

Integrantes:

Pedro Ribeiro Ramos - RM 559513

Thiago Cardoso Mendes Machado - RM 560032

Maria Clara Vieira da Silva - RM 560947

Polianna Albuquerque Lima- RM 561078

Alise Souza Galdino da Silva - RM561164

Qual o problema do cliente?

O problema identificado na operação do estacionamento é a dificuldade que os manobristas têm para encontrar vagas livres. O estacionamento possui capacidade para 5.000 veículos com vagas demarcadas, e essa falta de um sistema eficiente para localizar os espaços disponíveis causa transtornos e ineficiência na operação.

Lista da pesquisa e análise dos recursos disponíveis em softwares de informatização para estacionamento**1. Frontend (Interface do Usuário)**

O sistema precisa de uma interface simples e que funcione em múltiplos dispositivos.

- **Frameworks JavaScript:**
 - **React** ou **Vue.js**: Ideais para criar interfaces de usuário dinâmicas e reativas. Permitem construir uma experiência de uso simples e intuitiva para os operadores e gestores.
- **Design Responsivo:**
 - **Bootstrap** ou **Tailwind CSS**: Frameworks CSS que facilitam a criação de layouts que se adaptam a diferentes tamanhos de tela, como desktops, tablets e celulares.
- **Aplicativo Móvel (Opcional):**
 - **React Native** ou **Flutter**: Caso a integração com um aplicativo para clientes seja um requisito, essas ferramentas permitem criar aplicativos para Android e iOS a partir de um único código-base.

2. Backend (Servidor e Lógica de Negócio)

O backend será o cérebro do sistema, responsável por registrar operações, calcular tarifas e gerenciar dados.

- **Linguagem de Programação e Framework:**
 - **Node.js com Express.js**: Uma escolha moderna e eficiente para processar operações de entrada e saída rapidamente e construir APIs para comunicação com o frontend e sistemas externos.
 - **Python com Django**: Um framework robusto que agiliza o desenvolvimento, ideal para gerar relatórios e gerenciar permissões de usuários (operador, gestor).

- **Java com Spring:** Conhecido pela sua estabilidade e segurança, é uma ótima opção para garantir um sistema confiável em horários de pico e proteger os dados dos clientes.
- **API (Interface de Programação de Aplicações):**
 - **RESTful API:** Um padrão de arquitetura para criar a comunicação entre o frontend, o backend e possíveis integrações com sistemas de pagamento.

3. Banco de Dados

A escolha do banco de dados é crucial para garantir um sistema rápido, centralizado e que suporte relatórios.

- **Sistema de Gerenciamento de Banco de Dados (SGBD):**
 - **PostgreSQL ou MySQL:** Bancos de dados relacionais (SQL) que são extremamente confiáveis, rápidos e ideais para armazenar dados estruturados como histórico de veículos, pagamentos e registros de logs.

4. Arquitetura e Infraestrutura (DevOps)

Para atender aos requisitos de estabilidade, segurança e escalabilidade, a infraestrutura é fundamental.

- **Provedor de Nuvem:**
 - **Amazon Web Services (AWS), Google Cloud ou Microsoft Azure:** Oferecem servidores confiáveis, opções de backup diário automático, redes seguras e a capacidade de escalar o sistema conforme a demanda de veículos aumenta.
- **Containerização:**
 - **Docker:** Permite "empacotar" a aplicação e suas dependências, garantindo que ela funcione de maneira consistente em qualquer ambiente e facilitando a manutenção do código.
- **Orquestração de Containers:**
 - **Kubernetes:** Ajuda a automatizar a implantação e o gerenciamento de containers, sendo essencial para um sistema tolerante a falhas e escalável.

5. Integração com Hardware

O sistema precisa se comunicar com equipamentos físicos como sensores, catracas e impressoras.

- **SDKs (Software Development Kits):** Muitos fabricantes de hardware fornecem SDKs próprios (geralmente em linguagens como C++, Java ou C#) para facilitar a integração.
- **Serviços de Mensageria:**
 - **MQTT:** Um protocolo leve, ideal para a comunicação entre o software e os sensores de vagas, garantindo alertas em tempo real.
- **APIs de baixo nível:** Para impressoras de tickets e leitores de cartão, pode ser necessário desenvolver uma camada de software intermediária (um *driver*) em

Python ou **Node.js** que traduza os sinais do hardware em comandos para o sistema principal.

6. Ferramentas de Gestão e Documentação

Para garantir a manutenção e a qualidade do projeto.

- **Controle de Versão:**
 - **Git e GitHub/GitLab:** Essenciais para gerenciar as versões do código-fonte e facilitar o trabalho em equipe.
- **Documentação da API:**
 - **Swagger/OpenAPI:** Ferramentas para documentar a API do sistema de forma clara, o que é crucial para a manutenção e futuras expansões.

Entrevista: Lista de perguntas e respostas

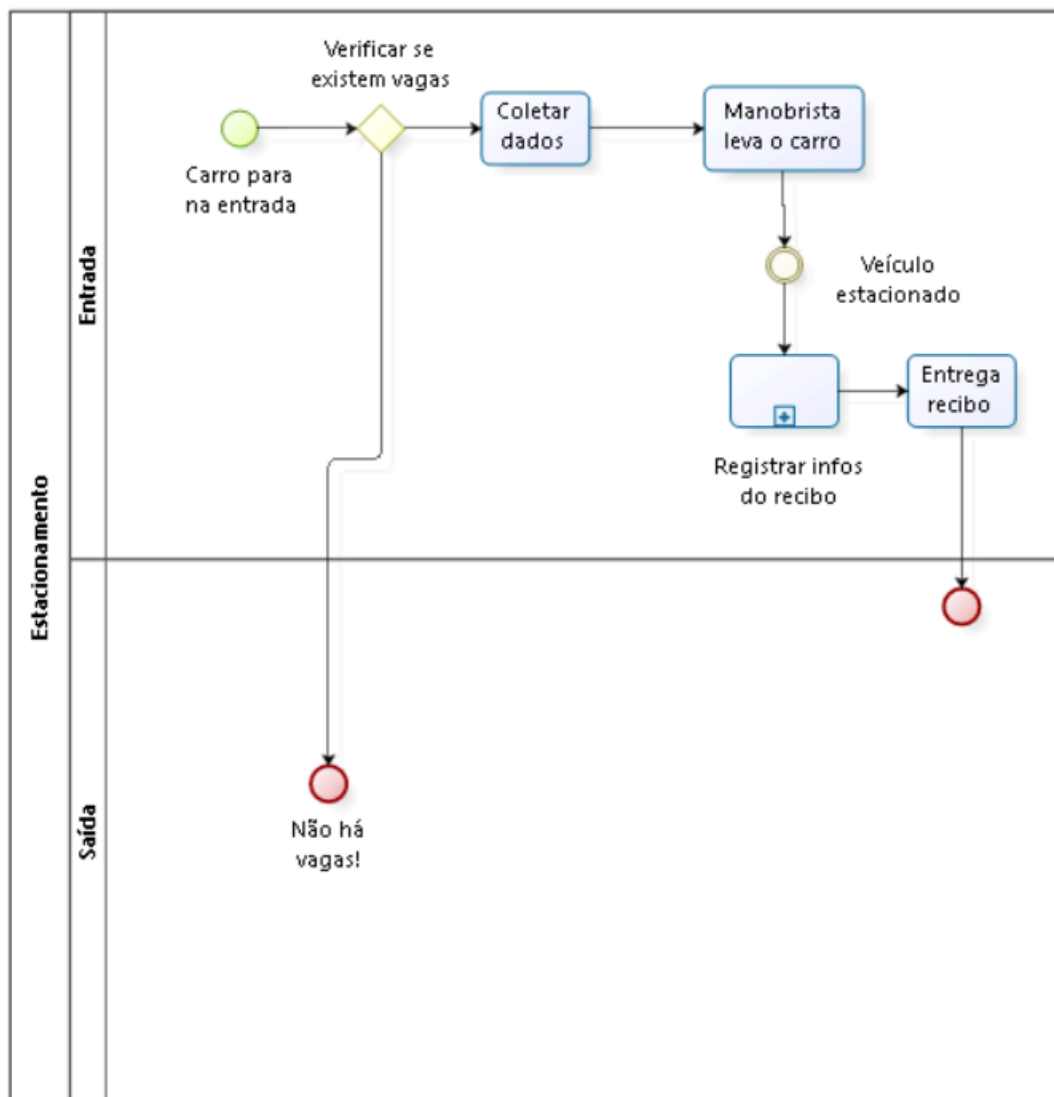
Entrevistador: "Bom dia! Gostaria de entender melhor suas expectativas ao usar um estacionamento. Posso fazer algumas perguntas?"

Consumidor: "Bom dia! Claro."

1. **Pergunta:** Qual é a primeira coisa que você busca ao escolher um estacionamento?
 - **Resposta:** Localidade e se o local é próximo ou não do meu destino.
2. **Pergunta:** Como você costuma realizar o pagamento? Você tem preferência por algum método?
 - **Resposta:** Cartão de débito, crédito ou pix.
3. **Pergunta:** Você se sentiria mais seguro se pudesse acompanhar o status do seu veículo pelo celular, sabendo que ele está estacionado e seguro?
 - **Resposta:** Com certeza. Seria uma tranquilidade a mais, especialmente em locais que não conheço bem.
4. **Pergunta:** O que você acha de sistemas que leem a placa do carro automaticamente na entrada e na saída para calcular o tempo, em vez de usar tickets de papel?
 - **Resposta:** Acho excelente! É muito mais prático e eu não correria o risco de perder o ticket.
5. **Pergunta:** Você estaria interessado em um programa de fidelidade que oferecesse descontos ou horas grátis com base na frequência de uso?
 - **Resposta:** Depende se for uma rede de estacionamento/shopping ou não.
6. **Pergunta:** Seria útil para você poder reservar uma vaga online antes mesmo de chegar ao local?
 - **Resposta:** Sim, principalmente em dias de eventos ou em áreas muito movimentadas. Isso economizaria muito tempo e estresse.
7. **Pergunta:** Na sua opinião, a sinalização de vagas livres e ocupadas dentro do estacionamento (com luzes verdes/vermelhas, por exemplo) facilitaria sua vida?
 - **Resposta:** Demais! É muito ruim ficar rodando sem saber para onde ir. Isso agilizaria muito o processo de encontrar uma vaga.

8. **Pergunta:** Qual a sua maior frustração no processo de saída de um estacionamento?
- **Resposta:** As filas para pagar no caixa.
9. **Pergunta:** Você gostaria de receber o recibo de pagamento de forma digital, por e-mail ou no aplicativo, em vez de impresso?
- **Resposta:** Sim, é mais sustentável e mais fácil de organizar para controle de despesas.
10. **Pergunta:** Que outra funcionalidade ou serviço você acha que um estacionamento moderno deveria oferecer?
- **Resposta:** Um serviço de limpeza externo no carro.

Rotina do estacionamento



Link do GitHub: <https://github.com/azul182/CP-2-Estudo-de-caso-estacionamento>