

MODULO 4 | Desarrollo de Aplicaciones Móviles

Android Kotlin | Ignacio Cavallo

https://github.com/cavigna/modulo_desarrollo_de_aplicaciones_moviles_android_kotlin

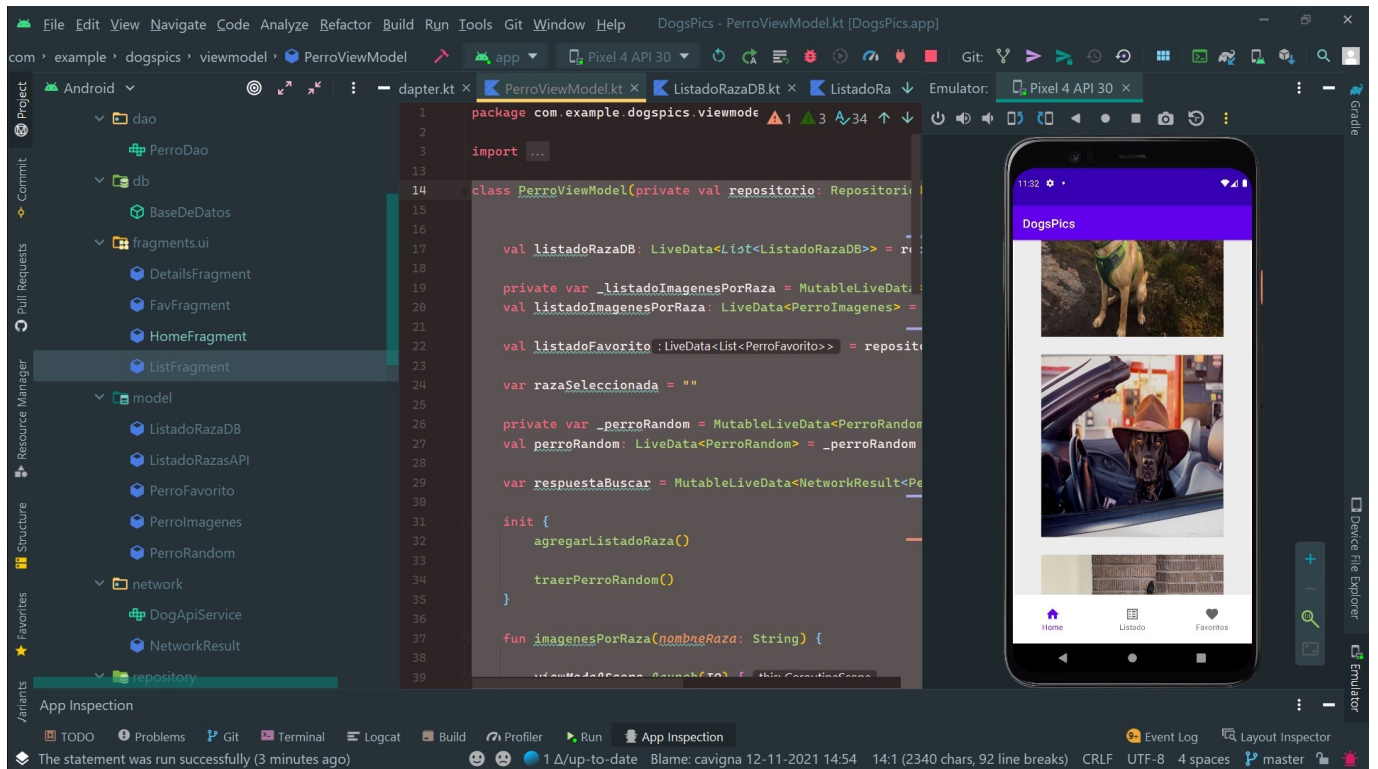
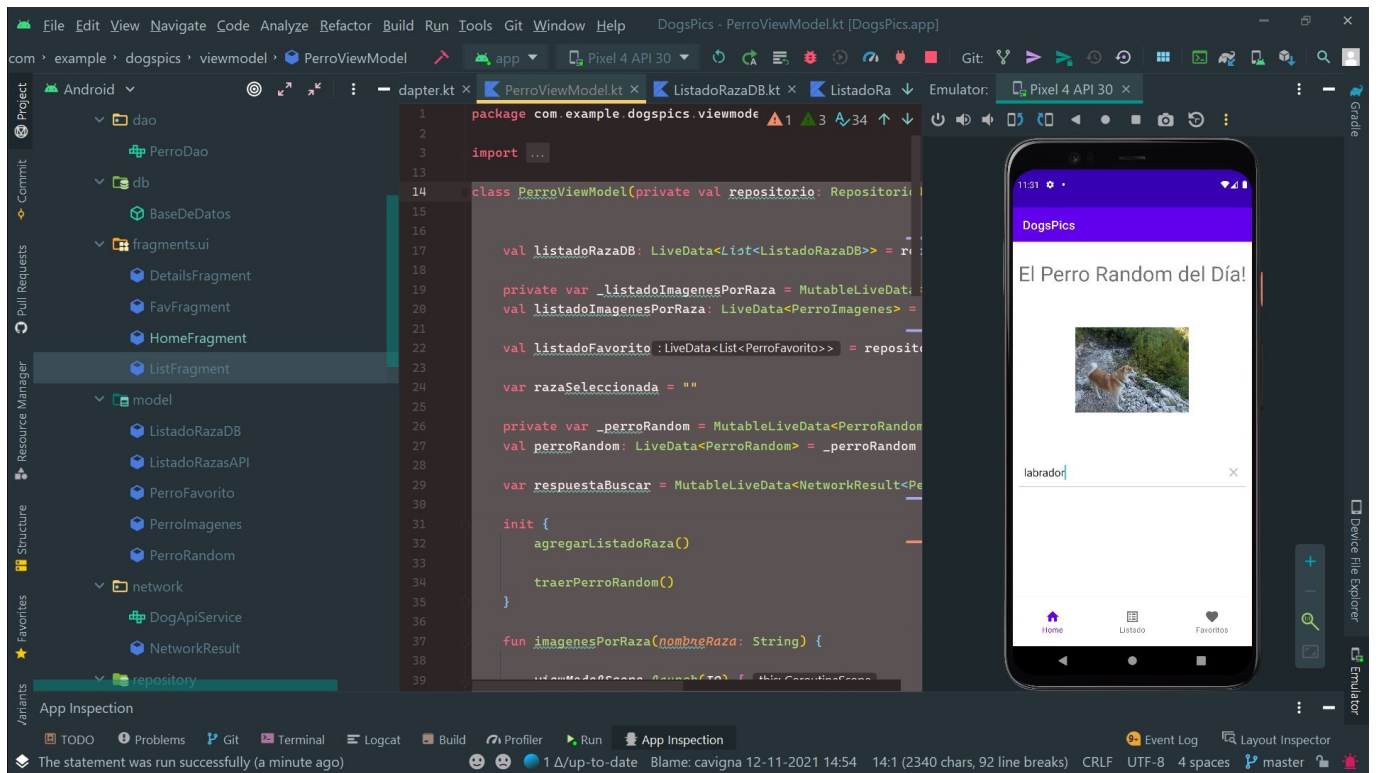
Clase 95 | 12-11

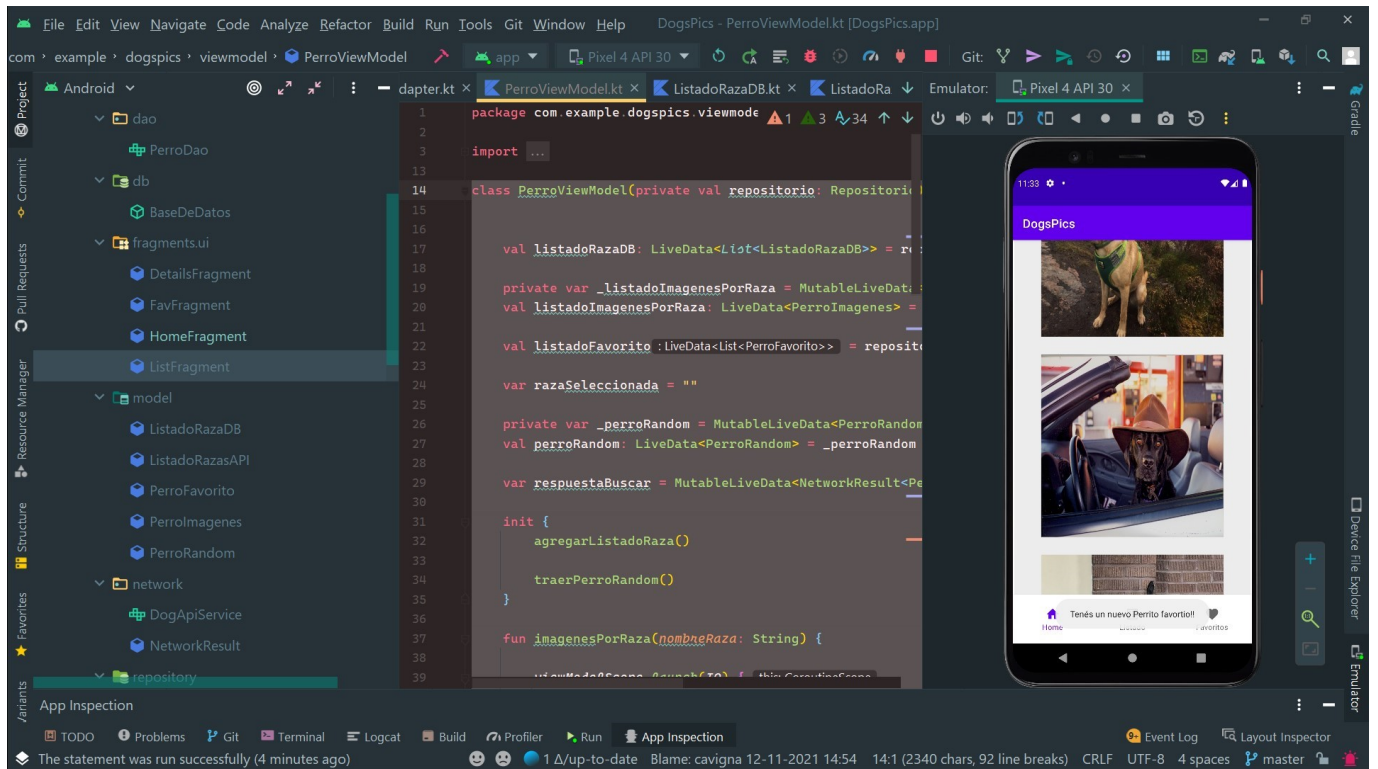
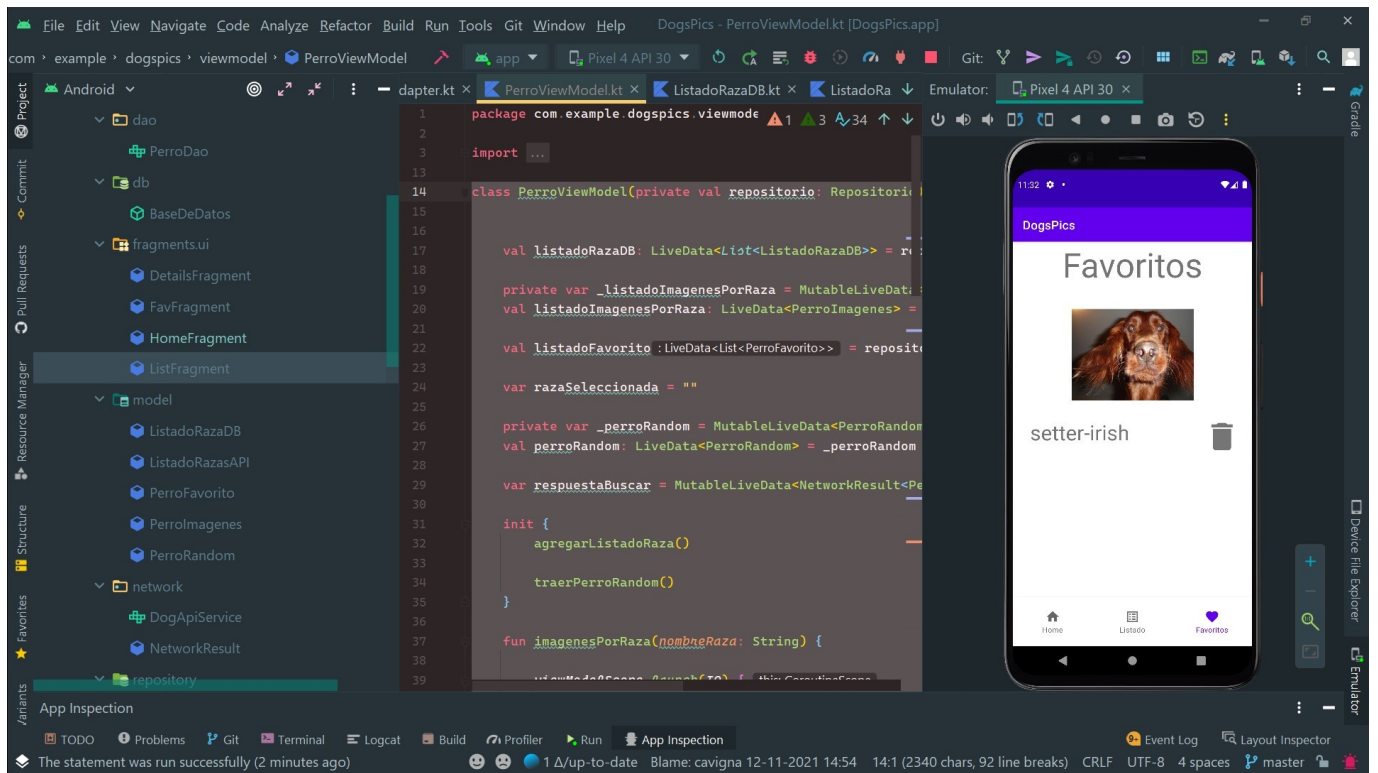
En esta clase algunos compañeros preguntaron dudas sobre la tarea. A continuación continuamos desarrollando el ejercicio propuesto la clase anterior. Dejo a continuación algunos avances:

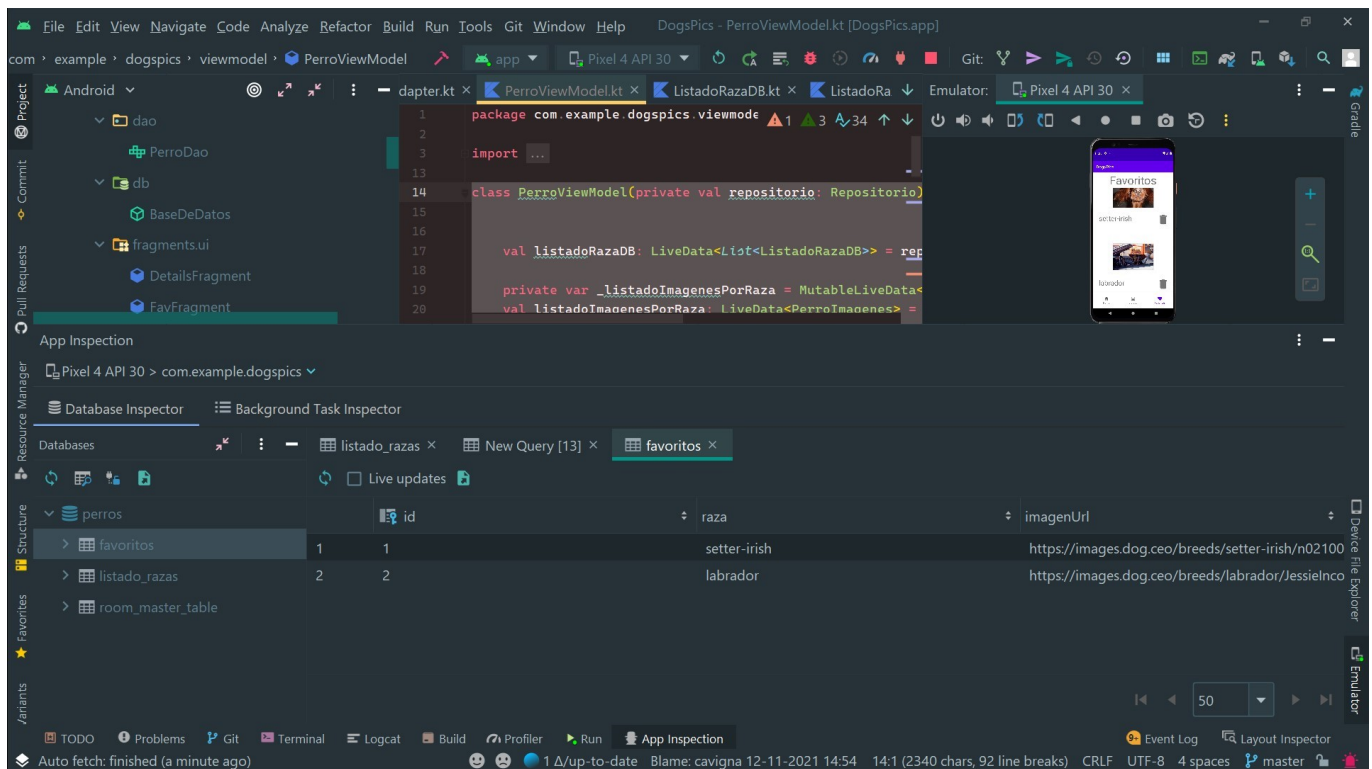
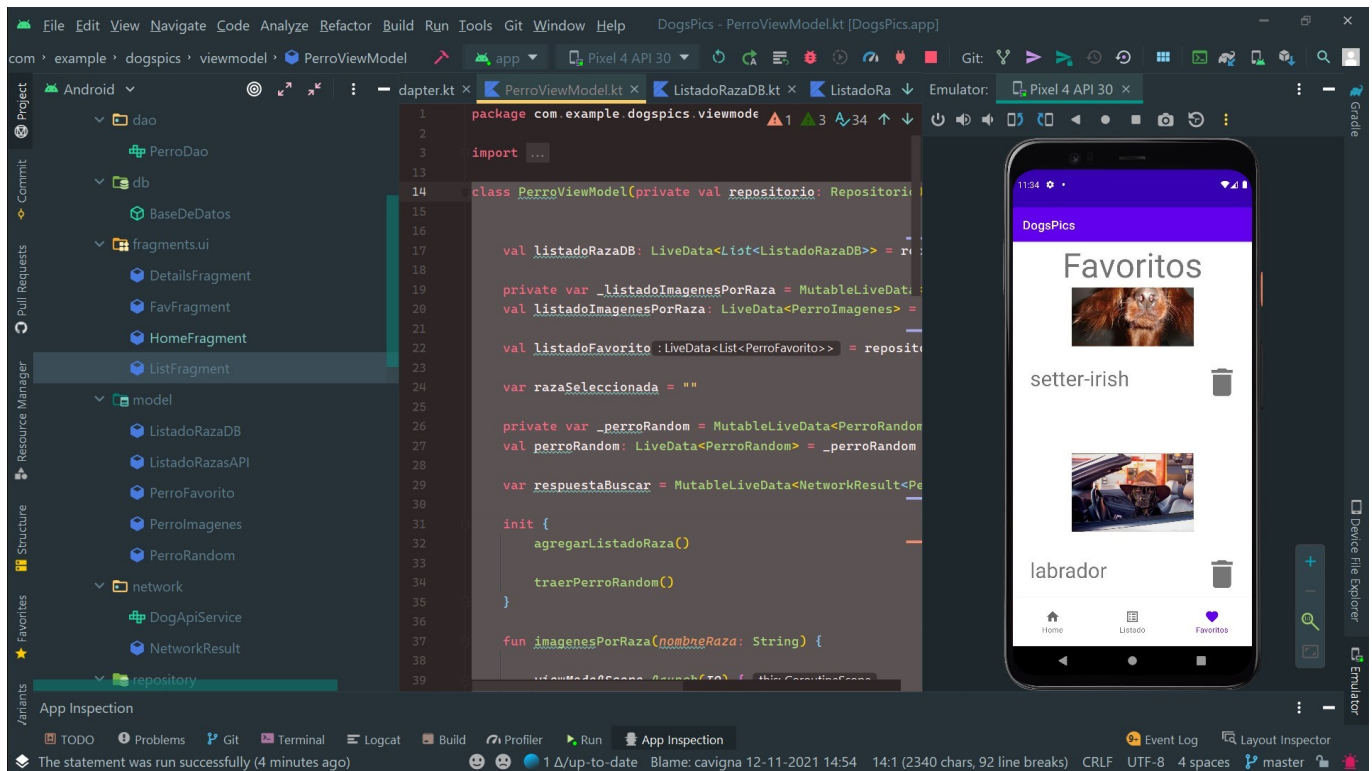
Aquí esta el link del repo

<https://github.com/cavigna/DogsPics>

A continuación muestro unas capturas de pantalla







CODIGO

PerroViewModel.kt

```
class PerroViewModel(private val repositorio: Repositorio) : ViewModel() {

    val listadoRazaDB: LiveData<List<ListadoRazaDB>> =
    repositorio.listadoRazasDB().asLiveData()
```

```

private var _listadoImagenesPorRaza = MutableLiveData<PerroImagenes>()
val listadoImagenesPorRaza: LiveData<PerroImagenes> = _listadoImagenesPorRaza

val listadoFavorito = repositorio.listadoFavoritos().asLiveData()

var razaSeleccionada = ""

private var _perroRandom = MutableLiveData<PerroRandom>()
val perroRandom: LiveData<PerroRandom> = _perroRandom

var respuestaBuscar = MutableLiveData<NetworkResult<PerroImagenes>>()

init {
    agregarListadoRaza()

    traerPerroRandom()
}

fun imagenesPorRaza(nombreRaza: String) {
    viewModelScope.launch(IO) {
        _listadoImagenesPorRaza.postValue(repositorio.imagenesPorRaza(nombreRaza))
    }
}

fun agregarListadoRaza() {
    viewModelScope.launch {
        val listado = repositorio.listadoRazaAPI()
        val listadoRazaDB = mutableListOf<ListadoRazaDB>()

        for (s in listado) {
            listadoRazaDB.add(ListadoRazaDB(raza = s))
        }

        repositorio.agregarListaRazaDB(listadoRazaDB)
    }
}

fun agregarFavorito(perro: PerroFavorito) {
    viewModelScope.launch {
        repositorio.asegarFavorito(perro)
    }
}

```



```

fun borrarFavorito(perro: PerroFavorito) {
    viewModelScope.launch {
        repositorio.borrarFavorito(perro)
    }
}

fun traerPerroRandom() {
    viewModelScope.launch(IO) {
        _perroRandom.postValue(repositorio.perroRandom())
    }
}

fun buscarRaza(nombreRaza: String){
    respuestaBuscar.postValue(NetworkResult.Loading())

    viewModelScope.launch {
        try {
            val respuestaExitosa =
repositorio.buscarRazaAPI(nombreRaza).body()!!
            respuestaBuscar.postValue(respuestaExitosa)
        }catch (e: Exception){
            respuestaBuscar.postValue(NetworkResult.Error(e.message))
        }
    }
}

fun buscadorRazaDB(nombreRaza: String){
    viewModelScope.launch {
        razaSeleccionada =
repositorio.buscarRazaDB(nombreRaza).asLiveData().value?.get(0)?.raza.toString()
    }
}
}

```