

Práctica base de datos



Saúl mellado y Eneko Rebollo

Contenido

[Paquete Model 2](#_Toc88921187)

[Paquete Pojo 2](#_Toc88921188)

[Clase Commit 2](#_Toc88921189)

[Clase Departamento 3](#_Toc88921190)

[Clase Issue 3](#_Toc88921191)

[Clase Programador 4](#_Toc88921192)

[Clase Proyecto 4](#_Toc88921193)

[Clase Repositorio 5](#_Toc88921194)

[Paquete PojoDTO 5](#_Toc88921195)

[Clase CommitDTO 5](#_Toc88921196)

[Clase DepartamentoDTO 6](#_Toc88921197)

[Clase IssueDTO 6](#_Toc88921198)

[Clase ProgramadorDTO 7](#_Toc88921199)

[Clase ProyectoDTO 7](#_Toc88921200)

[Clase RepositorioDTO 7](#_Toc88921201)

[Paquete Driver 8](#_Toc88921202)

[Paquete Repository 8](#_Toc88921203)

[Paquete Mapper 8](#_Toc88921204)

# Paquete Model

## Paquete Pojo

En este paquete tenemos nuestras clases pojo.

### Clase Commit

Texto

Descripción generada automáticamente

### Clase Departamento

Texto

Descripción generada automáticamente

### Clase Issue

Texto

Descripción generada automáticamente

### Clase Programador

Texto

Descripción generada automáticamente

### Clase Proyecto

Texto

Descripción generada automáticamente

### Clase Repositorio

Texto

Descripción generada automáticamente

## Paquete PojoDTO

En este paquete tenemos nuestros dto

### Clase CommitDTO

Texto

Descripción generada automáticamente

### Clase DepartamentoDTO

Texto

Descripción generada automáticamente

### Clase IssueDTO

Texto

Descripción generada automáticamente

### Clase ProgramadorDTO

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

### Clase ProyectoDTO

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

### Clase RepositorioDTO

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

# Paquete Driver

Aquí tenemos nuestro driver de la base de datos.

Definimos las variables que vamos a utilizar

Texto

Descripción generada automáticamente

El metodo initConfig inicializa la base de datos con una configuración predeterminada.

El metodo open abre la conexión del servidor con la base de datos. En nuestro caso como estamos utilizando sqlite le metemos el driver correspondiente.

Texto

Descripción generada automáticamente

Creamos dos consultas preparadas a la base de datos de tipo select con parámetros opcionales si son necesarios.

Texto

Descripción generada automáticamente

Creamos otra consulta de tipo select en la que le establecemos el desplazamiento y el numero de registros que tiene.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Creamos también consultas para insertar , actualizar y eliminar registros de nuestra base de datos.

Texto

Descripción generada automáticamente

Por último creamos otra consulta de update pero en este caso preparada utilizando prepared Statement.

Texto

Descripción generada automáticamente

# Paquete Mapper

En esta clase creamos nuestros Mapper.

## Clase RepositoryMapper

Creamos un metodo para crear Programadores a partir de valores pasados por parámetros

Texto

Descripción generada automáticamente

Creamos un metodo para crear Commit a partir de valores pasados por parámetros

Texto

Descripción generada automáticamente

Creamos un metodo para crear Departamento a partir de valores pasados por parámetros

Texto

Descripción generada automáticamente

Creamos un metodo para crear Departamento a partir de valores pasados por parámetros

Texto

Descripción generada automáticamente

Creamos un metodo para crear Repositorio a partir de valores pasados por parámetros

Texto

Descripción generada automáticamente

Creamos un metodo para crear Proyecto a partir de valores pasados por parámetros

Texto

Descripción generada automáticamente

## Clase CommitMapper

## Clase DepartamentoMapper

## Clase IssueMapper

## Clase ProgramadorMapper

## Clase ProyectoMapper

## Clase RepositorioMapper

# Paquete Repository

En este paquete creamos todos nuestros repositorios.