**FATEC SÃO CAETANO DO SUL – ANTONIO RUSSO**

**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS – AMS**

**São Caetano do Sul**

**2024**

**FATEC ANTONIO RUSSO**

Camille Almeida

Felipe Lins

Murillo Grillo Bastos

Rafael Galindo

Thiago Gerbi

Vitor Tamais Fischer

**Gerenciamento de Reservas em Restaurante**

Algoritmo em linguagem C

São Caetano do Sul

2024

**RESUMO**

O objetivo do Projeto de Gerenciamento de Reservas em Restaurantes é criar uma plataforma prática e fácil de entender para controlar, armazenar e administrar reservas em determinado restaurante para facilitar a agenda e os pedidos do estabelecimento. A ferramenta foi desenvolvida para auxiliar os usuários a realizarem reservas rapidamente, o que facilita a organização e permite o acesso rápido a informações importantes. Para facilitar o gerenciamento de reservas, o sistema organiza a lista em ordem de chegada por cliente ao adicionar uma nova reserva e permite exibir o painel de reservas para o administrador.

**Palavras-chave**: Reservas. Gerenciamento. Restaurantes. Sistema.

**LISTA DE FIGURAS**

Figura 1: Código struct 9

Figura 2: Lista encadeada 10

Figura 3: Adicionar 10

Figura 4: Remover 11

Figura 5: Exibir 11

Figura 6: Consultar 12

Figura 7: Relatórios 12

Figura 8: Validação data 13

Figura 9: Validação hora 13

Figura 10: Plano de Testes 15

Figura 11: Testes de Reserva .................................................................................................................16

Figura 12: Exibição de Reserva ..............................................................................................................16

Figura 13: Remoção de Reserva .. 17

Figura 14: Teste de nomes iguais ..........................................................................................................18

Figura 15: Teste de consulta específica ..................................................................................................19

Figura 16: Teste de relatório de ocupação .............................................................................................20

Figura 17: Teste de Capacidade .............................................................................................................20

**LISTA DE TABELAS**

Tabela 1: Tabela Estrutura .....................................................................................................................9

Tabela 2: Tabela GitHub ........................................................................................................................14

**SUMÁRIO**

**1.** **CAPA**  **1**

**2.** **CAPA DE ROSTO** **2**

**3.** **RESUMO** **3**

**4.** **LISTA DE FIGURAS** **4**

**5.** **LISTA DE TABELAS** **4**

**6.** **SUMÁRIO** **5**

**6.** **INTRODUÇÃO** **6**

**6.** **CONTEXTO (PROBLEMA)** **7**

**6.** **REVISÃO BIBLIOGRÁFICA** **8**

**6.** **IMPLEMENTAÇÃO** **9**

**6.** **TESTES DE SOFTWARE**  **15**

**6.** **CONCLUSÃO** **21**

1. **INTRODUÇÃO**

O gerenciamento eficiente de reservas em restaurantes é essencial para garantir uma experiência satisfatória aos clientes, minimizando o tempo de espera e maximizando a utilização dos recursos disponíveis. Com o objetivo de otimizar esse processo, desenvolveu-se um algoritmo em linguagem C capaz de gerenciar essas reservas de um modo prático e fácil.

Este algoritmo foi projetado para permitir a realização de diversas operações, tais como a adição de novas reservas, cancelamento de reservas existentes, consulta de disponibilidade em determinados horários e dias da semana, além de fornecer relatórios sobre a ocupação do restaurante em intervalos específicos de tempo.

A implementação deste algoritmo busca oferecer uma solução eficaz para os desafios enfrentados pelos estabelecimentos no controle de suas reservas, contribuindo para a melhoria da gestão operacional, consequentemente, para a satisfação dos clientes e para aumentar o lucro do estabelecimento.

Dessa forma, este trabalho apresenta uma contribuição significativa para a área de gestão de restaurantes, fornecendo uma ferramenta prática e eficiente para o gerenciamento de reservas, que pode ser facilmente adaptada e integrada aos sistemas existentes.

Serão utilizadas as seguintes estruturas aprendidas no semestre:

* Tipos básicos de dados;
* Ponteiro;
* Listas Lineares;
* Struct;
* Estruturas de Pilhas;
* Listas Ordenadas;

**2. CONTEXTO (PROBLEMA)**

Sabe-se que, diante de toda a evolução da tecnologia ao longo do tempo, ainda não existe um sistema de gerenciamento de reservas tão eficaz quanto a demanda do mercado exige.

Pense na seguinte situação: Um funcionário tem mais de 40 reservas e pedidos por dia, nisso ele não dispõe de um sistema capaz o suficiente para dar conta de “lembrar” de todas as reservas e precisa ficar anotando em um papel ou bloco de notas. Isso faz com que o funcionário perca muito tempo fazendo as anotações, ou seja, o tempo que poderia estar sendo aproveitado para outras funções está sendo desperdiçado e o estabelecimento poderá não obter o lucro total, em caso de esquecimento de alguma reserva ou conflitos com reservas em horários e dias iguais, entre outras inúmeras complicações.

Pensando nisso, foi desenvolvido um sistema considerando as principais críticas de funcionários, clientes, gerentes e proprietários de restaurantes, para que fosse possível suprir as suas necessidades, garantindo assim, qualidade e agilidade no processo de adicionar, remover e consultar reservas já feitas pelos clientes durante o dia em um determinado restaurante, evitando falhas de comunicação e obtendo um bom resultado no panorama geral.

Nosso sistema permite a visualização em tempo real das reservas feitas pelos clientes e com comandos fáceis a fim de ajudar nas ações de remover, adicionar, gerar relatórios de acordo com a data e horário para administradores e consulta em horários específicos prevenindo que mais clientes além do limite permitido tenham feito reservas.

**3. REVISÃO BIBLIOGRÁFICA**

Para a concluir o sistema, utilizamos conhecimentos estudados durante o semestre nas aulas de Estrutura de Dados pelo professor Carlos Henrique Veríssimo.

Durante as aulas do primeiro semestre, aprendemos diversos assuntos em estrutura de dados, alguns exemplos são:

* Ordenação de dados;
* Ponteiros;
* Filas;
* Pilhas;
* Listas ordenadas;
* Listas ligadas;
* Arrays;
* Struct, entre outros.

Esses temas foram fundamentais para o funcionamento pleno do algoritmo de reservas.

Como exemplos reais em nosso código utiliza-se:

* Nó - lista de reservas;
* Filas - exibição das reservas;
* Pilha – consulta de reservas;
* Listas – agrupamento das reservas;
* Struct – para a alocação da memória e das reservas;

Os tópicos mencionados serão explicados nos testes de software e na implementação juntamente com mais exemplos.

**4. IMPLEMENTAÇÃO**

Nesta seção, vamos explorar os principais aspectos da implementação do sistema de reservas para um restaurante, destacando as estruturas de dados utilizadas, as operações realizadas e a organização do código.

**4.1 Estruturas de Dados e Structs:**

Estruturas de Dados:

A principal estrutura de dados utilizada foi a lista encadeada, implementada na forma de uma lista simplesmente encadeada. Cada nó da lista representa uma reserva e contém informações como o nome do cliente, o número de pessoas, a data e o horário da reserva. A lista é dinamicamente alocada e manipulada por meio de ponteiros.

No código, foi definida uma estrutura de dados para representar uma reserva, denominada Reserva. Esta estrutura contém os seguintes campos:

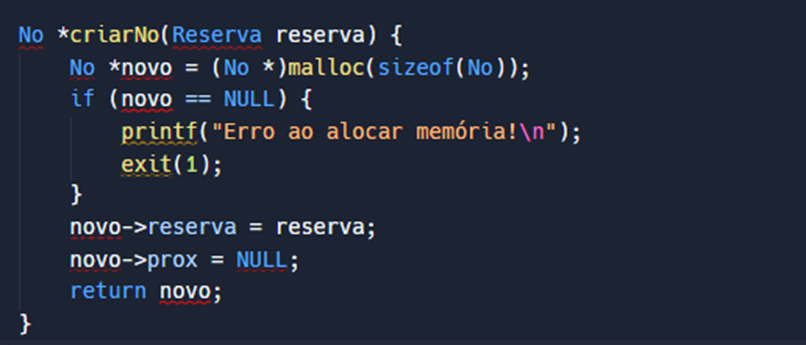
|  |  |
| --- | --- |
| nome | armazena o nome do cliente que fez a reserva. |
| num\_pessoas | indica o número de pessoas para a reserva. |
| data | armazena a data da reserva no formato AAAA-MM-DD. |
| horario | representa o horário da reserva no formato HH. |

***Tabela 1: estrutura***



***Figura 1: Código Struct***

Além disso, a implementação utiliza uma lista encadeada para armazenar as reservas. Cada nó da lista contém uma reserva e um ponteiro para o próximo nó na lista.

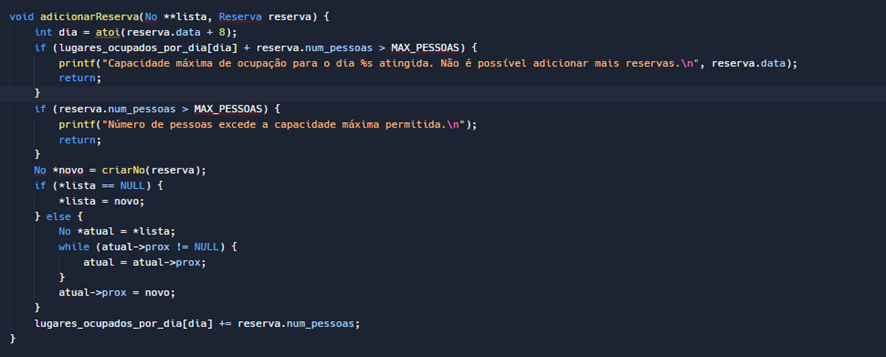


***Figura 2: Lista encadeada***

**4.2 Operações de Reserva:**

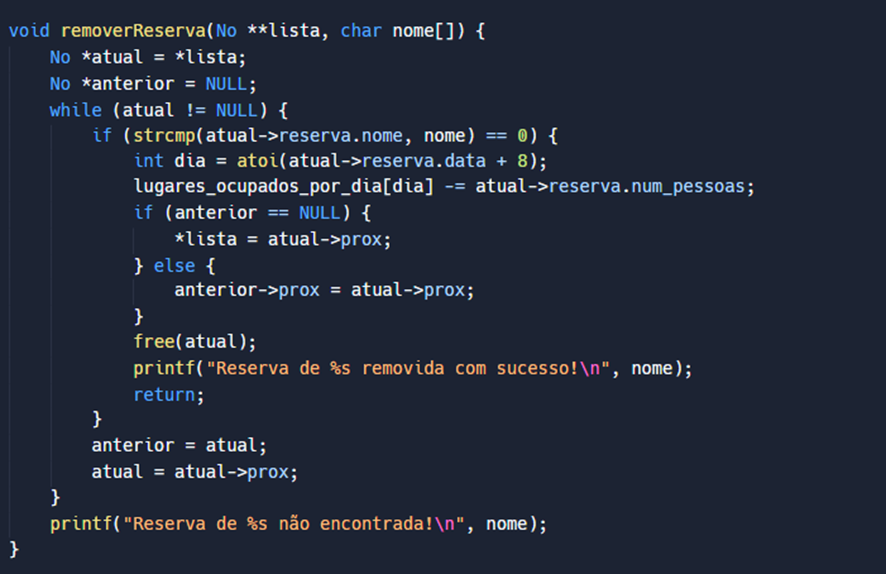
Foram implementadas diversas operações para manipular as reservas no sistema, tais como:

**Adicionar uma nova reserva:**



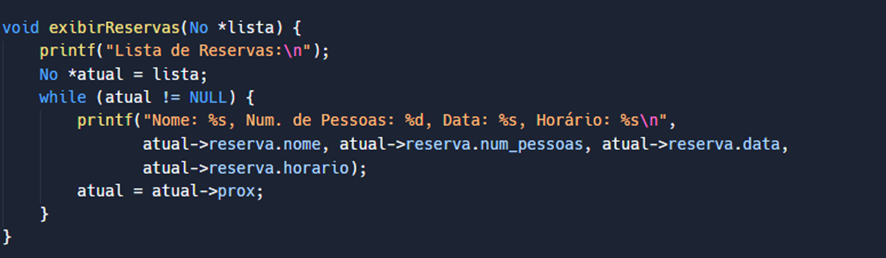
***Figura 3: adicionar***

**Remover uma reserva existente com base no nome do cliente:**



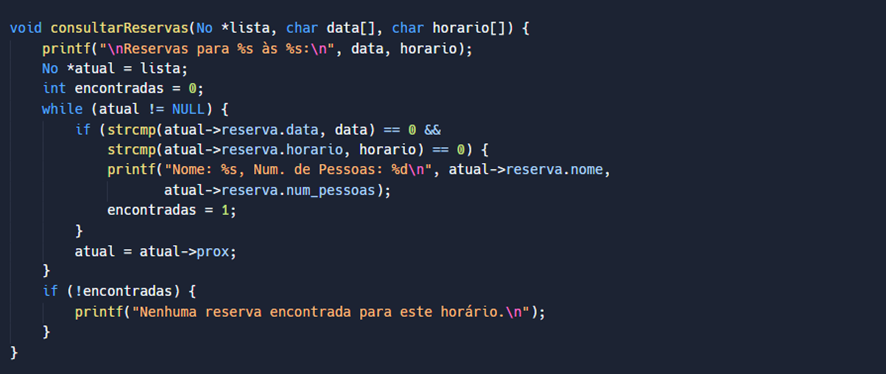
***Figura 4: remover***

**Exibir todas as reservas.**



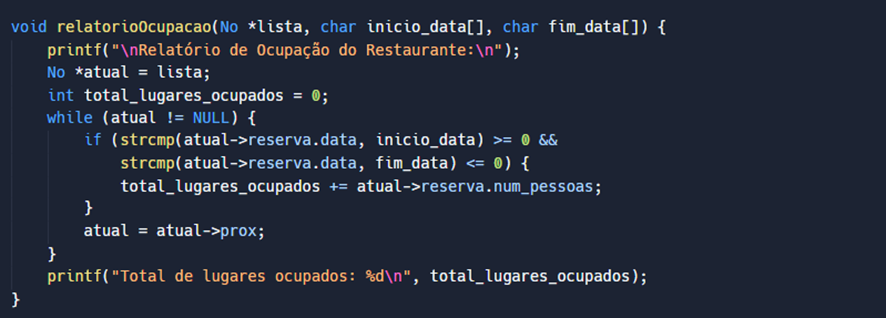
***Figura 5: exibir***

**Consultar as reservas em um horário específico.**



**Figura 6: consultar**

**Gerar um relatório de ocupação para um intervalo de datas:**



***Figura 7: relatórios***

Essas funções encapsulam as operações relacionadas às reservas, proporcionando uma organização mais modular e facilitando a manutenção do código.

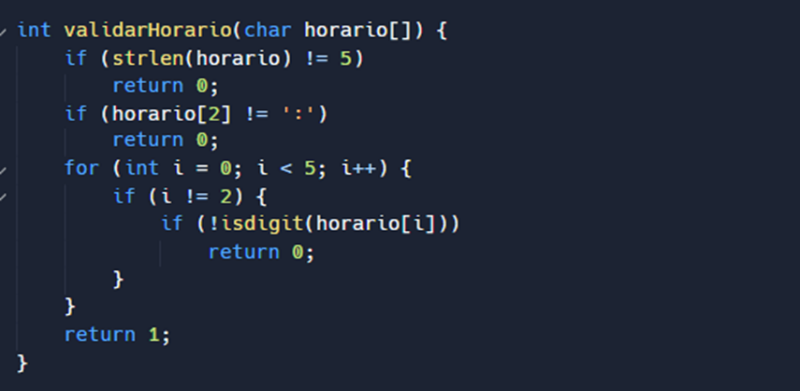
**4.3 Validação de Entrada de Dados:**

* **Validação de Data**



***Figura 8: validação data***

* **Validação de Hora**



***Figura 9: validação hora***

Essas operações foram implementadas como funções separadas no arquivo reserva.c, seguindo uma abordagem modular para melhor organização do código.

**Modularização do Código:**

O código foi organizado de forma modular, com as funções relacionadas a reservas encapsuladas no arquivo reserva.c. Além disso, foram definidos protótipos de função no arquivo reserva”H”, permitindo que as funções sejam acessadas por outros arquivos fonte do programa, como o arquivo main.c.

**Boas Práticas de Codificação:**

Durante a implementação, foram adotadas algumas boas práticas de codificação para garantir a legibilidade e manutenibilidade do código. Isso inclui o uso de nomes descritivos para variáveis e funções, comentários explicativos para documentar o código e a divisão do código em blocos lógicos coesos.

**Gerenciamento de Reservas para Restaurante**

|  |  |
| --- | --- |
| Camille Almeida | <https://github.com/CamilleGFAlmeida> |
| Felipe Lins | <https://github.com/lipe-slins> |
| Murillo Bastos | <https://github.com/MuriloBastosAMSptech> |
| Rafael Galindo | <https://github.com/Rafael-Galindo> |
| Thiago Gerbi | <https://github.com/Gerbimv> |
| Vitor Tamais | <https://github.com/VitorTamais> |

***Tabela 2: Tabela GitHub***

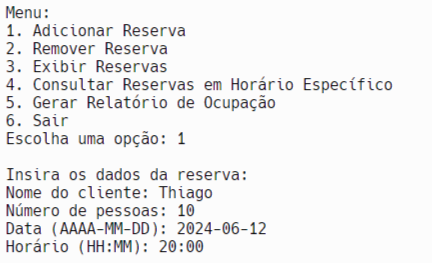
**5. TESTES DE SOFTWARE**

Para a realização de qualquer sistema de qualidade, o software passa por alguns processos de testes, no qual é feita uma verificação das etapas, para observar se será necessária alguma modificação. O sistema de reservas passou por etapas de testes para a validação do sistema, como o funcionamento efetivo de adição, remoção e visualização de reservas, além de relatórios de ocupação.

Para organizar os testes e realizar anotações nas quais são passadas para a equipe de desenvolvedores posteriormente, é realizado uma planilha de testes, na qual contêm todos os testes a serem realizados.

***Figura 10: plano de testes***

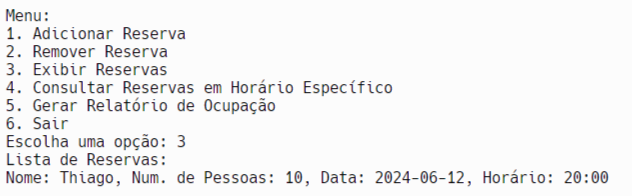
Os primeiros testes realizados foram de adição de reservas no restaurante, onde o funcionário entra com alguns dados como: nome do responsável pela reserva, quantidade de pessoas, dia e hora da reserva



***Figura 11: Teste de Reserva***

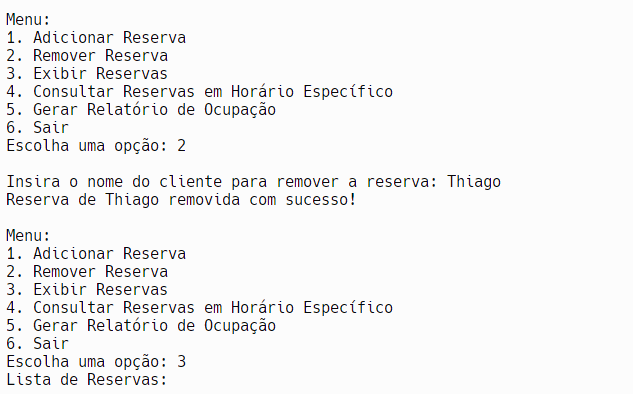
Ao realizar o primeiro teste, foram observados alguns problemas, como a forma na qual deve se inserir a data (yyyy-MM-dd), onde foi solicitado para o desenvolvedor modificar para a seguinte forma (dd-MM-yyyy).

O segundo teste realizado foi para verificar se a reserva realizada estava sendo exibida de forma correta, com o nome do responsável, a quantidade de pessoas, o dia e a hora da reserva.

***Figura 12: teste de exibição de reservas***

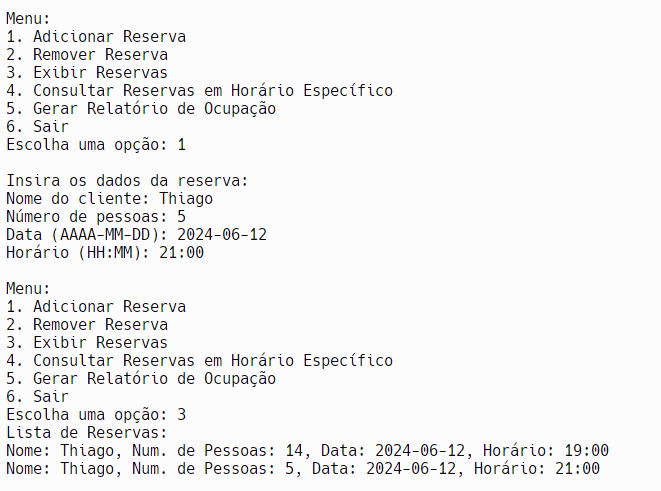
Ao realizar o teste de exibição não foi encontrado nenhum erro, apenas a recomendação de mudança do formato da data. A exibição de reserva ocorreu de forma correta.

O terceiro teste realizado foi o de remoção de reservas, onde o funcionário para remover a reserva insere o nome do responsável pela reserva, e assim faz a sua remoção.

***Figura 13: remoção de reserva***

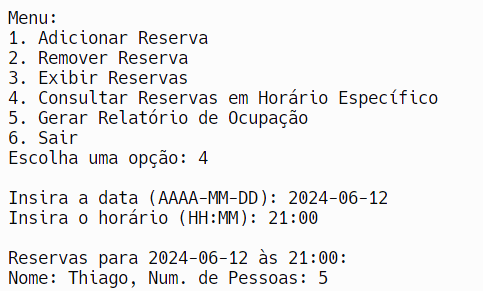
A remoção ocorreu de forma efetiva, removendo a reserva de determinado cliente.

O quarto teste foi o de realização de reservas com nomes iguais, onde o resultado esperado seria a solicitação de sobrenome pelo sistema, evitando assim conflitos na hora de reservas

***Figura 14: teste de nomes iguais***

Ao realizar o teste, foi detectado um problema na realização de reservas, uma vez que não é solicitado o sobrenome do responsável, permitindo assim com que haja conflitos nas reservas, se tornando muito difícil a visualização e a remoção de reservas. Foi solicitado que o desenvolvedor implemente um campo para a inserção do sobrenome na hora de realizar uma reserva

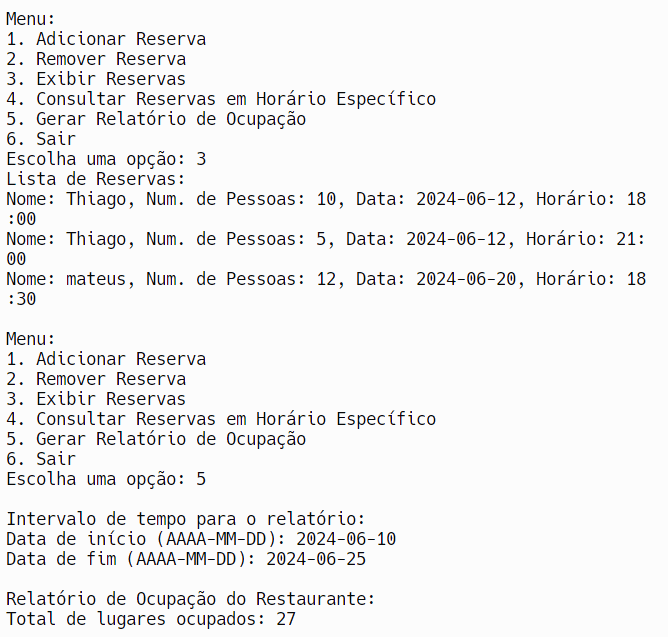
O quinto teste se trata de uma consulta de uma reserva em um horário específico, onde o funcionário irá inserir um dia e um horário e ele terá a visualização de todas as reservas daquele dia e horário



***Figura 15: teste de consulta específica***

Com a realização do teste, foi possível observar que a pesquisa ocorreu de forma esperada, onde apenas foi feita a recomendação para a mudança do formato de data.

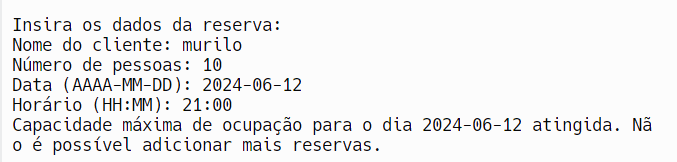
O sexto teste realizado foi o de relatórios de ocupação, onde o funcionário irá inserir a data início e a data final, e nesse meio tempo, será possível observar um relatório com a quantidade total de pessoas que foram ou que irão ao restaurante a partir de uma reserva.



***Figura 16: teste de relatório de ocupação***

No teste de ocupação, o resultado foi positivo, mostrando o total de lugares ocupados nesse prazo de tempo, porém foi recomendado a alteração do formato de datas.

No sétimo e último teste, foi realizado um teste para ver se era possível realizar mais reservas do que o permitido. Onde definimos um total de 20 lugares para a ocupação. Caso queira realizar uma reserva sem lugares disponíveis, o sistema irá barrar e avisar que não há lugares disponíveis



***Figura 17: teste de capacidade***

O teste foi bem-sucedido, uma vez que o sistema barrou a realização da reserva a partir do momento que a capacidade máxima de ocupação foi atingida.

**6. CONCLUSÃO**

Conclui-se que, os estudos e conteúdos compreendidos nas aulas da disciplina Estrutura de Dados, foram essenciais para a finalização do projeto utilizando a linguagem de programação C para atender os critérios de um algoritmo completo de gerenciamento de reservas para restaurantes, sendo ele, eficiente, prático, fácil e intuitivo.

Foram realizados testes para o controle da qualidade e funcionamento do algoritmo, registrados em Excel com cenários específicos para cada funcionalidade, verificando assim quais seriam os possíveis “pontos fracos” do projeto e prevenindo falhas posteriores.

Compreendemos que com este algoritmo conseguimos aplicar todos os conhecimentos em um único programa. De tal modo, a equipe aperfeiçoou suas práticas e o projeto apresentou um resultado satisfatório em relação aos requisitos propostos pelo professor Carlos Veríssimo.

Em resumo, a adoção de um sistema de gerenciamento de reservas de restaurantes, conforme as diretrizes estabelecidas nesta norma, representa um avanço significativo na gestão operacional e no atendimento ao cliente. A padronização dos processos, a automatização das tarefas e a melhoria na comunicação entre os restaurantes e seus clientes proporcionam benefícios tangíveis, como a redução de erros, a otimização do tempo e a maximização da satisfação dos consumidores. A implementação eficaz desse sistema contribui para a criação de um ambiente mais organizado e eficiente, fortalecendo a reputação do estabelecimento e promovendo a fidelização dos clientes.