

Tarefa Avaliativa - Projeto com Testes

Cálculos de Complexidade Ciclomática Simplificada (CCS)

CCS - Estoque

```
public boolean inserir(Produto novoProduto){
    if(novoProduto.getNome().isEmpty() || novoProduto.getValor() <= 0)
        return false;
    listaProdutos.add(novoProduto);
    return true;
}
```

Inserir -> $1 + 1 + 1 = 3$

```
public boolean editar(Produto produtoAntigo, Produto novoProduto){
    if(novoProduto.getNome().isEmpty() || novoProduto.getValor() <= 0)
        return false;
    for (Produto item : listaProdutos) {
        if(produtoAntigo.equals(item)){
            this.listaProdutos.remove(item);
            this.listaProdutos.add(novoProduto);
            return true;
        }
    }
    return false;
}
```

Editar -> $1 + 1 + 1 + 1 + 1 = 5$

```
public boolean excluir(Produto produto){
    for (Produto item : listaProdutos) {
        if(produto.equals(item)){
            this.listaProdutos.remove(item);
            return true;
        }
    }
    return false;
}
```

Excluir -> $1 + 1 + 1 = 3$

Totalização Estoque:

Estoque:inserir=3 CCS

Estoque:editar=5 CCS

Estoque:excluir=3 CCS

Total Estoque = $3 + 5 + 3 = 11$

CCS – Pedido

```
public boolean inserirProduto(Produto novoProduto){  
    if(novoProduto.getNome().isEmpty() || novoProduto.getValor() <= 0)  
        return false;  
    produtosPedido.add(e: novoProduto);  
    atualizaValor();  
    return true;  
}
```

InserirProduto -> 1 + 1 = 2

```
public boolean excluirProduto(Produto produto){  
    for (Produto item : produtosPedido) {  
        if(produto.equals(obj: item)){  
            this.produtosPedido.remove(o: item);  
            atualizaValor();  
            return true;  
        }  
    }  
    return false;  
}
```

ExcluirProduto -> 1 + 1 + 1 = 3

```
public void atualizaValor(){  
    valorTotal = 0.00;  
    for (Produto item : produtosPedido) {  
        valorTotal = valorTotal + item.getValor();  
    }  
}
```

AtualizaValor -> 1 + 1 = 2

Totalização Pedido:

Pedido:inserirProduto = 2 CCS

Pedido:excluirProduto = 3 CCS

Pedido:atualizaValor = 2 CCS

Total Pedido = 2 + 3 + 2 = 7

Caso de teste com classes válidas e inválidas

Classe Pagamento

Partições/Classes

| Condições de Entrada | Classe válida | Classe inválida |
|------------------------------|---------------|-----------------|
| Valor informado é parcelável | Sim (1) | Não (2) |
| A parcela é maior que 20 | Sim (3) | Não (4) |

Casos de teste

| # | Entrada | Saída esperada | Classe coberta |
|---|-------------------|--|----------------|
| 1 | 200.00 no Crédito | Lista com parcelamentos possíveis | 1,3 |
| 2 | 25.00 no Crédito | Valor de parcela menor que R\$ 20,00. | 1,4 |
| 3 | 200.00 à vista | Uma parcela possível, pagamento realizado com sucesso. | 2,4 |
| 4 | 20.00 à vista | Uma parcela possível, pagamento realizado com sucesso. | 2,4 |