

JPA

JPA → Java Persistence API

O que é?

É uma abstração (especificação) em cima do JDBC que torna possível a independência do SQL.

→ É utilizado anotações para mapear os elementos.

Principal pacote do JPA: `javax.persistence`

Principais componentes:

- Mapeamento Objeto Relacional (ORM – Object-Relational Mapping)
- Uma API (EntityManager)
- Uma linguagem de consulta (JPQL – Java Persistence Query Language)
- Entidades:
 - A classe da entidade representa uma tabela do banco de dados.
 - Um objeto da entidade representa uma linha na tabela

Principais regras:

- A entidade deve ser anotada com `@javax.persistence.Entity`.
- A anotação `@javax.persistence.Id` deve ser usada para criar uma chave primária simples.
- Possuir um construtor público.
- A entidade não pode ser um ENUM ou INTERFACE.

Implementação uma entidade:

- O nome da tabela herda o nome da classe
 - Você pode mudar o nome: `@Table(name="nome-tabela")`
- Os nomes das colunas são herdados dos atributos
 - `@Column(name="opcional")`

Mapeamento Objeto Relacional - ORM

- Serve como ponte entre o banco de dados e a POO
- Prós e Contras:

Prós

- É mais fácil consultar usando ORM do que MySQL.
- Oculta os processos complexos e apenas permite que você se preocupe com a implementação de alto nível.
- Não há necessidade de alterar o código se o banco de dados subjacente for alterado.

Contras

- Aprender um ORM do zero é demorado.
- Você não pode mergulhar diretamente em um ORM sem aprender MySQL. Mesmo que o ORM ajude a abstrair os detalhes complexos, é pertinente saber a consequência de cada comando nos bastidores.
- Você pode encontrar problemas de desempenho ao codificar consultas complexas usando um ORM.

- `@Table`: Por padrão, os nomes das tabelas são criadas com letra maiúscula, assim como o nome da classe. `@Table("opcional")`
- `@SecondaryTable`: Permite que os atributos de uma Entidade possa ser distribuído entre mais de uma tabela.

```
@SecondaryTables ({
    @SecondaryTable (name="cidade"),
    @SecondaryTable (name="pais")
})
```

- `@Id`: Define uma chave primária
- `@GeneratedValue`: Define a coluna que tem um valor gerado automaticamente
- `@Basic`: É tipo de mapeamento mais simples
- `@Column`: Define os atributos que serão colunas
- `@Temporal`: Usada para definir datas
- `@Transient`: Permite que um atributo não seja persistido
- Associações:
 - Unidirecional
 - Bidirecional
 - Com cardinalidade
- `@OneToOne`: um para um
- `@ManyToOne`: n para muitos
- `@ManyToMany`: n para n