



Software livre para o comércio justo e solidário: apoio à agricultura familiar

Free software for fair trade and solidarity economy: support for family farming

João Manoel da Luz Quevedo

joaque@alunos.utfpr.edu.br

Universidade Tecnológica Federal do Paraná, Pato Branco, Paraná, Brasil

Vinicius Pegorini

vinicius@utfpr.edu.br

Universidade Tecnológica Federal do Paraná, Pato Branco, Paraná, Brasil

Antonio Augusto Maccarini

antoniomaccarini@alunos.utfpr.edu.br

Universidade Tecnológica Federal do Paraná, Pato Branco, Paraná, Brasil

Maria Luisa Carvalho

mluisacarvalho@utfpr.edu.br

Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil

Almir Antonio Gnoatto

almirgnoatto@utfpr.edu.br

Universidade Tecnológica Federal do Paraná, Dois Vizinhos, Paraná, Brasil

Gabriel Prando

gprando@alunos.utfpr.edu.br

Universidade Tecnológica Federal do Paraná, Pato Branco, Paraná, Brasil

Gustavo Tiecker

gustavotiecker@gmail.com

Universidade Tecnológica Federal do Paraná, Pato Branco, Paraná, Brasil

Daniel Gustavo Favero

danielfavero17@gmail.com

Universidade Tecnológica Federal do Paraná, Pato Branco, Paraná, Brasil

RESUMO

Diferentes setores da sociedade necessitam de sistemas computacionais para resolver problemas específicos que não são atendidos pelos softwares livres existentes na comunidade, geralmente esses setores não possuem pessoas com conhecimento para desenvolvimento de software. Neste trabalho, por meio de uma demanda da marca Produtos da Terra, vinculada a Cooperativa Central da Reforma Agrária, foi desenvolvido um sistema de comércio eletrônico. Embora a cooperativa tivesse um site para comercialização, os



consumidores preferiam fazer os pedidos via Whatsapp, o que ocasionava erros de pedidos, retrabalhos, sobrecarga de trabalho e dificuldades no fechamento do caixa semanal. O projeto desenvolvido teve dois objetivos primários, com a criação da aplicação auxiliar a comunidade por meio da tecnologia e capacitar os alunos por meio da prática. As tecnologias foram escolhidas para melhor se encaixar na resolução da questão, entre elas: PostgreSQL, React, Javascript, Node.js, Express, ORM, GitHub. No fim do projeto, foi criado o sistema Merkur com a capacidade de cadastrar os itens vendidos, criação de cestas de compra pelos clientes, cadastros de usuários, cadastros de produtores, pesquisa de itens e criação de itens promocionais. A partir dos resultados obtidos foi possível observar que o software será importante no auxílio da cooperativa para a centralização dos pedidos.

PALAVRAS-CHAVE: Comércio eletrônico. Software livre. Agricultura familiar. Agroecologia. Economia Solidária.

ABSTRACT

Different sectors of society need computer systems to solve their problems, but the existing free software may not be capable to do that. Generally, these sectors do not have people with knowledge of software development. In this work, through a demand for the Produtos da Terra, linked to the Central Cooperative for Agrarian Reform, an electronic commerce system was developed. Although the cooperative had a website to sell their products, consumers preferred to place orders via Whatsapp, which caused order errors, rework, work overload, and difficulties in closing the weekly cash register. The project developed had two primary objectives, with the creation of the application to help the community through technology and to train students through practice. The technologies were chosen to best fit the resolution of the issue. Among them were used: PostgreSQL, React, Javascript, Node.js, Express, ORM, GitHub. At the end of the project, the Merkur system was created with the ability to register the items sold, create shopping baskets by customers, register users, register producers, search for items and create promotional items. From the results obtained, it was possible to observe that the software will be important in helping the cooperative to centralize orders.

KEYWORDS: E-commerce. Free software. Family farming. Agroecology. Solidarity economy.



INTRODUÇÃO

Um software livre respeita a liberdade do usuário e da comunidade, isso significa que os usuários têm a liberdade de executar, copiar, distribuir, estudar, mudar e melhorar o software. A liberdade de executar o programa, para qualquer propósito. A liberdade de executar o programa significa a liberdade de qualquer pessoa ou organização de usá-lo em qualquer sistema computacional para qualquer tipo de trabalho e propósito, sem a necessidade de comunicar o desenvolvedor ou qualquer entidade (STALLMAN, 2021).

Existem 4 ordens de liberdade. Liberdade 0: A liberdade de executar o programa como desejar; Liberdade1: A liberdade de estudar o código-fonte do programa e mudá-lo para fazer o que você deseja; Liberdade 2: A liberdade de ajudar seu próximo. Essa é a liberdade de redistribuir cópias do programa para outras pessoas quando desejar; e Liberdade 3: A liberdade de contribuir com sua comunidade. Se você tiver todas essas quatro liberdades, o programa é um software livre (STALLMAN, 2021).

Geralmente o software livre é criado por meio do envolvimento da comunidade, que contribui para o desenvolvimento de software para as mais diversas finalidades. Um exemplo de software livre aceito e adotado é o sistema operacional Linux, que veio para suprir a necessidade de um sistema operacional livre e sem custos para a comunidade. Nesse contexto, o surgimento de novos softwares livres visa atender setores carentes de softwares livres, tais como economia, transporte e agricultura (AGROLIVRE, 2021).

O uso de software na agricultura visa atender a demanda do setor agropecuário nas áreas de sistemas de manejo, sistemas de apoio à tomada de decisão, de apoio à pesquisa científica e de apoio a projetos de inclusão digital (AGROLIVRE, 2021).

O desenvolvimento de software livre também visa auxiliar a agricultura familiar, segundo a Freitas et al. (2009) “promovendo a consolidação e a disseminação de conhecimentos relativos ao mapeamento da produção e informações para auxílio à comercialização de produtos convencionais, orgânicos e artesanais”. Dessa maneira os softwares visam auxiliar na tomada de decisões por parte dos produtores, cooperativas, associações, comerciantes, governos e outros atores envolvidos na agricultura familiar.

A agricultura familiar é toda forma de cultivo da terra e produção rural em que a administração e mão de obra são formadas por um grupo familiar, no Brasil ela é realizada por quem têm a terra como fonte primária de renda. A estrutura, organização e produção são algumas das características que a diferenciam da agricultura patronal. Enquanto a agricultura patronal baseia-se na monocultura, contratação de funcionários para trabalhar nos grandes sistemas produtivos das propriedades e utilização de latifúndios, a agricultura familiar produz diversas variedades de alimentos, os funcionários são os próprios familiares e a produção ocorre em pequenas propriedades (JUNQUEIRA; LIMA, 2008).

Para algumas famílias afastadas das zonas urbanas, por vezes, é através do comércio dos produtos oriundos de suas propriedades que seu sustento é gerado. Dessa maneira, a agricultura familiar surte impacto dentro da economia, criando emprego e renda dentro da área rural. Outro ponto a ser citado são algumas questões relacionadas à sustentabilidade dessa forma de cultivo, que apresenta baixa utilização de agrotóxicos e responsabilidade socioambiental (DELGADO; BERGAMASCO, 2017; PICOLOTTO, 2014).

Um dos problemas das cooperativas de agricultura familiar é a venda e distribuição dos produtos. Nesse contexto esse trabalho atuou em parceria com a rede colaborativa, Produtos da Terra, que realiza a distribuição de cestas orgânicas dentro do município de Curitiba. Essa rede colaborativa une cooperativas agroecológicas e empreendimentos da Economia Solidária e consumidoras/es conscientes, que apoiam a Reforma Agrária, a Agricultura Familiar, a Agroecologia e a Economia Solidária em Curitiba e região. A Economia Solidária é um outro modelo econômico que congrega atividades produtivas, prestação de serviços, comercialização, consumo e finanças, urbanas e rurais pautadas nos princípios da autogestão, cooperação, solidariedade, comércio justo e solidário e respeito à natureza.



O desenvolvimento do projeto Merkur, foi idealizado para facilitar as necessidades da rede Produtos da Terra (TERRA, 2021), ela apresentava queixas em seu atendimento ao cliente, devido a uma descentralização dos canais de atendimento. A situação se agravou com a pandemia pois houve um aumento expressivo de consumidores devido às restrições de funcionamento do comércio e medidas de isolamento social. Assim, muitas pessoas passaram a comprar as cestas agroecológicas por poderem adquirir produtos sem agrotóxico e em condições seguras. O projeto visava assim, atender as demandas atuais do coletivo, favorecendo que mesmo após a atenuação e/ou fim da pandemia pudesse manter os novos consumidores. O sistema desenvolvido descrito neste artigo tem como objetivo primário estreitar e facilitar o contato do cliente com as cooperativas por meio da centralização dos pedidos de cestas orgânicas em apenas uma plataforma. Os produtores rurais também possuirão um canal de acesso, onde será feito o cadastro dos itens que desejam comercializar. O Merkur conta com diversas funcionalidades, como: cadastro dos itens vendidos, criação de cestas de compra, cadastros de usuários, cadastros de produtores, pesquisa de itens, criação de itens promocionais. Destaca-se que a escolha de uma solução tecnológica pautada no software-livre deve ao fato do mesmo ter princípios e valores consonantes com a Economia Solidária: democracia, cooperação, compartilhamento do saber.

MATERIAIS E MÉTODOS

A aplicação foi desenvolvida para plataforma web. Os materiais utilizados durante o desenvolvimento da aplicação estão listados no Quadro 1.

Quadro 1 – Materiais

Ferramenta	Versão	Finalidade
Figma	1.0	Prototipação da interface.
NodeJs	14.17.5	Desenvolvimento do <i>back-end</i> das aplicações.
Typescript	4.4.2	Linguagem de programação.
Jest	27.1.0	Biblioteca de testes.
PostgreSQL	12	Sistema gerenciador de banco de dados.
TypeORM	0.2.37	Biblioteca para mapeamento objeto-relacional.
BcryptJs	2.4.3	Biblioteca para encriptação de dados.
Swagger	2.2.1	Documentação do <i>back-end</i> da aplicação.
Nodemailer	6.6.3	Biblioteca para envio de emails.
Docker	3.6	Desenvolvimento e implantação da aplicação.
React.js	17.0.2	Desenvolvimento do <i>front-end</i> das aplicações.
Next.js	11.1.0	Desenvolvimento do <i>front-end</i> das aplicações.

Fonte: Autoria própria (2021).

O desenvolvimento foi dividido em seis etapas, coleta de requisitos, modelagem, prototipação, desenvolvimento, testes e implantação. Inicialmente foi realizada a coleta de requisitos, por meio de reuniões entre a cooperativa e a equipe de desenvolvimento. Os principais requisitos elencados estão apresentados de maneira resumida no Quadro 2.



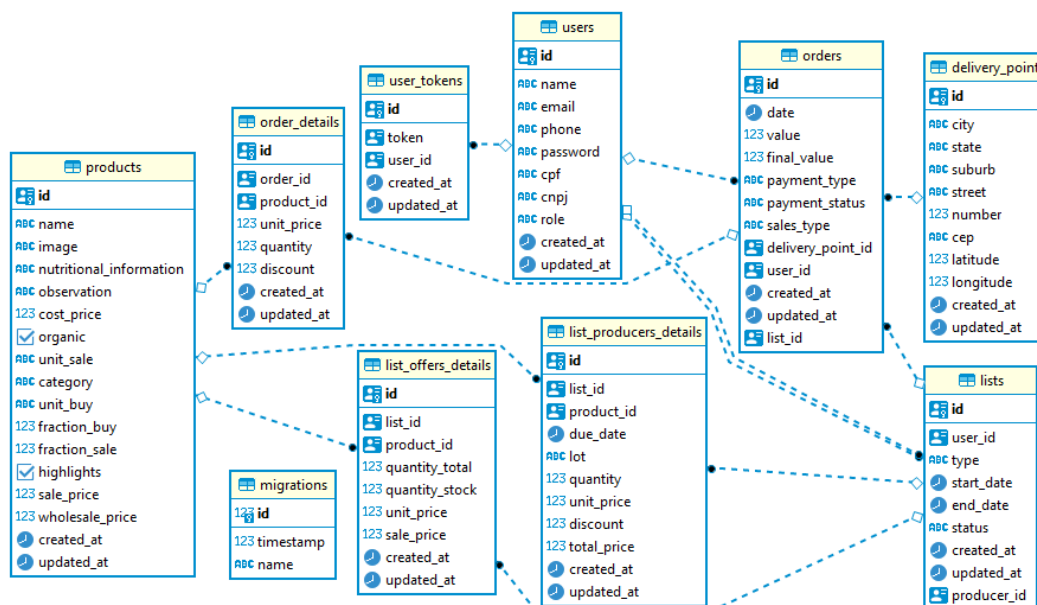
Quadro 2 – Requisitos do sistema

Requisito	Descrição
Cadastro de pessoas	O sistema deve cadastrar usuário, sendo ele do tipo cliente, produtores ou administrador. Essa seleção de tipo deve ser feita através de um campo de seleção.
Comércio Eletrônico - lista de produtos	O sistema deve contar com uma tela para que os clientes possam visualizar os produtos ofertados no período.
Comércio Eletrônico - cesta de compras	O sistema deve contar com uma tela para que os clientes possam visualizar os produtos adicionados para compra.
Comércio Eletrônico - meus pedidos	O sistema deve possuir uma página para que os clientes possam visualizar os pedidos realizados.
Área administrativa - cadastro de produto	O sistema deve possibilitar o cadastro de produtos, com os campos nome, unidade de medida, valor de venda, categoria do produto e lista de características.
Área administrativa - cadastro de ofertas semanais	O sistema deve contar com uma tela para cadastrar as ofertas semanais, em que o usuário pode selecionar os produtos a serem fornecidos e a quantidade de dias que a oferta ficará no ar.
Área administrativa - pedidos realizados	O sistema deve contar com uma tela para que os administradores possam consultar todos os pedidos realizados em um intervalo de datas.

Fonte: Autoria própria (2021).

Na segunda etapa foi realizada a modelagem do banco de dados, o diagrama de Entidade-Relacionamento pode ser visualizado na Figura 1. A tabela 'users' irá armazenar os dados de todos os tipos de usuários, sejam eles administradores da cooperativa, clientes e produtores. Já as tabelas 'lists' juntamente com a tabela 'lists_offers_detalhes' irão armazenar as listas de ofertas semanais que poderão ser compradas pelos clientes. E, as tabelas 'orders' e 'orders_details' irão armazenar os pedidos das cestas realizadas pelos clientes.

Figura 1 – Diagrama entidade-relacionamento



Fonte: Autoria própria (2021).

A prototipação da interface foi realizada na terceira etapa utilizando a ferramenta Figma (FIGMA, 2021). A primeira criação foi a página de autenticação e a página principal da aplicação. E, conforme surgiam a necessidade de novas interfaces, o processo de criação dos protótipos foi realizado.



Após finalizada a modelagem e prototipação, iniciou-se a construção do *back-end* da aplicação. Algumas das funcionalidades envolvidas foram as: *Migrations*, *Entities*, *Interface Repositories*, *Fake Repository*, *TypeORM Repository*, *Services*, *Controllers* e *Routes*. Essas funcionalidades estão descritas no Quadro 3.

Quadro 3 – Componentes do sistema

Componente	Descrição
Migrations	Auxiliou na criação da base de dados, a criação é realizada por meio de <i>frameworks</i> , como por exemplo, o TypeORM.
Entities	<i>Entity</i> é uma classe que mapeia para uma tabela de banco de dados (ou coleção ao usar MongoDB).
Interface Repositories	São listadas todas as funções que serão utilizadas relacionadas com o banco de dados. Como é uma interface só é necessário criar a assinatura do método, os parâmetros e o seu retorno. Ex: <i>findById(id: string): Promise<User undefined></i> ; esse método recebe como parâmetro um ID do tipo <i>string</i> e retorna um tipo <i>User</i> ou <i>null</i> .
Fake Repository	No <i>fake repository</i> são implementadas as interfaces feitas nos <i>repositories</i> . Entretanto, não é usado o banco de dados, por isso o nome 'fake' (do inglês falso). Esse repositório serve exclusivamente para os testes unitários. Então todas as funções que estiverem na <i>interface repository</i> são implementadas no <i>fake repository</i> .
Repository do TypeORM	Tem a mesma função do <i>fake repository</i> mas agora utilizando o banco de dados. São criados todos os métodos de CRUD (<i>create</i> , <i>read</i> , <i>update</i> e <i>delete</i>).
Services	Nos <i>services</i> estão as regras de negócio da aplicação. Ex: Em usuário da aplicação precisa cadastrar-se e atualizar a senha. Então são criados dois <i>services</i> (um para cadastrar e outro para a atualização da senha). Dentro de cada um são tratados apenas as funcionalidades específicas, são realizadas as validações e respostas ao cliente. Observação: Os CRUD são feitos em <i>services</i> separados. Para cada funcionalidade foi criado um <i>service</i> separado (ideia de unidade isolada). Isso porque para cada <i>service</i> gerado é criado um teste unitário (por isso a ideia de <i>service</i> separado). Os testes unitários vão testar cada unidade (<i>service</i>) do sistema e são testados individualmente, isto é, um teste não depende do outro. Pode-se ainda optar por escrever os testes e somente depois desenvolver o <i>service</i> , ou também o contrário.
Controllers	Ficam no <i>back-end</i> da aplicação e servem para atender as requisições do <i>front-end</i> e, então encaminhar para o respectivo <i>service</i> . Após o processamento retornam a resposta para o cliente. Trabalha diretamente com as rotas, isto é, para cada rota é chamado um método de <i>controller</i> .
Routes	Refere-se à determinação de como o <i>back-end</i> responde a uma solicitação do cliente por um <i>end point</i> específico, que é uma URI (ou caminho) e um método de solicitação HTTP específico (GET, POST, PUT e etc). No projeto o roteamento está organizado em arquivos de acordo com as funcionalidades do sistema, sendo que cada funcionalidade possui seu respectivo arquivo.

Fonte: Autoria própria (2021).

Na etapa de testes foram realizados os testes da aplicação. Os testes foram realizados durante todo o desenvolvimento da aplicação pelos desenvolvedores. O *fake repository* serviu de ambiente de testes das interfaces. Por fim, na etapa de implantação do sistema foi utilizado o Docker, que é uma plataforma para virtualizar sistemas.



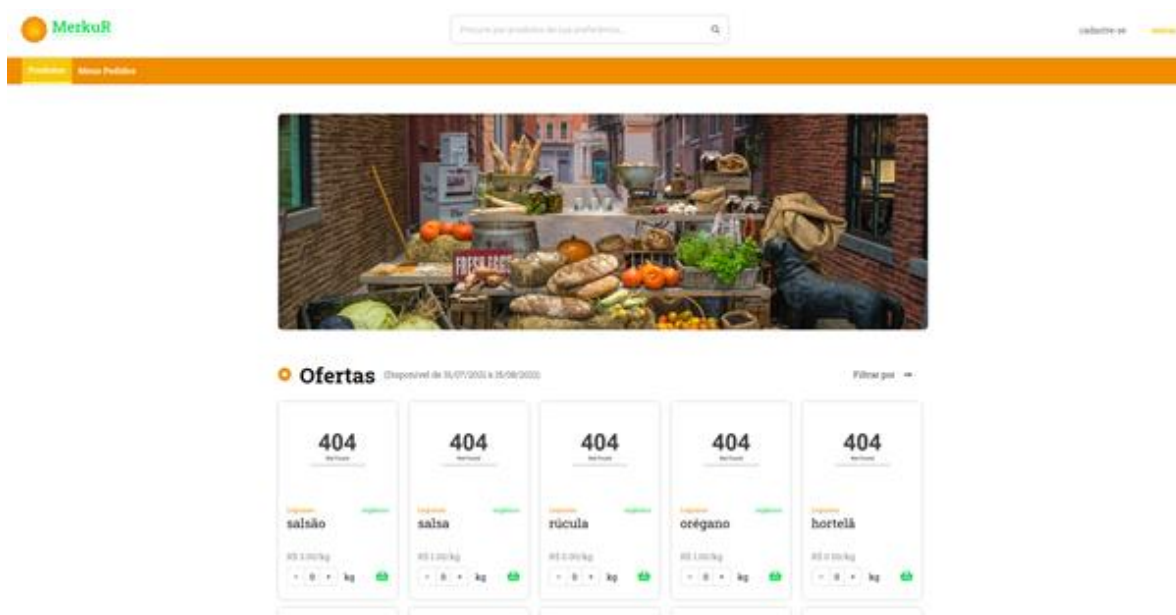
RESULTADOS E DISCUSSÕES

O sistema para comércio eletrônico resultante deste trabalho é uma aplicação com a capacidade de fazer o cadastro de clientes e cadastro de produtores, inserir quais serão os produtos para venda, criar cestas de compra para os clientes e criar listagem dos itens que se encontram em oferta.

Todos os códigos desenvolvidos durante o projeto são públicos e foram disponibilizados por meio do endereço eletrônico: <https://github.com/merkur-org>.

Na Figura 2 é possível visualizar a página inicial do sistema, que conta com um menu superior, no canto superior esquerdo está o logo da cooperativa que irá disponibilizar os produtos à venda. No centro é exibido um campo de busca que permitirá a consulta aos produtos disponibilizados. E na parte esquerda estão os links para o cliente realizar um novo cadastro e outro para o cliente efetuar a autenticação caso já tenha um cadastro. Na sequência da página é apresentado o menu para selecionar as categorias de produtos e um atalho que direciona o cliente para a sua lista de pedidos realizados. E no centro da página são exibidas as principais ofertas de produtos para aquela semana.

Figura 2 – Página inicial do sistema



Fonte: Autoria própria (2021).

Os produtos que serão apresentados como oferta, poderão ser selecionados pelo usuário da cooperativa, pois geralmente são ofertados os produtos sazonais, visando a venda antes da deterioração desses itens.

Para realizar uma compra é necessário estar autenticado no sistema, usando e-mail e senha ou telefone e CPF. Caso o cliente não possua cadastro é apresentada a página de cadastro, que pode ser visualizada na Figura 3. Nessa tela o cliente irá informar o nome completo, e-mail, telefone, CPF e a senha.



Figura 3 – Página de cadastro de cliente

Fonte: Autoria própria (2021).

O sistema desenvolvido também conta com uma cesta de compras, que armazena os produtos selecionados pelos clientes durante a navegação pelas telas de produtos. Nessa tela o cliente pode alterar a quantidade de produtos comprados e também selecionar o local de retirada dos produtos por meio do campo 'Ponto de entrega'. A Figura 4 exibe a tela de cesta de compras.

Figura 4 – Página de cesta de compras

Fonte: Autoria própria (2021).

Outra funcionalidade importante do sistema é o resumo dos pedidos realizados pelos clientes. Essa tela está disponível apenas para os usuários administradores do sistema e nela o usuário poderá emitir relatórios para separar os produtos que devem ser entregues para os clientes. Essa página e as demais páginas do sistema contam com o mesmo padrão de layout, de componentes e cores e não serão exibidas neste trabalho por restrição do número de páginas do documento.

CONCLUSÃO

O sistema desenvolvido permite que uma cooperativa de agricultura familiar ofereça produtos para seus clientes. Um sistema de comércio eletrônico também possibilita a conquista de novos clientes, pois sistemas web ampliam a visibilidade das empresas uma vez que qualquer usuário com acesso à Internet pode utilizar o sistema. É claro que o sistema proposto, por mais que possa ser acessado de qualquer lugar com conexão à Internet ele estará limitado à cidade da cooperativa, pois envolve a venda de produtos perecíveis.



Como o sistema foi desenvolvido com o propósito de ser software livre qualquer cooperativa ou outra entidade pode utilizar o software, bem como modificá-lo de acordo com suas necessidades. Essas modificações também podem tornar-se contribuições permitindo que o software esteja em constante desenvolvimento.

As tecnologias utilizadas para desenvolvimento do trabalho foram suficientes para implementar todas as funcionalidades. O aprendizado dessas tecnologias foi importante para complementar a formação dos alunos envolvidos no projeto, por serem frameworks atuais, os alunos adquiriram o conhecimento de tecnologias amplamente utilizadas e requisitadas por empresas que atuam no desenvolvimento de software.

A parte administrativa do software é responsável pelo cadastro de produtos, listas semanais, fornecedores, pontos de entrega e visualização dos pedidos. O administrador também possui as funções de edição, exclusão, visualização e exportação dos dados. Uma sugestão para uma continuação do trabalho, seria a integração de uma plataforma de pagamento, onde o cliente pudesse realizar o pagamento online e um sistema com emissão de notas fiscais e relatórios.

AGRADECIMENTOS

Agradecemos a todos os que colaboraram para a realização deste projeto, que auxiliaram no seu desenvolvimento e conclusão. Esse projeto foi financiado com uma bolsa de extensão da Universidade Tecnológica Federal do Paraná campus Pato Branco disponibilizada pela PROREC.

REFERÊNCIAS

AGROLIVRE. **Rede de Software Livre para Agropecuária.** Disponível em: <https://www.agrolivre.gov.br/index.html>. Acesso em: 06 Ago. 2021.

DELGADO, Guilherme Costa; BERGAMASCO, Sonia Maria Pessoa Pereira. **Agricultura Familiar Brasileira: Desafios e Perspectivas de Futuro.** Brasília: Ministério do Desenvolvimento Agrário, 2017.

FIGMA. **Figma: the collaborative interface design tool.** Disponível em: <https://www.figma.com/>. Acesso em: 25 jun. 2021.

FREITAS, Antônio Carlos Reis de et al. **e-Sispaf: Sistema de Informação e Promoção de Produtos e Serviços da Agricultura Familiar-Guia de Instalação e Configuração do Sistema.** Versão 0.1, 2009.

JUNQUEIRA, Clarissa Pereira; LIMA, Jandir Ferrera de. **Políticas públicas para a agricultura familiar no Brasil.** Semina: Ciências Sociais e Humanas, Universidade Estadual de Londrina, v. 29, n. 2, p. 159–176, jul. 2008. DOI:10.5433/1679-0383.2008v29n2p177.

PICOLOTTO, Everton Lazaretti. **Os atores da construção da categoria agricultura familiar no Brasil.** Revista de Economia e Sociologia Rural, Fap UNIFESP (SciELO), v. 52, suppl 1, p. 63–84, 2014. DOI:10.1590/s0103-20032014000600004.

STALLMAN, Richard. **The Free Software Foundation.** Disponível em: <https://www.gnu.org/philosophy/free-sw.html.en>. Acesso em: 01 jul. 2021.

TERRA, Produtos da. **Produtos da Terra é uma rede colaborativa que une cooperativas agroecológicas à consumidores conscientes que apoiam a Agricultura Familiar e Orgânica em Curitiba e região.** Disponível em: <https://www.produtosdaterrapr.com.br/> Acesso em: 25 jun. 2021.