Modelagem UML

(Unified Modeling Language, Linguagem de Modelagem Unificada)

Alunos:

- Lucas Chagas Santos
- Luis Miguel freitas de Jesus

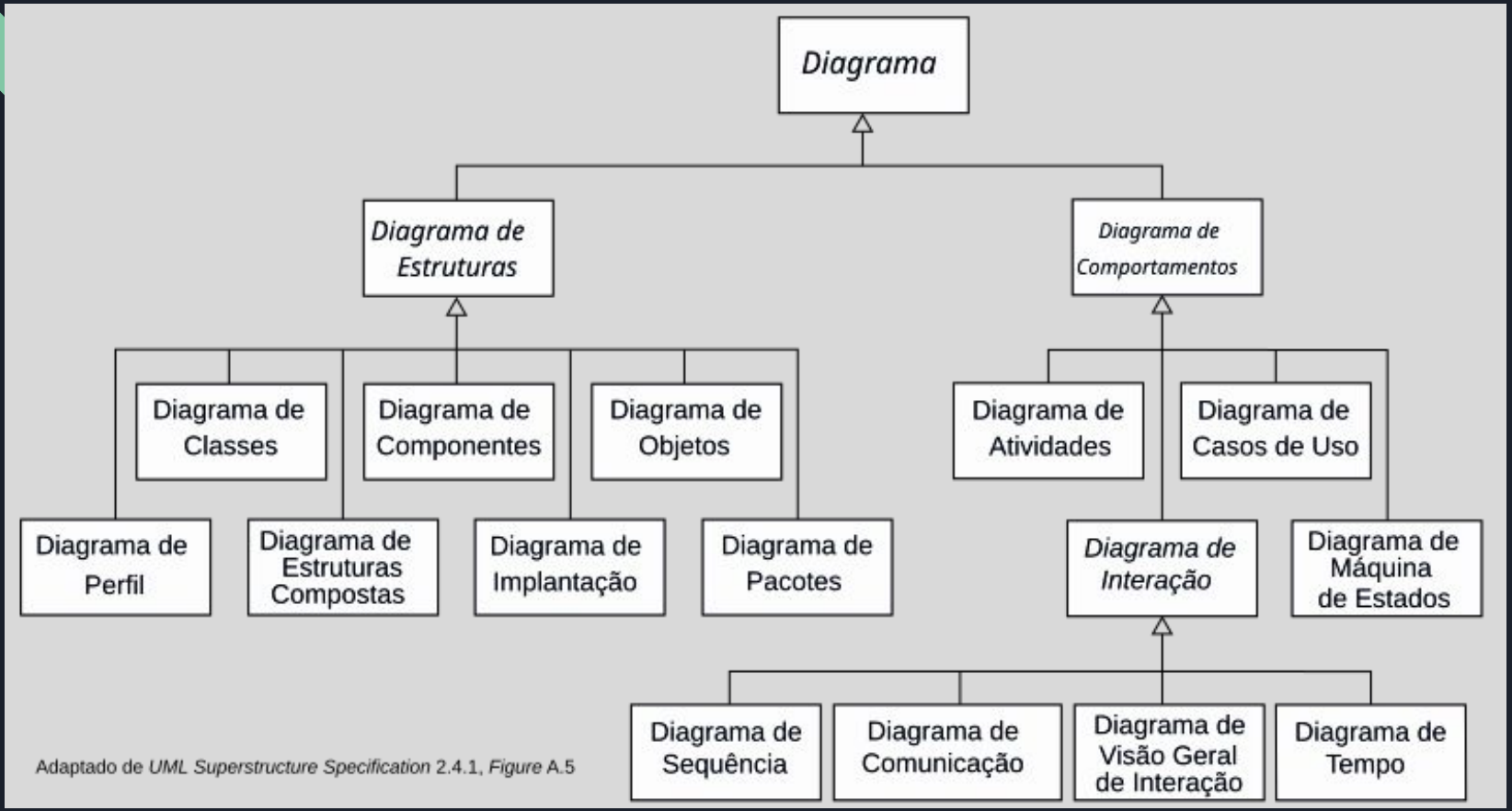
Modelagem UML

- O que é?

É uma linguagem-padrão para a elaboração da estrutura de projetos de software. Ela poderá ser empregada para a visualização, a especificação, a construção e a documentação de artefatos que façam uso de sistemas complexos de software. Em outras palavras, na área de Engenharia de Software, a UML é uma linguagem de modelagem que permite representar um sistema de forma padronizada (com intuito de facilitar a compreensão de pré-implementação).

UML é adequada para a modelagem de sistemas, cuja abrangência poderá incluir desde sistemas de informação corporativos a serem distribuídos a aplicações baseadas na Web e até sistemas complexos embutidos de tempo real. É uma linguagem muito expressiva, abrangendo todas as visões necessárias ao desenvolvimento e implantação desses sistemas.

Exemplo de diagramas da UML



Adaptado de *UML Superstructure Specification* 2.4.1, *Figure* A.5

PlantUML

- O que é?

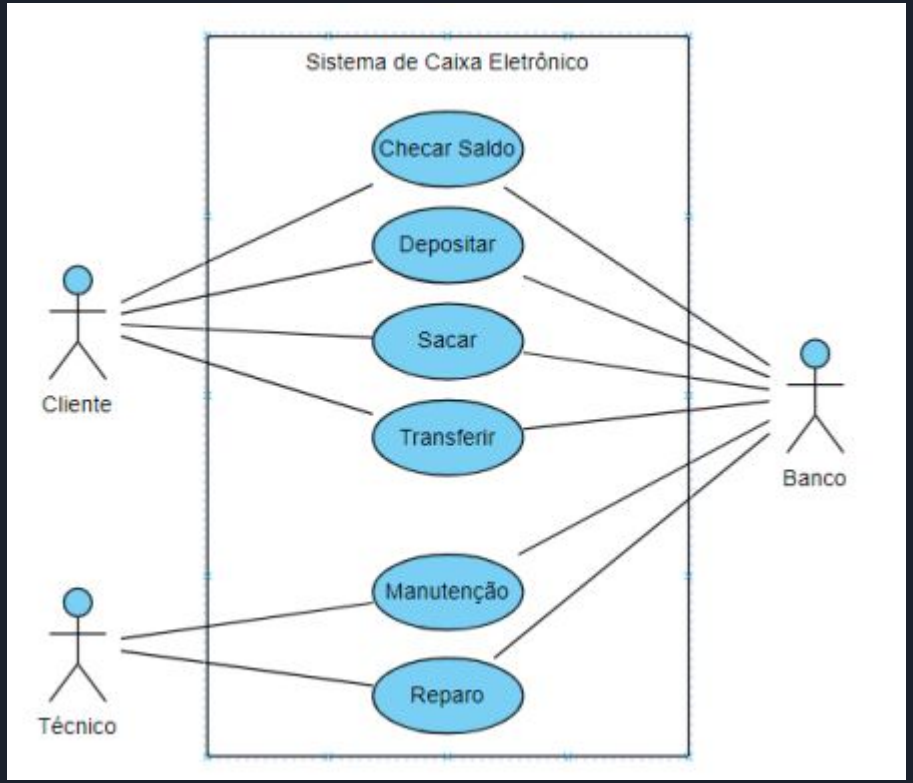
PlantUML é uma ferramenta que permite criar diagramas UML de maneira rápida e automatizada usando texto simples. A integração com ambientes de desenvolvimento facilita atualizações e mantém a documentação coerente com o código-fonte.

```
@startuml
class Usuario {
  +id
  +nome
  +tipoUsuario
}

class Livro {
  +id
  +titulo
  +autor
  +estado
}

Usuario "1" -- "*" Emprestimo : realiza
Livro "1" -- "*" Emprestimo : possui

class Emprestimo {
  +id
  +dataEmprestimo
  +dataDevolucao
  +renovacoes
}
@enduml
```

Principais diagramas da UML.

1. Diagrama de Casos de Uso (Use-Case Diagram)

Representa as funcionalidades do sistema sob a perspectiva do usuário, mostrando as interações que ocorrem. No contexto da biblioteca, incluiria casos como "Realizar Empréstimo", "Cadastrar Usuário", "Buscar Livros". Ajuda na definição clara de funcionalidades e papéis de cada usuário.



Principais diagramas da UML.

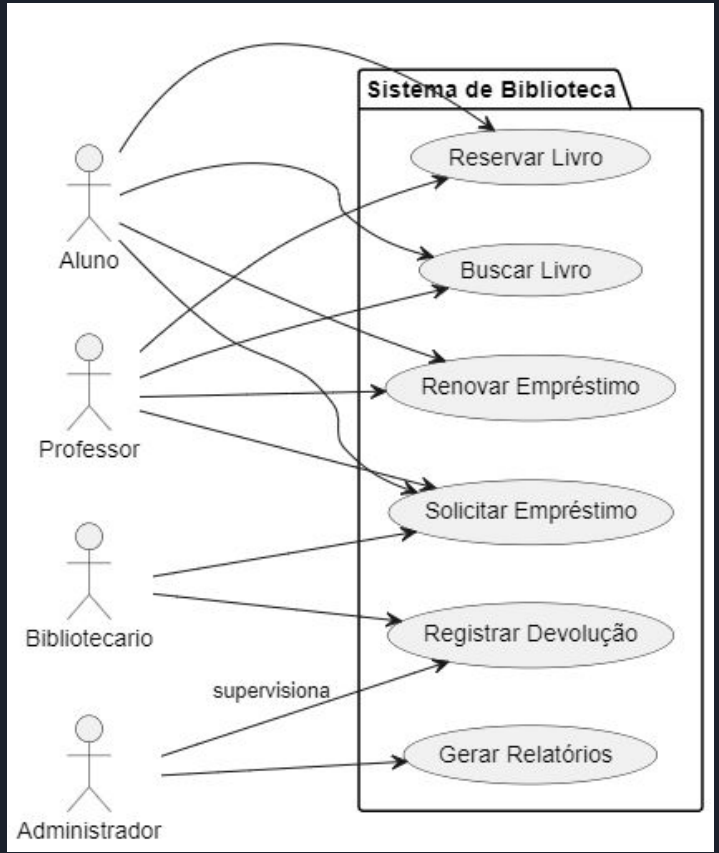
1. Diagrama de Casos de Uso (Use-Case Diagram)

- **Propósito**: Capturar *o que* o sistema fará do ponto de vista do usuário, não *como*.
- **Perguntas que responde**
  - Quem interage com o sistema? (atores)
  - Quais objetivos esses atores desejam atingir? (casos de uso)
  - Há dependências obrigatórias ou opcionais entre esses objetivos? (<<include>>, <<extend>>).
- **Quando usar**: logo no início do projeto, durante a elicitação de requisitos; excelente para conversar com stakeholders não técnicos.
- **Valor agregado**
  - Define a fronteira do sistema de forma visual.
  - Permite priorizar funções antes de mergulhar na arquitetura.
- **Exemplo prático**: “Aluno” reserva livro; “Bibliotecário” registra devolução – ambos são facilmente vistos no diagrama, tornando claras as responsabilidades.

Principais diagramas da UML.

Exemplo de diagrama de Casos de Uso (Use-Case Diagram)

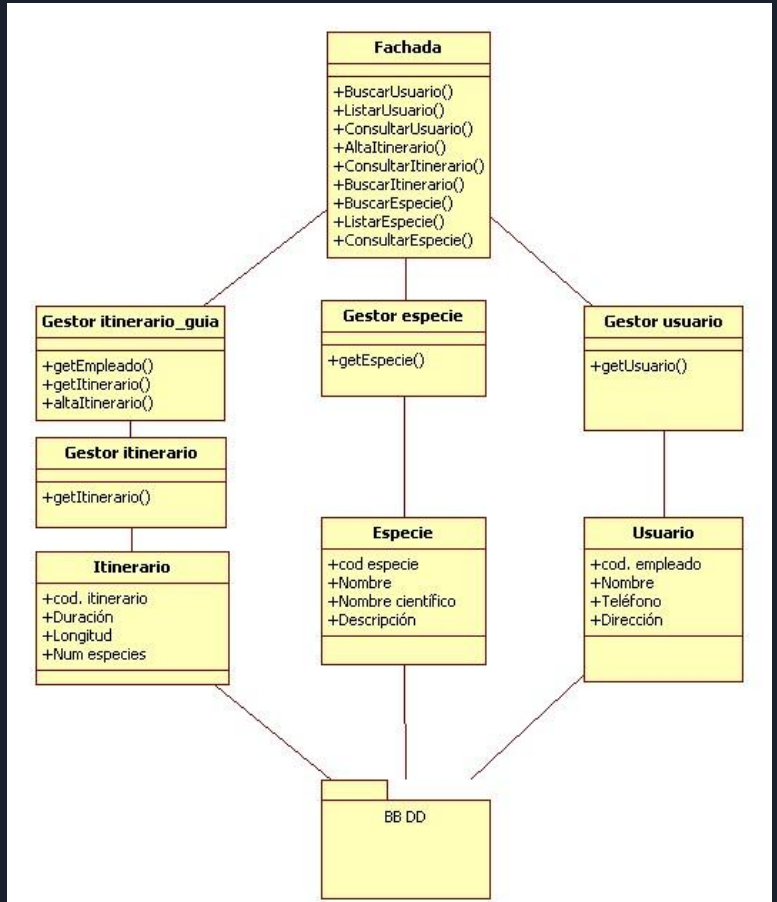
```
You, 20 hours ago | 1 author (You)
@startuml Biblioteca-UseCases
left to right direction
actor Aluno
actor Professor
actor Bibliotecario as BIB
actor Administrador as ADM

package "Sistema de Biblioteca" {
  usecase "Buscar Livro" as UC_Busca
  usecase "Reservar Livro" as UC_Reserva
  usecase "Renovar Empréstimo" as UC_Renova
  usecase "Solicitar Empréstimo" as UC_Solicita
  usecase "Registrar Devolução" as UC_Devolucao
  usecase "Gerar Relatórios" as UC_Relatorio
}

Aluno --> UC_Busca
Aluno --> UC_Reserva
Aluno --> UC_Renova
Aluno --> UC_Solicita

Professor --> UC_Busca
Professor --> UC_Reserva
Professor --> UC_Renova
Professor --> UC_Solicita

BIB --> UC_Devolucao
BIB --> UC_Solicita

ADM --> UC_Relatorio
ADM --> UC_Devolucao : supervisiona
@enduml
```



Principais diagramas da UML.

2. Diagrama de Classes (Class Diagram)

Mostra a estrutura estática do sistema, definindo classes, atributos, operações e as relações entre os objetos. Para a biblioteca, as classes principais seriam "Livro", "Usuário", "Empréstimo", com relacionamentos de associação, agregação ou composição, e herança para diferentes tipos de usuários (aluno, professor, bibliotecário).



Principais diagramas da UML.

2. Diagrama de Classes (Class Diagram)

- **Propósito**: Descrever a estrutura estática do domínio: classes, atributos, métodos e relacionamentos (associação, herança, composição, agregação, dependência).
- **Perguntas que responde**
  - Quais entidades compõem o domínio?
  - Como elas se relacionam e quais regras de cardinalidade existem?
  - Que operações cada entidade oferece?
- **Quando usar**: durante análise de domínio e desenho da solução; serve de base para gerar código ou schema de banco de dados.
- **Valor agregado**
  - Torna explícitas as regras de negócio (ex.: “Empréstimo contém 1..* Livros”).
  - Facilita detecção de ciclos e acoplamento excessivo.
- **Exemplo prático**: Usuario abstrato com subclasses Aluno, Professor; enum EstadoLivro prepara terreno para diagrama de estados.

Principais diagramas da UML.

Exemplo de diagrama de Classes (Class Diagram).

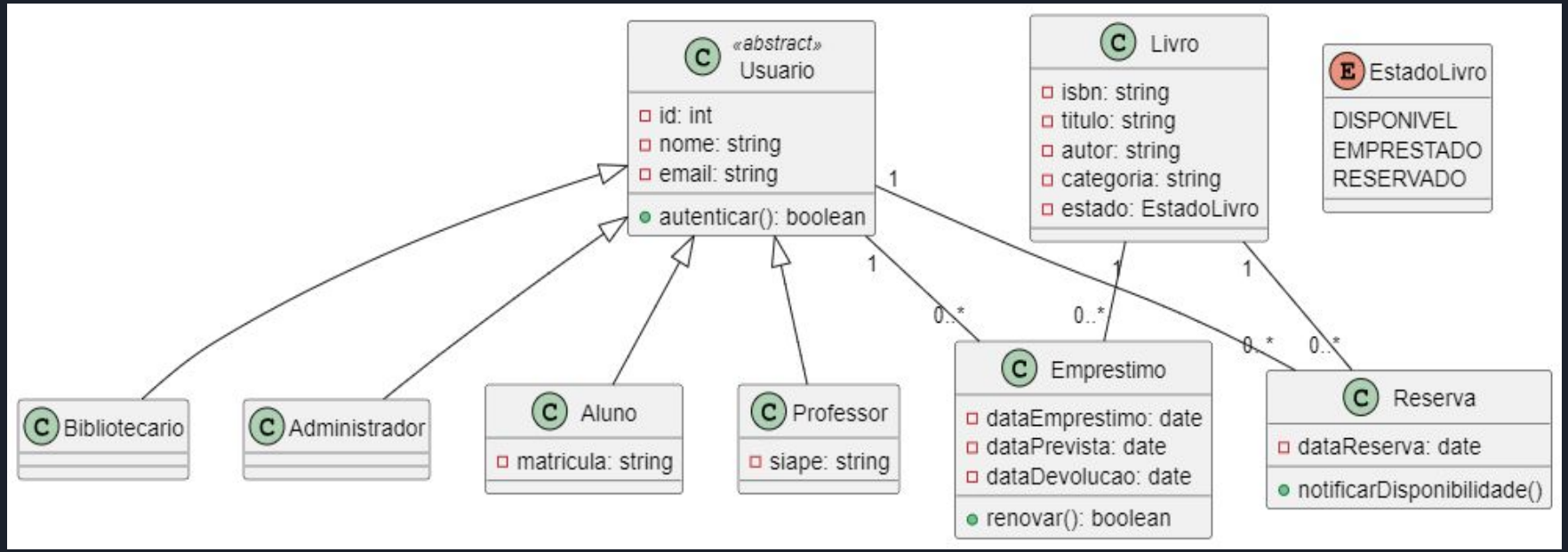
```
@startuml Biblioteca-Classes
class Usuario <<abstract>> {
  -id: int
  -nome: string
  -email: string
  +autenticar(): boolean
}

class Aluno {
  -matricula: string
}
class Professor {
  -siape: string
}
class Bibliotecario
class Administrador

Usuario <|-- Aluno
Usuario <|-- Professor
Usuario <|-- Bibliotecario
Usuario <|-- Administrador
```

```
class Livro {
  -isbn: string
  -titulo: string
  -autor: string
  -categoria: string
  -estado: EstadoLivro
}

enum EstadoLivro {
  DISPONIVEL
  EMPRESTADO
  RESERVADO
}

class Emprestimo {
  -dataEmprestimo: date
  -dataPrevista: date
  -dataDevolucao: date
  +renovar(): boolean
}
```

```
class Reserva {
  -dataReserva: date
  +notificarDisponibilidade()
}

Usuario "1" -- "0..*" Emprestimo
Livro   "1" -- "0..*" Emprestimo

Usuario "1" -- "0..*" Reserva
Livro   "1" -- "0..*" Reserva
@enduml
```

Principais diagramas da UML.

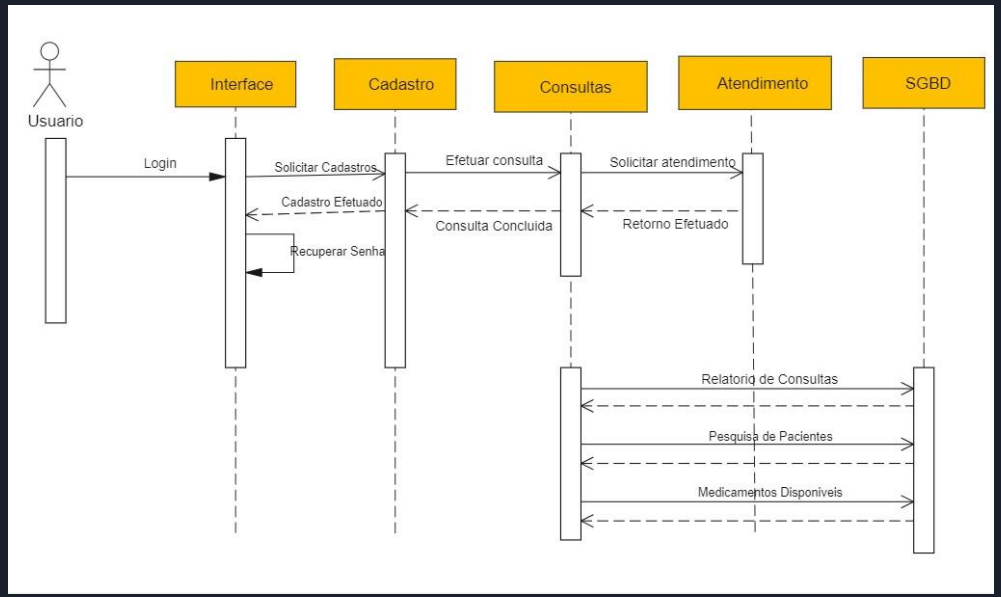
Exemplo de diagrama de Classes (Class Diagram).



Principais diagramas da UML.

3. Diagrama de Sequência (Sequence Diagram)

Descreve como objetos interagem ao longo do tempo para realizar uma função específica. Um exemplo prático seria detalhar o fluxo de "Empréstimo de Livros", mostrando as mensagens entre objetos como Usuário, Livro, e Sistema durante o processo de empréstimo.



Principais diagramas da UML.

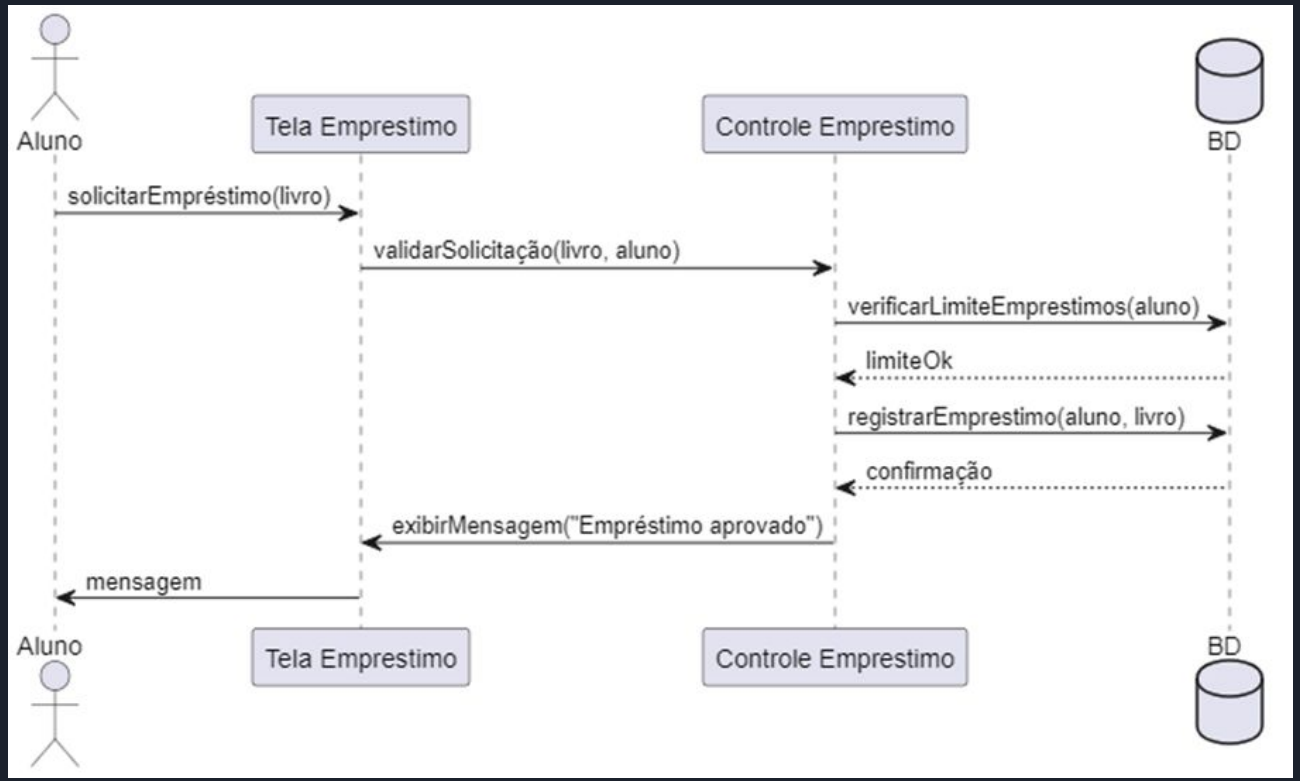
3. Diagrama de Sequência (Sequence Diagram)

- **Propósito**: Modelar a interação temporal entre objetos/partes do sistema para realizar um cenário específico.
- **Perguntas que responde**
  - Qual é a ordem exata das mensagens?
  - Que objeto inicia a operação? Onde ocorrem validações?
  - Há paralelismo ou chamadas assíncronas?
- **Quando usar**: ao refinar casos de uso críticos, definir APIs, ou depurar fluxos complexos (login, pagamento, transação bancária).
- **Valor agregado**
  - Revela dependências ocultas e gargalos de performance.
  - Alinha desenvolvedores sobre quem chama quem e em que ordem.
- **Exemplo prático**: sequência “Aluno solicita empréstimo” → sistema checa limite → bibliotecário confirma → banco registra.

Principais diagramas da UML.

Exemplo de diagrama de Sequência (Sequence Diagram).

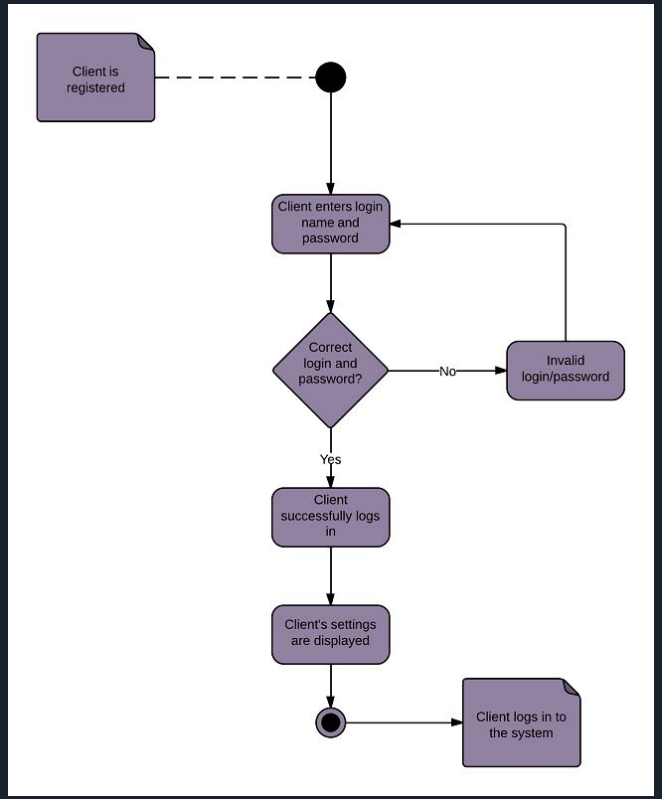
```
@startuml Seq-Emprestimo
actor Aluno
participant "Tela Emprestimo" as Tela
participant "Controle Emprestimo" as Ctrl
database "BD" as DB

Aluno -> Tela : solicitarEmpréstimo(livro)
Tela -> Ctrl  : validarSolicitação(livro, aluno)
Ctrl -> DB    : verificarLimiteEmprestimos(aluno)
DB --> Ctrl   : limiteOk
Ctrl -> DB    : registrarEmprestimo(aluno, livro)
DB --> Ctrl   : confirmação
Ctrl -> Tela  : exibirMensagem("Empréstimo aprovado")
Tela -> Aluno : mensagem
@enduml      You, 20 hours ago • Adicionando Arquivio
```



Principais diagramas da UML.

4. Diagrama de Atividades (Activity Diagram)

Representa o fluxo de atividades e decisões, como um fluxograma. No caso da biblioteca, pode modelar o processo de "Renovação de Empréstimos", destacando decisões como "Livro disponível para renovação?" e ações subsequentes.



Principais diagramas da UML.

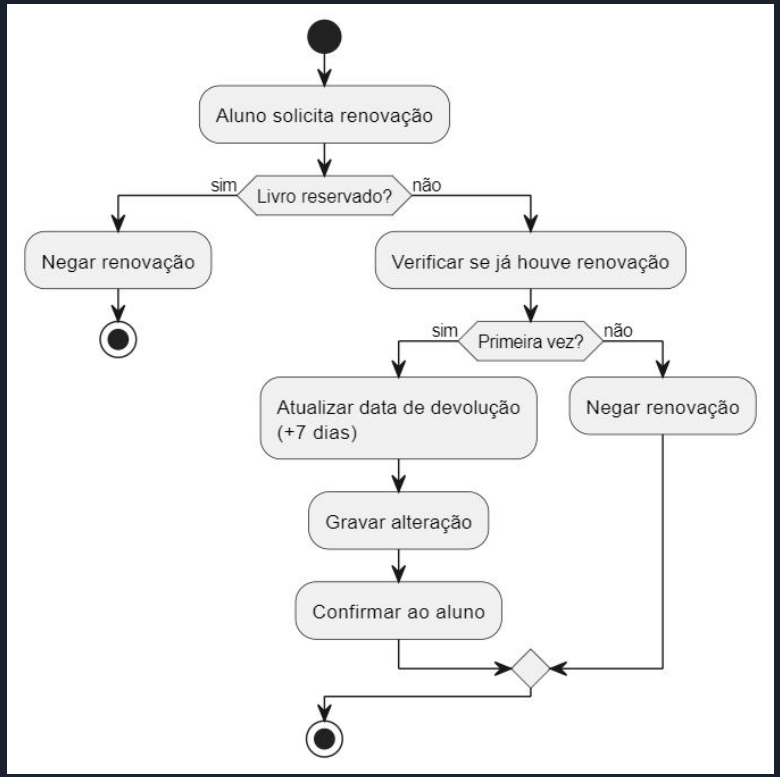
4. Diagrama de Atividades (Activity Diagram)

- **Propósito**: Representar fluxos de trabalho, algoritmos ou processos de alto nível, incluindo decisões, paralelismo e loops.
- **Perguntas que responde**
  - Quais passos compõem um processo?
  - Onde ocorrem bifurcações (decision nodes) e junções (merge/join)?
  - Quais atividades podem executar em paralelo?
- **Quando usar**: para documentar regras de negócio, processos de negócio BPM, ou lógica de métodos complexos.
- **Valor agregado**
  - Comunica claramente caminhos alternativos e exceções.
  - Serve de ponte entre análise de requisitos e modelagem de processo (BPMN).
- **Exemplo prático**: processo de renovação de empréstimo – vértice inicial, verificação de reserva, ramificação “permitido / negado”, vértice final.

Principais diagramas da UML.

Exemplo de Diagrama de Atividades (Activity Diagram).

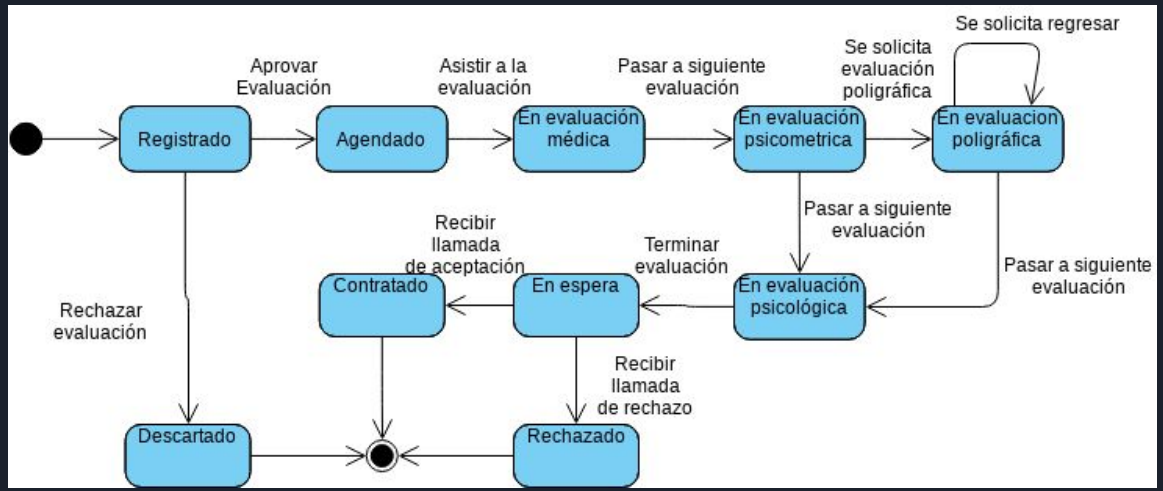
```
@startuml Atividade-Renovacao
start      You, 20 hours ago • Adicionando Arquivios d
:Aluno solicita renovação;
if (Livro reservado?) then (sim)
  :Negar renovação;
  stop
else (não)
  :Verificar se já houve renovação;
  if (Primeira vez?) then (sim)
    :Atualizar data de devolução\n(+7 dias);
    :Gravar alteração;
    :Confirmar ao aluno;
  else (não)
    :Negar renovação;
  endif
endif
stop
@enduml
```



Principais diagramas da UML.

5. Diagrama de Estados (State-Machine Diagram)

Mostra o ciclo de vida dos objetos, exibindo seus possíveis estados e as transições entre eles. Um exemplo para a biblioteca seria o estado do "Livro": Disponível -> Reservado -> Emprestado -> Disponível novamente.



Principais diagramas da UML.

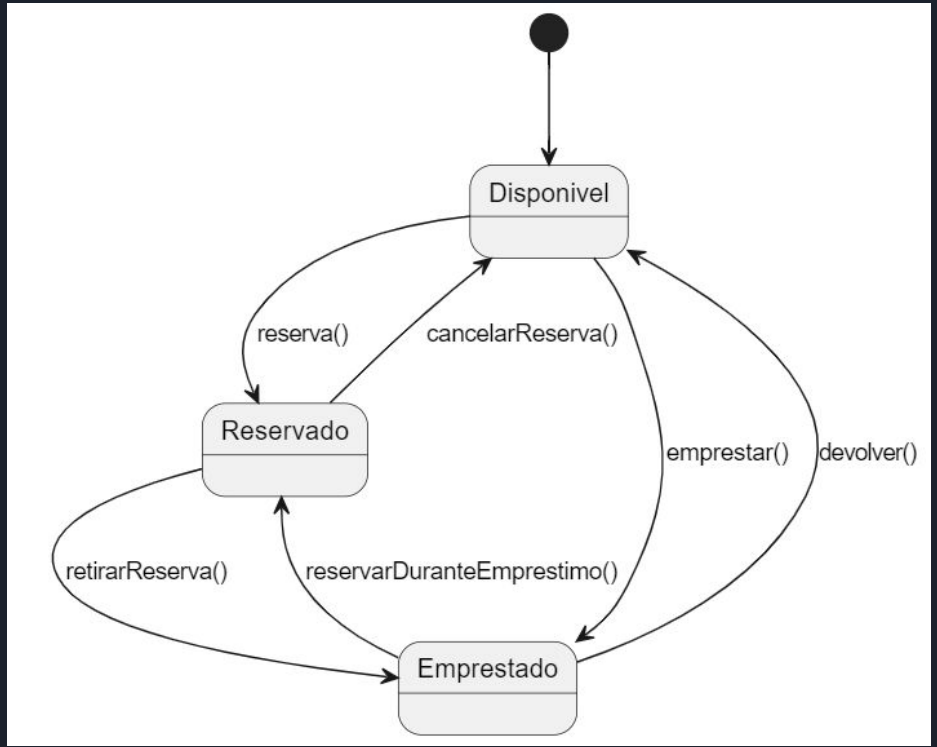
5. Diagrama de Estados (State-Machine Diagram)

- **Propósito**: Mostrar os estados possíveis de um objeto (ou sistema) e os eventos que provocam transições entre esses estados.
- **Perguntas que responde**
  - Quais são os estados válidos?
  - Que eventos disparam mudanças?
  - Existem transições condicionais ou ações de entrada/saída?
- **Quando usar**: em entidades com ciclo de vida complexo (documentos, dispositivos IoT, ordens de compra) ou para validar lógica de controle.
- **Valor agregado**
  - Detecta estados inalcançáveis ou transições faltantes antes da implementação.
  - Auxilia na escrita de testes baseados em eventos.
- **Exemplo prático**: Livro → estados *Disponível*, *Reservado*, *Emprestado*; eventos reservar(), devolver(), emprestar().

Principais diagramas da UML.

Exemplo de Diagrama de Estados (State-Machine Diagram).

```
You, 20 hours ago | 1 author (You)
@startuml Estado-Livro
[*] --> Disponivel

Disponivel --> Reservado : reserva()
Disponivel --> Emprestado : emprestar()

Reservado --> Emprestado : retirarReserva()
Reservado --> Disponivel : cancelarReserva()

Emprestado --> Disponivel : devolver()
Emprestado --> Reservado : reservarDuranteEmprestimo()

@enduml      You, 20 hours ago • Adicionando Arquivios do Projeto …
```



Em resumo:

- **Casos de Uso** definem escopo funcional e atores.
- **Classes** formalizam a estrutura estática do domínio.
- **Sequência** detalha a dinâmica de um cenário específico.
- **Atividades** mapeiam processos com fluxo de controle.
- **Estados** descrevem a evolução temporal de um objeto.

Juntos, esses cinco diagramas oferecem visão complementar que vai do requisito ao funcionamento interno, garantindo entendimento compartilhado e redução de risco durante o desenvolvimento.

Referências

- https://pt.wikipedia.org/wiki/UML
- https://www.devmedia.com.br/modelagem-de-sistemas-atraves-de-uml-uma-visao-geral/27913

Muito obrigado pela atenção!!!