CFGS DESARROLLO DE APLICACIONES MULTIPLATAFORMA

UD 5. Persistencia

Módulo: Programación multimedia y dispositivos móviles

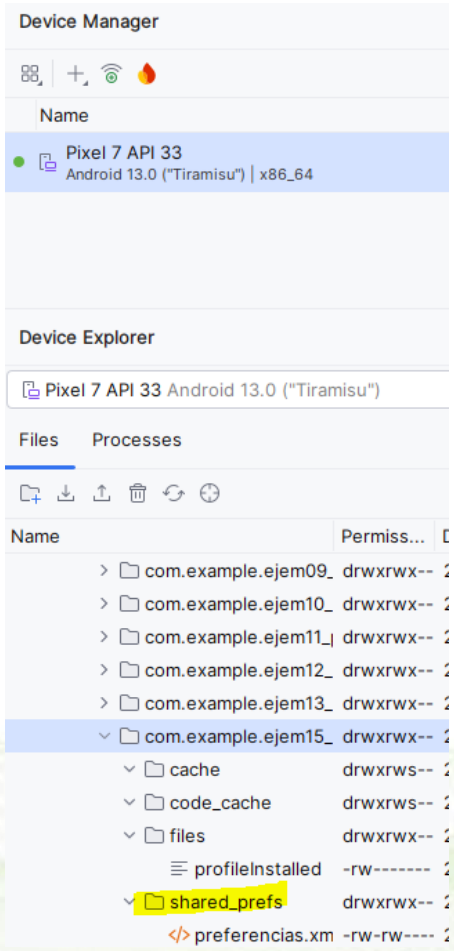
Víctor J. Vergel Rodríguez

Centro de Enseñanza
Gregorio Fernández

Persistencia. SharedPreference

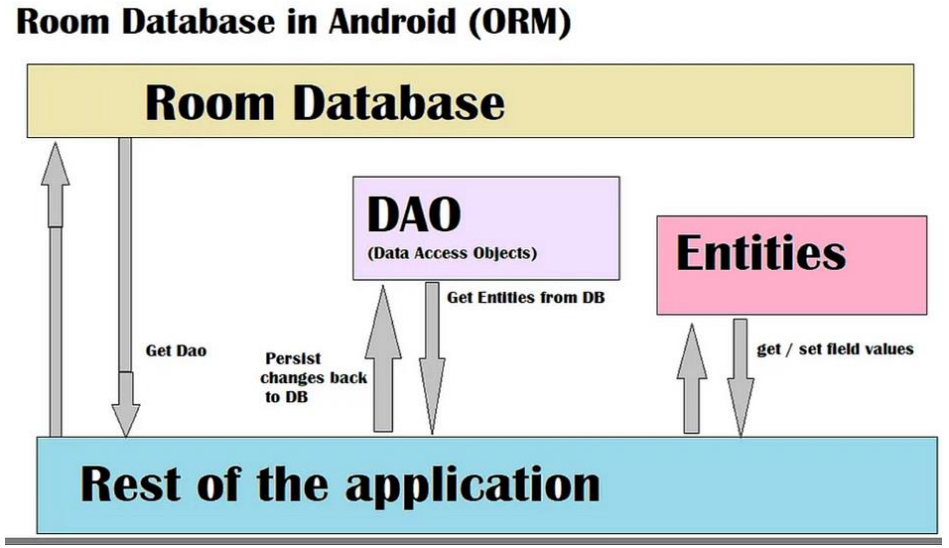
- Permite almacenar datos en fichero xml para próximas ejecuciones de la aplicación

```
val pref = getSharedPreferences("datos", MODE_PRIVATE)

.....

val editor = pref.edit()
editor.putString("datos", “cadena")


editor.commit()
```



Persistencia. Base de datos ROOM

- Nos permite una capa de abstracción del SQL. **Room es el ORM** (Object Relational Mapping) de Android propuesto por Google en Jetpack. Tres partes fundamentales



- **Database class**: Se encarga de albergar la base de datos y manejar aspectos como la apertura, conexión, cerrado, etc. También se encarga de proveer DAOs.
- **DAOs** (Data Access Objects): Proveen métodos que la aplicación usará como queries para recuperar o guardar datos.
- **Entities**: Representan las tablas de la base de datos.

Persistencia. Base de datos ROOM

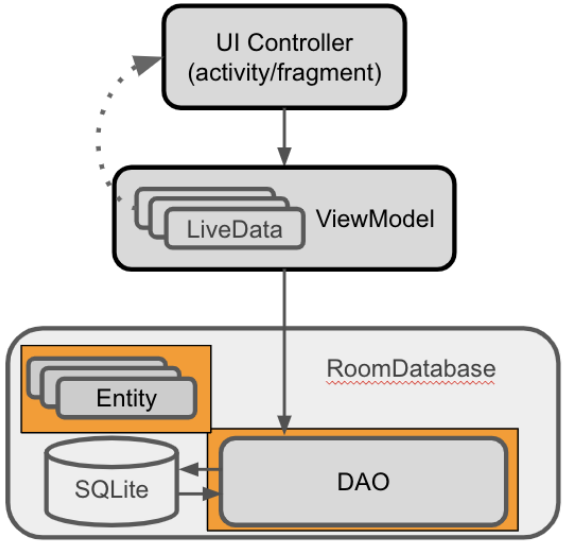
- **Anotaciones importantes:**

@Entity(tableName = "usuarios") – Vincula la clase con la tabla correspondiente

OBLIGATORIO -> @PrimaryKey(autoGenerate = true) – Un atributo de la clase será la clave primaria en la tabla

@ColumnInfo(name = "nombre") – Atributo de una clase vinculado a campo de una tabla

@Dao - Marca la interfaz o clase abstracta como un Data Access Object (DAO). Los DAOs proporcionan métodos para interactuar con la base de datos



Persistencia. Base de datos ROOM

- **Importaciones a realizar:**

```
plugins {
   //ROOM DATABASE
   id("com.google.devtools.ksp") version "2.0.21-1.0.25“  -> OJO, la versión de kotlin actual es 2.0.21 en libs.versions
}

……..

dependencies {

   val room_version = "2.8.4"

   implementation("androidx.room:room-runtime:$room_version")
   annotationProcessor("androidx.room:room-compiler:$room_version")

   // To use Kotlin Symbol Processing (KSP)
   ksp("androidx.room:room-compiler:$room_version")
```

En el Activity Principal:

```
val baseDatos: RoomDatabase=Room.databaseBuilder(this, BD::class.java, "bdUsuarios").allowMainThreadQueries().build()
```

NOTA.- Cuando se hace desde un Framgent en lugar de this colocaremos requireContext()

Persistencia. Base de datos ROOM


```
@Entity(tableName = "persona")
data class Persona(@PrimaryKey(autoGenerate = true) var id:Int = 0,
                   @ColumnInfo(name = "nombre") val nombre:String = "",
                   @ColumnInfo(name = "edad") val edad:Int = 0,
                   @ColumnInfo(name = "direccion") val direccion:String = "")

@Dao
abstract class PersonaDAO {
    @Query("SELECT * FROM persona")
    abstract fun listar(): List<Persona?>?
    @Query("SELECT * FROM persona WHERE id = :id")
    abstract fun recuperarUsuario(id: Int ): Persona?
    @Insert
    abstract fun insertar(p: Persona)
    @Delete
    abstract fun eliminar(p: Persona)
    @Update
    abstract fun actualizar(p: Persona)
}

@Database(entities = [Persona::class], version = 1)
public abstract class BD : RoomDatabase() {
    public abstract fun personaDao(): PersonaDAO?
}
```