

Catu 21 - Sistema Web para Divulgação de Produtos de Microempreendedores

Gabriela Lima dos Anjos, Rodolfo Francisco de Oliveira

¹Curso Superior de Análise e Desenvolvimento de Sistemas
Instituto Federal São Paulo (IFSP) - Campus Hortolândia
Av. Thereza Ana Cecon Breda, 1896 – 13183-250 – São Paulo – SP – Brasil

anjos.gabriela@aluno.ifsp.edu.br, rodolfo.oliveira@ifsp.edu.br

Abstract. *When opening a small business, the micro-entrepreneur may find it difficult to attract and retain customers, especially in the face of competition from entrepreneurs/companies already consolidated in the market. In this context, the present work includes the development of a web system called Catu 21, that aims to promote small entrepreneurs through a virtual catalog that displays their products and store. The project was primarily developed in Django, which is a full stack and open-source web framework written in Python. The system created is responsive and allows micro-entrepreneurs to register in this catalog, which aims to group a variety of registered stores, simplifying the search for specific products by different consumers.*

Resumo. *Ao abrir um pequeno empreendimento, o microempreendedor pode encontrar dificuldades para atrair e reter clientes, principalmente diante da concorrência de empreendedores/empresas já consolidados no mercado. Neste contexto, este trabalho contempla o desenvolvimento de um sistema web denominado Catu 21, que tem o propósito de fazer a divulgação de pequenos empreendedores através de um catálogo virtual que exiba seus produtos e loja. O projeto foi desenvolvido primordialmente em Django, um framework web full stack e de código aberto escrito em Python. O sistema criado é responsivo e permite que os microempreendedores realizem seus respectivos cadastros neste catálogo que visa agrupar uma variedade de lojas cadastradas simplificando a busca de produtos específicos por diferentes consumidores.*

1. Introdução

Conquistar novos clientes é uma das principais dificuldades das micro e pequenas empresas. Muitas vezes isso ocorre devido à falta de investimento em ações de *marketing* (principalmente quando são negócios mais familiares) ou devido à falta do planejamento estratégico necessário para atrair clientes [Serasa 2021].

O comércio é importante para a economia de uma região, especialmente para os pequenos empreendedores, que muitas vezes, ainda segundo [Serasa 2021], encontram dificuldades em se destacar em um mercado dominado por grandes empresas. Com o crescente uso da Internet e a facilidade de acesso à informação, um catálogo virtual de produtos pode ser uma solução eficiente para a divulgação desses pequenos negócios.

Este artigo apresenta um projeto de desenvolvimento *web* de um catálogo virtual de produtos de pequenos empreendedores, visando a divulgação de seus produtos e de

contato para possíveis consumidores. O intuito do projeto é oferecer uma plataforma de fácil acesso para o público geral, contribuindo para o fortalecimento dos pequenos negócios.

O objetivo deste trabalho é desenvolver um sistema *web* (*back-end* e *front-end*) responsivo que tem o propósito de divulgar microempreendedores do Brasil e conectá-los com potenciais consumidores. Os microempreendedores, mediante cadastro prévio, poderão fornecer suas informações de contato e o catálogo de suas lojas, além de atualizá-los periodicamente. Os usuários visitantes, que não precisam de cadastro, poderão visualizar informações das lojas cadastradas e seus catálogos. Além disso, os administradores podem gerir os usuários comerciantes.

A organização deste trabalho segue a seguinte estrutura: a seção 2 faz referência aos trabalhos correlatos, comparando-os ao presente projeto. A seção 3 detalha as ferramentas empregadas no desenvolvimento. A seção 4 mostra o desenvolvimento prático do trabalho, relatando os passos do processo. Por fim, a seção 5 encerra o trabalho e indica possíveis melhorias futuras.

2. Trabalhos Correlatos

Nesta seção estão resumidos 3 trabalhos correlatos encontrados em pesquisa dentro do Google Acadêmico e portal de Trabalhos de Conclusão de Curso do IFSP - Hortolândia [IFSP 2023], em 2023, através das palavras-chave: desenvolvimento web, produto, comércio eletrônico, Python e Django. Apesar da busca feita com tais palavras, foi possível encontrar trabalhos que fizeram uso de outras tecnologias *web*.

2.1. Sistema Web para divulgação e venda de produtos de uma confeitaria

O trabalho desenvolvido por [Santos 2021] tem como objetivo o desenvolvimento de um sistema *web* (*e-commerce*) para a divulgação e venda de produtos de uma confeitaria, permitindo a divulgação, customização e a realização de pedidos dos produtos, além de controlar o *status* dos pedidos em tempo real. Como base do desenvolvimento foi utilizado o padrão Spring MVC (*Model-View-Controller*), linguagem Java e o *framework* de mapeamento objeto-relacional Hibernate para a persistência de dados. O autor iniciou o trabalho através de pesquisa *online* com as panificadoras e confeitarias de sua localidade sobre o interesse dos mesmos na implantação de um comércio eletrônico; em seguida fez a modelagem, desenvolvimento do sistema e sua revisão. O método utilizado processo sequencial linear de [Pressman and Maxim 2016], onde as tarefas e atividades ocorrem de forma sequencial, iniciando pela especificação dos requisitos e em seguida avança para as fases de modelagem e entrega.

2.2. EasyProposal: Aplicação web para elaboração de propostas de vendas de máquinas de produção

O trabalho desenvolvido por [Schiarolli and Tomazeti 2022] tem o objetivo de oferecer uma plataforma *web* com funções de controle, gestão de vendas e geração de propostas para automatizar serviços de empresas que utilizam planilhas eletrônicas para controle interno na elaboração de propostas de venda de máquinas de produção. Dessa forma, o sistema automatiza o controle e elaboração de proposta das empresas. As principais ferramentas utilizadas foram o *framework web* ASP.NET e o React. O processo de

desenvolvimento utilizado foi o Incremental, onde pequenas partes do projeto são entregues e refinadas a cada entrega.

2.3. Desenvolvimento de uma plataforma virtual para negociação de produtos ou serviços

O objetivo do trabalho apresentado em [Konig and da Silva 2022] é desenvolver um sistema *web* que divulgue e controle pedidos de produtos e serviços de pequenas empresas de forma acessível à população. O projeto apresenta a estratégia de *marketing* e vendas *omnichannel* que flexibiliza a negociação entre cliente e vendedor. As tecnologias utilizadas foram a linguagem de programação Python, o banco de dados MySQL e o *framework* Django, que trabalha com a estrutura MTV (*Model, Template, View*), dividido na lógica de integração de dados (*Model*), arquivos estáticos (*Template*) e regra de negócio (*View*). O autor não informa a metodologia utilizada.

2.4. Comparação dos trabalhos

Os três trabalhos apresentados apresentam o desenvolvimento de sistemas *web* para facilitar a divulgação e comercialização de produtos e serviços. Onde cada um desses projetos apresenta diferentes abordagens e tecnologias, mas têm em comum a intenção de tornar a compra e venda de produtos e serviços mais acessível e eficiente por meio da Internet.

Ao comparar os trabalhos mencionados com este trabalho, é possível identificar que o sistema *web* desenvolvido (Catu 21) tem como objetivo fornecer um catálogo eletrônico exclusivamente de pequenos empreendedores. A proposta é contribuir para a divulgação dos pequenos comerciantes e facilitar a busca por seus produtos. Ademais, o sistema *web* é alimentado pelas informações do próprio comerciante, permitindo que tenha controle sobre sua apresentação e atualização do catálogo.

3. Fundamentação Teórica

Esta seção do artigo apresenta o detalhamento das ferramentas e recursos empregados, permitindo estabelecer uma estrutura conceitual sobre o funcionamento do projeto.

3.1. Processo Incremental

O processo incremental, que é uma evolução do modelo cascata, foi escolhido para o desenvolvimento do trabalho. De acordo com [Pressman 2006], a metodologia incremental trabalha com pequenas entregas de *software*, combinando com os elementos do modelo Cascata aplicados iterativamente. Logo, são realizados refinamentos sucessivos que são melhorados a cada iteração. A Figura 1, extraída do livro Engenharia de Software de [Pressman and Maxim 2016], é uma representação visual do processo em questão.

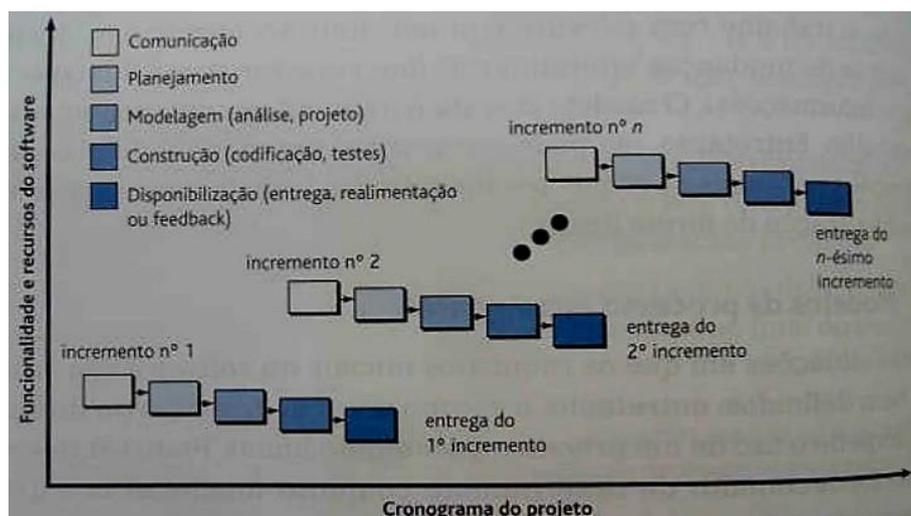


Figura 1. Processo Incremental (Fonte: [Pressman and Maxim 2016])

3.2. Python

A linguagem de programação escolhida para o projeto é o Python. De acordo com o *site* da [Amazon Web Services 2023], Python tem se tornado cada vez mais popular entre os desenvolvedores devido à sua clareza, simplicidade e flexibilidade, além de ser uma linguagem multiparadigma, versátil e poderosa, com ampla aplicação em diferentes áreas, como desenvolvimento *web*, inteligência artificial e ciência de dados. Uma das vantagens do Python é a grande comunidade de desenvolvedores que constantemente contribuem com bibliotecas e ferramentas, expandindo a capacidade da linguagem.

3.3. Django

Conforme o próprio *site* [Django Software Foundation 2023] do Django, o mesmo é um *framework web* de alto nível, que tem ganhado popularidade entre os desenvolvedores devido à sua facilidade de uso e flexibilidade. Gratuito e de código aberto, ele foi escrito em Python, e permite a criação rápida de aplicativos *web* robustos e escaláveis. Com uma arquitetura bem definida e uma sintaxe simples, o Django segue o padrão *Model-View-Template* (MVT) e oferece muitos recursos integrados, incluindo um ORM (*Object-Relational Mapping*), sistema de administração de banco de dados, sistema de autenticação de usuários e muito mais. O Django é usado por empresas e desenvolvedores em todo o mundo para criar *sites* e aplicativos *web* complexos, desde aplicativos corporativos até plataformas de comércio eletrônico e muito mais. Sua ampla gama de recursos e suporte ativo da comunidade o tornam uma escolha popular para o desenvolvimento *web* em Python.

No padrão MVT, o "modelo"(ou *model*) armazena as informações, é responsável por fazer o gerenciamento do banco de dados; a "visualização"(ou *view*) tem a função de descrever os dados que serão apresentados, ou seja, trabalha com a lógica do projeto, fazendo toda a manipulação das informações. Desta maneira, a "*view*"no MVT se diferencia do padrão MVC (*Model-View-Controller*) pois a primeira descreve quais dados serão apresentados, ou seja, formata e controla a apresentação e pode conter lógica adicional e a segunda exibe os dados; por fim, o *template* permite que as informações sejam visualizadas através da renderização das páginas.

No Django, as tabelas de banco de dados se tornam classes Python e, os aplicativos da *web* acessam e gerenciam dados por meio de modelos do próprio *framework*. Os campos do banco de dados são convertidos em atributos de classe [IBM 2023]. Ainda conforme o autor, o *framework* fornece a funcionalidade CRUD (criar, ler, atualizar, excluir), *HttpResponse* (resposta enviada por um servidor web após receber uma solicitação), lida com *cross-site scripiting* (vulnerabilidade *web* onde um código malicioso é inserido e executado no navegador, comprometendo a segurança), recursos de administração de *software*, e, reinterpretação de comandos não autorizados para ajudar a proteger o aplicativo e os usuários.

3.4. SQLite

Conforme [Lima 2023], o SQLite é um Sistema de Gerenciamento de Banco de Dados Relacional (SGBDR) de código aberto e de natureza leve e rápida. É amplamente adotado em aplicações e projetos que necessitam de um banco de dados embutido, ou seja, integrado diretamente à aplicação, eliminando a necessidade de um servidor separado. O SQLite utiliza a linguagem SQL (*Structured Query Language*) para consulta e manipulação de dados, incluindo operações comuns de banco de dados, como o CRUD (acrônimo para Criar, Ler, Atualizar e Excluir) que representa as operações básicas de gerenciamento de dados em bancos de dados.

O Django, por padrão, utiliza o SQLite3 como Sistema de Gerenciamento de Banco de Dados (SGBD) e cria um arquivo padrão chamado 'db.sqlite3' que armazena localmente os dados do projeto. O SGBD pode ser alterado se desejável pela equipe de desenvolvimento.

3.5. HTML

Conforme o *site* [Mozilla 2021], HTML (*Hypertext Markup Language*) é uma linguagem de marcação utilizada para criar a estrutura de conteúdo de uma página *web*. Ela é composta por elementos que representam diferentes tipos de conteúdo, como texto, imagens, *links* e formulários. Esses elementos são organizados em uma hierarquia de árvore, em que os elementos pai contêm elementos filhos.

Ao criar uma página HTML, é possível definir a aparência do conteúdo utilizando CSS (*Cascading Style Sheets*) e adicionar interatividade com JavaScript. Além disso, HTML é uma linguagem acessível, permitindo a utilização de recursos para tornar a página mais inclusiva para usuários com deficiências visuais ou auditivas. Em resumo, HTML é a base da *web* e é essencial para qualquer desenvolvedor que deseja criar conteúdo para a Internet.

3.6. CSS

De acordo com o publicado no *site* [Mozilla 2023], CSS é uma linguagem de folha de estilo usada para descrever a apresentação de um documento escrito em HTML ou XML (como uma página da *web*). Com o CSS, é possível definir o *layout*, as fontes, as cores e outros aspectos visuais do conteúdo da página. O CSS permite que os desenvolvedores separem a apresentação da estrutura do documento, tornando a manutenção e atualização da aparência visual do *site* mais fácil e eficiente. Além disso, o CSS permite que diferentes dispositivos e navegadores possam interpretar a página da mesma maneira, proporcionando uma experiência consistente para os usuários finais.

3.7. Bootstrap

Vide o próprio *site* do Bootstrap, [Bootstrap 2023], este é um *kit* de ferramentas de *front-end* repleto de recursos que permite a rápida criação de interfaces responsivas de *websites*. É um *framework* de código aberto construído em HTML, CSS e JavaScript que auxilia na construção de interfaces de *sites*. O Bootstrap contém ampla variedade de conjuntos de elementos previamente construídos (barras de navegação, formulários e outros) que podem ser utilizados dentro das páginas HTML dos projetos *web* e *mobile*, proporcionando praticidade no desenvolvimento e reduzindo a quantidade de *scripts* que podem ser escritos caso tal ferramenta não seja utilizada.

3.8. JavaScript

JavaScript, segundo [Silva 2020], é uma linguagem que permite definir, alterar e controlar de forma dinâmica a apresentação de um documento HTML, manipulando, também, as regras do CSS implementadas. Logo, com o JavaScript é possível que os *sites* sejam interativos, fornecendo ao usuário mais opções de navegação, como, por exemplo, recursos de acessibilidade (alterar fonte e alterar contraste da página).

3.9. API

API ou *Application Programming Interface* (Interface de Programação de Aplicações) são, segundo a [Red Hat 2023], são conjuntos de ferramentas, definições e protocolos para a criação de aplicações de software, conectando soluções e serviços sem a necessidade da equipe de desenvolvimento saber como elas são implementadas. Ainda conforme o autor, a API funciona como contratos com documentações que representam um acordo entre as partes interessadas, onde uma parte envia uma solicitação remota e a outra responde.

4. Desenvolvimento

Nesta seção são apresentados os detalhes sobre como o projeto foi desenvolvido, através de diagramas, fluxos de atividades e a descrição de como o projeto foi implementado.

4.1. Especificação dos Requisitos

Segundo [Sommerville 2019] a especificação de requisitos trabalha com a descrição dos serviços que o sistema irá prestar, ou seja, reflete as necessidades do cliente/usuário que devem ser atendidas. Realizar a elicitação e análise de requisitos é essencial para um bom desenvolvimento de *software*. Se essa fase for mal desenvolvida, erros podem ocorrer durante o processo, aumentando o tempo e os custos do projeto. Os requisitos também são importantes para estimar o preço da modelagem, projeto, testes, implementação e manutenção.

Os requisitos podem ser categorizados em dois conjuntos: requisitos funcionais e requisitos não funcionais.

Requisitos funcionais são declarações dos serviços que o sistema deve fornecer, do modo como o sistema deve reagir a determinadas entradas e de como deve se comportar em determinadas situações [Sommerville 2019].

Na Tabela 1 estão listados os requisitos funcionais dos usuários visitantes (é importante ressaltar que tais usuários não possuem cadastro no sistema, assim qualquer

pessoa interessada no *website* pode acessá-lo). Na Tabela 2, os requisitos do comerciante dos produtos e, na Tabela 3, os do administrador do sistema.

Tabela 1: Requisitos Funcionais dos Usuários Visitantes

Identificação	Nome	Descrição
RF01	Pesquisar produtos	Permitir que os usuários pesquisem os produtos por nome
RF02	Detalhar produtos	Mostrar detalhes do produto
RF03	Redirecionar ao comerciante	Permitir que o usuário seja redirecionado à página de contato do vendedor

Tabela 2: Requisitos Funcionais dos comerciantes

Identificação	Nome	Descrição
RF04	Cadastrar perfil	Permitir que novos comerciantes se cadastrem no <i>site</i>
RF05	Efetuar login	Permitir que os comerciantes tenham um perfil dentro do <i>site</i>
RF06	Cadastrar Loja	Usuário deve cadastrar a loja para liberar as funcionalidades de cadastro e edição de produto e perfil
RF07	Gerenciar perfil	Permitir que os comerciantes alterem suas informações de perfil
RF08	Cadastrar produto	Permitir comerciante cadastrar produtos no catálogo
RF09	Editar produto	Permitir que comerciante edite os produtos de seu catálogo
RF10	Remover produto	Permitir que comerciante exclua seus produtos do catálogo

Tabela 3: Requisitos Funcionais do Administrador

Identificação	Nome	Descrição
RF11	Gerenciar Usuário	Desativar conta dos usuários; alterar senha; cadastrar/remover categorias dos produtos
RF12	Gerenciar Categoria do Produto	Cadastrar, editar ou remover categorias dos produtos

Requisitos não funcionais são restrições sobre os serviços ou funções oferecidas pelo sistema. Eles incluem restrições de tempo, restrições sobre o processo de desenvolvimento e restrições impostas por padrão ([Sommerville 2019]), e, se aplicam ao sistema como um todo.

Tabela 4: Requisitos Não Funcionais

Identificação	Nome	Descrição
RNF01	Acesso à Internet	Usuário deve ter acesso à Internet para utilizar o <i>site</i>
RNF02	Responsividade	Garantir que o sistema se adapte a diferentes tamanhos e resoluções de tela
RNF03	Segurança	Garantir que apenas os usuários cadastrados e donos de seus próprios produtos possam editá-los

Os casos de uso, de acordo com artigo publicado no *site* da [IBM 2021], representam textualmente e de forma gráfica (diagrama) os requisitos funcionais do sistema e do negócio. Desta maneira, incluem os relacionamentos entre os atores e o sistema definindo a sequência de ações que representa um resultado observável de valor.

Os casos de uso estão representados na Figura 2.

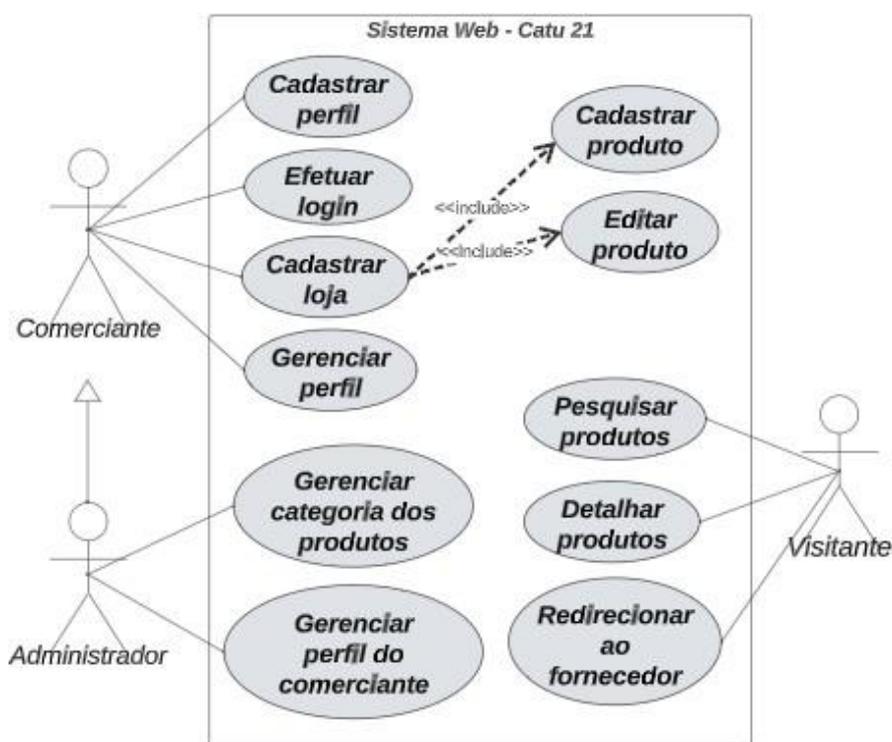


Figura 2. Diagrama de Casos de Uso - Catu 21

De forma resumida, o processo de utilização do sistema pelo comerciante é:

- Realizar cadastro pessoal e efetuar *login*;
- Cadastrar loja no sistema para conseguir cadastrar as informações detalhadas de seus produtos em um catálogo individual e excluir produto;
- O comerciante pode gerenciar seu perfil.

O processo de utilização do sistema pelo administrador é:

- Cadastrar ou remover categorias dos produtos conforme necessidade;
- Fazer o gerenciamento do perfil dos usuários cadastrados (comerciantes).

O processo de utilização do sistema pelo usuário visitante é:

- Usuário acessa o site e tem a visualização dos produtos de todos comerciantes na tela inicial, podendo detalhá-los e ser redirecionado à página da loja do comerciante;
- Na página do comerciante, o visitante pode ter acesso ao catálogo da loja e pode acessar as redes sociais e o endereço da loja física, se existir.

Em caráter de exemplo, na Tabela 5 estão detalhados os fluxos dos casos de uso RF08 e RF11. Porém, todos os fluxos citados estão detalhados no Anexo A.1.

Tabela 5: Fluxos dos Casos de Uso RF08 e RF11

RF08	Cadastrar Produto
1	Na tela inicial, o comerciante clica no botão 'Cadastrar Novo Produto'.
2	O comerciante preenche o formulário de cadastro de produto exibido pelo sistema.
3	Comerciante confirma o cadastro.
4	Sistema guarda as informações no banco de dados.
RF11	Gerenciar Usuário
1	O administrador acessa a lista de usuários cadastrados no sistema.
2	O sistema exibe a lista de usuários registrados.
3	O administrador seleciona um usuário específico para gerenciar suas informações.
4	O administrador salva as alterações.

O Diagrama de Entidade e Relacionamento (DER), segundo a [Miro 2023], é um diagrama que mostra como as entidades (sejam elas pessoas, objetos ou conceitos) interagem em um sistema, demonstrando visual e logicamente como os dados fluem de uma entidade para outra. Portanto é possível ter visualização ampla da estrutura de um sistema.

Ainda segundo [Miro 2023], o DER possibilita o mapeamento dos elementos-chave do banco de dados, permitindo uma prévia análise do funcionamento do *software* e a identificação de lacunas antes que o sistema entre em funcionamento. O DER possui três principais componentes: entidades, atributos e relacionamentos.

As entidades são a representação das pessoas, objetos, conceitos ou eventos. Os atributos são as características da entidade. E os relacionamentos mostram a interação entre as entidades.

Diagrama ER banco de dados - Catu 21

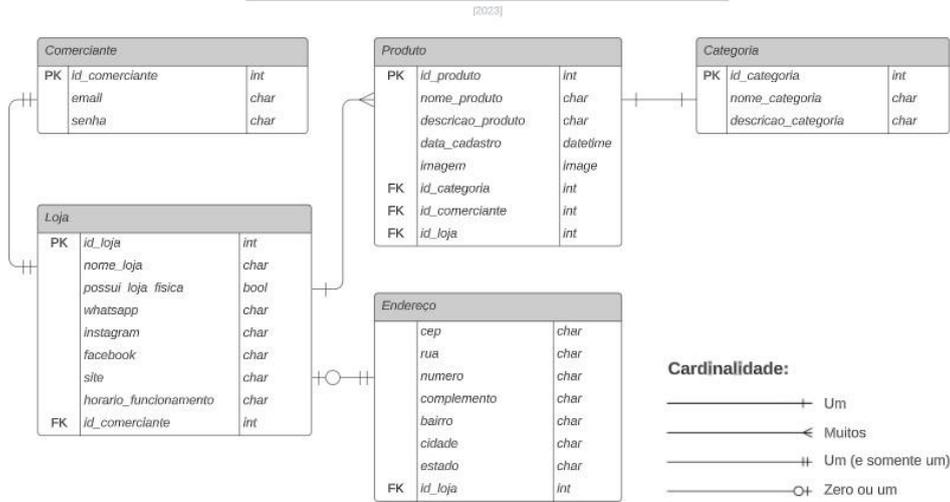


Figura 3. Diagrama Entidade Relacionamento - Catu 21

O DER, conforme apresentado na Figura 3, possui as entidades Comerciante, Loja, Endereço, Produto e Categoria.

A entidade Comerciante possui os atributos “id_comerciante”, “email” e “senha”, que armazenam, respectivamente a identificação única do usuário no sistema, o endereço de e-mail e senha que o usuário (comerciante) irá utilizar para acessar o sistema.

O comerciante pode informar se possui ou não loja física através do atributo “possui_loja_fisica”. Se possuir, então a entidade Endereço será utilizado para armazenar o endereço completo da loja (rua, número, complemento, bairro, cidade, estado e CEP) da loja através de seus respectivos atributos.

A entidade Loja possui os atributos “id_loja”, que permite identificação única da loja; “nome_loja”, que armazena o nome da loja associada ao comerciante; os atributos de redes sociais: “whatsapp”, “site”, “instagram” e “facebook” que armazenam os endereços das redes sociais da loja. O atributo “dias_funcionamento” armazena os dias e horários de funcionamento da loja.

A entidade Produtos permite armazenar todos os produtos do catálogo do comerciante, através dos atributos: “id_produto”, que permite identificação única do produto; “nome_produto”, responsável por armazenar o nome do produto; “descricao”, que contém uma descrição detalhada do produto cadastrado; “data_cadastro”, que é um campo de preenchimento automático da data na qual o produto foi cadastrado no sistema; “imagem”, que armazena uma imagem do produto; e “categoria”, que mostra a qual categoria o produto pertence.

A entidade Categoria é responsável pelo registro das categorias dos produtos no sistema. Seus atributos são o “nome_categoria” e “descricao_categoria” que guardam o nome da categoria do produto e sua descrição detalhada, respectivamente.

Resumidamente, o usuário (comerciante) faz seu cadastro e o da sua loja no sistema. A loja pode ou não ter endereço físico. O comerciante pode, então, iniciar o cadastro de produtos no seu respectivo catálogo. Um catálogo pode ter inúmeros

produtos e cada produto encaixa-se dentro de alguma categoria cadastrada no sistema pelo administrador.

4.2. Implementação do Sistema

O *Back-end* do sistema foi implementado através do *framework* Django, que utiliza, por padrão, o banco de dados SQLite e a linguagem de programação Python. No *Front-end* foram utilizados recursos do Bootstrap, JavaScript, HTML e CSS.

O desenvolvimento do trabalho iniciou-se com a criação das telas de *login* e de cadastro dos comerciantes. Ao realizarem o cadastro, os mesmos devem primeiramente fornecer dados pessoais (*e-mail* e senha). A senha deve ter no mínimo 6 caracteres e, se o usuário já estiver cadastrado no sistema, será automaticamente redirecionado para a tela de login.

Após tal procedimento, o comerciante é redirecionado a uma tela de cadastro da loja, onde deverá fornecer dados pertinentes (localização, nome da loja, redes sociais e dias/horário de funcionamento). O comerciante pode informar se a loja possui espaço físico ou não. Se existir, um formulário para preencher o endereço será exibido na tela, onde o usuário informará o CEP (Código de Endereçamento Postal) e o sistema irá consumir a API ViaCEP, que é gratuita e permite o preenchimento automático dos outros campos do endereço, exceto número e complemento. Os campos preenchidos automaticamente no projeto são: logradouro, bairro, cidade e UF (Unidade Federativa).

A Figura 4 apresenta a *view* do projeto Django que faz o consumo da API em questão: o código define a função "obter_endereco_por_cep" que recebe um CEP como argumento, realiza uma requisição à API "ViaCEP" para obter informações de endereço associadas ao CEP fornecido (linha 4), e, se a resposta for bem sucedida (linha 5), retorna os detalhes do endereço em formato JSON. Em caso de falha na requisição (linha 7), a função retorna uma resposta HTTP com a mensagem "Não foi possível encontrar o CEP digitado."

```
1 def obter_endereco_por_cep(cep):
2     url = f'https://viacep.com.br/ws/{cep}/json/'
3     try:
4         response = requests.get(url)
5         if response.status_code == 200:
6             return response.json()
7     except requests.exceptions.RequestException:
8         return None
```

Figura 4. View de consumo da API ViaCEP

O comerciante somente poderá cadastrar seus produtos após realizar o cadastro da loja. Caso não realize, terá funcionalidades bloqueadas (acesso à edição e cadastro de seus produtos).

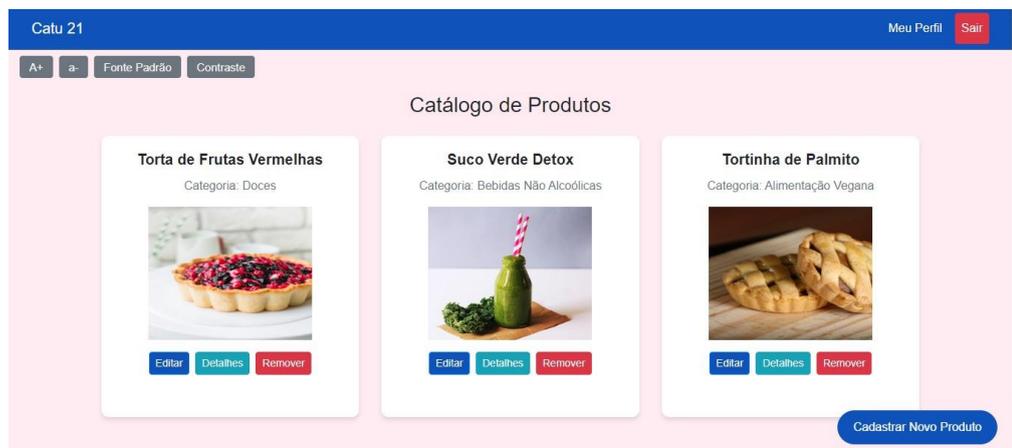


Figura 5. Tela Inicial do Comerciante

Com o cadastro completo, o comerciante poderá cadastrar e visualizar seus produtos. Na tela inicial são exibidos os produtos do comerciante (Figura 5). O botão 'Detalhes' abre a janela de um modal (Figura 6), que exibe o nome do produto, a data de cadastro do produto no sistema e sua descrição. O botão 'Editar' leva à página de edição do produto (Figura 7). Em todas as telas há botões de acessibilidade dispostos na área superior esquerda. Tais botões podem ser utilizados para alterar contraste e para aumentar, diminuir ou retomar a fonte padrão das páginas.

O botão 'Remover' abre um modal (Figura 8) para perguntar se o usuário confirma a remoção do produto. Se sim, será removido do banco de dados. Se não, o usuário clica em 'Voltar' e é redirecionado para a tela inicial. Já o botão 'Cadastrar Novo Produto' direciona o usuário para a tela de cadastro de novos produtos, onde todos os campos deverão ser preenchidos, conforme Figura 9.

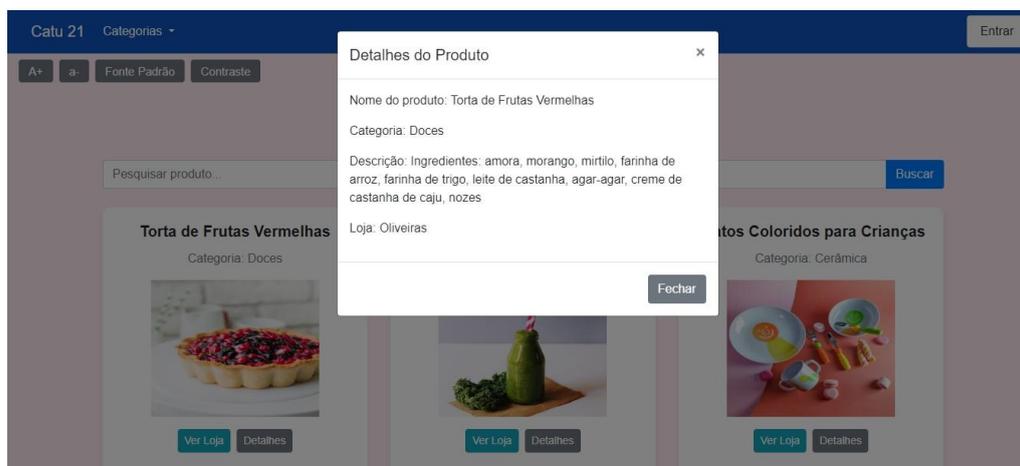


Figura 6. Modal de Detalhes do Produto

Nome do Produto:

Torta de Palmito

Descrição:

Deliciosa torta caseira de Palmito.
 Ingredientes: Palmito, farinha de arroz, alho, sal, ovos, ervilha, água

Categoria:

Alimentação Saudável

Imagem atual:



Selecione uma nova imagem:

Escolher arquivo Nenhum arquivo escolhido

Remover Salvar Voltar

Figura 7. Tela de Edição do Produto

Para cadastrar a categoria do produto, é necessário escolher um dos itens de uma lista que é referenciada no banco de dados. As categorias dos produtos são definidas pelo administrador do sistema (Figura 10). Os usuários não podem criar novas categorias pois limitar ações dos usuários torna o sistema mais padronizado.

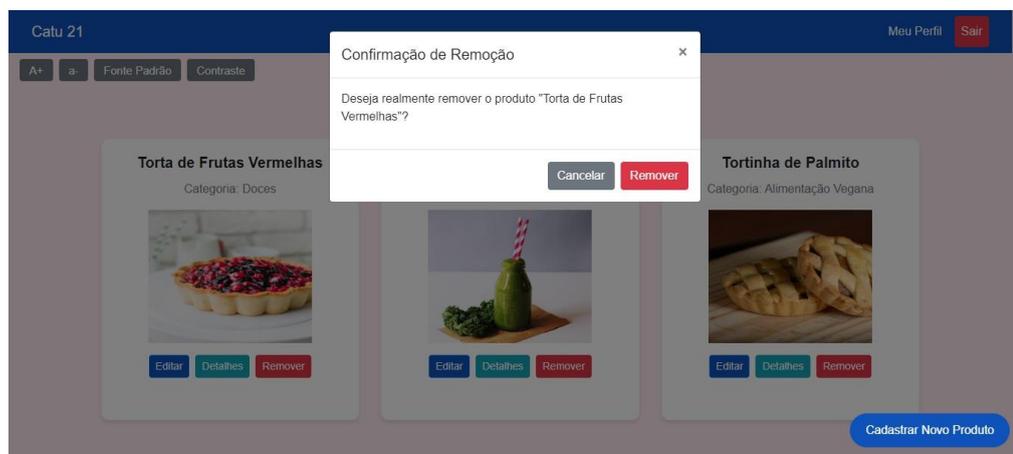


Figura 8. Modal de Remoção do Produto

Catu 21

Meu Perfil Sair

A+ a- Fonte Padrão Contraste

Imagem: Escolher arquivo suco_refrescante.jpg

Nome: Suco Refrescante

Descrição: Delicioso suco refrescante

Categoria: Bebidas Não Alcoólicas

Cadastrar Voltar

CATU 21 foi desenvolvido por Gabriela Lima dos Anjos em 2023 para o Trabalho de Conclusão de Curso - IFSP/HTO

Figura 9. Tela de Cadastro de Novo Produto

Adicionar categoria

Nome: Produtos de Limpeza

Descrição: Esses produtos são formulados com ingredientes ativos e aditivos que possuem propriedades de limpeza, desinfecção e desodorização. Entre os componentes mais comuns encontrados em produtos de limpeza, estão detergentes, alvejantes, solventes, agentes desinfetantes e fragrâncias.

SALVAR Salvar e adicionar outro(a) Salvar e continuar editando

Figura 10. Modal de Adição de Categoria

Como recurso de segurança, foram utilizadas restrições nas páginas dos usuários cadastrados, onde apenas os cadastrados terão acesso aos recursos de cadastro e edição. E, cada usuário conectado terá acesso à visualização e edição apenas de seus produtos. Entretanto, ao efetuar *logout* no sistema e acessar a página inicial do Catu 21, poderão visualizar todos produtos e lojas cadastradas.

A função "home_comerciante"(11) verifica a existência de uma sessão de usuário ativa (linha 2), obtém o usuário associado (linha 3), procura pela loja vinculada a esse usuário. Se não encontrar (linhas 4-7) redireciona para a tela de cadastro de loja. Em seguida, se encontrar a loja associada, recupera os produtos do usuário (linha 9) e renderiza um template 'home.html', passando os produtos e o ID do usuário logado como contexto (linha 11), redirecionando-o para a tela inicial da página do comerciante (5). Caso não haja sessão de usuário ativa, a função direciona para a página de login com um status de notificação (linha 13) de que o usuário não tem permissão de acesso.

O administrador do sistema pode excluir cadastros e trocar a senha dos usuários, caso necessário. Um exemplo da área do administrador pode ser visualizado na Figura 12.

```

1 def home_comerciante(request):
2     if request.session.get('usuario'):
3         usuario = Usuario.objects.get(id=request.session['usuario'])
4         try:
5             Loja.objects.get(usuario=usuario)
6         except Loja.DoesNotExist:
7             return redirect('cadastrar_loja')
8
9         productos = Productos.objects.filter(usuario=usuario)
10
11     return render(request, 'home.html', {'productos': productos,
12                                         'usuario_logado': request.session.get('usuario')})
13 else:
14     return redirect('/auth/login/?status=2')

```

Figura 11. View da página inicial do comerciante

Todas as páginas são responsivas, conforme Figura 13, que representa a visualização da página por uma tela de *smartphone*.

Dessa forma, tanto o *layout* quanto o conteúdo da página se adaptam automaticamente ao tamanho da tela, otimizando o uso do sistema para os usuários que utilizam diferentes dispositivos para acessá-lo.

Os usuários visitantes possuem outra visualização das páginas, onde todos os produtos disponíveis de todas as lojas são exibidos na mesma tela (vide Figura 14). Além disso, podem realizar a busca personalizada dos pedidos através do botão ‘Buscar’ referente ao digitado na barra de pesquisa da página e filtrar os produtos por suas categorias correspondentes, através do botão ‘Categorias’.



Figura 12. Tela do Administrador do Django



Figura 13. Visão da tela de um *smartphone*

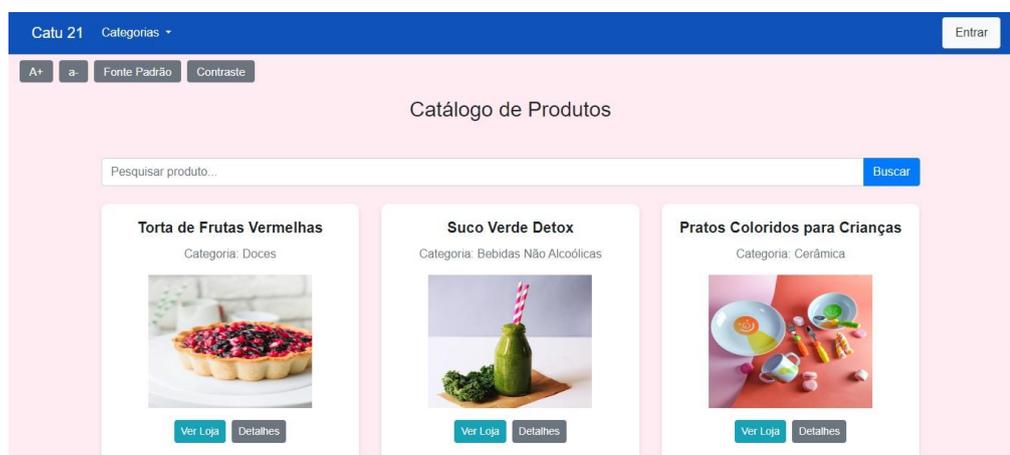


Figura 14. Página visitantes

Quando o visitante encontra um produto de seu interesse, pode selecionar o botão 'Ver Loja', onde será redirecionado à página da loja referente ao produto buscado, conforme Figura 15. Em tal página são exibidas as informações de contato do comerciante e seus produtos cadastrados no catálogo.

Se a loja possuir redes sociais, um *link* será gerado e o visitante poderá clicar e ser redirecionado a respectiva página da loja (que, neste caso, não faz mais parte do sistema Catu 21, pois este apenas faz o redirecionamento aos visitantes às páginas dos comerciantes), assim como poderá ser redirecionado ao endereço da loja diretamente no Google Maps, se a loja existir fisicamente.

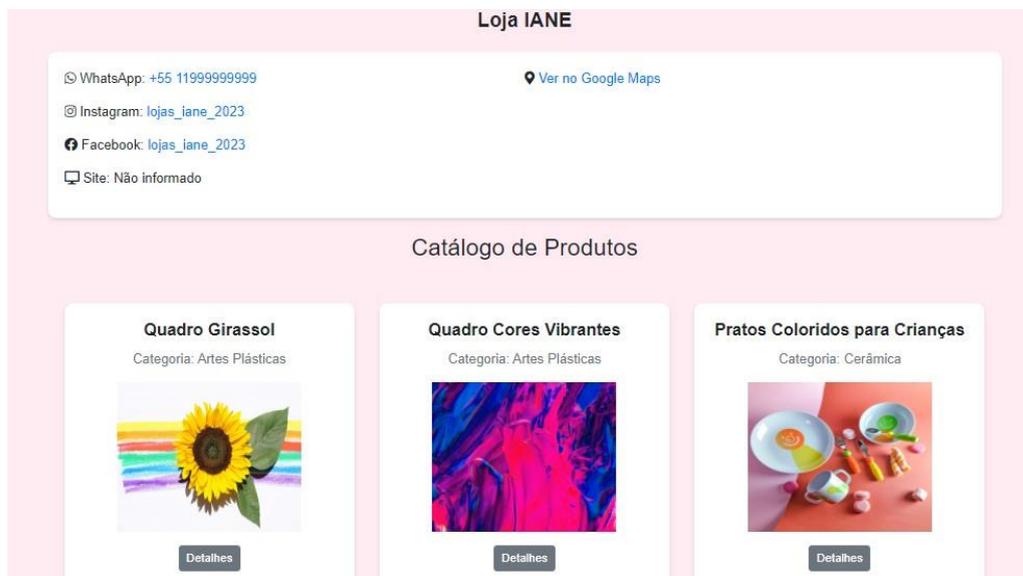


Figura 15. Página da Loja - Visualização dos visitantes

Com isso, o usuário pode ter a visualização das telas conforme necessidade. Se for comerciante, terá cadastro pessoal e poderá editar o catálogo continuamente. Se for visitante, terá a visualização completa de todos os produtos cadastrados no sistema como um todo.

5. Conclusão

O desenvolvimento do projeto que visava a criação de um *website*, incluindo o *back-end* e o *front-end*, de divulgação de produtos de pequenos empreendedores foi realizado conforme o objetivo, criando a oportunidade de conectar microempreendedores a potenciais consumidores. Os comerciantes precisam apenas ter cadastro no sistema, pois assim é possível divulgar suas lojas e produtos. Os usuários visitantes, ou seja, os consumidores, não têm a opção de realizar o cadastro porque o catálogo é aberto ao público e ainda não há um sistema de avaliação dos produtos e loja.

O projeto foi testado em nível de desenvolvimento. Durante o desenvolvimento do sistema *web*, surgiram desafios na implementação do *framework* Django, pois a aluna não possuía o conhecimento da ferramenta, mas ao longo do projeto pôde se familiarizar gradualmente com a ferramenta, aprimorando sua habilidade com o Django.

Para trabalhos futuros existem melhorias que podem ser implementadas no sistema para aprimorar a experiência dos usuários. Isso inclui a implementação de um filtro baseado na geolocalização dos usuários, fornecendo mais opções de refino das pesquisas de busca de acordo com seus interesses. Ao disponibilizar essa opção, os visitantes da plataforma teriam maior conveniência de encontrar as lojas mais próximas de sua localização atual, o que poderia potencializar a conexão entre comerciantes e consumidores locais e engajar a plataforma. Também é possível iniciar a comercialização dos produtos dentro do Catu 21 e criar uma área de cadastro para os visitantes, com

possibilidade dos usuários compartilharem suas avaliações da loja e dos produtos. E, por fim, é possível implementar a funcionalidade de suspensão do catálogo e cadastro do comerciante, caso este não acesse o sistema em um tempo superior ao configurável no sistema (três meses, por exemplo).

O desenvolvimento do trabalho contemplou o conhecimento adquirido em disciplinas do curso superior de Análise e Desenvolvimento de Sistemas, principalmente em relação às disciplinas: Interação Humano Computador, através dos botões de acessibilidade, permitindo que o sistema seja utilizado por um público maior; Desenvolvimento *Web*, pela integração com o banco de dados, consumo de API externa e pelo uso de JavaScript, HTML e CSS, criando uma interface adaptável a diferentes telas; Análise Orientada a Objetos, por onde foi possível elaborar diagramas e fluxos do sistema; Banco de Dados, pela integração e modelagem do banco.

Em síntese, a realização do trabalho permitiu a incorporação de conhecimentos adquiridos no curso e ao longo do projeto. Com isso, foi possível fazer a análise, criar requisições, desenvolver o projeto utilizando as linguagens de programação e, então, implementar o sistema localmente. Ademais, novos conhecimentos, que não foram abordados durante o curso, foram adquiridos: Python, Django e Bootstrap, expandindo as áreas de conhecimento no âmbito do desenvolvimento de sistemas.

Referências

- Amazon Web Services (2023) O que é Python? <https://aws.amazon.com/pt/what-is/python/>, acesso: 18 abr. 2023.
- Bootstrap (2023) Comece com o bootstrap, <https://getbootstrap.com/docs/5.3/getting-started/introduction/>, 10 mai. 2023.
- Django Software Foundation (2023) Django, <https://www.djangoproject.com/>, 18 abr. 2023.
- IBM (2021) Casos de uso, <https://www.ibm.com/docs/pt-br/elms/elm/6.0?topic=cases-use>, 26 abr. 2023.
- IBM (2023) Desenvolvimento web com Django, <https://www.ibm.com/br-pt/topics/django>, 20 abr. 2023.
- IFSP (2023) Trabalho de conclusão de curso (TCC), <https://hto.ifsp.edu.br/portal/index.php/ensino/cursos/ads/ads-tcc>, ago. 2023.
- Konig, F. and da Silva, R. B. (2022) Desenvolvimento de uma plataforma virtual para negociação de produtos ou serviços, <https://repositorio.ifsc.edu.br/handle/123456789/2022>, 22 mar. 2023.
- Lima, B. O. C. (2023) Aplicação web para automação de atendimento de bares com Python e Django, *UNIFACS*, <https://repositorio.animaeducacao.com.br/handle/ANIMA/33558>, 20 abr. 2023.
- Miro (2023) O que é diagrama entidade-relacionamento, <https://miro.com/pt/diagrama/o-que-e-diagrama-entidade-relacionamento/>, 12 jun. 2023.

- Mozilla (2021) Getting started with html, https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics, 18 abr. 2023.
- Mozilla (2023) What is CSS? Mozilla Developer Network, https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/What_is_CSS, 18 abr. 2023.
- Pressman, R. and Maxim, B. (2016). *Engenharia de Software-8ª Edição*. McGraw Hill Brasil.
- Pressman, R. S. (2006). *Engenharia de Software*. McGrawHill, Nova York, EUA, 6 edition.
- Red Hat (2023) O que é API, <https://www.redhat.com/pt-br/topics/api/what-are-application-programming-interfaces>, 24 ago. 2023.
- Santos, E. F. D. (2021) Sistema web para divulgação e venda de produtos de uma confeitaria, Universidade Tecnológica Federal do Paraná, <https://repositorio.utfpr.edu.br/jspui/bitstream/1/28491/1/sistemawebconfeitaria.pdf>, 22 mar. 2023.
- Schiarolli, L. and Tomazeti, D. M. (2022) Easyproposal: Aplicação web para elaboração de propostas de vendas de máquinas de produção, <https://hto.ifsp.edu.br/cloud/s/ZttizB9ycxibfqx>, 23 ago. 2023.
- IFSP (2023) Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP) – Campus Hortolândia, <https://hto.ifsp.edu.br/cloud/s/ZttizB9ycxibfqx>, 23 ago. 2023.
- Serasa (2021) Quais as maiores dificuldades das micro empresas e pequenas empresas e como superá-las?, <https://serasa.certificadodigital.com.br/blog/certificado-digital/>, 25. nov. 2022.
- Silva, M. S. (2020) JavaScript-Guia do Programador: Guia completo das funcionalidades de linguagem JavaScript, Novatec Editora, https://books.google.com.br/books/about/JavaScript_Guia_do_Programador.html?hl=pt-BR&id=BB9WDQAAQBAJ&redir_esc=y, 12. jun. 2023.
- Sommerville, I. (2019). *Engenharia De Software*. PEARSON BRASIL.

A. Anexos

A.1. Detalhamento dos Casos de Uso

Tabela 6: Fluxos dos Casos de Uso

RF01	Pesquisar produtos
1	O usuário clica na barra de pesquisa e digita o nome do produto que deseja procurar
2	O sistema exibe os resultados da pesquisa caso existam. Se o produto pesquisado não existir no sistema, a mensagem ‘Nenhum item correspondente encontrado’ é exibida.
RF02	Detalhar Produtos
1	O usuário clica no botão ‘Detalhes’ do produto exibido no catálogo.

2	O sistema exibe um modal com mais detalhes sobre o produto (nome, categoria, descrição, nome da loja).
RF03	Redirecionar ao Comerciante
1	O usuário clica no botão de contato ('Ver loja') do comerciante em um produto do mesmo.
2	O sistema redireciona o usuário para a página de contato do comerciante, onde são exibidas informações da loja e <i>links</i> relevantes.
RF04	Cadastrar Perfil
1	O comerciante acessa a página de cadastro.
2	O comerciante preenche as informações solicitadas (e-mail, senha).
3	O sistema verifica se todas as informações foram preenchidas corretamente.
4	O sistema salva os dados do usuário no banco de dados.
5	O sistema redireciona o usuário para a tela de <i>login</i> .
RF05	Efetuar Login
1	O usuário acessa a página de login.
2	O usuário preenche o e-mail e a senha cadastrados previamente.
3	O sistema verifica se o e-mail e a senha são válidos.
4	O sistema inicia a sessão do usuário.
5	Caso o comerciante tenha realizado o cadastro da loja, o sistema redireciona o usuário para a página principal do <i>site</i> .
6	Caso o comerciante ainda não tenha realizado o cadastro da loja, o sistema redireciona o usuário para a página de cadastro de loja.
RF06	Cadastrar Loja
1	O sistema exibe automaticamente a tela de cadastro de loja, onde um formulário é exibido.
2	O comerciante preenche o formulário com as informações obrigatórias e opcionais de cadastro de loja. O usuário é obrigado a preencher o formulário para ter acesso a outras funcionalidades do sistema.
3	Usuário confirma preenchimento dos campos através do botão de confirmação.
RF07	Gerenciar Perfil
1	O usuário acessa a página de gerenciamento de perfil através do botão 'Meu Perfil'.
2	O sistema exibe as informações de perfil do usuário e de sua loja.
3	O usuário pode alterar as informações editáveis.
4	Usuário confirma as alterações realizadas.
5	O sistema salva as informações alteradas no banco de dados.
RF08	Cadastrar Produto
1	Na tela inicial, o comerciante clica no botão 'Cadastrar Novo Produto'.
2	O comerciante preenche o formulário de cadastro de produto exibido pelo sistema.
3	Comerciante confirma o cadastro.
4	O sistema guarda as informações no banco de dados.
RF09	Editar Produto
1	O vendedor clica no botão 'Editar' referente ao produto que deseja fazer alterações.

2	O sistema exibe as informações cadastradas no banco de dados sobre o produto.
3	O comerciante altera as informações de seu interesse.
4	O vendedor confirma a alteração.
5	O sistema retorna automaticamente para a tela inicial com as informações atualizadas.
RF10	Remover Produto
1	O comerciante seleciona o botão 'Remover' correspondente ao produto que deseja remover.
2	O sistema exibe mensagem de confirmação de remoção.
3	O comerciante confirma a remoção.
4	O sistema remove o produto do catálogo do comerciante.
RF11	Gerenciar Usuário
1	O administrador acessa a lista de usuários cadastrados no sistema.
2	O sistema exibe a lista de usuários registrados.
3	O administrador seleciona um usuário específico para gerenciar suas informações.
4	O administrador salva as alterações.
RF12	Gerenciar Categoria do Produto
1	O administrador acessa a lista de categorias de produtos cadastradas no sistema.
2	O administrador decide editar ou adicionar uma categoria.
3	Caso o administrador escolha editar, deve selecionar a categoria específica e clicar no botão para gerenciar suas informações.
4	Caso o administrador escolha cadastrar nova categoria, deve selecionar o botão para adicionar categoria.
5	Administrador salva as alterações.

Documento Digitalizado Restrito

Artigo Final - Gabriela Lima dos Anjos

Assunto: Artigo Final - Gabriela Lima dos Anjos
Assinado por: Rodolfo Oliveira
Tipo do Documento: Anexo
Situação: Finalizado
Nível de Acesso: Restrito
Hipótese Legal: Informação Pessoal - dados pessoais e dados pessoais sensíveis (Art. 31 da Lei nº 12.527/2011)
Tipo do Conferência: Documento Digital

Documento assinado eletronicamente por:

- **Rodolfo Francisco de Oliveira, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 31/08/2023 15:10:30.

Este documento foi armazenado no SUAP em 31/08/2023. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifsp.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1425813

Código de Autenticação: 99eff285c1

