



DOCUMENTACION TICKETBOOM

Gregorio Castillo Jr 19580589
Sheila Lizeth Flores Acosta 19580595

Instituto Tecnológico de México
Instituto Tecnológico de Reynosa
Proyectos de innovación
Patricia Martínez Vázquez

Preparación

Antes de empezar a trabajar tendremos que definir cuál es el objetivo de nuestra aplicación, como queremos que el usuario interactúe con ella y que tipo de problema solucionara, de este modo fue que llegamos a plantearnos el realizar una aplicación para la compra-venta de tickets de conciertos, en la cual el usuario podrá comprar tickets para sus conciertos preferidos y estos se guardaran dentro de la aplicación, con la cual no será necesario descargarlos y puede obtener acceso a ellos desde cualquier dispositivo móvil mientras el usuario tenga su correo y contraseña.

La que se busca de la aplicación es lograr obtener una confianza con los usuarios otorgándoles la facilidad de llevar u obtener algún boleto de su artista favorito sin ningún obstáculo de por medio.

Para el almacenamiento de datos, haremos uso de firebase por parte de Google, esto porque nos ofrece ciertos servicios gratuitos hasta tener un alto flujo de movimientos, tambien facilita la implementación con los sistemas Android, haciéndolo una opción bastante fiable y sencilla para usar.

Desarrollo

Para la primera parte de nuestro trabajo realizaremos 2 activities, la primera será lo correspondiente al menú principal, en la primera se realiza un inicio de sesión en la cual ingresaremos un correo y una contraseña con la que anteriormente ya debimos registrarnos, esto en la parte inferior donde encontramos la opción “Regístrate en [TicketBoom](#)” al dar click en ticketboom este nos mandara a un formulario de registro, adicional a ello también contaremos con la opción de recuperar contraseña en el activity principal en el cual se solicitara el correo del cual quieres recuperar la contraseña.

Esta parte del código nos funciona para declarar los elementos con los que contamos dentro de nuestro activity main, de forma que nos permita hacer uso de EditTexts y TextViews, los cuales nos permiten mostrar un texto(TextViews) y modificar este(EditTexts), tambien tenemos de otro TextView que esta subrayado el cual nos permitirá movernos al activity de registro

```

public class MainActivity extends AppCompatActivity {
    Button test;

    private EditText Iemail, Ipassword ;
    private TextView registro;

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

        Iemail = (EditText) findViewById(R.id.IEmail);
        Ipassword = (EditText) findViewById(R.id.IPassword);

        //Subrayo que lleva al registro de ticketboom
        registro = findViewById(R.id.IRegistrarse);
        String Iregistro = "Registrate en TicketBoom";
        SpannableString ss = new SpannableString(Iregistro);

        ClickableSpan Rticketboom = (widget) -> {
            Intent a = new Intent ( packageContext: MainActivity.this, registro.class);
            startActivity(a);
        };

        ss.setSpan(Rticketboom, start: 14, end: 24, Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
        registro.setText(ss);
        registro.setMovementMethod(LinkMovementMethod.getInstance());
    }
}

```

También contaremos con la siguiente parte del código la cual nos permitirá hacer uso del botón de iniciar sesión

```

//Metodo para iniciar sesion
public void inicioS (View view){
    String email = Iemail.getText().toString();
    String contraseña = Ipassword.getText().toString();

    if(contraseña.length() == 0){
        Toast.makeText( context: this, text: "Ingresa una contraseña porfavor", Toast.LENGTH_LONG).show();
    }
    if(email.length() == 0){
        Toast.makeText( context: this, text: "Ingresa un correo porfavor", Toast.LENGTH_LONG).show();
    }
    if(!email.equals(null) && !contraseña.equals(null)){
        Toast.makeText( context: this, text: "Verificando correo", Toast.LENGTH_LONG).show();
    }
}
}

```

El código siguiente es el que nos permitirá movernos entre el activity main y el activity de registro, para ello haremos uso de un Intent, el cual nos moverá de nuestro MainActivity a nuestro registro

```

registro = findViewById(R.id.IRegistrarse);
String Iregistro = "Registrate en TicketBoom";
SpannableString ss = new SpannableString(Iregistro);

ClickableSpan Rticketboom = (widget) -> {
    Intent a = new Intent ( packageContext: MainActivity.this, registro.class);
    startActivity(a);
};

ss.setSpan(Rticketboom, start: 14, end: 24, Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
registro.setText(ss);
registro.setMovementMethod(LinkMovementMethod.getInstance());
}

```

Tras movernos al activity de registro contaremos con 1 texto, 6 campos de llenado y 1 checkbox, como se puede ver en el código siguiente donde agregaremos 1 Textview 6 EditText y 1 Checkbox con el respectivo campo de TextView con el mensaje "Registro" este para dar una idea al usuario de donde esta, los campos de llenado "Nombre", "Apellido", "Email", "Contraseña", "Confirmar contraseña", "Teléfono" y el checkbox "Términos y condiciones"

```

private EditText Rnombre, Rapellido, Remail, Rcontraseña, RCcontraseña, Rtelefono ;
private CheckBox RTyC;

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

    Rnombre = (EditText) findViewById(R.id.RNombre);
    Rapellido = (EditText) findViewById(R.id.RApellido);
    Remail = (EditText) findViewById(R.id.REmail);
    Rcontraseña = (EditText) findViewById(R.id.RContraseña);
    RCcontraseña = (EditText) findViewById(R.id.RCContraseña);
    Rtelefono = (EditText) findViewById(R.id.RTelefono);
}

public void registrar (View view){
    String nombre = Rnombre.getText().toString();
    String apellido = Rapellido.getText().toString();
    String email = Remail.getText().toString();
    String contraseña = Rcontraseña.getText().toString();
    String Ccontraseña = RCcontraseña.getText().toString();
    String Telefono = Rtelefono.getText().toString();
    CheckBox RTyC = findViewById(R.id.RTyCCond);
    boolean datosCompletos = true;
}

```

cada uno de ellos tendrá que ser llenados por el usuario para que pueda completar su registro en la aplicación, de no llenar todos los campos la aplicación solicitara que estos sean llenados, esto lo lograremos gracias a la siguiente parte del código, la cual evalúa

cada uno de los campos a llenar y si estos están vacíos, no procederá el registro, cambiando un valor booleano (datoscompletos) a falso, haciendo que no entre al proceso de mandar datos a firebase

```
if (!RTYC.isChecked()) {
    Toast.makeText( context: this, text: "Tienes que aceptar los términos y condiciones", Toast.LENGTH_LONG).show();
    datosCompletos = false;
}
if(Telefono.length() == 0){
    Toast.makeText( context: this, text: "Ingresa un telefono porfavor", Toast.LENGTH_LONG).show();
    datosCompletos = false;
}
if(contraseña.length() == 0){
    Toast.makeText( context: this, text: "Ingresa una contraseña porfavor", Toast.LENGTH_LONG).show();
    datosCompletos = false;
}
else if(Ccontraseña.length() == 0){
    Toast.makeText( context: this, text: "Confirma tu contraseña porfavor", Toast.LENGTH_LONG).show();
    datosCompletos = false;
}
else if(!contraseña.equals(Ccontraseña)){
    Toast.makeText( context: this, text: "Tu contraseña y la confirmacion no son igual", Toast.LENGTH_LONG).show();
    datosCompletos = false;
}
if(email.length() == 0){
    Toast.makeText( context: this, text: "Ingresa un correo porfavor", Toast.LENGTH_LONG).show();
    datosCompletos = false;
}
if(apellido.length() == 0){
    Toast.makeText( context: this, text: "Ingresa un apellido porfavor", Toast.LENGTH_LONG).show();
    datosCompletos = false;
}
if(nombre.length() == 0){
    Toast.makeText( context: this, text: "Ingresa un nombre porfavor", Toast.LENGTH_LONG).show();
    datosCompletos = false;
}
if(datosCompletos){
```

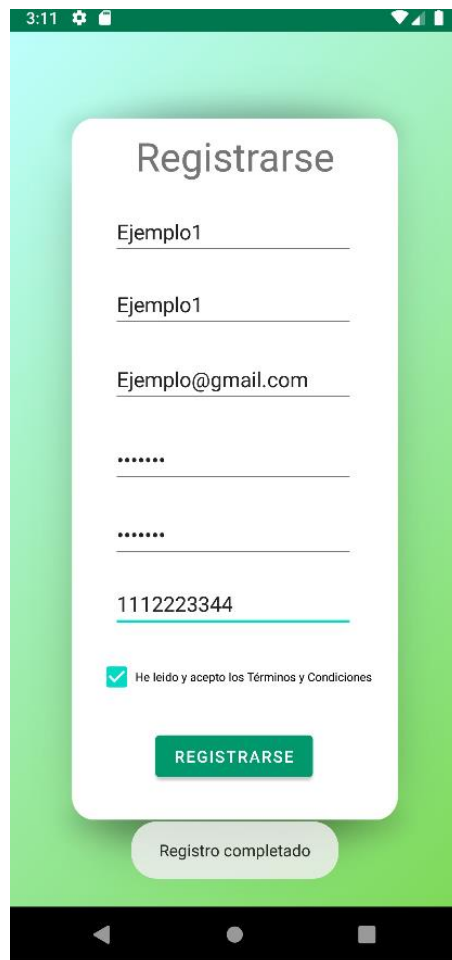
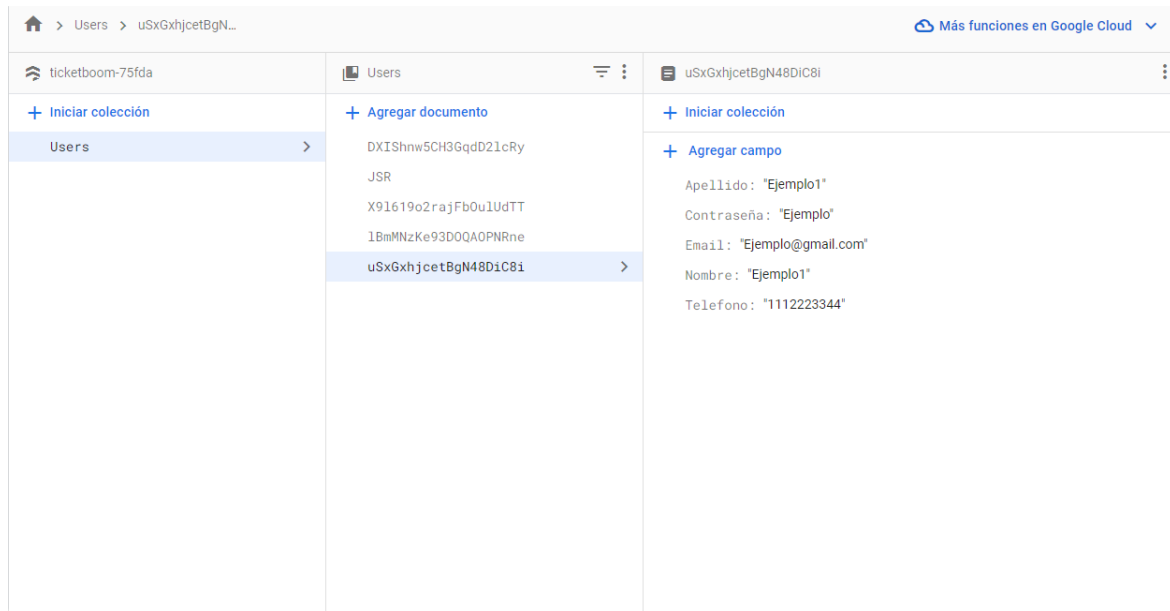
Tras los datos ser confirmados que no están vacíos, el booleano quedara con un valor = true, entonces empezara el proceso de mandar datos a firebase, en el cual se hará referencia al servicio donde deseamos mandar los datos, para después hacer la recolección de datos para guardarlos en un objeto User, de forma que será mandado y registrado en ese entorno, tras ello tendremos una línea que nos permitirá asignarle un id único a los datos, esto para evitar que los datos anteriormente no se sobrescriban, y de todo haber resultado con éxito se mostrara un mensaje de “Registro completado”

```
if(datosCompletos){
    FirebaseFirestore db = FirebaseFirestore.getInstance();

    Map<String, Object> User = new HashMap<>();
    User.put("Nombre", nombre);
    User.put("Apellido", apellido);
    User.put("Email", email);
    User.put("Contraseña", contraseña);
    User.put("Telefono", Telefono);

    db.collection( collectionPath: "Users").document().set(User).addOnCompleteListener(new OnCompleteListener<Void>() {
        @Override
        public void onComplete(@NonNull Task<Void> task) {
            if (task.isSuccessful()){
                Toast.makeText( context: registro.this, text: "Registro completado", Toast.LENGTH_LONG).show();
            }
        }
    });
}
```

De esta forma es como se verian los resultados del registro:



Debugging de código y aplicación

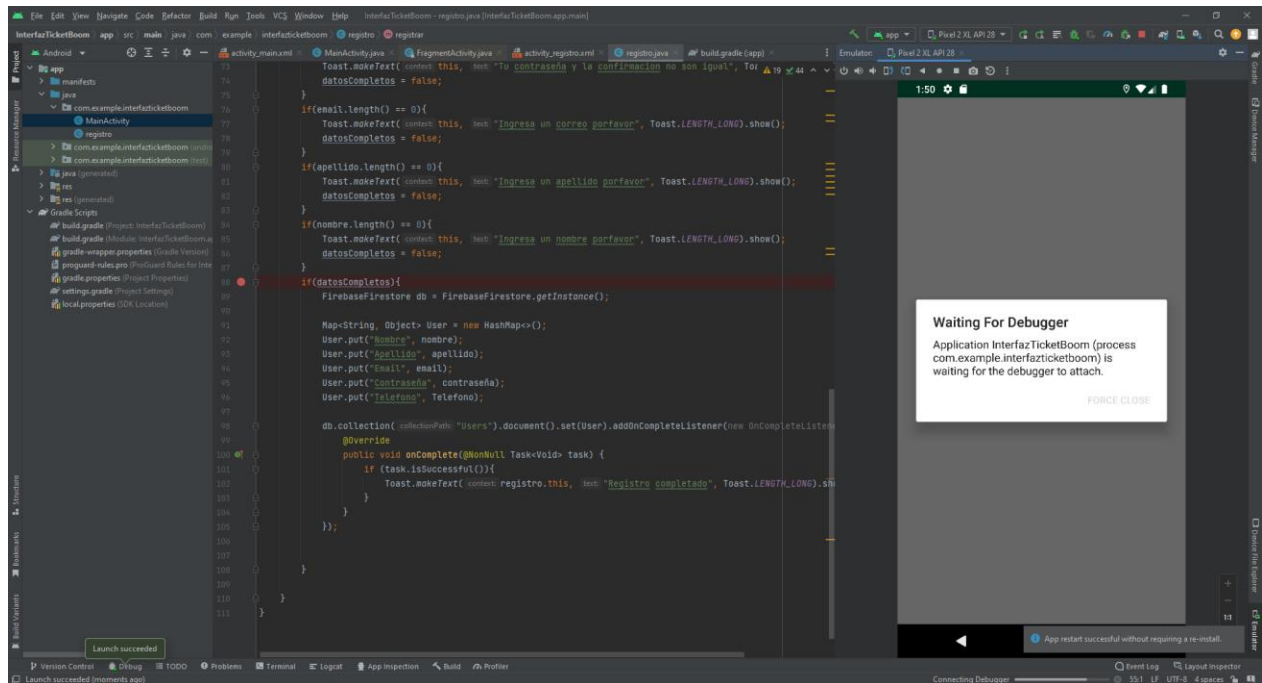
Para el proceso de debbuging tomamos en cuenta (al menos de momento) 4 puntos clave de nuestra aplicación, en los cuales teníamos que prestar mucha más atención estos fueron, el texto de registro, cuando se cambia de activity, cuando se evalúa el inicio de sesión, y cuando se comparan si los datos de registro están llenos, es asi que se vería de la siguiente manera a nivel de código

```
6 String registro = "Registrate en ticketboom";
7 SpannableString ss = new SpannableString(registro);
8
9 ClickableSpan Rticketboom = (widget) -> {
10     Intent a = new Intent ( packageContext: MainActivity.this, registro.class);
11     startActivity(a);
12 }
13
14 ss.setSpan(Rticketboom, start: 14, end: 24, Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
15 registro.setText(ss);
16 registro.setMovementMethod(LinkMovementMethod.getInstance());
17 }
18
19 //Metodo para iniciar sesion
20 public void inicioS (View view){
21     String email = Iemail.getText().toString();
22     String contraseña = Ipassword.getText().toString();
23
24     if(contraseña.length() == 0){
25         Toast.makeText( context: this, text: "Ingresa una contraseña porfavor", Toast.LENGTH_LONG).show();
26     }
27     if(email.length() == 0){
28         Toast.makeText( context: this, text: "Ingresa un correo porfavor", Toast.LENGTH_LONG).show();
29     }
30     if(!email.equals(null) || !contraseña.equals(null)){
31         Toast.makeText( context: this, text: "Verificando correo", Toast.LENGTH_LONG).show();
32     }
33 }
```

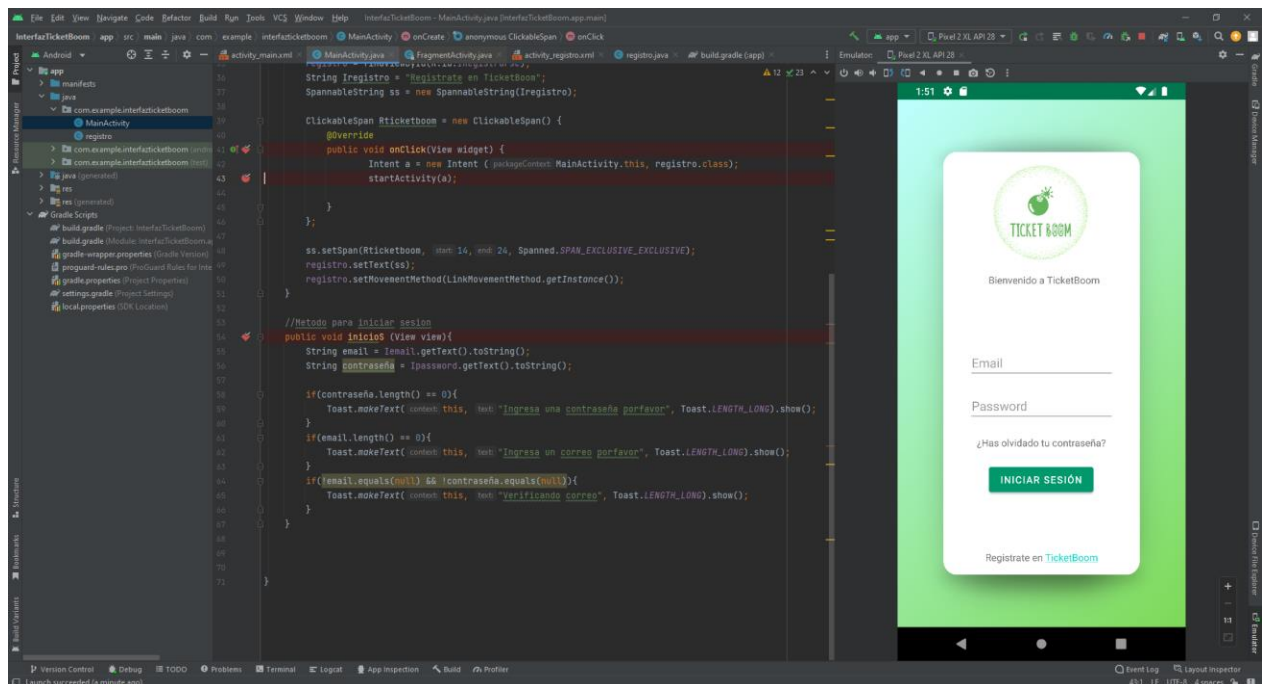
Estos 3 puntos de debug estarían situados en nuestro archivo MainActivity

```
1 }
2
3 if(datosCompletos){
4     FirebaseFirestore db = FirebaseFirestore.getInstance();
5
6     Map<String, Object> User = new HashMap<>();
7     User.put("Nombre", nombre);
8     User.put("Apellido", apellido);
9     User.put("Email", email);
10    User.put("Contraseña", contraseña);
11    User.put("Telefono", Telefono);
12
13    db.collection( collectionPath: "Users").document().set(User).addOnCompleteListener(new OnCompleteListener<Void>() {
14        @Override
15        public void onComplete(@NonNull Task<Void> task) {
16            if (task.isSuccessful()){
17                Toast.makeText( context: registro.this, text: "Registro completado", Toast.LENGTH_LONG).show();
18            }
19        }
20    })
21 }
```

Este último estaría situado en nuestro archivo registro

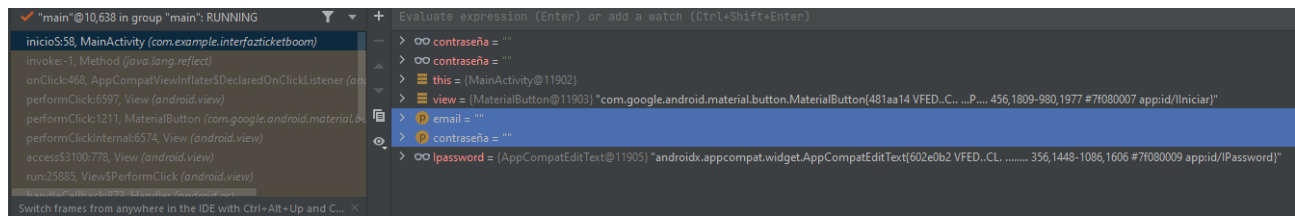


Aquí podemos notar que empieza el proceso de debugging, el cual nos permitirá encontrar errores dentro de nuestra codificación,



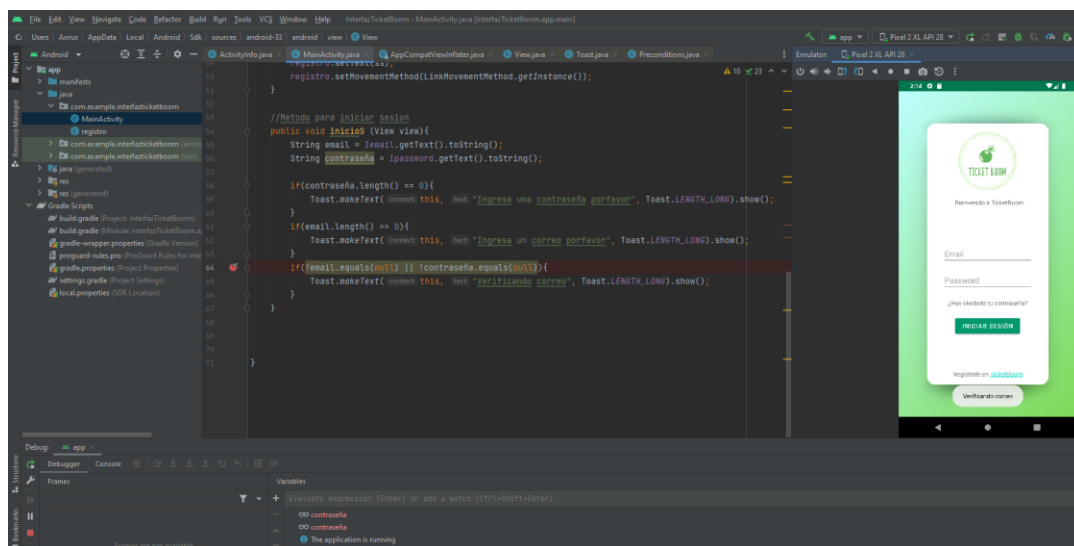
Una vez ya con la interfaz iniciada podemos proceder con el primer punto de nuestro debugg, el cual seria tratar de iniciar sesión sin datos

Tras presionar iniciar sesión sin poner ningún dato, en el debugger podremos encontrar los valores relacionados a los text box que tenemos

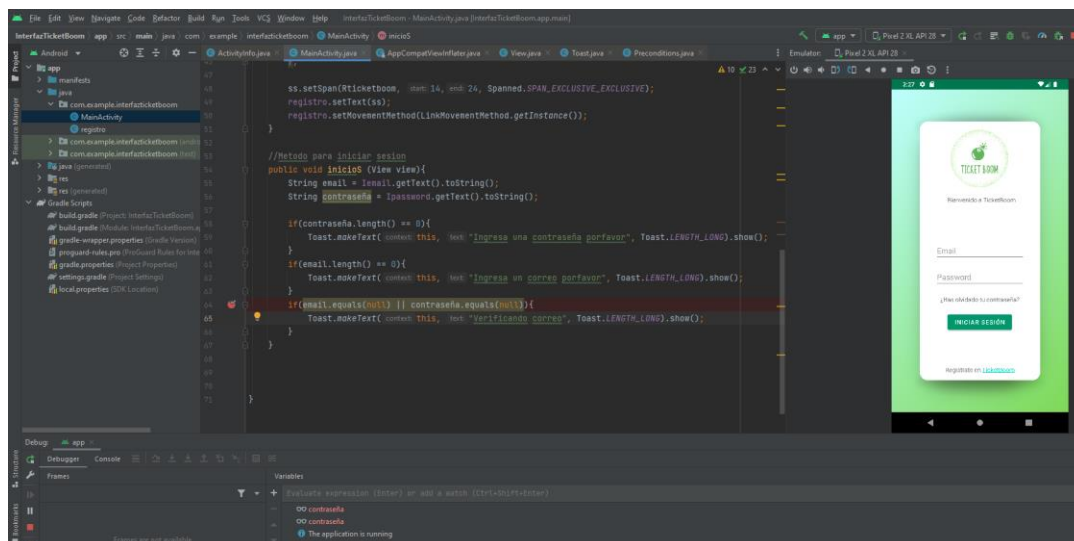


Es aquí por ejemplo que nos sirve el proceso de debbuging, donde encontramos el siguiente error, donde estamos haciendo la comparación de forma equivocada debido al negativo de la evaluación null de nuestra variable

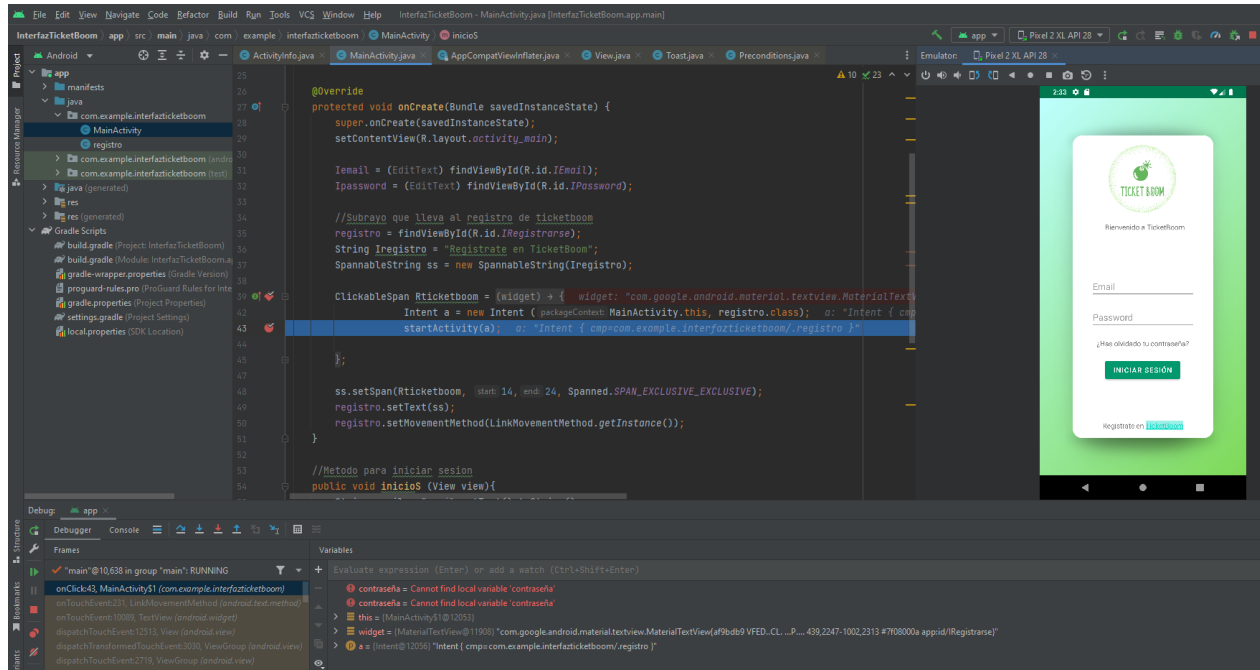
Siendo la incorrecta la línea de esta forma:



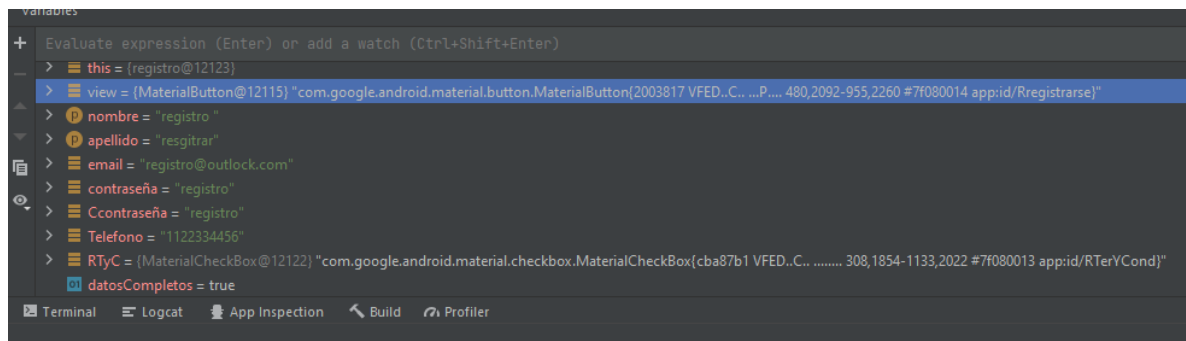
Y la corrección de la siguiente forma:



Al continuar con el proceso de debbuging presionaremos el texto subrayado que nos llevara al siguiente activity como se puede ver aquí:



Tras movernos de activity y rellenar el formulario de registro en el debugger tendremos las variables que hemos llenado, en este caso es de la siguiente forma:

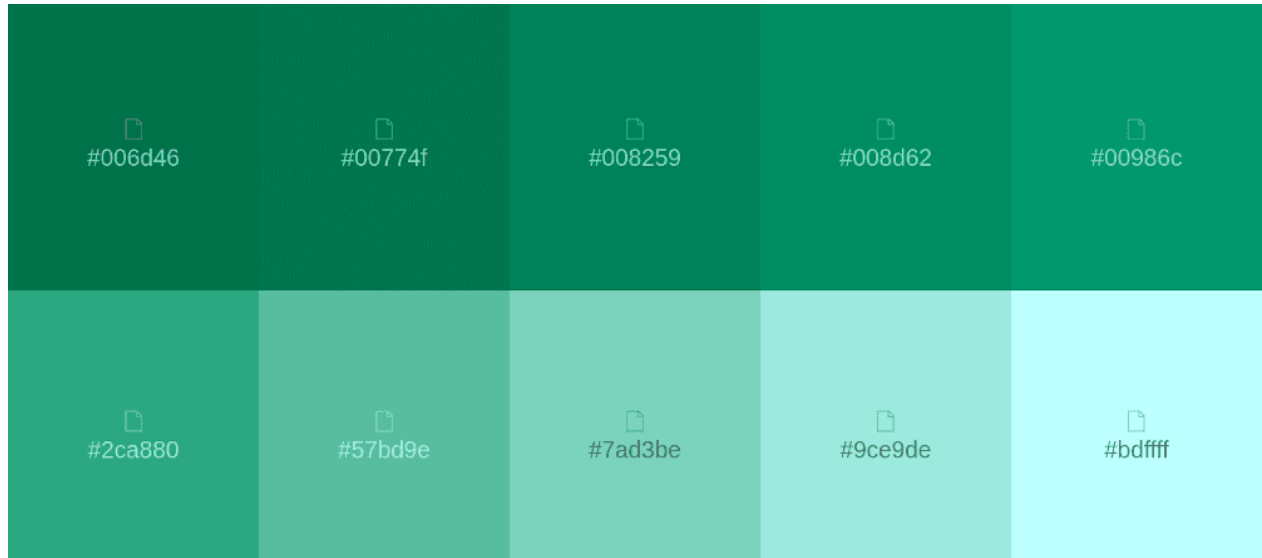


Donde al tener los datos completos, nuestro booleano se quedará en true, permitiéndonos mandar nuestros datos a la base de datos en la nube

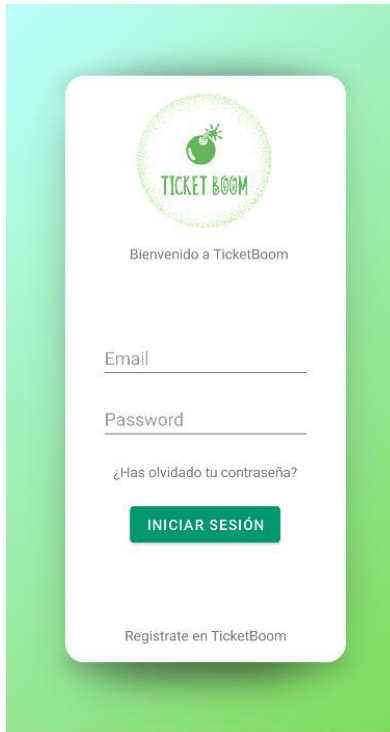


Diseño

En cuanto a nuestro diseño esta será la paleta de colores a utilizar, son unos colores entre verde y azul o más específicamente colores menta



Para la interfaz principal, usaremos una imagen ya editada como fondo de pantalla, la cual ya cuenta con las características de un recuadro blanco donde cuenta con puntas rebordeadas y sombras, además de un degradado de azul celeste a verde, es con esto mismo que los colores a utilizar tanto en logo y botón también serán dentro de la paleta de colores, presentando el logo al principio con un pequeño mensaje de bienvenida para seguir con el formulario de inicio de sesión, a través de un email y una contraseña, también estará presente la posibilidad de recuperar la contraseña junto al botón de iniciar sesión, si el usuario es nuevo, entonces tendrá la opción de registrarse, esto al final del diseño, esta se puso hasta el fondo porque es una opción que se usara muy poco y nos centraremos mas en el inicio de sesión

A login form for Ticket Boom. At the top is a circular logo with a green bomb icon and the text "TICKET BOOM". Below the logo is the text "Bienvenido a TicketBoom". The form contains two input fields: "Email" and "Password". Below the password field is a link that says "¿Has olvidado tu contraseña?". At the bottom of the form is a green button with the text "INICIAR SESIÓN". Below the button is a link that says "Regístrate en TicketBoom". The form is set against a green gradient background.

TICKET BOOM

Bienvenido a TicketBoom

Email

Password

¿Has olvidado tu contraseña?

INICIAR SESIÓN

Regístrate en TicketBoom

En cuanto a la parte de registro, utilizaremos la misma plantilla para el fondo y los mismos colores para el botón, eliminaremos el logo y pondremos un pequeño texto para identificar en el formato en el cual estamos, en este caso indicara “Registro”, en este apartado tendremos que rellenar con, “Nombre”, “Apellido”, “Email”, “Contraseña”, “Confirmar contraseña”, “Telefono”, un Checkbox aceptando los términos y condiciones y para finalizar el botón de “Registrarse”

A registration form titled "Registrarse". It contains several input fields: "Nombre", "Apellido", "Email", "Contraseña", "Confirmar contraseña", and "Telefono". Below these fields is a checkbox with the text "He leído y acepto los Términos y Condiciones". At the bottom of the form is a green button with the text "REGISTRARSE". The form is set against a green gradient background.

Registrarse

Nombre

Apellido

Email

Contraseña

Confirmar contraseña

Telefono

☐ He leído y acepto los Términos y Condiciones

REGISTRARSE