Classes e objetos

Na “modelação orientada por objetos”, procuramos identificar os conceitos do problema e representá-los como classes conceituais, ou tipos de “coisas” (= Classes).

A Classe é um “molde” do qual se podem obter “cópias” para guardar informação de ocorrências concretas. Estas ocorrência podem ser chamadas de instâncias ou objetos.

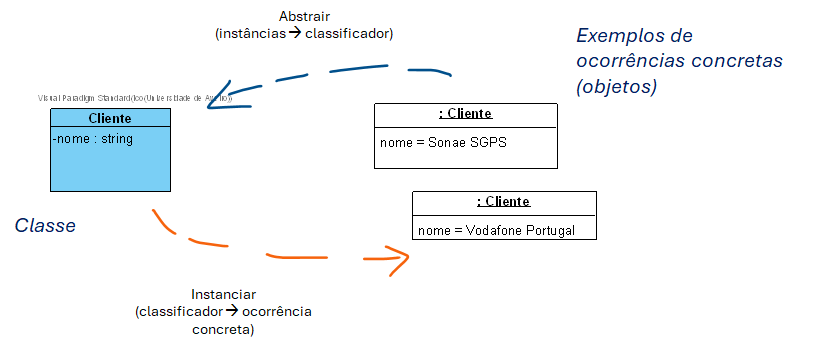
Considere a apresentação que se segue, por um responsável de uma (hipotética) empresa de consultadoria tecnológica:

* *A Sonae SGPS tem vários projetos connosco. Por exemplo, o projeto “Norte 5”, coordenado pelo Eng. Casimiro, e o projeto “Continente Wireless”, coordenado pela Dr. Sílvia, que é especilista em soluções IoT.*
* *A Sonae SGPS é o nosso melhor cliente, mas temos cerca de outros 10 grandes clientes em carteira.*
* *No projeto “Norte 5”, pretende-se monitorizar a distribuição espacial dos compradores em grandes superfícies, com técnicas de visão por computador (VC). O nosso colaborador Justino, que está no projeto, acredita que terá um grande impacto. Se o quiser contactar, pode usar o email “justino\_ninja@mymail.com”.*
* *O plano do projeto “Norte 5” está em preparação, mas prevê-se incluir as tarefas:*
  + *T1: Avaliação de frameworks de VC* 
    - *T1.1: Pesquisa e seleção de frameworks open-source*
    - *T1.2 Ensaio exploratório num Continente Bom Dia*
  + *T2: Levantamento de requisitos e cenários de utilização*

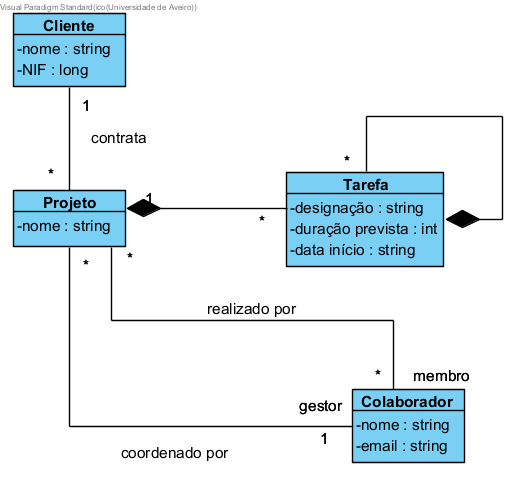
Construimos um “mapa” para os conceitos, ou modelo concetual, procurando:

* Encontrar conceitos/tipos de coisas que devem ser representados/memorizados num sistema de informação, e
* Abstrair as especificidade de cada exemplo/ocorrência, para nos focarmos nas propriedades gerais.

Por exemplo, existem muitos clientes; podem ser “classificados” pelo tipo “Cliente”, para os quais deve ser representado um nome.



Continuando o processo, e procurando identificar as associações:



O modelo (concetual, com classes) supõe que, mais à frente na implementação, haverá mecanismos (de programação) para:

* Criar/instanciar objetos quando forem necessários. E.g.: criar um objeto para representar um novo projeto.
* Ler e atualizar o estado em cada objeto. E.g.: podemos ler ou atualizar a informação de uma ocorrência de Cliente, a qualquer altura.
* Percorrer cada coleção de objetos. E.g.: listar os clientes, contar os projetos, filtrar projetos por uma caraterística (que esteja representada no modelo), etc.
* **Ligar objetos de acordo com as associações** (navegar entre objetos). E.g.: para um objeto de Projeto, é possível aceder ao respetivo objeto de Cliente.