



FACULDADE DE  
CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE DE  
**COIMBRA**

**Programação Orientada a Objetos –  
Licenciatura em Engenharia Informática**

# **POO Financial Services**

**Trabalho realizado por:  
Miguel Pereira e Carlos Pereira**

**Dezembro, 2024**

## Índice

1.	Objetivo e Introdução .....	3
1.1	Estrutura Geral .....	3
1.2	Visão geral das classes usadas .....	4
2.	Funcionamento .....	5
2.1	Menu principal .....	5
2.2	Criação e edição de clientes .....	5
2.3	Criação e edição de faturas .....	6
2.4	Parsing do ficheiro de faturas .....	6
3.	Conclusão .....	7

## Índice de Figuras

Figura 1 - UML da Aplicação .....	7
-----------------------------------	---

# 1. Objetivo e Introdução

O principal objetivo deste relatório é a apresentação das soluções obtidas durante o processo de desenvolvimento de um serviço de faturação utilizando métodos de programação orientada a objetos.

O programa é constituído por faturas, clientes e 5 tipos de produtos diferentes. Cada produto e subproduto tem características diferentes (tanto o formato das mesmas, método de cálculo do IVA, etc.).

É possível também visualizar as faturas de um modo geral, neste caso apresenta a fatura com o seu número de fatura, o número de contribuinte do cliente associado, o número de produtos e os preços do IVA, preços totais dos produtos sem IVA e preços totais com IVA inseridos no programa. Também é possível visualizar cada fatura detalhadamente, o nome, descrição, tipo de produto e subproduto, código do produto, preço do IVA, preço sem IVA e preço com IVA, de cada produto associado a cada fatura, bem como também os já mencionados preços para o total da fatura. Existe ainda também a visualização de todos os clientes, detalhando o nome, número de contribuinte e localização.

Quanto à interação com o utilizador, esta realizar-se-á através de uma Interface de Linha de Comando (CLI).

## 1.1 Estrutura Geral

Tendo em conta os objetivos do projeto, a estrutura decidida foi a ilustrada no final do relatório (Figura 1).

Adicionalmente, o método *main* está numa classe *main* à parte, usada para a CLI.

Dentro da pasta *src*, onde está o código fonte das classes criadas, existe também dois ficheiros, “*faturas.txt*”, que contém faturas previamente guardadas, e “*ficheiroObjetos.obj*”, que contém os clientes e faturas guardados como objetos.

## 1.2 Visão geral das classes usadas

### 1.2.1 Visão das classes

- **Main**, usada para gerir o menu e criar uma instância de POOFs.
- **POOFs** para gerir todas as faturas, produtos e clientes, bem como criar e editar.,
- **Cliente**, usada para gerir os atributos de cada cliente, sendo esses o nome, número de contribuinte e localização.
- **Fatura**, para gerir os atributos de cada fatura, sendo esses o número de fatura, o número de contribuinte do cliente associado à fatura, a lista de produtos que pertencem à fatura, o preço total sem IVA e o preço do IVA.
- **Produto**, uma classe abstrata que fará a gestão dos produtos. Desta herdam as seguintes classes:
  - **ProdutosAlimentares**, uma classe abstrata que fará a gestão de produtos alimentares, que podem ser ou não biológicos, mas, nomeadamente, das suas diferentes taxas de IVA:
    - **ProdutosAlimentaresNormal**, para gerir métodos e atributos específicos aos produtos alimentares que apenas são ou não biológicos.
    - **ProdutosAlimentaresIntermedio**, para gerir métodos e atributos específicos aos produtos alimentares que fazem parte das categorias *enlatados, vinho, congelados*.
    - **ProdutosAlimentaresReduzido**, para gerir métodos e atributos específicos aos produtos alimentares que têm algum dos certificados *ISO22000, FSSC22000, HACCP, GMP*.
  - **ProdutosFarmaceuticos**, uma classe abstrata que fará a gestão de produtos farmacêuticos, que podem ter ou não prescrição:
    - **ProdutosFarmaceuticosPrescritos**, para gerir métodos e atributos específicos aos produtos farmacêuticos que têm prescrição, inclui também o nome do médico.
    - **ProdutosFarmaceuticosNaoPrescritos**, para gerir métodos e atributos específicos aos produtos farmacêuticos que não têm prescrição, estes fazem parte de uma das categorias *BEM-ESTAR, ANIMAIS, BELEZA, BEBES, OUTROS*.

### 1.2.2 Visão das interfaces

- **LerConsola**, usada para ler da consola apenas uma das variáveis *int*, *double* ou *String*, garantido que é apenas essa a variável lida pela consola.
- **GerenciadorFicheiros**, usada para guardar e ler faturas em ficheiro *.txt* ou guardar e ler as listas de classes de faturas e clientes em ficheiro *.obj*.

## 2. Funcionamento

### 2.1 Menu principal

O programa começa por criar uma instância da classe *POOFs*, caso exista um ficheiro de objetos, esse ficheiro inicializa a classe, caso não, é criado associando uma classe não inicializada.

Já dentro do menu principal temos as opções de criar ou editar um cliente, listar clientes, criar ou editar faturas, listar faturas, visualizar uma fatura, guardar faturas em ficheiro *.txt*, ler faturas guardadas em ficheiro *.txt*, mostrar estatísticas gerais sobre *POOFs* e sair, guardando em ficheiro *.obj* as listas de classes de faturas e clientes.

### 2.2 Criação e edição de clientes

A criação dos clientes é feita associando à *ArrayList* de classe *Cliente* um objeto da classe *Cliente*.

Pede-se ao utilizador por um número de contribuinte, é verificado se o número fornecido contém 9 algarismo, um nome e uma localização *Continente*, *Açores* ou *Madeira*, para definir o cliente.

A edição dos clientes é feita modificando um objeto da classe *Cliente* da *ArrayList* de classe *Cliente*.

Pergunta-se ao utilizador se deseja alterar o nome, número de contribuinte ou localização, alterando-o no objeto *Cliente*. Caso seja modificado o número de contribuinte, é percorrido todas as faturas e em cada uma que tenha o número de contribuinte antigo é também alterado para o novo número de contribuinte.

## 2.3 Criação e edição de faturas

A criação das faturas é feita associando à *ArrayList* de classe *Fatura* um objeto da classe *Fatura*. Pede-se ao utilizador por um número de contribuinte e por produtos, que são associados a um *ArrayList* de classe *Produto*.

Na criação de produtos pergunta-se pelo nome do produto, descrição de produto, código de produto, quantidade e valor unitário. De seguida o tipo de produto e subtipo de produto, tendo cada subtipo de produto as suas nuances específicas para a criação do produto.

A edição de faturas é feita retirando ou adicionando produtos a faturas existentes, ou alterando o número de contribuinte associado à fatura.

Pede-se ao utilizador por um número de fatura, de seguida, dependendo da escolha do utilizador, altera-se o número de contribuinte. Caso escolha por remover ou adicionar produto, procede à remoção do produto via eliminação do objeto *Produto* da *ArrayList* de classe *Produto* verificando pelo código de produto, ou adiciona um produto da mesma maneira que na criação da fatura.

## 2.4 Parsing do ficheiro de faturas

O parsing dos ficheiros é efetuado por auxílio à interface *GerenciadorFaturas*, a qual contém funções para ler e guardar em *.obj* ou *.txt*.

Para os ficheiros *.obj* é guardado as *ArrayList* de *POOFs*, *Cliente clientes* e *Fatura faturas*, bem como também o inteiro que numera a fatura seguinte. A sua leitura é apenas carregando esses objetos outra vez para o *POOFs* quando o programa inicializa, através do construtor que recebe os valores guardados.

Para os ficheiros *.txt* é guardado em cada linha cada fatura no seguinte formato

<i>NumFatura</i>		<i>NumContribuinte</i>		<i>PrecoSemIva</i>		<i>PrecoDoIVA</i>		<i>NomeProduto</i>	
<i>CodigoProduto</i>		<i>DescricaoProduto</i>		<i>Quantidade</i>		<i>ValorUnitario</i>		<i>TipoDeProduto</i>	
<i>SubtipoDeProduto</i>									

Depois dependendo do tipo e subtipo de produto acrescenta-se ainda:

- |*SubtipoDeProduto*|*Categoria*, para Produtos Farmaceuticos Não Prescritos;

- `|SubtipoDeProduto|NomeDoMedico`, para Produtos Farmaceuticos Prescritos;
- `|SubtipoDeProduto|isBiologic`, para Produtos Alimentares de taxa Normal;
- `|SubtipoDeProduto|isBiologic|CategoriaAlimentar`, para Produtos Alimentares de taxa Intermédia;
- `|SubtipoDeProduto|isBiologic|Certificados{separadas por “,”}`, para Produtos Alimentares de taxa Reduzido.

A sua leitura é evidentemente feita pelo inverso, lendo via este formato, criando clientes e faturas, associando à fatura o cliente e os produtos.

### 3. Conclusão

Deste projeto, resultou, não só um entendimento mais profundo sobre programação orientada a objetos, como também um jogo programa de faturação bastante completo, que cumpre todos os objetivos colocados no início do projeto.

A estrutura final do projeto (Imagem 4.1), sofreu bastantes alterações desde a estrutura inicial, no entanto, todas elas foram justificadas e todas as soluções encontradas, apesar de não haver espaço para as descrever neste relatório.

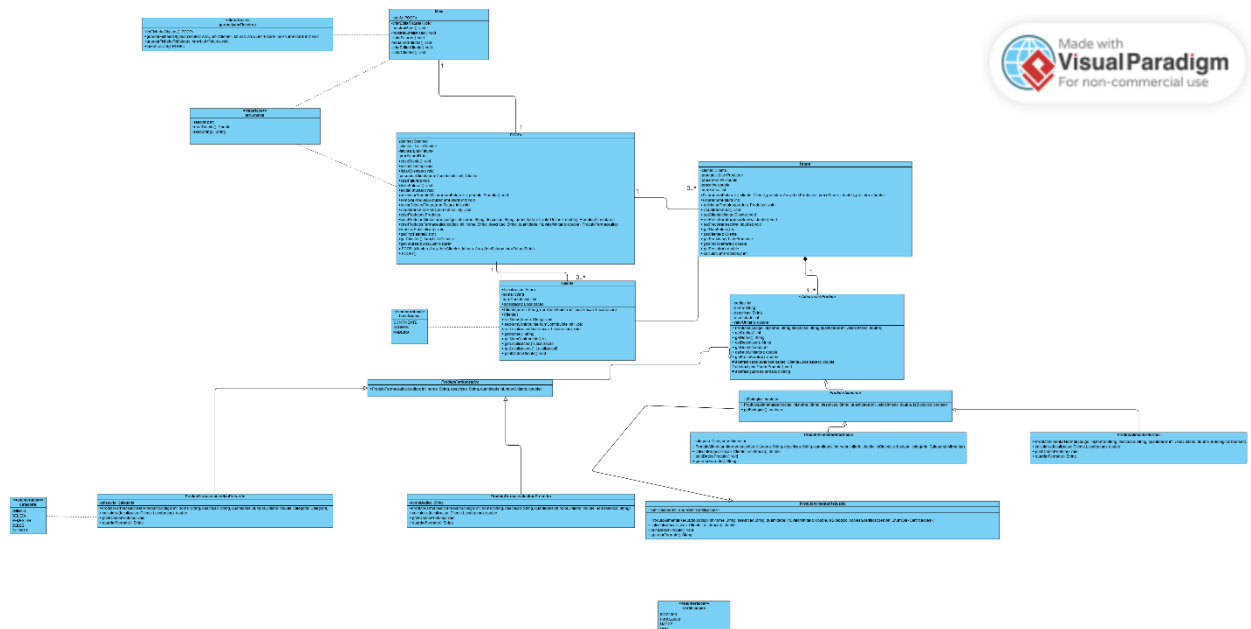


Figura 1 - UML da Aplicação

Figura 3.1 – UML da aplicação