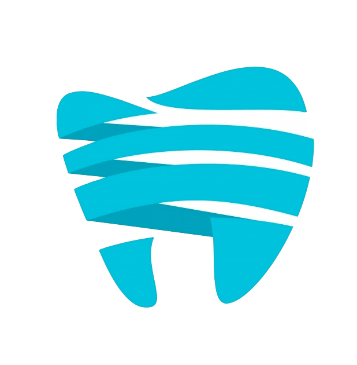


MANUAL

DE DESARROLLADOR



DentalApp

**INDICE**

[DESCRIPCIÓN DE LA APLICACIÓN 4](#_Toc71584488)

[REQUISITOS DE LA APLICACIÓN 5](#_Toc71584489)

[MANUAL DE DESARROLLADOR 6](#_Toc71584490)

[**DISEÑO DE PANTALLAS** 7](#_Toc71584491)

[INICIO DE SESIÓN 7](#_Toc71584492)

[REGISTRO 8](#_Toc71584493)

[AGENDAR CITA EDITAR CITA 10](#_Toc71584494)

[LISTA DE CITAS RECUPERAR CONTRASEÑA 12](#_Toc71584495)

[CREAR USUARIO 13](#_Toc71584496)

[EDITAR CITA 15](#_Toc71584497)

[PERFIL DE USUARIOS 16](#_Toc71584498)

[MENÚ LATERAL 17](#_Toc71584499)

[**PROGRAMACIÓN DE PANTALLAS (JAVA CLASS)** 18](#_Toc71584500)

[Login.java 18](#_Toc71584501)

[signUp.java 21](#_Toc71584502)

[sign\_up2.java 22](#_Toc71584503)

[sign\_up3.java 23](#_Toc71584504)

[MainActivity.java 24](#_Toc71584505)

[listService.java 25](#_Toc71584506)

[forgetPassword.java 26](#_Toc71584507)

[editAppointment.java 27](#_Toc71584508)

[dashboard.java 28](#_Toc71584509)

[addCatalogue.java 30](#_Toc71584510)

[addAppointment.java 31](#_Toc71584511)

[addUser.java 34](#_Toc71584512)

[addUser2.java 35](#_Toc71584513)

[addUser3.java 36](#_Toc71584514)

[editAppointment.java 37](#_Toc71584515)

[BottomSheet.java 40](#_Toc71584516)

[ListOfAppointments.java 41](#_Toc71584517)

[UserProfile.java 43](#_Toc71584518)

[**MODELS** 45](#_Toc71584519)

[Appointment.java 45](#_Toc71584520)

[Service.java 46](#_Toc71584521)

[User.java 47](#_Toc71584522)

[**ADAPTERS** 48](#_Toc71584523)

[AdapterAppointment.java 48](#_Toc71584524)

[AdapterService.java 49](#_Toc71584525)

[AdapterAppointmentDashboard.java 50](#_Toc71584526)

# DESCRIPCIÓN DE LA APLICACIÓN

**Dental App** es una aplicación móvil para sistema operativo Android a partir de la versión 4.1, la cual tiene como principal objetivo llevar el control de citas de los pacientes de una clínica dental además de permitir la visualización de los servicios ofrecidos por la misma y un aspecto a resaltar es que esta se encuentre al alcance de cualquier cliente de la clínica por lo cual el sitio le permite registrarse.

.

# REQUISITOS DE LA APLICACIÓN

Para el correcto funcionamiento de la aplicación DentalApp se pide como requisito lo siguiente:

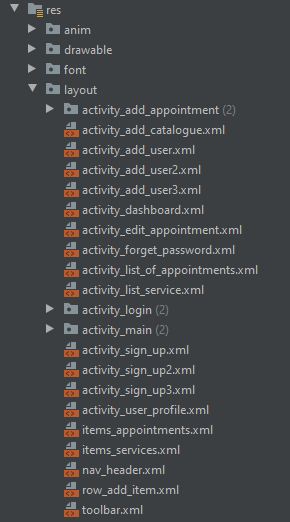
* Dispositivo con sistema operativo Android 4.1 o superiores.
* 1 GB de memoria RAM
* 512 MB de almacenamiento
* Conexión a internet

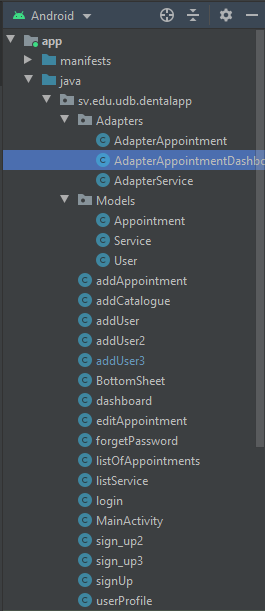
Para el correcto desarrollo de la aplicación DentalApp se piden los siguientes requisitos:

* Android Studio 4.1.3
* Acceso consola del proyecto DentalApp en Firebase (Administrador de base de datos de la app)

# MANUAL DE DESARROLLADOR

A continuación, se muestra la estructura del proyecto en Android llamado “DentalApp”. Al lado izquierdo se muestran las clases “Java” que se han utilizado en la programación de las diferentes pantallas que componen “DentalApp”. Al lado derecho se muestran los diferentes layouts (pantallas) que se han diseñado para “DentalApp”



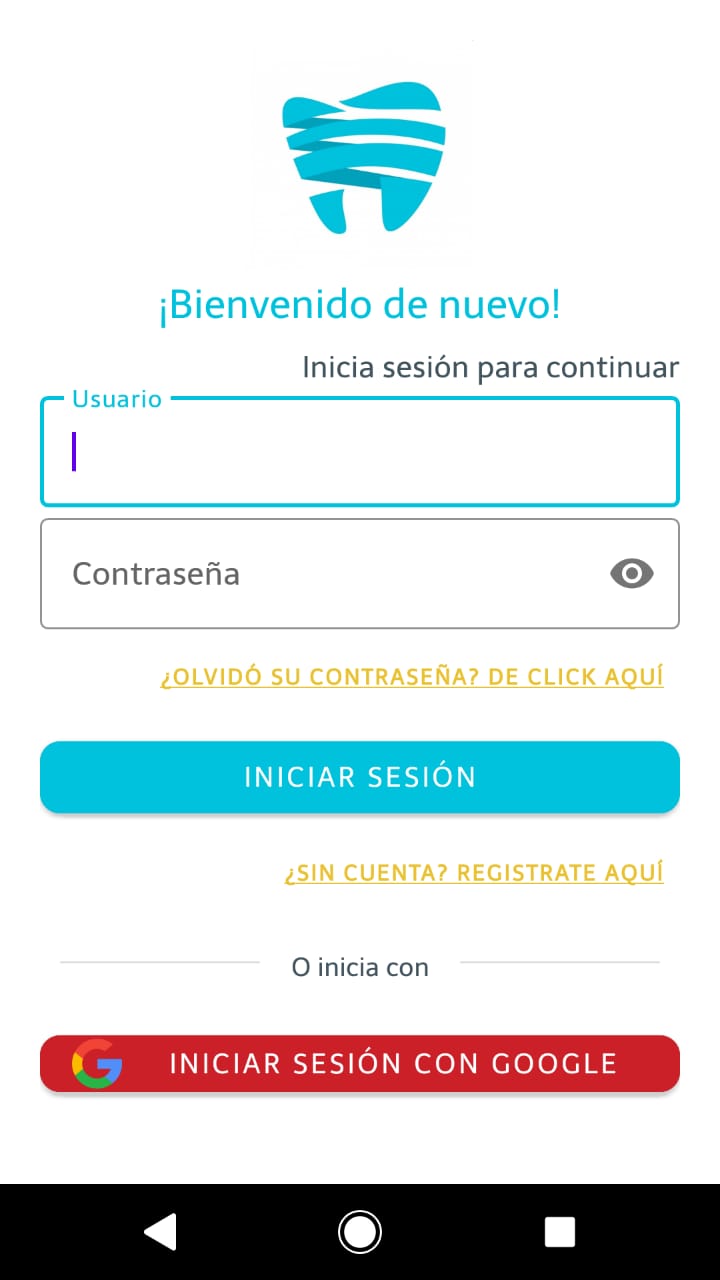


### **DISEÑO DE PANTALLAS**

A continuación, se muestra el diseño de cada una de las pantallas de la aplicación. Se ha colocado el nombre de la pantalla, el nombre el layout “nombrepantalla.xml” y la imagen con su respectivo id, con el cual se identifica a cada uno de los componentes dentro de cada clase “java” de programación de cada pantalla.

#### INICIO DE SESIÓN

##### normal\activity\_login.xml



|  |  |
| --- | --- |
| # | Id (Diseño) |
| 1 | edtUser |
| 2 | edtPass |
| 3 | recuperarcontra |
| 4 | btnLogin |
| 5 | BtnGoogle |

Ilustración 2: Pantalla de Inicio de Sesión

#### REGISTRO

##### activity\_sign\_up.xml

##### activity\_sign\_up2.xml

##### 

Ilustración 4: Pantalla 2 de registro

Ilustración 3: Pantalla 1 de registro

|  |  |
| --- | --- |
| # | Id (Diseño) |
| 1 | Registrername |
| 2 | Registrerlastname |
| 3 | Registrerphone |
| 4 | btnNext |
| 5 | btnLogin |

|  |  |
| --- | --- |
| # | Id (Diseño) |
| 1 | Registrername |
| 2 | Registrerlastname |
| 3 | Registrerphone |
| 4 | btnNext |
| 5 | btnLogin |
| 6 | btnBack |

|  |  |
| --- | --- |
| # | Id (Diseño) |
| 1 | RegistrerUser |
| 2 | RegistrerEmail |
| 3 | RegistrerPassword |
| 4 | RegistrerPasswordAgain |
| 5 | btnLogin |
| 6 | btnNext |

##### activity\_sign\_up3.xml



|  |  |
| --- | --- |
| # | Id (Diseño) |
| 1 | rbtnMale, rbtnFemale |
| 2 | DPfecha |
| 3 | btnNext |

Ilustración 5: Pantalla final de registro

#### AGENDAR CITA EDITAR CITA

#### 

##### activity\_edit\_appointment.xml

##### normal\activity\_add\_appointment.xml



Ilustración 7: Pantalla para editar cita

Ilustración 6: Pantalla para agendar cita

|  |  |
| --- | --- |
| # | Id (Diseño) |
| 1 | user |
| 2 | btnSelectTime\_Date |
| 3 | hour\_date |
| 4 | observation |
| 5 | listServicesAppointment |
| 6 | txtvServiceRequired |
| 7 | btnaddAppointment |
| 8 | btncancel |

|  |  |
| --- | --- |
| # | Id (Diseño) |
| 1 | user |
| 2 | btnSelectTime\_Date |
| 3 | hour\_date |
| 4 | observation |
| 5 | listServices |
| 6 | txtvServiceRequired |
| 7 | btnaddAppointment |
| 8 | btncancel |

##### *AÑADIR CATÁLOGO CATÁLOGO DE SERVICIOS*



##### activity\_list\_service.xml

##### activity\_add\_catalogue.xml

##### activity\_add\_catalogue.xml



Ilustración 9: Pantalla para ver servicios

Ilustración 8: Pantalla para agregar servicios

|  |  |
| --- | --- |
| # | Id (Diseño) |
| 1 | listServices |
| 2 | btnback |

|  |  |
| --- | --- |
| # | Id (Diseño) |
| 1 | txtServiceName |
| 2 | txtDescription |
| 3 | txtPrice |
| 4 | txtTime |
| 5 | btnAddCatalogue |
| 6 | btnCancel |

#### LISTA DE CITAS RECUPERAR CONTRASEÑA

##### activity\_list\_of\_appointments.xml

##### activity\_forget\_password.xml

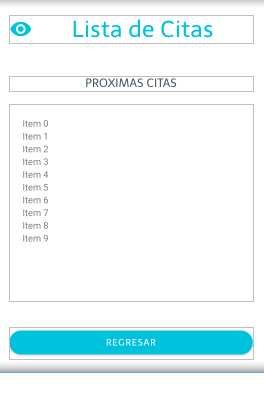


Ilustración 11: Pantalla para recuperar contraseña

Ilustración 10: Pantalla para ver citas

|  |  |
| --- | --- |
| # | Id (Diseño) |
| 1 | rcorreo |
| 2 | rbtn |
| 3 | btnBack |

|  |  |
| --- | --- |
| # | Id (Diseño) |
| 1 | listAppointments |
| 2 | btnback |

#### CREAR USUARIO

##### activity\_add\_user2.xml



##### activity\_add\_user.xml



Ilustración 13: Pantalla para agregar usuarios

Ilustración 12: Pantalla para agregar usuarios

|  |  |
| --- | --- |
| # | Id (Diseño) |
| 1 | btnBack |
| 2 | txtName |
| 3 | txtLastName |
| 4 | txtPhone |
| 5 | btnNext |

|  |  |
| --- | --- |
| # | Id (Diseño) |
| 1 | btnBack |
| 2 | txtAddUserEmail |
| 3 | txtAddUserPassword |
| 4 | txtAddUserUsuario |
| 5 | btnNext |

##### activity\_add\_user3.xml

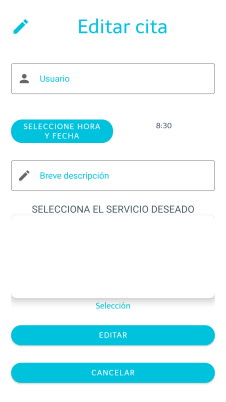


|  |  |
| --- | --- |
| # | Id (Diseño) |
| 1 | btnBack |
| 2 | rbtnAddUserMale |
| 3 | rbtnAddUserFemale |
| 4 | rbtnQuee |
| 5 | rdbTypeAdmin |
| 6 | rdbTypeClient |
| 7 | fechaAddUser |
| 8 | btnNext |

Ilustración 14: Pantalla para agregar usuarios

#### EDITAR CITA

##### activity\_edit\_appointment.xml



|  |  |
| --- | --- |
| # | Id (Diseño) |
| 1 | userEdit |
| 2 | hour\_dateEdit |
| 3 | observationEdit |
| 4 | ServicesAppointmentEdit |
| 5 | btnEditAppointment |
| 6 | btncancelEdit |

Ilustración 15: Pantalla de Editar citas

#### PERFIL DE USUARIOS

##### activity\_user\_profile.xml



|  |  |
| --- | --- |
| # | Id (Diseño) |
| 1 | txtNameProfile |
| 2 | txtLasNameProfile |
| 3 | txtEmailProfile |
| 4 | txtPasswordProfile |
| 5 | txtPhoneProfile |
| 6 | btnUpdate |

Ilustración 16: Pantalla del Perfil de Usuario

#### MENÚ LATERAL

##### activity\_dashboard.xml

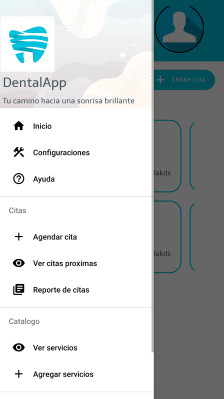


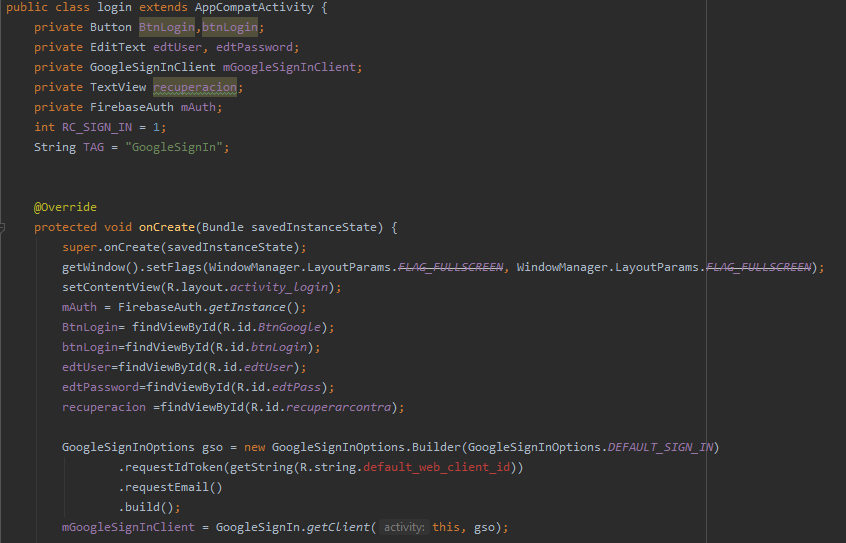
Ilustración 17: Menú lateral

### **PROGRAMACIÓN DE PANTALLAS (JAVA CLASS)**

A continuación, se muestran el código utilizado para cada pantalla diseñada (layout). Se coloca el nombre, y unas capturas con la muestra del código que se ha programado para la funcionalidad de “DentalApp”.

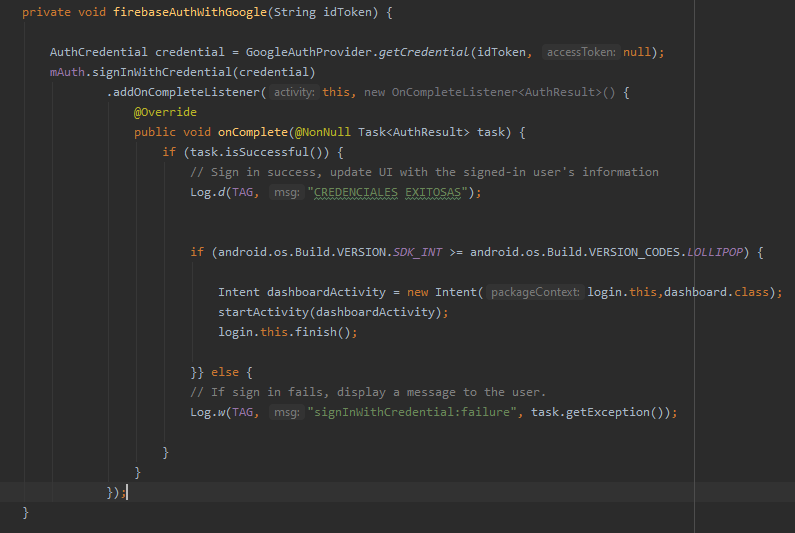
#### Login.java

En esta clase se programa la vista del inicio de sesión de usuarios, ya sean administradores o clientes de la clínica dental. Se puede iniciar sesión con el correo de Gmail con ayuda de la autenticación de Firebase o ingresando las credenciales en los inputs establecidos. Además de tener los métodos para acceder a la vista de olvido de contraseña.

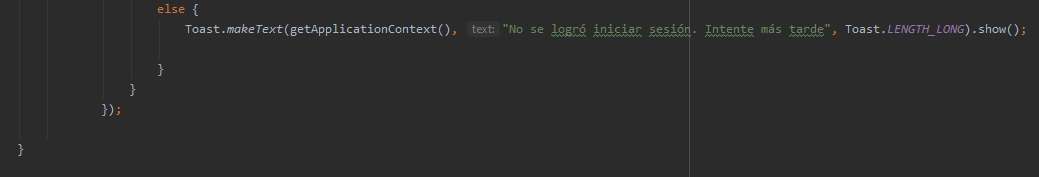


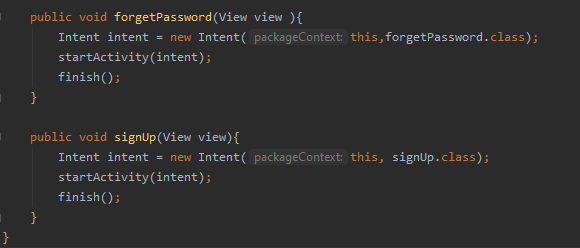






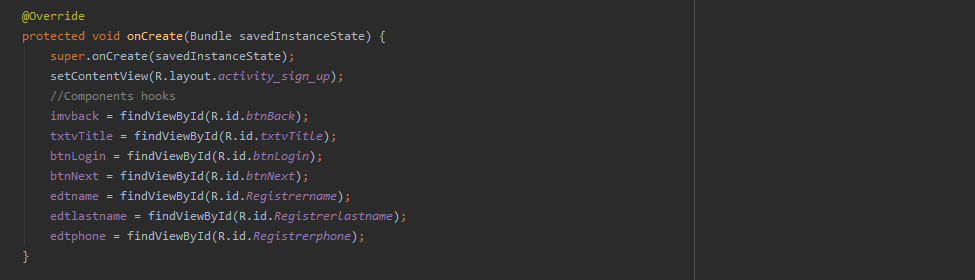




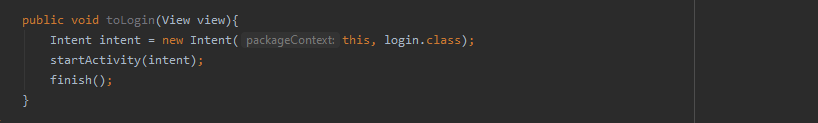


#### signUp.java

En esta clase se programa el primer paso del registro de usuarios tipo cliente. En la sesión de diseño, se muestran los id de los inputs o elementos utilizados en cada pantalla. En este caso se programa el ingreso de los datos personales básicos. Además del método para pasar del paso 1 al paso 2.



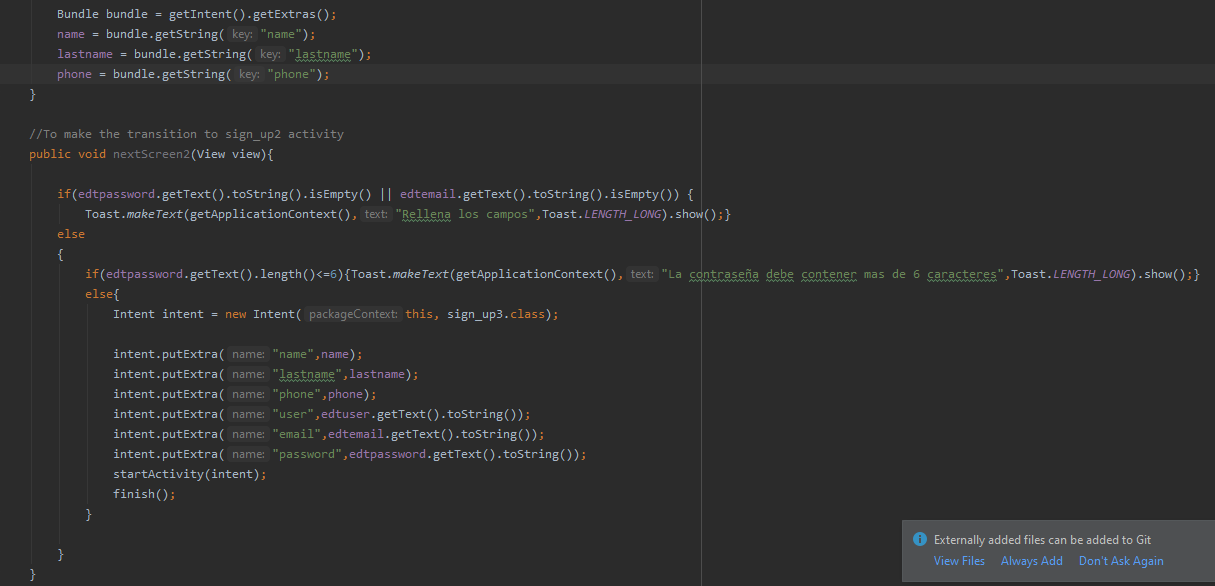


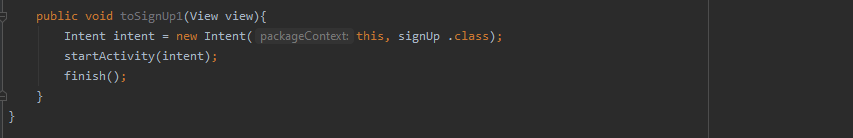


#### sign\_up2.java

En esta clase se programa el primer paso del registro de usuarios tipo cliente. En la sesión de diseño, se muestran los id de los inputs o elementos utilizados en cada pantalla. En este caso se programa el ingreso de los datos para iniciar sesión. Además del método para pasar del paso 2 al paso 3, y de regresar al paso 1.



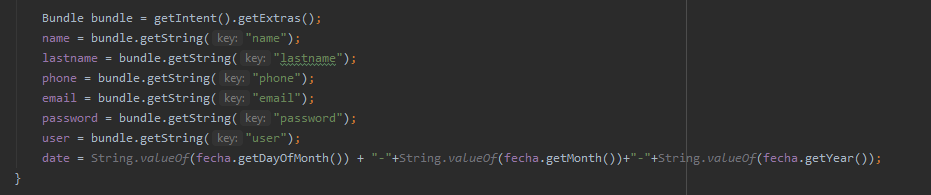




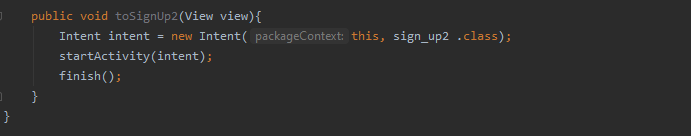
#### sign\_up3.java

En esta clase se programa el primer paso del registro de usuarios tipo cliente. En la sesión de diseño, se muestran los id de los inputs o elementos utilizados en cada pantalla. En este caso se programa el ingreso de los datos personales básicos del usuario. Además del método para regresar al paso 2.



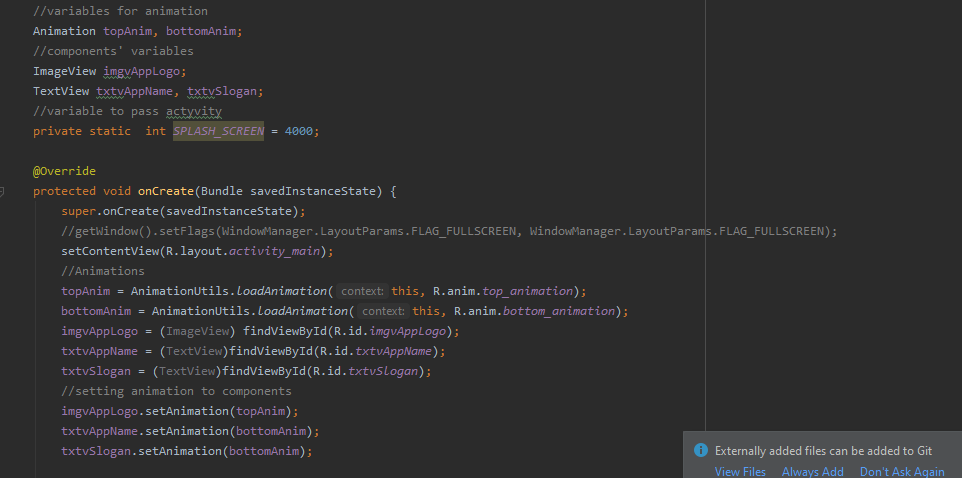


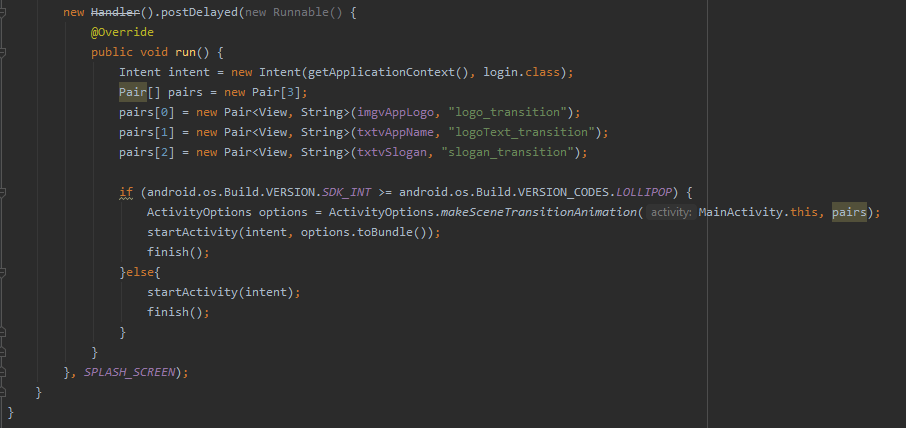




#### MainActivity.java

En esta clase se programa la pantalla principal, con la cual inicia la app, donde se encuentra el logo de la aplicación, el nombre y el logan. Se encuentra el método para la animación inicial de la app.

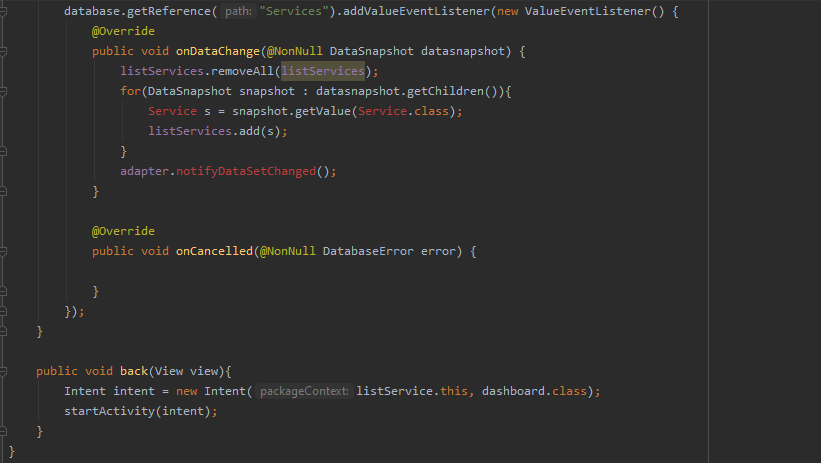




#### listService.java

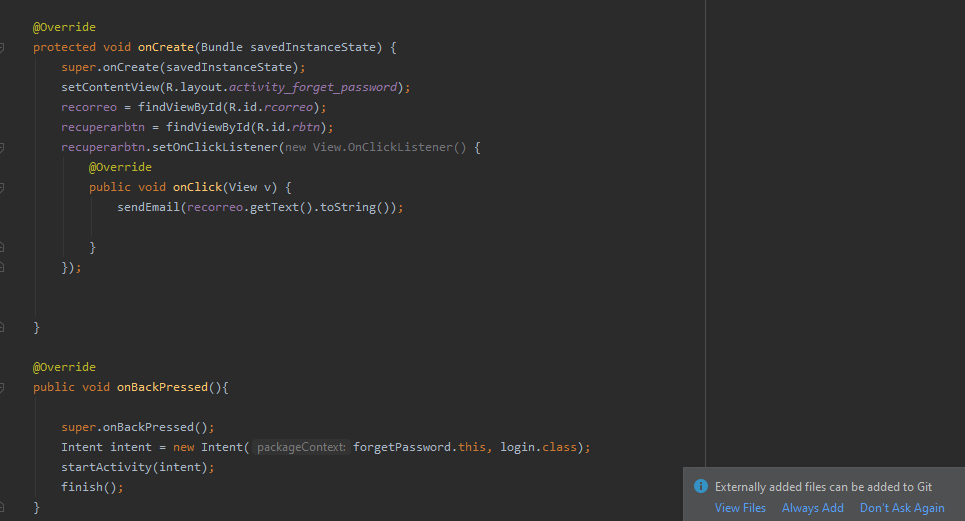
En esta clase se programa la lista de servicios que se muestran en el catálogo de servicios en un recyclerview.

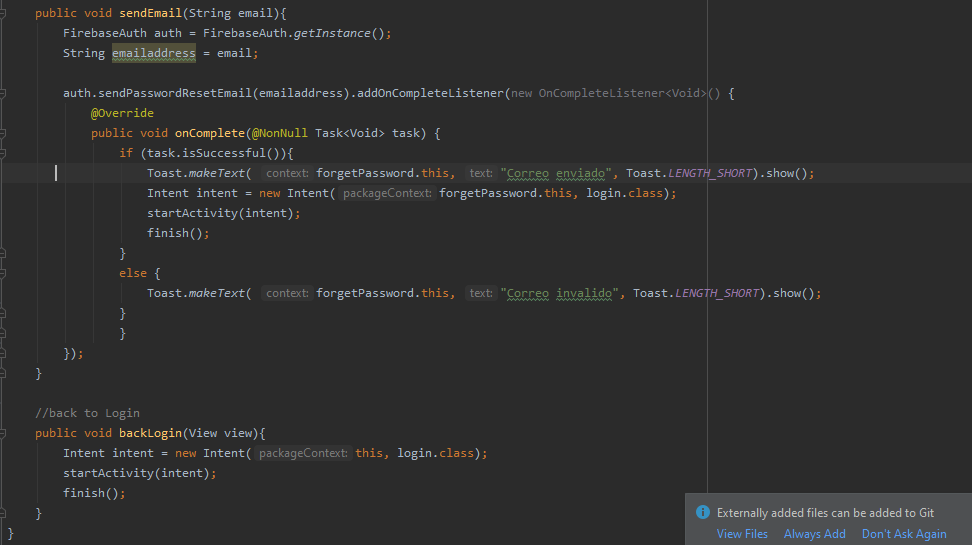




#### forgetPassword.java

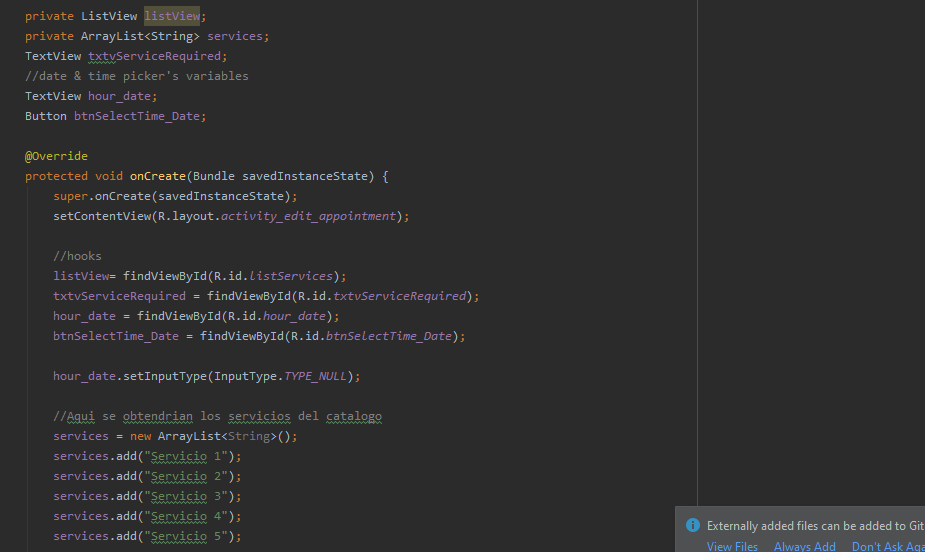
En esta clase se programa la vista de “recuperar contraseña”, con ayuda de Firebase se realiza este proceso. Se declara la variable de correo electrónico para obtener la recuperación con Firebase.



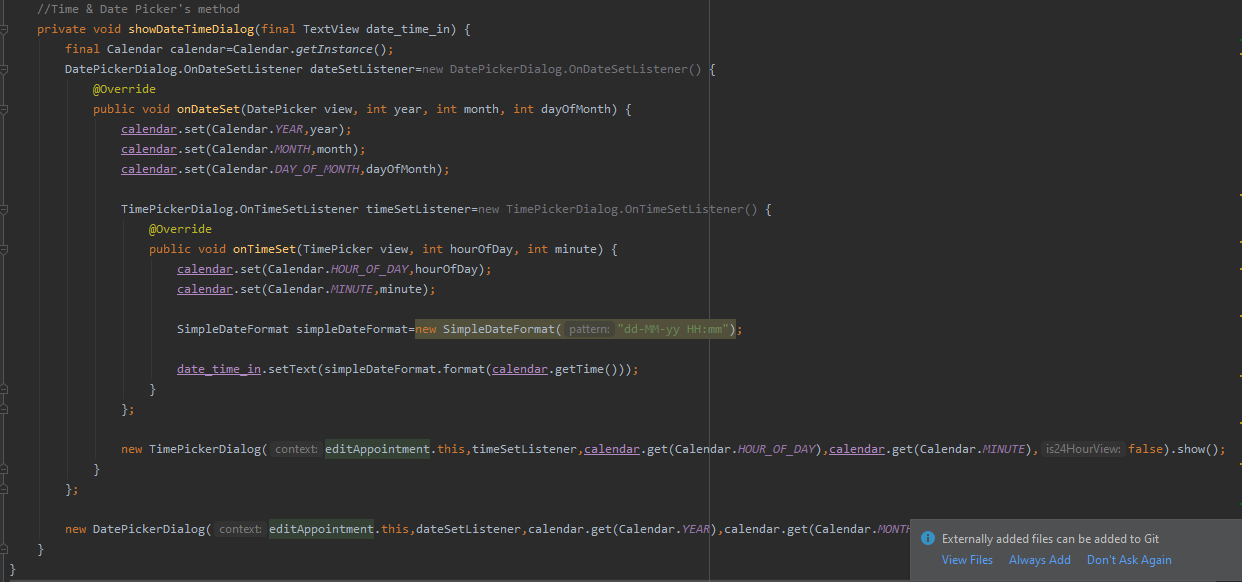


#### editAppointment.java

En esta clase se configura el “editar” o modificar la información de las citas de cada usuario.

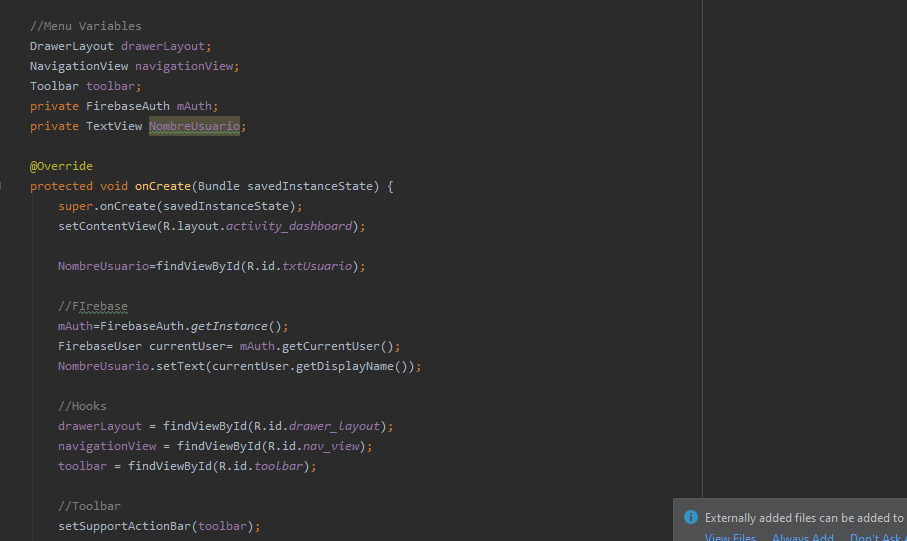


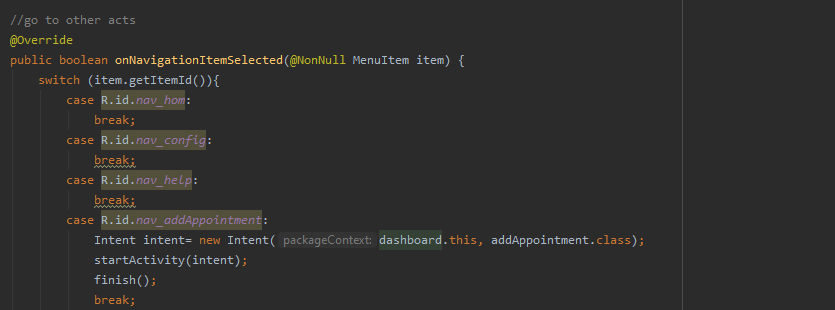


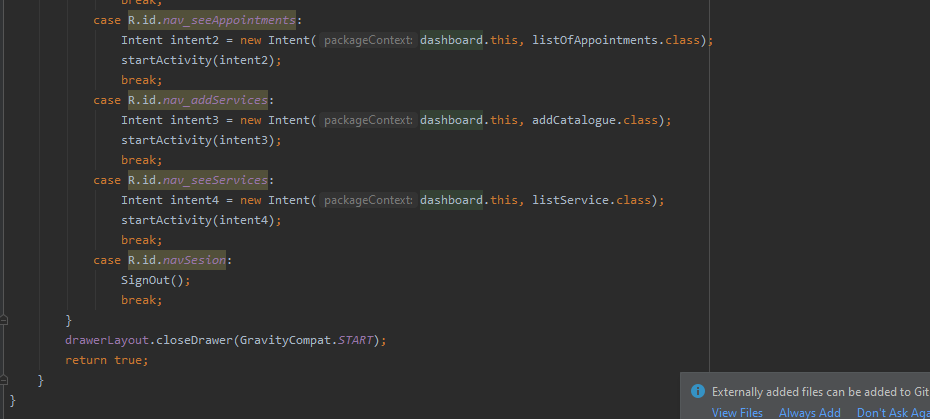


#### dashboard.java

En esta clase se configura el menú lateral de la aplicación, programando cada clasificación u opción que tiene cada usuario para navegar dentro de la aplicación.

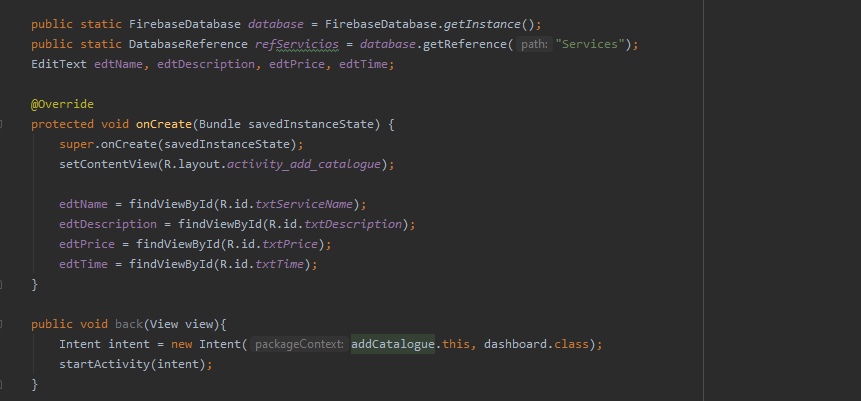






#### addCatalogue.java

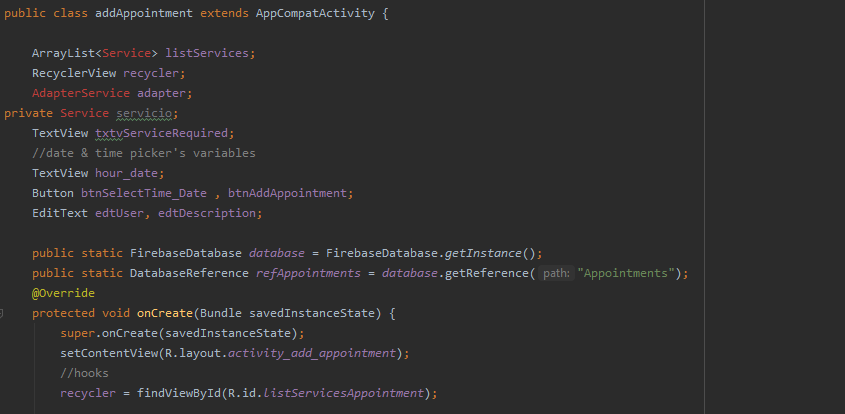
En esta clase se declaran los métodos que se utilizan para agregar los servicios al catálogo, siempre que sea un usuario administrador.

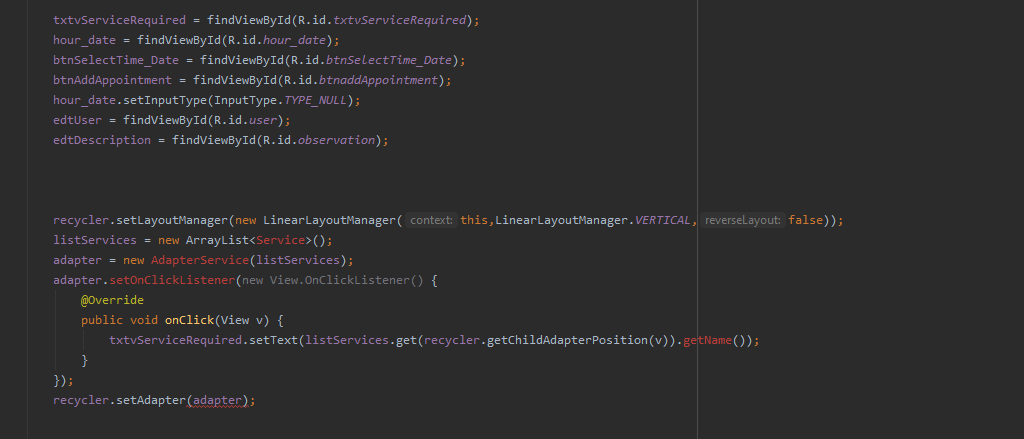


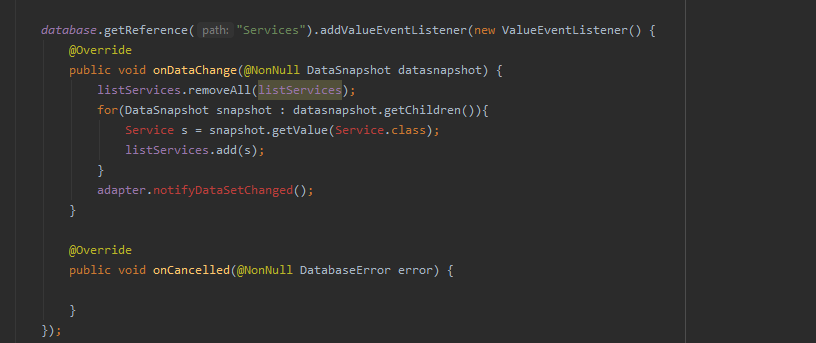


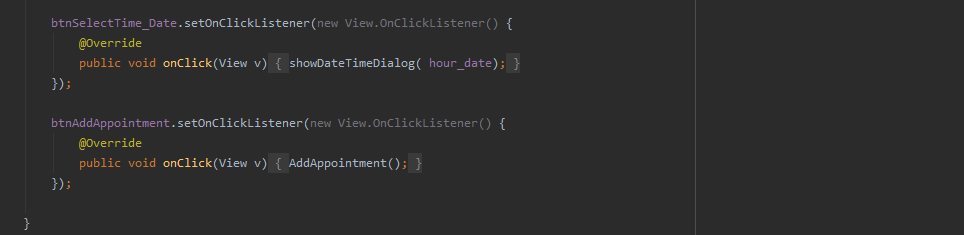
#### addAppointment.java

En esta clase se programa el “insertar” de las citas de los usuarios. Se han empleado alrededor de 5 métodos para el agregar citas para cada usuario. Algunos de estos son para obtener la información de cada elemento donde se ha seleccionado.









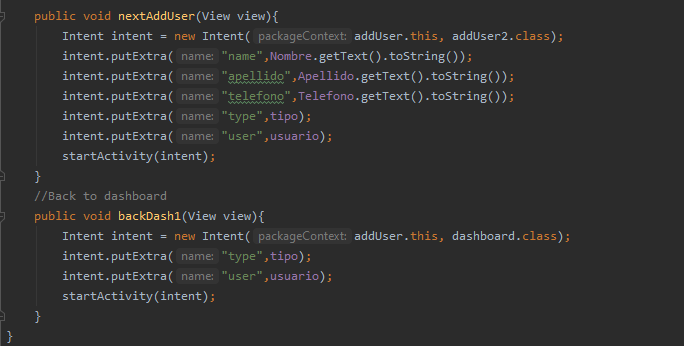




#### addUser.java

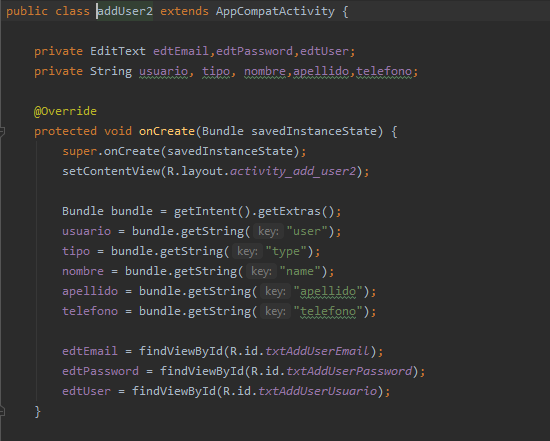
En esta clase se programa el “agregar” un nuevo usuario, que tendrá acceso a la aplicación. En esta clase se programa el primer paso, en el cual se especifican: nombre, apellido y número de teléfono. Luego se encuentra el botón ‘siguiente’ que permite avanzar hacia el siguiente paso.



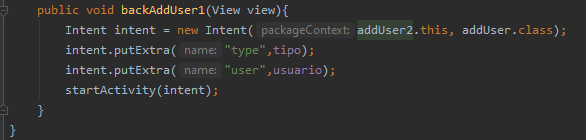


#### addUser2.java

En esta clase se programa el segundo paso del “agregar” un nuevo usuario, que tendrá acceso a la aplicación. En esta clase se obtienen los datos de: correo, contraseña y usuario. Además, se permite avanzar al siguiente paso.

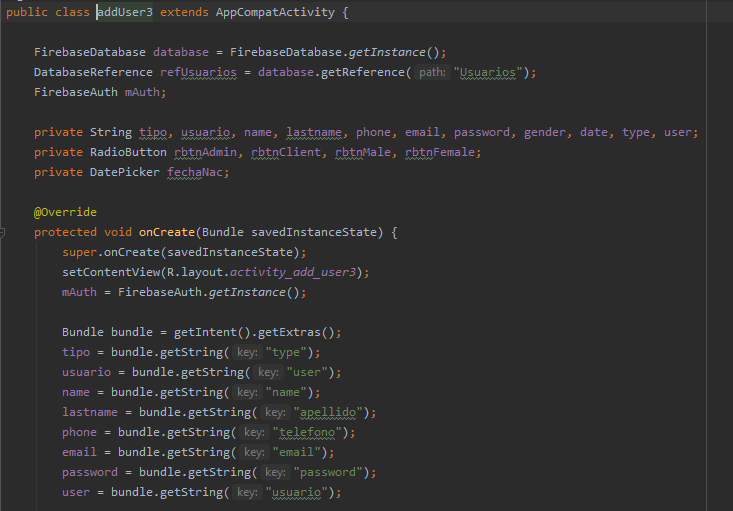




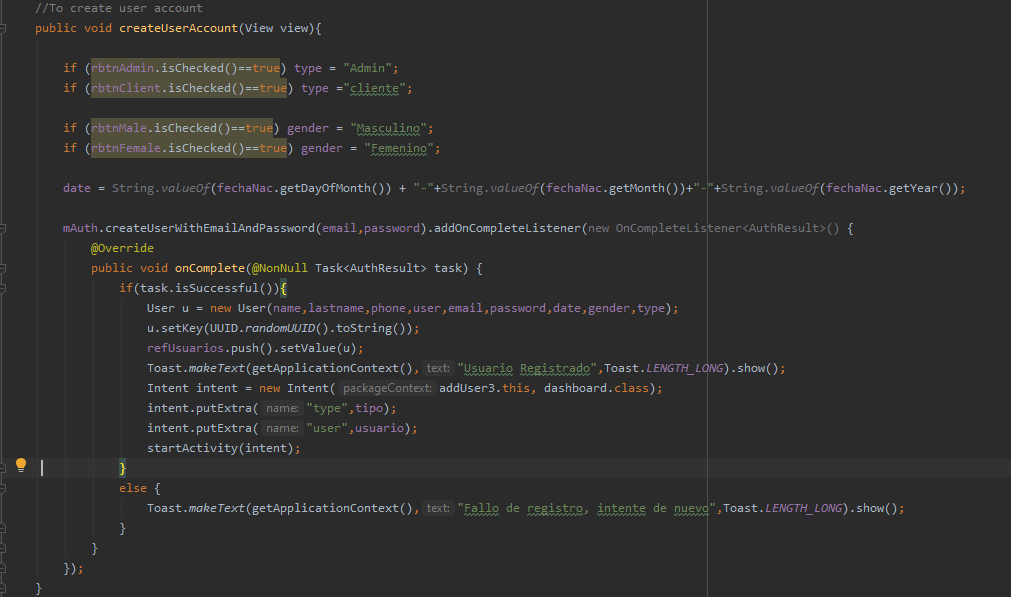


#### addUser3.java

En esta clase se programa el último paso del “agregar” un nuevo usuario, que tendrá acceso a la aplicación. En esta clase se obtienen los datos de: genero de usuario, tipo de usuario y la fecha de nacimiento. Además, se encuentra el botón de agregar usuario.

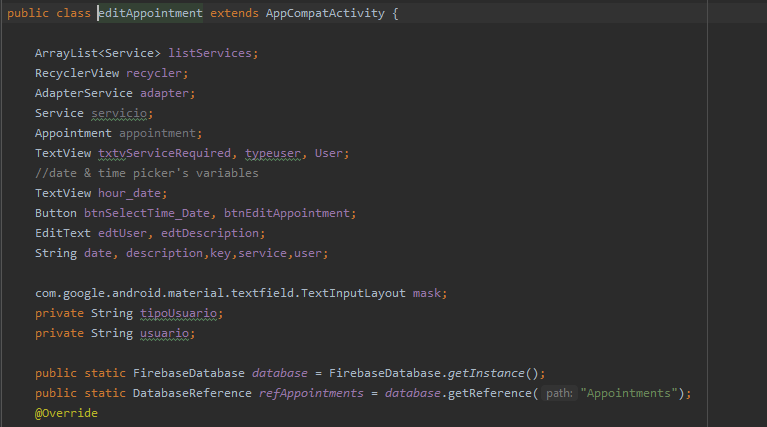






#### editAppointment.java

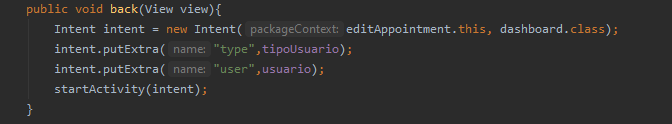
En esta clase se programa la opción “editar” citas según el usuario. Las opciones a modificar son: la hora de la cita, el servicio y agregar una breve descripción de la cita. Luego se presentan los botones de “editar” y “cancelar”.



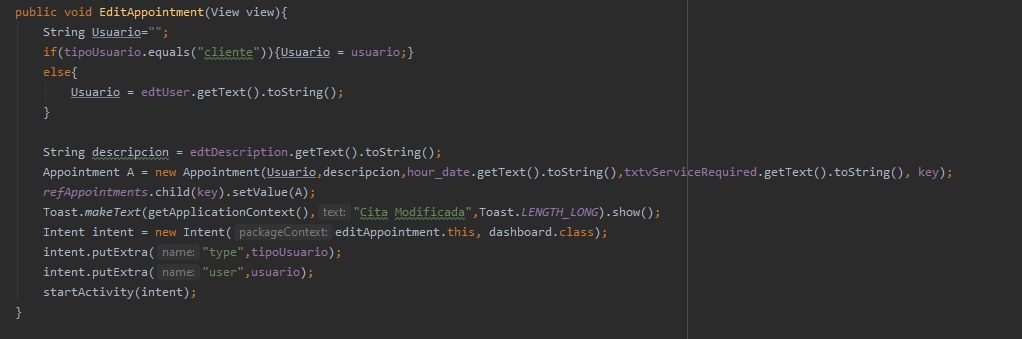






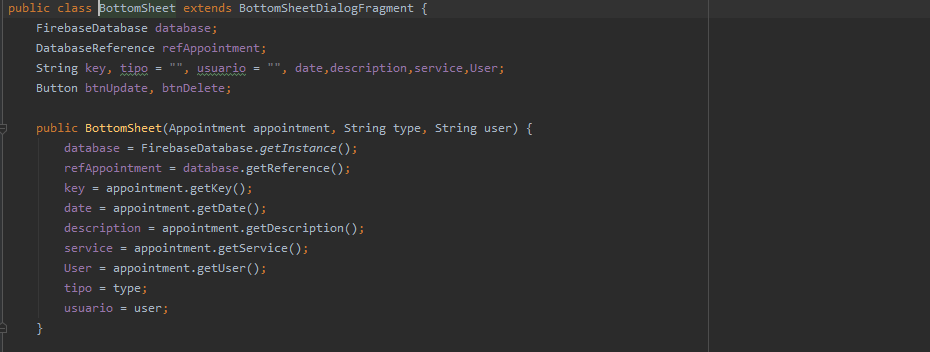






#### BottomSheet.java

En esta clase se programa la opción “Eliminar” una cita. Antes de eliminarla, se ha programado una notificación de confirmación de acción.



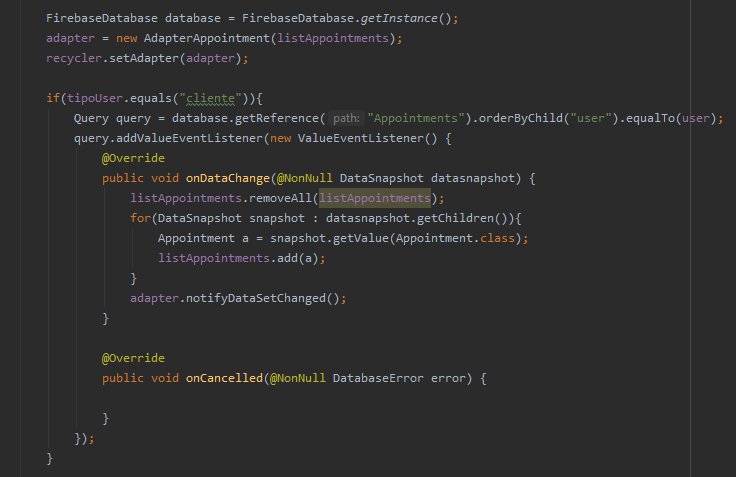


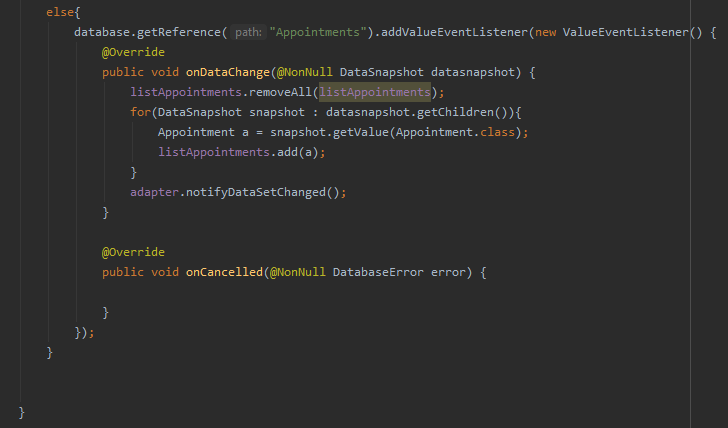


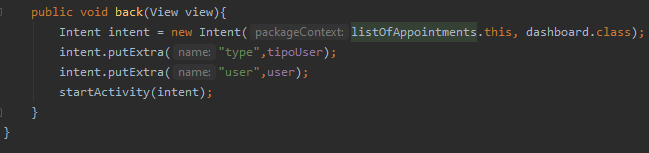
#### ListOfAppointments.java

En esta clase se programa la opción “Ver” citas por usuario. En ella se acumulan las citas que tiene programadas el usuario, es decir su historial. Hace referencia al modelo llamado *Appointments.*

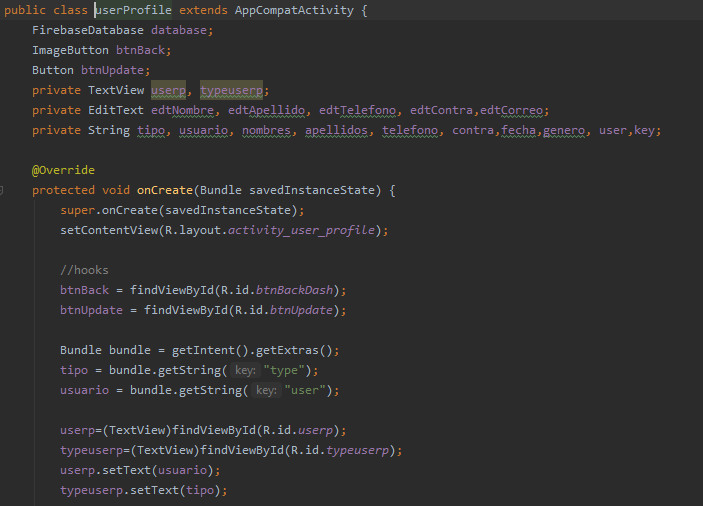


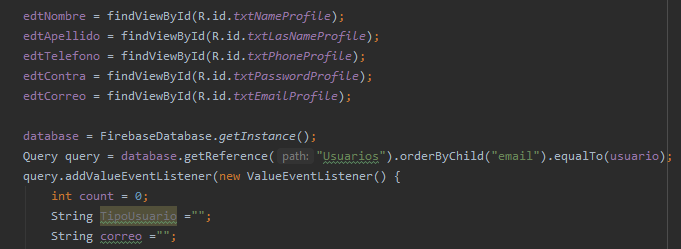




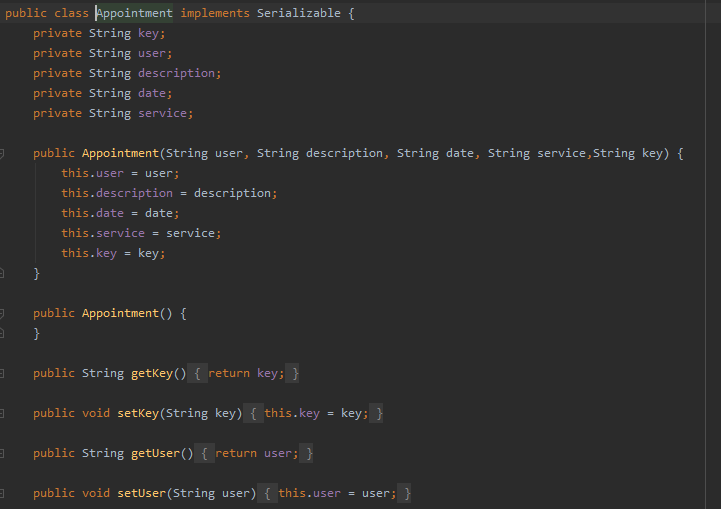


#### UserProfile.java

En esta clase se muestra el perfil con la información básica del usuario. Además, le permite modificar información como el nombre, apellido, correo electrónico, contraseña y el número de teléfono.



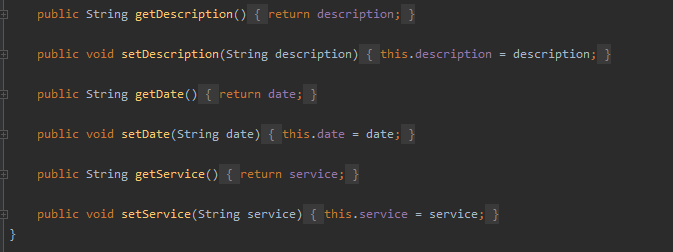




### **MODELS**

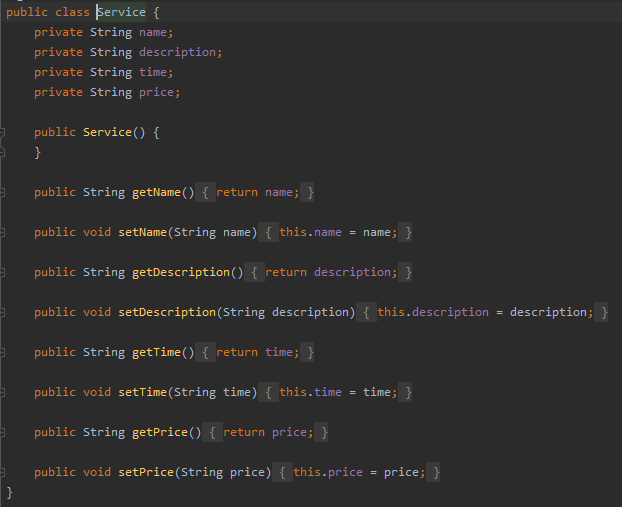
#### Appointment.java

En este modelo, se almacenan las variables que capturan la información que se ingresa sobre las citas. Para posteriormente, añadirlas en una lista en la clase llamada “ListOfAppointments”.



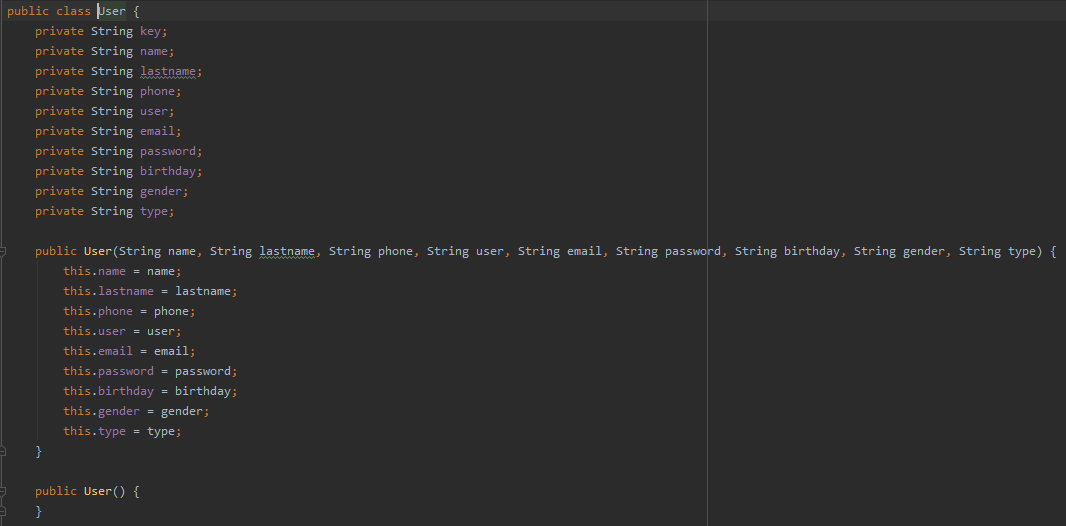
#### Service.java

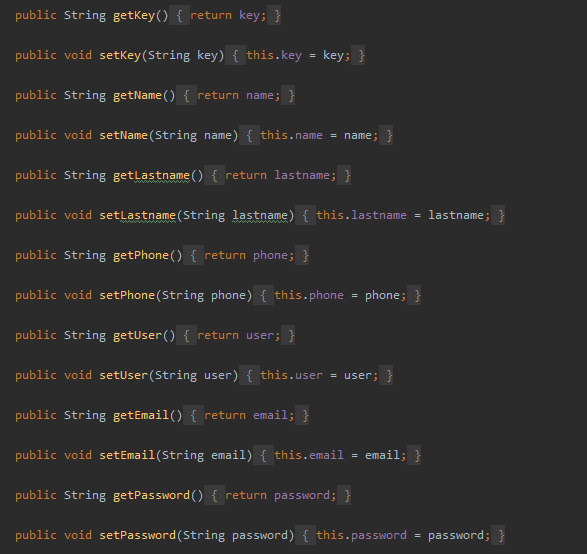
En este modelo, se almacenan las variables que capturan la información que se ingresa sobre los servicios. Para posteriormente, añadirlas en una lista en la clase llamada “ListService”.

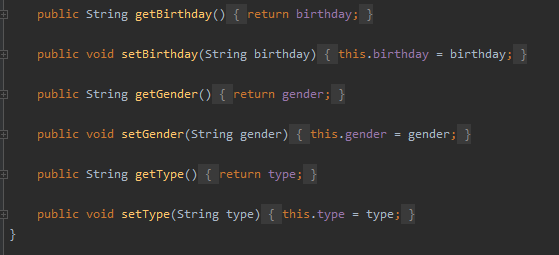


#### User.java

En este modelo, se almacenan las variables que capturan la información que se ingresa sobre los usuarios. Para posteriormente, referenciarlas en userProfile.

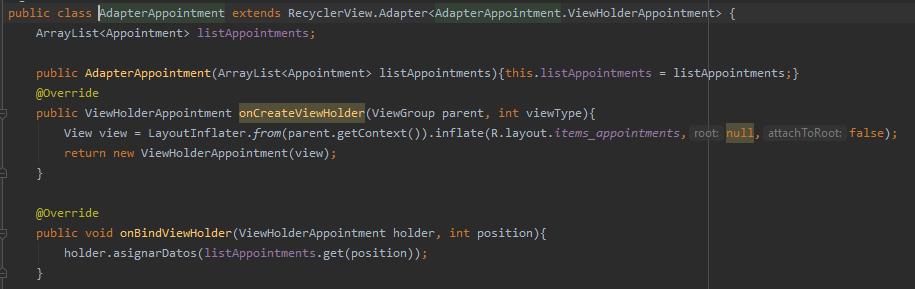






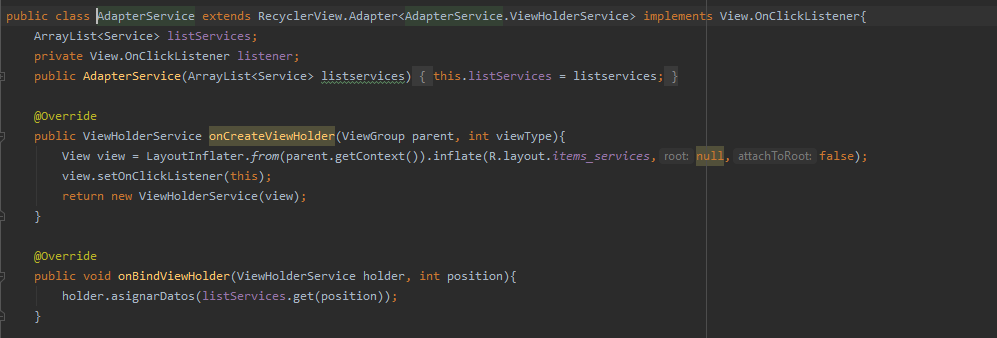
### **ADAPTERS**

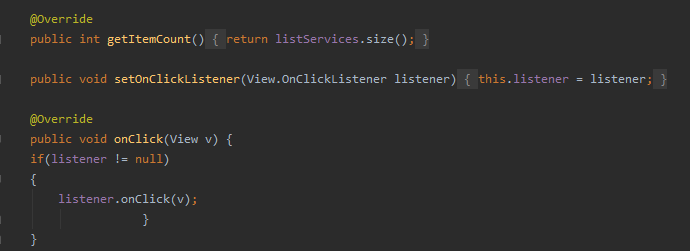
#### AdapterAppointment.java

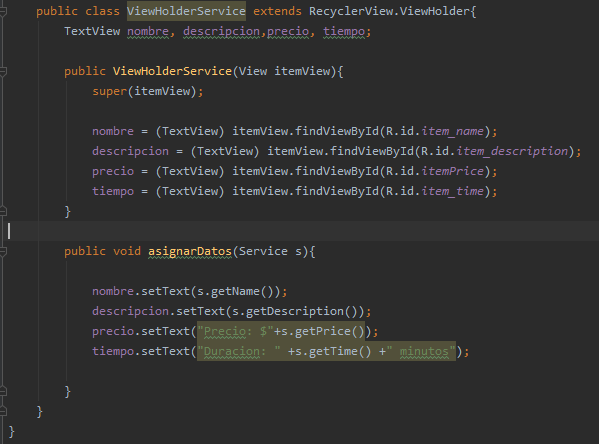




#### AdapterService.java







#### AdapterAppointmentDashboard.java

