

*Desenvolvimento Web com Java utilizando padrões  
de projeto e bootstrap*

---

Semana da  
engenharia

---





---

# Padrões de Projeto

---

- ❖ **Padrão de Projeto** (*design pattern*): É uma solução geral para um problema conhecido.
- ❖ É uma descrição ou modelo de como resolver um problema que pode ser utilizado em muitas situações diferentes.
- ❖ Não é o código fonte da aplicação!



---

# Padrão DAO

---

- ❖ Padrão de projeto DAO (*Data Access Object*): Objeto de acesso a dados.
- ❖ É um padrão de projeto que permite separar regras de negócio das regras de acesso a dados.
- ❖ Tem como objetivo construir uma camada de acesso aos dados de forma a isolar a camada do modelo da camada de persistência.
- ❖ <https://sergiotaborda.wordpress.com/desenvolvimento-de-software/java/patterns/dao/>



# Padrão DAO

Classes do  
Modelo

Pessoa

Livro

Carro

DAO

VIEW

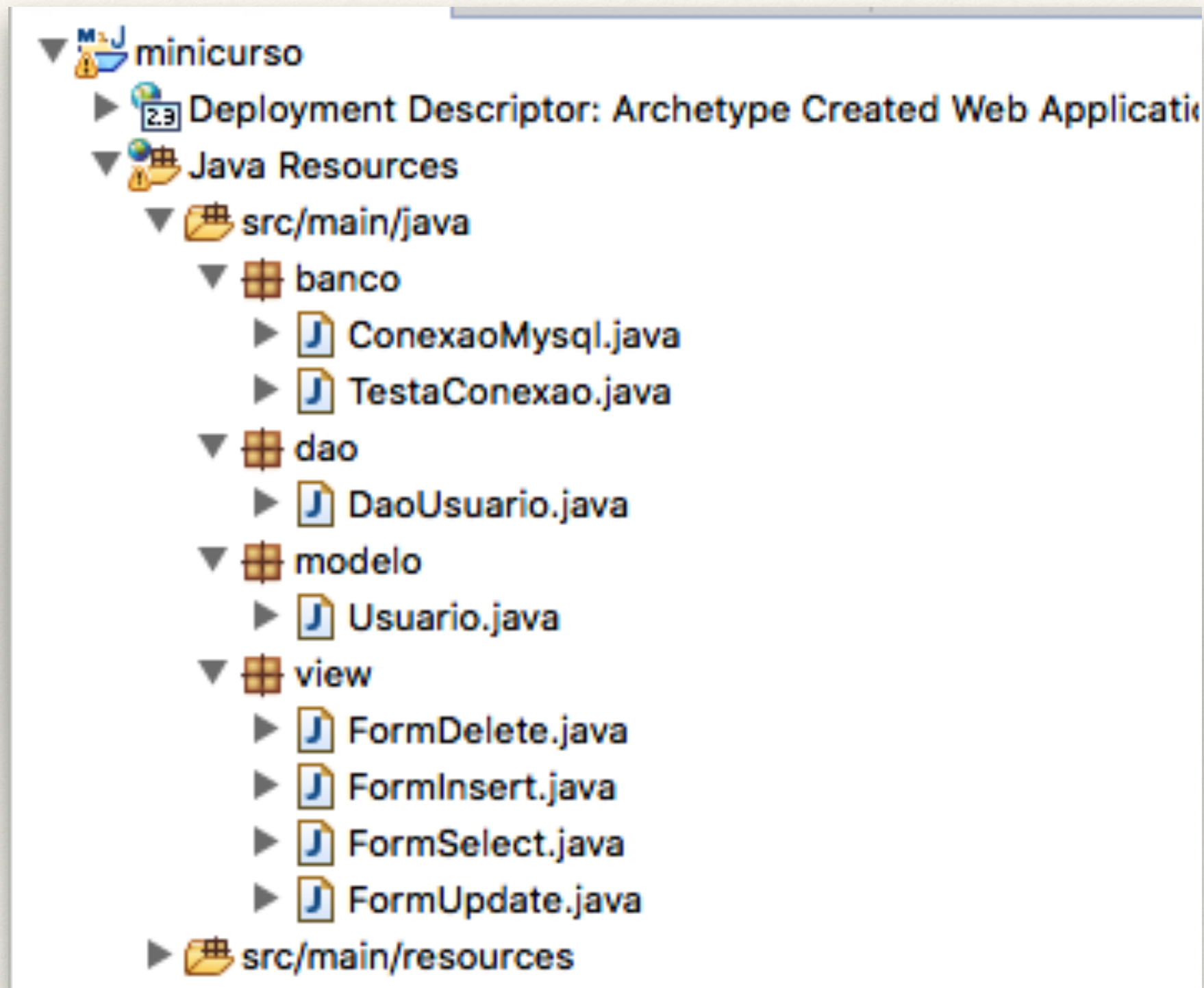
Persistência  
(Dados)

Oracle

XML



# Estrutura do projeto





---

# Implementação das classes

---

- ❖ **Banco:** Gerencia as conexões com a camada de persistência.
- ❖ **Modelo:** Atributos do objeto e métodos *getters* e *setters*.
- ❖ **Dao (Controle):** Operações CRUD (Create, Read, Update e Delete) de acordo com a camada de persistência de dados (interfaces).



---

# Classe Modelo

---

```
public class Usuario {  
    private int id;  
    private String nome;  
    private String email;  
    private int idade;
```

```
    public int getId() {  
        return id;  
    }  
  
    public void setId(int id) {  
        this.id = id;  
    }  
  
    public String getNome() {  
        return nome;  
    }  
  
    public void setNome(String nome) {  
        this.nome = nome;  
    }  
}
```



---

# Classe DAO

---

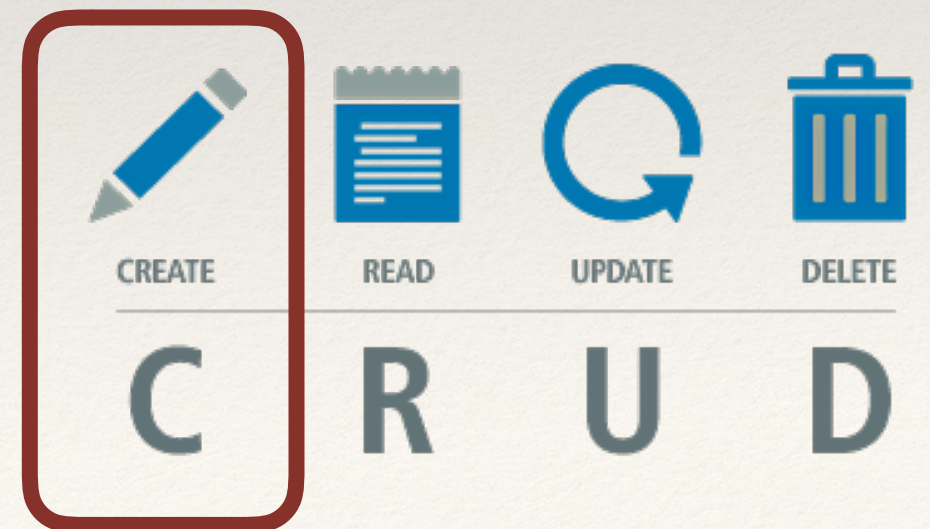
```
public class DaoUsuario {  
    private Connection con;  
    private ConexaoMysql conexao;  
  
    public DaoUsuario() {  
        this.conexao = ConexaoMysql.getInstancia();  
        this.con = conexao.conecta();  
    }  
}
```



# Classe DAO

Método para Inserir um registro no banco de dados.

```
public void insert(Usuario user) {  
    String sql = "insert into tb_usuarios "  
                + "(nome, email, idade) values (?, ?, ?)";  
    try {  
        PreparedStatement stmt = this.con.prepareStatement(sql);  
  
        stmt.setString(1, user.getNome());  
        stmt.setString(2, user.getEmail());  
        stmt.setInt(3, user.getIdade());  
  
        stmt.execute();  
        stmt.close();  
    } catch (SQLException e) {  
        throw new RuntimeException(e);  
    }  
}
```

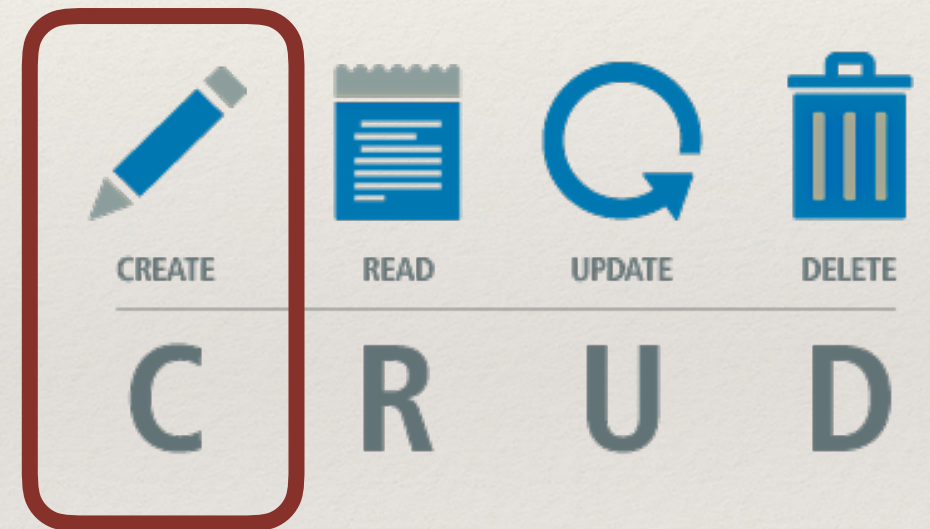




# Classe View

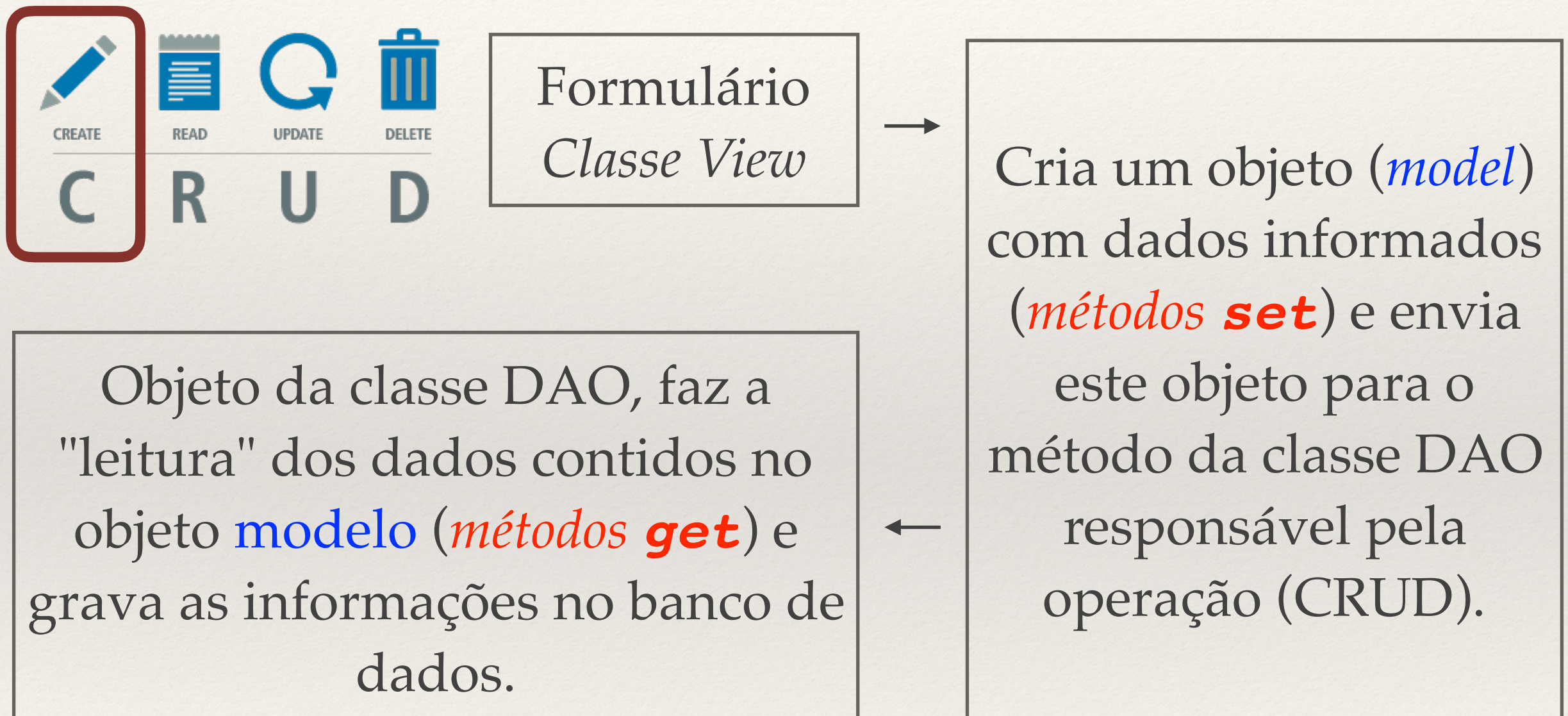
Teste de inclusão de um registro no banco de dados.

```
public class FormInsert {  
  
    public static void main(String[] args) {  
        Usuario user = new Usuario();  
        user.setNome("Tulio Calixto");  
        user.setEmail("tulio@e-mail.com");  
        user.setIdade(36);  
        user.gravarUsuario();  
  
        user.setNome("Maria Clara");  
        user.setEmail("maria@e-mail.com");  
        user.setIdade(25);  
        user.gravarUsuario();  
  
        System.out.println("Registros inseridos com sucesso.");  
    }  
}
```





# Resumo do processo





# Classe DAO



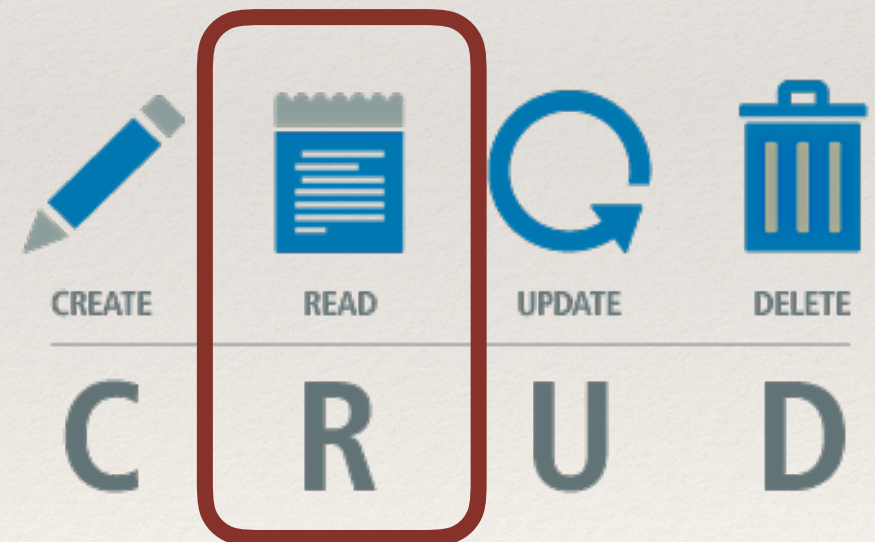
```
public List<Usuario> select(){  
    try {  
        List<Usuario> users = new ArrayList<Usuario>();  
        String sql = "select * from tb_usuarios";  
        PreparedStatement stmt = this.con.prepareStatement(sql);  
        ResultSet rs = stmt.executeQuery();
```

ID_LIVRO	ASSUNTO	AUTOR	EDITORA	ISBN	PUBLICO
1	teste	teste	teste	teste	teste
4	Desenvolvimento de software	Deitel	Makron	123456	Estudantes
5	Desenvolvimento WEB	Mauricio	Pearson	000000	Graduacao
6	C++	Deitel	Makron	99999	Pos-graduacao



# Classe DAO

```
while(rs.next()) {  
    Usuario u = new Usuario();  
    u.setId(rs.getInt("id_usuarios"));  
    u.setNome(rs.getString("nome"));  
    u.setEmail(rs.getString("email"));  
    u.setIdade(rs.getInt("idade"));  
  
    users.add(u);  
}  
rs.close();  
stmt.close();  
  
return users;  
} catch (SQLException e) {  
    throw new RuntimeException(e);  
}
```





# Classe View

```
public static void main(String[] args) {
    Usuario u = new Usuario();
    List<Usuario> users = null;
    users = u.select();

    int id, idade;
    String nome, email;
    String info = "Relação de usuários:";

    for(Usuario userTMP : users) {
        id = userTMP.getId();
        nome = userTMP.getNome();
        email = userTMP.getEmail();
        idade = userTMP.getIdade();

        info += "\n\nCódigo: " + id;
        info += "\nNome: " + nome;
        info += "\nE-mail: " + email;
        info += "\nIdade: " + idade;
    }
    System.out.println(info);
}
```





# Classe View

Relação de usuários:

Código: 1

Nome: Tulio Calixto

E-mail: tulio@e-mail.com

Idade: 36

Código: 2

Nome: Maria Clara

E-mail: maria@e-mail.com

Idade: 25

Código: 3

Nome: Joao Paulo

E-mail: jpe-mail.com

Idade: 19





# Classe DAO

```
public void update(Usuario user) {  
    String sql = "update tb_usuarios set "  
        + "nome = ?, email = ?, idade = ? "  
        + "where id_usuarios = ?";  
    try {  
        PreparedStatement stmt = this.con.prepareStatement(sql);  
  
        stmt.setString(1, user.getNome());  
        stmt.setString(2, user.getEmail());  
        stmt.setInt(3, user.getIdade());  
        stmt.setInt(4, user.getId());  
  
        stmt.execute();  
        stmt.close();  
    } catch (SQLException e) {  
        throw new RuntimeException(e);  
    }  
}
```





# Classe View



```
public static void main(String[] args) {  
    Usuario user = new Usuario();  
    user.setId(1);  
    user.setNome("Usuario Alterado");  
    user.setEmail("alteracao@e-mail.com");  
    user.setIdade(10);  
    user.update();  
  
    System.out.println("Registro alterado com sucesso.");  
}
```



# Classe DAO



```
public void delete(int id) {  
    String sql = "delete from tb_usuarios where id_usuarios = " + id;  
    try {  
        PreparedStatement stmt = this.con.prepareStatement(sql);  
        stmt.execute();  
        stmt.close();  
    } catch (SQLException e) {  
        throw new RuntimeException(e);  
    }  
}
```



# Classe View



```
public static void main(String[] args) {  
    Usuario u = new Usuario();  
    u.delete(2);  
  
    System.out.println("Registro excluído com sucesso.");  
}
```