

# UML

## Unified Modeling Language

# UML – Unified Modeling Language

## O que vamos aprender

- Conceitos a UML.
- Entender e elaborar o diagrama de Casos de Uso.
- Entender e elaborar o diagrama de Classes.
- Entender e elaborar o diagrama de Sequência.
- Entender e elaborar o diagrama de Atividades.
- Entender e elaborar o diagrama de Estados.
- Entender e elaborar o diagrama de Componente.

# UML – Unified Modeling Language

## Características da UML

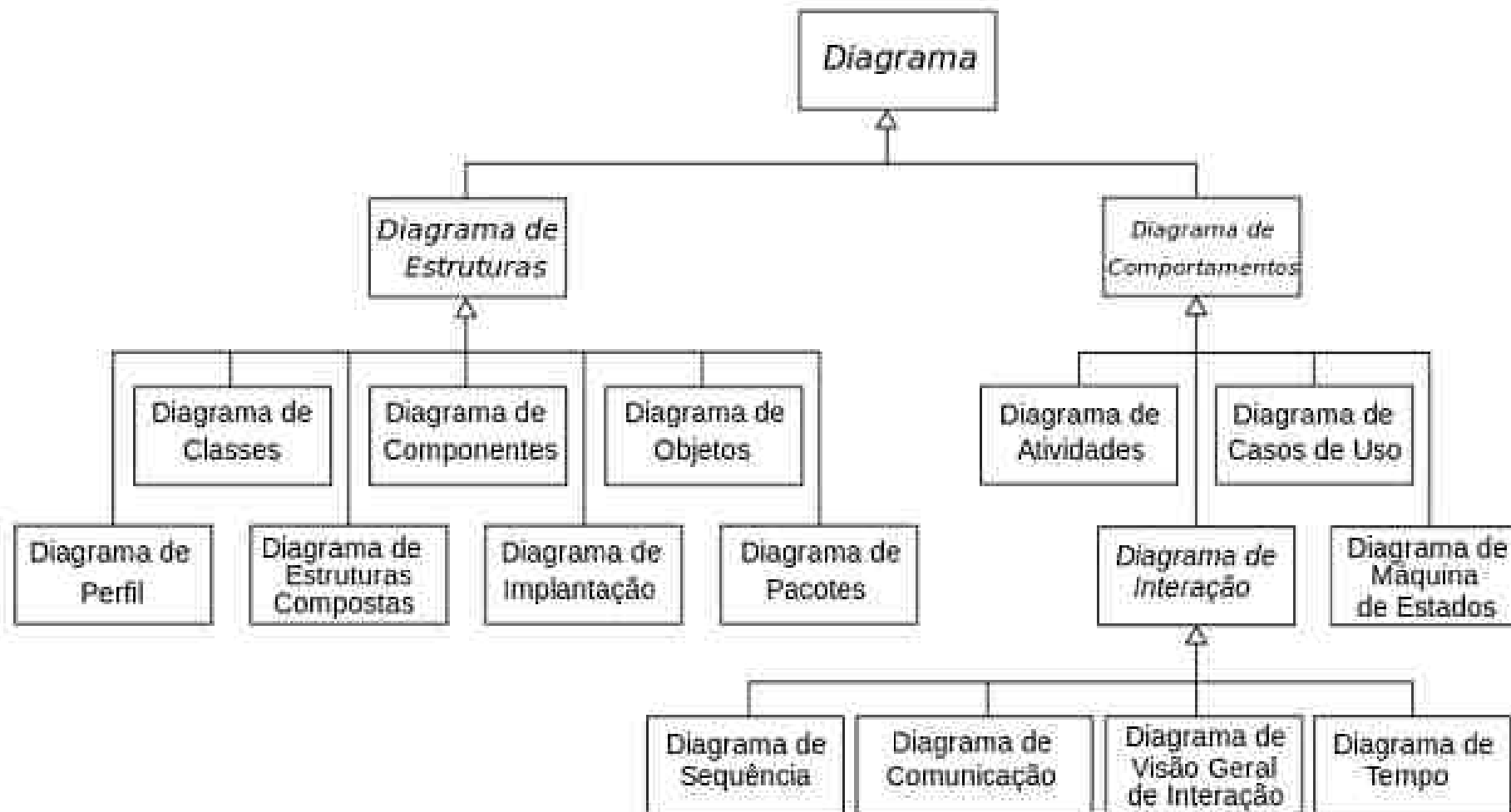
- Linguagem de notação para uso em projetos de sistemas.
- Expressa através de diagramas.
- Cada diagrama é composto por elementos que possuem relação entre si.
- Se dividem em dois grandes grupos:
  - Diagramas estruturais;
  - Diagramas comportamentais.

# UML – Unified Modeling Language

## Características da UML

- Diagramas estruturais - utilizados para especificar detalhes da estrutura do sistema (parte estática)
  - Classes;
  - Componentes;
  - Objetos.
- Diagramas comportamentais - utilizados para especificar detalhes do comportamento do sistema (parte dinâmica),
  - Funcionalidades;
  - Regra de negócio;
  - Fluxo de Comportamentos;
  - Comunicação entre componentes;

# UML – Unified Modeling Language



# UML – Unified Modeling Language

## Para que serve UML

- Passar uma mensagem de maneira padronizada, independente do idioma utilizado.
- É o famoso: “entendeu ou quer que desenha?”
- No passado era utilizado para documentar software existente, ou projeto preditivo de sistema (estimativa). Isso quase nunca é viável.
- No presente serve para uma boa comunicação em equipes, onde fazendo uso de uma linguagem que todos entendem, para deixar claro o que deve ser feito.

# UML – Unified Modeling Language

## Quando se deve usar UML

- Quando é necessário especificar o desejo do cliente que será materializado no software.
- Quando se precisa ter uma visão única e padronizada, seja no escopo funcional (requisitos, histórias de usuário ou modelos de processo) ou não funcional (foco na arquitetura/estrutura do sistema e integrações)
- Comunicar para o mundo externo protocolos (contratos) de interfaces do sistema que devem ser consumidas por terceiros ou ilustrar topologias arquiteturas físicas/lógicas.

# UML – Diagrama de Casos de Uso

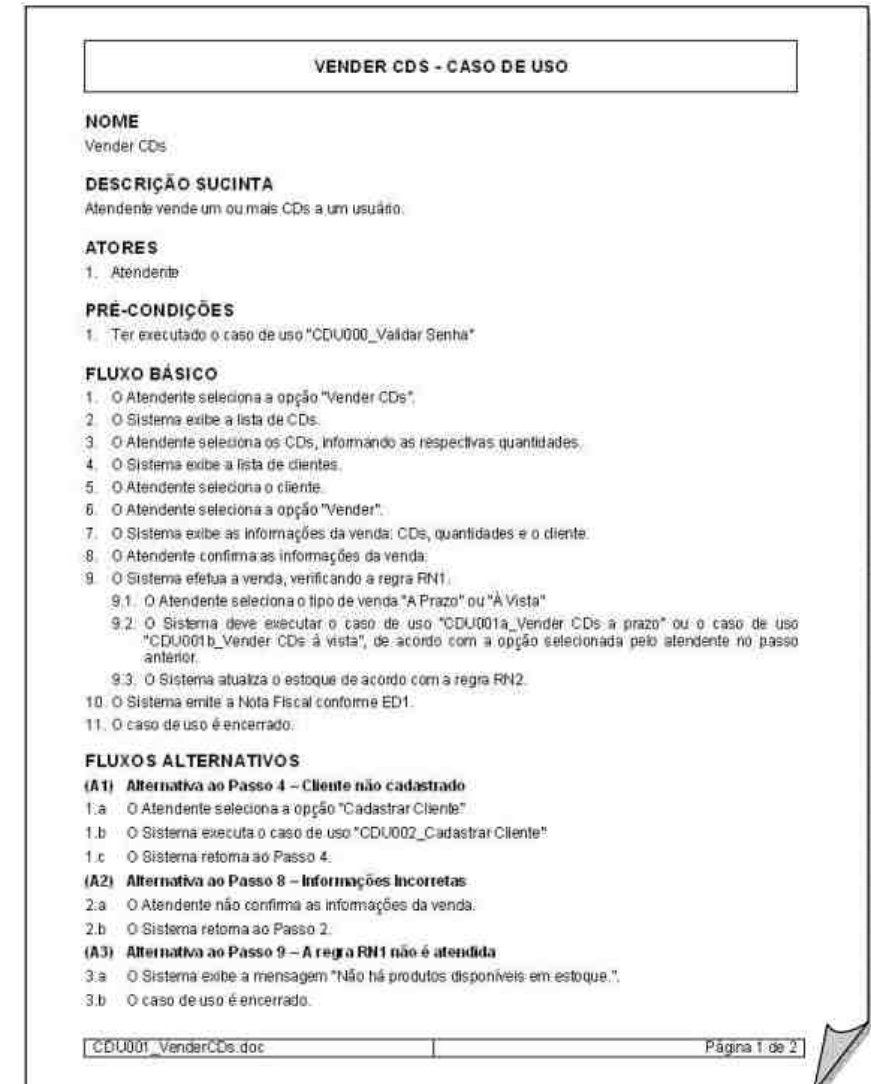
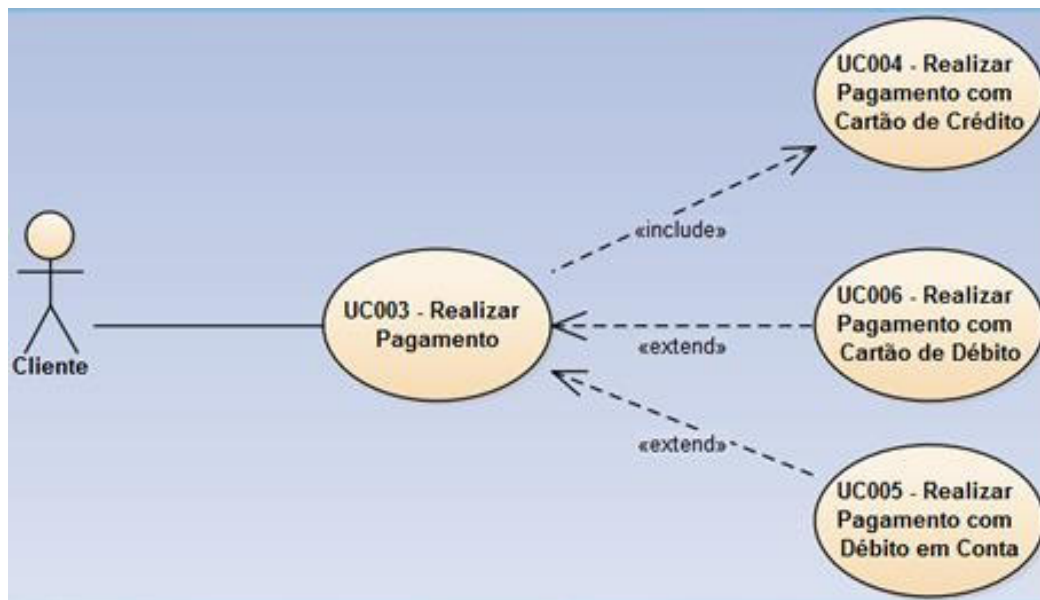
## Diagrama de Casos de Uso

- Documenta o que o sistema faz do ponto de vista do usuário.
- Descreve as principais funcionalidades do sistema e a interação dessas funcionalidades com os usuários do mesmo sistema.
- Não se aprofunda em detalhes técnicos que dizem como o sistema faz.
- Comumente derivado da especificação de requisitos.
- Diagramas de Casos de Uso são compostos basicamente por quatro partes:
  - Cenário: Sequência de eventos que acontecem quando um usuário interage com o sistema.
  - Ator: Usuário do sistema.
  - Use Case: É uma tarefa ou uma funcionalidade realizada pelo ator (usuário)
  - Comunicação: é o que liga um ator com um caso de uso



# UML – Diagrama de Casos de Uso

- Na especificação de um caso temos fluxos:
  - Fluxo Principal.
  - Fluxo Alternativo.
  - Fluxo de Exceção.

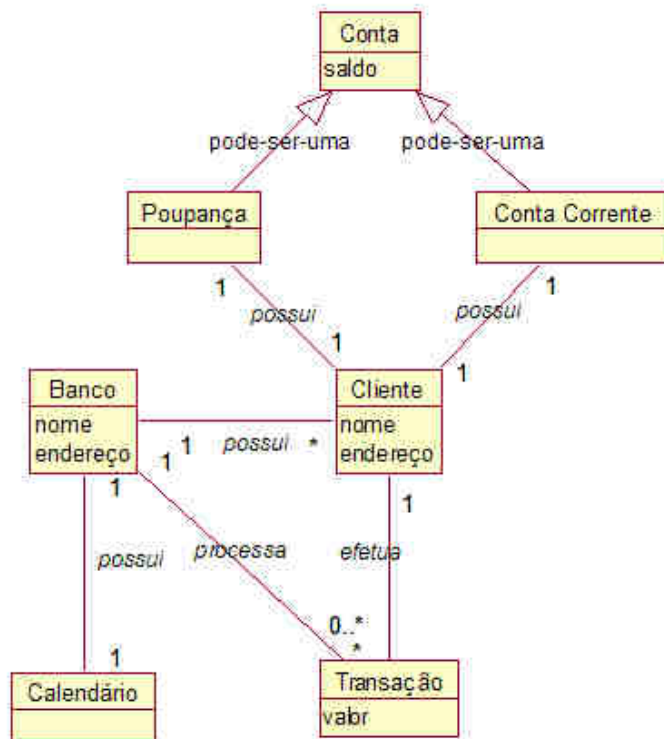


# UML – Diagrama de Classes

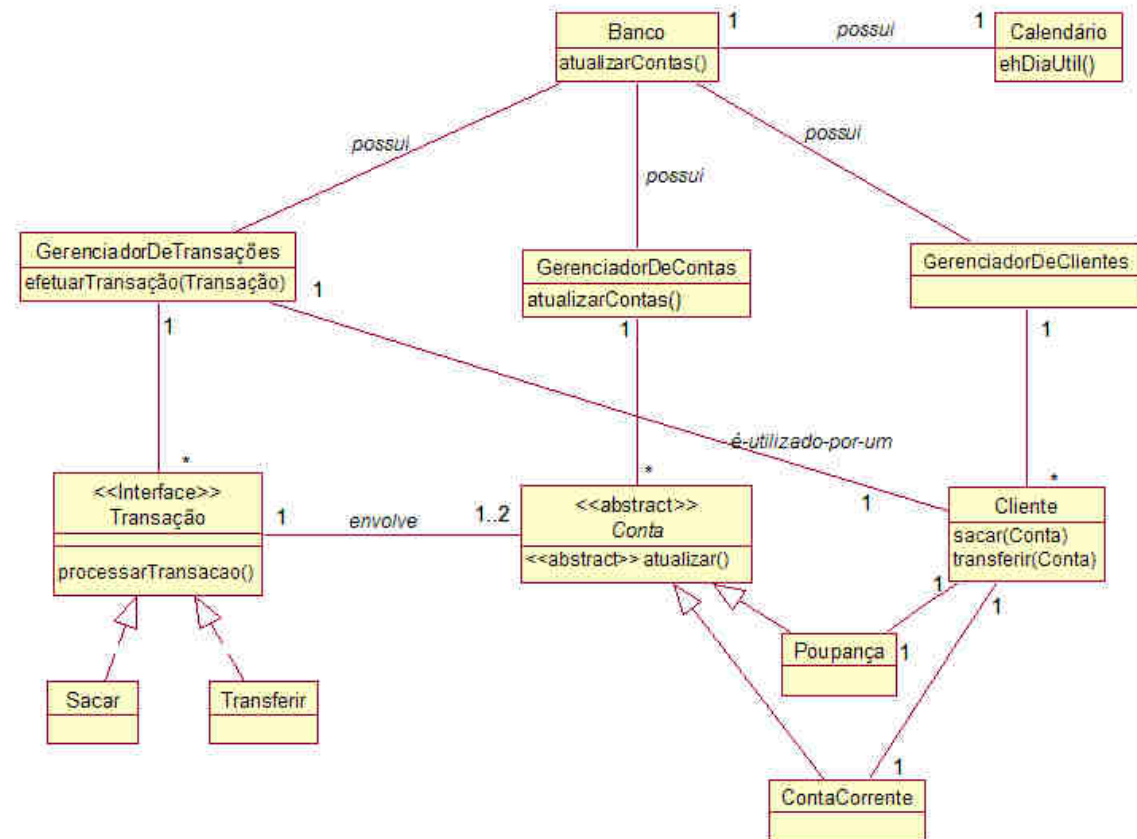
- Descrever os vários tipos de objetos no sistema e o relacionamento entre eles.
- Um diagrama de classes pode oferecer três perspectivas, cada uma para um tipo de observador diferente. São elas:
- Conceitual
  - Representa os conceitos do domínio em estudo.
  - Perspectiva destinada ao cliente.
- Especificação
  - Tem foco nas principais interfaces da arquitetura, nos principais métodos, e não como eles irão ser implementados.
  - Perspectiva destinada as pessoas que não precisam saber detalhes de desenvolvimento, tais como gerentes de projeto.
- Implementação - *a mais utilizada de todas*
  - Aborda vários detalhes de implementação, tais como navegabilidade, *tipo* dos atributos, etc.
  - Perspectiva destinada ao time de desenvolvimento.

# UML – Diagrama de Classes

Conceitual



Especificação



# UML – Diagrama de Classes

## Implementação

