

Universidade do Minho  
Licenciatura em Engenharia Informática  
Programação Orientada a Objectos



## **Trabalho Pratico**

Ano Lectivo de 2009/2010

54738 **João Gomes**  
54745 **André Pimenta**  
54825 **Daniel Santos**

8 de Junho de 2010

## **Resumo**

Este relatório serve essencialmente para apresentar e justificar as escolhas feitas ao longo do desenvolvimento do projecto de gestão de seguros para a unidade curricular de POO. Será então apresentadas todas as escolhas das estruturas de dados e as suas funcionalidades, tentaremos também explicar cada Class do projecto.

# Conteúdo

<b>Conteúdo</b>	<b>1</b>
0.1 Motivação e objectivos . . . . .	1
0.2 Estrutura . . . . .	1
<b>1 Desenvolvimento</b>	<b>2</b>
1.1 Análise do Problema . . . . .	2
1.2 Solução do problema . . . . .	3
1.2.1 Seguros . . . . .	3
1.2.2 Clientes e Contratos . . . . .	6
<b>2 Conclusão</b>	<b>8</b>
<b>3 Elementos do Grupo</b>	<b>9</b>

## 0.1 Motivação e objectivos

Inquestionavelmente este trabalho pratico tem como principal objectico e motivação a aplicação de todos os conhecimentos adquiridos ao longo do semestre em POO, a utilização dos conhecimentos teóricos aplicados em situações praticas.

É também importante referir que só o uso do conhecimento em situações praticas é uqe surgem as verdadeiras duvidas e se assimila o conhecimento.

## 0.2 Estrutura

O presente relatório é constituído por três capítulos: **Introdução**; **Desenvolvimento**, dividido em duas secções, **Análise do problema** e **Solução do problema** e, por fim, a **Conclusão**.

# Capítulo 1

## Desenvolvimento

### 1.1 Análise do Problema

Para o desenvolvimento deste projecto foi pedido a existência de diferentes tipos de seguros disponíveis para venda por uma seguradora, estes são seguro de vida, seguro medico, seguro da casa com ou sem recheio, e seguro de automóvel, tanto de passageiros, mercadorias, ou transportes públicos.

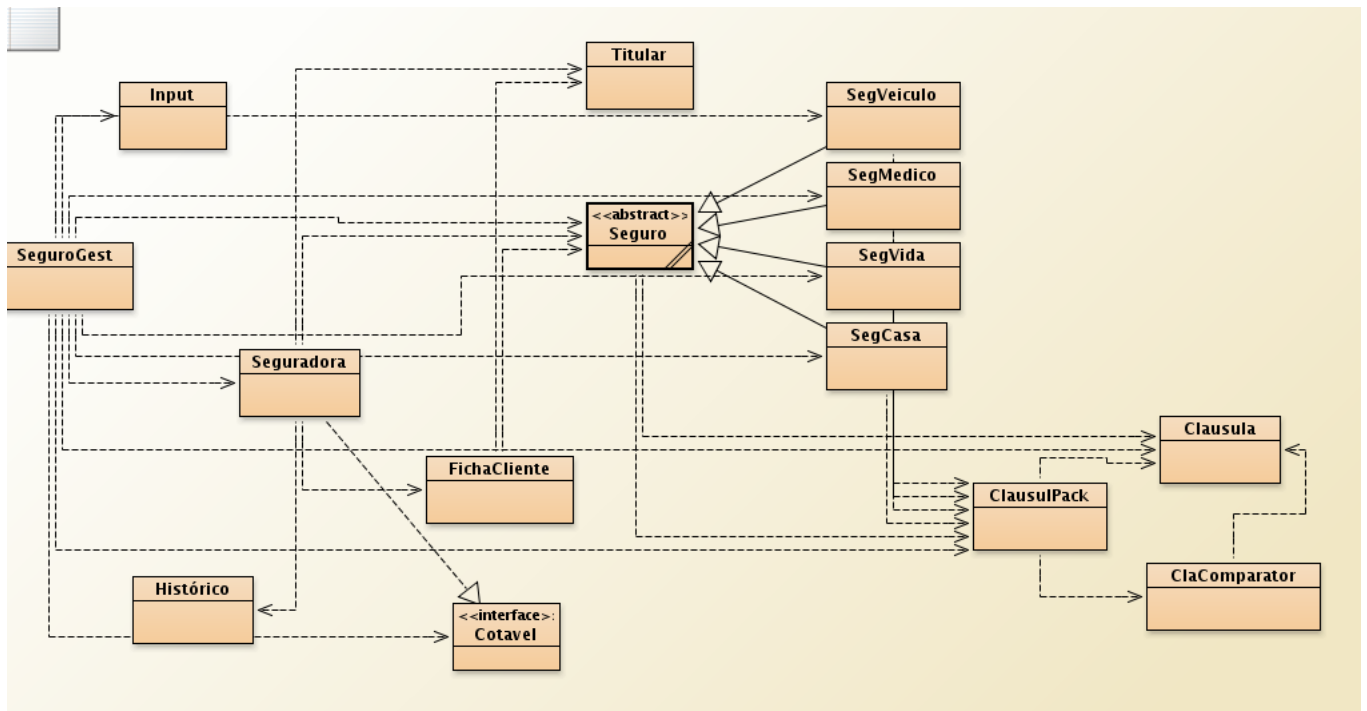
Sendo que os diferentes tipos de seguros não variam nos principais campos, tais como prémio base, código, pagamento em atraso, etc, decidimos criar uma superClass Seguro que se vai dividir em quatro subClass's da Class Seguro referidas anteriormente.

Relativamente aos titulares de seguros, criamos uma Class Titular que possui apenas os dados pessoas de um cliente ,dados estes que irão ser usados na Class FichaCliente.

Na class Fichacliente temos a informação pessoa de cada cliente proveniente da class Titular, mais o total a pagar por este e os seguros que este possui, sendo assim possível existir clientes com seguros de momento e clientes sem qualquer seguro, mas com registo na seguradora.

Chegamos agora a um ponto onde precisamos de conectar toda a informação disponível, para tal criamos a Class Seguradora que irá conter todas as Fichas de clientes ,todos os seguros disponíveis para venda, e informações de estatística tal como seguros vendidos e seguros por vender, valor em bolsa da seguradora e histórico de clientes.

Desta forma pensamos que na generalidade conseguimos cobrir todas as necessidades da seguradora e dos clientes.



**Figura 1.1:** Projecto em Ambiente BlueJ

## 1.2 Solução do problema

Após a exposição dos principais problemas , vamos expor as nossa solução para tais problemas. De forma sucinta podemos começar por dizer que para guardar os dados quer dos seguros quer dos clientes usamos Maps, pois esta Collection permite-nos ter os dados ordenados, ter bons tempos de execução, e não permite a existência de dados repetidos, e sempre que se revela necessário apresentar dados, usamos a Collection Set, pois evita dados repetidos e apresenta os valores pela ordem que são inseridos.

Para dar explicar o melhor o funcionamento do projecto vamos explicar as Class's e ideias com maior importância dentro do projecto.

### 1.2.1 Seguros

Os nossos seguros estão divididos em quatro tipos:

Seguro automóvel  
Seguro de Casa  
Seguro de Vida  
Seguro Médico

## Seguro Automóvel

Sabíamos de início que o seguro automóvel poderia ser de diferentes tipos mas vimos que isso não implicava que houvessem grandes discrepâncias entre os diferentes tipos, pois em quase todos os casos os veículos poderiam ter as mesmas cláusulas. Por exemplo quebra vidros poderia ser aplicado a veículos ligeiros, veículos mercadorias, veículos transporte público. O mesmo acontece para incêndio, furto, etc. Assim decidimos juntar todos esse tipos na mesma Class veículo sendo que essa contém uma String que identifica o tipo do veículo.

Assim temos as seguintes variáveis de instância para o tipo de seguro automóvel:

- marca - marca do veículo
- idade - idade do veículo
- valorComercial - valor actual de mercado do veículo segurado.
- TipoVeiculo - tipo do veiculo segurado (como explicamos acima)

## Seguro Medico

A classe SegMedico é semelhante à classe SegAuto diferindo basicamente no segurado que passa a ter as seguintes características:

- nome - nome do segurado
- idade - idade do segurado
- morada - local onde habita o segurado
- doenças - doenças de que padece o segurado

## Seguro Vida

A classe SegVida possui as seguintes é também semelhante a todas as outras possuindo apenas um segurado diferente para o qual guardamos as seguintes informações:

- nome - nome do segurado
- idade - idade do segurado
- morada - local onde habita o segurado

## Seguro Casa

O seguro casa é também um seguro especial que poderia ser aplicado em casos em que os clientes querem protecção não só do imóvel mas também do seu recheio. Assim decidimos criar um conjunto de packs que prevêem essas situações.

Possuímos packs que protejam por exemplo contra catástrofes naturais mas também contra outros acontecimentos como por exemplo furtos. Assim podemos ter não só uma protecção exclusiva do recheio da casa ou uma protecção apenas do imóvel mas um conjunto das duas que é uma hipótese que faz sentido.

## Packs de cláusulas

Escolha de variáveis de classe

Todas as classes dos diferentes tipos de seguros possuem um conjunto de packs de cláusulas, packs esses que estão armazenados em TreeMaps com uma correspondência <String, Clausul-Pack>. Escolhemos esta solução pois permite que a qualquer momento possamos aceder aos pacotes de uma forma rápida pois eles estão ordenados por código.

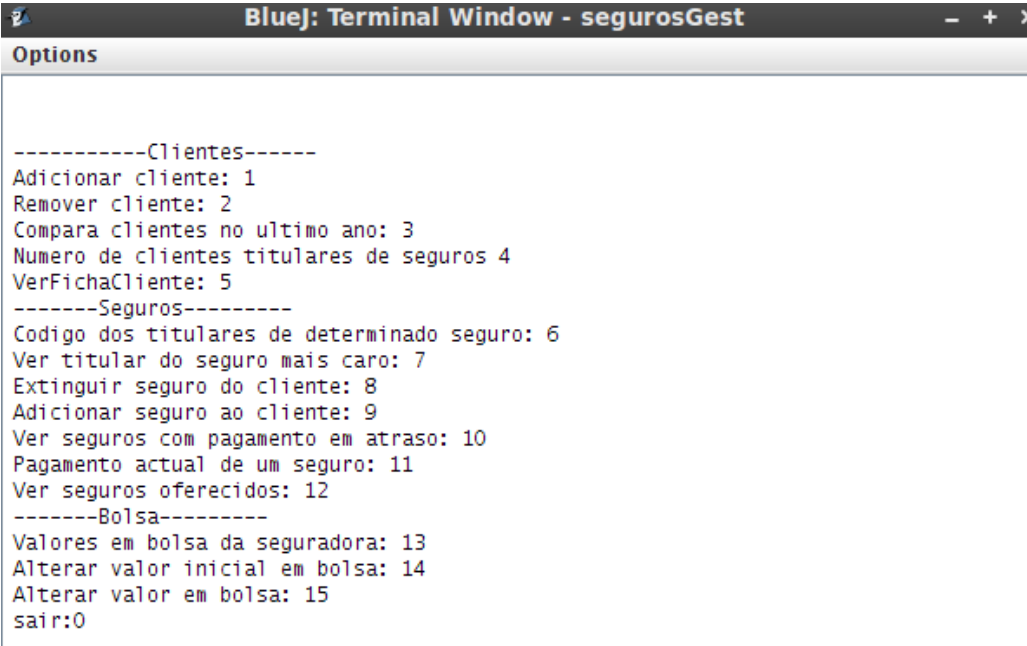
As colecções em que guardamos a colecção dos pacotes disponíveis para um determinado seguro são variáveis de classe. Esta é uma solução que permite uma grande facilidade por exemplo na obtenção de um pacote de cláusulas visto que todas as classes dos diferentes tipos de seguro possuem uma variável de classe que é este TreeMap com esse conjunto de packs. Assim ao criar uma instância de um seguro basta que seja passado como argumento um código do pack que o utilizador pretende adicionar ao seu seguro. Como o método é sempre o mesmo em todas as classe isso facilita a implementação na inserção de um novo utilizador pois não temos de testar qual é o tipo de seguro que o utilizador quer e a partir daí ir buscar os packs desse tipo de seguro.

As instâncias de ClausulPack possuem os seguintes as seguintes variáveis de instância:

String código - contém o código do pack

String descrição - contém uma descrição do pack , normalmente indica quais as protecções que esse pack oferece.

Set<Clausula> - contém uma colecção de cláusulas que vão definir os agravamentos a fazer ao seguro e indicar as protecções que oferecem caso acontece o que elas prevêm ao segurado.



```
BlueJ: Terminal Window - segurosGest
Options

-----Clientes-----
Adicionar cliente: 1
Remover cliente: 2
Compara clientes no ultimo ano: 3
Numero de clientes titulares de seguros 4
VerFichaCliente: 5
-----Seguros-----
Codigo dos titulares de determinado seguro: 6
Ver titular do seguro mais caro: 7
Extinguir seguro do cliente: 8
Adicionar seguro ao cliente: 9
Ver seguros com pagamento em atraso: 10
Pagamento actual de um seguro: 11
Ver seguros oferecidos: 12
-----Bolsa-----
Valores em bolsa da seguradora: 13
Alterar valor inicial em bolsa: 14
Alterar valor em bolsa: 15
sair:0
```

**Figura 1.2:** Programa em execução

### 1.2.2 Clientes e Contratos

Para o tratamento de dados de clientes criamos duas Class fundamentais, a Class Titular se serve para criar um cliente e conter os seus dados pessoais deste, e Class FichaCliente que irá fazer a ligação entre a seguradora e o cliente, ou seja, vai ser na FichaCliente que vão estar guardados os dados do cliente mais os seguros que ele possui e toda a informação sob estes, para além da informação de quanto está a pagar actualmente à seguradora por ano.

Posto isto iremos apresentar de forma mais detalhada estas duas class que cobrem as necessidades dos clientes dentro da seguradora.

#### Class Titular

Esta Class é bastante simples apenas contem os campos correspondentes a dados pessoais dos possíveis clientes. A class contem as seguintes variáveis de instâncias:

```
private String nome;  
private int idade;  
private String morada;  
private String nif;  
private String bi;
```

Esta Class apenas se resume as dados pessoais de cada cliente;

#### Class FichaCliente

Esta Class tal como já foi dito anteriormente faz a ligação entre o cliente e os seguros.

Nesta Class temos os dados pessoais do cliente, os seguros que este comprou e possui organizados num Map<Seguro>, e total a pagar, que está a ser calculado sempre que se adiciona um novo seguro ou se actualiza o prémio deste.

Esta Class possui as seguintes variáveis de Class:

```
private Titular cliente;  
private TreeMap<String,Seguro> seguros;  
private double premioTotal;
```

#### Class Seguradora

Esta é a Class com maior importância ou melhor dizendo, onde se desenvolvem os métodos principais. Na class Seguradora vamos ter a colecção de todos os seguros oferecidos pela seguradora armazenados num Map.

Para saber quais os seguros vendidos e qual a quantidade, temos uma variável de instância vendidos que é um Map com o numero de vezes que estes foram vendidos.

Para guardar o total de clientes de cada ano que a seguradora possui, criamos uma class Histórico, onde se guarda o total de cliente de cada ano, o total de clientes é actualizado sempre que se adiciona ou remove um cliente à seguradora.

Para a cotação em bolsa da seguradora foi criada a interface Cotável que contem métodos que alteram os valores em bolsa da seguradora.



Falta agora explicar a forma como estão estruturados os clientes, para estes temos um Map que contem Fichas de cliente.

Com estas variavas de estâncias conseguimos controlar todas as necessidades pedidas.

## Capítulo 2

# Conclusão

Após a elaboração do trabalho pratico podemos dizer que conseguimos assimilar e compreender melhor alguns princípios e regras fundamentais da programação orientada aos objetos, tal com o principio do encapsulamento.

Podemos dizer também que o uso de SuperClass e subClass para além de interfaces neste projecto foi importante para uma melhor compreensão da forma de programar genericamente, de forma a que futuras alterações ao código possam ser feitas sem se ter a necessidade de grandes alterações. Graças a realização do trabalho pratico conseguimos compreender e dar melhor utilidade a todas as possibilidades e potencialidades da programação orientada aos objectos e neste caso , mais em concreto a linguagem JAVA.

## Capítulo 3

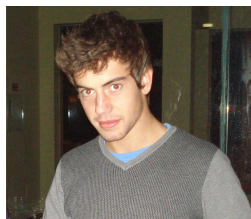
# Elementos do Grupo



**Figura 3.1:** João Miguel



**Figura 3.2:** André Pimenta



**Figura 3.3:** Daniel Santos