

Login:

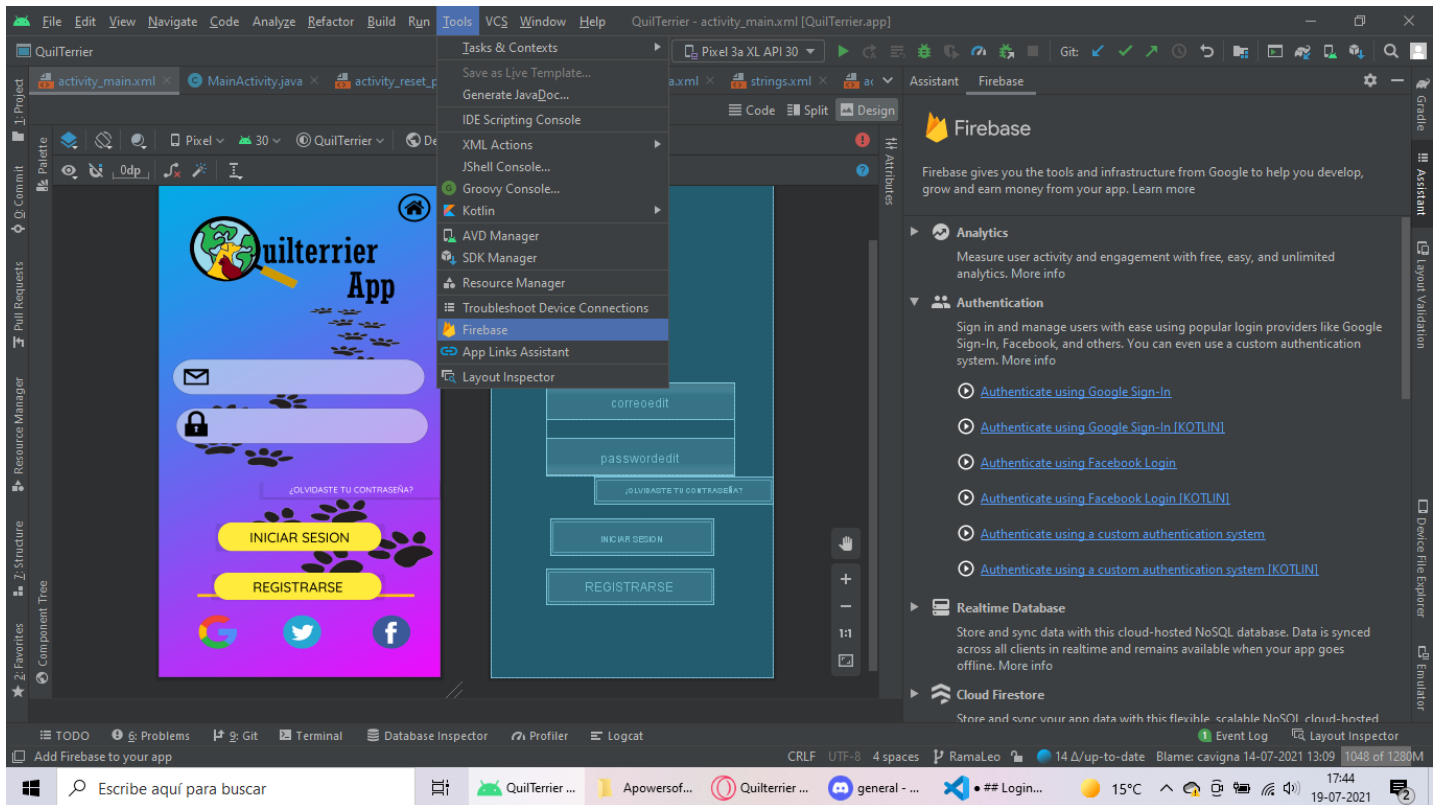
Para la aplicación se tiene pensado utilizar un sistema de Login (soportado en Android Firebase), con el fin de diferenciar entre usuario que puedan solamente ver el contenido de la app y los que puedan ver y realizar comentarios o solicitudes de búsqueda o avisar sobre mascotas encontradas.

Así, con esto en mente, el primer paso fue la implementación inicial del sistema de Login como se presenta a continuación (imágenes ilustrativas no representan la apariencia final de la aplicación, son solo un modo de representar visualmente el avance y el código de la misma):

1. Creación de las distintas interfaces de usuario, en este caso 4: una para Login, otra para recuperar contraseña, otra para crear cuenta y una última para cerrar sesión.

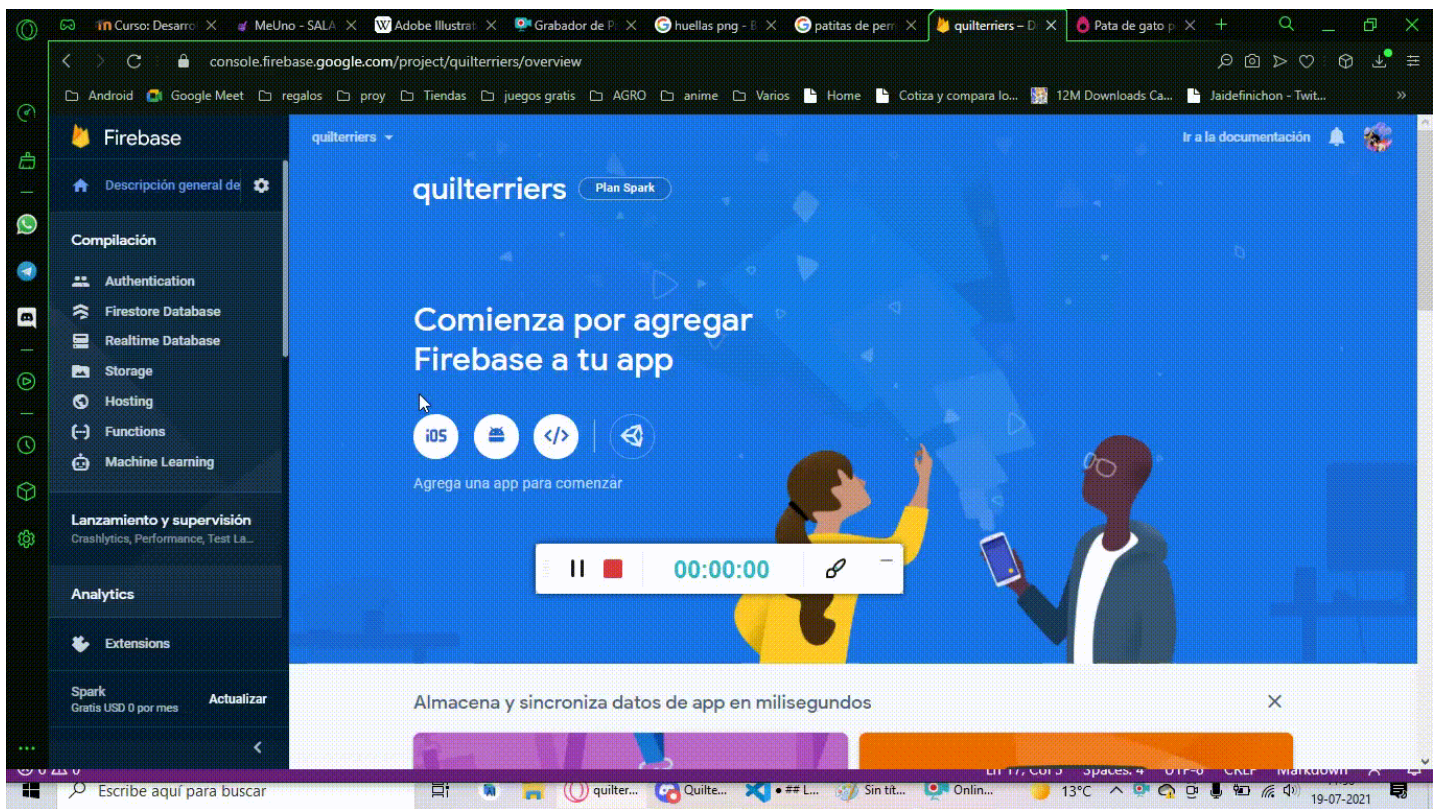


2. Una vez realizado lo anterior se realiza la conexión con Firebase (teniendo previamente una cuenta ya creada en el servicio). Esto se hace desde el menú Tools/Firebase y en la pantalla desplegable al lado izquierdo, seleccionando la opción Authentication.

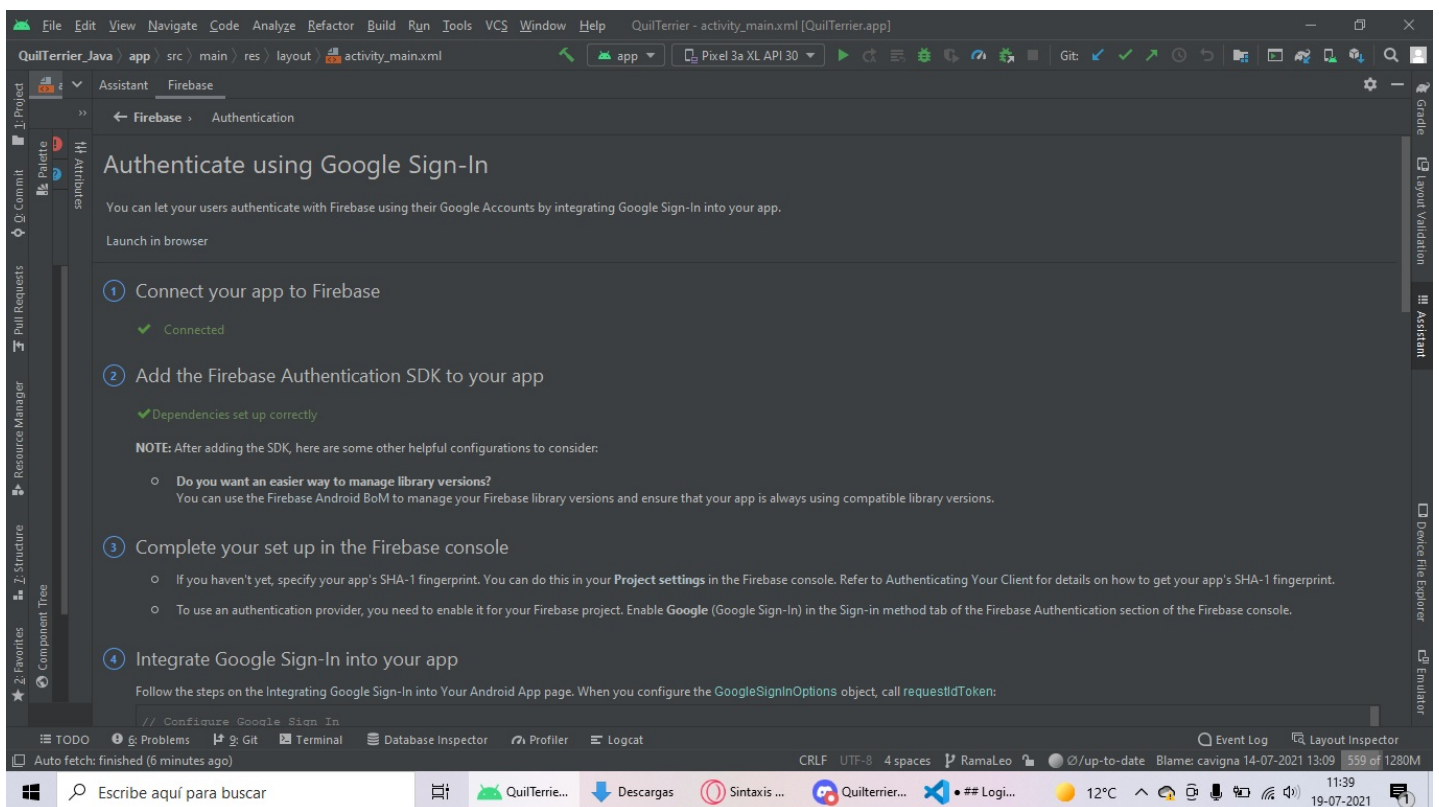


3. En Authentication seguir los pasos del asistente:

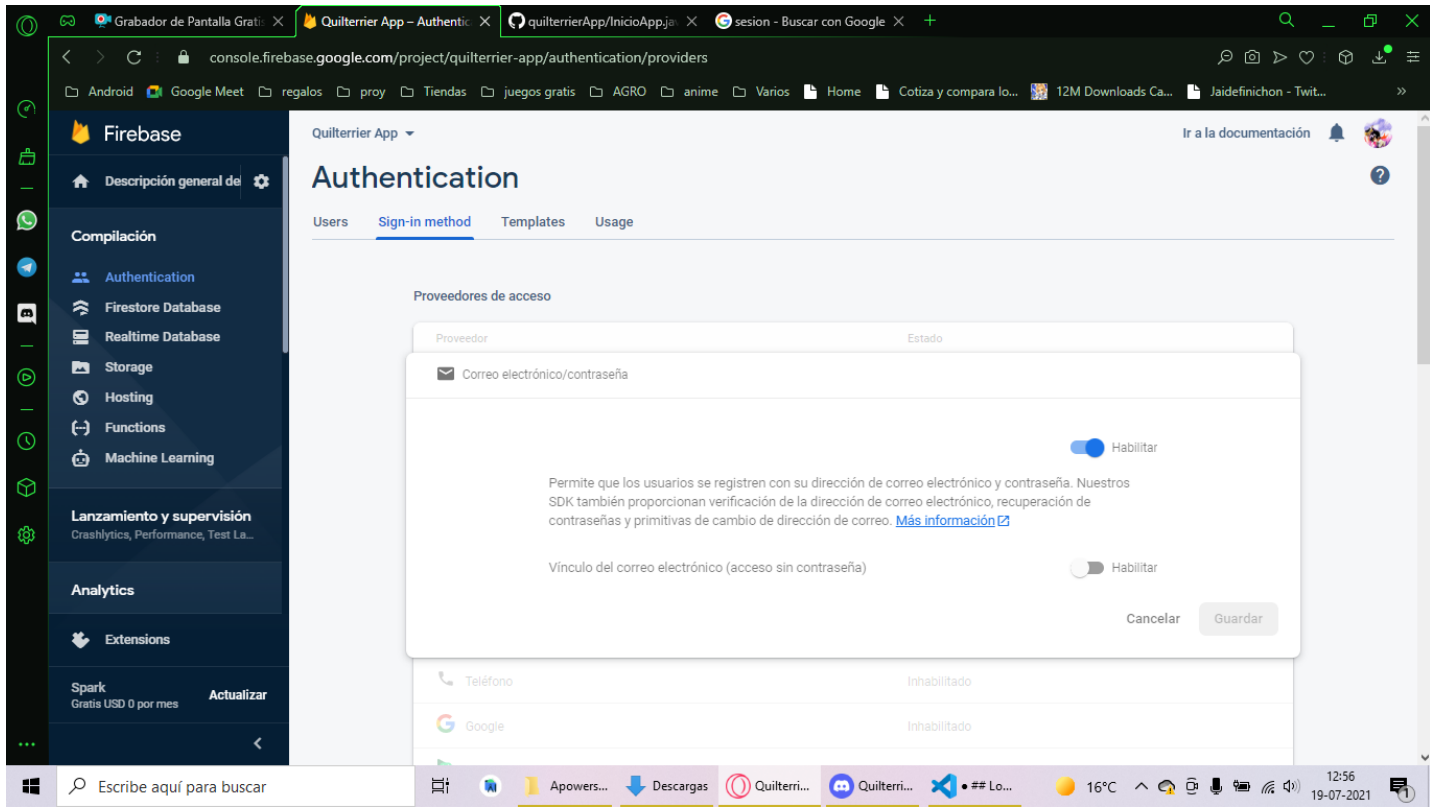
- Conectar app a Firebase: Este paso si bien se indica que funciona de manera automática, en mi caso en particular no fue así. Para esto se siguen las instrucciones a través del sitio web de Firebase, las cuales se resumen en: conexión de app, inscripción del proyecto en la base de datos, descarga del archivo google-services.json (el cual debe ser descargado y copiado de manera manual en la carpeta app/res de la aplicación) y verificado manual de las dependencias en los gradle project y modulo en la configuración de la app.



- En el siguiente paso, instalar la dependencias (dar click y si instalan de manera automatica. Tomar en cuenta que luego de esto el Gradle se va a sincronizar, por lo que cargará y se demorará un poco).



- Luego, desde la p+agina web de Firebase, en la consola de control seleccionar la opcion de logeo con correo y contraseña y habilitarla.



- Una vez realizado esto y sincronizado la app y aplicado las dependencias (a veces despues de este paso es necesario reiniciar Android Studio para que considere todos los cambios realizados) se comienza a codificar y a implementar los métodos correspondientes acada funciónn dentro del login. Estos métodos se detallan a continuación (incluidos sus comentarios en las líneas de código):

Método Login

```

package com.example.quilterrier;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

public class MainActivity extends AppCompatActivity {

    private EditText correo, contrasena;
    private FirebaseAuth nAuth; //objeto de la clase Firebase

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

        correo = findViewById(R.id.correoedit);
        contrasena = findViewById(R.id.passwordedit);

        nAuth = FirebaseAuth.getInstance(); //se instancia el objeto
    }

    public void restablecerPassword (View v) {

        Intent i = new Intent(this, resetPassword.class);
        startActivity(i);

    }

    public void crearCuenta (View v) {

        Intent i = new Intent(this, ActivityCrearCuenta.class);
        startActivity(i);

    }

    //Si inicia la app, que revise si ya tiene iniciada la sesión

    public void onStart (){

        super.onStart();
        //cheque si ya esta logueado y actualiza la UI si ese es el caso
        FirebaseUser currentUser = nAuth.getCurrentUser();
        //UpdateUI(currentUser); //comentado por que por ahora no se va a utilizar a menos que alguien ya esté logueado

    }

    //Acá va el método que lo que va a permitir es iniciar la sesión (antes hay que registrarse, si no es así, no resulta)

    public void iniciarSesion (View view) {

        // String contrasena = getText().toString();
        // String correo = getText().toString();

        /*if (!contrasena.isEmpty() && !correo.isEmpty()){ */
        nAuth.signInWithEmailAndPassword(correo.getText().toString(), contrasena.getText().toString())
            .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {
                    if (task.isSuccessful()) {
                        //Login exitoso, actualiza la UI del usuario con la información de registro
                        FirebaseUser user = nAuth.getCurrentUser();
                        //updateUI(user);
                        //si se loguea correctamente, al dar click al boton que sea enviado a la activity de inicio de perdidos y adopciones en la app
                        //Este Intent tiene que ser modificado para que te lleve al inicio de la app
                        Intent i = new Intent(getApplicationContext(), InicioApp.class);
                        startActivity(i);

                    }else{
                        //Si el logueo falla, tirele un mensaje al usuario por equivocarse
                        Toast.makeText(getApplicationContext(), "Logueo falló", Toast.LENGTH_SHORT).show();
                        //updateUI(null);
                    }
                }
            });
        /* }else{
            Toast.makeText(getApplicationContext(), "Debes llenar todos los campos solicitados", Toast.LENGTH_SHORT).show();
        }
        */
    }
}

```

```

package com.example.quilterrier;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

public class ActivityCrearCuenta extends AppCompatActivity {

    private FirebaseAuth nAuth;
    private EditText correo, contrasena, confimacontrasena;

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

        nAuth = FirebaseAuth.getInstance();

        correo = findViewById(R.id.correoedit);
        contrasena = findViewById(R.id.passwordedit);
        confimacontrasena = findViewById(R.id.passwordedit2);
    }

    public void volver(View v) {

        Intent i = new Intent(this, MainActivity.class);
        startActivity(i);

    }

    //Método para hacer una autenticación previa al logueo?? --> x Leo, esto entendí

    public void onStart() {

        super.onStart();
        //Chequea que el usuario se suscribe (no null) y updatea la UI de acuerdo a la suscripción
        FirebaseUser currentUser = nAuth.getCurrentUser();
        //updateUI(currentUser); //Este método se usa una vez que el usuario ya está logueado, por lo mismo, al inicio se comenta para que no genere errores
    }

    //Método para registrar el usuario

    public void registrarUsuario (View view) {

        //Este primer condicional es para revisar que las dos contraseñas sean iguales, en caso contrario el registro fallará

        if (contrasena.getText().toString().equals(confimacontrasena.getText().toString())) {

            nAuth.createUserWithEmailAndPassword(correo.getText().toString(), contrasena.getText().toString())
                .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
                    @Override
                    public void onComplete(@NonNull Task<AuthResult> task) {

                        if (task.isSuccessful()) {
                            //Registro exitoso, actualiza la UI del usuario con la información de registro
                            Toast.makeText(getApplicationContext(), "Usuario creado con éxito", Toast.LENGTH_SHORT).show();
                            FirebaseUser user = nAuth.getCurrentUser();
                            //si se registra correctamente, al dar click al boton que sea enviado a la main activity para que ahora escoja la opcion de loguearse con la cuenta recientemente creada
                            Intent i = new Intent(getApplicationContext(), MainActivity.class);
                            startActivity(i);

                            //updateUI(user); //comentado por la misma razon que el otro updateUI
                        } else {
                            //Si el registro falla, pasa lo siguiente (arroja esta parte del método y mensajes)

                            Toast.makeText(getApplicationContext(), "Fallo de autenticación", Toast.LENGTH_SHORT).show();
                            //updateUI(null); //comentado por la misma razon que los otros dos updateUI
                        }
                    }
                })
        }

    }

    } else {
        Toast.makeText(this, "Ambas contraseñas no son iguales", Toast.LENGTH_SHORT).show();
    }
}

```



```

package com.example.quilterrier;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.app.AlertDialog;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;

public class resetPassword extends AppCompatActivity {

    private EditText eIcorreo;
    private Button btnrestablecer;

    private String email = ""; //Variable local que mantiene almacenado lo que el usuario escribe en el correo
    private FirebaseAuth mAuth; //Esta variable u objeto se crea para usarla al restablecer el password

    //Se crea un progress con un loader, diciendo que se está ejecutando el envío del correo de restablecimiento
    private ProgressDialog mDialog;

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

        //Instancio las variables en el onCreate
        mAuth = FirebaseAuth.getInstance();
        eIcorreo = findViewById(R.id.edittextcorreo);
        btnrestablecer = findViewById(R.id.btnrestablecer);
        mDialog = new ProgressDialog(this); //Recibe como parametro el contexto = esta actividad = this.

        //se crea este método en el onCreate para que cuando el usuario de click al boton "restablecer" se active el metodo de cambiede contraseña
        btnrestablecer.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick (View view) {

                //Se trae al método lo que el usuario escribe en el editText (id edittextcorreo) y lo guarda en la variable anterior llamada email
                email = eIcorreo.getText().toString();

                //Se ejecuta una validación en donde comprueba que: si el campo editText (id edittextcorreo) NO esta vacio, ejecuta el método resetPassword
                //en caso contrario, arroja el mensaje de advertencia con el Toast.
                if (!email.isEmpty()){
                    //Si el usuario ingreso un correo se debe mostrar el progress dialog instanciado antes
                    mDialog.setMessage("Espera un momento...");
                    //el mDialog.setCanceledOnTouchside (false) es para que el usuario no puede quitar el mensaje tocando en algun lado
                    // mientras se esta ejecutando la función reset password
                    mDialog.setCanceledOnTouchOutside(false);
                    //el .show() va a hacer que los mensajes anteriores se muestren
                    mDialog.show();
                    //Luego del progress dialog, comienza a ejecutar el método (o al mismo tiempo, no es, es rápido esto, la cosas es que el mensaje y luego el método)
                    resetPassword();
                }else{
                    Toast.makeText(resetPassword.this, "Debe ingresar un correo", Toast.LENGTH_SHORT).show(); //ADVERTENCIA DE TOAST
                }
            }
        });
    }

    private void resetPassword (){
        //El primer paso que se utiliza, es establecer el idioma en que debe llegar el correo para restablecer la contraseña, aunque
        //no se si ahora es necesario aún, de todos modos por el tutorial se establece en español (eso significa la parte de "es")
        mAuth.setLanguageCode("es");
        //Se establece el método que va a recibir el email a ser validado y a enviar el correo de restablecimiento de contraseña.
        mAuth.sendPasswordResetEmail(email).addOnCompleteListener(new OnCompleteListener<Void>() {
            @Override
            public void onComplete(@NonNull Task<Void> task) {

                //Evalua si la tarea (task = enviar el correo al usuario) es exitosa, de ser así, Toast (a), si no ejecuta Toast (b)
                if(task.isSuccessful()){
                    Toast.makeText(resetPassword.this, "Se ha enviado un correo a "+email+" para reestablecer tu contraseña", Toast.LENGTH_SHORT).show(); //Toast (a)
                }else{
                    Toast.makeText(resetPassword.this,"No se pudo enviar el correo de reestablecer contraseña - culpa de Yayson", Toast.LENGTH_SHORT).show(); // Toast (b)
                }

                //en referencia al mDialog anterior, cuando la tarea termine de ejecutarse, el mensaje del Dialog se va ocultar, con el siguiente método
                mDialog.dismiss();
            }
        });
    }
}

```

Método cerrar sesión:

```

package com.example.quilterrier;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;

import com.google.firebase.auth.FirebaseAuth;

public class InicioApp extends AppCompatActivity {

    FirebaseAuth nAuth;

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

    public void cerrarSesion (View view) {

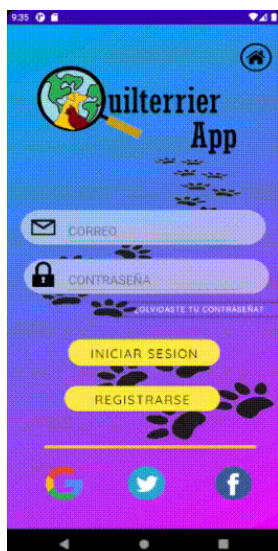
        nAuth.signOut();
        Intent i = new Intent(this, MainActivity.class);
        startActivity(i);
    }
}

```

De lo anterior se desprende lo siguiente, teniendo en cuenta que se inicia la cuenta de Firebase/Quilterrier sin ningun usuario como se observa a continuación:

The screenshot shows the Firebase Authentication console for the 'Quilterrier App' project. The left sidebar contains the Firebase navigation menu with options like 'Descripción general de', 'Compilación', 'Authentication', 'Firestore Database', 'Realtime Database', 'Storage', 'Hosting', 'Functions', 'Machine Learning', 'Lanzamiento y supervisión', 'Analytics', 'Extensions', and 'Spark'. The main content area is titled 'Authentication' and has tabs for 'Users', 'Sign-in method', 'Templates', and 'Usage'. The 'Users' tab is active, showing a table with columns: 'Identificador', 'Proveedores', 'Fecha de creación', 'Fecha de acceso', and 'UID de usuario'. A blue 'Agregar usuario' button is in the top right of the table. Below the table, a message states: 'Este proyecto todavía no tiene usuarios'. The browser's address bar shows 'console.firebase.google.com/project/quilterrier-app/authentication/users'. The bottom of the screen shows a Windows taskbar with various application icons and a system clock indicating 12:58 on 19-07-2021.

Se realiza como prueba de logueo en la app, obteniendo lo siguiente:



En el gif demostrativo en primera instancia se intenta loguear sin haber creado antes una cuenta, por lo que es de esperar que se genere un error (mostrado en el Toast como "Logueo falló").



Quilterrier App - Authentic...

console.firebase.google.com/project/quilterrier-app/authentication/users

Android Google Meet regalos proy Tiendas juegos gratis AGRO anime Varios Home Cotiza y compara lo... 12M Downloads Ca... Jaidefinichon - Twit...

Quilterrier App

Authentication

Users Sign-in method Templates Usage

Crea prototipos y haz pruebas de extremo a extremo con Local Emulator Suite, que ahora es compatible con Firebase Authentication. [Comenzar](#)

Buscar por dirección de correo electrónico, número de teléfono o UID de usuario [Agregar usuario](#)

Identificador	Proveedores	Fecha de creación	Fecha de acceso	UID de usuario ↑
leo.rodenas.e@gmail.com	✉	19 jul. 2021	19 jul. 2021	U9BwJ4ETn0ZyWh7XoLoh1hXCv...

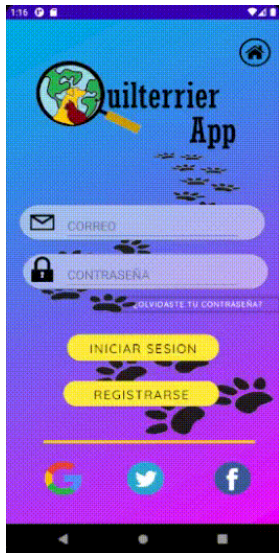
Filas por página: 50 1 - 1 of 1

Analytics Spark Actualizar

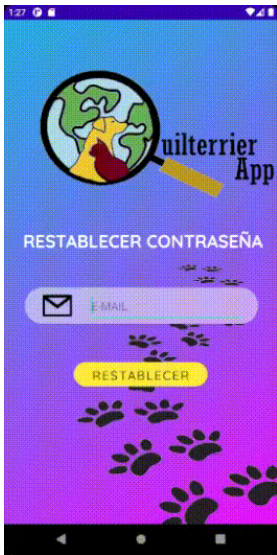
Escribe aquí para buscar

Quil... Apo... Apo... Quil... Quit... Andr... Previ... 9°C 21:31 19-07-2021

Luego se crea la cuenta, de manera que en un primer intento las contraseñas no sean iguales, por lo que arroja un Toast diciendo efectivamente esto, luego si se procede a crear la cuenta exitosamente (con un Toast de aviso "Usuario creado con éxito").
Notar que en la base de datos de Firebase ahora si aparece el correo registrado.

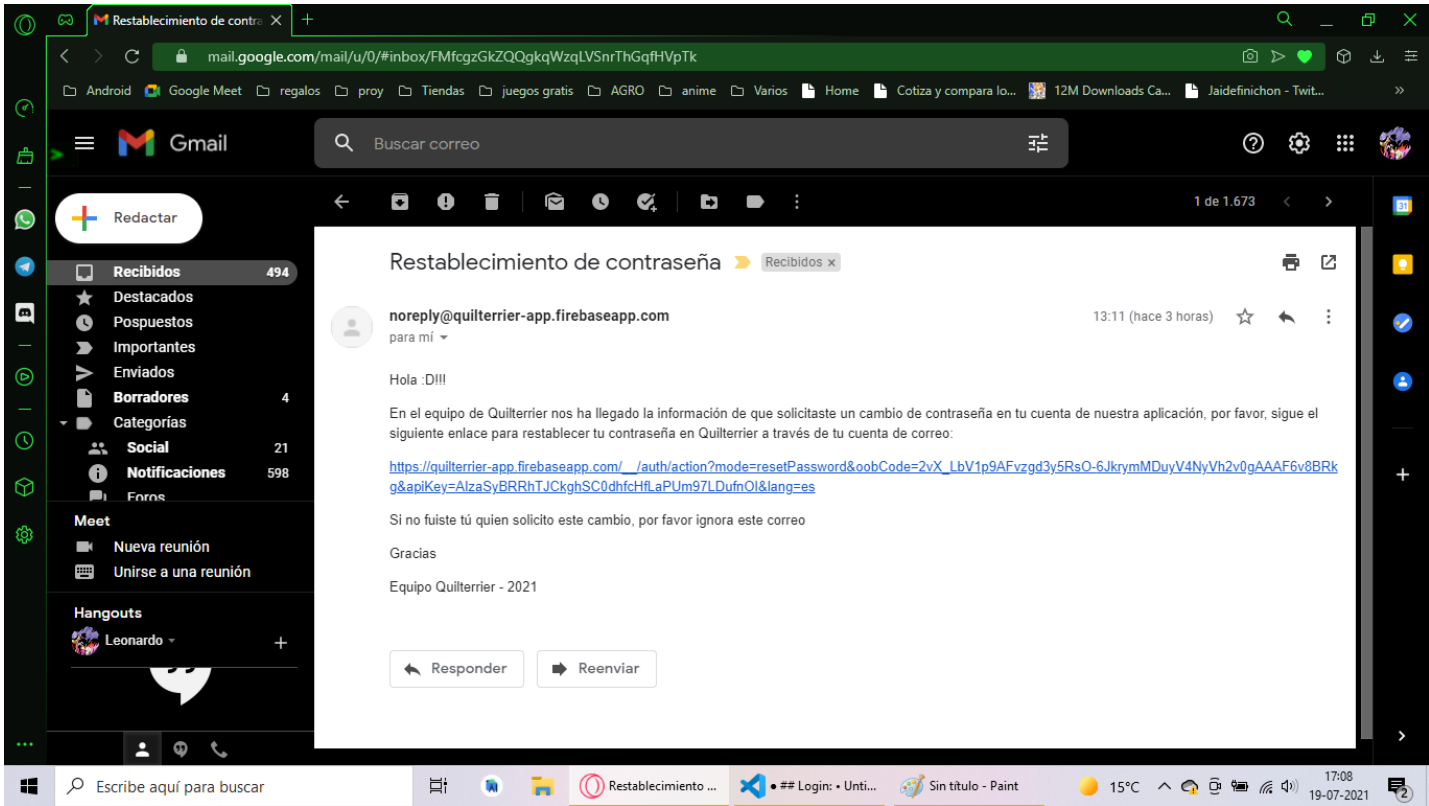


Siguiente con la cuenta creada exitosamente nos volvemos a loguear y somos enviados a una activity con acceso al boton "Cerrar sesión" desde en donde al darle click nos envia a Login nuevamente para poder loguearnos.



Finalmente, como ejercicio pedimos el restablecimiento de la contraseña con un correo falso (que no posee registro como usuario), generando un Toast de error y despues con un correo registrado para que así de esta manera nos envíe el correo de restablecimiento a nuestro correo.

- Correo de restablecimiento:



De esta manera se implementa el login en la aplicación y se espera que en futuras actualizaciones contenga nuevas funcionalidades diferenciadas entre usuarios registrados y no registrados.