***GUARDIÃO VEICULAR***

***DRS – DOCUMENTO DE REQUISITOS DE SOFTWARE***

***SUMÁRIO***

|  |  |  |
| --- | --- | --- |
| ***1*** | ***INTRODUÇÃO*** | ***3*** |
| ***2*** | ***REQUISITOS FUNCIONAIS*** | ***4*** |
| ***3*** | ***REQUISITOS NÃO FUNCIONAIS*** | ***9*** |
| ***4*** | ***REQUISITOS DE DESEMPENHO*** | ***13*** |
| ***5*** | ***REQUISITOS DE USABILIDADE*** | ***13*** |
| ***6*** | ***REQUISITOS DE CONFIABILIDADE*** | ***13*** |
| ***7*** | ***RFQUISITOS DE SEGURANÇA*** | ***14*** |
| ***8*** | ***REQUISITOS DE* MANUTENIBILIDADE** | ***14*** |
| ***9*** | ***REQUISITOS DE PORTABILIDADE*** | ***14*** |
| ***10*** | ***ASSINATURAS*** | ***15*** |

**INTRODUÇÃO**

* 1. **Propósito**

O propósito deste documento é descrever os requisitos funcionais e não funcionais do sistema Guardião Veicular. Ele servirá como base para o desenvolvimento, testes aceitação do software.

* 1. **Escopo do Sistema**

O sistema Guardião Veicular será responsável por oferecer uma solução de baixo custo para o monitoramento e segurança de veículos. O escopo inclui as seguintes funcionalidades: localização em tempo real, alertas de segurança proativos e gestão de múltiplos veículos através de uma interface web responsiva e intuitiva.

**1.3. Definições, Acrônimos e Abreviações**

* **DRS:** Documento de Requisitos de Software
* **RF:** Requisito Funcional
* **RNF:** Requisito Não Funcional
* **API (Application Programming Interface):** Conjunto de rotinas e padrões de programação para acesso a um aplicativo de software ou plataforma baseado na Web.
* **Backend:** A parte do sistema que roda no servidor, responsável pela lógica de negócio, acesso ao banco de dados e comunicação com o hardware.
* **ESP32:** Microcontrolador de baixo custo com Wi-Fi e Bluetooth integrados.
* **Firmware:** Software embarcado no microcontrolador (ESP32) para controlar o hardware.
* **Frontend:** A parte do sistema com a qual o usuário interage.
* **Geofencing (Cerca Virtual):** Delimitação de uma área geográfica virtual que dispara um alerta quando o dispositivo entra ou sai dela.
* **GPS (Global Positioning System):** Sistema de posicionamento global utilizado para obter as coordenadas geográficas.
* **IoT (Internet of Things):** Rede de dispositivos físicos ("coisas") que possuem sensores, software e outras tecnologias para se conectar e trocar dados pela internet.
* **MQTT (Message Queuing Telemetry Transport):** Protocolo de mensagens leve, ideal para a comunicação entre dispositivos IoT e servidores.
* **PWA (Progressive Web App):** Aplicação web que oferece uma experiência de usuário similar a de um aplicativo nativo, incluindo a capacidade de ser "instalada" e receber notificações push.
* **WebSocket:** Protocolo de comunicação que permite uma conexão interativa e em tempo real entre o navegador do cliente e o servidor.

1. **Requisitos Funcionais (RF)**

Para estabelecer a prioridade dos requisitos foram adotadas as denominações “essencial”, “importante” e “desejável”. A prioridade dos requisitos é utilizada no gerenciamento do escopo das etapas do projeto e na definição das prioridades durante o desenvolvimento do sistema.

* **Alta (1)**: Requisito essencial cujo fracasso em sua implementação significa que o sistema não irá atender as necessidades do cliente. Imprescindível que seja atendido pelo sistema, condição fundamental para o sucesso do projeto.
* **Média (2)**: Requisitos importantes para a eficácia ou eficiência do sistema. Sua não implementação afeta a satisfação do usuário e/ou o valor agregado do produto e o não atendimento não determina o fracasso do projeto.

**Baixa (3)**: Requisitos úteis, porém menos críticos, sendo usados menos frequentemente. Não possui muito significado para a satisfação do usuário e pode deixar de ser atendida

|  |  |
| --- | --- |
| **ID:** RF001 | **Nome do Requisito:** Cadastro de Novo Usuário |
| **Categoria:** Autenticação | **Prioridade:** Alta |
| **Descrição:** | O sistema deve permitir que um novo usuário se cadastre na plataforma, fornecendo nome, e-mail e senha. |
| **Informações:** | Entrada: Nome, E-mail, Senha.  Saída: Confirmação de cadastro. |
| **Regras de Negócio:** | O e-mail fornecido deve ser único na base de dados. A senha deve ter no mínimo 8 caracteres. |
| **ID:** RF002 | **Nome do Requisito:** Login de Usuário |
| **Categoria:** Autenticação | **Prioridade:** Alta |
| **Descrição:** | O sistema deve permitir que um usuário existente faça login. |
| **Informações:** | Entrada: E-mail, Senha.  Saída: Token de autenticação. |
| **Regras de Negócio:** | Após 5 tentativas sem sucesso, a conta deve ser bloqueada temporariamente. |
| **ID:** RF003 | **Nome do Requisito:** Login Social com Google |
| **Categoria:** Autenticação | **Prioridade:** Média |
| **Descrição:** | O sistema deve permitir o cadastro e login simplificado via contas Google. |
| **Informações:** | Entrada: Credencial Google.  Saída: Token de autenticação. |
| **Regras de Negócio:** | Obter apenas informações essenciais (nome, e-mail) com permissão do usuário. |
| **ID:** RF004 | **Nome do Requisito:** Recuperação de Senha |
| **Categoria:** Autenticação | **Prioridade:** Média |
| **Descrição:** | O sistema deve prover uma funcionalidade para redefinição de senha. |
| **Informações:** | Entrada: E-mail.  Saída: E-mail com link de redefinição. |
| **Regras de Negócio:** | O link de redefinição deve ser de uso único e expirar após 1 hora. |
| **ID:** RF005 | **Nome do Requisito:** Adicionar Novo Veículo |
| **Categoria:** Gestão de Veículos | **Prioridade:** Alta |
| **Descrição:** | Permitir que um usuário adicione um ou mais veículos à sua conta. |
| **Informações:** | Entrada: Dados do Veículo.  Saída: Confirmação de adição. |
| **Regras de Negócio:** | Um mesmo ID de Dispositivo não pode ser cadastrado por mais de um usuário. |
| **ID:** RF006 | **Nome do Requisito:** Detalhes do Veículo |
| **Categoria:** Gestão de Veículos | **Prioridade:** Alta |
| **Descrição:** | O usuário deve informar os detalhes do veículo ao cadastrá-lo. |
| **Informações:** | Entrada: Apelido, ID do Dispositivo, Placa, Tipo, Marca/Modelo, Cor, Ano. |
| **Regras de Negócio:** | "Apelido" e "ID do Dispositivo" são campos obrigatórios. |
| **ID:** RF007 | **Nome do Requisito:** Listar Veículos |
| **Categoria:** Gestão de Veículos | **Prioridade:** Alta |
| **Descrição:** | O sistema deve exibir uma lista de todos os veículos cadastrados pelo usuário. |
| **Informações:** | Entrada: N/A.  Saída: Lista de veículos associados à conta. |
| **Regras de Negócio:** | A lista deve exibir um status simplificado (Online/Offline) para cada veículo. |
| **ID:** RF008 | **Nome do Requisito:** Editar Veículo |
| **Categoria:** Gestão de Veículos | **Prioridade:** Média |
| **Descrição:** | O sistema deve permitir ao usuário editar as informações de um veículo. |
| **Informações:** | Entrada: Novos dados do veículo.  Saída: Confirmação de alteração. |
| **Regras de Negócio:** | O ID do Dispositivo não pode ser alterado via edição. |
| **ID:** RF009 | **Nome do Requisito:** Remover Veículo |
| **Categoria:** Gestão de Veículos | **Prioridade:** Média |
| **Descrição:** | O sistema deve permitir ao usuário remover um veículo de sua conta. |
| **Informações:** | Entrada: ID do Veículo a ser removido.  Saída: Confirmação de remoção. |
| **Regras de Negócio:** | A remoção exige uma confirmação do usuário antes de ser efetivada. |
| **ID:** RF010 | **Nome do Requisito:** Exibição no Mapa |
| **Categoria:** Monitoramento | **Prioridade:** Alta |
| **Descrição:** | Exibir em um mapa interativo a última localização conhecida do veículo. |
| **Informações:** | Entrada: ID do Veículo.  Saída: Coordenadas (latitude, longitude). |
| **Regras de Negócio:** | Se a localização for mais antiga que 24h, o marcador deve indicar "desatualizado". |
| **ID:** RF011 | **Nome do Requisito:** Atualização em Tempo Real |
| **Categoria:** Monitoramento | **Prioridade:** Alta |
| **Descrição:** | A posição do marcador no mapa deve ser atualizada em tempo real via WebSockets. |
| **Informações:** | Entrada: Evento de WebSocket com novas coordenadas.  Saída: Posição do marcador atualizada. |
| **Regras de Negócio:** | A atualização só deve ocorrer se o usuário estiver com o dashboard ativo. |
| **ID:** RF012 | **Nome do Requisito:** Endereço Reverso |
| **Categoria:** Monitoramento | **Prioridade:** Alta |
| **Descrição:** | Exibir o endereço textual correspondente às coordenadas atuais do veículo. |
| **Informações:** | Entrada: Coordenadas.  Saída: String de endereço formatado. |
| **Regras de Negócio:** | A conversão deve usar uma API externa (ex: Google Maps Geocoding). |
| **ID:** RF013 | **Nome do Requisito:** Exibição de Status |
| **Categoria:** Monitoramento | **Prioridade:** Alta |
| **Descrição:** | Exibir o status atual (Estacionado/Em Movimento) e a velocidade do veículo. |
| **Informações:** | Entrada: Dados de telemetria (velocidade).  Saída: Status textual e numérico. |
| **Regras de Negócio:** | "Em Movimento" só deve ser exibido para velocidades > 5 km/h. |
| **ID:** RF014 | **Nome do Requisito:** Seletor de Veículo |
| **Categoria:** Monitoramento | **Prioridade:** Alta |
| **Descrição:** | O usuário deve ser capaz de alternar a visualização entre seus veículos. |
| **Informações:** | Entrada: ID do novo veículo selecionado.  Saída: Dados completos do novo veículo. |
| **Regras de Negócio:** | A troca de veículo deve recarregar todos os dados do dashboard. |
| **ID:** RF015 | **Nome do Requisito:** Armar/Desarmar Alarme |
| **Categoria:** Segurança | **Prioridade:** Alta |
| **Descrição:** | Permitir que o usuário "arme" e "desarme" o alarme para um veículo. |
| **Informações:** | Entrada: ID do Veículo, Estado (armar/desarmar).  Saída: Confirmação de estado. |
| **Regras de Negócio:** | O comando deve ser enviado ao hardware para confirmação. |
| **ID:** RF016 | **Nome do Requisito:** Alerta de Vibração |
| **Categoria:** Segurança | **Prioridade:** Alta |
| **Descrição:** | Gerar um alerta de "Vibração Suspeita" se o hardware detectar um impacto. |
| **Informações:** | Entrada: Evento de alerta do hardware.  Saída: Notificação para o usuário. |
| **Regras de Negócio:** | O limiar de vibração deve ser configurável no firmware. |
| **ID:** RF017 | **Nome do Requisito:** Alerta de Invasão |
| **Categoria:** Segurança | **Prioridade:** Alta |
| **Descrição:** | Gerar um alerta de "Invasão Detectada" se o hardware detectar a abertura de porta. |
| **Informações:** | Entrada: Evento de alerta do hardware.  Saída: Notificação para o usuário. |
| **Regras de Negócio:** | Este alerta tem prioridade máxima sobre o de vibração. |
| **ID:** RF018 | **Nome do Requisito:** Envio de Notificação |
| **Categoria:** Segurança | **Prioridade:** Alta |
| **Descrição:** | O sistema deve enviar uma notificação push quando um alerta for gerado. |
| **Informações:** | Entrada: Tipo de alerta, dados do veículo.  Saída: Notificação Push. |
| **Regras de Negócio:** | Múltiplos alertas do mesmo tipo devem ser agrupados a cada 5 minutos. |
| **ID:** RF019 | **Nome do Requisito:** Seleção de Data |
| **Categoria:** Histórico | **Prioridade:** Alta |
| **Descrição:** | Permitir que o usuário selecione uma data para consultar o histórico de um veículo. |
| **Informações:** | Entrada: ID do Veículo, Data.  Saída: Dados do trajeto. |
| **Regras de Negócio:** | O sistema deve permitir a consulta de um histórico de, no mínimo, 90 dias. |
| **ID:** RF020 | **Nome do Requisito:** Desenho da Rota no Mapa |
| **Categoria:** Histórico | **Prioridade:** Alta |
| **Descrição:** | Desenhar no mapa o trajeto completo realizado por um veículo na data selecionada. |
| **Informações:** | Entrada: Lista de coordenadas do trajeto.  Saída: Linha desenhada no mapa. |
| **Regras de Negócio:** | O trajeto deve indicar visualmente o ponto de partida e o de chegada. |
| **ID:** RF021 | **Nome do Requisito:** Resumo do Trajeto |
| **Categoria:** Histórico | **Prioridade:** Média |
| **Descrição:** | Exibir um resumo com dados do trajeto (distância, velocidade máxima, etc.). |
| **Informações:** | Entrada: Lista de coordenadas e timestamps.  Saída: Dados estatísticos do trajeto. |
| **Regras de Negócio:** | A distância deve ser calculada somando a distância entre os pontos. |

1. **Requisitos Não Funcionais (RNF)**

Os requisitos não funcionais descrevem **como** o sistema deve se comportar. Eles se referem a qualidades do sistema como desempenho, segurança e usabilidade

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Prioridade** | **ID** | **Nome** | **Descrição** | **Interfaces e dependências** |
| Alta | RNF01 | Latência de Atualização | A atualização da posição do veículo no mapa (via WebSocket) deve ocorrer em menos de 2 segundos após o recebimento do dado pelo backend. | Frontend, Backend e Hardware (comunicação WebSocket) |
| Alta | RNF02 | Tempo de Carregamento | O tempo de carregamento da tela do dashboard principal não deve exceder 3 segundos em uma conexão de internet banda larga padrão. | Frontend e Backend (otimização de carregamento de dados) |
| Alta | RNF03 | Resposta da API | A API do backend deve ter um tempo de resposta médio inferior a 500ms para requisições padrão. | Backend |
| Alta | RNF04 | Interface como PWA | A interface web deve ser desenvolvida como um Progressive Web App (PWA), garantindo uma experiência otimizada para dispositivos móveis, capacidade de ser "instalada" na tela inicial e de receber notificações push. | Frontend (desenvolvimento da interface) |
| Média | RNF05 | Acessibilidade | As ações principais (visualizar mapa, armar/desarmar alarme) devem ser acessíveis a partir da tela principal com o mínimo de cliques. | Frontend (design de interface e experiência do usuário) |
| Média | RNF06 | Feedback Visual | O sistema deve fornecer feedback visual claro e imediato para as ações do usuário (ex: mensagens de sucesso, erro, indicadores de carregamento). | Frontend (design de interface e experiência do usuário) |
| Média | RNF07 | Intuitividade | A interface deve ser intuitiva, minimizando a necessidade de treinamento para um novo usuário. | Frontend (design de interface e experiência do usuário) |
| Alta | RNF08 | Reconexão Automática do Hardware | O firmware do dispositivo deve ser capaz de se reconectar automaticamente à rede Wi-Fi/GSM caso a conexão seja perdida. | Hardware (firmware do dispositivo) |
| Alta | RNF09 | Disponibilidade do Servidor (Uptime) | O backend do sistema deve ser projetado para ter uma disponibilidade (uptime) de 99.5% ou superior. | Backend |
| Alta | RNF10 | Backup do Banco de Dados | Devem ser realizados backups automáticos e periódicos do banco de dados para garantir a recuperação em caso de falhas. | Banco de Dados e Backend (infraestrutura) |
| Alta | RNF11 | Criptografia de Comunicação | Toda a comunicação entre o frontend, backend e hardware deve ser criptografada utilizando os protocolos HTTPS e MQTTS. | Frontend, Backend e Hardware (segurança da comunicação) |
| Alta | RNF12 | Armazenamento de Senhas | As senhas dos usuários devem ser armazenadas no banco de dados utilizando um algoritmo de hashing forte e com salt (ex: bcrypt). | Backend e Banco de Dados (segurança de dados) |
| Alta | RNF13 | Isolamento de Dados de Usuários | Um usuário só pode acessar e visualizar os dados dos veículos associados à sua própria conta. | Frontend, Backend e Banco de Dados (lógica de autorização e acesso) |
| Média | RNF14 | Documentação do Código | O código-fonte do firmware, backend e frontend deve ser devidamente comentado nos pontos críticos para facilitar o entendimento e futuras manutenções. | Frontend, Backend e Hardware (desenvolvimento e manutenção do código) |
| Média | RNF15 | Arquitetura Modular | A arquitetura do software deve ser modular, separando as responsabilidades para facilitar modificações e a adição de novas funcionalidades. | Frontend, Backend e Hardware (arquitetura do sistema) |
| Baixa | RNF16 | Controle de Versão | Todo o código-fonte do projeto deverá ser versionado utilizando Git, com um repositório centralizado (ex: GitHub). | Frontend, Backend e Hardware (desenvolvimento e colaboração) |
| Média | RNF17 | Compatibilidade de Navegadores | A aplicação web (PWA) deve ser compatível com as duas últimas versões estáveis dos principais navegadores (Google Chrome, Mozilla Firefox, Safari, Edge). | Frontend (desenvolvimento e testes) |
| Média | RNF18 | Portabilidade do Backend | A aplicação backend deve ser conteinerizada (utilizando Docker) para garantir a portabilidade entre diferentes ambientes de desenvolvimento e produção. | Backend (infraestrutura e arquitetura) |

**3.1. Requisitos de Desempenho**

* **RNF-01:** A atualização da posição do veículo no mapa (via WebSocket) deve ocorrer em menos de 2 segundos após o recebimento do dado pelo backend.
* **RNF-02:** O tempo de carregamento da tela do dashboard principal não deve exceder 3 segundos em uma conexão de internet banda larga padrão.
* **RNF-03:** A API do backend deve ter um tempo de resposta médio inferior a 500ms para requisições padrão.

**3.2. Requisitos de Usabilidade**

* **RNF-04:** A interface web deve ser desenvolvida como um Progressive Web App (PWA), garantindo uma experiência otimizada para dispositivos móveis, capacidade de ser "instalada" na tela inicial e de receber notificações push.
* **RNF-05:** As ações principais (visualizar mapa, armar/desarmar alarme) devem ser acessíveis a partir da tela principal com o mínimo de cliques.
* **RNF-06:** O sistema deve fornecer feedback visual claro e imediato para as ações do usuário (ex: mensagens de sucesso, erro, indicadores de carregamento).
* **RNF-07:** A interface deve ser intuitiva, minimizando a necessidade de treinamento para um novo usuário.

**3.3. Requisitos de Confiabilidade**

* **RNF-08:** O firmware do dispositivo deve ser capaz de se reconectar automaticamente à rede Wi-Fi/GSM caso a conexão seja perdida.
* **RNF-09:** O backend do sistema deve ser projetado para ter uma disponibilidade (uptime) de 99.5% ou superior.
* **RNF-10:** Devem ser realizados backups automáticos e periódicos do banco de dados para garantir a recuperação em caso de falhas.

**3.4. Requisitos de Segurança**

* **RNF-11:** Toda a comunicação entre o frontend, backend e hardware deve ser criptografada utilizando os protocolos HTTPS e MQTTS.
* **RNF-12:** As senhas dos usuários devem ser armazenadas no banco de dados utilizando um algoritmo de hashing forte e com salt (ex: bcrypt).
* **RNF-13:** Um usuário só pode acessar e visualizar os dados dos veículos associados à sua própria conta.

**3.5. Requisitos de Manutenibilidade**

* **RNF-14:** O código-fonte do firmware, backend e frontend deve ser devidamente comentado nos pontos críticos para facilitar o entendimento e futuras manutenções.
* **RNF-15:** A arquitetura do software deve ser modular, separando as responsabilidades para facilitar modificações e a adição de novas funcionalidades.
* **RNF-16:** Todo o código-fonte do projeto deverá ser versionado utilizando Git, com um repositório centralizado (ex: GitHub).

**3.6. Requisitos de Portabilidade**

* **RNF-17:** A aplicação web (PWA) deve ser compatível com as duas últimas versões estáveis dos principais navegadores (Google Chrome, Mozilla Firefox, Safari, Edge).
* **RNF-18:** A aplicação backend deve ser conteinerizada (utilizando Docker) para garantir a portabilidade entre diferentes ambientes de desenvolvimento e produção.

1. ***Assinaturas***

Os abaixo assinados estão de acordo com o conteúdo deste documento.

|  |  |
| --- | --- |
| Data: \_\_\_/\_\_\_/\_\_\_\_\_  *<Nome do responsável>*  Gestor e Patrocinador do Projeto  Cliente | Data: \_\_\_/\_\_\_/\_\_\_\_\_  *<Nome do responsável>*  Representante dos Usuários  Cliente |
| Data: \_\_\_/\_\_\_/\_\_\_\_\_  *<Nome do responsável>*  Gerente de Projetos  empresa | Data: \_\_\_/\_\_\_/\_\_\_\_\_  *<Nome do responsável>*  Analista Responsável  empresa |