

# LocarEasy

1. **Para:** Usuários que desejam locar carros para suprir suas necessidades
2. **Que estão insatisfeitos com:** O modelo atual de locação de automóveis oferecidos
3. **Nosso produto:** LocarEasy é uma startup inovadora que prioriza a flexibilidade do aluguel de carros, oferecendo mobilidade e tranquilidade para nossos clientes
4. **Que fornece:** a locação contabilizada por horas entre o check-in e check-out
5. **Ao contrário dos:** Sistemas de locação veicular disponíveis no atual mercado
6. **Nosso produto:** Oferece a capacidade do cliente alugar veículos de forma a pagar por horas. Além disso, temos parceria com postos de combustível e estacionamentos, dessa forma, é possível coletar e entregar veículos em nossos parceiros, sem a necessidade de locomoção até as unidades da LocarEasy.

Fig. 1 – Documento de Visão

## Requisitos Funcionais

### 1. Registro de usuários:

- os usuários devem se registrar fornecendo nome, endereço de e-mail e senha;
- devem também realizar confirmação de habilitação enviando uma foto de sua Carteira Nacional de Habilitação ou da CNH Digital;
- os dados de registro devem ser armazenados de forma segura no banco de dados.

### 2. Reserva de veículos:

- os usuários devem conseguir visualizar a lista de veículos disponíveis para a locação;
- a aplicação deve permitir que os usuários selecionem um veículo de sua escolha em uma das unidades da LocarEasy ou dos parceiros;
- o sistema deve verificar a disponibilidade do veículo;
- os usuários informam a hora de retirada e a hora de realocação do veículo;
- o sistema deve registrar no banco de dados a reserva.

### **3. Cancelamento de reserva:**

- os usuários devem ser capazes de cancelar uma reserva existente, indicando um motivo preestabelecido ou um outro;
- a aplicação deve permitir o cancelamento dentro de uma prazo determinado antes do horário de retirada do veículo;
- o banco de dados deve ser atualizado para refletir o cancelamento da reserva.

### **4. Pagamento:**

- o sistema deve oferecer opções de pagamento, como Pix, cartão de crédito ou dinheiro;
- os usuários devem fornecer informações necessárias e válidas para o pagamento de forma segura;
- o sistema deve processar o pagamento e registrar a transação no banco de dados.

### **5. Localização dos veículos:**

- a empresa deve integrar um sistema de rastreadores nos veículos em tempo real, bem como o status de atividade do veículo (ativo ou inativo);
- os usuários devem ver em um mapa na aplicação a localização atual dos veículos disponíveis;

### **6. Devolução dos veículos:**

- os usuários devem poder realizar a devolução dos veículos alugados após o término do período de locação;
- a devolução pode ser feita em uma das unidades da LocarEasy, em um dos postos de combustível parceiros ou em um dos estacionamentos parceiros;
- os usuários devem poder relatar quaisquer danos e/ou problemas em relação ao estado do veículo;
- a empresa deve poder fornecer uma inspeção do estado do veículo após a devolução;
- o sistema deve atualizar o status de disponibilidade do veículo;
- o sistema deve enviar uma notificação aos usuários confirmando a devolução bem-sucedida, bem como uma mensagem de agradecimento;
- os usuários devem receber um recibo detalhado com informações relevantes, como data, horário, valor etc.

### **7. Feedback:**

- os usuários devem poder fornecer um feedback sobre a experiência com a locação do veículo;

- a aplicação deve armazenar o feedback no banco de dados e disponibilizar para a avaliação de outros usuários.

## **Requisitos não funcionais**

### **1. Segurança:**

- utilização de criptografia para proteger dados confidenciais dos usuários durante o armazenamento;
- implementação do serviço de segurança CloudFlare, para a prevenção de ataques;
- uso do framework de segurança Flask Security, no desenvolvimento.

### **2. Desempenho:**

- utilização de técnicas de otimização de desempenho, como o cache de dados, para reduzir o tempo de resposta e aumentar a escalabilidade;
- utilização da linguagem Python para a construção do backend, utilizando o framework Flask;
- uso do Sistema Gerenciador de Banco de Dados (SGBD) MySQL no Amazon RDS ou Microsoft Azure;
- utilização de tecnologias para escalabilidade horizontal, como os contêineres do Docker ou orquestração Kubernetes, em caso de aumento de tráfego e demanda.

### **3. Usabilidade:**

- compatibilidade com os sistemas Android e iOS;
- uma interface de usuário responsiva, seguindo as melhores práticas de User Xperience (UX);
- utilização do framework front-end Web React, bem como desenvolvimento multiplataforma nativo, utilizando Flutter;
- realização de testes de usabilidade para garantir a facilidade de uso e atendimento às expectativas do usuário;

### **4. Disponibilidade:**

- implementação da plataforma em um serviço de hospedagem em nuvem confiável e escalável, por meio da Microsoft Azure ou Amazon Web Services (AWS);
- comunicação do front-end e do back-end por meio de API;
- implementação de uma ferramenta de monitoramento de erros, bugs e da saúde da plataforma, como o Sentry.

### **5. Confiabilidade:**

- realização de testes de unitários para garantir a estabilidade e funcionamento íntegro da plataforma;
- utilização de mecanismos de backup periódicos a fim de evitar perda de dados;
- implementação de replicação de servidores que garante a disponibilidade contínua do serviço;

#### **6. Suporte técnico:**

- disponibilização de meios de comunicação com o usuário, por meio de e-mail, WhatsApp, chatbot etc;
- utilização do Help Desk para gerenciar e “trackear” as solicitações de suporte;
- recrutamento e treinamento de uma equipe de suporte ao usuário capacitada e que deve ser 24h.