



Desarrollo de Aplicaciones Multiplataforma

DOCENTE: Daniel López Lozano





Firebase

Tema 5.

Introducción a Firebase

Índice de contenidos

- ❑ Introducción a FireBase.

- ❑ Si queremos usar firebase hay que obtener una referencia a la bases de datos.
- ❑ Después usando una notación parecida a XPath localizar el elemento sobre el que queremos actuar.
- ❑ Con esto podemos hacer operaciones de creación y actualización de datos.
- ❑ Se recomienda tanto en operaciones de escritura como de lectura usar POJO's ya que Firebase es capaz de trabajar con mapeo de objetos y resulta mucho más cómodo trabajar así.

Acceso a datos con Firebase

```
private lateinit var referencia: DatabaseReference

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

    referencia = FirebaseDatabase.getInstance().getReference()
}
```

Mapeo de los objetos en Firebase/Kotlin

```
data class Persona(var nombre: String? = null,
                  var apellido1: String? = null,
                  var apellido2: String? = null,
                  var email: String? = null,
                  var telefono: String? = null,
                  var edad: Int? = 0){
}
```

Mapeo de objetos en Firebase

```
data class Persona(var nombre: String? = null,  
                  var apellido1:String?=null,  
                  var apellido2:String?=null,  
                  var email:String?=null,  
                  var telefono:String?=null,  
                  var edad:Int?=0){  
}
```

```
"-LtZBOK68xIgdHbQS3M7" : {  
  "apellido1" : "Morty",  
  "apellido2" : "Martinez",  
  "edad" : 1,  
  "email" : "nomerobes.gmail.com",  
  "nombre" : "Chori",  
  "telefono" : "123456789"  
},  
"p1" : {  
  "apellido1" : "Martin",  
  "apellido2" : "Lopez",  
  "edad" : 99,  
  "email" : "antonio@email.com",  
  "nombre" : "Pepico",  
  "telefono" : "1177328"  
}
```

Crear con Firebase

```
//CREAR UN NUEVO DATO EN LA BASE DE DATOS
//GENERAR CLAVE UNICA PARA QUE NO SE PISE CON OTRA
var identificador = referencia.child( pathString: "personas").push().key!!

var nueva_persona = Persona( nombre: "Daniel",
                               apellido1: "Lopez",
                               apellido2: "Lozano",
                               email: "daniel@dani.com",
                               telefono: "123455678",
                               edad: 39)

referencia.child( pathString: "personas").child(identificador).setValue(nueva_persona)
```


Modificar con Firebase

```
//MODIFICAR VALORES INDIVIDUALES
```

```
referencia.child( pathString: "personas").child(identificador).child( pathString: "nombre").setValue("Perico")  
referencia.child( pathString: "personas").child(identificador).child( pathString: "apellido1").setValue("El de los ")  
referencia.child( pathString: "personas").child(identificador).child( pathString: "apellido2").setValue("Palotes")
```

```
//CAMBIAR PARCIALMENTE DE MANERA MAS RAPIDA
```

```
//EL RESTO SE VA A GUARDAR IGUAL
```

```
nueva_persona.nombre="Perico"
```

```
nueva_persona.apellido1="El de los"
```

```
nueva_persona.apellido2="Palotes"
```

```
referencia.child( pathString: "personas").child(identificador).setValue(nueva_persona)
```

```
//OTRO OBJETO NUEVO PARA SUSTITUIRLO POR OTRO
```

```
var otra_persona=Persona( nombre: "Danielin",
```

```
    apellido1: "Lopes",
```

```
    apellido2: "Losano",
```

```
    email: "danielsan@daniel.com",
```

```
    telefono: "4324234",
```

```
    edad: 40)
```

```
referencia.child( pathString: "personas").child(identificador).setValue(otra_persona)
```


Borrar con Firebase

```
//BORRAR UN PARCIALMENTE UN OBJETO : SOLO EL NOMBRE  
referencia.child( pathString: "personas").child(identificador).child( pathString: "nombre").removeValue()  
//BORRAR UN TOTALMENTE UN OBJETO  
referencia.child( pathString: "personas").child(identificador).removeValue()
```

- ❑ Para obtener datos la forma es algo distinta.
- ❑ Al cargar la activity hay que programar una serie de eventos que indican según lo sucedido la disponibilidad de datos en distintas circunstancias.
- ❑ Primero debemos seleccionar los datos y/o colecciones que queremos "escuchar" y después programar como queremos que se muestren en un nuestra activity.

Obtener datos todos los datos

```
referencia.child( pathString: "personas")
    .addListenerForSingleValueEvent(object : ValueEventListener {
        override fun onDataChange(snapshot: DataSnapshot) {
            var resultado: String = ""
            snapshot.children.forEach { hijo: DataSnapshot? ->
                val pojo_persona = hijo?.getValue(Persona::class.java)
                resultado += pojo_persona?.nombre + "\n" +
                    pojo_persona?.edad + "\n" +
                    pojo_persona?.email + "\n\n"
            }
            datos.setText("En la agenda hay\n\n" + resultado)
        }

        override fun onCancelled(error: DatabaseError) {
            println(error.message)
        }
    })
```

Si queremos la aplicación se actualice tiempo real poner addValueEvent

```
referencia.child( pathString: "personas")  
  .addValueEventListener(object : ValueEventListener {  
    override fun onDataChange(snapshot: DataSnapshot) {  
      var resultado: String = ""  
      snapshot.children.forEach { hijo: DataSnapshot? ->  
        val pojo_persona = hijo?.getValue(Persona::class.java)  
        resultado += pojo_persona?.nombre + "\n" +  
          pojo_persona?.edad + "\n" +  
          pojo_persona?.email + "\n\n"  
      }  
      datos.setText("En la agenda hay\n\n" + resultado)  
    }  
  })
```

Filtrar datos desde Firebase (consultas)

Solo se puede aplicar un filtro

```
//SELECCIONAR USUARIO POR ID
referencia.child("usuarios").child("id1")
//SELECCIONAR SOLO 10 USUARIOS PRIMEROS
referencia.child("usuarios").limitToFirst(10)

//SELECCIONAR SOLO 10 USUARIOS ULTIMOS
referencia.child("usuarios").limitToLast(10)

//SELECCIONAR USUARIO POR LOGIN
referencia.child("usuarios").orderByChild("login").equalTo("Jesusico")

//SELECCIONAR USUARIOS CON edad mayor o igual a 30
referencia.child("usuarios").orderByChild("edad").startAt(30)

//SELECCIONAR USUARIOS CON edad menor o igual a 30
referencia.child("usuarios").orderByChild("edad").endAt(30)

//SELECCIONAR USUARIOS CON edad menor a 30
referencia.child("usuarios").orderByChild("edad").endAt(29)

//SELECCIONAR USUARIOS CON edad entre 18 y 30
referencia.child("usuarios").orderByChild("edad").startAt(18).endAt(30)
```


Obtener solo el primer objeto (o el último)

```
referencia.child( pathString: "personas").limitToFirst( limit: 1)  
.addValueEventListener(object : ValueEventListener {  
    override fun onDataChange(snapshot: DataSnapshot) {  
        var resultado: String = ""  
        if(snapshot.hasChildren()){  
            val perso=snapshot.children.iterator().next().getValue(Persona::class.java)  
            resultado+=perso?.nombre + "\n" +  
                perso?.edad + "\n" +  
                perso?.email + "\n\n"  
        }  
    }  
})
```

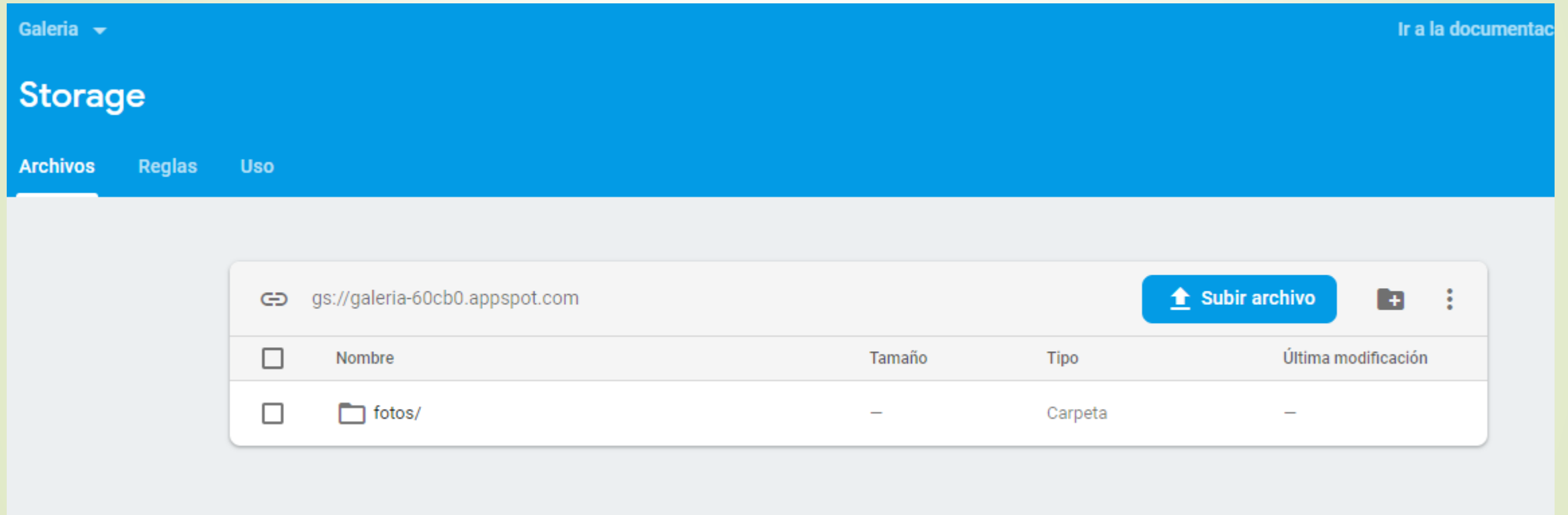
Obtener datos de un objeto único

```
referencia.child( pathString: "personas")
    .orderByChild( path: "nombre")
    .equalTo( value: "Elfo")
    .addValueEventListener(object : ValueEventListener {
        override fun onDataChange(snapshot: DataSnapshot) {
            var resultado: String = ""

            if(snapshot.hasChildren()){
                val pojo_persona=snapshot.children
                    .iterator()
                    .next().getValue(Persona::class.java)
                resultado+=pojo_persona?.nombre + "\n" +
                    pojo_persona?.edad + "\n" +
                    pojo_persona?.email + "\n\n"
            }
        }
    })
```


- ❑ En firebase existe un servicio que permite guardar y recuperar archivos en un servidor accesible desde internet con URL y de manera segura.
- ❑ Podemos subir nosotros mismos los ficheros usando el interfaz de firebase.
- ❑ Existe un sistema de caminos al igual que cuando usamos el servicio de bases de datos en tiempo real, donde cada paso del camino es una carpeta en el sistema.

Interfaz Storage










Interfaz Storage Firebase

gs://galeria-60cb0.appspot.com

> fotos

Subir archivo

<input type="checkbox"/>	Nombre	Tamaño	Tipo	Última modificación
<input type="checkbox"/>	 -LUqCmb9NQQWYoDEJDxm	87,17 KB	image/jpeg	28 dic. 2018
<input type="checkbox"/>	 -LUqCXcP4q4zYjeorvMU	71,41 KB	image/jpeg	28 dic. 2018
<input type="checkbox"/>	 -LUqD6KLAR6BPCXJRuTC	7,2 KB	image/jpeg	28 dic. 2018
<input type="checkbox"/>	 -LUqDYSw1YmTo81cna2q	91,42 KB	image/jpeg	28 dic. 2018
<input type="checkbox"/>	 -LVIXwlRX2kQ6w9Scikv	31,81 KB	image/jpeg	8 ene. 2019
<input type="checkbox"/>	 -LVt7Uyq6plozhskfXtF	7,31 KB	image/jpeg	10 ene. 2019
<input type="checkbox"/>	 -LWQhxbXVF-xy__O0v8f	7,54 KB	image/jpeg	17 ene. 2019

- ❑ Primeramente debemos configurar el acceso al storage de nuestro proyecto firebase.
- ❑ En un principio va estar abierto porque si no, es necesarios autenticar la aplicación en de firebase.
- ❑ Dicho proceso de autenticación lo veremos más adelante y servirá para asegurar tanto el storage como el servicio de bases de datos en tiempo real.
- ❑ Después en nuestro proyecto Android hay que conectar con firebase y añadir las librerías de storage y autenticación necesarias.
- ❑ También necesitaremos la librería Glide que permite renderizar una URL de internet usando un ImageView

Configuración de acceso a storage

```
1 service firebase.storage {  
2   match /b/{bucket}/o {  
3     match /{allPaths=**} {  
4       allow read, write;  
5     }  
6   }  
7 }  
8
```

Obtener la imagen de la galería

Para ello necesitamos crear un objeto `registerForActivityResult` que es un `activity` que devuelve un resultado en este caso esta configurada para acceder a la galería del dispositivo

```
private val obtener_url= registerForActivityResult(ActivityResultContracts.GetContent()){
    uri:Uri?->
    when (uri){
        null->Toast.makeText(applicationContext, text: "No has seleccionado una imagen",Toast.LENGTH_SHORT).show()
        else->{
            url_escudo=uri
            image_escudo.setImageURI(url_escudo)
            Toast.makeText(applicationContext, text: "Has seleccionado una imagen",Toast.LENGTH_SHORT).show()
        }
    }
}
```

Hay que invocar el objeto para recoger el

```
image_escudo.setOnClickListener(View.OnClickListener { it: View!
    obtener_url.launch( input: "image/*")
}))
```


Guardar en Storage

Con lo anterior en `url_escudo` tenemos una Uri del archivo seleccionado para enviarlo al servicio Storage dándole un nombre único, para eso usaremos una key generado por Firebase ya que el objetivo es asociarlo a un objeto que tengamos almacenado

```
sto_ref.child( pathString: "liga")  
        .child( pathString: "escudos")  
        .child(id)  
        .putFile(imagen)
```

Después para poder acceder a dicho fichero desde la app hay que obtener su url

```
sto_ref.child( pathString: "liga")  
        .child( pathString: "escudos")  
        .child(id)  
        .downloadUrl
```

Para hacer lo anterior necesitamos una referencia al storage

```
private lateinit var sto_ref:StorageReference  
  
override fun onCreate(savedInstanceState: Bundle?) {  
    sto_ref=FirebaseStorage.getInstance().getReference()
```