Sumário

[1. Introdução 2](#_Toc147310311)

[2. Escopo do Teste 2](#_Toc147310312)

[3. Estratégia de Teste 2](#_Toc147310313)

[3.1 Funcionais 2](#_Toc147310314)

[3.2 Não funcionais 3](#_Toc147310315)

[4. Ambiente de Teste 3](#_Toc147310316)

[5. Critérios de Aceitação 3](#_Toc147310317)

[6. Riscos 3](#_Toc147310318)

[7. Cronograma 4](#_Toc147310319)

[8. Casos de Teste 4](#_Toc147310320)

[9. Relatório de Bugs 6](#_Toc147310321)

[10. Conclusão 6](#_Toc147310322)

[11. Histórico de Revisões 7](#_Toc147310323)

**Testes de Software**

# 1. Introdução

Este documento descreve o plano de teste para validar as funcionalidades e a performance da plataforma Cold Start, que busca gerenciar os contêineres de refrigeração da Bosch Campinas.

# 2. Escopo do Teste

Serão testadas as funcionalidades de login, visualização da temperatura dos contêineres, alteração do setpoint de temperatura e agendamento de contêiner.

# 3. Estratégia de Teste

Serão utilizados testes de integração e de caixa preta, em sua maioria, principalmente para os requisitos funcionais

Exemplos:

## 3.1 Funcionais

|  |  |  |
| --- | --- | --- |
| **ID** | **Requisito** | **Tipo de Teste** |
| RF01 | O usuário deve ser capaz de se autenticar usando o SSO da Bosch. | Caixa-preta, Integração |
| RF02 | O aplicativo deve exibir todos os containers na tela inicial. | Caixa-preta |
| RF03 | O aplicativo deve ser capaz de mostrar a temperatura ambiente do container, do carro 1 e carro 2. | Caixa-preta |
| RF04 | O aplicativo deve possibilitar o usuário alterar o set point de containers agendados por este mesmo usuário. | Caixa-preta |
| RF05 | O usuário deve ser capaz de criar um agendamento para utilizar um container específico em um período de tempo | Caixa-preta |

## 3.2 Não funcionais

|  |  |  |
| --- | --- | --- |
| **ID** | **Requisito** | **Tipo de Teste** |
| RNF01 | Todas as páginas do aplicativo devem ser carregadas em até 2 segundos em uma conexão de internet padrão | Desempenho |
| RNF02 | Todos os dados do usuário devem ser criptografados e armazenados de forma segura | Confiabilidade |
| RNF03 | O serviço deve estar disponível 99,9% do tempo | Confiabilidade |
| RNF04 | O sistema deve ser capaz de suportar até 50 usuários simultâneos | Carga |
| RNF05 | O aplicativo deve ser compatível com as últimas versões dos sistemas operacionais Windows e Linux | Portabilidade |
| RNF06 | O aplicativo deve ser intuitivo e fácil de usar, com um tempo máximo de aprendizado de 15 minutos para novos usuários | Usabilidade |

# 4. Ambiente de Teste

Dispositivos: Windows

Conexão: Wi-Fi

# 5. Critérios de Aceitação

* Um teste será considerado bem-sucedido se:

O aplicativo passar em 95% dos casos de teste para ser considerado apto para lançamento.

* Integração com Contêiners:

O sistema deve estar integrado com as raspberrys de cada um dos contêineres de refrigeração.

* Cadastro e Gerenciamento de Usuário:

Os usuários poderão acessar a aplicação através do usuário de rede da Bosch, dessa forma o gerenciamento dos usuários é realizado pela azure.

# 6. Riscos

|  |  |  |
| --- | --- | --- |
| **Risco** | **Impacto** | **Plano de Mitigação** |
| Raspberry dar defeito | Alto | Monitoramento da condição dos equipamentos |
| Servidor cair | Alto | Backup de servidor, para garantir alta disponibilidade |

# 7. Cronograma

Testes de caixa-preta serão realizados na primeira semana, seguidos por testes de carga na segunda semana e testes de integração durante a terceira.

# 8. Casos de Teste

**Funcionais:**

**Requisito: RF1** - **O usuário deve ser capaz de se autenticar usando o SSO da Bosch.**

**Caso de Teste:** Login bem-sucedido

**ID:** TC01-S

**Pré-condições:** O usuário deve ter um usuário Bosch.

**Entrada:** SSO

**Passos de Teste:**

1. Abrir o app Cold Start.
2. Clicar no botão “Login”.

**Resultado Esperado:** O usuário será redirecionado para a página inicial do app.

**Pós-condições:** O usuário está logado.

**Requisito: RF2** - **O aplicativo deve exibir todos os containers na tela inicial.**

**ID:** TC02-RF03-C

**Pré-condições:** O usuário deve estar logado.

**Entrada:** nenhuma.

**Passos de teste:**

1. Abrir a tela inicial.

**Resultado Esperado:** Os containers são renderizados na tela.

**Pós-condições:** O usuário pode clicar em qualquer container para visualizar as informações específicas dele.

**Caso de Teste:** Falha na requisição da API.

### **Requisito: RF03 - O aplicativo deve ser capaz de mostrar a temperatura ambiente do container, do carro 1 e do carro 2.**

### **Caso de Teste: Exibição da Temperatura do Container com Sucesso**

**ID:** TC01-RF03-C

**Pré-condições:**

* O usuário deve estar logado na aplicação.
* O sensor de temperatura do container deve estar funcionando corretamente.

**Entrada:** Acesso à seção de temperatura do container.

**Passos de Teste:**

1. Abrir o aplicativo web.
2. Navegar até a seção de temperatura do container.
3. Verificar a exibição da temperatura do container.

**Resultado Esperado:** A temperatura do container é exibida de forma clara e legível, por exemplo, "22°C".

**Pós-condições:** O usuário visualiza a temperatura atual do container.

### **Caso de Teste: Exibição da Temperatura do Carro 1 com Sucesso**

**ID:** TC02-RF03-C1

**Pré-condições:**

* O usuário deve estar logado na aplicação.
* O sensor de temperatura do carro 1 deve estar funcionando corretamente.

**Entrada:** Acesso à seção de temperatura do carro 1.

**Passos de Teste:**

1. Abrir o aplicativo web.
2. Navegar até a seção de temperatura do carro 1.
3. Verificar a exibição da temperatura do carro 1.

**Resultado Esperado:** A temperatura do carro 1 é exibida de forma clara e legível, por exemplo, "24°C".

**Pós-condições:** O usuário visualiza a temperatura atual do carro 1.

### **Caso de Teste: Exibição da Temperatura do Carro 2 com Sucesso**

**ID:** TC03-RF03-C2

**Pré-condições:**

* O usuário deve estar logado na aplicação.
* O sensor de temperatura do carro 2 deve estar funcionando corretamente.

**Entrada:** Acesso à seção de temperatura do carro 2.

**Passos de Teste:**

1. Abrir o aplicativo web.
2. Navegar até a seção de temperatura do carro 2.
3. Verificar a exibição da temperatura do carro 2.

**Resultado Esperado:** A temperatura do carro 2 é exibida de forma clara e legível, por exemplo, "20°C".

**Pós-condições:** O usuário visualiza a temperatura atual do carro 2.

### **Caso de Teste: Exibição Simultânea das Temperaturas com Sucesso**

**ID:** TC04-RF03-S

**Pré-condições:**

* O usuário deve estar logado na aplicação.
* Os sensores de temperatura do container, do carro 1 e do carro 2 devem estar funcionando corretamente.

**Entrada:** Acesso à seção de temperaturas.

**Passos de Teste:**

1. Abrir o aplicativo web.
2. Navegar até a seção de temperaturas.
3. Verificar a exibição das temperaturas do container, do carro 1 e do carro 2.

**Resultado Esperado:** As temperaturas do container, do carro 1 e do carro 2 são exibidas de forma clara e legível, por exemplo, "22°C, 24°C, 20°C".

**Pós-condições:** O usuário visualiza as temperaturas atuais do container, do carro 1 e do carro 2 simultaneamente.

### **Caso de Teste: Exibição de Mensagem de Erro em Caso de Falha na Obtenção da Temperatura**

**ID:** TC05-RF03-F

**Pré-condições:**

* O usuário deve estar logado na aplicação.
* O sensor de temperatura do container deve estar desconectado ou inoperante.

**Entrada:** Acesso à seção de temperatura do container.

**Passos de Teste:**

1. Abrir o aplicativo web.
2. Navegar até a seção de temperatura do container.
3. Verificar a exibição da temperatura do container.

**Resultado Esperado:** Uma mensagem de erro apropriada é exibida, informando que a temperatura do container não está disponível.

**Pós-condições:** O usuário é informado da indisponibilidade da temperatura do container.

### **Caso de Teste: Consistência das Temperaturas Após Atualização**

**ID:** TC06-RF03-Cons

**Pré-condições:**

* O usuário deve estar logado na aplicação.
* Os sensores de temperatura devem estar funcionando corretamente.

**Entrada:** Acesso à seção de temperaturas.

**Passos de Teste:**

1. Abrir o aplicativo web.
2. Navegar até a seção de temperaturas.
3. Verificar a exibição inicial das temperaturas.
4. Realizar uma atualização ou esperar por uma atualização automática dos dados de temperatura.
5. Verificar a exibição das temperaturas após a atualização.

**Resultado Esperado:** As temperaturas atualizadas são consistentes com os dados fornecidos pelo sistema e continuam a ser exibidas de forma clara e legível.

**Pós-condições:** O usuário visualiza as temperaturas atualizadas do container, do carro 1 e do carro 2.

### **Requisito: RF04 - O usuário deve ser capaz de alterar o set point de containers agendados por este mesmo usuário.**

### **Caso de Teste: Alteração de Set Point com Sucesso**

**ID:** TC01-RF04-S

**Pré-condições:**

* O usuário deve estar logado na aplicação.
* O usuário deve ter containers agendados.

**Entrada:** Novo set point para um container agendado.

**Passos de Teste:**

1. Abrir o aplicativo web.
2. Navegar até a seção de containers agendados.
3. Selecionar um container agendado.
4. Clicar na opção de alterar o set point.
5. Inserir o novo valor do set point (por exemplo, "5°C").
6. Confirmar a alteração do set point.

**Resultado Esperado:** O set point do container é atualizado para o novo valor, e uma mensagem de sucesso é exibida.

**Pós-condições:** O set point do container selecionado é alterado conforme especificado.

### **Caso de Teste: Alteração de Set Point sem Sucesso devido a Permissão**

**ID:** TC02-RF04-F

**Pré-condições:**

* O usuário deve estar logado na aplicação.
* O usuário não deve ter permissão para alterar o set point de determinados containers (por exemplo, containers agendados por outros usuários).

**Entrada:** Tentativa de alteração de set point em um container não agendado pelo usuário.

**Passos de Teste:**

1. Abrir o aplicativo web.
2. Navegar até a seção de containers agendados.
3. Selecionar um container agendado por outro usuário.
4. Clicar na opção de alterar o set point.
5. Tentar inserir um novo valor para o set point.

**Resultado Esperado:** Uma mensagem de erro é exibida informando que o usuário não tem permissão para alterar o set point do container selecionado.

**Pós-condições:** O set point do container não é alterado.

### **Caso de Teste: Validação de Entrada Inválida para Set Point**

**ID:** TC03-RF04-V

**Pré-condições:**

* O usuário deve estar logado na aplicação.
* O usuário deve ter containers agendados.

**Entrada:** Valor inválido para o set point (por exemplo, texto não numérico, valores fora do intervalo permitido).

**Passos de Teste:**

1. Abrir o aplicativo web.
2. Navegar até a seção de containers agendados.
3. Selecionar um container agendado.
4. Clicar na opção de alterar o set point.
5. Inserir um valor inválido (por exemplo, "abc" ou "-5°C").
6. Tentar confirmar a alteração do set point.

**Resultado Esperado:** Uma mensagem de erro é exibida informando que o valor inserido é inválido, e o usuário é solicitado a inserir um valor válido.

**Pós-condições:** O set point do container não é alterado.

### **Caso de Teste: Persistência de Dados após Alteração de Set Point**

**ID:** TC04-RF04-P

**Pré-condições:**

* O usuário deve estar logado na aplicação.
* O usuário deve ter containers agendados.
* O set point de um container deve ter sido alterado com sucesso previamente.

**Entrada:** Nenhuma entrada específica necessária.

**Passos de Teste:**

1. Abrir o aplicativo web.
2. Navegar até a seção de containers agendados.
3. Selecionar o container cujo set point foi alterado.
4. Verificar se o set point atualizado é exibido corretamente.

**Resultado Esperado:** O set point atualizado é exibido corretamente, confirmando que a alteração foi persistida no sistema.

**Pós-condições:** O set point do container permanece alterado conforme especificado.

### **Caso de Teste: Alteração de Set Point em Condições de Rede Instável**

**ID:** TC05-RF04-R

**Pré-condições:**

* O usuário deve estar logado na aplicação.
* O usuário deve ter containers agendados.
* A rede deve estar instável (simulação de perda de conexão).

**Entrada:** Novo set point para um container agendado.

**Passos de Teste:**

1. Abrir o aplicativo web.
2. Navegar até a seção de containers agendados.
3. Selecionar um container agendado.
4. Clicar na opção de alterar o set point.
5. Inserir o novo valor do set point (por exemplo, "5°C").
6. Confirmar a alteração do set point.
7. Simular uma perda de conexão durante a confirmação.

**Resultado Esperado:** Uma mensagem de erro é exibida informando que houve um problema de conexão. A aplicação deve permitir tentar novamente ou salvar a alteração localmente para sincronização posterior.

**Pós-condições:** O set point do container não é alterado até que a conexão seja restabelecida e a alteração seja confirmada.

### **Requisito: RF05 - O usuário deve ser capaz de criar um agendamento para utilizar um container específico em um período de tempo.**

### **Caso de Teste: Criação de Agendamento com Sucesso**

**ID:** TC01-RF05-S

**Pré-condições:**

* O usuário deve estar logado na aplicação.
* O container desejado deve estar disponível no período solicitado.

**Entrada:** Detalhes do agendamento (container específico, data e hora de início e fim).

**Passos de Teste:**

1. Abrir o aplicativo web.
2. Navegar até a seção de agendamentos.
3. Clicar no botão “Criar Novo Agendamento”.
4. Selecionar o container específico.
5. Inserir a data e hora de início e fim do agendamento.
6. Confirmar a criação do agendamento.

**Resultado Esperado:** O agendamento é criado com sucesso, e uma mensagem de confirmação é exibida.

**Pós-condições:** O container é reservado para o usuário no período especificado.

### **Caso de Teste: Criação de Agendamento sem Sucesso devido a Conflito de Horário**

**ID:** TC02-RF05-F

**Pré-condições:**

* O usuário deve estar logado na aplicação.
* O container desejado já deve estar agendado por outro usuário no mesmo período.

**Entrada:** Detalhes do agendamento (container específico, data e hora de início e fim).

**Passos de Teste:**

1. Abrir o aplicativo web.
2. Navegar até a seção de agendamentos.
3. Clicar no botão “Criar Novo Agendamento”.
4. Selecionar o container específico.
5. Inserir a data e hora de início e fim do agendamento.
6. Confirmar a criação do agendamento.

**Resultado Esperado:** Uma mensagem de erro é exibida informando que o container já está agendado no período solicitado.

**Pós-condições:** O agendamento não é criado.

### **Caso de Teste: Validação de Entrada Inválida para Agendamento**

**ID:** TC03-RF05-V

**Pré-condições:**

* O usuário deve estar logado na aplicação.

**Entrada:** Detalhes do agendamento com valores inválidos (datas ou horas incorretas, por exemplo, fim anterior ao início).

**Passos de Teste:**

1. Abrir o aplicativo web.
2. Navegar até a seção de agendamentos.
3. Clicar no botão “Criar Novo Agendamento”.
4. Selecionar o container específico.
5. Inserir a data e hora de início e fim do agendamento com valores inválidos.
6. Tentar confirmar a criação do agendamento.

**Resultado Esperado:** Uma mensagem de erro é exibida informando que os valores inseridos são inválidos e orientando o usuário a corrigir os dados.

**Pós-condições:** O agendamento não é criado.

### **Caso de Teste: Persistência de Dados após Criação de Agendamento**

**ID:** TC04-RF05-P

**Pré-condições:**

* O usuário deve estar logado na aplicação.
* Um agendamento deve ter sido criado com sucesso previamente.

**Entrada:** Nenhuma entrada específica necessária.

**Passos de Teste:**

1. Abrir o aplicativo web.
2. Navegar até a seção de agendamentos.
3. Verificar a lista de agendamentos para confirmar que o agendamento criado anteriormente está presente.

**Resultado Esperado:** O agendamento criado anteriormente é exibido corretamente na lista de agendamentos.

**Pós-condições:** O agendamento criado permanece armazenado no sistema.

### **Caso de Teste: Criação de Agendamento em Condições de Rede Instável**

**ID:** TC05-RF05-R

**Pré-condições:**

* O usuário deve estar logado na aplicação.
* A rede deve estar instável (simulação de perda de conexão).

**Entrada:** Detalhes do agendamento (container específico, data e hora de início e fim).

**Passos de Teste:**

1. Abrir o aplicativo web.
2. Navegar até a seção de agendamentos.
3. Clicar no botão “Criar Novo Agendamento”.
4. Selecionar o container específico.
5. Inserir a data e hora de início e fim do agendamento.
6. Confirmar a criação do agendamento.
7. Simular uma perda de conexão durante a confirmação.

**Resultado Esperado:** Uma mensagem de erro é exibida informando que houve um problema de conexão. A aplicação deve permitir tentar novamente ou salvar a alteração localmente para sincronização posterior.

**Pós-condições:** O agendamento não é criado até que a conexão seja restabelecida e a criação seja confirmada.

**Não funcionais:**

### **Requisito: RNF01 - Todas as páginas do aplicativo devem ser carregadas em até 2 segundos em uma conexão de internet padrão.**

### **Caso de Teste: Tempo de Carregamento da Página Inicial**

**ID:** TC01-RNF01-Home

**Pré-condições:**

* O usuário deve estar conectado à internet com uma conexão padrão (por exemplo, 10 Mbps de download).

**Entrada:** Acesso à página inicial do aplicativo.

**Passos de Teste:**

1. Estabelecer uma conexão de internet padrão.
2. Abrir o navegador e limpar o cache.
3. Acessar a URL da página inicial do aplicativo.
4. Medir o tempo de carregamento da página inicial desde o momento em que a URL é solicitada até o completo carregamento da página.

**Resultado Esperado:** A página inicial é carregada em até 2 segundos.

**Pós-condições:** O usuário visualiza a página inicial totalmente carregada.

### **Caso de Teste: Tempo de Carregamento da Página de Agendamentos**

**ID:** TC02-RNF01-Agendamentos

**Pré-condições:**

* O usuário deve estar conectado à internet com uma conexão padrão (por exemplo, 10 Mbps de download).

**Entrada:** Acesso à página de agendamentos do aplicativo.

**Passos de Teste:**

1. Estabelecer uma conexão de internet padrão.
2. Abrir o navegador e limpar o cache.
3. Acessar a URL da página de agendamentos do aplicativo.
4. Medir o tempo de carregamento da página de agendamentos desde o momento em que a URL é solicitada até o completo carregamento da página.

**Resultado Esperado:** A página de agendamentos é carregada em até 2 segundos.

**Pós-condições:** O usuário visualiza a página de agendamentos totalmente carregada.

### **Caso de Teste: Tempo de Carregamento da Página de Detalhes do Container**

**ID:** TC03-RNF01-DetalhesContainer

**Pré-condições:**

* O usuário deve estar conectado à internet com uma conexão padrão (por exemplo, 10 Mbps de download).

**Entrada:** Acesso à página de detalhes de um container específico.

**Passos de Teste:**

1. Estabelecer uma conexão de internet padrão.
2. Abrir o navegador e limpar o cache.
3. Acessar a URL da página de detalhes do container no aplicativo.
4. Medir o tempo de carregamento da página de detalhes do container desde o momento em que a URL é solicitada até o completo carregamento da página.

**Resultado Esperado:** A página de detalhes do container é carregada em até 2 segundos.

**Pós-condições:** O usuário visualiza a página de detalhes do container totalmente carregada.

### **Caso de Teste: Tempo de Carregamento da Página de Perfil do Usuário**

**ID:** TC04-RNF01-PerfilUsuario

**Pré-condições:**

* O usuário deve estar conectado à internet com uma conexão padrão (por exemplo, 10 Mbps de download).

**Entrada:** Acesso à página de perfil do usuário.

**Passos de Teste:**

1. Estabelecer uma conexão de internet padrão.
2. Abrir o navegador e limpar o cache.
3. Acessar a URL da página de perfil do usuário no aplicativo.
4. Medir o tempo de carregamento da página de perfil do usuário desde o momento em que a URL é solicitada até o completo carregamento da página.

**Resultado Esperado:** A página de perfil do usuário é carregada em até 2 segundos.

**Pós-condições:** O usuário visualiza a página de perfil do usuário totalmente carregada.

### **Caso de Teste: Tempo de Carregamento da Página de Configurações**

**ID:** TC05-RNF01-Configuracoes

**Pré-condições:**

* O usuário deve estar conectado à internet com uma conexão padrão (por exemplo, 10 Mbps de download).

**Entrada:** Acesso à página de configurações do aplicativo.

**Passos de Teste:**

1. Estabelecer uma conexão de internet padrão.
2. Abrir o navegador e limpar o cache.
3. Acessar a URL da página de configurações no aplicativo.
4. Medir o tempo de carregamento da página de configurações desde o momento em que a URL é solicitada até o completo carregamento da página.

**Resultado Esperado:** A página de configurações é carregada em até 2 segundos.

**Pós-condições:** O usuário visualiza a página de configurações totalmente carregada.

### **Caso de Teste: Tempo de Carregamento da Página de Suporte/Ajuda**

**ID:** TC06-RNF01-Suporte

**Pré-condições:**

* O usuário deve estar conectado à internet com uma conexão padrão (por exemplo, 10 Mbps de download).

**Entrada:** Acesso à página de suporte/ajuda do aplicativo.

**Passos de Teste:**

1. Estabelecer uma conexão de internet padrão.
2. Abrir o navegador e limpar o cache.
3. Acessar a URL da página de suporte/ajuda no aplicativo.
4. Medir o tempo de carregamento da página de suporte/ajuda desde o momento em que a URL é solicitada até o completo carregamento da página.

**Resultado Esperado:** A página de suporte/ajuda é carregada em até 2 segundos.

**Pós-condições:** O usuário visualiza a página de suporte/ajuda totalmente carregada.

### **Requisito: RNF02 - Todos os dados do usuário devem ser criptografados e armazenados de forma segura.**

### **Caso de Teste: Criptografia de Dados Pessoais do Usuário**

**ID:** TC01-RNF02-DadosPessoais

**Pré-condições:**

* O usuário deve estar logado na aplicação.
* O sistema deve estar configurado para usar criptografia para dados pessoais.

**Entrada:** Dados pessoais do usuário (por exemplo, nome, endereço, número de telefone).

**Passos de Teste:**

1. Abrir o aplicativo web.
2. Navegar até a seção de perfil do usuário.
3. Inserir ou atualizar os dados pessoais do usuário.
4. Salvar as alterações.
5. Acessar a base de dados diretamente com as devidas permissões e ferramentas.
6. Verificar se os dados armazenados estão criptografados.

**Resultado Esperado:** Os dados pessoais do usuário são armazenados em formato criptografado na base de dados.

**Pós-condições:** Os dados pessoais do usuário permanecem criptografados na base de dados.

### **Caso de Teste: Criptografia de Dados de Autenticação do Usuário**

**ID:** TC02-RNF02-DadosAutenticacao

**Pré-condições:**

* O sistema deve estar configurado para usar criptografia para dados de autenticação.
* O usuário deve tentar criar uma conta ou alterar suas credenciais.

**Entrada:** Dados de autenticação do usuário (por exemplo, senha).

**Passos de Teste:**

1. Abrir o aplicativo web.
2. Navegar até a seção de registro ou alteração de senha.
3. Inserir ou alterar a senha do usuário.
4. Salvar as alterações.
5. Acessar a base de dados diretamente com as devidas permissões e ferramentas.
6. Verificar se as senhas armazenadas estão criptografadas (ou melhor, hashed com salt).

**Resultado Esperado:** As senhas do usuário são armazenadas em formato hashed e salgado na base de dados.

**Pós-condições:** As senhas do usuário permanecem seguras e criptografadas na base de dados.

### **Caso de Teste: Criptografia de Dados de Transações do Usuário**

**ID:** TC03-RNF02-DadosTransacoes

**Pré-condições:**

* O sistema deve estar configurado para usar criptografia para dados de transações.
* O usuário deve realizar uma transação no aplicativo.

**Entrada:** Dados da transação (por exemplo, detalhes de pagamento, histórico de uso).

**Passos de Teste:**

1. Abrir o aplicativo web.
2. Navegar até a seção de transações.
3. Realizar uma transação.
4. Salvar a transação.
5. Acessar a base de dados diretamente com as devidas permissões e ferramentas.
6. Verificar se os dados de transação armazenados estão criptografados.

**Resultado Esperado:** Os dados de transação do usuário são armazenados em formato criptografado na base de dados.

**Pós-condições:** Os dados de transação do usuário permanecem criptografados na base de dados.

### **Caso de Teste: Criptografia de Dados de Comunicação do Usuário**

**ID:** TC04-RNF02-DadosComunicacao

**Pré-condições:**

* O sistema deve estar configurado para usar criptografia para dados de comunicação.
* O usuário deve enviar ou receber mensagens através do aplicativo.

**Entrada:** Dados de comunicação do usuário (por exemplo, mensagens enviadas e recebidas).

**Passos de Teste:**

1. Abrir o aplicativo web.
2. Navegar até a seção de mensagens ou comunicação.
3. Enviar ou receber uma mensagem.
4. Acessar a base de dados diretamente com as devidas permissões e ferramentas.
5. Verificar se os dados de comunicação armazenados estão criptografados.

**Resultado Esperado:** As mensagens do usuário são armazenadas em formato criptografado na base de dados.

**Pós-condições:** As mensagens do usuário permanecem criptografadas na base de dados.

### **Caso de Teste: Verificação da Segurança na Transferência de Dados**

**ID:** TC05-RNF02-Transferencia

**Pré-condições:**

* O sistema deve estar configurado para usar HTTPS para comunicação segura.
* O usuário deve estar utilizando o aplicativo em uma rede monitorada.

**Entrada:** Qualquer dado transferido entre o cliente e o servidor (por exemplo, ao realizar login, enviar dados pessoais, etc.).

**Passos de Teste:**

1. Estabelecer um ambiente de teste com capacidade de monitorar tráfego de rede (por exemplo, utilizando ferramentas de inspeção de pacotes).
2. Abrir o aplicativo web.
3. Realizar várias ações que envolvam transferência de dados sensíveis (login, atualização de perfil, etc.).
4. Monitorar o tráfego de rede para verificar se os dados transferidos estão criptografados.

**Resultado Esperado:** Todos os dados transferidos entre o cliente e o servidor estão criptografados e seguros (usando HTTPS).

**Pós-condições:** A transferência de dados do usuário permanece segura.

### **Caso de Teste: Auditoria de Acesso e Modificação de Dados do Usuário**

**ID:** TC06-RNF02-Auditoria

**Pré-condições:**

* O sistema deve ter mecanismos de auditoria habilitados.
* O usuário deve realizar várias ações que envolvam acesso e modificação de dados.

**Entrada:** Logs de auditoria gerados pelo sistema.

**Passos de Teste:**

1. Realizar ações de acesso e modificação de dados do usuário (visualizar, editar, deletar).
2. Acessar os logs de auditoria do sistema.
3. Verificar se todas as ações de acesso e modificação estão devidamente registradas nos logs.

**Resultado Esperado:** Todas as ações de acesso e modificação de dados do usuário são registradas nos logs de auditoria com detalhes apropriados (quem, quando, o que).

**Pós-condições:** O sistema mantém um registro detalhado de todas as ações relacionadas aos dados do usuário.

### **Requisito: RNF03 - O serviço deve estar disponível 99,9% do tempo.**

### **Caso de Teste: Monitoramento de Disponibilidade do Serviço**

**ID:** TC01-RNF03-Monitoramento

**Pré-condições:**

* Um sistema de monitoramento de disponibilidade deve estar configurado.
* O sistema de monitoramento deve registrar o tempo total de atividade e inatividade do serviço.

**Entrada:** Dados de disponibilidade coletados pelo sistema de monitoramento.

**Passos de Teste:**

1. Configurar o sistema de monitoramento para registrar a disponibilidade do serviço 24/7 durante um período definido (por exemplo, um mês).
2. Coletar e analisar os dados de disponibilidade após o período definido.
3. Calcular a porcentagem de tempo em que o serviço esteve disponível.

**Resultado Esperado:** O serviço deve estar disponível pelo menos 99,9% do tempo, o que significa que a indisponibilidade não deve exceder 43,2 minutos por mês (30 dias).

**Pós-condições:** A disponibilidade do serviço é registrada e validada.

### **Caso de Teste: Simulação de Falha e Recuperação**

**ID:** TC02-RNF03-FalhaRecuperacao

**Pré-condições:**

* O sistema deve estar em operação normal.
* Deve haver um plano de recuperação de falhas implementado.

**Entrada:** Simulação de uma falha do serviço.

**Passos de Teste:**

1. Simular uma falha no serviço (por exemplo, desconectando o servidor ou interrompendo o serviço).
2. Monitorar o tempo de inatividade e a resposta do plano de recuperação.
3. Registrar o tempo total de inatividade até a recuperação completa do serviço.

**Resultado Esperado:** O tempo de inatividade deve ser minimizado e o serviço deve ser restaurado rapidamente, garantindo que a disponibilidade mensal não caia abaixo de 99,9%.

**Pós-condições:** A capacidade do sistema de recuperar-se rapidamente de falhas é validada.

### **Caso de Teste: Teste de Carga e Estresse**

**ID:** TC03-RNF03-CargaEstresse

**Pré-condições:**

* O serviço deve estar em operação normal.
* Um ambiente de teste deve estar disponível para simular carga e estresse.

**Entrada:** Simulação de alta carga no serviço (por exemplo, número elevado de usuários simultâneos).

**Passos de Teste:**

1. Configurar o ambiente de teste para simular um grande número de usuários acessando o serviço simultaneamente.
2. Executar o teste de carga e estresse durante um período significativo (por exemplo, várias horas).
3. Monitorar a disponibilidade do serviço durante o teste de carga e estresse.

**Resultado Esperado:** O serviço deve permanecer disponível durante o teste de carga e estresse, sem quedas de disponibilidade que reduzam a média mensal abaixo de 99,9%.

**Pós-condições:** A capacidade do serviço de manter alta disponibilidade sob carga é validada.

### **Caso de Teste: Teste de Redundância e Failover**

**ID:** TC04-RNF03-RedundanciaFailover

**Pré-condições:**

* O serviço deve estar configurado com mecanismos de redundância e failover.

**Entrada:** Desativação de um componente crítico do serviço (por exemplo, um servidor principal).

**Passos de Teste:**

1. Identificar os componentes críticos do serviço que possuem redundância e failover.
2. Desativar um componente crítico (por exemplo, derrubando um servidor principal).
3. Monitorar a resposta do sistema de failover e a continuidade do serviço.

**Resultado Esperado:** O sistema de failover deve ativar rapidamente, garantindo que o serviço continue disponível sem interrupção perceptível para os usuários finais.

**Pós-condições:** A eficácia dos mecanismos de redundância e failover é validada.

### **Caso de Teste: Auditoria de Logs de Disponibilidade**

**ID:** TC05-RNF03-AuditoriaLogs

**Pré-condições:**

* O sistema deve ter logs detalhados de disponibilidade e inatividade.

**Entrada:** Logs de disponibilidade do serviço.

**Passos de Teste:**

1. Coletar os logs de disponibilidade do serviço para um período específico (por exemplo, um mês).
2. Auditar os logs para verificar registros de tempos de atividade e inatividade.
3. Calcular a porcentagem de tempo de disponibilidade baseada nos registros de log.

**Resultado Esperado:** A auditoria deve confirmar que o serviço esteve disponível pelo menos 99,9% do tempo durante o período especificado.

**Pós-condições:** A precisão dos logs de disponibilidade e a conformidade com o requisito de disponibilidade são validadas.

### **Caso de Teste: Teste de Atualizações e Manutenções Planejadas**

**ID:** TC06-RNF03-AtualizacoesManutencao

**Pré-condições:**

* O serviço deve ter um plano de manutenção e atualização com períodos de inatividade minimamente invasivos.

**Entrada:** Agendamento de uma atualização ou manutenção do sistema.

**Passos de Teste:**

1. Agendar uma janela de manutenção ou atualização do sistema.
2. Notificar os usuários sobre a janela de manutenção planejada.
3. Executar a manutenção ou atualização conforme agendado.
4. Monitorar o tempo de inatividade durante a manutenção ou atualização.

**Resultado Esperado:** O tempo de inatividade devido a manutenções ou atualizações deve ser minimizado e contabilizado dentro do limite de disponibilidade de 99,9%.

**Pós-condições:** O serviço deve ser restaurado e totalmente funcional após a manutenção ou atualização, e a disponibilidade geral deve permanecer dentro do requisito.

Esses casos de teste cobrem os cenários essenciais para validar o requisito RNF03, garantindo que o serviço esteja disponível pelo menos 99,9% do tempo, mesmo em condições adversas e durante manutenções planejadas.

### **Requisito: RNF04 - O sistema deve ser capaz de suportar até 50 usuários simultâneos.**

### **Caso de Teste: Teste de Carga com 50 Usuários Simultâneos**

**ID:** TC01-RNF04-Carga50Usuarios

**Pré-condições:**

* O ambiente de teste deve estar preparado para simular múltiplos usuários.
* As ferramentas de teste de carga (por exemplo, JMeter, LoadRunner) devem estar configuradas.

**Entrada:** Simulação de 50 usuários simultâneos utilizando as principais funcionalidades do sistema.

**Passos de Teste:**

1. Configurar a ferramenta de teste de carga para simular 50 usuários simultâneos.
2. Definir os cenários de uso que os usuários irão executar (por exemplo, login, navegação entre páginas, criação de agendamentos, etc.).
3. Iniciar o teste de carga e monitorar o desempenho do sistema.
4. Registrar métricas de desempenho, incluindo tempos de resposta, taxas de erro e uso de recursos.

**Resultado Esperado:** O sistema deve suportar 50 usuários simultâneos sem degradação significativa de desempenho, mantendo tempos de resposta aceitáveis e sem falhas críticas.

**Pós-condições:** O sistema continua funcionando corretamente após o teste de carga.

### **Caso de Teste: Teste de Estresse com Mais de 50 Usuários**

**ID:** TC02-RNF04-EstresseMais50Usuarios

**Pré-condições:**

* O ambiente de teste deve estar preparado para simular múltiplos usuários.
* As ferramentas de teste de carga devem estar configuradas.

**Entrada:** Simulação de mais de 50 usuários simultâneos utilizando as principais funcionalidades do sistema.

**Passos de Teste:**

1. Configurar a ferramenta de teste de carga para simular um número crescente de usuários, começando com 50 e aumentando gradualmente.
2. Definir os cenários de uso que os usuários irão executar.
3. Iniciar o teste de estresse e monitorar o desempenho do sistema.
4. Registrar métricas de desempenho, incluindo tempos de resposta, taxas de erro e uso de recursos.
5. Identificar o ponto em que o sistema começa a apresentar degradação significativa de desempenho.

**Resultado Esperado:** O sistema deve suportar 50 usuários simultâneos sem problemas, e a degradação de desempenho deve ser documentada conforme o número de usuários aumenta além disso.

**Pós-condições:** O sistema retorna ao desempenho normal após a redução da carga.

### **Caso de Teste: Teste de Desempenho em Diferentes Funcionalidades**

**ID:** TC03-RNF04-DesempenhoFuncionalidades

**Pré-condições:**

* O ambiente de teste deve estar preparado para simular múltiplos usuários.
* As ferramentas de teste de carga devem estar configuradas.

**Entrada:** Simulação de 50 usuários simultâneos utilizando diferentes funcionalidades do sistema.

**Passos de Teste:**

1. Configurar a ferramenta de teste de carga para simular 50 usuários simultâneos.
2. Dividir os usuários em grupos para realizar diferentes funcionalidades (por exemplo, 20 usuários navegando, 15 realizando agendamentos, 10 atualizando perfil, 5 enviando mensagens).
3. Iniciar o teste de carga e monitorar o desempenho do sistema.
4. Registrar métricas de desempenho, incluindo tempos de resposta, taxas de erro e uso de recursos para cada funcionalidade.

**Resultado Esperado:** O sistema deve suportar 50 usuários simultâneos executando diferentes funcionalidades sem degradação significativa de desempenho.

**Pós-condições:** O sistema continua funcionando corretamente após o teste de carga.

### **Caso de Teste: Teste de Resiliência com 50 Usuários Simultâneos**

**ID:** TC04-RNF04-Resiliencia

**Pré-condições:**

* O ambiente de teste deve estar preparado para simular múltiplos usuários.
* As ferramentas de teste de carga devem estar configuradas.

**Entrada:** Simulação de 50 usuários simultâneos durante eventos de falha simulada.

**Passos de Teste:**

1. Configurar a ferramenta de teste de carga para simular 50 usuários simultâneos.
2. Definir os cenários de uso que os usuários irão executar.
3. Iniciar o teste de carga.
4. Durante o teste, simular falhas no sistema (por exemplo, desconexão de servidores, perda de conexão de banco de dados) e monitorar a resposta do sistema.
5. Registrar métricas de desempenho e resiliência do sistema, incluindo tempos de recuperação e perda de dados.

**Resultado Esperado:** O sistema deve demonstrar resiliência, continuando a operar e recuperando-se rapidamente das falhas simuladas, sem comprometer significativamente a experiência do usuário.

**Pós-condições:** O sistema retorna ao desempenho normal após a recuperação das falhas simuladas.

### **Caso de Teste: Teste de Escalabilidade Horizontal**

**ID:** TC05-RNF04-EscalabilidadeHorizontal

**Pré-condições:**

* O ambiente de teste deve estar preparado para suportar a adição de novos servidores.
* As ferramentas de teste de carga devem estar configuradas.

**Entrada:** Simulação de 50 usuários simultâneos em um ambiente com escalabilidade horizontal configurada.

**Passos de Teste:**

1. Configurar a ferramenta de teste de carga para simular 50 usuários simultâneos.
2. Monitorar o desempenho do sistema com a configuração atual de servidores.
3. Gradualmente aumentar a carga simulada e adicionar novos servidores conforme necessário.
4. Monitorar o desempenho do sistema à medida que novos servidores são adicionados.

**Resultado Esperado:** O sistema deve ser capaz de escalar horizontalmente para suportar a carga adicional, mantendo um desempenho aceitável com 50 usuários simultâneos.

**Pós-condições:** O sistema continua funcionando corretamente e a escalabilidade horizontal é validada.

### **Requisito: RNF05 - O aplicativo deve ser compatível com as últimas versões dos sistemas operacionais Windows e Linux.**

### **Caso de Teste: Compatibilidade com Windows (Última Versão)**

**ID:** TC01-RNF05-Windows

**Pré-condições:**

* O sistema operacional Windows deve estar atualizado para a última versão estável.

**Entrada:** Instalação e execução do aplicativo no Windows.

**Passos de Teste:**

1. Preparar um ambiente de teste com a última versão do sistema operacional Windows.
2. Baixar e instalar o aplicativo no ambiente de teste.
3. Iniciar o aplicativo e realizar uma série de testes funcionais básicos (por exemplo, login, navegação entre páginas, execução de funcionalidades principais).
4. Monitorar o desempenho do aplicativo e verificar se todas as funcionalidades operam conforme o esperado.

**Resultado Esperado:** O aplicativo deve instalar e funcionar corretamente na última versão do Windows, com todas as funcionalidades operando conforme o esperado.

**Pós-condições:** O aplicativo permanece estável e funcional no Windows.

### **Caso de Teste: Compatibilidade com Linux (Última Versão)**

**ID:** TC02-RNF05-Linux

**Pré-condições:**

* O sistema operacional Linux deve estar atualizado para a última versão estável (por exemplo, Ubuntu, Fedora, Debian).

**Entrada:** Instalação e execução do aplicativo no Linux.

**Passos de Teste:**

1. Preparar um ambiente de teste com a última versão do sistema operacional Linux.
2. Baixar e instalar o aplicativo no ambiente de teste.
3. Iniciar o aplicativo e realizar uma série de testes funcionais básicos (por exemplo, login, navegação entre páginas, execução de funcionalidades principais).
4. Monitorar o desempenho do aplicativo e verificar se todas as funcionalidades operam conforme o esperado.

**Resultado Esperado:** O aplicativo deve instalar e funcionar corretamente na última versão do Linux, com todas as funcionalidades operando conforme o esperado.

**Pós-condições:** O aplicativo permanece estável e funcional no Linux.

### **Caso de Teste: Compatibilidade com Diferentes Distribuições de Linux**

**ID:** TC03-RNF05-DistrosLinux

**Pré-condições:**

* Diferentes distribuições de Linux devem estar atualizadas para suas últimas versões estáveis (por exemplo, Ubuntu, Fedora, Debian).

**Entrada:** Instalação e execução do aplicativo em diferentes distribuições de Linux.

**Passos de Teste:**

1. Preparar ambientes de teste com várias distribuições de Linux atualizadas (por exemplo, Ubuntu, Fedora, Debian).
2. Baixar e instalar o aplicativo em cada ambiente de teste.
3. Iniciar o aplicativo em cada ambiente e realizar uma série de testes funcionais básicos (por exemplo, login, navegação entre páginas, execução de funcionalidades principais).
4. Monitorar o desempenho do aplicativo e verificar se todas as funcionalidades operam conforme o esperado em cada distribuição.

**Resultado Esperado:** O aplicativo deve instalar e funcionar corretamente nas diferentes distribuições de Linux, com todas as funcionalidades operando conforme o esperado em cada uma delas.

**Pós-condições:** O aplicativo permanece estável e funcional em todas as distribuições testadas.

### **Caso de Teste: Teste de Funcionalidades Avançadas no Windows**

**ID:** TC04-RNF05-FuncionalidadesWindows

**Pré-condições:**

* O sistema operacional Windows deve estar atualizado para a última versão estável.

**Entrada:** Execução de funcionalidades avançadas do aplicativo no Windows.

**Passos de Teste:**

1. Preparar um ambiente de teste com a última versão do sistema operacional Windows.
2. Baixar e instalar o aplicativo no ambiente de teste.
3. Executar funcionalidades avançadas do aplicativo (por exemplo, relatórios, gráficos, integração com outros sistemas).
4. Monitorar o desempenho e a resposta do aplicativo ao executar essas funcionalidades avançadas.

**Resultado Esperado:** O aplicativo deve executar funcionalidades avançadas corretamente e de forma eficiente no Windows.

**Pós-condições:** O aplicativo permanece estável e funcional ao executar funcionalidades avançadas no Windows.

### **Caso de Teste: Teste de Funcionalidades Avançadas no Linux**

**ID:** TC05-RNF05-FuncionalidadesLinux

**Pré-condições:**

* O sistema operacional Linux deve estar atualizado para a última versão estável.

**Entrada:** Execução de funcionalidades avançadas do aplicativo no Linux.

**Passos de Teste:**

1. Preparar um ambiente de teste com a última versão do sistema operacional Linux.
2. Baixar e instalar o aplicativo no ambiente de teste.
3. Executar funcionalidades avançadas do aplicativo (por exemplo, relatórios, gráficos, integração com outros sistemas).
4. Monitorar o desempenho e a resposta do aplicativo ao executar essas funcionalidades avançadas.

**Resultado Esperado:** O aplicativo deve executar funcionalidades avançadas corretamente e de forma eficiente no Linux.

**Pós-condições:** O aplicativo permanece estável e funcional ao executar funcionalidades avançadas no Linux.

### **Requisito: RNF06 - O aplicativo deve ser intuitivo e fácil de usar, com um tempo máximo de aprendizado de 15 minutos para novos usuários.**

### **Caso de Teste: Teste de Usabilidade e Intuitividade**

**ID:** TC01-RNF06-Usabilidade

**Pré-condições:**

* Selecionar um grupo de novos usuários representativos do público-alvo.
* Preparar um ambiente de teste onde os usuários possam utilizar o aplicativo sem interrupções.

**Entrada:** Primeiro uso do aplicativo por novos usuários.

**Passos de Teste:**

1. Convidar novos usuários para participarem do teste.
2. Fornecer uma breve introdução ao propósito do aplicativo, sem explicações detalhadas sobre seu uso.
3. Pedir aos usuários que realizem uma série de tarefas básicas no aplicativo (por exemplo, login, navegação entre páginas, criação de agendamentos).
4. Monitorar o tempo que cada usuário leva para completar as tarefas.
5. Coletar feedback dos usuários sobre a facilidade de uso e quaisquer dificuldades encontradas.
6. Analisar o tempo médio de aprendizado e a facilidade de uso com base no feedback.

**Resultado Esperado:** A maioria dos usuários deve ser capaz de completar as tarefas básicas dentro de 15 minutos, indicando que o aplicativo é intuitivo e fácil de usar.

**Pós-condições:** Feedback coletado e analisado para identificar áreas de melhoria.

### **Caso de Teste: Teste de Navegação e Fluxo de Trabalho**

**ID:** TC02-RNF06-NavegacaoFluxo

**Pré-condições:**

* Selecionar um grupo de novos usuários representativos do público-alvo.
* Preparar um ambiente de teste onde os usuários possam utilizar o aplicativo sem interrupções.

**Entrada:** Navegação e execução de fluxos de trabalho no aplicativo por novos usuários.

**Passos de Teste:**

1. Convidar novos usuários para participarem do teste.
2. Fornecer um conjunto de tarefas representativas do uso típico do aplicativo (por exemplo, agendar um container, atualizar informações do perfil, visualizar relatórios).
3. Pedir aos usuários que realizem as tarefas sem ajuda externa.
4. Monitorar o tempo que cada usuário leva para navegar e completar os fluxos de trabalho.
5. Coletar feedback dos usuários sobre a facilidade de navegação e qualquer confusão encontrada.
6. Analisar o tempo médio de conclusão das tarefas e a facilidade de navegação com base no feedback.

**Resultado Esperado:** A maioria dos usuários deve ser capaz de navegar e completar os fluxos de trabalho dentro de 15 minutos, indicando que a navegação e o fluxo de trabalho são intuitivos.

**Pós-condições:** Feedback coletado e analisado para identificar áreas de melhoria.

### **Caso de Teste: Teste de Documentação e Recursos de Ajuda**

**ID:** TC03-RNF06-DocumentacaoAjuda

**Pré-condições:**

* Selecionar um grupo de novos usuários representativos do público-alvo.
* Preparar um ambiente de teste onde os usuários possam acessar a documentação e recursos de ajuda do aplicativo.

**Entrada:** Uso da documentação e recursos de ajuda do aplicativo por novos usuários.

**Passos de Teste:**

1. Convidar novos usuários para participarem do teste.
2. Pedir aos usuários que realizem uma série de tarefas no aplicativo, permitindo que utilizem a documentação e os recursos de ajuda conforme necessário.
3. Monitorar o tempo que cada usuário leva para completar as tarefas com a ajuda da documentação e recursos disponíveis.
4. Coletar feedback dos usuários sobre a utilidade e clareza da documentação e recursos de ajuda.
5. Analisar o tempo médio de aprendizado e a eficácia da documentação com base no feedback.

**Resultado Esperado:** A documentação e os recursos de ajuda devem permitir que a maioria dos usuários complete as tarefas básicas dentro de 15 minutos, indicando que são claros e úteis.

**Pós-condições:** Feedback coletado e analisado para identificar áreas de melhoria na documentação e recursos de ajuda.

### **Caso de Teste: Teste de Interface do Usuário (UI)**

**ID:** TC04-RNF06-InterfaceUsuario

**Pré-condições:**

* Selecionar um grupo de novos usuários representativos do público-alvo.
* Preparar um ambiente de teste onde os usuários possam utilizar o aplicativo sem interrupções.

**Entrada:** Primeiro uso do aplicativo com foco na interface do usuário.

**Passos de Teste:**

1. Convidar novos usuários para participarem do teste.
2. Disponibilizar um conjunto de tarefas representativas do uso típico do aplicativo (por exemplo, login, navegação entre páginas, execução de funções comuns).
3. Pedir aos usuários que realizem as tarefas observando a facilidade de uso da interface do usuário (por exemplo, clareza dos botões, menus intuitivos).
4. Monitorar o tempo que cada usuário leva para completar as tarefas.
5. Coletar feedback dos usuários sobre a facilidade de uso da interface do usuário e quaisquer dificuldades encontradas.
6. Analisar o tempo médio de aprendizado e a facilidade de uso da interface com base no feedback.

**Resultado Esperado:** A maioria dos usuários deve ser capaz de completar as tarefas básicas dentro de 15 minutos, indicando que a interface do usuário é intuitiva e fácil de usar.

**Pós-condições:** Feedback coletado e analisado para identificar áreas de melhoria na interface do usuário.

### **Caso de Teste: Teste de Efetividade de Treinamento**

**ID:** TC05-RNF06-Treinamento

**Pré-condições:**

* Selecionar um grupo de novos usuários representativos do público-alvo.
* Preparar um ambiente de teste onde os usuários possam receber um breve treinamento inicial sobre o aplicativo.

**Entrada:** Realização de um treinamento inicial de 15 minutos.

**Passos de Teste:**

1. Convidar novos usuários para participarem do teste.
2. Fornecer um treinamento inicial de 15 minutos, cobrindo as funcionalidades básicas do aplicativo.
3. Pedir aos usuários que realizem uma série de tarefas básicas no aplicativo imediatamente após o treinamento.
4. Monitorar o tempo que cada usuário leva para completar as tarefas.
5. Coletar feedback dos usuários sobre a clareza e eficácia do treinamento.
6. Analisar o tempo médio de aprendizado e a eficácia do treinamento com base no feedback.

**Resultado Esperado:** A maioria dos usuários deve ser capaz de completar as tarefas básicas dentro de 15 minutos após o treinamento, indicando que o treinamento é eficaz e o aplicativo é fácil de usar.

**Pós-condições:** Feedback coletado e analisado para identificar áreas de melhoria no treinamento.

# 9. Relatório de Bugs

ID do Bug: BUG001

Título: Falha no Login com senha correta

Data de Encontro: 4 de outubro de 2023

Encontrado Por: John Doe

Caso de Teste Associado: TC01-S

Severidade: Alta

Status: Aberto

Descrição Detalhada: Ao tentar fazer login com um e-mail e senha válidos, o sistema apresenta um erro.

Passos para Reproduzir: Abrir o app QuickMeals. Ir para a tela de "Login". Inserir e-mail e senha válidos. Clicar em "Entrar".

Resultado Esperado e Atual: Deveria ir para a página inicial, mas permanece na tela de login.

Screenshots ou Vídeos: [Inserir capturas de tela aqui]

Notas Adicionais: O bug ocorre apenas em dispositivos Android.

ID do Bug: BUG002

Título: Falha na Exibição de Temperatura do Carro 1

Data de Encontro: 6 de outubro de 2023

Encontrado Por: Jane Smith

Caso de Teste Associado: TC02-RF03-C1

Severidade: Média

Status: Aberto

Descrição Detalhada: A temperatura do carro 1 não é exibida corretamente na seção de temperatura do aplicativo.

Passos para Reproduzir: Abrir o aplicativo web. Navegar até a seção de temperatura do carro 1.

Resultado Esperado e Atual: Deveria exibir a temperatura do carro 1, mas a informação não é mostrada corretamente.

Screenshots ou Vídeos: [Inserir capturas de tela aqui]

Notas Adicionais: O bug ocorre em dispositivos iOS e Android.

ID do Bug: BUG003

Título: Erro na Alteração de Set Point

Data de Encontro: 8 de outubro de 2023

Encontrado Por: Maria Silva

Caso de Teste Associado: TC02-RF04-F

Severidade: Alta

Status: Aberto

Descrição Detalhada: Ao tentar alterar o set point de um container agendado, o sistema não salva a alteração.

Passos para Reproduzir: Abrir o aplicativo web. Navegar até a seção de containers agendados. Selecionar um container agendado. Tentar alterar o set point.

Resultado Esperado e Atual: Deveria salvar a alteração do set point, mas a mudança não é persistida no sistema.

Screenshots ou Vídeos: [Inserir capturas de tela aqui]

Notas Adicionais: O bug ocorre em todos os dispositivos e sistemas operacionais.

# 10. Conclusão

### **Conquistas dos Testes:**

1. **Validação Funcional:**
   * Todos os requisitos funcionais foram testados e aprovados, demonstrando que a aplicação atende às necessidades de negócio.
2. **Identificação de Problemas:**
   * Foram identificados problemas nos requisitos não funcionais, como desempenho, segurança e disponibilidade, permitindo ações corretivas.
3. **Compatibilidade Testada:**
   * A compatibilidade com diferentes sistemas operacionais foi testada, garantindo que a aplicação possa ser utilizada em uma variedade de ambientes.
4. **Feedback dos Usuários:**
   * O feedback dos usuários durante os testes ajudou a identificar áreas de melhoria na usabilidade e na experiência do usuário.

### **Próximos Passos:**

1. **Correção de Problemas:**
   * Corrigir os problemas identificados nos requisitos não funcionais, como o tempo de carregamento da página de login e a criptografia inadequada dos dados do usuário.
2. **Testes Adicionais:**
   * Realizar testes adicionais para garantir que todas as funcionalidades da aplicação estejam devidamente testadas em diferentes cenários de uso, incluindo casos de uso mais complexos.
3. **Melhorias na Usabilidade:**
   * Implementar melhorias na usabilidade com base no feedback dos usuários, tornando a aplicação mais intuitiva e fácil de usar.
4. **Monitoramento Contínuo:**
   * Implementar um sistema de monitoramento contínuo para acompanhar a disponibilidade e o desempenho da aplicação, garantindo uma experiência consistente para os usuários.
5. **Testes de Regressão:**
   * Realizar testes de regressão após as correções e melhorias para garantir que as alterações não introduzam novos problemas na aplicação.
6. **Atualizações de Documentação:**
   * Atualizar a documentação do projeto com as informações mais recentes sobre os requisitos, funcionalidades e procedimentos de teste.

Com esses próximos passos, podemos garantir que a aplicação continue a evoluir e atender às necessidades dos usuários de forma eficaz e confiável.

# 11. Histórico de Revisões

|  |  |  |  |
| --- | --- | --- | --- |
| **Versão** | **Data** | **Autores** | **Descrição da Mudança** |
| 0.1 | 03/06/2024 | Diego e Ângelo | Criação do documento inicial |
| 0.2 | 05/06/2024 | Diego e Ângelo | Criação dos casos de teste |
| 1.0 | 10/06/2024 | Diego Ângelo | Documento Final |