

Universidade do Minho

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA

PROGRAMAÇÃO ORIENTADA A OBJETOS

Java Fatura

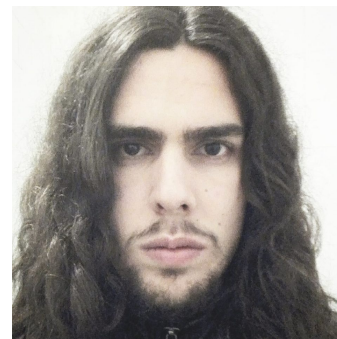
Membros do Grupo 36



Bruno Martins A80410



Leonardo Neri A80056



Márcio Sousa A82400

que apenas são instanciadas empresas, contribuintes e o admin. A classe `IdentidadeFiscal` contém as variáveis de instância que serão comuns às suas 3 subclasses, como o nome, o email, a password, etc. Cada uma das subclasses implementa depois o que as diferencia das outras subclasses, justificando a criação destas, como a lista de dependentes no caso do Contribuinte, o concelho a que pertence no caso das Empresas. A subclasse `Admin` foi criada para facilitar a forma de lidar com os privilégios extra deste utilizador.

2.2 Fatura

A classe `Fatura` é a classe a partir da qual serão instanciadas todas as faturas no programa, não havendo subclasses, pois todas as faturas têm a mesma informação, que consiste num numero de fatura, o NIF do emitente e do cliente, a descrição da despesa, o histórico de atividades atribuídas a essa fatura, o valor da despesa e a data. É importante salientar que é usado uma lista para as atividades devido à funcionalidade que possibilita ver o histórico de atividades que uma fatura já teve, sendo que estas são primeiramente emitidas como faturas pendentes, não lhes estando atribuída uma atividade específica, exceto se a empresa que a emitiu apenas se focar num setor económico. Quando é necessário aceder à atual atividade da `Fatura` em questão é apenas retornado o último elemento da lista.

2.3 Atividade

A classe `Atividade` é a classe principal das subclasses de atividades económicas, como a `AtividadeAlimentacao` e a `AtividadeSaude`, por exemplo. Estas subclasses têm apenas uma coisa em comum, o nome da atividade. Em cada subclasse da `Atividade` (exceto `Lazer` pois não conta para dedução) é definido o método de dedução que vai calcular o valor que deve ser deduzido em cada fatura, caso a `Atividade` correspondente seja uma das que contam para dedução fiscal. Neste algoritmo é tido em conta a atividade, a localização da empresa emitente, isto é, se é do interior ou não, o número de dependentes do contribuinte, e o valor máximo de dedução de cada tipo de despesa.

2.4 ControlClass

Se considerarmos o programa como uma caixa fechada com exceção de uma pequena abertura, sendo o interior as classes com a sua informação, e o ex-

terior o interface do utilizador, esta classe seria a entrada da caixa, isto é, faz a ligação entre a informação e o utilizador, retornando aquilo que o utilizador pede (apenas aquilo a que está autorizado), sem que este consiga ver ou alterar o estado interno. É nesta classe que se definem os métodos que proporcionam as "funcionalidades" do programa, e a partir daí são redirecionadas as respostas para o interface do utilizador.

3 Funcionalidades

3.1 Contribuinte

- **Ver Faturas emitidas em seu nome:** através do NIF do contribuinte, é relativamente simples procurar o contribuinte correto no HashMap das Identidades Fiscais, utilizando o NIF como key. Quando já se tem o Contribuinte em questão, é feito o *get* da Lista das faturas que todas as Identidades Fiscais têm, obtendo-se assim todas as faturas referentes a esse contribuinte.
- **Corrigir a Atividade de uma Fatura:** com a informação do contribuinte que está *logged in*, é utilizado o *getFaturas* para obter a lista das faturas. O contribuinte escolhe então uma fatura, e é adicionada à Lista de atividades da fatura a nova atividade.

3.2 Empresa

- **Obter as faturas da empresa ordenadas por data ou valor:** o procedimento é o mesmo que o que permite aos contribuintes verem todas as faturas emitidas em seu nome, mas desta vez, aplicado a uma empresa. Depois de ter a lista das faturas, esta é ordenada pela ordem desejada.
- **Obter a lista de faturas por contribuinte num intervalo de datas:** as faturas da empresa em questão são filtradas de acordo com o NIF cliente, e depois por data, sendo retornadas a lista de faturas que corresponde a esses dois critérios.
- **Obter a lista de faturas por contribuinte ordenadas por valor de despesa:** as faturas da empresa em questão são filtradas por NIF do contribuinte,

e é retornada a lista de Faturas ordenada por valor decrescente de despesa, recorrendo a um comparador.

- **Total faturado por uma empresa:** é percorrida a lista das faturas da empresa, somando o valor das faturas que estão dentro do período de tempo estipulado, sendo retornado o valor total.

3.3 Admin

- **Relação 10 contribuintes que mais gastam:** é percorrido o HashMap que contém todas as identidades fiscais e são filtrados todos os contribuintes individuais para uma lista que é ordenada segundo os gastos de cada um deles. No final são mantidos apenas na lista os primeiros dez contribuintes.
- **Relação 10 empresas que mais faturam:** é percorrido o HashMap que contém todas as identidades fiscais e são filtradas todas as Empresas para uma lista que é ordenada segundo o total faturado de cada um deles. No final mantemos apenas na lista os primeiros dez contribuintes

4 Estruturas de Dados

No que toca a estruturas de dados, foram usados ArrayLists, e um HashMap para guardar as identidades fiscais.

5 Conclusão

Este projeto foi uma excelente oportunidade para aplicar conhecimentos relativos ao paradigma de programação orientada a objetos, usando Java como linguagem de desenvolvimento do projeto. Relativamente a possíveis aspetos a melhorar no projeto, salienta-se que poderia ter sido usado um algoritmo de cálculo de dedução mais específico, e uma lista de concelhos que abrangesse todos os concelhos do país.