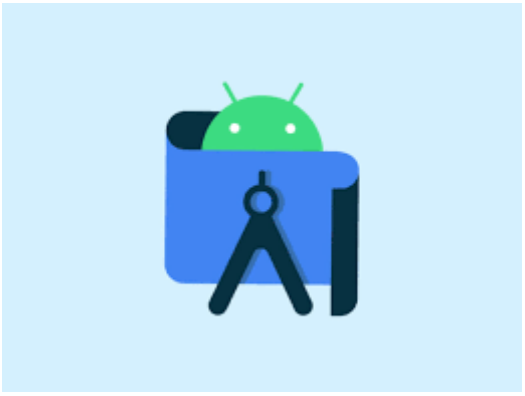
PROYECTO: ARQUITECTURA MVVM.

Juan Carlos
Ortega Lepe
2º DAM.



Introducción

Debido a que no he tenido el suficiente tiempo para realizar la práctica completa, a continuación se muestran los cambios que he realizado en mi aplicación de Android Studio con los apartados de esta práctica.

1. Uso de DataStore.

Para usar DataStore en nuestra aplicación, deberemos implementar una serie de opciones de nuestro main activity.

Primero, crearemos en el xml, un EditText, un Buttom, y un Checkbox, le ponemos un id, y le ponemos los atributos que queramos:

Creamos un editText para poder introducir texto:

```
<EditText
    android:id="@+id/etName"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:hint="Introduce tu  nombre"
    android:maxLines="1"
    android:singleLine="true"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintHorizontal_bias="0.966"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.622" />
```

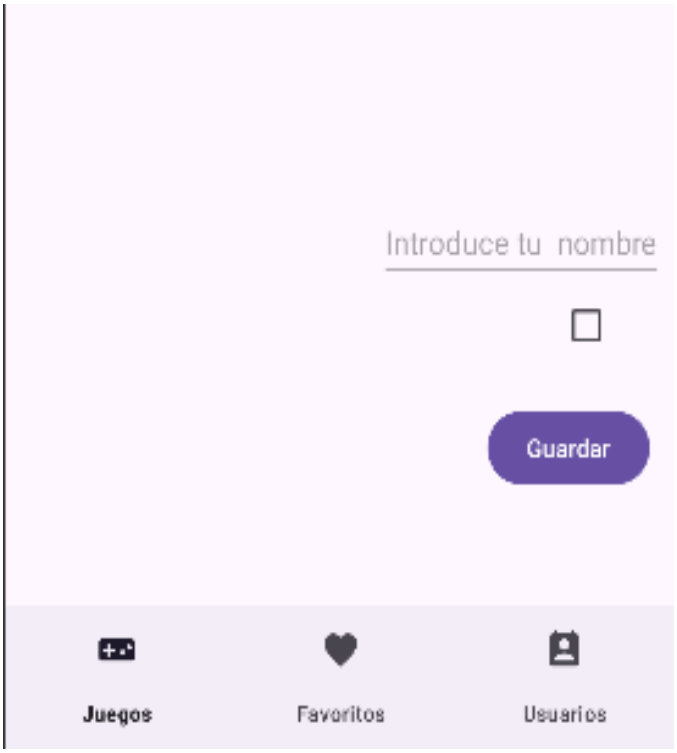
Creamos el CheckBox para marcar la casilla:

```
<CheckBox
    android:id="@+id/cbVIP"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="44dp"
    app:layout_constraintHorizontal_bias="0.914"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="@id/etName" />
```

Creamos un buttom para que se nos guarde y nos lleve a la siguiente actividad:

```
<Button
    android:id="@+id/btnSave"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="144dp"
    android:text="Guardar"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintHorizontal_bias="0.948"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent" />
```

Así se vería en vista de fragmento:



Una vez hecho esto, en el código del MainActivity creamos las siguientes variables:

```
//Datastore

val btnSave = findViewById<Button>(R.id.btnSave)
val etName = findViewById<EditText>(R.id.etName)
val cbVIP = findViewById<CheckBox>(R.id.cbVIP)
```

Una vez hecho esto, deberemos crear la siguiente función:

```
    new *
    private suspend fun saveValues(name: String, checked: Boolean){
        dataStore.edit { preferences ->
            preferences[stringPreferencesKey( name: "name")] = name
            preferences[booleanPreferencesKey( name: "vip")] = checked
        }
    }
}
```

Esta función nos dará errores, debido a que deberemos de crear una nueva actividad en mi proyecto.

En mi caso, la llamaré DetailActivity, y dentro de esta, ponemos el siguiente código:

```
activity_main.xml | MainActivity.kt | activity_detail.xml | DetailActivity.kt | UserProfile.kt

package com.marisma.juegos.adapter

import ...

new *
class DetailActivity : AppCompatActivity() {
    new *
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_detail)
        val backgroundView = findViewById<View>(R.id.viewBackground)
        val tvName = findViewById<TextView>(R.id.tvName)

        lifecycleScope.launch(Dispatchers.IO)  { this: CoroutineScope
        getUserProfile().collect{ it: UserProfile
            withContext(Dispatchers.Main){ this: CoroutineScope
                tvName.text = it.name
                if (it.vip){
                    backgroundView.setBackgroundResource(R.color.purple)
                }
            }
        }
        }

    }
```


```
36      }
        new *
37      private fun getUserProfile() = dataStore.data.map { preferences ->
38          UserProfile(
39              name = preferences[stringPreferencesKey( name: "name")].orEmpty(),
40               vip = preferences[booleanPreferencesKey( name: "vip")] ?: false
41          )
42
43      }
44  }
```

Una vez puesto el código, nos dará una serie de errores debido a que deberemos de crear una nueva clase en Android Studio, la llamaremos UserProfile. Una vez creada, le pondremos en la class los siguientes valores:

activity_main.xml | MainActivity.kt | activity_detail.xml | DetailActivity.kt | UserProfile.kt

```
1   package com.marisma.juegos.adapter
2
    new *
3   class UserProfile (val name:String, val vip:Boolean)
4
5
```

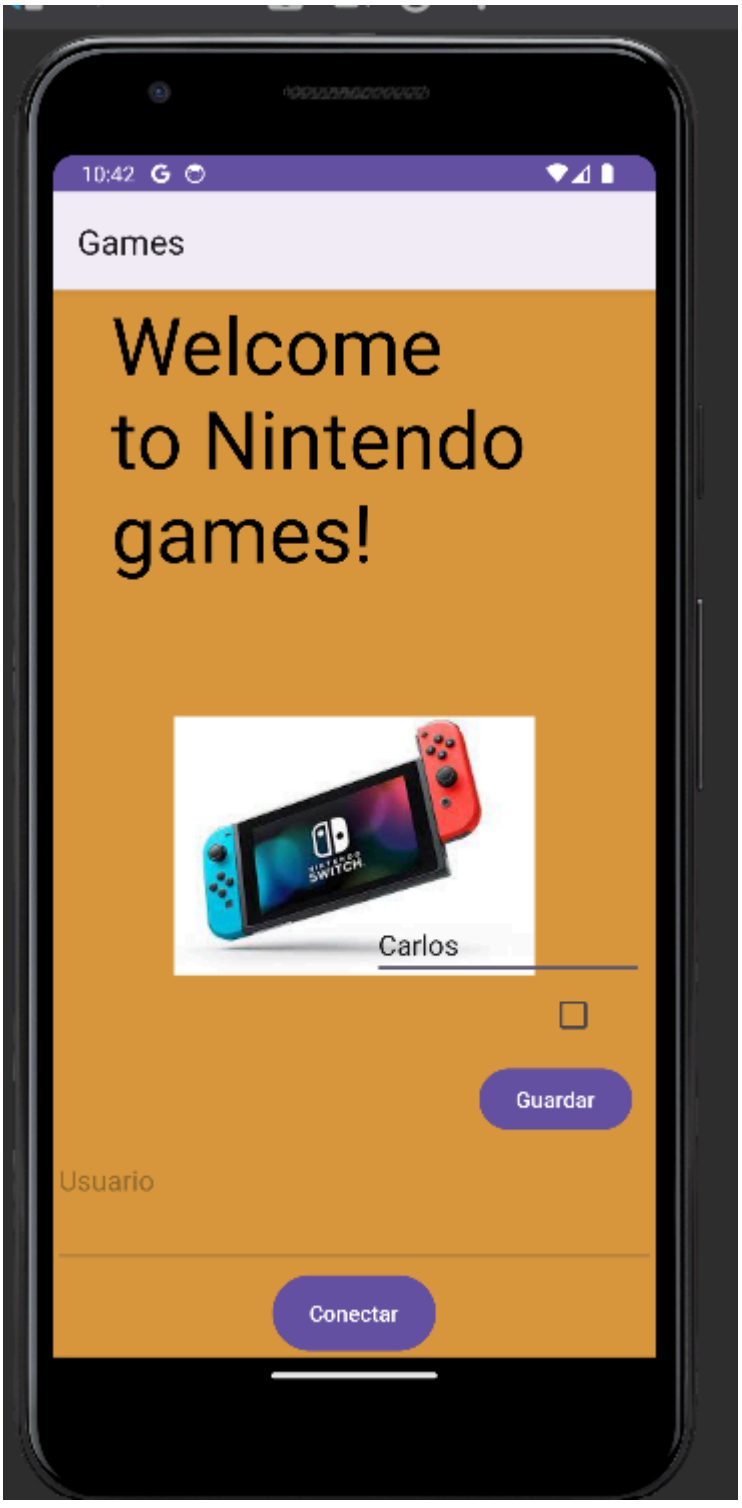
Estos valores son para que funcione correctamente el código de DetailActivity.

Una vez hecho esto, en activity_detail.xml, deberemos crear un TextView, donde se mostrará el resultado si lo hemos realizado correctamente.


```xml
<TextView
    android:id="@+id/tvName"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="26sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.775" />
```

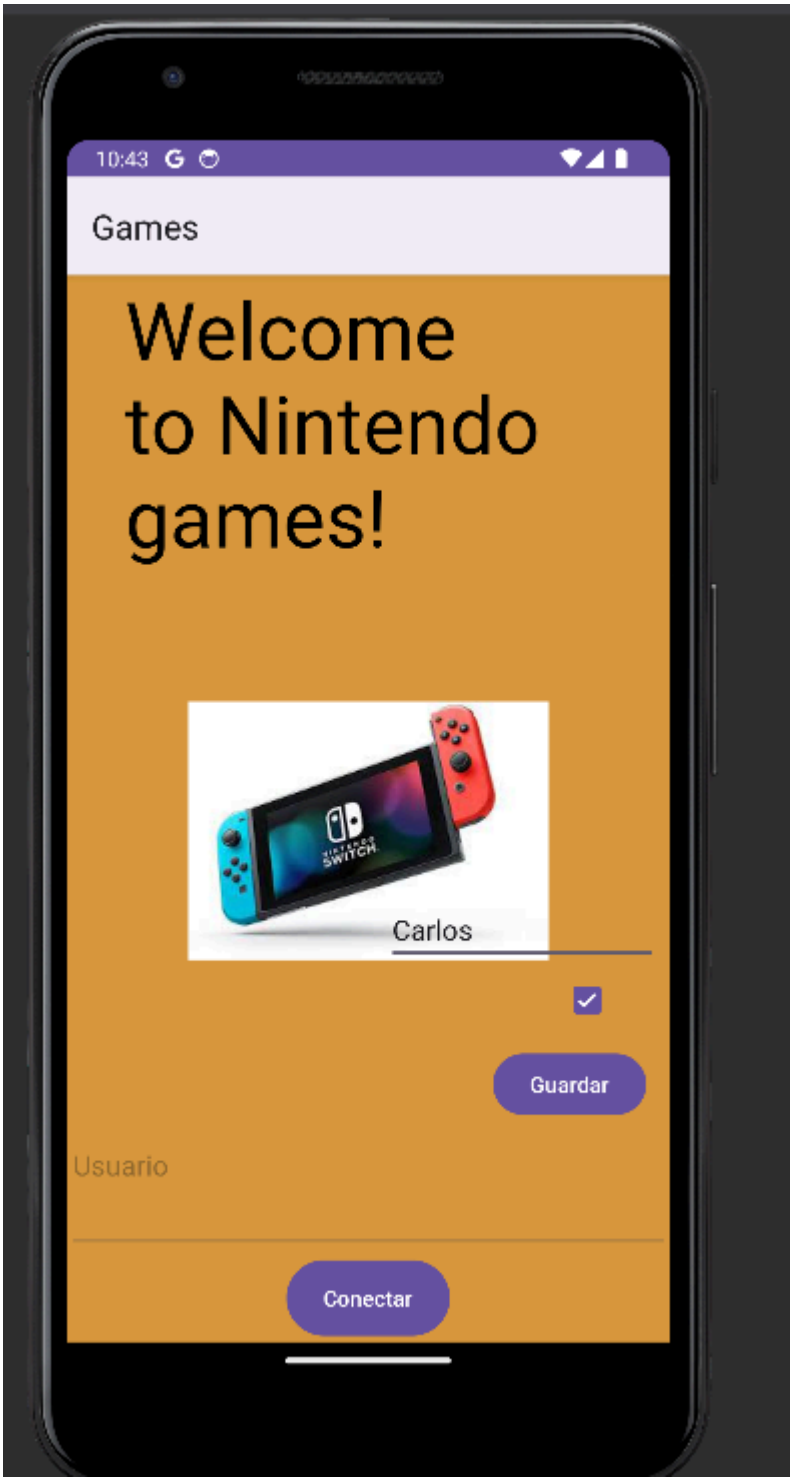
Una vez ya realizado el código, vamos a probar el funcionamiento de la aplicación:

Desde nuestro LoginFragment, podemos introducir los datos del DataStore:

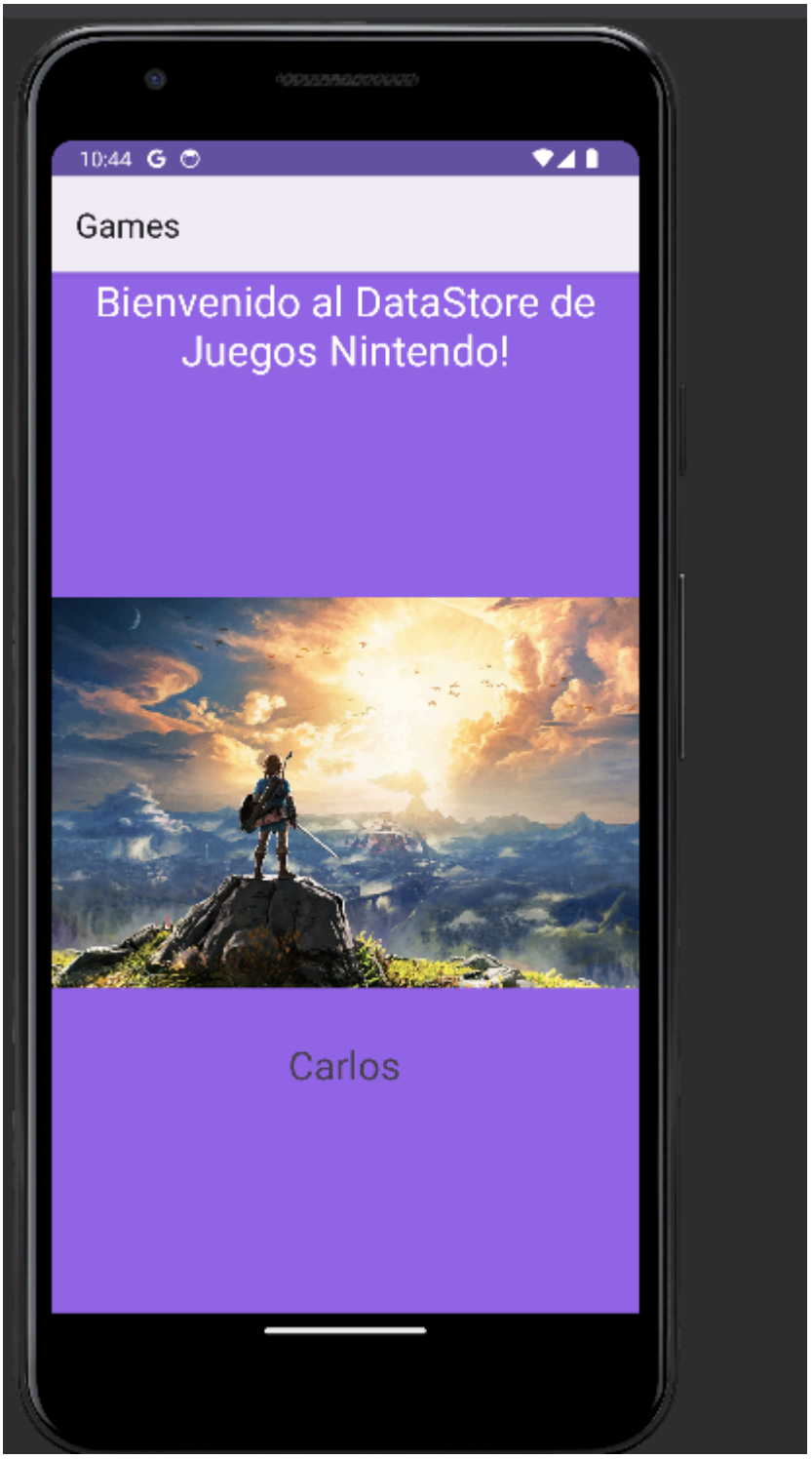


Si le damos a guardar, nos llevará a nuestro DetailActivity, pero antes de ello, vamos a pulsar en la casilla de justo arriba del botón guardar para que se nos actualice el

color de fondo como hemos definido anteriormente en el código:



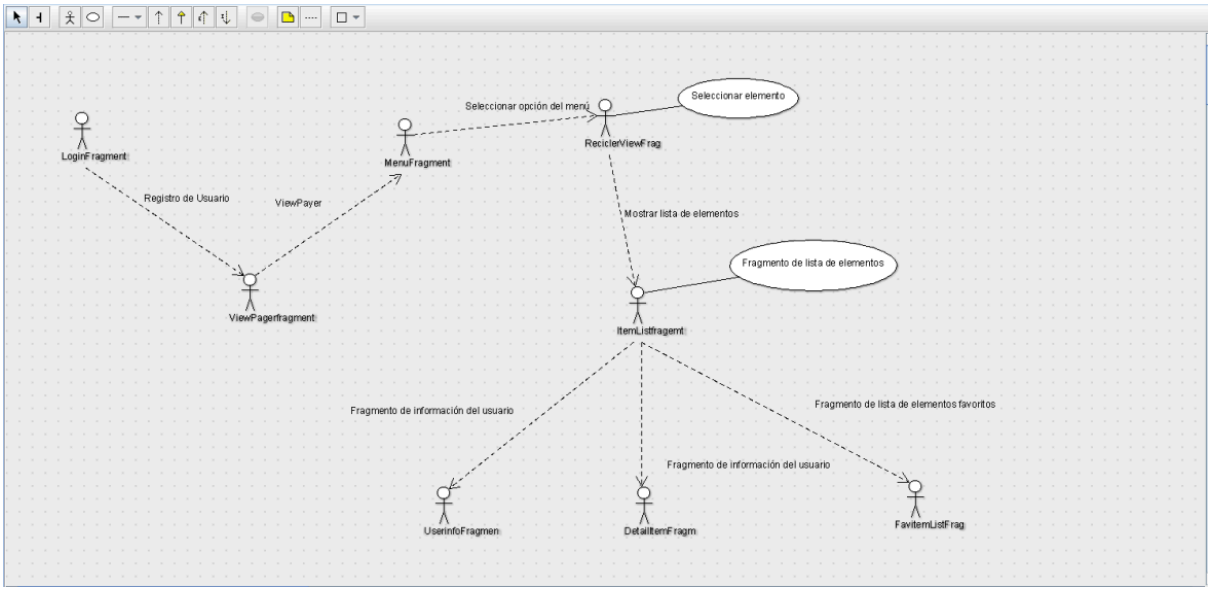
Le damos al botón guardar, y nos llevará a nuestra nueva actividad:



El proyecto está en el siguiente enlace:
https://github.com/Escaroz/JuegosProyectoExtra

Dentro de él, vienen los diagramas y los videos probando la aplicación desde mi móvil.

El diagrama de casos de uso, lo implemento en este pdf a continuación:



El diagrama de clases de mi aplicación, lo en este zip, y el archivo se llama DiagramaClase2.puml.