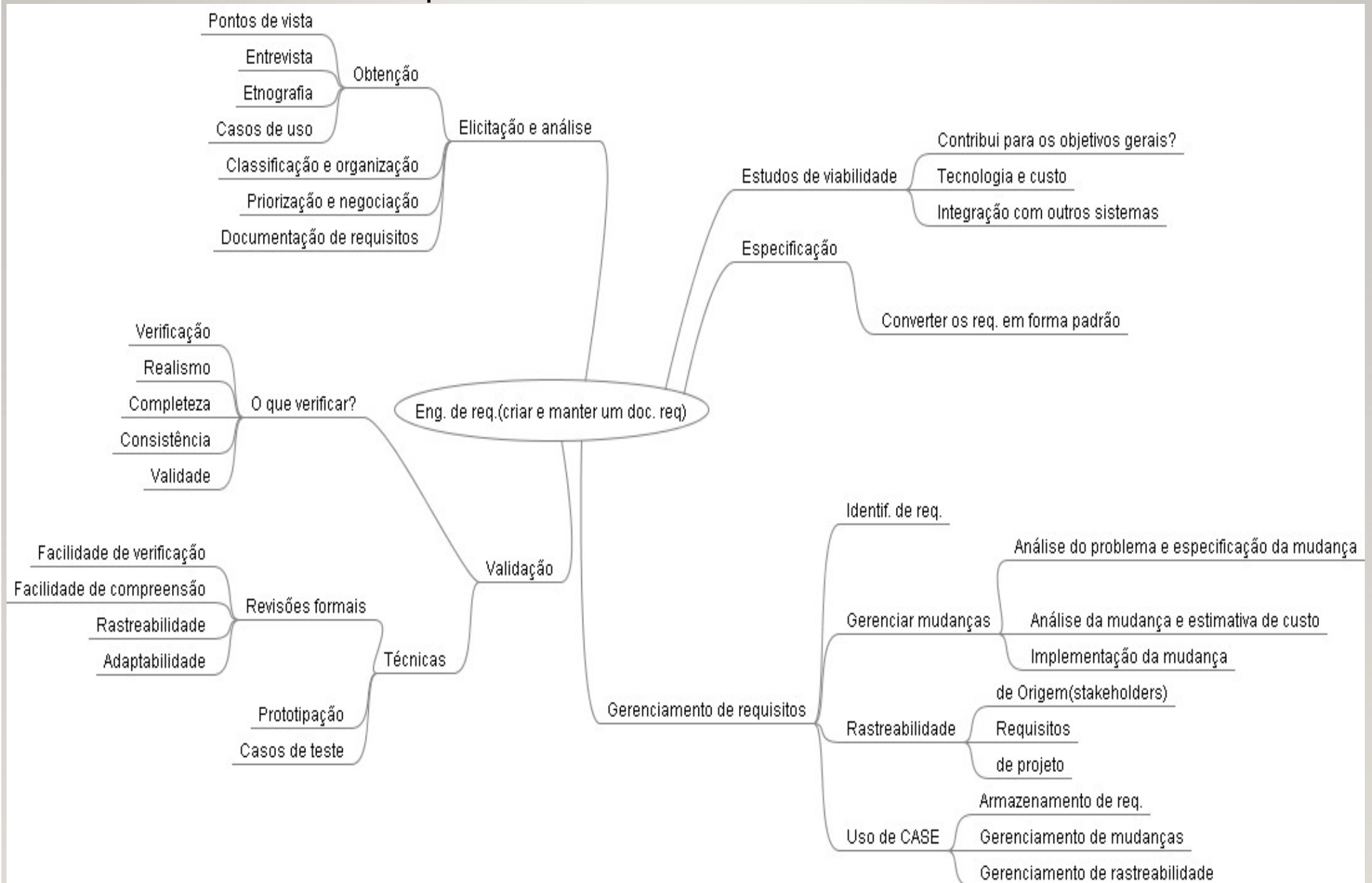


ENGENHARIA DE SOFTWARE 2 – AULA 6

PROF^a M^a DENILCE VELOSO
DENILCE.VELOSO@FATEC.SP.GOV.BR
DENILCE@GMAIL.COM

Recapitulando...



PROJETO E IMPLEMENTAÇÃO

Como dito anteriormente, apesar da diversidade, todo Processo de Software devem incluir de alguma forma as quatro atividades fundamentais (cada uma pode ser dividida em subatividades):

1. Especificação → **Projeto**
2. Desenvolvimento → **Implementação**
3. Validação → Testes
4. Evolução

PROJETO E IMPLEMENTAÇÃO

Envolve:

- Projeto Orientado a Objetos com UML -> alguns itens em ES2, outros em ES3 e LES
- Padrões de Projeto (Ex.: Projeto Singleton, Projeto Factory Method, Observer, Strategy etc) ⓘ
- Questões de Implementação (linguagens, frameworks, bibliotecas etc.)
- Desenvolvimento Open Source



<https://www.devmedia.com.br/conheca-os-padroes-de-projeto/957>

Slide 4

DV0

Padrões de Projeto - Reutilização de código, comunicação (linguagem comum), boas práticas, manutenção e evolução, soluções comprovadas (alto nível de abstração)

Denilce Veloso; 2024-09-19T17:32:36.661

DV0 0

Framework - Específicos para uma área ou linguagem.

Denilce Veloso; 2024-09-19T17:34:19.607

PROJETO E IMPLEMENTAÇÃO

Inclui pelo menos três fases principais:

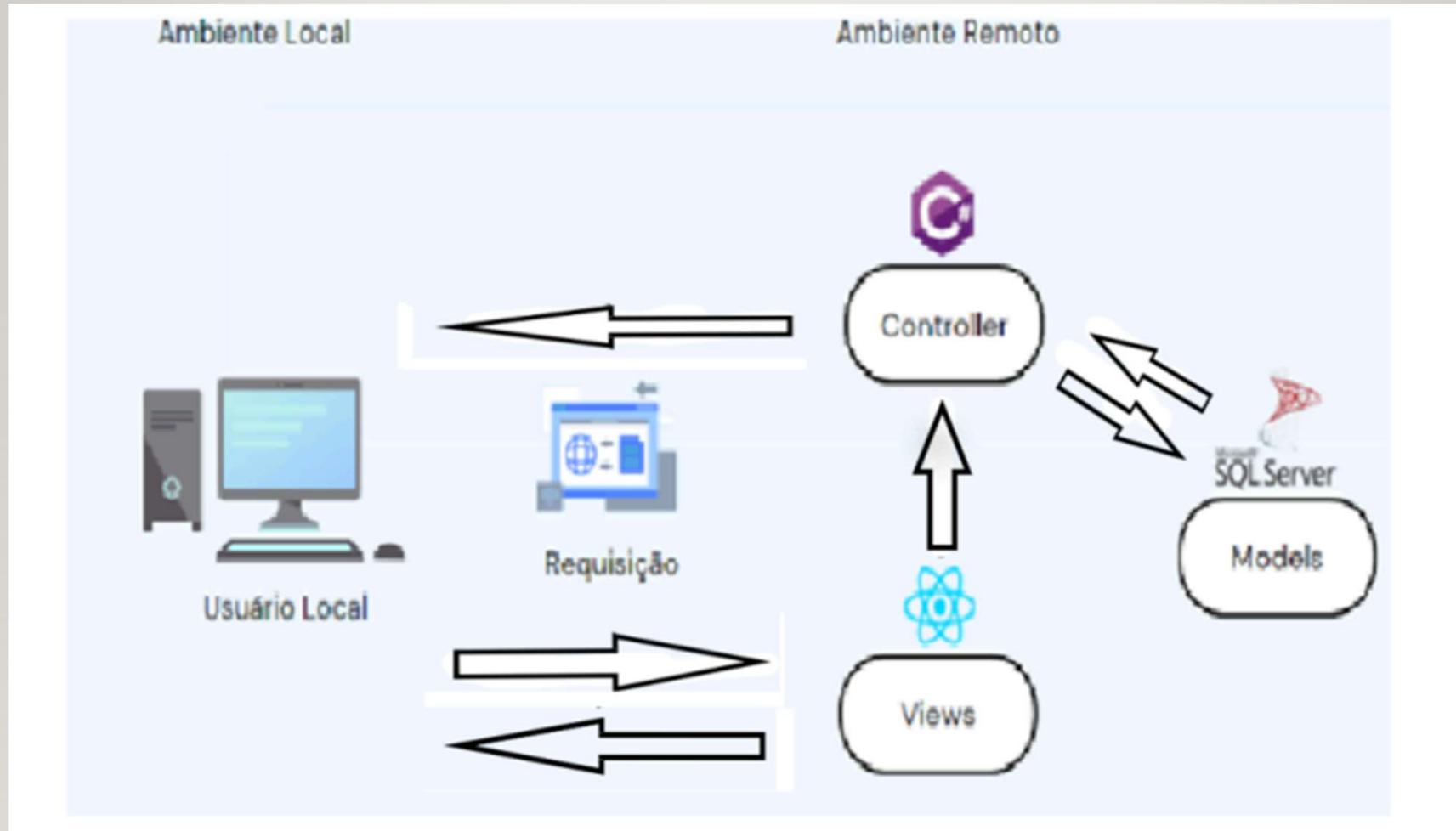
- Projeto **arquitetural** – define a estrutura modular do software (MVC, Camadas, Serviços etc), as interfaces e as estruturas de dados utilizadas (SQL, NoSQL). -> alguns itens são vistos em ES2 e outros ES3
- Projeto **detalhado** - define a solução para cada módulo do projeto preliminar. Ex. autenticação, gerenciamento usuário, pagamentos etc.(futuro-ES3)
- **Implementação** – transcreve as decisões de projeto para linguagem de programação. (futuro-ES3 e LESS)



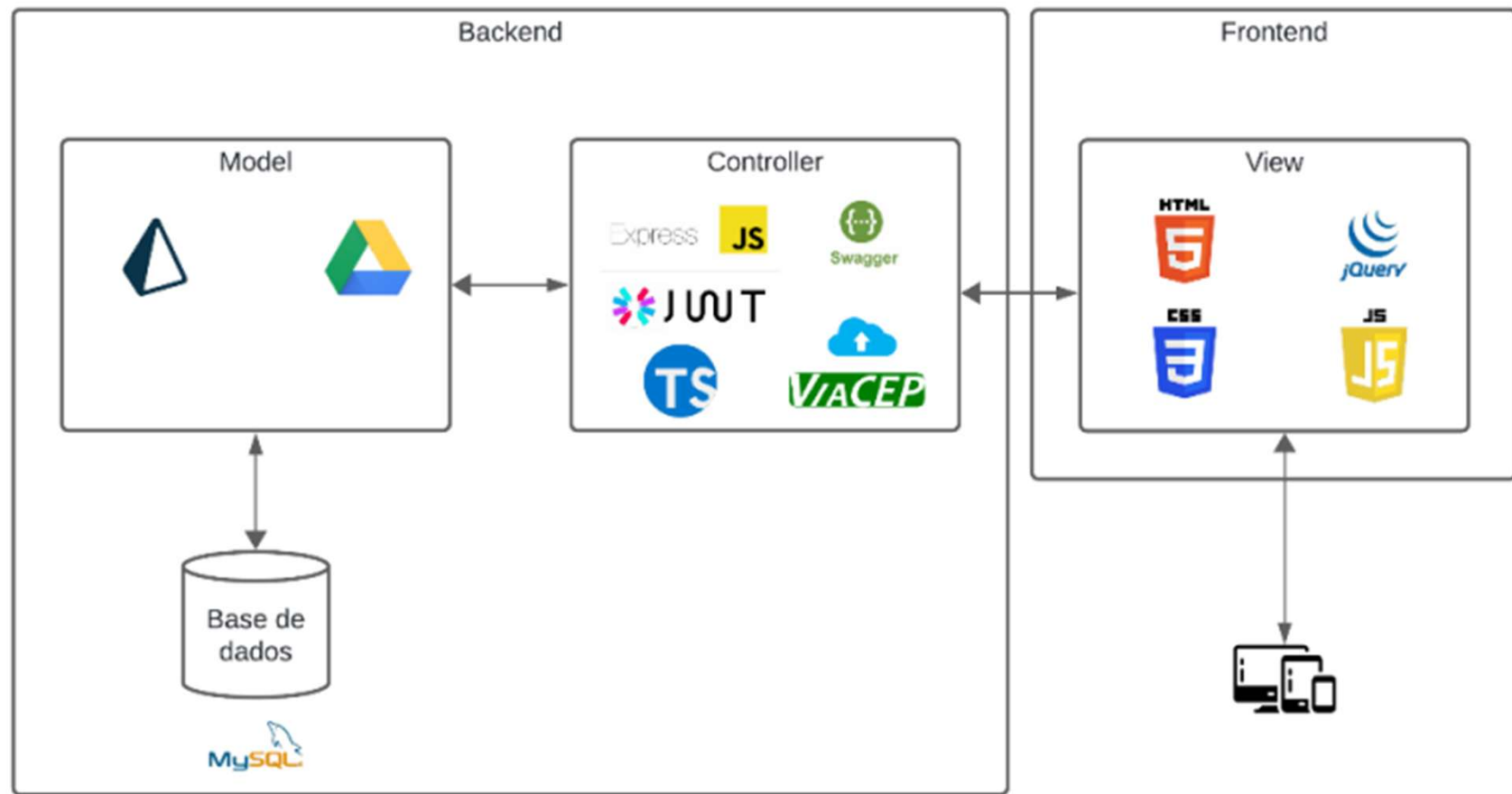
PROJETO ORIENTADO A OBJETOS COM UML – PROJETO ARQUITETURAL

- Definidas as interações entre o sistema e seu ambiente → definir projeto de arquitetura do sistema.
- Identificam-se os **componentes principais que compõem o sistema** e suas interações, e então pode-se organizar os componentes usando um padrão de arquitetura como por exemplo, um modelo em camadas ou cliente-servidor.

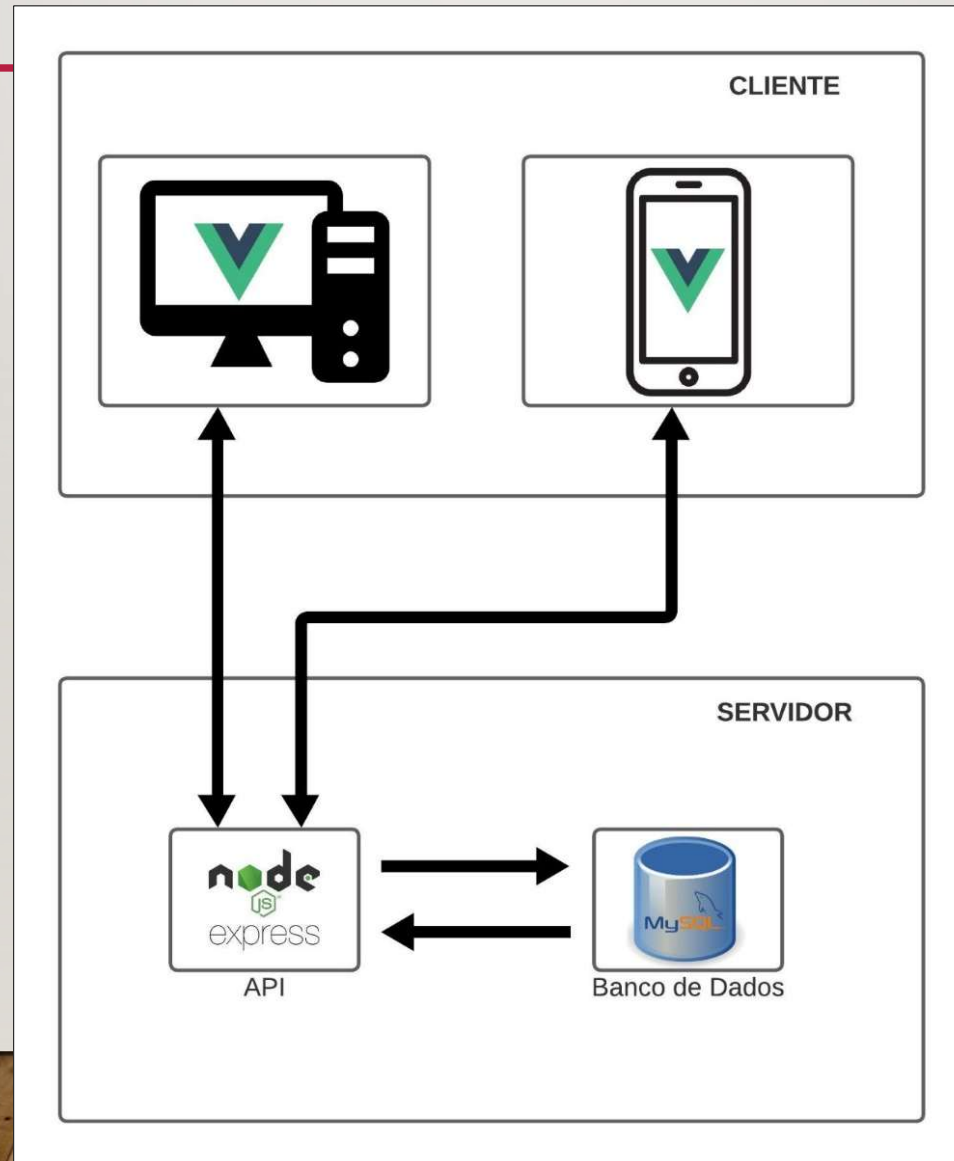
PROJETO E IMPLEMENTAÇÃO – ARQUITETURAL (EXEMPLO)



PROJETO E IMPLEMENTAÇÃO – ARQUITETURAL (EXEMPLO)



PROJETO E IMPLEMENTAÇÃO – ARQUITETURAL (EXEMPLO)



PROJETO E ORIENTAÇÃO A OBJETOS

- Em um processo de desenvolvimento **orientado a objetos**, o projeto da arquitetura normalmente é realizado por um arquiteto de software. O **projeto da arquitetura** visa distribuir as **classes** de objetos **relacionados** do sistema em subsistemas e seus componentes, distribuindo também esses componentes pelos recursos de hardware disponíveis.

IMPLEMENTAÇÃO

DV0

-
- É o estágio no processo de engenharia de software no qual um **sistema de software executável é desenvolvido**.
 - As atividades de projeto e implementação são invariavelmente **intercaladas**.
 - O **projeto de software** é uma **atividade criativa** na qual você identifica componente de software e seus relacionamentos baseando-se nos requisitos do cliente.
 - A **implementação** é o processo **de realização do projeto** em um programa de computador.



DV0

na prática, essas atividades estão intimamente relacionadas e muitas vezes ocorrem simultaneamente. O processo de desenvolvimento de software não é linear, e não é possível definir todas as especificações técnicas e detalhes de implementação antes de começar a escrever o código fonte.

DENILCE DE ALMEIDA OLIVEIRA VEL; 2023-03-30T13:44:25.259

PROJETO ORIENTADO A OBJETOS COM UML

- O que é UML?

A UML - *Unified Modeling Language* é **uma linguagem gráfica para visualizar, especificar, construir e documentar um software** através dos diagramas que são compostos de gráficos, símbolos e texto permitindo compreender de forma detalhada, conclusiva, sem ambiguidades e seus modelos pode ser conectados a diferentes linguagens de programação. Em outras palavras, diagramas são imagens, figuras ou esboço do projeto desenhado no papel.



PROJETO ORIENTADO A OBJETOS COM UML

- Os **processos de projeto orientados a objetos** envolvem o desenvolvimento de vários modelos (Caso de uso, Classes, Sequência etc.) diferentes de sistema.
- Eles **precisam de muito esforço no desenvolvimento e na manutenção desses modelos.**
- Código mais **organizado**, facilita **manutenção, reutilizável** e com **menos erros.**
- **Não é uma única abordagem** para todos os problemas de desenvolvimento de software, mas sim uma abordagem que pode ajudar a criar sistemas mais robustos e fáceis de manter.
- Analisar **custo-benefício.**



DV0

Abordagem estruturada: decomposição do sistema em módulos

Abordagem baseada em componentes: projetado como um conjunto de componentes independentes.

Abordagem ágil: A abordagem ágil é uma abordagem iterativa e incremental para o desenvolvimento de software, que se concentra em entregas frequentes e na resposta rápida às mudanças.

Abordagem model-driven: o modelo do sistema é criado primeiro e, em seguida, o código é gerado a partir do modelo. Isso permite que o desenvolvedor se concentre no design do modelo e não no código.

Abordagem baseada em processos: foco está nos processos de desenvolvimento de software, em vez de em uma abordagem específica.

DENILCE DE ALMEIDA OLIVEIRA VEL; 2023-03-30T13:54:14.606

PROJETO ORIENTADO A OBJETOS COM UML - ESTÁGIOS DO PROCESSO

Existe uma grande variedade de diferentes processos de projeto orientados a objetos, essa escolha dependerá da organização que está usando o processo. As atividades comuns nesses processos incluem:

- A **definição do contexto e interações do sistema**;
- O projeto de **arquitetura** do sistema;
- A **identificação dos principais objetos** de classe do sistema;
- O **desenvolvimento dos modelos** de projeto;
- As **especificações de interface** de objetos.



PROJETO ORIENTADO A OBJETOS COM UML - CONTEXTO E INTERAÇÕES DE SISTEMA

- Entender os relacionamentos entre o software que está sendo desenvolvido e seu ambiente externo é essencial na decisão de como prover a funcionalidade requerida para o sistema e como estruturar o sistema para se comunicar com seu ambiente. Ex. E-Commerce.
- Entender o contexto também lhe permite estabelecer os limites do sistema. Estabelecer os limites do sistema ajuda a decidir quais recursos serão implementados no sistema que está sendo desenvolvido e quais serão implementados em outros sistemas associados. Ex. Loja que vende fisicamente e virtualmente.

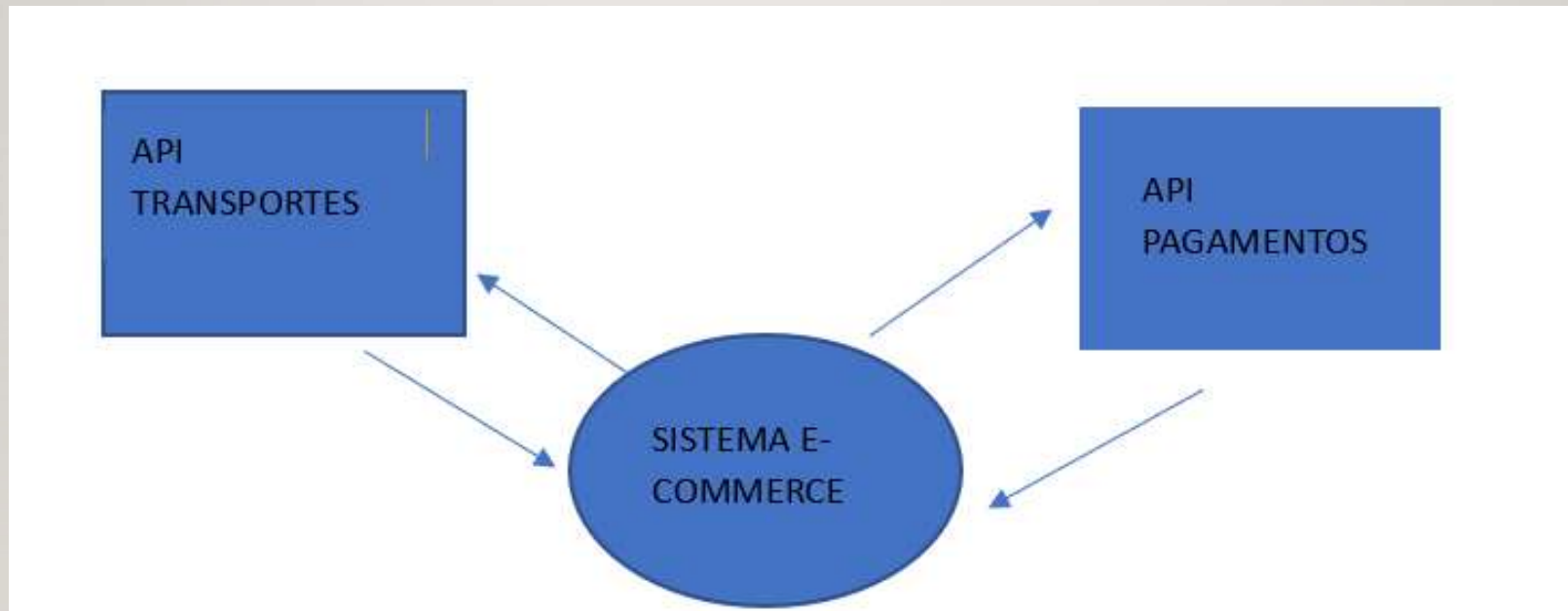


PROJETO ORIENTADO A OBJETOS COM UML – MODELOS DE CONTEXTO E INTERAÇÕES DE SISTEMA

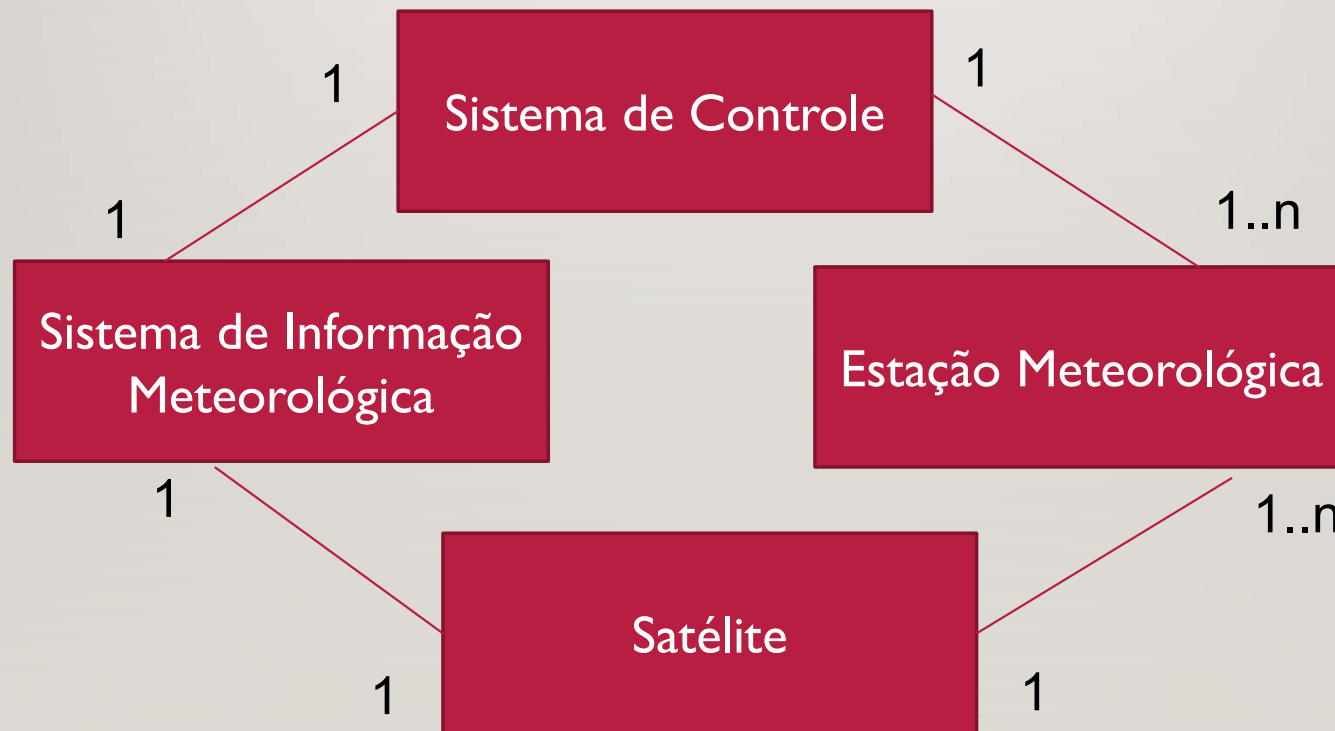
- Um modelo de contexto de sistema é um modelo estrutural que **mostra outros sistemas no ambiente do sistema que está sendo desenvolvido.**
- Um modelo de interação é um modelo dinâmico que mostra como o sistema interage com seu ambiente, durante o seu uso.



PROJETO ORIENTADO A OBJETOS COM UML – EXEMPLO: CONTEXTO DO SISTEMA DE E-COMMERCE



PROJETO ORIENTADO A OBJETOS COM UML – EXEMPLO: CONTEXTO DO SISTEMA ESTAÇÃO METEOROLÓGICA NO DESERTO



*** O exemplo é de um sistema que possui uma estação que coleta dados meteorológicos no deserto.*

DV0

1. Estação Meteorológica (no local do deserto):

Função Principal: Coletar dados meteorológicos diretamente no ambiente do deserto. Tem dispositivos como: sensores, sistema de comunicação (satélite).

2. Sistema de Controle (local centralizado, possivelmente mais distante do deserto):

Função Principal: Receber os dados das diversas estações meteorológicas espalhadas pelo deserto, processá-los e monitorar o funcionamento do sistema.

3. Sistema de Informação Meteorológica (acesso aos usuários):

Função Principal: Fornecer acesso aos dados meteorológicos coletados e processados para diversos usuários e aplicações. Com acesso às interfaces, geração relatórios etc.

4. Satélite (pode ser parte da infraestrutura de comunicação ou fonte de dados adicional).

DENILCE DE ALMEIDA OLIVEIRA VEL; 2025-03-17T22:33:59.886

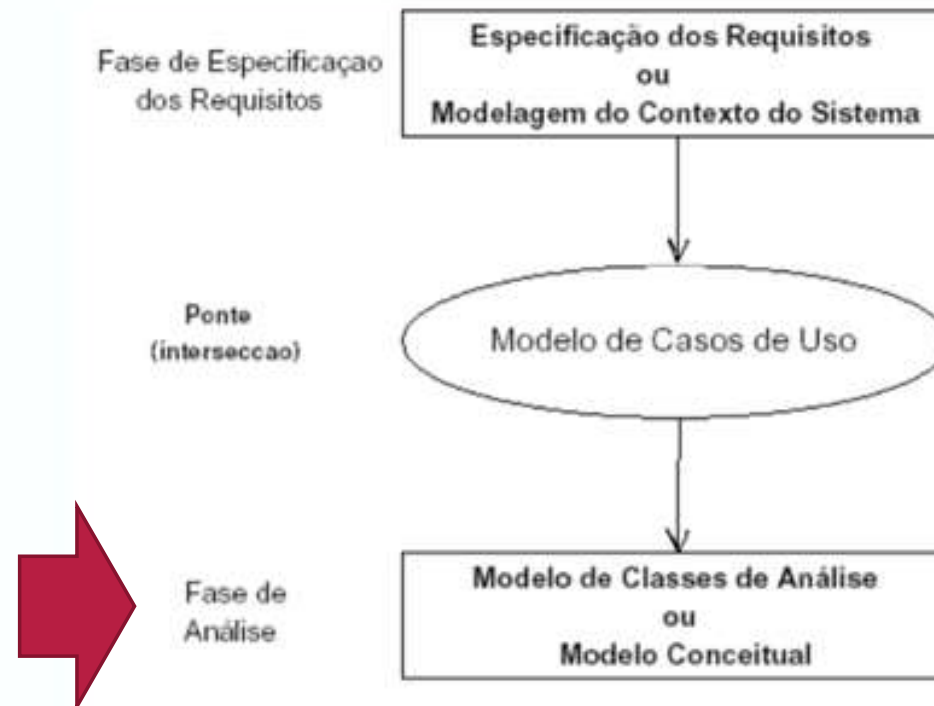
PROJETO ORIENTADO A OBJETOS COM UML

Como já visto anteriormente os objetivos principais são:

- Delimitação do contexto de um sistema;
- Documentação e o entendimento dos requisitos;
- Descrição dos requisitos funcionais;
- Principal saída da etapa de especificação de requisitos;
- **Principal entrada da etapa de análise.**

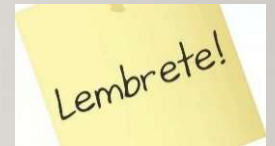
PROJETO ORIENTADO A OBJETOS COM UML – PONTE ENTRE REQUISITOS E ANÁLISE

Ponte entre Requisitos e Análise:



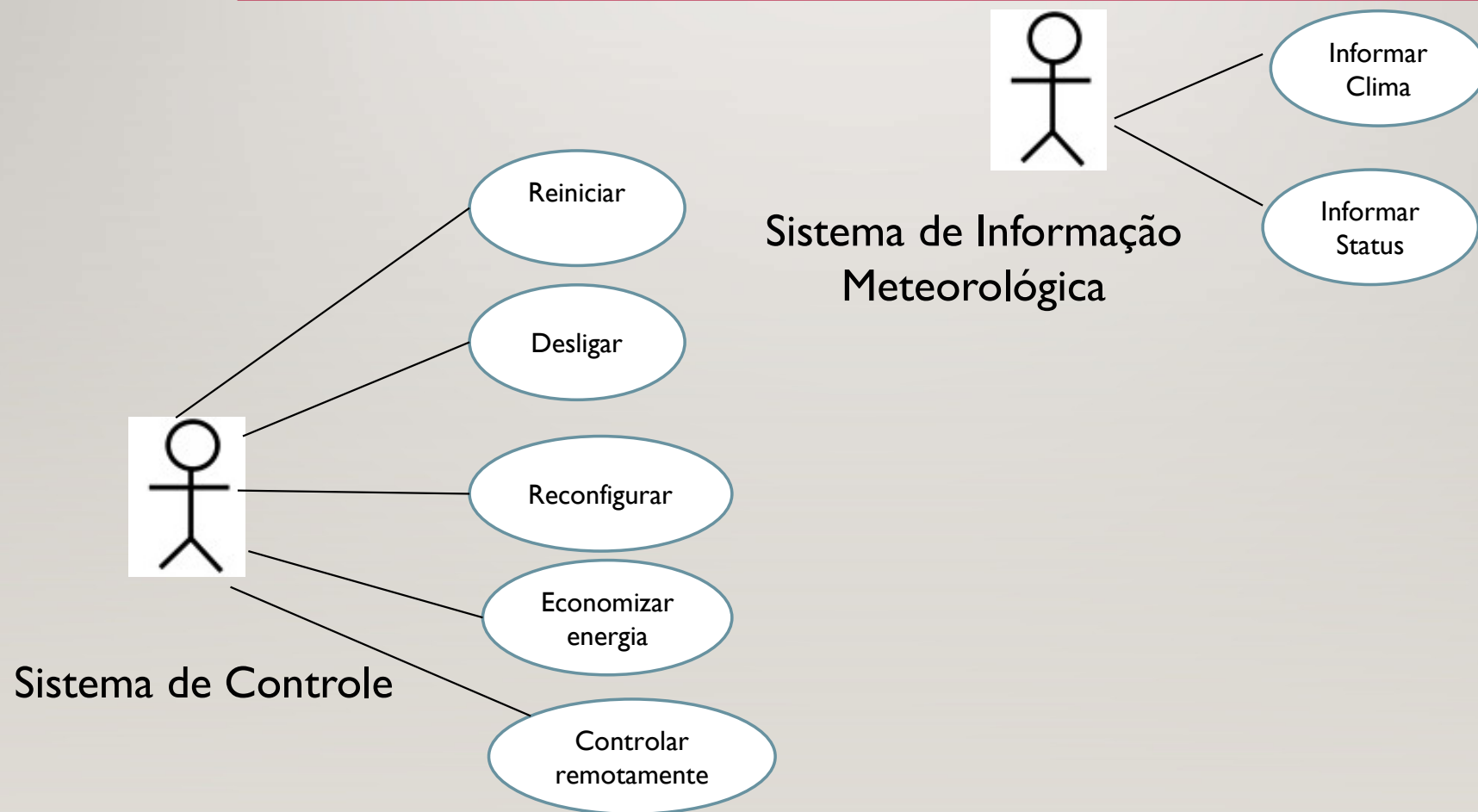
PROJETO ORIENTADO A OBJETOS COM UML

– CASOS DE USO

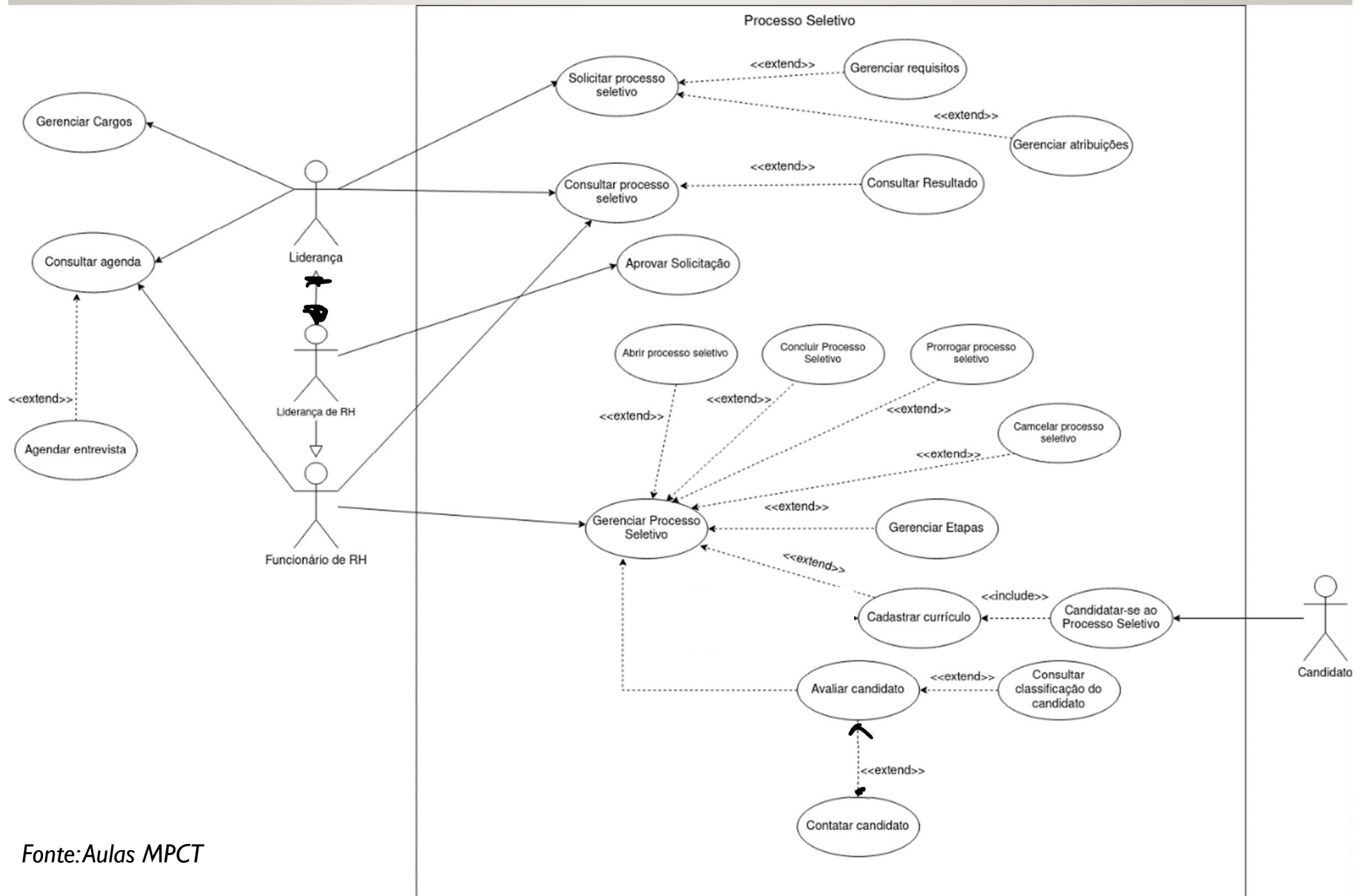


- **Caso de uso** é uma técnica de **especificação** que descreve uma sequência de ações que o sistema deve realizar para produzir uma resposta para um ator. O **caso de uso** detalha o que um sistema deve fazer, descrevendo como uma determinada funcionalidade é utilizada por um ator.

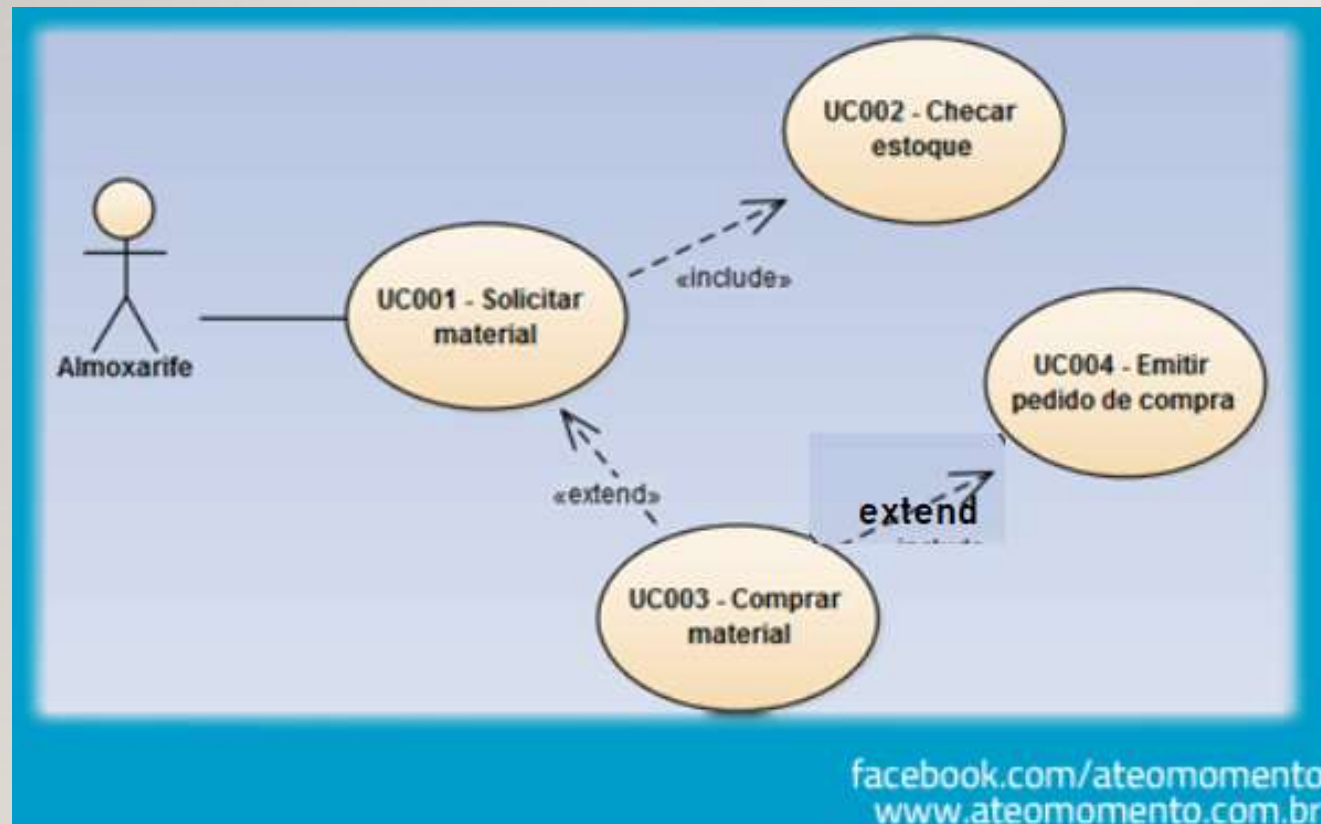
PROJETO ORIENTADO A OBJETOS COM UML – EXEMPLO: CASOS DE USO (ALTO NÍVEL) DO SISTEMA ESTAÇÃO METEOROLÓGICA NO DESERTO



PROJETO ORIENTADO A OBJETOS COM UML – EXEMPLO: CASOS DE USO (ALTO NÍVEL) DO PROCESSO SELETIVO



PROJETO ORIENTADO A OBJETOS COM UML – EXEMPLO: CASOS DE USO (ALTO NÍVEL) DO ALMOXARIFADO



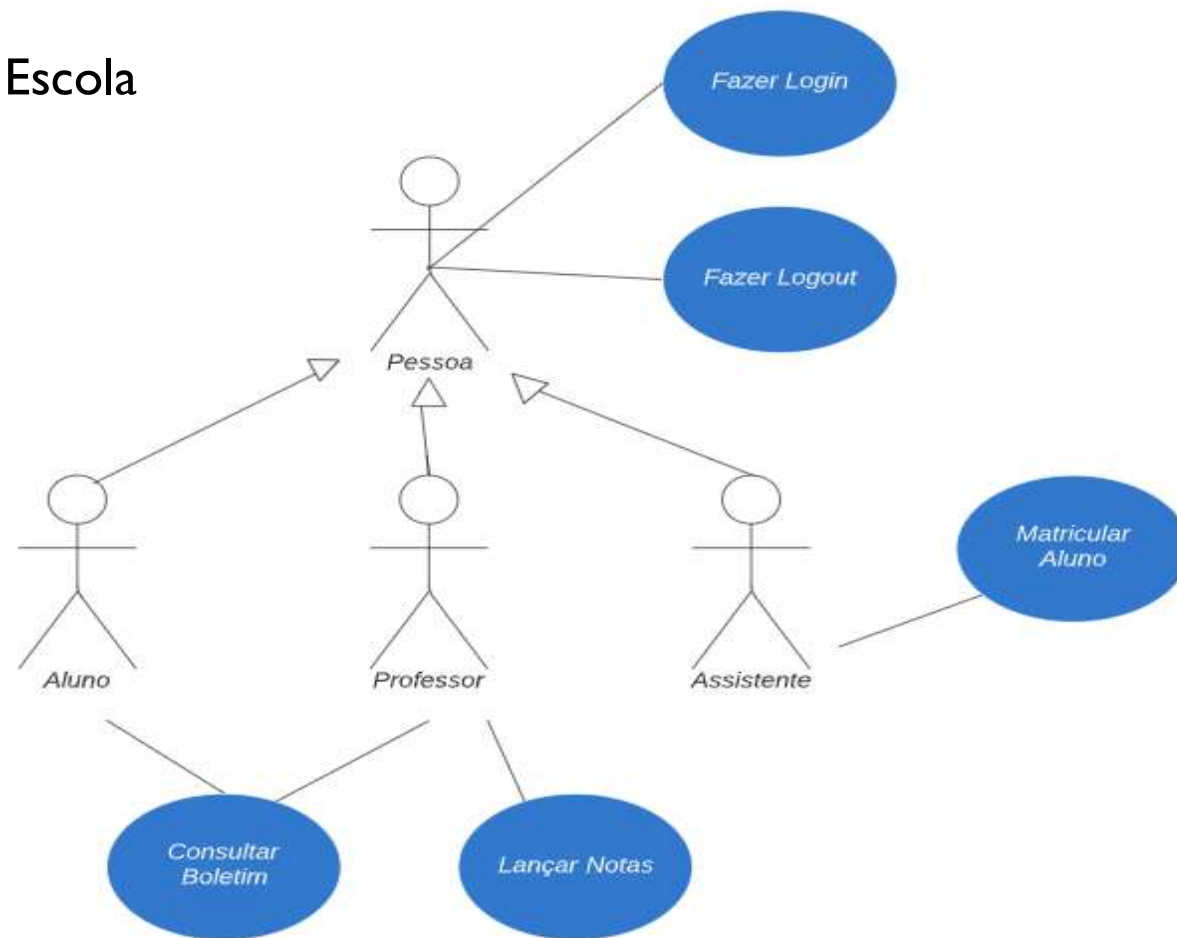
Include – **sempre** que Solicitar material deverá checar estoque.

Extend – quando Solicitar material **poderá** Comprar material

O QUE HÁ DE ERRADO NESSE DESENHO???

PROJETO ORIENTADO A OBJETOS COM UML – EXEMPLO: CASOS DE USO (ALTO NÍVEL) DO SISTEMA ESCOLA

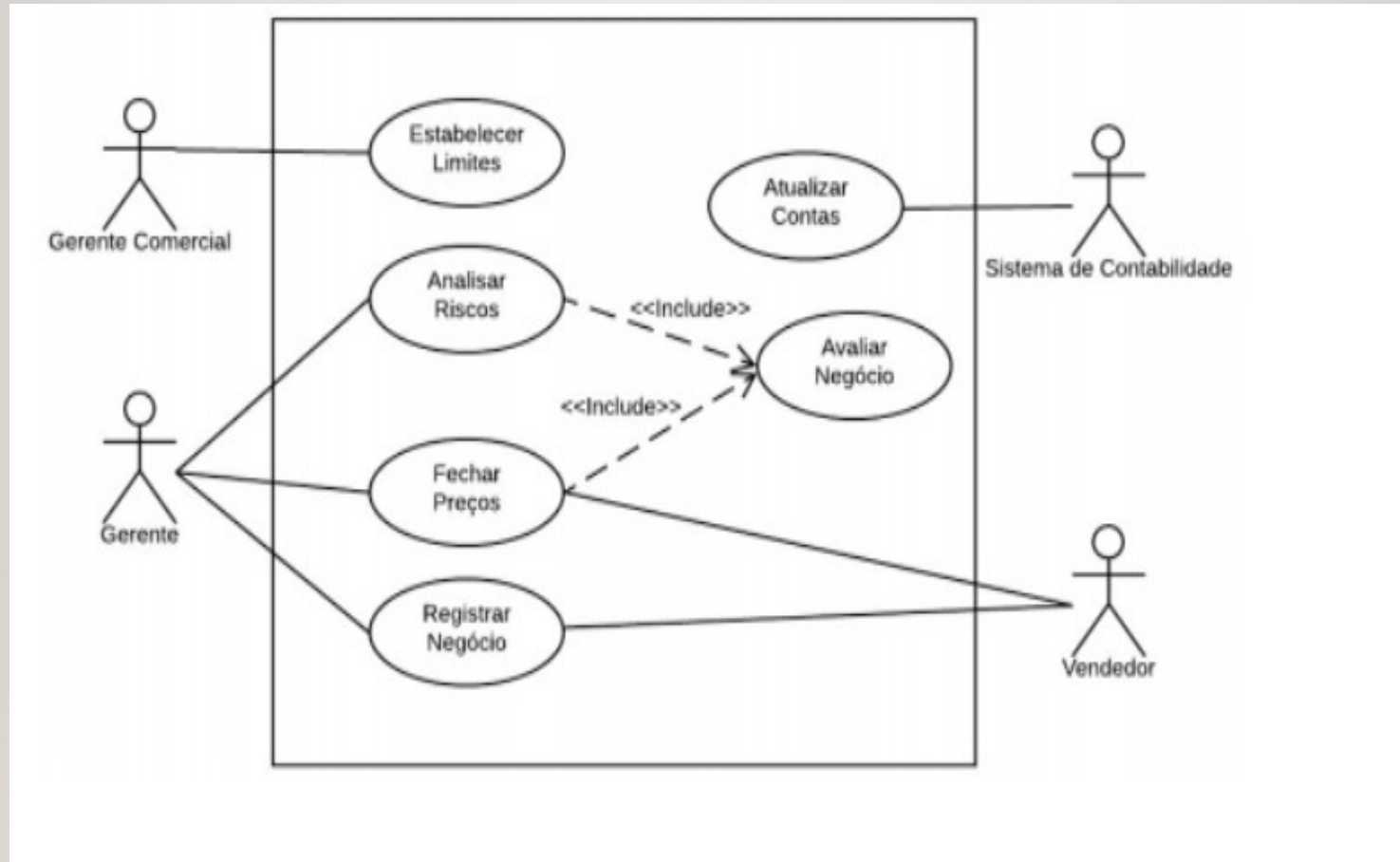
Sistema Escola



Generalização/Especialização - acontece quando os atores possuem características semelhantes

EXERCÍCIOS

1) Considerando o Diagrama de Caso de Uso apresentado e os elementos da notação UML, assinale a alternativa **CORRETA**:



- A) O caso de uso “Fechar Preços” está inserido no fluxo de execução do caso de uso “Avaliar Negócio”.
- B) O ator “Sistema de Contabilidade” está incorreto, pois atores devem representar pessoas que interagem com o sistema.
- C) O ator “Vendedor” executa o fluxo do caso de uso “Avaliar Negócio”.
- D) Setas direcionadas indicam relacionamento sequencial entre casos de uso.

Slide 26

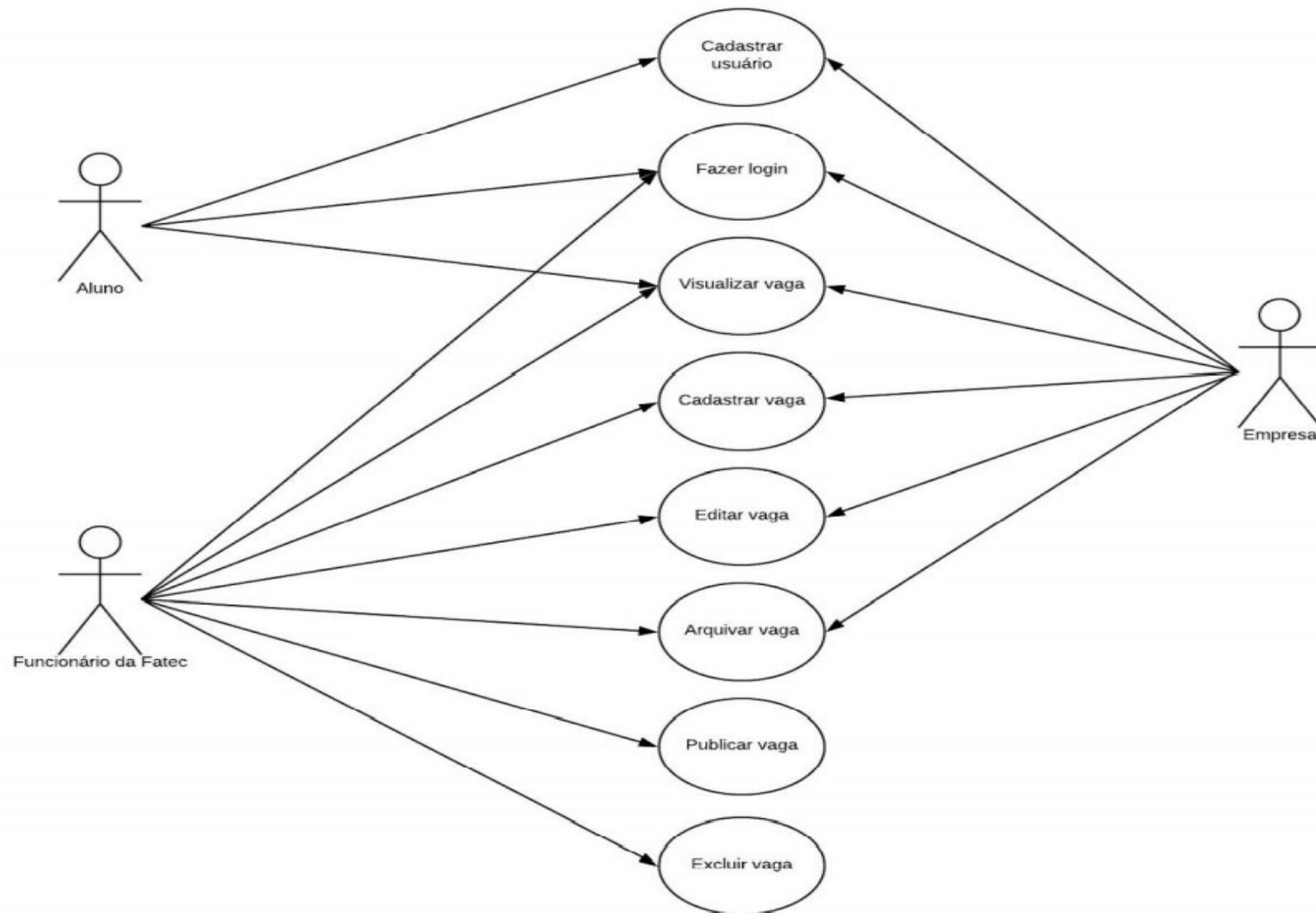
DV0

Resposta: C

DENILCE DE ALMEIDA OLIVEIRA VEL; 2024-03-21T16:20:32.411

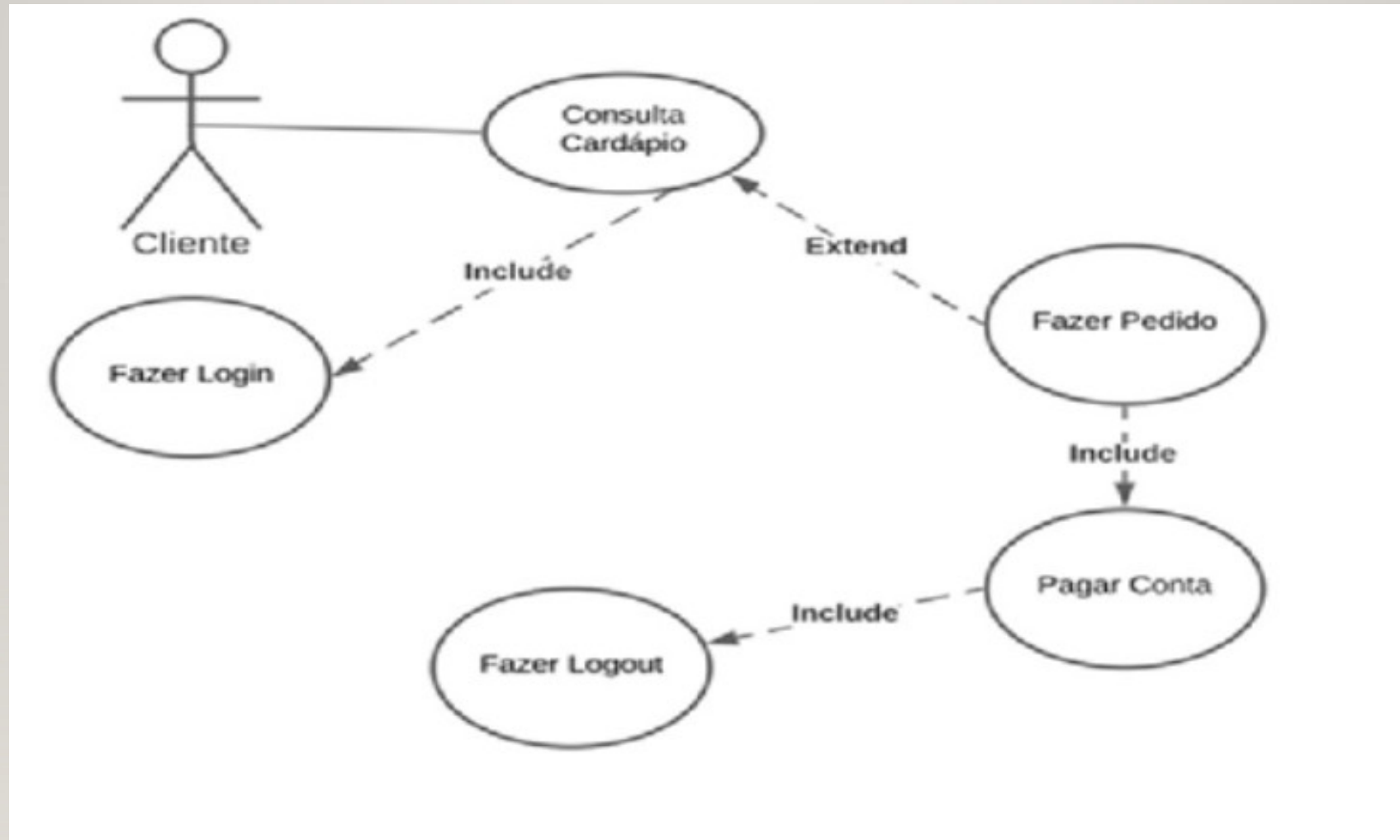
EXERCÍCIOS

2) Segundo Sommerville (2011), os casos de uso são uma maneira de escrever as interações entre os usuários e um sistema usando um modelo gráfico e um texto estruturado. Observe o caso de uso e informe o que pode ser melhorado no mesmo.



EXERCÍCIOS

3) Aponte um problema no diagrama de caso de uso abaixo:



PROJETO ORIENTADO A OBJETOS COM UML – EXEMPLO: DESCRIÇÃO DO CASO DE USO (BAIXO NÍVEL) FAZER LOGIN

Quadro 1. Caso de uso – Fazer Login

Caso de Uso	RF1: Fazer Login	
Ator Principal	Usuário	
Ator secundário	Fornecedor	
Pré-condição	O usuário deve estar previamente cadastrado no sistema	
Pós-condição		
Ações do Ator		Ações do Sistema
1- O usuário acessa o sistema.		
		2-O sistema mostra a tela de login para o usuário.
3-O Usuário coloca o e-mail e a senha e seleciona para entrada no sistema. Caso o usuário tenha esquecido a senha poderá chamar o RF2.		
		4-O sistema faz a validação do e-mail e da senha.
5-O Usuário consegue fazer a entrada no sistema.		
		6-O sistema libera as opções de acordo com o nível do usuário.

Fonte: Autor

PROJETO ORIENTADO A OBJETOS COM UML – EXEMPLO: DESCRIÇÃO DO CASO DE USO (BAIXO NÍVEL) RECUPERAR SENHA

Quadro 2. Caso de uso – Recuperar senha

Caso de Uso	RF 2: Recuperar Senha	
Ator Principal	Usuário	
Ator secundário	Fornecedor	
Pré-condição	O usuário deve estar previamente cadastrado.	
Pós-condição		
Ações do Ator		Ações do Sistema
1- O usuário escolhe a opção recuperar senha.		
		2- O sistema exibe a opção de recuperação de senha.
3- O usuário digita o e-mail		
		4- O sistema pede a confirmação do e-mail para o qual a senha será mandada.
5- O usuário confirma o e-mail.		
		6- O sistema manda a senha cadastrada para o e-mail informado.

Fonte: Autor

PROJETO ORIENTADO A OBJETOS COM UML – EXEMPLO: DESCRIÇÃO DO CASO DE USO (BAIXO NÍVEL) GERENCIAR USUÁRIO(I)

Quadro 11. Caso de uso – Gerenciar Usuário

Caso de Uso	RF11: Gerenciar Usuário	
Ator Principal	Comprador	
Ator secundário	Gerente	
Pré-condição	O usuário deverá estar logado.	
Pós-condição		
Ações do Ator	Ações do Sistema	
Cadastrar		
1- O usuário acessa a opção Gerenciar Usuário e Cadastrar Usuário.		
	2-O Sistema exibe na tela o formulário.	
3-O usuário preenche as informações como nome, e-mail e senha e confirma inclusão.		
	4-O Sistema valida se todos os campos estão preenchidos, retorna mensagem “Cadastro Efetuado com sucesso” e salva registro.	

PROJETO ORIENTADO A OBJETOS COM UML – EXEMPLO: DESCRIÇÃO DO CASO DE USO (BAIXO NÍVEL) GERENCIAR USUÁRIO (2)

Alterar	
1- O usuário acessa a opção Gerenciar Usuário e Alterar Usuário.	
	2-O Sistema exibe na tela os usuários.
3- O usuário escolhe o qual deseja alterar.	
	4-O sistema exibe os dados do usuário selecionado.
5-O usuário altera as informações necessárias e confirma a alteração.	
	6-O Sistema valida as informações. Se for alterada retorna mensagem “Alteração Efetuada com sucesso” e salva o registro.

PROJETO ORIENTADO A OBJETOS COM UML – EXEMPLO: DESCRIÇÃO DO CASO DE USO (BAIXO NÍVEL) GERENCIAR USUÁRIO (2)

Excluir

1- O usuário acessa a opção Gerenciar Usuário e Excluir Usuário.	
	2-O Sistema exibe na tela todos os usuários.
3-O usuário escolhe qual deseja excluir e confirma a exclusão.	
	4-O Sistema exclui o registro e retorna mensagem “Exclusão efetuada com sucesso.”

Fonte: Autor

SUGESTÕES DE FERRAMENTAS GRATUITAS PARA UML

- 1. Edraw Max**
- 2. Lucidchart**
- 3. Draw.io**
- 4. Creately**
- 5. Cacoo**
- 6. Visual Paradigm**
- 7. PlantUML**

EXERCÍCIO I - PROBLEMA



A Faculdade Alpha está criando uma aplicação para utilização de vários tipos de usuários (ATORES).

- Os alunos podem: **realizar a matrícula** pela aplicação, **solicitar histórico**, **solicitar transferência**, **consultar notas e frequência**.
- Quando os alunos **solicitam transferência** a mesma **deverá ser aprovada** pelo funcionário da secretaria.
- A secretaria também **pode cadastrar, alterar, excluir ou consultar um aluno e realizar a matrícula** de um aluno também.
- Os professores também **podem incluir, alterar, excluir ou consultar notas dos alunos e a mesma coisa com relação a frequência**.
- Todos os usuários necessitam **fazer login** no sistema e podem **fazer logout** e **recuperar a senha**, caso tenham esquecido.



EXERCÍCIO – PARTE 2 – O QUE FAZER

1. Listar os Requisitos Funcionais
2. Fazer o Diagrama de Caso de Uso

pasta



**** Subir como: *seuusuario/ES2N/Atividade5***

PRÓXIMO PASSO PARA O PROJETO (I



HOJE

1. Discussão sobre 5ª. Versão Proposta (se for o caso).
2. Discussão sobre 2ª versão dos Requisitos Funcionais.
3. Discussão sobre os formulários das pesquisas.

PRÓXIMO PASSO PARA O PROJETO (2)



- Acertar formulário e disparar pesquisa (21/03 a 01/04). Colocar cópia do form e os resultados dos gráficos no templates de pesquisa versão 3.0 (entregar dia 2/4)

Para próxima semana:

- 3ª versão Requisitos funcionais
- 1ª versão Diagrama de Caso de Uso
- Organizar repositório — deixar somente as últimas versões: proposta, requisitos funcionais, não funcionais e pesquisa.

REFERÊNCIAS

- BOOCH, Grady et al. *The Unified Modeling Language User Guide*. Addison Wesley, 2005.
- MEDEIROS, Ernani. *Desenvolvendo Software com UML 2.0: Definitivo*. Makron Books, 2006.
- PRESSMAN, Roger S. **Engenharia de software : uma abordagem profissional**. 7. ed. Porto Alegre: McGraw-Hill, 2011.
- SOMMERVILLE, Ian. **Engenharia de software**. 10. ed. São Paulo: Pearson, 2019.
- _____. Materiais Fornecidos pelo autor. Disponível em: <https://www.slideshare.net/software-engineering-book/ch4-req-eng> Acesso em: 01 de mar. de 2021.

