



# TI-2600 Laboratório de Desenvolvimento Multiplataforma

Antonio Carvalho - Treinamentos



# KMP SQLite com Room Database

## KMP Room Database

- ▶ A biblioteca de persistência Room fornece uma camada de abstração sobre o *database* SQLite para permitir acesso fluente ao banco de dados. Em particular, o Room oferece os seguintes benefícios:
  - ▶ Verificação em tempo de compilação sobre consultas SQL.
  - ▶ Anotações de conveniência que minimizam o código repetitivo e propenso a erros.
  - ▶ Caminhos de migração de banco de dados simplificados.

## KMP Room Database

- ▶ Existem três componentes principais no Room:
  - ▶ A classe Database que serve como o ponto de acesso principal para a conexão subjacente com os dados persistentes do aplicativo.
  - ▶ Entidades de dados que representam tabelas no banco de dados.
  - ▶ DAOs (*Data Access Object*) que fornecem métodos para o aplicativo pode usar para consultar, atualizar, inserir e excluir dados no banco de dados

# KMP Room Database

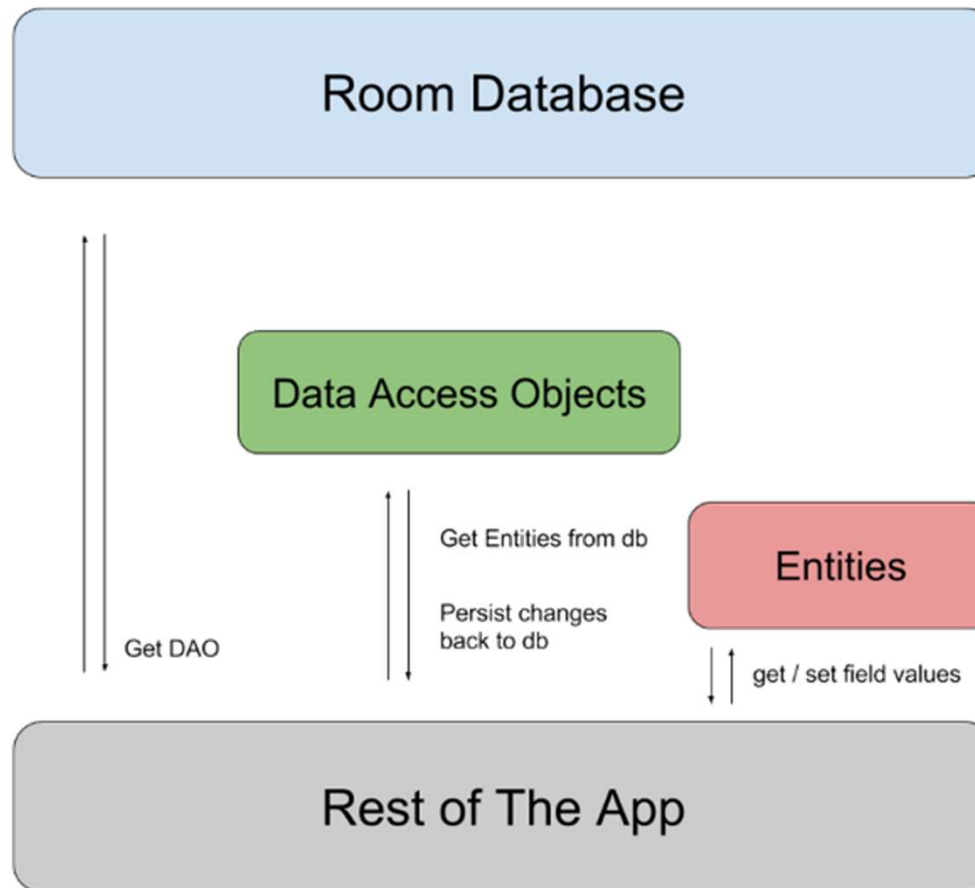


Diagrama da Arquitetura da Biblioteca Room

<https://developer.android.com/training/data-storage/room>

# KMP Room Database - Como instalar

► Para instalar o Room é preciso alterar o arquivo **libs.versions.toml**

```
[versions]
...
roomVersion = "2.7.1"

[libraries]
...
room-runtime = { group="androidx.room", name="room-runtime", version.ref="roomVersion"}
room-compiler = { group="androidx.room", name="room-compiler", version.ref="roomVersion"}
room-ktx = {group="androidx.room", name="room-ktx", version.ref="roomVersion"}

[plugins]
...
# KSP
google-devtools-ksp = { id = "com.google.devtools.ksp", version.ref = "kotlin" }
# Room
androidx-room = { id = "androidx.room", version.ref="roomVersion"}
```

# KMP Room Database - Como instalar

► adicionar as dependências no arquivo **build.gradle.kts** do módulo.

```
plugins {  
    // ksp  
    alias(libs.plugins.google.devtools.ksp)  
    // room  
    alias(libs.plugins.androidx.room)  
}  
  
kotlin {  
    // room  
    room {  
        schemaDirectory("$projectDir/schemas")  
    }  
    sourceSets {  
        commonMain.dependencies {  
            // room  
            implementation(libs.room.runtime)  
            ksp(libs.room.compiler)  
            implementation(libs.room.ktx)  
        }  
    }  
}
```

## KMP Room Database - Como instalar

- ▶ e em seguida adicionar o plugin no arquivo **build.gradle.kts** do projeto.

```
plugins {  
    ...  
    //ksp  
    alias(libs.plugins.google.devtools.ksp)  
}
```



## KMP Room Database - Como instalar

- Crie o arquivo para ser a entidade, com as anotações `@Entity` e `@Id` como exemplo:

```
import androidx.room.Entity
import androidx.room.PrimaryKey

@Entity(tableName = "Contato")
data class Contato (
    @PrimaryKey(autoGenerate = true)
    val id : Long = 0,
    val nome : String,
    val email : String,
    val telefone : String
)
```

# KMP Room Database - Como instalar

- ▶ Crie a interface DAO para permitir o acesso de gravação, consulta, atualização e remoção dos dados referente a tabela da entidade:

```
import androidx.room.Dao
import androidx.room.Delete
import androidx.room.Insert
import androidx.room.Query

@Dao
interface ContatoDAO {

    @Query("Select * from Contato")
    suspend fun getAll() : List<Contato>

    @Insert
    suspend fun insert(contato : Contato)

    @Delete
    suspend fun delete(contato : Contato)
}
```

## KMP Room Database - Como instalar

► Crie a classe para informar quais são as entidades e os DAOs:

```
import androidx.room.Database
import androidx.room.RoomDatabase

@Database(entities = [Contato::class], version = 1)
abstract class ContatoDatabase() : RoomDatabase() {
    abstract fun contatoDao() : ContatoDAO
}
```

## KMP Room Database - Como instalar

► Crie uma variável para fornecer o acesso ao banco de dados:

```
import androidx.room.Room

val db : ContatoDatabase = Room.databaseBuilder(
    applicationContext,
    ContatoDatabase::class.java,
    "database-agenda"
).build()
```