual dificulta la gestión de laboratorios, el análisis y la toma de decisión en base a esa información registrada.

## JUSTIFICACIÓN

Ofrecer a los alumnos de la UNIVERSIDAD POLITÉCNICA SALESIANA una aplicación móvil que mejore la eficiencia en el CECASIS en la gestión de laboratorios, materiales y mantenimiento apoyando a los objetivos estratégicos de la Universidad, al mejorar el acceso a la información relevante, concreta y oportuna a través de las tecnologías de la información y comunicación, como consecuencia reducirían costos lo que apoyará una gestión económica financiera, que facilita la toma de decisiones y el cumplimiento de los objetivos institucionales y ayudará a la comunidad universitaria de la UPS contar con información gestionada y conservada eficientemente.

## OBJETIVO GENERAL

Desarrollar una aplicación móvil para dispositivos Android, para la reserva de laboratorios, consulta los materiales disponibles y gestiona las notificaciones de mantenimiento de equipos del CECASIS y de este modo aportar el incremento de la eficiencia del grupo de trabajo.

## OBJETIVOS ESPECÍFICOS

* Levantar y diagramar procesos con la notación BPMN para tener una visión más clara del funcionamiento del CECASIS.
* Especificar los requerimientos necesarios con un modelo interactivo para lograr desarrollar una aplicación que cumpla con las necesidades de los interesados.
* Diseñar una aplicación móvil en Android Studio que sea fácil de aprender y usar
* Codificar la aplicación tomando en cuenta la limitación del Hardware de los teléfonos móviles.
* Utilizar el patrón MVC en la implementación de la aplicación para que el mantenimiento y futuras versiones sea más fácil de realizar.
* Realizar pruebas de validación de las funcionalidades requeridas.

## ALCANCE DEL PROYECTO

Para definir el alcance del proyecto se hizo uso de la Estructura de Descomposición del Producto o Product Breakdown Structure (PBS), en donde podemos presentar las partes funcionales de la APP.

La App móvil contará con 3 módulos:

* Noticias CECASIS
* Todos los usuarios tendrán acceso a ver las noticias y el detalle de las noticias del CECASIS
* Reserva de Materiales

Este módulo gestionará la reserva de laboratorios, para los estudiantes y los docentes de la Universidad Politécnica Salesiana.

* Consultas de Materiales

Todos los usuarios tendrán acceso a ver la disponibilidad de los materiales para préstamos en el CECASIS

* Reporte de Incidentes

Todos los usuarios tendrán acceso a reportar un incidente de maquina o laboratorio.

|  |
| --- |
| PBS del proyecto    Ilustración 1: PBS de la aplicación Móvil  Realizado por Fabricio Maldonado |

# CAPÍTULO 1

## 1 MARCO TEÓRICO

## 1.2. HERRAMIENTAS DE DESARROLLO

### 1.2.1. PLATAFORMAS DE DESARROLLO

#### 1.2.1.1. Android Studio

Android Studio es un IDE (Entorno de desarrollo Integrado) para la plataforma Android basado en el software IntelliJ IDEA especialmente diseñado para el desarrollo en Android.

Características:

* Renderizado en tiempo real
* Construcción con Gradle
* Soporte para desarrollar en Android Wear
* Templetes para el diseño de interfaces
* ProGruad, firma de aplicaciones
* Edición grafica de componentes
* Emulador Android para ejecutar y probar aplicaciones

(developer.android.com, 2017)

#### 1.2.2.2. BIZAGI MODELER

Bizagui es un Freeware que es utilizado para diagramar, documentar y simular procesos realizados con la notación BPMN (Business Process Model and Notation). Permite ejecutar flujos de trabajo y es considerada como una solución a la gestión de procesos. Bizagui automatiza procesos complejos.

(Bizagi, 2018)

#### 1.2.2.3. SAP POWERDESIGNER

PowerDesigner es una herramienta de modelado empresarial colaborativo que tiene soporte para diferentes tipos de modelos. El más importarte para este proyecto es el Modelo de datos el cual funciona con la mayoría de sistemas de base de datos relacionales y utilizar para crear el modelo conceptual y físico de la base de datos.

PowerDesigner permite visualizar, analizar y manipular metadatos para tener una mejor arquitectura empresarial de información brindando un enfoque basado en modelos lo cual permite a la tecnología alinear con el giro del negocio logrando una mejor implementación de arquitecturas de información

Está diseñado con buenas prácticas de análisis, diseño y gestión de metadatos brindando una mejor productividad individual y grupal.

(Wang, 2016)

#### 1.2.2.4. SERVIDOR HTTP APACHE

HTTP Apache es un servidor web multiplataforma que implementa el protocolo HTTP, es altamente configurable ya que es modular y extensible. Este es un programa diseñado para trasferir datos de hipertexto (páginas web) con todos sus elementos.

Apache es el componente de la plataforma de aplicaciones LAMP, junto a MySQL y los lenguajes de programación Phyton/Perl/ PHP y Ruby, ya que contiene varios paquetes redistribuidos de software propietario.

La licencia de Apache permite distribuir la derivación de código abierto y cerrado a partir de su código fuente original.

(Apache Software Foundation., 2018)

#### 1.2.2.5. XAMPP

Xampp es un software libre que gestiona el servidor apache, el gestor de base de datos MySQL, contiene intérpretes para lenguajes de Script PHP y Pearl.

Permite la instalación sencilla de Apache en el ordenador, sin importar el sistema operativo y es una herramienta que permite probar tu trabajo en tu propio ordenador sin necesidad de tener acceso a internet.

Paquetes importantes que se instalan con XAMPP:

* Apache, el servidor Web más famoso
* MySQL, una excelente base de datos de código libr
* PHP y Perl: lenguajes de programación
* ProFTPD: un servidor FTP
* OpenSSL: para soporte a la capa de sockets segura

( Apache Friends, 2018)

#### 1.2.2.6. JAVA

Java es un lenguaje de programación orientado a objetos, actualmente es un lenguaje más popular alrededor del mundo ya que tiene la capacidad de escribir una vez el código que se ejecutaría en cualquier dispositivo gracias a que cuenta con JVM o Java Virtual Machine

Ventajas de utilizar Java:

* Lenguaje simple
* Lenguaje orientado a objetos
* Aplicaciones distribuidas
* Interpretado y compilado
* Lenguaje Seguro.

(Oracle, 2018)

#### 1.2.2.7. PHP

PHP es un lenguaje de código abierto utilizado especialmente para el desarrollo web el cual puede ser incrustado en HTML.

PHP puede emplearse en todos los sistemas operativos también puede ejecutarse en cualquier servidor web que pueda utilizar el binario de PHP.

(The PHP Group, 2018)

#### 1.2.2.8. JSON

JSON (Notación de Objetos de JavaScript) es un formato liviano de intercambio de datos utilizado mayormente para la trasmisión de información de servicios web.

JSON está formado por dos estructuras:

* Pares de nombre y valor.
* Una lista ordenada de valores, se los conocen como arreglos, vectores, listas, etc.

(Nederlands, 2018)

#### 1.2.2.9. BPMN

BPMN (Modelo y Notación de Procesos de Negocio) es una notación estandarizada que permite modelar procesos de negocios en forma de flujo de trabajo.

|  |
| --- |
| https://upload.wikimedia.org/wikipedia/commons/thumb/2/2b/BPMN-UnProcesoConFlujoNormal-version2.svg/500px-BPMN-UnProcesoConFlujoNormal-version2.svg.pngEjemplo de diagrama de proceso de negocio.  Figure :Ejemplo de componentes de BPMN  Realizado: (Debevoise, 2016) |

La principal misión de esta notación es facilitar de manera gráfica un estándar de procesos que sea fácil legible y legible.

(Debevoise, 2016)

1.2.2.10. GRADLE

Gradle es un sistema de automatización de construcción de código que introduce un lenguaje específico de dominio (DSL) basado en Grovy.

Ventajas de Gradle en Android

* Reutilización de código.
* Fácil configuración y personalización en la compilación.
* Gestiona las dependencias de forma potente y cómoda.
* Fácil creación de diferentes versiones de la aplicación.

( Gradle Inc, 2018)

1.2.3.11. PENCIL PROYECT

Pencil Proyect es una herramienta de creación de prototipos funcionales de código abierto que está disponible para todos los sistemas operativos.

(Evolus, 2018)

1.2.3.12. MATERIAL DESIGNER

Material Design es una guía para el diseño visual, de movimientos y de interacción en Android brindando los siguientes elementos que permitirán crear aplicaciones Material Designer :

* Un tema nuevo Material Designer
* Nuevos widgets para vistas complejas
* Nuevas API (interfaces de programación de aplicaciones) para sombras y animaciones personalizadas.

(Google, 2018)

* + - 1. APACHE JMETER.

La aplicación Apache JMeter es un software de código abierto diseñado para realizar pruebas de carga y rendimiento funcional de una aplicación.

Se puede usar para simular una gran carga en un servidor para probar su rendimiento general por medio de diferentes tipos de carga.

(Apache Software Foundation, 2018)

### 1.2.2. BASE DE DATOS

### 1.2.2.1. POSTGRESQL

PostgreSQL es un sistema de gestión de bases de datos relacional orientado a objetos.

Características

* Alta concurrencia
* Amplia variedad de tipo de datos.
* PostgreSQL provee nativamente soporte para:
  + Números de precisión arbitraria.
  + Texto de largo indefinido.
  + Arrays.
  + Diferentes complementos para geo análisis.

(Riggs, 2018)

### 1.2.3. DEPENDENCIAS Y LIBRERÍAS

### 1.2.3.1. PDO

La Extensión Objetos de Datos es una interfaz para poder acceder a diferentes bases de datos en PHP exponiendo características concretas de la base de datos a conectar.

Para lograr una conexión se debe utilizar un controlador de PDO específico de la base de datos para tener acceso a un servidor de bases de datos que se lo habilita en PHP.

(The PHP Group, 2018)

### 1.2.3.2. JSON\_ENCODE

Un uso habitual de JSON es leer datos de un servidor web y exponer los datos en una página web.

Json\_encode: Devuelve la información JSON del valor dado

(Group, s.f.)

### 1.2.3.3. QUERY

Query se utiliza cuando se quiere crear una consulta a una base de datos relacional y no relacional. Es una consulta SQL

1.2.3.4. BIBLIOTECA APPCOMPAT V7

Esta biblioteca añade compatibilidad con los componentes de la interfaz de usuario de material design.

Appcompat v7 incluyen algunas clases como:

* ActionBar: proporciona una implementación de la barra de acciones.
* AppCompatDialog: añade una clase base para diálogos temáticos de AppCompat.
* ShareActionProvider: añade compatibilidad para un uso compartido como correo electrónico o publicación en aplicaciones sociales.

(Google, 2018)

1.2.3.5. SUPPORT:DESIGN

La biblioteca de soporte añade soporte para varios componentes de diseño, como cajones de navegación, botones de acción flotante, barras de aperitivos y pestañas.

(Google, 2018)

1.2.3.6. ANDROID-ASYNC-HTTP

Android-Async-http es un cliente Http basado en la devolución de llamadas asincrónicas en Android.

Todas las solicitudes se realizan fuera de la secuencia de interfaz de usuario principal de su aplicación, pero cualquiera devolución de llamada se ejecutará en el mismo subproceso que se creó mediante el uso de mensajes del Manejador de Android.

(Smith, 2016)

1.2.3.7. SMARTIMAGEVIEW

SmarthImagesView es una dependencia permite que se cargue imágenes de un URL o contactos de Android y se almacenen en cache en la memoria y en el disco para una carga súper rápida.

, esta dependencia es un reemplazo de ImageView Las imágenes se almacenan en caché

Características:

* La carga Asincrónica ocurre fuera del hilo de la interfaz de usuario
* Las imágenes se acumulan en caché en la memoria y en el disco para una carga súper rápida
* SmartImage puede cargar desde otras fuentes

(Smith, Android Smart Image View, 2016)

### 1.2.4. SERVICIO WEB

1.2.4.1. RESTFUL

Un servicio web es un grupo de protocolos utilizados para intercambiar datos entre aplicaciones.

Los softwares creados en diferentes lenguajes de programación pueden usar servicios web para intercambiar datos.

* GET: Proporciona acceso de solo lectura a datos.
* PUT: Consumido para crear nuevos datos.
* ELIMINAR: Consumido para eliminar datos.
* POST: Se usa para modificar o crear datos.

(World Wide Web, 2016)

### 1.2.5. METODOLOGIA

1.2.5.1. SCRUM

Scrum es una metodología de desarrollo ágil para software por lo cual es incremental interactiva. Se utiliza para guiar las siguientes actividades: requerimientos, análisis, diseño evolución y entrega.

### 1.2.5.1. SPRING

Un sprint es la unidad de tiempo que determina un ciclo de desarrollo con Scrum.

1.2.5.2. PROTOTIPADO

El prototipo debe ser construido en poco tiempo, usando los programas adecuados y no se debe utilizar muchos recursos.

El diseño del prototipo se centra en aquellos aspectos del software que serán visibles para el usuario final, el cual es evaluado por el cliente para una retroalimentación así refinamos los requisitos del software que se desarrollará.

El prototipado ayuda al desarrollado y al cliente a entender de mejor manera cuál será el resultado final del software y así lograr la satisfacción del cliente. De esta manera se involucra al cliente para lograr mejor agilidad en la construcción de la aplicación.

(Maner, 2018)

# CAPÍTULO 2

## 2. ANALISIS Y DISEÑO

En este capítulo se presenta el análisis previo realizado para cumplir con los objetivos planeados, así como los casos de uso y el prototipo a utilizarse para la construcción de la aplicación móvil.

## 2.1. REQUERIMIENTOS INICIALES.

### 2.1.1. REQUERIMIENTOS DE HARDWARE

Los requerimientos de hardware utilizados para el desarrollo de la aplicación en Android Studio son:

#### 2.1.1.1. COMPUTADOR

* Intel Core i7 6500U de 2.6 GHZ
* Memoria Ram: 12GB
* Tarjeta de video: AMD Radeon R5 m335 de 4GB
* Disco Duro: 1TB

#### 2.1.1.2. DISPOSITIVO MOVIL

* Marca: Nokia 6
* Pantalla: 5.5 pulgadas
* Resolución: 1,080x1,920 pixeles
* Procesador: Snapdragon 430 de ocho núcleos (1.4GHz)
* RAM: 3GB
* Almacenamiento: 32GB
* Ranura para tarjeta microSD: Sí
* Batería: 3,000mAh
* Cámara trasera: 16 megapixeles
* Cámara frontal: 8 megapixeles
* Sistema operativo: Android 8 Oreo
* Conectividad: Bluetooth 4.1, Wi-Fi 802.11n (2.4GHz y 5GHz)

### 2.1.2. ESPECIFICACION DE REQUERIMIENTOS DE SOFWARE

#### 2.1.2.1 PROPÓSITO

El presente proyecto tiene como propósito facilitar una aplicación móvil que automatice la gestión de reserva de laboratorios, disponibilidad de materiales,

La herramienta debe funcionar conjuntamente con la aplicación web desarrollada en el CECASIS.

La aplicación móvil debe permitir al profesor reservar un laboratorio exitosamente.

La aplicación móvil debe gestionar el mantenimiento de los laboratorios exitosamente.

La aplicación móvil debe verificar la disponibilidad y existencias de materiales exitosamente.

La aplicación móvil debe visualizar las noticias del CECASIS exitosamente de manera agradable a la vista.

2.1.2.2. ALCANCE

El proyecto se divide en cinco etapas: ingreso y autenticación con servicios web, revisión de ultimas noticias del CECASIS, disponibilidad y reserva de laboratorios según autenticación, disponibilidad de materiales que ofrece CECASIS, gestión de incidentes de laboratorios y materiales, cierre de sesión.

|  |
| --- |
| PBS del Proyecto    Figure 2. PBS que modela los componentes del proyecto  Elaborado por Fabricio Maldonado |

### 2.2.1. REQUERIMIENTOS ESPECÍFICOS

A continuación, se presenta los procesos funcionales del CECASIS:

#### 2.2.1.1. PROCESOS AS IS

Se utilizó los Proceso AS IS o “tal como está” que es levantamiento de procesos para tener una perspectiva de que se debe automatizar.

#### 2.2.1.1.1. RESERVA DE LABORATORIOS

La reserva de laboratorios se levantó junto con el dueño del proceso, en este caso con el personal del CECASIS. Se utilizó la notación BPMN para modelar dicho proceso.

|  |
| --- |
| Procesos AS IS Reserva de Laboratorios.  Ilustración 2: Procesos AS IS Reserva de laboratorios  Realizado por: Fabricio Maldonado |

En el proceso notamos que se registra manualmente la verificación de los laboratorios disponibles, la entrega del formulario y los registros de reservas de laboratorios. Estas acciones requieren automatización.

#### 2.2.1.1.2. CONSULTA DE MATERIALES

La consulta de materiales se levantó junto con el dueño del proceso, en este caso con el personal del CECASIS. Se utilizó la notación BPMN para modelar dicho proceso.

|  |
| --- |
| Proceso AS IS consulta de materiales    Ilustración 3:Procesos AS IS Consulta de Materiales  Realizado por: Fabricio Maldonado |

En el proceso notamos que el Estudiante o Docentes deben acercarse presencialmente al CECASIS para verificar la disponibilidad de los materiales de préstamos.

#### 2.2.1.1.3. INFORME DE MANTENIMIENTO

El informe de mantenimiento se levantó junto con el dueño del proceso, en este caso con el personal del CECASIS. Se utilizó la notación BPMN para modelar dicho proceso.

|  |
| --- |
| Procesos AS IS Informe de mantenimiento    Ilustración 4: Procesos AS IS informe de Mantenimiento  Realizado por: Fabricio Maldonado |

En el proceso notamos que se realiza un proceso manual por parte del Estudiante o Profesor al escribir el problema en el formulario. Existen tiempos muertos ya que una vez al día los técnicos revisan los problemas de mantenimientos.

### 2.2.1.2. PROCESOS TO BE

Se utilizó los Procesos TO BE o “como debe ser” que es el proceso ya optimizado que se debe automatizar.

Se maneja los siguientes roles:

* Estudiante
* Docente
* Técnico
* Secretaria

#### 2.2.1.2.1. CONSULTA Y RESERVA DE LABORATORIOS

En la optimización de proceso de consulta y reserva de laboratorios se separó dos importantes tipos de usuarios:

Estudiantes: los estudiantes pueden ver los laboratorios disponibles y reservar los laboratorios con estado pendiente.

Docentes: los docentes pueden ver la reserva del estudiante y tiene la potestad de aceptar o rechazar la reserva, así mismo puede hacer sus propias reservaciones.

|  |
| --- |
| Procesos To Be Consulta Y Reserva de Laboratorios  Ilustración 5:Procesos To Be Consulta y Reserva de Laboratorios  Realizado por : Fabricio Maldonado |

Tanto como docentes y estudiantes pueden hacer reservas para laboratorios, la diferencia es que para los estudiantes el estado se guarda como pendiente para que el docente sea quien apruebe la reserva.

2.2.1.2.2. CONSULTA DE MATERIALES

La consulta de materiales se levantó junto con el dueño del proceso, en este caso con el personal del CECASIS. Se utilizó la notación BPMN para modelar dicho proceso.

|  |
| --- |
| Procesos de Consulta de Materiales  Ilustración 6: Proceso TO BE de Consulta de Materiales  Realizado por Fabricio Maldonado |

2.2.1.2.3. GESTION DE MANTENIMIENTO

En la gestión del mantenimiento se eliminaron los procesos manuales los cuales fueron optimizados mediante registros de software

|  |
| --- |
| Procesos TO BE de Gestion de Mantenimiento    Ilustración 7: Procesos TO BE de Gestion de Mantenimiento  Realizado por: Fabricio Maldonado |

El estudiante o docente es encargado de registrar el incidente de los laboratorios.

El técnico asignado en ese horario será el encargado de solucionar el caso y cambiar de esto a la actividad para tener informado al personal del CECASIS sobre el incidente.

El técnico asignado será el encargado de solucionar y registrar la solución del trabajo realizado.

### 2.1.3. REQUERIMIENTOS NO FUNCIONALES

La aplicación funciona y trabaja en diferentes ambientes (web y aplicación móvil) y hace uso de las siguientes aplicaciones:

**Windows 10**. Sistema Operativo en el que se instala los programas de desarrollo.

Apache, Version 2.4.33 Utilizado para alojar los servicios web de la cual consume la aplicación móvil.

**PHP. Versión 5.6.36**. Lenguaje de programación para levantar los servicios web y consumir de la base de datos de la aplicación web.

**PostgreSQL.** Versión 10. Utilizado para almacenar la información que será consumida por el servicio web y presentarla en la aplicación móvil.

Java 8. Leguaje de programación en la que se encuentra escrita la aplicación móvil , la cual debe ser compilada y ejecutada.

**Android Studio 3.1.2.** IDE de desarrollo en la cual mediante un Material Designer y programación en JAVA se genera y ejecuta la aplicación web.

JVM OpenJDK. Máquina Virtual en la que corre la aplicación en el dispositivo móvil.

### 2.1.4. REQUERIMIENTOS FUNCIONALES

#### 2.1.4.1. DIAGRAMAS DE CASOS DE USO

#### 2.1.4.1.1. CASO DE USO – AUTENTICACIÓN E INGRESO

|  |  |
| --- | --- |
| Caso de Uso #1 | Autenticación e ingreso |
| Actores | Usuario |
| Camino Principal | 1. Instalar la aplicación en el dispositivo móvil 2. Ingresar correo institucional 3. Ingresar contraseña institucional |
| Precondiciones | Para ingresar a la aplicación el servicio web debe regresar valores y no valores nulls |
| Post-condiciones | Instalación de la aplicación  El servidor de servicios debe ser inicializado. |

Tabla 2:Caso de Uso Autenticación e Ingreso

Realizado por: Fabricio Maldonado

La autenticación y el ingreso es realizado consumiendo un servicio REST el cual está levantando en el servidor de aplicaciones que solicita dos variables, el correo institucional y la contraseña de usuario.



Ilustración 8:Diagrama de casos de uso: Autenticación e Ingreso

Realizado por: Fabricio Maldonado

2.1.4.1.2. DIAGRAMA DE CASO DE USO – RESERVAR LABORATORIO (DOCENTES, ESTUDIANTES)

|  |  |
| --- | --- |
| Caso de Uso #2 | Autenticación e ingreso |
| Actores | Docente / Estudiantes |
| Camino Principal | 1. Ingresar a opciones en el panel izquierdo superior de la pantalla 2. Seleccionar la opción Laboratorios 3. Seleccionar la fecha de reserva 4. Seleccionar la hora de reserva 5. Seleccionar el número de laboratorios 6. Seleccionar el software requerido 7. Buscar laboratorios disponibles 8. Ver y revisar el laboratorio disponible 9. Reservar el laboratorio disponible. |
| Precondiciones | Acceder a la cuenta de usuario y autentificarse como docente de la Universidad Politécnica Salesiana |
| Post-condiciones | Instalación de la aplicación  El servidor de servicios debe ser inicializado.  Disponibilidad de laboratorios |



Ilustración 9: Diagrama de casos de Uso: Reserva de Laboratorio

Realizado por: Fabricio Maldonado.

La diferencia entre los dos actores es que los estudiantes tienen la potestad de reservar un laboratorio en estado “Pendiente” mientras el tutor responsable asignado no lo apruebe. El docente tiene la potestad de reservar un laboratorio en estado “Reservado”

2.1.4.1.3. DIAGRAMA DE CASO DE USO – CONSULTA DE MATERIALES

|  |  |
| --- | --- |
| Caso de Uso #3 | Autenticación e ingreso |
| Actores | Docente / Estudiantes |
| Camino Principal | 1. Ingresar a Opciones 2. Ingresar a Materiales 3. Dar click en el botón buscar materiales 4. Escribir el material requerido 5. Click en el material requerido 6. Visualizar el material requerido 7. Verificar disponibilidad |
| Precondiciones | Acceder a la cuenta de usuario y autentificarse como docente o estudiante de la Universidad Politécnica Salesiana |
| Post-condiciones | Instalación de la aplicación  El servidor de servicios debe ser inicializado. |

Tabla 3: Diagrama de Caso de Uso: Consulta de Materiales

Realizado por: Fabricio Maldonado



Ilustración 10: Diagrama de casos de uso: Consulta de Materiales

Realizado por: Fabricio Maldonado

2.1.4.1.4. DIAGRAMA DE CASOS DE USO – GESTIÓN DE MANTENIMIENTO

|  |  |
| --- | --- |
| Caso de Uso #4 | Autenticación e ingreso |
| Actores | Docente / Estudiantes / Técnico / Secretaria |
| Camino Principal | 1. Ingresar a Opciones 2. Ingresar a Mantenimiento 3. Ingresar a Incidentes 4. Ingresar tipo de incidentes 5. Ingresar descripción de incidentes 6. Tomar foto del incidente 7. Guardar incidente 8. Ver el estado incidente guardado 9. Ingresar a Registro de Incidentes 10. Cambiar de estado al incidente 11. Revisar estado del incidente. |
| Precondiciones | Acceder a la cuenta de usuario y autentificarse como docente, estudiante, técnico o secretaria de la Universidad Politécnica Salesiana |
| Post-condiciones | Instalación de la aplicación  El servidor de servicios debe ser inicializado. |

Tabla 4: Diagrama de Casos de Uso: Gestión de Mantenimiento

Realizado por: Fabricio Maldonado

La Gestión del incidente consume Servicios REST los cuales deben estar levantados en el servidor de aplicaciones.

Los servicios que se deben consumir son: ingreso de incidentes, consulta de incidentes, modificación de estado de incidentes.



Ilustración 11: Diagrama de Casos de Uso: Consulta de Materiales

Realizado por: Fabricio Maldonado

El usuario Secretaria tiene acceso para supervisar el estado del incidente y que ayude a la toma de decisiones.

2.1.4.1.5. DIAGRAMA DE CASO DE USO – NOTICIAS

|  |  |
| --- | --- |
| Caso de Uso #4 | Autenticación e ingreso |
| Actores | Docente / Estudiantes / Técnico / Secretaria |
| Camino Principal | 1. Ingresar correo institucional 2. Ingresar contraseña institucional 3. Ver las noticias. |
| Precondiciones | Acceder a la cuenta de usuario y autentificarse como docente, estudiante, técnico o secretaria de la Universidad Politécnica Salesiana |
| Post-condiciones | Instalación de la aplicación  El servidor de servicios debe estar inicializado. |

Tabla 5:Diagrama de caos de Uso: Noticias

Realizado por: Fabricio Maldonado



Ilustración 12: Diagrama de Caso de Uso: Consulta de Materiales

Realizado por: Fabricio Maldonado

La consulta de materiales se lo realiza como método informativo para evitar la consulta presencial sobre la disponibilidad de dichos materiales.

## 2.2. DISEÑO

En esta sección se expone los bosquejos de las pantallas de la aplicación móvil para cumplir con los requerimientos establecidos.

### 2.2.1. DISEÑO DE INTERFACES.

2.2.1.1. DISEÑO DE INTERFAZ AUTENTICACION E INGRESO

La aplicación móvil debe permitir al usuario acceder a la aplicación, para ello necesita el correo institucional y contraseña, estar en la base de datos de los estudiantes, profesores, técnicos y secretaria. En caso que la información sea incorrecta o no existente la aplicación mostrara un mensaje de error sobre la autenticidad mediante un servicio web.

|  |
| --- |
| Prototipo de Autenticación e Ingreso    Ilustración 13:Prototipo de Autenticación e Ingreso  Realizado por: Fabricio Maldonado |

Una vez la autenticación sea correcta, o el usuario y contraseña sean correctos, la aplicación móvil automáticamente se dirigirá a una sección de noticias.

2.2.2. DISEÑO DE NOTICIAS ACTUALES.

Una vez realizada la autenticación satisfactoriamente la pantalla despliega una lista de las noticias actuales o información relevante del CECASIS, las cuales contienen una imagen relevante de la noticia y el titular de la misma.

Al tocar la noticia se despliega un popUp en el cual se visualizará de manera más detallada la imagen, el nombre y la descripción de la noticia.

|  |
| --- |
| Prototipo de Noticias Actuales  Ilustración 14: Prototipo de Noticias Actuales  Realizado por: Fabricio Maldonado |

### 2.2.3. DISEÑO DE INTERFAZ DE RESERVA DE LABORATORIOS

La primera sección que se visualiza es la reserva de laboratorios en la cual existen dos subsecciones: Reserva de Laboratorios y Ver Laboratorios. Esto se realizó con el objetivo de mejorar la usabilidad y experiencia del usuario.

En Reserva de Laboratorios procederemos a filtrar y reservar los laboratorios disponibles. En Ver laboratorios se verificará los laboratorios reservados.

|  |
| --- |
| Prototipo de Reserva de laboratorios  Ilustración 15:Prototipo de Reserva de Laboratorios  Realizado por: Fabricio Maldonado |

### 2.2.4. DISEÑO DE INTERFAZ DE DISPONIBILIDAD DE MATERIALES

De la interfaz unitaria se puede seleccionar la subsección Materiales la cual nos ayudara a ver la disponibilidad de los materiales que ofrece el CECASIS a los estudiantes y docentes.

|  |
| --- |
| Prototipo de Disponibilidad de Materiales.    Ilustración 16: Prototipo de Disponibilidad de Materiales  Realizado por Fabricio Maldonado |

### 2.2.5. DISEÑO DE INTERFAZ DE GESTIÓN DE INCIDENTES.

|  |
| --- |
| Prototipo de Gestión de Incidentes  Ilustración 17: Prototipo de Gestión de Incidentes  Realizado por: Fabricio Maldonado |

El diseño de la interfaz de Gestión de incidentes cuenta con 3 subseciones Reportes de Incidentes, Ver Incidentes y Ver asignación.

**Reporte de Incidentes:**

En esta sección se ingresará los reportes de incidentes por parte del usuario final de los laboratorios.

**Ver incidentes:**

En esta sección se verificará los incidentes ingresados y sus estados.

**Ver asignación:**

En esta sección se presentará una sección para el cambio de estado del incidente por parte del técnico.

### 2.2.2. DISEÑO DE LA BASE DE DATOS

|  |
| --- |
| Base de Datos del Sistema Web.    Ilustración 18: Base de Datos del Sistema Web  Realizado por: (Quilumba, 2018) |

La base de datos se encuentra actualmente en funcionamiento con el Sistema Web del Cecasis el cual gestiona Laboratorios, Materiales, Incidentes, Reservas, etc. El cual se está desarrollando junto a este proyecto.

Se está consumiendo servicios Web que tienen comunicación con las tablas necesarias para el funcionamiento de la aplicación y lograr su integración sin necesidad de conectarse a la base directamente.

Las tablas más importantes las cuales estas conectadas con el servicio web son:

2.2.2.1. DISEÑO DE BASE DE DATOS DE AUTENTICACION E INGRESO.

|  |
| --- |
| Tablas de Autenticación e Ingreso    Ilustración 19: Tablas de Autenticación e Ingreso  Realizado por: (Quilumba, 2018) |

Las tablas esenciales para realizar la autenticación son rol y usuarios, ya que la aplicación móvil maneja sesiones y habilitar opciones de acuerdo al rol del usuario.

#### 2.2.2.2. DISEÑO DE BASE DE DATOS DE RESERVA DE LABORATORIO.

|  |
| --- |
| Tablas de Reserva de Laboratorios.    Ilustración 20: Tablas de Reserva de Laboratorios  Realizado por: (Quilumba, 2018) |

Se muestran en la imagen las tablas esenciales para el funcionamiento de la subsección Laboratorios la cual realiza el filtrado de laboratorio por fecha, número de máquinas y el software que se encuentra instalado en los laboratorios.

#### 2.2.2.3. DISEÑO DE BASE DE DATOS DE DISPONIBILIDAD DE MATERIALES.

|  |
| --- |
| Tablas de Disponibilidad de Materiales.    Ilustración 21: Tablas de Disponibilidad de Materiales  Realizado por: (Quilumba, 2018) |

Se muestran en la imagen las tablas esenciales para el funcionamiento de la subsección Materiales la cual realiza la búsqueda y verificación de disponibilidad y estado de los materiales.

#### 2.2.2.4. DISEÑO DE BASE DE DATOS DE GESTION DE MANTENIMIENTO (INCIDENTES).

|  |
| --- |
| Tablas para Gestión de Mantenimiento    Ilustración 22: Tablas para Gestión de Mantenimiento  Realizado por: (Quilumba, 2018) |

Se muestran en la imagen las tablas esenciales para el funcionamiento de la subsección Mantenimiento, la cual se encarga de registrar un incidente o daño de la maquina o máquinas de los laboratorios del CECASIS.

2.2.3. DISEÑO DE AUTENTICACION DE APLICACIÓN MEDIANTE TOKENS

2.2.3. DISEÑO DEL SERVICIO WEB

Los servicios web tienen que estar constantemente levantados para ser consumidos, así mismo para lograr la integración con la aplicación web la cual va a gestión de manera más precisa Laboratorios, Materiales e Incidentes.

#### 2.2.3.1. SERVICIO AUTENTICACIÓN DE USUARIOS

El servicio se realizó mediante una consulta a las tablas usuarios enviando por servicio REST el correo institucional y la contraseña, para luego que el servidor de aplicaciones gestione la consulta y la convierta en un JSON para ser abierto y modelado por el dispositivo móvil.

|  |
| --- |
| Servicio Web Para la Autenticación e Ingreso.  Ilustración 23:Servicio web para la Autenticación e Ingreso  Realizado Por: Fabricio Maldonado |

2.2.3.2. SERVICIO DE CONSULTA DE LABORATORIOS

El servicio se realizó mediante una consulta a las tablas usuarios enviando por servicio REST el correo institucional y la contraseña, para luego que el servidor de aplicaciones gestione la consulta y la convierta en un JSON para ser abierto y modelado por el dispositivo móvil

|  |
| --- |
| Servicio Web de Consulta de Laboratorio.  Ilustración 24: Servicio Web Consulta de Laboratorios  Realizado Por: Fabricio Maldonado |

2.2.4. DISEÑO DE CLASES

2.2.4.1. DIAGRAMA DE CLASES

El Diagrama de clases nos facilita ver la estructura de la aplicación móvil, con sus atributos, métodos y operaciones.

2.2.4.1.1. DIAGRAMA DE CLASES DE AUTENTICACIÒN E INGRESO Y NOTICIAS

|  |
| --- |
| Diagrama de Clases: Autenticación e Ingreso    Ilustración 25: Diagrama de Clases Autenticación e Ingreso  Realizado por: Fabricio Maldonado |

La aplicación móvil comienza cuanta con una animación inicial para proceder con el Login de la aplicación, la clase principal **MainActivity**llama a un objeto **ConfigConexion** para la autenticación del usuario.

La clase principal **MainActivity** autentifica el usuario por medio de servicios web e inicia la clase **InicioNoticias**, la cual, mediante servicios web, presenta la información de noticias recientes, esta clase carga los datos en forma asíncrona.

2.2.4.1.2. DIAGRAMA DE CLASES DE RESERVA DE LABORATORIOS

|  |
| --- |
| Diagrama de Clases de Reserva de Laboratorios    Ilustración 26: Diagrama de Clases de Reserva de laboratorios  Realizado por: Fabricio Maldonado |

**LaboratoriosInicio**es la clase inicial de la subsección Laboratorios, la cual es llamada por medio de la clase principal MainActivity, esta clase llama a dos clases las cuales son:

* **LaboratoriosReserva**:
* Esta clase filtra y busca los laboratorios según la necesidad del usuario la cual da paso a **LaboratoriosDisponible** para realizar la confirmación de la reserva de laboratorios
* **LaboratoriosVerReserva**:
* Esta clase visualiza los laboratorios reservado por el usuario.

2.2.4.1.2. DIAGRAMA DE CLASES DE DISPONIBILIDAD DE MATERIALES

|  |
| --- |
| Diagrama de Clases de Disponibilidad de Materiales    Ilustración 27: Diagrama de clases de Materiales  Realizado por: Fabricio Maldonado |

**MaterialesInicio** es la clase inicial de la subsección Materiales la cual es llamada por medio de la clase principal **MainActivity**.

La clase **MaterialesInicio** llama **MAterialesBusqueda** para proceder con la búsqueda de materiales y así mismo la disponibilidad de cada uno de ellos.

2.2.4.1.2. DIAGRAMA DE CLASES DE GESTION DE MANTENIMIENTO (INCIDENTES)

|  |
| --- |
| Diagrama de Clases de Gestión de Mantenimiento    Ilustración 28: Diagrama de Clases de Gestion de Mantenimiento  Realizado por: Fabricio Maldonado |

**MantenimeintoInicio** es la clase inicial de la subsección Mantenimiento la cual es llamada por medio de la clase principal **MainActivity**.

**MantenimientoInicio** llama a dos clases:

**MantenimientoAsignacion:** en la cual se ingresa un incidente de laboratorios y se asigna automáticamente al técnico designado en el horario específico.

**MantenimientoReportes:** esta clase genera reportes y seguimiento a los incidentes para que el usuario pueda ver el estado del incidente ingresado.

# CAPÍTULO 3

# 3. CONSTRUCCIÓN

Para el desarrollo de la aplicación móvil se consideran las 5 etapas mencionadas en el capítulo de análisis y diseño: Autenticación e ingreso, Noticias Actuales, Consulta y Reserva de Laboratorios, Consulta de disponibilidad de materiales y Gestión de Incidentes.

## CONSTRUCCIÓN DE LA APLICACIÓN MÓVIL

* + 1. ARQUITECTURA DE LA APLICACIÓN

|  |
| --- |
| Arquitectura de la Aplicación Móvil    Ilustración 29: Diagrama : Arquitectura de la APP Móvil  Realizado por: Fabricio Maldonado |

La aplicación móvil se conecta por medio de internet a un servidor de aplicaciones en el cual actualmente se levantan los servicios web necesarios para el funcionamiento de la misma.

## CONSTRUCCIÓN DE INTERFAZ Y FUNCIONALIDAD

El diseño de las interfaces de la aplicación está realizado mediante el estándar de Material Designer que propone una estructura de xml para los diferentes componentes de la aplicación

* + 1. CONSTRUCCIÓN DE LA AUTENTICACION E INGRESO

La interfaz se realizó con un modelamiento basado en xml el que dispone de los siguientes componentes:

|  |
| --- |
| Componentes utilizados en Autenticación e Ingreso    Ilustración 30: Componentes utilizados en Autenticación e Ingreso  Realizado por: Fabricio Maldonado |

La característica más importante es la asignación del id. Este identificador es el encargado de darle una entidad al componente para luego ser llamado por el controlador de la aplicación.

|  |
| --- |
| Interfaz de Autenticación e Ingreso    Ilustración 31: Interfaz de Autenticación e Ingreso  Realizado por: Fabricio Maldonado |

El botón iniciar sesión crea un hilo el cual gestiona él envió de datos y la respuesta de datos por parte del servidor, así mismo guarda el estado del radiobutton para luego ser almacenado.

|  |
| --- |
| Código de Inicio de sesión  public void onClick(View v) {  guardarestadoradiobutton(); Thread tr=new Thread(){ @Override public void run() {  Ilustración 32:Código de Inicio de Sesión  Realizado por: Fabricio Maldonado |

El Radio Button es almacenado en una variable de memoria interna, para poder identificar si el usuario ya tuvo un inicio de sesión y guardarlo.

|  |
| --- |
| Código Guardar estado de Inicio de Sesión  public void guardarestadoradiobutton(){  SharedPreferences prefernece= getSharedPreferences(*STRING\_PREFERENCE*,*MODE\_PRIVATE*); prefernece.edit().putBoolean(*PREFERENCE\_ESTADO\_BUTTON\_SECION*,rbsesion.isChecked()).apply(); }  Ilustración 33Código Guardar estado de Inicio de Sesión  Realizado por: Fabricio Maldonado |

Así mismo se guarda en la memoria interna el usuario y la contraseña de usuario que inició sesión.

|  |
| --- |
| Código Guardado en Variables de memoria interna.  SharedPreferences prefernece= getSharedPreferences(*STRING\_PREFERENCE*,*MODE\_PRIVATE*); prefernece.edit().putString("nombre",nombrecompleto).apply(); prefernece.edit().putString("nombrerol",nombre\_rol).apply();  Ilustración 34: Código Guardado en variables de memoria interna  Realizado por: Fabricio Maldonado |

Se envía el usuario y contraseña al servicio web

|  |
| --- |
| Código Guardado en Variables de memoria interna.  url = new URL("http://"+ip+"/serv/inicio.php?usu="+usu+"&pas="+pas+"");  HttpURLConnection conection= (HttpURLConnection)url.openConnection(); respuesta=conection.getResponseCode();  Ilustración 35: Código Guardado en variables de memoria interna  Realizado por: Fabricio Maldonado |

Si el servicio web retorna una cadena JSON el usuario existe, en caso que el servicio web retorne un valor nulo el usuario no existe por lo cual no podrá iniciar sesión.

* + 1. CONSTRUCCIÓN DE NOTICIAS

Para crear la sección de noticias se necesita 2 diferentes interfaces, una que consuma el servicio y una interfaz de estilo de la presentación de los datos.

Así mismo una interfaz que gestione la presentación del popup de noticias.

La estructura de la interfaz es:

|  |
| --- |
| Componentes utilizados en Noticias    Ilustración 36:Componenetes utilizados en Autenticación e Ingreso  Realizado por: Fabricio Maldonado |

|  |
| --- |
| Interfaz de Noticias    Ilustración 37:Interfaz de Noticias  Realizado por: Fabricio Maldonado |

Se extrae los datos mediante una petición al servicio web, el cual retorna los datos de las noticias asincrónicamente, así mismo para una próxima carga, los datos se guardan en memoria caché.

|  |
| --- |
| Llamada Asincrónica a el servicio web noticias.  AsyncHttpClient client = new AsyncHttpClient(); client.get("http://"+ip+"/serv/noticias.php", new AsyncHttpResponseHandler() {  Ilustración 38:Llamada Asincrónica a el servicio web noticias.  Realizado por: Fabricio Maldonado |

Para presentar la imagen en la pantalla se utiliza smartimageview la cual genera una imagen a partir de una ubicación y servicio web.

Para luego, tanto como los datos y las imágenes, sean estructuradas para su presentación en la interfaz.

|  |
| --- |
| Llamada Asincrónica a las imágenes en el servidor web  String urlfinal="http://"+ip+"/serv/img/"+image.get(position).toString(); Rect rect =new Rect (smartimageview.getLeft(), smartimageview.getTop(),smartimageview.getRight(),smartimageview.getBottom()); smartimageview.setImageUrl(urlfinal,rect);  Ilustración 39:Llamada Asincrónica a las imágenes en el servidor web  Realizado por. Fabricio Maldonado |

* + 1. CONSTRUCCIÓN DE MÓDULO LABORATORIO

El módulo de laboratorio está conformado por dos sub módulos los cuales fueron modelados con la siguiente estructura:

* Reserva de laboratorios
* Ver laboratorios.
  + - 1. CONTRUCION DE RESERVA LABORATORIO

|  |
| --- |
| Interfaz de Reserva de Laboratorios    Ilustración 40:Interfaz de Reserva de Laboratorios  Realizado por: Fabricio Maldonado |

Antes de hacer la reserva de laboratorios, se filtra la fecha, la hora, el número de máquinas y el software disponible por laboratorio. Para esto se utilizó calendar, una librería de google que hace uso del calendario del teléfono celular.

|  |
| --- |
| Llamada a DatePicker de Android  DatePickerDialog datePickerDialog = new DatePickerDialog(this, new DatePickerDialog.OnDateSetListener() { @Override public void onDateSet(DatePicker view, int year, int monthOfYear, int dayOfMonth) {  dfecha.setText(dayOfMonth+"/"+(monthOfYear+1)+"/"+year);  datePickerDialog.show();  Ilustración 41:Llamada a DatePicker de Android  Realizado por: Fabricio Maldonado |

Se llama a un servicio web para ver las horas y el software disponible por laboratorio.

* + - 1. CONTRUCCION DE VER LABORATORIOS

En este sub módulo se visualizan los laboratorios filtrados para continuar con la reserva según los requerimientos de usuario.

|  |
| --- |
| Interfaz de ver laboratorios    Ilustración 42:Interfaz de ver laboratorios  Realizado por: Fabricio Maldonado |

La estructura de la interfaz es:

|  |
| --- |
| Componentes utilizados en ver Laboratorios    Ilustración 43:Componentes utilizados en ver Laboratorios  Realizado por: Fabricio Maldonado |

Para visualizar los laboratorios reservados se debe implementar un ListView y sus respectivos ArrayList por cada dato visualizado en la interfaz:

|  |
| --- |
| Código Declaración de los Arrays para exponer en el ListView  private ListView listView;  ArrayList idreserv = new ArrayList(); ArrayList labreserv = new ArrayList(); ArrayList horareserv= new ArrayList(); ArrayList diareserv= new ArrayList(); ArrayList estadoreserv= new ArrayList(); ArrayList nombrereserv= new ArrayList(); ArrayList apellidoreserv= new ArrayList();  Ilustración 44:Código Declaración de los Arrays para exponer en el ListView  Realizado por: Fabricio Maldonado |

Para poder llamar al servicio web y estructurarlos en cada uno de los campos.

* + 1. CONSTRUCCIÓN DE MÓDULO MATERIALES

El modulo materiales, nos ayuda a buscar y verificar la disponibilidad de los materiales de préstamos en el CECASIS

|  |
| --- |
| Interfaz de Consulta de Materiales    Ilustración 45:Interfaz de Consulta de Materiales  Realizado por: Fabricio Maldonado |

La estructura de la interfaz es:

|  |
| --- |
| Componentes utilizados en Consulta de laboratorios    Ilustración 46:Componentes utilizados en ver Laboratorios  Realizado por: Fabricio Maldonado |

Para crear la búsqueda dinámica se estructuró a un EditText que recorre el servicio web y lo agrega en un ListView.

|  |
| --- |
| Código búsqueda de Materiales desde un ListView  adapter=new ArrayAdapter<String>(this,android.R.layout.*simple\_list\_item\_1*,titulo); listView.setAdapter(adapter);  buscador.addTextChangedListener(new TextWatcher() { @Override public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {   }  @Override public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) { adapter.getFilter().filter(charSequence); }  Ilustración 47: Código búsqueda de Materiales desde un ListView  Realizado por: Fabricio Maldonado |

* + 1. CONSTRUCCIÓN DE MÓDULO MANTENIMIENTO

El módulo de gestión de mantenimiento guarda un incidente de alguna maquina o máquinas de un laboratorio. La información es ingresada por un usuario.

El técnico personal de CECASIS, revisará el incidente, cambiará estado y escribirá el arreglo por medio de la aplicación móvil.

|  |
| --- |
| Interfaz de Ingreso de Incidente  Ilustración 48:Interfaz de Ingreso de Incidente  Realizado por: Fabricio Maldonado |

La estructura de la interfaz es:

|  |
| --- |
| Componentes utilizados en el ingreso de incidentes    Ilustración 49:Componentes utilizados en el ingreso de incidentes  Realizado por: Fabricio Maldonado |

La notificación de ingreso de nuevos incidentes se realiza mediante AlertDialog el cual ayuda a enviar la notificación al usuario indicado.

|  |
| --- |
| Código de AlertDialog para guardar el incidente.  AlertDialog.Builder mBuilder = new AlertDialog.Builder(MantenimientoAsignacion.this); // mBuilder.setTitle("Noticias");  View mView = getLayoutInflater().inflate(R.layout.*activity\_mantenimiento\_asignacionpop*,null);  mBuilder.setView(mView); mBuilder.setNegativeButton("Guardar", new DialogInterfaz.OnClickListener() { @Override public void onClick(DialogInterfaz dialogInterfaz, int i) {  guardarincidente(); } });  AlertDialog mDialog = mBuilder.create(); mDialog.show();  Ilustración 50:Código de AlertDialog para guardar el incidente.  Realizado por: Fabricio Maldonado |

# PRUEBAS DEL SOFTWARE

## SOFTWARE Y HARDWARE

Se realizó la prueba en:

Software:

* Windows Server 2016 Standard Evaluation (64 bits)

Hardware:

* Procesador: Common KVM porcessor 2.67 GHZ (2 procesadores)
* Ram: 9,77 GB

## PRUEBA DE RENDIMIENTO DE LA BASE DE DATOS

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Número de Usuarios (Pool) | Número de Transacción por (usuario) | Número de Transacciones | Número de datos en la tabla | Número de campos en la Tabla | Petición JDBC | Transacciones Completas | Rendimiento | %Error | Kb / seg | Seg, |
| 25 | 10 | 250 / s | 30000 | 37 | 10 | 10 x usr | 0.41 /seg | 0,00 % | 4848.11 | 14 seg |
| 25 | 20 | 500 / s | 30000 | 37 | 20 | 20 x usr | 0.47 / seg | 0.00 % | 5536.72 | 25 seg |
| 25 | 35 | 875 / s | 30000 | 37 | 35 | 35 x urs | 1.2 / seg | 0.00 % | 6345.56 | 28 seg |
| 25 | 50 | 1250 / s | 30000 | 37 | 50 | 50 x urs | 1.7 / seg | 50.00 % | 6038.47 | 29 seg |
| 25 | 100 | 2500/ s | 30000 | 37 | 25 | 25 x urs | 3.6 / seg | 75.00 % | 6319.42 | 27 seg |
| 50 | 10 | 500 / s | 30000 | 37 | 10 | 10 x urs | 0.352 / seg | 0.00 % | 4076.68 | 17 seg |
| 50 | 20 | 1000/ s | 30000 | 37 | 20 | 20x urs | 0.455 / seg | 0.00 % | 5259.19 | 26 seg |
| 50 | 35 | 1750 / s | 30000 | 37 | 35 | 35 x urs | 0.438 / seg | 0.00 % | 5063.77 | 48 seg |
| 50 | 50 | 2500/ s | 30000 | 37 | 50 | 50 x urs | 0.547 / seg | 0.00 % | 6321.78 | 55 seg |
| 50 | 100 | 5000 / s | 30000 | 37 | 50 | 50 x urs | 1.6 / seg | 50.00 % | 5610.48 | 1:02 seg |
| 75 | 10 | 750 / s | 30000 | 37 | 10 | 10 x urs | 0.27 / seg | 0.00 % | 3118.65 | 22 seg |
| 75 | 20 | 1500 / s | 30000 | 37 | 20 | 20 x urs | 0.398 / seg | 0.00 % | 4598.68 | 30 seg |
| 75 | 35 | 2625 / s | 30000 | 37 | 35 | 35 x urs | 0.41/ seg | 0.00 % | 4749.53 | 51 seg |
| 75 | 50 | 3750 / s | 30000 | 37 | 50 | 50 x urs | 0.51/ seg | 0.00 % | 5918.84 | 58 seg |
| 75 | 100 | 7500 / s | 30000 | 37 | 75 | 75 x urs | 1.0 / seg | 25.00 % | 5420.97 | 36 seg |
| 100 | 10 | 1000 / s | 30000 | 37 | 10 | 10 x urs | 0.23 / seg | 0.00 % | 2663.50 | 26 seg |
| 100 | 20 | 2000 / s | 30000 | 37 | 20 | 20 x urs | 0.342 / seg | 0.00 % | 3959.02 | 35 seg |
| 100 | 35 | 3500/ s | 30000 | 37 | 35 | 35 x urs | 0.42 / seg | 0.00 % | 4261.43 | 50 seg |
| 200 | 20 | 4000 /s | 30000 | 37 | 0 | 0 x urs | 5 min | 100 % | 922.80 | 5. min |

Tabla 6: Valores de la prueba de carga y rendimiento de la BDD

Realizado por Fabricio Maldonado

## PRUEBAS DE ACEPTACIÓN

Las pruebas de aceptación fueron realizadas para verificar la funcionalidad de la aplicación móvil y comprobar que realiza lo descrito en las historias de usuario, cada historia de usuario se ejecutará y demostrará las funciones de los módulos.

La prueba de aceptación estará formada de los siguientes puntos:

1. Caso de pruebas
2. Número de caso
3. Número de historia de usuario
4. Nombre de caso de prueba
5. Descripción
6. Condiciones de ejecución
7. Entradas
8. Resultado esperado
9. Evaluación

### PERFIL DE USUARIOS

1. Estudiante
2. Profesor
3. Administrador
4. Técnico del CECASIS

Características de los perfiles

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Participante | Nivel de Experiencia con Android | Perfil de usuario | Interfaz Táctil | Género | Edad |
| 1 | Estudiante | Alto | Sí | Hombre | 23 |
| 2 | Estudiante | Medio | Sí | Mujer | 24 |
| 3 | Estudiante | Bajo | Sí | Hombre | 20 |
| 4 | Profesor | Alto | Sí | Hombre | 47 |
| 5 | Administrador | Alto | Sí | Mujer | 49 |
| 6 | Técnico de CECASIS | Alto | Sí | Hombre | 35 |

Tabla 7:Perfiles de Usuario

Realizado por Fabricio Maldonado

### PRUEBA DE ACEPTACIÓN: LOGIN

|  |  |
| --- | --- |
| Pruebas de Aceptación : Login | |
| Caso de Prueba : | Numero de Historia de Usuario : 1 |
| Nombre de Caso de Prueba : Login | |
| Descripción:  La aplicación utiliza servicios web para reconocer el usuario y contraseña de usuario para lograr una conexión. | |
| Condiciones de Ejecución:  El usuario debe ingresar el correo y la contraseña de la Universidad Politécnica Salesiana, para ser consumida por un servicio Web el cual verifica en la base de datos. | |
| Entradas:  Correo institucional  Contraseña Institucional | |
| Resultado Esperado: Ingresar a la aplicación. | |
| Evaluación: Positiva. | |

Tabla 8:Prueba de Aceptación: Login

Realizado por: Fabricio Maldonado

* + 1. PRUEBAS DE ACEPTACIÓN: VISUALIZAR NOTICIAS.

|  |  |
| --- | --- |
| Pruebas de Aceptación : Visualizar Noticias | |
| Caso de Prueba : 02 | Numero de Historia de Usuario : 1 |
| Nombre de Caso de Prueba : Visualizar noticias | |
| Descripción:  La aplicación visualizara las ultimas noticias ingresadas por el administrador, para tener una mejor comunicación. | |
| Condiciones de Ejecución: Tener una sesión iniciada | |
| Entradas: Hilo del usuario y contraseña. | |
| Resultado Esperado: Visualizar las ultimas noticias ingresadas desde la interfaz WEB | |
| Evaluación: Positiva | |

Tabla 9: Pruebas de Aceptación ; Visualizar Noticias

Realizado por: Fabricio Maldonado

* + 1. PRUEBAS DE ACEPTACIÓN: VER DETALLE DE NOTICIAS.

|  |  |
| --- | --- |
| Pruebas de Aceptación : Ver detalle de noticias | |
| Caso de Prueba : 1 | Numero de Historia de Usuario : 2 |
| Nombre de Caso de Prueba : Ver detalle de noticias | |
| Descripción:  El módulo de noticias tiene un detalle el cual se lo puede ver dando un click en la noticia. | |
| Condiciones de Ejecución  Ingresar a una noticia de interés personal. | |
| Entradas: Interacción con el módulo de noticias. | |
| Resultado Esperado:  Visualizar de manera amigable las noticias y su descripción | |
| Evaluación: Positiva | |

Tabla 10: Pruebas de Aceptacion: Ver detalles de Noticias

Realizado por: Fabricio Maldonado

* + 1. PRUEBAS DE ACEPTACIÓN: LOGIN RECORDANDO CONTRASEÑA

|  |  |
| --- | --- |
| Pruebas de Aceptación: Login recordando contraseña | |
| Caso de Prueba : 1 | Numero de Historia de Usuario : 2 |
| Nombre de Caso de Prueba :Login recordando contraseña | |
| Descripción:  Ingresar al Módulo de noticias sin necesidad de un usuario y contraseña | |
| Condiciones de Ejecución:  Guardar previamente el Usuario y contraseña con Shared Preference en Android | |
| Entradas:  Usuario y contraseña en Shared preference | |
| Resultado Esperado:  Ingreso al módulo de noticias sin usuario y contraseña | |
| Evaluación: Positiva | |

Tabla 11: Pruebas de Aceptación: Login recordando Contraseña

Realizado por: Fabricio Maldonado

* + 1. PRUEBAS DE ACEPTACIÓN: BUSCAR Y VISUALIZAR LABORATORIOS DISPONIBLES

|  |  |
| --- | --- |
| Pruebas de Aceptación : Buscar y visualizar laboratorios disponibles | |
| Caso de Prueba: 1 | Numero de Historia de Usuario : 3 |
| Nombre de Caso de Prueba : Buscar y visualizar laboratorios Disponibles | |
| Descripción:  Con las variables seleccionadas buscara el laboratorio de acuerdo a las necesidades del usuario | |
| Condiciones de Ejecución:  Ingreso correcto de Fechas, horas, número de máquinas y software disponible. | |
| Entradas:  Variables correctamente ingresadas. | |
| Resultado Esperado: Visualizar los laboratorios disponibles con las variables ingresadas. | |
| Evaluación: Positiva | |

Tabla 12: Prueba de aceptación: Buscar y Visualizar Laboratorio disponibles

Realizado por: Fabricio Maldonado

### PRUEBAS DE ACEPTACIÓN: VISUALIZAR LOS MATERIALES DISPONIBLES

|  |  |
| --- | --- |
| Pruebas de Aceptación : Visualizar los materiales Disponibles | |
| Caso de Prueba : | Numero de Historia de Usuario: 3 |
| Nombre de Caso de Prueba : Visualizar los materiales Disponibles | |
| Descripción:  Visualizar los materiales disponibles en el cecasis | |
| Condiciones de Ejecución:  Mediante servicio web obtener los materiales disponibles para su verificación | |
| Entradas:  Ingreso al módulo extrayendo el usuario y contraseña de Shared preference | |
| Resultado Esperado: Visualizar los materiales disponibles | |
| Evaluación: Positiva | |

Tabla 13: Prueba de Aceptación: Visualizar los materiales Disponibles

Realizado por: Fabricio Maldonado

* + 1. PRUEBAS DE ACEPTACIÓN: INGRESAR INCIDENCIA

|  |  |
| --- | --- |
| Pruebas de Aceptación: Ingresar Incidencia | |
| Caso de Prueba : 1 | Numero de Historia de Usuario : 4 |
| Nombre de Caso de Prueba : Ingreso al módulo mantenimiento y registrar una incidencia | |
| Descripción:  Ingresar al módulo de registro de incidencias, y registrar la incidencia | |
| Condiciones de Ejecución:  Ingreso al módulo Mantenimiento, reconociendo en usuario y la contraseña | |
| Entradas:  Ingreso al módulo extrayendo el usuario y contraseña de Shared preference | |
| Resultado Esperado: Ingreso y registrar sin error | |
| Evaluación: Positiva | |

Tabla 14: Prueba de Aceptación: Ingresar Incidencia

Realizado por Fabricio Maldonado

* + 1. PRUEBAS DE ACEPTACIÓN: VER INCIDENCIAS

|  |  |
| --- | --- |
| Pruebas de Aceptación : Ver incidencia registradas | |
| Caso de Prueba 1 | Numero de Historia de Usuario 4 |
| Nombre de Caso de Prueba : Visualizar las incidencias reigstras | |
| Descripción : Entrar al módulo de Mantenimiento y ver incidencias registradas | |
| Condiciones de Ejecución : Ingreso al módulo Mantenimiento, reconociendo en usuario y la contraseña | |
| Entradas: Usuario y contraseña del Shared preference junto al servicio web consumido | |
| Resultado Esperado: Visualizar los valores adecuados según el usuario registrado. | |
| Evaluación: Positiva | |

Tabla 15: Pruebas de Aceptación: Ver incidencias

Realizado por Fabricio Maldonado

* + 1. PRUEBAS DE ACEPTACIÓN: VER ASIGNACION DE INCIDENCIAS

|  |  |
| --- | --- |
| Pruebas de Aceptación : Ver Asignación de incidencias | |
| Caso de Prueba 1 | Numero de Historia de Usuario 4 |
| Nombre de Caso de Prueba : Asignación de Incidencia | |
| Descripción:  Ingresar al módulo de mantenimiento con el perfil de técnico y visualizar las incidencias asignadas al usuario. | |
| Condiciones de Ejecución : Usuario y contraseña extraído del Shared preference | |
| Entradas: Usuario y tipo de Usuario | |
| Resultado Esperado: Visualizar las incidencias de acuerdo al usuario logueado | |
| Evaluación: Positiva. | |

Tabla 16: Prueba de Aceptacion: Ver Asignacion de Inicidencias

Realizado por Fabricio Maldonado

# CONCLUSIONES

* El levantamiento y optimización de los procesos con la notación BPMN del CECASIS proporciona de manera gráfica un estándar de procesos que es fácil entendible y legible para personas no expertas.
* El desarrollo de la aplicación móvil utilizando Scrum fue incremental interactiva y permitió obtener los requerimientos, análisis, diseño evolución y entrega.
* El número de usuarios conectados al servicio web por defecto son 100 con 75 transacciones por segundo
* La utilización de servicios web es una solución efectiva, debido a que, con esta tecnología, se consigue evitar la duplicidad al contar con una sola conexión y optimizar recursos al trabajar con datos semiestructurados.

# RECOMENDACIONES

* Utilizar dispositivos móviles Android con sistemas operativos superiores al 4 JellyBean en adelante para garantizar el correcto funcionamiento de la aplicación móvil.
* Utilizar Apache 2 y PHP5.7 para la implementación de los servicios web ya que cuenta con componentes específicos que no tienen versiones anteriores.
* Utilizar estándares de Material Designer para que la aplicación móvil tenga mejor usabilidad para el usuario final.
* Cargar las dependencias de Gradle localmente para que el IDE se más fluido en su trabajo.

# BIBLIOGRAFIA

Apache Friends. (2018). *¿Qué es XAMPP?* Obtenido de https://www.apachefriends.org/es/index.html

Gradle Inc. (2018). *Gradle Release Notes*. Obtenido de https://docs.gradle.org/current/release-notes.html

Apache Software Foundation. (2018). *Apache JMeter*. Obtenido de https://jmeter.apache.org/

Apache Software Foundation. (2018). *What is the Apache HTTP Server Project?* Obtenido de http://httpd.apache.org/ABOUT\_APACHE.html

Bizagi. (2018). *La Plataforma de Negocios Digitales*. Obtenido de https://www.bizagi.com/es

Debevoise, N. T. (2016). The MicroGuide to Process Modeling in BPMN. En N. T. Debevoise.

developer.android.com. (09 de 2017). *Andorid Studio Notes* . Obtenido de https://developer.android.com/studio/releases/

Evolus. (2018). *Pencil Proyect* . Obtenido de https://pencil.evolus.vn/

Google. (2018). *Material Designer para Android* . Obtenido de https://developer.android.com/design/material/?hl=es-419

Google. (2018). *Support Library Packages*. Obtenido de https://developer.android.com/topic/libraries/support-library/packages

Group, T. P. (s.f.). *Json\_encode*. Obtenido de http://php.net/manual/es/function.json-encode.php

Maner, W. (2018). *PROTOTIPADO*. Obtenido de https://www.sidar.org/recur/desdi/traduc/es/visitable/maner/Prototipado.htm

Nederlands, D. (2018). *Introducing JSON*. Obtenido de https://www.json.org/

Oracle. (2018). *Java Development*. Obtenido de http://www.oracle.com/technetwork/java/javase/10-0-1-relnotes-4308875.html

Quilumba, M. (2018). *AUTOMATIZACIÓN DEL PROCESO DE RESERVACIÓN DE LABORATORIOS, PRÉSTAMOS Y MANTENIMIENTO DE EQUIPOS TECNOLÓGICOS DEL CECASIS, A TRAVÉS DE UNA APLICACIÓN WEB.*

Riggs, S. (2018). *Postgresq*. Obtenido de https://www.postgresql.org/community/

Smith, J. (2016). *Android Asynchronous Http Client*. Obtenido de http://loopj.com/android-async-http/

Smith, J. (2016). *Android Smart Image View*. Obtenido de http://loopj.com/android-smart-image-view/

The PHP Group. (2018). *¿Qué es PHP?* Obtenido de http://php.net/manual/es/intro-whatis.php

The PHP Group. (2018). *Objetos de datos de PHP* . Obtenido de http://php.net/manual/es/book.pdo.php

Wang, X.-Y. (2016). PowerDesigner Chief Architect,. En X.-Y. Wang.

World Wide Web. (2016). *RESTful Web Services - Introduction*. Obtenido de https://www.tutorialspoint.com/restful/restful\_environment.htm