

## Sistema de Gerenciamento de Hotel

**Objetivo:** Criar um sistema de gerenciamento de hotel que permita gerenciar hóspedes, quartos e reservas.

### Instruções:

#### 1. Diagrama de Classes UML:

- Os alunos devem criar um diagrama de classes UML que inclua as seguintes classes:
  - Hóspede (Guest)
  - Quarto (Room)
  - Reserva (Reservation)
  - Hotel (Hotel)
- Cada classe deve incluir atributos e métodos relevantes.
- O diagrama de classes deve ser feito em mermaid no arquivo *resolucao/diagramas.md*

#### 2. Implementação em Python:

- Os alunos devem traduzir o diagrama de classes UML para código Python.
- Implementar as classes com atributos, métodos e relacionamentos adequados.
- Usar propriedades na declaração dos atributos e type hints na declaração dos métodos não é obrigatório, mas agrega nota.
- Siga as regras e requisitos do sistema abaixo:

### Requisitos do Sistema:

#### Classe Hóspede (Guest):

##### Atributos:

- nome (str)
- id\_hospede (int)
- telefone (str)

##### Métodos:

- `exibir_informacoes()`: Exibir informações do hóspede.

- `fazer_reserva(hotel, quarto, check_in, check_out)`: Realizar uma reserva. Deve verificar a disponibilidade do quarto antes de realizar a reserva.

### Classe Quarto (Room):

#### Atributos:

- `numero` (int)
- `tipo` (str) [Ex: 'Simples', 'Duplo', 'Suite']
- `status` (str) [Disponível, Ocupado]

#### Métodos:

- `disponivel()`: Retornar True se o quarto estiver disponível, False caso contrário.
- `reservar()`: Alterar o status do quarto para 'Ocupado'.

### Classe Reserva (Reservation):

#### Atributos:

- `hospede` (Guest)
- `quarto` (Room)
- `check_in` (str) [Formato 'YYYY-MM-DD']
- `check_out` (str) [Formato 'YYYY-MM-DD']

#### Métodos:

- `confirmar_reserva()`: Confirmar a reserva e alterar o status do quarto para 'Ocupado'.
- `cancelar_reserva()`: Cancelar a reserva e alterar o status do quarto para 'Disponível'.

### Classe Hotel (Hotel):

#### Atributos:

- `lista_hospedes` (list)
- `lista_quartos` (list)

- `lista_reservas` (list)

### Métodos:

- `registrar_hospede(nome, telefone)`: Registrar um novo hóspede no sistema.
- `adicionar_quarto(numero, tipo)`: Adicionar um novo quarto no sistema.
- `fazer_reserva(hospede, quarto, check_in, check_out)`: Realizar uma nova reserva se o quarto estiver disponível.
- `cancelar_reserva(reserva)`: Cancelar uma reserva existente.

### Regras para Implementação:

1. **Verificação de Disponibilidade:** Ao fazer uma reserva, o método `fazer_reserva` da classe `Hotel` deve verificar se o quarto está disponível.
2. **Validação de Datas:** As datas de check-in e check-out devem estar no formato 'YYYY-MM-DD' e o check-out deve ser posterior ao check-in.
3. **Manutenção de Estado:** Ao confirmar uma reserva, o status do quarto deve ser alterado para 'Ocupado'. Ao cancelar uma reserva, o status deve ser alterado para 'Disponível'.
4. **Informações Consistentes:** Métodos que exibem informações (como `exibir_informacoes` na classe `Guest`) devem retornar dados formatados de maneira clara e concisa.
5. **Gestão de Listas:** A classe `Hotel` deve manter listas consistentes de hóspedes, quartos e reservas, garantindo que cada operação de adição ou remoção seja refletida corretamente nas listas correspondentes.