

**TCHUU-TCHUU**  
**ESCOLA SESI DE REFERÊNCIA - SESI SENAI**  
**TESTES DE SOFTWARE**

**TESTES DO SISTEMA TCHUU-TCHUU**

**Professora :**

**Djeniffer Caroline Machado**

**Equipe:**

**Davi Bousfield Castelucci**

**Guilherme José Schutz da Silva**

**Braian de Liz da Silva**

**Augusto Wilbert Friedrichsen**

**TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS INTEGRADO AO ENSINO MÉDIO**

**EMDS 2024**  
**AGOSTO/ 2025**

## Histórico de Revisões

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>
21/08/2025	<1.0>	Release Inicial	Escola SESI

# Índice

<b>1. INTRODUÇÃO .....</b>	<b>4</b>
<b>1.1 OBJETIVOS .....</b>	<b>4</b>
<b>1.2 O TCHUU-TCHUU .....</b>	<b>4</b>
<b>1.3 ESCOPO.....</b>	<b>4</b>
<b>1.4 IDENTIFICAÇÃO DE PROJETO .....</b>	<b>5</b>
<b>2. REQUISITOS A TESTAR.....</b>	<b>6</b>
<b>2.1 Teste Do Banco De Dados.....</b>	<b>6</b>
<b>2.2 Teste Funcional.....</b>	<b>6</b>
<b>2.3 Teste Do Ciclo De Negócios .....</b>	<b>6</b>
<b>2.4 Teste Da Interface Do Usuário.....</b>	<b>7</b>
<b>2.5 Perfil Da Performance.....</b>	<b>7</b>
<b>2.6 Teste De Carga .....</b>	<b>7</b>
<b>2.7 Teste De Stress .....</b>	<b>7</b>
<b>2.8 Teste De Volume .....</b>	<b>7</b>
<b>2.9 Teste De Segurança E De Controle De Acesso .....</b>	<b>7</b>
<b>2.10 Teste De Falha/Recuperação .....</b>	<b>7</b>
<b>2.11 Teste De Instalação .....</b>	<b>8</b>
<b>3. ESTRATÉGIA DE TESTE.....</b>	<b>9</b>
<b>3.1 Tipos De Teste.....</b>	<b>9</b>
<b>3.1.1 Teste De Integridade De Dados E Do Banco De Dados .....</b>	<b>9</b>
<b>3.1.2 Teste De Função.....</b>	<b>9</b>
<b>3.1.3 Teste Da Interface Do Usuário .....</b>	<b>10</b>
<b>3.1.4 Teste De Performance .....</b>	<b>10</b>
<b>3.1.5 Teste De Carga .....</b>	<b>12</b>
<b>3.1.6 Teste De Segurança E Controle De Acesso.....</b>	<b>12</b>
<b>3.1.7 Teste De Instalação.....</b>	<b>13</b>
<b>3.2 FERRAMENTAS.....</b>	<b>13</b>
<b>4. RECURSOS.....</b>	<b>14</b>
<b>4.1 Trabalhadores.....</b>	<b>14</b>
<b>4.2 Sistema.....</b>	<b>15</b>
<b>5. CRONOGRAMA .....</b>	<b>16</b>

## 1. INTRODUÇÃO

### 1.1 Objetivos

✓ Esse documento do Plano de Testes do TCHUU-TCHUU compõe-se dos seguintes objetivos:

- O objetivo dos testes no projeto TCHUU-TCHUU é garantir que o sistema funcione conforme especificado nos requisitos funcionais e não funcionais, assegurando confiabilidade, segurança e usabilidade. Os testes têm como foco principal:
- 
- Verificar o correto funcionamento das funcionalidades essenciais, como autenticação de usuários, monitoramento em tempo real dos trens, atualização de status e comunicação entre equipes.
- Validar a integridade e consistência dos dados entre o front-end, back-end e banco de dados.
- Assegurar a segurança no acesso às informações, especialmente em áreas restritas como gestão de rotas e manutenção.
- Confirmar que o sistema atende aos critérios de desempenho, como tempo de resposta inferior a 2 segundos e disponibilidade mínima de 99,5%.
- Identificar falhas ou comportamentos inesperados antes da entrega final, reduzindo retrabalho e riscos operacionais.
- Garantir que a interface seja intuitiva e acessível, especialmente para usuários com diferentes perfis (maquinista, gestor, equipe de manutenção).
- Os testes serão realizados de forma contínua durante o desenvolvimento, permitindo a detecção precoce de problemas e a validação progressiva do sistema como um todo.

### 1.2 O TCHUU-TCHUU

O projeto TCHUU-TCHUU é um sistema de gerenciamento ferroviário desenvolvido para simular uma solução integrada de monitoramento e controle operacional de trens em ambientes urbanos inteligentes. Voltado para maquinistas, gestores e equipes de manutenção, o sistema permite acompanhar em tempo real o status dos trens (ativos, ociosos ou em manutenção), visualizar rotas no mapa, receber alertas sobre condições adversas, como clima, e registrar intervenções técnicas com rastreabilidade completa. Através de uma interface web responsiva, o TCHUU-TCHUU centraliza funções essenciais para a operação ferroviária, como alteração de rotas, comunicação entre equipes e geração de relatórios operacionais e financeiros.

A arquitetura do sistema é full-stack, utilizando tecnologias web como HTML, CSS, JavaScript no front-end, PHP e Node.js no back-end e banco de dados MySQL para armazenamento seguro de informações. O projeto contempla requisitos funcionais e não

funcionais críticos, como autenticação segura, criptografia de dados, logs de auditoria, baixo tempo de resposta e compatibilidade com dispositivos móveis e navegadores. Toda a operação pode ser executada em uma intranet local, garantindo disponibilidade mesmo sem conexão externa constante, simulando um ambiente real de transporte ferroviário.

### **1.3 Escopo :**

O escopo dos testes no projeto TCHUU-TCHUU abrange os níveis de teste unitário, teste de integração, teste de sistema, teste de aceitação, teste de desempenho e carga, teste de segurança e teste de usabilidade, com foco nas funcionalidades críticas para a operação ferroviária, como autenticação de usuários, monitoramento em tempo real dos trens, atualização de status, gestão de rotas e comunicação entre equipes. Os testes de aceitação serão realizados em conjunto com os integrantes da equipe, outros estudantes e professores.

Os testes unitários serão aplicados às funções principais do sistema, como validação de login, cadastro de maquinistas, alteração de status de trens e envio de alertas. Já os testes de integração garantirão a correta comunicação entre os módulos, especialmente entre o front-end, o back-end (Node.js e PHP) e o banco de dados (MySQL), assegurando que operações como registrar manutenção ou alterar rota sejam refletidas em tempo real para todos os usuários conectados.

Os testes serão executados em ambiente controlado, separado do ambiente de produção, contemplando front-end, back-end, banco de dados e ferramentas específicas para cada tipo de teste.

## 1.4 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	▪ Sim
Plano de Projeto	▪ Sim	▪ Sim
Modelo de Análise	▪ Não	▪ Sim
Modelo de Projeto	▪ Não(ainda incompleto)	▪ Sim
Documento de Arquitetura	▪ Não	▪Não
Protótipo	▪ Sim	▪ Sim
Manual do Usuário	▪ Não	▪Não
Lista de Riscos	▪ Não	▪Não

## 2. REQUISITOS A TESTAR

A lista abaixo identifica aqueles itens – use cases, requisitos funcionais e não funcionais – que foram identificados como alvos de teste. Essa lista representa o que será testado.

### 2.1 Teste do Banco de Dados

- Verifique que as informações do usuário podem ser cadastradas, consultadas e removidas.
- Verifique que as informações como Trens podem ser inseridos, atualizados e consultados.
- Verifique que as informações específicas de cada usuário podem ser acompanhadas.
- Verifique que o sistema pode cadastrar, atualizar e remover um trem de seu banco de dados, caso o trem esteja com algum problema, precisa ser possível mandar uma requisição e mandar o trem para manutenção.
- Verifique que os Trens serão organizados em tabelas por atributos, tais como : marca, ano, número do trem .
- Verifique que os Vias Ferroviárias possam ser cadastrados, removidos e atualizados pelo administrador do sistema.
- Verifique que as informações sobre os usuários possam ser cadastradas, removidas e atualizadas pelo administrador do sistema.
- Verifique que os trens cadastrados possam ser consultados pelos usuários.
- Verifique que o usuário cadastrado pode acessar o mapa das ferrovias próximas do trajeto.
- Verifique que o usuário cadastrado pode acessar a tela do usuário.
- Verifique que o histórico de manutenção de cada trem pode ser consultado.

## 2.2 Teste Funcional

- Verifique que as informações úteis obtidas pelo subsistema responsável são automaticamente e periodicamente atualizadas.
- Verifique que qualquer usuário pode acessar sua própria conta através de login e senha.
- Verifique que o usuário cadastrado pode alterar as informações do usuário.
- Verifique que o usuário cadastrado pode acessar a comunicação via chat.
- Exibir informações de trens, viagens, horários e alertas em tempo real.
- Verifique se o painel de informações está com os trens, horários e destinos corretos.
- Verifique o recebimento de mensagens programadas de manutenção.
- Verifique que o sistema permite cadastrar um maquinista solicitando CPF, nome, telefone e número de funcionário.
- Verifique que o sistema não permite cadastro com dados incompletos ou inválidos.
- Verifique que qualquer usuário pode acessar sua própria conta através de login por email e senha.
- Verifique que o usuário pode ativar a opção “lembrar-me” e realizar login sem senha no navegador.
- Verifique que o usuário cadastrado pode alterar suas informações pessoais a qualquer momento.
- Verifique que o painel exibe informações de trens ociosos e em circulação.
- Verifique que o painel exibe horários de trens e alertas em tempo real.
- Verifique que o usuário recebe notificações de manutenção programada e alertas críticos.
- Verifique que o usuário pode acessar o mapa das rotas dos trens e acompanhar trajetos.
- Verifique que administradores podem gerenciar permissões de usuários corretamente.
- Verifique que gestores podem alterar rotas de trens e que essas alterações são refletidas no mapa.
- Verifique que o usuário pode acessar a documentação de funcionalidades e ajuda online.
- Verifique que o usuário pode selecionar rotas registradas e visualizar suas informações.
- Verifique que é possível cadastrar novos trens e que eles aparecem na listagem.
- Verifique que o estado de um trem (ativo, manutenção, pronto) pode ser alterado

e refletido corretamente.

- Verifique que o sistema registra o motivo da manutenção para cada trem.
- Verifique que o sistema permite navegação fluida entre todas as páginas.
- Verifique que o usuário recebe alertas de clima que possam impactar a operação dos trens.
- Verifique que o sistema exibe horários de chegada e partida corretamente.
- Verifique que o sistema permite pesquisar rotas existentes e exibe resultados corretos.
- Verifique que o sistema indica quais trens estão para manutenção.
- Verifique que o usuário pode enviar e receber mensagens via chat com a central de controle.
- Verifique que o sistema gera relatórios de manutenção corretamente.
- Verifique que é possível alterar o estado de componentes do trem (temperatura, pressão, motor, combustível).
- Verifique que o sistema solicita o número de registro do funcionário para recuperação de senha e envia instruções corretas.

### **2.3 Teste da Interface do Usuário**

- Navegue através de todos os use cases, verificando que cada tela de interface gráfica pode ser rapidamente entendida e facilmente utilizada.
- Verifique que toda ajuda online funciona.
- Verifique se os menus, botões e links levam às telas corretas.
- Verifique se as telas seguem um padrão do design.
- Verifique se as validações de campo funcionam (campos obrigatórios, formato de email, CPF válido etc.).
- Verifique que o sistema começa na tela de login.

### **2.4 Perfil da Performance**

- Verifique o tempo de resposta da rede interna, do servidor em relação aos terminais.
- Verifique o tempo de consulta/atualização do subsistema de informações úteis.
- Verifique que o tempo de resposta para operações que envolvam dados multimídia (imagens, vídeos, etc.) não ultrapassem 30 segundos.

### **2.5 Teste de Stress**

- As APIs com Node.js e PHP serão testadas com Postman para verificar seu limite de requisições(depõe muito das hospedagem).
- WebSocket do chat e de das atualizações de status dos trens serão testados com o Artillery(biblioteca do npm)
- banco de dados com o k6.



## **2.6 Teste de Volume**

- Poucos maquinistas e linhas, menor receita. Estratégia em planos acessíveis e foco em nichos regionais.
- Expansão gradual da base de usuários, necessidade de estabilidade. Estratégia em otimizar infraestrutura e suporte
- Grande adesão de maquinistas e empresas, pressão por desempenho. Estratégia em escalar tecnologia, garantir alta disponibilidade e oferecer funcionalidades avançadas.

## **2.7 Teste de Segurança e de Controle de Acesso**

- Verificar que usuários não cadastrados não podem acessar informações das rotas.
- Verificar que os usuários apenas veem suas próprias informações de conta no TchuuTchuu.
- Verificar que a atualização do sistema pode ser feita apenas a partir da rede interna do TchuuTchuu.

## **2.8 Teste de Falha/Recuperação**

- Interrupção do sistema ou perda de dados. Risco de atrasos operacionais e perda de confiança.
- Backup em nuvem, redundância de servidores e protocolos de resposta rápida. retomada do serviço com impacto mínimo.
- Monitoramento contínuo, testes periódicos de contingência e comunicação clara com usuários em incidentes.

### 3. ESTRATÉGIA DE TESTE

#### 3.1 Tipos de Teste

Nota: As transações abaixo se referem às “transações lógicas de negócio”. Essas transações são definidas como funções específicas que um usuário final do sistema é suposto de executar ao usar a aplicação, tais como adicionar ou modificar uma dada informação.

##### 3.1.1 Teste de Integridade de Dados e do Banco de Dados

<b>Objetivo do Teste:</b>	<ul style="list-style-type: none"> <li>Garantir que os métodos e processos de acesso ao banco de dados funcionam apropriadamente e sem corrupção dos dados.</li> </ul>
<b>Técnica:</b>	<ul style="list-style-type: none"> <li>Invocar cada método e processo de acesso ao banco de dados, alimentando cada um com dados ou requisições de dados válidos e inválidos.</li> <li>Inspecionar o banco de dados para garantir que os dados foram populados como pretendido, que todos os eventos do banco de dados ocorreram apropriadamente, ou revisar os dados retornados para garantir que os dados corretos foram recuperados pelas razões corretas.</li> </ul>
<b>Critério de Finalização:</b>	<ul style="list-style-type: none"> <li>Todos os métodos e processos de acesso à base de dados funcionam como projetados e sem nenhuma corrupção de dados.</li> </ul>
<b>Considerações Especiais:</b>	<ul style="list-style-type: none"> <li>O teste pode necessitar de um ambiente de desenvolvimento ou drivers de SGBD para inserir ou modificar os dados diretamente nas bases de dados</li> <li>Processos devem ser invocados manualmente</li> <li>Bases de dados pequenas ou minimizadas (número de registros limitados) devem ser usados para aumentar a visibilidade de eventos não-aceitáveis.</li> </ul>

##### 3.1.1 Teste de Função

<b>Objetivo do Teste:</b>	Garantir a funcionalidade apropriada do alvo do teste, incluindo navegação, entrada de dados, processamento e recuperação.
<b>Técnica:</b>	<p>Executar cada caso de uso, fluxo de caso de uso, usando dados válidos e inválidos, para verificar o seguinte:</p> <ul style="list-style-type: none"> <li>Os resultados esperados ocorrem quando dados válidos são usados</li> <li>As mensagens de erro ou aviso apropriadas são exibidas quando dados inválidos são usados.</li> <li>Cada regra de negócio é aplicada apropriadamente</li> </ul>

<b>Critério de Finalização:</b>	<ul style="list-style-type: none"> <li>▪ Todos os testes planejados foram executados.</li> <li>▪ Todos os defeitos identificados foram tratados.</li> </ul>
<b>Considerações Especiais:</b>	Nenhum

### 3.1.2 Teste da Interface do Usuário

<b>Objetivo do Teste:</b>	<p>Verificar o seguinte:</p> <ul style="list-style-type: none"> <li>• A navegação através dos alvos de teste reflete as funções e os requisitos do negócio apropriadamente, incluindo janela-a-janela, campo-a-campo, e o uso de métodos de acesso (tecla tab, movimentos do mouse, teclas aceleradoras)</li> <li>• Objetos e características da janela, tais como menus, tamanho, posição, estado e foco conformam-se aos padrões.</li> </ul>
<b>Técnica:</b>	<ul style="list-style-type: none"> <li>• Criar ou modificar os testes para cada janela para verificar a navegação e os estados de objeto apropriados para cada janela e objetos da aplicação.</li> </ul>
<b>Critério de Finalização:</b>	<ul style="list-style-type: none"> <li>• É verificado que cada janela permanece consistente com a versão de comparação ou dentro de padrões aceitáveis.</li> </ul>
<b>Considerações Especiais:</b>	<ul style="list-style-type: none"> <li>• Nem todas as propriedades para objetos personalizados e terceirizados podem ser acessadas.</li> </ul>

## 3.1.3 Teste de Performance

<b>Objetivo do Teste:</b>	<p>Verificar que os comportamentos de performance para as transações designadas ou funções de negócio sob as seguintes condições:</p> <ul style="list-style-type: none"> <li>• Carga de trabalho normal antecipada</li> <li>• Carga de trabalho no pior caso antecipada</li> </ul>
<b>Técnica:</b>	<ul style="list-style-type: none"> <li>• Usar Procedimentos de Teste desenvolvidos para Teste da Função ou Ciclo de Negócio</li> <li>• Modificar os arquivos de dados para aumentar o número de transações ou os scripts para aumentar o número de iterações que ocorrem a cada transação.</li> <li>• Scripts devem ser rodados em uma máquina (melhor caso para comparar um único usuário, uma única transação) e ser repetidas com múltiplos clientes (virtual ou real, ver Considerações Especiais abaixo).</li> </ul>
<b>Critério de Finalização:</b>	<ul style="list-style-type: none"> <li>• Único usuário ou transação: finalização com sucesso dos scripts de testes sem nenhuma falha e dentro da alocação de tempo por transação esperada ou requisitada.</li> <li>• Múltiplas transações ou usuários: finalização bem sucedida dos scripts de teste sem qualquer falha e dentro da alocação de tempo aceitável.</li> </ul>

<b>Considerações Especiais:</b>	<p>Um teste abrangente de performance inclui ter uma carga de trabalho no servidor.</p> <p>Há vários métodos que podem ser usados para executar isso, incluindo:</p> <ul style="list-style-type: none"> <li>• “Direcionar transações” diretamente para o servidor, usualmente na forma de chamadas SQL.</li> <li>• Criar carga de usuário “virtual” para simular muitos clientes, normalmente várias centenas. Ferramentas de Emulação de Terminal Remoto (RTE) são usadas para atingir essa carga. Essa técnica também pode ser usada para carregar uma rede com “tráfego”.</li> <li>• Usar múltiplos clientes físicos, cada um rodando scripts de teste para gerar uma carga no sistema.</li> <li>• O teste de performance deve ser executado em uma máquina dedicada ou em um tempo dedicado. Isso permite controle total e mensuração precisa.</li> <li>• As bases de dados usadas para o Teste de Performance devem ser ou do tamanho real ou proporcionalmente iguais.</li> </ul>
---------------------------------	---

## 3.1.4 Teste de Carga

<b>Objetivo do Teste:</b>	<ul style="list-style-type: none"> <li>• Verifique o tempo de resposta para as transações designadas ou casos de negócio sob condições variantes de carga de trabalho.</li> </ul>
<b>Técnica:</b>	<ul style="list-style-type: none"> <li>• Use testes desenvolvidos para o Teste do Ciclo de Negócio ou Função.</li> <li>• Modifique os arquivos de dados para aumentar o número de transações ou os testes para aumentar o número de vezes que cada transação ocorre.</li> </ul>
<b>Critério de Finalização:</b>	<ul style="list-style-type: none"> <li>• Múltiplas transações ou usuários: finalização bem sucedida dos testes sem qualquer falha e dentro da alocação de tempo aceitável.</li> </ul>
<b>Considerações Especiais:</b>	<ul style="list-style-type: none"> <li>• O teste de carga deve ser executado em uma máquina dedicada ou em um tempo dedicado. Isso permite o controle total da mensuração precisa.</li> <li>• As bases de dados usadas para os testes de carga devem ou ser do tamanho real ou igualmente dimensionadas.</li> </ul>

## 3.1.5 Teste de Segurança e Controle de Acesso

<b>Objetivo do Teste:</b>	<ul style="list-style-type: none"> <li>• Segurança do Nível de Aplicação: Verifique que um ator pode acessar apenas aquelas funções ou dados para os quais o seu tipo de usuário tem permissão.</li> <li>• Segurança do Nível de Sistema: Verifique que apenas aqueles atores com acesso ao sistema e aplicações têm permissão de acessá-los.</li> </ul>
<b>Técnica:</b>	<ul style="list-style-type: none"> <li>• Segurança do Nível de Aplicação: Identifique e liste cada tipo de usuário e as funções ou dados para os quais cada tipo tem permissão.</li> <li>• Crie testes para cada tipo de usuário e verifique cada permissão criando transações específicos para cada tipo de usuário.</li> <li>• Modifique o tipo de usuário e repita os testes para os mesmos usuários. Em cada caso, verifique que funções ou dados adicionais estão corretamente disponíveis ou negados.</li> <li>• Acesso de Nível de Sistema: Ver Considerações Especiais abaixo.</li> </ul>
<b>Critério de Finalização:</b>	<ul style="list-style-type: none"> <li>• Para cada tipo de ator conhecido as funções ou dados apropriados estão disponíveis, e todas as transações funcionam como esperado e rodam nos Testes de Função anteriores.</li> </ul>
<b>Considerações Especiais:</b>	<ul style="list-style-type: none"> <li>• O Acesso ao sistema deve ser revisado ou discutido com o administrador de rede ou de sistema apropriado. Esse teste pode não ser necessário já que ele pode ser uma função da administração da rede ou sistema.</li> </ul>

### 3.2 Ferramentas

As seguintes ferramentas serão empregadas para esse projeto:

Categoria	Ferramenta	Descrição
Gerenciamento de Teste	GitHub Actions	Automatização de pipelines de teste e deploy
	Jira	Gestão de tarefas e rastreamento de bugs
Execução de Testes	Postman	Testes manuais e automação de APIs REST
	Artillery	Testes de carga e stress para WebSocket e APIs
	Selenium	Testes de interface gráfica (se necessário)
Banco de Dados	Mysql ou Postgresql	Banco de dados relacional, manipulado com a linguagem SQL
Hospedagem	Render e ou Vercel	Hospedagem do back-end e front-end

## 4. RECURSOS

Essa seção apresenta os recursos recomendados para o projeto do Tchuu-Tchuu, suas

principais responsabilidades, e seus conhecimentos ou conjunto de habilidades.

#### 4.1 Trabalhadores

Essa tabela mostra as suposições de recrutamento para o projeto.

Recursos Humanos		
Trabalhador	Recursos Mínimos Recomendados	Responsabilidades Específicas ou Comentários
<b>Gerente de Teste, Gerente do Projeto de Teste</b>	Braian de Liz da Silva	<p>Fornece supervisionamento gerencial. Responsabilidades:</p> <ul style="list-style-type: none"> <li>• provê direcionamento técnico</li> <li>• adquire recursos apropriados</li> <li>• fornece relatórios de gerenciamento</li> </ul>
<b>Test Designer</b>	Augusto Wilbert Friedrichsen	<p>Identifica, prioriza, e implementa os casos de teste.</p> <p>Responsabilidades:</p> <ul style="list-style-type: none"> <li>• gera o plano de teste</li> <li>• cria o modelo de teste</li> <li>• avalia a efetividade do esforço de teste</li> </ul>
<b>Testador</b>	Guilherme José Schutz da Silva	<p>Executa os testes.</p> <p>Responsabilidades:</p> <ul style="list-style-type: none"> <li>• executar os testes</li> <li>• registrar os resultados</li> <li>• reestabelecer-se dos erros</li> <li>• documentar solicitações de mudança</li> </ul>
<b>Administrador do Sistema de Teste</b>	Davi Castelucci	<p>Garante que o ambiente e os bens de teste sejam gerenciados e mantidos.</p> <p>Responsabilidades:</p> <ul style="list-style-type: none"> <li>• administrar o sistema de gerenciamento teste</li> <li>• instalar e gerenciar o acesso do trabalhador ao sistema de testes</li> </ul>

<b>Gerente do Banco de Dados, Administrador do Banco de Dados</b>	Guilherme José Schutz da Silva	<ul style="list-style-type: none"> <li>▪ Garante que o ambiente e bens de teste de dados (banco de dados) sejam gerenciados e mantidos.</li> </ul> Responsabilidades: <ul style="list-style-type: none"> <li>▪ administrar os dados de teste (base de dados)</li> </ul>
<b>Designer</b>	Augusto Wilbert Friedrichsen	Identifica e define as operações, atributos, e associações das classes de teste. Responsabilidades: <ul style="list-style-type: none"> <li>• identificar e definir as classes de teste</li> <li>• identificar e definir os pacotes de teste</li> </ul>
<b>Implementador</b>	Braian de Liz da Silva	Implementa e faz os testes unitários das classes e pacotes de teste. Responsabilidades: <ul style="list-style-type: none"> <li>▪ cria as classes e pacotes de teste implementados no modelo de teste</li> </ul>

## 4.2 Sistema

Recursos do Sistema
<ul style="list-style-type: none"> <li>• Servidor de Banco de Dados <ul style="list-style-type: none"> <li>○ MySQL DataBase Server</li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>• Terminais Clientes <ul style="list-style-type: none"> <li>○ 2 PCs (conectados via LAN)</li> <li>○ 1 PC com tela sensível ao toque (conectado a uma LAN e à internet)</li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>• Repositório de Testes <ul style="list-style-type: none"> <li>○ 1 PC</li> <li>○ 3 PCs de Desenvolvimento de Teste</li> </ul> </li> </ul>

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



## 5. CRONOGRAMA

Milestone	Data de Início	Data de Término
Criar Gerenciamento dos testes	23/08/2025	30/08/2025
Projetar Teste	23/08/2025	23/08/2025
Implementar Teste	23/09/2025	20/11/2025
Executar Teste	23/09/2025	20/11/2025
Avaliar Teste	14/10/2025	20/11/2025