

Curso 2022-2023 **Profesor: Iván Lorenzo**





Hoja_UT3_03

Retrofit2

Vamos a realizar una aplicación para obtener los colegios e institutos de Gijón. Usaremos los datos abiertos que ofrece en su web (https://www.gijon.es/es/datos)

Crearemos la App 03 02.

DISEÑO

En esta primera práctica únicamente usaremos una Activity con un RecyclerView.

Así que diseñaremos el layout con un TextView con el título "Centros educativos" y un RecyclerView que ocupe el resto del ConstraintLayout.

Además, en el fichero **themes.xml** dentro de res → themes cambiaremos el parent a Theme.MaterialComponents.DayNight.NoActionBar para quitar la barra de navegación.

Crearemos también un diseño para mostrar únicamente el nombre del centro. Al layout podemos llamarlo item centro.xml

Habilitaremos el **viewBinding** (en el fichero build.gradle)

Mueve la clase **MainActivity** al paquete view

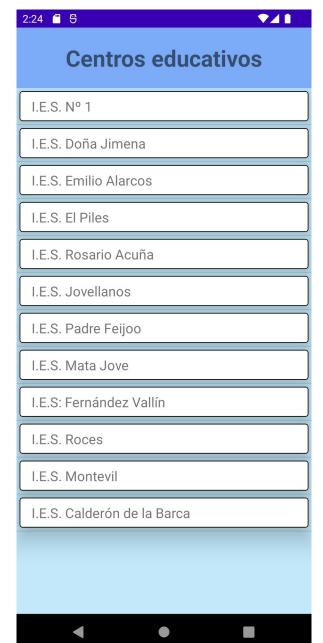
IMPLEMENTACIÓN

Vamos a crear una data class llamada **Centro** en el paquete model. Tendrá los atributos nombre, email, direccion, horario, localizacion, telefono y web

Crearemos un ViewModel llamado CentrosViewModel en el paquete viewmodel

Dentro de esta clase únicamente estableceremos un atributo llamado centrosLiveData que será un MutableLiveData que almacenará una lista de centros.

Por otro lado, dentro del paquete data.api crearemos **CentrosApiService**. El servicio se encuentra en la URL https://opendata.gijon.es/descargar.php?id=806&tipo=JSON





Curso 2022-2023 **Profesor: Iván Lorenzo**



PMDM

DAM2

Hoja_UT3_03

Previamente habrá que añadir las dependencias de Retrofit2 y de GSON. Además, podemos añadir ya las dependencias de LiveData, ViewModel y Fragments

```
//Retrofit2
implementation 'com.squareup.retrofit2:retrofit:2.9.0'
implementation 'com.squareup.retrofit2:converter-gson:2.9.0'

// LiveData
implementation 'androidx.lifecycle:lifecycle-livedata-ktx:2.5.1'

// ViewModel
implementation 'androidx.lifecycle:lifecycle-viewmodel-ktx:2.5.1'

//Fragments
implementation 'androidx.fragment:fragment-ktx:1.5.3'
```

Dentro de CentrosApiService tendremos un objeto retrofit, una constante con la url del servicio web, una interface llamada CentrosApiService que tendrá un método getCentros y un object CentrosApi

```
private const val URL_BASE =
   "https://opendata.gijon.es"

private val retrofit = Retrofit.Builder()
   .baseUrl(URL_BASE)
   .addConverterFactory(GsonConverterFactory.create())
   .build()

interface CentrosApiService
{
   @GET("descargar.php?id=806&tipo=JSON")
   suspend fun getCentros(): List<Centro>
}

object CentrosApi {
   val retrofitService: CentrosApiService by lazy { retrofit.create(CentrosApiService::class.java) }
}
```

Volviendo a la clase **Centro**, añade las anotaciones @SerializedName adecuadas para cada atributo.

Dentro del paquete model añade ahora la clase CentrosRepository

```
class CentrosRepository
{
    suspend fun get(): List<Centro> = CentrosApi.retrofitService.getCentros()
}
```



Curso 2022-2023 **Profesor: Iván Lorenzo**



PMDM

DAM2

Hoja_UT3_03

Modificaremos ahora la clase CentrosViewModel para crear un método cargarDatos que llamará al servicio web:

```
class CentrosViewModel: ViewModel()
{
    val centrosLiveData = MutableLiveData<List<Centro>>()

    private val repository = CentrosRepository()

    fun cargarDatos()
    {
        viewModelScope.launch {
            centrosLiveData.postValue(repository.get())
        }
    }
}
```

Por último, haremos una clase llamada CentroAdapter dentro del paquete view para establecer en el recyclerView.

En la clase MainActivity crearemos una instancia del viewModel utilizando el método viewModels()

private val centrosViewModel: CentrosViewModel by viewModels()

Y desde el método onCreate llamaremos al método cargarDatos del viewModel y observaremos el LiveData de centros para establecer el adapter al recyclerViewCentros.

```
centrosViewModel.cargarDatos()

centrosViewModel.centrosLiveData.observe(this) { animales ->
    with (binding.recyclerViewCentros) {
        adapter = CentroAdapter(animales) { centro -> onClickCentro(centro) }
        layoutManager = LinearLayoutManager(context)
        addItemDecoration(DividerItemDecoration(context, DividerItemDecoration.VERTICAL))
    }
}
```

Si hacemos click en un centro simplemente mostrará un Toast con el nombre del centro seleccionado.

Asegúrate también que has añadido el permiso de INTERNET a la aplicación.

AMPLIACIÓN:

Prueba a añadir también los colegios. Los colegios se pueden obtener de https://opendata.gijon.es/descargar.php?id=754&tipo=JSON