**Pontificia Universidade Católica de São Paulo**

**Faculdade de Ciências Exatas e Tecnologia**

**Ciências da Computação**

**GRUPO:JLM**

**João Alfredo – RA00338430**

**Leonardo Miguel dos Santos – RA00353295**

**Matheus Guido Noronha Moreira Passos - RA00352533**

**PROJETO HOTEL LIC2024**

**Controle de Reservas e Ocupação de Hotel**

**1) OBJETIVOS DO PROJETO**

Desenvolvemos um programa em C para controlar a ocupação e reservas dos apartamentos do hotel. Deve ser mantido o status de cada apartamento e as opções de check-in/check-out e a apresentação da ocupação.

**2) Manual do Operador**

**Considerações Gerais:**

**1. Limitação de Caracteres:**

- Devido a limitações da linguagem e IDE, o usuário não deve utilizar acentos ou cedilha na digitação

- O usuário não deve separar o DDD do Número do hóspede com parênteses “()” ou espaço “ “

**Layout das Telas:**

**1. Menu Principal:**

- Apresenta opções numeradas para o operador escolher as operações desejadas, como criar uma reserva, realizar check-in, exibir informações de hóspedes, etc.

**2. Check-in:**

- Solicita ao operador que insira o andar e número do quarto desejado, em seguida, solicita informações do hóspede como nome, telefone, email, etc. Confirma o Check-in.

**3. Check-out:**

- Solicita ao operador que insira o andar e número do quarto desejado, em seguida, conclui a estadia do hóspede, liberando o quarto e atualizando os registros. Confirma o Check-out.

**4. Realizar Reserva:**

- Solicita ao operador que insira o andar e número do quarto desejado, em seguida, solicita informações do hóspede como nome, telefone, email, etc. Confirma a Reserva.

**5. Cancelar Reserva:**

- Solicita ao operador que insira o andar e número do quarto desejado, em seguida, remove a reserva do hóspede, liberando o quarto e atualizando os registros. Confirma o Cancelamento da Reserva.

**6. Listagem de Hóspedes**

- Mostra ao operador os dados de hóspedes presentes ou reservas existentes,

**3) Como Proceder em Cada Situação Prevista que Ocorre na Recepção do Hotel**

**1. Realizar Check-in ou Reserva:**

- Selecione a opção correspondente no menu.

- Insira os dados solicitados.

- Confirme o check-in ou a reserva e receba a confirmação.

2. **Realizar Check-out ou Cancelar Reserva:**

- Selecione a opção correspondente no menu.

- Confirme o check-out ou cancelamento da reserva e receba a confirmação.

3. **Listagem de Hóspedes**

- Selecione a opção de listagem no menu.

- Insira os dados solicitados.

- Confirme os dados do hóspede ou da reserva.

**4. Sair**

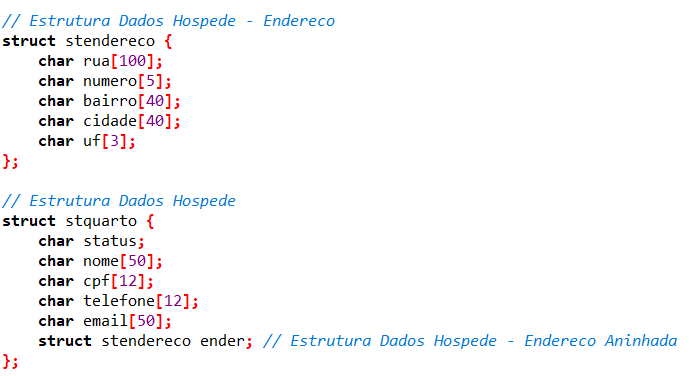
- Selecione a opção de sair no menu para encerrar o programa.

**5. Mostrar Mapa e Ocupação**

- Seleciona a opção de mostrar mapa e ocupação para mostrar o mapa do hotel e as porcentagens de ocupação, reserva e vagas.

**4) Apêndice Técnico**

**Estruturas de Dados Usadas**



### **Considerações Adicionais**

**1. Checagem para caracteres inválidos**:

- No registro de nomes, para não conter números, registro de quartos e andar para conter apenas números de 1 a 14, registro de email para ser necessário a presença do símbolo arroba (@), etc.

**2. Registro de dados na reserva:**

- Permite o registro de todos os dados no usuário tanto no momento da reserva como no check-in.

### **5) Conclusão**

Como o projeto consiste em um único arquivo main.c, toda a lógica e funcionalidades estão contidas nesse arquivo, organizadas principalmente em funções que são chamadas a partir da função main(). Desse modo trouxemos algumas ideias de melhorias e aprimoramentos possíveis para futuras versões desse projeto:

### **1. Modularização do Código:**

- Dividir as funcionalidades relacionadas ao hotel, como cadastro de hóspedes, reservas, gestão de quartos, em diferentes módulos (hotel.c, reservas.c, quartos.c) com seus respectivos cabeçalhos (hotel.h, reservas.h, quartos.h).

### **2. Melhoria nas Estruturas de Dados:**

**-** Utilizar mais estruturas de dados adequadas para representar hóspedes, quartos, reservas, etc., de maneira que facilite operações como busca, atualização e remoção de informações.

**- Melhor gerenciamento** e a alocação de memória, de forma eficiente para evitar vazamentos de memória e problemas de fragmentação.

### **3. Melhoria na Interface com o Usuário:**

**-** Aprimorar o menu de opções para o usuário, tornando-o mais intuitivo e fácil de navegar.

- Implementar mensagens de feedback mais claras para o usuário, indicando sucesso, erro ou instruções adicionais quando necessário.

### **4. Tratamento de Erros e Validação de Entrada:**

**-** Implementar validações robustas para garantir que os dados inseridos pelo usuário sejam corretos e consistentes.

**-** Adicionar tratamento de erros adequado para lidar com situações inesperadas, como falta de memória ou falhas na leitura/gravação de arquivos.