

---

**Estacione+**

---

**Estacione+**

**Especificação de Requisitos de Software  
Para Estacione+ Reconhecimento  
Inteligente de Placas**

**Versão 2.0**

Estacione+	Versão: 1.0
Especificação de Requisitos de Software	Data: 07/08/2025
ERS.001	

## Histórico da Revisão

Data	Versão	Descrição	Autor
07/08/2025	1.0	Início do documento.	Eduardo Kauan e Vinicius Pasqualini
05/10/2025	1.1	Finalização parcial do documento.	Eduardo Kauan e Vinicius Pasqualini
13/11/2025	2.0	Atualização completa com base no código final. Adição dos módulos de Preço, Vagas e novo Status do Pátio.	Eduardo Kauan e Vinicius Pasqualini

Estacione+	Versão: 1.0
Especificação de Requisitos de Software	Data: 07/08/2025
ERS.001	

# Índice

<b>1. Introdução</b>	<b>5</b>
<b>1.1 Objetivo</b>	<b>5</b>
<b>1.2 Escopo</b>	<b>5</b>
<b>1.3 Definições, Acrônimos e Abreviações</b>	<b>5</b>
<b>1.4 Referências</b>	<b>5</b>
<b>2. Descrição Geral</b>	<b>6</b>
2.1 Perspectiva do Produto	6
2.2 Funções do Produto	6
2.3 Características do Usuário	6
2.4 Premissas e Dependências	6
2.5 Subconjuntos de Requisitos	6
<b>3. Requisitos Específicos</b>	<b>7</b>
<b>3.1 Funcionalidade</b>	<b>7</b>
<b>3.1.1 RF001 - Cadastro de veículos</b>	<b>7</b>
<b>3.2 Utilidade</b>	<b>7</b>
<b>3.2.1 U001 – Facilidade de aprendizado</b>	<b>7</b>
<b>3.2.2 U002 – Interface padronizada e intuitiva</b>	<b>7</b>
<b>3.2.3 U003 – Tempo de execução das operações</b>	<b>7</b>
<b>3.2.4 U004 – Mensagens claras e feedback imediato</b>	<b>8</b>
<b>3.3 Confiabilidade</b>	<b>8</b>
<b>3.3.1 C001 – Disponibilidade</b>	<b>8</b>
<b>3.3.2 C002 – MTBF (Tempo médio entre falhas)</b>	<b>8</b>
<b>3.3.3 C003 – MTTR (Tempo médio para reparar)</b>	<b>8</b>
<b>3.3.4 C004 – Exatidão dos registros</b>	<b>8</b>
<b>3.3.5 C005 – Taxa máxima de erros críticos</b>	<b>8</b>
<b>3.3.6 C006 – Operação em modo degradado</b>	<b>8</b>
<b>3.4 Desempenho</b>	<b>8</b>
<b>3.4.1 D001 – Tempo de resposta para reconhecimento de placas</b>	<b>8</b>
<b>3.4.2 D002 – Capacidade de processamento</b>	<b>8</b>
<b>3.4.3 D003 – Modos de degradação</b>	<b>8</b>
<b>Caso a conexão com a API Plate Recognizer esteja indisponível, o sistema deve permitir que o operador utilize a funcionalidade de registro manual para continuar operando.</b>	<b>8</b>
<b>3.5 Suportabilidade</b>	<b>8</b>
<b>3.5.1 S001 – Padrões de codificação</b>	<b>8</b>
<b>3.5.2 S002 – Documentação do código</b>	<b>8</b>
<b>3.5.3 S003 – Modularidade</b>	<b>9</b>
<b>3.5.4 S004 – Controle de versão</b>	<b>9</b>

Estacione+	Versão: 1.0
Especificação de Requisitos de Software	Data: 07/08/2025
ERS.001	

<b>3.5.5 S005 – Acesso para manutenção</b>	<b>9</b>
<b>O sistema deverá permitir acesso restrito para atualizações e manutenção, protegendo dados e funcionalidades sensíveis.</b>	<b>9</b>
<b>3.6 Restrições</b>	<b>9</b>
<b>3.6.1 RD001 – Linguagem de Programação</b>	<b>9</b>
<b>3.6.2 RD002 – Framework de Interface Gráfica</b>	<b>9</b>
<b>3.6.3 RD003 – Integração com API de Reconhecimento</b>	<b>9</b>
<b>3.6.4 RD004 – Banco de Dados</b>	<b>9</b>
<b>3.6.5 RD005 – Plataforma de Execução</b>	<b>9</b>
<b>3.6.6 RD006 – Hardware para Captura de Imagens</b>	<b>9</b>
<b>3.6.7 RD007 - Fonte de Dados de Veículos</b>	<b>9</b>
A API de back-end deve utilizar o serviço web placafipe.com como fonte para obter os dados de modelo, cor e ano do veículo. Nenhuma outra fonte de dados para este fim será utilizada.	9
<b>3.7 Documentação do Usuário On-line e Requisitos do Sistema de Ajuda</b>	<b>9</b>
<b>3.8 Componentes Comprados</b>	<b>10</b>
<b>3.9 Interfaces</b>	<b>10</b>
<b>3.9.1 Interfaces com o Usuário</b>	<b>10</b>
<b>3.9.2 Interfaces de Software</b>	<b>10</b>

Estacione+	Versão: 1.0
Especificação de Requisitos de Software	Data: 07/08/2025
ERS.001	

# Especificação de Requisitos de Software

## 1. Introdução

### 1.1 Objetivo

O objetivo deste documento é especificar, de forma clara e detalhada, os requisitos do sistema Estacione+ – Reconhecimento Inteligente de Placas, um software para gerenciamento de estacionamentos com reconhecimento automático de placas de veículos.

Este documento serve como referência para todos os envolvidos no projeto, incluindo desenvolvedores, testadores, orientadores acadêmicos e demais partes interessadas, garantindo que todos possuam uma compreensão comum das funcionalidades e restrições do sistema.

### 1.2 Escopo

Estacione+ é composta por:

- Uma aplicação desktop, desenvolvida em Java com interface gráfica em Java Swing, responsável pela interação com o operador.
- Uma API de back-end, desenvolvida com o framework Spring Boot e Selenium, responsável por toda a lógica de negócio, regras e persistência de dados em um banco de dados PostgreSQL.

A aplicação cliente se integra à API Plate Recognizer para a leitura automática de placas de veículos via câmera. Uma vez que a placa é obtida, o cliente a envia para a API de back-end, que por sua vez enriquece os dados do veículo através de consulta ao serviço web Placa Fipe.

O sistema permitirá:

- Captura da placa no momento da entrada do veículo (via câmera ou manual).
- Identificação e registro de dados do veículo.
- Configuração do número total de vagas do pátio.
- Configuração da regra de cálculo de preço (valor por fração de tempo).
- Visualização gráfica do pátio (vagas livres e ocupadas).
- Cálculo em tempo real do tempo de permanência e valor atual para veículos estacionados.
- Registro de saída de veículos com cálculo de valor final.
- Geração de relatórios gerenciais (Movimentação e Financeiro).

O público-alvo são estacionamentos de pequeno porte, que buscam um sistema de fácil uso, baixo custo e integração com câmeras já disponíveis.

### 1.3 Definições, Acrônimos e Abreviações

- PlateRecognizer – Serviço de reconhecimento de placas de veículos por meio de inteligência artificial.
- Placa Fipe: Serviço web externo (não-oficial) utilizado pelo back-end para consultar e obter informações de modelo, cor e ano de um veículo a partir de sua placa.
- ERS: Especificação de Requisitos de Software.
- UI: Interface de Usuário (User Interface).

### 1.4 Referências

- Documentação oficial da API PlateRecognizer – <https://platerecognizer.com>
- Documentação Java Swing – <https://docs.oracle.com/javase/tutorial/uiswing/>
- Documentação Spring Boot – <https://spring.io/projects/spring-boot>

Estacione+	Versão: 1.0
Especificação de Requisitos de Software	Data: 07/08/2025
ERS.001	

## 2. Descrição Geral

### 2.1 Perspectiva do Produto

O Estacione+ é um sistema que opera em uma arquitetura Cliente-Servidor. O componente do cliente (front-end) é uma aplicação desktop instalada no computador do operador, com interface gráfica desenvolvida em Java Swing. O componente do servidor (back-end) é uma API REST desenvolvida em Java com Spring Boot, que centraliza as regras de negócio e o acesso ao banco de dados.

A comunicação entre o cliente e o servidor ocorre via protocolo HTTP. O cliente depende de conexão com a API Plate Recognizer para a função de reconhecimento de imagem, enquanto o back-end depende de conexão com a internet para consultar o serviço Placa Fipe.

### 2.2 Funções do Produto

O sistema oferecerá as seguintes funcionalidades principais:

- Cadastro de veículos e informações do proprietário.
- Reconhecimento de placas por câmera (realizado na aplicação cliente via Plate Recognizer) e enriquecimento de dados do veículo (realizado no back-end via PlacaFipe).
- Registro de entrada e saída de veículos com data e hora.
- Cálculo de valor a pagar proporcional ao tempo de permanência.
- Configuração de regras de preço (valor e minutos) e do total de vagas.
- Dashboard visual do pátio (Status do Pátio) com vagas ocupadas e livres.
- Cálculo em tempo real do tempo estacionado e valor atual para cada veículo no pátio.
- Dashboard com métricas em tempo real (total de veículos, vagas disponíveis).
- Geração de relatórios de Movimentação e Financeiro.

### 2.3 Características do Usuário

- Operadores: são os únicos usuários do sistema. Eles são responsáveis por monitorar entradas e saídas de veículos, cadastrar veículos e interagir com todas as funcionalidades disponíveis.

O sistema será projetado para usuários com conhecimento básico de informática, exigindo apenas noções simples de uso de teclado e mouse.

### 2.4 Premissas e Dependências

- A API Plate Recognizer e o serviço web placafipe.com devem estar operacionais e acessíveis pela internet.
- Disponibilidade de conexão com a internet para reconhecimento de placas.
- Banco de dados PostgreSQL instalado e configurado no ambiente local.

### 2.5 Subconjuntos de Requisitos

- Controle de entradas e saídas de veículos: o operador registra quando um veículo entra ou sai do estacionamento, garantindo a atualização correta do número de vagas disponíveis.
- Gerenciamento de vagas: o sistema mantém o controle em tempo real das vagas ocupadas e livres através de um painel visual, permitindo que o operador identifique rapidamente a disponibilidade e gerencie as saídas.
- Configuração do Sistema: o operador pode definir parâmetros de negócio, como o total de vagas e a política de precificação.

Estacione+	Versão: 1.0
Especificação de Requisitos de Software	Data: 07/08/2025
ERS.001	

### 3. Requisitos Específicos

#### 3.1 Funcionalidade

##### 3.1.1 RF001 - Cadastro de veículos

O sistema deve permitir o cadastro de veículos, armazenando placa, modelo, cor, ano e dados do proprietário (nome e telefone).

##### 3.1.2 RF002 - Reconhecimento automático de placas

A aplicação cliente deve, através da tela de registro de entrada, capturar a imagem de um veículo por meio de uma câmera. Esta imagem deve ser enviada para a API externa Plate Recognizer. O sistema deve então processar a resposta para extrair a placa do veículo em formato de texto. A placa extraída será então utilizada para registrar a entrada do veículo.

##### 3.1.3 RF003 – Registro de entrada de veículo

O sistema deve registrar a data e hora de entrada quando a placa for reconhecida (RF002) ou inserida manualmente (RF009).

##### 3.1.4 RF004 – Registro de saída de veículo

O sistema deve permitir o registro da saída de um veículo. Ao fazer isso, o sistema deve calcular o valor total a ser pago (com base no RF014) e registrar a data e hora de saída e o valor final no banco de dados.

##### 3.1.5 RF005 – Status do Pátio (Visual)

O sistema deve exibir um painel visual ("Status do Pátio") com uma grade que representa todas as vagas (configuradas no RF013). As vagas ocupadas devem exibir a placa do veículo e ser visualmente distintas das vagas livres, permitindo ao operador clicar em uma vaga ocupada para registrar a saída.

##### 3.1.6 RF006 – Geração de relatórios

O sistema deve permitir a geração de relatórios com base em um período (data de início e fim) selecionado pelo operador. Os relatórios devem ser divididos em:

- a) Relatório de Movimentação: Detalhando todas as entradas e saídas no período, com dados completos do veículo e cliente.
- b) Relatório Financeiro: Listando apenas os registros finalizados (com saída) no período, exibindo o valor pago por cada e um somatório total do valor arrecadado.

Ambos os relatórios devem ser gerados em formato PDF e visualizáveis dentro da aplicação.

##### 3.1.7 RF007 – Pesquisa de registros

O sistema deve permitir buscar registros de veículos por placa ou por período (histórico completo).

##### 3.1.8 RF008 - Dashboard de Métricas

O sistema deve apresentar uma tela de Métricas com cartões (cards) exibindo indicadores visuais, incluindo, a contagem em tempo real de veículos presentes no estacionamento e o total de vagas disponíveis.

##### 3.1.9 RF009 - Registro de Entrada Manual

Além do registro por câmera, o sistema deve fornecer uma tela para o registro de entrada totalmente manual, onde o operador pode digitar todos os dados do veículo (Placa, Nome do Cliente, Telefone).

##### 3.1.10 RF010 - Visualização de Ficha Detalhada

O sistema deve permitir que o operador visualize uma ficha com todos os dados detalhados de um registro de estacionamento (placa, cliente, modelo, cor, ano, horários de entrada e saída).

##### 3.1.11 RF011 - Reativação de Veículo

Ao visualizar a ficha de um veículo com registro já finalizado (saída registrada), o sistema deve oferecer ao operador a opção de registrar uma nova entrada para este mesmo veículo de forma rápida, reutilizando seus dados cadastrais.

##### 3.1.12 RF012 - Configuração de Preço

O sistema deve permitir ao operador, através de configuração, definir a regra de precificação. A regra consiste em um valor e uma fração de tempo correspondente em minutos.

Estacione+	Versão: 1.0
Especificação de Requisitos de Software	Data: 07/08/2025
ERS.001	

### 3.1.13 RF013 - Configuração de Vagas

O sistema deve permitir ao operador, através de configuração, definir o número total de vagas do estacionamento. Este número será usado como base para o "Status do Pátio" (RF005).

### 3.1.14 RF014 - Cálculo de Valor Proporcional

No momento da saída (RF004), o sistema deve calcular o valor total a ser pago de forma proporcional ao tempo de permanência (em minutos), com base na regra de preço configurada (RF012).

### 3.1.15 RF015 - Cálculo de Status em Tempo Real

O painel "Status do Pátio" (RF005) deve exibir, para cada veículo ocupante, o tempo de permanência atualizado e o valor a ser pago até o momento (calculado proporcionalmente, conforme RF014).

### 3.1.16 RF016 - Pré-visualização de Saída

Ao clicar em uma vaga ocupada (RF005), antes de confirmar a saída, o sistema deve exibir um diálogo de confirmação contendo os detalhes da estadia (placa, cliente, tempo, valor atual), solicitando ao operador a confirmação para registrar a saída.

## 3.2 Utilidade

### 3.2.1 U001 – Facilidade de aprendizado

O sistema deve permitir que um operador com conhecimento básico de informática aprenda todas as funcionalidades em no máximo 1 hora de uso do sistema.

### 3.2.2 U002 – Interface padronizada e intuitiva

A interface gráfica deve seguir um padrão de design consistente, com menus, botões e campos posicionados de forma uniforme em todas as telas, seguindo convenções comuns de aplicações desktop (menus na parte esquerda, botões de ação na superior).

### 3.2.3 U003 – Tempo de execução das operações

As operações comuns, como consulta de vagas, registro de entrada/saída e cadastro de veículo, devem ser concluídas em menos de 20 segundos após a captura ou digitação da placa.

### 3.2.4 U004 – Mensagens claras e feedback imediato

O sistema deve exibir mensagens objetivas para confirmar o sucesso ou indicar erro em qualquer ação executada, garantindo que o operador receba feedback imediato.

## 3.3 Confiabilidade

### 3.3.1 C001 – Disponibilidade

O sistema deve estar disponível para uso 90% do tempo durante o horário de funcionamento do estacionamento, considerando a manutenção preventiva fora desse período.

### 3.3.2 C002 – MTBF (Tempo médio entre falhas)

O sistema deve operar por no mínimo 500 horas consecutivas sem falhas críticas que impeçam seu uso.

### 3.3.3 C003 – MTTR (Tempo médio para reparar)

Em caso de falha, o tempo máximo permitido para que o sistema seja restaurado ao funcionamento é de 1 horas.

### 3.3.4 C004 – Exatidão dos registros

O sistema deve registrar entradas, saídas e dados de veículos com 100% de precisão nos campos de hora de entrada e hora de saída.

### 3.3.5 C005 – Taxa máxima de erros críticos

A taxa máxima permitida de erros críticos (falhas que impeçam o registro de entrada/saída ou

Estacione+	Versão: 1.0
Especificação de Requisitos de Software	Data: 07/08/2025
ERS.001	

causem perda de dados) não deve exceder 1 ocorrência a cada 1.000 operações.

### 3.3.6 C006 – Operação em modo degradado

O sistema oferece nativamente um fluxo de registro de entrada manual que não depende da câmera ou da API Plate Recognizer, servindo como alternativa primária ao fluxo automatizado.

## 3.4 Desempenho

### 3.4.1 D001 – Tempo de resposta para reconhecimento de placas

O sistema Estacione+ deve processar o reconhecimento de placas e liberar ou negar o acesso em no máximo 30 segundos após a captura da imagem pela câmera.

### 3.4.2 D002 – Capacidade de processamento

O sistema deve suportar o processamento de até 2 veículos por minuto, mantendo o tempo de resposta especificado.

### 3.4.3 D003 – Modos de degradação

Caso a conexão com a API Plate Recognizer esteja indisponível, o sistema deve permitir que o operador utilize a funcionalidade de registro manual para continuar operando.

## 3.5 Suportabilidade

### 3.5.1 S001 – Padrões de codificação

O código-fonte do sistema deverá seguir as convenções de nomenclatura e estilo da linguagem Java, garantindo legibilidade e facilidade de manutenção.

### 3.5.2 S002 – Modularidade

O sistema deverá ser desenvolvido com arquitetura modular, separando claramente as camadas de apresentação, lógica de negócio e acesso a dados, para facilitar manutenção e evolução.

### 3.5.3 S003 – Controle de versão

O projeto deverá utilizar sistema de controle de versão, permitindo histórico de alterações e colaboração organizada.

### 3.5.4 S004 – Acesso para manutenção

O sistema deverá permitir acesso restrito para atualizações e manutenção, protegendo dados e funcionalidades sensíveis.

## 3.6 Restrições

### 3.6.1 RD001 – Linguagem de Programação

O sistema deverá ser desenvolvido utilizando a linguagem Java, garantindo compatibilidade com as bibliotecas e frameworks escolhidos.

### 3.6.2 RD002 – Framework de Interface Gráfica

A interface gráfica da aplicação cliente deverá ser implementada exclusivamente com Java Swing.

### 3.6.3 RD003 – Integração com API de Reconhecimento

A aplicação cliente deve obrigatoriamente utilizar a API Plate Recognizer para o reconhecimento de placas a partir de imagens.

### 3.6.4 RD004 – Banco de Dados

O sistema utilizará o banco de dados PostgreSQL para armazenamento local das informações, sendo obrigatório seguir a estrutura definida no projeto.

Estacione+	Versão: 1.0
Especificação de Requisitos de Software	Data: 07/08/2025
ERS.001	

### 3.6.5 RD005 – Plataforma de Execução

O sistema será executado exclusivamente em ambiente desktop Windows, não sendo prevista versão para outros sistemas operacionais nesta etapa.

### 3.6.6 RD006 – Hardware para Captura de Imagens

A captura das imagens para reconhecimento será feita via câmera de smartphone configurada como webcam, não sendo permitida a utilização de câmeras IP ou dispositivos dedicados externos.

### 3.6.7 RD007 - Fonte de Dados de Veículos

A API de back-end deve utilizar o serviço web placafipe.com como fonte para obter os dados de modelo, cor e ano do veículo. Nenhuma outra fonte de dados para este fim será utilizada.

## 3.7 Documentação do Usuário On-line e Requisitos do Sistema de Ajuda

Não será implementado um sistema de help interativo. O suporte será centralizado em um manual digital (PDF) e em eventuais atualizações.

## 3.8 Componentes Comprados

O sistema Estacione+ depende de serviços externos para sua funcionalidade principal. O componente primário é a API Plate Recognizer, um serviço comercial de terceiros usado para o reconhecimento de placas (RF002).

O sistema é projetado para operar com o plano gratuito ('Free Tier') deste serviço, que requer que o operador se cadastre, obtenha uma chave de API (API Key) e a insira na tela de configuração do sistema para ativar a funcionalidade.

## 3.9 Interfaces

### 3.9.1 Interfaces com o Usuário

O sistema terá uma interface cliente desktop, desenvolvida em Java Swing. A navegação será feita por menus e botões, organizada nas seguintes telas principais:

- Tela Principal: Navegação central e acesso aos módulos.
- Tela de Entrada de Veículo (Registrar Veíc.): Com visualização da câmera (RF002) e opção de registro manual (RF009).
- Tela Status do Pátio (Gerenciar): Exibe uma grade visual de vagas ocupadas e livres (RF005), permitindo registrar saídas ao clicar em uma vaga ocupada (RF016).
- Tela de Consulta de Histórico (Consultar): Permite buscar registros por placa (RF007) e visualizar detalhes (RF010).
- Tela de Relatórios e Métricas (Relatórios): Apresenta o dashboard de métricas (RF008) e permite gerar relatórios por período (RF006).
- Tela de Visualização de PDF: Exibe o relatório gerado.
- Diálogo de Configuração de Preço (NOVO): Permite ao operador definir o valor e o tempo para o cálculo (RF012).
- Diálogo de Configuração de Vagas (NOVO): Permite ao operador definir o número total de vagas (RF013).
- Diálogo de Configuração de Chave de Serviço: Permite inserir o token da API Plate Recognizer.

### 3.9.2 Interfaces de Software

A aplicação desktop se comunicará com a API back-end desenvolvida em Spring Boot via requisições HTTP (REST). O back-end, por sua vez, se conectará ao banco de dados PostgreSQL via driver JDBC e fará requisições HTTP para as APIs externas (Plate Recognizer,

Estacione+	Versão: 1.0
Especificação de Requisitos de Software	Data: 07/08/2025
ERS.001	

Placa Fipe no back-end).