Aluno: Ariel Nunes da Silva

Professor: Jhony Luiz de Almeida

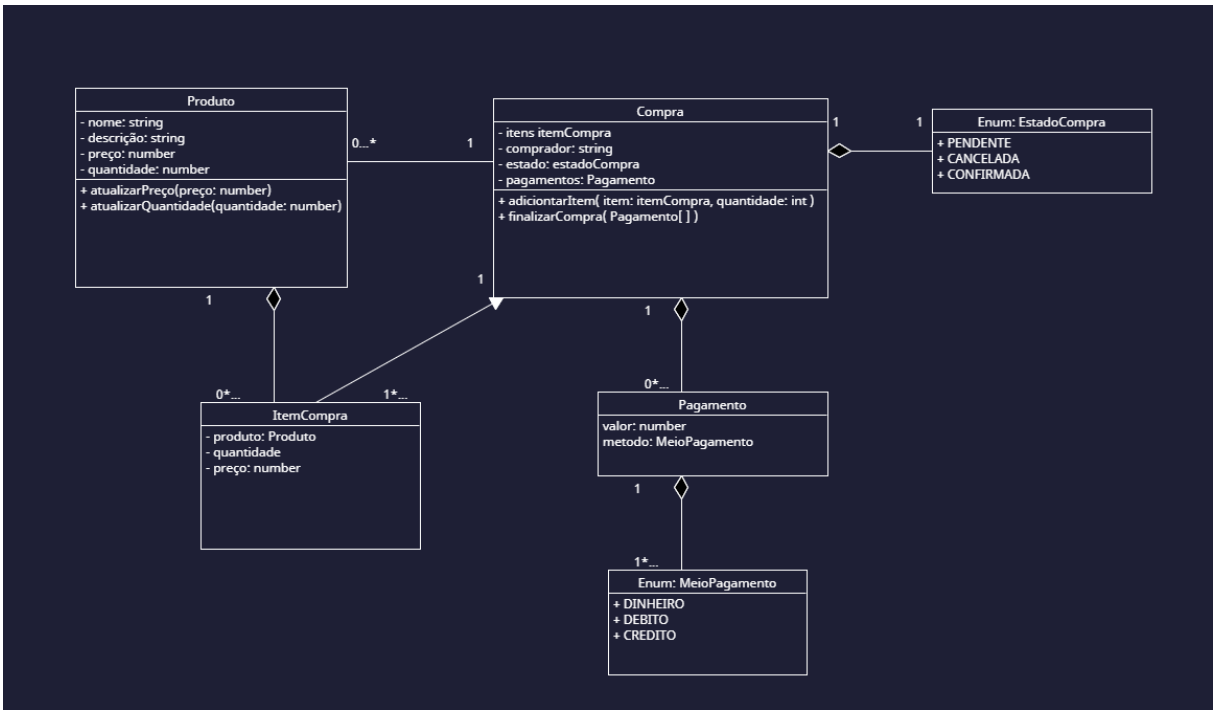
Matéria: Programação Orientada a Objetos

Data: 02/04/2025

https://github.com/ArielNunesS/faculdade

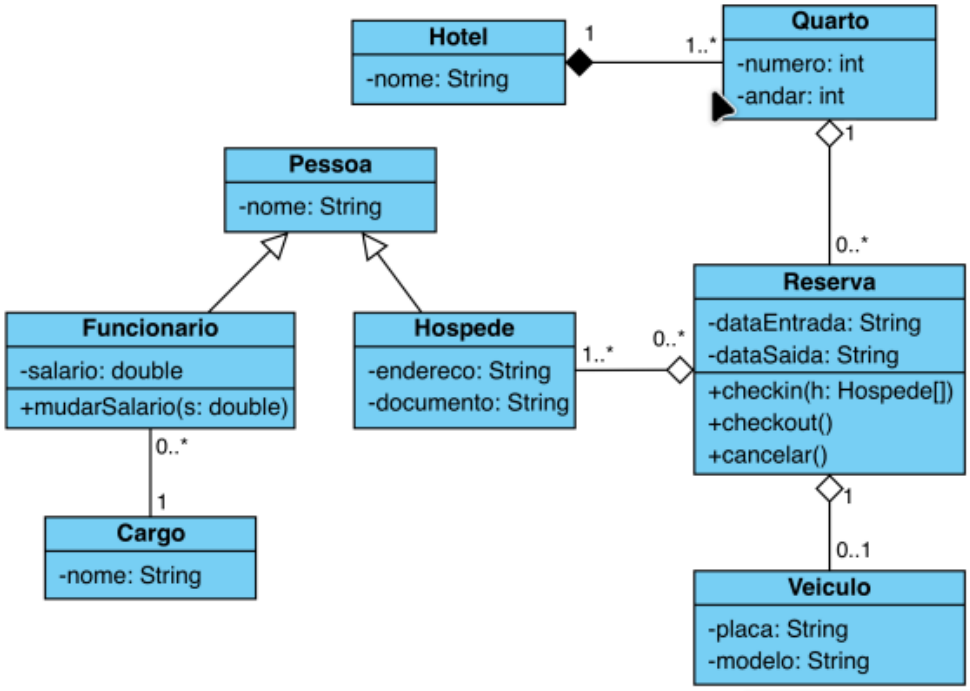
Projeto #1

Criar especificação UML a partir de descrição de requisitos Desenvolva um diagrama de classes na linguagem UML para a seguinte descrição de requisitos de um sistema: Em um sistema de compras online, cada produto é composto ao menos por um nome, uma descrição, um preço atual e uma quantidade em estoque. O nome e descrição do produto não podem ser alterados, mas seu preço e quantidade podem, verificando-se a validade dos valores (não podem ser zero ou negativo). Cada produto possui nenhuma ou várias compras associadas. Uma compra obrigatoriamente possui um ou mais itens, um comprador, um estado (que pode ser pendente, cancelada ou confirmada) e zero ou vários pagamentos. Cada item de compra possui produto, quantidade e preço e só pode estar associado a uma compra. Deve ser possível adicionar um item a uma compra, e finalizar a compra definindo seus pagamentos. Um pagamento possui valor e meio de pagamento (dinheiro, débito ou crédito). Você pode incluir requisitos adicionais que façam sentido, mas não foram citados no enunciado, desde que inclua tudo o que foi solicitado.



Projeto #2

Implementar código a partir de um diagrama de classes Implemente na linguagem escolhida as classes descritas no diagrama de classes abaixo:



CÓDIGO:

```typescript
class Hotel {
    constructor(public nome: string) {}
}

class Quarto {
  constructor(public numero: number, public andar: number) {}
}

class Pessoa {
  constructor(public nome: string) {}
}

class Funcionario extends Pessoa {
  constructor(nome: string, public salario: number) {
    super(nome);
  }

  mudarSalario(novoSalario: number): void {
    this.salario = novoSalario;
  }
}

class Cargo {
  constructor(public nome: string) {}
}

class Hospede extends Pessoa {
  constructor(nome: string, public endereco: string, public documento: string) {
    super(nome);
  }
}

class Veiculo {
  constructor(public placa: string, public modelo: string) {}
}

class Reserva {
  hospedes: Hospede[] = [];
  veiculo?: Veiculo;

  constructor(
    public dataEntrada: string,
    public dataSaida: string
  ) {}

  checkin(hospedes: Hospede[]): void {
    this.hospedes = hospedes;
  }

  checkout(): void {
    this.hospedes = [];
    console.log("Reserva finalizada");
  }

  cancelar(): void {
    this.hospedes = [];
    console.log("Reserva cancelada");
  }
}

```