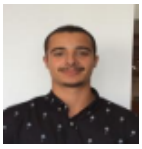
Projeto de Programação Orientada aos Objetos
Grupo 19

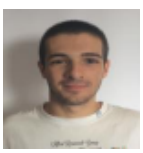
Henrique Pereira (a80261) Pedro Moreira (a82364)
Pedro Ferreira (a81135)

28 de Maio de 2018



(a) Henrique Pereira

(b) Pedro Moreira



(c) Pedro Ferreira

Conteúdo

1 Introdução

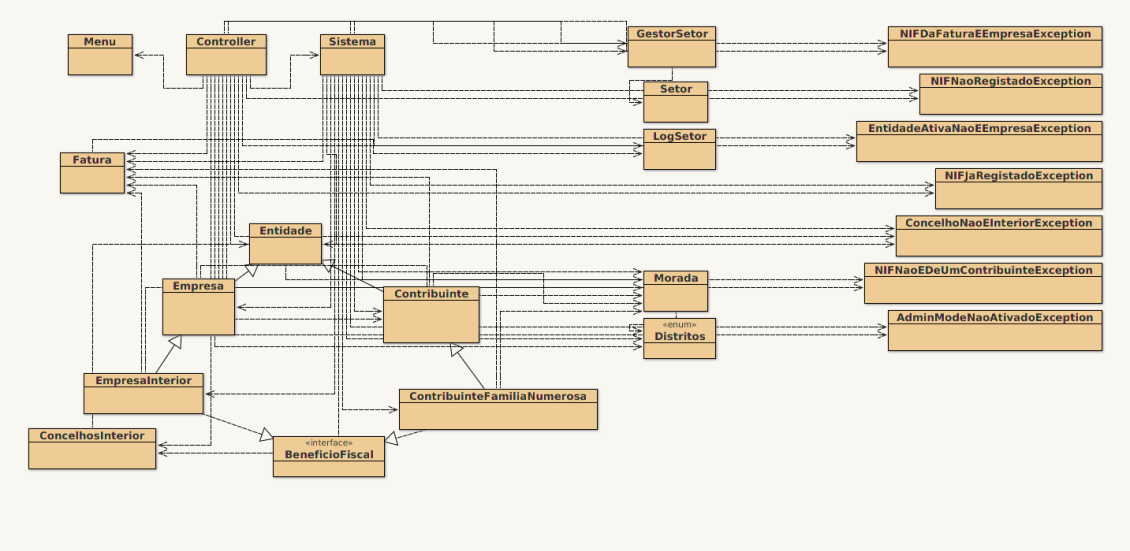
No contexto da Unidade Curricular Programação Orientada aos Objetos (POO), foi-nos proposto o desenvolvimento de uma aplicação chamada JAVA-Fatura. Tal como o nome indica, o enunciado proposto pelos docentes visa o desenvolvimento de uma versão da plataforma E-Fatura, de consulta e emissão de faturas, na linguagem *Java*. Os requisitos básicos são, por exemplo, por parte do contribuinte individual, verificar as faturas emitidas em seu nome e, por parte de uma empresa, obter o total faturado. Na secção 2 serão apresentados mais requisitos desta aplicação e também serão discutidas as classes criadas para a concretização dos requisitos apresentados. Na secção 2.4 será discutida a forma como os utilizadores podem interagir com a aplicação.

2 JAVA-Fatura

A aplicação teria que permitir, além da criação de novos contribuintes e empresas no sistema, de fazer a identificação correta da entidade que se encontra a utilizar o Java-Fatura, isto é, teria que ser capaz de efetuar o Login no Sistema com sucesso, perante as credenciais dadas. Para além dos requisitos já referidos, os contribuintes individuais devem ser capazes de verificar o montante de dedução fiscal acumulado (por si e pelo agregado familiar) e atribuir ou corrigir o setor de atividade económica de uma fatura, sendo que esta alteração teria que ser mantida em registo. As empresas deverão ser capazes de de obter vários tipos de listagens sobre as faturas que emitiram, ordenadas por valor ou data. Para além disso, a aplicação necessita de ter um admistrador capaz de determinar os contribuintes que mais gastam no sistema e as empresas que mais faturaram ou de adicionar um novo setor de atividade económica. Para a realização deste projeto, decidimos criar as classes que descrevemos na secção 2.1.

2.1 Classes

Antes de iniciarmos a descrição de cada uma das classes, começamos por mostrar o diagrama das classes implementadas (pelo *BLUEJ*).



2.1.1 BeneficioFiscal

Interface definida para que as famílias numerosas e as empresas do interior possam ter um beneficio fiscal aquando da dedução das suas faturas. Esta interface contém o método reducaoImposto não definido.

2.1.2 ConcelhosInterior

Classe que contém os concelhos que podem usufruir de beneficio fiscal. Pelo facto de serem imensos, decidimos optar por selecionar apenas nove concelhos abrangidos por este benefício, entre eles Penalva do Castelo, Vinhais, Resende, Ribeira de Pena, Baião, Celorico de Bastos, Tabuaço, Cinfães e Mirandela.

Possui apenas uma variável de instância: um mapa que associa a cada concelho do interior a sua taxa de dedução.

2.1.3 Contribuinte

Sendo subclasse de Entidade, esta classe tem como variáveis de instância a lista das faturas emitidas em seu nome (separadas pelas que são fiscalmente válidas ou inválidas/pendentes) e o número de dependentes familiares e os seus NIF's.

2.1.4 ContribuinteFamiliaNumerosa

Implementa a interface BeneficioFiscal e é subclasse de Contribuinte. É aqui que definimos a taxa de benefício fiscal das famílias numerosas e o bónus fiscal que cada dependente representa.

2.1.5 Controller

Classe intermediária entre a comunicação do utilizador com o sistema. É lá que está definida a *main* e as funções que nos permitem interagir com os menus por nós criados, e apresentados pelo terminal. A variável de instância que representa o estado do Sistema é criada aqui.

2.1.6 Distritos

Enum contendo todos os dezoito distritos do país.

2.1.7 Empresa

Subclasse de Entidade que contém as faturas emitidas e os setores de atividade económica. É nesta classe que estão definidos os métodos que permitem a ordenação das faturas, assim como a inserção destas no sistema.

2.1.8 EmpresaInterior

Subclasse de Empresa que implementa a interface BeneficioFiscal. É com esta classe que permitimos que os bónus fiscais sejam atribuidos às empresas do interior do país, isto é, cuja atividade está declarada como sendo efetuada num dos concelhos considerados pelo Sistema como sendo do Interior.

2.1.9 Entidade

Classe que contém as informações básicas das entidades do sistema, tais como o nome, o email e a morada, assim como as credenciais necessárias para o login (NIF e password).

2.1.10 Fatura

Classe onde são guardadas as informações de cada fatura, isto é, o nome e o NIF da empresa, a data de emissão da fatura, o NIF do cliente e a descrição e valor da fatura. Além disso, é também guardado o setor de atividade económica atribuído e o registo de alteração deste.

2.1.11 GestorSetor

Classe auxiliar à classe Fatura para que seja possível ao Contribuinte associar o setor de atividade económica que deseja à fatura em causa. Além disso, permite também saber qual a taxa de dedução fiscal associado a cada Setor, além de permitir também que sejam adicionados novos setores de atividade ao sistema.

2.1.12 LogSetor

Classe auxiliar à classe GestorSetor que guarda o histórico de todas as alterações do setor de atividade económica que um contribuinte realizou. Contém ainda informações relativas ao setor ativo.

2.1.13 Menu

Classe que faz o display das operações que o utilizador pode realizar, isto é, mostra-nos as operações possíveis nos diferentes Menus: inicial, contribuinte, empresa ou administrador.

2.1.14 Morada

Classe que permite agregarmos todas as informações relativas à morada de uma entidade, facilitando assim o seu manuseamento. É aqui que representamos a rua, o código postal, o concelho e o distrito de uma entidade.

2.1.15 Setor

Classe que contém as informações sobre a dedução relacionada com o setor de atividade económica, ou seja, as taxas de dedução, se as despesas relacionadas com esse setor são dedutíveis e, se assim forem, qual o máximo valor dedutível.

2.1.16 Sistema

Classe que guarda todas as informações da aplicação (dados de todas as entidades). Assim sendo, definimos aqui um HashMap com todas as entidades do sistema, um GestorSetor e as informações relativas às credenciais de admistrador, que são as seguintes:
NIF: `20172018`
Password: `miei`

2.2 Classes de Exceção

- AdminModeNaoAtivadoException
- ConcelhoNaoEInteriorException
- EntidadeAtivaNaoEEmpresaException
- NIFDaFaturaEEmpresaException
- NIFJaRegistadoException
- NIFNaoEDeUmContribuinteException
- NIFNaoRegistadoException

2.3 Decisões Tomadas

O desenvolvimento da aplicação foi feito procurando facilitar a implementação de outras versões gráficas da mesma, ou até mesmo, uma versão cliente-servidor. Para isso a classe Sistema, onde se encontra armazenada a informação sobre os utilzadores da aplicação, disponibiliza um conjunto de métodos tornados públicos através dos quais a parte gráfica deve estabelecer a conexão entre o utilizador da aplicação e o código que realiza operações, obtendo por parte do utilizador os dados com os quais deve alimentar o sistema. De modo a ser possível adicionar um novo setor de atividade económica sem que seja necessário proceder a alterações no código e recompilar o programa, optamos por mantar apenas um objeto da class GestorSetor em toda a aplicação, estando este contido no Sistema. Entrando na aplicação com privilégios de admin, é possivel adicionar um novo stor de atividade económico e este estará disponível para todos os restantes utilizadores da aplicação o usarem. As operações de I/O também passam a contemplar esta nova opção sem ser preciso reescrever código.

2.4 Funcionalidades

Qualquer utilizador quando inicia a aplicação é confrontado com o seguinte menu:

```
****************JavaFatura****************
1 - Login
2 - Registar Contribuinte
3 - Registar Empresa
0 - Sair
Operação a realizar:
```

Caso o utilizador escolha efetuar o login (1) e o efetue com sucesso (mediante registo prévio no sistema) o menu seguinte será diferente para cada uma das entidades apresentadas abaixo. Ao registar, quer um contribuinte, quer uma empresa, serão pedidas informações relativas a cada uma das entidades.

2.4.1 Contribuinte

```
****************JavaFatura****************
1 - Consultar Faturas
2 - Total Deduzido
3 - Deduções Agregado Familiar
4 - Atribuir Setor de Atividade Económica a uma Fatura
5 - Alterar Setor de Atividade Económica de uma Fatura
0 - Voltar Atrás
Operação a realizar:
```

2.4.2 Empresa

```
****************JavaFatura****************
1 - Emitir Fatura
2 - Consultar Faturas (por data de emissão)
3 - Consultar Faturas (por valor)
4 - Consultar Faturas (de um cliente em específico)
5 - Consultar Faturas emitidas entre determinada data
6 - Consultar Faturas emitidas entre determinada data (por valor)
7 - Consultar Faturas de um cliente específico, entre determinada data
8 - Total Faturado
0 - Sair
Operação a realizar:
```

2.4.3 Admistrador

```
***************JavaFatura***************
1 - Top 10 Contribuintes: total do valor das faturas
2 - Top X Empresas: relação entre o nº de faturas emitidas e o montante de deduções fiscais que as despesas registadas representam
3 - Adicionar novo setor de despesa
0 -  Sair
Operação a realizar:
```

3 Conclusões

Este projeto foi extremamente enriquecedor para o grupo, uma vez que nos permitiu pôr em prática os conhecimentos adquiridos nas aulas de POO e desenvolver uma aplicação em JAVA, tendo em conta as orientações dadas pelos docentes no que diz respeito a Programação Orientada aos Objetos.

Em suma, na nossa opinião, os requisitos propostos foram cumpridos e conjugados com êxito, sendo que o resultado final se revela bastante apelativo e funcional.

Foi, no geral, um trabalho altamente pedagógico, que nos permitiu desenvolver não só competências essenciais ao curso de Mestrado Integrado em Engenharia Informática, como também competências sociais, tais como o trabalho em grupo.