

Universidad Politécnica de Chiapas  
Ingeniería en Desarrollo de Software  
Proyecto de Investigación  
11° A

# Manual de Programador (Aplicación Móvil)

*Milton Neftalí Hernández Jiménez*

22 de mayo de 2020

# Índice General

<b>1. Introducción.</b>	<b>4</b>
<b>2. Requerimientos.</b>	<b>5</b>
2.1. Software . . . . .	5
2.2. Hardware . . . . .	5
<b>3. Configuración e Instalación.</b>	<b>6</b>
3.1. Instalación del IDE de Android studio. . . . .	6
3.2. Configuración del IDE de Android studio con el uso del proyecto. . . . .	6
3.3. Configuración del dispositivo móvil con sistema operativo Android. . . . .	6
<b>4. Descripción del código del proyecto.</b>	<b>7</b>
4.1. Composición del proyecto. . . . .	7
4.2. Carpeta Actividades. . . . .	7
4.2.1. Clase A_IniciarSesion. . . . .	7
4.2.1.1. EnableRuntimePermissionToAccessCamera: . . . . .	7
4.2.1.2. Validar_Email: . . . . .	7
4.2.1.3. Ir_A_La_Pantalla_de_Perfil: . . . . .	8
4.2.1.4. Iniciar_Sesion: . . . . .	8
4.2.2. Clase B_Perfil. . . . .	8
4.2.2.1. Ir_A_La_Pantalla_De_Crear_Muestra: . . . . .	8
4.2.2.2. Ir_A_La_Pantalla_De_Guardar_Localizacion: . . . . .	8
4.2.3. Clase C_Crear_Muestra. . . . .	8
4.2.3.1. Ir_A_La_Pantalla_De_Resultados: . . . . .	8
4.2.3.2. takePictureIntent: . . . . .	8
4.2.3.3. postFile: . . . . .	8
4.2.3.4. Lista_De_Lugares: . . . . .	9
4.2.3.5. Datos_Region: . . . . .	9
4.2.3.6. Informacion_Muestra: . . . . .	9
4.2.3.7. Datos_de_la_Muestra: . . . . .	9
4.2.3.8. FormatoDecimal: . . . . .	9
4.2.3.9. displayUserData: . . . . .	9
4.2.4. Clase D_Analisis_Croma. . . . .	9
4.2.4.1. Cambio_de_Colores: . . . . .	9
4.2.4.2. Actu_CRM: . . . . .	9
4.2.4.3. subir_Imagen: . . . . .	10
4.2.5. Clase E_Guardar_Coordenadas. . . . .	10
4.2.5.1. Permiso_De_Localizacion: . . . . .	10
4.2.5.2. Obtener_Coordenadas: . . . . .	10
4.2.5.3. nuevo_lista: . . . . .	10
4.2.5.4. lista_de_medidas: . . . . .	10
4.2.5.5. Guardar_Informacion: . . . . .	10

4.3.	Carpeta de API. . . . .	10
4.3.1.	Clase API. . . . .	10
4.3.2.	Clase ClienteRetrfit. . . . .	10
4.3.2.1.	ClienteRetrofit: . . . . .	11
4.3.2.2.	ClienteRetrofit con parámetro: . . . . .	11
4.3.2.3.	getInstance: . . . . .	11
4.3.2.4.	getInstance2: . . . . .	11
4.3.2.5.	getApi: . . . . .	11
4.4.	Carpeta de Modelo. . . . .	11

## **1. Introducción.**

El presente documento describe los aspectos técnicos informáticos del sistema. El manual de programador de la aplicación móvil hace referencia a la información necesaria con el fin de orientar al personal técnico encargado de las actividades de mantenimiento, revisión, solución de problemas y configuración del sistema. La redacción del manual técnico está orientado a personal con conocimientos en sistemas y tecnologías de la información, así como conocimientos medios o avanzados en programación orientada a objetos sobre entorno móvil.

## **2. Requerimientos.**

Para poder abrir el proyecto de la aplicación móvil se hace uso de tecnologías y equipo que continuación se en listan.

### **2.1. Software**

- IDE de Android Studio
- Emulador de equipos móviles Android (Esto es opcional en caso de no contar con un equipo móvil con sistema operativo Android)

### **2.2. Hardware**

- Dispositivo móvil con sistema operativo Android

### **3. Configuración e Instalación.**

Para poder modificar, ejecutar el proyecto hecho en Android studio, se debe tomar en cuenta los requerimientos del sistema que Android studio necesita para el buen manejo del entorno. (ver la página oficial de Android studio en la sección de System Requirements <https://developer.android.com/studio>)

#### **3.1. Instalación del IDE de Android studio.**

- Descargar el IDE de Android studio.
- Seguir todos los pasos que el instalador hace mención. (dentro de los pasos de instalación hace referencia a agregar la ruta del ejecutable en el PATH del sistema, es necesario aceptar esa opción)
- Ejecutar el IDE de Android studio y dejar instalar todos los componentes que automáticamente se ejecutan.

#### **3.2. Configuración del IDE de Android studio con el uso del proyecto.**

- Descargar el proyecto del repositorio de GitHub  
<https://github.com/Milhernandez08/ProyectoAppCroma.git>.
- Abrir el proyecto en el IDE de Android studio.
- Descargar todos los componentes y módulos que sean requeridos para poder modificar, ejecutar el proyecto.

#### **3.3. Configuración del dispositivo móvil con sistema operativo Android.**

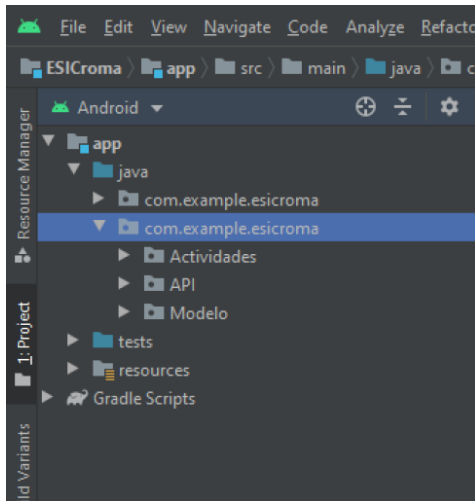
- Ir a ajustes del dispositivo.
- En la sección de Sistema, buscar el apartado de Acerca del teléfono.
- En la sección de Versión de Software presionar las veces necesarias hasta que salga la notificación de que el equipo está para uso del desarrollador.

## 4. Descripción del código del proyecto.

### 4.1. Composición del proyecto.

En esta sección se hará una descripción de cómo está compuesto el proyecto, indicando el uso de carpetas y el contenido en cada una de estas.

En este proyecto se muestran 3 carpetas que permiten la separación del código en partes lógicas que permite el funcionamiento del proyecto.



- Carpeta Actividades: En esta carpeta se almacenan todas las clases de la lógica para el funcionamiento de las Actividades (Parte lógica de lo visual).
- Carpeta API: Esta carpeta almacena las clases lógicas que permiten la comunicación del Back-End y la aplicación.
- Carpeta Modelo: En esta carpeta almacena el modelado de las peticiones que se piden al Back-End

### 4.2. Carpeta Actividades.

Se procede a explicar los componentes de cada clase que almacena esta carpeta.

#### 4.2.1. Clase A\_IniciarSesion.

En esta clase se hacen los procesos que permite hacer el inicio de sesión con las siguientes funciones:

##### 4.2.1.1. EnableRuntimePermissionToAccessCamera:

Esta función permite que la aplicación al momento de abrirla, muestre una ventana pidiendo el permiso del uso de la cámara.

##### 4.2.1.2. Validar Email:

Esta función recibe un parámetro de tipo String y permite validar si el texto proporcionado en el campo donde se ingresa el correo, evalúa si es un correo electrónico bien escrito.

#### **4.2.1.3. Ir\_A\_La\_Pantalla\_de\_Perfil:**

Esta función permite que la aplicación cambie de ventana, y pueda abrir la ventana de perfil.

#### **4.2.1.4. Iniciar\_Sesion:**

Esta función permite almacenar la información de los campos de texto (correo y contraseña) en variables con los mismos nombres, el cual hace una petición al back-end enviando la información almacenada en las dos variables y recibiendo la respuesta que el back-end envía, esta función evalúa la respuesta del back-end y en caso de que la información recibida no sea válida permanece en la misma ventana, en caso contrario esta función hace que guarde la respuesta del back-end en una variable con el nombre de Datos\_De\_Usuario y cambia de ventana usando la función Ir\_A\_La\_Pantalla\_de\_Perfil.

#### **4.2.2. Clase B\_Perfil.**

En esta clase solo se encuentran 2 funciones que permiten cambiar de ventana.

##### **4.2.2.1. Ir\_A\_La\_Pantalla\_De\_Crear\_Muestra:**

Esta función permite cambiar de ventana y muestra la ventana de C\_Crear\_Muestra.

##### **4.2.2.2. Ir\_A\_La\_Pantalla\_De\_Guardar\_Localizacion:**

Esta función permite cambiar de ventana y muestra la ventana de E Guardar Coordenadas.

#### **4.2.3. Clase C\_Crear\_Muestra.**

En esta clase es donde se hace el análisis de la muestra, hace una petición y recibe la respuesta del back-end.

##### **4.2.3.1. Ir\_A\_La\_Pantalla\_De\_Resultados:**

Esta función permite cambiar de ventana y muestra la ventana de Pantalla De Resultados.

##### **4.2.3.2. takePictureIntent:**

Esta función permite tomar la fotografía.

##### **4.2.3.3. postFile:**

Esta función permite enviar la fotografía tomada y recibir respuesta del servidor con el análisis hecho, esta misma función permite cambiar de ventana cuando recibe la respuesta del servidor y se dirige a la ventada D Analisis Croma.



#### **4.2.3.4. Lista\_De\_Lugares:**

Esta función permite traer la información de los lotes del usuario que ha iniciado sesión.

#### **4.2.3.5. Datos\_Region:**

Esta función permite traer los datos de municipio, estado y país del usuario que ha iniciado la sesión.

#### **4.2.3.6. Informacion\_Muestra:**

Esta función recibe un parámetro de tipo int es el identificador del croma, esto para hacer una petición al back-end y como respuesta del back-end se reciba los datos de la muestra.

#### **4.2.3.7. Datos\_de\_la\_Muestra:**

Esta función recibe un parámetro de tipo int que es el identificador del croma, esta petición al back-end y recibe como respuesta la información de todo el croma, la misma función muestra la latitud y longitud de la muestra tomada y se muestra para que el usuario sepa cuales son las coordenadas donde se tomó su muestra.

#### **4.2.3.8. FormatoDecimal:**

Esta función recibe un parámetro de tipo String y permite reducir la cantidad de decimales de un número.

#### **4.2.3.9. displayUserData:**

Esta función recibe un parámetro de tipo Object esta variable almacena toda la información del lote del usuario y esta función permite colocar en la parte visual el nombre del lote.

#### **4.2.4. Clase D\_Analisis\_Croma.**

En esta clase solo se hace uso de 3 de funciones para registrar los datos del croma analizado en la base de datos.

##### **4.2.4.1. Cambio\_de\_Colores:**

Esta función se utiliza para hacer cambio de colores en la parte visual de los campos donde se detectan que hay existencia de los componentes busados en el croma.

##### **4.2.4.2. Actu\_CRM:**

Esta función permite traer la información de un croma, para eso necesita 2 parámetros los cuales son uno de tipo int y otro de tipo Object, que con eso la función hace una petición al back-end.

#### **4.2.4.3. subir\_Imagen:**

Esta función permite almacenar la fotografía que se tomó, al servidor.

#### **4.2.5. Clase E\_Guardar\_Coordenadas.**

Esta clase permite obtener y guardar las coordenadas de la ubicación de donde la muestra que se obtuvo.

##### **4.2.5.1. Permiso\_De\_Localizacion:**

Esta función permite que muestre una ventana pidiendo el permiso del uso de la localización (GPS).

##### **4.2.5.2. Obtener\_Coordenadas:**

Esta función permite obtener las coordenadas, almacena la información en 2 variables (latitud y longitud) y las muestra en los campos respectivos en la parte visual del proyecto.

##### **4.2.5.3. nuevo\_lista:**

Esta función hace una petición al back-end para traer información de los lotes, el cual se muestran de forma visual los nombres de cada lote registrado por el usuario.

##### **4.2.5.4. lista\_de\_medidas:**

Esta función muestra de forma visual la lista de medidas que se toman para extraer la muestra de cada croma.

##### **4.2.5.5. Guardar\_Informacion:**

Esta función permite guardar el nombre de la muestra que se tomó y también permite guardar las coordenadas de donde se obtuvo la muestra a la base de datos.

### **4.3. Carpeta de API.**

Se procede a explicar los componentes de cada clase que almacena esta carpeta.

#### **4.3.1. Clase API.**

En esta clase se hace todas las peticiones que se solicitan al servidor, para obtener respuesta de lo solicitado.

#### **4.3.2. Clase ClienteRetrofit.**

Esta clase está orientada a la conexión con los servidores y tener respuestas con el back-end.

#### **4.3.2.1. ClienteRetrofit:**

Esta función permite conectarse al servidor donde se aloja el back-end de la información de usuario, etc.

#### **4.3.2.2. ClienteRetrofit con parámetro:**

Esta función recibe un parámetro de tipo String el cual tiene como propósito conectarse con el servidor que aloja el servicio de análisis del cromatograma.

#### **4.3.2.3. getInstance:**

Esta función permite crear una instancia de la conexión del CLienteRetrofit.

#### **4.3.2.4. getInstance2:**

Esta función permite crear una instancia de la conexión del CLienteRetrofit con parámetro.

#### **4.3.2.5. getApi:**

Esta función retorna la conexión hecha.

### **4.4. Carpeta de Modelo.**

En esta carpeta se almacenan todas las clases con el modelo que pertenecen respecto a las respuestas que el back-end envía a través de JSON.