Programação Orientada aos Objetos

2017/2018

LCC

Relatório do trabalho prático

Grupo nº 7:

A80298; Arlindo da Fonte Torres



A80584; Nelson José Dias Teixeira



A81241; José Alberto Martins Boticas



**Conteúdo**

1. **Introdução 3**
2. **Arquitetura de classes utilizada 4**
3. **Funcionalidades 8**
4. **Como utilizar 8**
5. **Possibilidades de alterações ao projeto 8**
6. **Conclusões 9**

**1 Introdução**

No âmbito da Unidade Curricular de Programação Orientada aos Objetos (POO), foi proposto que se realizasse um trabalho prático, recriando, com as devidas alterações, a plataforma e-fatura, que disponibiliza aos contribuintes informações sobre as faturas emitidas em seu nome e lhes permite, entre outros, verificar o montante de deduções fiscais e associar um tipo de despesa a uma fatura.

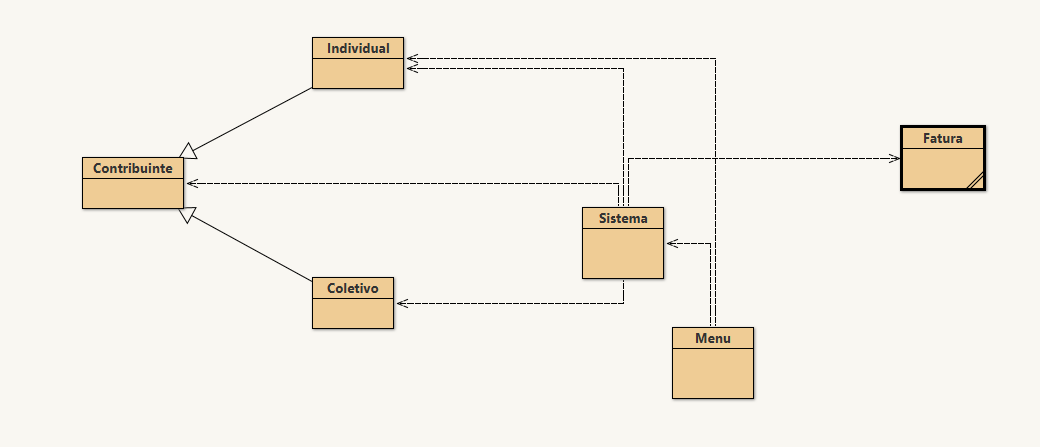
Para este projeto, deve desenvolver-se uma plataforma que permite:

1. registar contribuintes, quer sejam individuais ou coletivos (empresas);
2. validar o acesso à plataforma, utilizando as credenciais (NIF e password), por parte de diferentes entidades;
3. criar faturas associadas a despesas feitas por um contribuinte individual, sendo esta funcionalidade responsabilidade das empresas;
4. aos contribuintes individuais, consultar as suas faturas;
5. aos contribuintes individuais, associar uma atividade económica às faturas emitidas em seu nome e que ainda não tenham uma associada;
6. aos contribuintes individuais, visualizar o valor acumulado de deduções fiscais;
7. corrigir a classificação de atividade económica de uma fatura, sendo deixado um rasto para posterior rastreamento;
8. obter a listagem das faturas de uma determinada empresa, ordenada por data de emissão ou por valor;
9. às empresas, obter as listagens das faturas por contribuinte num intervalo de datas;
10. às empresas, obter as listagens das faturas por contribuinte ordenadas por valor decrescente da despesa;
11. indicar o total faturado por uma dada empresa, num determinado intervalo de tempo;
12. ao administrador da aplicação, determinar a relação dos 10 contribuintes que mais gastam em todo o sistema;
13. ao administrador da aplicação, determinar a relação das X empresas que mais faturas emitiram em todo o sistema e o montante de deduções fiscais que as despesas registadas por essas empresas representam;
14. gravar o estado da aplicação em ficheiro, para seja possível retomar mais tarde a execução.

Para além destes requisitos, foram impostas regras à forma como se emite uma fatura, sendo estas as seguintes:

1. se uma empresa tiver apenas uma atividade económica, então a despesa fica automaticamente catalogada e deve determinar-se qual o valor a acrescentar às deduções fiscais do contribuinte;
2. se a empresa tiver mais do que uma atividade económica, deve ser o contribuinte a catalogar a despesa e só aí deve ser calculado o valor de dedução fiscal;
3. enquanto o contribuinte não validar a despesa, esta fica no estado pendente, não deixando de ser associada a esse contribuinte;
4. se o contribuinte não validar a despesa, esta não é considerada para efeitos de deduções.

O cálculo das deduções fiscais dependerá sempre, essencialmente, do setor de atividade económica associado à fatura, mas também poderá ser influenciado pela empresa que a emite ou pelo agregado familiar do cliente, havendo bonificações para empresas do interior ou famílias numerosas (com mais de quatro filhos). Uma empresa terá bonificação de empresa do interior se se inserir num concelho com incentivo fiscal.

**2 Arquitetura de classes utilizada**

**2.1 – Contribuinte**

Esta é uma superclasse para gerar contribuintes e que define variáveis como: NIF, email, nome, password, morada, o seu coeficiente fiscal e uma lista contendo os índices das faturas associadas ao contribuinte.

**2.2 – Individual**

A classe Individual é uma subclasse de Contribuinte que representa os contribuintes individuais, contendo informações relativas ao agregado familiar, aos seus dependentes e às atividades económicas.

**2.3 – Coletivo**

Na classe Coletivo, que é também uma subclasse de Contribuinte, representamos os contribuintes coletivos, ou seja, as empresas, recorrendo a *HashMap*’s para representação das atividades económicas de venda e de compra. A localização da empresa representa também um papel importante devido ao facto dos benefícios fiscais das empresas que atuam em certos concelhos do interior.

**2.4 – Fatura**

Esta é a classe que contém todas as informações relativas às faturas emitidas pelas empresas e define variáveis como o NIF do emitente, nome do emitente, data e hora da fatura, NIF do cliente, descrição da despesa, o seu valor e um booleano indicando se a fatura ainda requer a validação de atividade económica por parte do cliente.

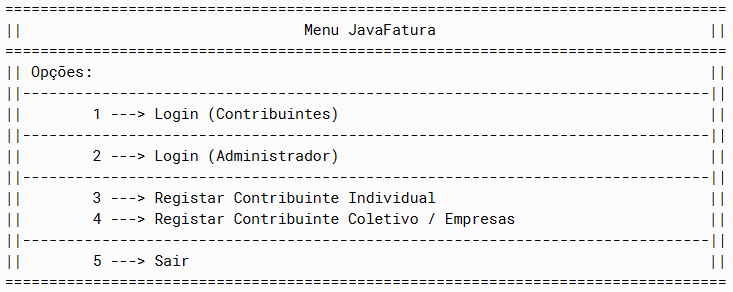
**2.5 – Sistema**

Classe que gere todas as funcionalidades do sistema, gerindo os contribuintes, as faturas, as atividades económicas disponíveis, a lista de agregados familiares, os concelhos com benefícios fiscais para empresas, entre outras funcionalidades de Input/Output. Esta classe contém também os métodos de salvaguarda do estado da aplicação.

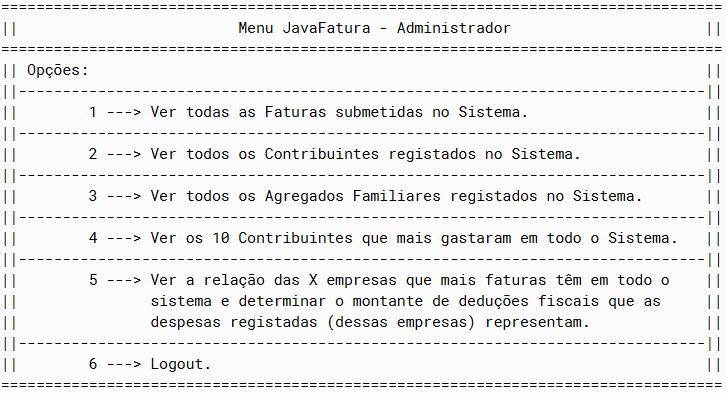
**2.6 – Menu**

Classe responsável pela apresentação e interação do sistema com o utilizador. Inicializa o sistema da aplicação carregando um estado, caso exista.

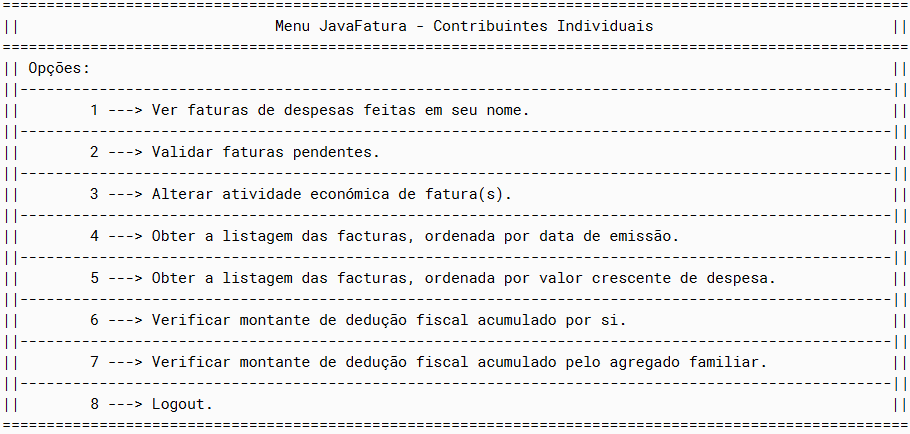
**2.6.1 – Visão Inicial**



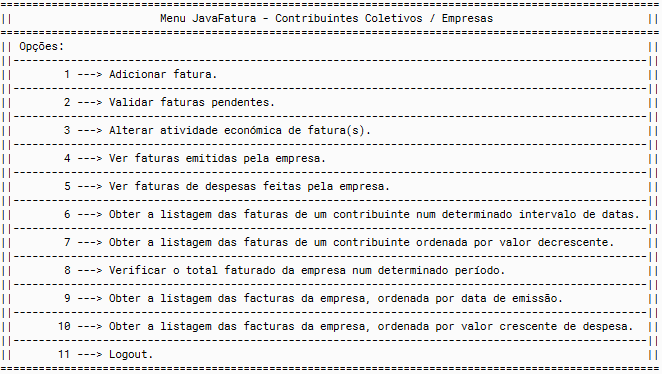
**2.6.2 – Visão Administrador**



**2.6.3 – Visão Contribuinte Individual**

****

**2.6.4 – Visão Contribuinte Coletivo**

****

**3 Funcionalidades**

A aplicação contém um sistema de registo e login de contribuintes. Os contribuintes individuais podem ver as faturas associadas ao seu NIF e validar faturas que estejam pendentes, bem como alterar a atividade económica das faturas, caso haja mais do que uma possibilidade.

O contribuinte individual também consegue obter a listagem das faturas, ordenadas tanto por data de emissão como por valor de despesa. Também é possível verificar o montante de dedução discal acumulado por si e pelo agregado familiar.

Os contribuintes coletivos podem emitir faturas, validar faturas pendentes que foram emitidas em seu nome, obter a listagem de faturas emitidas pela empresa e com alguns filtros (contribuinte, intervalo de datas, valores). Também é possível verificar o total faturado pela empresa num determinado período.

Existe um painel de administrador do sistema que nos permite ver todas as faturas no sistema, contribuintes registados, agregados familiares registados, ver os 10 contribuintes que mais gastaram e obter a relação de um certo número de empresas que mais faturas emitiram, determinando o montante de deduções fiscais que as despesas registadas representam.

**4 Como utilizar**

Para utilizar a aplicação desenvolvida deve-se, a partir do *BlueJ* executar a classe *Main*, sem argumentos. A partir deste ponto, aparecerá sempre um menu, com variadas opções. Tudo o que utilizador terá de fazer é escolher o número associado à opção que pretende executar. A aplicação é bastante intuitiva e não deverá causar dúvidas na sua utilização.

**5 Possibilidades de alterações ao projeto**

Em relação ao que foi implementado no projeto apresentado pelo nosso grupo, poderiam ter-se modificado ou acrescentado alguns métodos. Entre eles destacam-se:

1. Acrescentar às classes Fatura, Contribuinte e Coletivo métodos de ordenação (ordem natural, entre outros) e criar um *Map* que associasse um nome de um comparador - Chave - ao respetivo comparador - Valor.

Consequentemente, seria necessário adicionar um método que, dado o nome do comparador, retornaria o mesmo. Desta maneira, sempre que a classe Sistema necessitasse de um método para ordenar um *TreeSet* de elementos da classe Fatura, Contribuinte ou Coletivo, bastaria aceder ao comparador pretendido. Assim, uma vez que seria criado um comparador de cada vez, esta seria uma solução mais eficiente.

1. Possibilidade de registar as faturas numa variável de instância do tipo *Set*. Desta forma, garantidamente não existiriam faturas repetidas na classe Sistema.
2. Com a condição de se acrescentar uma variável respeitante aos rendimentos de cada um dos contribuintes individuais, seria possível fazer o cálculo do IRS do mesmo.
3. Acrescentar o cálculo de deduções fiscais para empresas, ou outros impostos, com a possibilidade de usar a variável que indica se esta é ou não do interior para eventuais benefícios.
4. Acrescentar métodos para fins estatísticos, quer ao nível do contribuinte - coletivo ou individual -, quer a nível global.
5. **Conclusões**

Este projeto serviu para aprofundarmos o conhecimento sobre programação orientada aos objetos em aplicações do mundo real. A realização de um trabalho deste tipo permite-nos questionar os melhores métodos de trabalho em equipa de modo a maximizar o nosso desempenho. A utilização do sistema de gestão de versões Git foi sem dúvida uma das decisões mais importantes no início deste projeto, já que facilitou bastante o trabalho em equipa.

Estes aspetos deveriam ser mais abordados no nosso curso, tanto da gestão das versões do nosso código, como de métodos de trabalho em equipa eficientes, pois são essenciais tanto para estes projetos de UC’s, e ainda mais quando entrarmos no mercado de trabalho.

O trabalho por nós desenvolvido foi testado e inclui várias medidas que evitam que o sistema falhe caso exista algum engano na escolha de alguma opção para execução. Todos os requisitos são garantidos, havendo algumas redundâncias, particularmente no acumular das despesas das empresas, que não é depois utilizado. No entanto, estas redundâncias deixam em aberto algumas opções para o futuro, particularmente para o cálculo das deduções fiscais das empresas. Poderiam ter sido tomadas outras opções, nomeadamente nas classes *Sistema* e *Menu*, mas optamos por aquelas que nos trariam menos problemas, tanto na sua construção como na sua interação com outras classes.