

Universidade do Minho

JAVA FACTURA

**MESTRADO INTEGRADO EM
ENGENHARIA INFORMÁTICA**

PROGRAMAÇÃO ORIENTADA A OBJECTOS

(2º ANO, 2º SEMESTRE, 2017/2018)

Grupo 11

A79003 Pedro Mendes Félix da Costa



Braga
Maio 2018

Índice

1	Introdução	3
2	Principais entidades do Sistema	4
2.1	Factura	4
2.2	Sectores de Actividade Económica	5
2.3	User	6
2.4	Concelhos	7
3	Emissão de Facturas e Calculo de Deduções	8
3.1	Emissão de Facturas	8
3.2	Calculo de Deduções	8
4	Interface Grafica	9
4.1	Admin	9
4.2	Contribuinte Individual	10
4.3	Contribuinte Empresarial	10
4.4	Consultar uma Factura	11
5	Extensibilidade da aplicação	12
6	Conclusões e Trabalho Futuro	13

Listas de Figuras

2.1	Diagrama de uma Factura	4
2.2	Implementação <i>Singleton</i> da classe Pendente	5
2.3	Implementação da interface Deductible	5
2.4	Diagrama de classes dos sectores de actividade económica	6
2.5	Hierarquia de Classes do User	7
2.6	Excerto do enum Concelho	7
3.1	Fórmula de cálculo implementada para o Sector Saude	8
4.1	Ecrã de <i>login</i>	9
4.2	Ecrã do Admin	9
4.3	Ecrã do individual	10
4.4	Ecrã do individual	10
4.5	Vista de uma Factura	11

Introdução

Este trabalho foi realizado no âmbito da unidade curricular programação orientada a objectos e teve como objectivo a implementação de um sistema similar ao EFatura, aplicando os conceitos leccionados nas aulas.

Principais entidades do Sistema

Factura

A classe **Factura** representa um documento de despesa no sistema. Esta guarda todas as informações pedidas nos requisitos:

- NIF do emitente
- Data de criação
- NIF do cliente
- Nome do emitente
- Valor da despesa

Além destes, é também guardado um histórico dos estados anteriores da factura que é actualizado sempre que o sector de actividade económica desta é alterado. A data de edição é também actualizada.

A alteração do sector económico tem de obedecer a duas restrições para que seja executada: 1º Uma factura não pode passar a ser Pendente depois de emitida e 2º o sector escolhido tem de ser um dos sectores da empresa que a emitiu.

Para além de todos estes dados, uma instância de **Factura** guarda também alguns dados necessários para calcular a sua dedução. Estes são:

- Se a empresa que a emitiu é uma empresa do interior
- O coeficiente fiscal do emitente
- O coeficiente fiscal do cliente
- O tamanho do agregado familiar do cliente
- Os sectores de actividade para os quais o cliente pode deduzir.

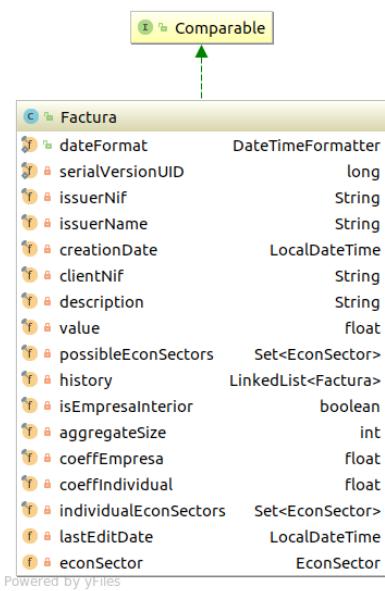


Figura 2.1: Diagrama de uma Factura

Sectores de Actividade Económica

Todos os sectores económicos extendem a classe `EconSector` que serve apenas para disponibilizar um conjunto de todos esses sectores e um método de converter o nome do sector na sua repetiva instância. Estas sublcasses por outro lado implementam um padrão *Singleton*, ou seja, em qualquer momento existe apenas uma instância de cada um destes objectos em memória. A opção pelo padrão *Singleton* foi tomada porque estes objectos não têm estado e, como tal, são imutaveis, podendo-se assim poupar memória, mas, sobretudo, para ter a garantia que as implementações por defeito do `equals()` e `hashcode()` funcionam perfeitamente.

```
public final class Pendente extends EconSector {  
  
    private static final Pendente instance = new Pendente();  
    private Pendente(){}
    public static Pendente getInstance(){  
        return instance;  
    }
    protected Object readResolve(){  
        return getInstance();  
    }
}
```

Figura 2.2: Implementação *Singleton* da classe `Pendente`

Os sectores económicos estão divididos em duas categorias: dedutíveis e não dedutíveis. Esta distinção é feita através de uma interface que obriga à implementação do método `deduction()` que calcula o valor que é deduzido do valor inicial da despesa.

```
public interface Deductible {  
    float deduction(float value, boolean interior, int numDependants,  
                    float coeffEmpresa, float coeffIndividual);  
}
```

Figura 2.3: Implementação da interface `Deductible`

Os dados necessários a este cálculo são:

- Valor da despesa
- Se a factura foi emitida por uma empresa do interior
- O número de dependentes do agregado familiar
- O coeficiente fiscal do cliente
- O coeficiente fiscal da empresa

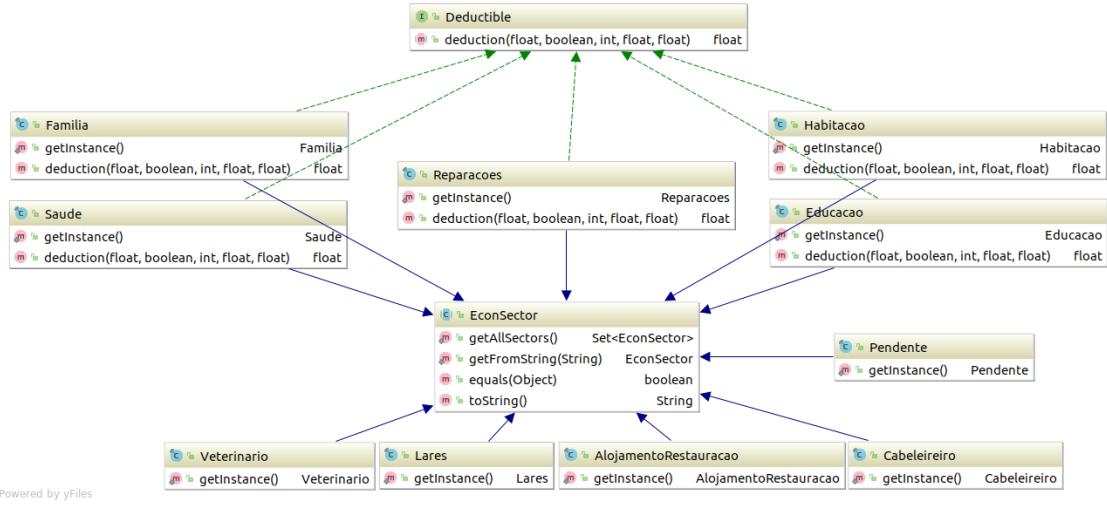


Figura 2.4: Diagrama de classes dos sectores de actividade económica

User

A interface `User` é implementada por duas classes: `Admin` e `Contribuinte`. Esta última, sendo uma classe abstracta, é ainda estendida por `ContribuinteEmpresarial` e `ContribuinteIndividual`.

Um `User` tem obrigatoriamente de implementar

- `getNif()`
- `getPassword()`
- `clone()`.
- `getName()`
- `setPassword()`

Um `Admin` implementa apenas a interface, não adicionando mais nenhuma funcionalidade. Note-se, no entanto, que o nif e nome do admin são sempre iguais a "admin".

A classe `Contribuinte`, para além de implementar a interface, guarda também como variáveis o email, morada, coeficiente fiscal, sectores de actividade económica e as facturas. Estes últimos dois atributos são vistos de forma diferente dependendo da subclasse que a extende.

O `ContribuinteEmpresarial` acrescenta ao `Contribuinte` o `Concelho` (secção 2.4) onde reside, este podendo ser do interior ou não. Para além disto vê as suas facturas como as vendas que realizou e os sectores de actividade económica como os possíveis sectores a que essas facturas podem pertencer. Este processo de emissão de facturas será explicado em mais detalhe na seccão 3.

O `ContribuinteIndividual` acrescenta ao `Contribuinte` o agregado familiar e define quantos dos seus elementos são dependentes. Este vê as suas facturas como as suas despesas e os sectores económicos como o conjunto de despesas de onde pode deduzir, tendo a capacidade de alterar o sector económico das suas facturas.

Mais tarde foi também adicionada a noção de **família numerosa**, que introduz a noção de famílias com mais apoio social, sendo que implementam um método que permite reduzir o imposto pago.

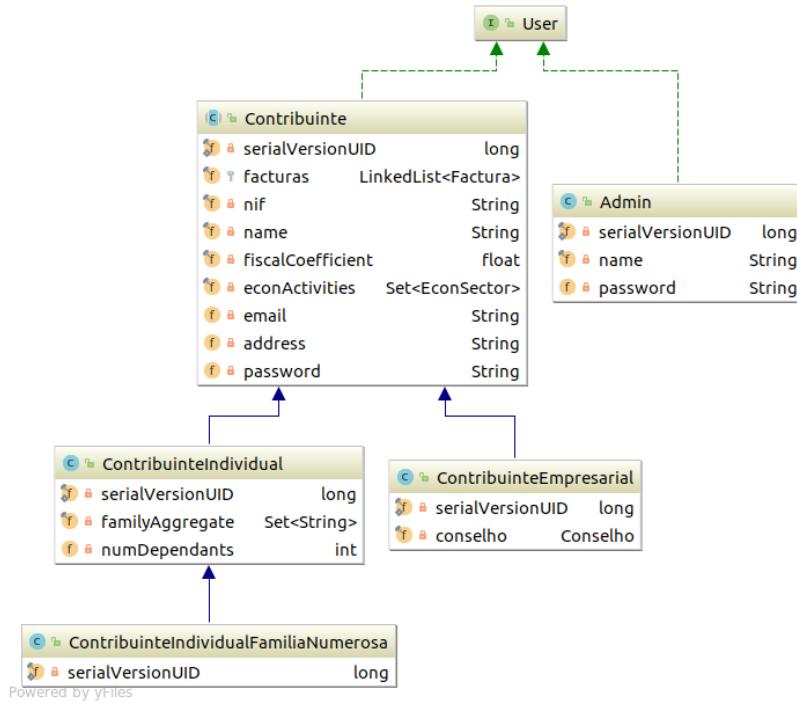


Figura 2.5: Hierarquia de Classes do User

Concelhos

Os concelhos disponíveis são mantidos pela aplicação na forma de um `enum` que associa a cada um um `boolean` a indicar se é um concelho do interior ou não. Desta forma simplifica-se bastante o processo de distinção das empresas que são ou não do interior. Para além disto, devido ao método como vai ser utilizado, não compromete a extensibilidade do código, sendo apenas necessário adicionar novas entradas ao `enum`, e toda a aplicação mantém o um comportamento correcto.

```

public enum Concelho {
    PORTO(false),
    BEJA(true);
    private boolean interior;
    Concelho(boolean interior){
        this.interior = interior;
    }
    public boolean isInterior(){
        return this.interior;
    }
}

```

Figura 2.6: Excerto do `enum` Concelho

Diagrama de classes

Devido à dimensão e complexidade do diagrama este não pode ser apresentado numa só imagem, por isso este foi dividido em duas seções.

Logica de negocio

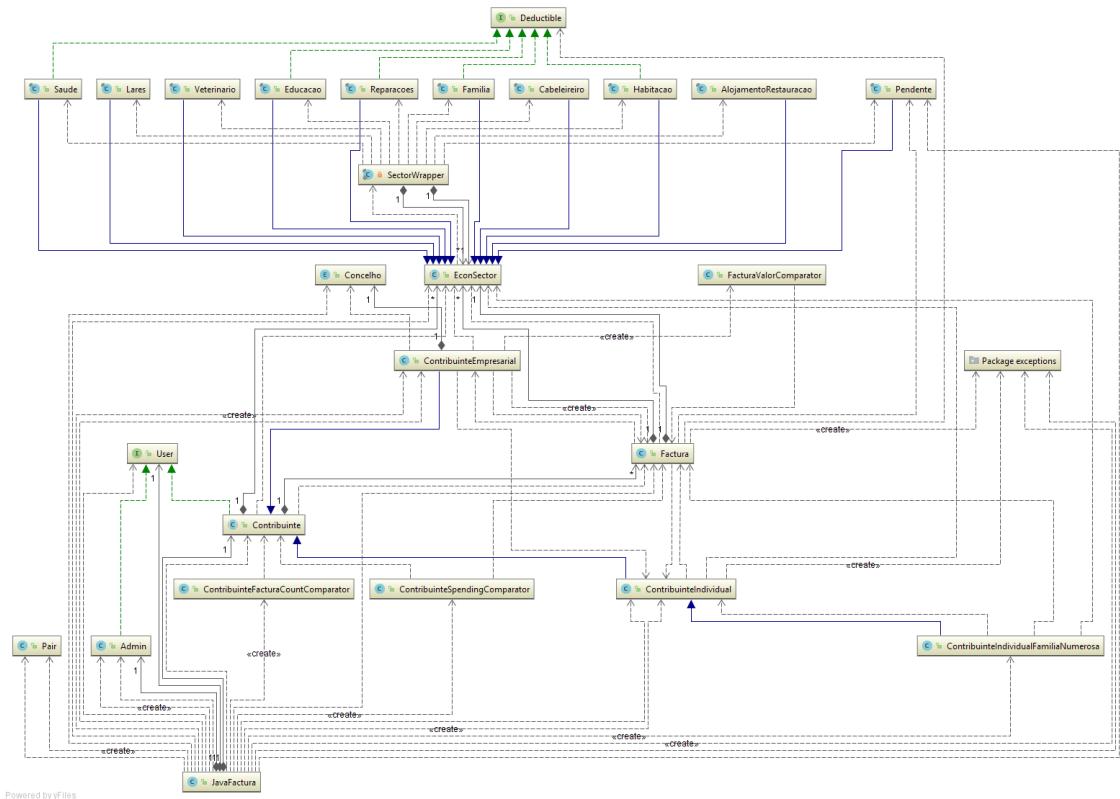


Figura 2.7: Logica de negocio

Interface grafica

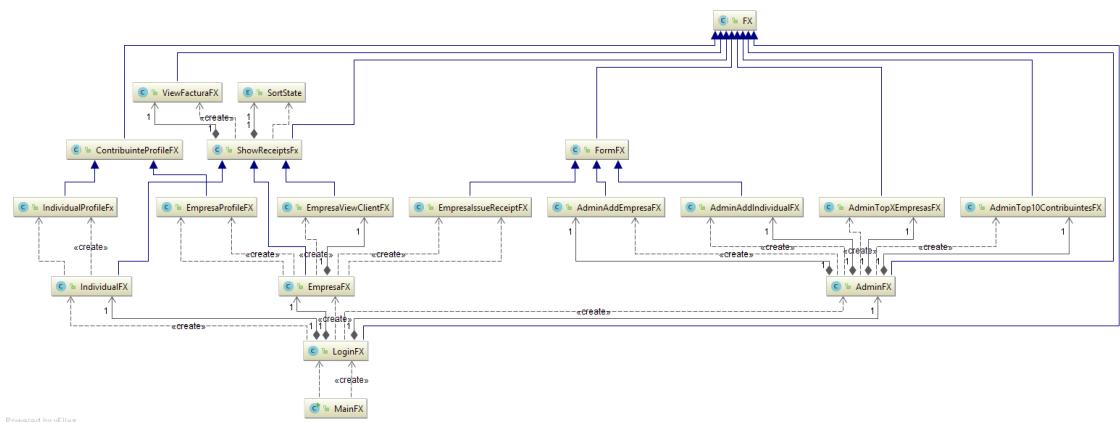


Figura 2.8: Interface grafica

Emissão de Facturas e Calculo de Deduções

Emissão de Facturas

Para garantir a consistência do estado interno da aplicação decidiu-se que, quando uma factura é emitida, tanto o emissor da factura como o cliente deviam guardar a mesma instância do objecto, assim, quando o cliente eventualmente alterar o sector económico da factura, esta alteração é automaticamente verificada pela empresa. Para que esta arquitetura não pudesse em causa o encapsulamento dos dados apenas instâncias de `ContribuinteIndividual` podem alterar facturas. Esta alteração é também muito pequena, tendo muito pouco impacto para a empresa, dados os requisitos pedidos.

Calculo de Deduções

A fórmula de cálculo de dedução, como foi referido anteriormente (secção 2.3), fica a cargo dos sectores económicos. Desta forma, para que uma despesa seja dedutível, a correspondente factura terá de ser de um sector `Deductible` e de este sector pertencer à lista de sectores para os quais o cliente está autorizado a deduzir. Um exemplo de uma destas fórmulas de cálculo pode ser vista na `Saude`:

```
public float deduction(float value, boolean interior, int numDependants,
                      float coeffEmpresa, float coeffIndividual){
    float deduction = 0;
    if(numDependants > 1) deduction += 0.3;
    if(interior) deduction += deduction * 1.5;
    deduction += coeffEmpresa + coeffIndividual;
    return value * deduction;
}
```

Figura 3.1: Fórmula de cálculo implementada para o Sector Saude

Interface Gráfica

Quando a aplicação inicia, o primeiro ecrã com que nos deparamos é o de *login*. Neste, se introduzir-mos as credenciais corretas, leva-nos a um dos três ecrãs principais.

The screenshot shows a 'Login' window. It has two input fields: 'NIF' containing 'admin' and 'Password' containing five black dots. Below the fields are two buttons: 'Sair' (gray) and 'Login' (blue outline). The window title is 'Login'.

Figura 4.1: Ecrã de *login*

Admin

O ecrã do administrador permite registar novos contribuintes, individuais ou empresariais, ver os 10 contribuintes individuais que gastaram mais dinheiro até ao momento ou ver os X contribuintes empresariais que mais facturas emitiram junto da quantidade monetária facturada. Este pode ainda alterar a sua password.

The screenshot shows a 'Bem vindo' (Welcome) window with several buttons: 'Add Contribuinte Individual' (highlighted with a blue border), 'Add Contribuinte Empresarial', 'Mudar password', 'Top 10 Contribuintes', 'Top X Empresas', and 'Log out'. The background is light gray.

Figura 4.2: Ecrã do Admin

Contribuinte Individual

O ecrã do contribuinte individual permite consultar o total deduzido por si e pelo seu agregado familiar, consultar as suas facturas ordenadas por data ou valor e consultar em detalhe cada factura (secção 4.4) e alterar o seu sector económico. Pode ainda consultar o seu perfil e alterar o seu email, morada e password.

The screenshot shows the 'Contribuinte Individual' interface. At the top right is a 'Perfil' button. Below it, there's a search bar labeled 'Procurar Empresa'. In the center, it displays '0 facturas pendente(s)', 'Total deduzido: 0,00', and 'Total deduzido pelo agregado familiar: 0,00'. There are two buttons: 'Ordenar por data' and 'Ordenar por valor'. Below these are two tables. The first table lists invoices with columns for Data, Sector, Empresa, and Valor. The second table lists clients with columns for NIF, Nome, and #Compras. At the bottom left is a 'Logout' button.

Data	Sector	Empresa	Valor
23-05-2018 20:50:26	Cabeleireiro	Um Oito	67,59
23-05-2018 20:21:33	Habitacao	Dois Quatro	195,95
23-05-2018 20:19:33	Saude	Oito	8,10

NIF	Nome	#Compras
7	Sete	1
9	Nove	1
31	Três Um	1

Figura 4.3: Ecrã do individual

Contribuinte Empresarial

O ecrã do contribuinte empresarial permite consultar o total facturado e as facturas emitidas, filtrar ambos por data e ordenar de forma análoga ao ecrã do contribuinte individual. Para além disto, permite emitir facturas novas e ver a lista dos seus clientes. Ao clicar num dos seus clientes pode ver todas as facturas que lhe foram emitidas e, aqui também, pode filtrar por data.

The screenshot shows the 'Contribuinte Empresarial' interface. At the top right is a 'Perfil' button. Below it, there's a 'Emitir Factura' button and two buttons: 'Ordenar por data' and 'Ordenar por valor'. On the left, a table lists invoices with columns for Data, Sector, Cliente, and Valor. On the right, a table lists clients with columns for NIF, Nome, and #Compras. At the bottom left is a 'Logout' button.

Data	Sector	Cliente	Valor
23-05-2018 20:18:43	Pendente	7	113,27
23-05-2018 20:18:43	Pendente	9	293,46
23-05-2018 20:18:43	Pendente	31	75,41

NIF	Nome	#Compras
7	Sete	1
9	Nove	1
31	Três Um	1

Figura 4.4: Ecrã do individual

Consultar uma Factura

A consulta de uma factura permite ver todas as alterações que a factura sofreu ao longo do tempo e, no caso de o utilizador autenticado ser um contribuinte individual, pode também alterar o sector económico desta.

NIF do Emitente:	2
Nome do Emitente:	Dois
Data de emissão:	23-05-2018 20:18:43
Data de edição:	23-05-2018 22:30:32
NIF do Cliente:	7
Descrição:	2 -> 7
Valor:	113.270256
Sector Económico:	Reparacoes

[Mudar Setor ▾](#)

[Mais Recente](#) [Mais Antiga](#) [Voltar](#)

Figura 4.5: Vista de uma Factura

Extensibilidade da aplicação

A modularidade da arquitetura concebida permite que seja facilmente estendida para incluir novos sectores económicos. Basta que seja adicionada uma classe que implemente o mesmo padrão *Singleton*, referido anteriormente (seccao 2.2), e adicionar esta à lista interna da super classe **EconSector**. Se esta for dedutível basta implementar a interface **Deductible**.

Conclusões e Trabalho Futuro

O trabalho realizado implementa todos os requisitos pedidos de forma estruturada, mantendo o encapsulamento dos dados.