

DRP – Documento de Requisitos de Projeto: IOTCARGO

11/07/2024

Versão v1.1.2

1. DESCRIÇÃO GERAL DO PROJETO	4
2. ESCOPO DO PROJETO	4
2.1 Autenticação de usuários	4
2.2 Aquisição de dados de telemetria	4
2.3 Visualização de dados	4
2.4 Geração de relatórios	4
2.5 Gerenciamento de usuários	4
2.6 Gerenciamento de filiais	4
2.7 Segurança	5
2.8 Escalabilidade	5
2.9 Disponibilidade	5
2.10 Monitoramento	5
3. REQUISITOS FUNCIONAIS	5
3.1 Tela de apresentação	5
3.1.1 Funcionalidades	5
3.2 Tela de login	5
3.3 Tela de dashboard	6
3.3.1 Dados apresentado	6
3.3.2 Funcionalidades	6
3.3.3 Informações da API Fulltrack	6
3.4 Tela de gerar relatórios	7
3.4.1 Funcionalidades	7
3.5 Tela de gerenciamento de conta	7
3.5.1 Funcionalidades	7
3.5.2 Tela cadastro usuário	7
3.5.3 Tela cadastro filiais	7
3.5.4 Tela cadastro carreta (implemento)	7
3.6 Segurança	7

3.6.1	Funcionalidades.....	8
3.7	Disponibilidade.....	8
3.7.1	Funcionalidades.....	8
3.8	Monitoramento.....	8
3.8.1	Funcionalidades.....	8
4.	REQUISITOS NÃO FUNCIONAIS.....	8
4.1	Desempenho	8
4.2	Usabilidade.....	9
4.3	Infraestrutura de Servidores	9
4.3.1	Vultr.....	9
4.3.2	Azure	9

1. DESCRIÇÃO GERAL DO PROJETO

O IOTCARGO visa solucionar problemas de monitoramento em tempo real da carreta, fornecendo dados para melhorar a logística. Este documento tem como objetivo detalhar os requisitos para o desenvolvimento da aplicação web de telemetria. A aplicação será construída utilizando o framework Angular e hospedada no servidor na nuvem Vultr. Também será utilizado Nginx para realizar o proxy reverso entre as duas APIs e possuirá assinatura SSL criada pelo Let's Encrypt.

2. ESCOPO DO PROJETO

2.1 Autenticação de usuários

Descrição: Integração com API externa para controle de login e autenticação dos usuários.

2.2 Aquisição de dados de telemetria

Descrição: Integração com API de terceiros para receber dados de telemetria em tempo real de 30 a 50 mil carretas.

2.3 Visualização de dados

Descrição: Criação de um dashboard para visualização dos dados telemetria em tempo real e histórico.

2.4 Geração de relatórios

Descrição: Implementação de funcionalidade para gerar relatórios personalizados dos dados de telemetria.

2.5 Gerenciamento de usuários

Descrição: Criação de telas para cadastro, consulta e edição de usuários.

2.6 Gerenciamento de filiais

Descrição: Criação de telas para cadastro, consulta e edição de filiais.

2.7 Segurança

Descrição: Implementação de medidas de segurança para proteger o acesso à aplicação e aos dados.

2.8 Escalabilidade

Descrição: A arquitetura da aplicação deve ser escalável para suportar um número crescente de usuários, sendo suportado inicialmente 300 a 500 usuários simultâneos.

2.9 Disponibilidade

Descrição: A aplicação deve estar sempre disponível para uso.

2.10 Monitoramento

Descrição: Implementação de mecanismos para monitorar o desempenho da aplicação e identificar falhas.

3. REQUISITOS FUNCIONAIS

3.1 Tela de apresentação

Descrição: Apresentar informações sobre o projeto e seus benefícios.

3.1.1 Funcionalidades

- Deve ter opção de direcionamento para tela login.

3.2 Tela de login

Descrição: Permitir que o usuário faça login para acessar os dados dos implementos cadastrados.

- Deve permitir que os usuários autentiquem-se utilizando e-mail e senha.
- Deve exibir mensagens de erro em caso de credencias inválidas.
- Deve exibir mensagens de erro quando dados do usuário não for informado ou estiver em formato inválido.

3.3 Tela de dashboard

Descrição: Mostrar os dados dos implementos cadastrados do cliente de forma clara e intuitiva através da API fornecida pela Fulltrack.

3.3.1 Dados apresentado

- Placa da carreta.
- Número de série do equipamento instalado.
- Odômetro da distância total percorrida pela carreta.
- Luz ABS.
- Estado de conexão do painel solar.
- Estado de violação do módulo.
- Localização da última atualização onde ao clicar é direcionado para o Google Map.
- Data da última atualização.

3.3.2 Funcionalidades

- Deve permitir a alteração da visualização dos dados, no formato cartão ou tabela.
- Deve permitir realizar filtro por placa dos implementos mostrado na tela.
- Deve permitir a filtragem na tabela por número de série, odômetro e data da última atualização.
- Deve possuir opção de recarregar novos dados.
- Deve possuir opção de gerar relatório.

3.3.3 Informações da API Fulltrack

Descrição: Fornece os dados usado para a telemetria da carreta, onde é possível realizar 3000 requistes por minuto e estes dados são armazenados no banco por 3 meses.

Dados usado da requeste:

- **ibutton:** Contém os dados da placa, odômetro, luz ABS, painel solar e status de violado do equipamento em formato hexadecimal.
- **Latitude e longitude:** Localização do equipamento.
- **Data da última comunicação:** Através da data do GRPs é informado a última comunicação com a API.
- **Número de série:** Através do campo veículo.

3.4 Tela de gerar relatórios

Descrição: Realiza relatório da carreta.

3.4.1 Funcionalidades

- Deve permitir que o usuário defina o período dos dados e o formato do relatório, tendo o período entre a data atual a 90 dias.
- Deve permitir a exportação do relatório para PDF ou Excel.
- Deve possuir no arquivo os dados mostrados no dashboard.

3.5 Tela de gerenciamento de conta

Descrição: Permite que gerente administrador crie novas contas de usuários de nível comum, cadastro de filias e vinculação das carretas ao usuários e filiais.

3.5.1 Funcionalidades

- Deve permitir que o gestor cadastre até 10 usuários e 50 filiais.
- Deve permitir que o gestor vincule as carretas às respectivas filiais.

3.5.2 Tela cadastro usuário

- Campos obrigatórios: nome, filial, e-mail e senha.
- Deve possuir opção de editar, deletar e criar novo usuário.

3.5.3 Tela cadastro filiais

- Campos obrigatórios: nome da filial, CEP, estado, rua, bairro e cidade.
- Deve possuir opção de editar, deletar e criar novo usuário.

3.5.4 Tela cadastro carreta (implemento)

- Deve possuir função de filtro por placa.
- Opção de editar filial responsável pela carreta
- Campos à ser mostrado: placa, número de série, filial e estado do módulo (instalado ou em estoque).

3.6 Segurança

Descrição: Fornece confiabilidade e proteção para o usuário.

3.6.1 Funcionalidades

- Deve possuir arquitetura escalável para suportar número crescente de usuários, onde o seu inicial é entre 300 a 500 usuários simultâneo.
- A aplicação deve ser capaz de se integrar com outros sistemas e APIs, incluindo integração cruciais com APIs de login e Fulltrack.
 - A integração com a API de login deve suporta entre 300 a 500 acessos.
 - A integração com a API Fulltrack deve suportar entre 30 a 50 mil requests diários, assegurando performance e confiabilidade.

3.7 Disponibilidade

Descrição: Permitirá acesso a aplicação constante.

3.7.1 Funcionalidades

- A aplicação deve ter mecanismos de failover para garantir a alta disponibilidade.

3.8 Monitoramento

Descrição: Realizara monitoramento com o objetivo de manter o constante desempenho da aplicação.

3.8.1 Funcionalidades

- Deve possuir monitoramento para identificar falhas e gargalos de desempenho.
- Deve possuir sistema de armazenamento de logs de erros e logs de atividades dos usuários para auditoria.

4. REQUISITOS NÃO FUNCIONAIS

4.1 Desempenho

- Possuir bom tempo de resposta.
- A aplicação deve ser capaz de processar um grande volume de dados de telemetria sem comprometer o desempenho.
- Definir métricas específicas para o tempo de resposta e throughput.

- Realizar testes de carga e estresse para garantir que o sistema funcione sobre alta demanda.

4.2 Usabilidade

- A interface da aplicação deve ser intuitiva e fácil de usar.
- A interface deve possuir responsividade para mobile e desktop.
- A interface deve possuir acessibilidade para tema claro e escuro.

4.3 Infraestrutura de Servidores

Descrição: Atualmente está sendo usado dois tipos de servidor de hospedagem.

4.3.1 Vultr

Descrição: Encontra-se o front-end desenvolvido em Angular com o nginx servindo de web serve para a ligação entre a API de login e a Fulltrack.

Configuração:

- 1 vCore
- 2GB de RAM
- 500 GB de armazenamento NVME

4.3.2 Azure

Descrição: Encontra-se o back-end desenvolvido com Node.js e Express onde está sendo usado um container docker para realizar a web serve.

Configuração – Azure Basic A2:

- 2 vCPUs
- 3.1GiB de RAM
- 30 GiB de armazenamento