

# Software Requirements Specification (SRS)

Projeto: BugBusters Gas Control App (Backend – Node + TypeScript)

## 1. Introdução

### 1.1 Propósito

Este documento tem como propósito registrar e detalhar todos os requisitos funcionais, não funcionais e regras de negócio do sistema **BugBusters Gas Control App**, garantindo clareza e alinhamento entre equipe, cliente e orientadora.

### 1.2 Escopo

O sistema consiste em uma **API RESTful** desenvolvida em **Node.js + TypeScript**, com banco **PostgreSQL + Prisma ORM**, responsável por:

- Autenticação via telefone
- Cadastro de clientes e fornecedores
- Solicitação de GLP
- Monitoramento de consumo via sensor simulado
- Detecção de vazamento e uso excessivo
- Notificações via WhatsApp API
- Geração de históricos e PDF
- Gestão de fornecedores (horário, endereço, pagamento)

### 1.3 Definições

- **GLP** – Gás Liquefeito de Petróleo
- **API** – Interface de comunicação entre sistemas
- **Prisma** – ORM utilizado para acesso ao banco
- **Sensor Simulado** – Script que irá envia dados periódicos para simular um sensor

### 1.4 Referências

- Documentação Node.js
- Documentação TypeScript
- Documentação Prisma

## **1.5 Visão Geral do Documento**

Este documento apresenta visão geral, requisitos funcionais, não funcionais, regras, casos de uso e critérios de aceitação.

## **2. Visão Geral do Sistema**

### **2.1 Descrição do Problema**

Usuários dependem do GLP para atividades domésticas. Não existe uma ferramenta integrada que combine:

- Monitoramento inteligente do consumo
- Notificações automáticas
- Solicitação rápida de gás
- Informações sobre fornecedores

### **2.2 Objetivos do Sistema**

- Centralizar gestão e solicitação de GLP
- Aumentar segurança com alertas de vazamento
- Automatizar notificações importantes
- Facilitar escolha de fornecedores

### **2.3 Usuários Envoltos**

- **Cliente**
- **Fornecedor**
- **Sistema de Sensor** (simulado)
- **WhatsApp API**

### **2.4 Restrições**

- O projeto é apenas back-end
- Sensor é simulado
- Equipe iniciante
- Tempo limitado

## 2.5 Premissas

- O app mobile integrará com a API
- Cada usuário tem um único botijão sendo monitorado

## 3. Requisitos Funcionais (RF)

**RF-01** – O sistema deve permitir o cadastro de usuário cliente.

**RF-02** – O sistema deve permitir o cadastro de fornecedor.

**RF-03** – O sistema deve autenticar usuários via telefone.

**RF-04** – O cliente deve poder solicitar GLP.

**RF-05** – O sistema deve permitir escolher entre entrega e retirada durante a compra.

- RF-05.1 – O cliente deve escolher: entrega no endereço cadastrado **ou** retirada no fornecedor.
- RF-05.2 – O sistema deve registrar o método selecionado e disponibilizar essa informação ao fornecedor.

**RF-06** – O sistema deve listar fornecedores ativos e inativos.

**RF-07** – O sistema deve aplicar filtros na lista de fornecedores.

- RF-07.1 – Distância
- RF-07.2 – Preço (quando disponível)
- RF-07.3 – Tempo médio de entrega
- RF-07.4 – Popularidade/avaliações
- RF-07.5 – Status aberto/fechado

**RF-08** – O sistema deve registrar leituras do sensor simulado.

**RF-09** – O sistema deve detectar consumo excessivo com base nas leituras do sensor.

**RF-10** – O sistema deve detectar possível vazamento de gás.

**RF-11** – O sistema deve enviar notificações via WhatsApp.

**RF-12** – O usuário deve registrar manualmente a troca de botijão.

- RF-12.1 – O usuário pode clicar em “Troquei o gás agora”.
- RF-12.2 – O sistema deve reiniciar o nível do botijão para 100%.

- RF-12.3 – O sistema deve registrar o timestamp da troca no histórico.
- RF-12.4 – Deve funcionar mesmo quando o sensor estiver offline.

**RF-13** – O sistema deve gerar histórico simples de consumo.

- RF-13.1 – Registrar diariamente peso atual e consumo diário.
- RF-13.2 – Apresentar em gráfico simples.
- RF-13.3 – Permitir visualização semanal ou dos últimos X dias.

**RF-14** – O sistema deve gerar histórico completo e permitir exportação em PDF.

- RF-14.1 – Exibir histórico completo mês a mês.
- RF-14.2 – Incluir no PDF: consumo diário, alertas, trocas registradas, pedidos realizados.
- RF-14.3 – O PDF deve ser disponibilizado para download no app.

**RF-15** – O fornecedor deve cadastrar informações operacionais:

- RF-15.1 – Formas de pagamento
- RF-15.2 – Horários de funcionamento
- RF-15.3 – Dias de funcionamento
- RF-15.4 – Endereço

## 4. Requisitos Não Funcionais (RNF)

**RNF-01** – A API deve responder em até 1s em condições normais.

**RNF-02** – O banco de dados deve ser PostgreSQL.

**RNF-03** – A comunicação deve acontecer exclusivamente via HTTPS.

**RNF-04** – O sistema deve ser escalável para alta carga de sensores.

**RNF-05** – Dados sensíveis devem ficar em variáveis de ambiente.

**RNF-06** – Disponibilidade mínima de 99%.

**RNF-07** – O sistema deve seguir arquitetura MVC + Services.

**RNF-08** – O sistema deve utilizar tokens JWT para autenticação.

**RNF-09** – Dados sensíveis como endereço e telefone devem ser criptografados.

**RNF-10** – Logs de acesso devem ser registrados para auditoria.

**RNF-11** – O sistema deve suportar múltiplos endereços por usuário.

**RNF-12** – O sistema deve armazenar coordenadas aproximadas para cálculos de distância.

## 5. Regras de Negócio (RN)

**RN-01** – Endereço é obrigatório para clientes.

**RN-02** – Fornecedor deve cadastrar horários e formas de pagamento.

**RN-03** – Sensor envia leituras em intervalos regulares.

**RN-04** – Notificações são enviadas sempre que consumo exceder limites predefinidos.

**RN-05** – O cliente só pode solicitar GLP se houver fornecedor ativo na região.

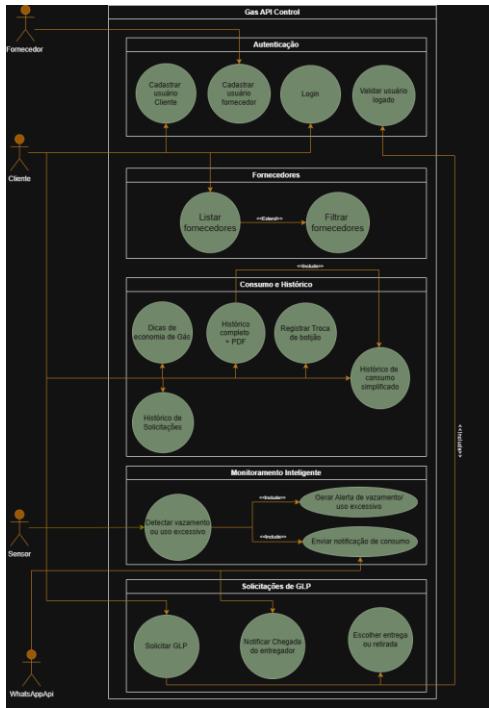
**RN-06** – A troca de botijão reinicia o nível de consumo para 100%.

## 6. Resumo dos Casos de Uso

ID	Nome do Caso de Uso	Atores	Descrição Resumida	Relações
UC01	Cadastrar Usuário Cliente	Cliente	Cadastra cliente com endereço obrigatório.	—
UC02	Cadastrar Fornecedor	Fornecedor	Cadastra fornecedor com endereço, horários e formas de pagamento.	—
UC03	Login	Cliente / Fornecedor	Autentica usuário via telefone.	—
UC04	Solicitar GLP	Cliente	Realiza solicitação de gás GLP.	<<include>> Validar Login
UC05	Escolher Entrega ou Retirada	Cliente	Escolhe modalidade de entrega.	—
UC06	Listar Fornecedores	Cliente	Lista fornecedores próximos.	<<extend>> Filtrar Fornecedores
UC07	Filtrar Fornecedores	Cliente	Aplica filtros de distância e status.	—

UC08	Registrar Leitura do Sensor	Sensor (simulado)	Envia nível de gás periodicamente.	—
UC09	Detectar Consumo Excessivo	Sistema	Identifica uso acima do normal.	<<include>> Enviar Notificação
UC10	Detectar Vazamento	Sistema	Detecta queda súbita no nível de gás.	<<include>> Enviar Alerta
UC11	Enviar Notificação de Consumo	WhatsApp API	Envia mensagens automáticas.	—
UC12	Enviar Alerta de Vazamento	WhatsApp API	Alerta urgente via WhatsApp.	—
UC13	Registrar Troca de Botijão	Cliente	Cliente registra troca manualmente.	—
UC14	Histórico de Solicitações	Cliente	Mostra compras anteriores de GLP.	—
UC15	Histórico de Consumo Simplificado	Cliente	Mostra percentual aproximado restante.	—
UC16	Histórico Completo + PDF	Cliente	Gera relatório detalhado mensal.	<<include>> Histórico Simplificado
UC17	Dicas de Economia	Cliente	Exibe lista de dicas.	—

Figura 1 — Diagrama de Casos de Uso do Sistema



## 7. Requisitos Futuros (Não no MVP)

- Dashboard analítico avançado
- Tela Inicial com manômetro digital
- API para integração com sensores reais
- Recomendação de fornecedores mais rápidos
- Pagamento via app.
- Estimativa de entrega após compra.

## 8. Critérios de Aceitação

- CA-01** – O usuário deve conseguir solicitar GLP sem erros.
- CA-02** – O sistema deve detectar consumo anormal em até 1 minuto.
- CA-03** – Notificações devem ser entregues ao WhatsApp API.
- CA-04** – Histórico deve gerar PDF válido.
- CA-05** – O sensor simulado deve gerar dados consistentes.