

delivEX: um aplicativo de *delivery* multissetorial

Talys Eugênio Batista Ferreira
Graduando em Ciência da Computação - Uni-FACEF
talysebf@gmail.com

Vinicius Ponce Cintra
Graduando em Sistemas da Informação - Uni-FACEF
vinicius.ponce19@hotmail.com

Orientador: Prof. Me Carlos Alberto Lucas
Mestre em Educação – Uni-FACEF
carlos@facef.br

Resumo

Este Trabalho de Conclusão de Curso (TCC) apresenta uma plataforma que faz a junção entre usuário/cliente e usuário/restaurante/conveniência, com a principal atividade sendo a exibição de alimentos e a venda por parte de restaurantes e conveniências, e o consumo destes pelos clientes. Estas ações geram serviços a terceiros que fazem a junção e satisfação entre restaurante/conveniência aos clientes, que neste caso são os *motoboy*s. Hoje todas as plataformas de *delivery* são focadas em apenas um nicho de comércio. Nosso objetivo é integrar em uma mesma solução, os recursos para suprir as necessidades que os empreendedores encontram na oferta de diferentes produtos. Para o desenvolvimento desta plataforma foram utilizadas as boas práticas de desenvolvimento de *software*, visando a melhor qualidade do desenvolvimento, a melhor solução custo/benefício e maior facilidade de manutenção deste *software*. Para o projeto da plataforma, foram utilizadas técnicas de *UX design*.

Palavras-chave: Boas práticas de projeto de *software*. Conveniência. *Delivery*. Multissetorial. *UX Design*.

Abstract

This Course Completion Work (TCC) presents a platform that brings together user/customer and user/restaurant/convenience, with the main activity being the display of food and the sale by restaurants and conveniences, and the consumption of these by customers. These actions generate services to third parties that make the junction and satisfaction between restaurant/customer convenience, which in this case are the *motoboy*s. Today all *delivery* platforms are focused on just one commerce niche. Our objective is to integrate in the same solution, the resources to supply the needs that entrepreneurs find in the offer of different products. For the development of this platform, good *software* development practices were used, aiming at the best development quality, the best cost/benefit solution and greater ease of maintenance of this *software*. For the platform design, *UX design* techniques were used.

Keywords: Convenience. *Delivery*. Good *software* design practices. Multisectoral. *UX Design*.

1. Introdução

No início da pandemia de COVID-19, foram impostas diversas restrições para as pessoas no mundo todo, tendo sido permitido sair às ruas apenas em condições imprescindíveis, pois o vírus ainda não era conhecido e tinha um alto poder de disseminação entre a população, com contágio muito rápido. Com isso, o consumo de alimentos em restaurantes e lanchonetes presencialmente foi proibido. Assim, o consumo de alimentação por plataformas de *delivery* foi uma das soluções para pessoas que antes tinham o consumo físico em determinados restaurantes e lanchonetes. Essa solução teve um aumento exponencial de acesso, comparado ao período pré-pandemia. Com a diminuição de casos, as restrições começaram a diminuir, e o consumo dentro de lanchonetes e restaurantes foi lentamente retornando aos padrões normais de antes da pandemia. No entanto, o consumo das plataformas de *delivery* se manteve, tendo pouca diminuição. As plataformas de *delivery* mais usadas são especializadas em algum nicho de mercado, deixando outros negócios sem atendimento, ou com atendimento inadequado para seus processos de comercialização e gestão. Por exemplo, comércios de bebidas, gelo, carvão, etc. não são atendidos em suas necessidades, ou, para usarem a plataforma, têm que modificar seus processos operacionais e gerenciais para se adequar. Tendo sido identificada a demanda por um aplicativo que possa abranger uma gama maior de empresas para atender à população com restrições de mobilidade, ou que desejam maior conveniência, foi estabelecida a questão problema deste projeto de TCC.

A proposta do projeto, é oferecer uma plataforma de serviços de *delivery* que atende inicialmente, restaurantes e lojas de conveniência em um único aplicativo. Uma solução que proporciona aos comerciantes o serviço centralizado com as funcionalidades necessárias para atendimento adequado de seus clientes e que ofereça recursos para gestão eficiente do negócio.

Nosso objetivo é integrar numa mesma solução, os recursos para suprir as necessidades que os empreendedores encontram em diferentes produtos, dificultando a utilização dos aplicativos concorrentes.

1.1 O Comércio Eletrônico - *e-Commerce*

O *e-Commerce* surgiu no início da década de 1990. Uma das pioneiras do setor é a Amazon.com, com sede em Seattle, nos Estados Unidos. No Brasil, a primeira empresa que comercializa produtos a partir de sites na internet foi a Livraria Cultura, começando suas atividades em 1995.

Desde então, o comércio eletrônico cresceu e se tornou popular em todo o mundo. No Brasil, em 2021, as vendas pela internet faturaram aproximadamente R\$161 bilhões, representando um aumento de quase 27% em relação ao ano anterior. Com estes dados pode-se notar o crescimento que ocorreu neste período no mercado digital. Mesmo em momentos de instabilidade, o comércio eletrônico vem apresentando crescimento, consolidando as vendas *online* no Brasil (EBIT, 2021).

Os números refletem a tendência de alta dos negócios *online*, que precisam se adaptar para atender às necessidades crescentes e em constante mudança. Embora muitas pessoas vejam esse processo como óbvio, hoje existem muitos fatores que tornam a forma de comprar e vender *online* adequada a quase todas as áreas de negócio. As lojas *online* não vendem apenas eletrodomésticos, mas também mantimentos, alimentos prontos, remédios, serviços e muito mais.

A população vem se adaptando cada vez mais às comodidades do comércio eletrônico: se no passado havia incertezas sobre esse tipo de serviço, atualmente os negócios *online* ganham a confiança do consumidor. Mais da metade da população brasileira está conectada e utiliza a internet em algum momento para fazer suas compras. Pesquisa, escolhe, e adquire produtos através do navegador ou de aplicativos em dispositivos móveis (TOMÉ, 2019).

Como vantagens do comércio eletrônico, na visão do consumidor, podem-se destacar a disponibilidade de lojas e produtos em qualquer horário, o acesso a partir de qualquer lugar, a ausência de filas, melhores ofertas e facilidade de encontrar o produto, além da maior facilidade para pesquisa de preços (TOMÉ, 2019).

De acordo com os conceitos de Albertin, comércio eletrônico é a “realização de toda a cadeia de valor dos processos de negócio num ambiente eletrônico, por meio da aplicação intensiva de tecnologia de comunicação e informação, atendendo aos objetivos da empresa”.

1.2 Uso de plataformas de *delivery*

Com a evolução da tecnologia, empresas em todo o mundo têm a oportunidade de expandir seus negócios atuais, com uma maior facilidade para chegar aos consumidores a qualquer momento, e em todos os lugares, graças à combinação de tecnologia e Internet. No entanto, a forma de consumo *online* está mudando o comportamento e a atitude dos consumidores. À medida que os consumidores estão mudando seus hábitos de compra, de alimentos, de vestuário, de produtos de higiene e beleza, as organizações continuarão a ter preferências consistentes se oferecerem facilidade de acesso.

O desenvolvimento e a proliferação de *smartphones* facilitam o atendimento das necessidades dos consumidores dessa área ao oferecer ferramentas e plataformas de *software*, que tornaram os aplicativos de entrega de alimentação populares e muito procurados pelos clientes. Velocidade na realização da operação, e qualidade no recebimento do produto são sempre importantes. Entender os fatores que influenciam as intenções e o comportamento de uso dessas tecnologias nesses segmentos de negócio é primordial para se entregar uma experiência de valor aos consumidores e aos empreendedores.

Estudos mais aprofundados sobre ferramentas de entrega de alimentos identificaram vários fatores que influenciam a aceitação e uso desses métodos. Um desses estudos é sobre pedidos de alimentos *online* conduzidos por Alagoz e Hekimoglu (2012), que constatou que fatores como utilidade, inovação e confiança moldam as atitudes dos clientes em relação a essa tecnologia. Outro é o caso chinês, conduzido por Cho et al. (2019), que encontraram valor percebido nas atitudes do consumidor em relação aos serviços de entrega. Os níveis da confiabilidade, do *design* e da autenticidade de um produto afetam significativamente o consumo.

Alagoz e Hekimoglu (2012) e Cho et al. (2019) destacam que, para determinar os fatores que influenciam a decisão de um cliente de usar um produto ou tecnologia, é importante primeiro entender qual é a finalidade do uso. Esse conceito é oriundo tanto do modelo proposto pela Teoria da Ação Racional (TRA), apresentado por Hill et al. (1977), quanto pela Teoria do Comportamento Planejado (TPB), concebida por Ajzen (1991). Em ambos os casos, a intenção é o principal fator que influencia as pessoas a se engajarem em determinados comportamentos. Os objetivos são analisados para determinar os fatores motivadores que influenciam o comportamento e representam o que as pessoas estão dispostas a tentar e quanto

esforço estão planejando exercer para realizar o comportamento. Nesse sentido, a intenção comportamental refere-se à possibilidade de o indivíduo realizar determinada ação, afetando assim o comportamento de uso, que está associado ao comportamento de consumir determinado produto ou tecnologia. A partir da definição desses construtos, apresenta-se a primeira hipótese desta pesquisa: a intenção de uso afeta o comportamento de uso.

Ao explorar os antecedentes diretamente relacionados à intenção e ao comportamento de uso de tecnologias de aplicativos, existem fatores considerados importantes para os consumidores enquanto estes constroem um relacionamento com as tecnologias. Como, por exemplo, o entendimento que os negócios têm sobre os interesses do consumidor, como a priorização da rapidez e a facilidade de encontrar informações que os consumidores precisam com ferramentas de melhor usabilidade, como os aplicativos de redes sociais, por exemplo. O que é comum é que as pessoas tenham um grande desejo de inovação e estejam dispostas a experimentar novos modelos, se interessando pelos resultados que advêm do uso da tecnologia. Em contraste, aqueles com baixo nível de inovação concentram-se no esforço, na facilidade de uso e no jogo da tecnologia.

1.3 Influência da pandemia no uso de aplicativos de *delivery*

O setor de *delivery* se apresentou como uma das principais alternativas para consumidores e comerciantes durante a pandemia de Coronavírus. Os pedidos de farmácias, restaurantes e supermercados aumentaram de maneira exponencial. Durante a pandemia, com a população que ficou isolada em casa e os negócios que ficaram fechados para evitar a propagação da Covid-19, as empresas utilizaram veículos de entrega para continuar operando. Durante a pandemia, o setor de *delivery* se apresentou como uma das principais alternativas para consumidores e comerciantes. Foi significativo o crescimento de diversos segmentos, como farmacêutico e alimentício.

No Brasil, a influência da tecnologia digital na dieta das pessoas aumentou com a pandemia de Covid-19. Com isso, o cenário de trabalho e lazer mudou e os brasileiros passaram a ficar mais em casa, fazendo uso maior do ambiente digital de consumo de alimentos. O ambiente alimentar digital pode ser entendido como a interface entre o consumidor e o alimento no contexto de realização pelo digital, principalmente no serviço de *delivery*. Considerando o grande aumento na utilização desses serviços, os estilos de vida tornaram-se mais digitais. Pesquisas sobre uso de mídia e comportamento do consumidor mostram que até março de 2021, cerca de 80% dos brasileiros com *smartphones* preferiam usar aplicativos de entrega para pedir comida. Essa taxa era de 7% em março de 2017 (HORTA, 2019).

Em resposta a esse aumento, os brasileiros se acostumaram com esse novo ambiente alimentar, o digital. A indústria de alimentos ultraprocessados alterou a sua forma de fazer propaganda de seus produtos como *pizzas*, lanches e outros tipos de alimentos, que estão entre as categorias mais pedidas nas plataformas *online*. Assim, o ambiente digital de entrega de alimentos pode ser entendido como um pântano de alimentos, pois concentra maior parte de alimentos não saudáveis em relação a alimentos saudáveis (HORTA, 2021).

O crescente acesso à internet, dispositivos móveis inteligentes, e uma cultura digital mais difundida são fatores que contribuem para a popularização dos aplicativos de *delivery* de comida no Brasil. A proporção de usuários de aplicativos de entrega também é maior em adultos jovens, população das classes A e B, na região

de maior concentração de renda, que residem na região Sudeste. Antes mesmo da pandemia, empresas de entrega de alimentos investem em medidas para diversificar sua base de clientes, como o programa que digitalizou o prato feito e uma porção de legumes em cidades brasileiras, entregando marmitas com preços a partir de R\$ 10,00 (BOTELHO; CARDOSO; CANELLA; 2020).

Além do aumento das vendas, esta iniciativa teve como objetivo atrair mais clientes das categorias econômicas C e D. Os eventos ao vivo, eventos *online* promovidos nas redes sociais apoiados pelas empresas de alimentos, melhoram a visibilidade dos serviços e fornecem conteúdo de entretenimento para os brasileiros assistirem em casa, incentivando o consumo de bebidas alcoólicas e outros alimentos.

2 Referencial Teórico

A solução proposta para resolver o problema de pesquisa escolhido foi o desenvolvimento do módulo de *backend* que suporte a operação de um sistema de *delivery*, e o protótipo de alta resolução do módulo de *frontend* de um aplicativo para compra de alimentos ou de bebidas de forma *online*, pretendendo atender lanchonetes, restaurantes, e outros comércios de alimentos e bebidas que sofreram com o impacto da pandemia. Podendo também um comerciante operar apenas *online* (sem loja física) vendendo seus produtos pela plataforma de *delivery*.

Mesmo com a amenização da pandemia, existem diversas pessoas que gostam de fazer as compras de refeições em aplicativos de *delivery* devido à comodidade de sua refeição ser entregue no local desejado. Seja por motivo de trabalho, caso a pessoa não possa se deslocar até um estabelecimento de vendas, ou uma pessoa que prefira receber suas refeições no conforto do seu lar.

O desenvolvimento dessa plataforma de *delivery* tem como propósito suprir algumas dores que usuários sinalizam ao usar plataformas existentes no mercado.

Neste referencial teórico, são mostradas as ferramentas utilizadas durante o desenvolvimento da plataforma, apresentando uma breve descrição de como funcionam, e como foram usadas no decorrer do desenvolvimento.

2.1 Docker

O Docker é uma coleção de produtos oferecidos como Plataforma Como Serviço (PAAS do termo em inglês *Platform As a Service*) que usam recursos de nível de sistema para entregar *software* em pacotes chamados contêineres. Os contêineres são isolados uns dos outros e contêm seus próprios softwares, bibliotecas e arquivos de configuração. Todos os contêineres são executados por um único *kernel* do sistema operacional e, usam menos recursos do que as máquinas virtuais (DOCKER, 2022).

Outra característica importante do Docker é que o *kernel* da máquina *host* é compartilhado com a máquina virtual ou o *software* em execução, capaz de combinar o aplicativo e suas dependências em um contêiner virtual que pode ser executado em qualquer distribuição Linux. Isso permite ter mais flexibilidade e portabilidade sobre onde o aplicativo pode ser executado, seja *on-premise*, nuvem, nuvem privada, etc.

Visando estabelecer um ambiente de desenvolvimento sólido, em que se tem uma maior facilidade de manipulação e centralização de ferramentas e

tecnologias, todos os serviços utilizados no desenvolvimento e também operacionalização de todo projeto, foram baseados no Docker.

2.2 Laravel

Laravel é um *framework* para linguagem de desenvolvimento PHP, livre e *open-source* criado para o desenvolvimento de sistemas web que utilizam o padrão *Model, View, Controller* (MVC). Algumas características englobadas ao Laravel são uma sintaxe simples e concisa, um sistema modular com gerenciador de dependências dedicado, várias formas de acesso a banco de dados relacionais, assim como muitas ferramentas úteis para auxiliar no desenvolvimento e manutenção do sistema (MELO, 2020).

A utilização do *framework* foi em todo contexto do *backend* do projeto, para estabelecer a comunicação com o *frontend* por meio de rotas de APIs. Essa característica foi considerada na escolha da ferramenta, bem como a alta segurança que o *framework* oferece, além do tempo de resposta das requisições feitas entre a comunicação do *frontend* com o *backend*.

2.3 Mysql

MySQL é um sistema de gerenciamento de banco de dados relacional (RDBMS – *Relational Database Management Systems*) com um modelo cliente/servidor. Um RDBMS é um *software* ou serviço de código aberto usado para criar e gerenciar bancos de dados com base em um modelo relacional (GÓIS, 2021).

Considerando o modelo de negócio, estabelecido no projeto, o uso de um banco de dados relacional se tornou necessário, pois fornece um serviço com maior estabilidade e disponibilidade. Outro ponto atraente e importante a ser mencionado é sua flexibilidade e escalabilidade. Por se tratar de armazenamento de alguns dados sensíveis de usuários, o MySQL fornece uma alta segurança de armazenamento dos dados, resultando em alta performance da aplicação.

2.4 PHPmyadmin

É um aplicativo desenvolvido em PHP para administração do MySQL pela Internet. A partir deste sistema é possível criar e deletar bancos de dados, criar, deletar e modificar tabelas, inserir, deletar e modificar campos, executar código SQL e gerenciar campos chave. O phpMyAdmin é usado por desenvolvedores da Web que geralmente desejam gerenciar bancos de dados. Normalmente, o phpMyAdmin é tratado como uma ferramenta obrigatória em quase todos os *hosts* da web (PHPMYADMIN, 2022).

O serviço do phpMyAdmin faz parte do ambiente de desenvolvimento do projeto, que foi englobado na containerização de serviços desenvolvida com Docker, conforme citado. Esta ferramenta contribui com todo gerenciamento da base de dados e também sua manipulação sempre que necessário, fornece uma total segurança de acesso das bases de dados controladas por meio de autenticação, evitando o acesso indevido de pessoas que não sejam autorizadas.

2.5 PHP

O PHP é uma linguagem com ênfase no Back End, ou seja, principalmente em scripts do lado do servidor. Portanto, você pode fazer qualquer

coisa que outro programa CGI pode fazer, como coletar dados de formulários, gerar páginas com conteúdo dinâmico ou enviar e receber cookies. Mas com ele pode-se fazer muito mais (MARQUES, 2022).

A linguagem definida para desenvolvimento do *backend* do projeto foi o PHP, sendo originalmente utilizado para desenvolvimento de aplicações presentes e atuantes no lado do servidor. Como o uso dela é independente, visa-se uma melhor consistência transacional dos dados entre a comunicação que é integrada com o *frontend* em ReactJS, oferecendo estabilidade confiável para a aplicação e também é aberta para aplicação de *patterns* de código (como, por exemplo, o padrão aplicado foi o SOLID) e há sempre suporte para as versões mais recentes, sendo confiável, segura, estável e oferece uma boa manutenibilidade para o código da aplicação.

2.6 ReactJS

O ReactJS foi criado pela equipe do Facebook em 2011 para otimizar a atualização e sincronização de atividades nos *feeds* de notícias da rede social, incluindo *chats*, *status*, listas de contatos e muito mais. ReactJS é uma biblioteca JavaScript para criar interface de usuário (REACT, 2022).

No início, todas essas atividades, chamadas de estados, tinham uma descrição muito complexa. Com o ReactJS, essa descrição fica mais simples, assim como a conexão entre HTML, CSS e JavaScript e todos os componentes da página.

Por sua eficiência, o ReactJS foi incorporado à interface de outras redes sociais do grupo, como o Instagram, nos anos seguintes, e em 2013 abriu seu código para a comunidade, ganhando popularidade. Hoje o ReactJS é uma das bibliotecas JavaScript mais conceituadas do mercado.

O uso do ReactJS é incentivado por ser flexível e poder interagir facilmente com outras bibliotecas e *frameworks*. ReactJS é JavaScript em primeiro lugar e, claro, é uma das melhores ferramentas para desenvolvimento com JavaScript do mercado. Como a comunicação foi feita utilizando rotas de API, este foi o *framework* adotado para uso do *frontend*.

2.7 JavaScript

JavaScript (JS) é uma linguagem de programação de uso geral adotada principalmente no desenvolvimento Web. Mais especificamente, o Javascript atua na programação *frontend* - a parte visual de um aplicativo. Neste caso, o Javascript é usado com outras duas linguagens iniciais, HTML e CSS (ROVEDA, 2022).

No entanto, para entender a força da linguagem, o Javascript também pode ser usado no *backend* de um aplicativo, por exemplo, para processar dados de uma tabela de um banco de dados.

Ponto importante a salientar: Javascript não é Java! Javascript e Java são linguagens muito diferentes, mas muitas pessoas confundem os nomes. Até o nome Javascript foi lançado como uma jogada de marketing por conta da popularidade do Java na década de 1990.

2.8 NodeJS

NodeJS é um ambiente de execução JavaScript que permite que aplicativos desenvolvidos na linguagem JavaScript sejam executados de forma autônoma, independente do navegador. Com ele, se pode criar praticamente qualquer

tipo de aplicação Web, desde servidores para sites estáticos, dinâmicos, até APIs e sistemas baseados em microsserviços (PESSÔA, 2022).

Foi criado em 2009 pelo engenheiro de *software* Ryan Dahl como substituto do Apache HTTP Server, o servidor Web mais popular da época. Dahl criticou algumas das limitações do Apache, como seu relativamente baixo suporte multi-conexão, e tentou resolver esse problema com uma abordagem centrada em eventos e execução assíncrona, sendo uma abordagem mais responsiva em comparação com servidores tradicionais e opções que consomem menos recursos.

Atualmente, o NodeJS é mantido pela OpenJS Foundation, criada em 2019 para promover o uso de JavaScript e tecnologias relacionadas.

2.9 Trello

O Trello é uma ferramenta de colaboração que organiza projetos em quadros e insere listas de tarefas a serem feitas individualmente ou em equipe. Cada lista recebe um cartão com uma descrição, uma data de vencimento e uma meta a ser concluída (TRELLO, 2022).

Os quadros podem ser compartilhados com qualquer pessoa com uma conta do Trello, e vários usuários podem ser adicionados aos cartões. Marcar alguém em um cartão não apenas garante que o membro da equipe seja responsável pela tarefa em questão, mas também permite que todos os outros membros do quadro saibam o que estão fazendo no momento.

O Trello também possui um sistema de rotulagem de cores. Além de ajudar a categorizar a atividade, o recurso também facilita a filtragem de *Cards* durante as buscas na plataforma. São dez opções de cores no total, que podem ganhar significados diferentes dependendo do contexto do quadro.

Todas essas características, além da facilidade de uso e o fato de ser gratuito, foram consideradas ao escolher utilizar essa ferramenta para auxiliar o gerenciamento do projeto executado pela equipe.

2.10 LucidChart

O LucidChart é um aplicativo de diagramação baseado na web que permite aos usuários colaborar visualmente no desenho, revisão e compartilhamento de gráficos e diagramas e melhorar processos, sistemas e estruturas organizacionais (LUCIDCHART, 2022).

Apesar das limitações da ferramenta por não ser gratuita, foi utilizada para definir e construir os diagramas funcionais do projeto, modelando os fluxogramas e requisitos levantados para a aplicação.

2.11 Figma

O Figma é um editor de gráficos vetoriais *online* que se concentra na prototipagem de interfaces gráficas e estruturas de design de experiência do usuário (GARRETT, 2021).

O grande diferencial do Figma é a proposta totalmente *online* que facilita a colaboração, permitindo que uma equipe de diferentes profissionais em diferentes localidades veja e trabalhe nas alterações de um projeto em tempo real.

2.12 GitHub

GitHub é um recurso de hospedagem de código para controle de versão e cooperação. É um *software* majoritariamente usado por desenvolvedores, pois permite um alojamento prático de código-fonte e arquivos em nuvem (GITHUB, 2022).

Permite que vários membros do mesmo time executem o trabalho de forma simultânea em um projeto, cada um fazendo a sua versão. Todos os integrantes da equipe têm a mesma chance de apresentar o trabalho, que passa pela aprovação do líder posteriormente ou também de forma individual. Esses arquivos ficam disponíveis para quem receber permissão, em qualquer lugar que possua internet, sendo essencial agora em que o trabalho remoto é o mais comum.

2.13 WebPack

É um conjunto de código para projetos web com foco em módulos de aplicação. Os módulos do programa são estáticos e migram para o JavaScript moderno. Uma vez que todas as aplicações são processadas, pode ser criado um gráfico que representa cada módulo e suas dependências. Isso gera um ou mais pacotes (NOLETO, 2021).

Em geral, o WebPack pode ser pensado como uma ferramenta de construção onde todos os seus ativos, incluindo JavaScript, imagens, fontes e CSS, são roteados para um gráfico de dependência. Isso permite que se use o código-fonte para selecionar arquivos locais e decidir o que fazer com eles.

3 Projeto (Engenharia de *Software*)

Para o desenvolvimento desse *software*, foram criados artefatos sugeridos pelas boas práticas de Engenharia de *Software*, que visam melhorar a experiência dos desenvolvedores e para uma melhor implementação e manutenção do código.

Itens esses que foram utilizados antes do desenvolvimento do código, durante o desenvolvimento, e também serão utilizados para manutenção do sistema.

3.1 *Business Process Model and Notation* (BPMN)

BPMN é usado para modelagem orientada a objetos, ajuda a organizar os processos de negócios para que nada se perca no caos diário e cada operação seja realizada da forma mais simples possível (TOTVS, 2022).

A modelagem de processos é uma solução que reduz custos para que a empresa possa operar de maneira otimizada com base em seus objetivos e fornece uma visão geral transparente de como o trabalho está sendo feito.

É uma representação gráfica composta por ícones que simbolizam um processo tecnológico. Isso significa que os processos podem ser mapeados a partir deste símbolo. Cada ícone é uma etapa do processo que está sendo modelado. Ele auxilia no processo de mapeamento usando símbolos padronizados em representações gráficas. Muito semelhante a um fluxograma tradicional, mas difere em termos de informação. Fornece informações mais completas devido ao modo simbólico.

Sua aplicação é semelhante ao aprendizado de programação em uma linguagem de desenvolvimento: primeiro é necessário aprender o básico. O BPMN é

uma forma de aumentar a eficiência das operações do negócio, otimizar resultados e, assim, reduzir o consumo de recursos. Aumenta a transparência estrutural e estratégica, permitindo uma visão abrangente de como as coisas funcionam na organização. Também incentiva a resposta competitiva e o interesse pela empresa.

3.2 Plano 5W1H

Plano 5W1H é uma ferramenta de levantamento e análise de características e contexto do problema a ser resolvido. Através de tópicos que caracterizam questões a serem explicitadas e aprofundadas, possibilita uma visão mais detalhada dos elementos presentes no problema. Neste projeto, as especificações de cada item são apresentadas a seguir.

What: ter um serviço de *delivery* que atenderá, inicialmente, restaurantes e conveniências em um só lugar. Uma plataforma com todo o serviço centralizado, dentre eles, cadastro do estabelecimento para utilização, cadastro de cardápio, formas de pagamento, entrega, gestão de pedidos e pagamentos

Why: visando sempre atender às necessidades dos clientes (consumidor e empreendedor), propomos uma plataforma de *delivery* com uma grande diversidade de segmentos de negócio, por exemplo, restaurantes, supermercados, conveniências, entre outros. Com a comodidade do consumidor fazer o pedido de qualquer estabelecimento cadastrado na plataforma e receber no conforto de sua casa.

Where: inicialmente será apenas por dispositivos móveis que possuem o aplicativo instalado e que possuam acesso à internet

When: poderá ser usado em qualquer lugar, contando apenas que o usuário possua acesso à internet. Desde uma pessoa que esteja em seu horário de intervalo e necessite de uma refeição, até uma pessoa que queira um doce, que queira receber o mesmo em sua residência

Who: desenvolvedores/estudantes responsáveis pelo projeto

How: será necessário fazer *download* do aplicativo, e necessitará de acesso à internet, e também de ter uma conta para receber seus produtos.

3.3 Diagrama de classe UML

Um diagrama de classes representa objetos ou grupos de objetos que compartilham valores e comportamentos. Uma classe, ou instância de uma classe, é uma instância comum em diagramas UML. Uma classe identifica os atributos, operações, relacionamentos e estudos de uma instância ou objeto da classe. Cada objeto que instala uma classe tem seu próprio conjunto de atributos. Em diagramas de classe, função, recurso e processo, as classes podem representar coisas como tabelas de banco de dados e páginas da Web para aplicativos, bem como objetos de negócios.

O nome de uma classe, que pode ser derivado da expressão do sistema modelo, indica sua função ou uma de suas funções no sistema. O editor gráfico exibe a classe como um retângulo tridimensional.

3.4 Diagrama de Estado

Representa estados ou situação que um objeto se encontra no decorrer da execução de seu sistema, partindo de um estado inicial para um estado final, estados que são passados por transições (JUNIOR, 2022).

4 Ferramentas e Métodos

Neste capítulo serão esclarecidas todas as ferramentas e também métodos que foram utilizados durante o desenvolvimento deste artigo.

4.1 Figma

Foi utilizado o Figma para desenvolvimento dos protótipos de baixa fidelidade e para o de alta fidelidade, pois atualmente é uma das ferramentas mais utilizadas para desenvolvimento de projetos de UX, e com uma interface de fácil entendimento. Foi utilizada a versão gratuita, pois durante o desenvolvimento dos protótipos, todos os itens presentes nesta versão satisfizeram as funcionalidades especificadas, e obteve-se a prototipação esperada.

O Figma serve para desenvolver testes de usabilidade antes do início do desenvolvimento do *frontend*, para se caso houver alguma correção durante a prototipação de alta fidelidade, haja menos retrabalho, tendo assim menos prejuízo, pois se for feita a correção após o desenvolvimento, o prejuízo será muito maior.

5 Desenvolvimento

Neste capítulo é mostrado como foi definida a comunicação do usuário com a plataforma de *delivery*. Nesta seção são descritas técnicas de aplicação de desenvolvimento de protótipos de tela sobre como será a interface da aplicação.

5.1 As cores

Foram utilizadas cores que compõem a paleta entre o azul e o verde, pois cada uma tem um significado, e o usuário ao visualizar a interface, sentirá o que cada cor transmite. A cor azul é frequentemente usada para descrever harmonia, calma e serenidade. Tons azulados ajudam a reduzir a pressão arterial e a circulação sanguínea, para o *delivery* essa cor indica a serenidade que em breve o pedido será entregue, e que o usuário pode ficar calmo, que em breve o seu desejo será saciado pelo seu pedido. Já o verde é a cor mais associada à aparência e sentimento de esperança, mas está associada a conceitos como saúde, dinheiro, poder e juventude. Por ser uma cor tranquila, demonstra a esperança que sua vontade será satisfeita, e que quando sentir novamente vontade de experimentar algo, saberá onde recorrer para saciar suas vontades.

5.2 Método 60/30/10

O método 60-30-10 ajuda a distribuir a renderização de cores de forma eficaz na interface. Apesar de ser um método principalmente usado na arquitetura e design de interiores, a técnica pode ser usada em produtos digitais, como sites e aplicativos, sendo possível utilizar essa relação na construção de interfaces, pois usamos a cor para destacar muitos elementos importantes da tela (VILLAIN, 2022).

É importante entender que 60-30-10 é uma relação de porcentagens, descrevendo quais cores estarão presentes na página. Esses valores são imaginários, pois não temos ideia se realmente usamos, por exemplo, 60% dessa cor na interface, mas os detalhes que eles, principalmente nesses princípios e nesse sistema, servem para entender a proporção que a cor que ocupará. Assim, essas porcentagens significam: 60% é a cor máxima, que terá mais presença nas interfaces. 30% é a segunda cor, que servirá de contraste com a cor de destaque e trará harmonia. 10% é uma cor de destaque aplicada aos elementos que precisam ser enfatizadas, como botões, links e ícones. No processo de *wireframe*, ainda construindo esses gráficos com baixa ou média fidelidade, sem a aplicação de cores se usará a regra 60-30-10 na íntegra para alcançar harmonia de composição.

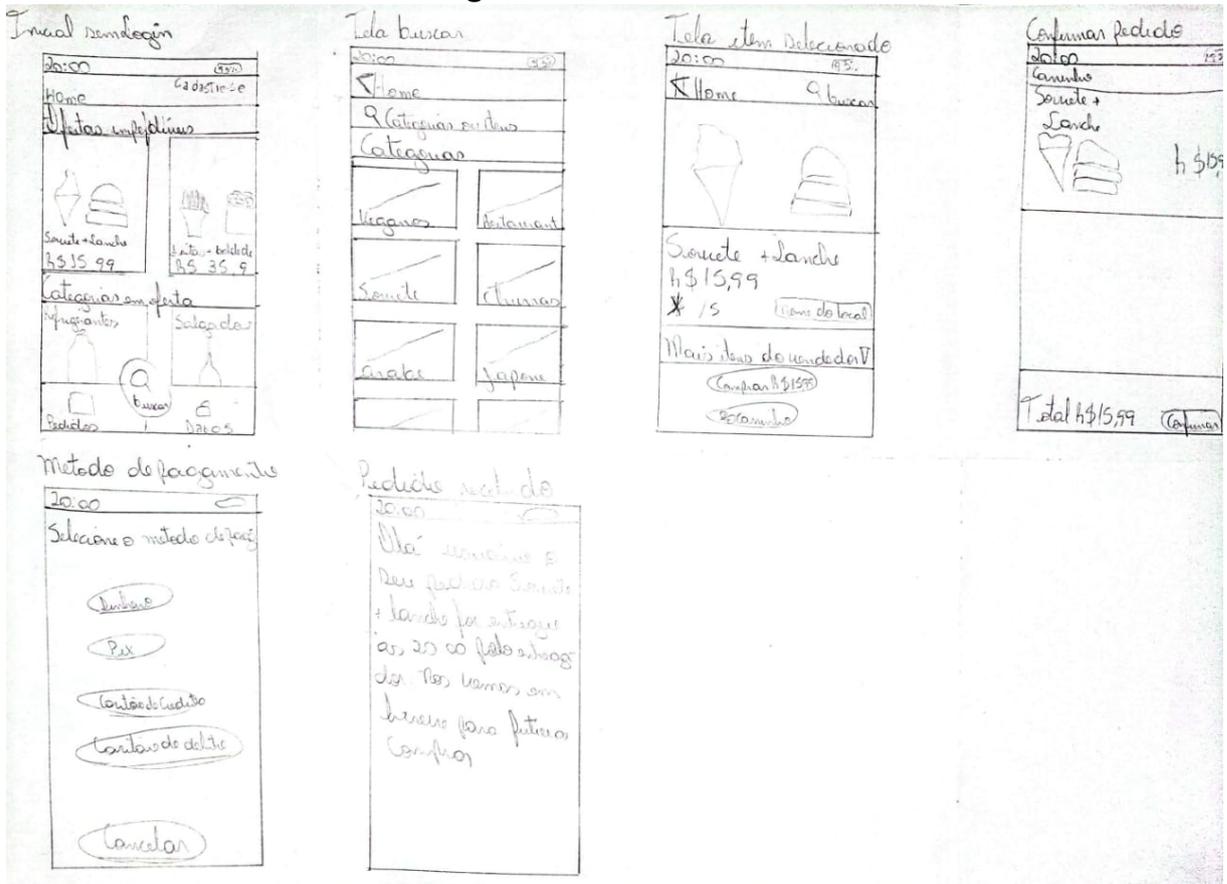
60% será a cor próxima à cor branca, 30% será um verde bem claro, indicando locais para navegar na plataforma, e 10% será um verde-azulado, indicando a cor que será mais forte. Os botões de confirmação e cancelamento, será nas cores verde, e na cor vermelho, sendo como elementos visuais para sinalizar ao clicar em um dos botões, o que estará ocorrendo.

5.3 Rabiscoframe

O rabiscoframe é a ideia inicial de como será o produto, sendo esquematizadas as telas de operação do aplicativo, proporcionando uma ideia de como será a aplicação.

Na Figura 1 é apresentado o desenvolvimento dessa parte do protótipo. E para construção do rabiscoframe, foi utilizada uma folha de papel A4, uma lapiseira e borracha. Por se tratar de um esboço, foi feito de maneira rápida, para apenas se ter uma noção do início do mapeamento de telas da plataforma.

Figura 1 - Rabiscoframe



Fonte: os autores

5.4 Wireframe

O *wireframe* é um esboço melhorado em comparação ao rabiscoframe, de como será a visualização do aplicativo. Representa uma figura estrutural de como serão as telas da plataforma e sua navegação (SOUZA, 2020).

Nas imagens (Figuras 2 a 7) são mostradas, em um diagrama de baixa resolução, como se propõe a construção em baixo nível de cada uma das telas. Neste processo não se define, ainda, nenhuma cor, ícone ou outros elementos da interface, sendo definidos a distribuição dos elementos nas telas e o fluxo de navegação entre elas.

Figura 2 - Tela inicial



Fonte: os autores

Figura 3 - Tela de Busca



Fonte: os autores

Figura 4 - Tela de item selecionado



Fonte: os autores

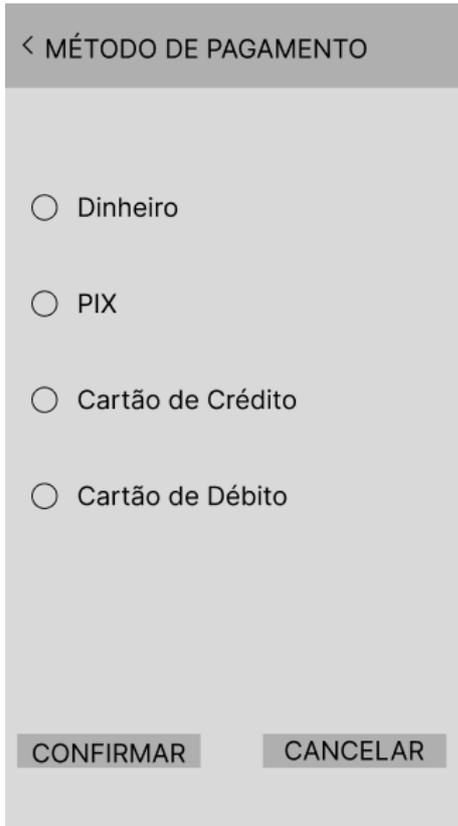
Figura 5 - Confirmar pedido



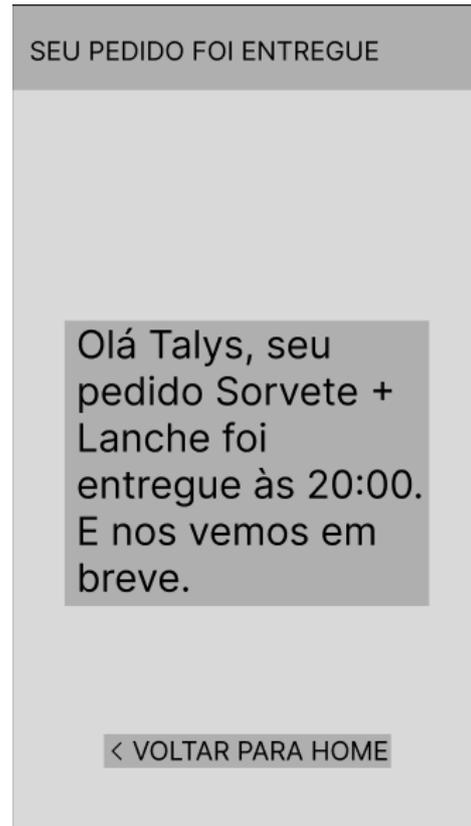
Fonte: os autores

Figura 6 - Método de pagamento

Figura 7 - Finalização do pedido



Fonte: os autores



Fonte: os autores

5.5 Arquitetura *backend*

Para o desenvolvimento do *backend*, foram utilizados métodos e tecnologias para melhor integrar os dados e informações entre clientes e lojistas em bases de dados, não seria muito seguro e performático armazenar os dados de todos os estabelecimentos e clientes em apenas uma base de dados. Aplicando uma arquitetura onde é possível ter um maior ganho de performance para a aplicação, segurança e integridade dos dados, foi utilizado pelo qual o modelo de arquitetura denominado de *Multi-Tenancy*, no ato do cadastro de cada estabelecimento é criado um banco de dados individual para cada um deles, proporcionando tratar com individualidade cada base.

Para melhor entendimento desse modelo de arquitetura, pode-se interpretá-la de forma em que cada loja cadastrada se trata de um inquilino, e cada base de dados é a morada, onde são armazenados os dados separadamente entre lojas.

A base de dados é integrada diretamente com a *API*, que é a principal responsável pela conexão e comunicação entre as bases de dados e por transacionar e validar todos os dados de clientes e lojistas, ela foi desenvolvida na linguagem PHP e todas as bases de dados utilizam os serviços do *MySQL*.

Na arquitetura de desenvolvimento da *API*, foi aplicado o conceito conhecido de *Design Pattern*, entre os tipos existentes foi aplicado o modelo de *MVC (Model, View, Controller)* separando toda a aplicação em camadas, separando hierarquia, tarefas de cada componente e isolando funcionalidades. Também foram aplicados conceitos de padrões de escrita de código, como *Clean Code*, *PSRs*, oferecendo maior segurança e maior facilidade na manutenibilidade do código.

6 Resultado

Nessa seção é apresentado o protótipo de alta resolução, para se ter uma visualização de como o usuário vivenciará o uso da plataforma de *delivery* para a realização futura de testes de usabilidade.

6.1 Protótipo de Alta Resolução

Protótipos com Alta Fidelidade têm mecanismos de respostas rápidas durante os testes. Estes modelos costumam se aproximar mais do software que os usuários terão contato. Significa que os usuários podem mexer nos protótipos como se realmente estivessem mexendo na aplicação de forma real (PERNICE, 2016).

Nas Figuras 8 a 17 são mostradas as telas do protótipo de alta resolução, tendo sido utilizada a plataforma de desenvolvimento de protótipos Figma, para o usuário conseguir mexer no protótipo como se estivesse mexendo na plataforma já desenvolvida e funcional.

Figura 8 - Splash screen



Fonte: os autores

Figura 9 - Tela Inicial sem login



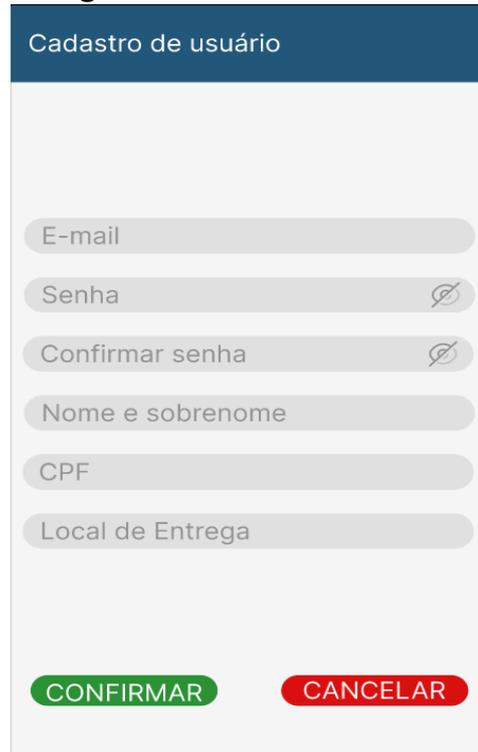
Fonte: os autores

Figura 10 - Tela de Login



Fonte: os autores

Figura 11 - Tela de Cadastro



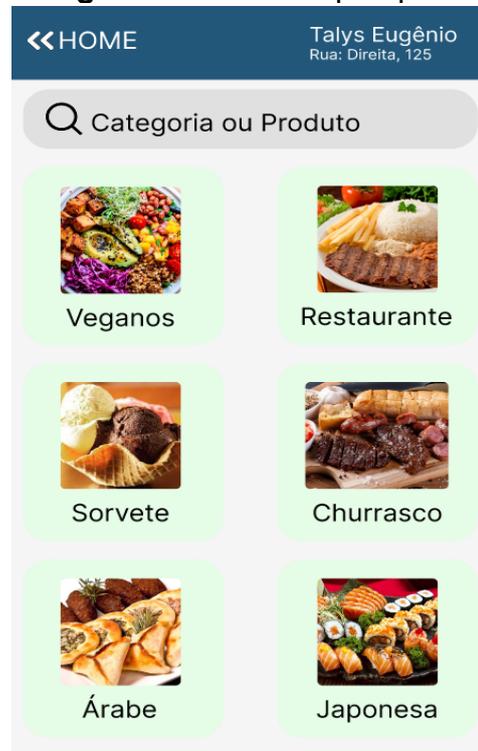
Fonte: os autores

Figura 12 - Tela Inicial logado



Fonte: os autores

Figura 13 - Tela de pesquisa



Fonte: os autores

Figura 14 - Tela de item selecionado



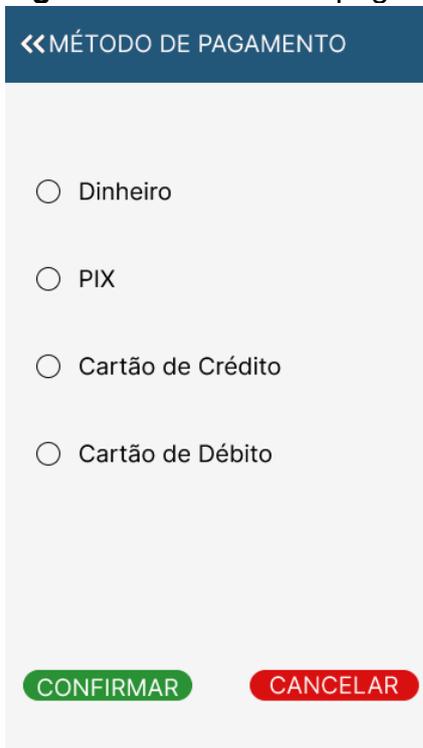
Fonte: os autores

Figura 15 - Confirmar pedido



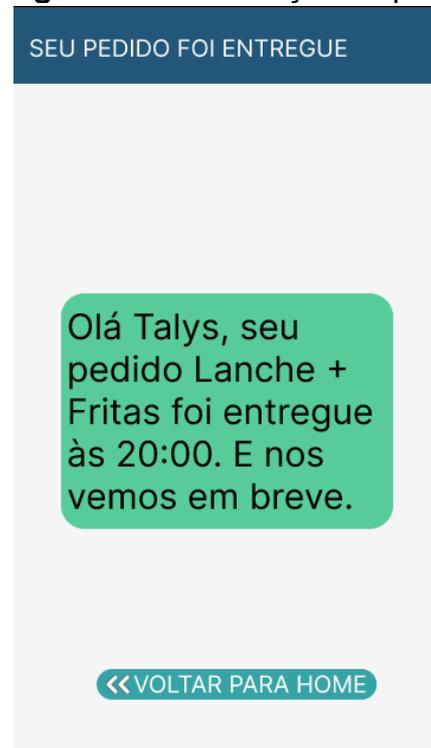
Fonte: os autores

Figura 16 - Método de pagamento



Fonte: os autores

Figura 17 - Finalização do pedido



Fonte: os autores

7 Considerações finais

Como resultado deste trabalho, concluímos que o objetivo de apresentar uma plataforma que faz a junção entre usuário/cliente e usuário/restaurante/conveniência, com a principal atividade sendo a exibição de alimentos e a venda por parte de restaurantes e conveniências, e o consumo destes pelos clientes foi alcançado.

No *backend*, foram implementadas todas as principais funcionalidades. No *frontend*, por necessidade de redefinir o escopo das atividades de projeto, foram desenvolvidos os estudos da interface, aplicando conceitos e fundamentos de Experiência do Usuário (*User Experience - UX*).

Apesar de ser citado no referencial teórico itens que seriam usados no desenvolvimento do *frontend*, sendo eles o ReactJS, o JavaScript, NodeJS e WebPack, por questões de escopo e adequação ao tempo de entrega, foi alterado o desenvolvimento para a prototipação da interface.

Após a apresentação deste artigo, serão estudadas formas de acessibilidade para pessoas que necessitem deste recurso, como exemplo deficientes visuais que necessitam de interfaces diferentes de uso da plataforma; deverá ter a opção de comando por voz, que no cadastro do usuário oferecerá como forma de acessibilidade. Também será estudada a possibilidade de acessibilidade por letras maiores, por contraste de cores para pessoas com daltonismo, e outras formas de acessibilidade. Após estudo sobre os demais concorrentes (*benchmarking*) nesse segmento, foi notado que nenhuma dessas plataformas em suas versões mobile oferecem versão com acessibilidade.

Quanto ao módulo de promoções, será feito um acordo com os parceiros de plataforma, para ser dividido os custos de cada item que está em promoção. A proposta de promoção é que seja firmado com um parceiro, que quanto maior a porcentagem de promoção que inserir, maior será a relevância desse anúncio, também será usada a classificação do item e do produto como média ponderada para ser dada relevância.

Para marketing e divulgação da plataforma será usado o método 4p do *marketing*: Produto, Preço, Praça, Promoção.

Referências

ALAGOZ, S. M; HEKİMOGLU, H. A study on TAM: Analysis of customer attitudes in online food ordering system. **ScienceDirect**

Disponível em:

<https://www.sciencedirect.com/science/article/pii/S1877042812036361>. Acesso em: 13 abr. 2022.

BOTELHO, L.V; CARDOSO, L.O; CANELLA, D.S. **COVID-19 and the digital food environment in Brazil**: reflections on the pandemic 's influence on the use of food delivery apps 2020. Disponível em:

<https://www.scielo.br/j/csp/a/pX8fFSjKVQXLLwFwbhWPYJd/>. Acesso em: 20 abr. 2022.

DOCKER. **What does Docker technology add to just plain LXC?** 2022. Disponível em: <https://docs.docker.com/engine/faq/#what-does-docker-technology-add-to-just-plain-lxc>. Acesso em: 28 nov. 2022.

EBIT (2021). **Webshoppers**. 45ª edição. Disponível em: <https://company.ebit.com.br/webshoppers/webshoppersfree>. Acesso em: 11 abr. 2022.

GARRETT, F. **O que é Figma?**. (2021). Disponível em: <https://www.techtudo.com.br/listas/2021/06/o-que-e-figma-quatro-perguntas-sobre-como-usar-o-site.ghtml>. Acesso em 25 ago. 2022.

GITHUB. **Olá, Mundo - Documentação do GitHub**. Disponível em: <https://docs.github.com/pt/get-started/quickstart/hello-world>. Acesso em 25 ago. 2022.

GÓIS, A. **MySQL: o que é e como usar o sistema?**. 2021. Disponível em: <https://www.tecmundo.com.br/software/223924-mysql-usar-o-sistema.htm>. Acesso em: 24 nov. 2022.

HORTA, P.M. **Ambiente alimentar digital e a pandemia de covid-19 no Brasil: um estímulo ao consumo de alimentos ultraprocessados**. Disponível em: <https://pp.nexojornal.com.br/opiniao/2021/Ambiente-alimentar-digital-e-a-pandemia-de-covid-19-no-Brasil-um-est%C3%ADmulo-ao-consumo-de-alimentos-ultraprocessados>. Acesso em: 21 abr. 2022.

JUNIOR, F. **UML - Diagrama de Estados**. Disponível em: http://help.market.com.br/csharp/diagrama_de_estado.htm. Acesso em 25 ago. 2022.

LUCIDCHART (2022). **O Que É Fluxograma de Processo** | Lucidchart. Disponível em: <https://www.lucidchart.com/pages/pt/o-que-e-fluxograma-de-processo>. Acesso em 28 nov. 2022.

MARQUES, R. **O que é PHP: entenda de forma descomplicada**. 2022. Disponível em: <https://www.homehost.com.br/blog/tutoriais/php/o-que-e-php/#:~:text=O%20PHP%20é%20uma%20linguagem%20com%20ênfase%20no%20Back%20End,pode-se%20fazer%20muito%20mais..> Acesso em: 25 nov. 2022.

MELO ,D . **O que é Laravel?** [Guia para iniciantes]. 2020. Disponível em: <https://tecnoblog.net/responde/o-que-e-laravel-guia-para-iniciantes/#:~:text=Laravel%20%C3%A9%20um%20framework%20PHP,agilizam%20o%20processo%20de%20desenvolvimento>. Acesso em: 25 nov. 2022.

NOLETO, C. **Webpack: aprenda o que é, como funciona e como usar!**. 2021. Disponível em: <https://blog.betrybe.com/ferramentas/webpack-tudo-sobre/#1>. Acesso em 25 ago. 2022.

PERNICE, K. **UX Prototypes: Low Fidelity vs. High Fidelity**. 2016. Disponível em: <https://www.nngroup.com/articles/ux-prototype-hi-lo-fidelity/>. Acesso em 8 set. 2022.

PESSÔA, C. (2022). **Node.JS**: definição, características, vantagens e usos possíveis. Disponível em: <https://www.alura.com.br/artigos/node-js-definicao-caracteristicas-vantagens-usos>. Acesso em 28 nov. 2022.

PHPMYADMIN. **Bringing MySQL to the web**. 2022. Disponível em: <https://www.phpmyadmin.net>. Acesso em: 24 nov 2022.

REACT **.React**: Uma biblioteca JavaScript para criar interfaces de usuário. 2022. Disponível em: <https://pt-br.reactjs.org/>. Acesso em 28 nov. 2022.

ROVEDA, U. **JAVASCRIPT**: O QUE É, PARA QUE SERVE E COMO FUNCIONA O JS?. 2022. Disponível em: <https://kenzie.com.br/blog/javascript/>. Acesso em 28 nov. 2022.

SOUZA, I. **Saiba o que é UI (User Interface) e a importância dele para os clientes**. Disponível em: <https://rockcontent.com/br/blog/o-que-e-ui/>. Acesso em 8 set. 2022.

TRELLO (2022). **O que é o Trello**: conheça recursos, usos e muito mais | Trello. Disponível em: <https://trello.com/pt-BR/tour>. Acesso em 28 nov. 2022.

TOTVS, E. **BPMN**: entenda o que é a modelagem de processos de negócios, como fazer e sua importância! Disponível em: <https://www.totvs.com/blog/gestao-industrial/bpmn/>. Acesso em: 26 ago. 2022.

VILLAIN, M. **Regra 60-30-10 em UI Design**. Disponível em: <https://www.alura.com.br/artigos/regra-60-30-10-ui-design>. Acesso em: 7 out. 2022.