



MAPEAMENTO OBJETO RELACIONAL

INTRODUÇÃO

INTRODUÇÃO

Antes do JPA, o JDBC.

- Modelo complexo;
- Exigia muito SQL no código;
- Além do código JDBC.



INTRODUÇÃO

Surge o Hibernate...

- Framework objeto-relacional;
- Simplifica a interação entre aplicação e banco de dados;
- Ajuda a representar tabelas do banco de dados através de classes;
- Automatiza tarefas;
- Gera SQL em tempo de execução.



HIBERNATE



INTRODUÇÃO

JPA (Java Persistence API)

Especificação oficial que descreve o comportamento dos frameworks de persistência Java.



INTRODUÇÃO

A persistência de dados ocorre para manter e recuperar objetos de um sistema de armazenamento não volátil.

A JPA fornece um mecanismo para gerenciá-la.



INTRODUÇÃO

Para persistir dados com JPA é necessário escolher uma implementação.

A implementação dá vida para a especificação, é o código que podemos chamar de framework.



IMPLEMENTAÇÃO

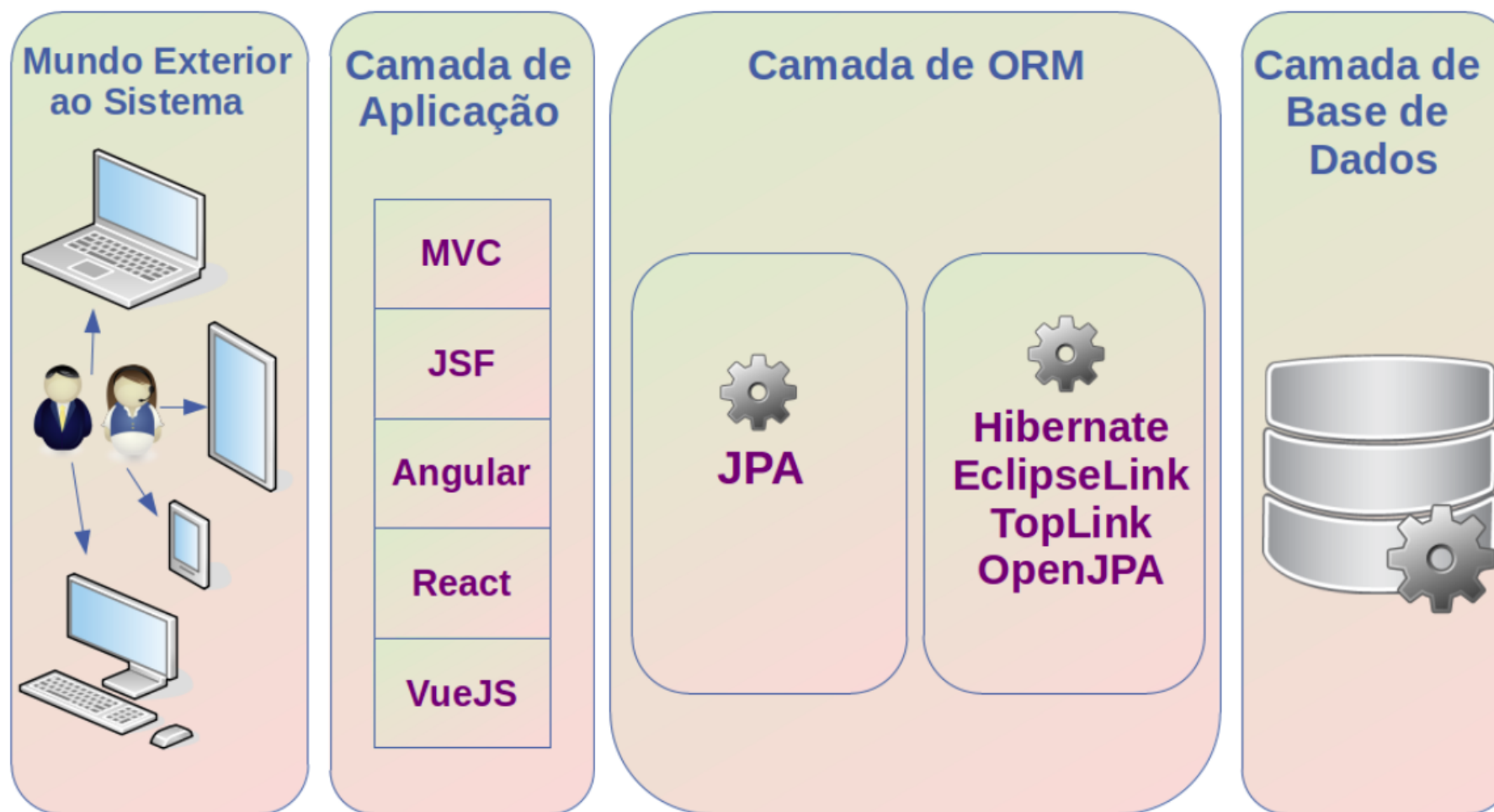
Qualquer pessoa pode escrever sua própria implementação JPA;

Dentre as mais famosas estão:

1. OpenJPA, da Apache;
2. Hibernate, da Red Hat;
3. EclipseLink, da Eclipse Foundation.

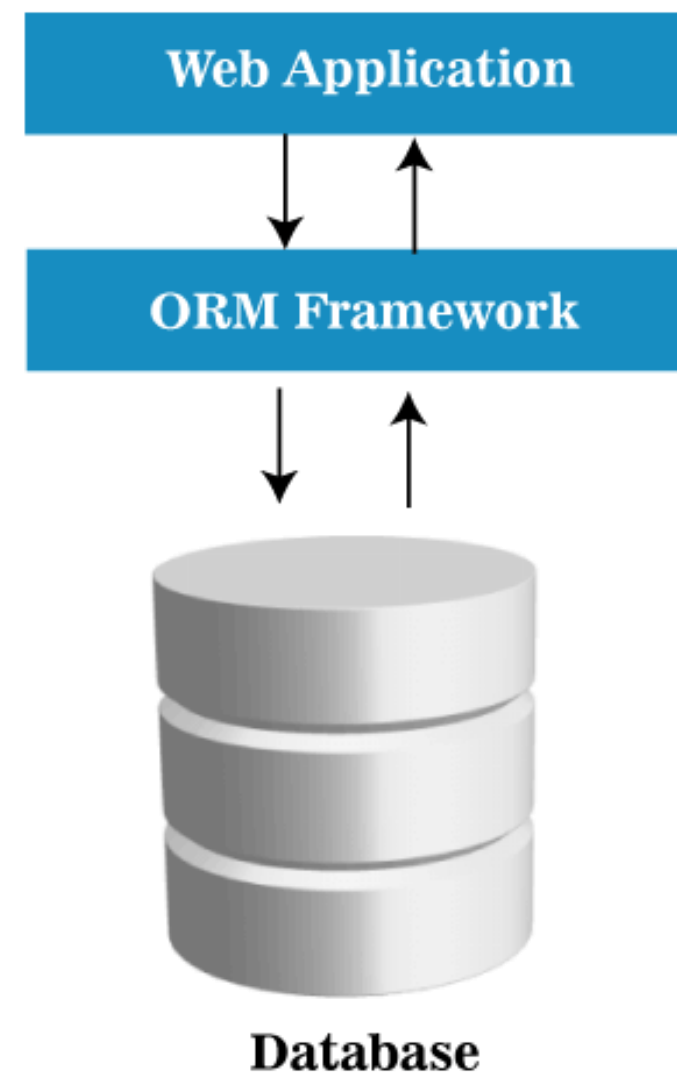


IMPLEMENTAÇÃO

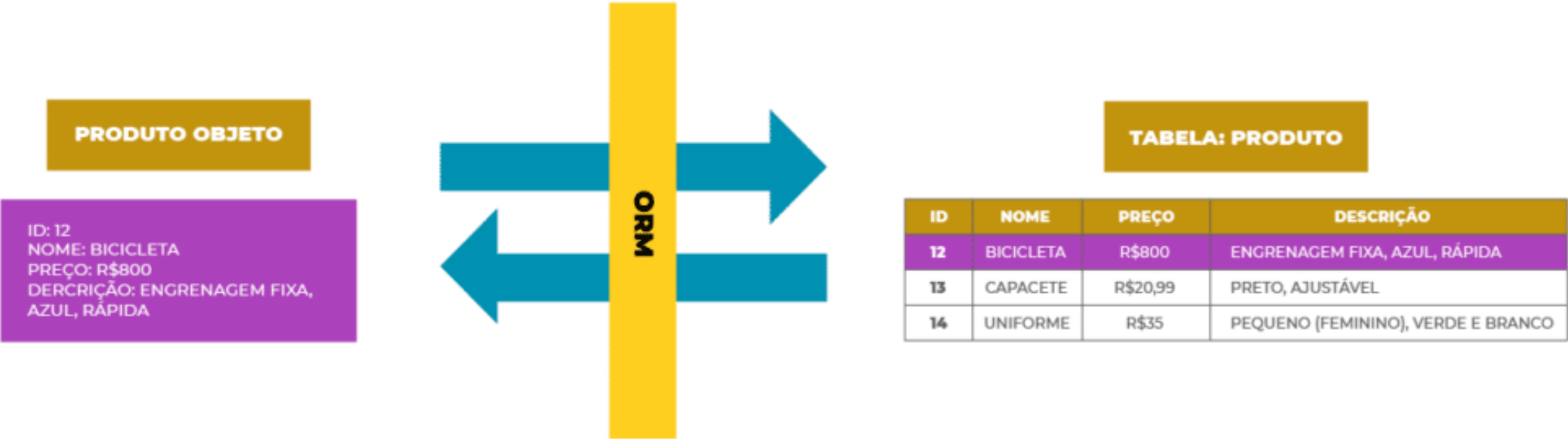


ORM

Mapeamento Objeto Relacional (ORM) é a representação da tabela de um banco de dados relacional através da classe Java.



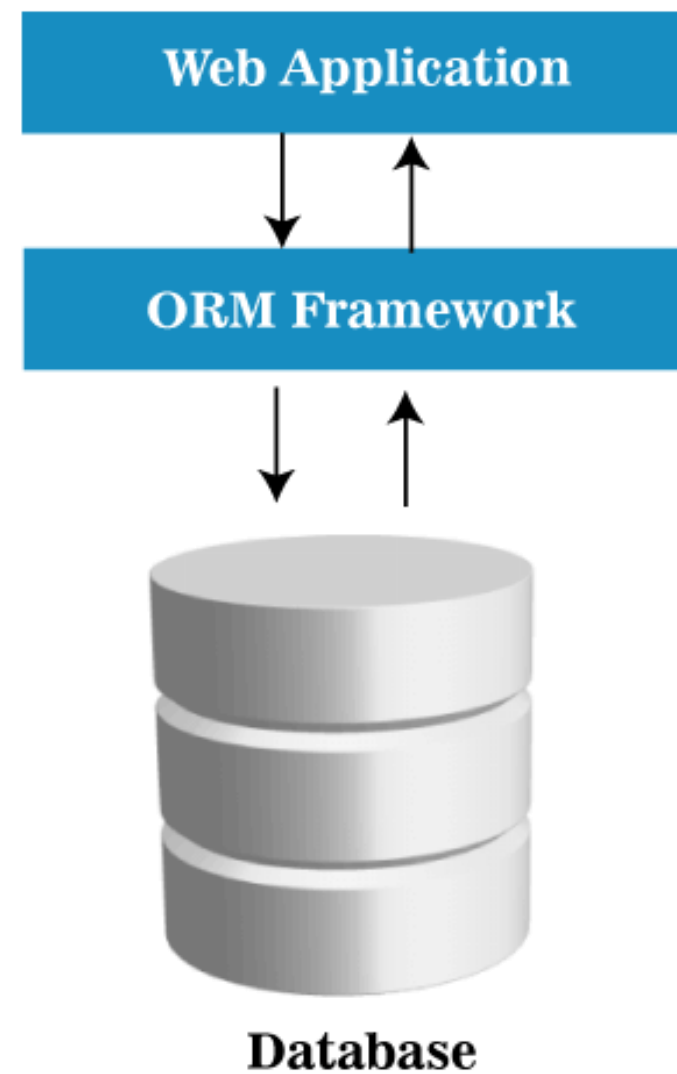
ORM



ORM

Observada a equivalência, para utilizar o ORM é preciso fazer uso de anotações.

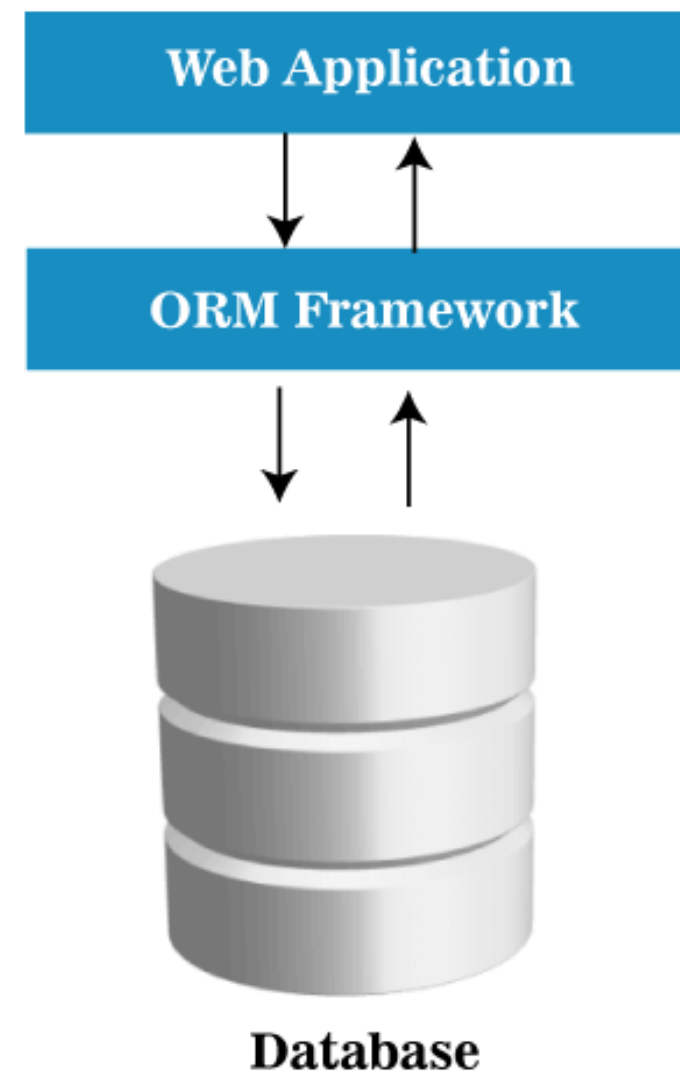
Elas serão responsáveis por adicionar metadados às classes e permitirão aos frameworks ORM entrarem em ação.



ORM

Exemplos mais usados:

- @Entity: indica que a classe também é uma entidade;
- @Table: indica a tabela correspondente;
- @Id: indica o atributo correspondente à PK;
- @Column: indica detalhes da coluna.



ORM

```
1  @Entity
2  @Table(name = "cliente")
3  public class Cliente {
4
5      @Id
6      @Column(name = "id")
7      private Integer id;
8
9      @Column(name = "nome")
10     private String nome;
11
12     // getters e setters omitidos
13
14 }
```



PQ JPA?

- Produtividade;
- SQL pronto, em tempo de execução;
- Simplificação do código fonte;
- Alternância entre SGDB;
- Manutenibilidade do código.



UNIVAS

Obrigado!

**Professor Gustavo Dias
luizdias@univas.edu.br**