

**MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PIAUÍ – UFPI
CAMPUS SENADOR HELVÍDIO NUNES DE BARROS – PICOS
BACHARELADO EM SISTEMAS DE INFORMAÇÃO
DISCIPLINA: ANÁLISE E PROJETO DE SISTEMAS**

EQUIPE:

**LAZARO CLAUBERT SOUSA RODRIGUES OLIVEIRA
KENNEDY CAMILO CARLOS DE SOUZA
MAURICIO BENJAMIN DA ROCHA
PEDRO ANTONIO VITAL DE SOUSA CARVALHO**

Para refletir 1

1. Qual a importância de utilizar requisitos em um projeto de desenvolvimento de Software?

A utilização de requisitos em um projeto de desenvolvimento de software é de extrema importância. Os requisitos são descrições detalhadas das funcionalidades, características e restrições do sistema que será desenvolvido. Eles servem como base para guiar todo o processo de desenvolvimento e garantir que o software atenda às necessidades e expectativas dos usuários finais.

Eles ajudam em diversos aspectos tais como compreensão do problema, comunicação efetiva, definição de escopo, orientação clara para o desenvolvimento, base adequada para testes e validação do que foi feito em relação ao proposto, controle de mudanças e etc.

2. Diferencie Elicitação de Especificação de requisitos.

Elicitação de requisitos refere-se ao processo de identificar, coletar e compreender as necessidades dos stakeholders e transformá-las em requisitos claros e concisos. É o estágio inicial em que são realizadas entrevistas, observações ou outras técnicas de engenharia de requisitos para obter informações relevantes. O objetivo é extrair informações sobre o que o sistema precisa fazer e como deve funcionar para atender aos objetivos e expectativas dos usuários finais. A elicitação de requisitos envolve interações com as partes interessadas para entender suas necessidades, identificar requisitos funcionais e não funcionais, e capturar os principais recursos e restrições do sistema.

Por outro lado, a especificação de requisitos é o processo de documentar os requisitos de forma detalhada, completa e precisa. Nessa etapa, os requisitos coletados durante a elicitação são refinados, organizados e estruturados em

um formato adequado para posterior desenvolvimento do software. A especificação de requisitos envolve a criação de documentos como casos de uso, especificações funcionais, diagramas de sequência, diagramas de classes, entre outros artefatos. Esses documentos descrevem, de maneira clara e compreensível, as funcionalidades, os comportamentos, as interfaces e outras características do sistema. A especificação de requisitos visa fornecer uma referência completa para a equipe de desenvolvimento entender o que deve ser implementado e como o sistema deve se comportar.

Em resumo, a elicitación de requisitos é o processo de coleta e compreensão das necessidades dos stakeholders, enquanto a especificação de requisitos é a documentação detalhada e estruturada desses requisitos. A elicitación está mais focada na obtenção de informações e entendimento das necessidades, enquanto a especificação está mais relacionada à formalização e organização dos requisitos para orientar o desenvolvimento do software. Ambos os processos são cruciais para o sucesso do projeto, pois garantem que as necessidades dos usuários sejam compreendidas e que as soluções sejam implementadas de acordo com essas necessidades.

3. Quais as principais técnicas de Elicitación de Requisitos? Descreva-as.

Existem diversas técnicas para a elicitación de requisitos, mas iremos abordar apenas cinco delas. sendo elas:

- Entrevista
- Brainstorming
- Workshops
- Casos de uso
- Questionários

Entrevista: É conhecida por ser a mais comum das técnicas de elicitación de requisitos. As entrevistas geralmente envolvem discussões individuais de um representante da equipe com o stakeholder do novo sistema.

Brainstorming: É uma discussão informal onde cada participante pode expressar suas ideias livremente para que um novo sistema seja desenvolvido.

Workshop: É uma espécie de seminário, grupo de discussão ou colóquio em que um palestrante apresenta um conteúdo específico e incentiva a reflexão no grupo.

Caso de uso: Representa uma possível utilização do sistema por um ator utilizando algum de seus serviços. Ele narra a interação entre o sistema e os atores envolvidos, para atingir um ou mais objetivos.

Questionários: São usados principalmente como uma ferramenta simples, com perguntas abertas ou fechadas durante a fase inicial da elicitación de requisitos para coletar o máximo de requisitos de diferentes pessoas que podem estar em lugares distintos.

4. Cite 5 Requisitos Funcionais para:

a) Sistema da padaria de pequeno porte;

- 1. Controle de caixa**
- 2. Estoque de matéria prima**
- 3. Relatórios de venda periódicos**
- 4. Sistema de cadastro de usuário**
- 5. Cadastrar os produtos e seus preços;**

b) Sistema de alocação docente.

- 1. Cadastro dos Docentes**
- 2. Cadastro das disciplinas que serão ofertadas**
- 3. Registro das turmas e horários disponíveis**
- 4. Consulta nas informações**
- 5. Relacionar as disciplinas com seus respectivos docentes**

5. Cite 5 Requisitos Não Funcionais para:

a) Sistema da padaria de pequeno porte;

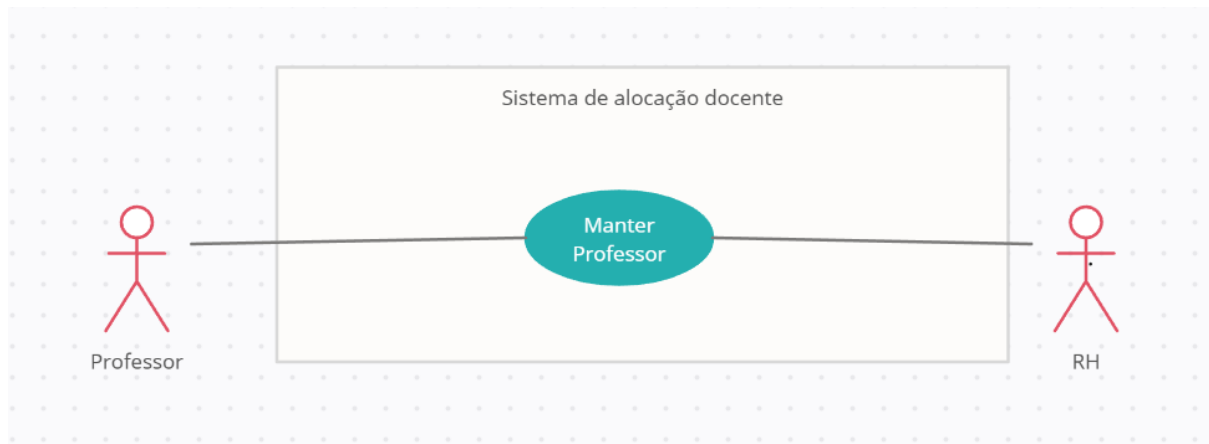
- 1. Interface simples e intuitiva (frases e imagens intuitivos)**
- 2. Sistema de de segurança baseado em login e senha**
- 3. Após pagar por um produto, o estoque daquele produto deve ser atualizado**
- 4. O relatório final deve ter todas as informações detalhadas de cada compra realizada no dia.**
- 5. O sistema deve ser capaz de gerenciar o caixa da padaria, registrando todas as entradas e saídas de dinheiro, incluindo as vendas.**

b) Sistema de alocação docente.

- 1. O sistema deverá ter uma interface com máximo de simplicidade possível, devido a possível entrada e saída de professores que irão interagir com o sistema.**
- 2. O sistema deverá cadastrar os docentes para as disciplinas de acordo com cada disponibilidade de horário e suas especialidades.**
- 3. O sistema deve através do login e senha identificar se é realmente um docente para que possa direcioná-los a "telas" com todas as funcionalidades referente ao seu cargo.**
- 4. O sistema deverá checar tanto os horários quanto as turmas disponíveis para que não ocorra choque de horários.**

5. O menu de cadastros deve ser simplificado, e de fácil utilização.

6. Escreva um caso de uso chamado Manter Professor para o sistema de alocação docente.



Para refletir 2

1. Faça uma pesquisa mais aprofundada sobre o padrão arquitetural MVC, observe seus detalhes e formas de adaptação.

O padrão arquitetural MVC divide uma aplicação em três componentes principais: o Modelo (Model), a Visualização (View) e o Controlador (Controller), os quais são representados pela sua sigla. Cada componente tem uma responsabilidade específica e trabalha em conjunto para criar uma estrutura organizada e modular. Isso resulta em um código mais modular, facilitando a manutenção e a evolução do software. Além disso, o MVC promove a reutilização de código e a escalabilidade do sistema, permitindo que cada componente seja modificado independentemente dos outros.

Componentes e funções:

- **Model:** O modelo representa a camada de dados da aplicação. Ele lida com a lógica de negócios, o acesso e a manipulação dos dados. O modelo é responsável por armazenar, recuperar e atualizar os dados, bem como notificar as visualizações e controladores sobre quaisquer alterações relevantes nos dados.
- **View:** A visualização é a camada de apresentação da aplicação. Ela exibe as informações ao usuário e fornece uma interface com a qual o usuário pode interagir. A visualização pode ser uma interface gráfica, uma página da web, uma saída de texto ou qualquer outro meio de exibição das informações.
- **Controller:** O controlador atua como intermediário entre o modelo e a visualização. Ele recebe as entradas do usuário, manipula essas entradas e atualiza o modelo de acordo. Ele também notifica as visualizações sobre as alterações ocorridas no modelo, permitindo que

elas sejam atualizadas adequadamente. O controlador é responsável por interpretar as ações do usuário e decidir como o modelo deve ser atualizado.

Para a adaptação desse padrão, são recomendadas algumas diretrizes, tais como: Identificar as responsabilidades do sistema e suas tarefas, separar a lógica de negócios e claro, a realização de testes.

2. Cite três diferenças entre a fase de análise e projeto.

1. Objetivos e foco: A fase de análise concentra-se em compreender as necessidades e requisitos dos usuários finais, bem como em identificar os problemas que o software deve resolver. É nessa fase que ocorre a elicitación de requisitos, a análise de processos e a modelagem conceitual do sistema. Por outro lado, a fase de projeto tem como objetivo transformar os requisitos identificados em uma solução técnica. Nessa etapa, os projetistas definem a arquitetura do sistema, projetam os componentes, estabelecem as interfaces e tomam decisões relacionadas à implementação, como a escolha de tecnologias e estruturas de dados. A fase de projeto está mais voltada para aspectos técnicos e de implementação do software.

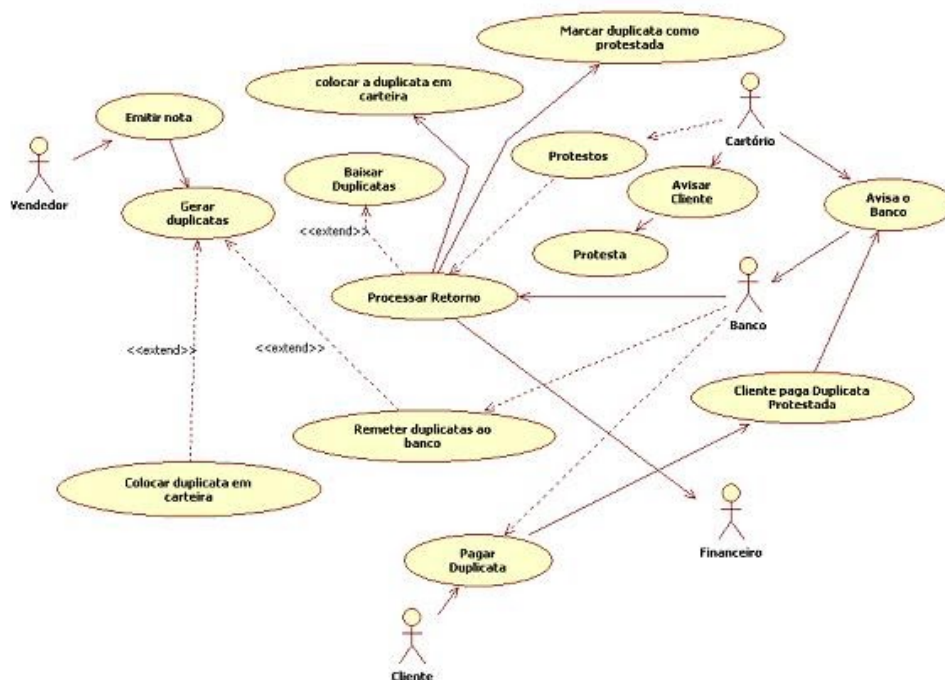
2. Artefatos produzidos: Na fase de análise, os principais artefatos produzidos incluem o documento de requisitos, modelos de processos, diagramas de caso de uso, diagramas de classe preliminares, entre outros. Esses artefatos visam capturar e representar os requisitos e o entendimento do sistema. Já na fase de projeto, os principais artefatos produzidos incluem a arquitetura do sistema, diagramas de sequência detalhados, diagramas de classe refinados, diagramas de bancos de dados, entre outros. Esses artefatos têm como objetivo traduzir os requisitos em soluções técnicas e detalhar a implementação do sistema.

3. Nível de detalhe e abstração: A fase de análise geralmente possui um nível de abstração maior. É uma fase mais exploratória, em que os analistas buscam compreender o contexto do sistema, identificar os problemas e definir as funcionalidades-chave. Os modelos e documentos criados nessa fase fornecem uma visão geral do sistema e são mais voltados para o entendimento do problema e dos requisitos. Por outro lado, a fase de projeto é mais detalhada e concreta. Os projetistas lidam com decisões técnicas específicas, definem a estrutura do sistema, projetam a interação entre os componentes e estabelecem os detalhes da implementação. Os artefatos produzidos nessa fase são mais detalhados e têm um foco maior na implementação efetiva do software.

3. Qual a importância da fase de análise para um projeto de desenvolvimento de software?

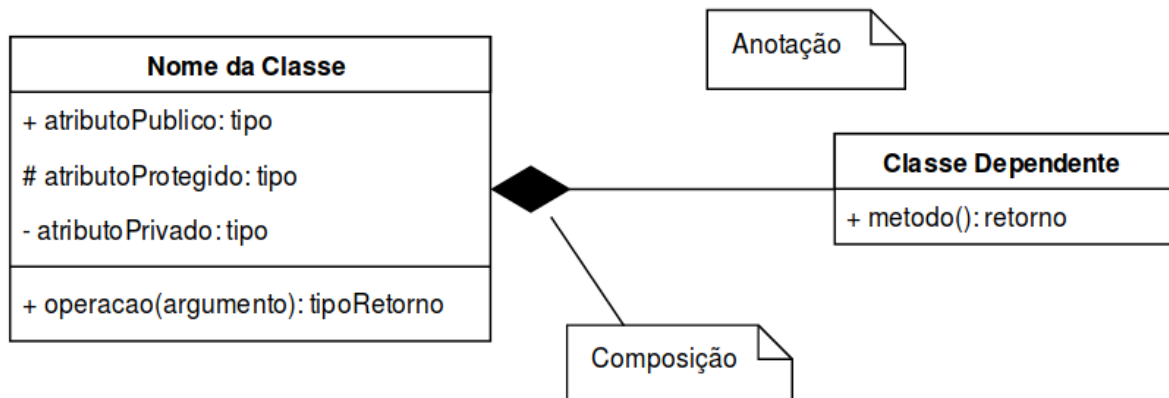
4. Encontre modelos criados para os seguintes diagramas: casos de uso, classes, interação, estados e atividades. Observe os elementos de cada um, as diferenças e semelhanças entre eles.

- **Elementos principais:** são atores (representando usuários ou sistemas externos) e casos de uso (representando funcionalidades do sistema).
- **Diferenças:** O diagrama de casos de uso tem foco nas interações entre os atores e os casos de uso, mostrando como os usuários interagem com o sistema. Ele não descreve a estrutura interna do sistema.



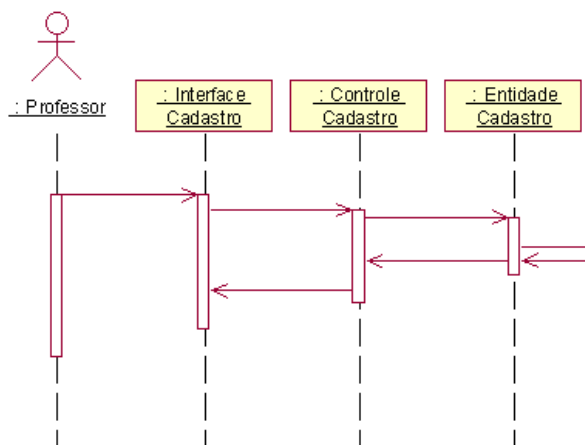
2. Diagrama de Classes:

- Elementos principais: são classes, atributos, métodos, associações, herança, agregação, entre outros.
- Diferenças: O diagrama de classes se concentra na estrutura interna do sistema, descrevendo as classes, seus atributos e métodos, bem como os relacionamentos entre elas. É usado para modelar a estrutura estática do sistema.



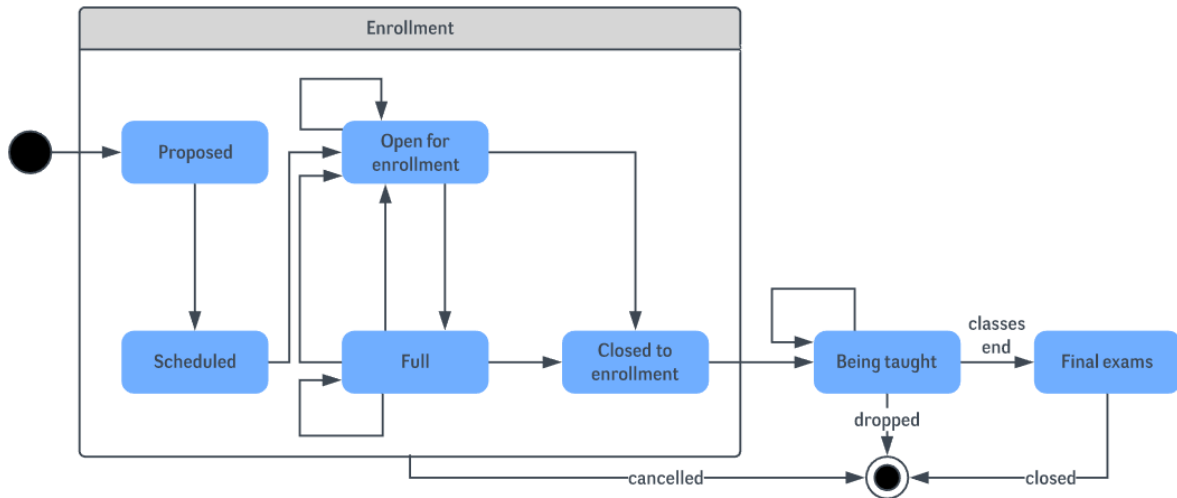
3. Diagrama de Interação (como Diagrama de Sequência e Diagrama de Comunicação):

- Elementos principais: são objetos, mensagens, linhas de vida e a ordem de interação entre os objetos.
- Diferenças: Os diagramas de interação, como o Diagrama de Sequência e o Diagrama de Comunicação, mostram como os objetos interagem entre si em um determinado cenário. Eles destacam a sequência temporal das mensagens trocadas entre os objetos.



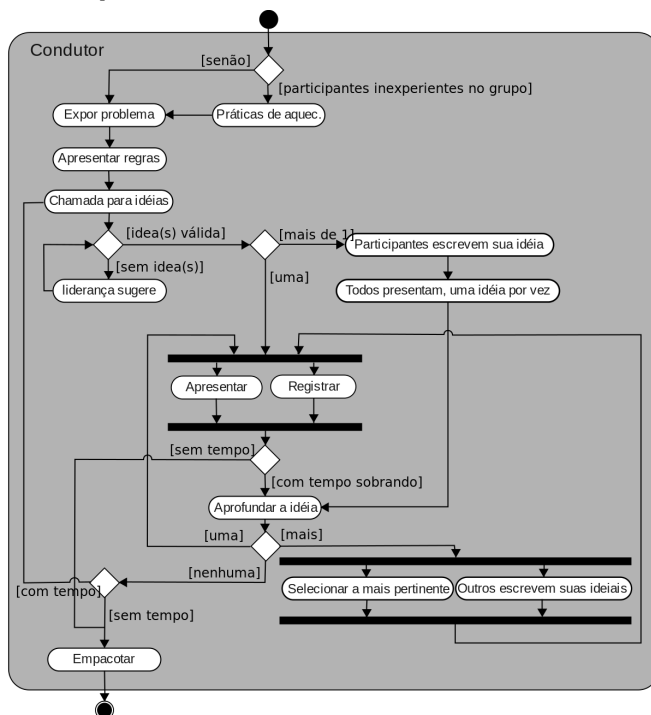
4. Diagrama de Estados:

- Elementos principais: são estados, transições, eventos, ações e condições.
- Diferenças: O diagrama de estados é usado para modelar o comportamento do sistema, descrevendo como ele responde a eventos e transita entre estados diferentes.



5. Diagrama de Atividades:

- Elementos principais: são atividades, decisões, bifurcações, junções, fluxos de controle, objetos, entre outros.
- Diferenças: o diagrama de atividades tem como função modelar o fluxo de trabalho ou processo do sistema, descrevendo as atividades e como elas se relacionam. É usado para representar a lógica sequencial de um processo.



As semelhanças entre estes diagramas está relacionada à representação visual para a compreensão do comportamento do sistema. Em resumo, cada um desses diagramas tem seu foco e finalidade específicos, e eles podem ser usados em conjunto para fornecer uma visão completa do sistema.