



UNIVERSIDADE DO MINHO

MESTRADO INTEGRADO DE ENGENHARIA  
INFORMÁTICA

**Programação Orientada a Objectos**

*Grupo 73:*

Rui Monteiro

A67671



Bernardo Viseu

A74618



Fábio Silva

A82331



# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Classes</b>	<b>3</b>
2.1	FaturasData . . . . .	4
2.2	FaturasApp . . . . .	5
2.3	Menu . . . . .	5
2.4	Contribuinte . . . . .	5
2.5	ContribuinteIndividual . . . . .	5
2.6	Empresas . . . . .	6
2.7	Despesa . . . . .	6
2.8	ExceptionErroLogin . . . . .	6
2.9	ComparatorDataDespesa . . . . .	6
2.10	ComparatorValorDespesa . . . . .	7
<b>3</b>	<b>Desenvolvimento da aplicação</b>	<b>8</b>
3.1	Menu Inicial . . . . .	8
3.2	Menu Empresa . . . . .	10
3.3	Menu Contribuinte Individual . . . . .	14
3.4	Menu Admin . . . . .	16
<b>4</b>	<b>Discussão</b>	<b>19</b>
<b>5</b>	<b>Conclusão</b>	<b>20</b>

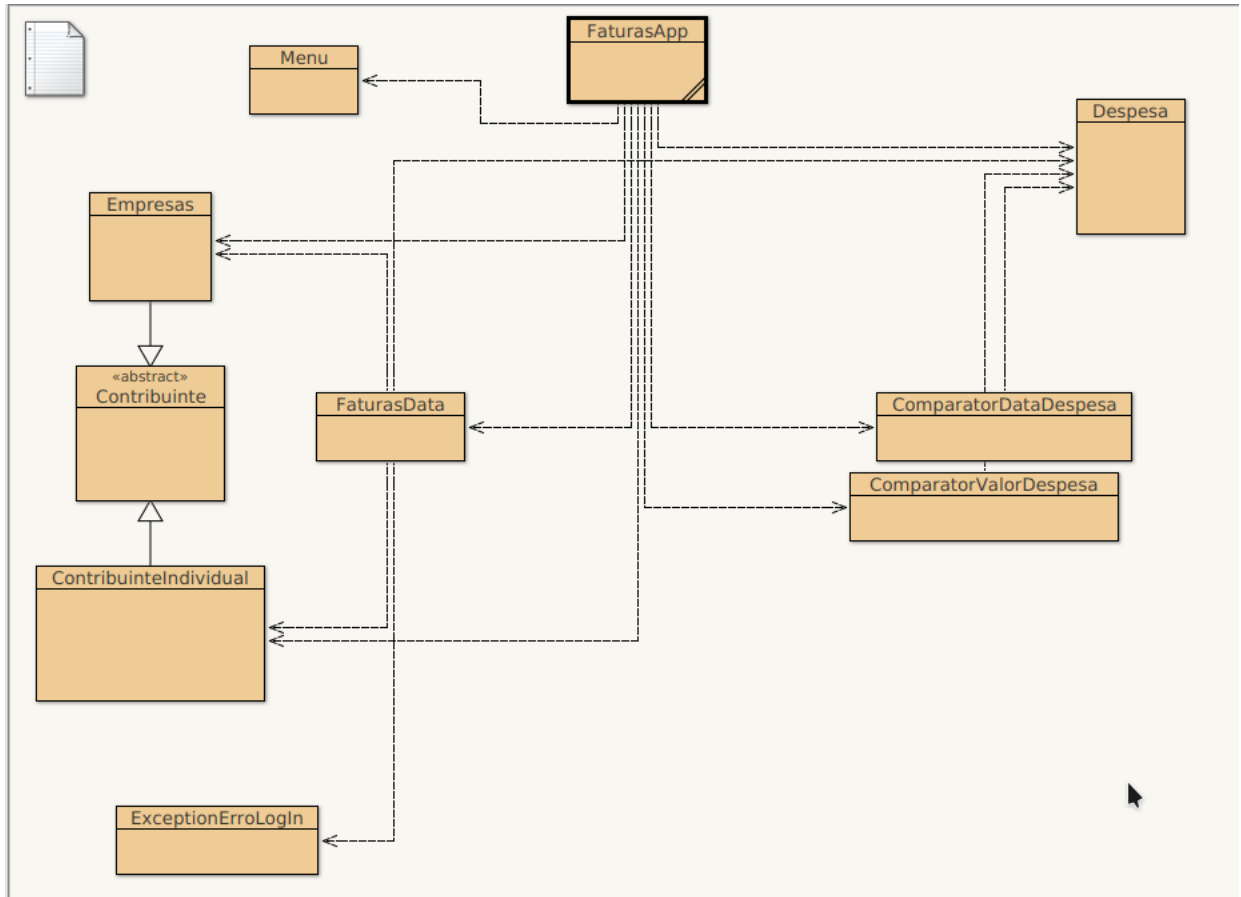
# 1 Introdução

No âmbito da disciplina de Programação Orientada a Objetos, foi-nos dada a oportunidade de criar um serviço de gestão de facturas nomeadamente chamado FacturasApp, recorrendo à linguagem de programação Java. Esta aplicação pode ser acedida por um utilizador identificado pelo seu respetivo número de contribuinte, seja ele um pessoa individual, empresa ou até mesmo o próprio administrador deste sistema.

Para além de ser possível aceder a todas as suas facturas emitidas (com o número de contribuinte associado), a aplicação também permite ao utilizador associar as facturas que não têm número de contribuinte associado ao respetivo sector. Sendo ainda possível visualizar e ter detalhe do cálculo associado, o valor acumulado de deduções fiscais de cada utilizador. Quanto às empresas é possível associar a factura de um serviço a um determinado contribuinte e organizar todas as faturas por, quer por data, quer por valor, de modo a que seja mais fácil organizar toda a informação, até para fins de gestão da própria empresa.

Esta aplicação, além de permitir aceder a todos os dados cujo nome de contribuinte foi associado, guarda os registos de todas as operações realizadas, ou seja, facturas emitidas por uma empresa, extrato de facturação de uma empresa num periodo de tempo decidido pelo utilizador, valor total das despesas de certo contribuinte tendo ainda mais funcionalidades que podem ser encontradas na leitura deste relatório. Sendo portanto uma aplicação capaz de gerir toda a vida económica de um contribuinte/empresa e tornando uma tarefa exaustiva mais acessível.

## 2 Classes



## 2.1 FaturasData

A classe FaturasData é onde se encontram todas as estruturas utilizadas no sistema para guardar os utilizadores (Contribuintes Individuais e Empresas). Usamos HashMap's (em que a key é um NIF) para guardar todos os contribuintes (uma HashMap para Empresas e outra para os Individuais), e usamos uma ArrayList para guardar as Despesas. Escolhemos esta estrutura pois foi a que achamos mais conveniente tendo em conta os dados que iamos ter e os requisitos do projecto. Nesta classe também se encontram varias funções essenciais para o programa, como os gets das estruturas, utilizadas para que o sistema possa analisar os dados destas em varias funções requisitadas, funções como addEmpresa(Empresa emp), para guardar uma empresa na estrutura, addIndividual(ContribuinteIndividual contr) e addDespesa(Despesa desp), para as adicionar nas respectivas estruturas.

Aqui encontra se métodos para iniciar sessao( iniciaSessao(int nif, String password) ) e terminar sessao ( logout() ), também varias funções relacionadas com o utilizador que esta atualmente logged in no sistema, como a checkAccount(), checkEmpresa(), checkIndividual() e checkAdmin(), que servem para identificar se existe um user logged in, e que tipo de user é (Individual, Empresa ou Admin), e também funções que permitem fazer get do nome do utilizador.

Nesta classe também estão as funções responsáveis por guardar o estado do programa num ficheiro (faturas.dados), utilizamos classes externas como ObjectOutputStream, e a partir destas guardamos todos os dados do sistema para que não se perca informação quando se sai do programa. Por fim temos o método log, que é utilizado para registar num ficheiro de texto (log.txt) o conteúdo actualizado das estruturas de dados.

## 2.2 FaturasApp

Esta classe é onde se encontra a "interface" que um utilizador do programa vê quando inicia a aplicação, e é aqui que se encontram as várias funções que analisam e modificam as estruturas de dados, e realizam os vários requisitos do projecto.

## 2.3 Menu

Esta classe tem os métodos necessários para apresentar o menu ao utilizador, e verificar a opção escolhida pelo utilizador.

## 2.4 Contribuinte

Esta é uma super classe abstrata. Como os contribuintes individuais e empresas têm várias informações em comum, decidimos que era melhor criar esta classe com essas informações, e depois ter as classes ContribuinteIndividual e Empresas com extend desta. Tem como variáveis de instância o NIF do utilizador (int NIF), email de utilizador (String email), nome (String nome), morada (String morada) e password do utilizador (String password).

## 2.5 ContribuinteIndividual

Esta é uma subclasse de Contribuinte, para além de já ter os atributos definidos nessa classe, também tem o número de dependentes da pessoa (int numeroDependentes), os números fiscais dos dependentes (ArrayList<Integer> numerosFiscais), o coeficiente fiscal do contribuinte (double coeficienteFiscal) e os códigos das actividades económicas do contribuinte, o nosso grupo decidiu que estes códigos seriam números, neste caso de 0 a 9, cada um deles representando uma actividade económica diferente (ArrayList<Integer> codigosActividades).

## 2.6 Empresas

Esta é uma subclasse de Contribuinte, para além de já ter os atributos definidos nessa classe, também tem o conjunto de actividades economicas da empresa (No caso dos individuais guarda se os codigos, aqui guardamos a String em si) (ArrayList<String> actividadesEconomicas) e o factor de dedução da mesma (double fatorDeducacao).

## 2.7 Despesa

Classe representante de cada despesa, tem como variáveis de instância o NIF do emitente (int NIFemitente), a designação do emitente (String designacaoEmitente), a data da despesa (Date dataDesp), o NIF do cliente (int NIFcliente), a descrição da despesa (String descricaoDesp), natureza da despesa (String naturezaDesp) e o valor da despesa (double valorDesp).

É de destacar que aqui existem métodos que, dado um código da actividade economica de uma despesa (ou a String), retorna o fator de dedução da mesma ( double getFatorDedCodigo(int codigo) e double getFatorDedString(String natureza) ), isto é necessario para depois calcular os valores dedutivos. Também existe um método para retornar a natureza da despesa em forma de código, em vez de String.

## 2.8 ExceptionErroLogin

Esta classe serve como maneira de indicar um erro a fazer log in, usamos um extend Exception para que possamos fazer throw deste erro. É utilizado em casos de password incorrecta, quando nao existe uma conta com o NIF registado, quando se tenta fazer login mas ja existe uma conta logged in, e quando se tenta registar uma conta com um NIF que ja esta no sistema.

## 2.9 ComparatorDataDespesa

Esta classe utiliza implements Comparator<Despesa>, para comparar duas despesas de maneira a dar sort de uma lista de despesas pela sua data, é utilizada para mostrar a uma empresa as suas faturas ordenadas por data.

## 2.10 ComparatorValorDespesa

Esta classe utiliza implements `Comparator<Despesa>`, para comparar duas despesas de maneira a dar sort de uma lista de despesas pelo seu valor, é utilizada para mostrar a uma empresa as suas faturas ordenadas por valor.



## 3 Desenvolvimento da aplicação

Com a intenção de permitir a um utilizador(cliente ou empresa) de usar a aplicação de maneira fácil e acessível. Para tal, a nossa interface possui apenas alguns menus(menu inicial, menu contribuinte individual, menu contribuinte de empresa e menu de administrador).

### 3.1 Menu Inicial

```
|=====| SISTEMA DE DESPESAS |=====|
1 -> Registrar Conta
2 -> Iniciar Sessão
0 -> Sair
Opção:
```

Este é o menu inicial que aparece na inicialização da aplicação, onde um utilizador pode se registrar ou fazer login.

```
|=====| SISTEMA DE DESPESAS |=====|
1 -> Registrar Conta Individual
2 -> Registrar Conta Empresas
0 -> Sair
Opção: 1
Email:
mail@exemplo.com
Nome:
Joao
Password:
pass123
Morada:
Morada
NIF:
123456789
Numero de dependentes:
1
NIF de dependente 1:
987654321
Coeficiente Fiscal:
0.15
Inserir codigos das Actividades Economicas.
Existem os seguintes codigos:
0 -> TRANSPORTES      1 -> CABELEIREIROS      2 -> ELECTRICIDADE      3 -> EDUCAÇÃO
4 -> GAS              5 -> SAUDE              6 -> AGUA              7 -> SERVIÇOS BANCARIOS
8 -> RESTAURAÇÃO     9 -> REPARAÇÃO DE VEICULOS
Insira qualquer outro numero para finalizar.
Codigo 1:
2
Codigo 2:
11
Done!
Adicionado Contribuinte Individual ao Sistema.
```

```

|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=| SISTEMA DE DESPESAS |=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|
1 -> Registrar Conta Individual
2 -> Registrar Conta Empresas
0 -> Sair
Opção: 2
Email:
mail2@exemplo.pt
Nome:
Google
Password:
passgoogle
Morada:
Morada2
NIF:
11111111
Inserir Actividades Economicas.
Existem as seguintes actividades disponiveis:
0 -> TRANSPORTES      1 -> CABELEIREIROS      2 -> ELECTRICIDADE      3 -> EDUCAÇÃO
4 -> GAS              5 -> SAUDE              6 -> AGUA              7 -> SERVIÇOS BANCARIOS
8 -> RESTAURAÇÃO     9 -> REPARAÇÃO DE VEICULOS
Insira qualquer outro numero para finalizar.
Codigo 1:
2
Codigo 2:
3
Codigo 3:
0
Codigo 4:
11
Fator de Dedução:
0.24
Done!
Adicionada Empresa ao Sistema.

```

## 3.2 Menu Empresa

```
|=====| SISTEMA DE DESPESAS |=====|
1 -> Registrar Conta
2 -> Iniciar Sessão
0 -> Sair
Opção: 2
NIF:
11111111
Password:
passgoogle

|=====| SISTEMA DE DESPESAS |=====|
--> Sessão iniciada: 'Google'.
1 -> Registrar Nova Fatura
2 -> Mostrar Lista Ordenada de Despesas
3 -> Mostrar Lista de Despesas por Contribuinte num intervalo de tempo
4 -> Mostrar Lista de Despesas por Contribuinte ordenadas pelo valor decrescente da Despesa
5 -> Mostrar valor total facturado num intervalo de tempo
6 -> Terminar Sessão
0 -> Sair
Opção:
```

Este é o menu que aparece caso seja efectuado o login por uma empresa. Nele o utilizador pode registar uma nova factura, mostrar ordenada de despesas, mostrar lista de despesas por contribuinte num dado intervalo de tempo (ordenadas por data ou por valor), mostrar lista de despesas por contribuinte ordenadas pelo valor decrescente da despesa, mostrar o valor total facturado num dado intervalo de tempo.

```

|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=| SISTEMA DE DESPESAS |=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|
--> Sessão iniciada: 'Google'.
1 -> Registrar Nova Fatura
2 -> Mostrar Lista Ordenada de Despesas
3 -> Mostrar Lista de Despesas por Contribuinte num intervalo de tempo
4 -> Mostrar Lista de Despesas por Contribuinte ordenadas pelo valor decrescente da Despesa
5 -> Mostrar valor total facturado num intervalo de tempo
6 -> Terminar Sessão
0 -> Sair
Opção: 1
Inserir NIF do Cliente:
123456789
Inserir designação da despesa:
Google
Inserir descrição da despesa:
PixelPhone
Inserir natureza economica da despesa:
Existem as seguintes naturezas economicas disponiveis:
0 -> TRANSPORTES      1 -> CABELEIREIROS      2 -> ELECTRICIDADE      3 -> EDUCAÇÃO
4 -> GAS               5 -> SAUDE              6 -> AGUA              7 -> SERVIÇOS BANCARIOS
8 -> RESTAURAÇÃO      9 -> REPARAÇÃO DE VEICULOS
3
Inserir valor da despesa:
599
Vai ser utilizada a data actual, ( Sun May 27 17:40:18 WEST 2018 )
Despesa adicionada com sucesso.

```

```

|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=| SISTEMA DE DESPESAS |=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|
--> Sessão iniciada: 'Google'.
1 -> Registrar Nova Fatura
2 -> Mostrar Lista Ordenada de Despesas
3 -> Mostrar Lista de Despesas por Contribuinte num intervalo de tempo
4 -> Mostrar Lista de Despesas por Contribuinte ordenadas pelo valor decrescente da Despesa
5 -> Mostrar valor total facturado num intervalo de tempo
6 -> Terminar Sessão
0 -> Sair
Opção: 2
Pretende ordenar as despesas por data, ou por valor?
1 -> Data
2 -> Valor
2

NIF Emitente = 11111111
Designacao do Emitente = Google
Data da Despesa = Sun May 27 17:40:18 WEST 2018
NIF Cliente = 123456789
Descricao da Despesa = PixelPhone
Natureza da Despesa = EDUCAÇÃO
Valor da Despesa = 599.0

```

```

|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=| SISTEMA DE DESPESAS |=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|
--> Sessão iniciada: 'Google'.
1 -> Registrar Nova Fatura
2 -> Mostrar Lista Ordenada de Despesas
3 -> Mostrar Lista de Despesas por Contribuinte num intervalo de tempo
4 -> Mostrar Lista de Despesas por Contribuinte ordenadas pelo valor decrescente da Despesa
5 -> Mostrar valor total facturado num intervalo de tempo
6 -> Terminar Sessão
0 -> Sair
Opção: 3
Indique a data inicial (DD/MM/YYYY):
20/10/2017
Indique a data final (DD/MM/YYYY):
20/10/2018

---->Despesas do NIF Cliente 123456789 :

NIF Emitente = 11111111
Designacao do Emitente = Google
Data da Despesa = Sun May 27 17:40:18 WEST 2018
NIF Cliente = 123456789
Descricao da Despesa = PixelPhone
Natureza da Despesa = EDUCAÇÃO
Valor da Despesa = 599.0

```

```

|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=| SISTEMA DE DESPESAS |=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|
--> Sessão iniciada: 'Google'.
1 -> Registrar Nova Fatura
2 -> Mostrar Lista Ordenada de Despesas
3 -> Mostrar Lista de Despesas por Contribuinte num intervalo de tempo
4 -> Mostrar Lista de Despesas por Contribuinte ordenadas pelo valor decrescente da Despesa
5 -> Mostrar valor total facturado num intervalo de tempo
6 -> Terminar Sessão
0 -> Sair
Opção: 4

---->Despesas do NIF Cliente 123456789 :

NIF Emitente = 11111111
Designacao do Emitente = Google
Data da Despesa = Sun May 27 17:40:18 WEST 2018
NIF Cliente = 123456789
Descricao da Despesa = PixelPhone
Natureza da Despesa = EDUCAÇÃO
Valor da Despesa = 599.0

```

```
|=====| SISTEMA DE DESPESAS |=====|
--> Sessão iniciada: 'Google'.
1 -> Registrar Nova Fatura
2 -> Mostrar Lista Ordenada de Despesas
3 -> Mostrar Lista de Despesas por Contribuinte num intervalo de tempo
4 -> Mostrar Lista de Despesas por Contribuinte ordenadas pelo valor decrescente da Despesa
5 -> Mostrar valor total faturado num intervalo de tempo
6 -> Terminar Sessão
0 -> Sair
Opção: 5
Indique a data inicial (DD/MM/YYYY):
20/10/2017
Indique a data final (DD/MM/YYYY):
20/10/2018
Valor total faturado no intervalo de tempo dado: 599 .
```

### 3.3 Menu Contribuinte Individual

```
|=====| SISTEMA DE DESPESAS |=====|
1 -> Registrar Conta
2 -> Iniciar Sessão
0 -> Sair
Opção: 2
NIF:
123456789
Password:
pass123

|=====| SISTEMA DE DESPESAS |=====|
--> Sessão iniciada: 'Joao'.
1 -> Ver despesas e montante de dedução fiscal
2 -> Alterar Classificação de uma Despesa
3 -> Terminar Sessão
0 -> Sair
Opção:
```

Este menu aparece caso o contribuinte faça login com uma conta individual. Nele o utilizador poderá ver as despesas e o montante de dedução fiscal ou alterar a classificação de uma despesa.

```
|=====| SISTEMA DE DESPESAS |=====|
--> Sessão iniciada: 'Joao'.
1 -> Ver despesas e montante de dedução fiscal
2 -> Alterar Classificação de uma Despesa
3 -> Terminar Sessão
0 -> Sair
Opção: 1

NIF Emitente = 11111111
Designacao do Emitente = Google
Data da Despesa = Sun May 27 17:40:18 WEST 2018
NIF Cliente = 123456789
Descricao da Despesa = PixelPhone
Natureza da Despesa = EDUCAÇÃO
Valor da Despesa = 599.0

Valor deduzido = 0.0.

A verificar faturas de dependentes do agregado...
A verificar faturas do contribuinte NIF = 987654321
NIF = 987654321 nao encontrado no sistema.
```

```

|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=| SISTEMA DE DESPESAS |=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|
--> Sessão iniciada: 'Joao'.
1 -> Ver despesas e montante de dedução fiscal
2 -> Alterar Classificação de uma Despesa
3 -> Terminar Sessão
0 -> Sair
Opção: 2

NIF Emitente = 11111111
Designacao do Emitente = Google
Data da Despesa = Sun May 27 17:40:18 WEST 2018
NIF Cliente = 123456789
Descricao da Despesa = PixelPhone
Natureza da Despesa = EDUCAÇÃO
Valor da Despesa = 599.0
Pretende mudar a classificação desta fatura?
1 -> Sim
2 -> Nao
1
Inserir nova classificação da despesa:
Existem as seguintes naturezas economicas disponiveis:
0 -> TRANSPORTES      1 -> CABELEIREIROS      2 -> ELECTRICIDADE      3 -> EDUCAÇÃO
4 -> GAS               5 -> SAUDE              6 -> AGUA              7 -> SERVIÇOS BANCARIOS
8 -> RESTAURAÇÃO      9 -> REPARAÇÃO DE VEICULOS
2

```



### 3.4 Menu Admin

```
|===== SISTEMA DE DESPESAS =====|
1 -> Registrar Conta
2 -> Iniciar Sessão
0 -> Sair
Opção: 2
NIF:
99999
Password:
passadmin

===== SISTEMA DE DESPESAS =====|
--> Sessão iniciada: 'ADMIN'.
1 -> Ver os 10 contribuintes que mais gastam
2 -> Ver as X empresas com mais faturas
3 -> Terminar Sessão
0 -> Sair
Opção:
```

Este menu aparece caso iniciada sessão como administrador, no qual que existem as opções de vêr os 10 contribuintes que mais gastam no sistema e ver as "X"empresas com mais facturas e o montante de dedução fiscal de cada uma delas.

```

|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=| SISTEMA DE DESPESAS |=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|
--> Sessão iniciada: 'ADMIN'.
1 -> Ver os 10 contribuintes que mais gastam
2 -> Ver as X empresas com mais faturas
3 -> Terminar Sessão
0 -> Sair
Opção: 1

NIF = 102
Nome = nomeInd102
Morada = moradaInd102
E-Mail = mailInd102
Password = pass102
Numero de dependentes = 1
Numeros Fiscais do agregado familiar = [101]
Coeficiente Fiscal = 0.12
Codigos das actividades economicas = [5, 8]
====>Gastos = 2341 .

NIF = 8
Nome = nome8
Morada = morada8
E-Mail = mail8
Password = pass8
Numero de dependentes = 1
Numeros Fiscais do agregado familiar = [7]
Coeficiente Fiscal = 0.17
Codigos das actividades economicas = [1, 2, 4]
====>Gastos = 1231 .

NIF = 10
Nome = nome10
Morada = morada10
E-Mail = mail10
Password = pass10
Numero de dependentes = 1
Numeros Fiscais do agregado familiar = [9]
Coeficiente Fiscal = 0.15
Codigos das actividades economicas = [5, 7, 6]
====>Gastos = 1203 .

```

```

|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=| SISTEMA DE DESPESAS |=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|=|
--> Sessão iniciada: 'ADMIN'.
1 -> Ver os 10 contribuintes que mais gastam
2 -> Ver as X empresas com mais faturas
3 -> Terminar Sessão
0 -> Sair
Opção: 2
Limite de empresas a mostrar:
2

NIF = 11
Nome = nome11
Morada = morada11
E-Mail = mail11
Password = pass11
Actividades Economicas = [TRANSPORTES, CABELEIREIROS, ELECTRICIDADE, EDUCAÇÃO, GAS, SAUDE, AGUA, SE
Fator de Dedução = 0.14
===>Numero de Faturas = 7 .
===>Montante de deduções fiscais = 5.6000000000000005

NIF = 33
Nome = nome33
Morada = morada33
E-Mail = mail33
Password = pass33
Actividades Economicas = [SAUDE, SERVIÇOS_BANCARIOS]
Fator de Dedução = 0.13
===>Numero de Faturas = 5 .
===>Montante de deduções fiscais = 1.9500000000000002

```

## 4 Discussão

Neste trabalho reparamos que existiam varias maneiras de abordar o projecto, e que isso se iria reflectir nas classes e subclasses do programa.

Nos utilizamos um sistema de codigos de números para representar as varias actividades económicas, ou seja, quando uma Empresa se regista, é apresentado uma lista das actividades economicas disponiveis, e o codigo de cada uma delas, e a Empresa insere os codigos respectivos, uma maneira alternativa seria utilizar Strings inseridas pela Empresa, mas reparamos que para os requisitos necessarios, era mais conveniente o uso de números, e depois utilizar métodos para converter esses códigos em Strings, que depois ficam guardadas nas informações de cada Empresa, e o mesmo acontece quando uma empresa regista uma fatura.

Outro caso de discussao deste projecto foi que estruturas utilizar para guardar os dados que iamos ter, decidimos utilizar HashMaps devido a ja termos experiencia com HashTables, mas uma alternativa seria utilizar TreeMap, isto no caso dos contribuintes individuais e Empresas, no caso de Despesas decidimos que ArrayList seria a melhor opção.

## 5 Conclusão

Com o desenvolvimento deste trabalho, podemos aplicar muito dos conhecimentos obtidos nas aulas de Programação Orientada a Objetos, aprendemos muito sobre a aplicação de estruturas e a utilização destas em Java, o uso de BlueJ foi-nos útil devido a ser possível ver as ligações entre as classes e subclasses. Ao longo deste trabalho encontramos alguns problemas que conseguimos resolver, no fim, o que também nos deu uma visão mais ampla do que é a disciplina de POO. Também concluímos que a linguagem de programação Java se trata de uma linguagem com funcionalidades práticas e acessíveis na criação de programas, e que se irá revelar uma ferramenta bastante prática e útil a nível profissional.