Evidencia día 01 - Semana 16

**Cristian Maturana Ibarra**

**ViewModel:**

class

SismosViewModel

(

application: Application) : AndroidViewModel(application)

{

private

var

repositorio: SismosRepositorio

init

{

val

sismosDao =

SismosDatabase.crearDatabase(application).obtenSismosDelDao()

repositorio = SismosRepositorio(sismosDao)

}

fun

traemeLoDelServer

()

{

repositorio.obtenDataDelServer()

}

fun

exponeSismosDeDB

()

:

LiveData<List<SismosModel>>

{

return

repositorio.exponeSismosDeLaBaseDeDatos()

}

fun

exponeSismosFavoritosDeDB

()

:

LiveData<List<SismosModel>>

{

return

repositorio.exponeFavoritosDeLaBaseDeDatos()

}

fun

cambiaeEstadoDeFavorito

(

sismo:

SismosModel

)

{

repositorio.cambiaeEstadoDeFavorito(sismo)

}

fun

borraSismosFavoritosDeDB

()

{

repositorio.borrarFavoritosDeLaBaseDeDatos()

}

}

**ViewModel Factory:**

class

Factory

(

private

val

application: Application) : ViewModelProvider.Factory {

override

fun

<

T : ViewModel

>

create

(

modelClass:

Class

<

T

>)

:

T

{

if

(modelClass.isAssignableFrom((SismosViewModel::

class

.

java

)))

{

return

SismosViewModel(application)

as

T

}

throw

IllegalArgumentException(

"ViewModel Desconocido"

)

}

}

**Database:**

@Database(entities = [SismosModel::class], version = 1)

abstract

class

SismosDatabase

:

RoomDatabase

()

{

abstract

fun

obtenSismosDelDao

()

:

SismoDao

companion

object

{

@Volatile

private

var

baseDeDatosCreada: SismosDatabase? =

null

fun

crearDatabase

(

context:

Context

)

:

SismosDatabase

{

if

(baseDeDatosCreada ==

null

)

{

synchronized(

this

)

{

baseDeDatosCreada = Room.databaseBuilder(

context,

SismosDatabase::

class

.

java

,

"base\_De\_Datos\_Sismos"

).build()

}

}

return

baseDeDatosCreada!!

}

}

}



(función favorito implementado sólo a nivel de Toast, no como funcionalidad de la app aún)

**Reflexión:**

Principalmente queda claro lo que enseña el profesor y tiene buena disposición para aclarar dudas, estoy ya más familiarizado con las API.