Modelação Estrutural + Notas Finais

(Diagrama de Instalação)

Desenvolvimento de Sistemas Software
José Creissac Campos / António Nestor Ribeiro

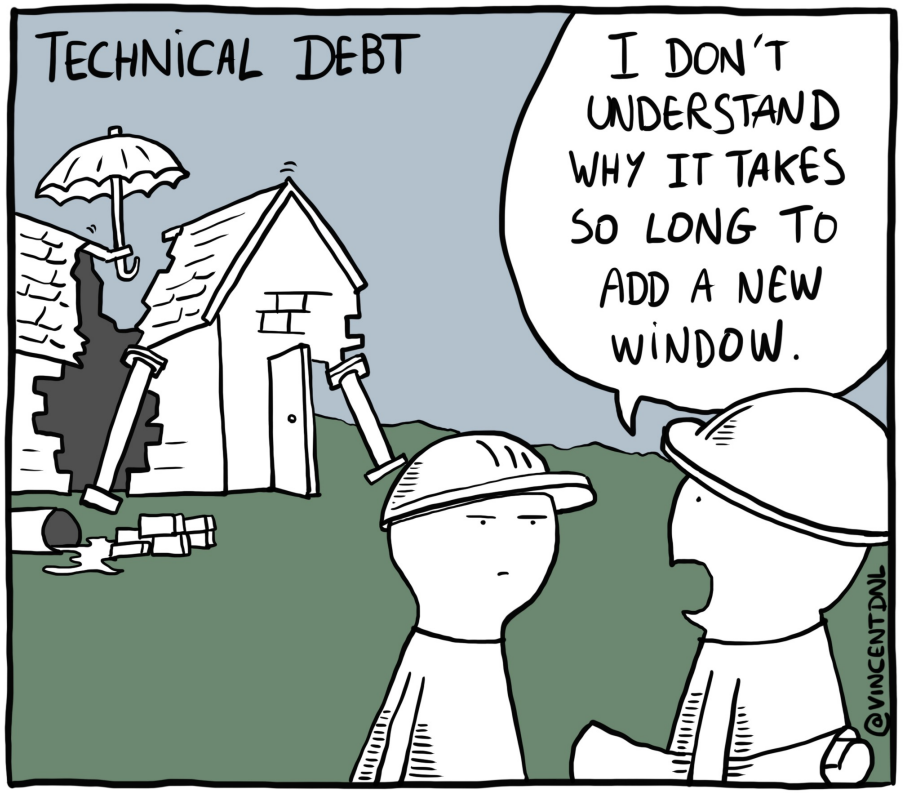
Da primeira aula...

Version 2 announcement:

Among Us 1 (…) started as a tiny local-multiplayer-only game and **has grown and grown and grown**. (…) Because of this, it's **extremely hard to add more things** (…) **because the game is so fragile**. Fixing [it would be] (…) **harder than just making a new game**.

So the first goal of **Among Us 2 is to be made to withstand growth**. We want to add to it at least as long as Among Us, but **with fewer bugs along the way**.

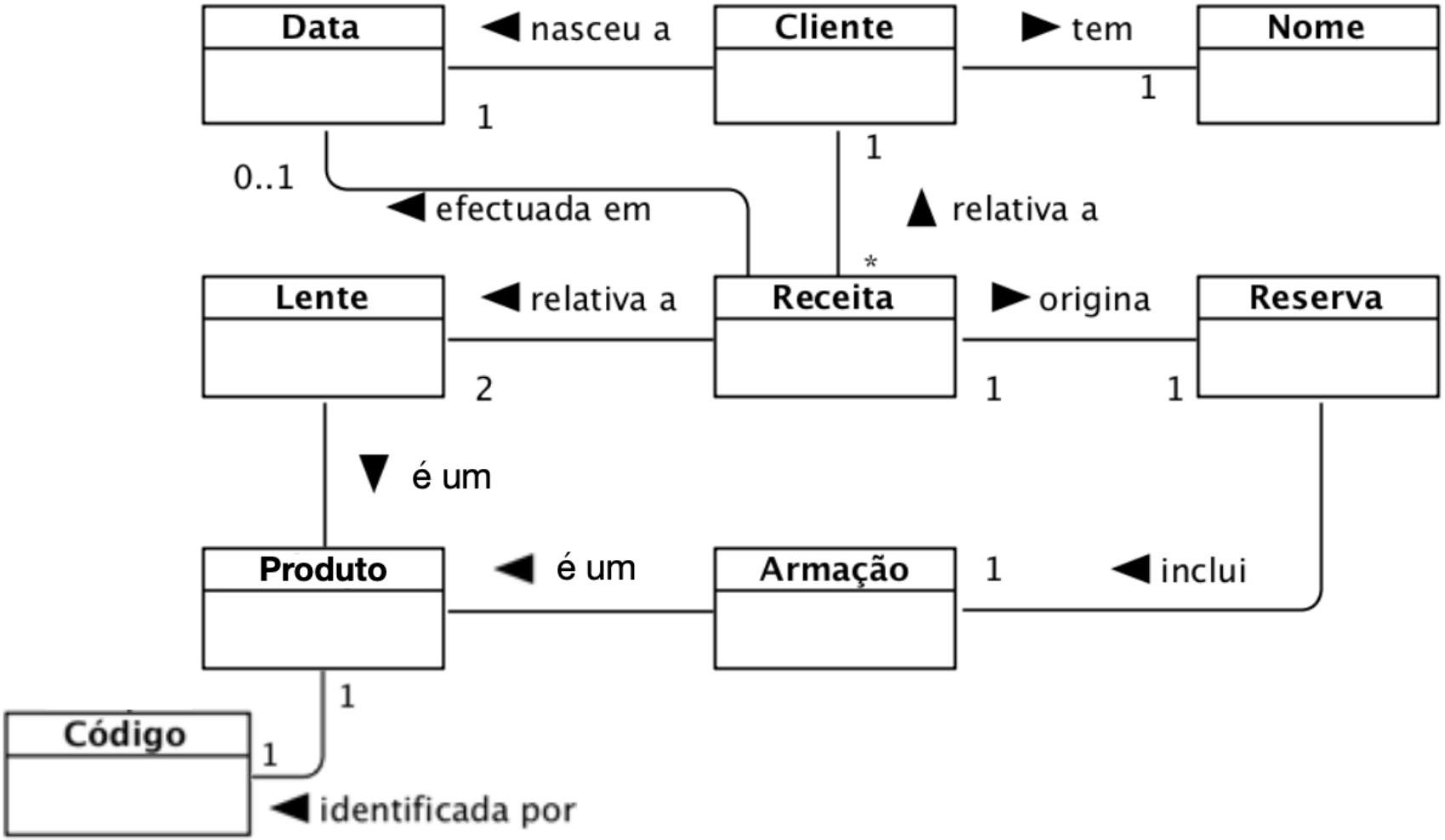
Dívida técnica *(technical debt)*



- Resulta de priorizar entregas rápidas em vez de qualidade da solução
- Software desenvolvido de forma parcial (com problemas por resolver)
- Erros e limitações afectam evolução do produto

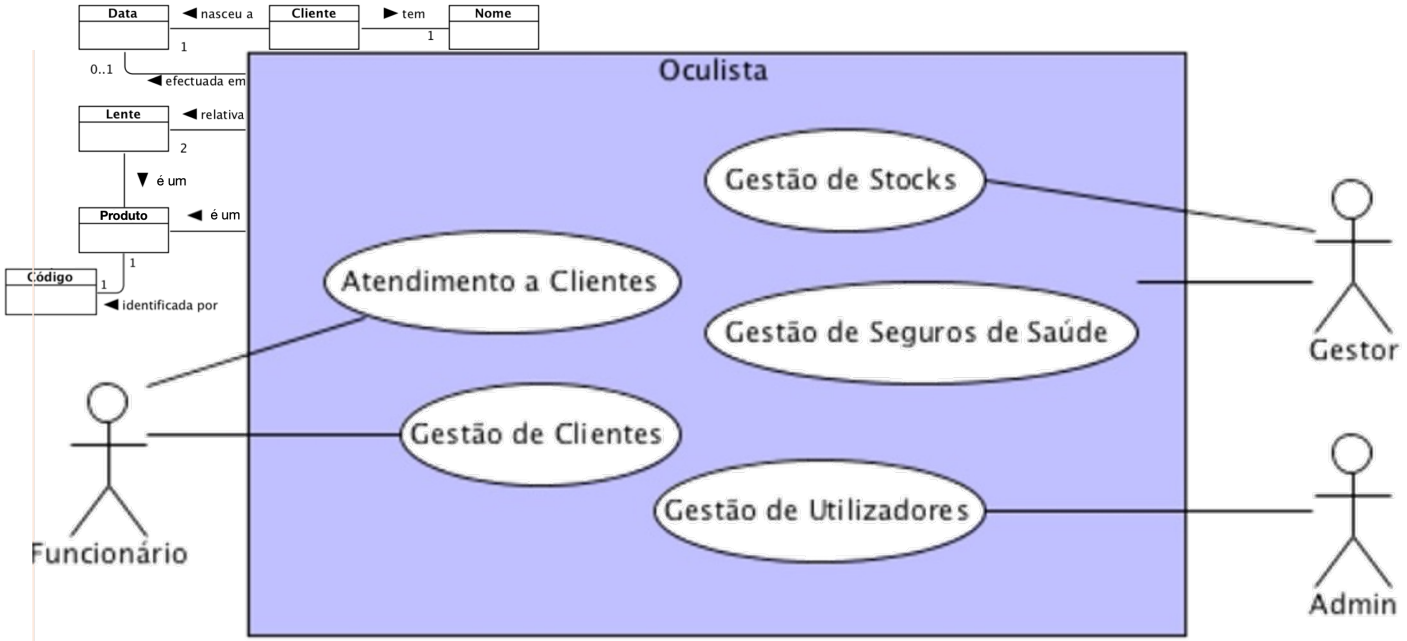
Análise - Perceber o problema

- Modelar o Domínio



Análise - Perceber os requisitos

- Modelo de Use Case



Use Case: Reservar armação e lentes

Descrição: Funcionário regista uma reserva de armação e lentes

Pós-condição: Reserva fica registada

Fluxo normal:

1. Funcionário indica nome e/ou data de nascimento do cliente
2. Sistema procura clientes
3. Sistema apresenta lista de clientes
4. Funcionário selecciona cliente
5. Sistema procura cliente
6. Sistema apresenta detalhes do cliente
7. Funcionário confirma cliente
8. Sistema procura produtos e apresenta lista
9. Funcionário indica Código de armação e lentes
10. Sistema procura detalhes dos produtos
11. Sistema apresenta detalhes dos produtos
12. Funcionário confirma produtos
13. Sistema regista reserva dos produtos
14. <<include>> imprimir talão

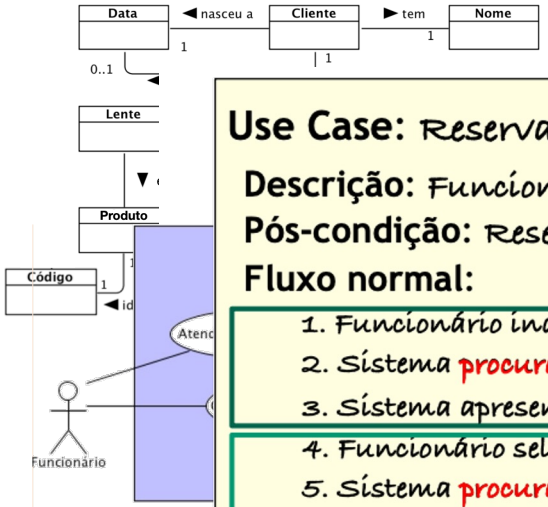
Fluxo de excepção: [cliente não quer produto] (passo 12)

11.1. Funcionário rejeita produtos

11.2. Sistema termina processo

Concepção - Conceber a solução

- Identificar a API da LN



Use Case: Reservar armação e lentes

Descrição: Funcionário regista uma reserva de armação e lentes

Pós-condição: Reserva fica registada

Fluxo normal:

1. Funcionário indica nome e/ou data de nascimento do cliente
2. Sistema **procura clientes**
3. Sistema apresenta lista de clientes
4. Funcionário selecciona cliente
5. Sistema **procura cliente**
6. Sistema apresenta detalhes do cliente
7. Funcionário confirma cliente
8. Sistema **procura produtos** e apresenta lista
9. Funcionário indica Código de armação e lentes
10. Sistema **procura detalhes dos produtos**
11. Sistema apresenta detalhes dos produtos
12. Funcionário confirma produtos
13. Sistema **regista reserva dos produtos**
14. <<include>> imprimir talão

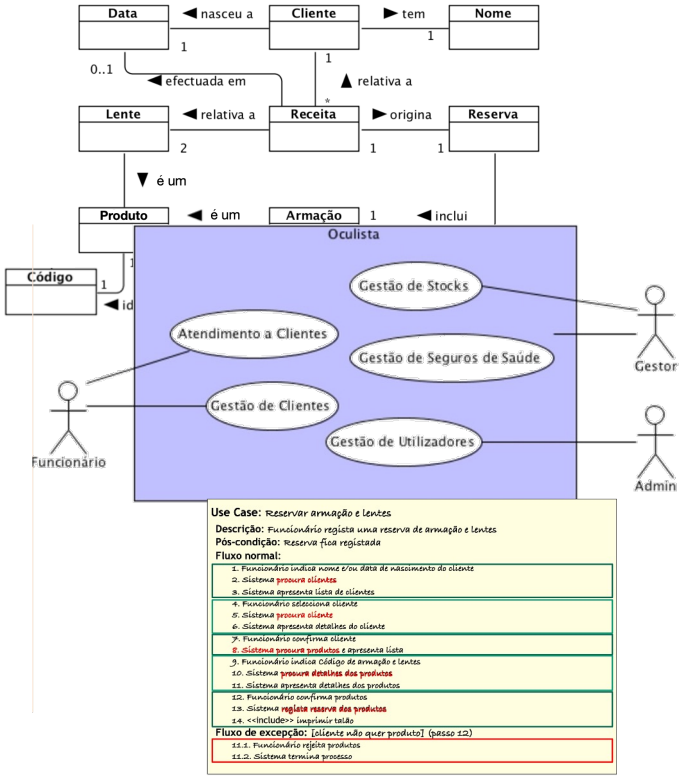
Fluxo de excepção: [cliente não quer produto] (passo 12)

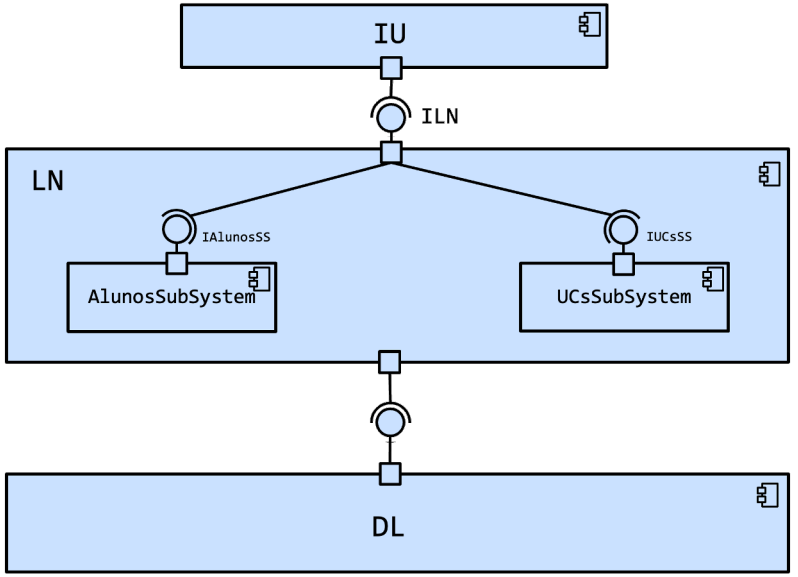
11.1. Funcionário rejeita produtos

11.2. Sistema termina processo

Concepção - Conceber a solução

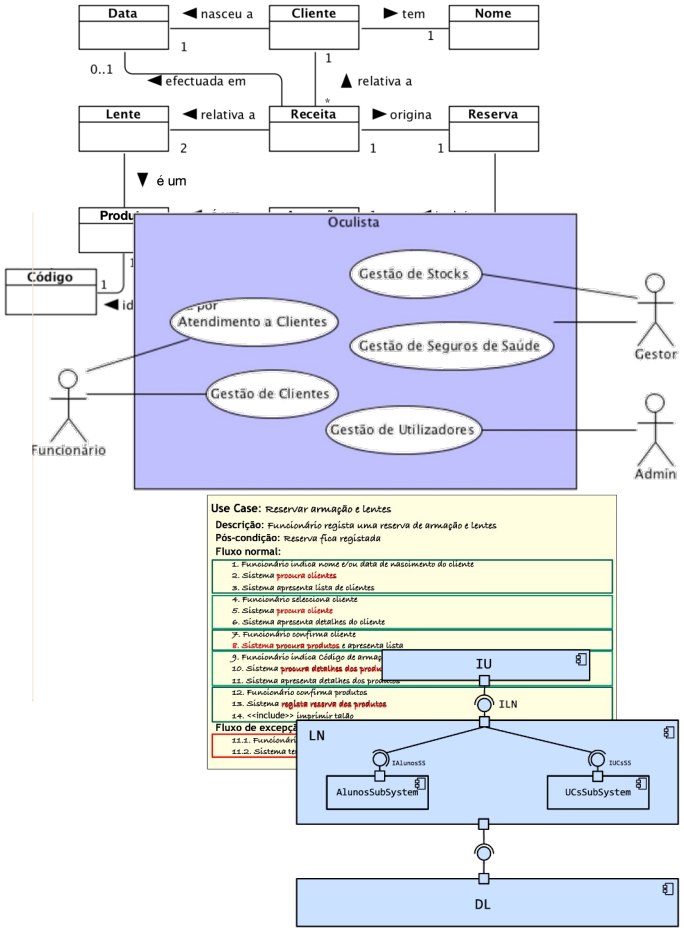
- Identificar subsistemas e suas APIs

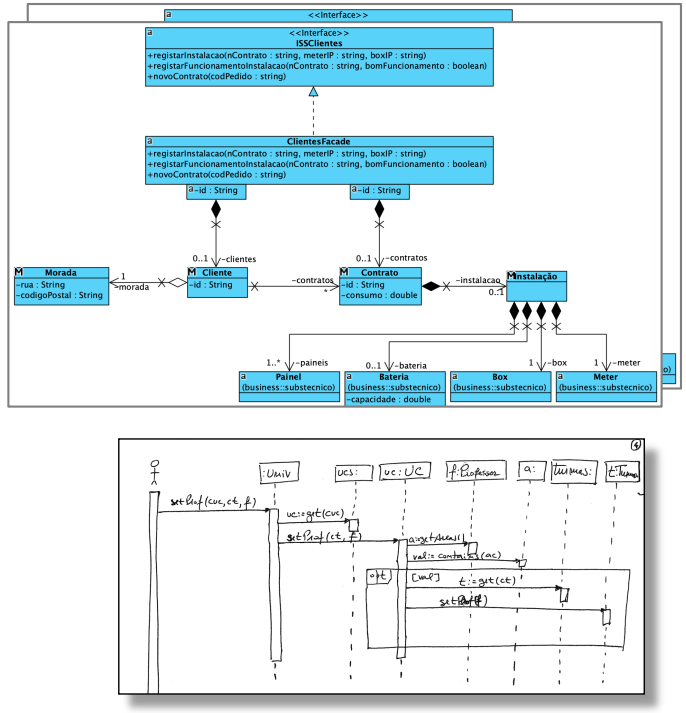




Concepção - Conceber a solução

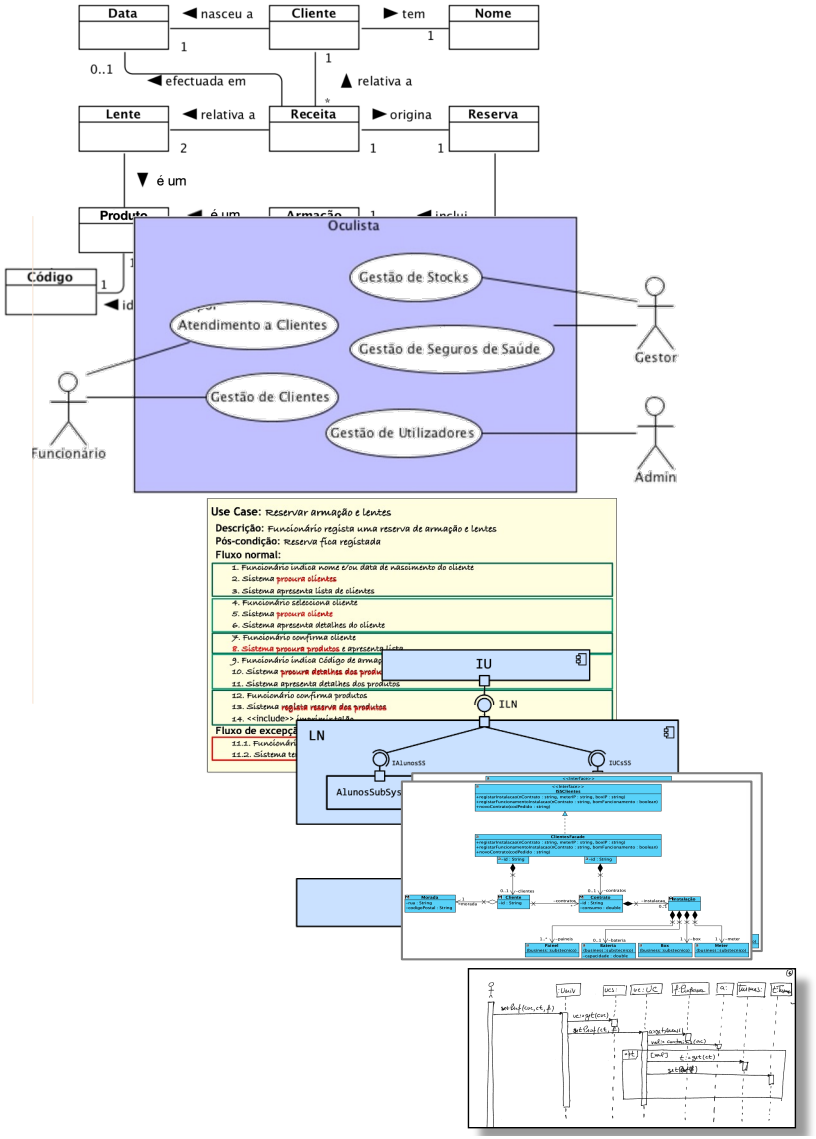
- Conceber os subsistemas

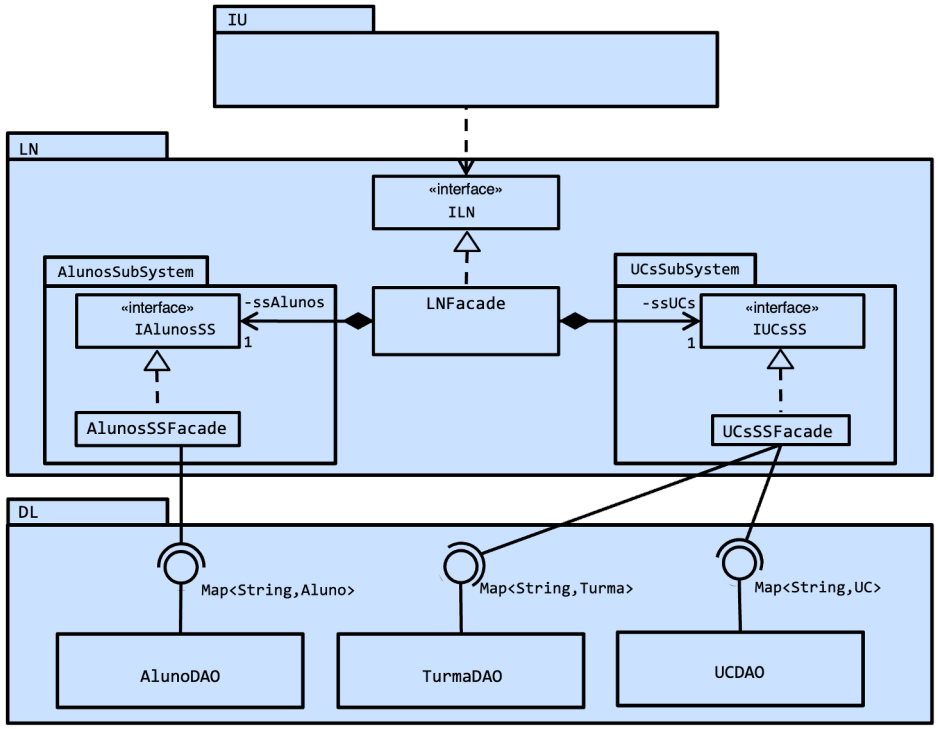


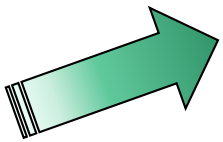


Concepção - Conceber a solução

- Ajustar à solução tecnológica concreta



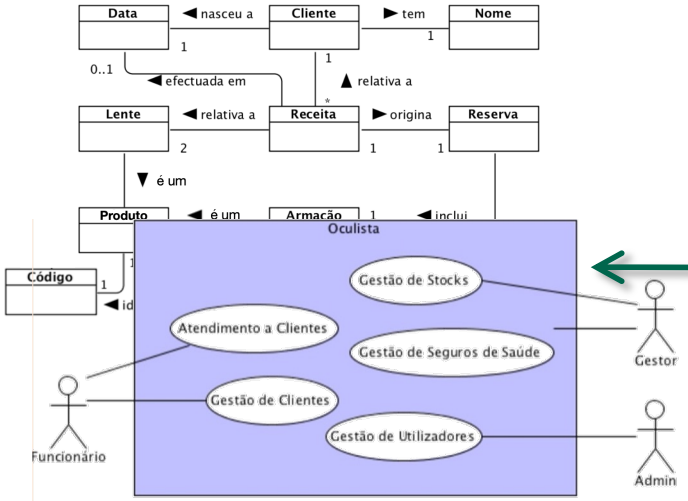


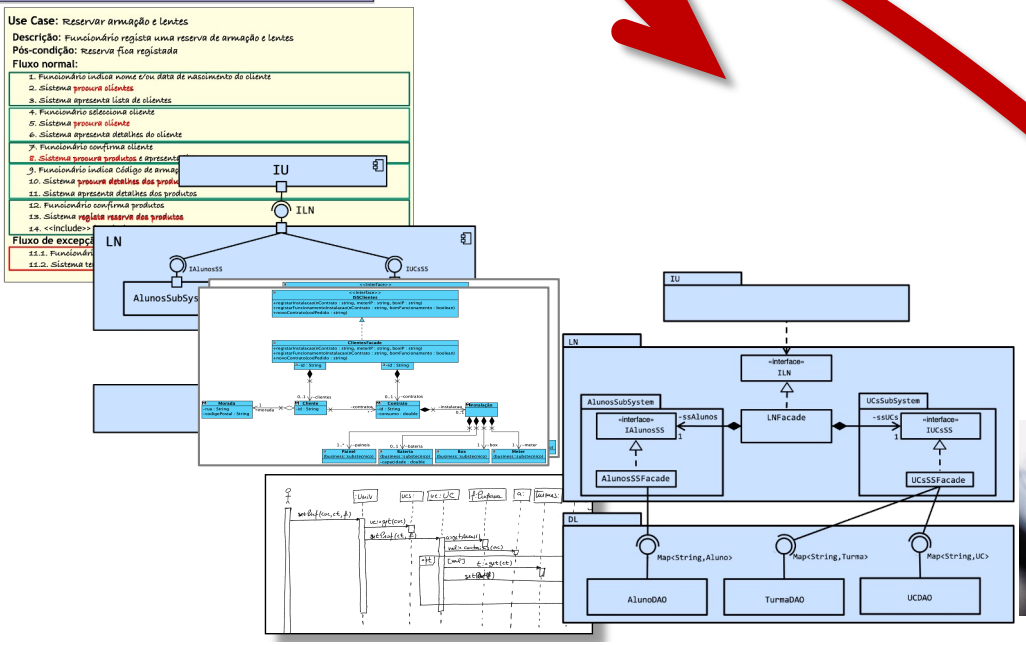


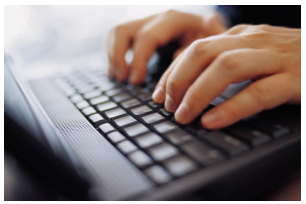
Análise → Implementação

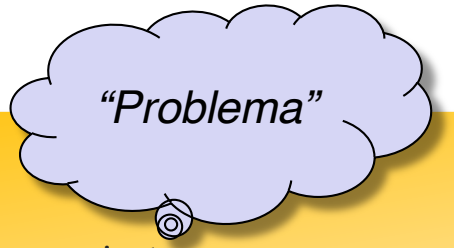
- Implementar, testar e instalar

Atenção!
Não é um processo em cascata!







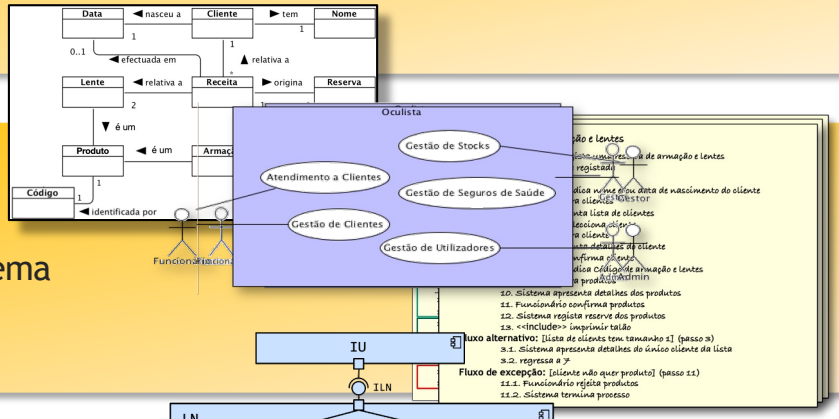


Planeamento

- Decisão de avançar com o projecto
- Gestão do projecto

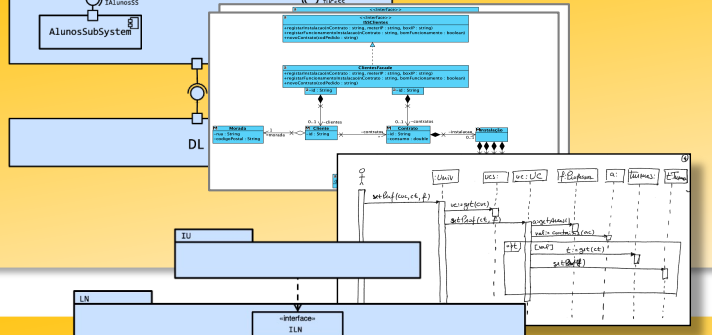
Análise

- Análise do domínio do problema
- Análise de requisitos



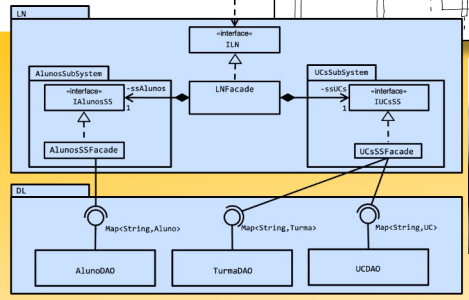
Concepção

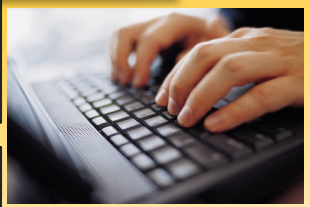
- Concepção da Arquitectura
- Concepção do Comportamento



Implementação

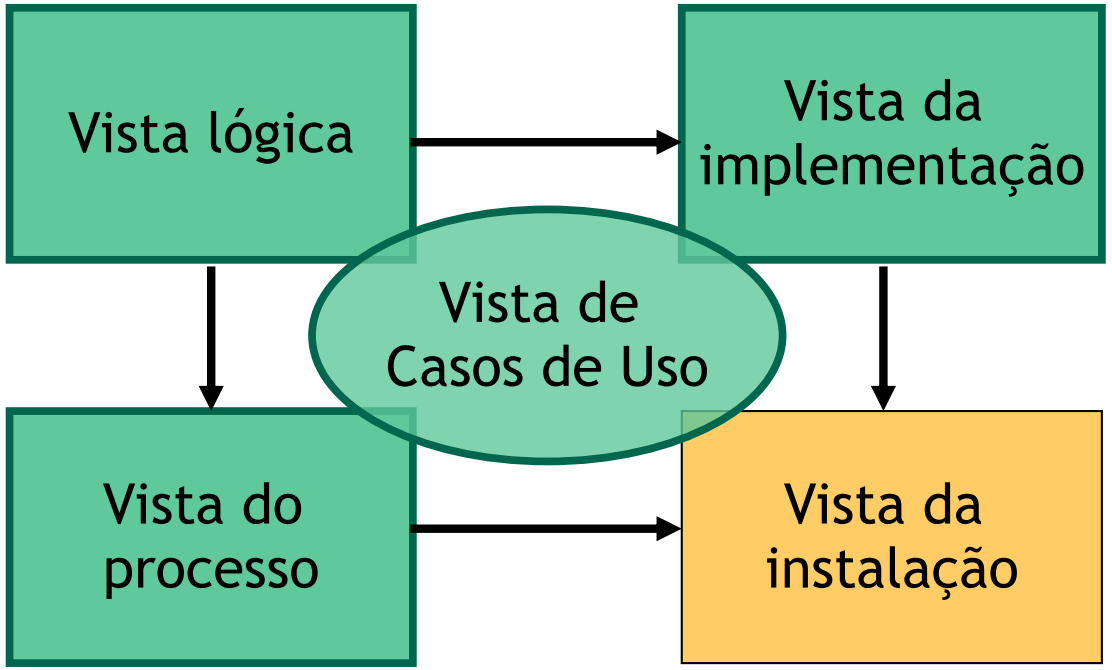
- Construção
- Teste
- Instalação
- Manutenção





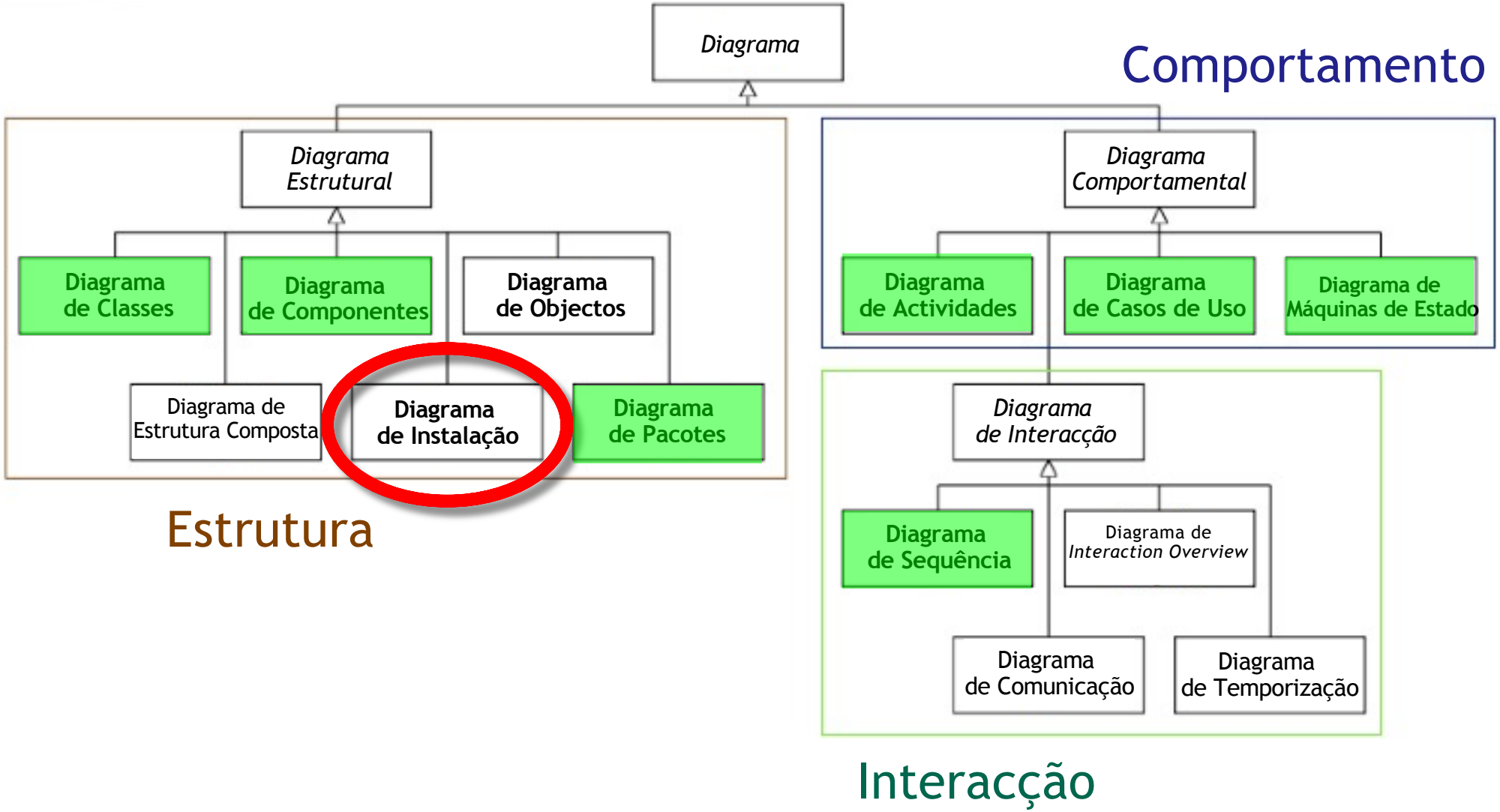
Uma ferramenta

Sobre a instalação do sistema...

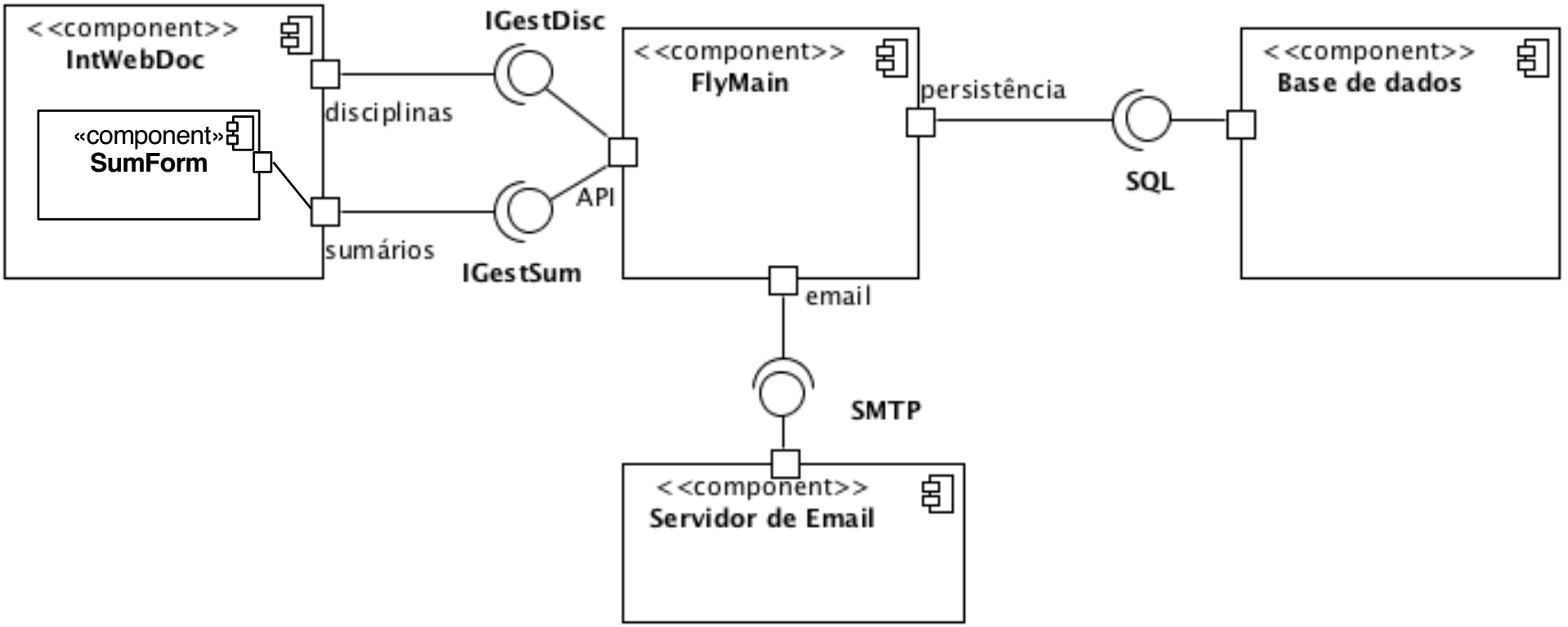


(Kruchten, 1995)

Diagramas da UML 2.x



Um sistema divide-se em Componentes



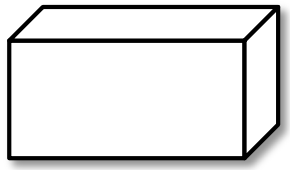
Diagramas de Instalação *(Deployment)*

- Qual a disposição física dos componentes que constituem o sistema?
  - Qual é a configuração do sistema em tempo de execução?
- Diagramas de Instalação especificam a arquitectura física do sistema
  - Topologia (ambiente) de hardware sobre a qual são executados os componentes de software
- Permitem:
  - Especificar a distribuição de componentes
  - Identificar estrangulamentos de desempenho

Diagramas de Instalação

- Elementos de um diagrama de deployment
  - Nós
  - Ligações

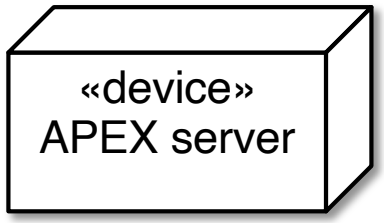
- **Nós (*nodes*):**
  - Computadores ou outros dispositivos (**hardware**)
  - Sistema operativo, *web servers*, *application servers*, etc. (**ambientes de execução**)
  - Os componentes localizados (*deployed*) em cada nó são representados explicitamente



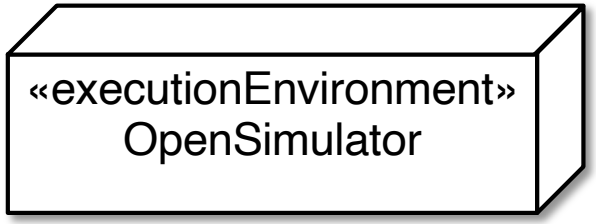
- **Ligações (*connections*):**
  - Representam **comunicação** entre os nós.
  - Podem ser decoradas com multiplicidades.
  - Podem ter estereótipos que indicam o tipo de ligação.
    - Exemplo: «TCP/IP» ou «RMI»

Diagramas de Instalação

- Por vezes utiliza-se o estereótipo **«device»** para identificar os nós de **hardware**



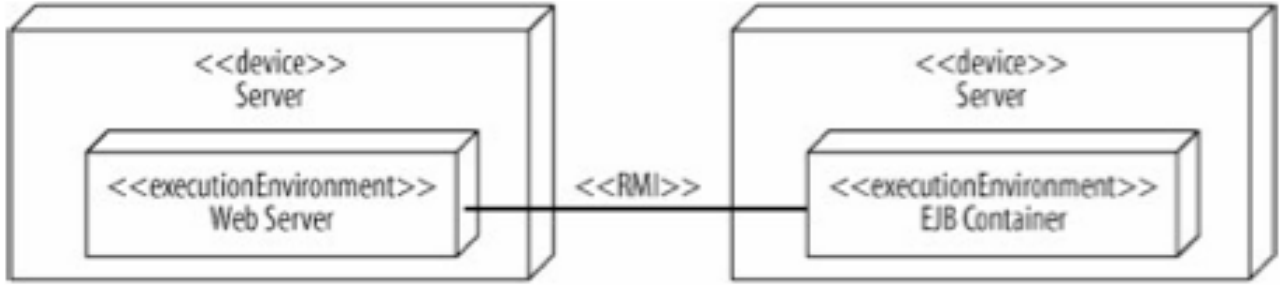
- Para identificar os **ambientes de execução** pode utilizar-se o estereótipo **«executionEnvironment»**



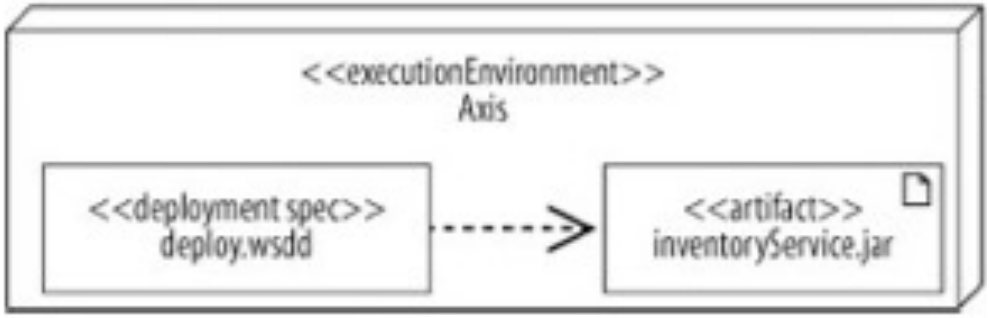
- Comunicação entre dois nós

Diagramas de Instalação

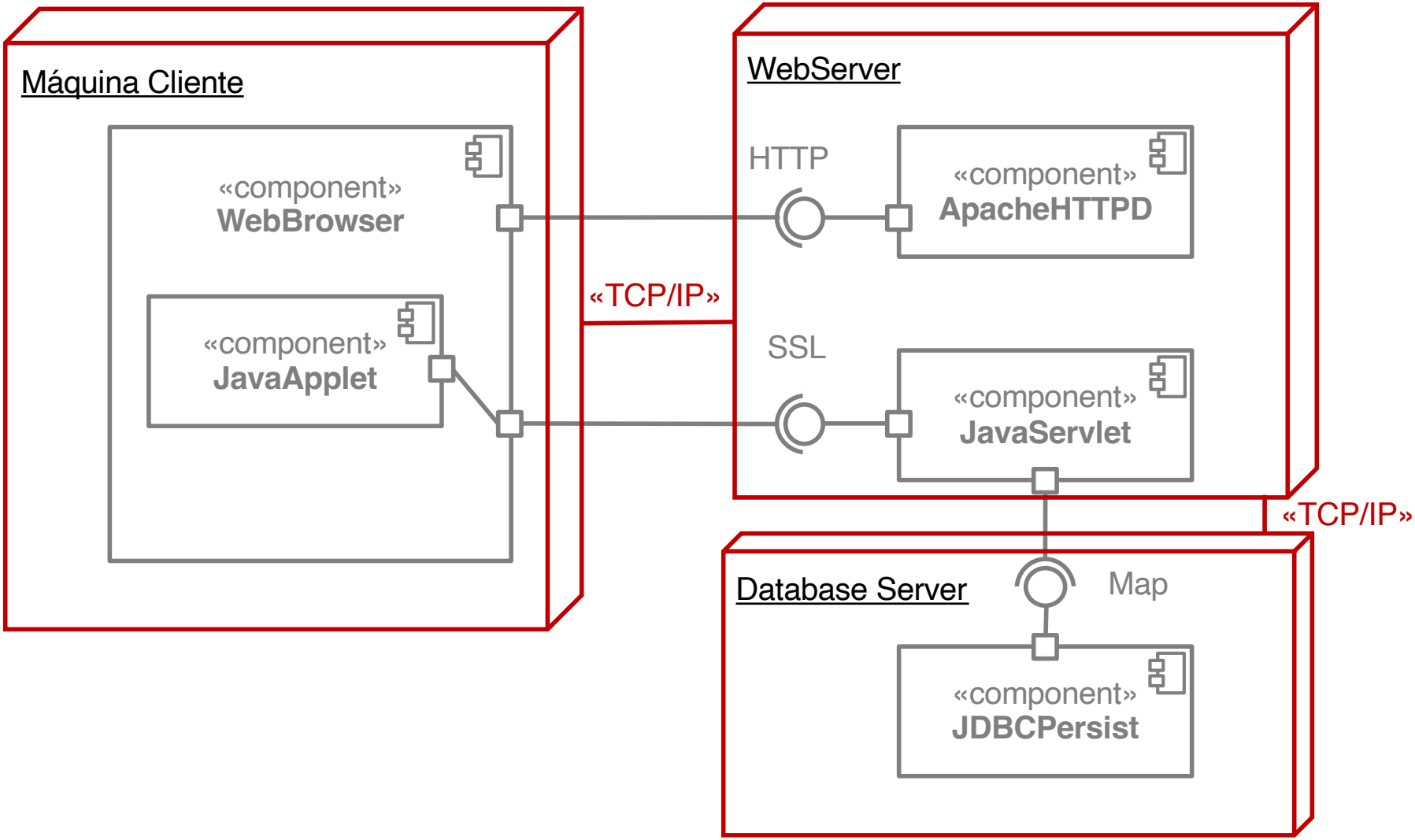
- A descrição pode ser refinada para detalhar os ambientes de execução em cada nó



- Especificação de dependências em tempo de execução

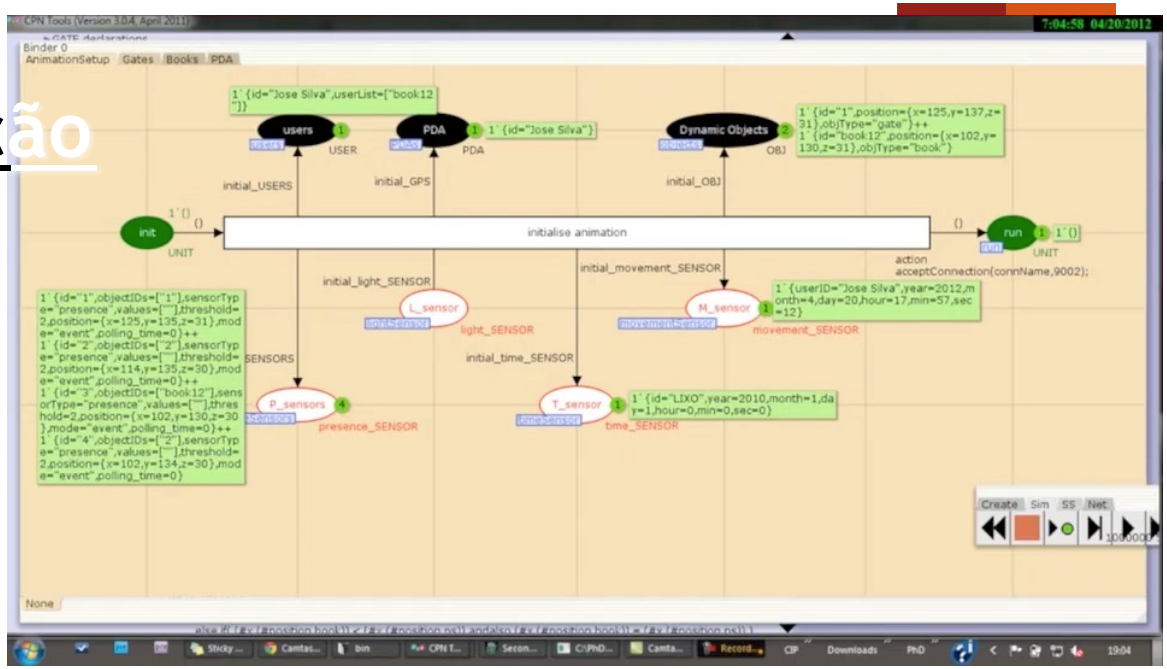


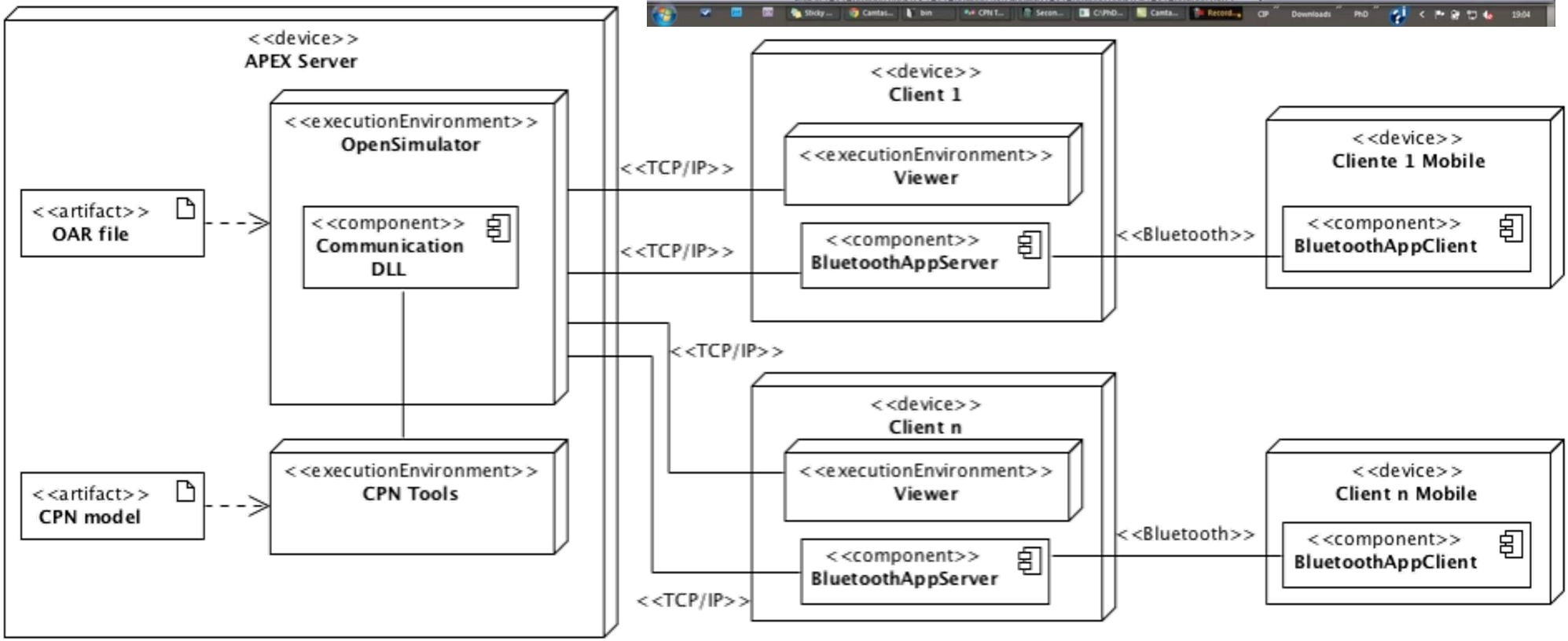
Instalação e componentes



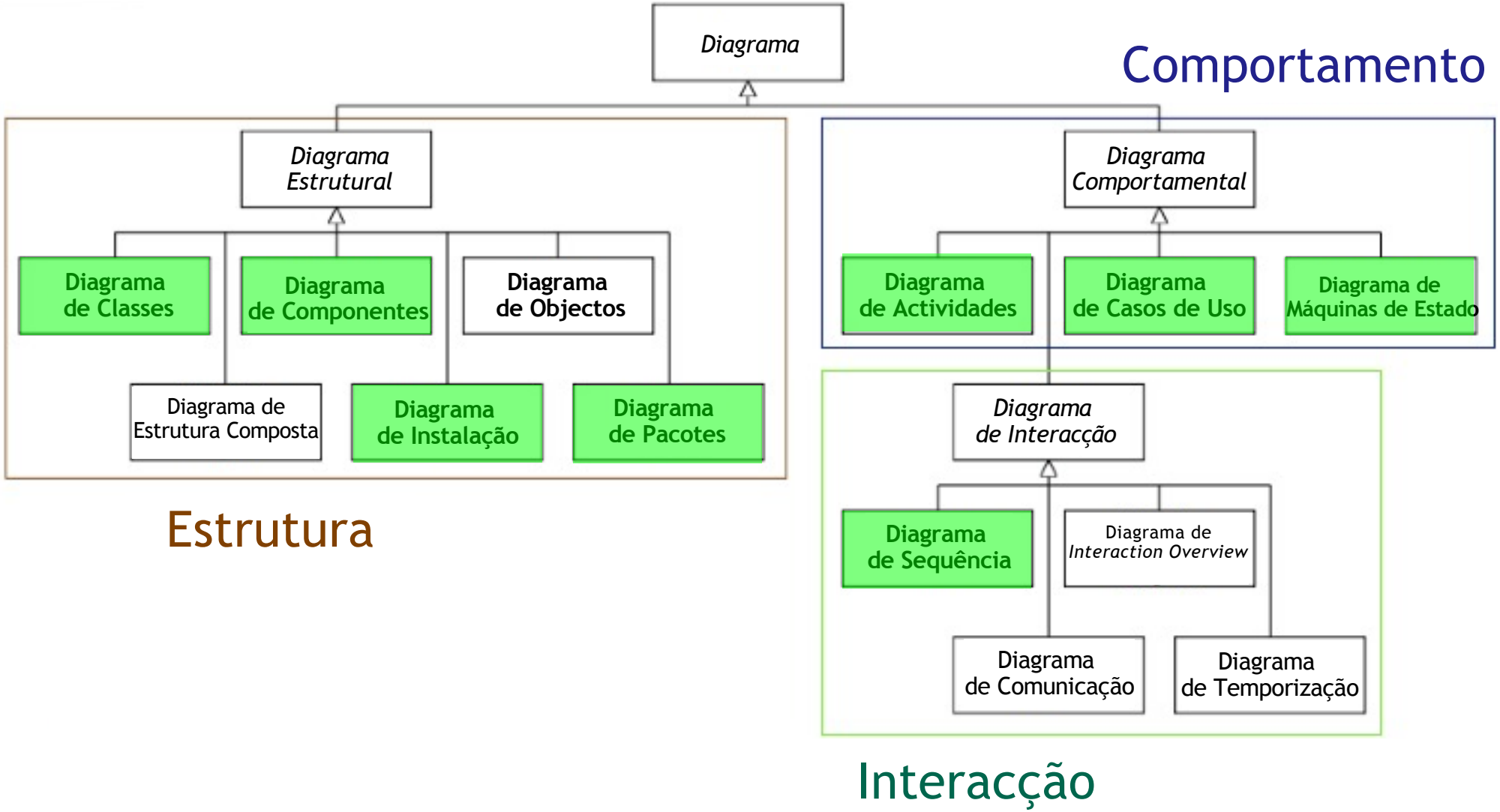
Diagramas de Instalação

- Um exemplo

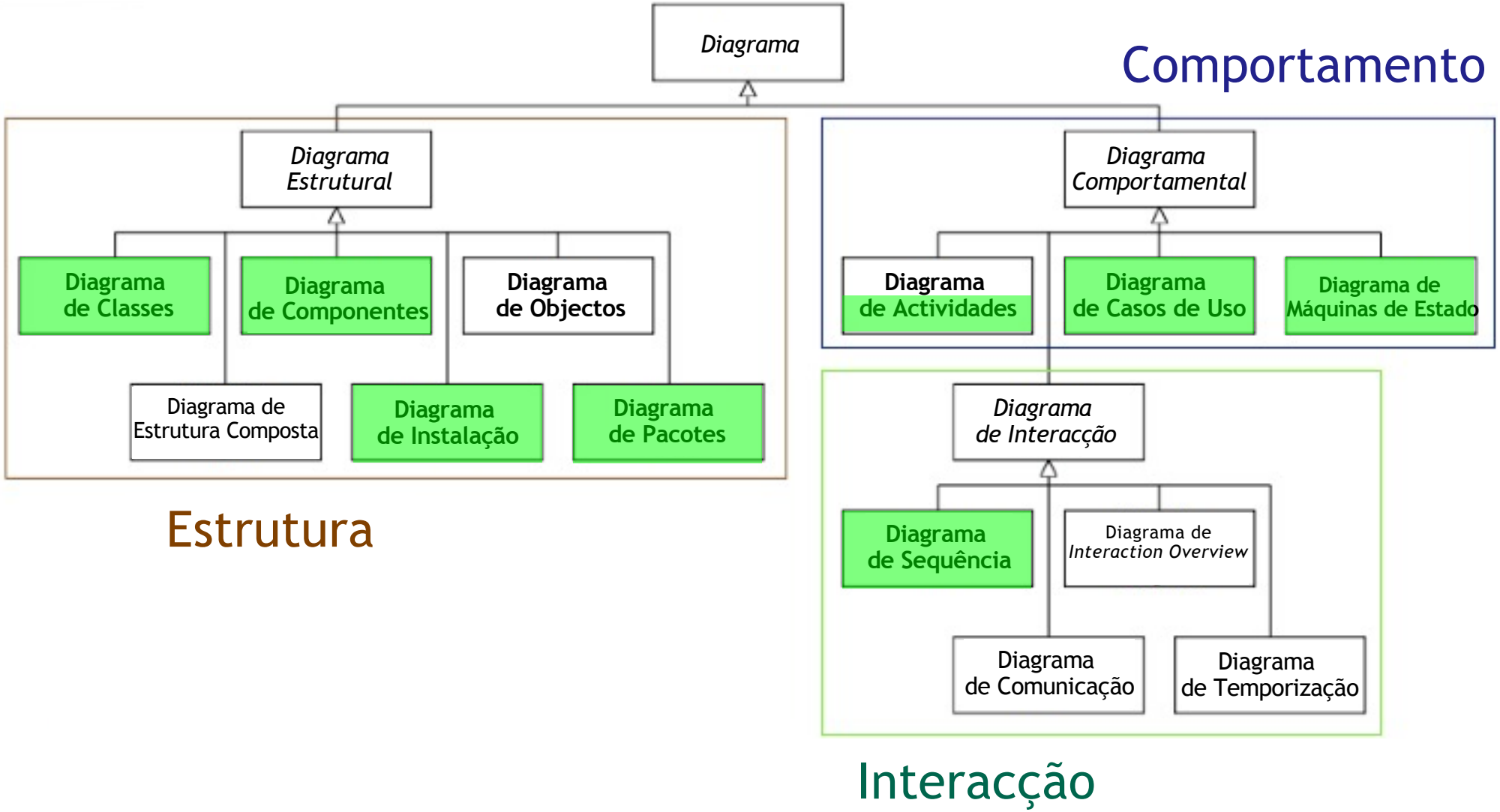




Diagramas da UML 2.x

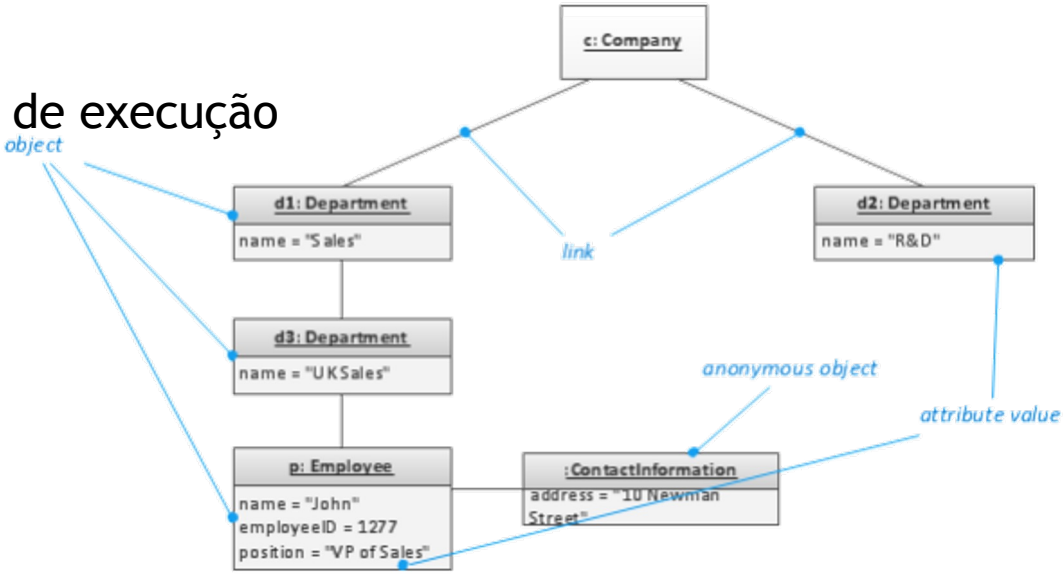


Diagramas da UML 2.x

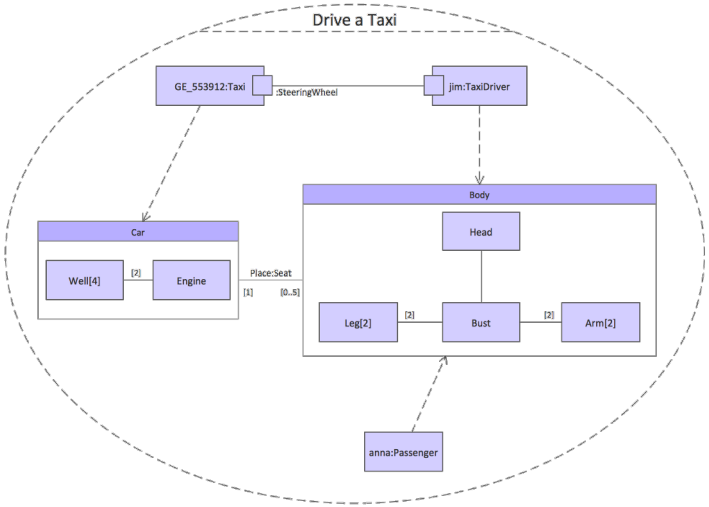


Diagramas de que não falamos

- Diagramas de objectos
  - Mostra instâncias em tempo de execução

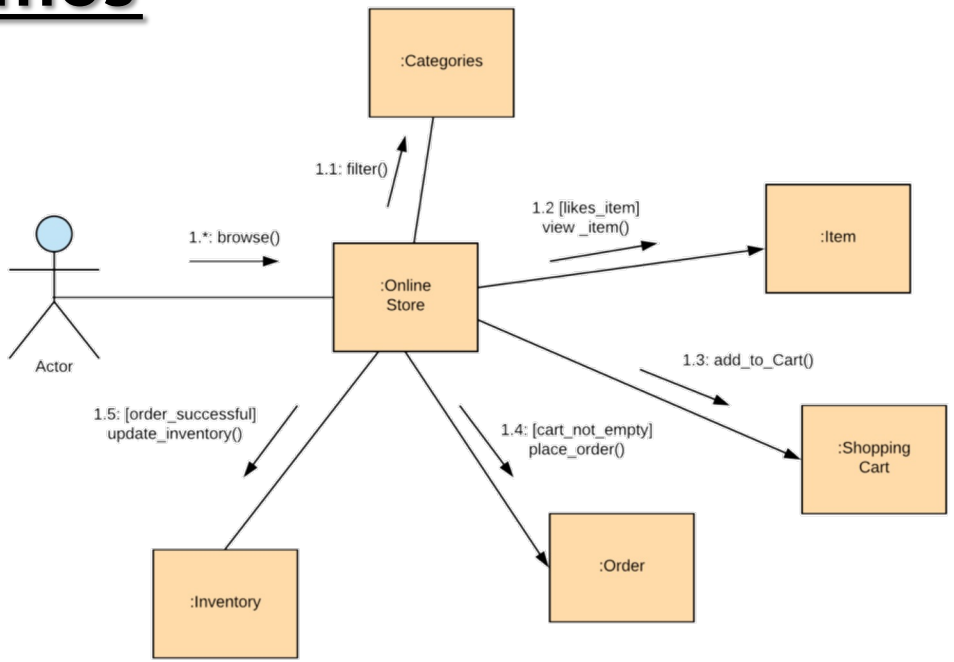


- Diagramas de estrutura composta
  - Mostra a estrutura interna de uma classe em termos de suas partes e das relações entre elas

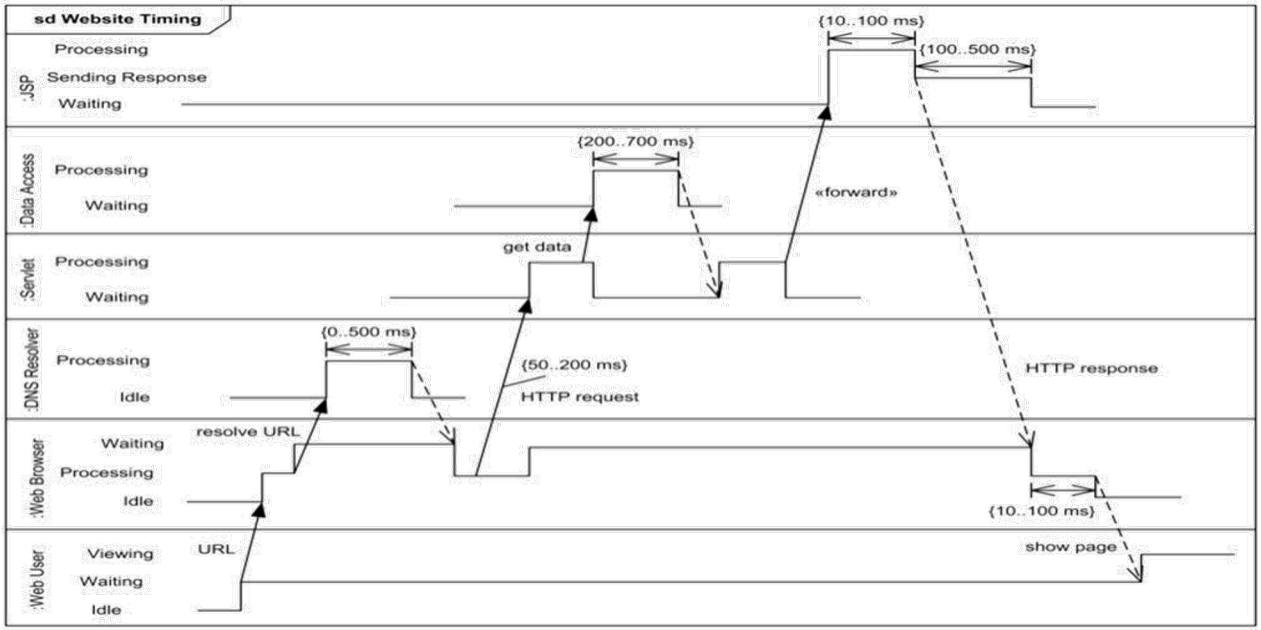


Diagramas de que não falamos

- Diagramas de Comunicação
  - Uma alternativa aos diagramas de sequência, focada na relação estrutural entre os objectos

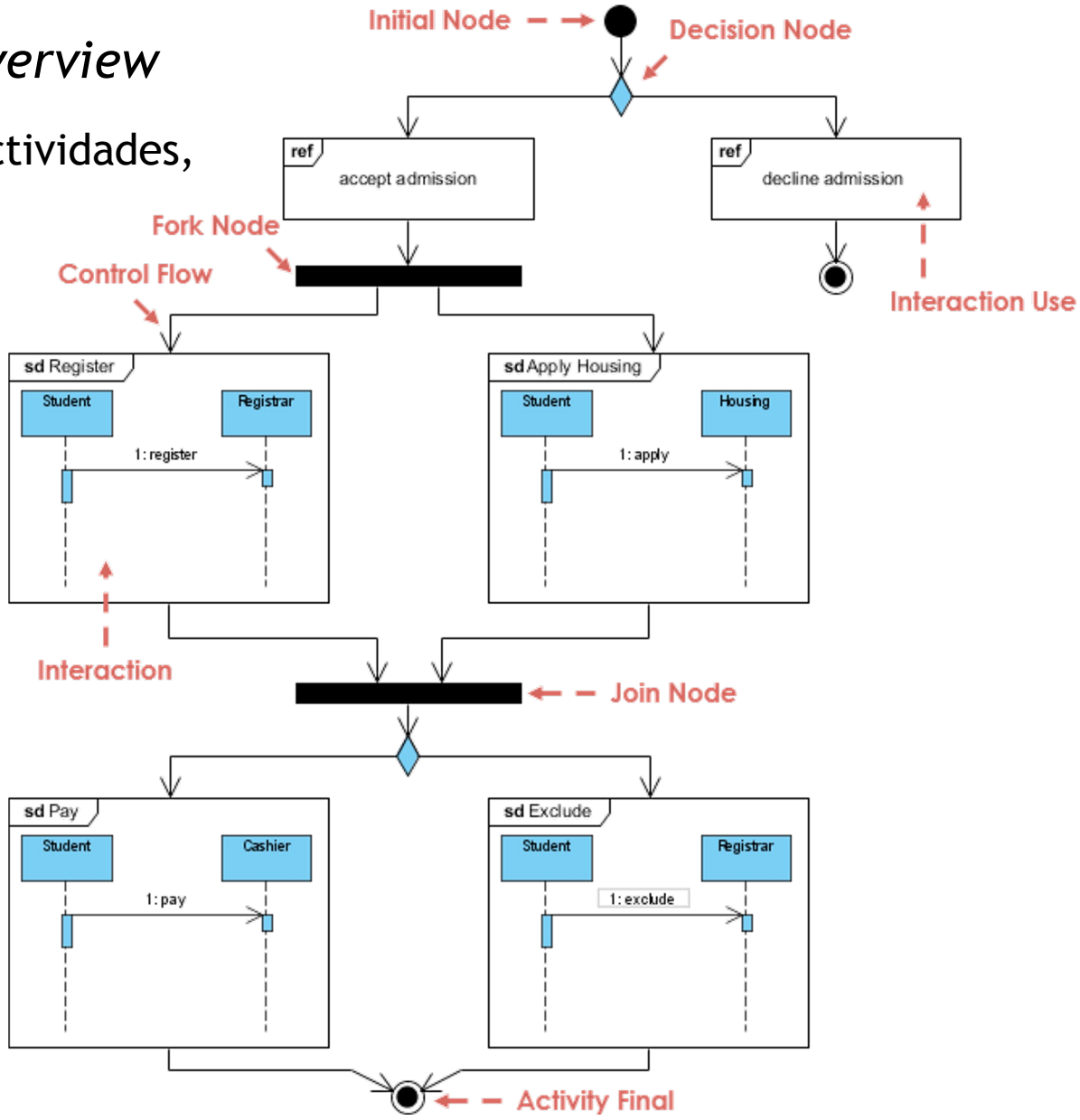


- Diagramas de Temporização
  - Focados na representação de tempo



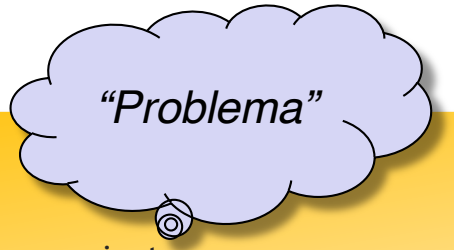
Diagramas de que não falamos

- Diagramas de *Interaction Overview*
  - Semelhante ao diagrama de actividades, com diagramas de sequência como actividades



Notas finais

VOXvote 195188
live.voxvote.com
Nickname: nº aluno!



a)

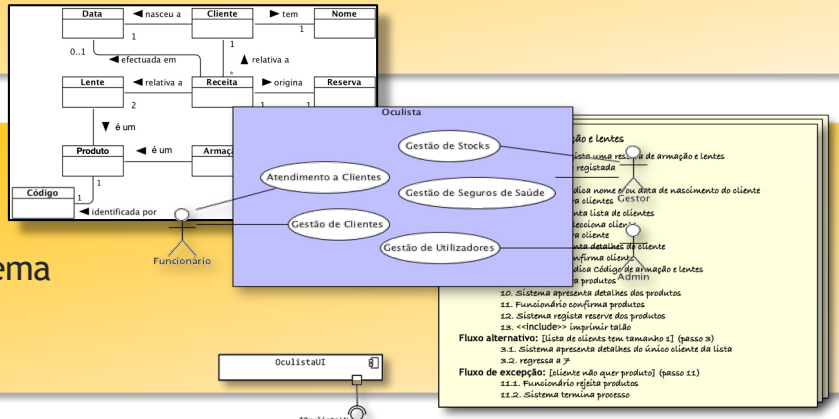
Planeamento

- Decisão de avançar com o projecto
- Gestão do projecto

b)

Análise

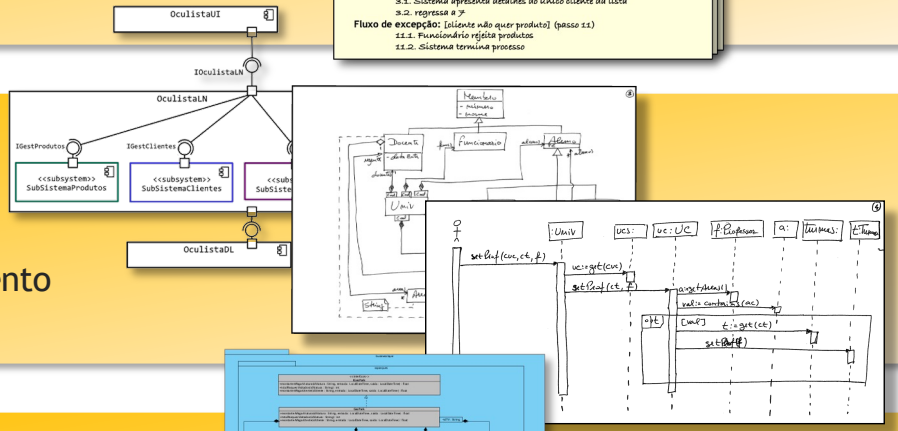
- Análise do domínio do problema
- Análise de requisitos



c)

Concepção

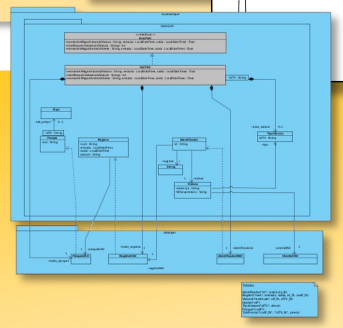
- Concepção da Arquitectura
- Concepção do Comportamento

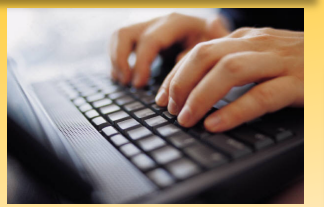


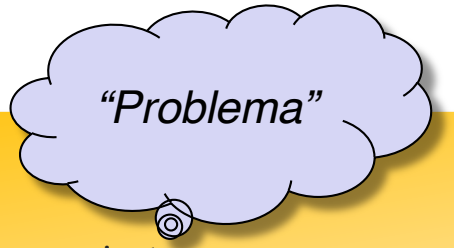
d)

Implementação

- Construção
- Teste
- Instalação
- Manutenção





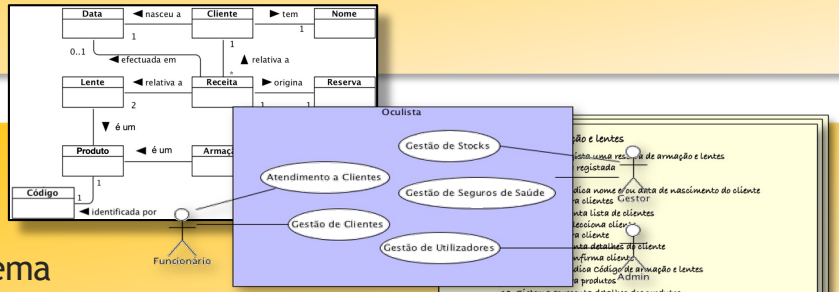


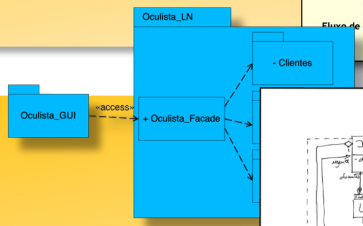
Planeamento

- Decisão de avançar com o projecto
- Gestão do projecto

Análise

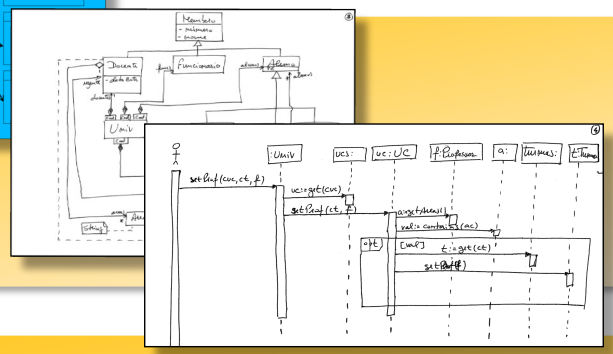
- Análise do domínio do problema
- Análise de requisitos

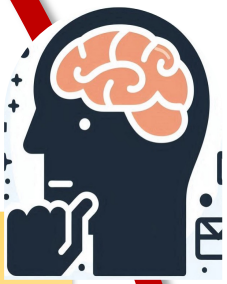




Concepção

- Concepção da Arquitectura
- Concepção do Comportamento





Implementação

- Construção
- Teste
- Instalação
- Manutenção

Avaliação

- Teste/Exame (≥ 8.0) - uma prova escrita sobre a matéria lecionada
  - Teste/Exame de consulta – duas folhas A4.
- Trabalho Prático (≥ 10.0)
- Objetivos
  - Reconhecer os diferentes tipos de diagramas da UML ;
  - Compreender modelos (de requisitos/estruturais/comportamentais) descritos em UML;
  - Conceber sistemas de software utilizando UML;
  - Implementar sistemas de software a partir de modelos UML.
- Classificação Final (**≥ 10.0**)

  .50 Teste/Exame + .50 Trabalho

Sobre o trabalho prático

- Dois *deliverables*:
  - Documento com análise e modelação
  - Uma aplicação informática que permita executar os use cases solicitados

- Na apresentação:
  - Avalia-se a qualidade do processo e como é que cada grupo executou as diferentes fases de análise e modelação
  - Avalia-se a construção da aplicação, na sua lógica multi-camada e na arquitetura apresentada

Teste exemplo

+1/1/60+

Nome: ..... **Número:** .....

☐0	☐0	☐0	☐0	☐0	☐0
☐1	☐1	☐1	☐1	☐1	☐1
☐2	☐2	☐2	☐2	☐2	☐2
☐3	☐3	☐3	☐3	☐3	☐3
☐4	☐4	☐4	☐4	☐4	☐4
☐5	☐5	☐5	☐5	☐5	☐5
☐6	☐6	☐6	☐6	☐6	☐6
☐7	☐7	☐7	☐7	☐7	☐7
☐8	☐8	☐8	☐8	☐8	☐8
☐9	☐9	☐9	☐9	☐9	☐9

DESENVOLVIMENTO DE SISTEMAS DE SOFTWARE

Teste Exemplo

Licenciatura em Engenharia Informática
Departamento de Informática, Universidade do Minho

2024/2025 · Duração máxima: 2h

Instruções:
Assinale as suas respostas com ■. Não se esqueça de preencher o nome e número. Indique também o número na tabela à direita, assinalando um dígito por coluna.

Leia cada questão da prova com atenção!

Parte 1 (8 valores)

Considere o seguinte excerto de código Java:

```
public interface Compras {
  public void compra(String id, String numBilhete);
  ...
}

public class ComprasFacade implements Compras {
  private String nome = "";
  private Map<String,Comprador> compradores;  //idComprador->Comprador
  ...
  public void compra(String id, String numBilhete) { ... }

  private Collection<String> espetaculos() { ... }
  ...
}

public class Comprador extends Entidade {
  private Map<String, List<Bilhete>> bilhetes; //Espetáculo -> Lista de Bilhete
  ...
  public Collection<Bilhete> getBilhetes() { ... }
  ...
}

public abstract class Entidade {
  private String id;                    // identificação da entidade
  ...
  public String getID() { return id; }
}

public class Bilhete { ... }
...
```

Responda às seguintes questões:

Teste exemplo

+1/2/59+

Questão 1 Analise o código e apresente o correspondente Diagrama de Classes, procurando ser o mais exaustivo possível na identificação de classes/interfaces e dos seus relacionamentos. Inclua todas as classes e interfaces que pode deduzir existirem a partir do código. Considere que todas as associações correspondem a composições. (4 valores)

☐0 ☐.1 ☐.2 ☐.3 ☐.4 ☐.5 ☐.6 ☐.7 ☐.8 ☐.9 ☐1

Teste exemplo

+1/3/58+

Questão 2 Pretende-se agora acrescentar à classe ComprasFacade a operação:
`bilhetesDe(id: String, d: Date): Collection<Bilhete>`
que calcula a coleção com todos os bilhetes comprados por um dado comprador antes da data dada. Caso não exista um comprador identificado por `id`, o resultado da operação não está definido. Desenhe um Diagrama de Sequência para o método pretendido e escreva a respetiva pré-condição. (4 valores)

☐0 ☐.1 ☐.2 ☐.3 ☐.4 ☐.5 ☐.6 ☐.7 ☐.8 ☐.9 ☐1

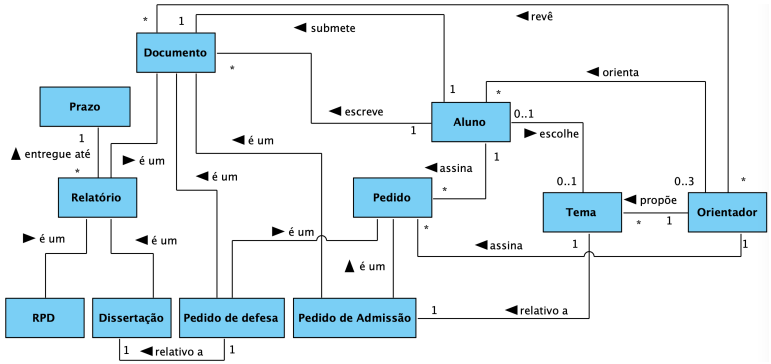
Teste exemplo

Parte 2 (7 valores)

Considere a seguinte descrição do processo de realização de uma dissertação de mestrado:

> O processo de realização de uma dissertação de mestrado, num dado tema, passa por várias fases até a dissertação estar publicada. O processo começa com a proposta do tema, ficando este a aguardar candidaturas. Após algum aluno se candidar ao tema, este passa a estar escolhido. O orientador decide então se aceita ou rejeita a candidatura do aluno ao tema. No caso de o orientador não aceitar a candidatura, o processo relativo a este tema regressa à fase de aguardar candidaturas. Pode ficar nesta fase até à "data limite de candidaturas". Se essa data for atingida, o processo termina.
>
> No caso de o orientador aceitar a candidatura, dois sub-processos passam a decorrer em paralelo. Um relativo ao tema, outro à dissertação. Por um lado, o tema passa ao estado de atribuído ao aluno. O aluno deverá então submeter um pedido de admissão à dissertação, ficando o tema pendente de aprovação. Após ser aceite pela Direcção de Curso, o tema passa a aceite.
>
> Por outro lado, a dissertação passa ao estado de escrita do relatório de pré-dissertação (RPD). Após a entrega do RPD, a dissertação passa ao estado de desenvolvimento (no qual o aluno está continuamente a escrever), até que o orientador considere a dissertação concluída, ou se atinja o "prazo de entrega". Caso o prazo de entrega seja atingido sem que a dissertação esteja pronta, o processo de realização da dissertação termina.
>
> Estando a dissertação concluída e o tema aceite, o aluno submete o pedido de defesa da dissertação, passando o processo à fase de aguardar defesa. Após a defesa, a dissertação pode ser aceite ou recusada pelo júri. Se for aceite, passa a publicada. Caso contrário, todo o processo de realização da dissertação termina.

Considere ainda que, no contexto do desenvolvimento de um sistema para a direcção de curso, foi proposto o modelo de domínio apresentado na figura.



Responda às seguintes questões:

Teste exemplo

+1/5/56+

Questão 3 Com base na descrição do processo e no modelo de domínio apresentados, especifique, num Diagrama de *Use Case*, os requisitos funcionais de um sistema que suporte uma direcção de curso na gestão do processo de realização das dissertações. O sistema deve permitir desde a submissão de propostas pelos orientadores, até à submissão das dissertações pelos alunos e posterior indicação da decisão do júri, pela direcção de curso. (4 valores)

☐0 ☐.1 ☐.2 ☐.3 ☐.4 ☐.5 ☐.6 ☐.7 ☐.8 ☐.9 ☐1

Teste exemplo

+1/6/55+

Questão 4 Utilizando um Diagrama de Máquina de Estados modelo o ciclo de vida do processo de realização de uma dissertação. Os estados da máquina deverão representar os estados porque passa o processo de realização da dissertação e as transições deverão representar as acções das entidades envolvidas (aluno, docente, direcção de curso) (2 valores)

Explique, de forma breve, a forma como modelou os prazos limite (*deadlines*) para as candidaturas, entrega do RPD e da dissertação. (1 valor)

☐0 ☐.1 ☐.2 ☐.3 ☐.4 ☐.5 ☐.6 ☐.7 ☐.8 ☐.9 ☐1

Teste exemplo

Parte 3 (5 valores)

Questão 5 Qual a frase que melhor descreve a seguinte expressão OCL? (1 valor)

```
context Encomenda::processItem(item: Item) : void
post: self.items->forall(i | i <> item)
```

- [ ] Todas as encomendas são *items*.
- [ ] O método não pode ser executado se o `item` não estiver na encomenda.
- [ ] Depois de o método ter sido executado, o `item` passado como parâmetro não deve existir nos items da encomenda.
- [ ] O método só pode ser executado se o `item` não estiver na encomenda.
- [ ] Depois de o método ter sido executado, o `item` passado como parâmetro deve existir nos items da encomenda.
- [ ] Nenhuma das anteriores

Questão 6 Considere o modelo de domínio apresentado anteriormente. Qual das seguintes frases é uma **descrição verdadeira da informação apresentada no diagrama**? (1 valor)

- [ ] Todos os orientadores de um aluno assinam o pedido de admissão desse aluno.
- [ ] Nenhum dos orientadores de um aluno assina o pedido de defesa desse aluno.
- [ ] Um aluno é orientado por quem propôs o tema da dissertação.
- [ ] Cada aluno tem obrigatoriamente um orientador.
- [ ] O pedido de defesa de um aluno tem que estar assinado por um orientador.
- [ ] Nenhuma das anteriores

Questão 7 ♣ Considere que está a desenvolver um editor gráfico, em que existem diversas formas geométricas que é possível adicionar à imagem em edição e sobre as quais é possível calcular o perímetro, a área, etc. Relembre os princípios de design orientado a objetos discutidos nas aulas, analise as afirmações que se seguem e indique as que considera verdadeiras: (1 valor)

- [ ] O Open/Closed Principle, ao dizer que a arquitetura do editor deve permitir que a adição de nova funcionalidade seja feita através da extensão de entidades existentes, em vez de através da sua modificação, implica que é preferível definir uma hierarquia de classes para representar as formas geométricas, em vez de ter uma única classe com um atributo que diz qual o tipo da forma geométrica.
- [ ] O Dependency Inversion Principle (DIP), ao dizer que o código do editor deve depender de interfaces em vez de classes concretas, implica que cada classe representativa de uma forma geométrica deve implementar uma interface correspondente (a classe Quadrado implementa a interface IQuadrado, etc.).
- [ ] O Single Responsibility Principle, ao dizer que cada classe do editor deve ter uma e apenas uma razão para mudar, implica que cada classe deve ter apenas um atributo.
- [ ] O Single Layer of Abstraction Principle, ao dizer que não devemos misturar diferentes níveis de abstração no mesmo código, promove que os atributos de cada classe sejam apenas acedidos dentro da própria classe.
- [ ] Nenhuma das anteriores

Teste exemplo

Questão 8 Considere que se vai desenvolver uma arquitetura a partir do modelo de domínio apresentado anteriormente. Sabendo que cada documento é identificado por um número e que se pretende que os alunos tenham acesso fácil ao documento que submetem, quais das seguintes afirmações considera verdadeira? (1 valor)

- ☐ A classe Aluno deverá ser uma subclasse de Documento.
- ☐ A classe Aluno deverá ter uma associação para o documento que o aluno submeteu.
- ☐ A classe Documento deverá ser uma subclasse de Aluno.
- ☐ A classe Aluno deverá ter um atributo com o número do documento que o aluno submeteu.
- ☐ A classe aluno deverá ser uma interface.
- ☐ Nenhuma das anteriores

Questão 9 ♣ Quais dos seguintes diagramas UML são os mais adequados para modelar a forma como um sistema reage a eventos externos? (1 valor)

- ☐ Diagrama de Máquinas de Estado
- ☐ Diagrama de Sequência
- ☐ Diagrama de Instalação
- ☐ Diagrama de Classes
- ☐ Diagrama de Use Case
- ☐ Diagrama de Actividades
- ☐ Nenhuma das anteriores