



**Recursos Educacionais / Educational Resources**

**Arquitetura Empresarial:  
na perspetiva da integração de diagramas UML**

**Luís Cavique**  
DCeT, Univ. Aberta  
Luis.Cavique@uab.pt

**Lisboa, outubro 2020**



Este trabalho está licenciado com uma Licença Creative Commons  
Attribution-NonCommercial-ShareAlike CC BY-NC-SA

## Resumo

Este documento pretende complementar a bibliografia da UC de Modelação de Sistemas de Informação, oferecida no 2º semestre do 1º ano, na Licenciatura em Engenharia Informática.

Na bibliografia recomendada (Silva, Videira 2008 e O'Neill et al. 2010) são magistralmente apresentadas as ferramentas UML em capítulos distintos. Os capítulos referem com detalhe os diagramas de 'use-cases', os diagramas de classe, os diagramas de atividades, os diagramas de estados, os diagramas de sequência e os diagramas físicos.

Nesta abordagem pretende-se dar uma visão integrada do sistema de informação, interligando as ferramentas UML com a matriz CRUD, procurando encontrar alinhamentos verticais e validações horizontais do sistema. Servimo-nos do conceito de Arquitetura Empresarial para desenvolvimento do procedimento proposto.

O documento está estruturado da seguinte forma: (i) três camadas da Arquitetura Empresarial, (ii) diagramas UML associados à Arquitetura Empresarial, (iii) regras heurísticas do alinhamento da Arquitetura Empresarial, (iv) procedimento para modelar a Arquitetura Empresarial, finalizando com uma (v) visão integrada das ferramentas UML.

Em anexo são incluídos três exemplos da aplicação do procedimento criado que garante o alinhamento do sistema.

## 1- Três camadas da Arquitetura Empresarial

Quando se fala em Arquitetura Empresarial é comum referir o modelo das três camadas: a arquitetura do negócio, a arquitetura do sistema de informação e a arquitetura tecnológica, conforme representa a Figura 1.

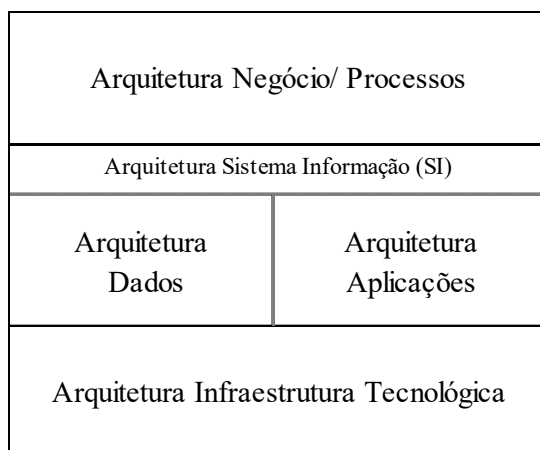


Figura 1 – Três camadas da Arquitetura Empresarial

Quando falamos em Arquitetura Empresarial existem muitos sinónimos dependendo das metodologias ou ferramentas utilizadas, pelo que se torna importante realizar uma síntese explicativa com um número reduzido de nomes.

Neste trabalho existem apenas três elementos essenciais, com vários sinónimos. Os elementos essenciais da Arquitetura Empresarial são: os atores, as atividades e os dados. Alguns dos sinónimos são os seguintes:

- atores = linhas de responsabilidades
- atividades = aplicações, requisitos, tarefas, 'uses-cases', processos-operacionais
- dados = classes, entidades-informacionais

Nos exemplos seguintes iremos utilizar atores ( $\alpha$ ,  $\beta$ ), as atividades (A,B,C) e os dados (X,Y,Z,W). Não resisto a mencionar a versão popular de arquitetura empresarial com três camadas:

|              |   |  |
|--------------|---|--|
| 'Peopleware' | Arq. Negócio/ processos                         | Os processos são constituídos por atividades (A,B,C) e geridos por atores humanos ( $\alpha$ , $\beta$ )             |
| Software     | Arq. Sistema de Informação (dados + aplicações) | Existem dois grandes grupos: os dados (X,Y,Z,W) e as aplicações (A,B,C)  |
| Hardware     | Arq. Infraestrutura tecnológica                 | Constituído pelas componentes físicas e software de base (sistemas operativos e sistemas gestores de bases de dados) |

## 2 - Diagramas associados à Arquitetura Empresarial

Neste trabalho pretendemos detalhar a arquitetura empresarial ao nível da modelação e análise de sistemas. A Figura 2 apresenta as três camadas da Arquitetura Empresarial associada a diagramas. Esquemáticamente teremos o diagrama dos processos, sequência de atividades, (em cima) na 1ª camada. Na 2ª camada, a lista das aplicações (no centro à direita), o diagrama de dados (no centro à esquerda), a matriz CRUD, aplicações versus dados (no meio). Finalmente, na 3ª camada, o diagrama da infraestrutura (em baixo).

Para representar o diagrama dos processos pode ser utilizado o BPMN (Business Process Model and Notation), um diagrama de atividades com partições ('swimlanes') em UML ou ainda uma versão simplificada num diagrama de casos-de-uso em UML.

Para a representação dos dados pode ser utilizado um diagrama Entidade-Relação ou um diagrama de classes em UML. Ainda na 2ª camada existe uma lista de aplicações e uma matriz CRUD (create, read, update, delete).

Finalmente para o diagrama da infraestrutura pode ser representado por um diagrama de arquitetura ou implementação em UML utilizando componentes e nós.

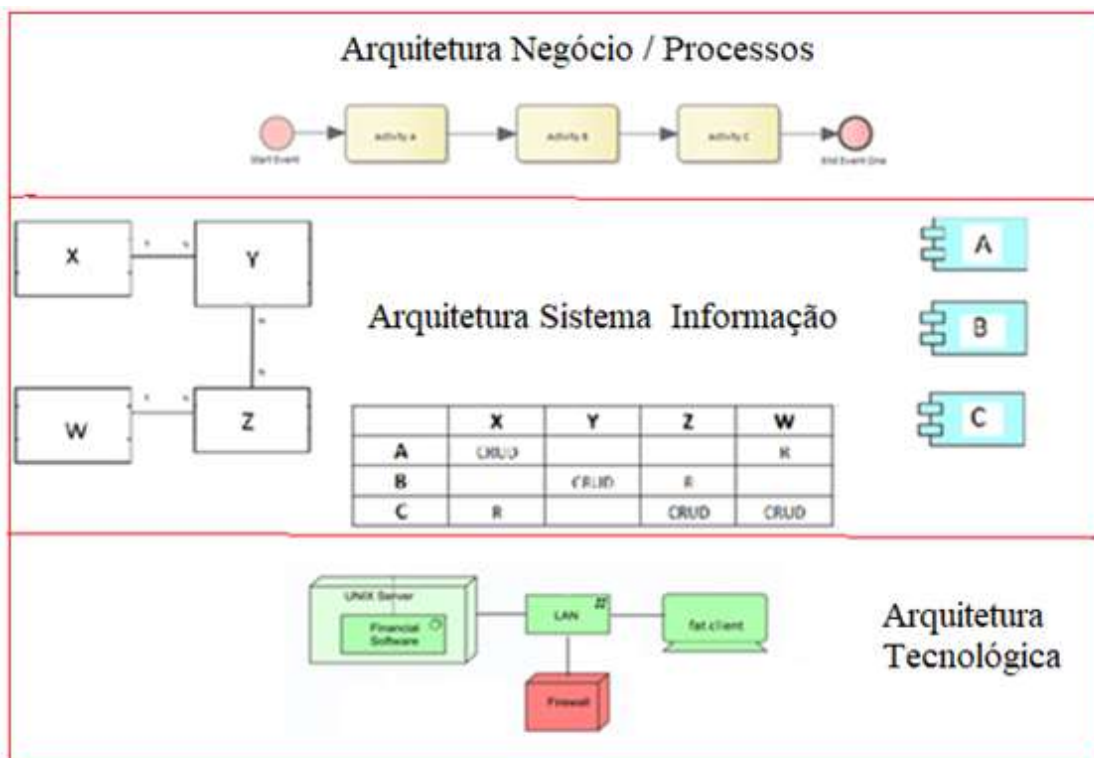


Figura 2 – Três camadas da Arquitetura Empresarial associada aos diagramas

### 3 – Regras heurísticas do alinhamento da Arquitetura Empresarial

O conjunto de diagramas devem estar alinhados, isto é, serem coerentes na sua articulação. Geralmente são considerados dois alinhamentos, o alinhamento vertical e o horizontal.

Toda a arquitetura tem início com uma descrição da área do negócio, aqui denominada por Narrativa da Área do Negócio. A referida narrativa pode ser obtida através de reuniões, 'focus groups', entrevistas (estruturadas ou não estruturadas) ou estudos de caso. A narrativa inclui geralmente um levantamento do sistema 'as-is' (passado e presente) e o sistema pretendido ou 'to-be' (futuro), associados a um conjunto de requisitos funcionais.

A análise do sistema considera duas visões possíveis: a visão do analista funcional que lida com os utilizadores finais orientada para a arquitetura de processos (1ª camada) e a visão do analista orgânico/sistemas orientada para a arquitetura dos dados (2ª camada).

Das atividades passíveis de serem automatizadas denominamos aplicações. O cruzamento dos dados com as aplicações é representado na matriz CRUD.

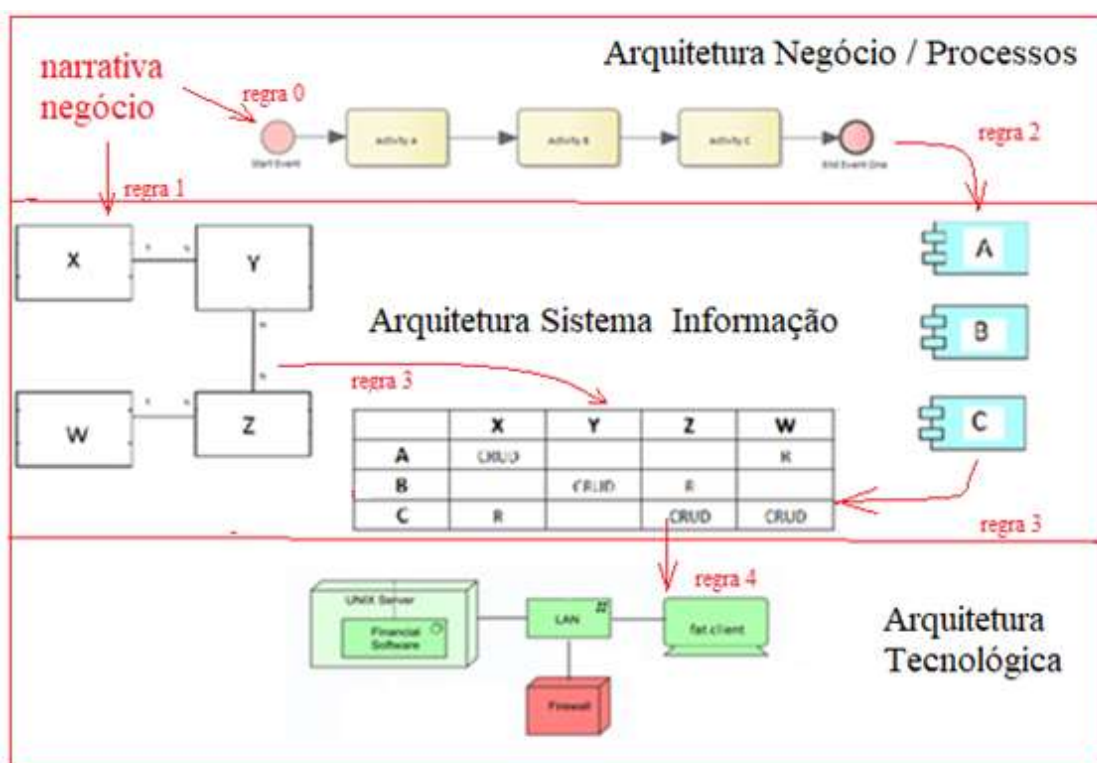


Figura 3 – Regras heurísticas do alinhamento da Arquitetura Empresarial

Os autores (Pereira, Sousa 2005) e (Vasconcelos 2017) apresentaram um conjunto de três regras heurísticas que garantem o alinhamento do sistema de informação. Baseado nos referidos trabalhos adaptámos as regras ao nosso modelo, que está associado aos já referidos diagramas.

A Figura 3 ilustra um conjunto de regras com vista a alcançar o alinhamento empresarial. As regras heurísticas são as seguintes:

- #0 A arquitetura dos processos deve suportar a narrativa da área de negócio;
- #1 A arquitetura dos dados deve suportar a narrativa da área de negócio;
- #2 Cada atividade do processo é automatizada por uma única aplicação;
- #3 Cada conjunto dados é gerido (CRUD) por uma única aplicação;
- #4 Associar uma infraestrutura a uma ou mais aplicações.

O desenvolvimento do sistema tem início com a Narrativa do Negócio. A aplicação das regras #0 e #1 iniciam as duas visões possíveis do sistema. De seguida é aplicada a regra #2 que é responsável pela transformação digital das atividades em aplicações.

A regra #3 utiliza os dados e as aplicações na criação da matriz CRUD. A matriz CRUD tem analogias com as técnicas 'Design Structure Matrix' (Eppinger, Browning 2012) onde se pretende ter uma visão compactada do sistema com vista a otimizar o projeto.

Finalmente, a regra #4, associa uma infraestrutura tecnológica a uma ou mais aplicações.

Note que as regras #0, #1, #2 e #4 garantem o alinhamento vertical e a regra #3 garante o alinhamento horizontal.

De seguida iremos detalhar o procedimento para modelar a arquitetura empresarial em três camadas.

#### **4 – Procedimento para modelar a Arquitetura Empresarial**

##### **Procedimento:**

- 0) definir a narrativa do negócio
  - i) definir arquitetura processos: listar atividades e atores, diagramas de atividades
  - ii) definir arquitetura do sistema de informação
    - ii.a) definir arquitetura dados: diagrama de classes
    - ii.b) definir matriz CRUD: dados versus aplicações
    - ii.c) detalhar CRUD com diagramas de sequência
  - iii) definir arquitetura física: diagrama de infraestrutura
- iv) verificar as regras de alinhamento e listar os elementos essenciais e rever o procedimento

4.0 – Definir a narrativa do negócio. A narrativa deve referir de uma forma lógica, os atores, as atividades e os dados, bem como a forma como se articulam no negócio.

4.i – Arquitetura dos processos (ou sequência de atividades). Enumerar as atividades, enumerar os atores e preencher a seguinte matriz (atividades versus atores).

| atividades vs atores | $\alpha$ | $\beta$ |
|----------------------|----------|---------|
| A                    |          |         |
| B                    |          |         |
| C                    |          |         |

Elaborar um diagrama de casos de uso UML, ou um diagrama de atividades UML ou ainda um diagrama BPMN onde deve incluir linhas de responsabilidade com os respetivos atores do sistema. Numa abordagem preliminar o número atividades deve ser  $7 \pm 2$ . Aplicar a regra #0, onde a arquitetura dos processos deve suportar a narrativa da área de negócio.

4.ii – Arquitetura do Sistema de Informação

Neste ponto da Arquitetura de Sistema de Informação vamos detalhar 3 subpontos:

- a) definir arquitetura dados: diagrama de classes
- b) definir matriz CRUD: dados versus aplicações
- c) detalhar CRUD com diagramas de sequência

4.ii.a – Arquitetura de dados. Criar para a arquitetura dos dados um diagrama Entidade-Relação ou um diagrama de classes UML que represente os dados necessários que correspondem aos requisitos referidos na narrativa. Devem ser considerados exclusivamente os símbolos da Figura 4 para o diagrama de classes UML. Numa abordagem preliminar o número de classes deve ser  $7 \pm 2$ . Aplicar a regra #1, a arquitetura dos dados deve suportar a narrativa da área de negócio.

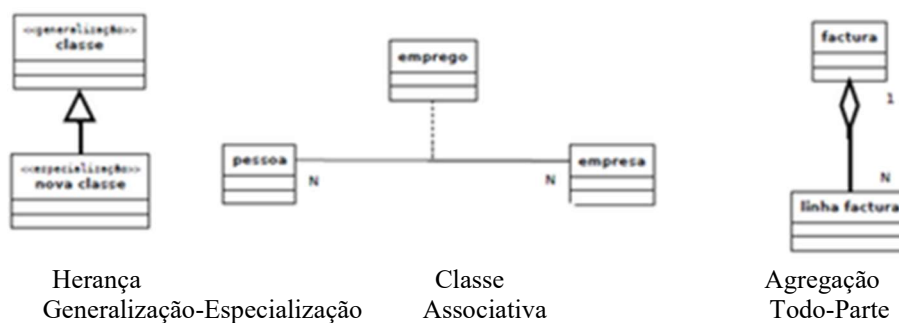


Figura 4 – Símbolos do diagrama de classes

4.ii.b – Aplicar a regra #2 onde cada atividade, que é automatizada, deve corresponder a uma única aplicação. Produzir a matriz CRUD representativa do relacionamento entre aplicações e classes (dados). Aplicar a regra #3, onde cada conjunto dados é gerido (CUD) por uma única aplicação.

| aplicações vs classes | X    | Y    | Z    | W    |
|-----------------------|------|------|------|------|
| A                     |      |      |      |      |
| B                     |      |      |      |      |
| C                     |      |      |      |      |
| outros                |      |      |      |      |
| contadores CRUD       | 1311 | 1111 | 1111 | 1111 |

Na matriz CRUD, devem existir os contadores CRUD, que validam possíveis inconsistências. O número 1311 indica que houve 1 Create, 3 Reads, 1 Update e 1 Delete. Deve existir uma única aplicação que realize o ‘Create’, ‘Update’ e o ‘Delete’, (CUD), podendo existir várias aplicações com ‘Read’. De preferência teremos contadores CRUD com 1N11, i.e. ‘Creates’, ‘Updates’ e ‘Deletes’ únicos e vários ‘Reads’.

4.ii.c – Detalhar a matriz CRUD para cada aplicação, associando um diagrama de sequência (Figura 5).

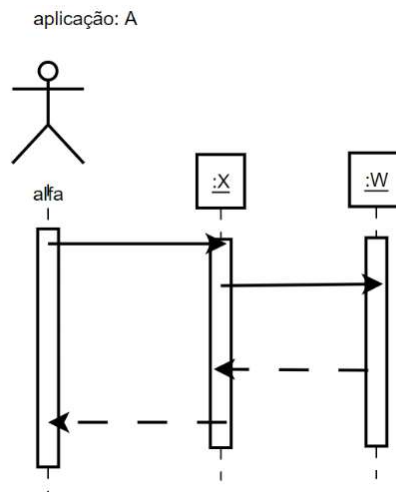


Figura 5 – Diagrama de sequência

4.iii– Arquitetura tecnológica. Modelar em UML a infraestrutura (Figura 6) incluindo plataformas tecnológicas, servidores, computadores clientes, bases de dados, sistemas operativos, etc. Aplicar a regra #4, associando a cada infraestrutura uma ou mais aplicações.

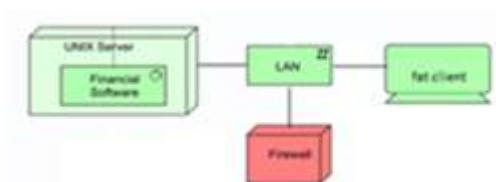


Figura 6 – Diagrama de infraestrutura

4.iv – Verificar as regras heurísticas de alinhamento e criar três listas dos elementos essenciais do seu sistema 'to-be'. Note que existem alguns sinónimos para os elementos: ator, atividade e classe.

- a lista dos atores
- a lista das atividades\aplicações
- a lista das dados\classes

## 5 – Visão integrada das ferramentas

Conforme já foi referido, a análise UML considera essencialmente as seguintes entidades numa organização: os atores, os casos-de-utilização/aplicações e as classes dos dados. Embora o CRUD não seja uma ferramenta UML, pode completar uma visão do sistema como representado na Figura 7.

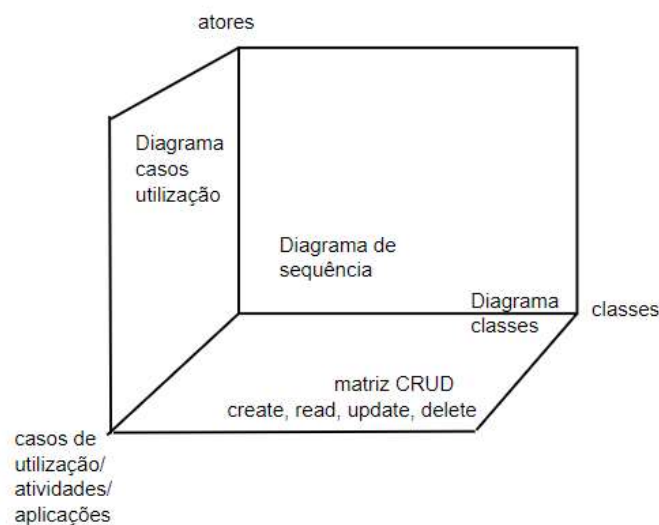


Figura 7 – Diagramas UML e matriz CRUD

Com os três vetores (atores, aplicações, classes) podemos realizar combinações de um elemento de pares e de um trio.

O único diagrama com um único elemento é o diagrama de classes, que é consequência da narrativa do negócio.

O par (atores, aplicações/use-cases) tem o nome de diagrama de casos-de-utilização. Um segundo par (aplicações, classes) é representado pela matriz CRUD, que valida o alinhamento horizontal. Um terceiro par (atores, classes) é representado pelo diagrama de sequência.

Finalmente, podemos dizer que o conjunto de todos os diagramas de sequência representam o trio (aplicação, atores, classes).

## **BIBIOGRAFIA**

Eppinger, S.D., T.R. Browning, (2012), Design Structure Matrix Methods and Applications, MIT Press, Cambridge.

O’Neill H., M. Nunes, P. Ramos, (2010), Exercícios de UML, FCA, ISBN: 978-972-722-616-0.

Pereira, C.M., P. Sousa, (2005), Enterprise architecture: business and it alignment, In H. Haddad, L.M. Liebrock, A. Omicini, R.L. Wainwright, eds., SAC, 1344–1345, ACM.

Silva A., C. Videira. (2008), UML - Metodologias e Ferramentas CASE. Edições Centro Atlântico, volume nº1, 2ª edição, ISBN: 978-989-615-009-9.

Vasconcelos, A., (2017), Slides Aulas Arquiteturas Tecnológicas Empresariais, DEI, Instituto Superior Técnico, Universidade Lisboa, Portugal.

## Estudo de Caso A: Rede Imobiliária

### Procedimento Geral:

- 0) definir a narrativa do negócio
- i) definir arquitetura processos: listar atividades e atores, diagrama atividades
- ii) definir arquitetura dados: diagrama de classes
- iii) definir matriz CRUD: dados versus aplicações
- iv) definir diagramas de sequência e de infraestrutura
- v) verificar as regras de alinhamento e listar elementos essenciais e rever o procedimento

### 0) definir a narrativa do negócio

Pretende-se que faça os passos iniciais do desenvolvimento iterativo de algumas funcionalidades de uma Rede Imobiliária. Após reunião com a comissão da empresa, a sua empresa de *software* recolheu as seguintes informações:

Uma agência tem associado um valor inteiro que a identifica. Algumas agências operam em regime de *franchising* enquanto outras pertencem à empresa mãe. As primeiras têm associado um custo mensal a pagar à empresa mãe.

Os funcionários da agência podem ser comerciais ou angariadores. Um deles é o diretor comercial da agência. Para além da informação típica de funcionário, os comerciais e angariadores podem trabalhar numa mesa de atendimento diferente em cada dia de trabalho. O diretor comercial é responsável por todos os funcionários e é ele que gere a lista de funcionários (inserção, remoção, atualização dos seus dados).

O comercial atende clientes à procura de comprar ou alugar casas. O angariador atende clientes que pretendem vender ou colocar casas a alugar. Assim, os clientes da agência podem ter dois papéis: compradores ou vendedores. De reparar que uma mesma pessoa, ao longo do tempo, pode ser comprador e vendedor. Os clientes são locais às agências pois não existe um registo centralizado.

Designamos por negócio a associação de um comercial, um comprador, um vendedor e um dos seus imóveis. Cada negócio tem associado uma data de início e uma data de conclusão. Também tem um estado: iniciado, em negociação, e concluído (com sucesso ou insucesso). A conclusão com sucesso de um negócio implica atualizar a lista de imóveis disponíveis.

Um imóvel pode ser de venda ou de aluguer. O imóvel possui uma morada, a área total, o preço de mercado, e uma lista de preços (começando pelo preço desejado pelo cliente proprietário). Um imóvel tem uma tipologia (apartamento, vivenda, terreno), e pode ter associado um conjunto de anexos (arrecadações, garagens, anexos). Vamos considerar que um imóvel pode ser composto, i.e., ser constituído por um conjunto de imóveis. Um imóvel pode estar disponível ou não (caso seja parte de um negócio fechado). Um imóvel pode voltar a estar disponível.

O comercial insere clientes compradores, e é capaz de iniciar negócios. Para além disso, o comercial pode incluir propostas da parte do comprador e/ou do vendedor ao longo do respetivo negócio.

O angariador insere clientes vendedores, e possui a capacidade de introduzir novos imóveis no sistema.

Um cliente tem um nome, profissão, contactos e um historial de negócios. Pode possuir também uma lista de imóveis para venda/aluguer.

### **i) definir arquitetura processos: listar atividades e atores, diagrama atividades**

Criámos um glossário com os termos específicos deste sistema:

Angariador – funcionário da agência imobiliária que trabalha com clientes à procura de vender imóveis, clientes vendedores ou senhorios (arrendamento/aluguer)

Cliente comprador – pretende comprar/ter alugado um imóvel

Cliente vendedor – pretende vender/alugar um imóvel

Comercial – funcionário da agência imobiliária que trabalha com clientes à procura de comprar imóveis, clientes compradores ou inquilinos (arrendamento/aluguer)

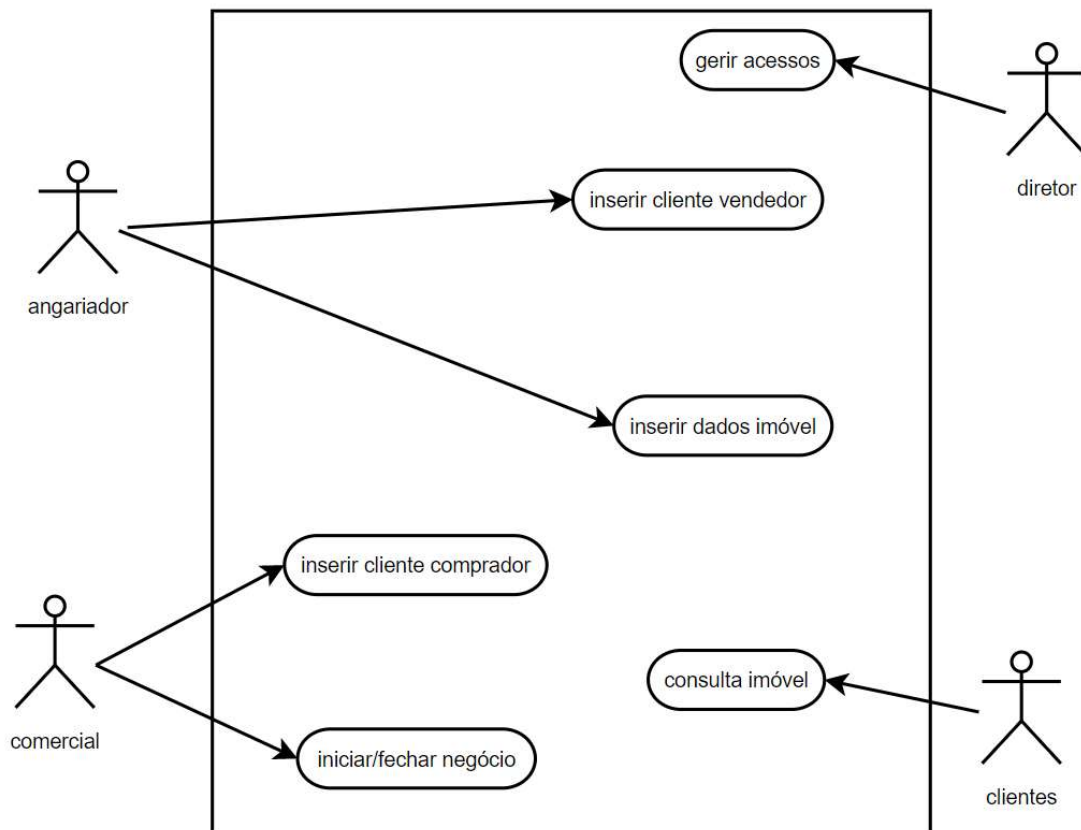
Imóvel – bem imóvel com tipologia (apartamento, vivenda, terreno) podendo ter anexos (arrecadações, garagens, anexos)

Negócio – a associação de um imóvel para venda ou aluguer, a um comercial, um cliente comprador e um cliente vendedor

De seguida construímos a Matriz Atores (substantivos) versus Casos-de-uso (verbos)

| <b>casos-de-uso \ atores</b>    | diretor comercial | angariador | comercial | cliente comprador | cliente vendedor |
|---------------------------------|-------------------|------------|-----------|-------------------|------------------|
| gerir acessos                   | X                 |            |           |                   |                  |
| inserir cliente vendedor        |                   | X          |           |                   |                  |
| inserir dados imóvel            |                   | X          |           |                   |                  |
| inserir cliente comprador       |                   |            | X         |                   |                  |
| iniciar/fechar negócio          |                   |            | X         |                   |                  |
| consultar imóveis (extra texto) |                   |            |           | X                 | X                |

Apresentamos na figura seguinte o diagrama de casos-de-uso em UML:



Podemos ainda incluir as dependências por forma a definir o fluxo do processo:

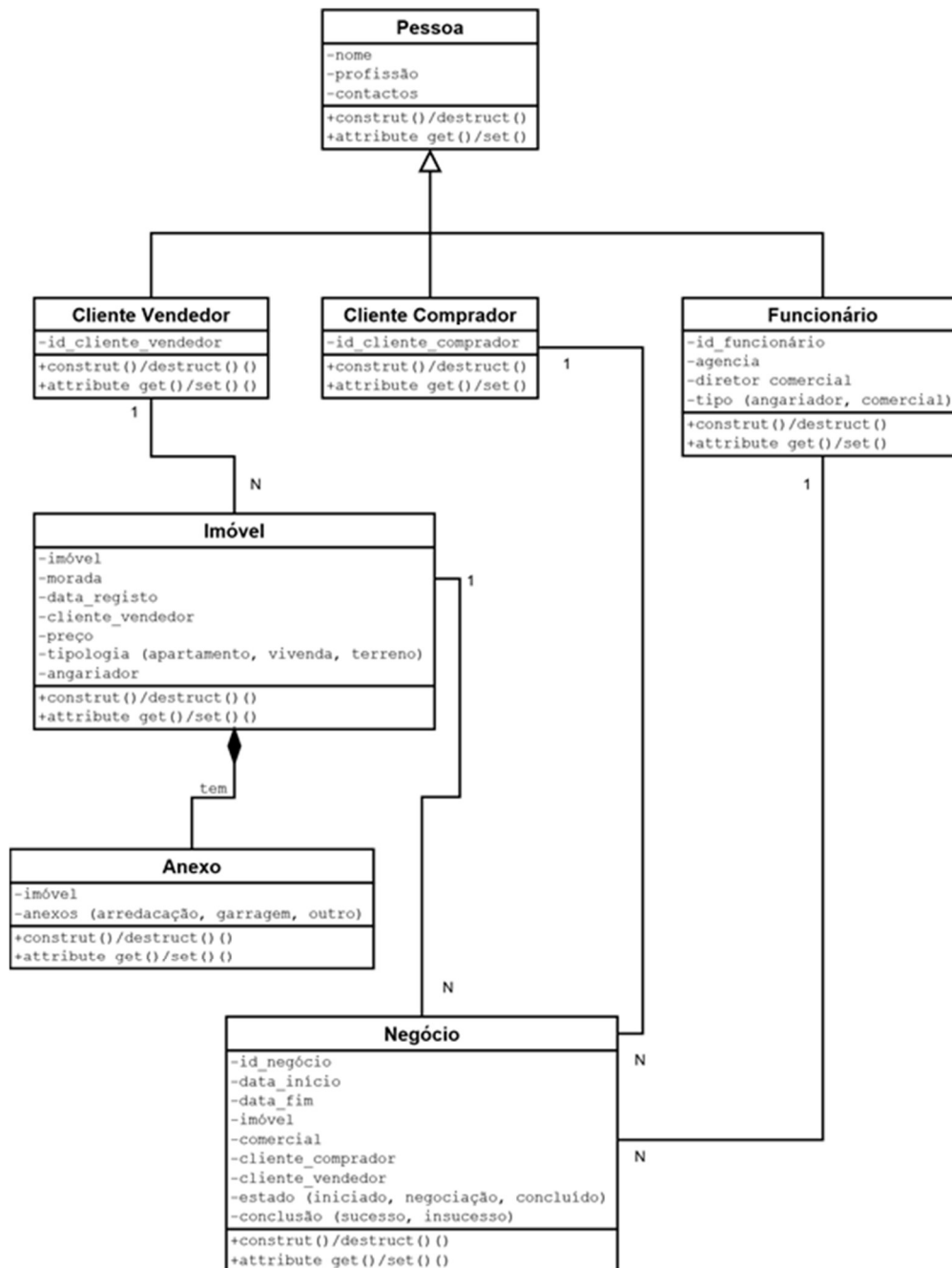
|   | casos-de-uso \ atores           | diretor<br>comercial | angariador | comercial | cliente<br>comprador | cliente<br>vendedor | precedências |
|---|---------------------------------|----------------------|------------|-----------|----------------------|---------------------|--------------|
| 1 | gerir acessos                   | X                    |            |           |                      |                     |              |
| 2 | inserir cliente vendedor        |                      | X          |           |                      |                     | 1            |
| 3 | inserir dados imóvel            |                      | X          |           |                      |                     | 1,2          |
| 4 | inserir cliente comprador       |                      |            | X         |                      |                     | 1            |
| 5 | iniciar/fechar negócio          |                      |            | X         |                      |                     | 1,3,4        |
| 6 | consultar imóveis (extra texto) |                      |            |           | X                    | X                   | 1            |

## ii.a) definir arquitetura dados: diagrama de classes

A agência imobiliária faz negócios ao encontrar para o imóvel do cliente vendedor, um cliente comprador.

Classe relevantes:

- Pessoas (cliente comprador, cliente vendedor e funcionário)
- Imóveis
- Negócio



## ii.b) definir matriz CRUD: dados versus aplicações

A matriz CRUD inicial teremos:

| <b>casos-de-uso \ classes</b> | C. Vendedor | C. Comprador | Funcionário | Imóvel | Negócio |
|-------------------------------|-------------|--------------|-------------|--------|---------|
| gerir acessos                 |             |              | CRUD        |        |         |
| inserir cliente vendedor      | CRUD        |              |             |        |         |
| inserir dados imóvel          | R           |              |             | CRUD   |         |
| inserir cliente comprador     |             | CRUD         |             |        |         |
| iniciar/fechar negócio        | R           | R            | R           | R      | CRUD    |
| <b>contadores CRUD</b>        | 1311        | 1211         | 1211        | 1211   | 1111    |

É de notar que os contadores CRUD respeitam a forma 1N11.

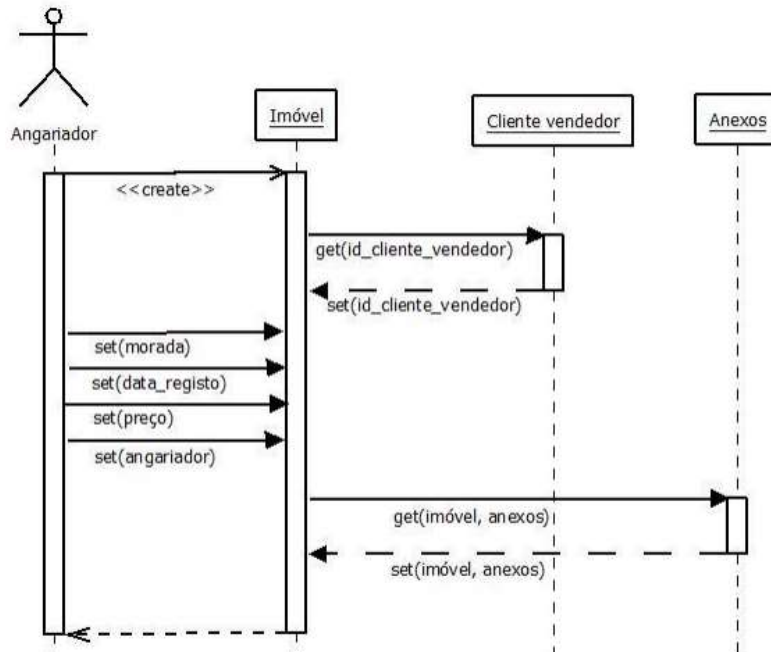
Com vista a reduzir a complexidade da matriz à sua diagonal principal, realizam-se trocas e agregações de linhas e colunas. Assim, depois de algumas iterações encontramos a seguinte matriz CRUD:

| <b>casos-de-uso \ classes</b> | Funcionário | C. Vendedor | Imóvel | C. Comprador | Negócio |
|-------------------------------|-------------|-------------|--------|--------------|---------|
| gerir acessos                 | CRUD        |             |        |              |         |
| inserir cliente vendedor      |             | CRUD        |        |              |         |
| inserir dados imóvel          |             | R           | CRUD   |              |         |
| inserir cliente comprador     |             |             |        | CRUD         |         |
| iniciar/fechar negócio        | R           | R           | R      | R            | CRUD    |
| <b>contadores CRUD</b>        | 1211        | 1311        | 1211   | 1211         | 1111    |

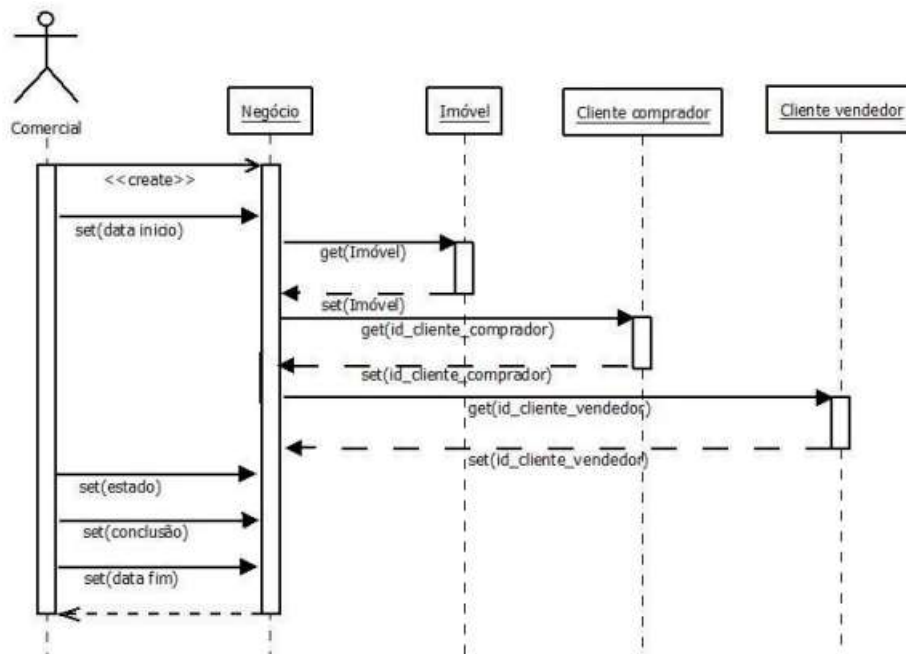
## ii.c) definir diagramas de seqüência

Exemplificamos dois diagramas de seqüência:

Aplicação inserir dados imóvel:



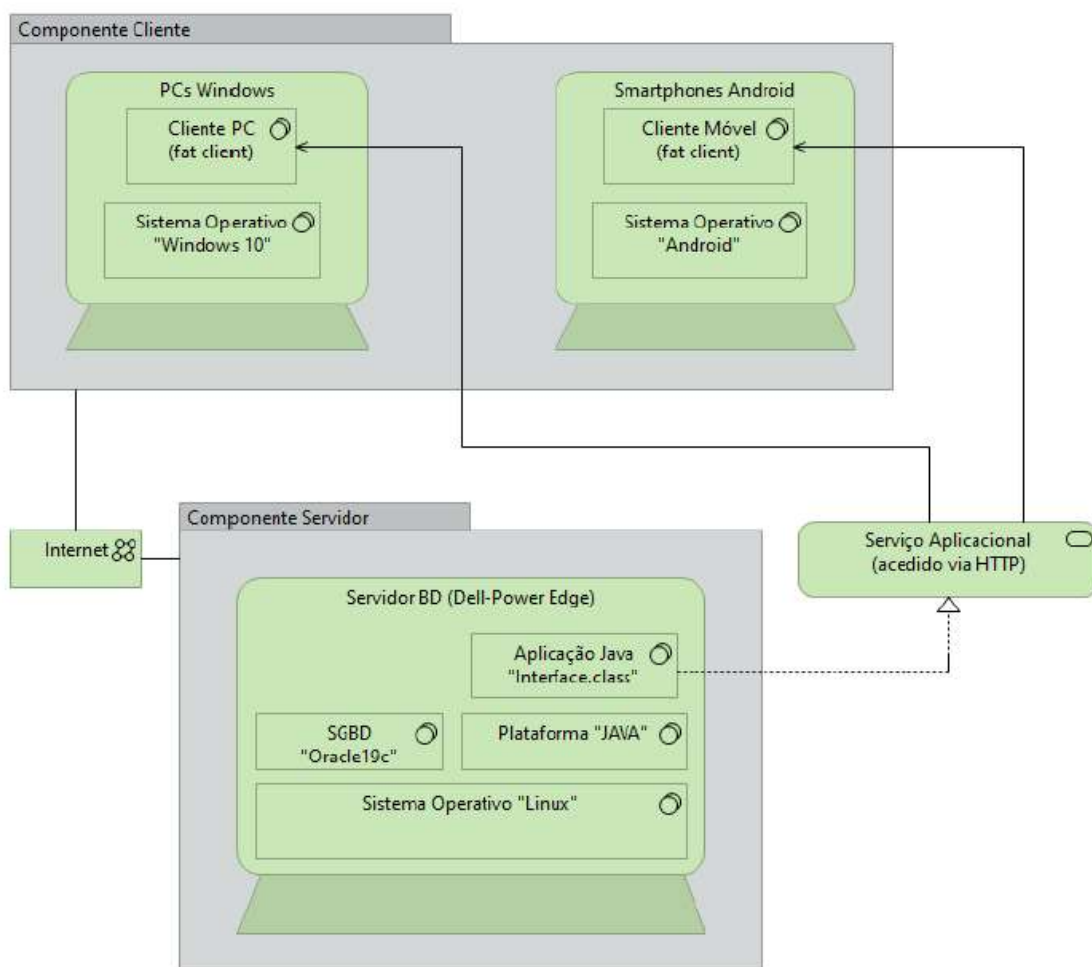
Aplicação iniciar/fechar negócio:



Note que para cada diagrama de seqüência, existe coerência com a matriz CRUD e com o diagrama de 'use-cases', garantido um alinhamento vertical com dois níveis.

### iii) definir diagramas de infraestrutura

Para a infraestrutura tecnológica especifique uma infraestrutura ou utilize a seguinte. O sistema de informação empresarial da organização baseado numa solução Cliente/Servidor. No servidor encontra-se a base de dados que é mantida pelo SGBD “Oracle19c”, o qual por sua vez corre sobre uma máquina “Dell-PowerEdge” com o sistema operativo “Linux”. Por outro lado, o cliente, sendo um “fat-client”, corresponde a uma aplicação Windows com duas variantes: PCs com “Windows 10” e smartphones com “Android”. A interface consiste numa aplicação Java, “Interface.class” que é acedida via HTTP e que se encontra instalada na máquina “Dell-PowerEdge”. Modele no ArchiMate o ponto de vista da infraestrutura (Infrastructure Viewpoint) incluindo plataformas tecnológicas, servidores, computadores clientes, sistemas operativos, etc.



**iv) verificar as regras de alinhamento e listar elementos essenciais e rever o procedimento**

As regras de alinhamento vertical e horizontal foram respeitadas para os seguintes elementos do sistema de informação.

|                         |
|-------------------------|
| <b>atores</b>           |
| diretor comercial       |
| angariador<br>comercial |
| cliente comprador       |
| cliente vendedor        |

|                                 |
|---------------------------------|
| <b>aplicações</b>               |
| gerir acessos                   |
| inserir cliente vendedor        |
| inserir dados imóvel            |
| inserir cliente comprador       |
| iniciar/fechar negócio          |
| consultar imóveis (extra texto) |

|                      |
|----------------------|
| <b>classes dados</b> |
| Funcionário          |
| C. Vendedor          |
| Imóvel               |
| C. Comprador         |
| Negócio              |

## **Estudo de caso B: Surto de Gripe**

### **Procedimento:**

- 0) definir a narrativa do negócio
- i) definir arquitetura processos: listar atividades e atores, diagrama atividades
- ii) definir arquitetura dados: diagrama de classes
- iii) definir matriz CRUD: dados versus aplicações
- iv) definir infraestrutura
- v) verificar as regras de alinhamento e listar elementos essenciais e rever o procedimento

### **0) definir a narrativa do negócio**

Pretende-se que faça a modelação de um sistema de informação das principais funcionalidades de um observatório de saúde pública que permite acompanhar a evolução de um surto de gripe.

Uma importante fonte de informação sobre uma crise de gripe são os ofícios e comunicados oficiais emitidos por organismos responsáveis, por exemplo, um comunicado do Ministério da Saúde. Assim, o sistema deverá manter um registo dos organismos responsáveis, sendo cada organismo caracterizado pela sua designação, URL do *site* e uma descrição. O observatório deverá disponibilizar uma lista de contactos para o público em geral. Sobre cada contacto pessoal deverá ser disponibilizado o nome, o telefone e o *e-mail*. Note que um organismo pode ter mais do que uma pessoa para contacto. Deverá ser mantido um registo de todos os ofícios emitidos pelos organismos oficiais. Os ofícios referem casos de gripe detetados.

Será ainda de manter no sistema a informação que diz respeito às diferentes estirpes de vírus dos casos detetados. Sobre cada estirpe é necessário saber a designação, os sintomas, a data em que foi identificada a estirpe. Além disso, é necessário saber quais os medicamentos disponíveis para combater as diferentes estirpes do vírus da gripe. Sobre o medicamento é necessário saber qual o laboratório que o disponibiliza.

No observatório existe dois conjuntos de funcionários. Uns digitalizam os ofícios e atualizam os dados dos casos detetados da gripe e outros atualizam a informação ligada às estirpes de vírus. Quando existe um surto de gripe, o diretor do observatório, pretende ter um gráfico dos casos de gripe por semana e por estirpe.

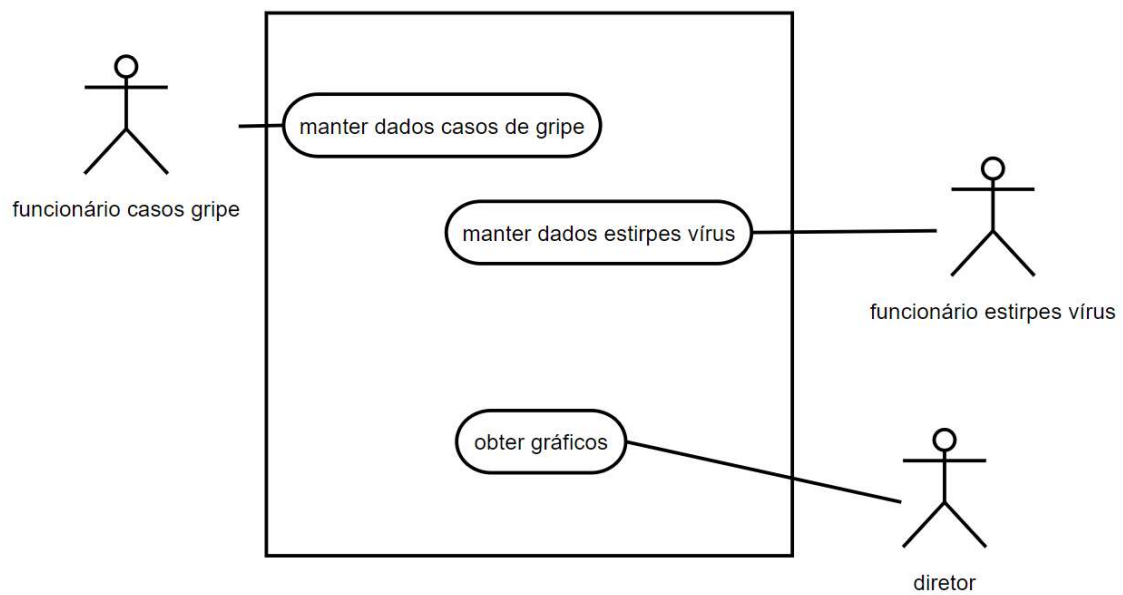
Na sua análise não exceda 7 atores, 7 casos-de-utilização (funcionalidades) e 7 classes.

**i) definir arquitetura processos: listar atividades e atores, diagrama atividades**

De seguida construímos a Matriz Atores (substantivos) versus Casos-de-uso (verbos)

| <b>casos-de-uso \ atores</b> | funcionário casos gripe | funcionário estirpes vírus | diretor |
|------------------------------|-------------------------|----------------------------|---------|
| manter dados casos gripe     | X                       |                            |         |
| manter dados estirpes vírus  |                         | X                          |         |
| obter gráficos               |                         |                            | X       |

Apresentamos na figura seguinte o diagrama de casos-de-uso em UML:



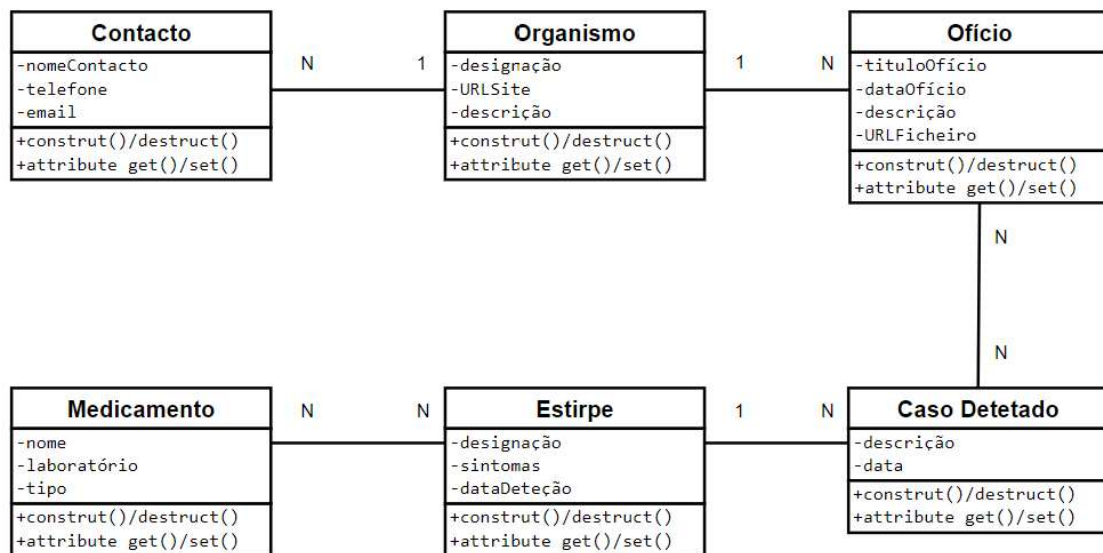
Podemos ainda incluir as dependências por forma a definir o fluxo do processo:

|   | <b>casos-de-uso \ atores</b> | funcionário casos gripe | funcionário estirpes vírus | diretor | <i>precedência</i> |
|---|------------------------------|-------------------------|----------------------------|---------|--------------------|
| 1 | manter dados casos gripe     | X                       |                            |         |                    |
| 2 | manter dados estirpes vírus  |                         | X                          |         |                    |
| 3 | obter gráficos               |                         |                            | X       | 1,2                |

## ii.a) definir arquitetura dados: diagrama de classes

Resumo das classes dos dados:

- os casos de surtos de gripe são informados por ofícios (ou cartas do poder local ou central)
- os ofícios pertencem a organismos do poder local ou central e têm diversos contactos
- cada caso de gripe corresponde a uma ou mais estirpes de vírus para os quais existem vários medicamentos



## ii.b) definir matriz CRUD: dados versus aplicações

A matriz CRUD inicial teremos:

| <b>casos-de-uso \ classes</b> | contacto | organismo | oficio | caso detetado | estirpe | medicamento |
|-------------------------------|----------|-----------|--------|---------------|---------|-------------|
| manter dados casos gripe      | CRUD     | CRUD      | CRUD   | CRUD          |         |             |
| manter dados estirpes vírus   |          |           |        |               | CRUD    | CRUD        |
| obter gráficos                |          |           |        | R             | R       |             |
| <b>contadores CRUD</b>        | 1111     | 1111      | 1111   | 1211          | 1211    | 1111        |

É de notar que os contadores CRUD respeitam a forma 1N11.

Com vista a reduzir a complexidade da matriz à sua diagonal principal, realizam-se trocas e agregações de linhas e colunas. Assim, depois de algumas iterações encontramos a seguinte matriz CRUD:

| <b>casos-de-uso \ classes</b> | casos | estirpes |
|-------------------------------|-------|----------|
| manter dados casos gripe      | CRUD  |          |
| manter dados estirpes vírus   |       | CRUD     |
| obter gráficos                | R     | R        |
| <b>contadores CRUD</b>        | 1211  | 1211     |

## iii) definir diagramas de sequência e de infraestrutura

NA- não aplicável, idêntico ao caso A

**iv) verificar as regras de alinhamento e listar elementos essenciais e rever o procedimento**

As regras de alinhamento vertical e horizontal foram respeitadas para os seguintes elementos do sistema de informação.

|                            |
|----------------------------|
| <b>atores</b>              |
| funcionário casos gripe    |
| funcionário estirpes vírus |
| diretor                    |

|                                 |
|---------------------------------|
| <b>casos-de-uso / aplicação</b> |
| manter dados casos gripe        |
| manter dados estirpes vírus     |
| obter gráficos                  |

|                |
|----------------|
| <b>classes</b> |
| contacto       |
| organismo      |
| ofício         |
| caso detetado  |
| estirpe        |
| medicamento    |

## **Estudo de caso C: Gamificação**

### **Procedimento Geral:**

- 0) definir a narrativa do negócio
- i) definir arquitetura processos: listar atividades e atores, diagrama atividades
- ii) definir arquitetura dados: diagrama de classes
- iii) definir matriz CRUD: dados versus aplicações
- iv) definir diagramas sequência e infraestrutura
- v) verificar as regras de alinhamento e listar elementos essenciais e rever o procedimento

### **0) definir a narrativa do negócio**

Considere um subsistema de gamificação de um 'contact-center'. No 'contact-center' os funcionários tratam as solicitações dos clientes através de mensagens de voz e de texto. O subsistema de gamificação recebe informação do subsistema de mensagens. O subsistema de gamificação tem os seguintes requisitos:

- Os utilizadores têm um identificador, nome e contactos;
- Os utilizadores agrupam-se em equipas; cada equipa tem um identificador, um nome e um chefe de equipa;
- Existem ainda atividades semanais de gamificação como os desafios ('challenges') e as missões ('quests');
- Os desafios têm um nome, número da semana (1 a 52), ano e um KPI ('key performance indicator'); um exemplo de desafio será responder a uma determinada quantidade de solicitações de clientes numa determinada semana; os desafios estão associados aos utilizadores;
- Por outro lado, as missões estão associadas às equipas e devem ser registados os atributos: nome, semana, ano e KPI da equipa;
- Os utilizadores que atingem os KPI são distinguidos com distintivos ('badges') e as equipas são recompensadas com taças ('cup');

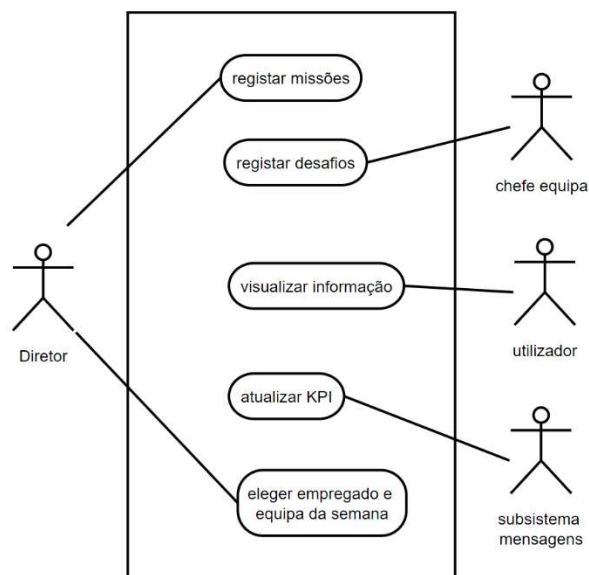
Quem regista as missões das equipas é o diretor. Quem regista os desafios individuais é o chefe da equipa. Todos os utilizadores podem visualizar os seus resultados-semanais individuais e da sua equipa. Os KPI são atualizados pelo subsistema de mensagens. Todas as semanas o diretor elege o empregado/utilizador da semana e a equipa da semana com base nos distintivos e nas taças. Pretende-se ainda que o subsistema de gamificação tenha um 'leaderboard' para os utilizadores que acumula a informação semanal.

**i) definir arquitetura processos: listar atividades e atores, diagrama atividades**

De seguida construímos a Matriz Atores (substantivos) versus Casos-de-uso (verbos)

| <b>casos-de-uso \ atores</b> | diretor | chefe equipa | utilizador | subsistema mensagens |
|------------------------------|---------|--------------|------------|----------------------|
| registar missões             | X       |              |            |                      |
| registar desafios            |         | X            |            |                      |
| visualizar informação        |         |              | X          |                      |
| atualizar KPI                |         |              |            | X                    |
| eleger empregado/equipa      | X       |              |            |                      |

Apresentamos na figura seguinte o diagrama de casos-de-uso em UML:

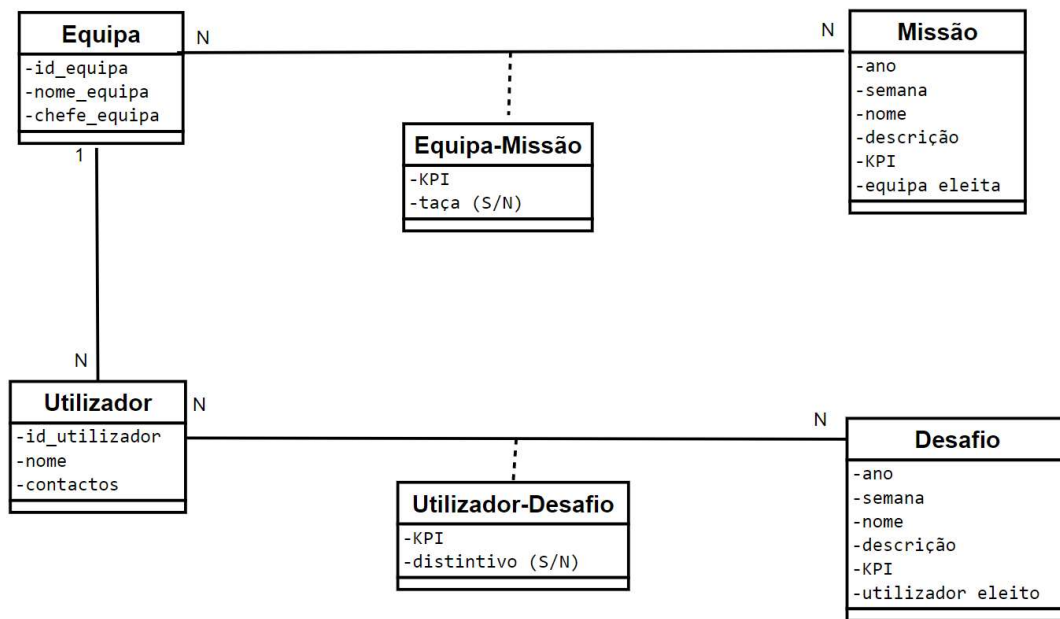


Podemos ainda incluir as dependências por forma a definir o fluxo do processo:

|   | <b>casos-de-uso \ atores</b>   | diretor | chefe equipa | utilizadores | subsistema mensagens | <i>precedências</i> |
|---|--------------------------------|---------|--------------|--------------|----------------------|---------------------|
| 1 | registar missões               | X       |              |              |                      |                     |
| 2 | registar desafios              |         | X            |              |                      |                     |
| 3 | visualizar informação          |         |              | X            |                      | 1,2,4               |
| 4 | atualizar KPI                  |         |              |              | X                    | 1,2                 |
| 5 | eleger empregado/equipa semana | X       |              |              |                      | 1,2,4               |

## ii.a) definir arquitetura dados: diagrama de classes

- cada equipa tem vários utilizadores
- as equipas ganham taças nas missões semanais
- os utilizadores ganham distintivos nos desafios semanais



## ii.b) definir matriz CRUD: dados versus aplicações

A matriz CRUD inicial teremos:

| <b>casos-de-uso \ classes</b>                   | equipa | utilizador | missão  | desafio | missão semanal | desafio semanal |
|---|--------|------------|---------|---------|----------------|-----------------|
| registar missões                                |        |            | CRUD    |         |                |                 |
| registar desafios                               |        |            |         | CRUD    |                |                 |
| visualizar informação                           | R      | R          | R       | R       | R              | R               |
| atualizar KPI                                   |        |            |         |         | CRUD           | CRUD            |
| eleger empregado/equipa semana                  |        |            | RU      | RU      |                |                 |
| outros  | CRUD   | CRUD       |         |         |                |                 |
| <b>contadores CRUD</b>                          | 1211   | 1211       | 1321(a) | 1321(a) | 1211           | 1211            |
| (a) update realizado sobre atributos diferentes |        |            |         |         |                |                 |

É de notar que os contadores CRUD não respeitam a forma 1N11.

Com vista a reduzir a complexidade da matriz à sua diagonal principal, realizam-se trocas e agregações de linhas e colunas. Assim, depois de algumas iterações encontramos a seguinte matriz CRUD:

| <b>casos-de-uso \ classes</b>  | missão  | desafio | missão semanal | desafio semanal | equipa | utilizador |
|--------------------------------|---------|---------|----------------|-----------------|--------|------------|
| registar missões               | CRUD    |         |                |                 |        |            |
| registar desafios              |         | CRUD    |                |                 |        |            |
| eleger empregado/equipa semana | RU      | RU      |                |                 |        |            |
| atualizar KPI                  |         |         | CRUD           | CRUD            |        |            |
| outros                         |         |         |                |                 | CRUD   | CRUD       |
| visualizar informação          | R       | R       | R              | R               | R      | R          |
| <b>contadores CRUD</b>         | 1321(a) | 1321(a) | 1211           | 1211            | 1211   | 1211       |

## iii) definir diagramas de sequência e de infraestrutura

NA- não aplicável, idêntico ao caso A

**iv) verificar as regras de alinhamento e listar elementos essenciais e rever o procedimento**

As regras de alinhamento vertical e horizontal foram respeitadas para os seguintes elementos do sistema de informação.

|                      |
|----------------------|
| <b>atores</b>        |
| diretor              |
| chefe equipa         |
| utilizadores         |
| subsistema mensagens |

|                                  |
|----------------------------------|
| <b>casos-de-uso / aplicações</b> |
| registar missões                 |
| registar desafios                |
| visualizar informação            |
| atualizar KPI                    |
| eleger empregado/equipa semana   |

|                         |
|-------------------------|
| <b>classes de dados</b> |
| equipa                  |
| utilizador              |
| missão                  |
| desafio                 |
| missão semanal          |
| desafio semanal         |