



UNIVERSIDADE FEDERAL DE LAVRAS

Projeto final IALG

Marcos Morais - RA: 202410496

Lavras

2024

Sumário

1	Introdução	2
2	Estruturas Utilizadas	2
3	Ordem dos Dados Armazenados	3
4	Acertos e Erros Durante o Desenvolvimento	4
5	Conclusão	5

1. INTRODUÇÃO

Este trabalho prático tem como objetivo demonstrar a aplicação de ferramentas da qualidade, com foco em estruturas de dados dinâmicas e a interação com a biblioteca `ncurses`. O projeto busca desenvolver uma solução que permita listar e filtrar dados relacionados a acomodações e hosts, utilizando técnicas de alocação dinâmica e interação com o usuário por meio de uma interface de linha de comando, sem a necessidade de inserção manual de dados, evitando a entrada de valores incoerentes.

2. ESTRUTURAS UTILIZADAS

O programa utiliza várias estruturas de dados fundamentais para organizar e manipular as informações. As principais estruturas utilizadas incluem:

- **Arrays Dinâmicos:** Para armazenar e gerenciar de forma eficiente os dados das acomodações e hosts, sem a necessidade de utilizar a biblioteca `vector`. Isso permite que o tamanho do array seja ajustado dinamicamente, de acordo com os dados inseridos.
- **Estruturas de Controle:** O fluxo do programa é baseado em comandos de entrada e saída por meio de `cin` e `cout`, e a interface de interação com o usuário é realizada utilizando a biblioteca `ncurses`.
- **Arquivos de Dados:** O programa armazena os dados em arquivos, organizando-os de forma eficiente para garantir a correta manipulação e recuperação das informações.
- **Estrutura acomodacoes:** A estrutura `acomodacoes` contém diversas variáveis para armazenar informações sobre as acomodações, como `id`, `name`, `host_id`, `price`, entre outras. A estrutura é projetada para armazenar dados como inteiros, floats e strings, com limites de tamanho fixos para alguns campos, como o nome do host e o tipo de propriedade.

A lógica do programa envolve a leitura dos dados de um arquivo, armazenamento das informações em arrays dinâmicos e, por fim, a exibição interativa dos dados para o

usuário. O processo de seleção é automatizado, permitindo ao usuário filtrar os dados sem precisar inserir valores manualmente, evitando assim a entrada de dados incoerentes.

3. ORDEM DOS DADOS ARMAZENADOS

Os dados são armazenados em um arquivo binário, onde cada registro contém as informações de uma acomodação. A estrutura `acomodacoes` é organizada da seguinte forma:

- **id** (long long): Um identificador único da acomodação.
- **name** (char[100]): O nome da acomodação (máximo de 100 caracteres).
- **host_id** (long long): O identificador único do host.
- **host_name** (char[100]): O nome do host (máximo de 100 caracteres).
- **host_since** (int): O ano em que o host iniciou suas atividades.
- **host_response_time** (char[50]): O tempo de resposta do host.
- **host_response_rate** (float): A taxa de resposta do host.
- **host_is_superhost** (char): Indica se o host é superhost (caractere 'f' ou 't').
- **neighbourhood_cleansed** (char[100]): O bairro da acomodação (máximo de 100 caracteres).
- **neighbourhood_group_cleansed** (char[100]): O grupo de bairros ao qual a acomodação pertence.
- **latitude** (float) e **longitude** (float): A localização geográfica da acomodação.
- **property_type** (char[100]): O tipo de propriedade.
- **room_type** (char[100]): O tipo de quarto.
- **accommodates** (int): A quantidade de pessoas que a acomodação pode acomodar.
- **bathrooms** (int): O número de banheiros da acomodação.
- **bedrooms** (int): O número de quartos da acomodação.

- **beds** (int): O número de camas.
- **price** (float): O preço por noite da acomodação.
- **minimum_nights** (int): O número mínimo de noites para a reserva.
- **availability_365** (int): A disponibilidade da acomodação durante o ano.
- **number_of_reviews** (int): O número de avaliações da acomodação.
- **review_scores_rating** (float): A média das avaliações de usuários.
- **license** (char[50]): O número de licença da acomodação.
- **instant_bookable** (char): Indica se a acomodação está disponível para reserva instantânea (caractere 'f' ou 't').

Esses dados são gravados de forma sequencial no arquivo, e a estrutura permite um fácil acesso e manipulação para fins de filtragem e exibição.

4. ACERTOS E ERROS DURANTE O DESENVOLVIMENTO

Durante o desenvolvimento do trabalho, foram encontrados e corrigidos alguns erros e acertos importantes:

- **Acertos:**
 - A implementação da alocação dinâmica funcionou corretamente, permitindo ajustar o tamanho dos arrays conforme a necessidade.
 - A integração com a biblioteca **ncurses** foi bem-sucedida, proporcionando uma interface de interação fluida.
 - A automatização da seleção de dados eliminou a necessidade de inserção manual, prevenindo a entrada de dados incoerentes.
- **Erros:**
 - Houve dificuldades iniciais na manipulação de arquivos binários, o que exigiu ajustes na estrutura de leitura e escrita dos dados.

- A implementação da filtragem de dados teve um desempenho insatisfatório inicialmente, necessitando de otimizações na alocação de memória.

Esses desafios foram superados por meio de ajustes na lógica do programa e refatoração do código.

5. CONCLUSÃO

O trabalho permitiu a aplicação prática de conceitos de qualidade e estruturas de dados, com destaque para a alocação dinâmica e o uso da biblioteca `ncurses` para interação com o usuário. Os resultados obtidos demonstram que a solução atende aos requisitos, possibilitando a visualização e filtragem dos dados de forma eficiente. A automatização da seleção de dados foi um ponto chave, garantindo que o usuário não precisasse inserir valores manualmente e evitando entradas incoerentes. Embora tenha sido necessário corrigir alguns erros durante o desenvolvimento, o aprendizado e a aplicação das ferramentas da qualidade foram altamente positivos, resultando em uma solução funcional e bem-sucedida.