

**DESARROLLO DEL APLICATIVO MÓVIL SOON PARA LA UNIVERSIDAD  
COOPERATIVA DE COLOMBIA SEDE BOGOTÁ**

**MANUAL DEL SISTEMA**

**Jonathan Alexander Hernández Ruiz**

**Sergio Alejandro Higuera Rojas**

**UNIVERSIDAD COOPERATIVA DE COLOMBIA**

**FACULTAD DE INGENIERÍA**

**PROGRAMA TECNOLOGÍA EN SISTEMAS**

**BOGOTÁ D.C.**

**2017**

**DESARROLLO DEL APLICATIVO MÓVIL SOON PARA LA UNIVERSIDAD  
COOPERATIVA DE COLOMBIA SEDE BOGOTÁ**

**Trabajo de grado en Modalidad de Análisis Sistemático de literatura  
Para optar por el título de  
Tecnólogo en Sistemas**

**Director**

**Ing. Jaime Páez.**

**Línea de investigación  
Desarrollo de Software**

**UNIVERSIDAD COOPERATIVA DE COLOMBIA  
FACULTAD DE INGENIERÍA  
PROGRAMA TECNOLOGÍA EN SISTEMAS  
BOGOTÁ D.C.**

**2017**

## TABLA DE CONTENIDO

<b>1. Descripción de la aplicación.....</b>	<b>4</b>
<b>2. Desarrollo de la aplicación.....</b>	<b>4</b>
<b>3. Recursos.....</b>	<b>10</b>
<b>4. Diseño lógico.....</b>	<b>11</b>
<b>5. Diseño gráfico.....</b>	<b>13</b>
<b>6. Requisitos del sistema.....</b>	<b>17</b>
<b>7. Instalación de la aplicación.....</b>	<b>17</b>
<b>8. Configuración de la aplicación.....</b>	<b>19</b>

## 1. Descripción de la aplicación

SOON es una aplicación totalmente gratuita para Smartphones Android versión 4.1 o posterior que necesita un espacio en memoria de 30 Mb. Cuenta con 3 módulos que permiten consultar información y ubicación de los distintos bloques de la Universidad Cooperativa de Colombia sede Bogotá, realizar el cálculo del valor de la matricula respecto al valor de los créditos, y realizar llamadas en caso de emergencia. La aplicación es nativa, por lo que las conexiones son rápidas y la interfaz está simplificada para el uso en el teléfono. Las funciones de la aplicación son ver un listado de los bloques con su descripción, características, transporte y ubicación como también una lista de programas de pregrado con información como el SNIES, duración, valor del crédito con su respectiva calculadora, y por ultimo una serie de pasos para realizar llamadas de emergencia.

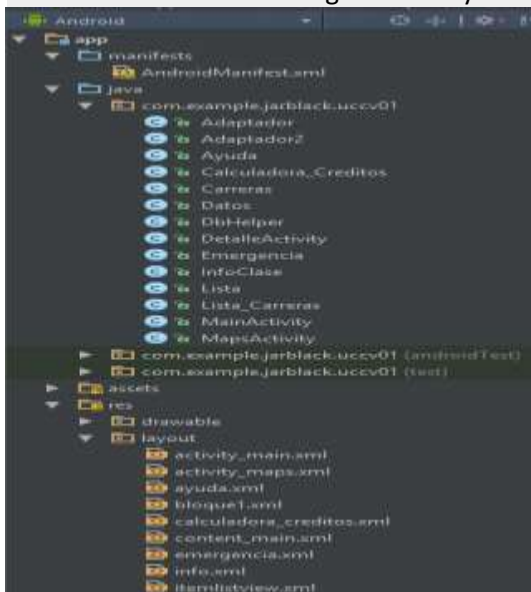
## 2. Desarrollo de la aplicación

Para el desarrollo de la aplicación fue necesario acudir a software web y de escritorio muy reconocido para crear un producto de calidad que cumpliera todos los estándares, a continuación se detalla cada uno de ellos:

### Android Studio V2.33 API 25:

Es un nuevo entorno de desarrollo integrado para el sistema operativo Android lanzado por Google, diseñado para ofrecer nuevas herramientas para el desarrollo de aplicaciones y alternativa al entorno Eclipse, hasta ahora el IDE más utilizado.

Al crear un nuevo proyecto en Android Studio, la estructura del proyecto aparece con casi todos los archivos dentro del directorio SRC, un cambio a un sistema de generación basado Gradle que proporcionará una mayor flexibilidad para el proceso de construcción. Además, gracias a su sistema de emulación integrado, Android Studio permite ver los cambios que realizamos en nuestra aplicación en tiempo real, pudiendo además comprobar cómo se visualiza en diferentes dispositivos Android con distintas configuraciones y resoluciones de forma simultánea.



Android Studio tiene un Modelo vista controlador donde podemos encontrar una Carpeta llamada Java donde solo irán las clases o actividades que estarán del lado del servidor (Backend) y otra carpeta de recursos donde irán todos los recursos necesarios para que el usuario pueda visualizar el diseño gráfico en pantalla (Frontend)

En Android Studio los controladores se denominan actividades, las cuales nos permiten darle funcionalidad a todo lo que se haya desarrollado en el frontend en especial los layouts, sin una actividad el layout será simplemente una vista. Estas clases deben ser programadas en Java y para que sean actividades deben heredar de la clase AppCompatActivity que ya viene por defecto en una de las librerías de Android.

```
public class MainActivity extends AppCompatActivity {

    private Toolbar toolbar;
    private Button ingresar, carreras, emergencia;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

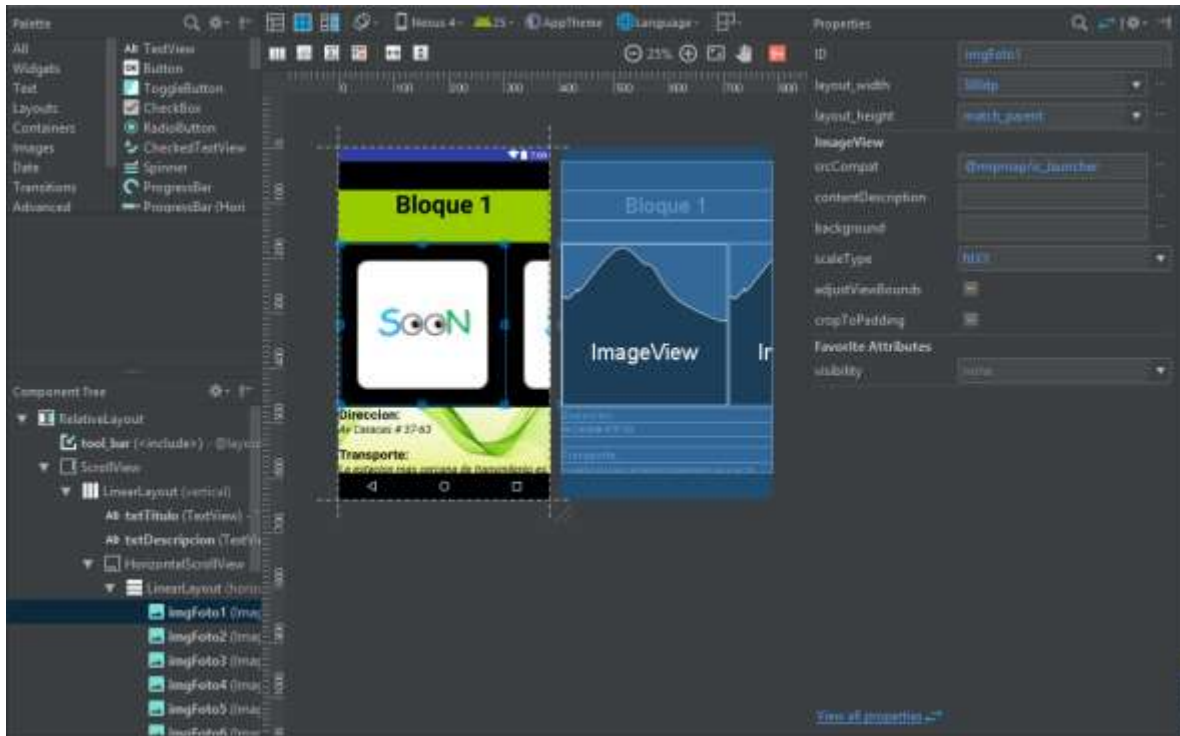
        toolbar=(Toolbar) findViewById(R.id.tool_bar);
        setSupportActionBar(toolbar);

        ingresar = (Button) findViewById(R.id.ingresar);
        carreras=(Button) findViewById(R.id.creditos);
        emergencia=(Button) findViewById(R.id.emergencia);

        ingresar.setOnClickListener((arg0) → { ejecutar_lista(null); });
        carreras.setOnClickListener((v) → {
            Intent intent=new Intent(MainActivity.this,Lista_Carreras.class);
            startActivity(intent);
        });

        emergencia.setOnClickListener((v) → {
            Intent intent=new Intent(MainActivity.this,Emergencia.class);
            startActivity(intent);
        });
    }
}
```

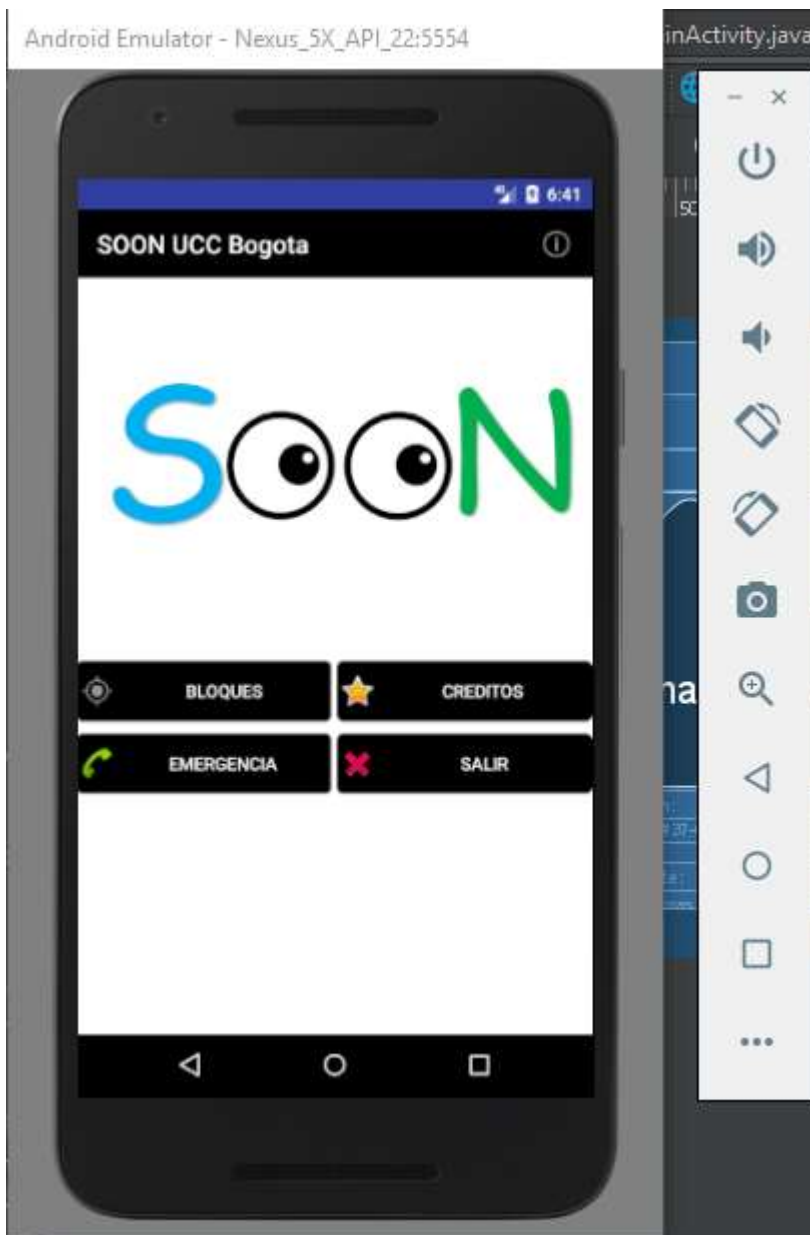
Para el diseño de los layouts Android Studio versión 2.33 nos ofrece una interfaz gráfica en la cual podemos organizar nuestros elementos que van a ser visualizados en el dispositivo móvil por el usuario. Por un lado tenemos una paleta de diseño donde podemos encontrar distintas opciones de gráficos para añadir a nuestro proyecto, por otros dos simuladores de pantalla, uno para ver como se ve y el otro para saber cómo está organizado. Por ultimo en la parte derecha nos encontramos con las propiedades del elemento que estemos manipulando.



Si el usuario lo desea puede también diseñar el layout con código xml, sin embargo cuando diseñamos el layout con la ayuda gráfica, Android Studio genera automáticamente el código xml.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/back4"
    app:layout_behavior="android.support.design.widget.AppBarLayout$ScrollingView..."
    tools:layout_editor_absoluteX="0dp"
    tools:layout_editor_absoluteY="81dp">
```

Android Studio viene con un emulador incorporado donde podemos diseñar un celular con las características que necesitemos para ejecutar la aplicación, también es posible utilizar un celular real para llevar a cabo estas pruebas de compilación.



### SQLite:

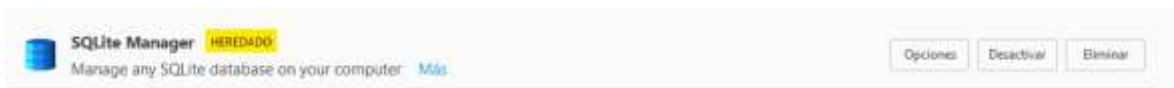
Es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña (~275 kiB)<sup>2</sup> biblioteca escrita en C. SQLite es un proyecto de dominio público<sup>1</sup> creado por D. Richard Hipp.

A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción.

Para el manejo de la base de datos SQLite diseñada para contener la información de la aplicación SOON se utilizó el navegador Mozilla Firefox



En Firefox existe una extensión que podemos instalar llamada SQLite Manager para la gestión de bases de datos en SQLite



El SQLite Manager aunque tiene funcionalidades básicas tiene lo necesario para gestionar bases de datos de este tipo. Toda base de datos creada para Android Studio tiene que tener obligatoriamente una tabla denominada `android_metada` de lo contrario es muy probable que no funcione.



SQLite Manager - C:\Users\Jarblack\Desktop\CarrerasUcc.db

Base de Datos Tabla Índice Vista Trigger Herramientas Ayuda

Carpeta (Seleccionar Perfil Base de Datos) Ir

Estructura Explorar & Buscar Ejecutar SQL Configuración DB

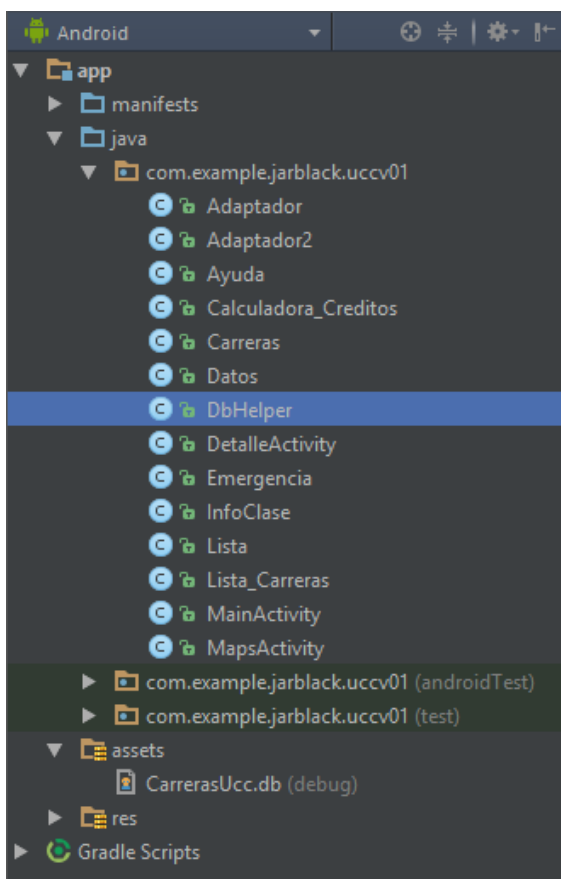
TABLE Carreras Buscar (2) Mostrar Todo

id	Carrera	Creditos	Valor_Credito	Cantidad_Semestr...	Ciudad	SNIES
1	Administración de...	143	169019.882	8	Bogotá	15111
2	Comercio Internac...	152	169910.118	8	Bogotá	3527
3	Comunicación So...	165	178125.882	8	Bogotá	4115
4	Contaduría públic...	144	150341.647	8	Bogotá	144
5	Derecho	158	220381.765	10	Bogotá	2722
6	Economía	151	180063.352	8	Bogotá	1821
7	Ingeniería Ambien...	168	209991.765	10	Bogotá	103670
8	Ingeniería Electron...	159	204837.765	10	Bogotá	54474
9	Ingeniería Industri...	167	201168.941	10	Bogotá	32023
10	Ingeniería de Siste...	162	198151.529	10	Bogotá	3314
11	Ingeniería de Tele...	164	198000	10	Bogotá	54475
12	Mercadeo	144	168967	8	Bogotá	1622
13	Psicología	147	198789.647	8	Bogotá	4064

1 para 13 de 13

SQLite 3.19.3 GeoStudio 56.0.2 (U.S.S.T.) signed Exclusive Número de archivos en la carpeta seleccionada: 8 ET: 3 ms

Para utilizar la base de datos en Android Studio, se debe importar a una carpeta llamada Assets y configurar en una clase que en este caso llamamos DBHelper



## **Adobe Photoshop**

Es un editor de gráficos rasterizados desarrollado por Adobe Systems Incorporated. Usado principalmente para el retoque de fotografías y gráficos, su nombre en español significa literalmente "taller de fotos". Es líder mundial del mercado de las aplicaciones de edición de imágenes y domina este sector de tal manera que su nombre es ampliamente empleado como sinónimo para la edición de imágenes en general.

*Referencia de <https://www.adobe.com/la/products/photoshop/free-trial-download.html>*

## **StarUML**

El lenguaje unificado de modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el Object Management Group (OMG).

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

Es importante remarcar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

Se puede aplicar en el desarrollo de software gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional, Rational Unified Process o RUP), pero no especifica en sí mismo qué metodología o proceso usar.

## **3. Recursos:**

### **Hardware:**

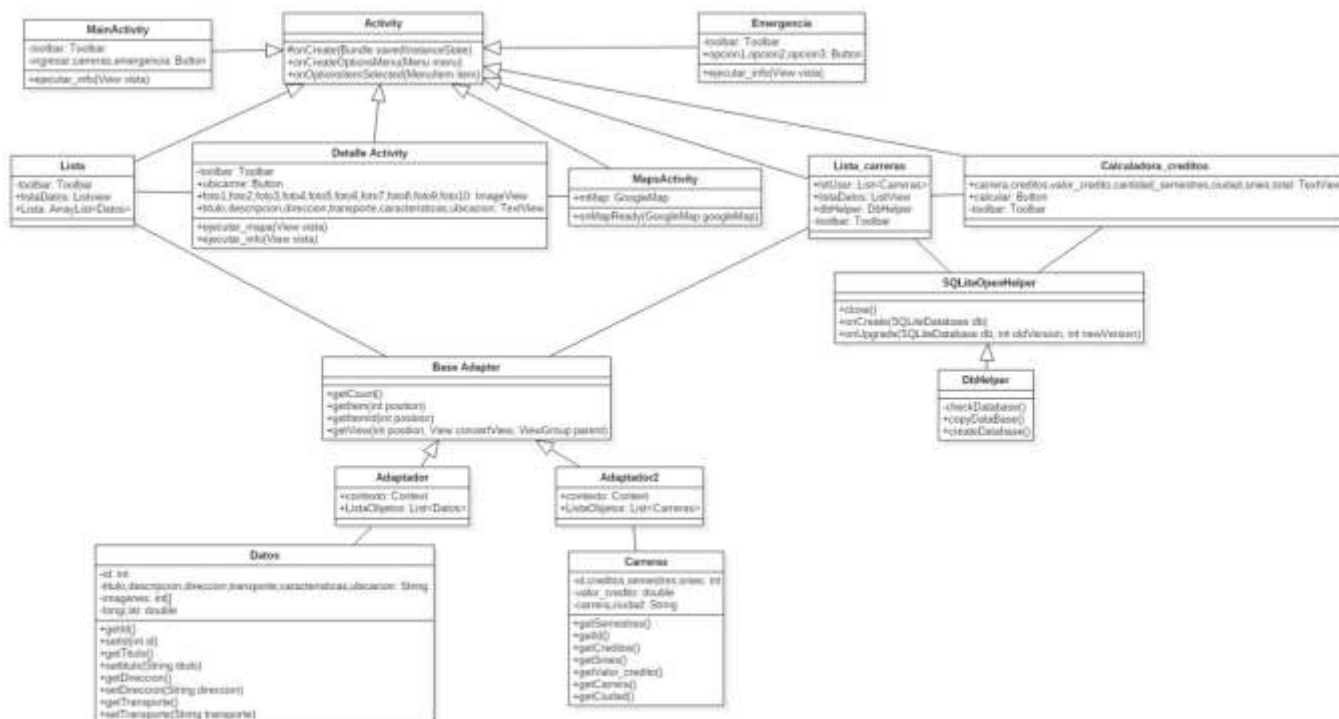
- Portátil Toshiba Intel Core i5 cuarta generación 2.5GHz con 6Gb de RAM y 750Gb de disco duro.
- Portátil Asus Amd A10-8700P Radeon R6 1.80GHz 8gb de RAM y 1Tb de disco duro.
- 

### **Software:**

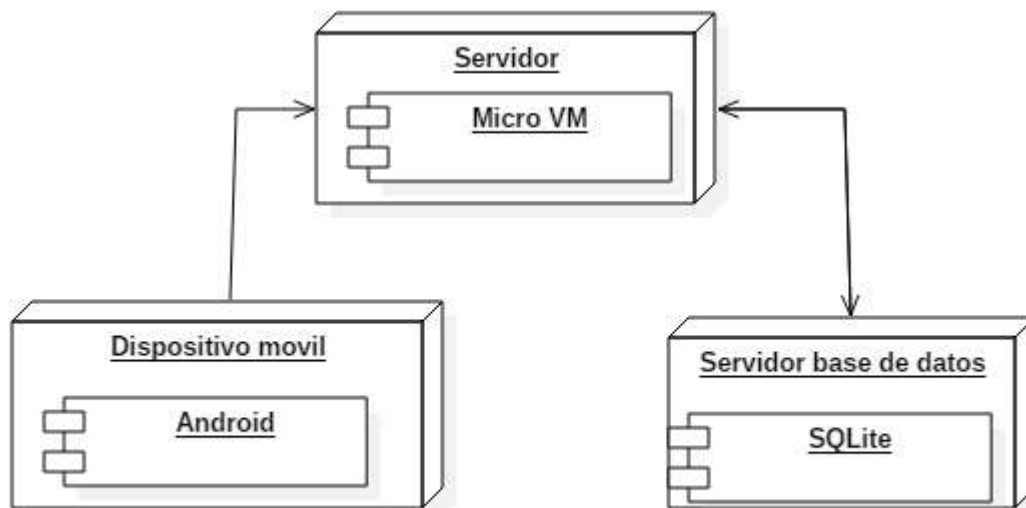
- Windows 10.
- Android Studio
- Photoshop.
- SQLite.
- StarUML

## 4. Diseño lógico

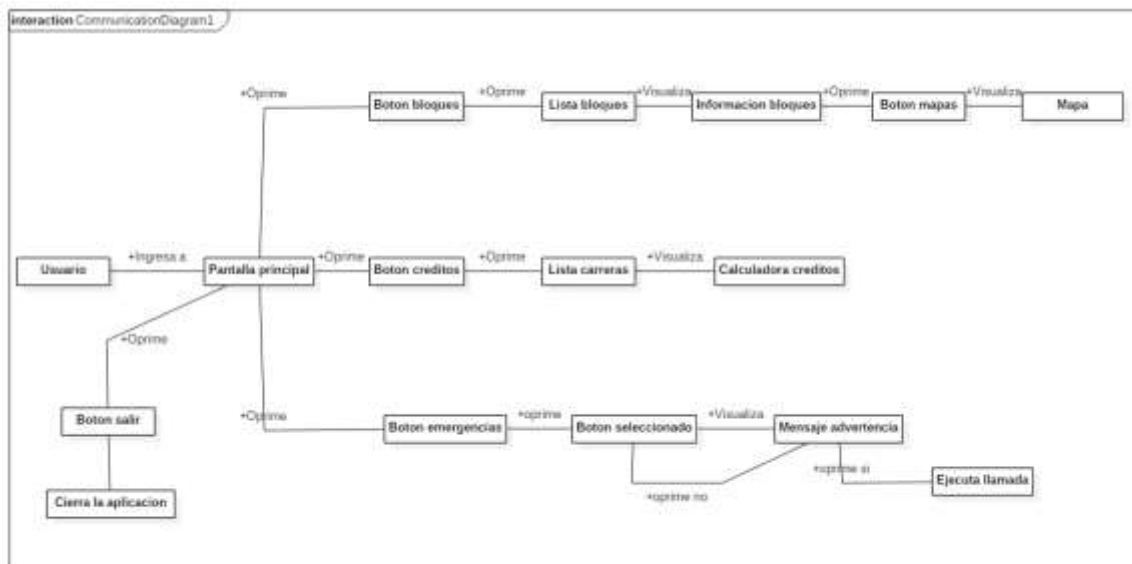
### Diagrama de clases



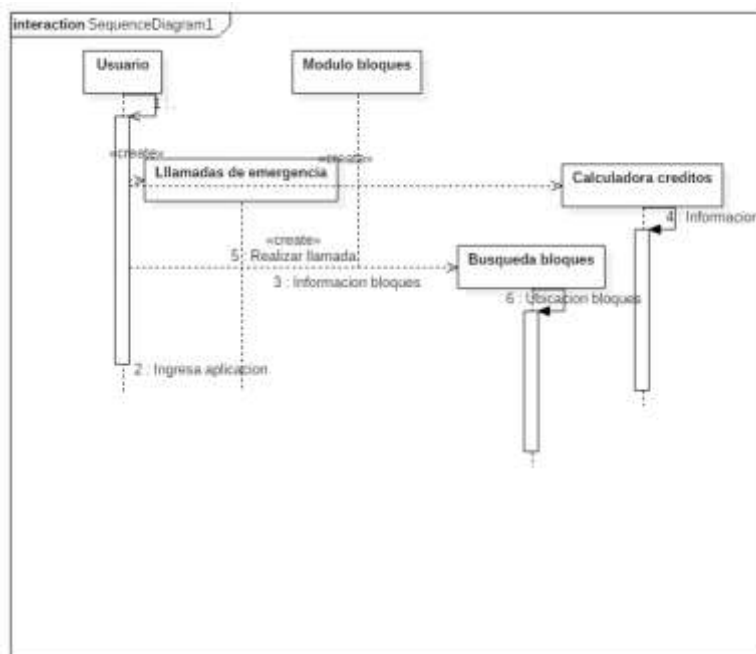
### Diagrama de componentes



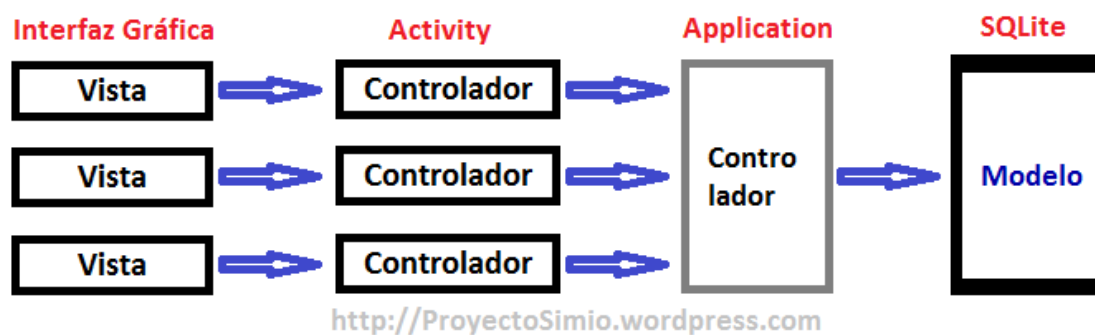
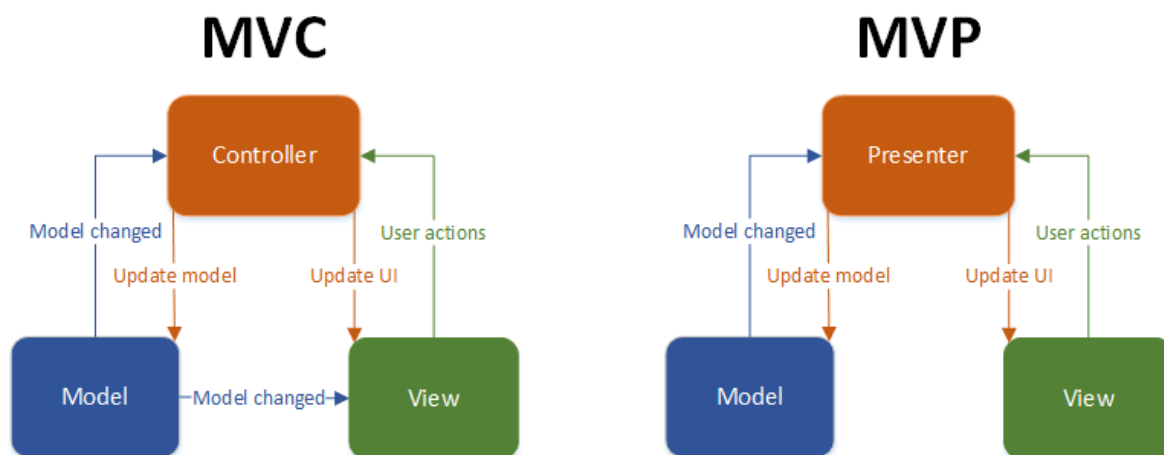
## Diagrama de interacción



## Diagrama de secuencia



## Modelo vista

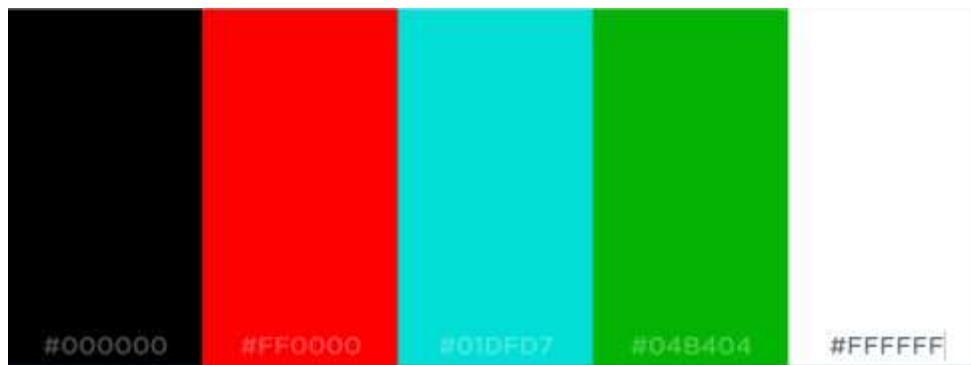


## 5. Diseño grafico

### Logo



## Paleta de colores



## Pantalla principal



## Pantalla lista bloques



## Pantalla información bloques



### Pantalla lista carreras



### Pantalla calculadora créditos





## Pantalla llamadas de emergencia



### Antes de oprimir el boton tenga en cuenta:

1. Esto es solo para casos de emergencia ocurridos dentro de la Universidad.
2. En el menu de opciones marque la opcion 1.
3. Cuando este comunicado con el asesor tendra que indicarle el nit de la universidad:  
Nit: 860029924-7
4. Al momento de la atencion debe presentar el carné de la universidad



## 6. Requisitos del sistema:

Para que la aplicación se ejecute correctamente es necesario tener un dispositivo móvil que cumpla con los siguientes requisitos:

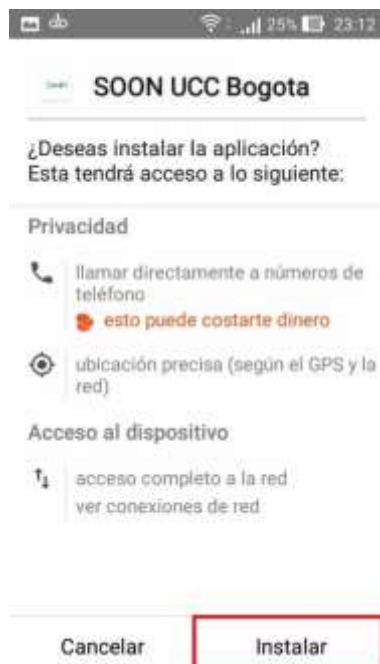
- Sistema operativo Android versión 4.1 o superior
- Espacio de almacenamiento de 50 mb, min 35mb
- Wifi, Gps, bandas de telefonía abiertas
- Google Maps

## 7. Instalación de la aplicación

Para realizar la instalación de la aplicación será necesario tener en el almacenamiento del dispositivo móvil la apk, ya sea descargada o guardada.



Se solicitarán unos permisos para poder realizar llamadas y tener acceso a la ubicación del dispositivo ya que sin estos la aplicación no funcionará correctamente. Para ello le daremos al botón de instalar



Una vez instalada, el usuario podrá navegar por todas las opciones que presenta la aplicación sin ningún problema de manera rápida y eficiente. Para ello le daremos al botón de abrir.

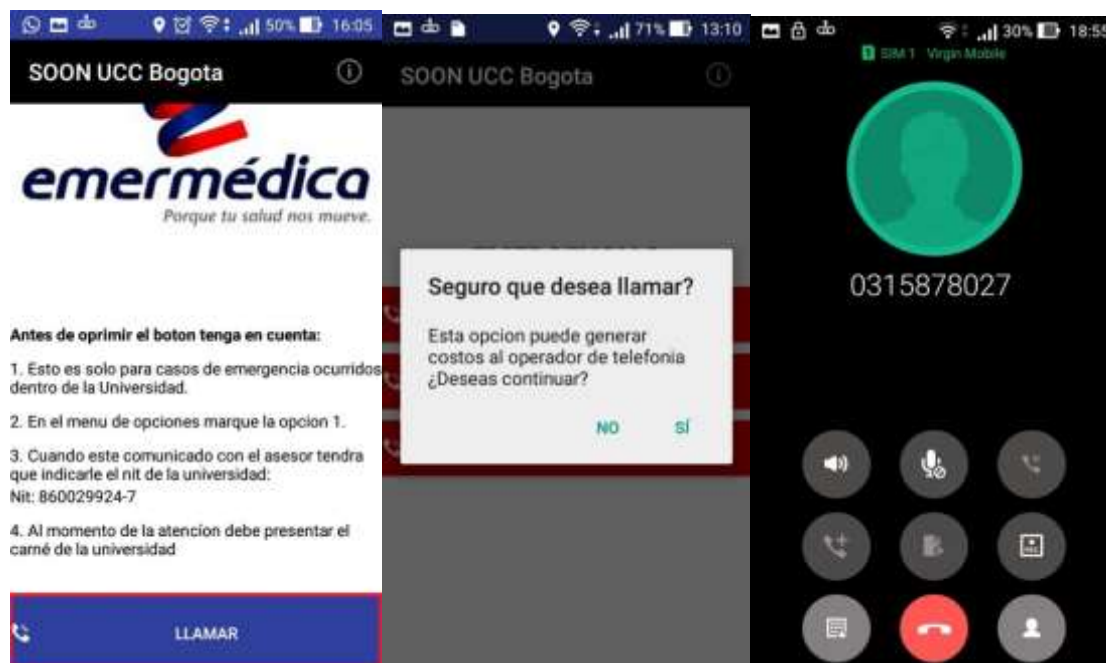


## 8. Configuración de la aplicación

### Llamadas de emergencia

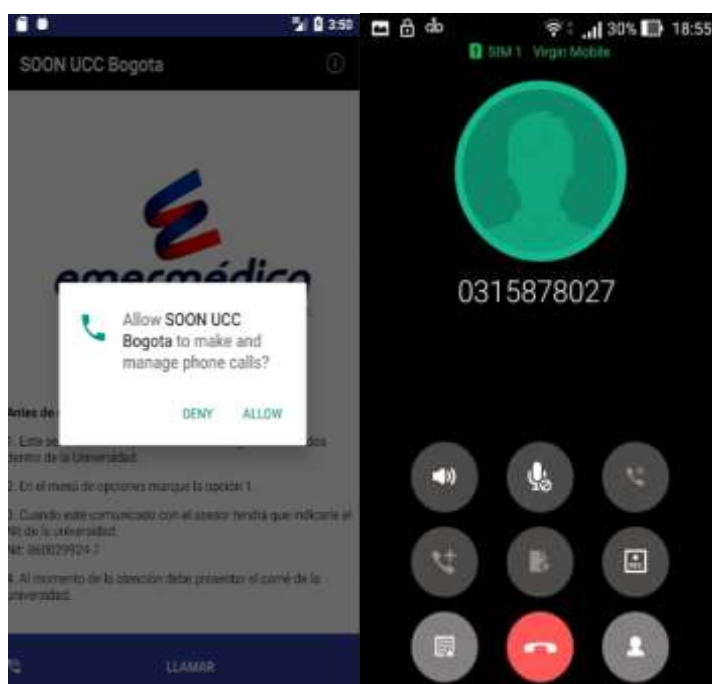
#### Versiones de Android 5.9 o anteriores:

Si el usuario tiene un celular con una versión de Android inferior a la 5.9 (Lollipop) al oprimir el botón de llamar ejecutara una advertencia en donde se le preguntara si ¿desea continuar? Si oprime la opción afirmativa la aplicación ejecutara la llamada de lo contrario cerrara la advertencia.



#### Versiones de Android 6.0 o posteriores

Si el usuario tiene un celular con una versión de Android 6.0 o superior (Marshmallow) al oprimir el botón de llamar la aplicación ejecutara una advertencia en donde se le preguntara si ¿desea continuar? Si oprime la opción afirmativa la aplicación pedirá unos permisos para realizar la llamada, al aceptarlos al darle de nuevo al botón solo será necesario pasar la advertencia de nuevo y se ejecutara la llamada. Si la respuesta es negativa simplemente se cerrara la advertencia.



En caso de que el usuario no acepte los permisos, si intenta ejecutar de nuevo la llamada la aplicación le denegara la ejecución hasta que no solicite y acepte los permisos.

