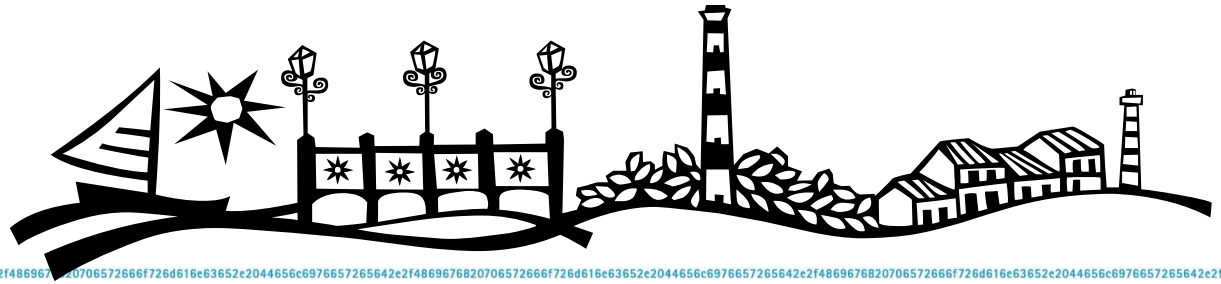
accenture

Strategy | Digital | Technology | Operations

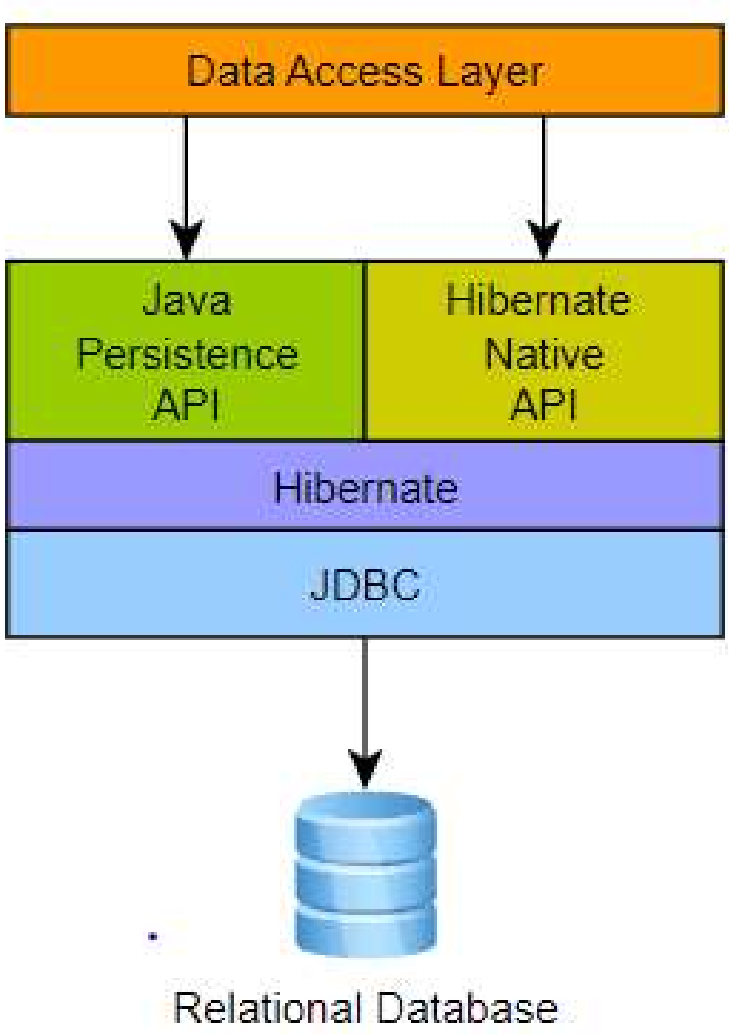
High performance. Delivered.

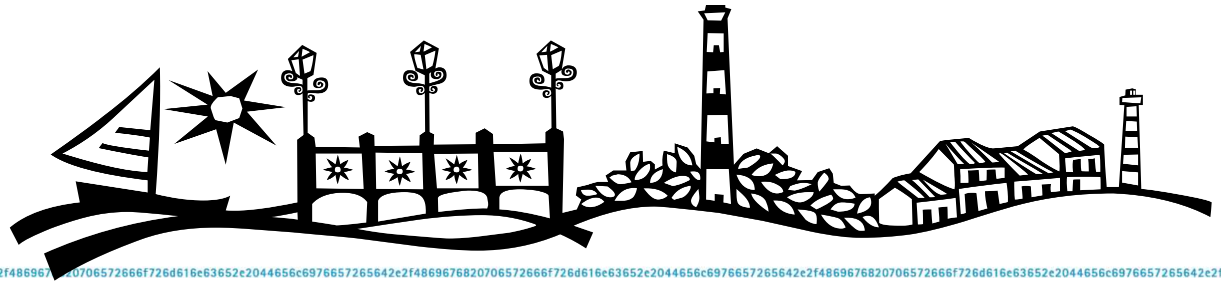
Application Delivery Fundamentals: SPRING BOOT

Module 8: ORM



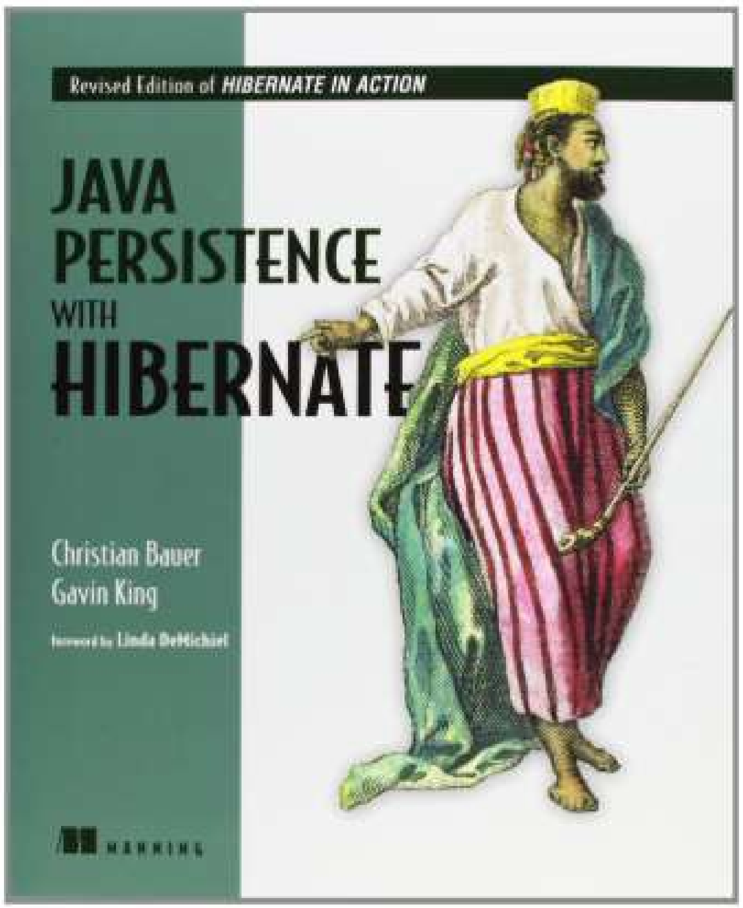
- **CARACTERISTICAS**:

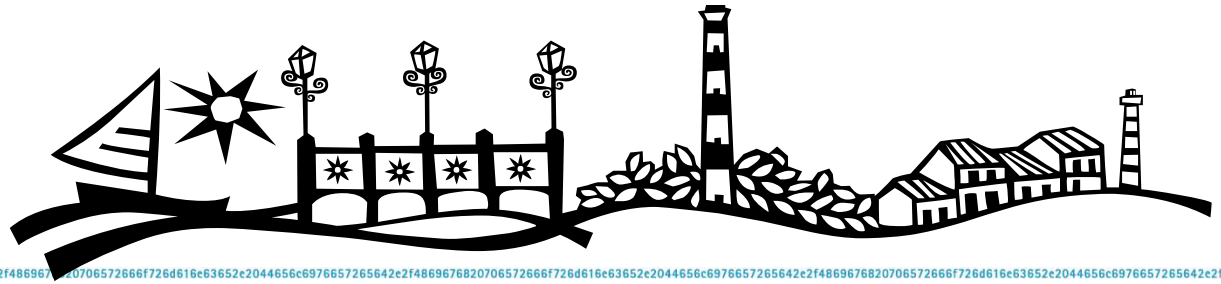




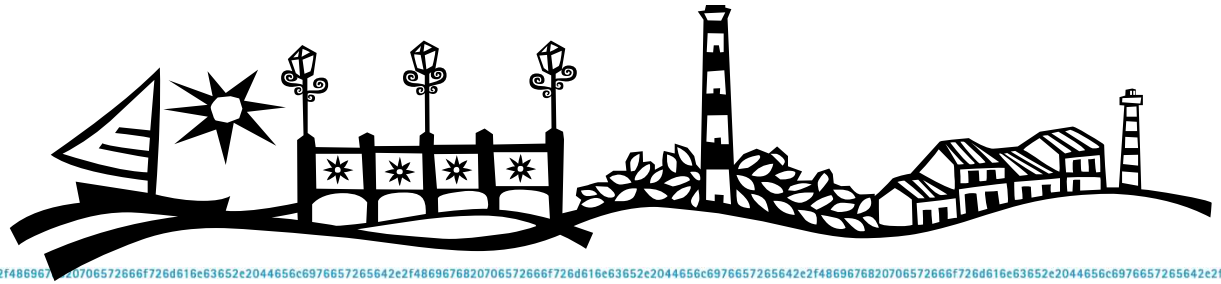
2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f48696768207

- **DICA DE LIVRO**:
- **Java Persistence with Hibernate: Revised Edition of Hibernate in Action** Paperback – November 24, 2006;



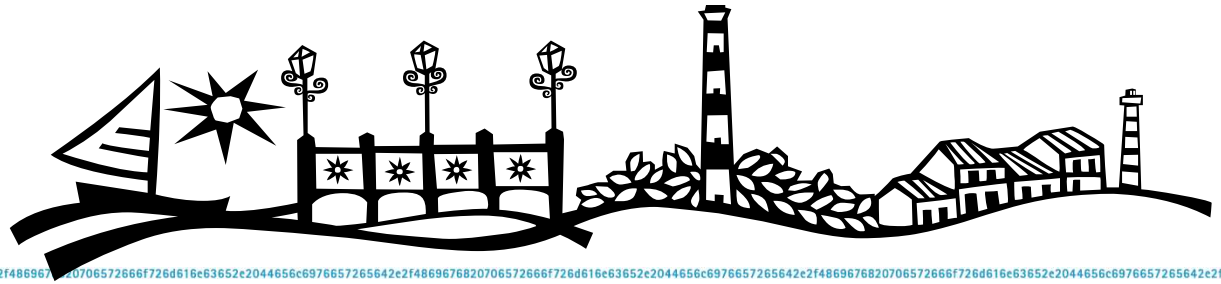


- **Object Relational Mapping - ORM**:
  - Como iremos mapear o mundo relacional com o mundo Orientado a objeto?
  - No mundo relacional os dados estão em forma de tabelas que contém linhas e colunas.
  - No mundo OO os dados estão em forma de objetos.



- **Object Relational Mapping - ORM - Possui**:
  - Uma API para montar o CRUD;
  - Uma query para especificar as classes e suas propriedades;
  - Um Local para especificação dos metadados;
  - Uma técnica para interagir com objetos relacionais para executar otimizações, recuperação de dados, etc..

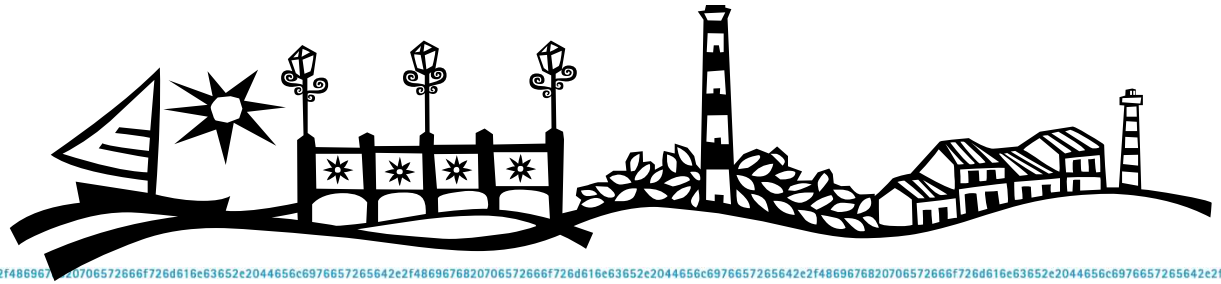
  - Proteje o desenvolvedor de SQL confusos;
  - O Desenvolvedor não precisa entender de BD relacionais;



2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207

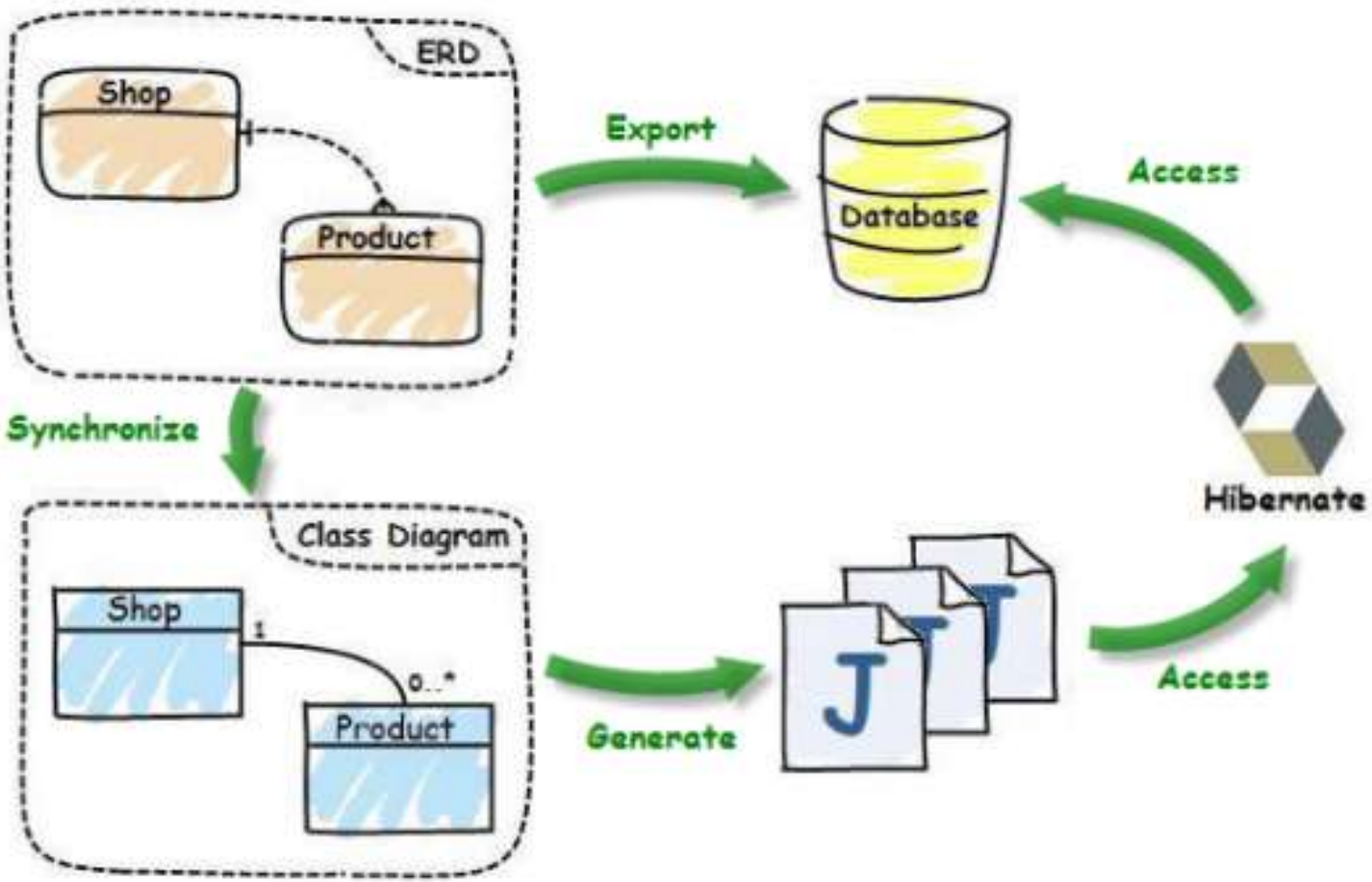
O que é Mapeamento Objeto-Relacional?

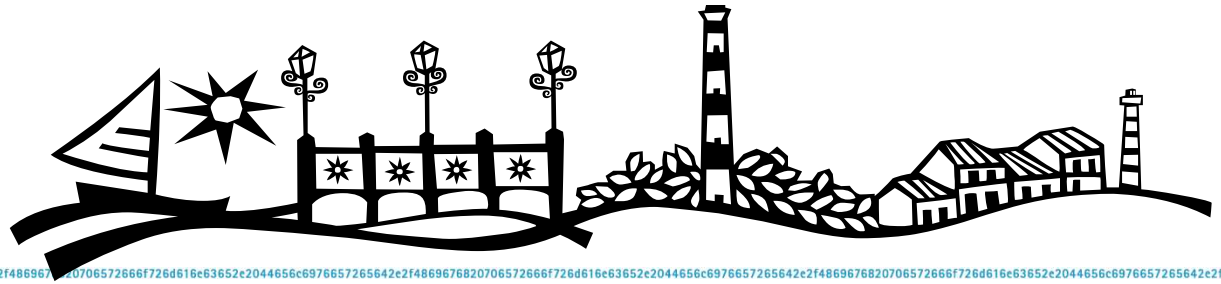
- ORM é uma técnica que consiste da conversão das classes da aplicação para tabelas do banco de dados e vice-versa.



2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207

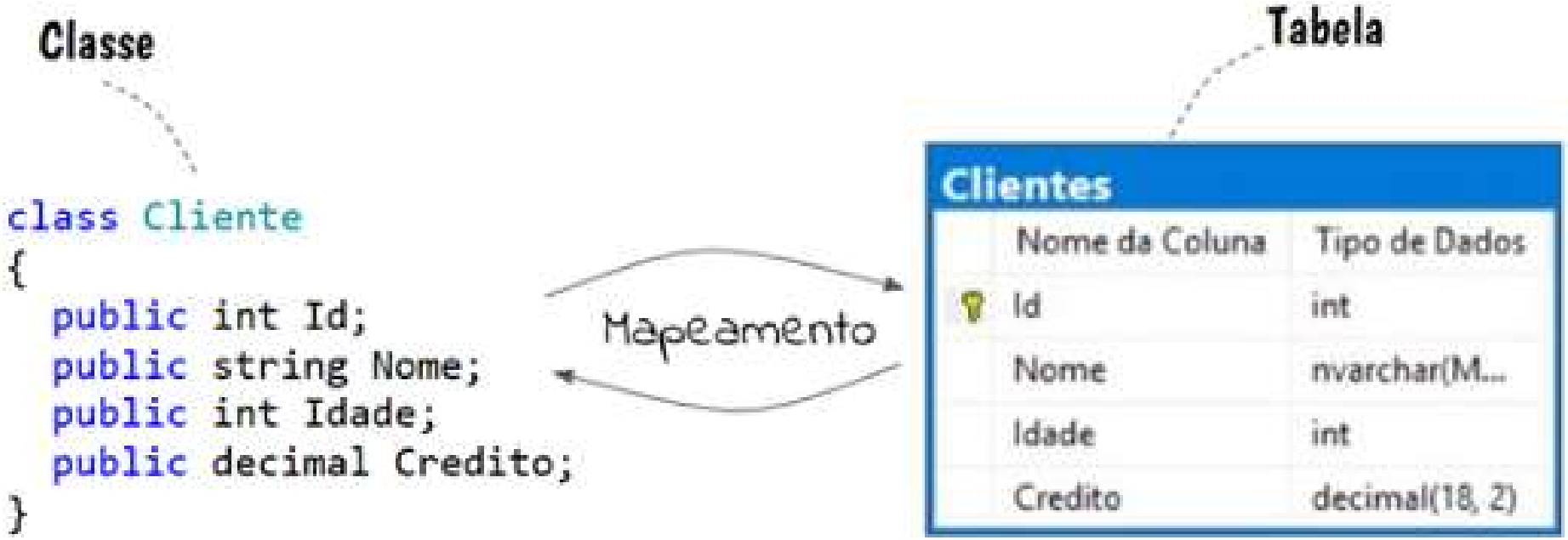
- **Object Relational Mapping - ORM**:

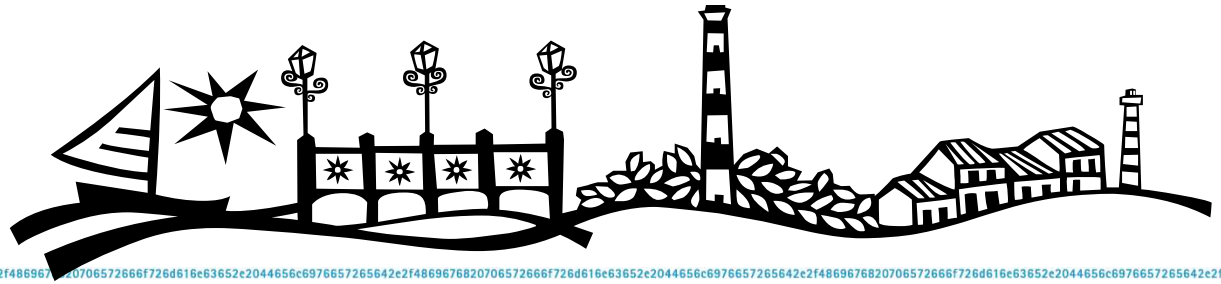




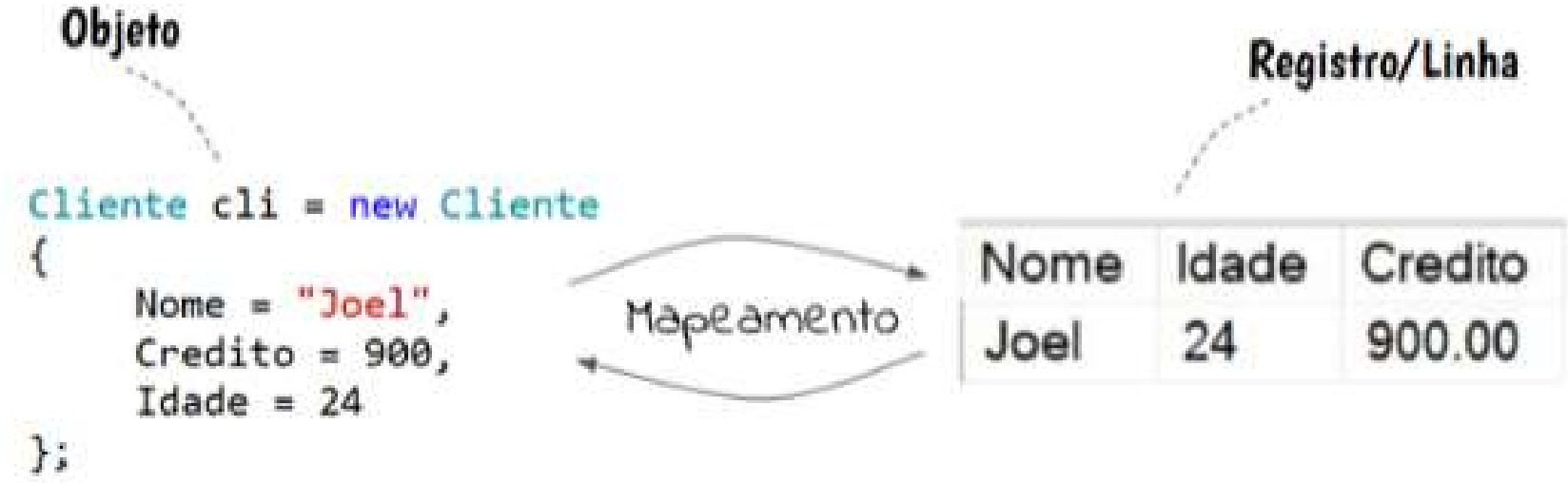
2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207

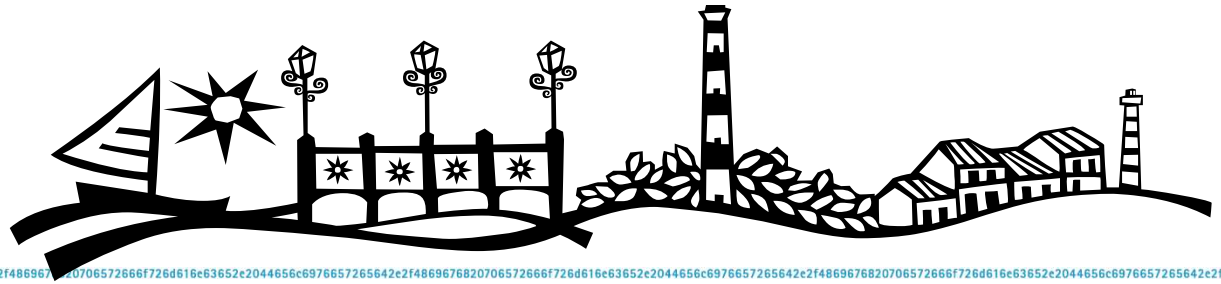
- **Object Relational Mapping - ORM**:





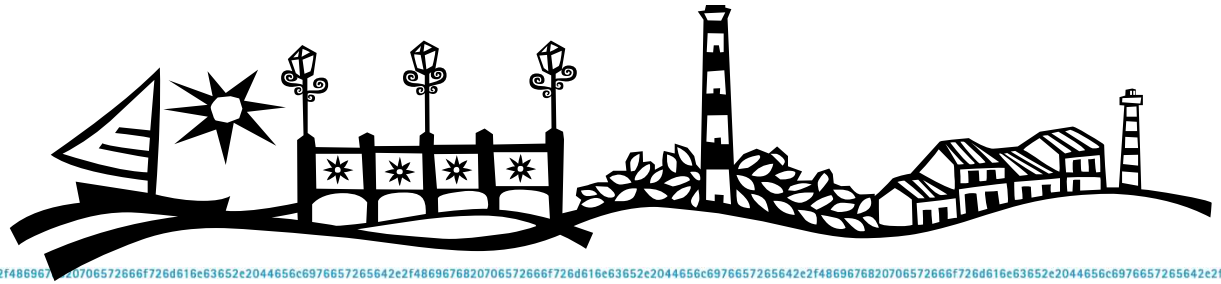
- **Object Relational Mapping - ORM**:
- Da mesma forma, também faz parte desse processo a conversão entre os objetos da aplicação e as linhas da tabela Ou seja, enquanto no código trabalhamos com classes e objetos, esses são gravados e recuperados do banco de dados na forma de registros/linhas.





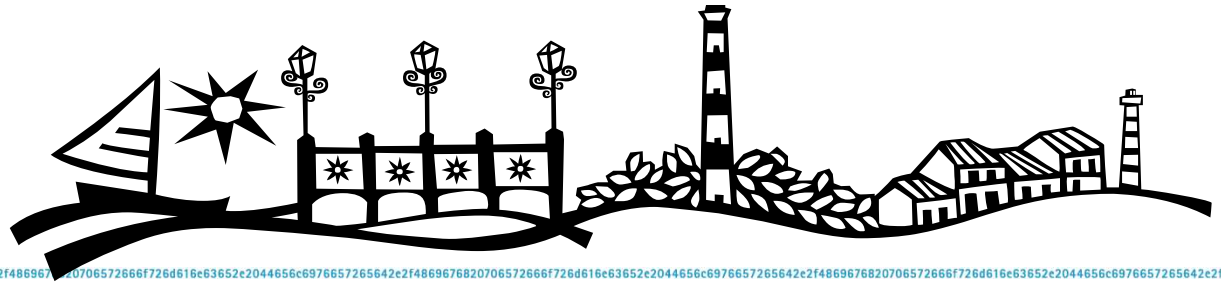
2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207

- **Associação**?
  - Entidades podem ser associadas à outras entidades;
  - Ou Collections;
  - **One-to-one, one-to-many, many-to-one, e many-to-many**;
  - Tecnicas de carregamento **Lazy** e **Eager**;



2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207

- **Annotations**
- https://hibernate.org/
  - @Entity
  - @Table
  - @Column (length = 100)
  - @Id @GeneratedValue(strategy = GenerationType.AUTO)
  - **@OneToOne(fetch =** FetchType.LAZY**, mappedBy = "**stock**", cascade =** CascadeType.ALL**)**
  - **@OneToMany**
  - **@ManyToOne**
  - **@ManyToMany**
  - **@Inheritance**
  - **@PreUpdate**
  - **@PrePersist**

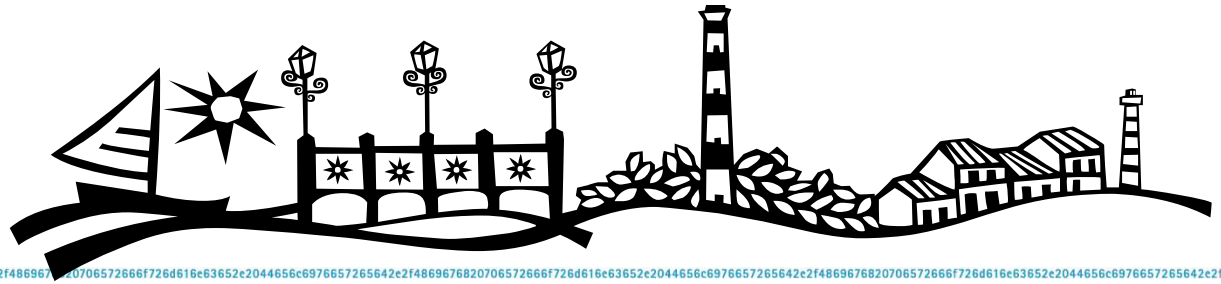


2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207

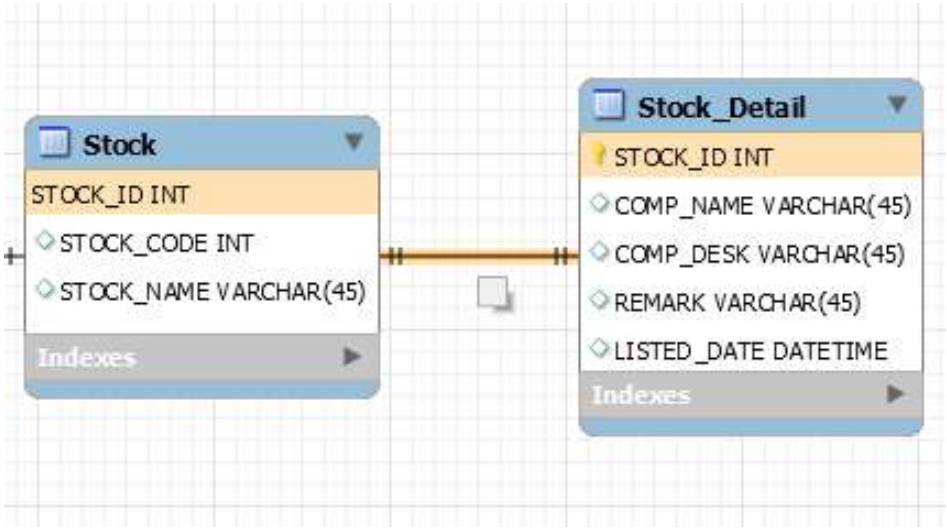
Estratégia	Descrição
GenerationType.AUTO	Valor padrão, deixa com o provedor de persistência a escolha da estratégia mais adequada de acordo com o banco de dados.
GenerationType.IDENTITY	Informamos ao provedor de persistência que os valores a serem atribuídos ao identificador único serão gerados pela coluna de auto incremento do banco de dados. Assim, um valor para o identificador é gerado para cada registro inserido no banco. Alguns bancos de dados podem não suportar essa opção.
GenerationType.SEQUENCE	Informamos ao provedor de persistência que os valores serão gerados a partir de uma sequence. Caso não seja especificado um nome para a sequence, será utilizada uma sequence padrão, a qual será global, para todas as entidades. Caso uma sequence seja especificada, o provedor passará a adotar essa sequence para criação das chaves primárias. Alguns bancos de dados podem não suportar essa opção.
GenerationType.TABLE	Com a opção TABLE é necessário criar uma tabela para gerenciar as chaves primárias. Por causa da sobrecarga de consultas necessárias para manter a tabela atualizada, essa opção é pouco recomendada.

2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f48696768207

Aplicação One To One



- **Annotations**
  - **@OneToOne(fetch =** FetchType.***LAZY*****, mappedBy = "**stock**", cascade =** CascadeType.ALL**)**

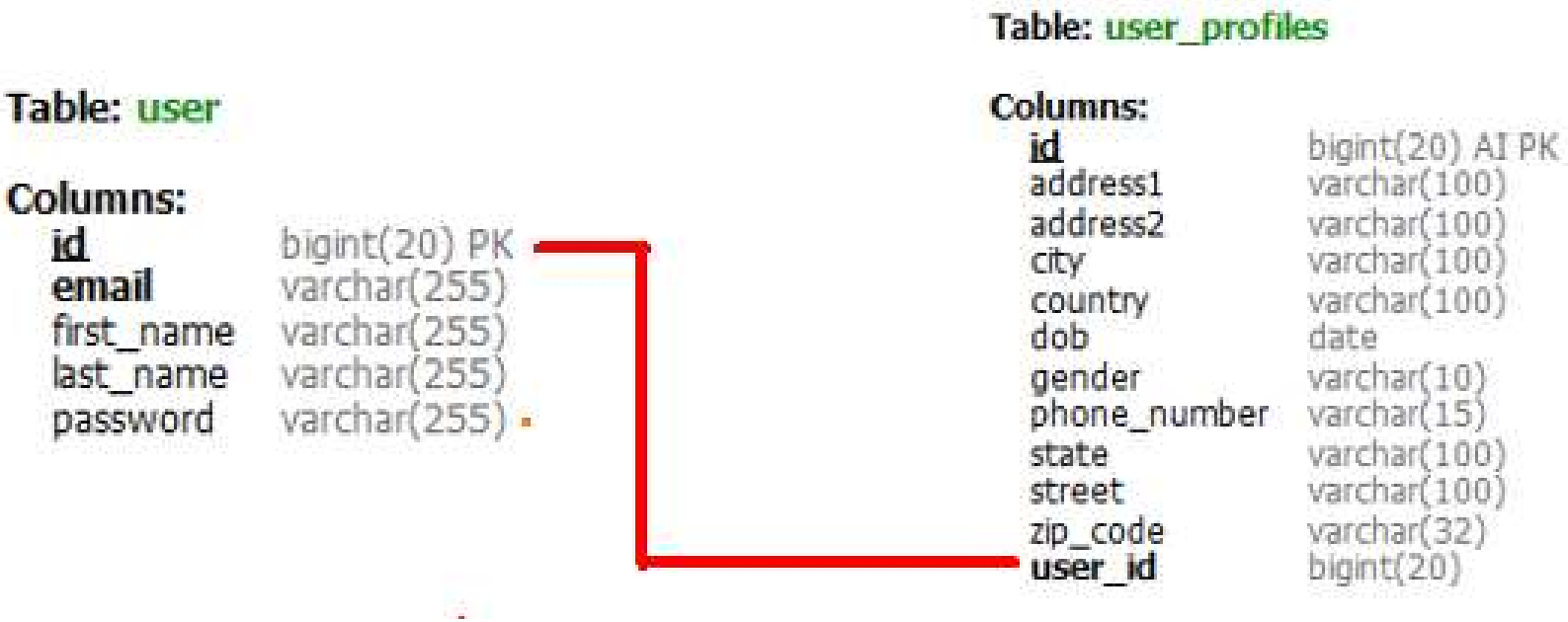


  - **@OneToOne(fetch** = FetchType.***LAZY***)
  - **@PrimaryKeyJoinColumn**

Spring BOOT JPA

2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207

Relacionamento user vs user_profile



Spring BOOT JPA

2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207

Mapeamento

```java
@Entity
@Table(name = "users")
public class User implements Serializable {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
```

Table: user

Columns:

id	bigint(20) PK
email	varchar(255)
first_name	varchar(255)
last_name	varchar(255)
password	varchar(255)

```java
    @OneToOne(fetch = FetchType.LAZY,
            cascade =  CascadeType.ALL,
            mappedBy = "user")
    private UserProfile userProfile;
```

Table: user_profiles

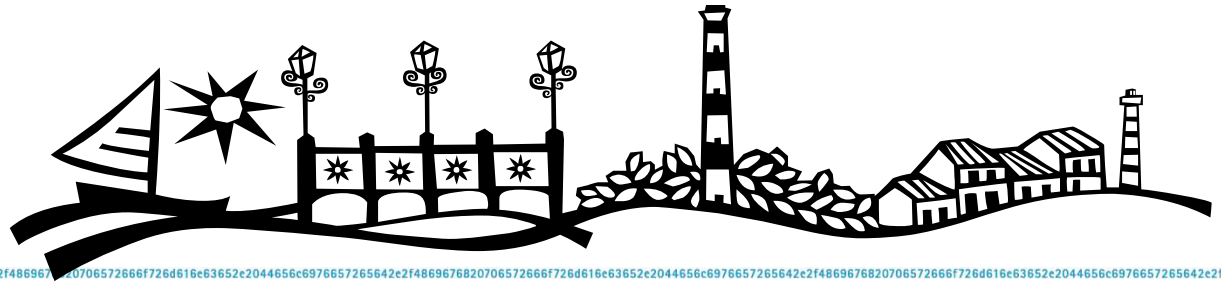
Columns:

id	bigint(20) AI PK
address1	varchar(100)
address2	varchar(100)
city	varchar(100)
country	varchar(100)
dob	date
gender	varchar(10)
phone_number	varchar(15)
state	varchar(100)
street	varchar(100)
zip_code	varchar(32)
user_id	bigint(20)

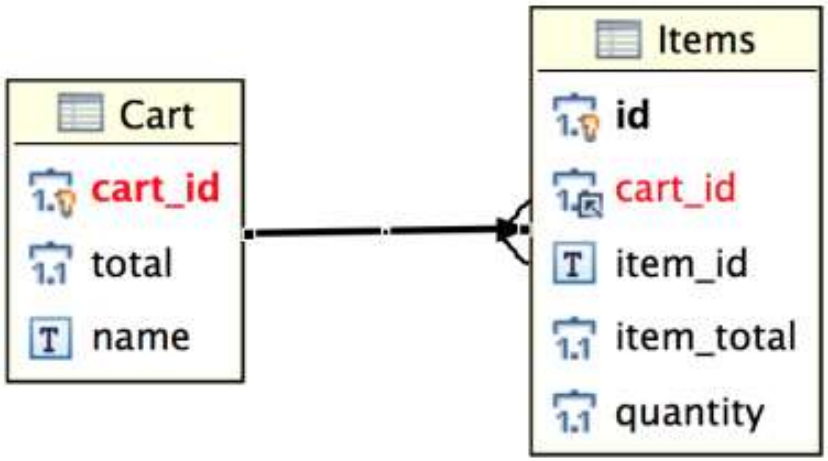
```java
    @OneToOne(fetch = FetchType.LAZY, optional = false)
    @JoinColumn(name = "user_id", nullable = false)
    private User user;
```

2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f48696768207

Aplicação One To Many



- **One-to-many**:
  - @**OneToMany**(mappedBy="cart")

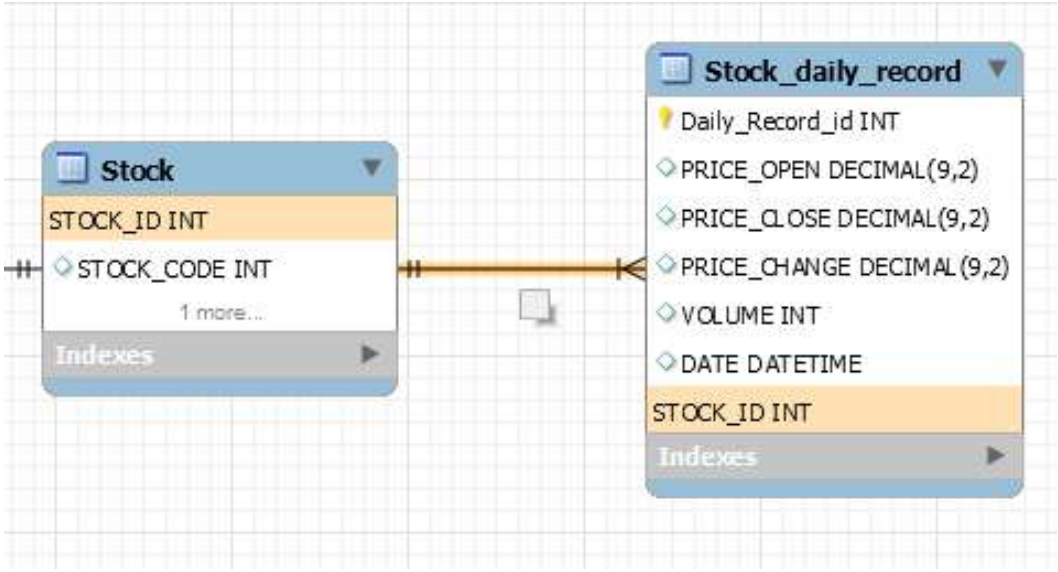


  - @**ManyToOne**
  - @**JoinColumn**(name="cart_id", nullable=false)

Spring BOOT JPA

2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f48696768207

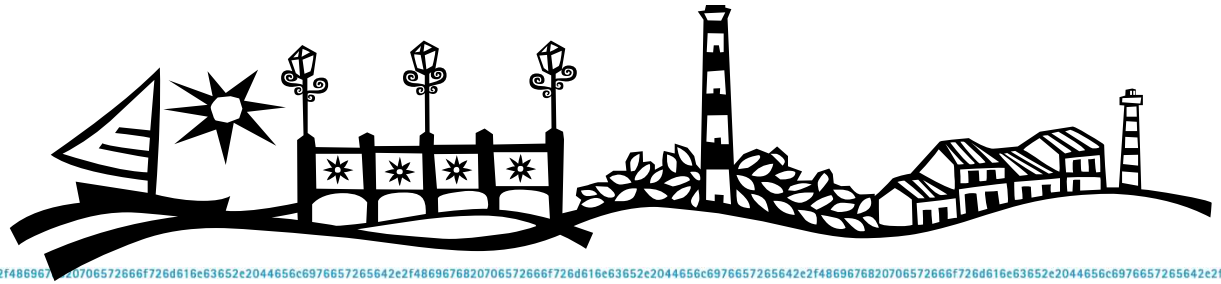
- **One-to-many**:
  - @**OneToMany**(fetch = FetchType.***LAZY,*** mappedBy **= *"stock")***



  - @**ManyToOne**(fetch = FetchType.***LAZY)***
  - @**JoinColumn**(name="STOCK_ID", nullable=false)

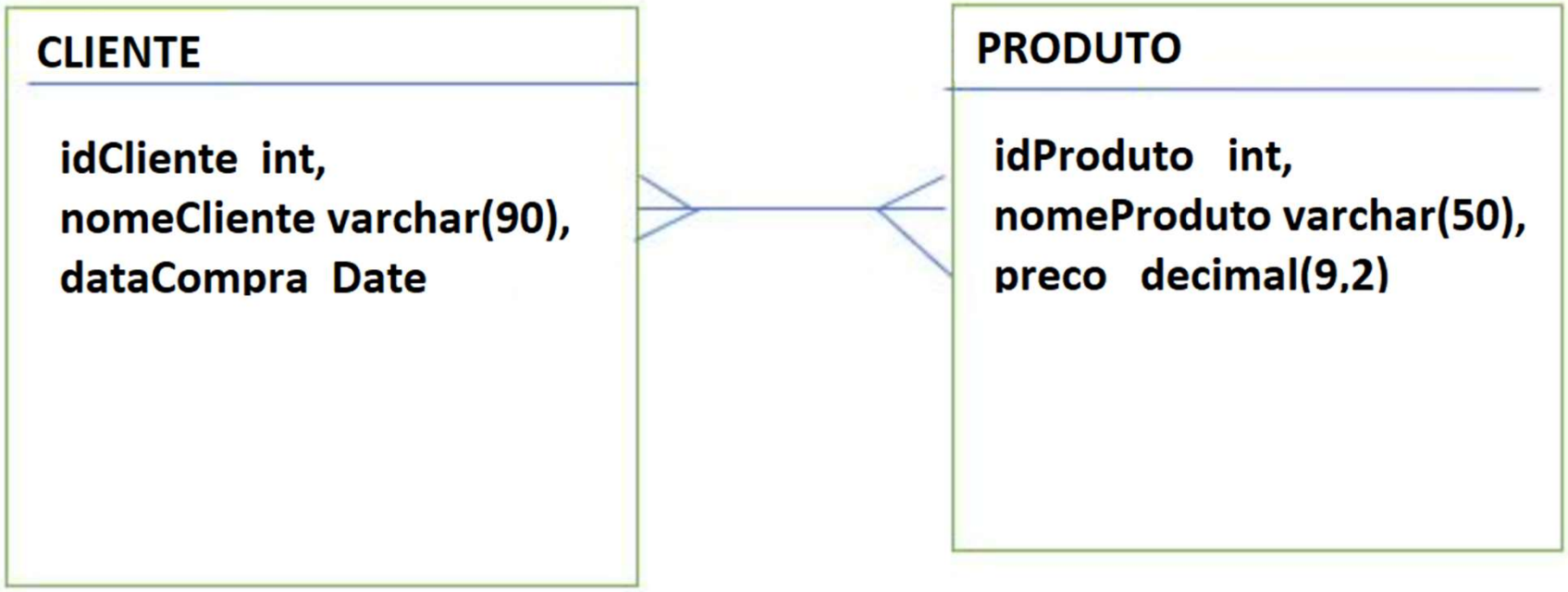
2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f48696768207

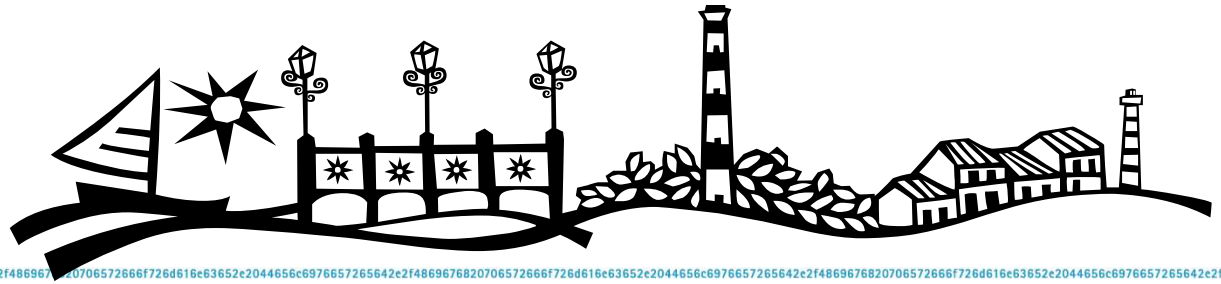
Aplicação
Many To Many



2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207

- **Many-to-many**:





2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207065726f726d616e63652e2044656c6976657265642e2f48696768207

Funcionario.java

```java
@Entity
@Table (name="funcionario")
public class Funcionario implements Serializable {
    @Id
    @Column(name="id")
    @GeneratedValue(strategy = GenerationType.AUTO, generator = "funcionario_seq_gen")
    @SequenceGenerator(name = "funcionario_seq_gen", sequenceName = "funcionario_id_seq")
    private Long id;
    private String nome;

    @ManyToMany
    @JoinTable(name = "depto_funcionario", joinColumns = (@JoinColumn(name = "id_funcionario")),
    inverseJoinColumns =(@JoinColumn(name = "id_depto")))
    private List<Depto> deptos;

    public List<Depto> getDeptos() {
        return deptos;
    }
```

Depto.java

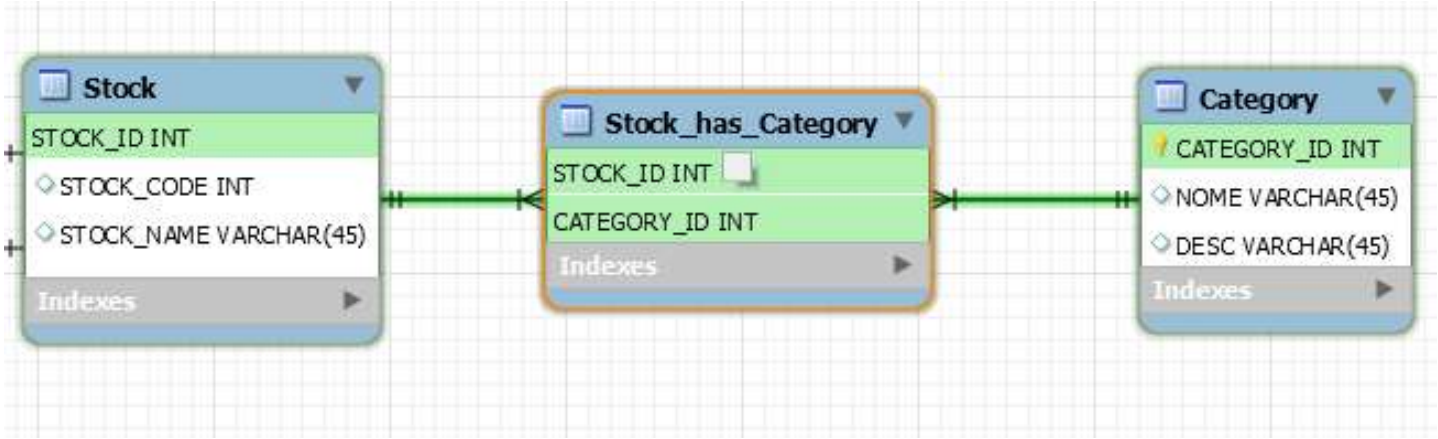
```java
@Entity
@Table (name="depto")
public class Depto implements Serializable {
    @Id
    @Column(name="id")
    @GeneratedValue(strategy = GenerationType.AUTO, generator = "depto_seq_gen"
    @SequenceGenerator(name = "depto_seq_gen", sequenceName = "depto_id_seq")
    private Long id;

    private String nome;

    @ManyToMany(mappedBy = "deptos")

    private List<Funcionario> funcionarios;

    public List<Funcionario> getFuncionarios() {
        return funcionarios;
    }
```

uso do mapped By

Spring BOOT JPA

2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207065726666f726d616e63652e2044656c6976657265642e2f48696768207

- **Annotations**
- **@ManyToMany**(fetch = FetchType.***LAZY, cascade = CascadeType.ALL***)
- **@JoinTable**(name = "stock_category", catalog = "mydb", joinColumns = **{**
- @JoinColumn(name = "STOCK_ID", nullable = **false,** updatable **= false) },**
- inverseJoinColumns = **{**
- @JoinColumn(name = "CATEGORY_ID", nullable = **false,** updatable **= false) })**



– **@ManyToMany**(fetch = FetchType.***LAZY, mappedBy = "categories")***

Dúvidas?