

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

APARECIDO SOUZA DA CRUZ JUNIOR

**LEVANTAMENTO DE REQUISITOS E
DESENVOLVIMENTO DA PLATAFORMA DIGITAL
GREENFOODS**

CURITIBA

2023

APARECIDO SOUZA DA CRUZ JUNIOR

**LEVANTAMENTO DE REQUISITOS E
DESENVOLVIMENTO DA PLATAFORMA DIGITAL
GREENFOODS**

Requirements gathering and development of the GreenFoods digital platform

Trabalho de conclusão de curso de Pós Graduação apresentado como requisito para obtenção do título de Especialista em Tecnologia Python para Negócios do curso Especialização em Tecnologia Python para Negócios da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Yuri Kaszubowski Lopes.

Coorientador: Pedro Henrique de Alencar Machado.

CURITIBA

2023



[4.0 Internacional](https://creativecommons.org/licenses/by-nc/4.0/)

Esta licença permite compartilhamento do trabalho, mesmo para fins comerciais, sem a possibilidade de alterá-lo, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

APARECIDO SOUZA DA CRUZ JUNIOR

**LEVANTAMENTO DE REQUISITOS E
DESENVOLVIMENTO DA PLATAFORMA DIGITAL
GREENFOODS**

Trabalho de Conclusão de Curso de Especialização
apresentado como requisito para obtenção do título
de Especialista em Especialização em Tecnologia
Python para Negócios da Universidade Tecnológica
Federal do Paraná (UTFPR).

Data de aprovação: 15 / Abril / 2023

Yuri Kaszubowski Lopes
Doutorado
Universidade do Estado de Santa Catarina

Rodolfo Adamshuk Silva
Doutorado
Universidade Tecnológica Federal do Paraná (UTFPR)

André Roberto Ortoncelli
Doutorado
Universidade Tecnológica Federal do Paraná (UTFPR)

Pedro Henrique de Alencar Machado
Mestrado
Universidade Tecnológica Federal do Paraná (UTFPR)

CURITIBA

2023

Dedico este trabalho à minha família.

AGRADECIMENTOS

Agradeço ao meu orientador Prof. Dr. Yuri e coorientador Prof. Pedro, pela sabedoria com que me guiou nesta trajetória.

Ao meu amigo Daniel que me auxiliou durante todo decorrer do curso e que compartilhou comigo várias horas de estudo.

Quero deixar registrado também, o reconhecimento aos meus familiares, minha esposa Camilla, que teve muita paciência e sempre incentivou durante toda minha experiência da especialização.

Em especial, a mim mesmo, pelo foco, determinação, vontade de evoluir como pessoa e profissionalmente.

Enfim, a todos os que por algum motivo contribuíram para a realização desse trabalho.

RESUMO

Conforme dados, cerca de 30% dos alimentos produzidos no país são desperdiçados, o equivalente a cerca de 46 milhões de toneladas de alimentos por ano. Estes alimentos tanto deveriam chegar a uma parcela carente da população, quanto representam um capital investido que é perdido. Vários projetos atacam o primeiro deste ponto, mas há a necessidade de atacar este problema pelo outro ponto de vista; i.e., que os alimentos desperdiçados são capital investido perdido e que recuperar parte deste valor pode ao mesmo tempo proporcionar um serviço atrativo a uma parcela da classe média. Nesse sentido, esse trabalho tem como objetivo o desenvolvimento de um MVP na plataforma de Marketplace que possibilite a venda de alimentos que de outra maneira seriam desperdiçados. O projeto será uma ferramenta Web baseada no framework Django nativa da linguagem Python com banco de dados Sqlite. Para o layout das páginas será utilizado a tecnologia Bootstrap e JavaScript. Pela plataforma será possível a venda de produtos comercializados por restaurantes, que ao invés de serem descartados podem ser vendidos a um preço acessível, assim, evitando o desperdício desses alimentos e ao mesmo tempo recuperando o capital investido nestes produtos perecíveis.

Palavras-chave: Python. Django. Desperdício.

ABSTRACT

According to data, about 30% of food produced in the country is wasted, equivalent to about 46 million tons of food per year. These foods should both reach a needy part of the population and represent invested capital that is lost. Several projects tackle the first from this point of view, but there is a need to tackle this problem from the other point of view; i.e., that wasted food is lost invested capital and that recovering part of this value can at the same time provide an attractive service to a portion of the middle class. In this sense, this work aims to develop an MVP on the Marketplace platform that enables the sale of food that would otherwise be wasted. The project will be a Web tool based on the Django framework native to the Python language with a Sqlite database. For the layout of the pages, Bootstrap and JavaScript technology will be used. Through the platform, it will be possible to sell products marketed by restaurants, which instead of being discarded can be sold at an affordable price, thus avoiding the waste of these foods and at the same time recovering the capital invested in these perishable products.

Keywords: Python. Django. Waste.

LISTA DE ILUSTRAÇÕES

Figura 1 – Modelo mental MVC.	17
Figura 2 – Fluxo MTV Django.	18
Figura 3 – Fluxo MTV Django.	19
Figura 4 – Diagrama UML.	24
Figura 5 – Models.py.	25
Figura 6 – Urls.py.	26
Figura 7 – Views.py.	27
Figura 8 – Login.	28
Figura 9 – Corpo e rodapé.	29
Figura 10 – Perfil anunciante.	30
Figura 11 – Finalizar pedido.	31

LISTA DE TABELAS

Tabela 1 – Lista de requisitos funcionais.	23
Tabela 2 – Lista de requisitos não funcionais.	23

LISTA DE ABREVIATURAS E SIGLAS

GUI	Graphical User Interface	16
MVP	Minimum Viable Product	12
FAO	Food and Agriculture Organization	14
ONU	Organização das Nações Unidas	14
S.O	Sistemas Operacionais	16
MVC	Model View Control	17
MTV	Model Template View	17
HTML	HyperText Markup Language	18
CSS	Cascading Style Sheets	18
IDE	Integrated Development Environment	19
W3C	World Wide Web Consortium	20
SGBD	Sistema Gerenciador de Banco de Dados	21

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Tema	12
1.2	Objetivo	12
1.2.1	Objetivo específico	13
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	Desperdício de comida e aplicativos de entrega	14
2.1.1	FDA (Food Delivery App)	14
2.1.3	Phenix	15
2.1.4	Zomato	15
2.1.5	Ifood	15
2.1.6	Deliveroo	15
2.2	Python	16
2.3	Framework Django	17
2.4	Bootstrap	19
2.5	Html	20
2.6	Css	20
2.7	Javascript	21
2.8	Sqlite	21
3	MATERIAIS E MÉTODOS	22
3.1	Objetivo	22
3.2	Requisitos	22
3.2.1	Requisitos funcionais	22
3.2.2	Requisitos não funcionais	23
3.2.3	Deliveroo	24
3.3	Estrutura da aplicação	24
3.3.1	Models.py	25
3.3.2	Url.py	25
3.3.3	Views.py	26
4	RESULTADO	28
5	CONCLUSÃO	32
6	TRABALHOS FUTUROS	33
6.1	Limitações	33

1 INTRODUÇÃO

A nível global é possível perceber o aumento crescente da população. Baseado nesse cenário, o desperdício de comida próprias para o consumo atingiu proporções imensuráveis. Esse quadro é agravado pelo aumento da escassez de recursos naturais utilizados para produção de alimentos e por conta dessa realidade vemos uma produção exagerada de resíduos (MEDEIROS,2021).

Neste contexto surge a oportunidade de desenvolver soluções capazes de viabilizar a comercialização de alimentos que, de outra forma, seriam desperdiçados. No mercado internacional, uma solução já praticada é uma plataforma digital para a venda online de alimentos sobressalentes. No entanto, o mercado nacional ainda possui carência de tal solução.

Este projeto foca, portanto, na criação de uma solução nacional para a questão de comercialização de alimentos sobressalentes, além da criação de um Produto Mínimo Viável (do Inglês, Minimum Viable Product (MVP)).

Esse projeto está relacionado com trabalho desenvolvido pelo estudante Daniel Gil com o tema "Construção de um modelo teórico de negócio para viabilização de um produto de marketplace - GreenFoods", com o foco no desenvolvimento do modelo de negócio que irá viabilizar o desenvolvimento da aplicação chamada GreenFoods, uma aplicação web do tipo Marketplace de alimentos (PLANELLES, 2023).

Este trabalho irá focar no levantamento de requisitos e desenvolvimento da plataforma digital deste MVP.

1.1 TEMA

Observou-se que dentro do tema de sustentabilidade que o desperdício de comida é uma questão importante a ser solucionada. Durante a pandemia uma alternativa encontrada foi pedir alimentos via aplicativos, o que alavancou as entregas via delivery.

Analisando os acontecimentos atuais e o crescimento dos aplicativos de alimentos, emergiu a oportunidade de gerar uma ferramenta que irá auxiliar nessa demanda que vem crescendo a cada ano.

1.2 OBJETIVO

Esse trabalho tem como objetivo de desenvolver uma aplicação web no formato Marketplace, onde os anunciantes irão disponibilizar produtos próprios para consumo que de outra maneira seriam desperdiçados.

1.2.1 Objetivo específico

- Gerar uma solução que monetize a produção excedente de comida dos restaurantes;
- Diminuir o desperdício de alimentos; e
- Conectar anunciantes e consumidores que estão abertos a uma nova proposta de consumo sustentável.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 DESPERDÍCIO DE COMIDA E APLICATIVOS DE ENTREGA

O desperdício alimentar ocorre em qualquer umas das fases da cadeia alimentar, tanto na produção do alimento quanto no consumo (BATISTA; ABRANJA, 2021).

Cerca de 14% dos alimentos do mundo (avaliados em US\$ 400 bilhões por ano) continuam sendo perdidos depois de colhidos antes de chegarem às lojas. E outros 17% de nossos alimentos acabam sendo desperdiçados no varejo e pelos consumidores, principalmente nas residências. Segundo estimativas da (Food and Agriculture Organization (FAO)), os alimentos perdidos e desperdiçados poderiam alimentar 1,26 bilhão de famintos todos os anos (AUTOR, 2022b).

No Brasil, cerca de 30% dos alimentos produzidos no país são desperdiçados, o equivalente a cerca de 46 milhões de toneladas de alimentos por ano (AUTOR, 2023a). Em valores, essa quantia de comida jogada fora é cerca de R\$ 61,3 bilhões por ano e segundo a Organização das Nações Unidas (ONU) o Brasil ocupa 10º posição no ranking do desperdício.

De acordo com a nossa fonte, os alimentos são desperdiçados devido a compras excessivas, falta de planejamento de refeições e descuido com a conservação.

2.1.1 FDA (Food Delivery App)

Nessa seção são descritas algumas ferramentas (aplicativos) de delivery.

2.1.2 TooGoodToGo

A empresa TooGoodToGo foi fundada por Brian Christensen, Thomas Bjorn Momsen, Stian Olesen, Klaus Bagge Pedersen, e Adam Sigbrand na cidade de Copenhaga, na Dinamarca.

Sua ideia é de resgatar a comida que seria jogada fora por restaurantes, cafés, padarias, supermercados, hotéis, entre outros (CARVALHO; CARDOSO, 2021).

Esse modelo de negócio, segundo (CARVALHO; CARDOSO, 2021), permite às empresas lucrar com alimentos e refeições que seriam descartados. Simultaneamente, atrair um novo segmento de clientes que pretendem experimentar novos conceitos de negócio.

Dentro da plataforma existe aqueles que buscam comprar produtos a preços mais baixos ou que apenas querem contribuir para um mundo mais sustentável. De qualquer forma, esses dois pensamentos impulsionam os comércios locais.

A startup dinamarquesa Too Good To Go propicia a seus usuários a possibilidade de adquirirem refeições a preços acessíveis, colaborando para uma economia rotativa, reduzindo o

desperdício de alimentos com uma pegada ecológica (BATISTA; ABRANJA, 2021).

2.1.3 Phenix

A Phenix é um aplicativo Frances fundada por Jean Moreau. A plataforma está disponível em mais de 5 países. Segundo o site oficial, foram salvos mais de 150.000 refeições por dia e evitou-se o desperdício de cerca 60 toneladas de resíduos diariamente (AUTOR, 2023).

A ferramenta permite aos seus utilizadores comprar caixas dos produtos dos saldos excedentes de produção de restaurantes, padarias, pastelarias, mercearias, frutarias ou floristas, a preço abaixo do mercado (BATISTA; ABRANJA, 2021).

2.1.4 Zomato

Zomato fundada sob o nome 'Foodiebay' em 2008 por Deepinder Goyal, com sede na Índia é uma plataforma online de busca e entrega de restaurantes. O App fornece detalhes como avaliações e classificações autônomas. Nomeada primeiramente como Foodiebay, sofreu uma alteração para Zomato em novembro de 2010 com intuito de aumentar seu alcance entre as pessoas (GUPTA, 2019).

2.1.5 Ifood

Ifood é uma empresa brasileira que começou em 2011, anteriormente chamada de Disk Cook e fundada pelos sócios Patrick Sigrist, Eduardo Baer, Guilherme Bonifácio e Felipe Fioravante.

Atualmente o Ifood atua no mercado da América Latina e segundo (AUTOR, 2021) o aplicativo conta com mais de 1,5 milhões de downloads por mês. A plataforma opera em mais de 1200 cidades do país, tem mais de 270 mil restaurantes parceiros e alcançou o patamar de 60 milhões de pedidos entregues todo mês.

2.1.6 Deliveroo

A Deliveroo é uma empresa de entrega de comida que serve como intermediária entre clientes e restaurantes. A Deliveroo foi fundada por Will Shu em 2013 ao lado de Greg Orlowski.

Devido ao crescimento da plataforma no Reino Unido, as operações foram ampliadas para Europa Ocidental, Hong Kong, Austrália, Kuwait e Emirados Árabes Unidos.

A Deliveroo tem 160.000 restaurantes em sua plataforma, um aumento de 14,2% em relação aos números de 2020. E está disponível 800 cidades em 12 países e em quase todas as cidades do Reino Unido (MANSUR, 2023).

2.2 PYTHON

A linguagem Python foi concebida no ano de 1980 por Guido van Rossum no Instituto de Pesquisa Nacional para Matemática e Ciência da Computação (CWI). Projetada para substituir a linguagem de programação ABC, Python é uma linguagem de programação de alto nível, destacando a legibilidade do código, oferece suporte a vários paradigmas (multiparadigma) de programação incluindo orientação a objetos, estruturada, programação funcional e estilo processuais. Os interpretadores da linguagem estão disponíveis em diversos sistemas operacionais possibilitando a execução do script em uma variedade de Sistemas Operacionais (S.O). Uma importante característica do Python é uma resolução dinâmica de nomes, que liga nomes de métodos e variáveis durante a execução do programa. Python possui funções de mapear, reduzir e filtrar; compreensões para listas, dicionários, e conjuntos gerador expressões (SCHOOL, 2021), (TULCHAK; MARCHUK, 2016).

Python transformou-se, uma linguagem simples e robusta, aplicada principalmente em áreas como Data Science, Data Analytics, Inteligência Artificial, Deep Learning, Desenvolvimento Web e aplicações de forma geral. Além disso, porta uma gama de bibliotecas que permitem o reuso de funcionalidades no desenvolvimento de software. Em 2020 contava com 246.267 bibliotecas disponíveis. (SANTIAGO et al., 2020).

Internamente, Python possui dois tipos de módulos padrões: `functools` e `itertools`. Além disso, o Python possui diversas vantagens sobre outras linguagens de programação (TULCHAK; MARCHUK, 2016) incluindo:

- Sintaxe limpa;
- Programas de tolerância, característica da maioria das linguagens interpretadas;
- A distribuição normal tem muitos módulos úteis, incluindo o módulo para desenvolvimento de Graphical User Interface (GUI);
- A utilização do Python em modo interativo, muito útil para experimentar e resolver problemas simples;
- Adequado para resolver problemas matemáticos. Uma maneira de trabalhar com números complexos, opera com números inteiros de tamanho arbitrário. A caixa de diálogo pode ser usada como uma poderosa calculadora; e
- Python armazena uma variedade de bibliotecas padrão. Essa é sua especialidade, ele fornece ferramentas adequadas para diversos tipos de tarefas, por isso é descrito como pilhas inclusas”.

2.3 FRAMEWORK DJANGO

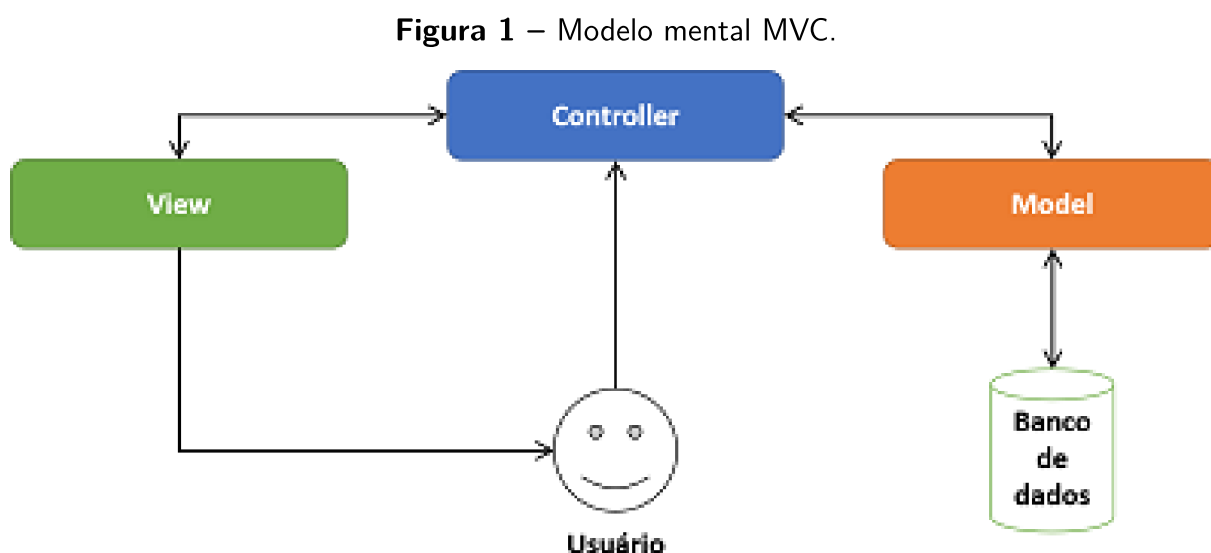
Django é um framework web Python que permite o desenvolvimento rápido, seguro e de fácil manutenção. Sua concepção foi planejada para diminuir a sobrecarga de trabalho cuidando da grande parte do desenvolvimento web, permitindo que o foco seja no desenvolvimento do aplicativo. O Django é gratuito e de código aberto (CONTRIBUTORS, 2022).

O framework foi criado no ano de 2005 por um grupo de programadores do Lawrence Journal World com intuito de agilizar o desenvolvimento web de projetos dentro do jornal. O Django tornou-se um aliado dos desenvolvedores por resolver problemas tradicionais e possibilitar a reutilização de código. Por conta disso, ficou muito famoso por fornecer soluções, como por exemplo, autenticação de usuário, administração de conteúdos entre outros (SANTIAGO et al., 2020).

Ao contrário de outros frameworks Web que utilizam a arquitetura Model View Control (MVC), o Django aplica livremente a arquitetura Model Template View (MTV) e suporta oficialmente os bancos de dados PostgreSQL, MySQL, Oracle e SQLite.

A arquitetura MVC separa os principais componentes e auxilia no desenvolvimento. Ela é responsável por armazenar os dados, manipular, aplicar as funções com entrada e saída dos dados disponíveis para o usuário.

O componente Model é objeto da aplicação, a View a interface disponível para o usuário e o Controller é responsável por ligar o model a View fazendo repasse dos models (LEMOS et al., 2013).



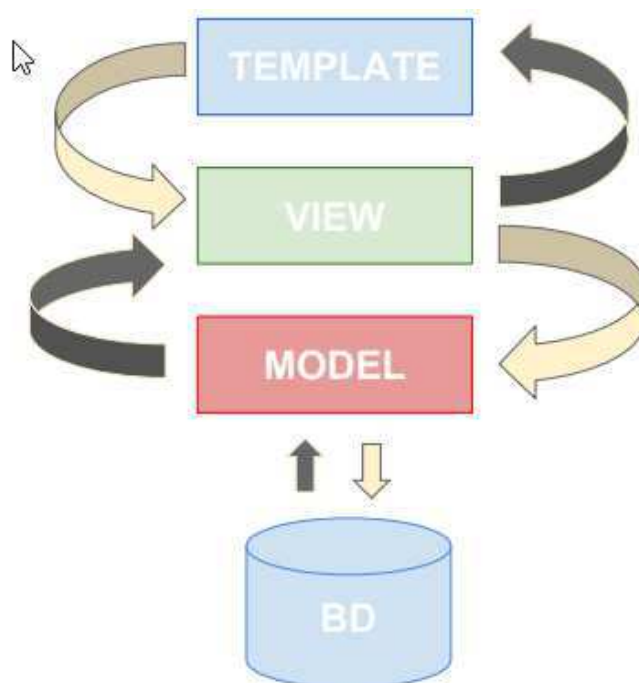
Fonte: (GUEDES, 2020).

A Figura 1 mostra a facilidade do MVC e o fluxo da informação entre as camadas partindo da solicitação do usuário.

O modelo de arquitetura MVT é padrão de projeto do Django baseada nessas 3 camadas conectadas entre si. As camadas tem as seguintes funções (OLIVEIRA; LUZ et al., 2020):

- O Model é a mesma definição do MVC, no qual funciona com uma ligação para manusear os dados entre o banco de dados e a camada View;
- A View é responsável por formatar os dados que estão circulando entre o banco dados através do Model; e
- Já o Template é responsável pela visualização do usuário final, ele contém arquivos HyperText Markup Language (HTML), Cascading Style Sheets (CSS) e JavaScript.

Figura 2 – Fluxo MTV Django.

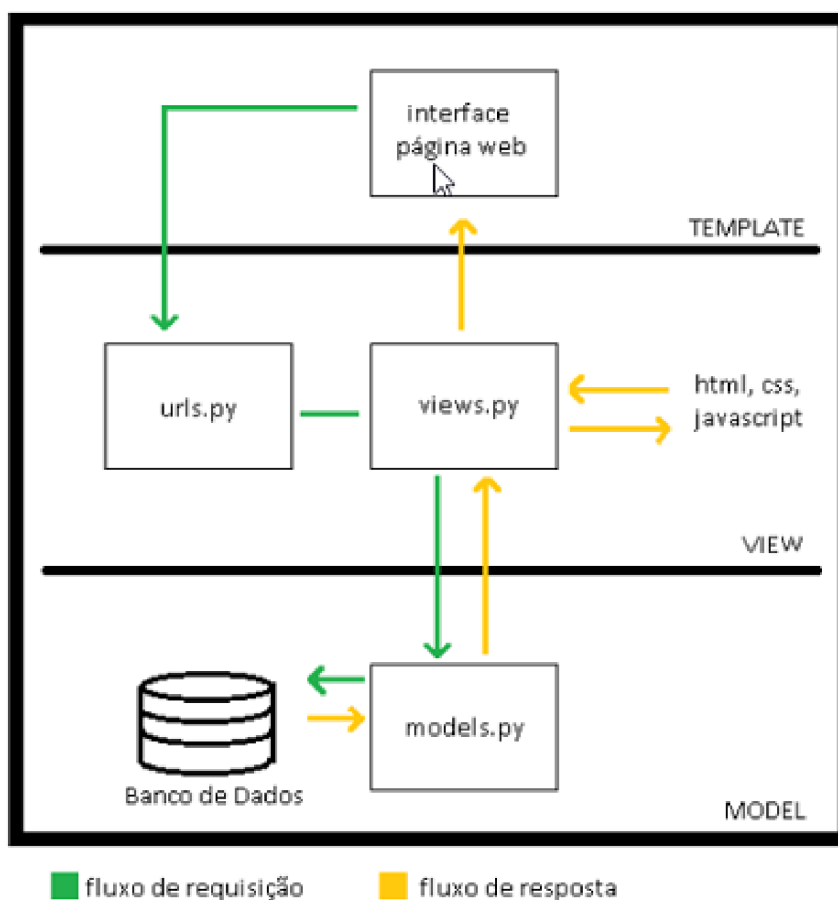


Fonte: OLIVEIRA, LUZ et al. (2020).

A Figura 2 dá uma ideia de como o MTV é utilizado no padrão do Framework Django.

Em sua essência, os arquivos `models.py`, `views.py`, `urls.py` fornecem funções para criar, recuperar, atualizar e excluir os dados. `View.py` é responsável em transmitir o conteúdo do site modelo HTML. A função do arquivo `urls.py` é configurar qual arquivo de visualização é chamado. Com esta estrutura e funções de arquivo, a alteração do código não requer a edição em vários lugares dentro do projeto, com isso, permitindo uma economia de tempo (GHIMIRE, 2020).

Figura 3 – Fluxo MTV Django.



Fonte: Ghimire (2020).

A Figura 3 mostra como funciona o fluxo da informação, desde a solicitação saindo da página web (Template), passando pelos arquivos internos do Django camadas (View/ Model), até a chega no banco de dados.

2.4 BOOTSTRAP

O Bootstrap é um framework utilizado para desenvolvimento de aplicações front-end e de código aberto (SANTIAGO et al., 2020). Além de conter HTML e modelos de design fundamentado em CSS para tipografia, formulários, botões, tabelas, navegação, modais, carrosséis de imagens e muitos outros (GAIKWAD; ADKAR, 2019), o Bootstrap conta também com outros componentes embutidos, como, JavaScript e ainda uma variedade de plug-ins tornando a sua utilização em aplicações mais amigáveis e responsivas (SANTIAGO et al., 2020).

Várias Integrated Development Environment (IDE) suportam o Bootstrap. Isso propicia uma facilidade na codificação para os programadores. Outra vantagem é a extensibilidade usando JavaScript e o suporte a todos os navegadores. Ele ainda conta com suporte interno para plug-ins jQuery e uma API JavaScript. Além de tudo pode ser utilizado com todas

as tecnologias ou linguagem back-end, como, ASP.NET, PHP, Ruby on Rails ou Python (SANTIAGO et al., 2020).

2.5 HTML

A Linguagem HTML é a linguagem de marcação padrão para criar páginas web e aplicações web (GAIKWAD; ADKAR, 2019). Em sua quinta revisão, lançada em 2011 pelo World Wide Web Consortium (W3C), foi apoiado pelos principais navegadores da web (Chrome, IE, Firefox, Opera, Safari) atualmente (SHAH, 2015).

Html é uma linguagem estrutural descritiva, que não requer nenhum compilador, simplesmente um bom editor de texto de programação, muitos dos quais os downloads é gratuito e de código aberto. Junto com estilo CSS e JavaScript formam uma tríade de tecnologias fundamentais para a World Wide Web (GAIKWAD; ADKAR, 2019).

A HTML5 disponibiliza vários recursos novos que fornecem não apenas suporte a rich media, mas também, aprimoram o suporte para aplicativos da web podendo interagir com os usuários, seus dados locais e servidores, de forma mais fácil e eficaz do que anteriormente. O HTML5 foi desenvolvido com base nos seguintes padrões (SHAH, 2015):

- Novos recursos devem ser baseados em HTML, CSS, DOM e JavaScript;
- A necessidade de plugins externos (como Flash) precisa ser reduzida;
- O tratamento de erros deve ser mais fácil do que nas versões anteriores;
- Os scripts devem ser substituídos por mais marcações;
- O HTML5 deve ser independente do dispositivo; e
- O processo de desenvolvimento deve ser visível ao público.

2.6 CSS

A W3C define o CSS como uma linguagem que descreve a apresentação de páginas da Web, cores, fontes e layouts (SHAH, 2015). É possível controlar a cor do texto, o estilo de fontes, o espaçamento entre os parágrafos como as colunas são dimensionadas e dispostas. E quais imagens ou cores de fundo são usados, designs de layout, variações na exibição para diferentes dispositivos e tamanhos de tela, bem como uma variedade de outros efeitos (GAIKWAD; ADKAR, 2019).

Existem duas regras de formatação de estilo para o CSS, o estilo embutido e o estilo externo. No estilo externo a regra de estilo fica em arquivo diferente, relacionando por uma tag exclusiva no documento HTML. A cascata que parte do CSS, refere-se a como as regras são aplicadas aos elementos HTML, assim, o documento é estilizado hierarquicamente. Portanto,

é responsabilidade do CSS encontrar a precedência das regras de estilo, conseqüentemente, acabando por estabelecer um efeito cascata (SHAH, 2015).

2.7 JAVASCRIPT

JavaScript foi inicialmente projetado e implementado em maio de 1995 no Netscape por Brendan Eich. Nesse ano, a Web e os navegadores da Web eram novas tecnologias que estouravam no mundo, e o Netscape Communications Corporation liderava o desenvolvimento de navegadores da Web (WIRFS-BROCK; EICH, 2020).

JavaScript é muito usado não apenas nas aplicações web do lado cliente, mas também passou a fazer parte de aplicativos móveis, desktop e aplicativos de servidor. Muitos programas JavaScript são projetos complexos de softwares, incluindo por exemplo, sites altamente interativos, jogos online, ambientes de desenvolvimento integrado, clientes de e-mail e processadores de texto (ANDREASEN et al., 2017).

As principais características do Javascript são tipos e operadores, objetos centrais e métodos. Sua sintaxe é derivada das linguagens Java e C, por isso, muitos das estruturas dessas linguagens também se aplicam ao JavaScript. A linguagem não contém classes; em vez disso, a funcionalidade é realizada por meio de protótipos de objetos (SHAH, 2015).

O outro principal diferença é que as funções são objetos, portanto, as funções são fornecidas com a capacidade de manter o código executável e eles podem ser passados como qualquer outro objeto.

2.8 SQLITE

Criado por D. Richard Hipp no ano de 2000, para cumprir o propósito de 'nenhuma administração necessária' para operar um programa. Sua versão 1.0 foi lançada com o gerenciador de banco de dados GNU. No ano de 2011 a Hipp anunciou a adição da interface UNQL ao banco de dados SQLite para desenvolver o UNQLite (banco de dados orientado a documentos) (S, 2023).

O SQLite é uma biblioteca que implementa um mecanismo de banco de dados SQL transacional, independente, sem servidor, com configuração zero. O código para SQLite é de domínio público e, portanto, é gratuito para uso para qualquer finalidade, comercial ou privada (AUTOR, 2022).

Conforme descrito por (NUNES, 2015), o SQLite funciona como um "Mini Sistema Gerenciador de Banco de Dados (SGBD)", desenvolvida totalmente em C padrão (ANSI), capaz de criar, ler e escrever em um arquivo alocado no disco do dispositivo móvel. Este arquivo detém a extensão db e é capaz de armazenar uma grande quantidade de tabelas.

3 MATERIAIS E MÉTODOS

O capítulo será apresentado as ferramentas utilizadas para desenvolvimento da aplicação Web.

3.1 OBJETIVO

O objetivo da aplicação é integrar estabelecimentos que produzem um excedente de alimentos durante o dia, aos clientes que desejam consumir esses produtos a um baixo custo.

O estabelecimento poderá comercializar vários tipos de produtos por um valor inferior, eles devem estar aptos para consumo imediato, método no qual evita descarte no lixo.

Com isso, os clientes poderão consumir e comprar refeições/alimentos mais baratos dos restaurantes, bem inferior aos preços comercializados normalmente.

Com a aplicação, o anunciante irá realizar um cadastro no qual poderá criar, visualizar, editar/atualizar e excluir seus produtos. Além disso, poderá incluir o nome, descrição, preço e uma foto do alimento disponibilizado para venda.

Já o cliente após seu cadastro, utilizará a aplicação para selecionar os produtos, a partir da sua escolha, gerar um pedido com os itens. Será possível selecionar a forma de pagamento e visualizar a distância atual até o estabelecimento.

3.2 REQUISITOS

O conceito de requisito mudou bastante durante os anos, anteriormente, os requisitos eram relacionados apenas a funções do sistema.

Hoje sua definição foi ampliada e conforme (HIGOR, 2013) é uma condição necessária para satisfazer um objetivo, ou seja, um requisito é um aspecto que o sistema proposto deve fazer ou uma restrição no desenvolvimento do sistema.

3.2.1 Requisitos funcionais

Os requisitos funcionais referem-se ao que o sistema deve fazer. Com base nessa definição, foram levantadas as seguintes funções da aplicação Web definidas pela Tabela 1:

Tabela 1 – Lista de requisitos funcionais.

Código	Nome	Descrição	Categoria
RF001	Cadastrar comprador/anunciante	A aplicação Web deverá permitir o cadastro de usuário (comprador/anunciante)	Obrigatório
RF002	Cadastrar produto	A aplicação Web deverá permitir o cadastro de produtos para o usuário anunciante	Obrigatório
RF003	Cadastrar pedido	A aplicação Web deverá permitir a geração de um pedidos através do carinhos de compras	Obrigatório
RF004	Confirmar Pedido	A aplicação Web deverá permitir o aceite do pedido por parte do anunciante	Obrigatório

Fonte: Próprio autor

3.2.2 Requisitos não funcionais

Os requisitos não funcionais referem-se aos critérios que qualificam os requisitos funcionais.

A Tabela 2 descreve a lista dos requisitos não funcionais da aplicação Web.

Tabela 2 – Lista de requisitos não funcionais.

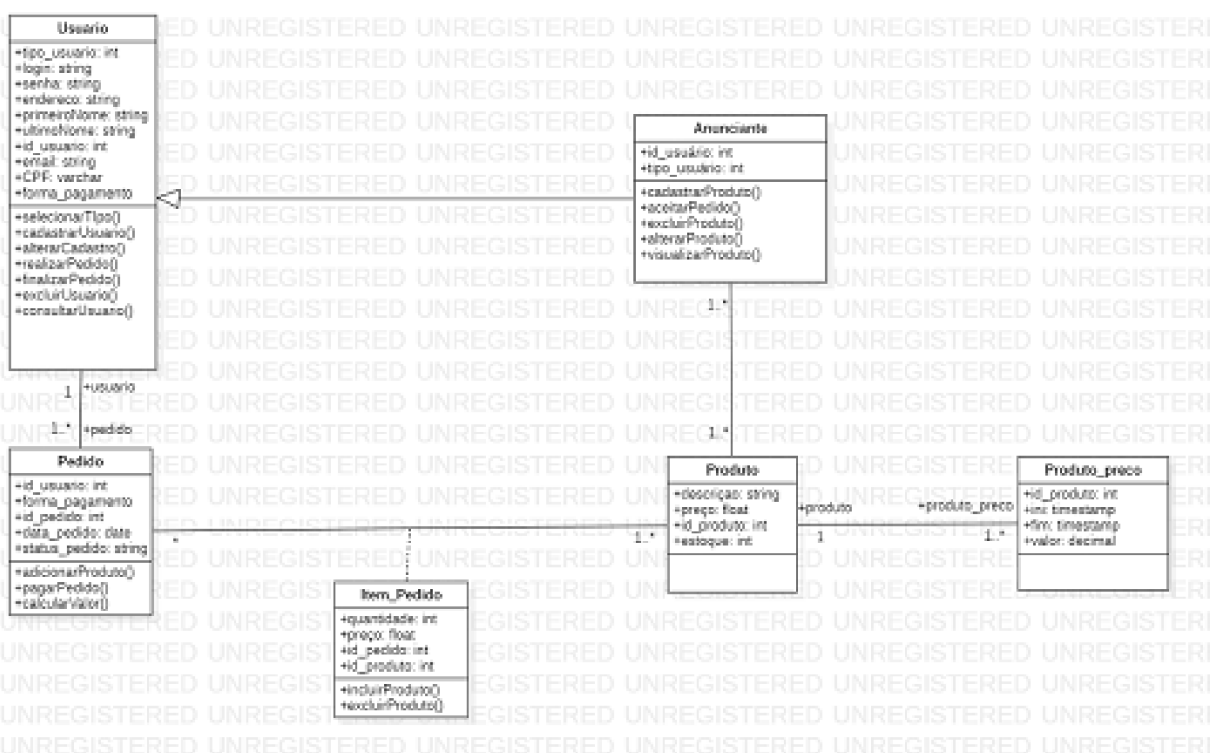
Código	Nome	Descrição	Categoria
RNF01	Disponibilidade	Aplicação web deve estar sempre acessível e disponível na internet	Obrigatório
RNF02	Back-end	O aplicação Web deve ser desenvolvida com o framework Django	Obrigatório
RNF03	Browsers	A aplicação Web deve ser compatível com todos os navegadores web	Obrigatório
RNF04	Usabilidade	A aplicação Web deve ser de fácil usabilidade	Obrigatório

Fonte: Próprio autor

3.2.3 Diagrama de classes

O diagrama de classes auxilia na representação formal das classes do sistema e seus relacionamentos. A Figura 4 apresenta o diagrama de classes que especifica as classes do sistema, o relacionamento entre elas com os seus respectivos atributos e métodos.

Figura 4 – Diagrama UML.



Fonte: Próprio autor.

3.3 ESTRUTURA DA APLICAÇÃO

O projeto foi desenvolvido com IDE PyCharm 2021.3 Community Edition e Python na versão 3.10.4. Para melhor estruturar o projeto, foi criado ambiente virtual com o comando "python3 -m venv tutorial-env" a fim de separar o projeto em um único lugar e não interferir em outros projetos. Para o gerenciamento e controle das versões incluímos o projeto na plataforma GitHub, onde a cada atualização local é realizado um commit com as alterações.

Para a elaboração da aplicação web, foi utilizado o framework web Django em sua versão 3.2.8.

3.3.1 Models.py

O arquivo Models.py mapeia os relacionamentos entre as tabelas e estrutura o banco de dados. Todos os dados essenciais são armazenado conforme o tipo declarado na criação dos fields.

Figura 5 – Models.py.



```
9 class State(models.Model):
10     name = models.CharField(max_length=20, verbose_name="Estado")
11     uf___ = models.CharField(max_length=2)
12
13     def __str__(self):
14         return self.name
15
16 class City(models.Model):
17     state = models.ForeignKey(State, on_delete=models.CASCADE)
18     name___ = models.CharField(max_length=40)
19
20     def __str__(self):
21         return self.name
22
23 class Neighborhood(models.Model):
24     city = models.ForeignKey(City, on_delete=models.CASCADE)
25     name = models.CharField(max_length=40)
26
27     def __str__(self):
28         return self.name
29
30 class Cep(models.Model):
31     city___ = models.ForeignKey(City, on_delete=models.CASCADE)
32     cep___ = models.CharField(max_length=10)
33     citywide = models.BooleanField(default=False)
```

Fonte: Próprio autor.

Conforme mostra a Figura 5 as classes State, City, Neighborhood, CEP dentro do arquivo Models.py, compõem o cadastro do usuário e as tabelas onde serão armazenados os dados do endereço.

O tipo ForeignKey demonstra que o atributo é um relacionamento muito para muitos e requer a classe a qual o modelo está relacionado. No exemplo temos a classe City que se relaciona com a classe State.

3.3.2 Urls.py

A URL é uma string usada para identificar um endereço da web ou recurso da web.

O arquivo urls.py contém as URLs que serão utilizadas no projeto. Ele armazena o mapeamento de URLs (urlpatterns), redirecionando requisições HTTP para as funções

localizadas no arquivo View.py com base na URL solicitada (SANTIAGO et al., 2020).

Figura 6 – Urls.py.



```
7
8 urlpatterns = [
9     path('signup/', SignUpView.as_view(), name="signup"),
10
11     path("", TemplateView.as_view(template_name="home.html"), name="home"),
12
13     path("card/", ListCardProd.as_view(), name="list_card_prod"),
14     path("card/<int:pk>", DetailCardProd.as_view(), name="detail_card_prod"),
15
16     path("restaurant", ListRestaurant.as_view(), name="list_restaurant"),
17     path("restaurant/<int:pk>", DetailRestaurant.as_view(), name="detail_restaurant"),
18
19
20     path("login_teste/", TemplateView.as_view(template_name="login_teste.html"), name="login_teste"),
21
22 ]
```

Fonte: Próprio autor.

O Django usa algo chamado URLconf (configuração de URL). URLconf é um conjunto de padrões que o Django tentará combinar com a URL solicitada para encontrar a visualização correta. Na Figura 6 vemos a aplicação desse padrão no projeto.

Dentro do método `path()` o parâmetro `"card/<int:pk>"` é correspondente a URL da View. Os sinais de menor e maior (`<`, `>`) são usados para definir partes de uma URL que serão capturadas e passadas para a view correspondente como argumentos nomeados.

O segundo parâmetro `"DetailCardProd.as_view()"` é a View correspondente à URL informada. Já o terceiro e último parâmetro `"detail_card_prod"` é o nome para chamada no template.

3.3.3 Views.py

O arquivo Views.py é utilizado pelo Django para armazenar classes e métodos, onde as requisições são enviadas conforme o comportamento de uma determinada URL.

Figura 7 – Views.py.

```
class ListCardProd(ListView):  
    model = Product  
    template_name = 'card.html'  
  
class DetailCardProd(DetailView):  
    template_name = 'detail_prod.html'  
    model = Product  
  
class ListRestaurant(ListView):  
    template_name = 'restaurant.html'  
    model = Restaurant
```

Fonte: Próprio autor.

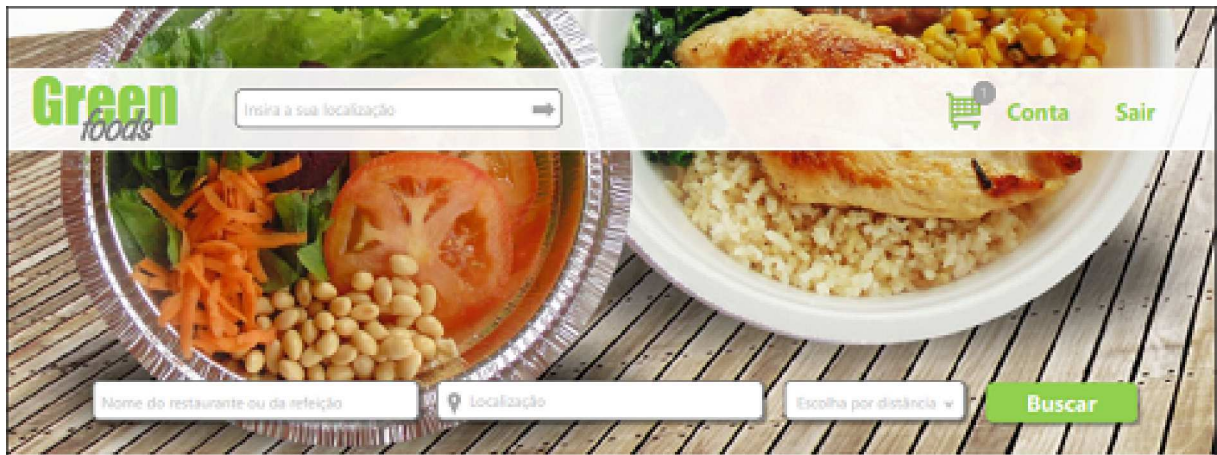
A Figura 7 apresenta algumas visualizações exibidas através de uma lista genérica baseada em classe. No exemplo a classe "ListCardProd" mostra a lista dos cards de produtos onde consta os itens do anunciante.

A exibição genérica consulta o banco de dados para obter todos os registros do modelo especificado (Product) e renderiza o modelo localizado.

4 RESULTADOS

Os templates do projeto foram desenvolvidos através de uma prototipagem onde se aplica um layout de fácil navegação para os usuários. A Figura 8 mostra a navbar da página Home.

Figura 8 – Login.



Fonte: Próprio autor.

Vejamos que é possível através do navbar pesquisar o restaurante por localização, ter acesso ao carrinho onde através dele os produtos selecionados ficam armazenados e o link para o login.

No corpo da página Home será exibido os anúncios mais recentes através de cards e uma lista de restaurantes mais próximos do endereço cadastrado na plataforma. A Figura 9 mostra o protótipo do template.

Figura 9 – Corpo e rodapé.

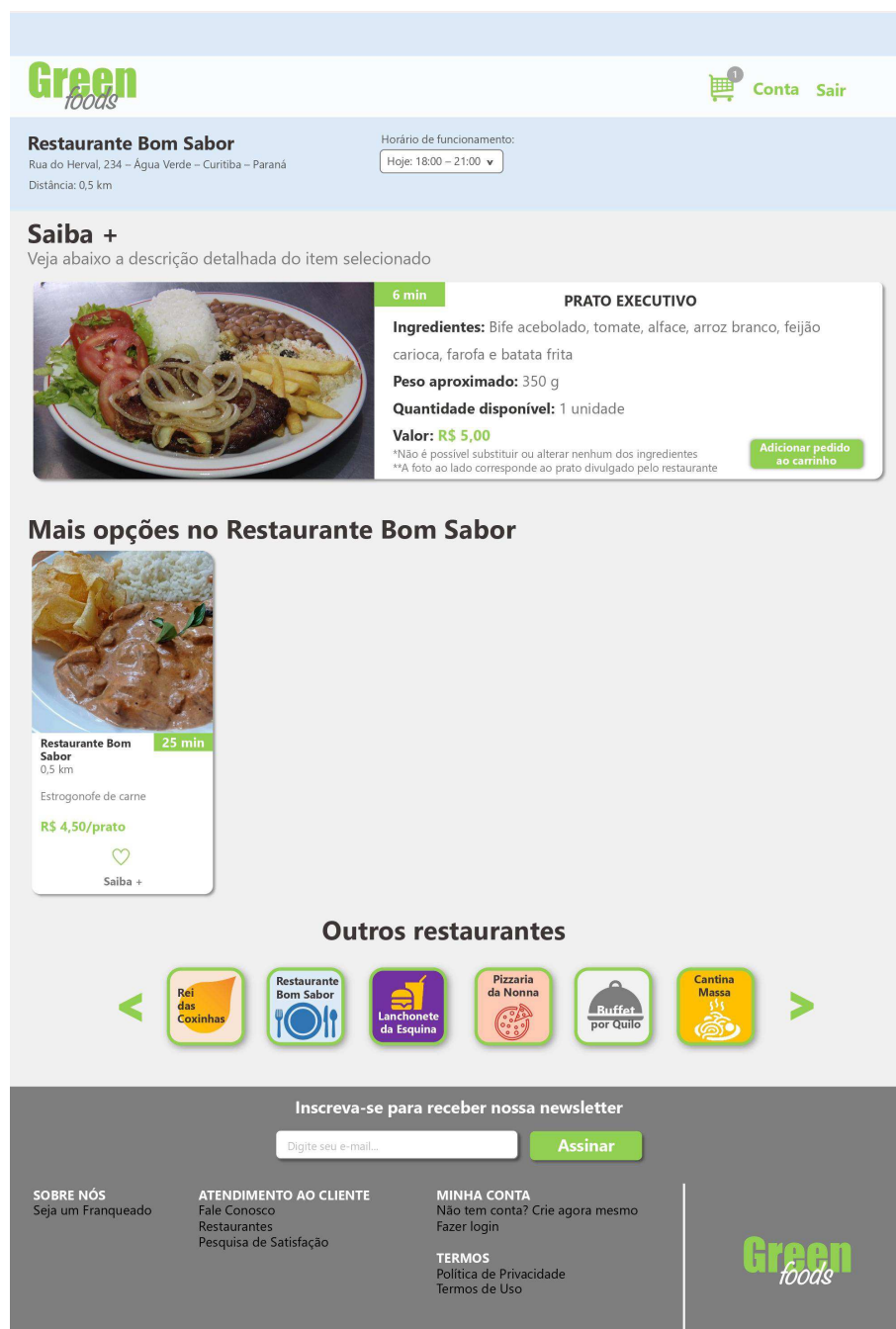


Fonte: Próprio autor.

Outra prototipagem desenvolvida foi a simulação de um acesso ao perfil de um dos anunciantes.

É possível visualizar a imagem do produto anunciado e ao lado vemos o card com as informações de ingredientes, peso, quantidade e valor. Ainda dentro do card está disponível um botão que adiciona o produto ao carrinho.

Figura 10 – Perfil anunciante.

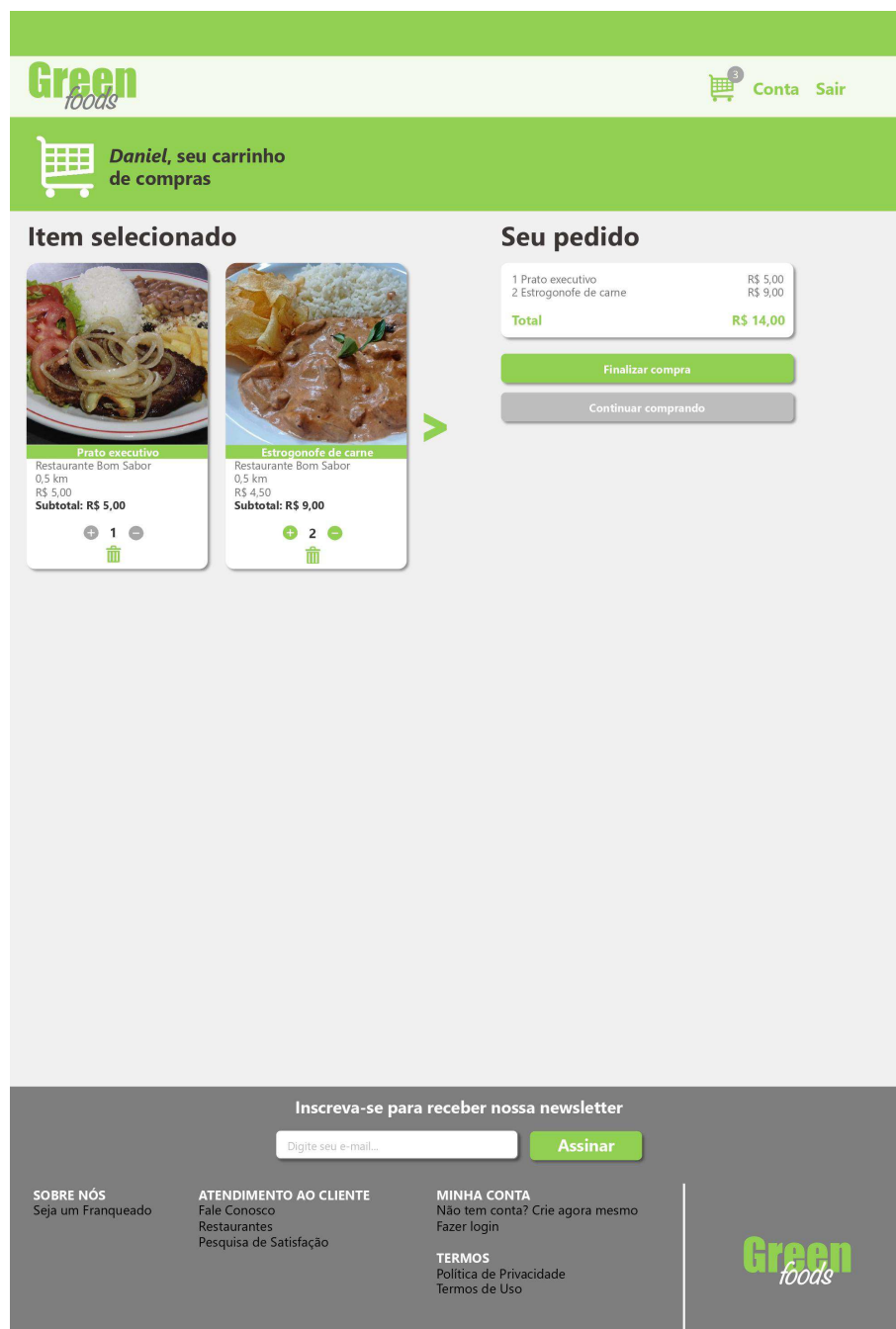


Fonte: Próprio autor.

A Figura 10 ainda sugere outras opções de produtos do mesmo anunciante e uma lista de outros perfis disponíveis.

E por último, uma demonstração do protótipo de como será o Template de finalização do pedido, esse é o processo no qual o usuário irá efetivar sua compra. Em cada card vemos os produtos selecionados e sua foto.

Figura 11 – Finalizar pedido.



Fonte: Próprio autor.

Na Figura 11 mostra o card que irá apresentar as informações da quantidade de cada item e logo após o valor total. No card ao lado vemos a imagem do produto e logo abaixo é exibido o nome do restaurante, sua distância do endereço cadastrado e valor. Nesse template será possível continuar selecionando itens e incluindo ao carrinho ou finalizar a compra.

O projeto está com sua estrutura de arquivos desenvolvida e regras de negócio de desenvolvimento. O Front-end é uma das suas maiores dores e por esse motivo realizamos a prototipagem das telas.

5 CONCLUSÃO

O tema do desperdício de alimentos deve ser amplamente discutido e novos métodos devem ser desenvolvidos para minimizar, ou até mesmo, sanar essa dor da humanidade. Com surgimento de novas tecnologias podemos lutar contra esse problema de maneira mais eficaz e a conscientização tanto dos consumidores quanto dos produtores deve alcançar um patamar elevado sobre o tema.

Podemos validar o resultado de outras plataformas que incentiva o consumo consciente dos alimentos que poderiam ser descartados. O projeto GreenFoods tem a mesma linha de pensamento, unindo o coletivo a favor do planeta.

Desta forma, o aplicativo proposto minimiza o desperdício de alimentos ao mesmo tempo que possibilita recuperar parte do capital investido nestes produtos. É importante destacar que, ao monetizar o redirecionamento dos alimentos que seriam desperdiçados estamos incentivando a cultura de reaproveitamento e gerando valor sobre isso. Assim, o impacto do presente projeto abrange duas dimensões: social e econômico.

Para a execução técnica do projeto, utilizou-se Python e o Framework Django. Conclui-se que, o framework Django é uma ferramenta poderosa que vai auxiliar no desenvolvimento do MVP proposto nesse trabalho, por conta da sua facilidade e economia de tempo durante o desenvolvimento. E ainda, unido com as demais tecnologias apresentadas que facilitaram no decorrer do desenvolvimento.

6 TRABALHOS FUTUROS

Para a versão 2.0 da plataforma podemos evoluir e desenvolver os seguintes pontos:

- Implementar o pagamento dos pedidos diretamente na plataforma.
- Expandir a rede de cobertura para demais cidade do Paraná.
- Promover iniciativas sustentáveis perante a sociedade para mobilizar esforços e levante discussão sobre o tema do desperdício de alimentos.
- Desenvolver um aplicativo para as plataformas Android e IOS.

6.1 LIMITAÇÕES

- Encontrar uma resistência na aderência da plataforma por parte dos usuários. - Realizar parcerias com estabelecimentos.
- Encontrar mão de obra especializada com conhecimento das linguagens de programação utilizada para o desenvolvimento do projeto.
- Encontrar uma hospedagem na internet para a plataforma com recursos necessários a um preço acessível para o início do projeto.

REFERÊNCIAS

- ANDREASEN, Esben et al. A survey of dynamic analysis and test generation for javascript. **ACM Computing Surveys (CSUR)**, ACM New York, NY, USA, v. 50, n. 5, p. 1–36, 2017. páginas 21
- AUTOR, Sem. **Ifood, muito mais que um pedido**. 2021. Url<https://canaltech.com.br/empresa/ifood/> en:Acesso em 25, Março de 2023. páginas 15
- _____. **About SQLite**. 2022. Disponível em: <<https://sqlite.org/about.html>:Acessoem20, Março de 2023>. páginas 21
- _____. **Tackling food loss and waste: A triple win opportunity**. 2022. Disponível em: <<https://www.fao.org/newsroom/detail/FAO-UNEP-agriculture-environment-food-loss-waste-day-2022/en>:Acessoem22, Março de 2023>. páginas 14
- _____. **Brasil é o 10º país que mais desperdiça alimentos no mundo**. 2023. Disponível em: <<https://mercadoconsumo.com.br/26/01/2023/sustentabilidade/brasil-e-o-10o-pais-que-mais-desperdica-alimentos-no-mundo/>>. páginas 14
- _____. **Quem somos nós**. 2023. Disponível em: <<https://www.wearephenix.com/pt-pt/phenix-combate-o-desperdicio/que-somos/en>:Acessoem22, Março de 2023>. páginas 15
- BATISTA, Diogo Plácido Pinto; ABRANJA, Nuno. Desperdício alimentar: Um problema do passado, do presente e do futuro. **Tourism and Hospitality International Journal**, v. 17, n. 1, 2021. páginas 14, 15
- CARVALHO, Luísa Cagica; CARDOSO, Carlos. Quando a sustentabilidade resolve problemas ambientais: o caso da aplicação too good to go. **Casos de marketing público y no lucrativo-Casos de Marketing Público e Não Lucrativo**, v. 8, p. 69–79, 2021. páginas 14
- CONTRIBUTORS, MDN. **Aprendendo desenvolvimento web**. 2022. Url<https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/Django/Introduction>. páginas 17
- GAIKWAD, Suraj Shahu; ADKAR, PRATIBHA. A review paper on bootstrap framework. **IRE Journals**, v. 2, n. 10, p. 349–351, 2019. páginas 19, 20
- GHIMIRE, Devndra. Comparative study on python web frameworks: Flask and django. 2020. páginas 7, 18, 19
- GUEDES, Marylene. **O que é MVC?** 2020. Url<https://www.treinaweb.com.br/blog/o-que-e-mvc/> en:Acesso em 22, Março de 2023. páginas 7, 17
- GUPTA, Mitali. A study on impact of online food delivery app on restaurant business special reference to zomato and swiggy. **International Journal of Research and Analytical Reviews**, v. 6, n. 1, p. 889–893, 2019. páginas 15
- HIGOR. **Introdução a Requisitos de Software**. 2013. Url<https://www.devmedia.com.br/introducao-a-requisitos-de-software/29580> en:Acesso em 29, Março de 2023. páginas 22

- LEMOS, Maxmilian Ferreira de et al. Aplicabilidade da arquitetura mvc em uma aplicação web (webapps). **RE3C-Revista Eletrônica Científica de Ciência da Computação**, v. 8, n. 1, 2013. páginas 17
- MANSUR, IQBAL. **Estatísticas de receita e uso da Deliveroo (2023)**. 2023. Url<https://www.businessofapps.com/data/deliveroo-statistics/> en:Acesso em 25, Março de 2023. páginas 15
- MEDEIROS, Francisco Manuel Fernandes. O impacto ambiental do desperdício alimentar: visões, percepções e o caso too good to go. 2021. páginas 12
- NUNES, Fernando. Avaliação de técnicas e mecanismos para entrada e saída de informações em dispositivos móveis. 2015. páginas 21
- OLIVEIRA, Lucas Carmichael dos Santos de; LUZ, Larissa Pavarini da et al. Estruturação de dados semânticos com base em um modelo de repositório digital. 2020. páginas 7, 18
- PLANELLES, Daniel Gil. Construção de um modelo teórico de negócio para viabilização de um produto de marketplace - greenfoods. 2023. páginas 12
- S, Ravikiran A. **What is SQLite? Everything You Need to Know**. 2023. Disponível em: <<https://www.simplilearn.com/tutorials/sql-tutorial/what-is-sqlite?tag=sqlite>: Acessoem20,Marçode2023>. páginas 21
- SANTIAGO, Cynthia Pinheiro et al. Desenvolvimento de sistemas web orientado a reuso com python, django e bootstrap. **Sociedade Brasileira de Computação**, 2020. páginas 16, 17, 19, 20, 25
- SCHOOL, Tokio. **A história do Python. As versões de uma linguagem única**. 2021. Url<https://tokioschool.pt/noticias/historia-python/>. páginas 16
- SHAH, Mahamad. Responsive web development using the twitter bootstrap framework. Turun ammattikorkeakoulu, 2015. páginas 20, 21
- TULCHAK, LV; MARCHUK, . **History of python**. Tese (Doutorado) — , 2016. páginas 16
- WIRFS-BROCK, Allen; EICH, Brendan. Javascript: the first 20 years. **Proceedings of the ACM on Programming Languages**, ACM New York, NY, USA, v. 4, n. HOPL, p. 1–189, 2020. páginas 21