**COLD START**

**Documento de Requisitos de Software**

Sumário

[Introdução 3](#_Toc160646394)

[Visão Geral do Produto 3](#_Toc160646395)

[Descrição dos Requisitos 4](#_Toc160646396)

[Requisitos Funcionais: 4](#_Toc160646397)

[Requisitos Não Funcionais: 6](#_Toc160646398)

[Casos de Uso 6](#_Toc160646399)

[Matriz de Rastreabilidade 9](#_Toc160646400)

[Anexos e Apêndices 10](#_Toc160646401)

[Histórico de Revisões 11](#_Toc160646402)

[Equipe TCC 12](#_Toc160646403)

**Especificação de Requisitos de Software**

# Introdução

* Objetivo:

Este Documento de Especificação de Requisitos de Software (SRS) serve como a cartilha definitiva para entender, desenvolver e validar o projeto Cold Start. Destina-se a ser o núcleo informativo que engenheiros, desenvolvedores, gerentes de projeto, stakeholders e testadores consultarão em sua jornada para transformar a visão do Cold Start em realidade tangível. Este documento não é apenas um conjunto de especificações, é o DNA codificado do que Cold Start aspira ser.

* Escopo:

O Cold Start é uma plataforma de gerenciamento de contêineres de refrigeração de carros, que visa simplificar a experiência do usuário ao máximo. Imagine ter acesso as temperaturas dos carros de teste ao toque de um botão, com opções de personalização da temperatura desejada, tudo em uma interface de usuário elegante e fácil de usar. Mas não vamos apenas controlar os contêineres; pretendemos economizar energia, tempo e dinheiro através de uma experiência verdadeiramente eficiente.

## Visão Geral do Produto

2.1 Perspectiva do Produto

O Cold Start é uma aplicação web que se concentra em exibir as temperaturas dos contêineres de refrigeração utilizados pela PS para testar a partida de carros em baixa temperatura. Inicialmente disponível pela internet, com foco em mostrar as temperaturas dos carros para o usuário e emitir um aviso quando a temperatura desejada for atingida. O objetivo principal é fornecer uma maneira eficiente e sustentável de gerenciar os testes dia a dia.

2.2 Funções do Produto

* Visualizar Contêineres: o usuário pode visualizar todos os contêineres, junto de seu número e status (Ocupado ou Disponível).
* Visualizar Temperaturas de um Contêiner: o usuário pode visualizar as temperaturas de um contêiner específico.
* Definir temperatura desejada para uma posição de um contêiner: o usuário pode definir a temperatura que deseja e receber um aviso quando ela for atingida.
* Agendar contêiner: o usuário pode agendar o contêiner para dias desejados.
* Gerenciar acessos de usuários: o administrador pode gerenciar o acesso de todos os outros usuários que não sejam administradores.
* Gerenciar agendamentos: o administrador pode gerenciar os agendamentos, caso algum usuário tenha travado um contêiner só para si.
* Login de usuário: os usuários podem fazer login através de seus usuários de rede, caso não possuam acesso, o mesmo pode ser solicitado.

2.3 Restrições

* Conformidade Regulatória

O aplicativo deve estar em conformidade com as leis e regulamentos locais de segurança alimentar e proteção ao consumidor.

* Área Geográfica de Serviço

O serviço estará disponível inicialmente apenas na Bosch Campinas.

* Idiomas Suportados

O aplicativo será lançado inicialmente apenas em português.

* Horário de Operação

O serviço estará disponível 24 horas por dia, com possível baixas para manutenção.

# Descrição dos Requisitos

Descrição: Aqui é onde os requisitos do projeto são minuciosamente detalhados. Isso inclui requisitos funcionais (o que o sistema deve fazer), requisitos não funcionais (como o sistema deve ser), regras de negócio, entre outros.

* 1. Requisitos Funcionais:
  2. Requisitos Não Funcionais:

Exemplo de Preenchimento:

## Requisitos Funcionais:

RF01 – Autenticação de Usuário:

* O usuário deve ser capaz de se autenticar usando o SSO da Bosch.

RF02 – Lista de Containers:

* O aplicativo deve exibir todos os containers na tela inicial.
* O aplicativo deve ser capaz de suportar novos containers que forem adicionados.

RF03 – Visualizar temperaturas:

* O aplicativo deve ser capaz de mostrar a temperatura ambiente do container.
* O aplicativo deve ser capaz de mostrar a temperatura do carro 1 e do carro 2.
* O aplicativo deve ser capaz de mostrar a temperatura do set point 1 e do set point 2.

RF04 - Agenda:

* O usuário deve ser capaz de criar um agendamento para utilizar um container específico em um período de tempo (de 1 dia até no máximo uma semana).
* O usuário deve poder alocar o carro dele se houver uma posição livre no container de outra pessoa.

RF05 - Informações do Restaurante:

* Detalhes como menu, avaliações e tempo estimado de entrega devem ser visíveis.
* O usuário deve poder ver fotos dos pratos oferecidos.

RF06 - Carrinho de Compras:

* O usuário deve ser capaz de adicionar itens ao carrinho.
* O usuário deve poder visualizar os itens no carrinho antes do checkout.
* Função de "Salvar para Depois" para itens que o usuário não quer comprar imediatamente.

RF07 - Checkout:

* O usuário deve poder escolher entre várias opções de pagamento (cartão de crédito, PayPal, etc.).
* Deve haver uma opção para inserir um endereço de entrega ou escolher um endereço previamente salvo.

RF08 - Rastreamento de Pedido:

* O usuário deve receber atualizações em tempo real sobre o status do pedido ("Preparando", "Saiu para entrega", etc.).
* Deve haver um mapa mostrando a localização atual do entregador.

RF09 - Avaliação e Comentários:

* Após a entrega, o usuário deve poder avaliar o restaurante e o entregador.
* O usuário deve poder escrever um comentário e dar uma nota de 1 a 5 estrelas.

RF10 - Histórico de Pedidos:

* O usuário deve poder visualizar um histórico de todos os pedidos feitos.
* O histórico deve incluir detalhes como data do pedido, itens comprados e custo total.

RF11 - Ofertas e Promoções:

* O usuário deve poder visualizar ofertas e promoções atuais na página inicial do aplicativo.
* O usuário deve ser capaz de aplicar códigos de desconto no checkout.

RF12 - Notificações:

* O usuário deve receber notificações push para atualizações importantes, como alterações no status do pedido ou novas promoções.

## Requisitos Não Funcionais:

RNF01 - Tempo de Carregamento:

* Todas as páginas do aplicativo devem ser carregadas em até 2 segundos em uma conexão de internet padrão.

RNF02 - Segurança:

* Todos os dados do usuário devem ser criptografados e armazenados de forma segura.

RNF03 - Disponibilidade:

* O serviço deve estar disponível 99,9% do tempo.

RNF04 - Escalabilidade:

* O sistema deve ser capaz de suportar até 10.000 usuários simultâneos.

RNF05 - Compatibilidade:

* O aplicativo deve ser compatível com as duas últimas versões dos sistemas operacionais Android e iOS.

RNF06 - Usabilidade:

O aplicativo deve ser intuitivo e fácil de usar, com um tempo máximo de aprendizado de 15 minutos para novos usuários.

# Casos de Uso

Descrição: Esta seção descreve, em termos de "atores" e suas interações com o sistema, como o software será usado. Cada caso de uso oferece cenários que ilustram o fluxo da atividade, ajudando assim no entendimento e na construção do sistema.

Exemplo de Preenchimento:

Caso de Uso 1: Registro de Usuário (RF01)

* Ator Principal: Novo Usuário
* Pré-condição: O usuário não está registrado no sistema.
* Fluxo Principal:
  + - 1. O usuário seleciona "Registrar-se".
      2. O sistema apresenta os campos de registro.
      3. O usuário fornece e-mail, senha e número de telefone.
      4. O sistema envia um código de verificação via SMS ao número fornecido.
      5. O usuário insere o código de verificação.
      6. O sistema valida o código e completa o registro.
      7. Pós-condição: O usuário está registrado e logado no sistema.

Caso de Uso 2: Autenticação de Usuário (RF02)

* Ator Principal: Usuário Registrado
* Pré-condição: O usuário está registrado, mas não está logado.
* Fluxo Principal:

1. O usuário seleciona "Login".
2. O usuário insere seu e-mail e senha.
3. O sistema valida as credenciais.
4. O usuário é autenticado e tem acesso às funcionalidades do sistema.
5. Pós-condição: O usuário está logado e tem acesso às funcionalidades disponíveis para usuários autenticados.

Caso de Uso 3: Lista de Restaurantes (RF03) <EXEMPLO RESUMIDO>

* Ator Principal: Usuário Logado
* Fluxo Principal:

1. O usuário logado acessa a página inicial e vê uma lista de restaurantes baseada em sua localização.

Caso de Uso 4: Filtragem de Restaurantes (RF04) <EXEMPLO RESUMIDO>

* Ator Principal: Usuário Logado
* Fluxo Principal:

1. O usuário logado aplica filtros como tipo de cozinha, avaliação e preço para encontrar restaurantes que correspondam aos critérios.

Caso de Uso 5: Informações do Restaurante (RF05) <EXEMPLO RESUMIDO>

* Ator Principal: Usuário Logado
* Fluxo Principal:

1. O usuário clica em um restaurante específico para ver mais informações, como menu, avaliações e tempo de entrega estimado.

Caso de Uso 6: Carrinho de Compras (RF06) <EXEMPLO RESUMIDO>

* Ator Principal: Usuário Logado
* Fluxo Principal:

1. O usuário adiciona itens ao carrinho e pode visualizar ou modificar esses itens antes do checkout.

Caso de Uso 7: Checkout (RF07) <EXEMPLO RESUMIDO>

* Ator Principal: Usuário Logado
* Fluxo Principal:

1. O usuário revisa os itens no carrinho, fornece informações de pagamento e conclui o pedido.

Caso de Uso 8: Rastreamento de Pedido (RF08) <EXEMPLO RESUMIDO>

* Ator Principal: Usuário Logado
* Fluxo Principal:

1. Após realizar o pedido, o usuário pode rastrear o status e a localização do pedido em tempo real.

Caso de Uso 9: Avaliação e Comentários (RF09)

A: Avaliar Restaurante (RF09)

* Atores Principais: Usuário Logado, Restaurante
* Pré-condição: O pedido foi entregue ao usuário.
* Fluxo Principal:

1. O usuário acessa a seção "Histórico de Pedidos".
2. O usuário seleciona o pedido recente.
3. O sistema exibe uma opção para avaliar o restaurante.
4. O usuário dá uma classificação de estrelas e deixa um comentário opcional.
5. O sistema registra a avaliação e atualiza a classificação geral do restaurante.
6. Pós-condição: A avaliação do usuário está visível para outros usuários e para o restaurante.

9B: Visualizar e Responder Avaliações (RF09)

* Atores Principais: Restaurante
* Pré-condição: O restaurante está listado no aplicativo.
* Fluxo Principal:

1. O restaurante acessa seu painel de controle.
2. O sistema exibe as avaliações recentes.
3. O restaurante escolhe uma avaliação para responder.
4. O sistema permite que o restaurante escreva uma resposta.
5. O restaurante envia a resposta.
6. Pós-condição: A resposta do restaurante fica visível para os usuários que visualizam a avaliação.

Caso de Uso 10: Histórico de Pedidos (RF10) <EXEMPLO RESUMIDO>

* Ator Principal: Usuário Logado
* Fluxo Principal:

1. O usuário acessa uma seção separada onde pode ver o histórico de todos os pedidos realizados.

Caso de Uso 11: Ofertas e Promoções (RF11) <EXEMPLO RESUMIDO>

* Ator Principal: Usuário Logado
* Fluxo Principal:

1. O usuário vê as ofertas atuais na página inicial e pode aplicar códigos de desconto durante o checkout.

Caso de Uso 12: Notificações (RF12) <EXEMPLO RESUMIDO>

* Ator Principal: Usuário Logado
* Fluxo Principal:

1. O usuário recebe notificações push para atualizações importantes, como mudanças no status do pedido ou novas promoções.

# Matriz de Rastreabilidade

Descrição: A Matriz de Rastreabilidade é uma tabela usada para rastrear os requisitos ao longo do ciclo de vida de um projeto. Ela serve para entender a relação entre os requisitos e os componentes do sistema que os implementam, assim como outros artefatos do projeto, como casos de teste ou documentos.

* ID do Requisito: Identificação única do requisito.
* Descrição do Requisito: Uma breve descrição do que o requisito envolve.
* Fonte do Requisito: Quem solicitou o requisito ou de onde ele veio.
* Caso de Uso Associado: O caso de uso que implementa ou é relacionado a esse requisito.
* Componente do Sistema: A parte do sistema que está diretamente relacionada ao requisito.
* Casos de Teste: Identificações dos casos de teste que validam o requisito.

Observação: não é obrigatório ter todas essas informações na Matriz de Rastreabilidade. A matriz pode ser adaptada para atender às necessidades específicas do seu projeto. Algumas matrizes podem ser bastante simples, incluindo apenas os IDs dos requisitos e os componentes correspondentes do sistema. Outras podem ser mais complexas e incluir informações adicionais, como responsáveis, estados dos requisitos (por exemplo, "Não Iniciado", "Em Desenvolvimento", "Concluído"), entre outros.

A ideia é que a Matriz de Rastreabilidade seja uma ferramenta útil para o gerenciamento do projeto. Portanto, ela deve conter as informações que são mais relevantes para ajudar a equipe a entender as relações entre os diversos componentes e requisitos do projeto.

Exemplo de Preenchimento:

Tabela

Descrição gerada automaticamente

# Histórico de Revisões

|  |  |  |  |
| --- | --- | --- | --- |
| **Versão** | **Data** | **Autor(es)** | **Descrição da Mudança** |
| 0.1 | 07/03/2024 | Diego Lopes e Ângelo Carnevale | Criação do Documento Inicial |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Equipe TCC

|  |
| --- |
| **Nome** |
| Diego Castan Lopes |
| Ângelo Carnevale |
| Giovanna Vieira |