



A89583 - Bruno Filipe de Sousa Dias | A89513 - Francisco Alves Andrade | A89597 - Luís Enes Sousa

## **Programação Orientada Objetos**

Projeto - Grupo 25

# Introdução

Este trabalho foi realizado no âmbito da Unidade Curricular de Programação Orientada aos Objetos. O trabalho passa pela implementação de uma aplicação que permita conjugar as actuais necessidades de entrega de encomendas a pessoas que estão confinadas nas suas habitações e requerem um serviço deste género.

A primeira fase do trabalho passaria pelo desafio de implementar este sistema na linguagem JAVA. Desta forma, este trabalho teria como função aumentar o conhecimento dos alunos na em JAVA, numa fase ainda um pouco crua da aprendizagem da linguagem, incentivando sempre os alunos a respeitarem os paradigmas da POO (Programação Orientada a Objetos) e consolidarem os conhecimentos obtidos ao longo do semestre.

Além dos aspetos referidos, o trabalho tinha também como focos prioritários a modularidade, o encapsulamento de dados, o código reutilizável (essenciais para uma boa prática da Programação Orientada a Objetos) e também o MVC (Modelo Vista Controlador).

Uma parte muito importante do trabalho foi a análise prévia e cuidada de como poderiam ser implementadas as diferentes estruturas. No entanto, ao longo da realização do trabalho, foram surgindo pormenores que teriam de ser ajustados de forma a conseguirmos alcançar todos os objetivos do Projeto.

## Descrição da Estrutura do Projeto

A arquitetura do software é definida por 5 entidades principais: As Encomendas (onde estão contidas também Linhas de Encomendas), os Utilizadores, as Lojas, os Voluntários e as Empresas Transportadoras (que estendiam os Voluntários e lhes acrescentavam ainda mais informação). Todas estas entidades se encontram interligadas, por várias razões diferentes. Qualquer entidade é fundamental e imprescindível para o bom funcionamento do serviço e da aplicação em geral implementada neste Projeto.

Os dados utilizados por todas estas entidades são carregados a partir de um ficheiro de logs fornecido pelos docentes. No entanto, dentro da aplicação temos opções para registar novas entidades, sejam elas quais forem. O Usuário da Aplicação tem ainda a opção de guardar o estado da aplicação em disco e ainda de dar load a esses dados e a esse estado de aplicação guardado.

# Encomenda

A Encomenda é talvez o ponto principal no qual roda todo o Projeto, uma vez que, como referido anteriormente, este Trabalho passaria essencialmente pelo desenvolvimento de uma aplicação de Gestão de Encomendas.

Assim cada Encomenda terá dois tipos de dados: Dados imutáveis e Dados que vão sendo alterados ao longo do Projeto.

Dos dados Imutáveis fazem parte:

- Código da Encomenda, que é um Identificador da Encomenda;
- Código da Loja onde a encomenda foi requerida, que posteriormente terá de ser aceite ou recusada;
- Código do Utilizador que requereu essa Encomenda e para o qual se destina a mesma;
- O peso total da Encomenda;
- Uma Lista de Linhas de Encomenda onde estão contidos todos os Produtos de uma dada Encomenda e suas descrições;
- Um booleano que indica se a Encomenda em questão é médica, isto é, se possui algum produto de teor médico;
- Uma data que remete ao momento em que a Encomenda foi requerida por parte do Utilizador;

Dos dados Mutáveis fazem parte:

- Código de Transportador, que será atribuído à Encomenda aquando da entrega da mesma;
- Um booleano que indica se a Encomenda já foi aceite pela devida Loja;
- Um booleano que indica se a Encomenda já foi entregue ao seu Utilizador;
- A distância percorrida na realização da entrega;
- O custo da entrega desta Encomenda;
- Tempo despendido na sua entrega;
- Condições em que a entrega da Encomenda foi feita;

## Linha Encomenda

Como auxiliar à Encomenda, temos uma estrutura Linha Encomenda, que terá cada Produto de uma Encomenda. Aqui todos os dados são imutáveis no programa desde que o pedido da Encomenda a que pertencem é feito. Aqui vamos ter os seguintes dados:

- Código do Produto da Linha de Encomenda;
- A descrição do Produto indicado;
- A quantidade de Produtos do mesmo tipo que pretendem ser encomendados;
- O valor por Unidade do Produto em estudo;

# GPS - Sistema de Coordenadas

Antes de começarem a ser apresentadas as diferentes entidades, apresentar-se-á a estrutura GPS, que será uma estrutura auxiliar nas entidades que se irão falar a seguir. Esta estrutura tem, como o nome indica, as informações ligadas à localização de uma dada Entidade. A sua estrutura contém dois parâmetros: a Latitude e a Longitude. Os seus objetivos principais serão, como iremos ver mais á frente, o cálculo de distâncias entre Localizações de entidades e ainda a averiguação da possibilidade ou impossibilidade de alcance de certas localizações. Para esta última teremos sempre em conta o ponto de partida e o raio limite de movimento (não podendo este ser excedido).

Inicialmente utilizamos referências de distâncias globais (como medir distâncias entre latitudes e longitudes num globo, como no planeta Terra), no entanto, e devido à obtenção de distâncias de escala abominante comparativamente aos dados dos logs recebidos, este cálculo de distâncias foi visto como um plano XY. No entanto, a maneira inicial de calcular distâncias encontra-se igualmente disponível no Projeto sobre a forma de comentário e basta retirar o comentário que todo o Sistema se irá adaptar a essa forma de cálculo de distâncias.

## Utilizador

O Utilizador é uma das entidades desta aplicação na qual o Usuário pode dar login e realizar um conjunto de tarefas para o qual está preparado. Assim, tal como todas as outras estruturas, o Utilizador vai ter dados importantes e imprescindíveis que lhe permitem realizar diferentes tarefas. Destes fazem parte:

- O Nome do Utilizador;
- O código que é utilizado como identificador do Utilizador;
- Coordenadas da Localização do Utilizador;
- Password para a efetivação do seu login;
- Map com informações de entregas de encomendas por Avaliar (Código da Encomenda, Tempo demorado na Entrega e Custo da Entrega);
- Set com códigos de Encomendas pendentes (por entregar ou por aceitar por parte das Lojas);
- Map com informações de pedidos de entregas de encomendas por parte das Transportadoras por Aceitar (Código da Encomenda, Estimções de Tempo de demora e Custo da Entrega);
- Map com todas as Encomendas que foram Entregues ao Utilizador;

Como vimos anteriormente, as Encomendas têm como destino um Utilizador em específico, e desta forma é normal que também um Utilizador tenha a opção de realizar um pedido de uma Encomenda a uma dada Loja. Além de requerer essa Encomenda, ele deverá, aquando da sua entrega, classificar a entidade que a realizou, seja esta um Voluntário ou uma Transportadora (esta opção aparece como forma de “Notificações” que o

Utilizador pode abrir e responder). No caso de uma Encomenda ser entregue por uma Transportadora, ao fim de esta realizar um pedido de Entrega a Encomenda irá ficar em stand-by até que o Utilizador autorize ou recuse a entrega da mesma por parte da Transportadora (a opção de recusar ou aceitar irá aparecer, também, como forma de “Notificação” ao qual Utilizador poderá aceder). Esta função existe uma vez que a entrega por parte de uma Transportadora tem um custo associado, e o Utilizador pode não estar disposto a pagar, recusando consequentemente o pedido de Entrega da Transportadora.

O Utilizador pode ainda listar os 10 Utilizadores com mais encomendas a si entregues por ordem decrescente. À semelhança das outras entidades, o Utilizador pode ainda listar todas as Entidades que se encontram também no Sistema e o seu próprio Histórico.

## Loja

A Loja é uma das entidades desta aplicação na qual o Usuário pode dar login e realizar um conjunto de tarefas para o qual está preparada. Assim, tal como todas as outras estruturas, a Loja vai ter dados importantes e imprescindíveis que lhe permitem realizar as diferentes Tarefas. Destes fazem parte:

- O Nome da Loja;
- O código que é utilizado como identificador da Loja;
- Coordenadas da Localização da Loja;
- Password para a efetivação do seu login;
- Booleano que indica se Loja tem fila de espera ou não;
- Set com códigos de Encomendas que foram pedidos pelos Utilizadores à loja em estudo e que esta tem de aceitar ou recusar;
- Set com códigos de Encomendas pendentes na Loja e que se encontram prontas para serem entregues aos Utilizadores;
- Map com todas as Encomendas que foram inicialmente pedidas a esta loja que foram Entregues aos seus Utilizador;

As Encomendas são então como um “stock” de uma Loja. Estas Lojas são então o local obrigatório no qual qualquer Entidade que realize a entrega da Encomenda tem de passar para executar a entrega da mesma (para entregar uma Encomenda tem de se passar em pelo menos uma loja). Assim, Loja tem então a funcionalidade de poder aceitar e colocar no seu stock uma Encomenda pedida por um dado Utilizador, ou recusar e não ter Encomenda disponível para entregas pedida pelo Utilizador.

Como todas as outras entidades, a Loja pode listar todos os dados das Entidades que se encontram também no Sistema e o seu próprio Histórico.

# Voluntário

O Voluntário é uma das entidades desta aplicação na qual o Usuário pode dar login e realizar um conjunto de tarefas para o qual está preparado. Assim, tal como todas as outras estruturas, o Voluntário vai ter dados importantes e imprescindíveis que lhe permitem realizar as diferentes Tarefas. Destes fazem parte:

- O Nome do Voluntário;
- O código que é utilizado como identificador do Voluntário;
- Coordenadas da Localização inicial do Voluntário;
- Password para a efetivação do seu login;
- Booleano que indica se Voluntário está preparado para entregar Encomendas Médicas;
- Booleano que indica se Voluntário está disponível para realizar entregas;
- Booleano que indica se Voluntário está disponível para realizar entregas Médicas;
- Uma classificação overall do Voluntário;
- Um número total de Classificações;
- Um número total de Entregas de Encomendas;
- Velocidade Média do Voluntário;
- Raio de Distribuição para o qual Voluntário consegue entregar Encomendas;
- Map com todas as Encomendas que foram inicialmente pedidas a esta loja que foram Entregues aos seus Utilizador;

O Voluntário vai ser uma das duas entidades responsáveis pela entrega das Encomendas aos Utilizadores. Assim Voluntário vai poder ter como opção realizar Entregas de Encomendas. No entanto, só irá poder realizar Encomendas onde quer a Loja onde se encontra em stock a Encomenda, quer o Utilizador que pediu e irá receber a Encomenda, se encontrem ambos no alcance do raio de Distribuição do Voluntário. Como o Voluntário tanto se pode mostrar Disponível para entregar Encomendas, como Indisponível, existe uma opção para dar mudar essa Disponibilidade sempre que o Voluntário quiser. Existem ainda uma opção para mudar a disponibilidade para entregar Encomendas que contenham produtos médicos e precisem de um cuidado especializado por parte das Transportadoras.

Como todas as outras Entidades, Voluntário tem ainda a opção de listar todas as Entidades presentes no Sistema e o seu próprio Histórico.

# Transportadora

A Transportadora é uma das entidades desta aplicação na qual o Usuário pode dar login e realizar um conjunto de tarefas para o qual está preparado. Assim, tal como todas as outras estruturas, a Transportadora vai ter dados importantes e imprescindíveis que lhe permitem realizar as diferentes Tarefas. No entanto a Transportadora é uma Entidade Especial, uma vez que a classe que a define dá extend a uma outra classe. Essa classe é o Voluntário. Desta forma, a Transportadora vai conter todos os dados que um Voluntário possui e ainda os seguintes:

- NIF (Número de Identificação Fiscal);
- O custo da Entrega feita pela Transportadora (custo por Km);
- Limite de Encomendas a entregar por parte Transportadora em simultâneo;

A Transportadora vai ser a outra Entidade responsável pela entrega de Encomendas aos Utilizadores. Assim, vai ter, tal como no Voluntário, uma opção de realizar a entrega de uma Encomenda, no entanto, a entrega não vai ser realizada imediatamente, sendo enviado antes um pedido de Entrega de Encomenda ao Utilizador que a requereu. Tal como no Voluntário e de forma semelhante ao explicado acima, Transportadora tem a opção de mudar a Disponibilidade não só para a Entrega de Encomendas (normais), mas também para a Entrega de Encomendas médicas, que requer um cuidado especial.

A Transportadora pode ainda listar até às 10 Transportadoras com mais Quilómetros feitos na realização de todas as suas Entregas. Além disso, e como Transportadora tem um custo associado à Entrega das Encomendas, ela pode ainda listar o seu total Faturado.

Mais uma vez, como todas as outras Entidades, a Transportadora pode também listar todas as outras Entidades que se encontrem no Sistema e o seu próprio Histórico.

# TrazAqui

TrazAqui é a Entidade superior que contém catálogos com todas as informações do Projeto. Desta forma a classe TrazAqui contém:

- Catálogo (Map) de Lojas que possui todas as Lojas do Sistema;
- Catálogo (Map) de Voluntários que possui todos Voluntários do Sistema;
- Catálogo (Map) de Transportadoras que possui todas as Transportadoras do Sistema;
- Catálogo (Map) de Utilizadores que possui todos os Utilizadores do Sistema;
- Catálogo (Map) de Encomendas que possui todas as Encomendas do Sistema;
- Lista com código de Encomendas que foram aceites mas ainda não foram Entregues;
- String com o Utilizador Atual da Aplicação;

O TrazAqui é então o centro de todo o Sistema no que toca a Dados e à parte do Model do MVC (Modelo Vista Controlador). Uma das maiores facetas deste Projeto passa por realizar Encomendas e consequentemente mudar os dados das mesmas. Primeiro é importante perceber que para evitar haver discrepância de Dados e de muitas dependências, as Encomendas encontram-se apenas no catálogo de Encomendas do TrazAqui. As Encomendas que existem fora desse contexto são as Encomendas que se encontram no Histórico das Entidades. No entanto, estas últimas nunca são alteradas, nem nos Históricos nem no Catálogo de Encomendas.

Assim quando vamos realizar operações vamos trabalhar sempre com cópias de Encomendas e se posteriormente quisermos que a Encomenda permaneça como nessa cópia, iremos substituir no Catálogo a Encomenda com o código de Encomenda da Cópia, pela Encomenda Cópia que temos neste momento. Desta forma vamos sempre trabalhar com cópias de Encomendas, e sempre que queremos remeter em qualquer Entidade, nas suas variáveis/dados auxiliares para uma Encomenda utilizamos sempre os Códigos das Encomendas ou alguns dados em particular da mesma e nunca Encomendas em si.

Vamos então ter o seguinte percurso de uma Encomenda:

- Utilizador faz um pedido de uma Encomenda a uma dada Loja, insere-a no Catálogo de Encomendas do TrazAqui e o seu código nas suas Encomendas Pendentes;
- Loja aceita ou recusa o pedido de Encomenda feito pelo Utilizador:
  - Caso aceite, vai mudar a Encomenda no Catálogo das Encomendas do TrazAqui e colocá-la como aceite pela Loja, colocando ainda o seu código respetivo na lista de Encomendas por Entregar na Loja ;
  - Caso recuse, vai remover a Encomenda do Catálogo de Encomendas e o código respectivo da lista de Encomendas Pendentes do Utilizador;
- No caso de a Loja ter aceite o pedido, a Encomenda vai então estar no Sistema pronta a ser entregue e distribuída por um Voluntário ou Por uma Transportadora. Agora um Voluntário ou uma Transportadora podem prontificar-se a entregar uma Encomenda, tendo de estar tanto a Loja, quando o Utilizador da Encomenda no raio de distribuição da Entidade que irá realizar a Entrega;
  - Caso o Voluntário se prontifique, a Encomenda é entregue imediatamente, sendo calculados os dados do transporte da mesma. Após isso, Encomenda é alterada com esses dados no Catálogo e é posta como entregue no Catálogo, sendo copiada para o Histórico da Loja, do Utilizador e do Voluntário, e o seu código é removido das variáveis das Entidades envolvidas na Encomenda e que a têm como pendente para ser entregue;
  - Caso a Transportadora se prontifique, a Encomenda não é entregue imediatamente. Os dados são calculados tal como no Voluntário e colocados na Encomenda no Catálogo. Posteriormente se o Utilizador recusar o Transporte por parte daquela Transportadora, a Encomenda dos Catálogo volta para os valores que envolvem entregas iniciais e volta tudo a como estava antes. Caso o Utilizador aceite, Encomenda é posta como entregue no Catálogo sendo copiada para o Histórico da Loja, do Utilizador e da Transportadora, e o seu código é removido das variáveis das Entidades envolvidas na Encomenda e que a têm como pendente para ser entregue;



# Diagrama de Classes

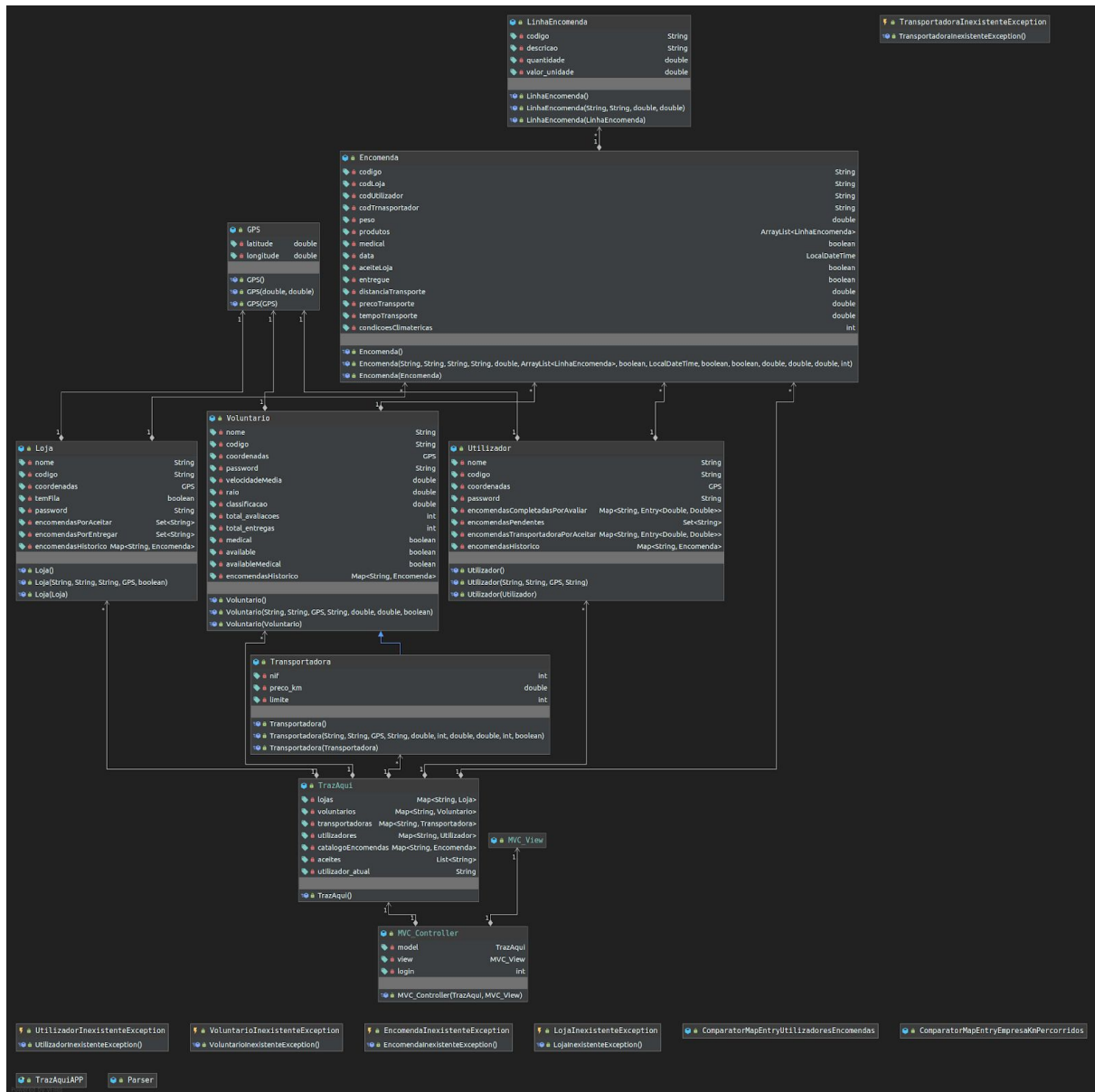
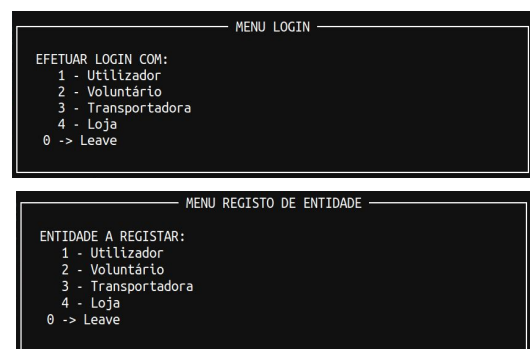
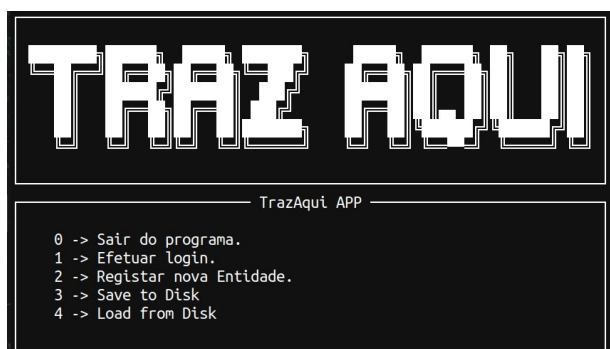


Diagrama de Classes gerado Automaticamente pelo IntelliJ.

# Funcionalidades das Entidades



```

MENU UTILIZADOR
0 -> Logout.
1 -> Listar entidades no sistema.
2 -> Fazer pedido de encomenda.
3 -> Avaliar Encomendas que foram Entregues (0).
4 -> Aceitar Entregas propostas por uma Transportadora (0).
5 -> Dez utilizadores ordenados com mais encomendas realizadas.
6 -> Mostrar Histórico.

```

```

MENU TRANSPORTADORA
0 -> Logout.
1 -> Listar entidades no sistema.
2 -> Fazer pedido para entregar encomenda.
3 -> Altera disponibilidade de entrega.
4 -> Altera disponibilidade de entrega médica.
5 -> Dez Transportadoras ordenadas com mais Km feitos.
6 -> Total faturado pelos Transportes da Transportadora.
7 -> Mostrar Histórico.

```

```

MENU VOLUNTÁRIO
0 -> Logout.
1 -> Listar entidades no sistema.
2 -> Fazer pedido para entregar encomenda.
3 -> Altera disponibilidade de entrega.
4 -> Altera disponibilidade de entrega médica.
5 -> Mostrar Histórico.

```

```

MENU LOJA
0 -> Logout.
1 -> Listar entidades no sistema.
2 -> Aceitar pedidos de Encomenda (0).
3 -> Mostrar Histórico.

```

## Conclusão

Concluído o trabalho, achamos que foram atingidos os objetivos do mesmo e foram retidos os pontos essenciais. Da utilização de Java recebemos um leque de funções e API's que podemos utilizar livremente, e assim fazer com que o uso das mesmas facilite a realização dos objetivos em mente e torne mais fácil o progresso até o mesmo ser alcançado. Conseguimos perceber que por vezes é difícil conseguir uma boa segurança dos dados (encapsulamento) por parte de terceiros, ou por vezes, até do próprio utilizador, sendo então este um ponto forte deste tipo de trabalho - a Segurança. Relembramos ainda a importância da reutilização de código e da generalização do mesmo, uma vez que com um bom código que respeitasse essas condições conseguir-se-ia reformular o trabalho depois de alterar as estruturas, apenas mudando alguns pontos chave em algumas funções de forma bastante simples, sendo este um parâmetro ainda mais acrescido quando se está a trabalhar com Paradigmas de POO (Programação Orientada a Objetos).