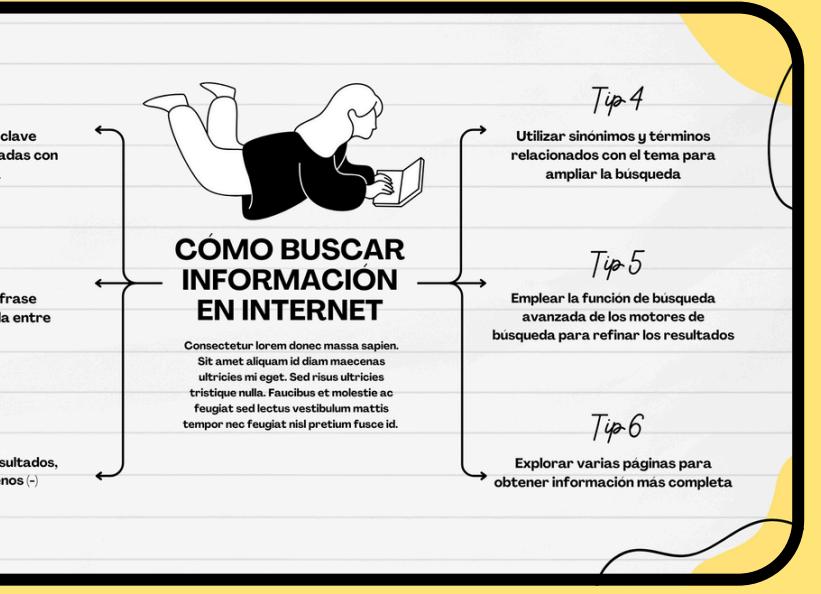


# API REST: JUEGOS OLIMPICOS



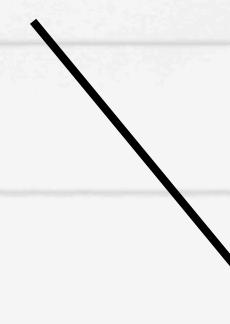
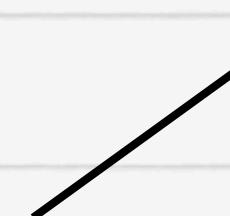
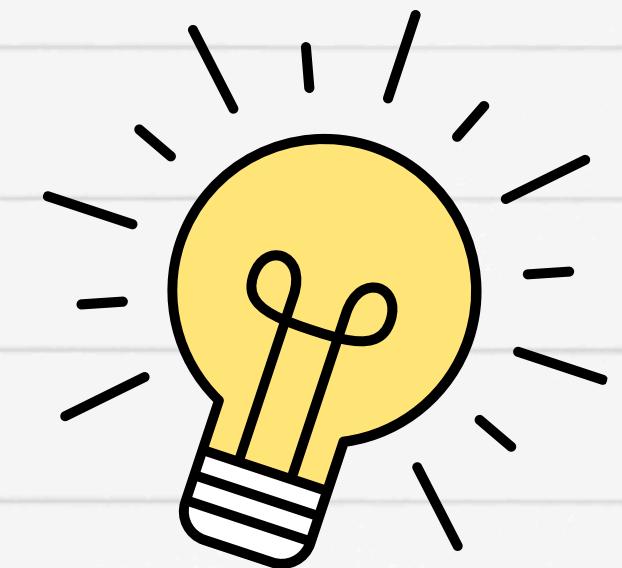
# IMPLEMENTACIÓN

API Rest

SpringBoot

Maven

Java



# Repositories

Clases que encapsulan la persistencia y consultas a la base de datos

# Controllers

Clases que sirven como endpoints y dependen de los repositories

# DBConnection

Clase que encapsula la conexión a la base de datos

# TipoPuntaje

Mayor numero de variables, ahora es una clase y no un Enumerator, esta relacionada con Modalidad.

# Equipo

Clase necesaria para conocer los participantes de un equipo.

# Multiplicidad

Cambio la multiplicidad en un gran numero de clases

## CAMBIOS EN DIAGRAMA DE CLASES

# CAMBIO EN FIGMA



**Reducción del scope a  
funcionalidades del Juez**

#### Juez - Inicio de sesión

9:30



Email  
Email

Contraseña  
Contraseña

Ingresar

#### Juez - Selección de compet...

9:30

Seleccione para calificar

- Atletismo - Carrera de velocidad  
22/6/2024
- Escríma - Florete  
29/6/2024
- Escríma - Espada  
30/6/2024
- Natación - Carrera de relevos  
22/6/2024
- Atletismo - Carrera de velocidad  
02/8/2024
- Atletismo - Carrera de Vallas  
02/8/2024
- Halterofilia  
09/8/2024
- Surf - Longboard  
14/10/2024

Finalizado

#### Juez - Selección de particip...

9:30

Escríma | Florete

- J José Abady
- J Javier Dannonne
- F Fernando Planchado
- R Rodrigo Lujo vio
- A Ana Bergallo
- B Bettina Carrizo
- S Sebastian Feirrés

Finalizado

#### Juez - Calificar participante

9:30

Escríma | Florete

J José Abady

Attaque 1 Attaque 2 Attaque 3

- Derecho a ataque  5
- Toque  10
- Combate  10
- Técnica  10

Calificar

#### Juez - Selección de Grupo

9:30

Atletismo | Carrera de velocidad

Equipo 1

- A Equipo 1
- B Equipo 2
- C Equipo 3
- D Equipo 4
- E Equipo 5
- F Equipo 6
- G Equipo 7

Finalizado

#### Juez - Selección de particip...

9:30

Equipo 1

Bianca Luzzato

J Juan Cosentino

L Lucas Alegre

S Santiago Ferraro

Finalizado

#### Juez - Calificar grupo-indivi...

9:30

Atletismo | Carrera de relevo

B Bianca Luzzato

Tiempo  1:38

Palancas  2

Velocidad  38

Técnica  7

Calificar



# CASOS DE USO

- INICIAR SESIÓN
- VER DISCIPLINAS/MODALIDADES/CATEGORÍAS
- VER PARTICIPANTES DE DICHA CATEGORÍA Y MODALIDAD
- CALIFICAR PARTICIPANTE SEGÚN PUNTAJES DE LA CATEGORÍA
- VER CALIFICACIONES/CALIFICADOS
- VER EQUIPOS DE DICHA CATEGORÍA Y MODALIDAD
- CALIFICAR MIEMBROS DE EQUIPOS SEGÚN PUNTAJE DE LA CATEGORÍA

## **DEPENDENCY INVERSION**

Los controllers dependen de interfaces, en vez de en las clases concretas o de los detalles. Esto nos permite reducir el acoplamiento al no depender de implementaciones concretas.

## **SINGLE RESPONSABILITY**

Mediante la clase DatabaseConnection se separó la lógica de la conexión a la base de datos de los repositorios, porque si a futuro se quieren añadir nuevos métodos ya poseemos una razón para cambiar en los Repository.

## **INTERFACE SEGREGATION**

Los controllers (principalmente) dependen de interfaces específicas de los repositorios.

En vez de depender de una interfaz general IRepository que implementaran todos los repositorios

# **PRINCIPIOS SOLID UTILIZADOS**

# Patrones utilizados

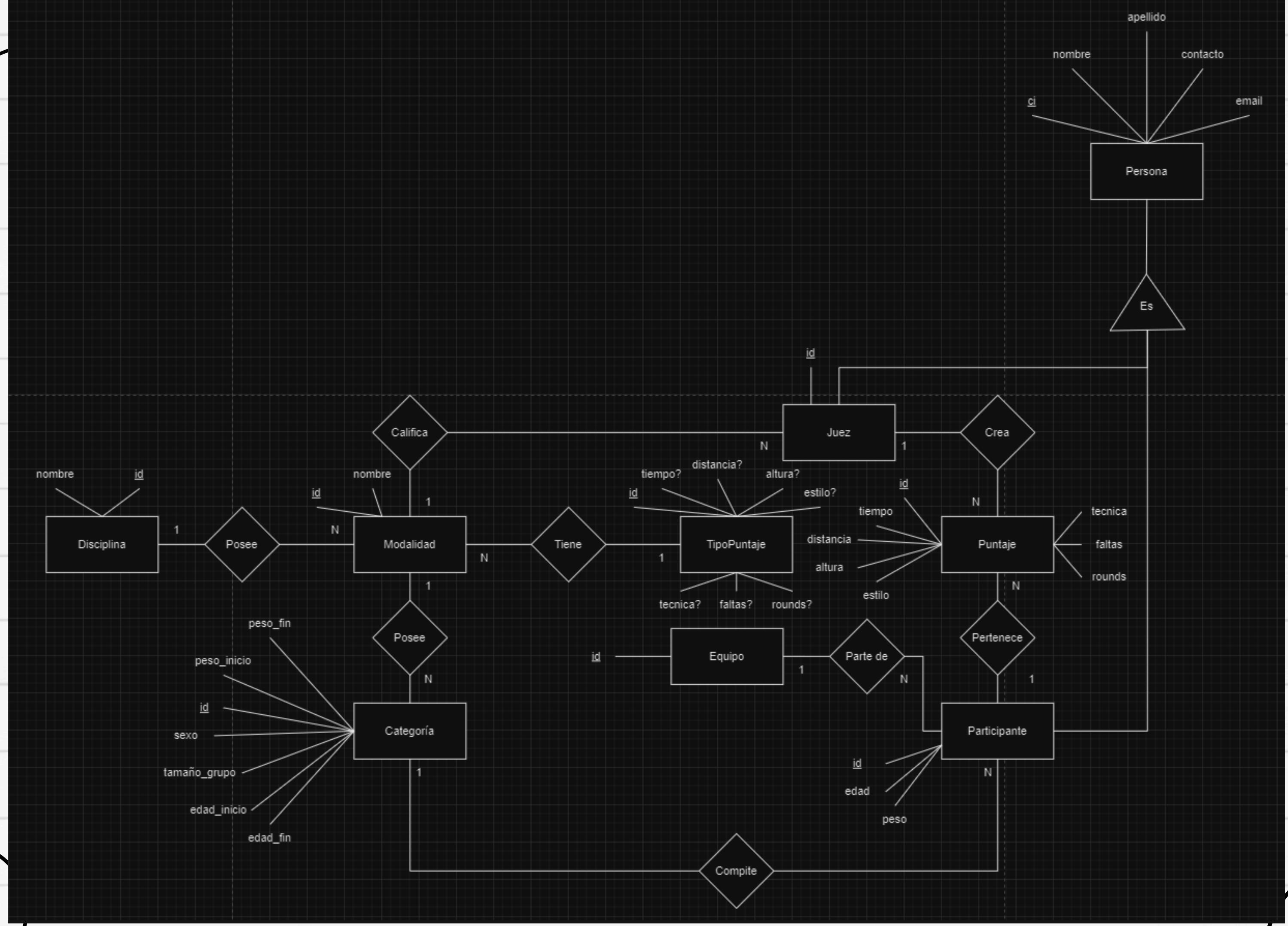
## SINGLETON

A partir de la clase “DatabaseConnection” se proporciona una única instancia para realizar la conexión a la base de datos. De esta manera se proporciona un punto global de acceso, el cual es compartido por todo el programa.

## REPOSITORY

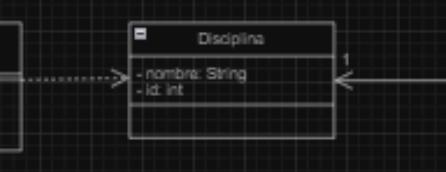
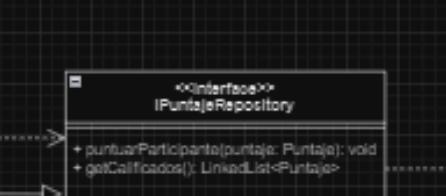
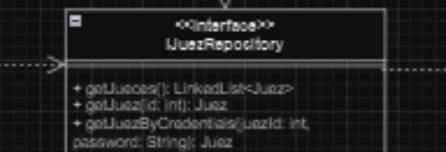
Las clases de tipo Repository de las mismas encapsulan la lógica de conexión, almacenamiento y obtención de datos de la base de datos emulando una colección de datos.

El cliente que usa estas clases no conoce la verdadera implementación que existe dentro de ellas.

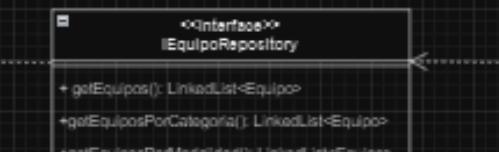
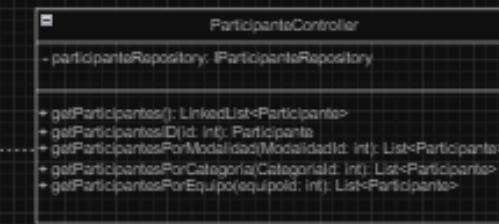
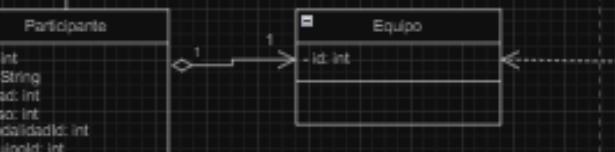
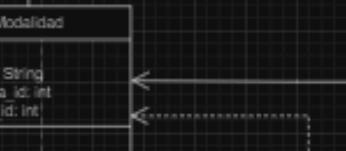
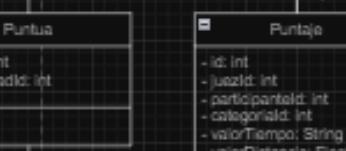
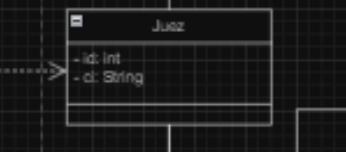




Todos los repositorios dependen de DataBaseConnection, por un tema de legibilidad se decidió no representarlo en el diagrama.



Los gets y sets de todas las clases son implícitos





# **DEMO POSTMAN**