

## Consultas Estatísticas

### 1.1

#### Número total produtos

Ver tamanho de TreeSet<Produto> arvoreProdutos.

#### Total produtos diferentes comprados

Ver tamanho de TreeSet<FichaProdutoContabilidade> produtos.

#### Total produtos nao comprados

Percorrer TreeSet<FichaProdutoContabilidade> produtos e ver quantos nodos têm numComprasProduto = 0.

#### Número total clientes

Ver tamanho de TreeSet<Cliente> arvoreClientes.

#### Total clientes que fizeram compras

Percorrer TreeSet<FichaClienteCompras> arvoreClientes e ver quantos nodos têm numUnidadesCompradasCliente = 0.

#### Total de compras de valor zero

Consultar variável numeroComprasValorZero das compras.

#### Facturação total

Consultar variável totalFacturado da contabilidade.

### 1.2

#### Número compras por mês

Consultar matriz totalComprasPorMes[meses][tipo\_compra] das compras.

#### Facturação por mês

Consultar matriz totalFacturadoPorMes[meses][tipo\_compra] das compras

#### Número clientes diferentes em cada mês

Consultar matriz numeroClientesPorMes[meses][tipo\_compra]

#### Total de registos de compras inválidas

Ver tamanho de ArrayList<Compra> comprasInvalidas.

## Consultas Interactivas

### 1

#### Nº e lista produtos nao comprados

Percorrer TreeSet<FichaProdutoContabilidade> produtos e para cada nodo em que a variável totalCompras = 0, adicionar a uma lista e devolver a lista. Para saber o nº, basta o main calcular o tamanho da lista.

### 2

#### Nº e lista clientes sem compras

Percorrer TreeSet<FichaClienteCompras> arvoreClientes e para cada nodo em que a variável numComprasCliente = 0, adicionar a uma lista e devolver a lista. Para saber o nº, basta o main calcular o tamanho da lista.

### 3

#### Mês -> Nº de compras

Consultar matriz totalComprasPorMes[meses][tipo\_compra] na linha correspondente ao mês pretendido.

#### Mês -> Total clientes

Consultar matriz numeroClientesPorMes[meses][tipo\_compra] na linha correspondente ao mês pretendido.

4

#### **Codigo Cliente -> N° Compras**

“Transformar” o codigo de cliente numa ficha de cliente e procurar a ficha na árvore TreeSet<FichaClienteCompras> arvoreClientes. Depois basta consultar a variável numComprasCliente da ficha.

#### **Codigo Cliente -> N° Produtos distintos**

“Transformar” o codigo de cliente numa ficha de cliente e procurar a ficha na árvore TreeSet<FichaClienteCompras> arvoreClientes. Depois basta ver qual o tamanho de TreeSet<FichaProdutoDeClienteCompras> produtosCliente dessa ficha.

#### **Codigo Cliente -> € gasto**

“Transformar” o codigo de cliente numa ficha de cliente e procurar a ficha na árvore TreeSet<FichaClienteCompras> arvoreClientes. Depois basta ver qual o tamanho de TreeSet<FichaProdutoDeClienteCompras> produtosCliente dessa ficha.

5

#### **Codigo Produto -> N° compras mês a mes**

“Transformar” o codigo de produto numa ficha de produto e procurar a ficha na arvore TreeSet<FichaProdutoContabilidade> produtos. Depois basta consultar a matriz totalComprasPorMes[meses][tipo\_compra] dessa ficha.

#### **Codigo Produto -> N° Clientes diferentes mês a mês**

“Transformar” o codigo de produto num par produto – nº clientes e procurar a o par na arvore TreeSet<ParProdutoNClientes> arvoreParesProdutoNClientes. Depois basta consultar a matriz numeroClientesPorMes[meses] do par obtido.

#### **Codigo Produto -> Facturado mês a mês**

“Transformar” o codigo de produto numa ficha de produto e procurar a ficha na arvore TreeSet<FichaProdutoContabilidade> produtos. Depois basta consultar a matriz facturacaoProdutoPorMes[meses][tipo\_compra] dessa ficha.

6

#### **Codigo Produto -> N° Compras N e P mês a mês**

Mesmo que o 1º ponto da querie 5.

#### **Codigo Produto -> Facturação para N e P mês a mês**

Mesmo que o 3º ponto da querie 5.

7

#### **Codigo Cliente -> Produtos mais comprados (ord qtd + alfa)**

“Transformar” o codigo de cliente numa ficha de cliente e procurar a ficha na árvore TreeSet<FichaClienteCompras> arvoreClientes. Da ficha obtida, percorrer TreeSet<FichaProdutoDeClienteCompras> produtosCliente e para cada elemento adiciona-lo a um array. Ordenar o array pela variavel numUnidadesCompradasProdutoCliente de cada ficha e devolver o array ao utilizador. (Ter o cuidado de criar copias ao passar para o array)

8

#### **X Produtos mais vendidos (em unidades)**

Adicionar cada elemento da arvore TreeSet<FichaProdutoContabilidade> produtos a um array, e ordenar o array de acordo com a variável numUnidadesProdutoVendidas de cada ficha. Do array ordenado, pegar nos X primeiros e copia-los para um outro array, que sera devolvido ao utilizador.

#### **X Produtos mais vendidos -> N clientes distintos de cada produto**

Para cada produto, “Transformar” o código de produto num par (produto, nº clientes) e procurar a o par na árvore `TreeSet<ParProdutoNClientes> arvoreParesProdutoNClientes`. Depois basta consultar a variável `numeroClientes`;

**9**

**X -> X Clientes com mais produtos diferentes comprados**

Criar uma classe que seja um par (Cliente, nº produtos comprados), percorrer a árvore `TreeSet<FichaClienteCompras> arvoreClientes` e formar um par (Cliente, nº produtos comprados) para cada nó. Junta esses nós num array e ordenar segundo a segunda componente dos pares.

**10**

**Código Produto -> X clientes que mais compraram + € gasto**

Criar uma classe que seja um triplo (Cliente, nº produtos comprados, total gasto). Percorrer a árvore `TreeSet<FichaClienteCompras> arvoreClientes` em que para cada nó dessa árvore também se consulta a árvore `TreeSet<FichaProdutoDeClienteCompras> produtosCliente` para saber quanto o cliente comprou e quanto gastou. Com essa informação construir o triplo. No fim, juntar os triplos todos num array e ordenar pela segunda componente do triplo.