**UNIVERSIDAD POLITÉCNICA SALESIANA**

**SEDE QUITO**

**CARRERA:**

**INGENIERIA 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**

**Autores:**

**Fabricio Ricardo Maldonado Taipe**

**fmaldonadota@gmail.com**

**TUTOR:**

**Quito, marzo del 2018**

# RESUMEN

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

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 del desarrollo de software que se realizó de este proyecto.

Contenido

[RESUMEN 2](#_Toc516834659)

[INTRODUCCIÓN 5](#_Toc516834660)

[ANTECEDENTES 5](#_Toc516834661)

[PROBLEMA 6](#_Toc516834662)

[JUSTIFICACIÓN 6](#_Toc516834663)

[OBJETIVOS GENERAL 7](#_Toc516834664)

[OBJETIVO ESPECIFICO 7](#_Toc516834665)

[ALANCE DEL PROYECTO 7](#_Toc516834666)

[CAPITULO 1 8](#_Toc516834667)

[1 MARCO TEORICO 8](#_Toc516834668)

[1.2. HERRAMIENTAS DE DESARROLLO 8](#_Toc516834669)

[1.2.1. PLATAFORMAS DE DESARROLLO 8](#_Toc516834670)

[1.2.2. BASE DE DATOS 15](#_Toc516834671)

[1.2.2.1. POSTGRESQL 15](#_Toc516834672)

[1.2.3. DEPENDENCIAS Y LIBRERÍAS 16](#_Toc516834673)

[1.2.3.1. PDO 16](#_Toc516834674)

[1.2.3.2. JSON\_ENCODE 16](#_Toc516834675)

[1.2.3.3. QUERY 17](#_Toc516834676)

[1.2.4. SERVICIO WEB 19](#_Toc516834677)

[1.2.5. METODOLOGIA 20](#_Toc516834678)

[1.2.5.1. SPRING 20](#_Toc516834679)

[CAPÍTULO 2 21](#_Toc516834680)

[2. ANALISIS Y DISEÑO 21](#_Toc516834681)

[2.1. REQUERIMIENTOS INICIALES. 22](#_Toc516834682)

[2.1.1. REQUERIMIENTOS DE HARDWARE 22](#_Toc516834683)

[2.1.2. ESPECIFICACION DE REQUERIMIENTOS DE SOFWARE 23](#_Toc516834684)

[2.2.1. REQUERIMIENTOS ESPECIFICOS 24](#_Toc516834685)

[2.2.1.2. PROCESOS TO BE 27](#_Toc516834686)

[2.1.3. REQUERIMIENTOS NO FUNCIONALES 29](#_Toc516834687)

[2.1.4. REQUERIMIENTOS FUNCIONALES 30](#_Toc516834688)

[2.2. DISEÑO 39](#_Toc516834689)

[2.2.1. DISEÑO DE INTERFACES. 39](#_Toc516834690)

[2.2.3. DISEÑO DE INTERFACE DE RESERVA DE LABORATORIOS 41](#_Toc516834691)

[2.2.4. DISEÑO DE INTERFACE DE DISPONIBILIDAD DE MATERIALES 41](#_Toc516834692)

[2.2.5. DISEÑO DE INTERFACE DE GESTIÓN DE INCIDENTES. 42](#_Toc516834693)

[2.2.2. DISEÑO DE LA BASE DE DATOS 44](#_Toc516834694)

[CAPÍTULO 3 54](#_Toc516834695)

[3. CONSTRUCCIÓN 54](#_Toc516834696)

[3.1. CONSTRUCCIÓN DE LA APLICACIÓN MÓVIL 54](#_Toc516834697)

[3.2. CONSTRUCCIÓN DE INTERFACE Y FUNCIONALIDAD 55](#_Toc516834698)

[4. PRUEBAS DEL SOFTWARE 63](#_Toc516834699)

[4.1. SOFTWARE Y HARDWARE 63](#_Toc516834700)

[4.2. PRUEBA DE RENDIMIENTO DE LA BASE DE DATOS 64](#_Toc516834701)

[4.3. PRUEBAS DE ACEPTACIÓN 66](#_Toc516834702)

[4.3.1. PERFIL DE USUARIOS 66](#_Toc516834703)

[4.3.2. PRUEBA DE ACEPTACIÓN: LOGIN 67](#_Toc516834704)

[4.3.7. PRUEBAS DE ACEPTACIÓN: VISUALIZAR LOS MATERIALES DISPONIBLES 70](#_Toc516834705)

[CONCLUSIONES 73](#_Toc516834706)

[RECOMENDACIONES 73](#_Toc516834707)

[ANEXOS 74](#_Toc516834708)

[2.2.3. ETAPA DE CONSULTA – RESERVA DE LABORATORIOS 75](#_Toc516834709)

[2.2.4. ETAPA DE CONSULTA DE DISPONIBILIDAD DE MATERIALES 80](#_Toc516834710)

[2.2.5. ETAPA DE GESTIÓN DE INCIDENTES. 82](#_Toc516834711)

# INTRODUCCIÓN

## ANTECEDENTES

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

El centro de capacitaciones y servicios informáticos de la Universidad Politécnica Salesiana Campus Sur (CECASIS), ofrece una amplia gama de servicios a los estudiantes como también soporte preventivo y correctivo a las computadoras situadas en los laboratorios del CECASSIS pertenecientes a la Universidad Politecnica Salesiana campus SUR

Servicios del CECASIS

* Servicios de préstamo de hardware como en software para estudiantes internos y externos a la Universidad Politécnica Salesiana
* Servicio de capacitación tanto en hardware como en software para estudiantes internos y externos a la Universidad Politécnica Salesiana
* Mantenimiento preventivo como correctivo de las computadoras ubicadas en elñ centro de cómputo.
* Reparación y soporte de incidentes y problemas a las computadoras del centro de cómputo.

## PROBLEMA

Los estudiantes y profesores de la Universidad Politécnica Salesiana notan una 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 se lo haría 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, reducirá costos que apoyará a 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.

## OBJETIVOS GENERAL

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 y de este modo aportar el incremento de la eficiencia del grupo de trabajo.

## OBJETIVO ESPECIFICO

* 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.

## ALANCE 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 los cuales son:

* Gestión de Reservas: Este módulo gestionará, la reserva de laboratorios y el pedido de materiales, para los estudiantes y los docentes de la Universidad Politécnica Salesiana
* Mensajes a Mantenimiento: Este módulo se encargará de enviar un mensaje a soporte técnico del CECASIS y registrarlo para una próxima solución.
* Personalización: Este módulo se encargará de personalizar la aplicación al gusto del usuario final.

# CAPITULO 1

## 1 MARCO TEORICO

## 1.2. HERRAMIENTAS DE DESARROLLO

### 1.2.1. PLATAFORMAS DE DESARROLLO

#### 1.2.1.1. ANDROID STUDIO

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

Caracteristicas:

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

#### 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), es considerada como una solución a la gestión de procesos que permite ejecutar flujos de trabajo.

Bizagui automatiza procesos complejos.

#### 1.2.2.3. SAP POWERDESIGNER

PowerDesigner es una herramienta de modelado empresarial colaborativo que tiene soporte para diferentes tipos de modelos:

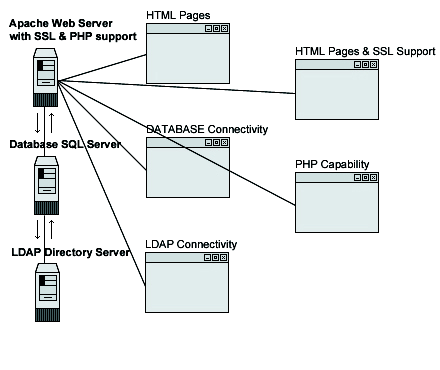
El más importarte para este proyecto es el Modelo de datos el cual funciona con la mayoría de sistemas de base de datos relacionales la cual vamos a utilizar para crear el modelo conceptual y el modelo físico de la base de datos.

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.

#### 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, e 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.

#### 1.2.2.5. XAMPP

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

Permite la instalación sencilla de Apache en el ordenador, sin importar tu sistema operativo, es una herramienta que permite probar tu trabajo en tu propio ordenar 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 libre.
* PHP y Perl: lenguajes de programación.
* ProFTPD: un servidor FTP.
* OpenSSL: para soporte a la capa de sockets segura.

#### 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.

#### 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, lo que distingue a PHP es que de lado del cliente como Java Script es ejecutado en el servidor, generando HTML y enviando al cliente.

* Existe tres campos principales donde se usan script de PHP
* Script del lado del servidor.
* Script desde la línea de comandos
* Escribir aplicaciones de escritorios

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

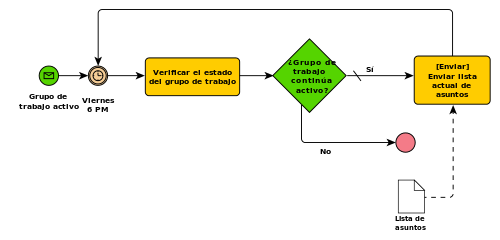
#### 1.2.2.8. JSON

JSON(Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos, JSON es un formato de texto que es completamente independiente.

JSON está constituido por dos estructuras:

* Una colección de pares de nombre/valor. Lo cual es conocido como objeto, registro, estructura, diccionario, etc.
* Una lista ordenada de valores, lo cual es conocido como arreglos, vectores, listas, etc

#### 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.

El principal objetivo de esta notación es proporcionar de manera gráfica un estándar de procesos que sea fácil entendible y legible para personas que no son expertos en el giro del negocio.

1.2.2.10. GRADLE

Gradle es un sistema de automatización de construcción de código abierto que introduce un lenguaje especifico de domingo (DSL) basado en Grovy en vez de la forma XML utilizada por Apache Maven.

Ventajas de Gradle en Android

* Permite reutilizar facilmente código.
* Hace sencilla la tarea de configurar y personalizar la compilación.
* Permite la distribución sencilla de código al resto del mundo, y fomenta el trabajo en equipo.
* Gestiona las dependencias de forma potente y cómoda (está basado en Maven).
* Permite la compilación desde consola, lo que nos puede hacer más sencilla la tarea de compilación en sistemas sin el entorno de desarrollo montado.
* Lo más importante es que hace increíblemente fácil la creación de diferentes versiones de la aplicación, por ejemplo para hacer múltiples versiones para móbiles o tables, versiones de pago o gratuitas

1.2.3.11. PENCIL PROYECT

Pencil Proyect es una herramienta de creación de prototipos de GUI de código abierto que está disponible para todas las plataformas.

Pencil está diseñado con el propósito de proporcionar una herramienta de creación de prototipos de GUI gratuita y de código abierto que las personas puedan instalar y usar fácilmente para crear maquetas en plataformas de escritorio populares.

1.2.3.12. MATERIAL DESIGNER

Material Design es una guía integral para el diseño visual, de movimientos y de interacción en distintas plataformas y dispositivos. Android ahora es compatible con las aplicaciones de Material Design.

En Android, se proporcionan los siguientes elementos que te permitirán crear aplicaciones en Material Design:

* Un tema nuevo
* Nuevos widgets para vistas complejas
* Nuevas API (interfaces de programación de aplicaciones) para sombras y animaciones personalizadas.
* Para obtener más información sobre la implementación de Material Design en Android, consulta Creación de aplicaciones con Material Design.
  + - 1. APACHE JMETER.

La aplicación Apache JMeter ™ es un software de código abierto, una aplicación Java 100% pura diseñada para cargar el comportamiento funcional de la prueba y medir el rendimiento.

Se puede usar para simular una gran carga en un servidor, grupo de servidores, red u objeto para probar su fortaleza o analizar el rendimiento general bajo diferentes tipos de carga.

### 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 y libre, publicado bajo la licencia PostgreSQL,1​ similar a la BSD o la MIT.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre o apoyados por organizaciones comerciales.

Características

* Algunas de sus principales características son, entre otras:
* Alta concurrencia
* Amplia variedad de tipos nativos
* PostgreSQL provee nativamente soporte para:
* Números de precisión arbitraria.
* Texto de largo ilimitado.
* Figuras geométricas (con una variedad de funciones asociadas).
* Direcciones IP (IPv4 e IPv6).
* Bloques de direcciones estilo CIDR.
* Direcciones MAC.
* Arrays.

### 1.2.3. DEPENDENCIAS Y LIBRERÍAS

### 1.2.3.1. PDO

La extensión Objetos de Datos de PHP (PDO por sus siglás en inglés) define una interfaz ligera para poder acceder a bases de datos en PHP. Cada controlador de bases de datos que implemente la interfaz PDO puede exponer características específicas de la base de datos, como las funciones habituales de la extensión. Se ha de observar que no se puede realizar ninguna de las funciones de las bases de datos utilizando la extensión PDO por sí misma; se debe utilizar un controlador de PDO específico de la base de datos para tener acceso a un servidor de bases de datos.

### 1.2.3.2. JSON\_ENCODE

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

Json\_encode — Retorna la representación JSON del valor dado

Ejemplos

Ejemplo #1 Un ejemplo de json\_encode()

<?php

$arr = array('a' => 1, 'b' => 2, 'c' => 3, 'd' => 4, 'e' => 5);

echo json\_encode($arr);

?>

El resultado del ejemplo sería:

{"a":1,"b":2,"c":3,"d":4,"e":5}

### 1.2.3.3. QUERY

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

1.2.3.4. BIBLIOTECA APPCOMPAT V7

Esta biblioteca agrega compatibilidad para el patrón de diseño de la interfaz de usuario de la barra de acciones. Esta biblioteca incluye compatibilidad con las implementaciones de la interfaz de usuario de material design.

A continuación, se muestran algunas de las clases claves incluidas en la biblioteca appcompat v7:

* ActionBar: proporciona una implementación del patrón de interfaz de usuario de la barra de acciones. Para obtener más información sobre el uso de la barra de acciones, consulta la guía del desarrollador sobre la barra de acciones.
* AppCompatActivity: agrega una clase de actividad de aplicación que puede usarse como una clase base para actividades que usan la implementación de la barra de acciones de la biblioteca de compatibilidad.
* AppCompatDialog: agrega una clase de diálogo que se puede usar como una clase base para diálogos temáticos de AppCompat.
* ShareActionProvider: agrega compatibilidad para una acción estándar de uso compartido (como correo electrónico o publicación en aplicaciones sociales) que se pueden incluir en una barra de acciones.

1.2.3.5. SUPPORT:DESIGN

Biblioteca de soporte de diseño

El paquete de diseño proporciona API para admitir la adición de componentes y patrones de diseño de materiales a sus aplicaciones.

La biblioteca de soporte de diseño agrega soporte para varios componentes y patrones de diseño de materiales para los desarrolladores de aplicaciones, como cajones de navegación, botones de acción flotante ( FAB ), barras de aperitivos y pestañas .

1.2.3.6. ANDROID-ASYNC-HTTP

Una biblioteca Http Client basada en devolución de llamada para Android

Un cliente Http basado en devolución de llamada asíncrono para Android construido sobre las bibliotecas HttpClient de Apache. Todas las solicitudes se realizan fuera de la secuencia de interfaz de usuario principal de su aplicación, pero cualquier lógica de devolución de llamada se ejecutará en el mismo subproceso que se creó mediante el uso de mensajes del Manejador de Android. También puede usarlo en el servicio o en el hilo de fondo, la biblioteca reconocerá automáticamente en qué contexto se ejecuta.

1.2.3.7. SMARTIMAGEVIEW

Esta dependencia permite que se Cargue imágenes de URL o contactos de Android con almacenamiento en caché

SmartImageView es un reemplazo ImageView para ImageView estándar de Android que además permite cargar imágenes desde las direcciones URL o la libreta de direcciones de contactos del usuario. Las imágenes se almacenan en caché en la memoria y en el disco para una carga súper rápida.

Características

* Reemplazo ImageView para ImageView
* Cargar imágenes desde una URL
* Cargar imágenes de la libreta de direcciones de contactos del teléfono
* Carga asíncrona de imágenes, la carga ocurre fuera del hilo de la interfaz de usuario
* Las imágenes se almacenan en caché en la memoria y en el disco para una carga súper rápida
* SmartImage clase SmartImage es fácilmente extensible para cargar desde otras fuentes

### 1.2.4. SERVICIO WEB

1.2.4.1. RESTFUL

Un servicio web es una colección de protocolos abiertos y estándares utilizados para intercambiar datos entre aplicaciones o sistemas. Las aplicaciones de software escritas en varios lenguajes de programación y que se ejecutan en varias plataformas pueden usar servicios web para intercambiar datos a través de redes informáticas como Internet de forma similar a la comunicación entre procesos en una sola computadora. Esta interoperabilidad (por ejemplo, entre Java y Python, o aplicaciones de Windows y Linux) se debe al uso de estándares abiertos.

* GET: proporciona acceso de solo lectura a un recurso.
* PUT: Usado para crear un nuevo recurso.
* ELIMINAR: Se usa para eliminar un recurso.
* POST: se usa para actualizar un recurso existente o crear un nuevo recurso.

### 1.2.5. METODOLOGIA

1.2.5.1. SCRUM

Scrum es una metodología de desarrollo ágil para software por lo cual es incremental interactiva, la cual es utilizada 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 Modelo de prototipos, en Ingeniería de software, pertenece a los modelos de desarrollo evolutivo. El prototipo debe ser construido en poco tiempo, usando los programas adecuados y no se debe utilizar muchos recursos.

El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles para el cliente o el usuario final. Este diseño conduce a la construcción de un prototipo, el cual es evaluado por el cliente para una retroalimentación; gracias a ésta se refinan los requisitos del software que se desarrollará. La interacción ocurre cuando el prototipo se ajusta para satisfacer las necesidades del cliente. Esto permite que al mismo tiempo el desarrollador entienda mejor lo que se debe hacer y el cliente vea resultados a corto plazo.

Este modelo es útil cuando el cliente conoce los objetivos generales para el software, pero no identifica los requisitos detallados de entrada, procesamiento o salida.

También ofrece un mejor enfoque cuando el responsable del desarrollo del software está inseguro de la eficacia de un algoritmo, de la adaptabilidad de un sistema operativo o de la forma que debería tomar la interacción humano-máquina

Se puede reutilizar el código.

La construcción de prototipos se puede utilizar como un modelo del proceso independiente, se emplea más comúnmente como una técnica susceptible de implementarse dentro del contexto de cualquiera de los modelos del proceso expuestos. Sin importar la forma en que éste se aplique, el paradigma de construcción de prototipos ayuda al desarrollado de software y al cliente a entender de mejor manera cuál será el resultado de la construcción cuando los requisitos estén satisfechos. De esta manera, este ciclo de vida en particular, involucra al cliente más profundamente para adquirir el producto.

# 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 PROPOSITO

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.



### 2.2.1. REQUERIMIENTOS ESPECIFICOS

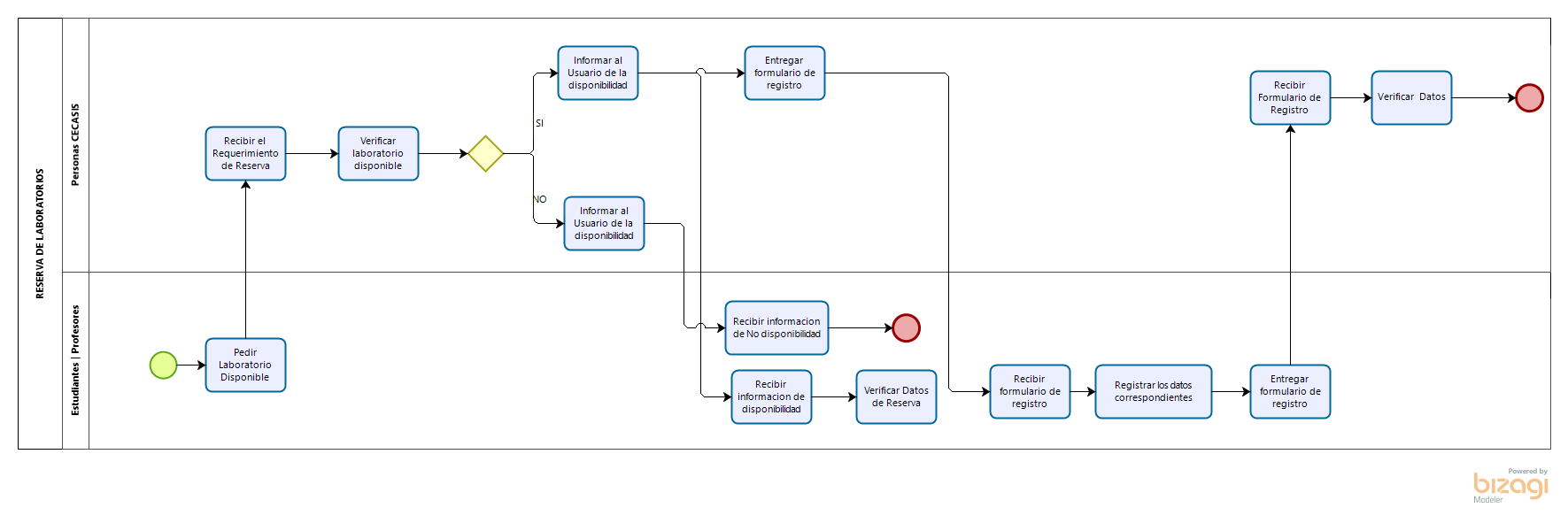
A continuación, se presenta los procesos funcionales del CECASIS:

#### 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

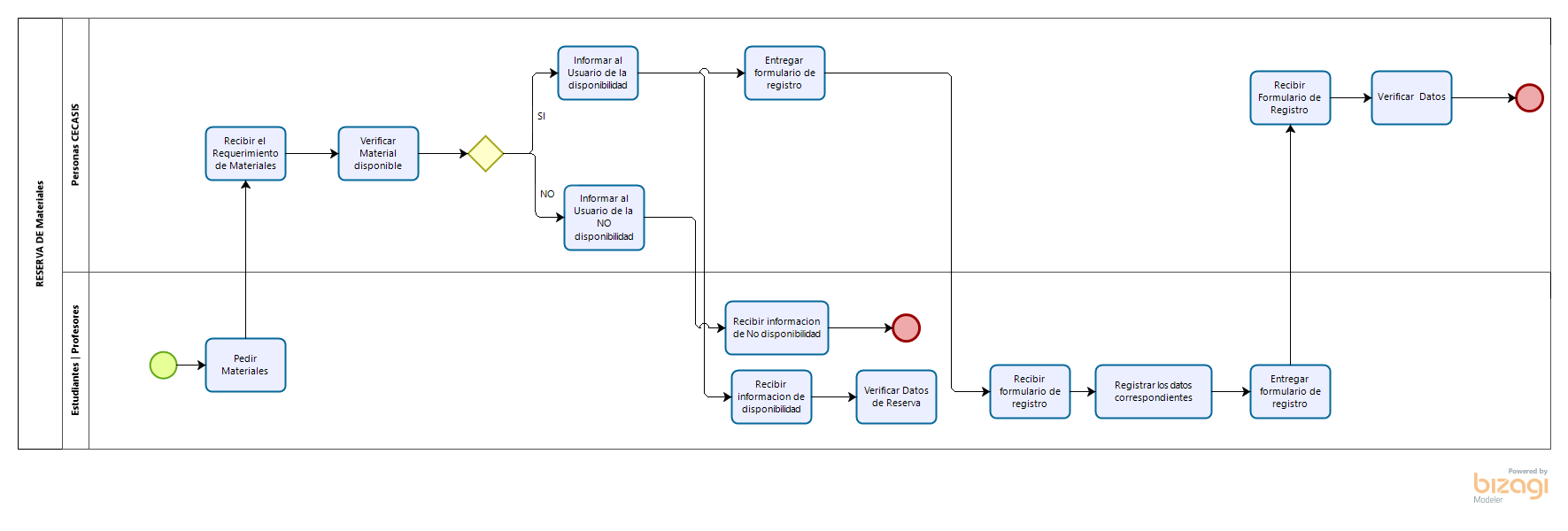
En la reserva de laboratorios se levanto junto al dueño del proceso en este caso con el personal del CECASIS para lo cual se utilizó la notación BPMN para modelar dicho proceso.



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 los cuales son necesarios automatizar.

#### 2.2.1.1.2. CONSULTA DE MATERIALES

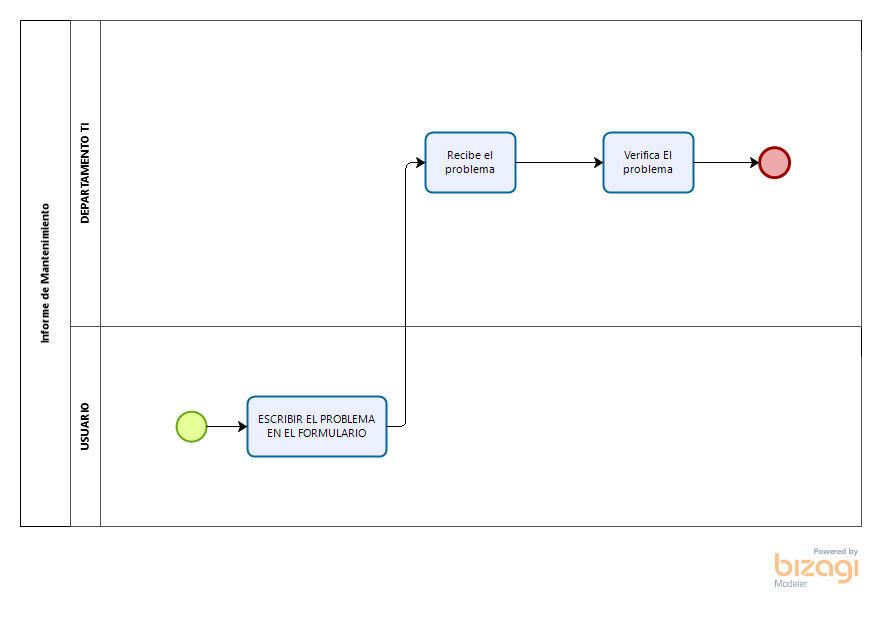
En la consulta de materiales se levantó junto al dueño del proceso en este caso con el personal del CECASIS para lo cual se utilizó la notación BPMN para modelar dicho proceso.



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

En el informe de mantenimiento se levantó junto al dueño del proceso en este caso con el personal del CECASIS para lo cual se utilizó la notación BPMN para modelar dicho proceso.



En el proceso notamos que se realiza un proceso manual por parte del Estudiante o Profesor al escribir el problema en el formulario, existe 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 Proceso TO BE o “como debe de ser” es el proceso ya optimizado el cual se debe automatizar.

Se maneja los siguientes roles:

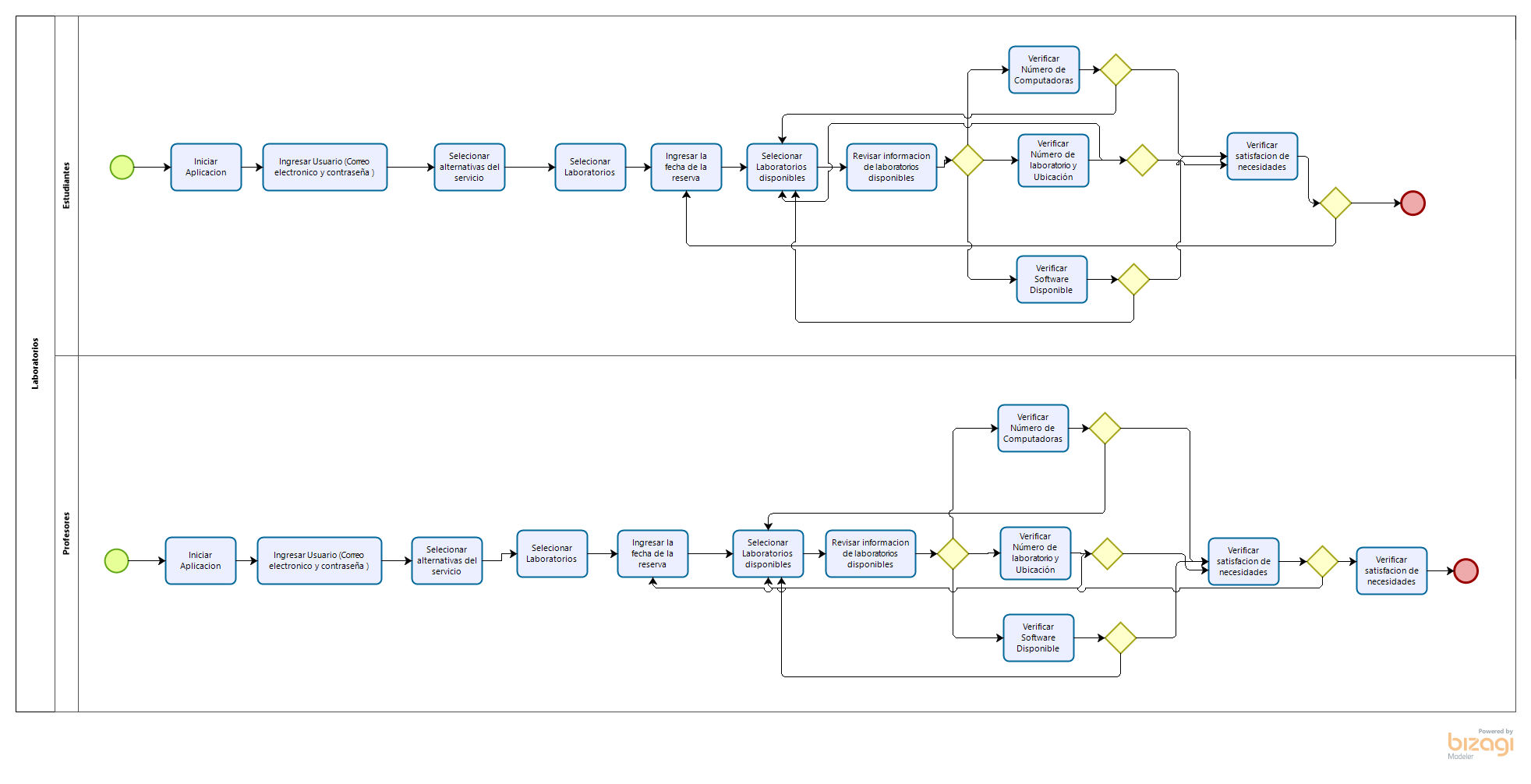
* Estudiante
* Docente
* Técnico
* Secretaria

#### 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.

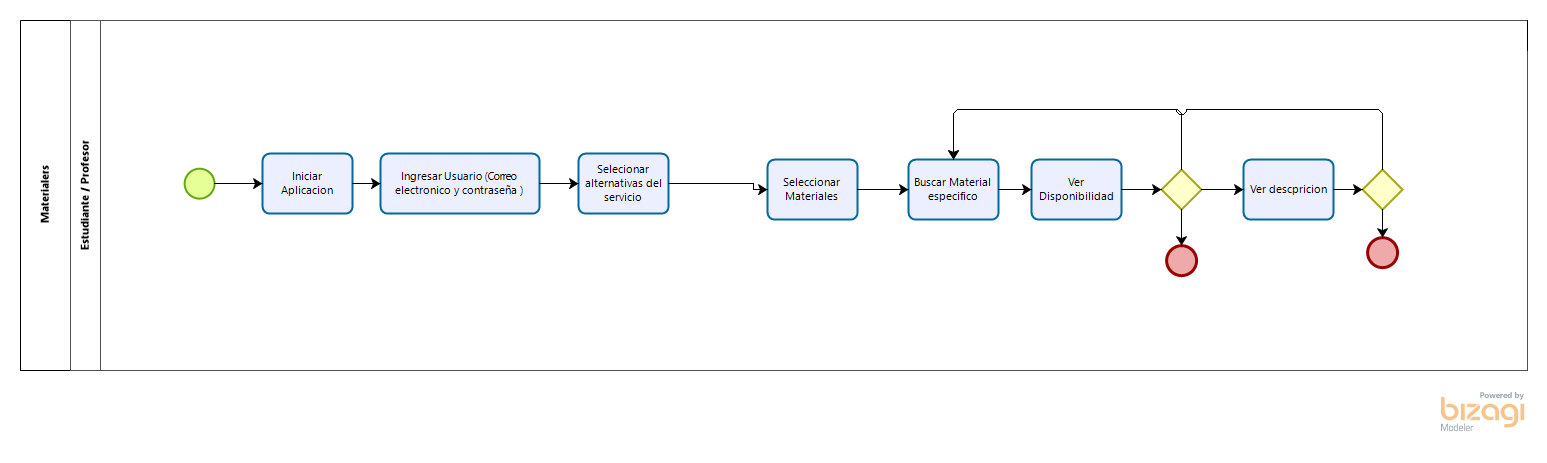
Docentes: Los docentes pueden ver la reserva del estudiante y tiene la potestad de aceptar o rechazar la reserva, así mismo puede hacer sus propias reservaciones.



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

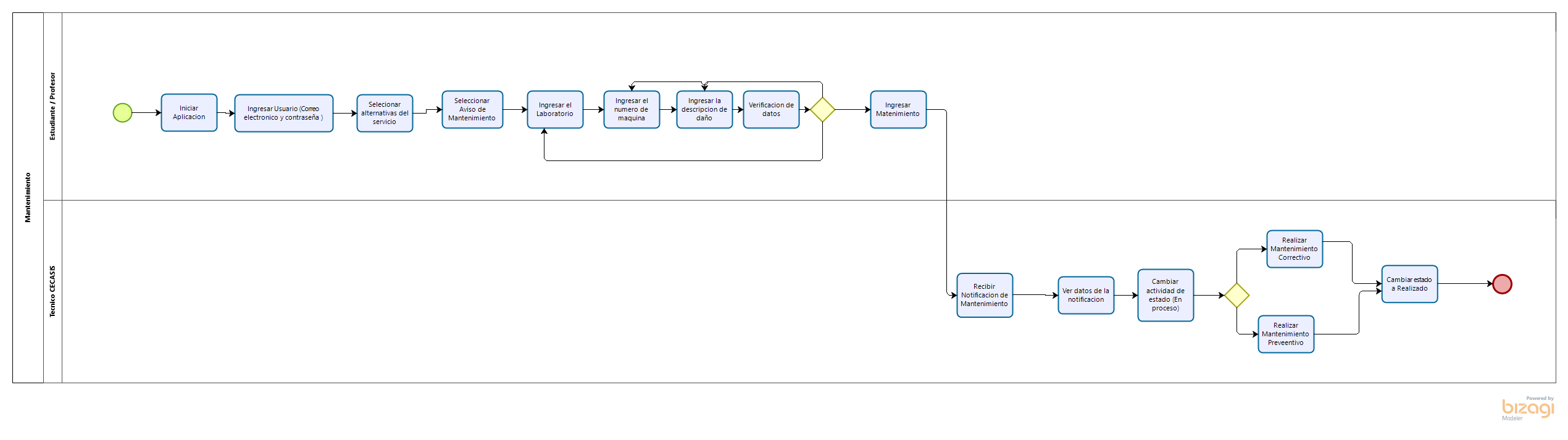
2.2.1.2.2. CONSULTA DE MATERIALES

En el informe de mantenimiento se levantó junto al dueño del proceso en este caso con el personal del CECASIS para lo cual se utilizó la notación BPMN para modelar dicho proceso.



2.2.1.2.3. GESTION DE MANTENIMIENTO

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



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 estar inicializado. |

La autenticación y el ingreso se lo realiza 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.



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 | Estar logueado y autentificado como docente de la Universidad Politécnica Salesiana |
| Post-condiciones | Instalación de la aplicación  El servidor de servicios debe estar inicializado.  Disponibilidad de laboratorios |



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 | Estar logueado y autentificado como docente o estudiante de la Universidad Politécnica Salesiana |
| Post-condiciones | Instalación de la aplicación  El servidor de servicios debe estar inicializado. |



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 | Estar logueado y autentificado como docente, estudiante, técnico o secretaria de la Universidad Politécnica Salesiana |
| Post-condiciones | Instalación de la aplicación  El servidor de servicios debe estar inicializado. |

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.



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 | Estar logueado y autentificado como docente, estudiante, técnico o secretaria de la Universidad Politécnica Salesiana |
| Post-condiciones | Instalación de la aplicación  El servidor de servicios debe estar inicializado. |



La consulta de materiales se lo realiza como método informativo para evitar el estar presencialmente preguntando 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 INTERFACE 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.

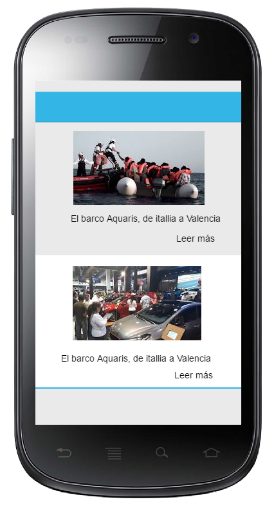


Una vez la autenticación sea correcta, o el usuario y contraseña sean correctos, la aplicación móvil automáticamente se dirigirá a una sección de noticias.

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 nombre de las noticias.

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.



### 2.2.3. DISEÑO DE INTERFACE DE RESERVA DE LABORATORIOS

La primera sección que se visualiza es la reserva de laboratorios en la cual existen dos subsecciones: Reserva de Laboratorios, Ver Laboratorios esto se lo realizo para mejor la usabilidad y experiencia del usuario.

La cual en la reserva de laboratorios procederemos a filtrar y reservar los laboratorios disponibles

Ver laboratorios se verificará los laboratorios reservados.



### 2.2.4. DISEÑO DE INTERFACE DE DISPONIBILIDAD DE MATERIALES

De la interface 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.



### 2.2.5. DISEÑO DE INTERFACE DE GESTIÓN DE INCIDENTES.



El diseño de la interface 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



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 tiene 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.



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

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



Se muestran en la imagen las tablas esenciales para el funcionamiento de la subsección Laboratorios la cual realiza el filtrado de laboratorio por fecha, numero 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.



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).

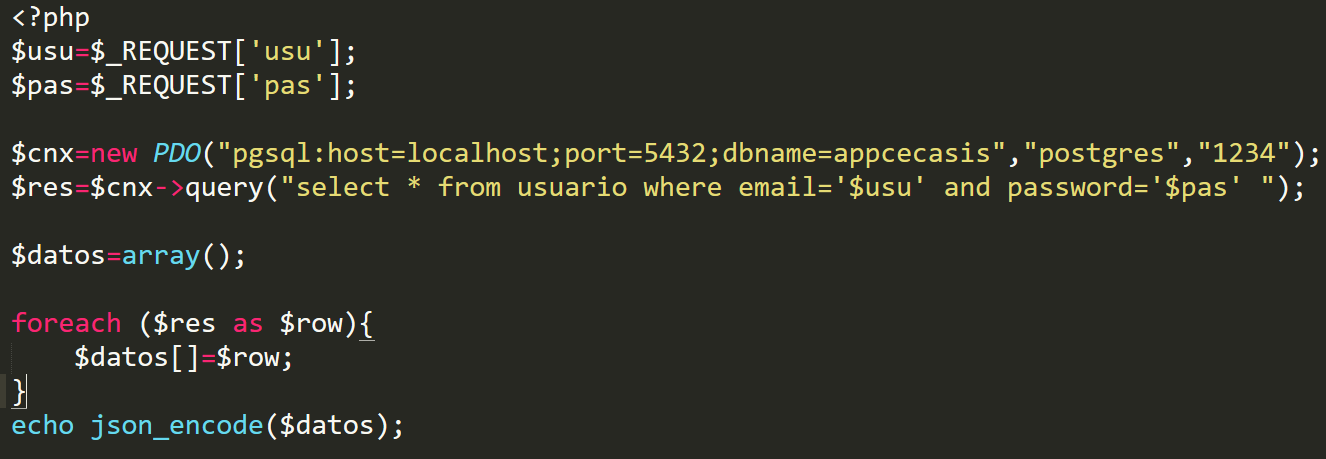


Se muestran en la imagen las tablas esenciales para el funcionamiento de la subsección Mantenimiento, la cual se encarga de registrar un incidente o daño de la maquina o máquinas de los laboratorios del CECASIS.

2.2.3. DISEÑO DEL SERVICIO WEB

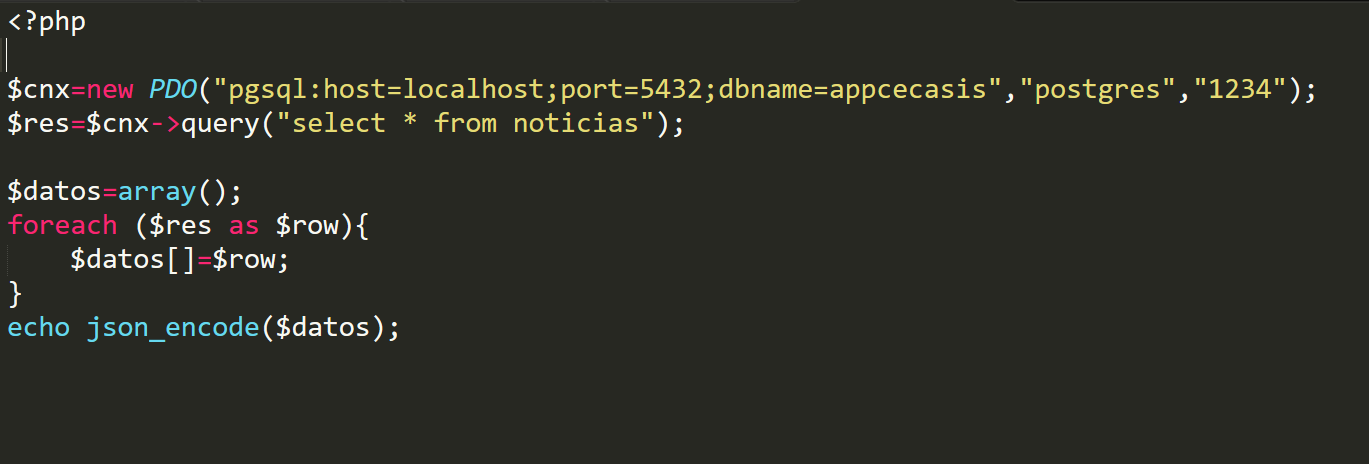
Los servicios web tienen que estar constantemente levantados para ser consumidos, así mismo para lograr la integración con la aplicación web la cual va a gestión de manera más precisa Laboratorios, Materiales e Incidentes.

#### 2.2.3.1. SERVICIO AUTENTICACIÓN DE USUARIOS

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

2.2.3.2. SERVICIO DE CONSULTA DE LABORATORIOS

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



2.2.4. DISEÑO DE CLASES

2.2.4.1. DIAGRAMA DE CLASES

Con 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 AUTENTICACION E INGRESO Y NOTICIAS



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



**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



**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)



**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 especifico.

**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 considera 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



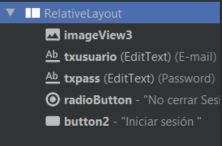
La aplicación móvil se conecta por medio de internet a un servidor de aplicaciones en el cual están levantado los servicios web necesarios para el funcionamiento de la aplicación móvil.

## CONSTRUCCIÓN DE INTERFACE Y FUNCIONALIDAD

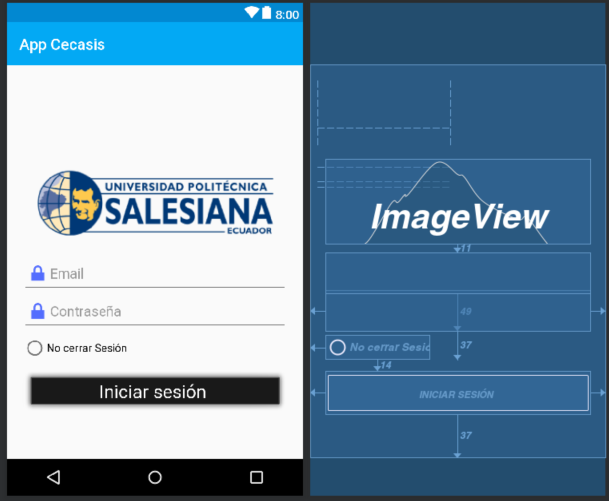
El diseño de las interfaces de la aplicación esta realizado mediante el estándar de Material Designer que propone una estructura de xml para los diferentes componentes de la aplicación

* + 1. CONSTRUCCIÓN DE LA AUTENTICACION E INGRESO

La interface se lo realizo con un modelamiento basado en xml los cuales disponen de los siguientes componentes:



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.



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

public void onClick(View v) {  
 guardarestadoradiobutton();  
 Thread tr=new Thread(){  
 @Override  
 public void run() {

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.

public void guardarestadoradiobutton(){  
 SharedPreferences prefernece= getSharedPreferences(*STRING\_PREFERENCE*,*MODE\_PRIVATE*);  
 prefernece.edit().putBoolean(*PREFERENCE\_ESTADO\_BUTTON\_SECION*,rbsesion.isChecked()).apply();  
}

Así mismo se guarda en la memoria interna el usuario, la contraseña de usuario que inicio sesión.

SharedPreferences prefernece= getSharedPreferences(*STRING\_PREFERENCE*,*MODE\_PRIVATE*);  
prefernece.edit().putString("nombre",nombrecompleto).apply();  
prefernece.edit().putString("nombrerol",nombre\_rol).apply();

Se envía el usuario y contraseña al servicio web

url = new URL("http://"+ip+"/serv/inicio.php?usu="+usu+"&pas="+pas+"");  
  
HttpURLConnection conection= (HttpURLConnection)url.openConnection();  
respuesta=conection.getResponseCode();

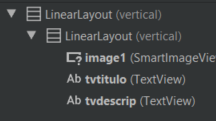
Si el servicio web retorna una cadena JSON, el usuario existe en caso de que retorne un valor nulo el usuario no existe por lo cual no podrá iniciar sesión.

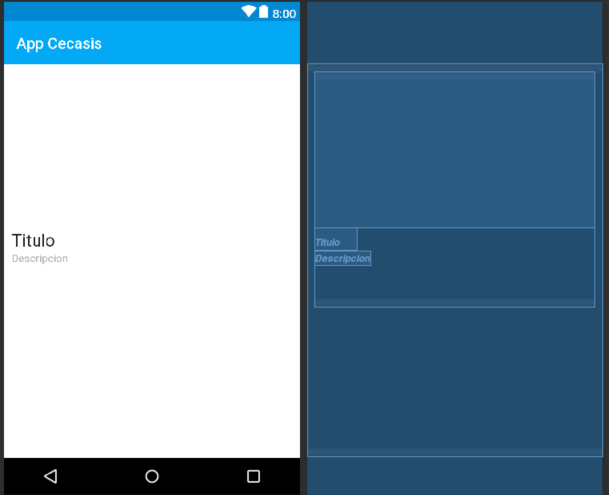
* + 1. CONSTRUCCIÓN DE NOTICIAS

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

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

La estructura de la interface es:





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

AsyncHttpClient client = new AsyncHttpClient();  
client.get("http://"+ip+"/serv/noticias.php", new AsyncHttpResponseHandler() {

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

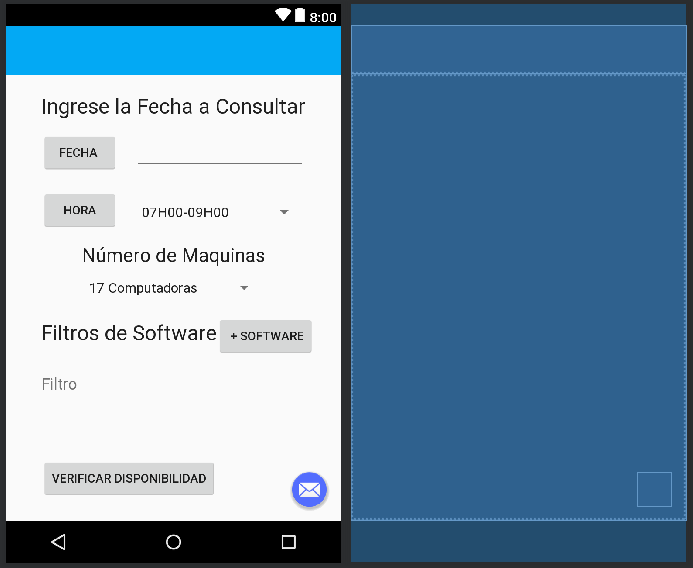
String urlfinal="http://"+ip+"/serv/img/"+image.get(position).toString();  
Rect rect =new Rect (smartimageview.getLeft(), smartimageview.getTop(),smartimageview.getRight(),smartimageview.getBottom());  
smartimageview.setImageUrl(urlfinal,rect);

Para luego tanto como los datos y las imagines sean estructuradas para su presentación en la interface.

* + 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



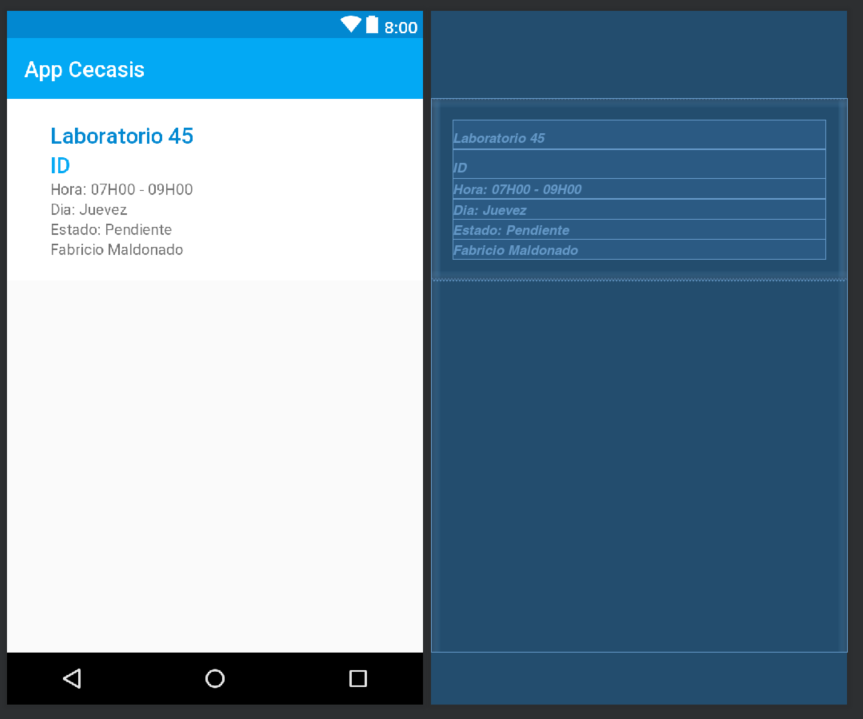
Antes de hacer la reserva de laboratorios, se filtra la fecha, la hora, el número de máquinas, y el software disponible por laboratorio para lo cual se utilizo calendar una librería de google que hace uso del calendario del teléfono celular.

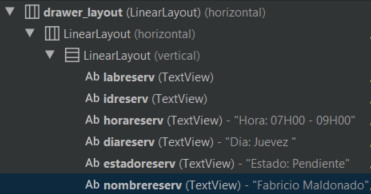
DatePickerDialog datePickerDialog = new DatePickerDialog(this, new DatePickerDialog.OnDateSetListener() {  
 @Override  
 public void onDateSet(DatePicker view, int year, int monthOfYear, int dayOfMonth) {  
  
 dfecha.setText(dayOfMonth+"/"+(monthOfYear+1)+"/"+year);

datePickerDialog.show();

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

* + - 1. CONTRUCCION DE VER LABORATORIOS

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

La estructura de la interface es:

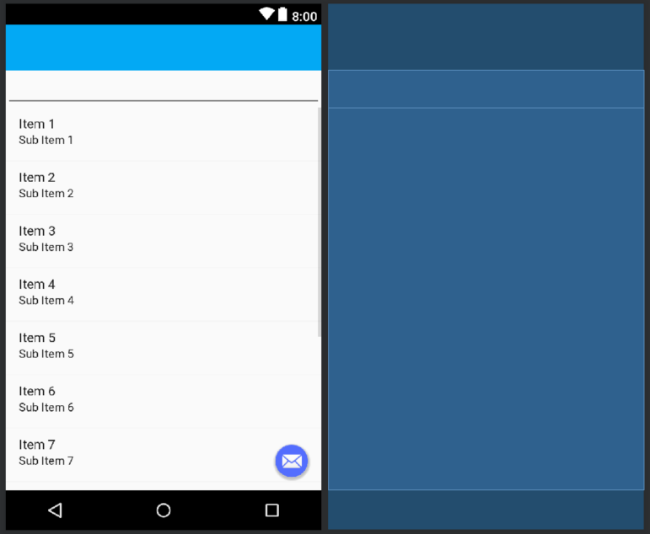
Para visualizar los laboratorios reservados se debe implementar un ListView y sus respectivos ArrayList por cada dato visualizado en la interface:

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();

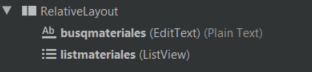
Para poder llamar al servicio web y estructurarlos en cada uno de los campos.

* + 1. CONSTRUCCIÓN DE MÓDULO MATERIALES

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



La estructura de la interface es:



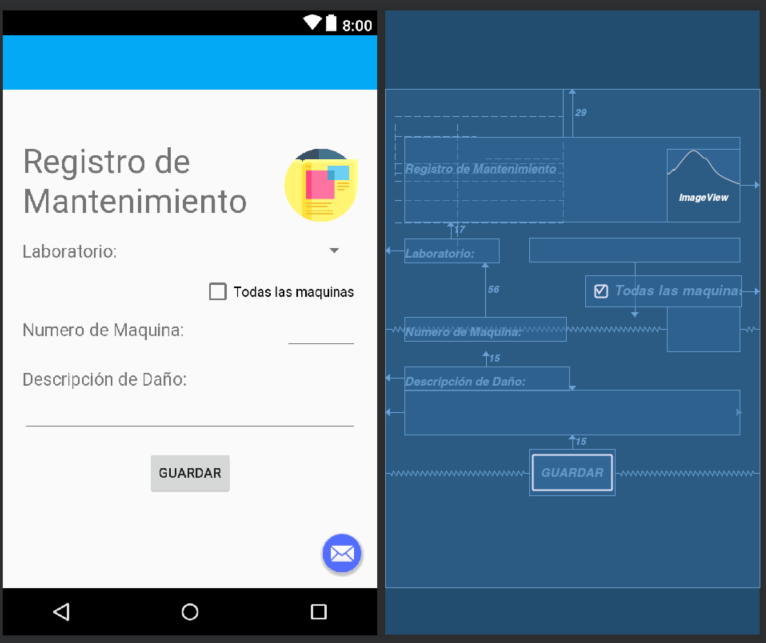
Para crear la búsqueda dinámica se estructuró a un EditText que recora el servicio web y lo agrege en un ListView.

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); }

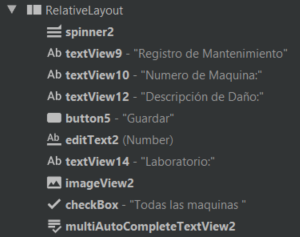
* + 1. CONSTRUCCIÓN DE MÓDULO MANTENIMIENTO

El módulo de gestión de mantenimiento guarda un incidente de algúna maquina o máquinas de un laboratorio, este es ingresado por un usuario.

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



La estructura de la interface es:



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

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 DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {  
 dialogInterface.dismiss();  
 }  
});  
  
AlertDialog mDialog = mBuilder.create();  
mDialog.show();

# 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 |

## PRUEBAS DE ACEPTACIÓN

Las pruebas de aceptación son 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 ejecutara y demostrar 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 Andorid | Perfil de usuario | Interface Táctil | Género | Edad |
| 1 | Estudiante | Alto | Si | Hombre | 23 |
| 2 | Estudiante | Medio | Si | Mujer | 24 |
| 3 | Estudiante | Bajo | Si | Hombre | 20 |
| 4 | Profesor | Alto | Si | Hombre | 47 |
| 5 | Administrador | Alto | Si | Mujer | 49 |
| 6 | Técnico de Cecasis | Alto | Si | Hombre | 35 |

### PRUEBA DE ACEPTACIÓN: LOGIN

|  |  |
| --- | --- |
| Pruebas de Aceptación : Login | |
| Caso de Prueba : | Numero de Historia de Usuario : 4 |
| 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. | |

* + 1. PRUEBAS DE ACEPTACIÓN: VISUALIZAR NOTICIAS.

|  |  |
| --- | --- |
| Pruebas de Aceptación : Visualizar Noticias | |
| Caso de Prueba : 02 | Numero de Historia de Usuario : 01 |
| 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 interface WEB | |
| Evaluación: Positiva | |

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

|  |  |
| --- | --- |
| Pruebas de Aceptación : Ver detalle de noticias | |
| Caso de Prueba : 1 | Numero de Historia de Usuario : 4 |
| 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 | |

* + 1. PRUEBAS DE ACEPTACIÓN: LOGIN RECORDANDO CONTRASEÑA

|  |  |
| --- | --- |
| Pruebas de Aceptación: Login recordando contraseña | |
| Caso de Prueba : 1 | Numero de Historia de Usuario : 4 |
| 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 | |

* + 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 : 4 |
| 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, numero de máquinas, y software disponible. | |
| Entradas:  Variables correctamente ingresadas. | |
| Resultado Esperado: Visualizar los laboratorios disponibles con las variables ingresadas. | |
| Evaluación: Positiva | |

### PRUEBAS DE ACEPTACIÓN: VISUALIZAR LOS MATERIALES DISPONIBLES

|  |  |
| --- | --- |
| Pruebas de Aceptación : Visualizar los materiales Disponibles | |
| Caso de Prueba : | Numero de Historia de Usuario |
| 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 | |

* + 1. PRUEBAS DE ACEPTACIÓN: INGRESO AL MÓDULO MANTENIMIENTO

|  |  |
| --- | --- |
| Pruebas de Aceptación: Ingreso al Módulo Mantenimiento | |
| Caso de Prueba : 1 | Numero de Historia de Usuario : 4 |
| Nombre de Caso de Prueba : Ingreso al módulo mantenimiento | |
| Descripción:  Ingresar al módulo laboratorios desde la interface inicial | |
| 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 sin error | |
| Evaluación: Positiva | |

* + 1. PRUEBAS DE ACEPTACIÓN: INGRESO NÚMERO DE LABORATORIO EN MODULO MANTENIMIENTO

|  |  |
| --- | --- |
| Pruebas de Aceptación : Ingreso Numero de laboratorio | |
| Caso de Prueba 1 | Numero de Historia de Usuario 4 |
| Nombre de Caso de Prueba : Ingreso número de laboratorio | |
| Descripción : | |
| Condiciones de Ejecución | |
| Entradas: | |
| Resultado Esperado: | |
| Evaluación: | |

* + 1. PRUEBAS DE ACEPTACIÓN: INGRESAR MAQUINA

|  |  |
| --- | --- |
| Pruebas de Aceptación : Ingresar Maquina | |
| Caso de Prueba 1 | Numero de Historia de Usuario 4 |
| Nombre de Caso de Prueba : Ingresar maquina | |
| Descripción:  Ingresar maquina donde se produjo el daño | |
| Condiciones de Ejecución : | |
| Entradas: | |
| Resultado Esperado: | |
| Evaluación: | |

# CONCLUSIONES

* El levantamiento y optimización de los procesos con la notación BPMN del CECASIS proporciona de manera gráfica un estándar de procesos que sea fácil entendible y legible para personas que no son expertos en el giro del negocio.
* Android Studio renderiza el diseño en tiempo real tiene una construcion con Gradle, diferentes soportes para desarrollar en Android Wear, Templetes para el diseño de interfaces, edición grafica de componentes y un emulador Andorid para ejecutar y probar aplicaciones lo cual facilita la creación y diseño de aplicaciones móviles.
* El desarrollo de la aplicación móvil utilizando scrum fue incremental interactiva para lo cual se utilizó para 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

Abiztar. (s.f.). Casos a incluir y casos a extender. Recuperado el 06 de octubre de 2017, de http://www.abiztar.com.mx/articulos/casos-a-incluir-casos-aextender.html Albasanz. (s.f.). Manual de XML. Recuperado el 18 de septiembre de 2017, de XML Que es?: http://www.mundolinux.info/que-es-xml.htm Alberto Basalo, M. A. (28 de Agosto de 2014). desarrolloweb.com. Obtenido de Qué es AngularJS: https://desarrolloweb.com/articulos/que-es-angularjsdescripcion-framework-javascript-conceptos.html Alvarez, M. A. (23 de Noviembre de 2009). desarrolloweb.com. Obtenido de CodeIgniter: https://desarrolloweb.com/articulos/codeigniter.html AngularJS. (s.f.). AngularJS. Recuperado el 9 de Febrero de 2017, de https://angularjs.org/ Apache. (s.f.). Apache JMeter. Recuperado el 18 de septiembre de 2017, de Apache JMeter ™: http://jmeter.apache.org/ Bahit, E. (diciembre de 2014). POO y MVC en PHP. Recuperado el 17 de sepriembre de 2017, de El paradigma de la Programación Orientada a Objetos en PHP y el patrón de arquitectura de Software MVC: http://www1.herrera.unt.edu.ar/biblcet/wpcontent/uploads/2014/12/eugeniabahitpooymvcenphp.pdf BBVAOPEN4U. (23 de marzo de 2016). BBVA API MARKET. Recuperado el 18 de septiembre de 2017, de API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos: https://bbvaopen4u.com/es/actualidad/api-rest-quees-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos Cardona, M. P. (14 de octubre de 2016). SEO Y SEM. Recuperado el 17 de septiembre de 2017, de Firebase, qué es y para qué sirve la plataforma de Google: http://www.iebschool.com/blog/firebase-que-es-para-que-sirve-la-plataformadesarroladores-google-seo-sem/ CEPE/ONU. (2012). Guia de Implementacion de la Facilitacion de Comercio. Recuperado el 11 de septiembre de 2017, de Proceso "Tal Como Está" (As is): http://tfig.unece.org/SP/contents/as-is-process.htm Cordova, J. (12 de Septiembre de 2016). Solución informática para mejorar la eficacia en el proceso de registro de evaluación y asistencia del instituto de educación superior tecnológico público Chocope, a partir del año 2015. Obtenido de Universidad Nacional de Trujillo: http://dspace.unitru.edu.pe/handle/UNITRU/7851 ddsutn. (2012). Introducción a arquitectura Web. Recuperado el 21 de Febrero de 2017, de Arquitectura Web: http://ddsutn.com.ar/cursos/cursadas- 98 anteriores/miercoles-a-la-noche2012/seguimientoclasesmienoche2012/introduccion-web develapps. (s.f.). Modelo Vista Presentador. Recuperado el 03 de agosto de 2017, de http://www.develapps.com/es/noticias/modelo-vista-presentador-mvp-enandroid Developers. (s.f.). Manifiesto de al app. Recuperado el 05 de agosto de 2017, de https://developer.android.com/guide/topics/manifest/manifestintro.html?hl=es-419 Diputacion Foral de BIzkaia. (Junio de 2006). Voluntariado Hechar una Mano. Recuperado el 5 de septiembre de 2017, de http://ibdigital.uib.es/greenstone/collect/cd2/index/assoc/responsa/ble1.dir/res ponsable1.pdf