

Manual Técnico

SubiteSv



DESARROLLADORES:

2

Kevin Adonay Martínez Cerón
Patrick Ernesto Rosales Mendoza
Stanley Adonay Mejía Amaya

Introducción

3

La finalidad de todo manual técnico es la de proporcionar al lector las pautas de configuración y la lógica con la que se ha desarrollado una aplicación, la cual se sabe que es propia de cada programador; por lo que se considera necesario ser documentada.

Requerimientos de desarrollo:

4

**android
studio**



Es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android y está basado en IntelliJ IDEA.





Android SDK

Con el podremos desarrollar aplicaciones y ejecutar un emulador del sistema Android de la versión que sea.



Android JDK

Se puede definir como un conjunto de herramientas, utilidades, documentación y ejemplos para desarrollar aplicaciones Java.

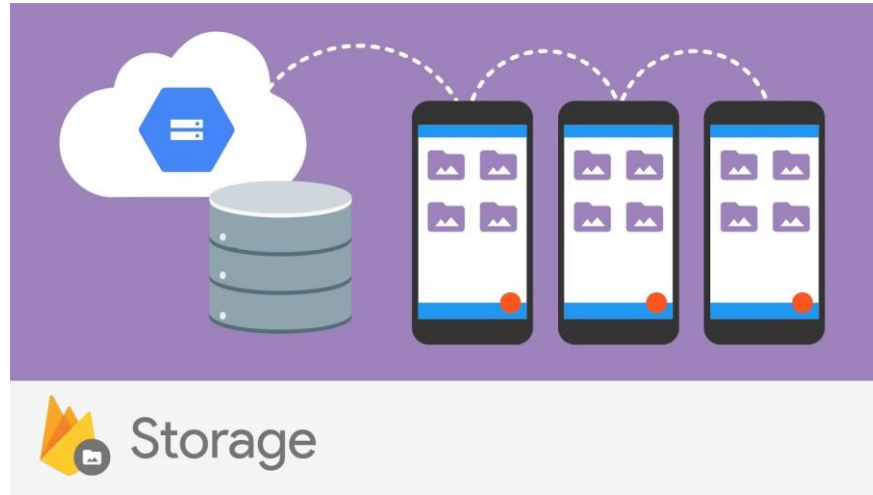
Firestore



Realtime database: Una de las herramientas más destacadas y esenciales de Firestore son las **bases de datos en tiempo real**. Estas se alojan en la **nube**, son **No SQL** y almacenan los datos como **JSON**. Permiten alojar y disponer de los datos e información de la aplicación en tiempo real, manteniéndolos actualizados aunque el usuario no realice ninguna acción.

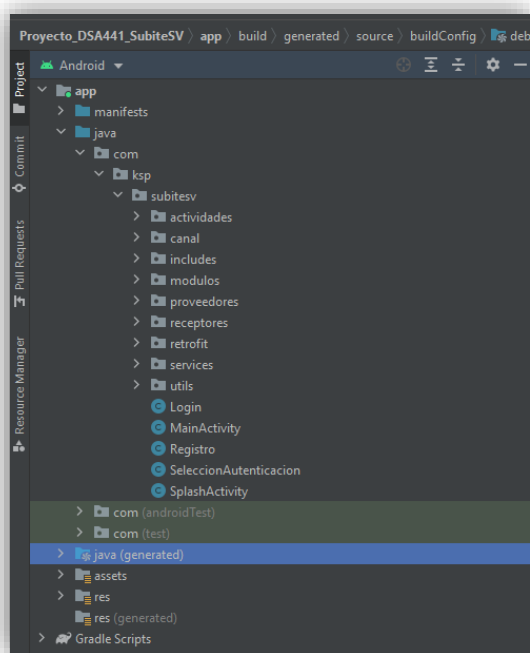
Firestore Authentication: proporciona servicios de backend, SDK fáciles de usar y bibliotecas de IU ya elaboradas para autenticar a los usuarios en tu app. Admite la autenticación mediante contraseñas, números de teléfono, proveedores de identidad federada populares, como Google, Facebook y Twitter, y mucho más.

Firestore: se diseñó para ayudarte a almacenar y procesar con rapidez y facilidad el contenido generado por usuarios, como fotos y videos.



Estructura Proyecto.

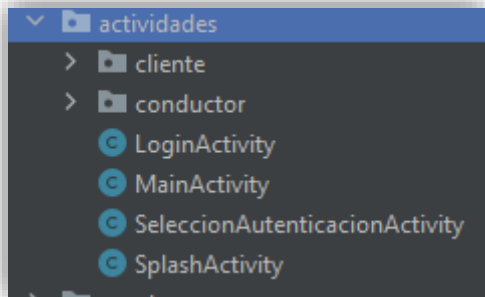
9



Esta es la estructura de nuestro Proyecto “SubiteSV”

Package actividades.

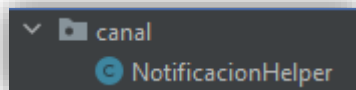
10



En este package se encuentran organizadas las “**Activities**” que hemos utilizado en el desarrollo de nuestro proyecto, además que se encuentran organizadas por cliente y conductor, etc.

Package canal.

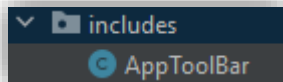
11



En este package se encuentran la clase **NotificacionHelper** en la cual se encuentra organizada la lógica de las notificaciones que recibirá el conductor cuando un cliente requiera de sus servicios.

Package includes.

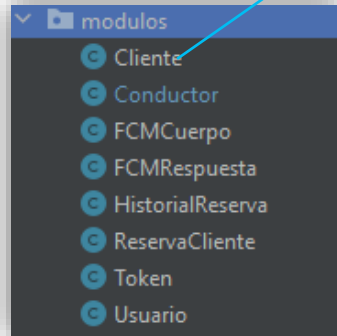
12



En este package se encuentra la clase **AppToolBar** en la cual agregamos datos específicos para las toolbars de nuestra app.

Package modulos.

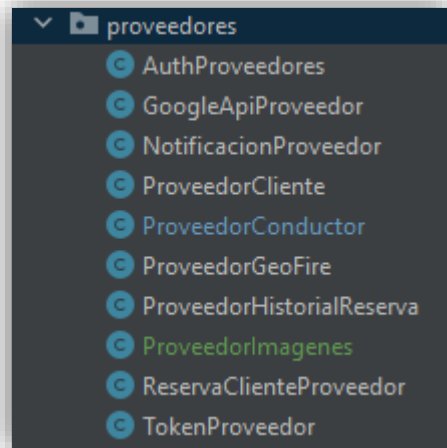
En este package se encuentran las clases necesarias que hemos utilizado, donde creamos los constructores, métodos get y set, etc.



```
1 package com.ksp.subitesv.modulos;
2
3 public class Cliente {
4     String id;
5     String nombre;
6     String correo;
7     String imagen;
8
9     public String getId() { return id; }
10
11     public void setId(String id) { this.id = id; }
12
13     public String getNombre() { return nombre; }
14
15     public void setNombre(String nombre) { this.nombre = nombre; }
16
17     public String getCorreo() { return correo; }
18
19     public void setCorreo(String correo) { this.correo = correo; }
20
21     public Cliente(String id, String nombre, String correo) {
22         this.id = id;
23         this.nombre = nombre;
24         this.correo = correo;
25     }
26
27     public Cliente() {
28     }
29
30     public String getImagen() { return imagen; }
31
32     public void setImagen(String imagen) { this.imagen = imagen; }
33
34     public Cliente(String id, String nombre, String correo, String imagen) {
35         this.id = id;
36         this.nombre = nombre;
37         this.correo = correo;
38         this.imagen = imagen;
39     }
40 }
```

Package proveedores.

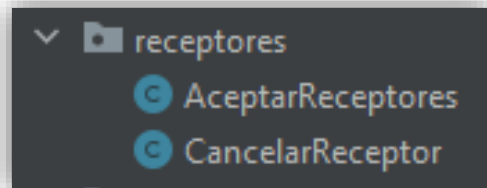
14



En este package se encuentra las clases donde se crea la lógica para firebase, además de GoogleApi.

Package includes.

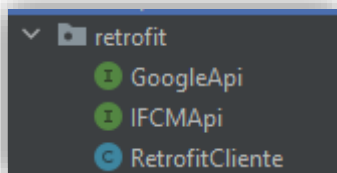
15



En este package se encuentran las clases donde se creo la lógica para aceptar y cancelar solicitud que llega por notificación.

Package retrofit.

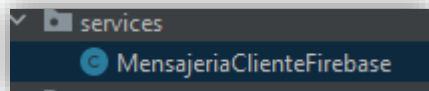
16



En este package se encuentra las clases que creamos para obtener datos con retrofit.

Package includes.

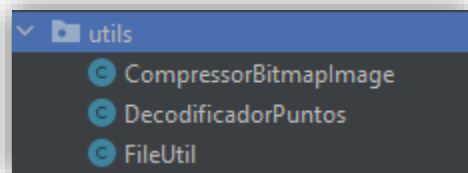
17



En este package la clase **MensajeriaClienteFirebase** la cual contiene la lógica para la notificación que será enviada al conductor.

Package utils.

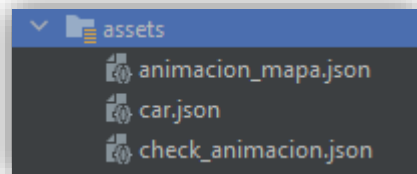
18



En este package se encuentra parte de la lógica para poder insertar imágenes.

Carpeta assets.

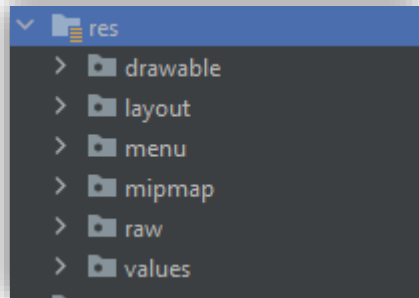
19



En esta carpeta se encuentran las animaciones en tipo json que utilizamos en nuestra aplicación.

Carpeta res.

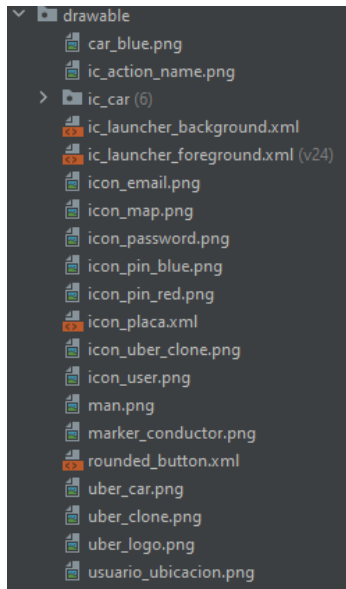
20



En esta carpeta se encuentran organizados los recursos utilizados en nuestra app.

Carpeta drawable.

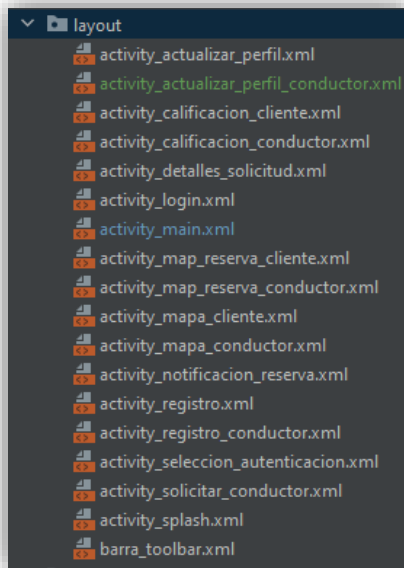
21



En esta carpeta se encuentran las imágenes que hemos utilizado.

Carpeta layout.

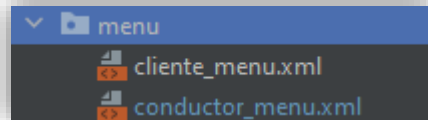
22



En esta carpeta se encuentran cada uno de los layout con el diseño de las pantallas de nuestra app.

Carpeta menu.

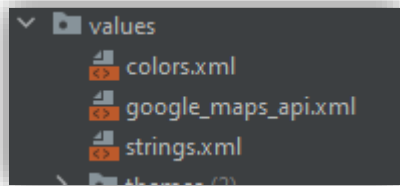
23



En esta carpeta se encuentran los layout que hemos utilizado para el diseño del menú que se muestra en la pantalla principal del cliente y conductor.

Carpeta values.

24



En esta carpeta encuentran organizados los recursos que utilizamos en nuestra app, como la paleta de colores en el archivo **colors.xml** y las strings en el archivo **strings.xml**.