

Requisitos de *Software*

Prof.^a Ma. Jessica Oliveira

Aula 04 – 31/03/2025

Modelagem de Requisitos: diagramas e representações.

1. Introdução à Modelagem de Requisitos.

Definição.

- A modelagem de requisitos é uma atividade fundamental no processo de desenvolvimento de *software* que **visa representar, de forma estruturada e clara, as necessidades e expectativas dos *stakeholders* em relação ao sistema a ser desenvolvido.**
- O objetivo da modelagem de requisitos é garantir que todos os envolvidos compreendam o que o *software* deve realizar, evitando ambiguidades e mal-entendidos ao longo do projeto.

Classificação dos requisitos.

- **Requisitos Funcionais:** descrevem as funcionalidades e serviços que o sistema deve fornecer. Exemplos incluem autenticação de usuários, cadastro de produtos e geração de relatórios.
- **Requisitos Não Funcionais:** definem características de qualidade do sistema, como desempenho, segurança, usabilidade e manutenibilidade.

A importância da Modelagem de Requisitos no Desenvolvimento de *Software*.

Clareza e Precisão.

- Facilita a comunicação entre desenvolvedores, analistas, clientes e outros *stakeholders*.
- Evita interpretações divergentes sobre o que o sistema deve realizar.

Redução de ambiguidades.

- A modelagem formal permite identificar e corrigir inconsistências logo no início do projeto, evitando retrabalho e custos elevados em fases posteriores.

Documentação padronizada.

- A utilização de padrões e linguagens formais, como a UML (*Unified Modeling Language*), garante que a documentação seja compreendida por diferentes profissionais.

Facilita a validação e verificação.

- Os modelos ajudam a verificar se os requisitos estão completos e consistentes.
- Permite a validação junto ao cliente, garantindo que as expectativas estão alinhadas com o que será desenvolvido.

Apoio à implementação.

- Os diagramas gerados na modelagem servem como base para o desenvolvimento e documentação do sistema.



2. Principais tipos de Modelagem de Requisitos.

Diagrama de Casos de Uso.

Diagrama UML.

Conceito e objetivo.

- O diagrama de casos de uso é uma representação gráfica que descreve as interações entre os usuários (atores) e o sistema.
- Seu principal objetivo é identificar e documentar os principais serviços que o sistema deve oferecer, agrupando-os de maneira clara e lógica.



Componentes principais.

- Um diagrama de casos de uso é composto pelos seguintes elementos:
 - **Ator:** representa um usuário ou sistema externo que interage com o software. É simbolizado por um **boneco** estilizado ou um **ícone** que indique o tipo de ator.
 - **Caso de Uso:** representa uma funcionalidade ou serviço que o sistema fornece. É ilustrado por uma **elipse** contendo o nome da funcionalidade.

Componentes principais.

- **Relações:** representam a forma como atores e casos de uso interagem. As principais relações são:
 - Associação: conexão direta entre um ator e um caso de uso.
 - Inclusão (<<include>>): representa um comportamento comum a múltiplos casos de uso.
 - Extensão (<<extend>>): representa uma funcionalidade opcional que estende um caso de uso básico.
 - Generalização: representa uma herança entre atores ou casos de uso.

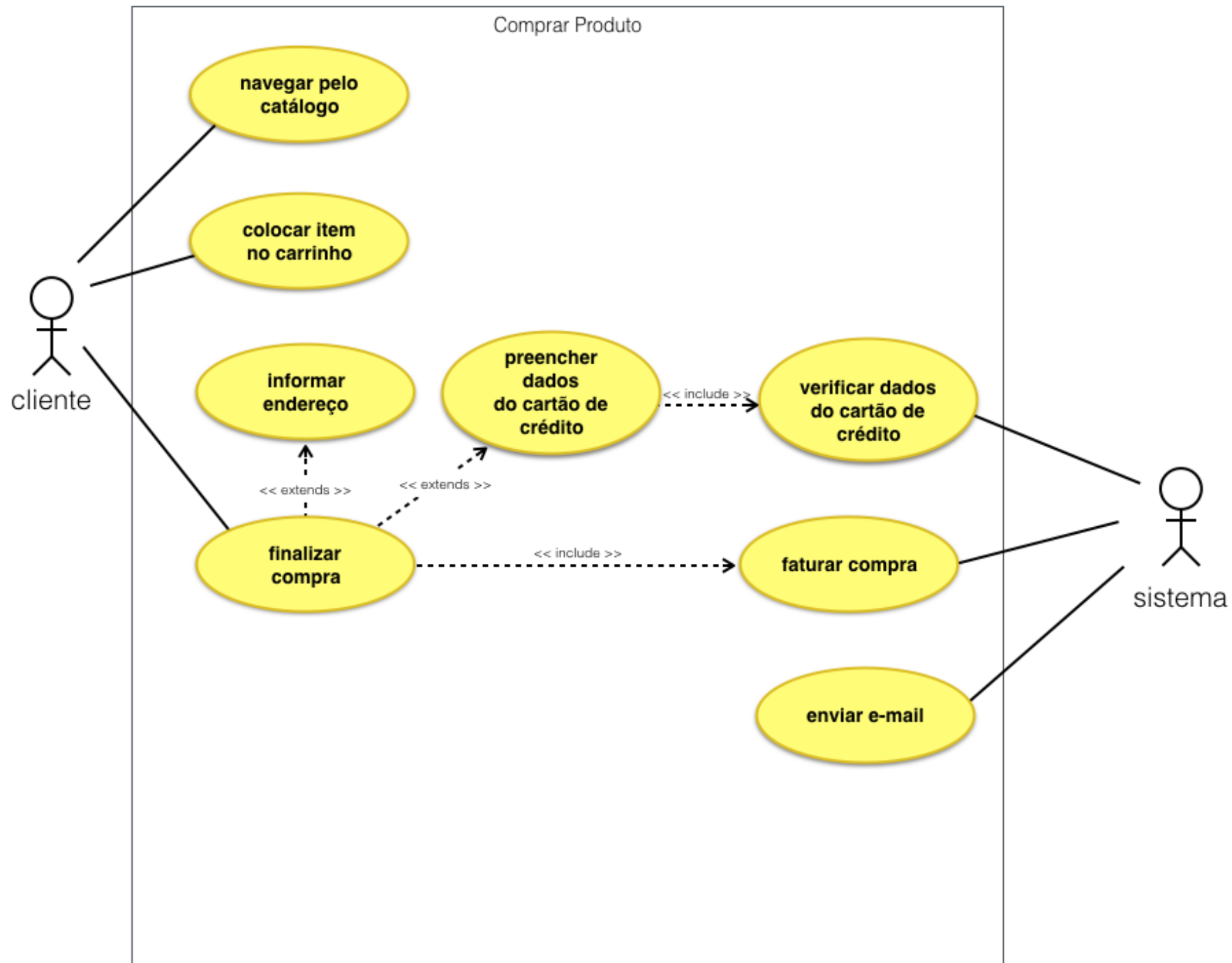


Diagrama de Atividade.

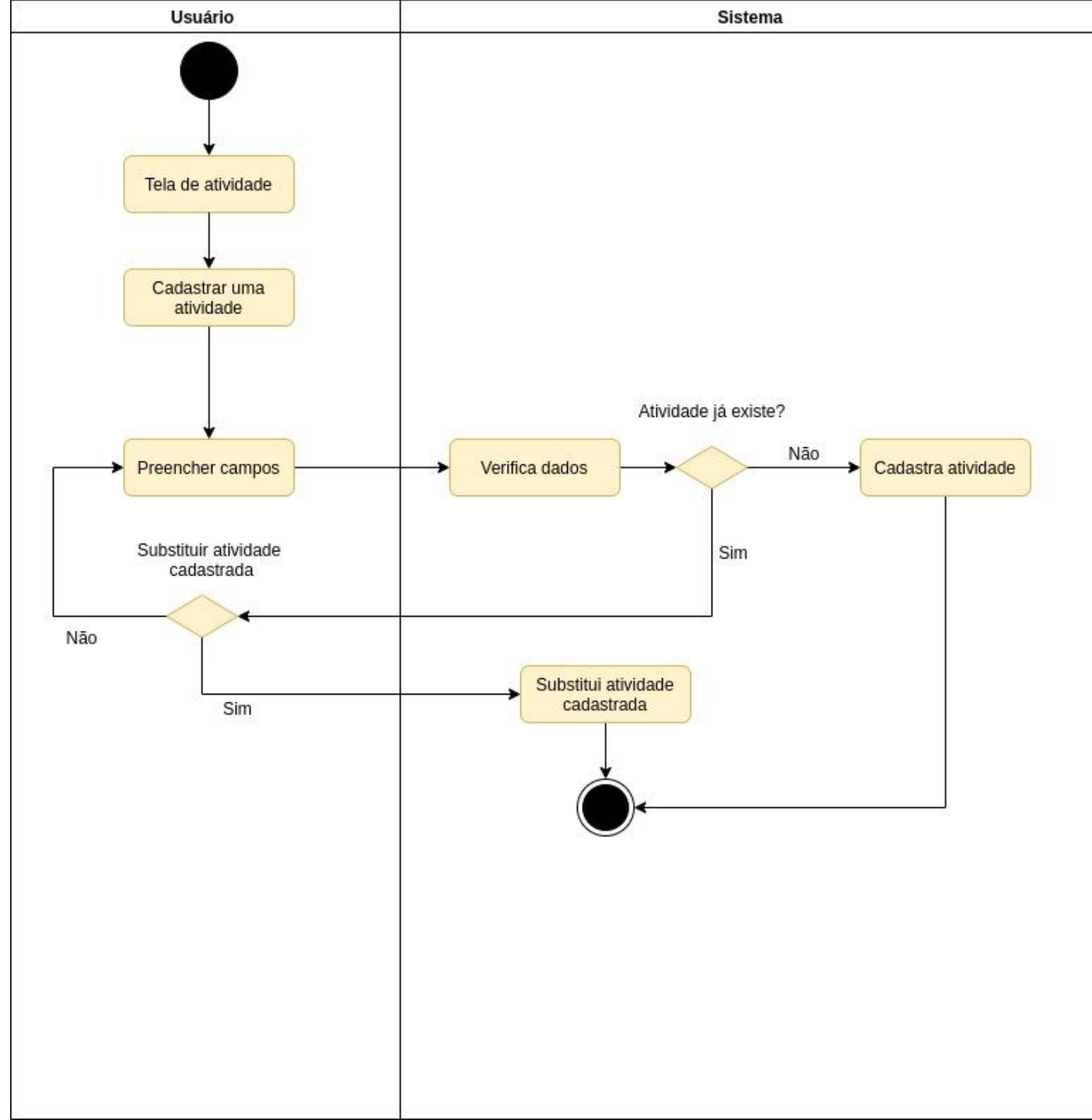
Diagrama UML.

Conceito e objetivo.

- O diagrama de atividade é utilizado para **representar fluxos de trabalho e processos dentro do sistema.**
- Ele ilustra a **sequência de atividades realizadas para atingir um objetivo específico**, detalhando tanto os passos principais quanto os caminhos alternativos ou condicionais.

Componentes principais.

- **Ação:** passo específico realizado dentro de um fluxo.
- **Decisão:** ponto no fluxo onde há uma bifurcação, geralmente com condições que determinam o caminho a seguir.
- **Fluxo de Controle:** linha que conecta as atividades, indicando a sequência de execução.
- **Atividade Inicial e Final:** marcam o começo e o término do fluxo de atividades.
- **Atividades Paralelas:** representam ações que podem ser executadas simultaneamente.
- **Objetos de Dados:** indicam os dados manipulados ou produzidos durante o fluxo.



Abordagens ágeis.



User Stories.

- São utilizadas em metodologias ágeis, como Scrum e XP (*Extreme Programming*), para capturar funcionalidades de forma simples e direta.
- A estrutura padrão de uma *user story* é: "Como [ator], eu quero [ação], para que [benefício]."
- Boas práticas na criação de *User Stories*:
 - Clareza: a história deve ser objetiva e compreensível.
 - Valor para o usuário: a ênfase deve estar nos benefícios que a funcionalidade trará.
 - Testabilidade: deve ser possível verificar se a história foi atendida.
- Exemplo de *User Story*: "Como usuário, eu quero redefinir minha senha, para que eu possa acessar o sistema caso a esqueça."

Storytelling.

- É uma técnica que utiliza narrativas para descrever situações de uso do sistema, permitindo um entendimento mais intuitivo e empático sobre as funcionalidades.
- A história pode incluir cenários, personagens e contextos específicos.
- Benefícios:
 - Humaniza os requisitos, conectando funcionalidades a histórias reais.
 - Facilita o entendimento de fluxos complexos.
 - Permite a visualização do impacto de funcionalidades na vida dos usuários.

Storytelling: exemplo.

- **Cenário:** um estudante acessa o portal acadêmico para verificar suas notas.
- **Narrativa:** Maria, uma estudante universitária, acessa o portal acadêmico para verificar as notas do último semestre. Ela navega até a seção de boletim, escolhe o semestre atual e visualiza a lista de disciplinas com suas respectivas médias. Satisfeita com o resultado, Maria decide compartilhar as notas com seus pais.

Vamos para a prática?



Análise do Cenário

- Cada um deve analisar o cenário escolhido, identificando os principais atores e funcionalidades.
- Definir quais requisitos são essenciais para o funcionamento do sistema.

Modelagem de Casos de Uso

- Utilizando a ferramenta de modelagem, criar um Diagrama de Casos de Uso que inclua:
 - Atores principais e secundários.
 - Casos de uso principais.
 - Relacionamentos entre casos de uso (inclusão e extensão).
- Nomear adequadamente cada elemento, garantindo clareza e precisão.

Modelagem de Atividades

- Criar um Diagrama de Atividade que represente um fluxo relevante, como o processo de cadastro de usuários ou realização de pedidos.
- Incluir:
 - Atividades principais.
 - Decisões e bifurcações.
 - Ações paralelas (quando aplicável).
- Garantir que o fluxo esteja correto e consistente.

Elaboração da *User Story*

- Criar uma user story para uma funcionalidade essencial do sistema.
- Utilizar o formato padrão:

"Como [ator], eu quero [ação], para que [benefício]."
- Revisar a história para garantir clareza e objetividade.

Dúvidas?

jessica.oliveira@p.ucb.br