TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS

JOÃO ANTONIO CARDOSO

FASTOCK

LONDRINA

2023

Lista de Figuras

Lista de Imagens

Sumário

1. INTRODUÇÃO

Algumas empresas, quando iniciam seu projeto, não conseguem prever o futuro e como tudo irá acontecer, por diversos motivos, sejam eles internos ou externos, e conforme os anos se passam o projeto inicial de atuação da empresa já não é mais o mesmo. Esta crescente mutação pode trazer alguns problemas para o negócio, cabe a empresa transformar estes problemas em uma oportunidade.

Foi o que aconteceu com a Fastcar Premium, uma oficina de pintura e funilaria, que já atua a mais de uma década na cidade de Londrina – PR. Com o passar dos anos a empresa conseguiu acumular uma quantidade relativamente grande de peças automotivas, sem nem mesmo te um estoque para tal. O acumulo gigantesco das peças em sua dependência começou a trazer alguns problemas para o negocio como falta de espaço e desorganização do ambiente.

O problema obrigou os sócios a contratarem funcionários com a função de organizar o “estoque”. Aos poucos o estoque foi se estruturando, trazendo organização para as dependências da empresa. A empresa conseguiu construir um inventário gigantesco de peças em sua dependência, agora com um estoque organizado e limpo, é fácil ver a oportunidade que o acumulo de peças trouxe, a ideia dos sócios neste momento é poder comercializar estas peças e aumentar o lucro de suas atividades.

Uma organização eficiente desempenha um papel fundamental no sucesso de qualquer empresa. Para o caso da Fastcar Premium, um sistema de estoque de peças bem estruturado e organizado garante a disponibilidade dos itens necessários no momento certo, reduzindo o tempo de espera dos clientes e aprimorando a qualidade do atendimento.

A criação de um sistema para um estoque de peças simples e dinâmico tem como embasamento uma adaptação de alguns dos princípios do

Lean Production (MAIA et al., 2011). No caso desse projeto, primeiramente, pode-se eliminar atividades desnecessárias para os objetivos finais, reduzindo o tempo de espera dos clientes, como por exemplo, a obtenção de informações sobre peças estar vinculada a algum funcionário especifico ou ser feita de forma manual durante um atendimento. E em segundo, a busca de melhoria contínua, buscando uma forma de gerenciamento que possibilite uma visualização geral do estoque, assim facilitando a concepção de propostas para melhorias constantes.

1.1 OBJETIVOS

O objetivo principal do projeto é desenvolver um sistema de estoque de peças eficiente e otimizado para a empresa Fastcar Premium. O sistema visa controlar de forma precisa e organizada o inventário de estoque, maximizando a eficiência operacional, garantindo também que o trabalho de organização continue sendo aplicado. Com essa solução, a empresa poderá aprimorar sua gestão de estoque e melhorar o atendimento aos clientes.

1.2 PROPOSTA E JUSTIFICATIVA

O sistema proposto para o estoque de peças se trata de um aplicativo com diversas funcionalidades. A primeira é o cadastramento de peças, permitindo que novos produtos sejam inseridos no sistema através inclusão de informações como nome da peça, descrição da peça, estado da peça (nova ou usada), código, tipo de carro, marca do carro, ano do carro, entre outras. Além disso, o sistema oferece a possibilidade de realizar

alterações no cadastro das peças, permitindo a atualização de dados, bem como sua realocação caso a peça mude de posição. A baixa de peças se dará por duas maneiras, a exclusão e a venda, ambas proporcionadas pelo sistema. Por fim, o sistema oferece a funcionalidade de consulta, onde é possível buscar informações sobre uma peça específica, verificando sua disponibilidade e sua locação em estoque.

Com essas propostas, o sistema de estoque de peças por meio do aplicativo proporcionará uma gestão mais eficiente e organizada do estoque. Também será possível atribuir acesso a outras oficinas, para poderem consultar a disponibilidade das peças ampliando assim a game de clientes.

2. FUNDAMENTAÇÃO TEÓRICA

Para Moreira (2012), “os estoques são quantidades de bens físicos que permanecem armazenados de forma improdutiva por algum intervalo de tempo”.

De acordo com Slack (2002), o estoque é caracterizado pela acumulação de materiais armazenados em um sistema de transformação. Essa acumulação envolve qualquer quantidade de material que tenha valor financeiro para uma empresa ou organização, sendo separada para uso futuro em atividades relacionadas à linha de produção.

A gestão eficiente de estoque é essencial para o sucesso de qualquer empresa, independentemente do seu porte ou setor de atuação. Um sistema de gestão de estoque bem desenvolvido e implementado pode ajudar as organizações a otimizar seus processos, reduzir custos, minimizar riscos e melhorar a satisfação dos clientes. Nessa perspectiva, o desenvolvimento de um *software* de gestão de estoque é uma estratégia inteligente para automatizar e aprimorar as operações relacionadas ao controle de estoques.

“A ciência da computação engloba o estudo dos computadores e suas capacidades, bem como suas limitações inerentes. Envolve a compreensão do design e das características dos computadores reais, além das diversas aplicações que podem ser desenvolvidas para resolver problemas” (DE PÁDUA PAULA FILHO, 2003). Uma das áreas mais importantes no vasto mundo da computação, é a de desenvolvimento de *software* e a interação com o usuário para auxilio nos problemas.

O desenvolvimento de um *software* de gestão de estoque oferece diversos benefícios. Em primeiro lugar, a automação dos processos de controle de estoque permite uma gestão mais precisa e eficiente, eliminando erros humanos e reduzindo o tempo gasto em tarefas manuais. Além disso, o *software* possibilita o acesso rápido a informações atualizadas sobre o estoque,

como níveis de estoque, histórico de movimentações, datas de validade, entre outros, facilitando a tomada de decisões estratégicas.

Um dos principais desafios deste trabalho, consiste em desenvolver um *software* que seja amigável para o usuário, levando em consideração sua busca por praticidade e facilidade, onde o usuário deseja encontrar todas as informações necessárias em um único lugar, sem a necessidade de decifrar o sistema ao utilizá-lo.

3. DESENVOLVIMENTO DO SISTEMA

3.1 ARQUITETURA

Por se tratar de um trabalho acadêmico, e mesmo assim sem perder o total poder que a aplicação pode ter, a linguagem de programação proposta para o desenvolvimento da aplicação é a Python, ela é uma linguagem de alto nível, ou seja, ela é mais próxima da linguagem humana do que da linguagem de máquina. Python também é conhecida por sua sintaxe clara e legibilidade, facilitando o aprendizado e o desenvolvimento de jovens programadores.

A linguagem Python possui uma comunidade gigantesca e ativa. Isso resulta em amplo suporte, documentação abrangente e uma grande quantidade de recursos disponíveis, facilitando a resolução dúvidas, problemas e erros.

E uma linguagem com um enorme número de usuários, faz com que estes usuários assíduos e engajados criem e desenvolvam uma grande variedade de bibliotecas e *frameworks*.

3.2 AMBIENTE DE DESENVOLVIMENTO

O ambiente de desenvolvimento de um sistema é fundamental para o sucesso do projeto. Ele inclui ferramentas de programação, controle de versão e ambientes de teste, que permitem o desenvolvimento eficiente, a identificação de erros e aprimoramentos antes da implantação. Além disso, o ambiente de produção adequado garante a disponibilidade e desempenho do sistema, com uma configuração otimizada e escalável.

As ferramentas são a chave para um bom ambiente de desenvolvimento, elas proporcionam um espaço propício para a criação e programação de *software*. Isso inclui ambientes integrados de desenvolvimento (IDEs), editores de código, compiladores, depuradores e outras essenciais. Essas ferramentas permitem aos desenvolvedores escreverem, testarem e depurarem o código de forma eficiente, facilitando o processo de criação de um *software* de alta qualidade. Além disso, o ambiente de desenvolvimento também engloba a configuração de ambientes virtuais, bibliotecas e *frameworks* relevantes para a linguagem de programação utilizada, um ecossistema completo para o desenvolvimento do projeto.

As ferramentas para o ambiente de desenvolvimento deste projeto será:

- **Pycharm Community**: Está IDE é uma versão gratuita do PyCharm, desenvolvido pela JetBrains. ela é muito popular entre programadores da linguagem Python, oferecendo recursos como edição de código, depuração, testes e integração com sistemas de controle de versão, permitindo um desenvolvimento eficiente e produtivo.
- **GitHub**: Uma plataforma de hospedagem de código-fonte e colaboração para desenvolvimento de *software*. Ele permite que os desenvolvedores armazenem, gerenciem e compartilhem seus projetos, facilitando a colaboração entre equipes de desenvolvimento e fornecendo um histórico completo das alterações feitas no código. A documentação completa e clara também é essencial para orientar os usuários na correta utilização do *software*, isto também é facilmente proposto dentro do GitHub.

3.3 INTERFACE

A interface de um programa é a forma pela qual os usuários interagem com o *software*. Ela engloba todos os elementos visuais, como menus, botões, campos de entrada e exibição de informações, que permitem ao usuário interagir e controlar as funcionalidades do programa. A interface de um programa deve ser projetada de forma intuitiva, amigável e eficiente, facilitando a utilização e compreensão por parte dos usuários

Para a criação e o desenvolvimento da interface da aplicação, será utilizada uma biblioteca chamada Streamlit. Lançada em 2020, a Streamlit é conhecida por poder criar rapidamente e facilmente, aplicativos web interativos para visualização e análise de dados, bem como painéis interativos, e uma visibilidade simplista e bonita, o que é excelente para o projeto proposto neste trabalho.

Algumas outras vantagens da utilização da Streamlit para o desenvolvimento da interface são:

- A possibilidade de poder se integra perfeitamente com outras bibliotecas e ferramentas populares em Python, como Pandas, Matplotlib e Plotly.
- Com ela é possível fazer a implantação dos aplicativos desenvolvidos, permitindo que sejam compartilhados e acessados rapidamente por outros usuários, sem a necessidade de configurações complexas de hospedagem.
- A Streamlit também possui uma comunidade ativa de desenvolvedores e usuários assíduos, o que significa que há suporte, documentação e exemplos disponíveis. Ainda existem vários pacotes e extensões criados pela comunidade.

3.4 BANCO DE DADOS

Da mesma maneira e com o mesmo propósito de termos um ambiente de desenvolvimento para o *software*, também é necessário ter um ambiente de desenvolvimento para o banco de dados. Um ambiente de desenvolvimento de banco de dados também contém um conjunto de ferramentas, *software* e recursos que permitem projetar, desenvolver e testar sistemas de banco de dados.

O Firebase é uma plataforma de desenvolvimento de aplicativos móveis e web, fornecida pelo Google de forma gratuita. Ele oferece uma ampla gama de serviços e ferramentas que auxiliam os desenvolvedores a criar e escalar aplicativos de forma rápida e eficiente. Seu grande diferencial é que a estrutura do banco de dados é feita de forma não relacional, também conhecida como NoSQL, isso garante o desenvolvimento de um banco de dados com maior performance e flexibilidade.

O termo NoSQL surgiu em 1998 por Carlo Strozzi, referindo-se a um banco de dados relacional de código aberto que não possuía uma interface. Strozzi argumentava que o movimento NoSQL era significativamente diferente do modelo relacional e, portanto, deveria ser chamado de "NoREL" ou algo similar. (STROZZI, 2000). E esse fato de ser uma estrutura relativamente novas faz com que tenha uma fácil integração com tecnologias modernas, sendo ideal para *BigData* e computação na Nuvem.

Com uma base de dados estruturada no formato NoSQL é possível criar uma REST API e assim interagir com serviços web de forma rápida, simples e segura. Ela permite que diferentes sistemas se comuniquem de forma eficiente, facilitando a integração entre eles, independentemente da plataforma.

Seguindo essa linha de pensamento, e também pensando no futuro e na ampliação desta aplicação que o Firebase foi escolhido como ambiente de desenvolvimento do banco de dados. Vale ressaltar também que, o Firebase inclui recursos como autenticação de usuários, armazenamento em

nuvem, banco de dados em tempo real, hospedagem de aplicativos, mensagens em tempo real, entre outros.

3.5 REGRAS DE NEGOCIO E REQUISITOS

Uma regra de negócio é uma linha que define como um determinado processo deve ser executado dentro de uma organização. Esta linha estabelece as condições, restrições e procedimentos que orientam nas operações, ajudando nas decisões da empresa, garantindo uma consistência e uma eficiência em conformidade com as políticas da empresa.

Seguindo este pensamento, de forma bem resumida, a regra de negocio da Fastcar Premium é garantir a organização e o desenvolvimento do seu estoque de forma organizada, limpa e ágil. O *software* deve ser capaz de fazer com que qualquer pessoa em seu uso posso incluir e consultar de forma rápida e fácil qualquer peça do seu estoque. E principalmente, o *software* deve ser segura o desenhado fielmente de forma que não possa conter erros. Fazer com que o *software* seja simplista reduz o erro humano.

Agora, pensando em cima das regras de negócio da Fastcar Premium, devemos esboçar os requisitos funcionais e não funcionais, que irão nortear o desenvolvimento do projeto.

Requisitos funcionais são especificações que descrevem as funcionalidades e comportamentos que um sistema de *software* deve possuir. Eles definem o que o sistema deve fazer, incluindo interações com usuários ou outros sistemas. Já os requisitos não funcionais, são características e restrições que definem a qualidade e o desempenho do sistema, abrangendo aspectos como segurança, usabilidade, confiabilidade, escalabilidade e compatibilidade.

Logo abaixo está descrito os requisitos funcionais e não funcionais aplicados as regras de negócio:

- Inclusão: O *software* deve incluir cada peça do estoque em uma locação dentro das dependências da empresa. O usuário deve entrar com as informações da peça o sistema deve criar uma locação e incluir a peça na sua lista de consulta. O *backend* irá pegar os dados informados pelo usuário, enviar para o Firebase onde será incluído no banco de dados.
- Consulta: O usuário será capaz de consultar as peças dentro do estoque de forma simples, utilizando um filtro claro ele é capaz de saber rapidamente se a peça tem disponibilidade e sua localização. A comunicação com o Firebase é feita através de uma API Rest, devolvendo uma lista com as peças existentes no estoque, e conforme for aplicando o filtro a lista irá excluir as linhas que não se encaixam na pesquisa.
- Alteração: Caso seja necessário alterar alguma informação de uma peça cadastrada no sistema, o usuário irá informar qual a peça e quais as novas informações que serão alteradas na base de dados. O sistema por sua vez irá localizar através de um index que representa a peça, criado no momento em que a peça foi inserida no estoque, e enviar as novas informações para o Firebase fazer a alteração dos dados da peça.
- Exclusão: Em poucos casos, a peça fica indisponível para uso, logo ela deve ser excluída do estoque. Para manter uma segurança a peça simplesmente é alterada para uma outra raiz da arvore da base de dados, passando de disponível (“peças em estoque”) para indisponível (peças excluídas), desta maneira, caso a peça volte a ficar disponível, é possível facilmente retornar ela para a base de dados das peças disponíveis.
- Venda: Será o principal meio de baixa do estoque do sistema, o usuário seleciona os itens que serão baixados (vendidos) e através das informações fornecidas pelo usuário, de quem está

comprando ou para que carro as peças estão indo, o sistema consegue alterar a peça de estado disponível para peça vendida, sendo possível listar estas peças vendidas.

3.6 DIAGRAMAS

Diagramas são representações gráficas que ajudam a visualizar e compreender informações de forma clara e organizada. No desenvolvimento de *software*, eles representam visualmente informações, processos e interações do sistema. Eles auxiliam na análise, modelagem, identificação de problemas e definição da arquitetura do *software*. Também ajudam na documentação, identificação de problemas, tomada de decisões e alinhamento de partes interessadas.

Um diagrama de caso de uso demonstra as interações que o usuário tem dentro de um sistema. Ele captura as principais funcionalidades do sistema do ponto de vista do usuário. Ele é usado para modelar requisitos funcionais do *software*, identificar interações e definir a funcionalidade que o sistema deve fornecer.

A Figura 1 (página 16) demonstra o Diagrama do caso de uso do sistema, nela o ator a principio só pode executar duas ações Cadastrar e Consultar, porém dentro da consulta também é permitido que o ator possa Alterar, Vender e Excluir.

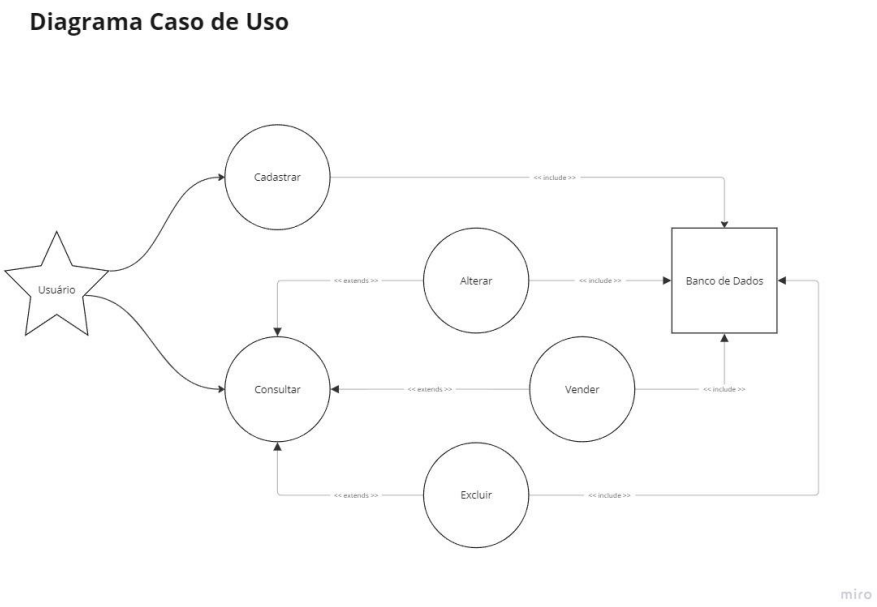

Figura 1 Diagrama de Caso de Uso. Fonte: Autor.

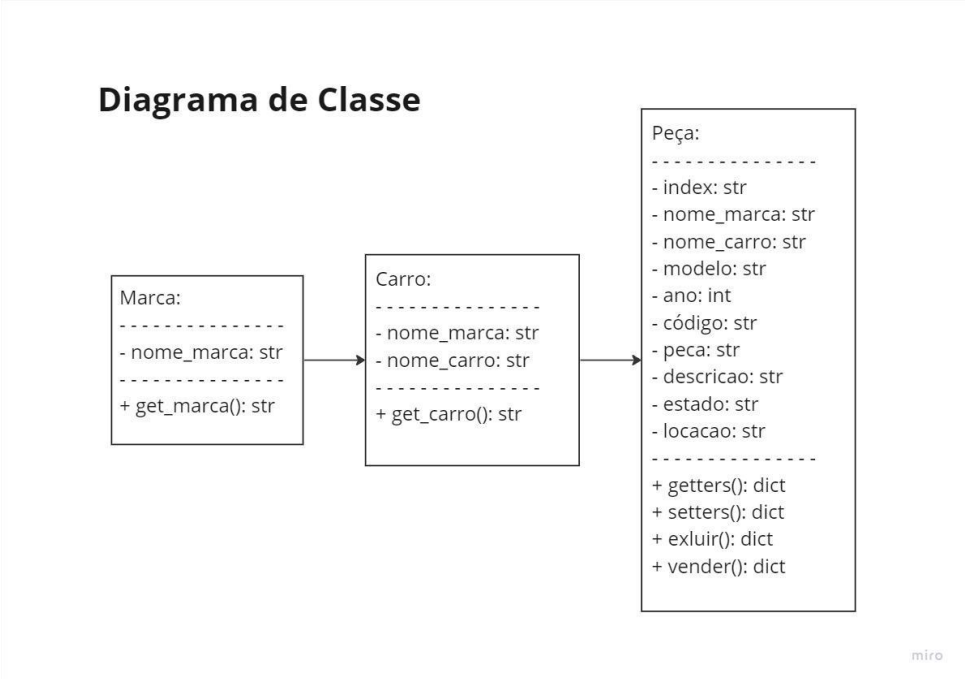

Figura 2: Diagrama de Classes. Fonte: Autor.

Já a Figura 2 (página 16) temos uma Diagrama de Classe, nele é possível ter uma visão geral da estrutura e dos relacionamentos das classes

dentro da aplicação. É usado para modelar a organização das classes, seus atributos, seus métodos, suas propriedades, seus relacionamentos e as interações entre os objetos do sistema. Em resumo, o sistema irá tratar de apenas uma entidade, “Peça”, e esta entidade deve obrigatoriamente pertencer a pelo menos um “Carro”, e todo carro tem uma “Marca”.

Em seguida, na página 18, temos a Figura 3: Diagrama de Sequência. Um diagrama de sequência mostrar de forma sucinta a interação sequencial entre usuário, objetos e partes de um sistema. Mostra a troca de mensagens (solicitações e respostas) e o fluxo de controle durante a execução de um determinado cenário ou função. Em outras palavras, diagramas de sequência representam interações entre objetos em uma série de etapas e nos ajudam a entender como um sistema se comporta ao longo da utilização, no caso da sequência é fácil distinguir que a primeira coluna de quadrados é a interface do sistema, seguido pela aplicação e por fim o banco de dados.

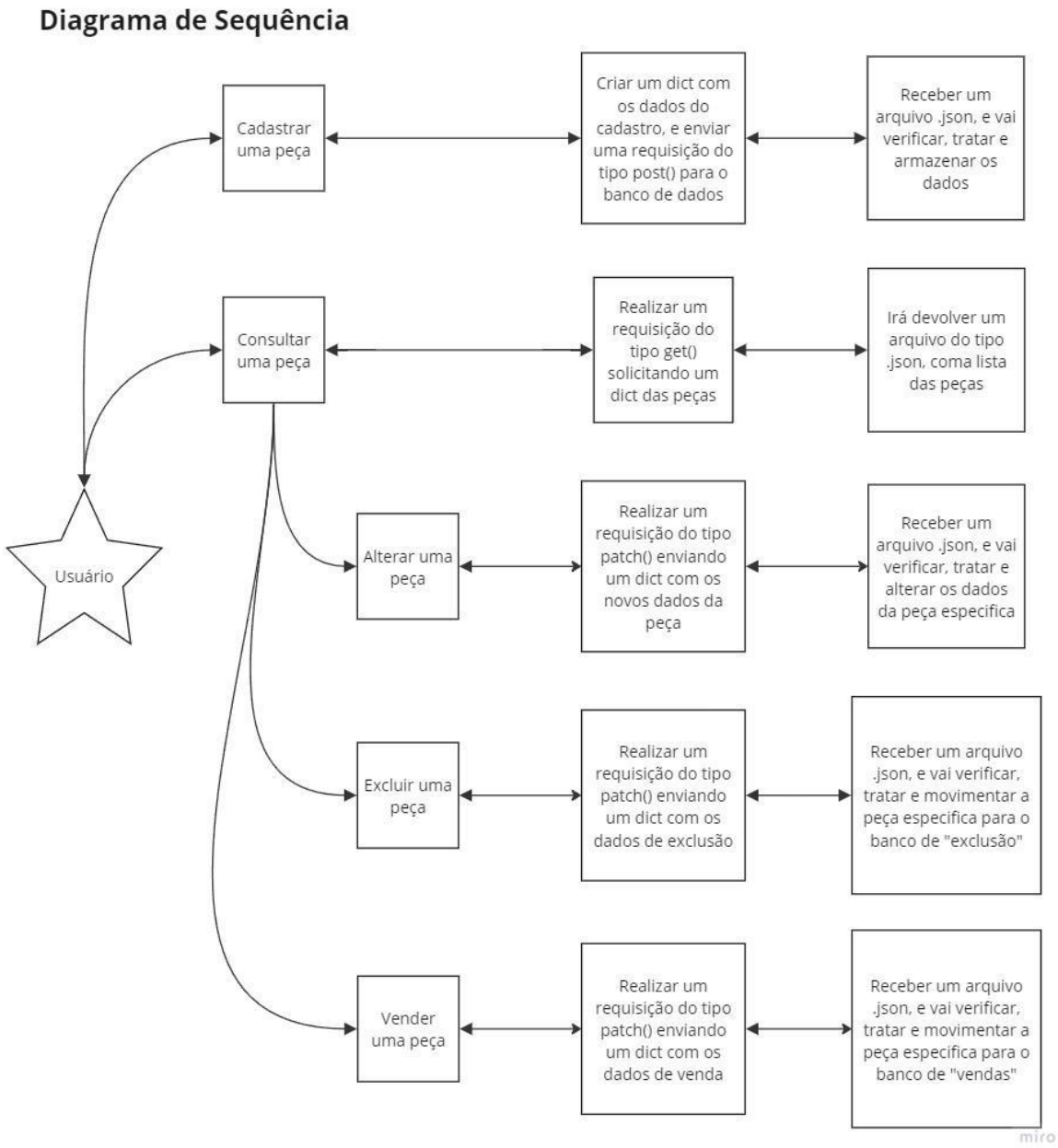

Figura 3: Diagrama de Sequência. Fonte: Autor.

Outro diagrama muito importante para auxiliar no desenvolvimento de sistemas, é o Diagrama de Entidade e Relacionamento (DER; figura 4), com ele é possível visualizar as entidades de forma aberta, vendo os atributos, os métodos e os relacionamentos dentro do escopo do banco de dados. Esta visão permite modelar a organização lógica dos dados e a entender a comunicar da estrutura do seu banco de dados de forma clara e sistemática, facilitando no desenvolvimento do projeto, e na implementação e manutenção do banco de dados. Como dito anteriormente, nele só será representado uma única entidade (“Peça”) que é a chave e o propósito da criação desta aplicação.

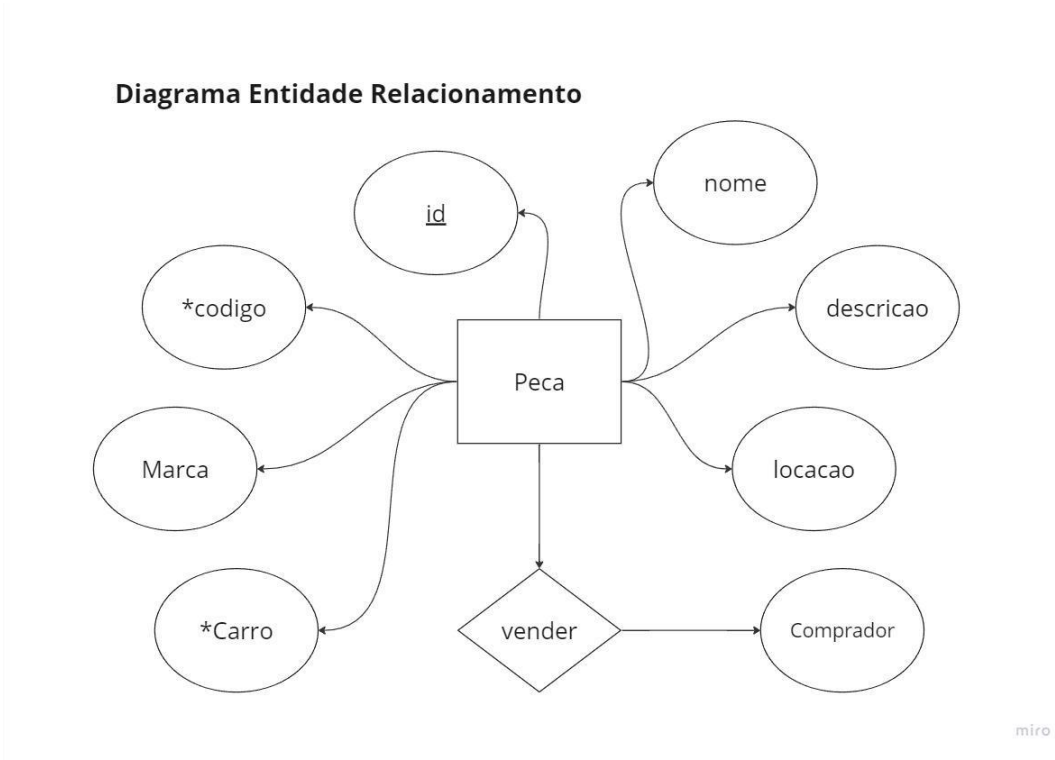

Figura 4: Diagrama Entidade Relacionamento. Fonte: Autor.

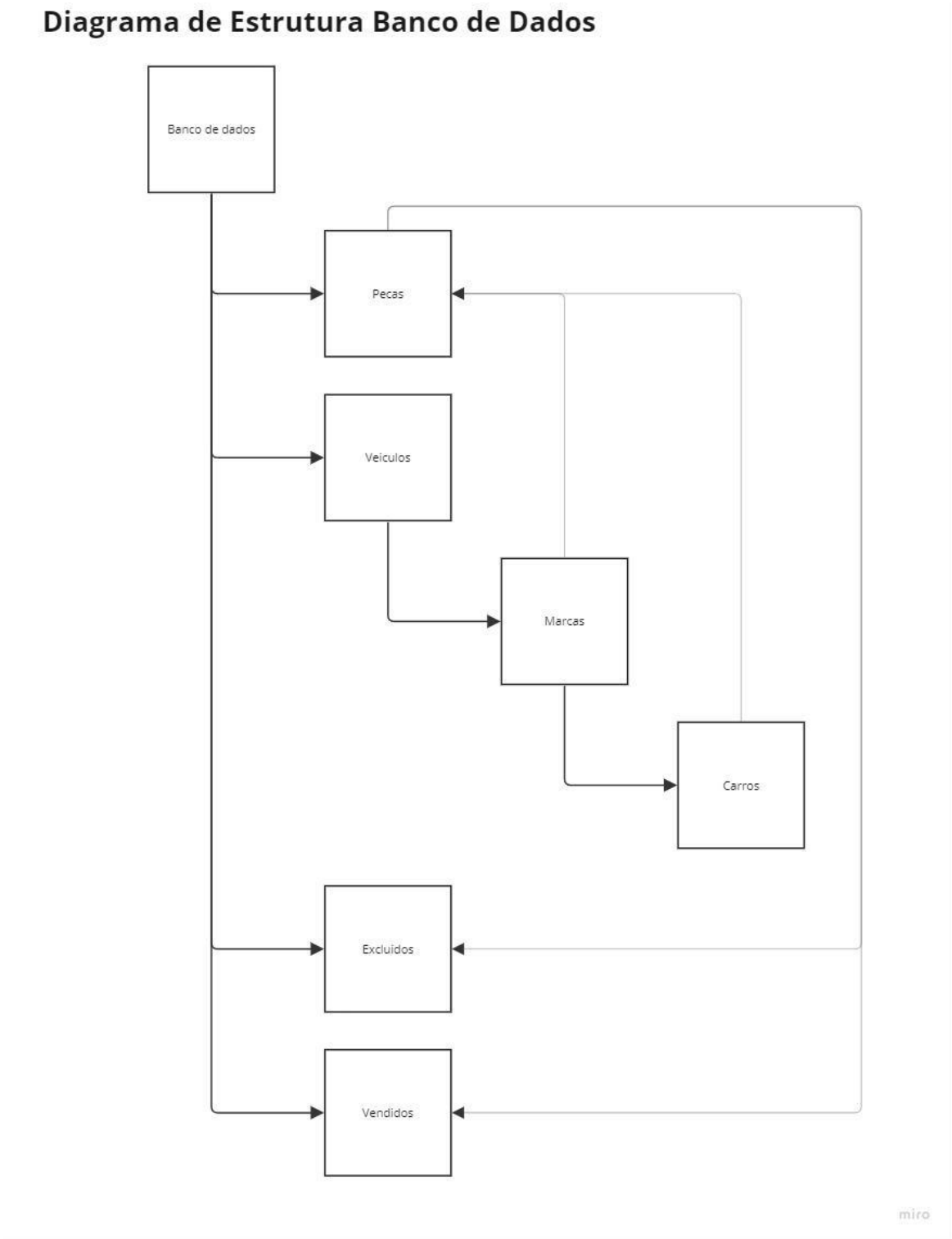

Figura 5: Estrutura Banco de Dados. Fonte: Autor.

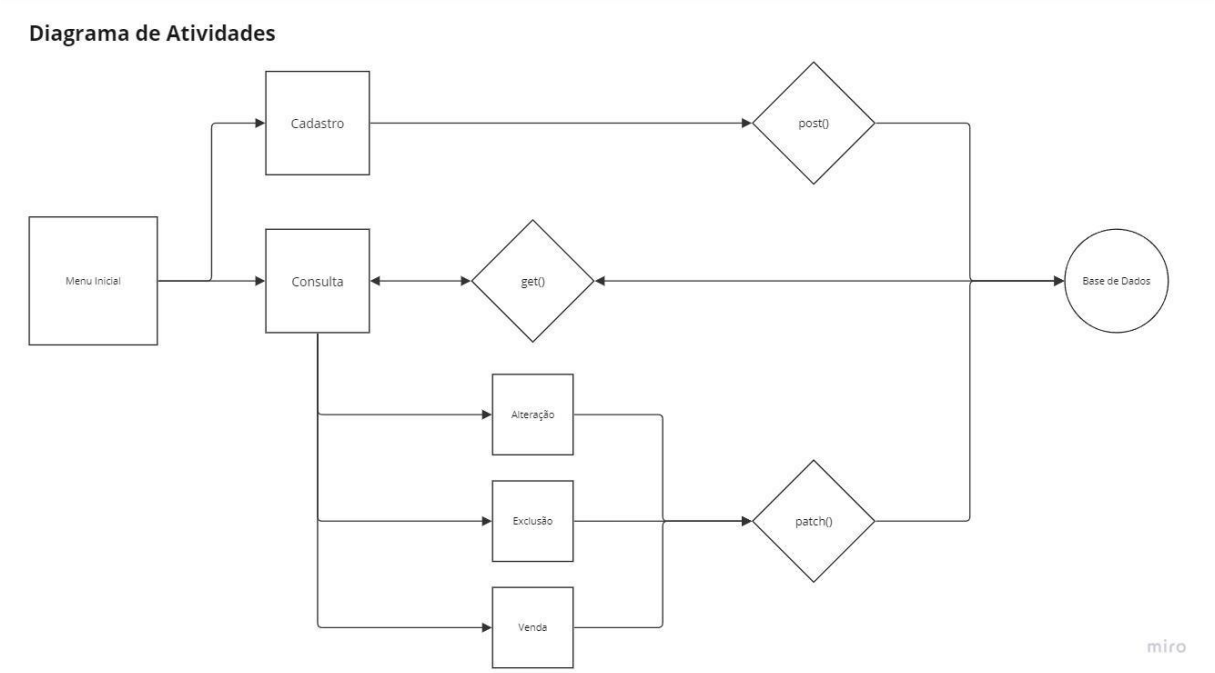

Figura 6: Diagrama de Atividades. Fonte: Autor.

Na figura 6, está representando um Diagrama de Atividades, nele é possível ver o fluxo de atividades do sistema, ele descreve a sequência de ações, decisões e ramificações, mostrando as etapas e decisões do processo de forma clara e organizada, e também auxilia a compreender as etapas e a lógica de um processo.

3.7 PROTÓTIPOS

Imagem 1: Página inicial.

Imagem 2: Menu lateral.

Formulário de Cadastro

Qual a marca do veículo?

AUDI

Qual o nome do veículo?

100

Qual o modelo do veículo?

Insira qual o modelo especifico do veículo!

Qual o ano do veículo?

Choose an option

Qual o código da peça?

Insira todos os códigos que se aplica a peça!

Qual a peça cadastrada?

Insira o nome da peça cadastrada

Qual a peça cadastrada?

Nova

Qual a descrição da peça cadastrada?

Insira a descrição da peça cadastrada

Qual a locação da peça?

Insira a locação da peça!

Cadastrar

Imagem 3: Tela de Cadastro.

Imagem 4: Tela de consulta.

4 CONSIDERAÇÕES FINAIS

Para um bom desenvolvimento de sistema, e para que ele seja bem sucedido primeiramente, é essencial entender os requisitos e necessidades da empresa, garantindo que o sistema atenda às suas demandas específicas.

Além disso, a escolha adequada de tecnologias é um fator chave. No caso mencionado, a utilização da linguagem Python, em combinação com o ambiente de desenvolvimento do PyCharm Community e do GitHub, juntamente com o *framework* Streamlit e o banco de dados Firebase, demonstra uma abordagem moderna e eficiente, com grande potencial de expansão e evolução.

A usabilidade e a experiência do usuário são aspectos cruciais para esta regra de negócio e para o sucesso da aplicação. Garantir uma interface intuitiva, responsiva e amigável para o usuário facilita a interação com o sistema e aumenta a eficiência no controle do estoque de peças.

O desenvolvimento bem-sucedido da aplicação, também envolve uma arquitetura de *software* bem definida e modular, que permita a escalabilidade e manutenção do sistema. A separação clara de responsabilidades, como o uso de camadas para lidar com a lógica de negócios, a interface com o usuário e a comunicação com o banco de dados, contribui para a organização e reusabilidade do código. Também vale mencionar, que durante todo o processo de desenvolvimento da aplicação, diversos testes e validações, bem como a consideração de cenários de uso e possíveis exceções, foram feitos para garantir a qualidade do *software* e a confiabilidade dos dados manipulados.

Com uma abordagem cuidadosa em relação a esses aspectos, é possível obter um desenvolvimento bem-sucedido da aplicação para controle de estoque de peças na empresa Fastcar Premium, fornecendo uma ferramenta confiável e eficiente para auxiliar em suas operações.

5 REFERÊNCIAS

ARRUDA, Clara F.; **Desenvolvimento de um Software Online de Gerenciamento de Controle de Estoque Interno para Laboratórios**. UNESP. Botucatu – SP, 2023. Acessado em 19 de junho de 2023: < https://repositorio.unesp.br/handle/11449/242758 >

DE PÁDUA PAULA FILHO, W. **Engenharia de software**. Vol. 2. LTC, 2003

GONÇALVES, Gustavo C.; **Desenvolvimento de uma organização para controle de estoque de fábricas através do software CRM *Salesforce***. UNESP. Guaratinguetá – SP, 2023. Acessado em 17 de junho de 2023: < https://repositorio.unesp.br/handle/11449/242512 >

MOREIRA, D. A. **Administração da produção e operações**. 2. ed. São Paulo: Cengage Learning, 2012.

SLACK, N.; CHAMBERS, S.; JOHNSTON, R. **Administração da produção**. 2. ed. São Paulo: Atlas, 2002.

SOUZA, Fernando J.; SANTANA, Paulo H. A.; **Estudo de Caso: Análise entre Banco de Dados Relacional e Não Relacional**. Centro Universitário de Anápolis – Unievangélica. Anápolis – GO, 2017. Acessado em 21 de junho de 2023: < http://repositorio.aee.edu.br/handle/aee/46 >