FACULDADE DE TECNOLOGIA ARTHUR DE AZEVEDO

CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Documentação de software:

reunion

Gabriel Danilo do Nascimento e Silva

LUCAS VIEIRA DA SILVA

Marcelo felipe belotto

Vinícius Emanuel da Silva

MOGI MIRIM

2024

RESUMO

Este documento apresenta o desenvolvimento e a implementação de um sistema de gerenciamento de reservas de salas de reunião, o ReuniON, projetado para otimizar o agendamento e o uso de salas em ambientes corporativos. O sistema foi estruturado de forma modular, com camadas distintas para bibliotecas e implementação, facilitando a manutenção e a escalabilidade. A estratégia de testes incluiu testes unitários, de integração e de sistema, assegurando a confiabilidade e a robustez do software. Durante o desenvolvimento, foram identificadas limitações, como a interface de linha de comando e desafios no controle de concorrência, que indicam áreas para melhorias futuras. O sistema alcançou os objetivos de eficiência e organização, com resultados positivos na gestão de reservas. Futuras expansões podem incluir o desenvolvimento de uma interface gráfica e a integração com calendários externos. As lições aprendidas destacam a importância da modularidade e dos testes contínuos.

Palavras-chave: gerenciamento de reservas; salas de reunião; sistema modular; testes de software; interface de usuário; controle de concorrência; escalabilidade.

SUMÁRIO

[1 introdução 4](#_Toc183161160)

[1.1 TEMA 4](#_Toc183161161)

[1.1.1 Delimitação do Problema 4](#_Toc183161162)

[1.1.2 Escolha do tema 4](#_Toc183161163)

[1.2 OBJetivos 4](#_Toc183161164)

[1.2.1 Objetivos Específicos 5](#_Toc183161165)

[1.3 modelo de Trabalho 5](#_Toc183161166)

[1.4 Estrutura do documento 5](#_Toc183161167)

[2 DESCRIÇÃO GERAL DO sISTEMA 6](#_Toc183161168)

[2.1 principais envolvidos e suas características 6](#_Toc183161169)

[2.1.1 Usuários do Sistema 6](#_Toc183161170)

[2.1.2 Desenvolvedores do Sistema 6](#_Toc183161171)

[2.1.3 Regras de Negócio 6](#_Toc183161172)

[3 requisitos do sistema 7](#_Toc183161173)

[3.1 backlog do produto 7](#_Toc183161174)

[3.2 CASOS DE USO 9](#_Toc183161175)

[3.3 Requisitos funcionais 10](#_Toc183161176)

[3.3.1 RF01 – Reservar Sala 10](#_Toc183161177)

[3.3.2 RF02 – Alterar Reserva 10](#_Toc183161178)

[3.3.3 RF03 – Cancelar Reserva 11](#_Toc183161179)

[3.3.4 RF04 – Verificar Disponibilidade de Salas 12](#_Toc183161180)

[3.3.5 RF05 – Cadastrar Salas 12](#_Toc183161181)

[3.3.6 RF06 – Cadastrar Usuário 13](#_Toc183161182)

[3.3.7 RF07 – Alterar Salas 14](#_Toc183161183)

[3.3.8 RF08 – Listar Usuários 14](#_Toc183161184)

[3.3.9 RF09 – Alterar Usuário 15](#_Toc183161185)

[3.3.10 RF10 – Alterar Senha 16](#_Toc183161186)

[3.3.11 RF11 – Exibir Reservas 16](#_Toc183161187)

[3.3.12 RF12 – Listar Reservas 17](#_Toc183161188)

[3.4 Requisitos não-funcionais 17](#_Toc183161189)

[3.4.1 Desempenho 17](#_Toc183161190)

[3.4.2 Segurança 17](#_Toc183161191)

[3.4.3 Usabilidade 18](#_Toc183161192)

[3.4.4 Confiabilidade 18](#_Toc183161193)

[3.4.5 Escalabilidade 18](#_Toc183161194)

[3.4.6 Compatibilidade 18](#_Toc183161195)

[3.4.7 Manutenibilidade: 18](#_Toc183161196)

[4 arquitetura do sistema 19](#_Toc183161197)

[4.1 Visão Geral da Arquitetura 19](#_Toc183161198)

[4.2 Componentes do Sistema 20](#_Toc183161199)

[4.3 Tecnologias Utilizadas 20](#_Toc183161200)

[5 IMPLEMENTAÇÃO DO SISTEMA 21](#_Toc183161201)

[5.1 ESTRUra do código 21](#_Toc183161202)

[5.2 Principais Algoritmos e Funções 21](#_Toc183161203)

[5.3 Padrões de Codificação 22](#_Toc183161204)

[5.4 Ferramentas de Desenvolvimento 22](#_Toc183161205)

[6 Testes e Validação 23](#_Toc183161206)

[6.1 Estratégia de Testes 23](#_Toc183161207)

[6.2 Casos de Teste 23](#_Toc183161208)

[6.3 Ferramentas de Teste 24](#_Toc183161209)

[6.4 Resultados dos Testes 24](#_Toc183161210)

[7 Conclusão 26](#_Toc183161211)

[7.1 Resultados Alcançados 26](#_Toc183161212)

[7.2 Limitações e Desafios 26](#_Toc183161213)

[7.3 Trabalhos Futuros 26](#_Toc183161214)

[7.4 Lições Aprendidas 27](#_Toc183161215)

[REFERÊNCIAS 28](#_Toc183161216)

# introdução

O presente documento tem como objetivo apresentar o projeto de desenvolvimento do ReuniON, um sistema de gerenciamento de reservas de salas de reunião, que visa otimizar o agendamento e o uso de salas de reunião em uma empresa ou organização, facilitando o arranjo e o uso eficiente dos recursos internos.

## TEMA

O tema do projeto é o desenvolvimento de um sistema de gerenciamento de reservas de salas de reunião, que busca atender às necessidades de agendamento e utilização eficaz das salas de reunião dentro de uma organização.

### Delimitação do Problema

O problema central que o projeto visa resolver é o gerenciamento das reservas de salas de reunião. Muitas organizações enfrentam dificuldades em organizar e otimizar o uso de suas salas de reunião, resultando em conflitos de agendamento e uso ineficiente dos espaços disponíveis.

### Escolha do tema

A escolha do tema se deu pela percepção de que o desenvolvimento de um sistema seria desafiador, tanto do ponto de vista técnico quanto organizacional. Além disso, a implementação de tal sistema pode trazer benefícios significativos para as organizações, melhorando a eficiência e a produtividade.

## OBJetivos

O objetivo principal do projeto é otimizar o agendamento e o uso de salas de reunião, proporcionando uma ferramenta que facilite a organização e o uso eficiente dos recursos internos.

### Objetivos Específicos

Os objetivos específicos incluem a criação de um sistema intuitivo e acessível para os usuários, que permita o agendamento rápido e eficiente das salas, além de fornecer informações sobre a utilização dos espaços.

## modelo de Trabalho

Para a realização do projeto, será utilizado o modelo tradicional de desenvolvimento de software, conhecido como modelo cascata. Este método prevê uma abordagem sequencial, onde cada fase do desenvolvimento deve ser concluída antes do início da próxima.

## Estrutura do documento

O documento abordará desde a introdução e descrição geral do sistema até a implementação, testes e apresentação do software.

# DESCRIÇÃO GERAL DO sISTEMA

O ReuniON visa melhorar o agendamento e a utilização de salas em empresas e organizações. A falta de um sistema eficaz resulta em desorganização, uso ineficiente dos recursos e conflitos de agenda. A solução é um sistema intuitivo que permite agendamentos rápidos e fornece informações sobre o uso dos espaços.

## principais envolvidos e suas características

### Usuários do Sistema

Existem dois tipos de usuários: o usuário geral, que pode gerenciar suas próprias reservas, e o administrador, que além dessas funções, pode cadastrar e gerenciar salas e visualizar todas as reservas. O sistema é adequado para qualquer organização com salas de reunião.

### Desenvolvedores do Sistema

O desenvolvimento está a cargo de Gabriel Danilo do Nascimento e Silva, Lucas Vieira da Silva, Marcelo Felipe Belotto e Vinicius Emanuel da Silva.

### Regras de Negócio

As regras de negócio incluem:

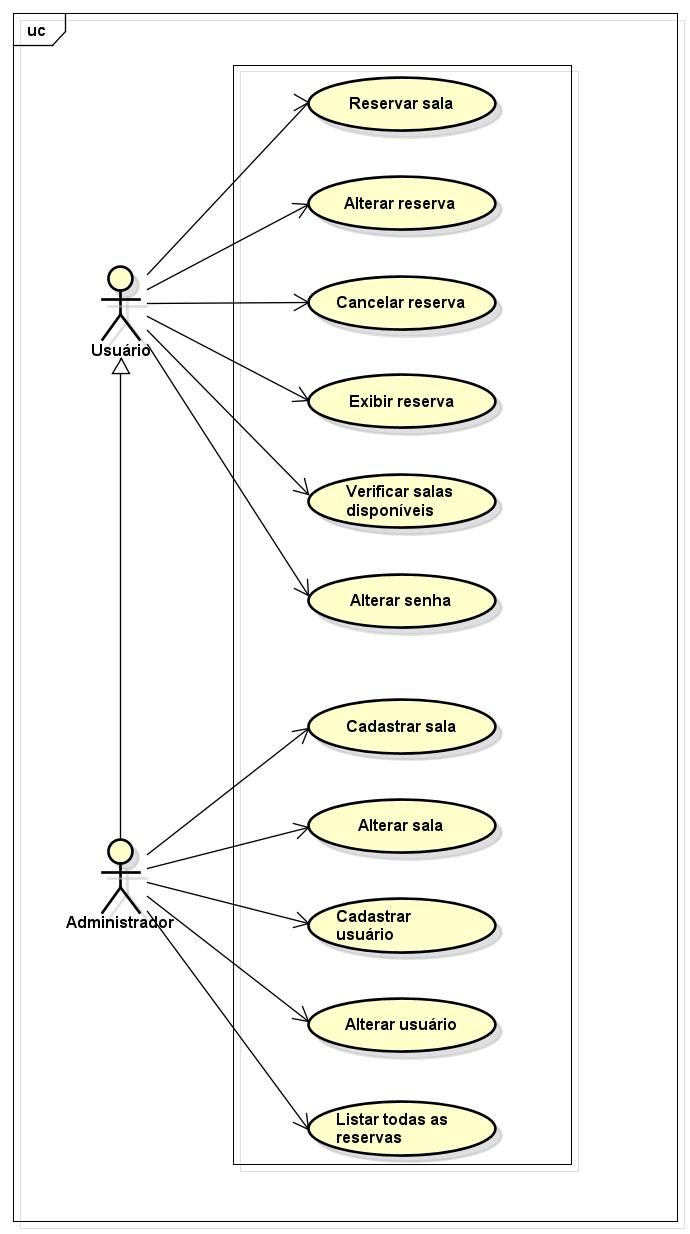
* Acesso: Apenas usuários autenticados podem fazer reservas, com privilégios adicionais para administradores.
* Desempenho: O sistema deve responder rapidamente às solicitações.
* Confiabilidade: Deve lidar com falhas de rede ou servidor, garantindo a integridade dos dados.
* Escalabilidade: Capacidade de armazenar um grande volume de dados, acompanhando o crescimento de usuários e salas.
* Ferramentas: Uso de tecnologias que suportem a escalabilidade e manutenção do sistema.

# requisitos do sistema

## backlog do produto

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nome do Aplicativo** | | | | |
| **ReuniON** | | | | |
| **Visão do Produto** | | | | |
| Para empresas que desejam otimizar o uso de suas salas de reunião e evitar conflitos de agendamento, o **ReuniON** é um sistema de controle de reservas que oferece, de forma prática e eficiente, a possibilidade de reservar e gerenciar salas conforme a necessidade, permitindo controle de horários e permissões de acesso. Diferentemente de outras soluções convencionais, o **ReuniON** tem uma interface de linha de comando simples e eficaz, permitindo consultas em tempo real, proporcionando maior organização e eficiência no uso dos espaços corporativos. | | | | |
| **Backlog** | | | | |
| **id** | **Prioridade** | **Título** | **Descrição** | **Detalhamento** |
| *RF01* | *100* | Reservar Sala | Como USUÁRIO, quero reservar uma sala específica, garantindo que ela esteja disponível para minha reunião. | Para efetuar a reserva, o usuário deve apontar: - A sala escolhida, dentre as exibidas pelo sistema; - Data e hora do início da reserva; - Data e hora do fim da reserva;  O sistema deverá fazer a validação da disponibilidade com os parâmetros inseridos. |
| *RF02* | *95* | Alterar Reserva | Como USUÁRIO, quero alterar uma reserva já efetuada. | O usuário deverá apontar: - Qual reserva deseja modificar dentre todas as suas próprias reservas; - Os dados a serem alterados (início e fim da reserva, tipo de sala)  O sistema deverá: - Exibir uma lista com todas as reservas feitas utilizando o id do usuário em questão; - Fazer a validação da disponibilidade com os novos parâmetros inseridos. |
| *RF03* | *90* | Cancelar Reservas | Como USUÁRIO, quero cancelar uma reserva efetuada. | O usuário deverá apontar qual reserva deseja cancelar dentre todas as suas próprias reservas.  O sistema deverá exibir uma lista com todas as reservas feitas utilizando o id do usuário em questão |
| *RF04* | *85* | Verificar Disponibilidade de Salas | Como USUÁRIO, quero verificar a disponibilidade de uma sala para saber se o horário desejado está ocupado antes de fazer a reserva. | O usuário deverá apontar: - O tipo de sala que deseja verificar; - A data que deseja verificar;  O sistema deverá exibir uma lista com todas as salas do tipo escolhido, mostrando todos horários ***indisponíveis*** |
| *RF05* | *80* | Cadastrar Salas | Como ADMINISTRADOR, quero cadastrar novas salas no sistema | o administrador deverá apontar: - O número da sala - O tipo da sala (PA, PC, MA, MC ou GA)\*\* - O status da sala (Ativa, Em manutenção, Inativa)  O sistema deverá validar se o número de sala já não foi utilizado |
| *RF06* | *75* | Cadastrar Usuário | Como ADMINISTRADOR, quero cadastrar novos usuários no sistema | o administrador deverá apontar: - CPF - Nome do usuário - Perfil (usuário geral ou administrador) - Senha - O status do usuário (Ativo, Inativo)  O sistema deverá: - Validar o CPF - Validar se o documento já não foi utilizado para outro usuário - Gerar um id sequencial e automático, com uma sequência de 5 dígitos, sendo o id 00001 reservado para o perfil "Master" |
| *RF07* | *70* | Alterar Salas | Como ADMINSITRADOR, quero alterar uma sala já cadastrada. | O administrador deverá apontar: - Qual sala deseja modificar dentre todas as exibidas pelo sistema; - Os dados a serem alterados (número, tipo de sala e status)  O sistema deverá: - Exibir uma lista com todas as salas cadastradas. - Em caso de mudança de status, deverá verificar se não há nenhum agendamento programado. |
| *RF08* | *65* | Listar usuários | Como ADMINISTRADOR, quero listar todos os usuários cadastrados no sistema | O sistema deverá exibir uma lista de todos os usários cadastros, com os dados completos para cada um. |
| *RF09* | *60* | Alterar Usuário | Como ADMINISTRADOR, quero alterar um usuário cadastrado | O administrador deverá apontar: - Qual usuário deseja modificar dentre todas os exibidos pelo sistema; - Os dados a serem alterados (CPF, nome, perfil, senha e status)  O sistema deverá: - Exibir uma lista com todas as salas cadastradas. - Em caso de mudança de status, deverá verificar se não há nenhum agendamento programado. |
| *RF10* | *55* | Alterar senha | Como USUÁRIO, desejo alterar minha senha de acesso. | O usuário deverá apontar a senha atual e a nova senha.  O sistema deverá validar a senha atual. |
| *RF11* | *50* | Exibir Reserva | Como USUÁRIO, desejo ver todas as reservas efetuadas utilizando meu id e senha. | O sistema exibirá uma lista com todas as reservas efetuadas pelo usuário em questão, exibindo o status da reserva. |
| *RF12* | *45* | Listar Reservas | Como ADMINISTRADOR, quero listar todas as reservas efetuadas no período estipulado para o tipo de sala específico | O administrador deverá apontar o tipo de sala e o período que deseja consultar. |
|  |  |  |  |  |
| \* O perfil ADMINISTRADOR pode executar todas as funções atribuídas ao perfil USUÁRIO.  \*\* Legenda: PA - Sala P (até 5 lugares), com recursos audiovisuais MA - Sala M (até 15 lugares), com recursos audiovisuais GA - Auditório (até 100 lugares), com recursos audiovisuais PC - Sala P (até 5 lugares), sem recursos audiovisuais MC - Sala M (até 15 lugares), sem recursos audiovisuais | | | | |

## CASOS DE USO



## Requisitos funcionais

### RF01 – Reservar Sala

O sistema deve permitir que o usuário reserve uma sala, solicitando o número da sala, data e hora de início e fim da reserva.

|  |  |
| --- | --- |
| **FLUXO DE EVENTOS NORMAL** | |
| 1 | O sistema exibe o menu inicial. |
| 2 | O usuário insere seu ID e senha para autenticação. |
| 3 | O sistema exibe o menu de opções do usuário. |
| 4 | O usuário seleciona a opção "Reservar Sala". |
| 5 | O sistema exibe uma lista de salas disponíveis. |
| 6 | O usuário escolhe a sala desejada e informa a data e hora de início e fim da reserva. |
| 7 | O sistema valida a disponibilidade da sala com os parâmetros inseridos. |
| 8 | O sistema registra a reserva no banco de dados. |
| 9 | O sistema exibe uma mensagem de confirmação da reserva. |
| 10 | O sistema retorna ao menu de opções do usuário. |
| **FLUXO DE EVENTOS EXCEÇÃO** | |
| ***2a - Erro de autenticação:***  2a.1 - O sistema exibe uma mensagem de erro informando que as credenciais são inválidas. 2a.2 - O sistema bloqueia o acesso temporariamente e retorna ao menu inicial. | |
| ***3a - Cancelamento da operação pelo usuário:***  3a.1 - O usuário opta por cancelar a operação de reserva.  3a.2 - O sistema retorna ao menu inicial. | |
| ***7a - Nenhuma sala disponível no horário desejado:***  7a.1 - O sistema exibe uma mensagem informando que não há salas disponíveis para o período solicitado. 7a.2 - O sistema oferece ao usuário a opção de ajustar a data ou horário, retornando ao passo 6 do fluxo normal.  ***7b - Dados inválidos para a reserva:***  7b.1 - O sistema detecta que a data ou horário de início/fim da reserva está incorreto (ex: data/hora de fim anterior à data/hora de início).  7b.2 - O sistema exibe uma mensagem de erro detalhando as informações inválidas. 7b.3 - O sistema retorna ao passo 7 do fluxo normal para que o usuário corrija as informações. | |
| ***8a - Falha no sistema ao registrar a reserva:***  8a.1 - O sistema exibe uma mensagem de erro informando a falha ao registrar a reserva. 8a.2 - O sistema orienta o usuário a tentar novamente, retornando ao passo 5 do fluxo normal. | |

### RF02 – Alterar Reserva

O sistema deve permitir que o usuário altere uma reserva já efetuada, solicitando a seleção da reserva e os dados a serem alterados, como data e hora de início e fim da reserva, além do tipo de sala.

|  |  |
| --- | --- |
| **FLUXO DE EVENTOS NORMAL** | |
| 1 | O sistema exibe o menu inicial. |
| 2 | O usuário insere seu ID e senha para autenticação. |
| 3 | O sistema valida as credenciais do usuário. |
| 4 | O sistema exibe o menu de opções do usuário. |
| 5 | O usuário seleciona a opção "Alterar Reserva". |
| 6 | O sistema exibe uma lista de todas as reservas do usuário. |
| 7 | O usuário seleciona a reserva que deseja alterar. |
| 8 | O sistema solicita as novas informações (sala, data e hora de início e fim). |
| 9 | O usuário insere as novas informações da reserva. |
| 10 | O sistema valida a disponibilidade da nova sala e horário. |
| 11 | O sistema atualiza a reserva no banco de dados. |
| 12 | O sistema exibe uma mensagem de confirmação da alteração. |
| 13 | O sistema retorna ao menu de opções do usuário. |
| **FLUXO DE EVENTOS EXCEÇÃO** | |
| ***3a - Erro de autenticação:***  3a.1 - O sistema exibe uma mensagem de erro informando que as credenciais são inválidas. 3a.2 - O sistema retorna ao menu inicial. | |
| ***4a - Cancelamento da operação pelo usuário:***  4a.1 - O sistema retorna ao menu inicial. | |
| ***6a - Nenhuma reserva encontrada:***  6a.1 - O sistema exibe uma mensagem informando a ausência de reservas.  6a.2 - O sistema retorna ao menu de opções do usuário. | |
| ***9a - O sistema detecta que as novas informações (data ou horário de início/fim) estão incorretas:***  9a.1 - O sistema retorna ao passo 8 do fluxo normal para que o usuário corrija as informações. | |
| ***10a - Sala indisponível no período solicitado:***  10a.1 - O sistema exibe uma mensagem de erro informando a indisponibilidade. 10a.2 - O sistema oferece a opção de selecionar outra sala ou modificar o período, retornando ao passo 8 do fluxo normal. | |
| ***11a - Falha no sistema ao atualizar a reserva:*** 11a.1 - O sistema exibe uma mensagem de erro informando a falha ao atualizar a reserva. 11a.2 - O sistema orienta o usuário a tentar novamente, retornando ao passo 5 do fluxo normal. | |

### RF03 – Cancelar Reserva

O sistema deve permitir que o usuário cancele uma reserva já efetuada, solicitando a seleção da reserva a ser cancelada.

|  |  |
| --- | --- |
| **FLUXO DE EVENTOS NORMAL** | |
| 1 | O sistema exibe o menu inicial. |
| 2 | O usuário insere seu ID e senha para autenticação. |
| 3 | O sistema valida as credenciais do usuário. |
| 4 | O sistema exibe o menu de opções do usuário. |
| 5 | O usuário seleciona a opção "Cancelar Reserva". |
| 6 | O sistema exibe uma lista de todas as reservas do usuário. |
| 7 | O usuário seleciona a reserva que deseja cancelar. |
| 8 | O sistema solicita a confirmação do cancelamento. |
| 9 | O usuário confirma o cancelamento. |
| 10 | O sistema altera o status da reserva para “cancelada” no banco de dados. |
| 11 | O sistema exibe uma mensagem de confirmação do cancelamento. |
| 12 | O sistema retorna ao menu de opções do usuário. |
| **FLUXO DE EVENTOS EXCEÇÃO** | |
| ***3a - Erro de autenticação:***  3a.1 - O sistema exibe uma mensagem de erro informando que as credenciais são inválidas.  3a.2 - O sistema retorna ao menu inicial. | |
| ***4a - Cancelamento da operação pelo usuário:***  4a.1 - O sistema retorna ao menu inicial. | |
| ***6a - Nenhuma reserva encontrada:***  6a.1 - O sistema exibe uma mensagem informando a ausência de reservas.  6a.2 - O sistema retorna ao menu de opções do usuário. | |
| ***8a - O usuário opta por não confirmar o cancelamento:***  8a.1 - O sistema retorna ao menu de opções do usuário sem efetuar o cancelamento. | |
| ***10a - Falha no sistema ao remover a reserva:***  10a.1 - O sistema exibe uma mensagem de erro informando a falha ao remover a reserva.  10a.2 - O sistema orienta o usuário a tentar novamente, retornando ao passo 5 do fluxo normal. | |

### RF04 – Verificar Disponibilidade de Salas

O sistema deve permitir que o usuário verifique a disponibilidade de uma sala para saber se o horário desejado está ocupado antes de fazer a reserva.

|  |  |
| --- | --- |
| **FLUXO DE EVENTOS NORMAL** | |
| 1 | O sistema exibe o menu inicial. |
| 2 | O usuário insere seu ID e senha para autenticação. |
| 3 | O sistema valida as credenciais do usuário. |
| 4 | O sistema exibe o menu de opções do usuário. |
| 5 | O usuário seleciona a opção "Verificar Disponibilidade de Salas". |
| 6 | O sistema solicita o tipo de sala e a data que o usuário deseja verificar. |
| 7 | O usuário insere o tipo de sala e a data desejada. |
| 8 | O sistema exibe uma lista com todas as salas do tipo escolhido, mostrando todos os horários indisponíveis. |
| 9 | O sistema retorna ao menu de opções do usuário. |
| **FLUXO DE EVENTOS EXCEÇÃO** | |
| ***3a - Erro de autenticação:*** 3a.1 - O sistema exibe uma mensagem de erro informando que as credenciais são inválidas. 3a.2 - O sistema retorna ao menu inicial. | |
| ***4a - Cancelamento da operação pelo usuário:***  4a.1 - O sistema retorna ao menu inicial. | |
| ***7a - O sistema detecta que o tipo de sala ou a data inserida está incorreta.:*** 7a.1 - O sistema exibe uma mensagem de erro detalhando as informações inválidas. 7a.2 - O sistema retorna ao passo 6 do fluxo normal para que o usuário corrija as informações. | |
| ***8a - Falha no sistema ao exibir a disponibilidade:*** 8a.1 - O sistema orienta o usuário a tentar novamente, retornando ao passo 5 do fluxo normal. | |

### RF05 – Cadastrar Salas

O sistema deve permitir que o administrador cadastre novas salas no sistema, solicitando o número da sala, tipo e status.

|  |  |
| --- | --- |
| **FLUXO DE EVENTOS NORMAL** | |
| 1 | O sistema exibe o menu inicial. |
| 2 | O administrador insere sua matrícula e senha para autenticação. |
| 3 | O sistema valida as credenciais do administrador. |
| 4 | O sistema exibe o menu de opções do administrador. |
| 5 | O administrador seleciona a opção "Cadastrar Nova Sala". |
| 6 | O sistema solicita as informações da sala (número, tipo, status). |
| 7 | O administrador insere as informações solicitadas. |
| 8 | O sistema valida as informações fornecidas, verificando se o número da sala já não foi utilizado. |
| 9 | O sistema registra a nova sala no banco de dados. |
| 10 | O sistema exibe uma mensagem de confirmação do cadastro. |
| 11 | O sistema retorna ao menu de opções do administrador. |
| **FLUXO DE EVENTOS EXCEÇÃO** | |
| ***3a - Erro de autenticação:*** 3a.1 - O sistema exibe uma mensagem de erro informando que as credenciais são inválidas. 3a.2 - O sistema retorna ao menu inicial. | |
| ***4a - Cancelamento da operação pelo usuário:***  4a.1 - O sistema retorna ao menu inicial. | |
| ***8a - O sistema detecta que as informações inseridas estão incorretas ou o número da sala já foi utilizado:*** 8a.1 - O sistema retorna ao passo 6 do fluxo normal para que o administrador corrija as informações. | |
| ***9a - Falha no sistema ao registrar a sala:*** 9a.1 - O sistema orienta o administrador a tentar novamente, retornando ao passo 5 do fluxo normal. | |

### RF06 – Cadastrar Usuário

O sistema deve permitir que o administrador cadastre novos usuários no sistema, solicitando CPF, nome, perfil, senha e status.

|  |  |
| --- | --- |
| **FLUXO DE EVENTOS NORMAL** | |
| 1 | O sistema exibe o menu inicial. |
| 2 | O administrador insere sua matrícula e senha para autenticação. |
| 3 | O sistema valida as credenciais do administrador. |
| 4 | O sistema exibe o menu de opções do administrador. |
| 5 | O administrador seleciona a opção "Cadastrar Novo Usuário". |
| 6 | O sistema solicita as informações do usuário (CPF, nome, perfil, senha, status). |
| 7 | O administrador insere as informações solicitadas. |
| 8 | O sistema valida as informações fornecidas, verificando se o CPF é válido e se já não foi utilizado para outro usuário. |
| 9 | O sistema gera um ID sequencial e automático para o novo usuário. |
| 10 | O sistema registra o novo usuário no banco de dados. |
| 11 | O sistema exibe uma mensagem de confirmação do cadastro. |
| 12 | O sistema retorna ao menu de opções do administrador. |
| **FLUXO DE EVENTOS EXCEÇÃO** | |
| ***3a - Erro de autenticação:*** 3a.1 - O sistema exibe uma mensagem de erro informando que as credenciais são inválidas. 3a.2 - O sistema retorna ao menu inicial. | |
| ***4a - Cancelamento da operação pelo usuário:***  4a.1 - O sistema retorna ao menu inicial. | |
| ***8a - O sistema detecta que as informações inseridas estão incorretas ou o CPF já foi utilizado:*** 8a.1 - O sistema retorna ao passo 6 do fluxo normal para que o administrador corrija as informações. | |
| ***10a - Falha no sistema ao registrar o usuário:*** 10a.1 - O sistema orienta o administrador a tentar novamente, retornando ao passo 5 do fluxo normal. | |
|  | |

### RF07 – Alterar Salas

O sistema deve permitir que o administrador altere uma sala já cadastrada, solicitando o número, tipo e status da sala.

|  |  |
| --- | --- |
| **FLUXO DE EVENTOS NORMAL** | |
| 1 | O sistema exibe o menu inicial. |
| 2 | O administrador insere sua matrícula e senha para autenticação. |
| 3 | O sistema valida as credenciais do administrador. |
| 4 | O sistema exibe o menu de opções do administrador. |
| 5 | O administrador seleciona a opção "Alterar Sala". |
| 6 | O sistema exibe uma lista de todas as salas cadastradas. |
| 7 | O administrador seleciona a sala que deseja modificar. |
| 8 | O sistema solicita as novas informações da sala (número, tipo, status). |
| 9 | O administrador insere as novas informações. |
| 10 | O sistema valida as informações fornecidas. |
| 11 | O sistema verifica se não há nenhum agendamento programado em caso de mudança de status. |
| 12 | O sistema atualiza as informações da sala no banco de dados. |
| 13 | O sistema exibe uma mensagem de confirmação da alteração. |
| 14 | O sistema retorna ao menu de opções do administrador |
| **FLUXO DE EVENTOS EXCEÇÃO** | |
| ***3a - Erro de autenticação:*** 3a.1 - O sistema exibe uma mensagem de erro informando que as credenciais são inválidas. 3a.2 - O sistema retorna ao menu inicial. | |
| ***4a - Cancelamento da operação pelo usuário:***  4a.1 - O sistema retorna ao menu inicial. | |
| ***10a - Informações inválidas:*** 10a.1 - O sistema retorna ao passo 8 do fluxo normal para que o administrador corrija as informações. | |
| ***11a - Sala com agendamentos programados:*** 11a.1 - O sistema exibe uma mensagem informando a impossibilidade de alteração de status. 11a.2 - O sistema retorna ao passo 6 do fluxo normal. | |
| ***13a - Falha no sistema ao atualizar a sala:*** 13a.1 - O sistema orienta o administrador a tentar novamente, retornando ao passo 5 do fluxo normal. | |

### RF08 – Listar Usuários

O sistema deve permitir que o administrador liste todos os usuários cadastrados no sistema, exibindo os dados completos para cada um.

|  |  |
| --- | --- |
| **FLUXO DE EVENTOS NORMAL** | |
| 1 | sistema exibe o menu inicial. |
| 2 | O administrador insere sua matrícula e senha para autenticação. |
| 3 | O sistema valida as credenciais do administrador. |
| 4 | O sistema exibe o menu de opções do administrador. |
| 5 | O administrador seleciona a opção "Listar Usuários". |
| 6 | O sistema recupera e exibe uma lista de todos os usuários cadastrados, com seus dados completos. |
| 7 | O sistema retorna ao menu de opções do administrador. |
| **FLUXO DE EVENTOS EXCEÇÃO** | |
| ***3a - Erro de autenticação:*** 3a.1 - O sistema exibe uma mensagem de erro informando que as credenciais são inválidas. 3a.2 - O sistema retorna ao menu inicial. | |
| ***4a - Cancelamento da operação pelo usuário:*** 4a.1 - O sistema retorna ao menu inicial. | |
| ***6a - Falha no sistema ao recuperar a lista de usuários:*** 6a.1 - O sistema orienta o administrador a tentar novamente, retornando ao passo 5 do fluxo normal. | |

### RF09 – Alterar Usuário

O sistema deve permitir que o administrador altere um usuário cadastrado, solicitando o dado a ser alterado (CPF, nome, perfil, senha ou status).

|  |  |
| --- | --- |
| **FLUXO DE EVENTOS NORMAL** | |
| 1 | O sistema exibe o menu inicial. |
| 2 | O administrador insere sua matrícula e senha para autenticação. |
| 3 | O sistema valida as credenciais do administrador. |
| 4 | O sistema exibe o menu de opções do administrador. |
| 5 | O administrador seleciona a opção "Alterar Usuário". |
| 6 | O sistema exibe uma lista de todos os usuários cadastrados. |
| 7 | O administrador seleciona o usuário que deseja modificar. |
| 8 | O sistema solicita as novas informações do usuário (CPF, nome, perfil, senha, status). |
| 9 | O administrador insere as novas informações. |
| 10 | O sistema valida as informações fornecidas. |
| 11 | O sistema verifica se não há nenhum agendamento programado em caso de mudança de status. |
| 12 | O sistema atualiza as informações do usuário no banco de dados. |
| 13 | O sistema exibe uma mensagem de confirmação da alteração. |
| 14 | O sistema retorna ao menu de opções do administrador. |
| **FLUXO DE EVENTOS EXCEÇÃO** | |
| ***3a - Erro de autenticação:*** 3a.1 - O sistema exibe uma mensagem de erro informando que as credenciais são inválidas. 3a.2 - O sistema retorna ao menu inicial. | |
| ***4a - Cancelamento da operação pelo usuário:*** 4a.1 - O sistema retorna ao menu inicial. | |
| ***10a - Informações inválidas:*** 10a.1 - O sistema retorna ao passo 8 do fluxo normal para que o administrador corrija as informações. | |
| ***11a - Usuário com agendamentos programados:*** 11a.1 - O sistema exibe uma mensagem informando a impossibilidade de alteração de status. 11a.2 - O sistema retorna ao passo 6 do fluxo normal. | |
| ***13a - Falha no sistema ao atualizar o usuário:*** 13a.1 - O sistema orienta o administrador a tentar novamente, retornando ao passo 5 do fluxo normal. | |

### RF10 – Alterar Senha

O sistema deve permitir que o usuário altere sua senha de acesso, solicitando a senha atual e a nova senha.

|  |  |
| --- | --- |
| **FLUXO DE EVENTOS NORMAL** | |
| 1 | O sistema exibe o menu inicial. |
| 2 | O usuário insere seu ID e senha para autenticação. |
| 3 | O sistema valida as credenciais do usuário. |
| 4 | O sistema exibe o menu de opções do usuário. |
| 5 | O usuário seleciona a opção "Alterar Senha". |
| 6 | O sistema solicita a senha atual e a nova senha. |
| 7 | O usuário insere a senha atual e a nova senha. |
| 8 | O sistema valida a senha atual. |
| 9 | O sistema atualiza a senha no banco de dados. |
| 10 | O sistema exibe uma mensagem de confirmação da alteração de senha. |
| 11 | O sistema retorna ao menu de opções do usuário. |
| **FLUXO DE EVENTOS EXCEÇÃO** | |
| ***3a - Erro de autenticação:*** 3a.1 - O sistema exibe uma mensagem de erro informando que as credenciais são inválidas. 3a.2 - O sistema retorna ao menu inicial. | |
| ***4a - Cancelamento da operação pelo usuário:*** 4a.1 - O sistema retorna ao menu inicial. | |
| ***8a - Senha atual inválida:*** 8a.1 - O sistema exibe uma mensagem de erro informando que a senha atual é inválida. 8a.2 - O sistema retorna ao passo 6 do fluxo normal para que o usuário insira novamente as senhas. | |
| ***9a - Falha no sistema ao atualizar a senha:*** 9a.1 - O sistema orienta o usuário a tentar novamente, retornando ao passo 5 do fluxo normal. | |

### RF11 – Exibir Reservas

O sistema deve permitir que o usuário veja todas as reservas efetuadas utilizando seu ID e senha.

|  |  |
| --- | --- |
| **FLUXO DE EVENTOS NORMAL** | |
| 1 | O sistema exibe o menu inicial. |
| 2 | O usuário insere seu ID e senha para autenticação. |
| 3 | O sistema valida as credenciais do usuário. |
| 4 | O sistema exibe o menu de opções do usuário. |
| 5 | O usuário seleciona a opção "Exibir Reservas". |
| 6 | O sistema recupera e exibe uma lista de todas as reservas efetuadas pelo usuário, exibindo o status de cada reserva. |
| 7 | O sistema retorna ao menu de opções do usuário. |
| **FLUXO DE EVENTOS EXCEÇÃO** | |
| ***3a - Erro de autenticação:*** 3a.1 - O sistema exibe uma mensagem de erro informando que as credenciais são inválidas. 3a.2 - O sistema retorna ao menu inicial. | |
| ***4a - Cancelamento da operação pelo usuário:*** 4a.1 - O sistema retorna ao menu inicial. | |
| ***6a - Falha no sistema ao recuperar as reservas:*** 6a.1 - O sistema orienta o usuário a tentar novamente, retornando ao passo 5 do fluxo normal. | |
|  | |

### RF12 – Listar Reservas

O sistema deve permitir que o administrador liste todas as reservas efetuadas no período estipulado para um tipo de sala específico.

|  |  |
| --- | --- |
| **FLUXO DE EVENTOS NORMAL** | |
| 1 | O sistema exibe o menu inicial. |
| 2 | O administrador insere sua matrícula e senha para autenticação. |
| 3 | O sistema valida as credenciais do administrador. |
| 4 | O sistema exibe o menu de opções do administrador. |
| 5 | O administrador seleciona a opção "Listar Reservas". |
| 6 | O sistema solicita o tipo de sala e o período que o administrador deseja consultar. |
| 7 | O administrador insere o tipo de sala e o período desejado. |
| 8 | O sistema recupera e exibe uma lista de todas as reservas efetuadas no período e tipo de sala especificados. |
| 9 | O sistema retorna ao menu de opções do administrador. |
| **FLUXO DE EVENTOS EXCEÇÃO** | |
| ***3a - Erro de autenticação:*** 3a.1 - O sistema exibe uma mensagem de erro informando que as credenciais são inválidas. 3a.2 - O sistema retorna ao menu inicial. | |
| ***4a - Cancelamento da operação pelo usuário:*** 4a.1 - O sistema retorna ao menu inicial. | |
| ***8a - Falha no sistema ao recuperar as reservas:*** 8a.1 - O sistema orienta o administrador a tentar novamente, retornando ao passo 5 do fluxo normal. | |
|  | |

## Requisitos não-funcionais

### Desempenho

* O sistema deve ser capaz de processar solicitações de reserva, alteração e cancelamento em menos de 2 segundos.
* O sistema deve suportar até 100 usuários simultâneos sem degradação de desempenho.

### Segurança

* O sistema deve criptografar todas as senhas de usuário no banco de dados.
* O sistema deve implementar autenticação de dois fatores para acesso administrativo.
* Todas as transações de dados sensíveis devem ser realizadas através de conexões seguras (HTTPS).

### Usabilidade

* A interface do usuário deve ser intuitiva e fácil de navegar, com um tempo de aprendizado estimado de menos de 30 minutos para novos usuários.
* O sistema deve fornecer mensagens de erro claras e úteis para ajudar os usuários a corrigirem problemas.

### Confiabilidade

* O sistema deve ter uma disponibilidade de 99,9%, garantindo que esteja operacional 24/7.
* O sistema deve realizar backups automáticos dos dados a cada 24 horas.

### Escalabilidade

* O sistema deve ser capaz de escalar horizontalmente para acomodar um aumento no número de usuários e transações sem necessidade de reestruturação significativa.

### Compatibilidade

* O sistema deve ser compatível com os principais SO’s (Windows, Linux e MacOS) e dispositivos móveis.

### Manutenibilidade:

* O sistema deve ser modular para facilitar atualizações e manutenção.
* O código deve ser documentado de forma clara para permitir que novos desenvolvedores entendam e modifiquem o sistema facilmente.

# arquitetura do sistema

## Visão Geral da Arquitetura

A arquitetura do ReuniON é projetada de forma modular, organizada em duas camadas principais: a Camada de Biblioteca e a Camada de Implementação.

* **Camada de Dados (databases):** Esta camada é responsável por armazenar e organizar todos os registros e informações inseridas no sistema em arquivos de dados (.csv). Os arquivos principais incluem sala.csv e usuario.csv, onde são mantidos os dados das salas e dos usuários, respectivamente. Essa estrutura de dados em arquivos facilita o armazenamento persistente e a integridade das informações, permitindo acesso estruturado e seguro para as outras camadas do sistema.
* **Camada de Biblioteca (lib):** Esta camada contém os arquivos de cabeçalho (.h) que definem as interfaces e estruturas de dados utilizadas no sistema. Os arquivos incluem bancoDados.h, menu.h, reservas.h, salas.h, usuario.h e utils.h. Essa organização facilita a modularização e a reutilização do código, permitindo que as interfaces sejam claramente definidas e acessíveis para os componentes que necessitam delas.
* **Camada de Implementação (src):** Nesta camada, encontram-se os arquivos de implementação (.c) correspondentes, onde as funcionalidades são efetivamente implementadas. Os arquivos incluem bancoDados.c, menu.c, reservas.c, salas.c, usuario.c e utils.c.
* **Camada principal:** Esta camada contém o ponto de entrada do sistema, representado pelo arquivo main.c, que orquestra a execução de todas as funcionalidades implementadas nas camadas anteriores. Além do main.c, esta camada inclui todos os arquivos e dependências necessárias para o funcionamento do sistema, gerenciando a inicialização, fluxo e coordenação entre os componentes para uma experiência integrada e coesa.

Essa estrutura modular permite que cada componente funcione de forma independente e interaja com os demais por meio de interfaces definidas, facilitando a manutenção e futuras expansões do sistema.

## Componentes do Sistema

Os principais componentes do sistema e suas responsabilidades são:

* **Banco de Dados:** Responsável por gerenciar a persistência das informações sobre salas e reservas, garantindo que os dados sejam armazenados e recuperados de forma eficiente.
* **Menu:** Define e controla a interface de interação com o usuário, exibindo as opções disponíveis e capturando as entradas do usuário.
* **Reservas:** Lida com o processo de reserva de salas, incluindo a verificação de disponibilidade e o cancelamento de reservas, assegurando que os conflitos de agendamento sejam evitados.
* **Salas:** Gerencia o cadastro e as informações das salas de reunião, permitindo que novas salas sejam adicionadas ou modificadas conforme necessário.
* **Usuário:** Lida com o cadastro e autenticação dos usuários, controlando os níveis de permissão (usuário e administrador) para garantir a segurança e a integridade do sistema.
* **Utils:** Provê funções utilitárias, como validações e manipulações de strings, que são utilizadas por outros componentes para realizar operações comuns.

Esses componentes são independentes, mas interagem para formar um sistema coeso e funcional.

## Tecnologias Utilizadas

O sistema foi desenvolvido utilizando as seguintes tecnologias:

* **Linguagem de Programação:** C, escolhida por sua eficiência e controle sobre a gestão de memória e desempenho, permitindo um sistema rápido e de baixo nível.
* **Ambiente de Desenvolvimento:** Visual Studio Code, utilizado como IDE para facilitar o desenvolvimento e a depuração do código.
* **Controle de Versão:** Git, utilizado para gerenciar o código-fonte e colaborar de forma eficaz durante o desenvolvimento.

Essas tecnologias foram selecionadas para oferecer um sistema robusto e eficiente, com controle total sobre os recursos do sistema.

# IMPLEMENTAÇÃO DO SISTEMA

## ESTRUra do código

O sistema é organizado em uma estrutura de diretórios clara, que separa as definições de interfaces, implementações e dados. No diretório lib, encontram-se os arquivos de cabeçalho (.h), como bancoDados.h, menu.h, reservas.h, salas.h, usuario.h e utils.h. Esses arquivos definem as funções, tipos de dados e constantes, promovendo a modularidade do sistema. Já o diretório src abriga os arquivos de implementação (.c), onde a lógica dos módulos é realizada. O arquivo main.c está localizado na camada principal, e desempenha um papel central, coordenando a execução do sistema ao chamar funções de diferentes módulos para operações de reserva, consulta e gestão de salas. Além disso, o diretório databases contém os arquivos de dados (.csv), como sala.csv e usuario.csv, que armazenam as informações persistentes. Essa organização modular facilita a manutenção e evolução do código, permitindo que cada módulo tenha uma responsabilidade bem definida.

## Principais Algoritmos e Funções

O sistema implementa uma série de algoritmos e funções essenciais para seu funcionamento:

* **Banco de Dados:** Funções que garantem a persistência dos dados, salvando e recuperando informações sobre reservas e salas.
* **Menu:** Funções que exibem opções ao usuário, capturam entradas e direcionam para as operações adequadas.
* **Reservas:** Funções que gerenciam a lógica de reserva, verificando a disponibilidade de salas e resolvendo conflitos.
* **Salas:** Funções que cuidam do cadastro e gestão das informações das salas de reunião.
* **Usuário:** Funções que realizam o cadastro e autenticação dos usuários, além de controlar permissões.
* **Utilitárias (utils):** Funções de suporte que realizam validações, formatação e manipulação de strings.

Essas funções são organizadas de forma que o main.c centraliza o controle do fluxo de execução, garantindo que as operações sejam realizadas de maneira coordenada.

## Padrões de Codificação

Durante o desenvolvimento, foram seguidos padrões de codificação que asseguram a clareza e a consistência do código:

* **Nomenclatura de Variáveis:** Utilização de nomes claros e descritivos, adotando o padrão camelCase para melhorar a legibilidade.
* **Formatação do Código:** O código é formatado de maneira consistente, com indentação e espaçamento adequados, facilitando a leitura e manutenção.
* **Comentários:** Inclusão de comentários para explicar blocos de código complexos e documentar o propósito de cada função, auxiliando na compreensão do código por outros desenvolvedores.

Essas práticas tornam o código mais acessível e colaborativo, facilitando futuras expansões.

## Ferramentas de Desenvolvimento

Para o desenvolvimento e depuração do código, foram utilizadas as seguintes ferramentas:

* **IDE:** Visual Studio Code, que oferece um ambiente de desenvolvimento integrado com suporte a extensões e ferramentas de depuração.
* **Compilador:** GCC, amplamente utilizado para compilar projetos em C, garantindo compatibilidade e eficiência.
* **Controle de Versão:** Git, utilizado para rastrear alterações e colaborar de forma eficiente no projeto.

Essas ferramentas foram escolhidas para proporcionar um fluxo de trabalho ágil e eficiente, especialmente no contexto de desenvolvimento em C.

# Testes e Validação

## Estratégia de Testes

A estratégia de testes adotada para o ReuniON é abrangente, cobrindo diferentes níveis de verificação para assegurar a qualidade e a robustez do sistema. A abordagem inclui:

* **Testes Unitários:** Cada módulo do sistema, como banco de dados, menu, reservas, salas, usuário e utils, é testado de forma isolada. Esses testes garantem que as funções individuais operem conforme o esperado, permitindo a identificação precoce de problemas em funções específicas antes de avançar para testes mais complexos.
* **Testes de Integração:** Após a validação dos módulos individualmente, são realizados testes para verificar a integração entre eles. Isso assegura que a interação entre os módulos ocorra corretamente, como a comunicação entre o módulo de reservas e o banco de dados, garantindo que as reservas sejam armazenadas e recuperadas adequadamente.
* **Testes de Sistema:** Testes completos são realizados no sistema, simulando o fluxo de uso real desde o login até a realização de reservas e consultas. Esses testes cobrem tanto casos de sucesso quanto situações de erro, garantindo que o sistema funcione corretamente em um cenário real.

Essa abordagem permite identificar problemas em diferentes níveis do sistema, assegurando tanto a confiabilidade das funcionalidades individuais quanto a robustez do sistema como um todo.

## Casos de Teste

Os principais casos de teste definidos para o sistema incluem:

* **Cadastro de Usuário:** Verifica se um usuário pode ser cadastrado com informações válidas e se o sistema impede cadastros com dados inválidos.
* **Autenticação de Usuário:** Testa o login para usuários com permissões diferentes (usuário comum e administrador) e verifica se o sistema lida corretamente com tentativas de login inválidas.
* **Reserva de Sala:** Verifica se o sistema permite reservas de salas disponíveis e bloqueia reservas em horários ou datas conflitantes.
* **Cancelamento de Reserva:** Testa a capacidade de cancelar uma reserva existente e verifica se o sistema lida corretamente com tentativas de cancelamento de reservas inexistentes.
* **Consulta de Salas:** Verifica se as informações de salas cadastradas são exibidas corretamente ao usuário.
* **Persistência de Dados:** Testa se os dados de salas e reservas são armazenados e recuperados corretamente após o sistema ser fechado e reiniciado.

Esses casos cobrem as funcionalidades principais e verificam tanto o funcionamento em condições normais quanto o tratamento de erros.

## Ferramentas de Teste

Para a realização dos testes, foram utilizadas as seguintes ferramentas:

* **Compilador GCC com Flags de Depuração:** Permite compilar o código com opções de depuração, facilitando a identificação de erros.
* **Testes Manuais:** Testes manuais foram realizados para simular fluxos de usuário, garantindo que as funcionalidades principais e cenários comuns estejam funcionando como esperado.
* **Logs e Prints:** Utilização de mensagens de log e printf para verificar o fluxo do programa e identificar pontos de falha durante a execução dos testes.

## Resultados dos Testes

Os resultados dos testes realizados indicaram:

* **Falhas Identificadas:** Durante os testes iniciais, foram encontrados alguns problemas, como erros de validação no cadastro de usuários e conflitos na reserva de salas quando mais de um usuário tentava reservar o mesmo horário.
* **Correções Implementadas:** Para resolver essas falhas, foram adicionadas verificações mais rigorosas de validação no cadastro de usuários e bloqueios temporários durante o processo de reserva para evitar conflitos. Funções auxiliares em utils foram ajustadas para garantir a integridade dos dados ao longo do fluxo de reserva e cancelamento.
* **Resultados Finais:** Após as correções, o sistema passou em todos os casos de teste, mostrando-se funcional tanto em condições normais quanto em cenários de erro.

# Conclusão

## Resultados Alcançados

* **Eficiência na Gestão de Reservas:** O sistema alcançou os objetivos iniciais, permitindo que os usuários reservem salas de forma fácil e rápida, eliminando conflitos de agendamento e melhorando a organização.
* **Facilidade de Uso:** A interface de menu clara possibilita que os usuários realizem operações básicas, como cadastro, reserva e cancelamento, sem dificuldades.
* **Persistência de Dados:** A implementação de um banco de dados assegura que as informações de reservas e usuários sejam mantidas mesmo após o fechamento do sistema, aumentando a confiabilidade.
* **Modularidade e Extensibilidade:** A arquitetura modular facilita a manutenção e futuras melhorias, permitindo a escalabilidade do sistema e a adição de novas funcionalidades.

## Limitações e Desafios

* **Interface limitada:** A interface de linha de comando pode ser uma barreira para usuários menos experientes, que poderiam se beneficiar de uma interface gráfica.
* **Controle de conflitos em tempo real:** Em ambientes com muitos usuários tentando reservar simultaneamente, pode haver atrasos na atualização dos dados.
* **Escalabilidade:** O sistema atende bem a ambientes de pequeno a médio porte, mas pode precisar de ajustes para suportar um volume muito alto de usuários e reservas simultâneas.

## Trabalhos Futuros

* **Interface gráfica (GUI):** Desenvolver uma interface gráfica para tornar o sistema mais acessível e fácil de usar.
* **Integração com calendários externos**: Integrar com sistemas de calendário como Google Calendar ou Outlook para sincronizar reservas com calendários pessoais.
* **Controle de conflitos e concorrência:** Melhorar o gerenciamento de reservas simultâneas, implementando bloqueios e sincronização mais eficazes.
* **Notificações e alertas:** Adicionar notificações para lembrar os usuários de suas reservas e alertá-los sobre mudanças.

## Lições Aprendidas

* **Importância da modularidade:** A arquitetura modular facilitou o desenvolvimento, manutenção e teste do sistema, destacando a importância de uma estrutura de código bem-organizada.
* **Testes contínuos:** Realizar testes durante todo o desenvolvimento ajudou a identificar e resolver problemas rapidamente, evitando a acumulação de erros.
* **Planejamento de interface:** Uma interface intuitiva é fundamental para a experiência do usuário, mesmo em sistemas de linha de comando. A clareza do menu e das opções contribuiu significativamente para a usabilidade.
* **Gerenciamento de conflitos:** O controle de concorrência mostrou-se um desafio, destacando a necessidade de explorar melhores práticas de sincronização de dados para evitar conflitos

###### REFERÊNCIAS

SOMMERVILLE, Ian. **Engenharia de Software**. Tradução Ivan Bosnic e Kalinka G. de O. Gonçalves; revisão técnica Kechi Hirama. 9. ed. São Paulo: Pearson Prentice Hall, 2011. Título original: Software engineering. ISBN 978-85-7936-108-1.

UFES - Universidade Federal do Espírito Santo. **Engenharia de Requisitos: Notas de Aula**. Ricardo de Almeida Falbo. 2017.

UFES - Universidade Federal do Espírito Santo. **Engenharia de Software: Notas de Aula**. Ricardo de Almeida Falbo. 2014.

SERRA, Ana Paula Gonçalves. **Documentação de um Produto de Software**. Versão 3.0. Revisor: Prof. Fernando Giorno. 2005.