****

**DOCUMENTO**

**DE REQUISITOS**

Versão 1.0 - 29/05/2025

PROJETO Tec Web 01

**SUMÁRIO**

[HISTÓRICO DE REVISÕES](#_heading=h.okjoa86ume4)

[1. INTRODUÇÃO](#_heading=h.x100hokc1z5e)

[1.1 Propósito deste documento](#_heading=h.bhaoimownpmu)

[1.2 Escopo do produto](#_heading=h.s8yiqnyuszgq)

[1.3 Definições, acrônimos e abreviações](#_heading=h.5zn4npu5ohmw)

[1.4 Visão geral do documento](#_heading=h.f4wv4b9k7889)

[2. VISÃO GERAL DO PRODUTO](#_heading=h.qwnsghb6jcj2)

[2.1 Panorama geral do produto](#_heading=h.msb8epsi2nq5)

[2.1 Funções do produto](#_heading=h.xzr8uf7fo89g)

[2.3 Restrições do produto](#_heading=h.ebjh0qwz0wa3)

[2.4 Tipos de atores](#_heading=h.2pk6nj30u5l2)

[2.5 Suposições e dependências](#_heading=h.6m08e9ncurho)

[2.6 Distribuição da especificação](#_heading=h.p7ufuq3ba9hh)

[3. ESPECIFICAÇÃO](#_heading=h.90atbwa6wx7w)

[3.1 Módulo 1](#_heading=h.j416w4l4t9ij)

[Como ATOR 1, quero FAZER LOGIN para ACESSAR meu dashboard](#_heading=h.4qzpe2vi9br5)

[4. INTERFACES EXTERNAS](#_heading=h.t1n9uuju47v9)

[4.1 Interfaces de software](#_heading=h.p3kc39jqaw9v)

[4.2 Interfaces de usuário](#_heading=h.drlr783a0ngx)

[4.3 Interfaces de hardware](#_heading=h.9pteh2ng57md)

[5. VERIFICAÇÃO](#_heading=h.w27jpx2znrmi)

[6. ANEXOS](#_heading=h.2a8yob8p815b)

# HISTÓRICO DE REVISÕES

| **Responsável** | **Data** | **Descrição breve das alterações** | **Versão** |
| --- | --- | --- | --- |
| Francisco Felipe Rodrigues de Sousa | 27/05/2025 | Versão inicial da especificação | 1.0 |

# 

# INTRODUÇÃO

## 

## 1.1 Propósito deste documento

Este documento tem como objetivo especificar os requisitos do projeto Sistema de Gerenciamento de Laboratórios UFC, incluindo suas restrições, dependências e interfaces. Fazem parte do público-alvo deste documento: Coordenadores, professores, alunos, técnicos de laboratórios, pesquisadores, colaboradores e todos que atuam nas tarefas de especificação, projeto, construção, testes e verificação.

## 1.2 Escopo do produto

O Sistema de Gerenciamento de Laboratórios (SGL) abrange os seguintes módulos principais:

* Módulo de Cadastro e Gestão de Laboratórios ;
* Módulo de Consulta de Disponibilidade ;
* Módulo de Gerenciamento de Agendamentos ;
* Módulo de Notificações ;
* Módulo de Relatórios e Análises (Dashboard Analítico) ;

## 1.3 Definições, acrônimos e abreviações

Para garantir clareza e padronização ao longo da documentação e do desenvolvimento do projeto, as seguintes definições, acrônimos e abreviações serão utilizadas:

* **API (Application Programming Interface):** Conjunto de regras e protocolos que permite que diferentes softwares se comuniquem entre si. No contexto deste projeto, refere-se às interfaces de comunicação entre o front-end e o back-end.
* **Back-end:** A parte do sistema que lida com a lógica do servidor, banco de dados e APIs. No projeto, será desenvolvido com **Node.js** e **Express**.
* **Bootstrap:** Framework de código aberto para desenvolvimento de interfaces web responsivas e mobile-first, utilizado para padronizar o design e a responsividade do front-end.
* **Campus Mucambinho:** Um dos campi da Universidade Federal do Ceará (UFC) em Sobral, Ceará, onde o sistema será implementado para gerenciar as salas de aula do Bloco das Engenharias.
* **CSS3 (Cascading Style Sheets, Nível 3):** Linguagem de estilo utilizada para descrever a apresentação de um documento escrito em HTML, controlando cores, fontes, layout, etc.
* **Express:** Framework web minimalista e flexível para Node.js, utilizado para construir APIs REST no back-end.
* **Front-end:** A parte do sistema com a qual o usuário interage diretamente, incluindo a interface gráfica. Será desenvolvido com **HTML5**, **CSS3** e **JavaScript** com **Bootstrap**.
* **HTML5 (HyperText Markup Language, Versão 5):** Linguagem de marcação utilizada para estruturar o conteúdo de páginas web.
* **JavaScript:** Linguagem de programação amplamente utilizada para criar interatividade e dinamismo em páginas web.
* **JWT (JSON Web Tokens):** Padrão aberto e seguro para a criação de tokens de acesso que verificam a identidade do usuário e as permissões de acesso. Será utilizado para autenticação no sistema.
* **MySQL:** Sistema de gerenciamento de banco de dados relacional de código aberto, utilizado para armazenar informações de usuários, salas e reservas.
* **Node.js:** Ambiente de tempo de execução JavaScript de código aberto e multiplataforma, utilizado para construir aplicações de rede escaláveis, sendo a base do back-end do sistema.
* **UFC Sobral:** Universidade Federal do Ceará, Campus Sobral, instituição à qual o Campus Mucambinho pertence e onde o sistema será implementado.
* **Sistema Web:** Aplicação de software que funciona em um navegador web, acessível pela internet, que é o tipo de sistema a ser desenvolvido neste projeto.

## 1.4 Visão geral do documento

O restante deste documento está organizado em quatro seções: (i) visão geral do produto, em que são abordadas as principais características do produto especificado, tais como o panorama geral e funções do produto, tipos de atores, suposições e dependências; (ii) especificação, em que são descritas as funcionalidades previstas para o produto; (iii) verificação, em que são mencionados os critérios de verificação de conformidade dos requisitos especificados e, por fim, (iv) anexos, onde são listados os textos complementares a este documento.

# VISÃO GERAL DO PRODUTO

Esta seção descreve os fatores gerais que afetam o desenvolvimento deste produto, sua especificação e projeto. Ressalta-se que a especificação e o projeto em si não são detalhados nesta seção, mas as informações aqui descritas oferecem um pano de fundo para estes requisitos, que são definidos detalhadamente na Seção 3.

## 2.1 Panorama geral do produto

O **Sistema de Gerenciamento de Laboratórios (SGL)** é uma **aplicação web** que surge como uma solução para otimizar a administração e o uso dos espaços no Bloco das Engenharias do Campus Mucambinho da UFC Sobral. Atualmente, a gestão de salas enfrenta desafios como **conflitos de agendamento, falta de visibilidade em tempo real sobre a disponibilidade e ausência de um registro centralizado** do histórico de uso, resultando em retrabalho administrativo e insatisfação dos usuários.

O SGL será uma plataforma intuitiva e integrada, acessível via navegador web, projetada para **facilitar o agendamento, a liberação e o monitoramento** do uso das salas. Sua estrutura analítica do projeto (EAP) pode ser visualizada pelos seguintes módulos principais:

* **Módulo de Cadastro e Gestão de Salas:** Responsável por permitir a inclusão, edição e remoção de informações detalhadas sobre cada sala de aula.
* **Módulo de Consulta de Disponibilidade:** Oferecerá interfaces visuais (calendário e lista) para que os usuários possam verificar a ocupação das salas em tempo real.
* **Módulo de Gerenciamento de Agendamentos:** Permitirá a solicitação, aprovação, rejeição e cancelamento de reservas, gerando o fluxo completo de um agendamento.
* **Módulo de Notificações:** Encarregado de enviar alertas e comunicações importantes aos usuários e administradores sobre o status das reservas.
* **Módulo de Relatórios:** Produzirá dados analíticos sobre o uso das salas, auxiliando na tomada de decisões e otimização de recursos.

## 2.2 Funções do produto

As funções esperadas para cada módulo do SGL são:

* **Módulo de Cadastro e Gestão de Salas:**
  + Cadastrar novas salas com nome, capacidade, recursos (projetor, ar-condicionado), localização.
  + Editar informações de salas existentes.
  + Excluir salas do sistema (com validação de dependências).
  + Definir o status da sala (ativa/inativa).
* **Módulo de Consulta de Disponibilidade:**
  + Visualizar calendário de agendamentos por dia, semana e mês.
  + Filtrar salas por recursos disponíveis e capacidade.
  + Consultar disponibilidade em formato de lista.
* **Módulo de Gerenciamento de Agendamentos:**
  + Solicitar reserva de uma sala para um período específico.
  + Administrador aprovar ou rejeitar solicitações de reserva.
  + Modificar detalhes de uma reserva existente (por administradores).
  + Cancelar reservas (por usuários e administradores).
  + Verificar e alertar sobre conflitos de agendamento.
* **Módulo de Notificações:**
  + Enviar e-mails de confirmação de reserva.
  + Enviar e-mails de rejeição ou modificação de reserva.
  + Enviar e-mails de cancelamento de reserva.
  + Gerar alertas internos para administradores sobre novas solicitações ou conflitos.
* **Módulo de Relatórios:**
  + Gerar relatório de uso por sala (mensal/anual).
  + Gerar relatório de reservas por usuário.
  + Calcular e exibir a taxa de ocupação das salas.

## 2.3 Restrições do produto

As seguintes restrições são impostas ao desenvolvimento e operação do SGL:

* **Alcance Geográfico:** O sistema será restrito ao gerenciamento de laboratórios do **Campus Mucambinho da UFC Sobral**.
* **Tecnológicas:** O desenvolvimento do Front-end será em **HTML5, CSS3, JavaScript com Bootstrap**; o Back-end em **Node.js com Express**; o Banco de Dados será **MySQL**; e a autenticação será via **JWT**.
* **Segurança:** A autenticação dos usuários é mandatória para acesso às funcionalidades, garantindo que apenas usuários autorizados possam interagir com o sistema.
* **Integração Externa:** Não haverá integração direta com outros sistemas da UFC Sobral (e.g., sistema acadêmico, diretor de usuários) em sua versão inicial.
* **Acesso:** O sistema exige uma conexão de internet estável para seu funcionamento.

## 2.4 Tipos de atores

Os atores são usuários, equipamentos ou sistemas externos com os quais o produto proposto pode interagir. Neste contexto, estão previstos **3** atores distintos:

* **Administrador do Sistema (Gestor de Salas):**
  + **Papel:** É o ator com controle total sobre o sistema. Ele será responsável por cadastrar e gerenciar salas, aprovar ou rejeitar todas as solicitações de reserva, visualizar e gerenciar todos os usuários, e acessar todos os relatórios analíticos do sistema. Atua como o principal responsável pela manutenção e organização dos dados.
* **Professor/Docente:**
  + **Papel:** Representa o corpo docente da UFC Sobral. Poderá consultar a disponibilidade das salas de aula, submeter solicitações de reserva para suas aulas, bancas ou outras atividades acadêmicas, e receber notificações sobre o status de suas próprias reservas.
* **Técnico Administrativo:**
  + **Papel:** Refere-se aos servidores técnico-administrativos da universidade. Poderá consultar a disponibilidade das salas e solicitar agendamentos para reuniões, eventos administrativos ou outras atividades relacionadas à sua função. Assim como os professores, receberá notificações sobre suas reservas.

## 2.5 Suposições e dependências

As seguintes suposições e dependências impactam o desenvolvimento e sucesso do SGL:

* **Disponibilidade de Infraestrutura de Hospedagem:** Assume-se que a UFC Sobral fornecerá ou disponibilizará o ambiente de servidor necessário para a hospedagem e operação do sistema.
* **Cooperação Institucional:** A colaboração da coordenação do Bloco das Engenharias e da equipe de TI da UFC Sobral é crucial para o fornecimento de dados iniciais (listagem de salas, definição de regras de uso, identificação de usuários) e para a validação das funcionalidades desenvolvidas.
* **Acesso à Internet:** É uma premissa que os usuários finais terão acesso à internet para conseguir acessar e interagir com o sistema web.
* **Conhecimento Básico de Informática:** Pressupõe-se que os atores envolvidos possuam conhecimento fundamental sobre o uso de navegadores web e sistemas online.
* **Segurança de Dados:** O sistema dependerá da adesão a práticas de segurança no desenvolvimento e na infraestrutura para proteger os dados sensíveis dos usuários e dos agendamentos.
* **Manutenção de Dados:** A qualidade dos relatórios e a eficácia do sistema dependem da correta inserção e atualização dos dados de agendamento pelos usuários.

## 2.6 Distribuição da especificação

Nesta seção é apresentado o modo como a especificação dos requisitos está organizada na **Seção 3**. O detalhamento segue um padrão que favorece a clareza, a rastreabilidade e a manutenção futura:

* **Agrupamento por módulos**: Cada módulo definido na visão geral (Seção 2.1) dá origem a uma subseção específica em 3.x, contendo todos os requisitos funcionais relacionados ao seu escopo.
* **Identificador único**: Todo requisito funcional recebe um código (por exemplo, PRJ‑WEB‑XXX), garantindo referência inequívoca em discussões, testes e rastreamento de alterações.
* **Descrição estruturada**: Para cada requisito funcional são apresentados:  
  + **Título e prioridade**, indicando importância e ordem de implementação.
  + **Fluxo principal**, descrevendo o caminho de sucesso completo.
  + **Fluxos alternativos** e **exceções**, mapeando cenários de erro e variações de uso.
  + **Critérios de aceitação**, quando aplicáveis, para guiar a validação e os testes.
* **Requisitos não funcionais** : Ao final de cada módulo, listam‑se os requisitos não funcionais relevantes, organizados em categorias como desempenho, segurança, usabilidade, disponibilidade e portabilidade.
* **Rastreabilidade** : cada requisito funcional referencia o ator responsável e o módulo correspondente, facilitando o vínculo com casos de uso, testes de aceitação e implementação de código.

# ESPECIFICAÇÃO

A especificação dos requisitos funcionais do Sistema de Gerenciamento de Laboratórios de aula foi realizada utilizando o padrão de **Requisitos Funcionais por Módulos**. Este padrão organiza os requisitos com base nas funcionalidades agrupadas em módulos lógicos do sistema, conforme definido no "Escopo do Produto". Essa abordagem facilita a compreensão da abrangência de cada parte do sistema e a rastreabilidade entre as funcionalidades e os componentes que as implementam.

## 3.1 Módulo 1

| **IDENTIFICADOR:**  PRJ-WEB-001 | **TÍTULO:** Login para acessar o dashboard |
| --- | --- |
| **PRIORIDADE:**  Essencial | **LINK EXTERNO:** |
| **Link Figma**: /Figma <[Click para acessar o projeto](https://www.figma.com/design/3qtvHu0LTJWuRp55mQMpe5/Sistema-de-Gerenciamento-de-Salas?node-id=2-2&t=YsstOP3PJXjkEiYd-1)>  **Quais categorias de usuário devem ter acesso ao software?**  alunos, professores, responsáveis por laboratórios e o servidor responsável pelo setor (Coordenação) .  **Quais informações são necessárias para realizar o login?**  O login deve ser realizado mediante a informação do usuário(matrícula, e-mail institucional ou número de registro) e senha do usuário previamente cadastrados.  **Se o usuário esquecer a senha, o que deve ser feito?**  A página de login deve possuir um link chamado "Esqueci minha senha", que encaminha o usuário para um formulário de recuperação de senha.  **A conta deve ser bloqueada após um determinado número de erros de senha?**  Na versão inicial do software, a conta não deve ser bloqueada.(Pode sofrer alterações em versões futuras)  **Qual o modelo (pattern) da senha?**  Por segurança, a senha do usuário deverá conter no mínimo 8 caracteres, com pelo menos uma letra maiuscula, um número e um símbolo especial ($, #, @ e afins).  **O que deve ser feito após a realização do login?**  Deve ser exibida a dashboard do usuário com um menu de opções/funções de acordo com nivel de hierarquia pré-cadastrado na criação do usuário.  **Fluxo principal :**   1. Usuário abre a página de login. 2. O sistema exibe formulário contendo os campos "E-mail", "Senha", botão "Entrar" e link "Esqueci minha senha". 3. O usuário preenche os campos "E-mail" e "Senha", e clica no botão "Entrar". 4. O sistema verifica se os campos "E-mail" e "Senha" seguem os padrões determinados. 5. A aplicação valida os dados do usuário e, caso tudo esteja correto, cria a sessão e exibe a dashboard específica para o perfil do usuário.   **Fluxo alternativo 1 (usuário esqueceu a senha):**   1. Se o usuário esquecer a senha, deve clicar no link "Esqueci minha senha". 2. O sistema encaminha o usuário para a página de recuperação de senha.   **Fluxo de exceção 1 (dados do usuário são inválidos):**   1. Se os dados do usuário não forem válidos, o sistema deve exibir novamente a página de login com a mensagem "Usuário e/ou senha não foram encontrados". 2. Os campos de "Login" e "Senha" devem ser marcados com uma cor diferente (vermelho, por exemplo) e preenchidos com os valores informados pelo usuário.   **Fluxo de exceção 2 (usuário novo):**   1. Se este for o primeiro acesso do usuário, sistema informa que o usuário precisa alterar sua senha 2. Sistema exibe uma tela de alteração de senha contendo os campos "Nova senha", "Confirmação de nova senha" e um botão "Atualizar senha". 3. Usuário preenche os campos conforme determinado. 4. A dashboard do usuário (conforme seu perfil) deve ser exibido somente após a alteração da senha; caso o usuário não altere a senha, seu acesso ao sistema não deve ser permitido. | |
|

| **IDENTIFICADOR:**  PRJ-WEB-001 | **TÍTULO:**Cadastro |
| --- | --- |
| **PRIORIDADE:**  Essencial | **LINK EXTERNO:** |
| **Link Figma**: /Figma <[Click para acessar o projeto](https://www.figma.com/design/3qtvHu0LTJWuRp55mQMpe5/Sistema-de-Gerenciamento-de-Salas?node-id=2-2&t=YsstOP3PJXjkEiYd-1)>  **Quais categorias de usuário devem ter acesso ao modulo?**  Servidores da Coordenação e o responsável pelos laboratórios.  **O que ocorre quando a tabela de horários já está preenchida ?**  O usuário consegue visualizar a partir de ícones os horários preenchidos e o sistema o notifica quando tenta desmarcar algum deles.(Apesar de ter total liberdade de edição sobre eles)  **O que deve ser feito na tela de cadastro?**  No dashboard do usuário com permissão de cadastro, deve constar uma opção de cadastro de salas, onde o mesmo poderá criar as salas, editá-las e excluí-las. Os horários disponíveis para cada sala também serão geridos nesta opção.  **Fluxo principal :**   1. Usuário abre a página de cadastro e seleciona a opção salas. 2. Clica sobre a opção adicionar nova sala. 3. Preenche as informações de nome, número, capacidade e horários reservados(selecionar o período de vigência e descrição da ocupação). 4. Ao clicar em salvar, a aplicação irá mostrar as salas cadastradas, com opções de visualizar, editar e excluir em cada uma.   **Fluxo alternativo 1 (usuário esqueceu a senha):**   1. Se o usuário esquecer a senha, deve clicar no link "Esqueci minha senha". 2. O sistema encaminha o usuário para a página de recuperação de senha.   **Fluxo de exceção 1 (sala já existente):**   1. Se o nome da sala a ser cadastrada já estiver no sistema, será exibido uma mensagem "Sala já existente, edite ou exclua". 2. O botão de salvar fica inoperante enquanto o nome não for diferente da sala já cadastrada. | |
|

| **IDENTIFICADOR:**  PRJ-WEB-001 | **TÍTULO:**Reservas |
| --- | --- |
| **PRIORIDADE:**  Essencial | **LINK EXTERNO:** |
| **Link Figma**: /Figma <[Click para acessar o projeto](https://www.figma.com/design/3qtvHu0LTJWuRp55mQMpe5/Sistema-de-Gerenciamento-de-Salas?node-id=2-2&t=YsstOP3PJXjkEiYd-1)>  **Quais categorias de usuário devem ter acesso ao módulo?**  Alunos, professores, responsáveis por laboratórios e o servidor responsável pelo setor (Coordenação) .  **Quais informações são necessárias para fazer uma reserva ?**  Sala, professor responsável, horário e nome do projeto.  **Se o horário já estiver preenchido?**  Deve-se buscar um horário alternativo livre, não é permitido pegar horários já agendados/indisponíveis.  **Fluxo principal :**   1. Usuário abre a página de reservas. 2. Seleciona uma das salas disponíveis. 3. Procura um horário livre. 4. Clica sobre o horário para selecionalo.. 5. Preenche as informações de nome do projeto, professor responsável e clica em solicitar agendamento 6. Na mesma tela, terá disponível em uma seção agendamentos, com informações de status, horário e uma opção de imprimir comprovante.. | |
|

| **IDENTIFICADOR:**  PRJ-WEB-001 | **TÍTULO:**Usuários e Permissões |
| --- | --- |
| **PRIORIDADE:**  Essencial | **LINK EXTERNO:** |
| **Link Figma**: /Figma <[Click para acessar o projeto](https://www.figma.com/design/3qtvHu0LTJWuRp55mQMpe5/Sistema-de-Gerenciamento-de-Salas?node-id=2-2&t=YsstOP3PJXjkEiYd-1)>  **Quais categorias de usuário devem ter acesso ao módulo?**  Apenas a coordenação.  **Quais informações são necessárias para fazer uma reserva ?**  Sala, professor responsável, horário e nome do projeto.  **Se o horário já estiver preenchido?**  Deve-se buscar um horário alternativo livre, não é permitido pegar horários já agendados/indisponíveis.  **Fluxo principal :**   1. Usuário abre o dashboard. 2. Seleciona a opção de Usuários e permissões. 3. Faz uma pesquisa por o usuário que interessa. 4. Clica sobre o dropdown de opções de níveis de acesso. 5. Seleciona o nível de acesso que deve dar ao usuário.(Módulos que ficaram disponíveis). 6. Na mesma tela, terá disponível em uma seção agendamentos, com informações de status, horário e uma opção de imprimir comprovante..   **Fluxo alternativo 1 (usuário esqueceu a senha):**   1. O usuário pode utilizar a barra de filtro para selecionar usuários de determinado grupo de permissões. 2. Uma vez filtrados(em exibição na tela), é possível imprimilos clicando em imprimir. | |
|

| **IDENTIFICADOR:**  PRJ-WEB-001 | **TÍTULO:**Notificações |
| --- | --- |
| **PRIORIDADE:**  Essencial | **LINK EXTERNO:** |
| **Link Figma**: /Figma <[Click para acessar o projeto](https://www.figma.com/design/3qtvHu0LTJWuRp55mQMpe5/Sistema-de-Gerenciamento-de-Salas?node-id=2-2&t=YsstOP3PJXjkEiYd-1)>  **Quais categorias de usuário devem ter acesso ao módulo?**  Professores e níveis acima.  **Como funciona a autorização ?**  O usuário recebe uma notificação que pode ser de informação ou solicitação, em caso de notificação, será disponível apenas um botão de ok para confirmar o recebimento, já em casos de solicitação, os botões de autorizar ou não autorizar ficarão à disposição.  **Se a data tivesse passado ?**  Automaticamente a notificação de solicitação vira uma de informação.  **Fluxo principal :**   1. Usuário abre o dashboard. 2. Seleciona a opção de Notificações. 3. Filtra a notificação de seu interesse.. 4. Clica sobre um dos botões. | |
|

# INTERFACES EXTERNAS

Nesta seção são descritas as interfaces que o sistema possui com outros softwares, pessoas ou hardwares.

## 4.1 Interfaces de software

* **Back-end (Node.js e Express):** Será desenvolvido com Node.js e o framework Express para lidar com a lógica do servidor, banco de dados e APIs. Isso significa que o front-end interage com o back-end através das APIs construídas com Express e Node.js.
* **Front-end (HTML5, CSS3, JavaScript com Bootstrap):** O front-end, que é a parte com a qual o usuário interage diretamente, será desenvolvido com HTML5, CSS3 e JavaScript, utilizando Bootstrap para padronizar o design e a responsividade. Isso implica que estas são as interfaces de software que interagem com o navegador web do usuário.
* **Banco de Dados (MySQL):** O MySQL será o sistema de gerenciamento de banco de dados relacional usado para armazenar informações de usuários, salas e reservas. A interação entre o back-end e o MySQL é uma interface de software crucial.
* **JWT (JSON Web Tokens):** Será utilizado para autenticação no sistema, verificando a identidade do usuário e as permissões de acesso. Isso é um padrão de comunicação e segurança que o sistema utiliza.

## 4.2 Interfaces de usuário

* **Aplicação Web:** O sistema será uma aplicação web, acessível via navegador web. Isso significa que a principal interface de usuário será uma página da web.
* **Tecnologias Front-end:** A interface gráfica será desenvolvida com HTML5, CSS3 e JavaScript, utilizando Bootstrap para padronizar o design e a responsividade.
* **Interfaces Visuais:** O Módulo de Consulta de Disponibilidade, por exemplo, oferecerá interfaces visuais como calendário e lista para que os usuários possam verificar a ocupação das salas em tempo real.
* **Dashboard Analítico:** O Módulo de Relatórios e Análises será um dashboard analítico que fornecerá dados para apoio à tomada de decisões administrativas. Embora não seja uma interface de entrada de dados primária, é uma interface de visualização importante para certos tipos de usuários.
* **Fluxo de Login:** O sistema terá uma página de login com campos para e-mail e senha, e um link para recuperação de senha. Após o login, a dashboard do usuário será exibida.
* **Diferentes Níveis de Acesso:** O sistema definirá diferentes níveis de acesso (e.g., usuário comum, administrador), o que implica que a interface de usuário poderá se adaptar e exibir diferentes funcionalidades e informações dependendo do perfil do usuário.

## 

## 4.3 Interfaces de hardware

Não há interação direta do Sistema de Gerenciamento de Laboratórios (SGL) com hardware específico.

* **Ausência de Interação Direta com Hardware Específico:** O Sistema de Gerenciamento de Laboratórios (SGL) é concebido como uma aplicação web e, em sua versão inicial, não requer ou possui interfaces diretas com hardware dedicado ou específico, como pin pads para pagamentos, leitores de código de barras para identificação, impressoras térmicas para emissão de comprovantes, ou outros dispositivos periféricos especializados. A arquitetura do sistema é focada em uma interação totalmente via software, mediada por navegadores web.
* **Dependência de Hardware Padrão do Usuário para Acesso:** A funcionalidade do sistema dependerá exclusivamente do hardware padrão que o usuário final possui para acessar a internet e navegar em aplicações web. Isso inclui:
* **Computadores (Desktops e Notebooks):** Utilizados para acessar o sistema através de navegadores web instalados.
* **Dispositivos Móveis (Smartphones e Tablets):** Igualmente capazes de acessar o sistema através de seus respectivos navegadores, aproveitando a responsividade da interface desenvolvida com Bootstrap. A única exigência de hardware para a interação com o SGL é que o dispositivo possua um navegador web compatível e uma conexão de internet estável. O sistema não impõe requisitos de hardware adicionais além desses para sua operação

## 

# VERIFICAÇÃO

# 

# A verificação do cumprimento dos requisitos após a implementação será realizada de forma abrangente, combinando abordagens manuais e automatizadas, e integrando-as a um processo de desenvolvimento contínuo para garantir a qualidade e a conformidade do sistema.

#### **1. Estratégia de Testes**

# Serão desenvolvidos testes em diferentes níveis para validar as funcionalidades e requisitos do sistema:

# Testes de Unidade: Serão realizados para verificar a menor parte testável de uma aplicação, como métodos ou funções individuais no back-end (Node.js/Express) e componentes específicos no front-end (JavaScript). Estes testes serão automatizados e escritos pelos próprios desenvolvedores durante o processo de codificação.

# Testes de Integração: Destinam-se a verificar se diferentes módulos ou componentes do sistema funcionam corretamente quando combinados. Isso incluirá a interação entre o front-end e o back-end, e a comunicação do back-end com o banco de dados (MySQL). Estes testes também serão automatizados.

# Testes de Sistema: Irão validar o sistema como um todo, garantindo que ele atenda a todos os requisitos funcionais e não-funcionais especificados. Cenários de uso completos, envolvendo a interação do usuário com a interface, o fluxo de agendamento, notificações, etc., serão testados. Uma parte significativa desses testes será automatizada, mas também haverá testes manuais para cenários mais complexos e validação da experiência do usuário.

# Testes de Aceitação do Usuário (UAT - User Acceptance Testing): Serão conduzidos com a participação de usuários finais (administradores do Campus Mucambinho, professores/servidores que agendam salas) para garantir que o sistema atende às suas necessidades e expectativas. Estes testes serão predominantemente manuais e baseados em casos de uso reais.

#### **2. Automação de Testes**

# A maior parte dos testes (unidade e integração) será automatizada. Para o back-end, serão utilizadas bibliotecas de teste como Mocha/Chai ou Jest. Para o front-end, frameworks como Jest ou React Testing Library (se for o caso de um framework como React ser adotado posteriormente para o front-end) serão considerados para componentes e interações.

# Ferramentas de automação de testes de sistema, como Cypress ou Playwright, serão avaliadas para simular interações do usuário na interface web e validar fluxos completos.

#### **3. Pipeline CI/CD (Integração Contínua/Entrega Contínua)**

# Será implementada uma pipeline de CI/CD para automatizar o processo de construção, teste e implantação do sistema.

# Cada vez que o código for enviado para o repositório de controle de versão (e.g., Git), a pipeline será acionada automaticamente, executando os testes automatizados (unidade e integração).

# Se os testes passarem, o código será integrado e, dependendo da configuração da pipeline, poderá ser automaticamente implantado em um ambiente de desenvolvimento ou staging para testes adicionais (incluindo testes de sistema automatizados e manuais).

# A implementação de uma pipeline CI/CD visa garantir que bugs sejam identificados e corrigidos precocemente, que a qualidade do código seja mantida e que as novas funcionalidades possam ser entregues de forma mais rápida e confiável. Ferramentas como GitHub Actions ou GitLab CI/CD serão consideradas para a implementação da pipeline.

#### **4. Testes Manuais**

# Embora a automação seja priorizada, os testes manuais serão cruciais para cenários que exigem avaliação humana, como:

# Usabilidade e Experiência do Usuário (UX): Avaliar a intuição da interface, o fluxo de navegação e a facilidade de uso do sistema.

# Testes Exploratórios: Realizar testes sem roteiro pré-definido para descobrir falhas inesperadas ou comportamentos incomuns.

# Testes de Regressão Críticos: Mesmo com testes automatizados, algumas funcionalidades críticas podem ter um teste manual de regressão para maior segurança.

# Validação de Relatórios: A precisão e a clareza dos relatórios gerados (mensal e anual) serão verificadas manualmente para garantir que os dados apresentados são consistentes e úteis para a tomada de decisões.

# Em resumo, a estratégia de verificação combinar a eficiência da automação de testes em uma pipeline CI/CD com a percepção crítica dos testes manuais, garantindo que o sistema não apenas funcione conforme especificado, mas também seja robusto, intuitivo e atenda plenamente às necessidades da UFC Sobral.

# 

# ANEXOS

Proposta Inicial : [Link01](https://docs.google.com/document/d/1dLSPmC-rZe2RJDYUYg5_mJ4qlXRftn7iQ3N5UqK5NUk/edit?usp=sharing)

Prototipagem das Telas : [Link02](https://www.figma.com/design/3qtvHu0LTJWuRp55mQMpe5/Sistema-de-Gerenciamento-de-Salas?node-id=2-2&t=YsstOP3PJXjkEiYd-1)







