Projeto Integrador II – Profa. Silvia Garcia

1. O que é o PlantUML?

PlantUML é uma ferramenta *open source* que permite criar diagramas a partir de texto simples escrito em uma linguagem de marcação própria. É muito utilizada em documentação técnica, engenharia de software e projetos ágeis.

Vantagens:

- Simplicidade: base textual
- Integração com editores (VSCode, IntelliJ, Obsidian)
- Pode ser usado offline ou online: https://plantuml.com/plantuml

---

2. Tipos de Diagramas que podem ser criados

2.1 Diagrama de Casos de Uso

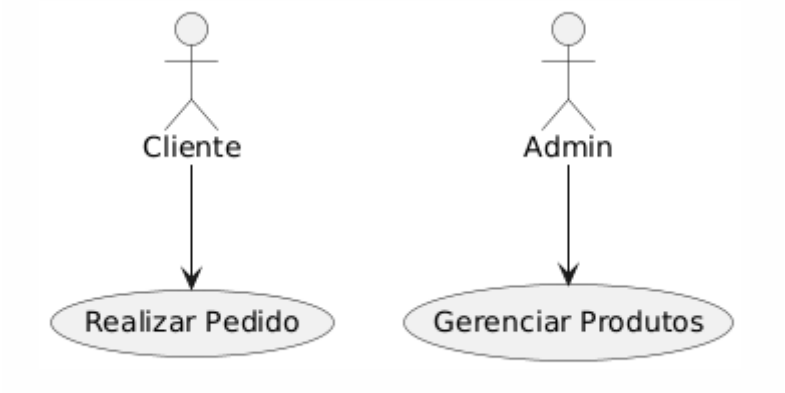
Mostra as funcionalidades (casos de uso) do sistema e os atores envolvidos.

Exemplo:

```
@startuml
actor Cliente
actor Admin

usecase "Realizar Pedido" as RP
usecase "Gerenciar Produtos" as GP

Cliente --> RP
Admin --> GP
@enduml
```



2.2 Diagrama de Classes

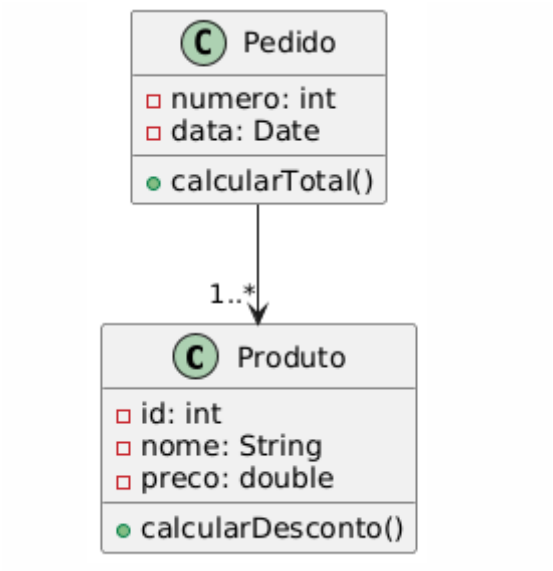
Modela a estrutura do sistema com classes, atributos, métodos e relacionamentos.

Exemplo:

```
@startuml
class Produto {
  - id: int
  - nome: String
  - preco: double
  + calcularDesconto()
}

class Pedido {
  - numero: int
  - data: Date
  + calcularTotal()
}

Pedido --> "1..*" Produto
@enduml
```



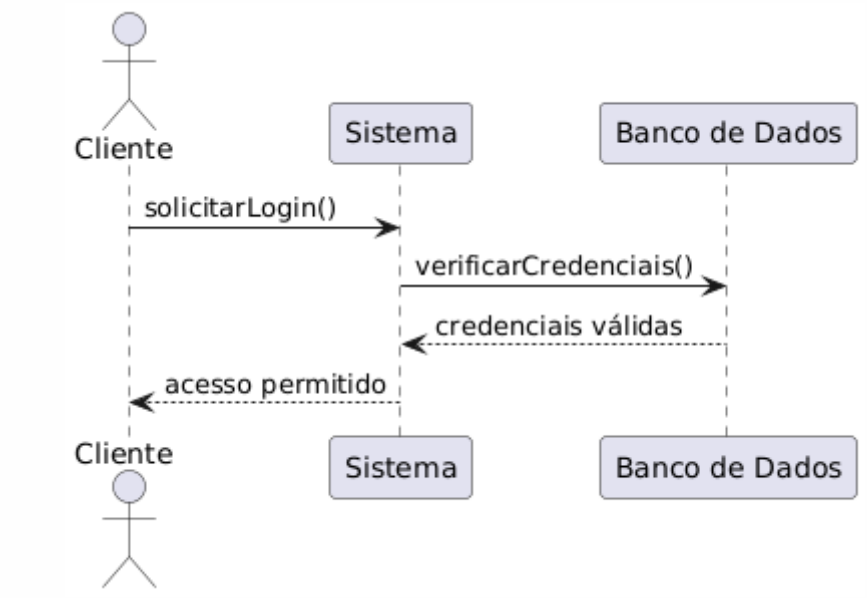
2.3 Diagrama de Sequência

Mostra a interação entre objetos no tempo.

Exemplo:

```
@startuml
actor Cliente
participant "Sistema" as S
participant "Banco de Dados" as BD

Cliente -> S: solicitarLogin()
S -> BD: verificarCredenciais()
BD --> S: credenciais válidas
S --> Cliente: acesso permitido
@enduml
```

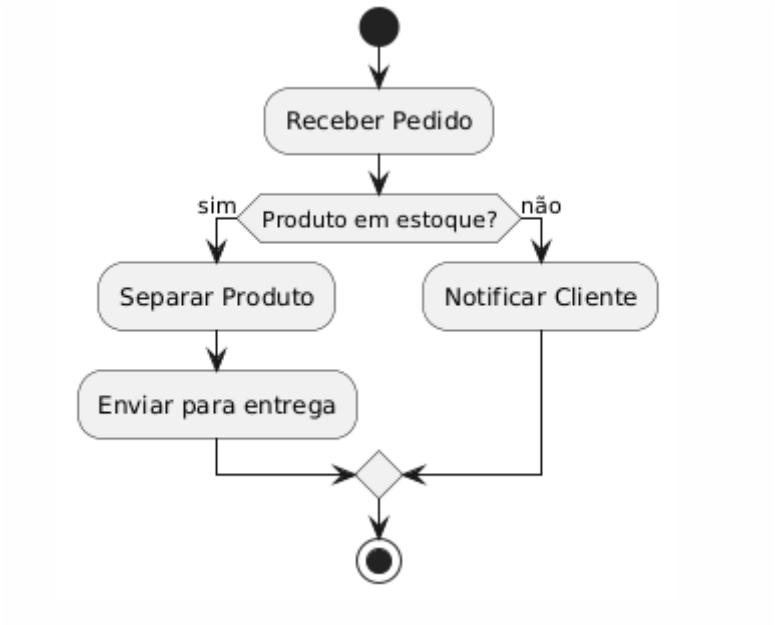


2.4 Diagrama de Atividades

Representa o fluxo de atividades de um processo.

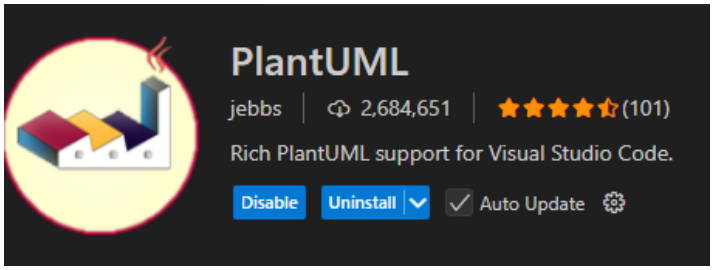
Exemplo:

```
@startuml
start
:Receber Pedido;
if (Produto em estoque?) then (sim)
  :Separar Produto;
  :Enviar para entrega;
else (não)
  :Notificar Cliente;
endif
stop
@enduml
```



VSCode

Extensão: PlantUML



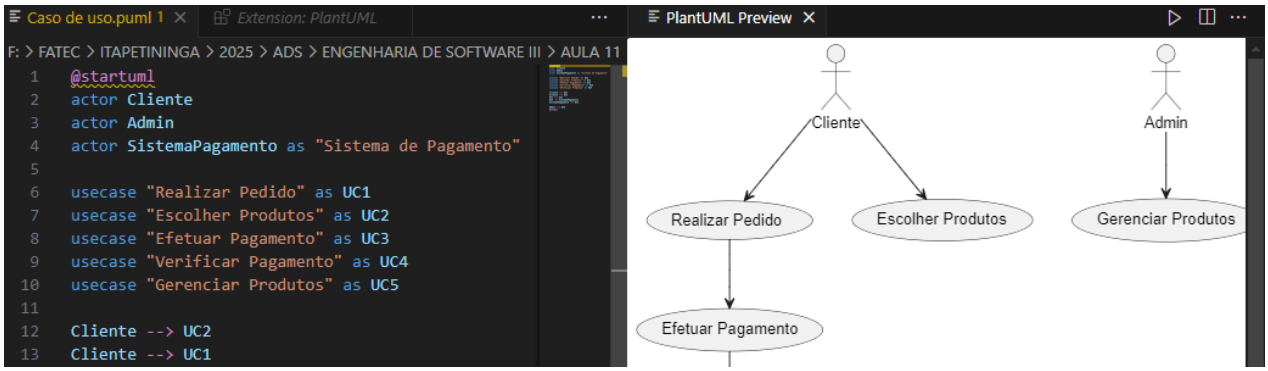
Tipo de arquivo: puml

File – New File

Salvar com a extensão .puml

Após digitar o código utilize as teclas ALT + D para abrir o **Preview** do diagrama

Exemplo:



3. Exemplo Prático

Contexto:

Uma empresa de **e-commerce** está desenvolvendo um novo sistema para **Gestão de pedidos**. Você faz parte da equipe de análise e deve representar os seguintes aspectos do sistema:

Tarefas:

1. **Diagrama de Casos de Uso** representando:
    - Cliente realizando pedido
    - Sistema verificando pagamento
    - Admin gerenciando produtos
2. **Diagrama de Classes** com:
    - Classes: Cliente, Produto, Pedido
    - Relacionamentos e atributos básicos
3. **Diagrama de Atividades** que represente o fluxo:
    - Cliente faz login
    - Escolhe produtos
    - Realiza pagamento
    - Confirmação do pedido
4. **Diagrama de Sequência** que represente o fluxo:

Mostre a comunicação entre os seguintes participantes:

- Cliente
- Sistema
- Sistema de Pagamento
- Banco de Dados

Fluxo sugerido:

1. Cliente solicita login
2. Sistema valida
3. Cliente realiza pedido
4. Sistema envia para pagamento
5. Pagamento retorna confirmação

RESOLUÇÃO:

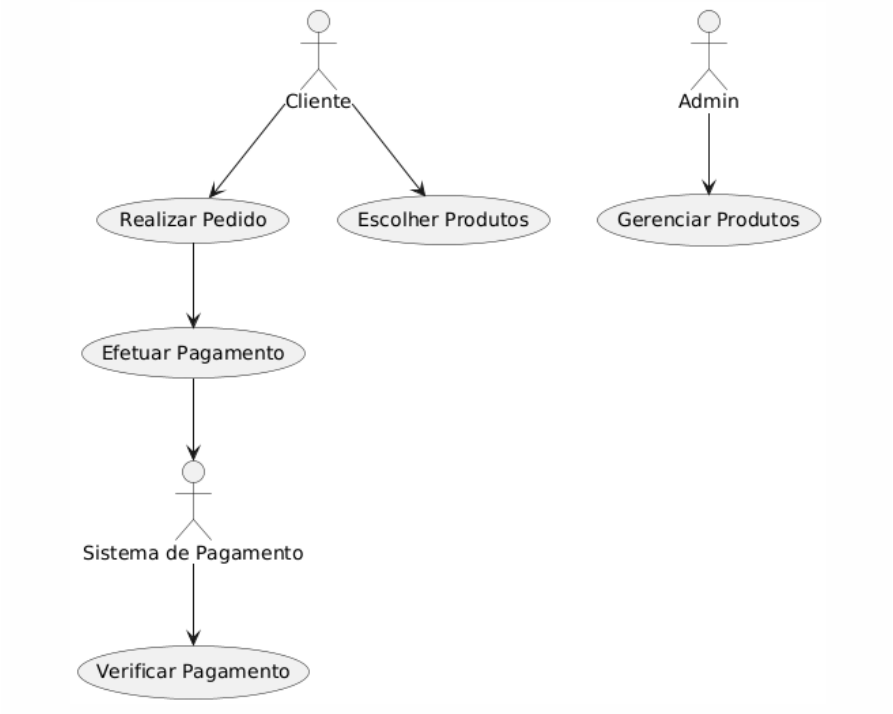
1 – Diagramas de Casos de uso

```
@startuml
actor Cliente
actor Admin
actor SistemaPagamento as "Sistema de Pagamento"

usecase "Realizar Pedido" as UC1
usecase "Escolher Produtos" as UC2
usecase "Efetuar Pagamento" as UC3
usecase "Verificar Pagamento" as UC4
usecase "Gerenciar Produtos" as UC5

Cliente --> UC2
Cliente --> UC1
UC1 --> UC3
UC3 --> SistemaPagamento
SistemaPagamento --> UC4

Admin --> UC5
@enduml
```



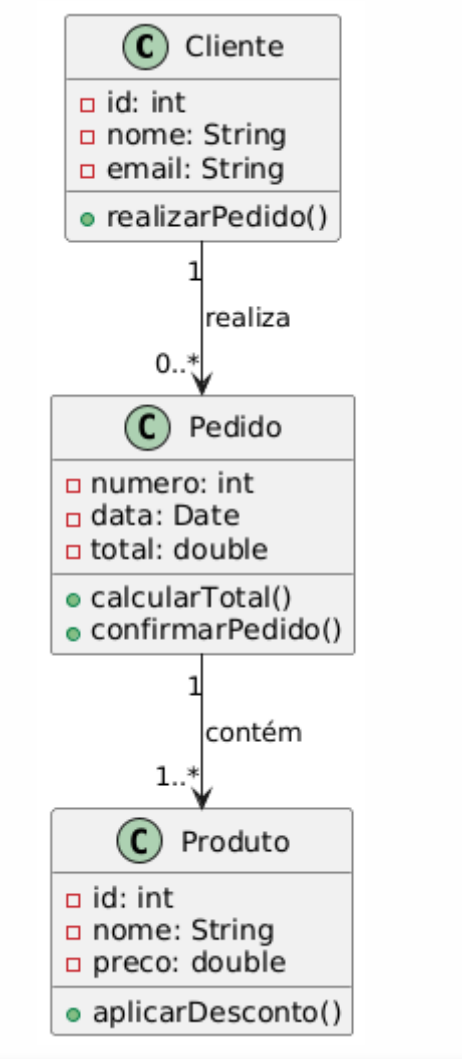
2. Diagrama de Classes

```
@startuml
class Cliente {
  - id: int
  - nome: String
  - email: String
  + realizarPedido()
}

class Produto {
  - id: int
  - nome: String
  - preco: double
  + aplicarDesconto()
}

class Pedido {
  - numero: int
  - data: Date
  - total: double
  + calcularTotal()
  + confirmarPedido()
}

Cliente "1" --> "0..*" Pedido : realiza
Pedido "1" --> "1..*" Produto : contém
@enduml
```



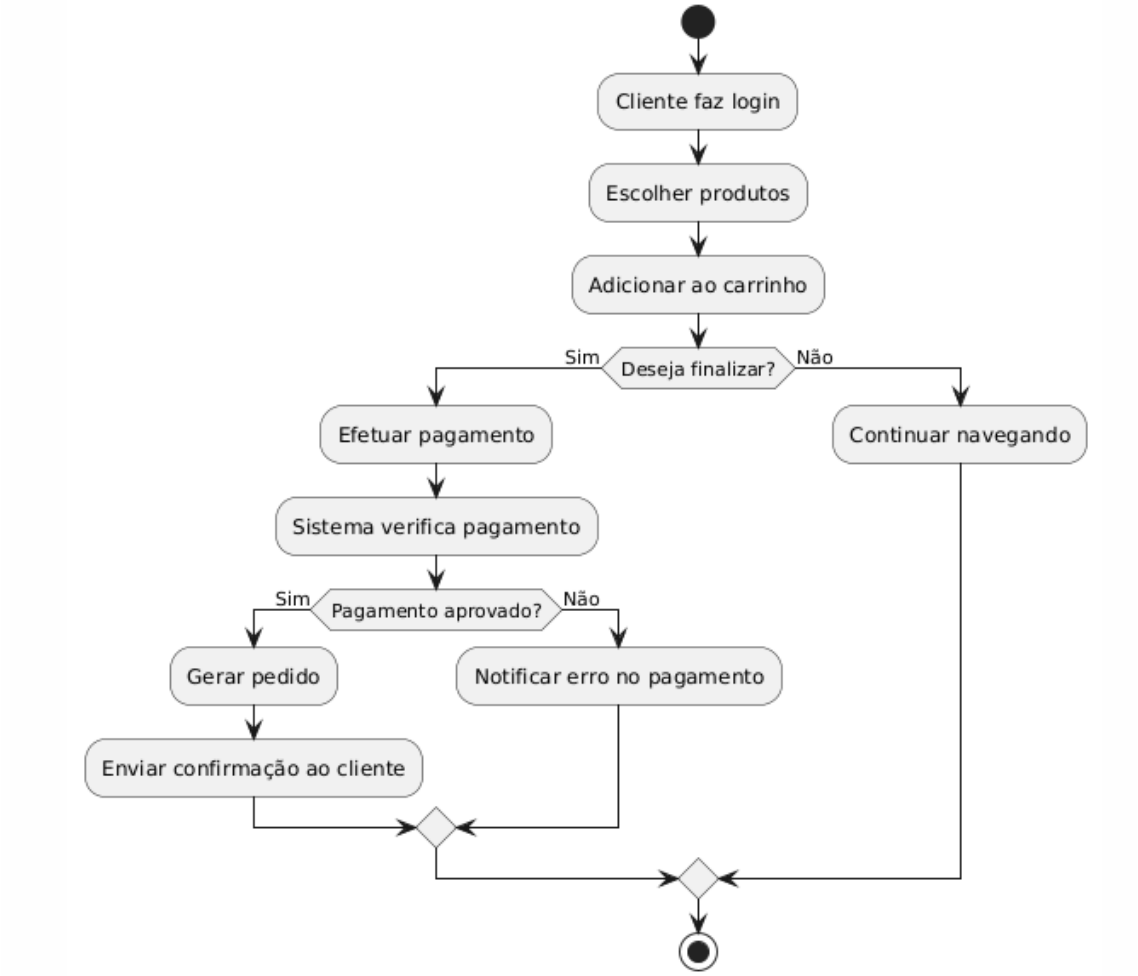
3. Diagrama de Atividades:

```
@startuml
start

:Cliente faz login;
:Escolher produtos;
:Adicionar ao carrinho;

if (Deseja finalizar?) then (Sim)
  :Efetuar pagamento;
  :Sistema verifica pagamento;
  if (Pagamento aprovado?) then (Sim)
    :Gerar pedido;
    :Enviar confirmação ao cliente;
  else (Não)
    :Notificar erro no pagamento;
  endif
else (Não)
  :Continuar navegando;
endif

stop
@enduml
```



4. Diagrama de Sequência:

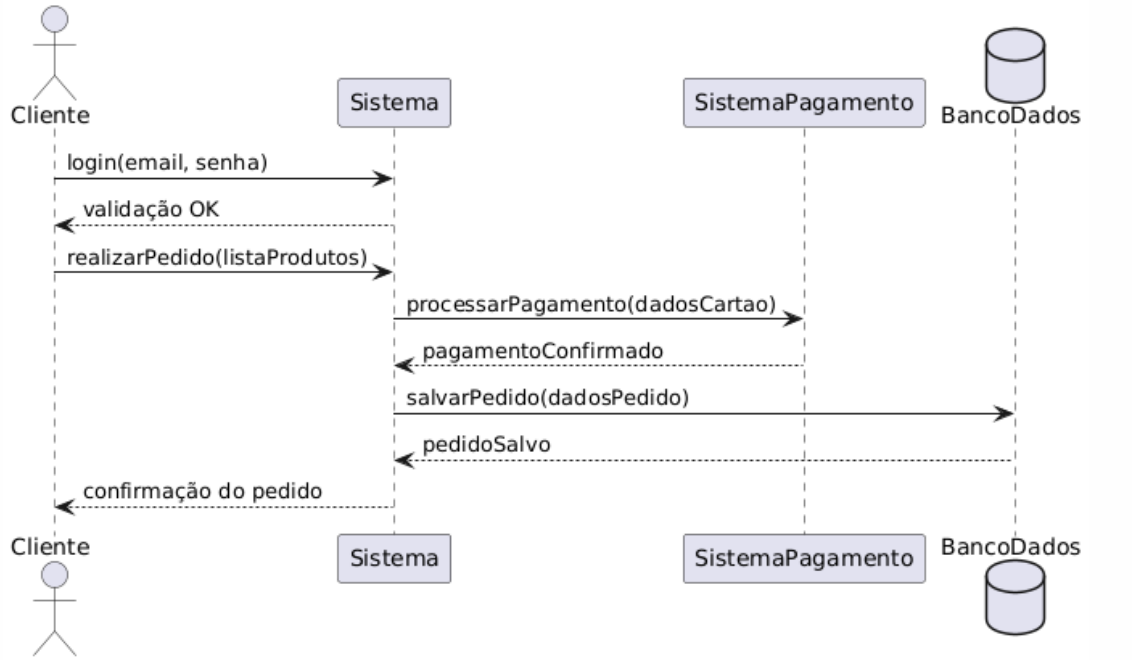
```
@startuml
actor Cliente
participant Sistema
participant SistemaPagamento
database BancoDados

Cliente -> Sistema : login(email, senha)
Sistema --> Cliente : validação OK

Cliente -> Sistema : realizarPedido(listaProdutos)
Sistema -> SistemaPagamento : processarPagamento(dadosCartao)
SistemaPagamento --> Sistema : pagamentoConfirmado

Sistema -> BancoDados : salvarPedido(dadosPedido)
BancoDados --> Sistema : pedidoSalvo

Sistema --> Cliente : confirmação do pedido
@enduml
```



ATIVIDADE PRÁTICA - PI:

- De acordo com o Levantamento de Requisitos realizado e os níveis de usuário a seguir, elabore os diagramas:

  - Diagrama de Caso de uso

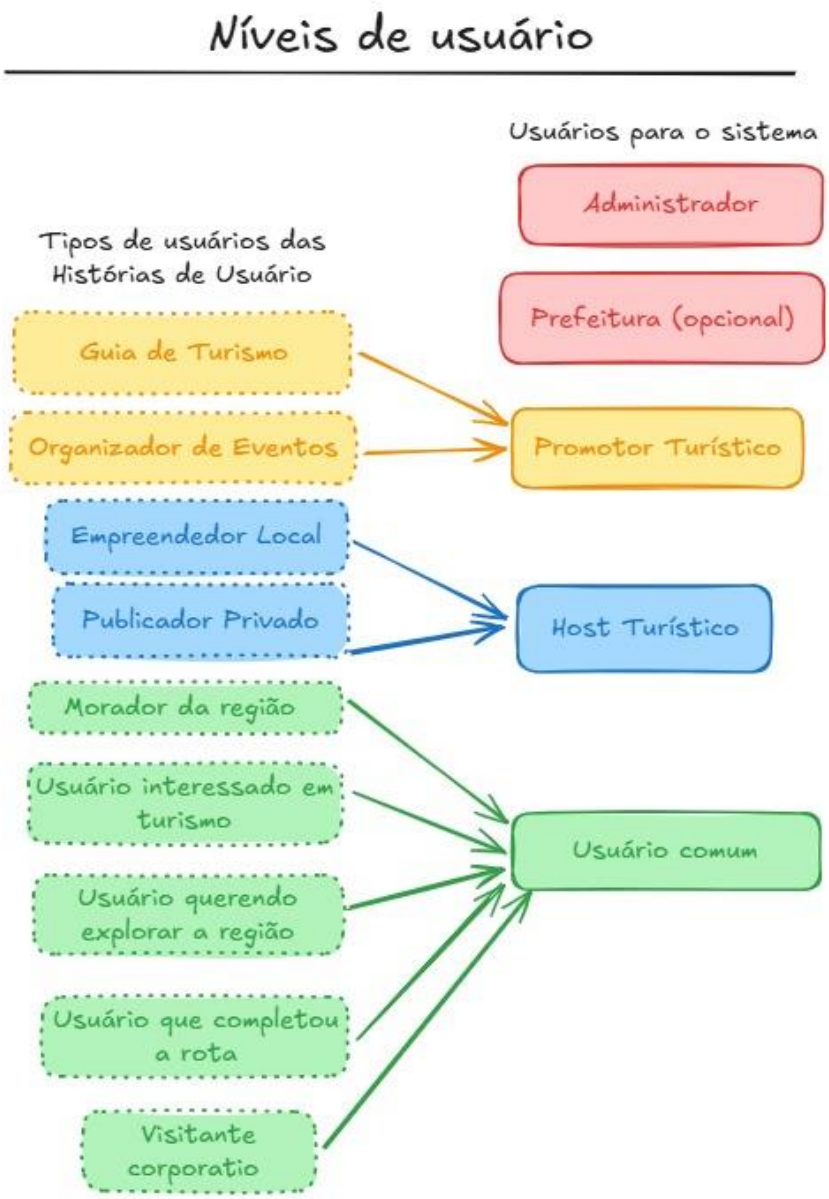
  - Diagrama de Classes

  - Diagrama de Atividades

  - Diagrama de Sequência

Entregar arquivo único em PDF, contendo:
- Diagramas (PNG) e códigos (para cada diagrama solicitado).

- **Níveis e usuário:**



Referências

- Site oficial: https://plantuml.com/
- Cheat Sheet: https://plantuml.com/guide
- Editor online: https://www.plantuml.com/plantuml/uml