

# **Escopo 1: Sistema de Gerenciamento de Reservas de Quartos de Hotel**

## **Descrição:**

Desenvolver um sistema de gerenciamento de reservas de quartos de hotel que permitirá aos usuários realizar várias operações, como verificar disponibilidade de quartos, fazer reservas, cancelar reservas e exibir informações sobre quartos e reservas. O sistema deve ser baseado em objetos e classes para facilitar a organização e reutilização do código.

## **Classes e seus requisitos:**

1. **Classe Quarto:**
  - 1.1. **Atributos:**
    - 1.1.1. Número do Quarto
    - 1.1.2. Tipo de Quarto (simples, duplo, suíte, etc.)
    - 1.1.3. Preço por Noite
    - 1.1.4. Disponibilidade (indicando se o quarto está ocupado ou disponível).
  - 1.2. **Métodos:**
    - 1.2.1. `exibirInfo()`:
      - 1.2.1.1. Exibe informações detalhadas sobre o usuário
2. **Classe Cliente:**
  - 2.1. **Atributos:**
    - 2.1.1. Nome
    - 2.1.2. ID
    - 2.1.3. E-mail
    - 2.1.4. Reservas (uma lista de objetos de reserva, com limite de reservas por cliente)
  - 2.2. **Métodos:**
    - 2.2.1. `exibirInfo()`:
      - 2.2.1.1. Exibe informações detalhadas sobre o usuário
3. **Classe Reserva:**
  - 3.1. **Atributos:**
    - 3.1.1. Cliente (referência ao cliente que fez a reserva)
    - 3.1.2. Quarto (referência ao quarto reservado)
    - 3.1.3. Data de Check-In
    - 3.1.4. Data de Check-Out
  - 3.2. **Métodos:**
    - 3.2.1. `calcularTotal()`

- 3.2.1.1. Calcula o valor total da reserva com base nas datas de check-in e check-out e no preço do quarto.
- 3.2.2. `exibirInfo()`:
  - 3.2.2.1. Exibe informações detalhadas sobre o livro.

#### **4. Classe Hotel**

##### **4.1. Atributos:**

- 4.1.1. Nome
- 4.1.2. Endereço
- 4.1.3. Lista de Quartos (limite de quartos disponíveis)

##### **4.2. Métodos:**

- 4.2.1. `verificarDisponibilidade()`:
  - 4.2.1.1. Verifica se há pelo menos um quarto disponível para reserva.
- 4.2.2. `fazerReserva()`:
  - 4.2.2.1. Permite aos clientes fazerem uma reserva de quarto, marcando o quarto como ocupado e associando-o ao cliente.
- 4.2.3. `cancelarReserva()`:
  - 4.2.3.1. Permite aos clientes cancelarem uma reserva, marcando o quarto como disponível novamente.
- 4.2.4. `localizarQuartoPorTipo()`:
  - 4.2.4.1. Permite a busca de quartos disponíveis por tipo (simples, duplo, suíte, etc.).
- 4.2.5. `exibirInfo()`:
  - 4.2.5.1. Exibe informações detalhadas sobre o hotel, incluindo a lista de quartos e clientes.

## **Escopo 2: Sistema de Gestão de Produtos**

### **Descrição:**

Desenvolver um sistema de gestão de produtos que permitirá aos usuários realizar várias operações, como cadastrar produtos, atualizar informações, verificar disponibilidade, gerar relatórios e exibir informações detalhadas. O sistema deve ser baseado em objetos e classes para facilitar a organização e reutilização do código.

### **Classes e seus requisitos:**

#### **1. Classe Produto**

##### **1.1. Atributos:**

- 1.1.1. ID do Produto
- 1.1.2. Nome
- 1.1.3. Descrição
- 1.1.4. Preço
- 1.1.5. Quantidade em Estoque

##### **1.2. Métodos:**

- 1.2.1. `exibirInfo()`: Exibe informações detalhadas sobre o produto.

#### **2. Classe Fornecedor**

##### **2.1. Atributos:**

- 2.1.1. Nome
- 2.1.2. CNPJ
- 2.1.3. Lista de Produtos Fornecidos (uma lista de objetos de produto, com limite de produtos por fornecedor)

##### **2.2. Métodos:**

- 2.2.1. `exibirInfo()`: Exibe informações detalhadas sobre o fornecedor.

#### **3. Classe Estoque**

##### **3.1. Atributos:**

- 3.1.1. Lista de Produtos em Estoque (Com limite)

##### **3.2. Métodos:**

- 3.2.1. `verificarDisponibilidade()`: Verifica a disponibilidade de produtos em estoque.
- 3.2.2. `adicionarProduto()`: Permite adicionar um novo produto ao estoque.
- 3.2.3. `atualizarProduto()`: Permite atualizar as informações de um produto existente.
- 3.2.4. `excluirProduto()`: Permite excluir um produto do estoque.
- 3.2.5. `gerarRelatorio()`: Gera um relatório com informações sobre os produtos em estoque.

#### 4. **Classe Gestor Produtos**

##### 4.1. **Atributos:**

- 4.1.1. Lista de Fornecedores (limite de fornecedores)

##### 4.2. **Métodos:**

- 4.2.1. cadastrarFornecedor(): Permite cadastrar um novo fornecedor.
- 4.2.2. buscarProdutoPorNome(): Permite a busca de produtos por nome.
- 4.2.3. buscarProdutoPorID(): Permite a busca de produtos por ID.
- 4.2.4. exibirInfoFornecedor(): Exibe informações detalhadas sobre o fornecedor.
- 4.2.5. exibirInfoProduto(): Exibe informações detalhadas sobre o produto.

## **Escopo 3: Sistema de Reservas de Voos**

### **Descrição:**

Desenvolver um sistema de reservas de voos que permitirá aos usuários realizar várias operações, como pesquisar voos, fazer reservas, cancelar reservas e exibir informações detalhadas sobre voos e reservas. O sistema deve ser baseado em objetos e classes para facilitar a organização e reutilização do código.

### **Classes e seus requisitos:**

#### **1. Classe Voo**

##### **1.1. Atributos:**

- 1.1.1. Número do Voo
- 1.1.2. Origem
- 1.1.3. Destino
- 1.1.4. Data e Hora de Partida
- 1.1.5. Data e Hora de Chegada
- 1.1.6. Capacidade de Passageiros
- 1.1.7. Assentos Reservados (indicando quantos assentos já foram reservados)

##### **1.2. Métodos:**

- 1.2.1. `exibirInfo()`: Exibe informações detalhadas sobre o Voo.

#### **2. Classe Passageiro**

##### **2.1. Atributos:**

- 2.1.1. Nome
- 2.1.2. ID
- 2.1.3. E-mail
- 2.1.4. Reservas (uma lista de objetos de reserva, com limite de reservas por passageiro)

##### **2.2. Métodos:**

- 2.2.1. `exibirInfo()`: Exibe informações detalhadas sobre o passageiro.

#### **3. Classe Reserva**

##### **3.1. Atributos:**

- 3.1.1. Passageiro (referência ao passageiro que fez a reserva)
- 3.1.2. Voo (referência ao voo reservado)
- 3.1.3. Data da Reserva
- 3.1.4. Assentos Reservados (indicando quantos assentos foram reservados nesta reserva)

##### **3.2. Métodos:**

- 3.2.1. `exibirInfo()`: Exibe informações detalhadas sobre a reserva..

#### **4. Classe Aeroporto**

##### **4.1. Atributos:**

- 4.1.1. Nome
- 4.1.2. Localização
- 4.1.3. Lista de Reservas (Com limite)

##### **4.2. Métodos:**

- 4.2.1. buscarVoos(): Permite aos usuários pesquisar voos disponíveis com base no número
- 4.2.2. fazerReserva(): Permite aos passageiros fazerem uma reserva de voo.
- 4.2.3. cancelarReserva(): Permite aos passageiros cancelarem uma reserva.
- 4.2.4. exibirInfo(): Exibe informações detalhadas sobre o aeroporto, incluindo a lista de voos disponíveis.

## **Escopo 4: Sistema de Agendamento de Eventos**

### **Descrição:**

Desenvolver um sistema de agendamento de eventos que permitirá aos usuários realizar várias operações, como criar eventos, verificar disponibilidade de datas, fazer reservas de espaços para eventos e exibir informações detalhadas sobre eventos e reservas. O sistema deve ser baseado em objetos e classes para facilitar a organização e reutilização do código.

### **Classes e seus requisitos:**

#### **1. Classe Evento**

##### **1.1. Atributos:**

- 1.1.1. ID do Evento
- 1.1.2. Nome do Evento
- 1.1.3. Data e Hora do Evento
- 1.1.4. Local do Evento

##### **1.2. Métodos:**

- 1.2.1. `exibirInfo()`: Exibe informações detalhadas sobre o evento.

#### **2. Classe Organizador**

##### **2.1. Atributos:**

- 2.1.1. Nome
- 2.1.2. ID
- 2.1.3. E-mail
- 2.1.4. Eventos Organizados (uma lista de objetos de evento, com limite de eventos por organizador)

##### **2.2. Métodos:**

- 2.2.1. `exibirInfo()`: Exibe informações detalhadas sobre o organizador

#### **3. Classe ReservaEspaco**

##### **3.1. Atributos:**

- 3.1.1. Organizador (referência ao organizador que fez a reserva)
- 3.1.2. Evento (referência ao evento reservado)
- 3.1.3. Data da Reserva
- 3.1.4. Espaço Reservado (indicando o local reservado para o evento)

##### **3.2. Métodos:**

- 3.2.1. `exibirInfo()`: Exibe informações detalhadas sobre a reserva de espaço.

#### **4. Classe Calendario**

##### **4.1. Atributos:**

- 4.1.1. Lista de Eventos Agendados (limite de eventos agendados)

#### **4.2. Métodos:**

- 4.2.1. verificarDisponibilidade(): Verifica a disponibilidade de datas para agendar eventos.
- 4.2.2. inserirEvento(): Permite aos organizadores inserirem os eventos.
- 4.2.3. cancelarEvento(): Permite aos organizadores cancelarem um evento..
- 4.2.4. exibirInfo(): Exibe informações detalhadas sobre o calendário, incluindo a lista de eventos agendados.



## **Escopo 5: Sistema de Rede Social Simplificada**

### **Descrição:**

Desenvolver um sistema de rede social simplificada que permitirá aos usuários realizar várias operações, como criar perfis de usuário, adicionar amigos, compartilhar mensagens e fotos, visualizar o feed de notícias e exibir informações detalhadas sobre os perfis e conteúdo. O sistema deve ser baseado em objetos e classes para facilitar a organização e reutilização do código.

### **Classes e seus requisitos:**

#### **1. Classe Usuario**

##### **1.1. Atributos:**

- 1.1.1. ID do Usuário
- 1.1.2. Nome de Usuário
- 1.1.3. Nome Real
- 1.1.4. Data de Nascimento
- 1.1.5. Lista de Amigos (limite de amigos)
- 1.1.6. Lista de Mensagens (limite de mensagens)
- 1.1.7. Lista de Fotos (limite de fotos)

##### **1.2. Métodos:**

- 1.2.1. `exibirInfo()`: Exibe informações detalhadas sobre o usuário.
- 1.2.2. `adicionarAmigo()`: Permite aos usuários adicionar outros usuários como amigos.
- 1.2.3. `compartilharMensagem()`: Permite aos usuários compartilharem mensagens com seus amigos.
- 1.2.4. `compartilharFoto()`: Permite aos usuários compartilharem fotos com seus amigos.
- 1.2.5. `verFeedNoticias()`: Exibe o feed de notícias do usuário, mostrando mensagens e fotos compartilhadas por amigos.

#### **2. Classe Mensagem**

##### **2.1. Atributos:**

- 2.1.1. Autor (referência ao usuário que escreveu a mensagem)
- 2.1.2. Conteúdo da Mensagem
- 2.1.3. Data e Hora de Publicação

##### **2.2. Métodos:**

- 2.2.1. `exibirInfo()`: Exibe informações detalhadas sobre a mensagem.

#### **3. Classe Foto**

##### **3.1. Atributos:**

- 3.1.1. Autor (referência ao usuário que compartilhou a foto)
- 3.1.2. Descrição da Foto

3.1.3. Data e Hora de Upload

3.1.4. URL da Imagem

**3.2. Métodos:**

3.2.1. `exibirInfo()`: Exibe informações detalhadas sobre a foto.

**4. Classe RedeSocial**

**4.1. Atributos:**

4.1.1. Lista de Usuários (limite de usuários)

**4.2. Métodos:**

4.2.1. `inserirUsuario()`: Permite inserir um usuário na Rede social.

4.2.2. `buscarUsuarioPorNome()`: Permite a busca de usuários por nome (username).

4.2.3. `visualizarFeedNoticias()`: Exibe o feed de notícias dos usuários.