

“CERES”: PLATAFORMA *WEB* PARA DIVULGAÇÃO DE PRODUTOS AGRÍCOLAS DE PEQUENOS PRODUTORES

SAMUEL ANDREATA CANEVARI¹; FILIPE EVANGELISTA AVILA²; MAURÍCIO DUARTE³; LUIZ CARLOS QUERINO FILHO⁴

¹ Discente em Big Data no Agronegócio na FATEC Pompéia “Shunji Nishimura”, Pompeia-SP, samuel.canevari@fatec.sp.gov.br

² Discente em Big Data no Agronegócio na FATEC Pompéia “Shunji Nishimura”, Pompeia-SP, filipe.avila@fatec.sp.gov.br

³ Docente do curso Big Data no Agronegócio, FATEC Pompéia “Shunji Nishimura”, Pompeia-SP, luiz.querino@fatec.sp.gov.br

⁴ Docente do curso Big Data no Agronegócio, FATEC Pompéia “Shunji Nishimura”, Pompeia-SP, mauricio.duarte@fatec.sp.gov.br

RESUMO

Com o crescimento da Internet como ferramenta para venda de produtos e serviços, propiciando conforto e praticidade tanto ao vendedor quanto ao consumidor, o site “Ceres” foi desenvolvido pensando nesses pontos e visando otimizar os negócios dos pequenos produtores. Estima-se que 77% dos estabelecimentos da Agricultura têm enquadramento familiar, e que a comercialização de seus produtos normalmente é feita por meio de intermediários, que realizam a venda desses produtos e os negociam para mercados médios. Sendo assim, o pequeno produtor, de característica familiar acaba ficando dependente de terceiros, uma vez que não possui empoderamento ou força para poder gerenciar e vender seus produtos diretamente ao cliente final. Dessa forma, esse trabalho focou no desenvolvimento de uma plataforma *web* denominada “Ceres” e foram utilizados HTML, CSS, JavaScript e Django como tecnologias de desenvolvimento, oferecendo um recurso para que o pequeno produtor possa divulgar seus produtos regionalmente, assim como seus dados para contato e localização, e um sistema de chat online para que ele consiga entrar diretamente em contato com seus eventuais compradores. Foi possível verificar que os benefícios seriam amplos: não só o vendedor conseguiria divulgar sua produção, mas também o cliente se beneficiaria ao conseguir localizar, por meio dessa plataforma, e em sua região, produtos de seu interesse, e negociá-los diretamente com o produtor, evitando ir a um mercado.

Palavras-chave: *Website*. Produtores agrícolas. Agricultura familiar. Chat em tempo real.

1. INTRODUÇÃO

1.1. As pequenas produções agrícolas no Brasil

A agricultura familiar tem como característica uma produção de alimentos variados e, muitas vezes, focada primeiramente no consumo interno de quem produz. É desenvolvida, como o próprio nome já diz, pela família, e a produção ocorre em pequenas propriedades de terras, diferenciando-se das grandes propriedades do Agronegócio, que geralmente produzem em massa um único tipo de plantaço visando à produção, exportação e negociação em massa.

Estima-se que 77% dos estabelecimentos agrícolas no Brasil são classificados do tipo familiar, de acordo com o último Censo Agropecuário, realizado em 2017 pelo IBGE (GODOY, 2021). Sendo assim, pode-se afirmar que a maioria dos alimentos que chegam às mesas do consumidor, em algum momento, foram provenientes desse tipo de estabelecimento. Trata-se de pequenas propriedades, cujos proprietários são chefes de famílias e responsáveis por cultivar frutas, verduras e legumes, não tendo muitas vezes estruturas suficientes para manter o negócio funcionando e rendendo grandes lucros, produzindo apenas o necessário para a subsistência da família (MANGIATERRA, 2020).

No Brasil, as pequenas produções envolvem aproximadamente 4,4 milhões de famílias, segundo informações do Ministério da Agricultura, Pecuária e Abastecimento (MAPA). Além disso, a Organização das Nações Unidas (ONU) estima que 80% de todo alimento do planeta venha desse tipo de produção (CRESOL, 2020).

No Brasil, a maior concentração de produções do tipo agricultura familiar ocorre nas regiões Norte e Nordeste, em estados como Pernambuco, Ceará e Acre (ELDER, 2022). Já na região Sudeste, pelo fato de ser mais desenvolvida e tecnológica, o local possui menores níveis de concentração dessa forma de agricultura, dedicando-se mais à produção em larga escala visando exportações.

O censo ainda mostrou que a agricultura familiar no país é responsável por empregar mais de 10 milhões de pessoas, sendo os homens a maioria, com 81%, e as mulheres 19%. A faixa etária fica em torno de 45 e 54 anos (CRESOL, 2020).

Outros dados interessantes mostram que esses pequenos agricultores são responsáveis por produzir cerca de 70% do feijão nacional, 34% do arroz, 87% da mandioca, 60% da produção de leite e 59% do rebanho suíno, 50% das aves e 30% dos bovinos (CRESOL, 2020).

Diante do exposto, fica clara a importância do desenvolvimento de uma plataforma que possa dar suporte à agricultura familiar, pois, segundo Vânia Marques, secretária da Confederação Nacional dos Trabalhadores na Agricultura (Contag): ela

é fundamental para manter o mercado interno abastecido e a alimentação na mesa da população brasileira, garantindo a segurança alimentar e nutricional que tem como base a produção orgânica e agroecológica, além de movimentar a economia local, principalmente, nos pequenos municípios (GODOY, 2021).

1.2. Dificuldades enfrentadas pelos pequenos produtores agrícolas

O pequeno agricultor passa por diversas dificuldades em suas atividades, como, por exemplo, a gestão e manutenção das pessoas envolvidas no negócio, que geralmente são seus descendentes. Esse é um problema vivido pelos agricultores mais velhos que, por um lado, precisam de mão de obra na propriedade, e por outro, almejam proporcionar melhores condições de estudo para os filhos (MANGIATERRA, 2020).

Outro problema é o fato de a comercialização dos produtos ser feita através de intermediários que realizam a operação de compra e depois vendem tais produtos para grandes mercados. Sendo assim, o pequeno produtor fica dependente de uma situação na qual não há empoderamento ou força para gerar outras opções comerciais diferentes daquela (MANGIATERRA, 2020).

Em um país como o Brasil, com dimensões continentais e grande problemas infraestruturais de trajetos, como por exemplo o alto custo por quilômetro rodado e o escoamento e transporte da produção, muitas vezes não é economicamente viável para os pequenos agricultores transportar mercadorias e isso impede que eles tenham contato direto ao grande mercado” (MANGIATERRA, 2020).

Além disso, em função da pandemia do coronavírus, surgiu uma outra dificuldade: muitas feiras onde ocorria o comércio de produtos agrícolas de cidades locais foram fechadas, e os produtores tiveram que divulgar seus produtos em redes sociais como Facebook, Instagram, grupos de Whatsapp entre outros, para garantir uma renda e sustentar suas famílias; além disso, o uso de tais meios não era totalmente dominado por eles.

1.3. Objetivos e foco do trabalho desenvolvido

Em decorrência de todos os pontos citados e, principalmente, das dificuldades vividas pelos pequenos agricultores, esse trabalho de graduação teve como objetivo desenvolver uma aplicação *web* para proporcionar uma plataforma de divulgação de

mercadorias de pequenos produtores, além de criar um canal de comunicação direto, via *chat online*, entre vendedor e consumidor.

Em razão das condições logísticas de transporte, e da perecibilidade das mercadorias envolvidas, optou-se por não envolver as operações de transação no projeto, dando liberdade ao vendedor de vender seus itens da maneira como achar mais conveniente, seja levando-os diretamente às casas dos compradores, seja marcando um local de negociação, entre outros.

2. MATERIAL E MÉTODOS

2.1 Figma

Para o desenvolvimento inicial do *site*, foi utilizada a plataforma colaborativa Figma que, segundo Villain (2022), é adequada para a construção e prototipação de design de interfaces. Com ela é possível construir o *design* de produtos digitais inteiros, independentemente do nível de complexidade dele, quer seja algo simples, quer seja mais elaborado. Com o Figma é possível explorar o máximo possível do *design* de interface e da criação de fluxos inteiros de navegação entre páginas.

2.2 HTML

Para a estruturação do site, utilizou-se o HTML (*HyperText Markup Language*, ou Linguagem de Marcação de Hipertexto) que, segundo Longen (2022), permite o desenvolvimento de funcionalidades dinâmicas, em que, com o auxílio de elementos, *tags* e atributos, podem-se criar seções, parágrafos e *links*. Diante disso, deve-se ressaltar que o HTML não é considerado necessariamente uma linguagem de programação, mas sim uma linguagem do computador que estrutura a maior parte das plataformas digitais.

2.3 CSS

Também foi necessário o uso do CSS que, conforme Gonçalves (2022) explica, CSS, ou *Cascading Style Sheet*, é utilizado para a estilização dos elementos de uma linguagem de marcação, como, por exemplo, o HTML, separando o conteúdo da representação visual do *site*. Um exemplo do seu uso é a possibilidade de alterar cores do texto e do fundo do site, tamanho de fonte e espaçamento entre os

parágrafos. Desse modo, utilizar essa tecnologia, em conjunto com o HTML, foi de suma importância na construção de todo o *layout* do *site* e suas respectivas telas, e na estilização de formulários, *cards*, entre outros componentes.

2.4 JavaScript

Recorreu-se ao uso do JavaScript que, de acordo com o Kinsta (2022), “é uma linguagem de *script* popular para adicionar funcionalidade interativa e outros conteúdos dinâmicos da *web* às páginas *web*”, o que permitiu, diante dessa funcionalidade, juntamente com o HTML e CSS mencionados anteriormente, otimizar o desenvolvimento da plataforma *web*, oferecendo ao usuário final uma alta capacidade de interatividade com os recursos e funcionalidades do site.

2.5 MySQL

A fim de salvar as informações do usuário, como seus dados de cadastro e mensagens de suas conversas, utilizou-se o banco de dados MySQL, que é “um *software* de código aberto ou serviço usado na criação e gerenciamento de bancos de dados baseados no modelo relacional” (LONGEN, 2021). Vale ressaltar que código aberto se refere à liberdade que se tem de manusear e modificá-lo; banco de dados se refere ao local onde são armazenados e gerenciados os dados obtidos, e modelo relacional diz respeito à forma como eles são organizados, que, no caso, são em tabelas.

2.6 Python e Django

Para o desenvolvimento rápido e design limpo e prático, foi utilizado o *framework web* Django, que é escrito na linguagem de programação Python, que, segundo a AWS (2022), é de grande eficiência e de facilidade no aprendizado, podendo ser implementado em diferentes tipos de sistemas, agilizando o seu desenvolvimento. Em conjunto com o Python, o Django (2022) possibilita a solução de grande parte dos incômodos do desenvolvimento *web*, como o tratamento de requisições e a organização das respostas HTTP, objetivando o propósito do desenvolvimento em si.

2.7 Desenvolvimento do chat em tempo real

Primeiramente, é necessário compreender que, no desenvolvimento de uma aplicação como, nesse caso, um *chat* em tempo real, é preciso dividir o processo em duas partes: o *Front-end* e o *Back-end*. O primeiro, segundo Souto (2022), se refere respectivamente à parte visual da aplicação, ou, de forma mais simples, àquilo com que o usuário consegue interagir, onde são utilizadas normalmente as tecnologias base da *web*, como HTML, CSS e o JavaScript. Já o segundo é aquilo que se tem por trás da aplicação, fazendo geralmente a ponte entre as informações que vêm de um banco de dados e do *Front-end*.

Diante disso, no desenvolvimento *Back-end*, foi utilizado o Django em conjunto com um dos seus pacotes, o chamado Channels, que, de acordo com Tomazic (2022), estende os recursos internos do próprio Django e fornece suporte para eles, como, por exemplo, recursos de autenticação, permitindo não lidar apenas com HTTP, mas também com protocolos que exigem conexões de longa duração, como WebSocket e outros recursos em tempo real.

Já, no *Front-end*, foi utilizado o JavaScript com WebSocket, sendo este uma “tecnologia avançada para criar uma ligação entre um cliente e um servidor (*browser* e servidor) e permitir a comunicação entre eles em tempo real” (APPMASTER, 2022).

O WebSockets foi usado no lugar do HTTP, uma vez que, no HTTP, todas as vezes em que se precisa receber novas informações, é necessário enviar uma nova solicitação ao servidor, que ao recebê-la, envia os dados solicitados, atrasando o processo. Já no caso do WebSocket, uma vez estabelecida a conexão, os dados são enviados por si mesmos sem que seja necessário enviar um pedido separado. Essa é a vantagem de utilizá-lo em um *chat* em tempo real, onde se requer envio e recebimento de dados de forma constante.

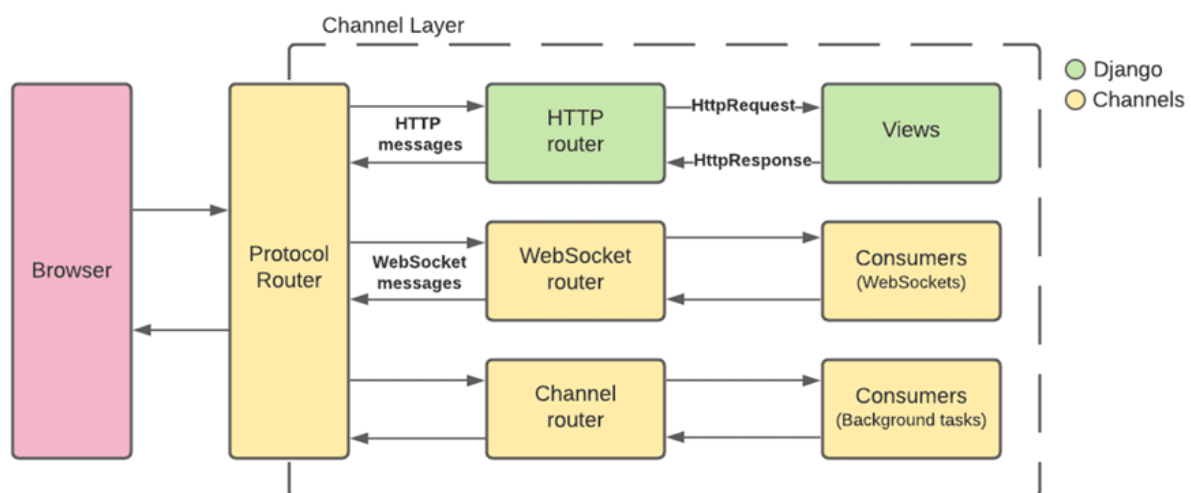
Entre outros fatores que surgiram durante o desenvolvimento, alguns devem ser ressaltados, tendo em vista algumas diferenças entre o Channels e o *framework* Django, foi necessário a alternância na execução de códigos síncronos, onde os processos são executados em sequência e de forma ordenada, seguindo uma fila, e os de códigos assíncronos, onde os processos são executados ao mesmo tempo, sem que nada impeça que um processo comece mesmo que outro já tenha sido iniciado anteriormente. Essa alternância pode ser notada quando o Django precisa

utilizar códigos síncronos para acessar o banco de dados, enquanto o Channels precisa ser acessado utilizando-se um assíncrono.

Outro fator importante é o fato do Channels, que de acordo com sua documentação, utiliza como unidade básica de código os chamados *consumers*, que recebem este nome por serem responsáveis em consumir eventos, sendo que estes eventos representam as interações do usuário, como por exemplo, quando o usuário enviar ou receber mensagens em um *chat*.

Assim, para consolidar a ideia de como o Channels opera, a Figura 1 mostra de forma simplificada uma configuração típica do Channels. Nela é possível observar que, após uma conexão ser efetuada, o *browser*, ou seja, o navegador da internet, consegue enviar e receber solicitações. Essas solicitações são enviadas a um protocolo de roteamento, que tem como função especificar como outros roteadores se comunicam e distribuem as informações a eles. Por essa razão, é possível trabalhar com o HTTP e WebSocket de forma separada, definindo um roteador específico para cada um deles. Por fim, esses roteadores encaminham essas solicitações para o *consumer* responsável por ele.

Figura 1 - Configuração básica do Channels



Fonte: Tomazic (2022)

3. RESULTADOS E DISCUSSÃO

Tendo em vista a facilidade que as plataformas de mensagens instantâneas oferecem aos seus usuários, para uma comunicação simples e direta, percebe-se uma crescente demanda por funcionalidades que criem um canal de comunicação

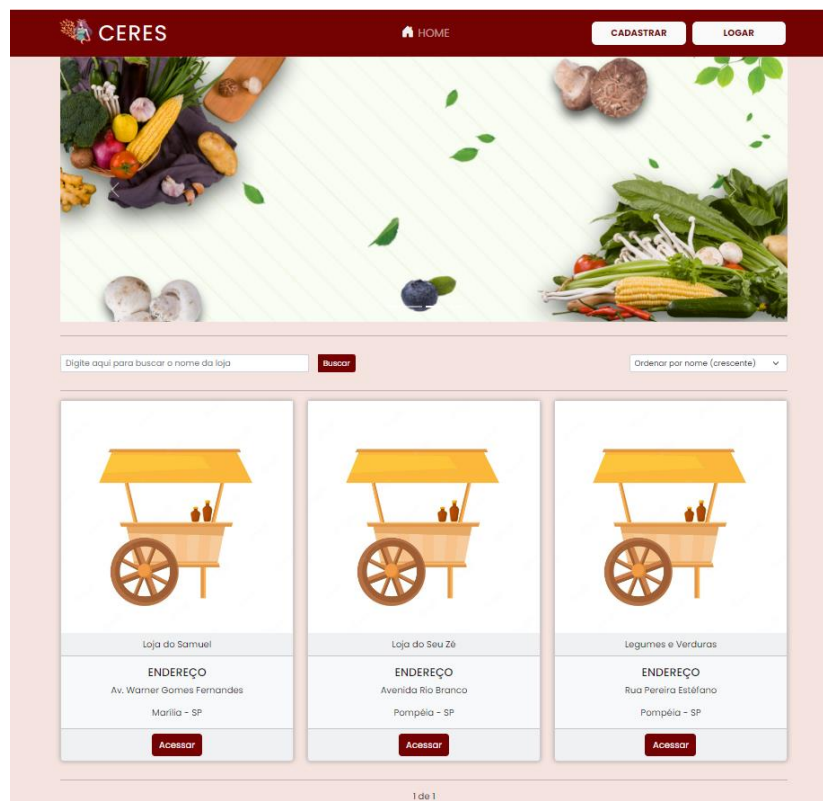
entre uma parte que oferece um produto/serviço, e outra, que o adquire - o cliente. Sendo assim, decidiu-se pela implementação dessa função, na qual desenvolveu-se uma plataforma *web* com os materiais e métodos mencionados no item 2, com o objetivo de ser de fácil acesso e manuseio, possibilitando ao usuário uma navegação limpa e clara de compreensão.

Diante disso, o *site* possibilita ao usuário localizar páginas de loja, juntamente com seus respectivos produtos; permite uma fácil comunicação entre vendedor e consumidor e vendedor; e até mesmo, propicia a esse último a possibilidade de expor seus próprios produtos, estabelecendo assim, uma publicidade de seu negócio.

Tratando-se de um sistema de exposição de produtos agrícolas, optou-se por nomeá-lo de “Ceres”, referindo-se a deusa romana, que representava o poder produtivo do solo, a agricultura, e, em geral, a prosperidade.

A tela inicial, ou tela Home, ilustrada na Imagem 1, é composta pelas lojas já cadastradas por quem deseja divulgar seus produtos para venda. Nessa tela, o usuário pode fazer uma pesquisa por uma loja específica ou apenas visualizá-las. Nela também é possível acessar outras telas e funções da plataforma, como Perfil, Cadastro e Login.

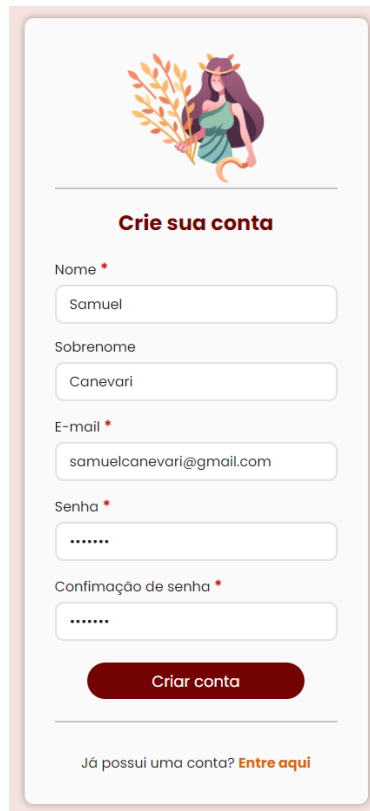
Imagem 1 - Tela Home



Fonte: Elaborado pelos autores (2022).

Caso o usuário queira acessar todos os recursos do site, é necessário que ele crie uma conta, devendo primeiramente preencher o formulário de cadastro com seu nome/sobrenome, e-mail e uma senha, conforme mostra a Imagem 2.

Imagem 2 - Tela Cadastro de usuário




A imagem mostra a tela de cadastro de usuário. No topo, há uma ilustração de uma mulher com longos cabelos castanhos, vestindo um vestido verde, segurando uma cesta de colheita com espigas de milho. Abaixo da ilustração, o título "Crie sua conta" está em negrito. O formulário contém cinco campos de entrada: "Nome" com o valor "Samuel", "Sobrenome" com "Canevari", "E-mail" com "samuelcanevari@gmail.com", "Senha" e "Confirmação de senha", ambos com pontos para ocultar o texto. Um botão vermelho com o texto "Criar conta" está posicionado abaixo dos campos. Na base da tela, há um link que diz "Já possui uma conta? [Entre aqui](#)".

Fonte: Elaborado pelos autores (2022).

Ao completar o cadastro, o usuário é redirecionado à tela Login, onde poderá acessar sua conta cadastrada anteriormente, com seu e-mail e senha, conforme mostra a imagem 3.

Imagem 3 - Tela Login de usuário



Acesse sua conta

E-mail *

Senha *

Entrar

Não tem uma conta? [Cadastre-se aqui](#)

Fonte: Elaborado pelos autores (2022).

Havendo o usuário logado, ele terá a opção de criar sua própria loja; para isso, deverá preencher um novo formulário, informando os dados da futura loja como nome, bairro e endereço onde a mercadoria é produzida ou vendida, além do estado, cidade e um telefone para contato, conforme ilustrado na Imagem 4.

Imagem 4 - Tela Cadastro de loja

Crie sua loja

Nome da loja *

Bairro *

Endereço * Número

Estado * Cidade *

- Selecione - ▾ - Selecione - ▾

Telefone para contato *

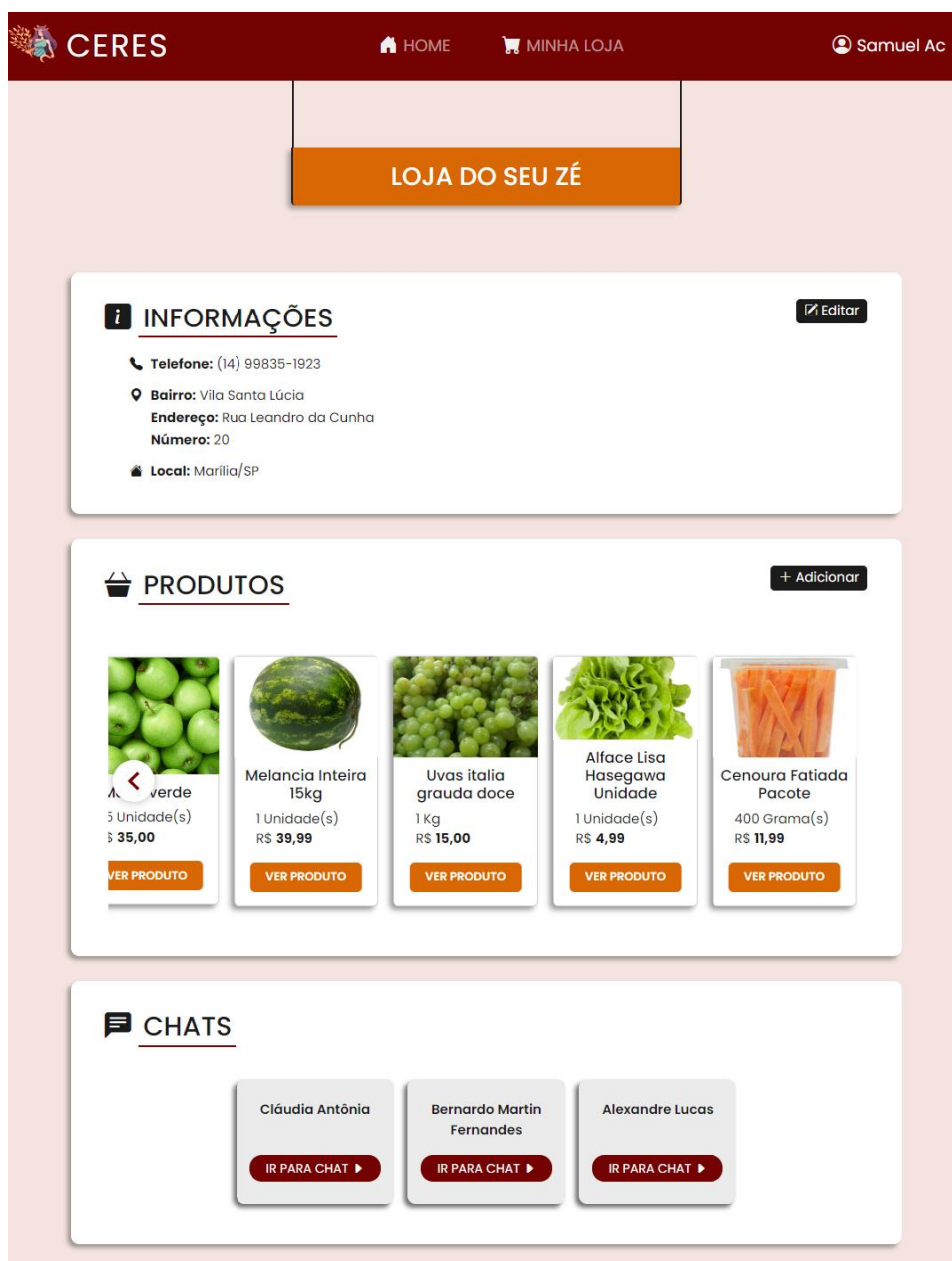
Criar loja

[Voltar à página principal](#)

Fonte: Elaborado pelos autores (2022).

Assim que a loja é criada, conforme a Imagem 5, o proprietário pode editar as informações da mesma a qualquer momento, bem como adicionar ou remover os produtos que serão vendidos, e acessar os chats com seus clientes. Já, o outro usuário, que seria o cliente, poderá, caso tenha interesse em algum produto, ver seus detalhes mais específicos e iniciar uma conversa com o dono da loja, para buscar mais informações, como quantidade, preço ou qualidade da mercadoria.

Imagem 5 - Tela Loja



Fonte: Elaborado pelos autores (2022).

O proprietário da loja pode adicionar produtos ao preencher um formulário, indicado na Imagem 6, informado o nome, a categoria, o preço, a quantidade e uma breve descrição sobre o produto. No final do cadastro, ele pode optar por adicionar o produto e voltar à tela da Loja, ou adicionar vários produtos em sequência.

Imagem 6 - Tela Cadastro de Produto

Adicionar produto

Nome do produto *
Cenoura Fatiada Pacote

Categoria *
Verduras

Quantidade *
400 Grama(s)

Preço total *
R\$ 11,99

Descrição *
Cenoura Fatiada Pacote (Pacote com aprox. 400g)

Foto *
Escolher arquivo cenoura.jpg

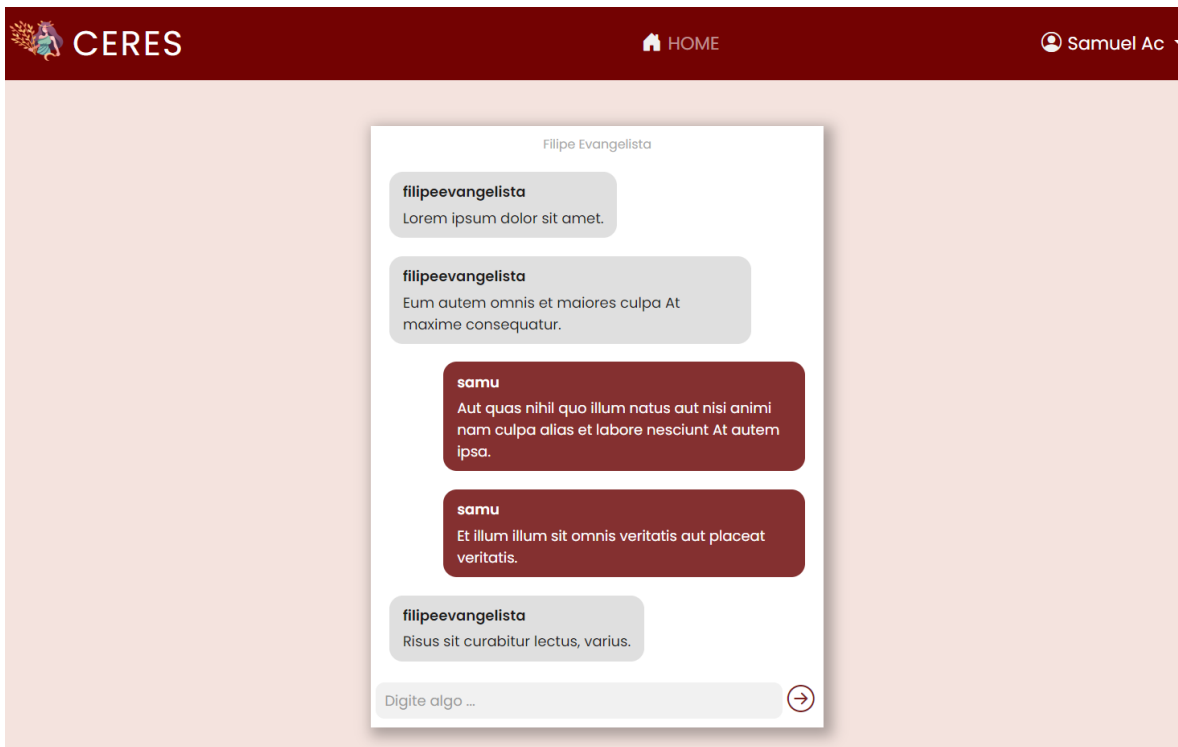
Adicionar produto **Continuar adicionando**

[Voltar à pagina principal](#)

Fonte: Elaborado pelos autores (2022).

Por fim, na Tela Chat, registrada na imagem 7, o cliente e o proprietário da loja entram em contato direto um com o outro e podem negociar a venda dos produtos, estabelecendo um outro valor, a forma de pagamento e a de entrega, sendo que esta poderá ocorrer de duas maneiras: ou o cliente retira diretamente com o proprietário, ou ele recebe a mercadoria no endereço fornecido por ele, uma vez que o sistema não conta com funcionalidade de transação monetária. Vale ressaltar que essa tela somente poderá ser acessada por usuários logados.

Imagem 7 - Tela Chat



Fonte: Elaborado pelos autores (2022).

4. CONCLUSÕES

O site criado alcançou as metas pré-estabelecidas, gerando um fluxo estável de informações rápidas, tanto na parte de apresentação, quanto na de comunicação. Tendo em vista o alcance dessas metas, vê-se que já seria possível a utilização dessa plataforma, possibilitando ao pequeno produtor diversas oportunidades de mercado, e visando, em um futuro próximo, a possibilidade de se fazer um sistema de entrega como nova funcionalidade, inclusive tornar a plataforma capaz de operar transações monetárias diretamente no próprio site.

6. REFERÊNCIAS

APPMASTER. **O que são WebSockets e Como Criá-los?**. 2022. Disponível em: <https://appmaster.io/pt/blog/o-que-sao-websockets-e-como-cria-los>. Acesso em: 23. out. 2022.

AWS. **O que é Python?**. 2022. Disponível em: <https://aws.amazon.com/pt/what-is/python/>. Acesso em: 12. out. 2022.

CHANNELS. **Django Channels - Channels 4.0.0 documentation**. 2022. Disponível em: <https://channels.readthedocs.io/en/stable/introduction.html>. Acesso em: 12. out. 2022.

CRESOL. **Agricultura familiar: um legado que resiste.** 2020. Disponível em: <https://blog.cresol.com.br/agricultura-familiar-um-legado-que-resiste/>. Acesso em: 26. set. 2022.

DJANGO. **Django - The web framework for perfectionists with deadlines.** 2022. Disponível em: <https://www.djangoproject.com>. Acesso em: 12. out. 2022.

ELDER. **QUAL A SITUAÇÃO DA AGRICULTURA FAMILIAR NO BRASIL?** 2022. Disponível em: <https://regularizacaofundiariamt.com.br/2022/03/06/qual-a-situacao-da-agricultura-familiar-no-brasil>. Acesso em: 27. set. 2022

GODOY, Thalya. **Pequenos produtores rurais modificam modelo de negócio durante a pandemia.** 2021. Disponível em: <https://sba1.com/noticias/noticia/14908/Pequenos-produtores-rurais-modificam-modelo-de-negocio-durante-a-pandemia>. Acesso em: 28. set. 2022.

GONÇALVES, Ariane. **O que é CSS? Guia Básico para Iniciantes.** 2022. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css>. Acesso em: 12. out. 2022.

KINSTA. **O que é JavaScript? Uma Visão Geral da Linguagem de Script Mais Popular da Web.** 2022. Disponível em: <https://kinsta.com/pt/base-de-conhecimento/o-que-e-javascript/#o-que--javascript>. Acesso em: 12. out. 2022.

LONGEN, Andrei. **O que é HTML? Guia Básico Para Iniciantes.** 2022. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-html-conceitos-basicos>. Acesso em: 12. out. 2022.

LONGEN, Andrei. **O que é MySQL? Guia Para Iniciantes.** 2021. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-mysql>. Acesso em: 12. out. 2022.

MANGIATERRA, Bárbara. **A situação e os desafios do pequeno agricultor no Brasil.** 2020. Disponível em: <https://revistacultivar.com.br/artigos/a-situacao-e-os-desafios-do-pequeno-agricultor-no-brasil>. Acesso em: 28. set. 2022.

SOUTO, Mario. **Front-end, Back-end e Full Stack.** 2022. Disponível em: <https://www.alura.com.br/artigos/o-que-e-front-end-e-back-end>. Acesso em: 23. out. 2022.

TOMAZIC, Nik. **Introduction to Django Channels.** 2022. Disponível em: <https://testdriven.io/blog/django-channels/>. Acesso em: 12. out. 2022.

VILLAIN, Mateus. **Figma: o que é a ferramenta, Design e uso.** 2022. Disponível em: <https://www.alura.com.br/artigos/figma>. Acesso em: 23. out. 2022.