

Innovation OASIS

Plano de teste

Time

Kayllane Justino - 202211010154

Gustavo Souto - 202211010137

Guilherme Firpo - 202211010320

Arthur Willyams - 202211010080

Maio
2024

Histórico de Revisões

Data	Versão	Descrição	Autor

Sumário

1. Introdução	5
1.1 Objetivos	5
2. Escopo	5
2.1 Definição do Produto	5
2.2 Identificação de Projeto	5
3. Requisitos a Testar	6
3.1 Teste do Banco de Dados	6
3.2 Teste Funcional	6
3.3 Teste do Ciclo de Negócios	6
3.4 Teste da Interface do Usuário	6
3.5 Perfil da Performance	7
3.6 Teste de Carga	7
3.7 Teste de Stress	7
3.8 Teste de Volume	7
3.9 Teste de Segurança e de Controle de Acesso	7
3.10 Teste de Falha/Recuperação	8
3.11 Teste de Instalação	8
4. Estratégia de Teste	8
4.1 Tipos de Teste	8
4.1.1 Teste de Integridade de Dados e do Banco de Dados	8
4.1.2 Teste de Função	8
4.1.3 Teste da Interface do Usuário	8
4.1.4 Teste de Performance	9
4.1.5 Teste de Carga	9
4.1.6 Teste de Segurança e Controle de Acesso	9
4.1.7 Teste de Instalação	10
4.2 Ferramentas	10
5. Recursos	10
5.1 Trabalhadores	10
5.2 Sistema	11
6. Cronograma	11

1. INTRODUÇÃO

1.1 Objetivos

Assegurar que todas as funcionalidades do aplicativo de reserva de salas "InnovationOasis" estejam funcionando conforme especificado nos requisitos.

Garantir a qualidade e produtividade para indivíduos, desde Pessoa Física à jurídica, em um ambiente de trabalho apropriado. Garantindo uma experiência do cliente satisfatória

2. ESCOPO

Critérios de Aceitação:

- Todos os campos de entrada devem validar corretamente os formatos e restrições.
- Os usuários devem ser capazes de recuperar a senha sem falhas.
- As reservas devem refletir precisamente as preferências do usuário e a disponibilidade atual das salas.
- Os usuários devem poder cadastrar locais sem erros e com feedback adequado.

Áreas de Teste Incluídas:

- Cadastro de Usuário:
 - Testar tanto o cadastro de pessoas físicas quanto de empresas, verificando a validação de campos como nome, email, CPF/CNPJ, telefone e senha.
- Login e Recuperação de Senha:
 - Testar os mecanismos de login e de recuperação de senha, garantindo a segurança e a eficácia do acesso dos usuários.
- Reserva de Salas:
 - Testar o processo completo de reserva de salas, desde a seleção de filtros (localização, tamanho, capacidade) até a confirmação da reserva.
- Cadastro de Local:
 - Verificar a funcionalidade de cadastro de um novo local, incluindo a inclusão de mídias e descrições.

Áreas de Teste Excluídas:

- Testes de performance e carga em larga escala podem ser excluídos do escopo inicial, focando-se inicialmente em funcionalidade e usabilidade.
- Integração com sistemas de terceiros não especificados nas informações fornecidas (por exemplo, sistemas de pagamento ou redes sociais).

Reserva de Salas:

- Testar o processo completo de reserva de salas, desde a seleção de filtros (localização, tamanho, capacidade) até a confirmação da reserva.
-

Cadastro de Local:

- Verificar a funcionalidade de cadastro de um novo local, incluindo a inclusão de mídias e descrições.

Ferramentas e Tecnologias de Teste:

- Considerar o uso de ferramentas para automação de testes com Selenium para testes de interface.
- Usar pytest com pytest-django para testar requisições e respostas do servidor
- Utilizar frameworks como pytest para testar funções e classes isoladamente.
- Simular interações com o banco de dados usando pytest-mock para evitar operações de I/O durante os testes.
- Validar restrições e triggers para garantir que as regras de negócio sejam respeitadas
- Usar Jest para testar componentes React isoladamente.
- Implementar testes com React Testing Library para garantir que os componentes renderizem corretamente.

1.1 Definição do Produto

Consiste numa aplicação que fornece ao usuário o livre acesso a poltronas e salas de estudos, proporcionando-lhes um ambiente estruturado e de alta qualidade.

1.2 Identificação de Projeto

A tabela abaixo identifica a documentação e disponibilidade usados para desenvolver o plano de testes:

Documento	Criado ou Disponível	Recebido ou Revisado
Especificação de Requisitos	■ Sim □ Não	■ Sim □ Não
Plano de Projeto	■ Sim □ Não	■ Sim □ Não
Modelo de Análise	■ Sim □ Não	■ Sim □ Não
Modelo de Projeto	■ Sim □ Não	■ Sim □ Não
Documento de Arquitetura	■ Sim □ Não	■ Sim □ Não
Protótipo	□ Sim ■ Não	□ Sim ■ Não
Manual do Usuário	□ Sim ■ Não	□ Sim ■ Não
Lista de Riscos	■ Sim □ Não	■ Sim □ Não

3. REQUISITOS A TESTAR

A lista a seguir identifica os elementos específicos: casos de uso, requisitos funcionais e não funcionais. Estes itens foram selecionados como alvos de teste e representam o escopo das atividades de teste.

3.1 Teste do Banco de Dados

- Verificar se as informações do usuário podem ser cadastradas, consultadas e removidas no sistema..
- Verificar se as informações podem ser inseridas, atualizadas e consultadas no sistema.
- Verificar se as informações úteis obtidas pelo subsistema responsável podem ser atualizadas e apresentadas corretamente.
- Verificar se as informações específicas de cada usuário podem ser acompanhadas no sistema.
- Verificar se o sistema pode cadastrar, atualizar e remover um novo cliente de seu banco de dados.
- Verificar se o sistema pode remover categorias de salas cadastrados conforme necessário.

3.2 Teste Funcional

- Verificar se as informações úteis obtidas pelo subsistema responsável são atualizadas automaticamente e de forma periódica.
- Verificar se qualquer usuário pode acessar sua própria conta por meio de login e senha.
- Verificar se o relatório da conta do cliente é preciso e exato.
- Verificar se as informações podem ser acessadas em qualquer um dos idiomas disponíveis no sistema.

3.3 Teste do Ciclo de Negócios

- Verificar a integração e funcionamento correto de todas as etapas dos processos de negócio essenciais.
- Verificar que todas as transações comerciais se completam sem erros, cobrindo cenários típicos e atípicos de uso.
- Verificar a continuidade dos processos de negócio mesmo em face de condições adversas ou entradas inesperadas.

3.4 Teste da Interface do Usuário

- Verificar a clareza e a funcionalidade dos componentes de interface do usuário em diferentes dispositivos e resoluções.
 - Verificar a acessibilidade da interface para usuários com diferentes necessidades.
 - Verificar a consistência da interface em toda a aplicação, assegurando uma experiência do usuário coesa.
-

3.5 Perfil da Performance

- Verificar os tempos de resposta das funcionalidades sob condições normais de uso.
- Verificar o consumo de recursos do sistema, como memória e CPU, durante operações críticas.
- Verificar a eficiência de execução de consultas ao banco de dados e outras operações de I/O.

3.6 Teste de Carga

- Verificar a resposta do sistema com 10 usuários.
- Verificar a resposta do sistema com 50 usuários.
- Verificar a resposta do sistema com 100 usuários.
- Verificar a resposta do sistema com 200 usuários.
- Verificar a resposta do sistema com 500 usuários.

3.7 Teste de Stress

- Verificar a capacidade do sistema de operar sob carga extrema por períodos prolongados.
- Verificar a estabilidade do sistema quando submetido a picos súbitos de carga.
- Verificar as respostas do sistema e a geração de logs em condições de stress.

3.8 Teste de Volume

- Verificar o comportamento do sistema quando aumentado progressivamente o volume de dados armazenados e processados.
- Verificar a escalabilidade do sistema para suportar crescimento contínuo dos dados.
- Verificar a integridade dos dados após operações em grande escala.

3.9 Teste de Segurança e de Controle de Acesso

- Verificar se apenas usuários cadastrados podem acessar informações restritas aos cadastrados no mercado eletrônico.
 - Verificar se somente o administrador tem permissão para inserir, atualizar ou remover dados do sistema.
 - Verificar se os clientes visualizam apenas suas próprias informações de conta e reservas de salas.
 - Verificar se os usuários do sistema podem acessar apenas as funcionalidades e dados associados ao seu próprio tipo de perfil de usuário.
 - Verificar se as atualizações do sistema podem ser realizadas apenas a partir de fontes autorizadas dentro da plataforma..
-

3.10 Teste de Falha/Recuperação

- Verificar a capacidade do sistema de se recuperar automaticamente após falhas de hardware ou software.
- Verificar a eficácia dos procedimentos de backup e restauração de dados.
- Verificar os mecanismos de alerta e notificação em caso de falhas.

3.11 Teste de Instalação

- Verificar a facilidade e a robustez do processo de instalação do sistema em diversos ambientes.
- Verificar que todas as dependências do sistema são resolvidas durante a instalação.
- Verificar a capacidade de desinstalação completa do sistema sem deixar resíduos.

4. ESTRATÉGIA DE TESTE

4.1 Tipos de Teste

Nota: As transações a seguir referem-se às "transações lógicas de negócio". Essas transações representam funções específicas que um usuário final do sistema é esperado para executar ao utilizar a aplicação, como adicionar/modificar determinadas informações.

4.1.1 TESTE DE INTEGRIDADE DE DADOS E DO BANCO DE DADOS

Objetivo do Teste:	Verificar a integridade dos dados armazenados no banco de dados da aplicação que gerencia poltronas e sala de estudos
Técnica:	<p>Inserir dados de teste: Criar um conjunto de dados de teste que representam diferentes casos de uso da aplicação.</p> <p>Executar consultas de validação: Executar consultas SQL que verifiquem se os dados armazenados no banco de dados estão corretos e consistentes.</p> <p>Comparar resultados: Comparar os resultados das consultas de validação com os dados de teste esperados.</p>
Critério de Finalização:	O teste será considerado bem-sucedido se todos os resultados das consultas de validação forem iguais aos dados de teste esperados.

Considerações Especiais:	<p>Os dados de teste devem ser cuidadosamente criados para garantir que todos os casos de uso da aplicação sejam testados.</p> <p>As consultas de validação devem ser abrangentes e verificar todos os aspectos dos dados armazenados no banco de dados.</p> <p>Os resultados das consultas de validação devem ser documentados e armazenados para referência futura.</p>
---------------------------------	---

1.1.1 TESTE DE FUNÇÃO

Objetivo do Teste:	Verificar se a aplicação que gerencia poltronas e sala de estudos está funcionando corretamente de acordo com seus requisitos funcionais.
Técnica:	<p>Análise de Requisitos: Revisar os requisitos funcionais da aplicação para identificar as funcionalidades que devem ser testadas.</p> <p>Criação de Casos de Teste: Criar casos de teste que representem cada um dos requisitos funcionais da aplicação.</p> <p>Execução de Testes: Executar os casos de teste na aplicação e registrar os resultados.</p> <p>Análise de Resultados: Analisar os resultados dos testes para identificar qualquer falha ou defeito na aplicação.</p>
Critério de Finalização:	O teste será considerado bem-sucedido se todos os casos de teste forem executados com sucesso e se não forem encontrados falhas ou defeitos na aplicação.
Considerações Especiais:	<p>Os casos de teste devem ser abrangentes e testar todas as funcionalidades da aplicação.</p> <p>Os testes devem ser realizados em diferentes ambientes, como diferentes navegadores web ou dispositivos móveis.</p> <p>Os resultados dos testes devem ser documentados e armazenados para referência futura.</p>

4.1.2 TESTE DA INTERFACE DO USUÁRIO

Objetivo do Teste:	O objetivo principal deste teste é avaliar a usabilidade e a experiência do usuário ao interagir com a interface da aplicação de gerenciamento de poltronas e salas de estudo. Isso significa verificar se a interface é clara, intuitiva, fácil de usar e atende às expectativas dos usuários.
---------------------------	---

Técnica:	<p>Testes de Usabilidade: Envolvem a observação direta de usuários reais enquanto eles interagem com a interface. Isso pode ser feito através de testes presenciais ou remotos.</p> <p>Testes Heurísticos: Utilizam um conjunto de princípios de usabilidade para avaliar a interface.</p> <p>Inspeção de Walk-through: Simula o fluxo de trabalho de um usuário na interface para identificar problemas de usabilidade.</p> <p>Testes de Card Sorting: Permitem avaliar a organização e a rotulagem da interface.</p> <p>Questionários de Usabilidade: Coletam feedback dos usuários sobre a interface.</p>
Critério de Finalização:	<p>Taxa de sucesso: A porcentagem de usuários que podem concluir as tarefas com sucesso.</p> <p>Tempo de Tarefa: O tempo que os usuários levam para concluir as tarefas.</p> <p>Satisfação do Usuário: O nível de satisfação dos usuários com a interface.</p> <p>Número de Erros: O número de erros que os usuários cometem ao usar a interface.</p> <p>Facilidade de Uso: A facilidade com que os usuários podem aprender a usar a interface</p>
Considerações Especiais:	<p>O teste será realizado com usuários reais que nunca usaram a aplicação antes.</p> <p>O teste será realizado em um ambiente de estudo simulado.</p> <p>A funcionalidade de reserva de poltrona será testada para diferentes tipos de poltronas e horários.</p>

4.1.3 TESTE DE PERFORMANCE

Objetivo do Teste:	<p>O objetivo principal deste teste é avaliar o desempenho da aplicação de gerenciamento de poltronas e salas de estudo sob carga. Isso significa verificar se a aplicação pode suportar um número elevado de usuários e solicitações sem apresentar falhas ou lentidão significativa.</p>
---------------------------	--

Técnica:	<p>Testes de Carga: Simulam um grande número de usuários simultâneos acessando a aplicação para avaliar o tempo de resposta e a capacidade de processamento da aplicação.</p> <p>Testes de Estresse: Submetem a aplicação a um grande volume de solicitações por um período prolongado para identificar pontos de falha e gargalos de desempenho.</p> <p>Testes de Desempenho: Medem o tempo de resposta da aplicação para diferentes operações e funcionalidades.</p> <p>Monitoramento de Desempenho: Coletam dados sobre o desempenho da aplicação em tempo real para identificar problemas de desempenho antes que eles causem impacto aos usuários.</p>
Critério de Finalização:	<p>Tempo de resposta: O tempo que a aplicação leva para responder a solicitações dos usuários.</p> <p>Taxa de Erro: A porcentagem de solicitações que falham.</p> <p>Capacidade de Processamento: O número máximo de usuários simultâneos que a aplicação pode suportar.</p> <p>Uso de Recursos: O uso de CPU, memória e outros recursos do sistema pela aplicação.</p>
Considerações Especiais:	<p>Carga de Trabalho Realística: O teste deve simular uma carga de trabalho realística que a aplicação enfrenta em um ambiente de produção.</p> <p>Infraestrutura: O teste deve ser realizado em uma infraestrutura similar à que será utilizada em produção.</p> <p>Tipos de Usuários: O teste deve considerar os diferentes tipos de usuários que utilizarão a aplicação e suas respectivas demandas de desempenho.</p> <p>Impacto nos Usuários: O teste deve ser realizado de forma a minimizar o impacto aos usuários reais da aplicação.</p>

4.1.4 TESTE DE CARGA

Objetivo do Teste:	<p>O objetivo principal deste teste é avaliar o desempenho da aplicação de gerenciamento de poltronas e salas de estudo sob carga. Isso significa verificar se a aplicação pode suportar um número elevado de usuários e solicitações sem apresentar falhas ou lentidão significativa.</p>
---------------------------	--

Técnica:	<p>Testes de Carga Simulados: Simulam um grande número de usuários simultâneos acessando a aplicação para avaliar o tempo de resposta e a capacidade de processamento da aplicação. Essa técnica pode ser realizada utilizando ferramentas de teste de carga como o LoadRunner, JMeter ou Taurus.</p> <p>Testes de Estresse: Submetem a aplicação a um grande volume de solicitações por um período prolongado para identificar pontos de falha e gargalos de desempenho. Essa técnica pode ser útil para identificar problemas que podem ocorrer em cenários de uso extremos.</p> <p>Testes de Inundação: Inundam a aplicação com um grande número de solicitações simultâneas para avaliar a capacidade da aplicação de lidar com picos de tráfego. Essa técnica pode ser útil para avaliar o desempenho da aplicação em situações de alta demanda.</p>
Critério de Finalização:	<p>Tempo de resposta: O tempo que a aplicação leva para responder a solicitações dos usuários.</p> <p>Taxa de Erro: A porcentagem de solicitações que falham.</p> <p>Capacidade de Processamento: O número máximo de usuários simultâneos que a aplicação pode suportar.</p> <p>Uso de Recursos: O uso de CPU, memória e outros recursos do sistema pela aplicação.</p> <p>Satisfação do Usuário: A percepção dos usuários sobre o desempenho da aplicação.</p>
Considerações Especiais:	<p>Carga de Trabalho Realística: O teste deve simular uma carga de trabalho realística que a aplicação enfrenta em um ambiente de produção. Isso inclui considerar o número de usuários simultâneos, o tipo de solicitações realizadas e a distribuição das solicitações ao longo do dia.</p> <p>Infraestrutura: O teste deve ser realizado em uma infraestrutura similar à que será utilizada em produção. Isso inclui considerar o hardware, o software e a rede que serão utilizados para hospedar a aplicação.</p> <p>Tipos de Usuários: O teste deve considerar os diferentes tipos de usuários que utilizarão a aplicação e suas respectivas demandas de desempenho. Isso pode incluir usuários que acessam a aplicação a partir de diferentes dispositivos, como computadores, tablets e smartphones.</p> <p>Impacto nos Usuários: O teste deve ser realizado de forma a minimizar o impacto aos usuários reais da aplicação. Isso pode ser feito realizando o teste em um ambiente de teste separado ou durante períodos de menor atividade.</p>

4.1.5 TESTE DE SEGURANÇA E CONTROLE DE ACESSO

Objetivo do Teste:	O objetivo principal deste teste é avaliar a segurança e o controle de acesso da aplicação de gerenciamento de poltronas e salas de estudo. Isso significa verificar se a aplicação está protegida contra acessos não autorizados, ataques cibernéticos e outras violações de segurança.
Técnica:	<p>Testes de Penetração: Envolvem a simulação de ataques reais à aplicação para identificar vulnerabilidades de segurança.</p> <p>Análise Estática de Código: Examina o código-fonte da aplicação para identificar falhas de segurança.</p> <p>Análise Dinâmica de Aplicação: Monitora o comportamento da aplicação em execução para identificar vulnerabilidades de segurança.</p> <p>Testes de Autenticação e Autorização: Verificam se a aplicação autentica e autoriza os usuários corretamente.</p> <p>Testes de Controle de Acesso: Verificam se a aplicação controla o acesso aos recursos da aplicação de forma adequada.</p>
Critério de Finalização:	<p>Ausência de Vulnerabilidades: A aplicação não deve apresentar nenhuma vulnerabilidade de segurança crítica.</p> <p>Funcionamento Correto da Autenticação e Autorização: A aplicação deve autenticar e autorizar os usuários corretamente.</p> <p>Controle de Acesso Adequado: A aplicação deve controlar o acesso aos recursos da aplicação de forma adequada.</p> <p>Conformidade com Leis e Regulamentos: A aplicação deve estar em conformidade com todas as leis e regulamentos relevantes relacionados à segurança de dados.</p>
Considerações Especiais:	<p>O teste será realizado utilizando uma ferramenta de análise estática de código.</p> <p>O teste será realizado em todo o código-fonte da aplicação.</p> <p>O impacto aos usuários reais da aplicação será minimizado durante o teste.</p>

4.1.6 TESTE DE INSTALAÇÃO

Objetivo do Teste:	O objetivo principal deste teste é verificar se a aplicação de gerenciamento de poltronas e salas de estudo pode ser instalada com sucesso em diferentes ambientes e dispositivos. Isso significa garantir que a instalação seja fácil, rápida e livre de erros.
---------------------------	--

Técnica:	<p>Testes de Instalação Manual: Envolvem a instalação da aplicação manualmente em diferentes ambientes e dispositivos para identificar problemas de instalação.</p> <p>Testes de Instalação Automatizados: Utilizam ferramentas de automação para instalar a aplicação em diferentes ambientes e dispositivos de forma rápida e eficiente.</p> <p>Testes de Compatibilidade: Verificam se a aplicação é compatível com diferentes sistemas operacionais, hardwares e softwares.</p> <p>Testes de Desinstalação: Verificam se a aplicação pode ser desinstalada com sucesso sem deixar arquivos ou configurações residuais.</p>
Critério de Finalização:	<p>Sucesso da Instalação: A aplicação deve ser instalada com sucesso em todos os ambientes e dispositivos testados.</p> <p>Facilidade de Instalação: A instalação da aplicação deve ser fácil e intuitiva.</p> <p>Tempo de Instalação: O tempo de instalação da aplicação deve ser razoável.</p> <p>Ausência de Erros: A instalação da aplicação deve ser livre de erros.</p> <p>Compatibilidade: A aplicação deve ser compatível com todos os sistemas operacionais, hardwares e softwares testados.</p> <p>Desinstalação Completa: A aplicação deve ser desinstalada com sucesso sem deixar arquivos ou configurações residuais.</p>
Considerações Especiais:	<p>O teste será realizado em um computador com SO Windows 10.</p> <p>O usuário que realizará o teste não tem conhecimentos técnicos avançados.</p> <p>A aplicação será instalada usando o arquivo de instalação baixado do site.</p>

4.2 Ferramentas

- **Fase de Desenvolvimento Inicial**

- **React**

- **Função:** Biblioteca JavaScript para construir interfaces de usuário.
 - **Uso:** Utilizado no desenvolvimento front-end para criar componentes reativos e dinâmicos.

- **Django**

- **Função:** Framework de alto nível para Python que promove desenvolvimento rápido e design limpo e pragmático.
 - **Uso:** Utilizado no back-end para estruturar o servidor, modelos e a lógica de negócios.

- **SQL Server**

- **Função:** Sistema de gerenciamento de banco de dados relacional.
 - **Uso:** Armazenamento de dados, execução de transações e consultas.

- **PyTest**

- **Função:** Framework para escrever testes simples e escaláveis em Python.
 - **Uso:** Testar componentes do back-end, garantindo que a lógica de

negócios funcione como esperado.

- **Fase de Desenvolvimento e Teste Intermediário**

- **Django REST Framework**

- **Função:** Kit de ferramentas poderoso e flexível para construir APIs web.
 - **Uso:** Utilizado em conjunto com Django para criar interfaces API RESTful que facilitam a comunicação entre o back-end e o front-end.

- **Postman**

- **Função:** Plataforma para desenvolvimento API que simplifica cada etapa do ciclo de vida da API e permite testar manualmente as APIs.
 - **Uso:** Testar e verificar as APIs do Django REST, assegurando que respondam corretamente às requisições do React.

- **React Testing Library**

- **Função:** Conjunto de utilitários para testar componentes React de forma mais consistente.
 - **Uso:** Testar componentes front-end, garantindo que funcionem corretamente em diferentes cenários.

- **Fase de Implantação e Monitoramento**

- **Docker**

- **Função:** Plataforma de contêineres que permite empacotar aplicações dentro de contêineres.
 - **Uso:** Containerizando a aplicação para facilitar implantações consistentes e isoladas em qualquer ambiente.

- **Sentry**

- **Função:** Ferramenta de monitoramento de aplicativos que ajuda a rastrear bugs e problemas em tempo real.
 - **Uso:** Monitorar a aplicação em produção, identificando e resolvendo exceções e erros rapidamente.

- **Fase de Escalabilidade e Manutenção**

- **Kubernetes**

- **Função:** Sistema de orquestração de contêineres para automatizar a implantação, escalonamento e operações de aplicações em contêineres.
 - **Uso:** Gerenciar o escalonamento e a disponibilidade do serviço, especialmente útil à medida que a demanda pela aplicação cresce.

- **Jenkins**

- **Função:** Ferramenta de integração e entrega contínua para automatizar várias fases do desenvolvimento de software.
 - **Uso:** Automatizar o pipeline de CI/CD, desde testes até a implantação, assegurando agilidade e redução de erros.

- **Elasticsearch**

- **Função:** Motor de busca e análise distribuído para todos os tipos de dados, incluindo textuais, numéricos, geoespaciais,
-

estruturados e não estruturados.

- **Uso:** Analisar grandes volumes de logs gerados pela aplicação, ajudando na identificação de padrões ou problemas.

5. RECURSOS

5.1 Trabalhadores

Esta tabela apresenta as suposições relacionadas ao recrutamento para o projeto.

Recursos Humanos		
Funções	Usuário	Responsabilidades Específicas ou Comentários
<ul style="list-style-type: none"> Gerente de Teste; Gerente do Projeto de Teste 	<ul style="list-style-type: none"> Gustavo Lima Maria Paula 	Fornece supervisionamento gerencial. Responsabilidades: <ul style="list-style-type: none"> provê direcionamento técnico adquire recursos apropriados fornece relatórios de gerenciamento
<ul style="list-style-type: none"> Testador 	<ul style="list-style-type: none"> Renato Limb Monica lima 	Executa os testes. Responsabilidades: <ul style="list-style-type: none"> executar os testes registrar os resultados reestabelecer-se dos erros documentar solicitações de mudança
<ul style="list-style-type: none"> Recepcionista 	<ul style="list-style-type: none"> Sheila Melo 	Será a pessoa que irá administrar o sistema durante o uso: <ul style="list-style-type: none"> Irá ver se a sala está realmente reservada assim que o usuário chegar no local E será responsável por disponibilizar a sala após a saída dos usuários

5.2 Sistema

A tabela seguinte expõe os recursos do sistema para o projeto de teste.

Recursos do Sistema
Servidor de Banco de Dados <ul style="list-style-type: none"> SQL Server
Terminais Clientes <ul style="list-style-type: none"> 2 PCs (conectados via LAN) 1 PC com tela sensível ao toque (conectado a uma LAN e à internet)

Repositório de Testes:

- 1 servidor de testes dedicado
- 3 estações de trabalho para execução de testes automatizados
- Acesso a uma variedade de dispositivos móveis para teste de compatibilidade
- Licenças de software de teste, com Selenium
- Ambiente de virtualização para criação de ambientes de teste isolados: Utilizando tecnologias como Docker e possivelmente Kubernetes para gerenciar e isolar ambientes de teste.
- Ferramentas de gestão de testes, como Jira ou TestRail
- Servidor de banco de dados para armazenamento de resultados de testes e dados de teste: SQL Server
- Acesso a ferramentas de monitoramento de desempenho e ferramentas de análise de cobertura de código: Ferramentas como PyTest para testes e cobertura no backend, e React Testing Library para o frontend, complementadas por ferramentas de CI/CD como Jenkins para automação e monitoramento.

6. CRONOGRAMA

Milestone	Data de Início	Data de Término
• Negócio/Solução Digital	15/04/2024	01/05/2024
• Plano de Teste e Projeto de Teste	01/05/2024	15/05/2024
• MVP (Minimum Viable Product)	015/05/2024	22/05/2024