

Sistema de Reservas de Hotel (Projeto N2)

Disciplina Programação Estruturada e Modular

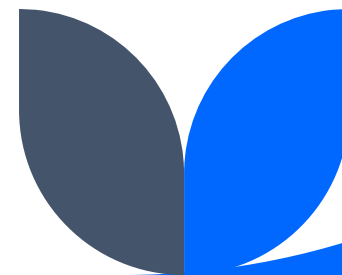
Cassiel Okada
Enzo Santos
Matheus Emanuel
Matheus Galvão

Contextualização

Disciplina: Concentra-se na organização lógica do código e na divisão eficiente de sistemas complexos. Com o foco de utilizar estruturas de controle claras para garantir um fluxo previsível e quebrar o programa em módulos.

Objetivo do projeto: desenvolver um programa na linguagem C para gerenciar de forma eficiente utilizando o que foi aprendido em sala de aula.

Importância do tema: Criar um programa para automatizar pequenos hotéis.



Problema proposto e solução

Problema: Lugares como pousadas ou hotéis de pequeno e médio porte usando reservas manuais/papel, causando erros, duplicidade e lentidão

Requisitos: Cadastro de quartos, hospedes. Realizar e cancelar reservas. Listar reservas ativas, histórico de reservas e verificar disponibilidade de quartos por período

Solução: Um sistema em C para centralizar hóspedes, quartos e reservas, garantindo integridade e agilidade



Arquitetura e Modularização

- **Módulo de Quartos:** Gerenciamento de cadastro, preços e status.
- **Módulo de Hóspedes:** Validação de CPF e histórico de reservas.
- **Módulo de Reservas:** O "coração" do sistema. Valida datas e disponibilidade.
- **Módulo de Persistência:** Gravação e leitura de arquivos binários
- **Main (Interface):** Menu principal que integra todas as funções

Implementação Técnica (Destques do Código)

- **Alocação Dinâmica de Memória:**

- Uso de realloc e vetores dinâmicos em vez de matrizes fixas.
- Permite cadastrar hóspedes e reservas ilimitados (enquanto houver memória RAM).

- **Structs e Modelagem:**

- Dados organizados em estruturas: Quarto, Hospede, Reserva.

- **Manipulação de Strings:**

- Uso de strcmp para buscas (ex: validar login ou achar CPF)

Desafios e Lógica de Datas

- **O Problema das Datas:**

- Como calcular diárias e evitar conflitos (sobreposição) de reservas?.

- **A Solução Lógica:**

- Conversão de string ("DD/MM/AAAA") para números inteiros.
- Cálculo de "Dias Absolutos" para saber a diferença exata entre check-in e check-out.

- **Validação de Conflitos:**

- O sistema impede que um quarto seja reservado se já existir outra reserva ativa naquele intervalo de datas.

Testes Realizados (Casos de Uso)

- CT1: Cadastro de Quarto:** Validação de número único e gravação no arquivo.
- CT2: Reserva com Sucesso:** Cálculo automático do valor total e atualização do status para "Ocupado".
- CT3: Bloqueio de Conflito:** O sistema recusou corretamente uma reserva em data sobreposta.



Conclusão

Objetivos Alcançados:

- Sistema funcional, seguro e atendendo aos requisitos da disciplina.

Clean Code:

- Código indentado, comentado e com nomes de variáveis intuitivos.

Aprendizados:

- Domínio sobre ponteiros e memória dinâmica.
- Importância da modularização para trabalho em equipe.
- Lógica robusta para validação de regras de negócio (datas e status)