

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARAÍBA
Campus João Pessoa

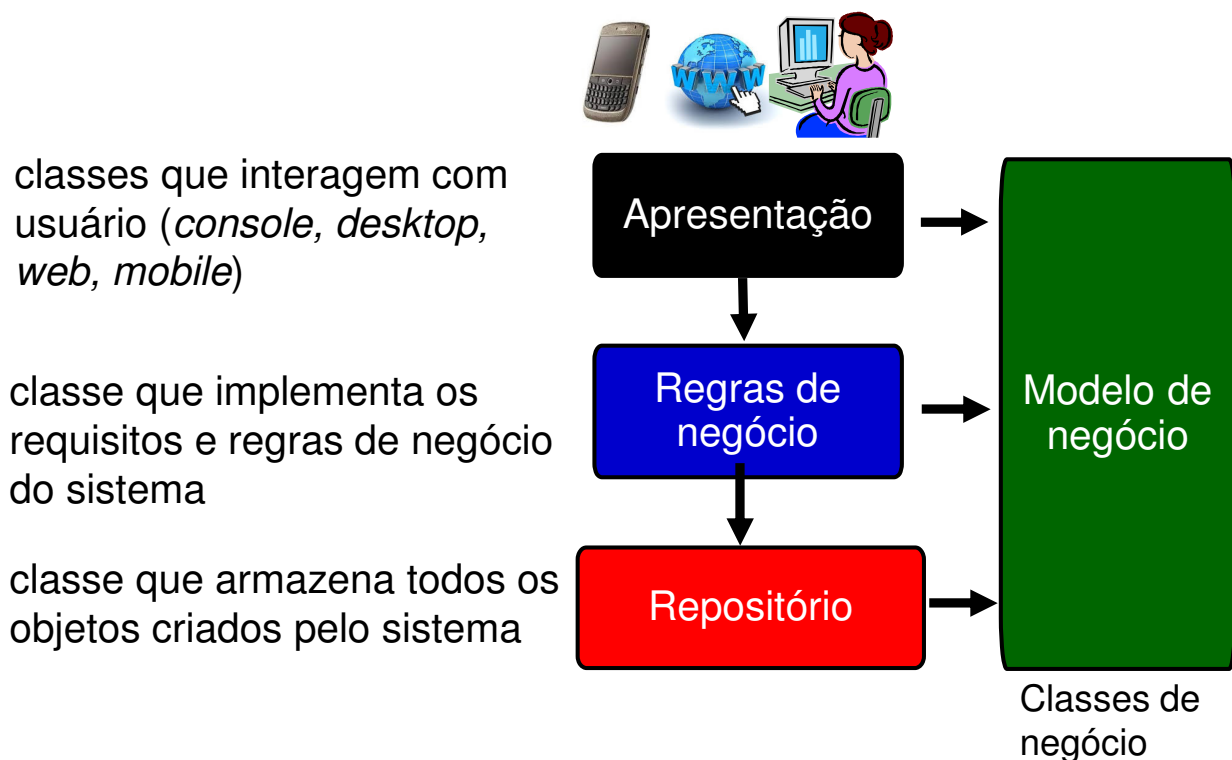
Programação Orientada a Objetos

Fausto Maranhão Ayres

11

Desenvolvimento em Camadas

Camadas do sistema



Desenvolvimento em Camadas

■ Vantagem:

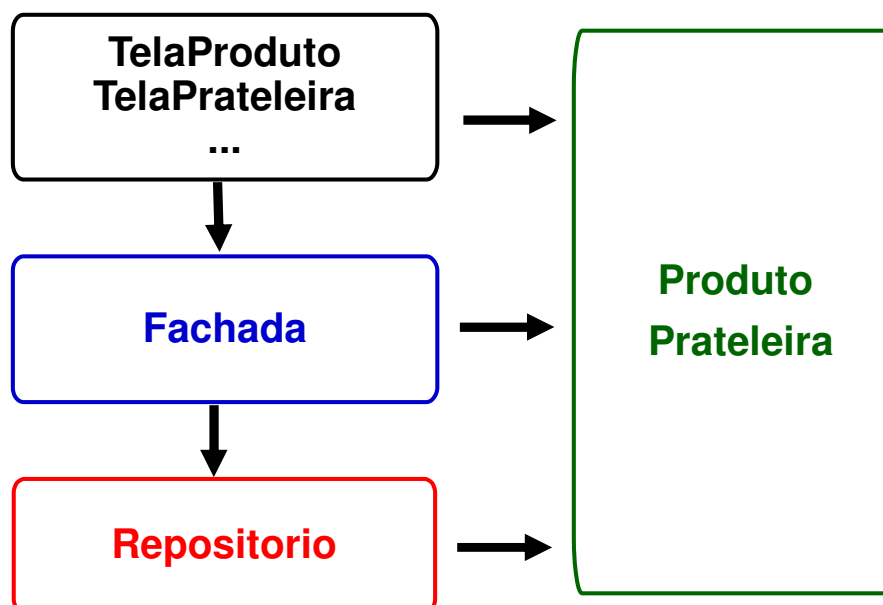
- Facilita o **teste e manutenção** (é mais fácil localizar a classe que tem problema)
- Facilita o **trabalho em equipe** (cada um trabalha numa camada)
- Facilita a **portabilidade** para web e mobile (troca a tela sem trocar a fachada)
- Permite adicionar **outras camadas** com pouca interferência nas existentes (p.ex camada de acesso a banco de dados)

fausto.ayres@ifpb.edu.br

3

Exemplo: Sistema Almoxarifado

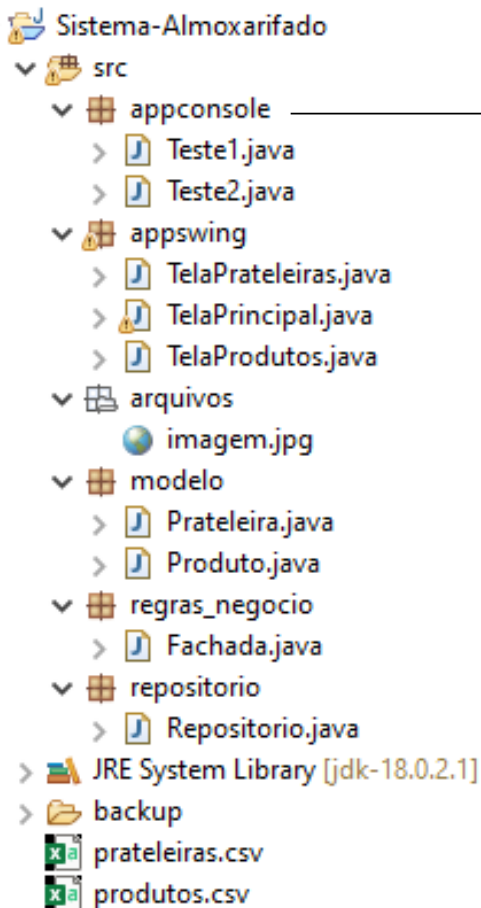
■ Classes



fausto.ayres@ifpb.edu.br

4

Camadas (*packages*) do sistema



Cada camada é representada por um ou mais *packages*

Para criar um *package*
File / New / package

5

Aplicação console (classe Teste1)

Chama os métodos da classe Fachada

```
Fachada.criarProduto("arroz", 3.0);
Fachada.criarProduto("feijao", 5.0);
Fachada.criarProduto("leite", 2.0);
Fachada.criarProduto("carne", 30.0);
Fachada.criarProduto("oleo", 10.0);
Fachada.criarPrateleira(200);
Fachada.criarPrateleira(300);
Fachada.criarPrateleira(400);
Fachada.inserirProdutoPrateleira(1, "arroz");
Fachada.inserirProdutoPrateleira(1, "feijao");
Fachada.inserirProdutoPrateleira(2, "leite");
Fachada.inserirProdutoPrateleira(3, "carne");
Fachada.removeProdutoPrateleira("carne");
Fachada.apagarProduto("arroz");
ArrayList<Produto> lista1 = Fachada.listarProdutos();
ArrayList<Prateleira> lista2 = Fachada.listarPrateleiras();
```

Classe Fachada

A classe utilitária **Fachada** implementa todas os requisitos e regras de negócio do sistema. Acessa o repositório de objetos

```
public class Fachada {  
    private static Repositorio repositorio = new Repositorio();  
    private Fachada() {}  
  
    public static ArrayList<Produto> listarProdutos() {  
    public static ArrayList<Prateleira> listarPrateleiras() {  
    public static void criarProduto(String nome, double preco) throws Exc  
    public static void criarPrateleira(double tam) throws Exception{  
    public static void inserirProdutoPrateleira(int id, String nome) thro  
    public static void removerProdutoPrateleira(String nome) throws Except  
    public static void apagarProduto(String nome) throws Exception {  
    public static void apagarPrateleira(int id) throws Exception {
```

Métodos
estáticos

fausto.ayres@ifpb.edu.br

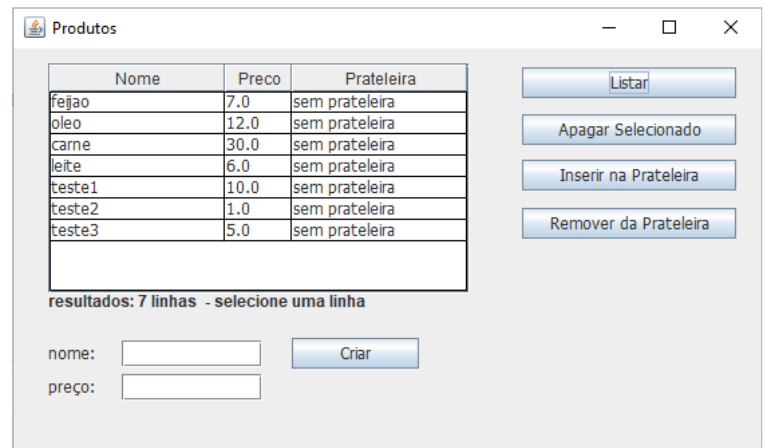
7

Classe Repositorio

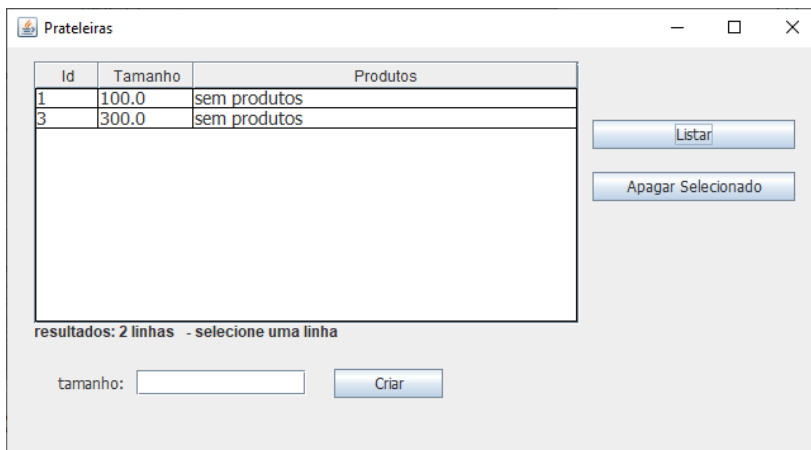
Reúne na memória todos os objetos criados pela Fachada

```
public class Repositorio {  
    private ArrayList<Produto> produtos = new ArrayList<>();  
    private ArrayList<Prateleira> prateleiras = new ArrayList<>();  
  
    public Repositorio() {  
    public void adicionar(Produto p) {  
    public void remover(Produto p) {  
    public Produto localizarProduto(String nome) {  
    public void adicionar(Prateleira p) {  
    public void remover(Prateleira p) {  
    public Prateleira localizarPrateleira(int id) {  
    public ArrayList<Produto> getProdutos() {  
    public ArrayList<Prateleira> getPrateleiras() {  
    public int getTotalProdutos() {  
    public int getTotalPrateleiras() {  
    public void apagarProdutos() {  
    public void apagarPrateleiras() {  
    public int gerarId() {  
  
    public void carregarObjetos() {  
    public void salvarObjetos() {
```

8



TelaProduto

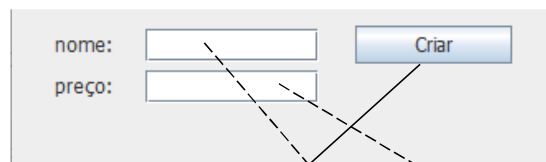


TelaPrateleira

9

Requisito: criar produto

TelaProduto



Requisito

```

public static void criarProduto(String nome, double preco){
    Produto p = repositorio.localizarProduto(nome);
    if (p!=null)
        throw new Exception("produto ja cadastrado");
    p = new Produto(nome, preco);
    repositorio.adicionar(p);
    return p;
}

```

regra de negócio

Fachada

Repositório

```

public class Repositorio{...
    public void adicionar(Produto p){
        produtos.add(p);
    }
}
...

```

```

public class Produto{
    ...
}

```

Requisito: apagar produto

| Nome | Preco | Prateleira |
|--------|-------|----------------|
| feijao | 7.0 | sem prateleira |
| oleo | 12.0 | sem prateleira |
| carne | 30.0 | sem prateleira |
| leite | 6.0 | sem prateleira |
| teste1 | 10.0 | sem prateleira |
| teste2 | 1.0 | sem prateleira |
| teste3 | 5.0 | sem prateleira |

TelaProduto

Requisito

Fachada

```
public static void apagarProduto(nome) {  
    Produto p = repositorio.localizarProduto(nome);  
    if (p==null)  
        throw new Exception("produto inexistente");  
    repositorio.remover(p);  
    ...  
}
```

Regra de negócio

Repositório

```
public class Repositorio{  
    public void remover(Produto p){  
        produtos.remove(p);  
    }  
    ...  
}
```

fausto.ayres@ifpb.edu.br

11

Requisito: listar produtos

| Nome | Preco | Prateleira |
|--------|-------|----------------|
| feijao | 7.0 | sem prateleira |
| oleo | 12.0 | sem prateleira |
| carne | 30.0 | sem prateleira |
| leite | 6.0 | sem prateleira |
| teste1 | 10.0 | sem prateleira |
| teste2 | 1.0 | sem prateleira |
| teste3 | 5.0 | sem prateleira |

TelaProduto

Requisito

Fachada

```
public static ArrayList<Produto> listarProdutos() {  
    return repositorio.getProdutos();  
}
```

Repositório

```
public class Repositorio{...  
    public ArrayList<Produto> getProdutos() {  
        return produtos;  
    }  
}
```

fausto.ayres@ifpb.edu.br

12