# DIAGRAMA DE CLASSES

*“Uma representação visual da estrutura de um sistema, destacando suas classes, atributos, métodos e os relacionamentos.”*

*Programação Orientada à Objetos*

*Código da Disciplina: TI0196*

# INTRODUÇÃO À UML

A UML – Unified Modeling Language ou Linguagem de Modelagem Unificada – é uma linguagem visual utilizada para modelar softwares baseados no paradigma de orientação a objetos.

# POR QUE USAR DIAGRAMAS UML?

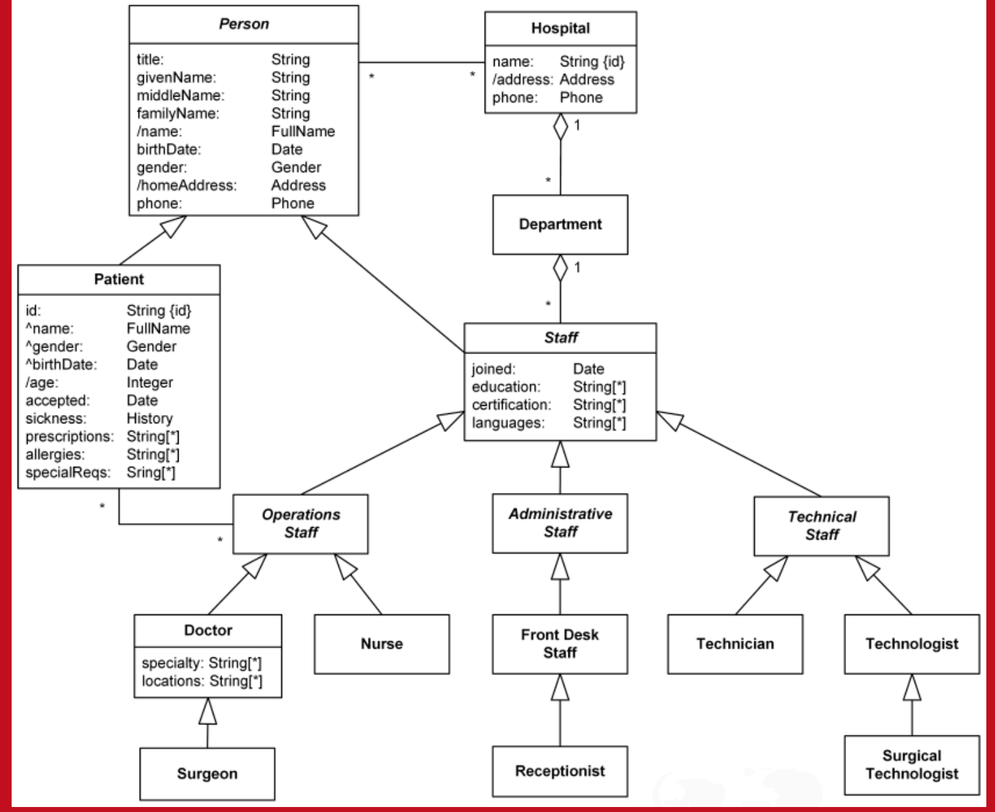
01 Facilita a compreensão de ideias e sistemas

02 Transforma códigos complexos em um diagrama visual

03 Usar uma notação padrão torna-se mais compreensível à todos

04 Permite ter o panorama geral de um sistema

05 Orienta os colaboradores sobre os processos e o software




# INTRODUÇÃO À DIAGRAMA DE CLASSES

- Permite a visualização das classes do sistema, seus atributos, métodos e relacionamentos.
- Mostra a organização das classes e define sua estrutura lógica.
- Componentes principais: classes e associações.
- Fases de desenvolvimento:
  - Análise
  - Projeto

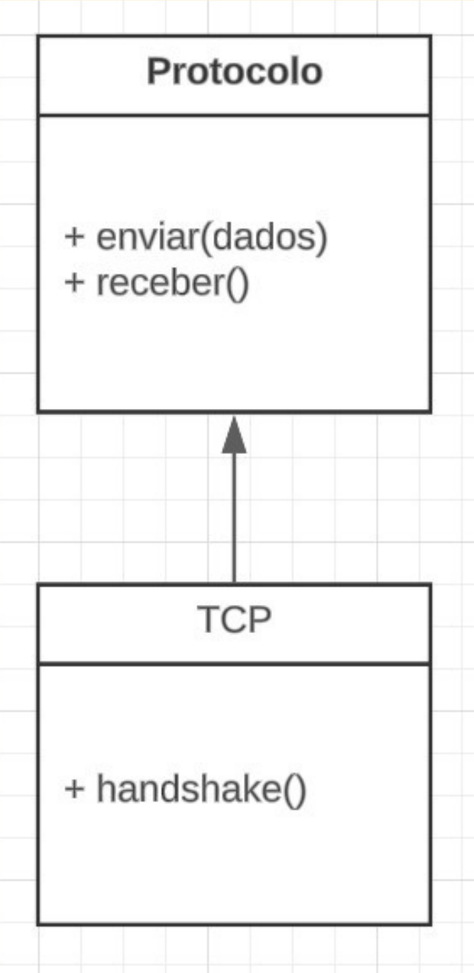
# ATRIBUTOS

- Representam as características de uma classe.
- São apresentados na segunda divisão da classe, contendo duas informações: o nome que identifica o atributo e o tipo de dado que ele armazena.

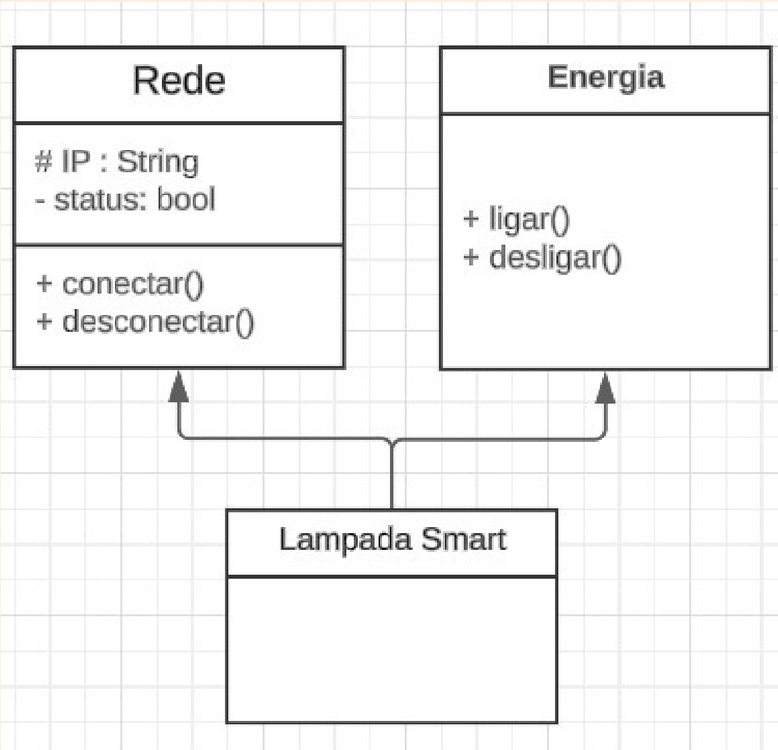
# MÉTODOS

- Representam uma atividade que um objeto de uma classe pode executar.
- Podem, ou não, receber parâmetros e, tende a retornar valores.

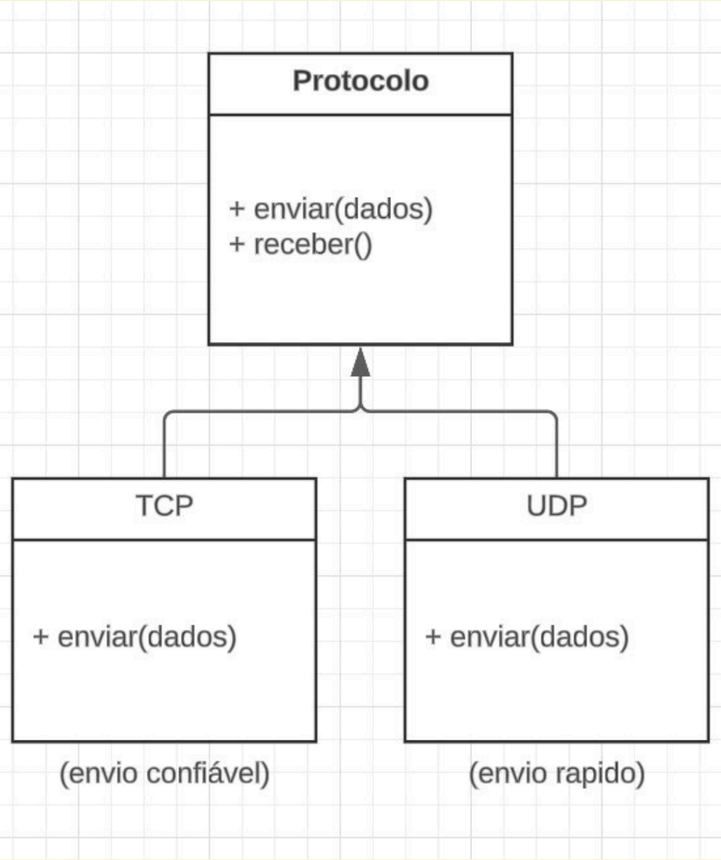
# HERANÇA




# HERANÇA MÚLTIPLA




# POLIMORFISMO



# RELACIONAMENTOS (ASSOCIAÇÕES)

Associações são relações estabelecidas entre classes, possibilitando uma troca de informações e a colaboração entre objetos de uma classe na execução de tarefas.
Navegabilidade: determina se uma classe reconhece outra.

## ASSOCIAÇÃO UNÁRIA OU REFLEXIVA

Relacionamento de um objeto de uma classe com objetos da mesma classe.

## ASSOCIAÇÃO BINÁRIA

Relacionamentos entre objetos de duas classes distintas.

## ASSOCIAÇÃO TERNÁRIA OU N-ÁRIA

Conectam objetos de mais de duas classes.

# RELACIONAMENTOS (ASSOCIAÇÕES)

Associações são relações estabelecidas entre classes, possibilitando uma troca de informações e a colaboração entre objetos de uma classe na execução de tarefas.
Navegabilidade: determina se uma classe reconhece outra.

## AGREGAÇÃO

Tipo de associação em que se complementa informações entre objetos.

## COMPOSIÇÃO

Caracteriza-se por um vínculo mais forte e dependente entre um objeto "todo" e seus objetos "parte".

## GENERALIZAÇÃO/ ESPECIALIZAÇÃO

Representa a herança entre as classes, identificando as classes-mãe, e classes-filhas.

# RELACIONAMENTOS (ASSOCIAÇÕES)

Associações são relações estabelecidas entre classes, possibilitando uma troca de informações e a colaboração entre objetos de uma classe na execução de tarefas.
Navegabilidade: determina se uma classe reconhece outra.

## CLASSE ASSOCIATIVA

São usadas quando atributos da relação entre classes não cabem em nenhuma delas.

## ASSOCIAÇÃO QUALIFICADA

Maneira de identificar individualmente um objeto dentro de uma coleção

## DEPENDÊNCIA

Identifica certo grau de dependência de um elemento em relação à outro.

## REALIZAÇÃO

Realização é uma dependência que herda comportamento, mas não estrutura.

# PORTAS

Uma porta é um ponto de comunicação entre um classificador e seu ambiente ou suas partes internas, representado como um pequeno quadrado na borda da classe.

# INTERFACES

## FORNECIDAS

Descreve serviços de uma classe, representada por um círculo ligado à classe, indicando os serviços que oferece aos clientes.

## REQUERIDAS

Descreve os serviços que outras classes devem fornecer a uma classe. É representada por um semicírculo ligado à classe e requer uma porta de comunicação.

# RESTRIÇÕES

- Definem condições para métodos, associações ou atributos, podendo detalhar requisitos não funcionais.
- Representados por textos limitados com chaves.

# RESTRIÇÕES EM OCL

- Restrição adicionada a esteriótipos;
- Restrições aplicadas em associações;
- Restrições aplicadas a atributos;
- Restrições representando o “ou” exclusivo (xor);
- Restrições aplicadas à coleções;
- Restrições para classes especializadas.

# ESTEREÓTIPOS DO DIAGRAMA DE CLASSES

Utilizados para especializar um determinado componente e atribuir-lhe características extras além daquelas comuns àquele tipo de componente.

## <<ENUMERATION>>

Tipo de dado cujos valores são enumerados no modelo como literais de enumeração.

## PROJETO NAVEGACIONAL

- <<server page>>
- <<client page>>
- <<form>>
- <<link>>
- <<submits>>
- <<builds>>

## <<BOUNDARY>>

Identifica uma classe que serve de comunicação entre os atores externos e o sistema propriamente dito.

# ESTEREÓTIPOS DO DIAGRAMA DE CLASSES

Utilizados para especializar um determinado componente e atribuir-lhe características extras além daquelas comuns àquele tipo de componente.

## <<CONTROL>>

Identifica classes cujos objetos servem de intermédio entre os objetos das classes <<boundary>> e os objetos de classes de entidade.

## <<ENTITY>>

Torna explícito que uma classe contém informações recebidas e armazenadas pelo sistema ou geradas por ele (entidade).

```
class IP():
    def __init__(self, addr:int) -> None:
        self.addr = addr

class Service():
    def __init__(self, scr_port:int, dst_port:int) -> None:
        self._scr_port = scr_port
        self._dsr_port = dst_port
    def Protocolo():
        pass

class Host():
    def __init__(self,_addr, port:int) -> None:
        self._addr = _addr
        self.port = port
    def get_addr(self):
        return self._addr
    def get_addr_IP(self, IP):
        pass

class Protocolo():
    def send(self,data):
        pass
    def receive(self):
        pass
```

```
class UDP(Protocolo):
    def send(self,data):
        pass
    def receive(self):
        pass

class TCP(Protocolo):
    def handshake(self):
        pass
    def open(self):
        pass
    def close(self):
        pass
    def send(self, data):
        pass
    def receive(self):
        pass
```

# AGRADECEMOS A ATENÇÃO!

*Programação Orientada à Objetos*

*Código da Disciplina: TI0196*

Professor: Tarcisio Ferreira Maciel

Antônia Marília de Oliveira Coelho - 568538
Bruno Azevedo Rocha - 565294
Camila Luiza Saldanha de Aquino - 571135
Jose Elenilton da Silveira Junior - 554497
Mateus Barbosa Mesquita - 552487
Ruan Pereira Alves - 569551
Vanessa Gonçalves de Araújo - 556421

REFERÊNCIAS BIBLIOGRÁFICAS:

Guedes, Gilleanes TA. UML 2-Uma abordagem prática. Novatec Editora, 2018.

O que é diagrama UML e como fazer? Veja tipos, modelos e exemplos. Disponível em: <https://miro.com/pt/diagrama/o-que-e-uml/>. Acesso em: 1 dez. 2024.

*Programação Orientada à Objetos*

*Código da Disciplina: TI0196*