



**UTT**

UNIVERSIDAD TECNOLÓGICA DE TIJUANA

**GOBIERNO DE BAJA CALIFORNIA**

**TEMA:**

Autenticación con firebase

**PRESENTADO POR:**

Daniel Chavez Madrigal

**GRUPO:**

9-B

**MATERIA:**

Desarrollo para Dispositivos Inteligentes

**PROFESOR:**

Ray Brunett Parra Galaviz

Tijuana, Baja California, 31 de octubre del 2024

## Explicación del Código de Autenticación con Firebase

Este código se compone de dos archivos principales:

1. **LoginMailActivity**: Maneja la autenticación mediante Firebase utilizando el correo y la contraseña del usuario.
2. **MainActivity**: Configura la navegación entre fragmentos después de que el usuario ha iniciado sesión correctamente.

### 1. LoginMailActivity: Autenticación de Usuario con Firebase

Este archivo contiene el flujo de inicio de sesión de un usuario en Firebase usando un correo y contraseña.

#### *Código de la clase LoginMailActivity*

kotlin

Copy code

```
class LoginMailActivity : AppCompatActivity() {
    private lateinit var auth: FirebaseAuth
    private lateinit var etemail: EditText
    private lateinit var etpassword: EditText

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_login_mail)

        // Inicializar FirebaseAuth
        auth = FirebaseAuth.getInstance()

        // Referencias a los campos de email y contraseña
        etemail = findViewById(R.id.et_email)
        etpassword = findViewById(R.id.et_password)
    }

    fun login(view: View){
        val email = etemail.text.toString()
        val password = etpassword.text.toString()

        auth.signInWithEmailAndPassword(email,
password).addOnCompleteListener { task ->
            if(task.isSuccessful){
                // Iniciar MainActivity si el inicio de sesión es exitoso
                val intent = Intent(this, MainActivity::class.java)
                startActivity(intent)
                finish()
            }
        }.addOnFailureListener { exception ->
```

```

        // Mostrar mensaje de error en caso de fallo en el inicio de
sesión
        Toast.makeText(applicationContext,
exception.localizedMessage, Toast.LENGTH_LONG).show()
    }
}
}

```

### *Desglose de la funcionalidad*

- **Inicialización de Firebase:** `auth = FirebaseAuth.getInstance()` crea una instancia de `FirebaseAuth` para manejar la autenticación.
- **Captura de Email y Contraseña:** Los campos de texto `email` y `password` reciben la entrada del usuario para autenticarse.
- **Método login:**
  - Usa `auth.signInWithEmailAndPassword(email, password)` para iniciar sesión en Firebase.
  - Si el inicio de sesión es exitoso, navega a `MainActivity` mediante un `Intent`.
  - En caso de fallo, muestra un mensaje `Toast` con la descripción del error (`exception.localizedMessage`), proporcionando retroalimentación al usuario.

## 2. MainActivity: Navegación entre Fragmentos

La clase `MainActivity` configura la interfaz de navegación que permite al usuario alternar entre las opciones "Profile", "Users", y "Chats" una vez que ha iniciado sesión.

### *Código de la clase MainActivity*

```

kotlin
Copy code
class MainActivity : AppCompatActivity() {
    private lateinit var binding: ActivityMainBinding
    private lateinit var firebaseAuth: FirebaseAuth

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        binding = ActivityMainBinding.inflate(layoutInflater)

        enableEdgeToEdge()
        setContentView(binding.root)

        firebaseAuth = FirebaseAuth.getInstance()

        viewProfile()
        binding.bottomNV.setOnItemClickListener { item ->

```

```

        when(item.itemId){
            R.id.item_profile -> {
                viewProfile()
                true
            }
            R.id.item_users -> {
                viewUsers()
                true
            }
            R.id.item_chats -> {
                viewChats()
                true
            }
            else -> {false}
        }
    }
}

private fun goToLogin(){
    startActivity(Intent(applicationContext,
optionsLoginActivity::class.java))
}

private fun viewProfile(){
    binding.tvTitle.text = "Profile"
    val fragment = FragmentProfile()
    val fragmentTransaction =
supportFragmentManager.beginTransaction()
    fragmentTransaction.replace(binding.fragmentFL.id, fragment)
    fragmentTransaction.commit()
}






private fun viewUsers(){
    binding.tvTitle.text = "Users"
    val fragment = FragmentUsers()
    val fragmentTransaction =
supportFragmentManager.beginTransaction()
    fragmentTransaction.replace(binding.fragmentFL.id, fragment)
    fragmentTransaction.commit()
}

private fun viewChats(){
    binding.tvTitle.text = "Chats"
    val fragment = FragmentChats()
    val fragmentTransaction =
supportFragmentManager.beginTransaction()
    fragmentTransaction.replace(binding.fragmentFL.id, fragment)
    fragmentTransaction.commit()
}
}

```

## Desglose de la funcionalidad

- **Autenticación:** `firebaseAuth = FirebaseAuth.getInstance()` se utiliza para verificar si el usuario está autenticado. El bloque comentado `//if (firebaseAuth.currentUser == null){ goToLogin() }` puede descomentarse para que el usuario vuelva al inicio de sesión si no está autenticado.
- **Navegación entre Fragmentos:**
  - La barra de navegación `bottomNV` permite al usuario alternar entre las opciones "Profile", "Users", y "Chats".
  - La función `setOnItemSelectedListener` escucha las selecciones de menú y ejecuta el método correspondiente a cada opción seleccionada.
- **Métodos `viewProfile`, `viewUsers`, `viewChats`:**
  - Cada método configura el título de la pantalla (`tvTitle`) y carga el fragmento adecuado (`FragmentProfile`, `FragmentUsers`, `FragmentChats`) usando el `supportFragmentManager`.
  - Se realiza una transición de fragmento (`fragmentTransaction.replace`) que muestra el fragmento correspondiente en el `FrameLayout` (`fragmentFL`), configurado en el diseño XML de la actividad principal.

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			
amaury@gmail.com		31 oct 2024	31 oct 2024	g80HkCnSjPhdeOmPjxeMCtt5...			
gaelbr@hotmail.com		31 oct 2024	31 oct 2024	Yv1H9j7yJSeWV1b9Q0Jd1nYL...			
danixd@mail.com		30 oct 2024	30 oct 2024	XL1FtuR6NfeaHhvLEnmckyof...			
hola2@gmail.com		30 oct 2024	30 oct 2024	AspjHkvDn3dnr2mGrXNYSIGA...			
hola@gmail.com		30 oct 2024	30 oct 2024	Jl460STAxgXZscvy6BBID6bs...			

## LOGIN

- Email

hola@gmail.com

- Password

.....



INGRESAR

[Regístrate aquí](#)

