Software para Dispositivos Móviles
Grado en Ingeniería Informática del Software
Escuela de Ingeniería Informática – Universidad de Oviedo

Servicios en

Dr. Juan Ramón Pérez Pérez

Departamento de Informática

jrpp@uniovi.es

Consola de Firebase

- https://firebase.google.com/
- Utilizar una cuenta de Google para hacer login
- Ir a la consola de Firebase
- Crear proyecto →

Configuración de un proyecto Firebase

Crear nuevo proyecto

- En la consola aparecen todos los proyectos Firebase de esa cuenta
- Normalmente un proyecto concentra todos los servicios de distintos tipos para una app
- Para un proyecto hay que introducir: nombre, id, habilitar Google Analytics, cuenta de Analytics

Vincular Firebase con una aplicación

- Registrar App
  - Ir al proyecto de la app en Android Studio
  - Introducir el nombre del paquete
  - Descargar `google-services.json`
  - Copiarlo a la carpeta app de nuestro proyecto
  - Añadir dependencias en el Gradle →

Dependencias del gradle (I)
libs.version.toml

```
[versions]
...
firebaseBom = "33.4.0"
services = "4.4.2“
crashlytics=  "3.0.2"
```

```
[libraries]
...
firebase-bom = { module = "com.google.firebase:firebase-bom", version.ref =
"firebaseBom" }
firebase-crashlytics = {module = "com.google.firebase:firebase-crashlytics" }
```

```
[plugins]
...
googleServices = { id = "com.google.gms.google-services", version.ref= "services"}
crashlytics = { id = "com.google.firebase.crashlytics", version.ref= "crashlytics"}
```

Dependencias del gradle (II)
build.gradle.kts (:app)

```
dependencies {
    // Import the Firebase BoM
    implementation(platform(libs.firebase.bom))
    implementation(libs.firebase.analytics)
    implementation(libs.firebase.crashlytics)
...
```

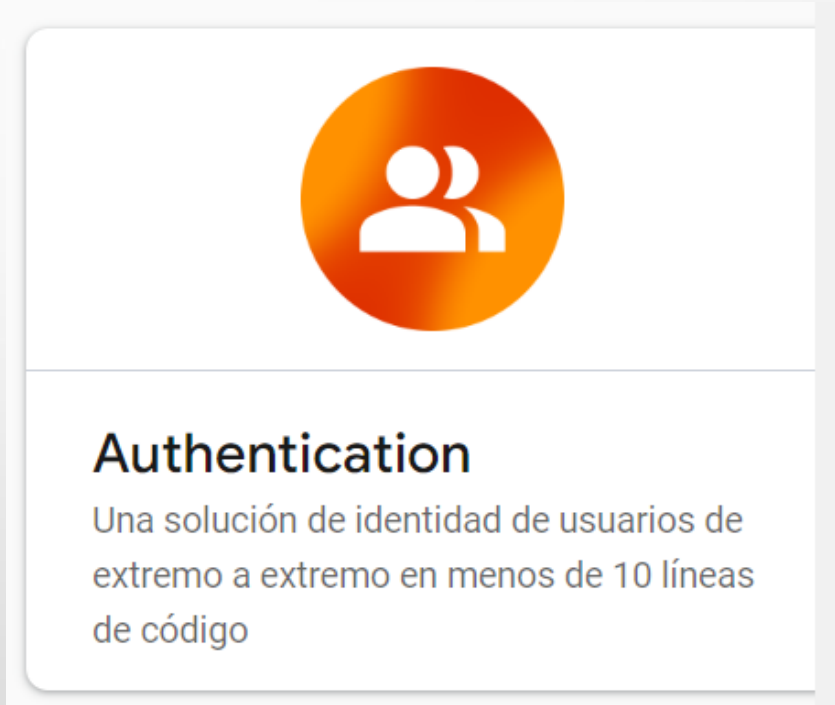
```
plugins {
...
    alias(libs.plugins.googleServices)
    alias(libs.plugins.crashlytics)
}
```

Recursos gráficos

- Muchas veces herramientas y guias nos proporcionan recursos, no reinventar la rueda
- https://firebase.google.com/brand-guidelines

Autenticación

- Gestión de forma sencilla de usuarios
  - Login / Logout
  - Registro

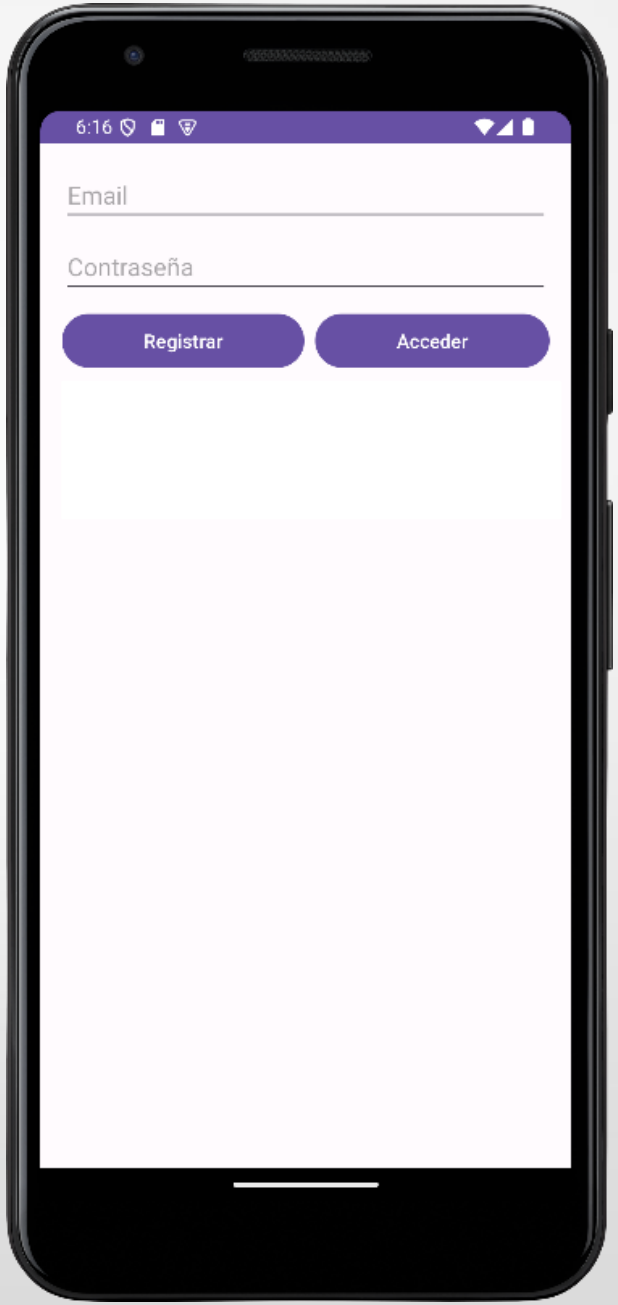


Inicialización del servicio

- En la consola de Firebase
- Servicio de autenticación
- Distintas formas y proveedores de autenticación
- Autenticación por correo electrónico y contraseña

Prueba de concepto

- Diseño de la interfaz
  - activity_auth
  - EditText email / contraseña
  - Botón Registrar
  - Botón Acceder
- Funcionalidad
  - Botón de registro → Registra usuarios
  - Botón de acceder → Autentica usuarios



Configuración de librerías en el gradle

- libs.version.toml
- (el boom establece versión global y aquí no hay que ponerla)

```
firebase-auth = {module = " com.google.firebase:firebase-auth"}
```

- build.gradle [Module :app]

```
implementation(libs.firebase.auth)
```

Qué hay que implementar

- Necesitamos referencias para los EditText y los botones
- Registramos el evento click para cada botón
  - setOnClickListener { }
- Tenemos un singleton para acceder a los servicios de autenticación
  - **FirebaseAuth**.getInstance()

Registro (SignUp) de un usuario

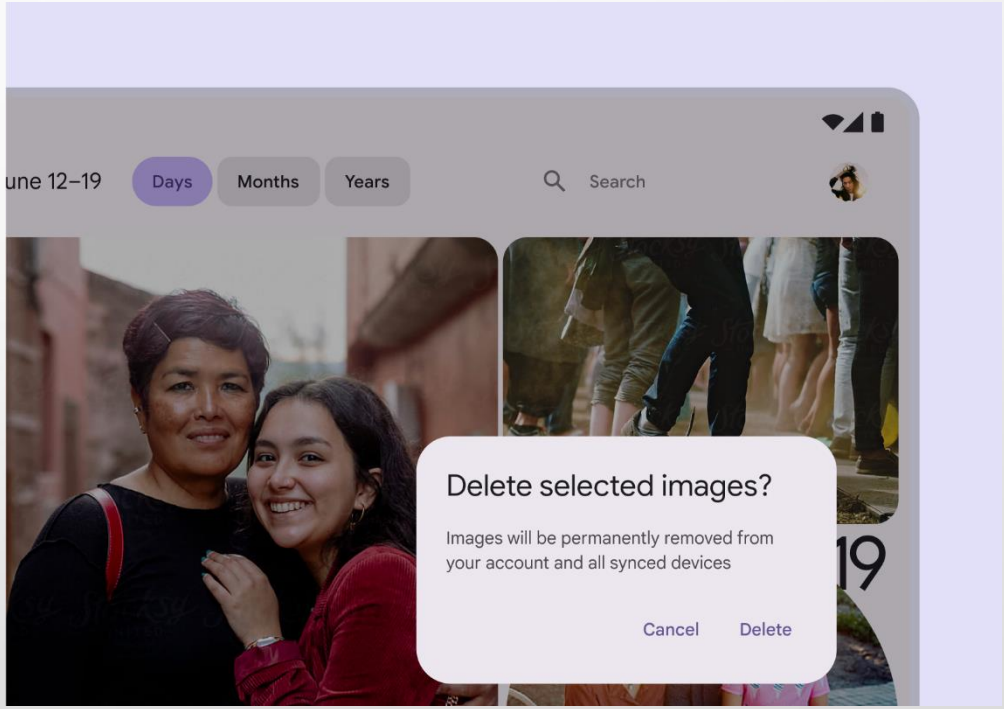
- **FirebaseAuth**.getInstance().createUserWithEmailAndPassword()
- Ojo esto hace una petición a un servicio remoto → establecer una función de callback para esperar resultados
  - .addOnCompleteListener { … }
- Seguimos el patrón típico:
  - ver si todo fue bien
  - Si es así, podemos acceder al resultado

Login (SignIn) de un usuario

- **FirebaseAuth**.getInstance(). signInWithEmailAndPassword()
- Ojo esto hace una petición a un servicio remoto → establecer una función de callback para esperar resultados
  - .addOnCompleteListener { … }
- Seguimos el patrón típico:
  - ver si todo fue bien
  - Si es así, podemos acceder al resultado

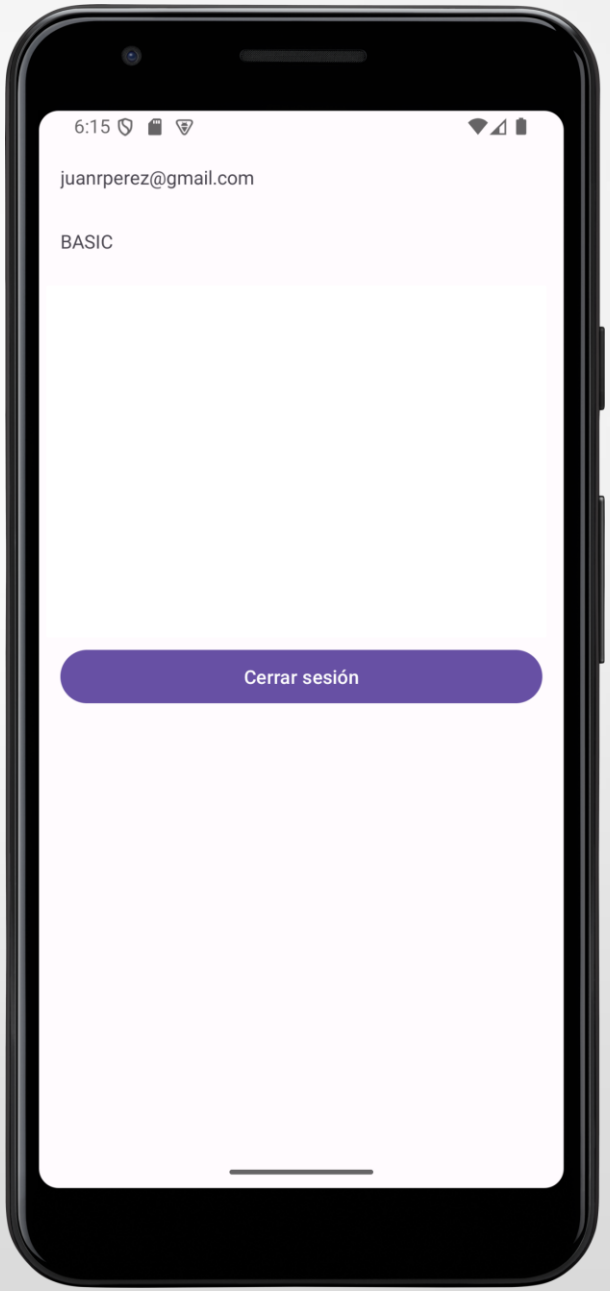
Diálogos en Android

- Componentes que obligan a interactuar al usuario ante una opción importante
- Normalmente no ocupan toda la pantalla aunque podrían
- Normalmente se crean dinámicamente
- https://m3.material.io/components/dialogs/overview



HomeActivity

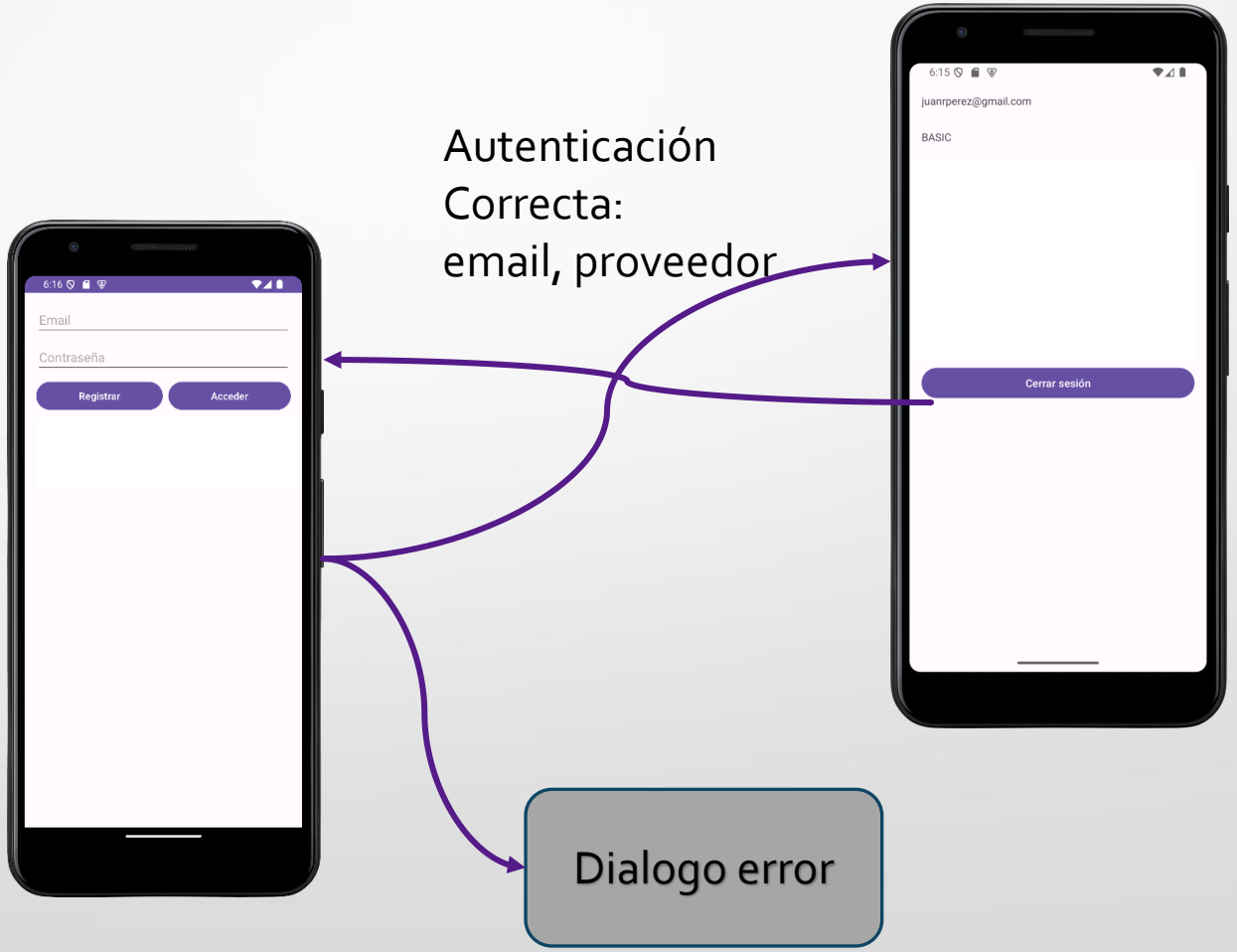
- Diseño de la interfaz
  - activity_home
  - TextView email
  - TextView proveedor
  - Botón Cerrar sesión
- Funcionalidad
  - Muestra email y proveedor
  - Botón de cerrar sesión → logout usuario



Ejercicio de recordatorio

- En función del resultado del registro / login
- Si hay error
  - Abrir dialogo error
- Si va bien
  - Abrir Activity Home
  - Pasar como parámetros: email, proveedor
- Volver desde la ActivityHome

  FirebaseAuth.getInstance().signOut()
  onBackPressed()



Activity de autenticación sólo si es necesaria

- Primero de usabilidad: “No obligar al usuario a repetir cosas que ya hizo”
- Si el usuario ya está logeado → app debería aparecer en home directamente
- ¿Cómo se comprueba si el usuario ya está logeado?
  - **val** currentUser= FirebaseAuth.getInstance().currentUser

Cloud Firestore

- Base de datos NoSQL (Como MongoDB):
  - Colecciones y documentos
- Crear base de datos
- Ubicación geográfica de la BD
- Utilizar modo de pruebas en principio
- Funciones de la consola:
  - Datos, Reglas, Índices, Uso

Configuración de librerías

- libs.version.toml

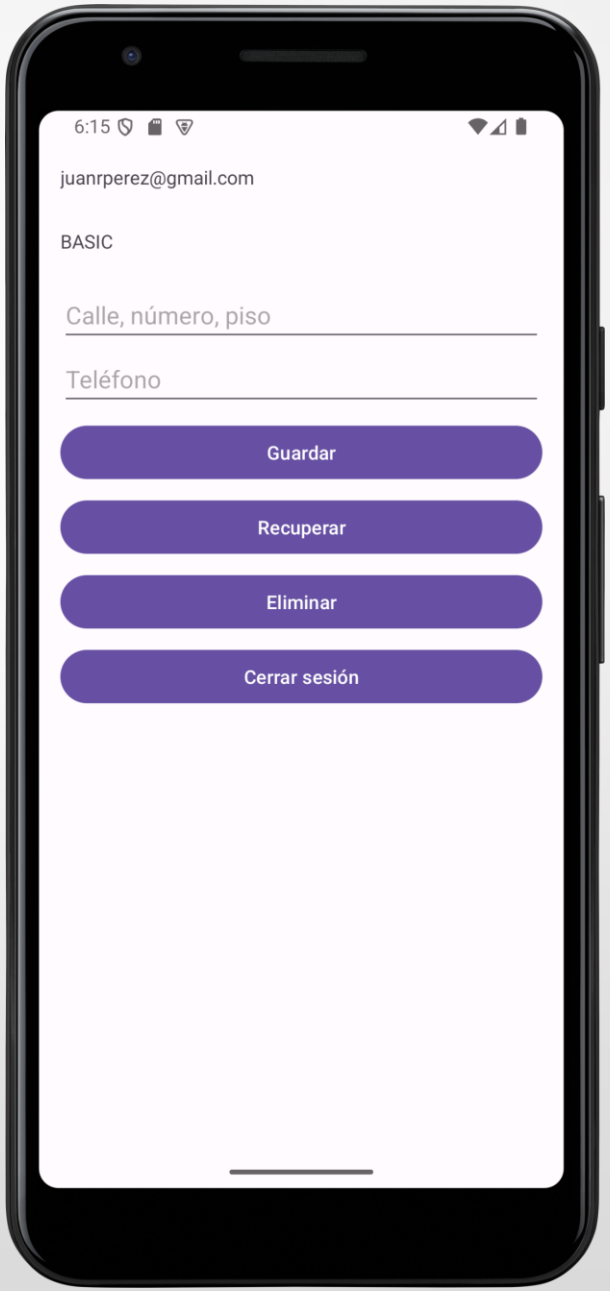
```
firebase-firestore = { module= "com.google.firebase:firebase-firestore"}
```

- build.gradle [Module :app]

```
implementation(libs.firebase.firestore)
```

Prueba de concepto

- Modelo de datos:
  - Colección de usuarios
  - Documentos con los siguientes campos: email (clave), provider, dirección y teléfono
- Interfaz ActivityHome, añadir:
  - EditText: address y pone
  - Botones: Guardar, Recuperar, Eliminar



Métodos para trabajar con el servicio Firestore (I)

- Recuperar del singleton la instancia de la BD
  - private val db= FirebaseFirestore.getInstance()
- Crear un documento nuevo en la BD
  - db.collection("users").document(email).set()
- Pasar los campos del documento como un HashMap
  - hashMapOf("provider" to provider,
    "address" to addressEditText.text.toString(),
    "phone" to phoneEditText.text.toString())

Métodos para trabajar con el servicio Firestore (II)

- Recuperar un documento de la BD
  - db.collection("users").document(email).get().addOnSuccessListener { addressEditText.setText(it.get("address") as String?) phoneEditText.setText(it.get("phone") as String?)
- Borrar un documento de la BD
  - db.collection("users").document(email).delete()