

LeCard: Web Aplicativo para comandas virtuais

Elias Douglas Wenceslau PORTELA¹

Lucas De Castro SANTOS²

Ely Fernando do PRADO³

Resumo: Atualmente o modelo mais usado em bares, lanchonetes e restaurantes no controle dos consumos dos clientes é o de comandas de papel, onde funcionários registram os produtos consumidos e gerenciam os gastos de seus clientes. Muitas empresas usam a forma manual deste processo utilizando papel e caneta para anotar os pedidos, todavia, alguns dos problemas encontrados com este método é o risco de perdas da comanda, falta de controle e agilidade na qual os pedidos transitam pelo estabelecimento. Assim sendo, o objetivo deste trabalho é desenvolver uma aplicação *Web Mobile*, denominada LeCard, que visa automatizar os processos de uma comanda e é composto pela parte em que os clientes poderão acompanhar os produtos consumidos, os pedidos e o cardápio, a parte do garçom com função de gerenciar as comandas além das APIs para integração com sistemas de caixas e ERPs. Apresentamos as tecnologias empregadas na construção do sistema e a metodologia dos processos, a começar pela fase de levantamento de requisitos até a implantação e por fim, exibimos uma pesquisa proposta aos funcionários na empresa consultada visando obter os resultados adquiridos com a utilização da comanda virtual LeCard.

Palavras-chave: clientes; pedidos; comanda;

Abstract: *Currently the most used model in bars, snack bars and restaurants in control of customers' consumption is the use of cards, where employees record the products consumed and manage the expenses of their customers. Many companies use the manual form of this process using paper and pen to write down the orders, however, some of the problems encountered with this method is the risk of loss of the cards, lack of control and agility in which the orders transit through the establishment. Therefore, the objective of this work is to develop a Web Mobile application, called LeCard, which aims to automate the processes of a card and is composed by the part in which the customers can follow the products consumed, the orders and the menu, the part of the waiter with the function of managing the commands besides the APIs for integration with systems of cash flow and ERPs. We present the technologies used in the construction of the system and the methodology of the processes, starting with the requirements gathering phase until the implementation, and finally, we present a research proposed to the employees in the company consulted in order to obtain the results acquired with the use of the virtual card LeCard.*

Keywords: *consumer; orders; card;*

¹ Discente do curso de Bacharelado em Sistemas de Informação do Uni-FACEF

² Discente do curso de Bacharelado em Sistemas de Informação do Uni-FACEF

³ Docente do curso de Bacharelado em Sistemas de Informação do Uni-FACEF

1. Introdução

A tecnologia evolui constantemente a cada ano, e todos os dias novos sistemas para automatização de tarefas manuais estão sendo criados ou estão em plena evolução. A evolução da tecnologia por sua vez, contribuiu para o surgimento da revolução digital que por consequência mudou a forma de realizar tarefas antes feitas há mão. O setor de serviços tem se beneficiado muito com essas transformações, pois o uso de inovações contribui na transformação da empresa e resulta em mais qualidades de serviços aos clientes da mesma forma auxilia na redução dos custos de operação. Posto isto, o mercado alimentício, especialmente no setor de bares, restaurantes e lanchonetes, necessitam cada vez mais ter suas finanças estruturadas como fator essencial para poder se manter no mercado, fazendo se uso da tecnologia para alcançar este objetivo. (SEBRAE, 2017)

Em plano geral, dos estabelecimentos deste nicho, utilizam a comanda de pedidos para anotações dos consumos de seus clientes, e muitas das vezes este processo é feito de modo manual, um papel impresso em cada mesa dos clientes. Os problemas mais comuns que ocorrem quando é usada somente esta abordagem é o fato dos funcionários deixarem de anotar os pedidos, podendo ter também várias outras complicações, como, perdas de comandas e falta de controle no momento de anotar o que será enviado para a cozinha.

Em caso de perda de comanda, o consumidor é estipulado a pagar certa multa fixa, na qual pode existir grande dificuldade na fidelização dos clientes. Outro fator é o prejuízo que pode ser acarretado ao estabelecimento no caso de esquecimento em anotações de consumo e também a possibilidade de existirem funcionários que agem de má fé contra a empresa na qual trabalham, deixando de marcar os pedidos na comanda por interesses próprios.

Em razão disso, o objetivo deste trabalho é desenvolver um web aplicativo que funcione como uma comanda virtual, o qual poderá ser acessado por funcionários do estabelecimento e também pelo próprio consumidor. Através desta aplicação, poderá ser evitado o risco de perda do papel, que poderá ainda ser utilizado para controle interno, mas com uma cópia online. O funcionário da mesma forma deverá marcar os pedidos na comanda virtual para realizar o envio ao setor responsável pela preparação dos alimentos impossibilitando o esquecimento e também o controle de fraudes. O aplicativo, nomeado como LeCard, poderá ser integrado com sistemas já implementados na empresa por meio de APIs, para tanto será disponibilizado um documento com detalhes de cada requisição bem como o seu funcionamento.

2. Referencial Teórico

O desenvolvimento da aplicação utilizou-se de conceitos com grande difusão no ambiente tecnológico e como sendo uma aplicação voltada para dispositivos móveis, discorreu-se sobre sua definição em conjunto com a mobilidade digital. Posto às limitações de tamanho e memória, os sistemas operacionais móveis também se diferenciam para este tipo de software e, sendo assim, houve uma breve explicação dentre os dominantes no mercado, o Android da Google e o *IOS* da Apple. Com o advento do HTML5 e os novos recursos dos navegadores comparou-se também sobre os softwares nativos em contrapartida aos de web aplicativos. Por fim, foram apresentadas todas as tecnologias que compõem a construção da aplicação, desde as ferramentas de engenharia de software, bem como os *frameworks* e bibliotecas aderidos no desenvolvimento do projeto.

2.1 Mobilidade Digital

Mobilidade é a capacidade de dispositivos se manterem funcionais mesmo em movimento (B'FAR, 2005). Segundo B'Far (2005), com a mobilidade, os aparelhos móveis oferecem recursos e características que não encontramos em sistemas comuns, tal como, monitoramento de nível de energia e perda de dados em caso de pane, armazenamento de dados local ou remoto através de uma conexão com ou sem fio e sincronização de dados com outros sistemas. A vantagem maior e mais trivial da mobilidade, a capacidade de acessar dados em qualquer lugar e a qualquer momento (MENDONÇA, 2018).

2.2 Dispositivos móveis

Dispositivos móveis são computadores que encaixam no conceito de mobilidade, e possuem características físicas como, tela pequena, teclado em miniatura e tamanho reduzido o suficiente para ser operado com as mãos. Os dispositivos mais comuns são: *Smartphone*, PDA (*Personal digital assistant*), celular, console portátil, televisão portátil, tablets entre outros (MENDONÇA, 2018).

O ábaco, um dos primeiros computadores desenvolvidos a mais 500 A.C, pode ser considerado, por conta do tamanho e portabilidade que oferece, o pioneiro dos dispositivos móveis (B'FAR, 2005). Na era dos eletrônicos, para usar os benefícios da mobilidade precisou ser inventado dispositivos que não necessitam o uso de fio, foi então que em 1947 engenheiros da Bell Labs tiveram a ideia de criar um sistema de comunicação entre telefones sem cabos de redes, era o início da criação dos primeiros aparelhos celulares. Entretanto como tecnologia da época não era suficiente a primeira ligação real entre dois aparelhos apenas foi

efetuada em abril de 1973 (VANNI, 2018). Com o decorrer das décadas, iniciou-se um nova era de aparelhos mais compactos e inteligentes e em 1992 foi lançado o *palmtop* Newton Message PAD (*Personal digital assistant*) pela Apple na qual tinha tela sensível ao toque, memória total de 1Mb, rodava com o sistema operacional Newton OS e vinha com uma série de softwares para ajudar na organização e gerenciamento de dados pessoais, no entanto não teve muita repercussão para época (MENDONÇA, 2018). Daquele período em diante houve vários lançamentos dos mais diversos fabricantes e também a incorporação de funcionalidades em um único aparelho, como, câmera, GPS, *WIFI* entre outros, até chegar nos mais modernos dispositivos como conhecemos hoje (MENDONÇA, 2018).

2.3 Sistemas operacionais móveis

Sistema operacional é um conjunto de softwares que tem como função gerenciar os recursos de hardware, como, processador, memória, sistema de arquivos entre outras coisas de um dispositivo computacional, bem como oferecer uma interface para o usuário (VANNI, 2018). Devido às limitações de memória e processador, os dispositivos móveis usam sistemas operacionais diferentes de outras plataformas como os usados em computadores *desktops* (KRISHNA, DEVARAKONDA, 2015). Dentre os SOs mais usados para plataformas mobiles o Android da Google e o IOS da Apple são os com maior números de usuários ativos no mundo (VANNI, 2018).

2.3.1 Android

A Android Inc foi fundada em Palo Alto, Califórnia em outubro de 2003 por Andy Rubin, Rich Miner e Chris White. O objetivo até então, era desenvolver um sistema operacional para câmeras digitais, porém, a companhia mudou para o ramo de *smartphones*, visto que era um mercado recente, maior, e em plena expansão (MARKOFF, 2007). Foi adquirido pela Google em 17 de agosto de 2005, e a equipe, liderada por Rubin começou a desenvolver o Sistema operacional usando o Kernel do Linux, a Google também começou a comercializar a plataforma para fabricantes de dispositivos móveis e operadoras de telefonia móvel juntamente com a promessa de um sistema flexível e renovável e sugeriu uma parceria sólida e aberta para qualquer um estivesse disposto a participar (RABELLO, 2018).

Em 2007 foi criado o *Open Handset Alliance* (OHA) um consórcio de empresas de tecnologias lideradas pela Google com o intuito de desenvolver projetos *open-sources* (Código aberto) para dispositivos móveis. O android tornou-se o seu primeiro produto (RABELLO, 2018).

O Android permite que seus usuários instalem aplicativos, de terceiros, sendo obtido tanto pela loja oficial, a *Play Store*, quanto por outras lojas como a *Amazon Appstore* ou baixando e instalando arquivos *APK (Android Package)*. Os aplicativos nativos são desenvolvidos na linguagem de programação Java usando o kit de desenvolvimento de software do Android (SDK). O SDK compreende um conjunto de ferramentas de desenvolvimento incluindo depuradores, bibliotecas e emuladores (RABELLO, 2018).

2.3.2 IOS

Ios é um sistema operacional desenvolvido pela Apple para iPhone, iPad e iPod touch. É baseado no núcleo do MacOS, portanto sobre o sistema operacional Darwin (QADIR, 2010).

O desenvolvimento iniciou-se em 2005, com Scott Forstall assumindo a liderança do projeto. A ideia inicial era desenvolver um telefone que rodasse com o Mac OS X, e o primeiro protótipo foi chamado de P2. Para garantir o sigilo, apenas alguns funcionários fizeram parte do desenvolvimento do software (QADIR, 2010).

O sistema operacional original foi lançado em 9 de janeiro de 2007 junto com o iPhone na *MacWorld Conference and Expo*. Na primeira versão, o iOS não suportava a instalação de software desenvolvidos por terceiros, apesar disso, a Apple divulgou em fevereiro de 2008 um Kit de Desenvolvimento de Software (SDK) (QADIR, 2010).

Os aplicativos passaram a ser introduzidos com o iPhone OS 2.0 juntamente com o lançamento da *App Store*. Atualmente a IDE, *Integrated Development Environment* ou Ambiente Integrado de Desenvolvimento em português tanto para Mac e iOS é o Xcode incluindo com ele a linguagem de programação Swift que substituiu o Object-C (ANDRADE, 2017).

2.4 HTML5

HTML5 é uma linguagem de marcação com objetivo de estruturar e exibir conteúdos na *World Wide Web*, sendo a quinta e atual versão padrão do HTML. Foi publicado em outubro de 2014 pela W3C com novos elementos e novas funcionalidades que possibilitam melhores experiências e integrações se comparado com versões anteriores da linguagem (ROBBINS, 2013).

Devido falta de atualizações do HTML4, houve a criação de diversos plugins com o intuito de incrementar a linguagem, um exemplo é o Adobe Flash Player, na qual veio dar suporte a arquivos multimídias. Diversas funcionalidades como geolocalização,

armazenamento de dados do lado do cliente, notificações, acesso a câmera, informações de acelerômetro e giroscópio, animações 3D com aceleração na GPU, além de uma API nativa para o suporte a multimídias foram adicionadas nativamente apenas com advento do HTML5 (LOPES, 2013).

2.4.1 PWA

Progressive Web Application, traduzida como aplicações webs progressivas, é uma nova metodologia de desenvolvimento de software que, de maneira oposta aos tradicionais aplicativos nativos ou híbridos, utilizam recursos dos navegadores para oferecer uma aplicação, com as mesmas vantagens de um aplicativo convencional (GOOGLE DEVELOPERS, 2018). De acordo com o Google *Developers*, um PWA contém como características, a progressividade, ou seja, funciona para qualquer usuário independentemente do navegador, a responsividade de se adaptar a qualquer formato de tela, independência de conectividade com a internet para trabalhar em ambientes offlines ou com baixas conexões, segurança, descobrível, envolvente, instalável pois permite o anexo a tela inicial sem necessitar acessar uma loja de aplicativos e linkável na qual é de fácil compartilhamento por URL (GOOGLE DEVELOPERS, 2018).

2.5 Aplicativos vs Web

O HTML5 trouxe diversas novas APIs de acesso a recursos do dispositivo, entretanto, há muitas funcionalidades exclusivas que ainda não são possíveis obter acesso com a *Web*. Apesar disto, dependendo do escopo do projeto, é possível usar o navegador para criar aplicações com as mesmas funcionalidades de um aplicativo. As vantagens de usar esse método é a portabilidade, ou seja, capacidade de usar em diferentes plataformas e a não obrigatoriedade de instalar um software no seu dispositivo, ganhando assim os usuários que não desejam ceder parte da memória de seu aparelho. Rodar dentro do navegador também tem diversas vantagens em relação segurança e privacidade, pois é mais seguro abrir um site da *Web* do que instalar um aplicativo com inúmeras permissões aos recursos do aparelho, mesmo com restrições explícitas de cada plataforma (LOPES, 2013).

2.6 Tecnologias adotadas

Decidiu-se usar no desenvolvimento da aplicação da comanda virtual tecnologias e ferramentas consolidadas no mercado. No gerenciamento do desenvolvimento da aplicação escolheu-se o *Scrum*, uma metodologia ágil para gestão e planejamento de projetos. Foram priorizadas ferramentas *open-sources* com o intuito de reduzir o custo e também dado às grandes comunidades de código livre no mercado (BROMHEAD, 2017).

2.6.1 BackEnd

O *Backend* de uma aplicação é responsável pela regra de negócios e o desenvolvimento das APIs (LAMIM, 2014).

Como linguagem foi utilizada o PHP, acrônimo para PHP: *Hypertext Preprocessor*, uma linguagem de *script* projetada para o desenvolvimento *web*, mas também usada como uma linguagem de programação de propósito geral (PHP, 2018). O PHP começou a ser desenvolvido em 1994 por Rosmus Lerdorf com o nome Personal Home Page e foi implementado em sua homepage pessoal. Com o decorrer do tempo, ele estendeu as funcionalidades para aceitar trabalhar com formulários da web e para se comunicar com bancos de dados (PHP, 2018). Em 1999 Zeev Suraski e Andi Gutmans reescreveram o núcleo da linguagem que cresceu e adquiriu uma grande popularidade com as novas versões PHP 3, PHP 4 e o PHP 5 que passou a dar suporte completo a orientação a objeto. Atualmente a versão principal é o PHP 7 que originalmente foi chamado de PHP da próxima geração (PHP, 2018).

Além da linguagem PHP, o *framework CodeIgniter* foi utilizado, consiste em um kit de ferramentas para desenvolvimento de aplicações em PHP com ampla variedade de bibliotecas, minimizando uma grande quantidade de código (CodeIgniter, 2018). Sua licença de uso é aberta e possibilita empregar a abordagem MVC (*Model View Controller*) para a separação de lógica e sua apresentação (TEIXEIRA, 2013).

2.6.2 FrontEnd

O *Frontend* de uma aplicação tem como responsabilidade a interface e a interação com o usuário (LAMIM, 2014). Utilizou-se HTML (*Hyper Text Markup Language*) para a estrutura das telas, CSS (*Cascading Style Sheets*) para o estilo e o *JavaScript* como linguagem de desenvolvimento.

O JavaScript é uma linguagem de programação interpretada usada como parte dos navegadores web. Inicialmente implementado apenas no frontend, os mecanismos Javascript passaram a ser usados no *backend* de aplicações com o Node.js (DEVELOPER MOZILLA, 2018).

Desenvolvido por Brendan Eich originalmente para o navegador Netscape na década de 1990 o JavaScript inicialmente levava o nome de LiveScript. No começo de Dezembro de 1995, o Java estava no seu ápice o que levou a alteração de LiveScript para o nome JavaScript como uma manobra de marketing da Netscape para dar mais reconhecimento

a linguagem. Em 1996 a Microsoft lançou uma linguagem idêntica para o *Internet Explorer 3* e para conter, a Netscape decidiu normatizar a linguagem através da organização *ECMA International*, uma companhia especializada em padrões e normativas. O nome JavaScript era patenteado até então pela *Sun Microsystems* e não poderia ser usado, foi decidido então pelo nome *ECMAScript*. Atualmente as normas da linguagem são mantidas pelo Comitê Técnico 39 (TC39). Este comitê é composto por especialistas de grandes empresas como Microsoft, Mozilla e Google dentre outras gigantes do munda da tecnologia (MDN WEB DOCS, 2018).

Vue.js é um *framework* Javascript progressivo para construção de componentes *web* reativos. Destaca-se pela sua simplicidade e velocidade em comparação a outros *frameworks* Javascript como Angular e React (SCHMITZ, 2016).

2.6.3 Banco de Dados

Banco de dados consiste em um conjunto de arquivos relacionados entre si, com o intuito de armazenar dados de uma determinada aplicação. O SGBD (Sistema gerenciador de banco de dados) é uma ferramenta responsável pelo gerenciamento de um banco de dados e o seu objetivo é abstrair das aplicações a responsabilidade de gerenciar o acesso, a persistência, a manipulação e a organização dos dados (KORTH, SILBERSCHATZ, 1994).

O modelo Relacional surgiu para aumentar a independência de dados nos SGBDs e prover um conjunto de funções apoiadas na álgebra relacional. Eles implementam estruturas de dados organizadas em relações e utilizam restrições para evitar aspectos indesejados, como: integridade referencial, chaves e integridade de junções de relações. (FERREIRA, TAKAI, 2018)

Utilizou-se o Mysql, um SGBD que dispõe da linguagem SQL (*Structured Query Language*) como interface e a principal opção de banco de dados para aplicativos baseados na *web*, sendo comumente usado, por exemplo, pelo Facebook, Twitter e Youtube e todos os cinco principais sites classificados pela Alexa Internet, Inc (ORACLE, 2018).

A Oracle oferece o servidor de banco de dados MySQL sob um modelo de licença dupla. Para softwares *Open Source* é usada a licença GPL (*General Public Licence*) em que os desenvolvedores devem distribuir suas aplicações também sob o modelo GPL. Para usos comerciais, na qual não é desejável a distribuição do código fonte, é necessário adquirir a licença comercial do MySQL (MYSQL, 2018). A aplicação LeCard está licenciada sob o modelo GPL.

2.6.4 Ferramentas de Engenharia de Software

O BPMN ou *Business Process Model and Notation* (em português, Notação de modelagem de processos de Negócios) é uma representação gráfica para especificar processos em um modelo de negócios. O objetivo do BPMN é apoiar os processos tanto para usuários técnicos como para usuários corporativos, fornecendo uma notação intuitiva e compreensível porém capaz de representar uma semântica complexa dos processos (NOGUEIRA, 2014). Para a criação de diagramas BPMN usou-se o Bizagi Modeler⁴.

UML ou linguagem de modelagem Unificada é uma linguagem de modelagem de desenvolvimento de propósito geral no campo da engenharia de software que se destina a fornecer uma maneira padrão de visualizar o *design* de um sistema (LUCID CHART, 2018). Para os demais diagramas utilizou-se o software Astah⁵, uma ferramenta UML completa que permite a criação de qualquer diagrama necessário com facilidade e rapidez e, por fim, o Balsamic, um software desenvolvido pela Adobe para o desenvolvimento dos protótipos de telas e mockups.

2.6.5 APIs

API é uma sigla vinda do inglês *Application Programming Interface* que em português significa interface de programação de aplicativos, é um conjunto que realiza rotinas e padrões de programação para acessar aplicações *web* ou algum aplicativo de software (CANAL TECH, 2018).

API são ferramentas que faz o compartilhamento de ações, formas padronizadas e protocolos como por exemplos em compras onde um site consome uma API de uma adquirente de cartão de crédito para aprovar uma venda (Battistelli, 2018).

3. Metodologia

Em muitos estabelecimentos comerciais que utilizam algum tipo de comanda para gerenciar suas vendas, costumam trazer uma multa de um determinado valor caso o cliente perca ou venha rasurar o papel da comanda. Esta prática segundo o procon é abusiva e ilegal, pois não é de responsabilidade do consumidor ter controle daquilo que consumiu (PROCON PARANÁ, 2018), sendo assim, o leCard foi concebido para trazer mais confiança e controle para o gestor e auxiliando-o através de uma comanda totalmente digital onde os

⁴ <https://www.bizagi.com/pt>

⁵ <http://astah.net>

funcionários podem inserir dados de consumo dos clientes e estas informações ficarão para consulta para todos os envolvidos.

Ao iniciar o projeto procurou-se uma empresa que usasse comandas de papéis para entender as regras de negócios do setor e também um ambiente disponível para realizar os testes da aplicação.

O estabelecimento escolhido foi a pizzaria Napolitchello que está localizada na cidade de São Tomás de Aquino no estado de Minas Gerais e possui um quadro de seis funcionários, sendo, dois garçons para atender e servir os clientes que irão consumir dentro do estabelecimento, pizzaiolo, forneiro responsável por abrir as massas e assar as pizzas, auxiliar de cozinha e um gerente para gerir, atender e entregar pedidos dos consumidores que irão consumir fora da pizzaria.

Com as perguntas elaboradas anteriormente sobre o negócio, o gerente da empresa Marcos concedeu uma entrevista para a compreensão da regra de negócio no local, e ver na prática o funcionamento de uma comanda. Foram realizadas entrevistas fechadas com perguntas pré-definidas, e também entrevistas abertas em que os *stakeholders* detalharam os requisitos do sistema, tornando possível a especificação dos requisitos do sistema a ser desenvolvido.

O *checklist* do levantamento de requisitos do sistema, bem como as imagens e áudios da entrevista com o cliente se encontram no github do projeto (SANTOS, PORTELA, 2018).

3.1 Levantamento dos requisitos

O levantamento de requisitos é umas das etapas mais importantes na construção de um software bem como o sucesso ou fracasso no desenvolvimento da aplicação (FSMA, 2018). Compreende as identificações e análises de tarefas que determinam as necessidades ou condições a serem atendidas para um produto ou projeto. A identificação dos requisitos ao cliente engloba várias técnicas que podem incluir o desenvolvimento de cenários, casos de usos, observações no local de trabalho (etnografia) e entrevistas individuais ou em grupos. Os requisitos devem ser documentados e definidos com um nível de detalhamento suficiente para as partes interessadas ao projeto, podem também ser documentados em várias maneiras, como formulários, *checklists*, imagens, gravações de áudio e vídeo (NOGUEIRA, 2014).

Em um primeiro momento, usou-se de observações do estabelecimento, e como os funcionários trabalhavam para atender os clientes, depois, o gerente concedeu uma entrevista respondendo as dúvidas levantadas, além de esclarecer com mais detalhes todas as regras de negócio. Para documentar, foi usada a gravação das perguntas propostas ao entrevistado junto a imagens das comandas e pedidos em papéis.

O documento de requisitos está disponível no *GitHub* do projeto no endereço <https://github.com/eliasportela/appcomanda/tree/master/docs>.

3.2 Diagramas Engenharia de Software

Os diagramas na engenharia de software são utilizados para ajudar o entendimento, compreensão e clareza em sistemas. São divididos em dois tipos de diagramas, os estruturais que inclui os diagramas estáticos (Diagramas de Classes), diagramas de Componentes e distribuição “*Deployment*”, e comportamentais que são compostos pelos diagramas de casos de uso, Interação, Sequência, comunicação (Colaboração), atividades e máquina de estado (RIBEIRO GUDWIN, 2015, p. 27).

3.2.1 BPMN

O BPMN (*Business Process Modeling Language*) é uma forma gráfica de apresentar processos comuns, com a não existência de formato de armazenamento especificada. Este auxilia uma compreensão descomplicada dos processos da organização (REIS, 2008).

É um método que utiliza de fluxograma tem como objetivo traçar a modelagem de processos de negócios planejados de uma ponta à outra, este descreve visivelmente as atividades e o fluxos das informações para conclusão de um processo (LUCID CHART, 2018).

Definidos os requisitos do sistema foi realizado a criação do BPMN que se encontra na página 5 no documento de requisitos do projeto (SANTOS, PORTELA, 2018).

3.2.2 Casos de Uso

Em engenharia de software, caso de uso é uma lista de ações ou eventos definidos pela interação entre um ator e as ações da aplicação. Um ator pode ser um usuário, uma empresa, um sistema ou qualquer ação que deve ser capaz de tomar decisões.

Um diagrama de caso de uso representa a relação entre atores, como exemplo usuários e sistema e utilizam representações de bonecos que simbolizam os atores do sistema e suas interações, sendo que não seguem uma ordem de execução (MICROSOFT, 2015).

O diagrama de caso de uso representa possíveis interações sistêmicas com o intuito de atingir uma meta específica. Alguma particularidade dessa organização de requisitos funcionais, representa caminhos e eventos, retrata um fluxo principal (curso básico de uma ação), multi níveis pois um caso de uso pode desfrutar de funcionalidades de outro (ROUSE, 2007).

O diagrama de casos de uso da aplicação LeCard pode ser encontrada no documento de requisitos do projeto (SANTOS, PORTELA, 2018, p. 5).

3.3 Modelagem do banco de dados

Diagrama de entidade-relacionamento é representado como um fluxograma que estes que retratam as entidades e relacionamentos, pessoas, objetos, como elas se relacionam dentro do sistema. São estruturas gramaticais, as entidades são caracterizadas como substantivos e os relacionamentos os verbos (LUCID CHART, 2018).

As siglas E-R conceituam o termo modelagem conceitual, os objetos são exibidos de uma forma abstrata independem da implementação em computador, os dados serão armazenados em um banco de dados. Os modelos usados com mais frequência são MER (modelo de entidade de relacionamento) aglomerado de conceitos e elementos que o profissional de banco de dados necessita entender para desenvolver um DER de qualidade e DER (diagrama entidade relacionamento) resultado de modelagem que o profissional de banco de dados atinge tendo grande conhecimento sobre o MER da aplicação (WORDPRESS, 2009).

O MER e o DER da aplicação LeCard podem ser encontrados no documento de requisitos do projeto (SANTOS, PORTELA, 2018, p. 27).

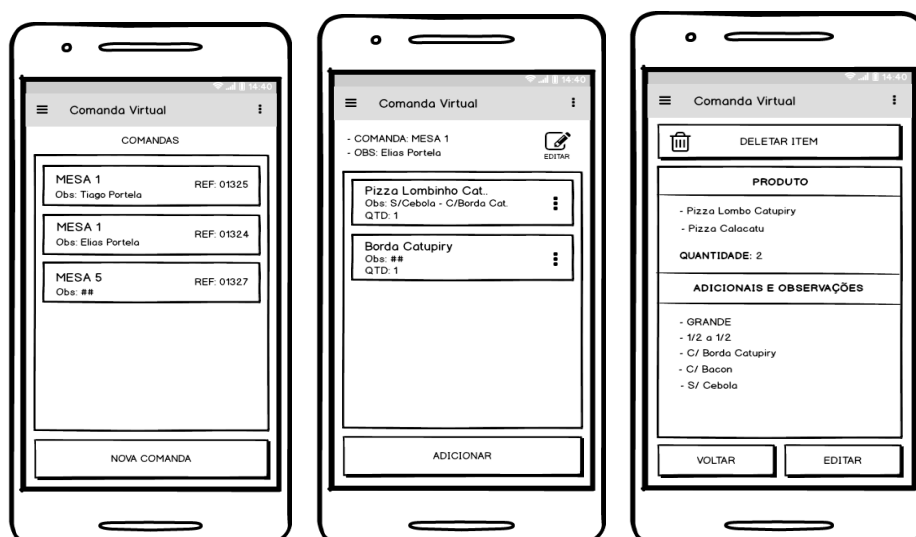
3.4 Prototipação das Telas

Técnica de construção e elaboração que envolve diversos tipos de projeto incluindo a preparação de um software, esta forma de trabalho é chamada de prototipagem. No início de um projeto na fase de criação de uma aplicação é imprescindível a primeira etapa o levantamento de requisitos, entender como e aonde o cliente quer chegar, este processo torna-se complexo com a não utilização de ferramentas específicas a prototipação de telas

consegue minimizar alguns riscos de entendimento com dois níveis baixo e alto, sendo o baixo a prototipação não será visível e sem funcionalidades confirmadas do sistema, o nível alto é o reverso podendo ser o nível final do sistema (ALVES, PAOLIELLO, 2018).

Segundo ROSA, GURGEL e PASSOS (2012) prototipagem “é uma representação limitada de um design que permite aos usuários interagirem com ele e explorarem suas conveniências.”. A Figura 1 apresenta exemplos dos protótipos que foram desenvolvidos para o aplicativo LeCard.

Figura 1: Mockups das telas de comandas, detalhes e produtos



Fonte: Os autores

3.5 Desenvolvimento

O desenvolvimento foi dividido entre *frontend* e *backend*, e iniciado com a criação dos protótipos das telas definidas com os *stakeholders* (cliente e orientador do projeto). Com os protótipos aprovados e embasados no MER e DER do projeto, realizado a criação do modelo físico do banco de dados, em seguida houve a criação das APIs, utilizando o *framework* PHP *codeigniter*. Sendo assim, o sistema foi decomposto entre o *WebApp*, usado pelo Garçon e cliente, e as APIs da aplicação, incumbida para as tarefas de cadastros em geral bem como o recebimento dos pedidos e o fechamento das comandas.

Figura 2: Função inserção de comando

```
if ($acesso_aprovado) {  
    $dataRegister = $this->input->post();  
  
    if ($dataRegister == null) {  
        echo json_encode(array('result' => 'Erro nos parametros enviados', 'id_comanda' => null),  
            options: JSON_UNESCAPED_UNICODE);  
        $this->output->set_status_header('500');  
        return;  
    }  
  
    $mesa = $dataRegister['mesa'];  
    $observacao = $dataRegister['observacao'];  
  
    $ref_comanda = rand(1000, 9000);  
  
    $dataModel = array('mesa' => $mesa, 'observacao' => $observacao, 'ref_comanda' => $ref_comanda);  
    $res = $this->Crud_model->InsertID('comanda', $dataModel);  
    if ($res) {  
        $dataModel = array('ref_comanda' => $ref_comanda + $res);  
        $res2 = $this->Crud_model->Update('comanda', $dataModel, array('id_comanda' => $res));  
        if ($res2) {  
            echo json_encode(array('result' => 'Sucesso', 'id_comanda' => $res),  
                options: JSON_UNESCAPED_UNICODE);  
            return;  
        } else {  
            echo json_encode(array('result' => 'Erro no Servidor', 'id_comanda' => null),  
                options: JSON_UNESCAPED_UNICODE);  
            $this->output->set_status_header('500');  
            return;  
        }  
    }  
}
```

Fonte: Os autores

Figura 3: Função de nova comanda

```
novaComanda() {  
    if (this.dados.mesa !== '') {  
        openLoading("Cadastrando comanda..");  
        let options = {emulateJSON: true};  
        this.$http.post(base_url + 'comandas/inserir/' + this.token, this.dados, options)  
            .then(response => {  
                let id_comanda = response.data.id_comanda;  
                if (id_comanda !== null) {  
                    closeLoading();  
                    this.$router.push("comanda-detahes/" + id_comanda);  
                } else {  
                    openModalMsg("Ops!", response.data.result);  
                }  
            });  
    } else {  
        openModalMsg("Ops!", "Selecione uma mesa");  
    }  
}
```

Fonte: Os autores

Um dos exemplos de APIs desenvolvidas para o projeto é a função “InserirComanda”, mostrado na figura 2, que tem como objetivo criar um registro na tabela “comanda”. Ela se inicia com a validação do token de acesso pela função ‘ValidarToken’, na

condição de sucesso faz o recebimento dos dados enviados através do protocolo de requisição HTTP POST, em seguida realiza o tratamento e inserção nas variáveis “\$mesa” e “\$Observação”, a referência da comanda é criada baseada em um número aleatório de quatro dígitos juntamente com a soma do identificador único da tabela, resultando assim em um número não repetível. Por fim é retornando status 200 para sucesso, 500 em casos de erro ou 401 para acesso negado.

Já nas telas do WebApp foi utilizado HTML, CSS e o framework Javascript Vue. A função “novaComanda”, exibido na figura 3, utiliza a API ‘InserirComanda’ para inserir um novo registro. Esta função inicia com a validação se o campo ‘Mesa’ e logo após insere um tela de espera para mostrar ao usuário que a requisição está sendo processada, em seguida faz se o uso de uma requisição Ajax para enviar o *token* de acesso, os dados com as informações de mesa e observação. Obtendo sucesso é fechado há tela de espera e o sistema redireciona o usuário para a tela Detalhes da Comanda, todavia se houver algum impeditivo, é informado o erro e a descrição na tela.

Encontram-se no repositório do projeto o código fonte das aplicações e o *dump* do modelo físico do banco de dados.

3.6 Reuso de software (código)

Reuso de software consistem em reutilizar o que é vital ao ciclo de vida do sistema, não existindo a necessidade de recriar códigos (BASTOS FILHO, 2013).

Com o uso do reuso de software torna-se possível a redução em 40% de tempo e custo, pois não havendo a necessidade de criar produtos do zero, aproveitando componentes específicos e testados (TUEIV, 2014).

Como uma parte de reuso de código na aplicação, foi utilizado a biblioteca do *Codeigniter* “CI_Model”, cujo objetivo é encurtar o tempo no processo de chamadas ao banco de dados. Primeiramente é necessário preencher os dados de conexão no arquivo de configuração do *framework* e depois criar uma classe que herda os métodos da classe “CI_Model”, através dela é possível fazer chamadas ao banco de dados passando apenas como parâmetro o nome da tabela e os dados a serem processados.

3.7 Testes

A última etapa do desenvolvimento de um software é a realização de teste, pois com ela conseguimos detectar possíveis erros decorrentes do desenvolvimento, varrendo a

aplicação de ponta a ponta a procura de erros de compilação que possam dificultar o funcionamento do software (FACOM, 2017).

Segundo Rios e Filho (2013, p. 17) teste de software é "verificar se o software está fazendo o que deveria fazer, de acordo com os seus requisitos, e não o que não deveria fazer".

3.8 Implantação

O software foi implantado na empresa Napolitchello, e iniciou com uma simulação entre os próprios funcionários para fim de treinamento. Logo após, empregou-se ao garçom algumas tarefas de testes, como abrir as comandas e realizar pedidos, onde foram notadas certas dificuldades ao manusear a aplicação pela primeira vez, porém superados em certo período de manuseio. Após isto, foi oferecida a utilização para clientes, onde puderam visualizar suas respectivas comandas, acessar o cardápio, produtos consumidos e os pedidos realizados.

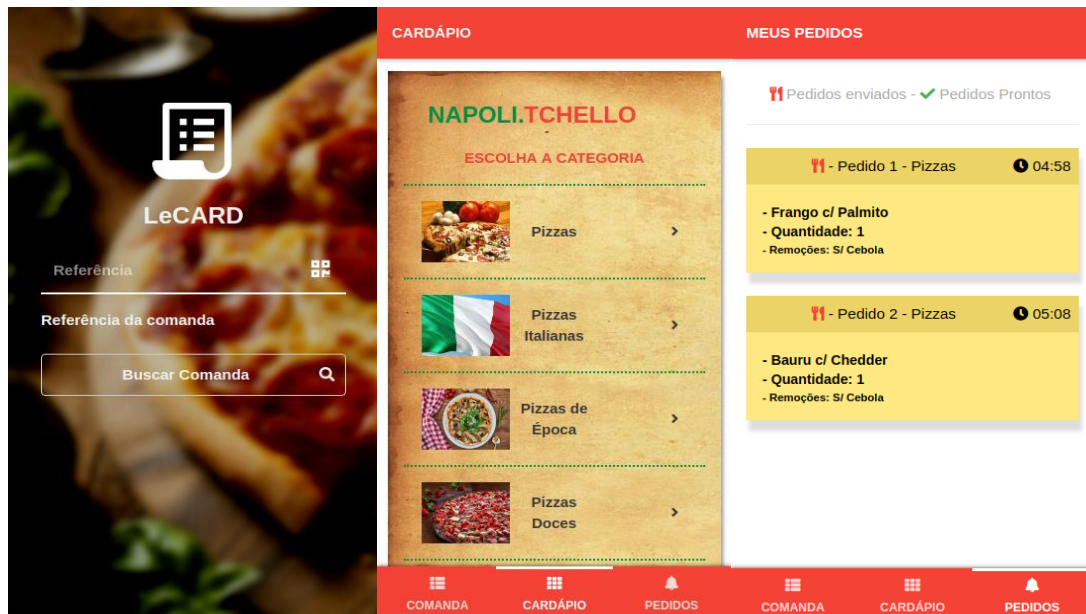
4. Análise de Resultados

O software entregue foi dividido entre a parte acessada pelo garçom e a parte acessada pelo cliente, tal como as APIs, que são destinadas para integração com softwares de terceiros. Buscou-se simplificar o máximo possível os processos realizados nas telas da aplicação. A Figura 4 é mostrado parte das telas da área do cliente tais como a tela de buscas de comandas, o cardápio virtual e os meus pedidos. A figura 5 são telas destinadas ao garçom, sendo, login, gerenciamento de comandas e detalhes das comandas. Demais telas se encontram no documento de requisitos do sistema (SANTOS, PORTELA, 2018, p. 35).

Levou-se ao trabalho documentar todas as APIs oferecidas pela aplicação através de exemplos de requisições e tabelas com os parâmetros suportados e está disponível no github do projeto (SANTOS, PORTELA, 2018).

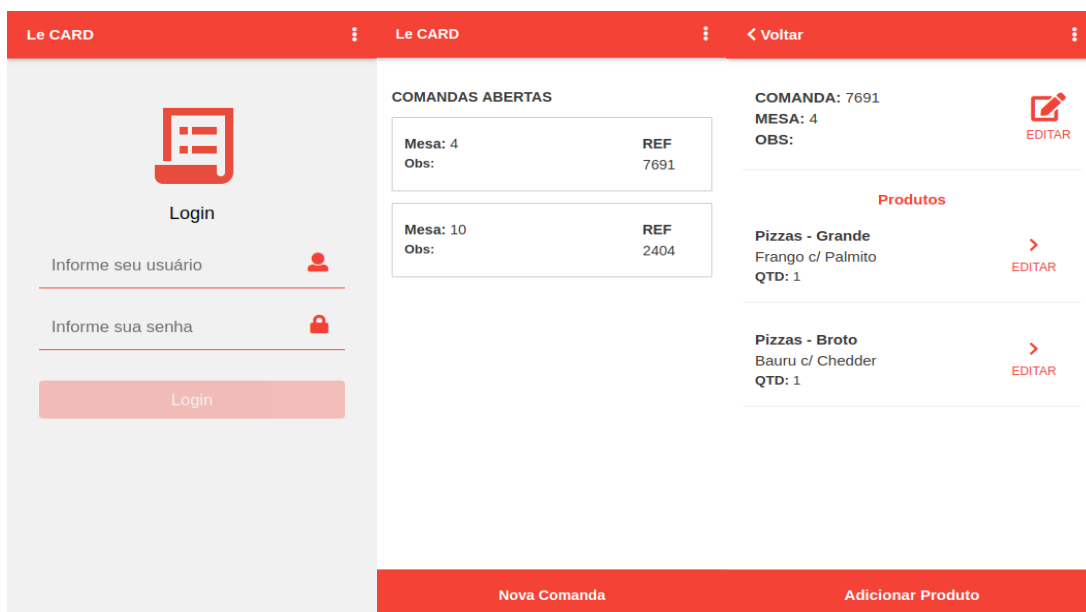
Desenvolveu também um *dashboard* onde foi colocado em prática o uso de integrações, simulações de consumo e gerenciamento dos dados, como em um *software* externo, todavia sua documentação e código fonte não foram utilizados, pois foram considerados fora do escopo do projeto pelo fato da linha de defesa deste trabalho ser voltado para o conceitos de uso de integrações com APIs.

Figura 4: Tela de busca de comandas, cardápio e meus pedidos.



Fonte: Os autores

Figura 5: Tela de login, comandas e detalhes da comanda.



Fonte: Os autores

Após a implantação, foram entrevistados funcionários que utilizaram o sistema, e fez-se algumas simples perguntas sobre a opinião do entrevistado em relação à utilização da plataforma, assim como a opinião em relação à ajuda adquirida através da comanda virtual LeCard aos processos antes realizados de forma manual.

Através da análise do conteúdo das respostas dos entrevistados verificou-se uma grande satisfação em relação à plataforma. De acordo com a percepção dos funcionários, os pedidos transitaram de uma forma mais ágil, reduzindo o tempo entre a mesa e a cozinha além de oferecer um meio mais confiável de acompanhamento daquilo que foi pedido pelo cliente, conforme presente na Tabela 1. Ainda, os colaboradores ponderaram redução no uso do papel, trazendo benefícios ao meio ambiente além de reduzir custos para a empresa.

Como ponto negativo, apresentou-se a dificuldade inicial no manuseio por parte dos garçons, mas conforme o uso da aplicação houve a redução deste esforço. Notaram-se também alguns contratempos de lentidão no consumo das APIs, mas nada impeditivo para que o sistema fluísse normalmente.

Tabela 1. Parte das respostas dos entrevistados

Funcionários	<p>...O sistema ajuda ao evitar o erro de marcar produtos que não foram pedidos pelo cliente...</p> <p>...Os pedidos transitam com mais rapidez e facilidade proporcionando mais agilidade...</p> <p>...Como cada sistema no início, não se está acostumado a manusear, você tem uma certa dificuldade, mas achei interessante e fácil de mexer e os pedidos saem conforme o garçom insere.</p> <p>...Em relação ao custo, há uma grande economia de papel, os clientes também podem perceber que estamos sempre dispostos a melhorar e inovar nosso atendimento...</p> <p>...O cliente monitorando reduz o erro de existir produtos que foram pedidos incorretamente...</p> <p>...A utilização do sistema facilitou nos processos de fabricação das pizzas, inclusive no acompanhamento do cliente na mesa...</p> <p>...Não houve nenhuma dificuldade pois a interface segue os padrões de outros aplicativos, tudo sendo bastante intuitivo.</p> <p>Em relação na forma digital a forma manual, vem a consciência com o meio ambiente, pois se reduz o uso de papéis. Houve uma melhora na rapidez, o que gastaria por exemplo um minuto para chegar na produção,</p>
---------------------	---

	<p>quando houver o pedido na mesa já chega na cozinha e já está pronto para ser produzido...</p> <p>...Antigamente tinha que levar o papel para a zona de preparo, e se errassem o pedido o cliente não iria saber. Com a forma digital o próprio cliente poderá acompanhar aquilo que foi pedido, confirmando caso houver alguma inconsistência, como adicionais ou remoções não marcados...</p> <p>...Quase não houve dificuldades encontradas, apenas algumas partes que demoraram um pouquinho para processar...</p> <p>...Com o aplicativo ficou mais rápido realizar os pedidos, pois já traz os ingredientes que você poderá remover ou adicionar, não havendo atrasos para chegar a cozinha...</p>
--	--

Fonte: Os autores

5. Empreendedorismo

Nos estudos de (Baggio, Baggio, 2014) o "Empreendedorismo pode ser compreendido como a arte de fazer acontecer com criatividade e motivação", baseia-se em sentir-se bem ao realizar qualquer projeto, seja ele pessoal ou organizacional, e assumir um comportamento proativo diante dos riscos e oportunidades apresentadas (BAGGIO; BAGGIO, 2014).

O empreendedor é capaz de conceder uma nova funcionalidade a algo que já existe e empenha-se em descobrir novos meios de inovar (BRITO; PEREIRA; LINARD, 2013).

5.1 Startup

O termo *Startup* começou a se popularizar nos Estados Unidos a partir de 1990 com o início da popularização da internet, entretanto apenas chegou ao Brasil no período entre 1999 e 2001 (SIGNORI et al., 2014). *Startup* pode ser considerada como uma organização formada para pesquisar um modelo de negócios que possa ser repetido e escalável, e assim, para o crescimento do negócio será indispensável um aumento de capital ou pessoal na mesma proporção (Blank, 2006).

5.2 Startup enxuta

Dentro do conceito de *Startup* Enxuta (*Lean Startup* em Inglês) entra o MVP ou Mínimo Produto Viável que significa entregar uma amostra do produto ou serviço ao mercado e suas fragilidades serão expostas e melhoradas em ciclos curtos de desenvolvimento. Isso simboliza um conjunto de hipóteses que necessitam ser validadas ou abolidas o mais rápido possível em pequenos períodos de tempo para gerar produtos entregáveis e que validam o negócio em seu mercado de consumo (SIGNORI et al., 2014).

5.3 Modelo canvas

O termo modelo de negócio pode ser referido como a possibilidade de visualizar a descrição de um negócio e suas partes de maneira que as ideias sejam compreendidas da forma como pretendia o criador do modelo. Dentre as várias ferramentas criadas para o empreendedor expor seus modelos de negócios existe o Canvas, criado por Osterwalder e Pigneur cujo o objetivo é facilitar a identificação de todos os aspectos de um novo empreendimento (SEBRAE, 2013).

O canvas consiste em um quadro dividido em nove blocos que compõem um negócio e agrupado em quatro questões, vou fazer o que, para quem vou fazer, como vou fazer e quanto irá custar, para serem respondidas pelo empreendedor (SEBRAE, 2013).

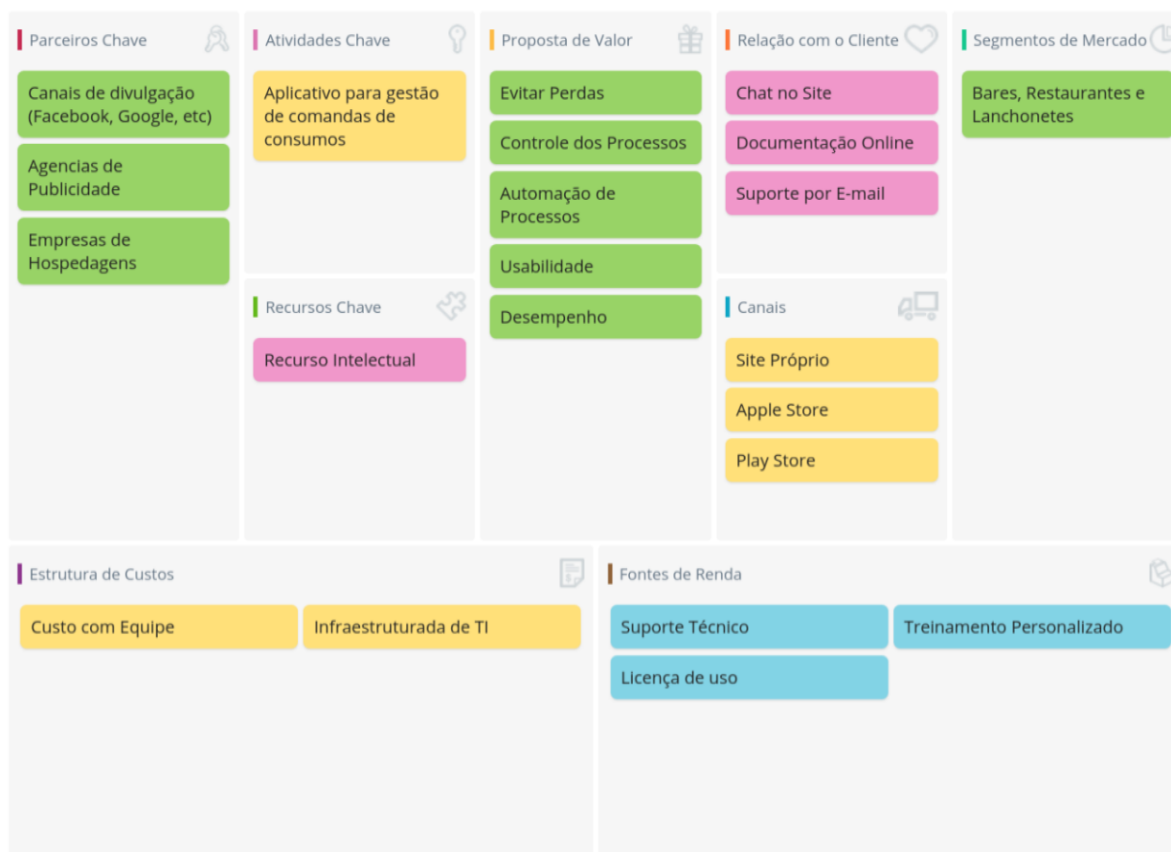
5.4 Canvas do projeto

LeCard é um sistema de controle virtual de comandas para bares, restaurantes e lanchonetes e tem como objetivo auxiliar o controle dos gastos do cliente e eliminar o uso de papéis. O principal público da aplicação são os pequenos e médios empreendedores que buscam um método mais eficiente para gerir e evitar perdas dos pedidos dos seus clientes. O sistema será feito de forma genérica mais apoiada em um mapeamento do modelo de negócio de uma pizzaria que utiliza um exemplar impresso de comandas. O custo de desenvolvimento e manutenção será utilizado com, armazenamento dos dados em nuvens, computadores e salários da equipe do projeto.

Dos blocos que compõe o Canvas, o LeCard será segmentado para o mercado de Bares, Restaurantes e Lanchonetes e terá como canais de distribuição as lojas de aplicativos virtuais Apple Store, Play Store e o próprio site da aplicação. A atividade chave do negócio será uma aplicação para gestão de comandas de consumos, usando assim o recurso intelectual no processo da criação. A relação com os clientes será feito mediante a suporte por e-mail, documentação e chat no site e a proposta de valor será fundamentado em melhorias no desempenho, melhor usabilidade, automação e controle dos processos do

negócio bem como ajudar o cliente a evitar perdas com o uso digital da comanda. Serão realizadas parcerias com empresas de Hospedagens, agências de publicidades e canais de divulgação como Google e Facebook. A estrutura de custos será dividida entre infraestrutura e gastos com a equipe. Por fim, a fonte de renda será obtida pela cobrança de uma licença de uso, suporte técnico e treinamentos personalizados. Figura 6 Canvas do projeto.

Figura 6: Canvas do projeto LeCard.



Fonte: Os autores

6. Considerações Finais

Conseguiu-se entregar de forma satisfatória os objetivos propostos no início do projeto, sendo desenvolvido um web aplicativo de comanda virtual para bares, lanchonetes e restaurantes em conjunto com a documentação das *APIs*, na qual a proposta é auxiliar na integração com softwares de terceiros.

Como resultado, foi realizada a implantação do sistema em uma pizzaria e houve ótima satisfação dos colaboradores que utilizaram a ferramenta, permitindo melhorias significativas na transição dos pedidos entre as mesas para a cozinha, ajudando a evitar o risco

de perdas de comandas, fraudes além de reduzir o tempo de atendimento dos garçons, de preparo e consequentemente a duração de espera dos consumidores.

O desfecho obtido no desenvolvimento deste trabalho serviu para facilitar o entendimento do conteúdo ministrado em sala de aula, sendo de suma importância na transferência do conhecimento teórico para a prática. Em relação às experiências adquiridas com os processos de construção de um software, desde o levantamento dos requisitos até a implantação do sistema na empresa, contemplou-se a enorme importância de compreender as regras de negócios como uma forma de atender todas as necessidades do cliente. A escolha das tecnologias foi igualmente fundamental para o sucesso do projeto pois o uso de linguagens, frameworks e ferramentas atuais que possuem comunidades grandes e ativas puderam aumentar o êxito na soluções de empecilhos que apareceram no decorrer do desenvolvimento.

Sendo o LeCard um Mínimo produto viável (MVP), precisará ainda ser evoluído em alguns pontos como integrações com sistemas de caixas e ERPs, melhorias na interação com usuário (UX) e na infraestrutura, que deverá ser auto escalável para conseguir demandar novos clientes. Para o melhor aproveitamento da tecnologia também se faz necessário que as empresas e colaboradores recebam treinamentos mais longos para adquirirem conhecimento na utilização das funcionalidades da aplicação assim como o software deve ser sempre melhorado através da adesão e experiências dos clientes com a plataforma.

Vale ressaltar que ao surgir novos clientes que demandem novas regras de negócios, existirá a necessidades de incluir outras funcionalidades como *delivery*, auto-pedidos que consiste em pedidos realizados pelo próprio cliente, dentre outras funções requeridas pelo contratante.

Referências

ALVES, Bruno Leite, PAOLIELLO Rafael Saraiva. Prototipagem de interfaces na prática. Disponível em: <<http://www.univale.com.br/unisite/mundo-j/artigos/48prototipagem.pdf>>. Acesso em: 25 jul. 2018.

ANDRADE, Tiago. Eu quero desenvolver Apps para iOS, por onde eu começo?.

InfiniteLoop. 2017. Disponível em: <<http://www.infiniteLoop.com.br/eu-quiero-desenvolver-apps-para-ios-por-onde-eu-comeco/>>. Acesso em: 1 mai. 2018.

BAGGIO, Adelar Francisco; BAGGIO, Daniel Knebel. Empreendedorismo: Conceitos e Definições. IMED. 2014. Disponível em:

<<https://seer.imed.edu.br/index.php/revistasi/article/download/612/522>>. Acesso em: 12 jun. 2018.

BASTOS FILHO, Cesar. REUSO DE SOFTWARE. 2013. Disponível em:

<<http://www.uel.br/cce/dc/wp-content/uploads/VersaoPreliminarTCC-CezarBastos.pdf>>. Acesso em: 26 jul. 2018.

BATTISTELLI, Juliana. O que é API e para que serve dentro da programação?. 2018.

Disponível em: <<https://blog.mastertech.tech/tecnologia/o-que-e-api-e-para-que-serve-dentro-da-programacao/>>. Acesso em: 03 set. 2018.

BLANK, Steve. The Four Steps to the Epiphany. STANFORD. 2006. Disponível em:

https://web.stanford.edu/group/e145/cgi-bin/winter/drupal/upload/handouts/Four_Steps.pdf. Acesso em: 12 jun. 2018.

BRITO, Andréia Matos; PEREIRA, Pedro Silvino; LINARD, Ângela Patrícia.

Empreendedorismo. Instituto Federal de Educação , Ciências e Tecnologias. Juazeiro do Norte, CE. 2013. Disponível em:

<http://estudio01.proj.ufsm.br/cadernos/ifce/tecnico_edificacoes/empreendedorismo.pdf>. Acesso em: 12 jun. 2018.

B'FAR, Reza. Mobile Computing Principles: Designing and Developing Mobile with UML and XML. Nova York: Cambridge University Press, 2005.

BROMHEAD, Ben. 10 advantages of open source for the enterprise: Enterprises looking to make smart use of open source software have plenty of great reasons to do so. 2017.

Disponível em: <<https://opensource.com/article/17/8/enterprise-open-source-advantages>>. Acesso em: 1 mai. 2018.

CANAL TECH. O que é api?. Disponível em: <<https://canaltech.com.br/software/o-que-e-api/>>. Acesso em: 03 set. 2018.

CODEIGNITER. Codeigniter at a glance. Disponível em:

<https://codeigniter.com/user_guide/overview/at_a_glance.html>. Acesso em: 21 jul. 2018.

DEVELOPER MOZILLA. JavaScript. Developer Mozilla. Disponível

em: <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>. Acesso em: 1 mai. 2018.

FACOM. Teste de software. Disponível em:

<https://www.pjf.mg.gov.br/secretarias/sarh/edital/interno/selecao2015/arquivos/apostilas/mo_dulo4_p.pdf>. Acesso em: 30 jul. 2018.

FACULDADE SALESIANA. Previsão de carga a curto prazo: modelagem em duas etapas.

Disponível em: <http://www.fsma.edu.br/si/edicao3/rna_paper.pdf>. Acesso em: 07 out. 2018.

FONSECA FILHO, Clézio. Historia da Computação: O caminho do pensamento e da tecnologia. Porto Alegre: EDIPUCRS, 2007. 205 p. Disponível em:

<<http://www.pucrs.br/edipucrs/online/historiadacomputacao.pdf>>. Acesso em: 6 abr. 2018.

EDUARDO, João Ferreira; KOTARO, Osvaldo Takai. Banco de Dados: Modelo de Dados

Relacional. 2018. Disponível em: <<https://www.ime.usp.br/~jef/bd03.pdf>>. Acesso em: 01 ago. 2018.

GOOGLE DEVELOPERS. Your First Progressive Web App. Developers Google. 2018.

Disponível em: <<https://codelabs.developers.google.com/codelabs/your-first-pwapp/#0>>.

Acesso em: 1 mai. 2018.

GUILHERME, José Santa Rosa; GURGEL, Andrei; PASSOS, Marcel. Técnicas Baseadas em Etnografia e Prototipagem no Design de interface de Aplicativo Mobile para Gerenciamento Acadêmico. 2012. Disponível em:

<https://repositorio.ufrn.br/jspui/bitstream/123456789/18526/1/Jos%C3%A9%20Guilherme%20da%20Silva%20Santa%20Rosa_%20T%C3%A9cnicas%20Baseadas%20em%20etnografia.pdf>. Acesso em: 25 jul. 2018.

KORTH, H. F; SILBERSCHATZ, A. Sistemas de Bancos de Dados. 2. ed. Makron Books, 1994.

KRISHNA, Y. K. Sundara; DEVARAKONDA, G. K. Mohan. A Survey on Architectures of Mobile Operating Systems: Challenges and Issues. ARC Journals. 2015. Disponível em:

<<https://www.arcjournals.org/pdfs/ijrscse/v2-i3/18.pdf>>. Acesso em: 1 mai. 2018.

LAMIM, Jonathan. Afinal, o que é Frontend e o que é Backend?: Você sabe separar o que é o frontend de uma aplicação e o que é o backend? Se não sabe, esse artigo vai te ajudar. Oficina da Net. 2014. Disponível em: <<https://www.oficinadanet.com.br/post/13541-afinal-o-que-e-frontend-e-o-que-e-backend->>. Acesso em: 1 mai. 2018.

LOPES, Sergio. A Web Mobile: Programe para um mundo de muitos dispositivos. São Paulo: Casa do Código, 2013.

LUCID CHART. O que é um diagrama uml?. Disponível em: <<https://www.lucidchart.com/pages/pt/o-que-e-uml>>. Acesso em: 01 ago. 2018.

LUCID CHART. What is business process modeling notation. Disponível em: <<https://www.lucidchart.com/pages/bpmn>>. Acesso em: 22 jul. 2018.

MARKOFF, John. I, Robot: The Man Behind the Google Phone. The New York Times, 4 nov 2007. Disponível em: <https://www.nytimes.com/2007/11/04/technology/04google.html?_r=2&hp=&pagewanted=all>. Acesso em: 1 mai. 2017.

MICROSOFT. Diagramas de caso de uso uml: diretrizes. Disponível em: <<https://msdn.microsoft.com/pt-br/library/dd409432.aspx>>. Acesso em: 07 out. 2018.

MDN WEB DOCS. Javascript. Disponível em: <<https://developer.mozilla.org/pt-br/docs/web/javascript>>. Acesso em: 11 jul. 2018.

MENDONÇA, Aderval. Mobilidade em análise. Devmedia. Disponível em: <<https://www.devmedia.com.br/mobilidade-em-analise/3309>>. Acesso em: 15 abr. 2018.

MYSQL. Lixeira comercial para OEMs, ISVs e VARs. Disponível em: <<https://www.mysql.com/about/legal/licensing/oem/>>. Acesso em: 17 set. 2018.

NOGUEIRA, Rhaíssa Arantes. Artigo Introdução ao Business Process Modeling Notation (BPMN). 2014. Disponível em: <<https://www.devmedia.com.br/introducao-ao-business-process-modeling-notation-bpmn/29892>>. Acesso em: 01 ago. 2018.

ORACLE. Mysql: O banco de dados de código aberto mais conhecido no mundo. Oracle. Disponível em: <<https://www.oracle.com/br/mysql/index.html>>. Acesso em: 1 mai. 2018.

PROCON. Perda da comanda – saiba seus direitos. Disponível em:

<<http://www.procon.pr.gov.br/modules/conteudo/conteudo.php?conteudo=565>>. Acesso em: 07 out. 2018.

PHP. História do php. Disponível em: <https://secure.php.net/manual/pt_br/history.php.php>. Acesso em: 01 ago. 2018.

PHP. O que é o PHP ?. PHP. Disponível em:<https://secure.php.net/manual/pt_BR/intro-what.php>. Acesso em: 1 mai. 2018.

QADIR, Yasir Abdul. AN INTRODUCTION TO IPHONE HARDWARE, OPERATING SYSTEM, APPLICATIONS AND DEVELOPMENT OF IPHONE APPLICATIONS. 2010.

Disponível

em:<<https://www.theseus.fi/bitstream/handle/10024/17152/Bachelors%20Thesis.pdf?sequence=1&isAllowed=y>>. Acesso em: 1 mai. 2018.

RABELLO, RAmon Ribeiro. Android: um novo paradigma de desenvolvimento móvel.

Disponível em:<https://dpwinfo.files.wordpress.com/2013/04/wm18_android.pdf>. Acesso em: 1 abr. 2018.

REIS, Glauco Dos Santos. Modelagem de processos de negócios com bpmn. 1 ed. São Paulo: Editora PortalBPM Ltda, 2008. 7 p.

RIBEIRO GUDWIN, Ricardo. Engenharia de Software: Uma Visão Prática. 2a Edição. 2015.

Disponível em:

<<http://faculty.dca.fee.unicamp.br/gudwin/sites/faculty.dca.fee.unicamp.br.gudwin/files/ea975/ESUVP2.pdf>>. Acesso em: 09 set. 2018.

RIOS, Emerson; FILHO, Trayahú Moreira. Teste de software. 3 ed. Rio de Janeiro: ALTA BOOKS, 2013. 17 p.

ROBBINS, Jennifer Niederst. HTML5: Pocket Reference. 5. ed. O'Reilly, 2013.

ROUSE, Margaret. Use case. Disponível em:

<<https://searchsoftwarequality.techtarget.com/definition/use-case>>. Acesso em: 22 jul. 2018.

SANTOS, PORTELA, Lucas de Castro, Elias Douglas Wenceslau, GitHub contendo documentação do TCC. Disponível em

<<https://github.com/eliasportela/appcomanda/tree/master/docs>> . 2018.

SCHMITZ, Daniel. CONHEÇA O VUE.JS, UM FRAMEWORK JAVASCRIPT PARA CRIAÇÃO DE COMPONENTES WEB REATIVOS. Tableless. 2016. Disponível em:

<<https://tableless.com.br/conheca-o-vue-js-um-framework-javascript-para-criacao-de-componentes-web-reativos/>>. Acesso em: 1 mai. 2018.

SEBRAE. O quadro de modelo de negócios: Um caminho para criar, recriar e inovar em modelos de negócios. SEBRAE. 2013. Disponível em:

<https://www.sebraecanvas.com.br/downloads/cartilha_canvas.pdf>. Acesso em: 12 jun. 2018.

SEBRAE. Bares e restaurantes: um setor em expansão: Alimentação fora de casa é um ramo que cresce a cada ano, mas requer planejamento adequado.. SEBRAE. 2017. Disponível em:

<<http://www.sebrae.com.br/sites/PortalSebrae/artigos/bares-e-restaurantes-um-setor-em-expansao,1038d53342603410VgnVCM100000b272010aRCRD>>. Acesso em: 7 abr. 2018.

SIGNORI, Gláuber Guilherme et al. Startup e inovação: inovando na forma de pensar e decretando o fim das velhas ideias. ANPROTECT. 2014. Disponível em:

<<http://www.anprotec.org.br/Relata/ArtigosCompleto/ID%20107.pdf>>. Acesso em: 12 jun. 2018.

TEIXEIRA, José Ricardo. artigo introdução ao framework php codeigniter. 2013. Disponível em: <<https://www.devmedia.com.br/introducao-ao-framework-php-codeigniter/27346>>.

Acesso em: 21 jul. 2018.

TUEIV, Marcelo. Reúso de Software. 2014. Disponível em:

<https://www.ibm.com/developerworks/community/blogs/tlcbr/resource/mp/TLC-BR_Mini_Paper_Ano_10_N_222_Reuso_Software.pdf?lang=en>. Acesso em: 27 jul. 2018.

VANNI, Renata Maria Porto. Dispositivos Móveis: Uma visão geral sobre a história e tecnologia para dispositivos móveis. Instituto Federal de Educação , Ciências e Tecnologias.

Araraquara, SP. Disponível em:<<https://pt.slideshare.net/MauricCarvalho/dispositivos-mveis-15375049>>. Acesso em: 1 mai. 2018.

WORDPRESS. Mer -modelo de entidades e relacionamentos e der-diagrama entidade relacionamento. Disponível em: <<https://nati57.wordpress.com/2009/01/14/mer-modelo-de-entidades-e-relacionamentos-e-der-diagrama-entidade-relacionamento/>>. Acesso em: 22 jul. 2018.