

Serviço Nacional de Aprendizagem Comercial do Rio Grande do Sul
Faculdade Senac Porto Alegre
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

JORGE DAMIÃO ABRANTI DA SILVA

RELATÓRIO DE PROJETO PARCIAL

HUNGER MANAGEMENT

Sistema *web* para gestão de entregas de comida com *chatbot* de atendimento

Porto Alegre
2017

JORGE DAMIÃO ABRANTI DA SILVA

RELATÓRIO DE PROJETO PARCIAL

HUNGER MANAGEMENT

Sistema *web* para gestão de entregas de comida com *chatbot* de atendimento

Relatório de Projeto Parcial apresentado como requisito parcial para obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas pela Faculdade Senac Porto Alegre.

Orientador: Prof. Me. Luciano Zanuz

Porto Alegre

2017

RESUMO

No período tecnológico contemporâneo, a utilização de ferramentas informatizadas na gestão comercial e na relação com clientes se mostram cada vez mais eficazes em aumentar os lucros e reduzir custos e despesas, seja em empresas pequenas, médias ou grandes. A utilização de *bots* (diminutivo do inglês robots), programa de computador fabricado para automatizar procedimentos, aplicada em *chats* (do inglês conversa), tem se tornado um canal de comunicação e atendimento promissor. Seguindo esta tendência, o sistema HUNGER MANAGEMENT traz um canal de atendimento automatizado para pedidos, unido a um sistema web de gestão com relatórios e métricas em uma interface visual intuitiva, utilizando as mais atuais tecnologias para trazer agilidade no atendimento e economia para a empresa, reduzindo custos com atendimento humano e telefonia, e sempre mantendo a disponibilidade do serviço de pedidos.

Palavras-chave: Comercial, *Delivery*, *Web*, *Chatbot*.

LISTA DE FIGURAS

Figura 1 – Pesquisa IBOPE Inteligência 2017	8
Figura 2 – Visão técnica da solução	19
Figura 3 – Fluxo de processos metodologia ágil Scrum Solo	20
Figura 4 – Diagrama BPM - Fluxo principal de atendimento do <i>chatbot</i>	25
Figura 5 – Diagrama de Entidade Relacional	27
Figura 6 – <i>Wireframe</i> da tela de fila de pedidos	32
Figura 7 – Template inicial <i>chatbot</i>	33
Figura 8 – Template inicial boas vindas e ver cardápio do <i>chatbot</i>	34
Figura 9 – Template inicial boas vindas e Horário de atendimento do <i>chatbot</i>	35
Figura 10 – Template inicial boas vindas e pedidos em aberto do <i>chatbot</i>	35
Figura 11 – Template horários de atendimento do <i>chatbot</i>	36
Figura 12 – Template de cardápio do <i>chatbot</i>	37
Figura 13 – Template de produtos do <i>chatbot</i>	37
Figura 14 – Template de sabores do produto do <i>chatbot</i>	38
Figura 15 – Template de resposta para exceções do <i>chatbot</i>	39
Figura 16 – Template de recibo de pedido do <i>chatbot</i>	39
Figura 17 – Template de recibo de pedido maximizado do <i>chatbot</i>	40

LISTA DE TABELAS

Tabela 1 – Tabela comparativa de sistemas	11
Tabela 2 – <i>Product Backlog</i>	22
Tabela 3 – <i>User Stories</i>	23
Tabela 4 – <i>Sprint Backlog</i>	24
Tabela 5 – Regras de Negócio	26
Tabela 6 – Dicionário de dados - Tabela Empresas	28
Tabela 7 – Dicionário de dados - Tabela Produtos	29
Tabela 8 – Dicionário de dados - Tabela Tipos Produtos	29
Tabela 9 – Dicionário de dados - Tabela Clientes	30
Tabela 10 – Dicionário de dados - Tabela Pedidos Sabores Produtos	30
Tabela 11 – Dicionário de dados - Tabela Pedidos	31
Tabela 12 – Dicionário de dados - <i>Collection BotIO</i>	31
Tabela 13 – Dicionário de dados - <i>Collection Requests</i>	32
Tabela 14 – Escala Likert - Níveis de concordância	42
Tabela 15 – Cronograma do Projeto TCC1	43
Tabela 16 – Cronograma do Projeto TCC2	44

LISTA DE ABREVIATURAS E SIGLAS

IBOPE	Instituto Brasileiro de Opinião Pública e Estatística
BPM	Business Process Management - Gestão de Processos de Negócio
SQL	Structured Query Language - Linguagem de Consulta Estruturada
JSON	JavaScript Object Notation - Notação de Objetos JavaScript

SUMÁRIO

1	APRESENTAÇÃO GERAL DO PROJETO	8
2	DEFINIÇÃO DO PROBLEMA	10
3	OBJETIVOS	12
3.1	OBJETIVO GERAL	12
3.2	OBJETIVOS ESPECÍFICOS	12
4	ANÁLISE DE TECNOLOGIAS/FERRAMENTAS	13
4.1	LINGUAGENS DE PROGRAMAÇÃO E FRAMEWORKS	13
4.1.1	Node.js	13
4.1.2	Express framework	13
4.1.3	Mongoose	13
4.1.4	Laravel framework	14
4.1.5	Plataforma Facebook Messenger	14
4.2	BANCO DE DADOS	14
4.2.1	Mysql	14
4.2.2	Mysql workbench	14
4.2.3	MongoDB	15
4.2.4	Robomongo	15
4.3	AMBIENTE DE DESENVOLVIMENTO	15
4.3.1	Netbeans IDE	15
4.4	FERRAMENTAS DE MODELAGEM E CONTROLE DE VERSÃO	15
4.4.1	Astah	15
4.4.2	Bizagi process modeler	16
4.4.3	Balsamiq mockup	16
4.4.4	Git / Github	16
4.5	FERRAMENTAS PARA TESTE DE SOFTWARE	16
4.5.1	Postman	17
5	DESCRIÇÃO DA SOLUÇÃO	18
5.1	ASPECTOS FUNCIONAIS	18
5.2	ASPECTOS TÉCNICOS	18
6	ABORDAGEM DE DESENVOLVIMENTO	20
7	ARQUITETURA DO SISTEMA	21
7.1	MODELAGEM FUNCIONAL	21

7.1.1	Product backlog	21
7.1.2	User stories	22
7.1.3	Sprint backlog	24
7.2	MODELAGEM DE PROCESSO DE NEGÓCIO	24
7.2.1	Diagramas BPMN	25
7.2.2	Lista de regras de negócio	26
7.3	MODELAGEM DE DADOS	26
7.4	DICIONÁRIO DE DADOS	28
7.4.1	Relacional sql - Mysql	28
7.4.2	Nosql - MongoDB	31
7.5	MODELAGEM DE INTERFACE GRÁFICA DO USUÁRIO	32
8	FUNCIONAMENTO DO SISTEMA	33
8.1	MÓDULO DE ATENDIMENTO AUTOMATIZADO CHATBOT	33
8.1.1	Template Inicial de atendimento do <i>chatbot Facebook Messenger</i>	33
8.1.2	Template de boas vindas e menu inicial do <i>chatbot Facebook Messenger</i>	34
8.1.3	Template de horários de atendimento do <i>chatbot Facebook Messenger</i>	36
8.1.4	Template de cardápio do <i>chatbot Facebook Messenger</i>	36
8.1.5	Template de produtos do <i>chatbot Facebook Messenger</i>	37
8.1.6	Template de sabores do produto do <i>chatbot Facebook Messenger</i>	37
8.1.7	Template de resposta para exceções do <i>chatbot Facebook Messenger</i>	38
8.1.8	Template de recibo do pedido do <i>chatbot Facebook Messenger</i>	39
9	VALIDAÇÃO	41
9.1	ESTRATÉGIA	41
10	CRONOGRAMA	43
11	COMPONENTES REUTILIZADOS	45
	REFERÊNCIAS	46

1 APRESENTAÇÃO GERAL DO PROJETO

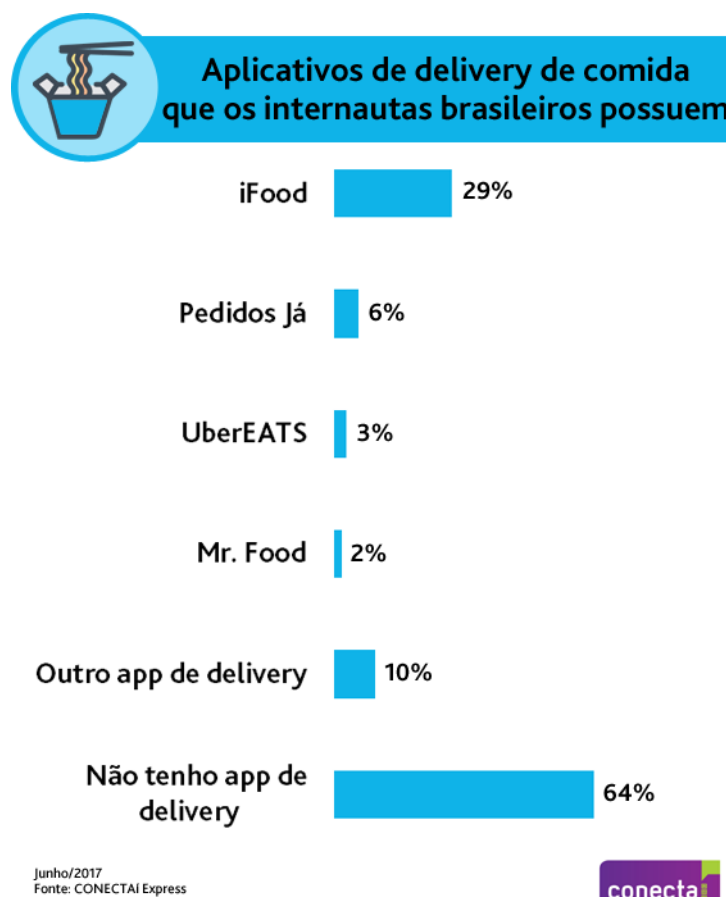
No dia a dia corrido da sociedade moderna, é comum que se busque um melhor aproveitamento do tempo, principalmente em tarefas rotineiras. A grande maioria dos indivíduos vive um constante “corre-corre”, com grande parte do seu tempo aplicado às tarefas profissionais e/ou no meio discente. Muitos acabam consumindo comida rápida e utilizando o segmento de *delivery*, palavra em inglês atualmente muito utilizada no âmbito de restaurantes, que entregam comida na casa ou no trabalho das pessoas, para se alimentar de maneira fácil e rápida.

No mercado de entrega de comida, a qualidade e disponibilidade da comunicação com cliente é essencial, sendo o primeiro contato, uma demonstração do compromisso e seriedade da empresa em fornecer seus serviços, um telefone ocupado ou um erro humano ao transcrever um pedido, pode gerar uma imagem negativa da empresa.

Com a popularização do acesso à Internet, o serviço de entrega de comida tem expandido seus meios de contato com os clientes, deixando de se manter rigidamente vinculado à comunicação telefônica e passando a utilizar outros meios, como chat de mensagens, aplicativos *mobile* e sistemas *Web* para efetuar seus pedidos.

Segundo pesquisa da IBOPE Inteligência de junho de 2017 efetuada por meio de CONECTAi Express (INTELIGÊNCIA, 2017), 36% dos internautas brasileiros possuem algum aplicativo de *delivery* em seus *smartphones* e *tablets*. A pesquisa ainda destaca que há muita oportunidade neste segmento, já que 2/3 dos internautas ainda não utilizam esse serviço, a Figura 1 apresenta o resultado da pesquisa.

Figura 1 – Pesquisa IBOPE Inteligência 2017



Fonte: (INTELIGÊNCIA, 2017)

Ao menos 64% dos brasileiros que participaram da pesquisa não utilizam nenhum aplicativo de *delivery*, grande parte opta por utilizar o método de contato por telefone, que proporciona um contato mais intimista e dá a sensação de maior compromisso com o tempo de entrega estabelecido, já que o tempo de entrega pode variar bastante utilizando um *app*.

Usuários de *smartphones* passam a maior parte do seu tempo dentro de aplicativos de *email* e de mensagens, e a maioria prefere se comunicar usando mensagens de texto. Os *chatbots* possibilitam as empresas oferecer serviços a seus usuários através dos aplicativos de mensagens, sem a necessidade de *download* ou instalação de um novo *app*. (REDAÇÃO, 2016).

Os avanços tecnológicos na última década, tornaram populares os assistentes virtuais, como a Siri(da Apple) e o assistente do Google, “robôs” que interagem com seres humanos de forma natural. Aplicativos de troca de mensagens como o *Facebook Messenger*, carregam milhares de *chatbots*, robôs que interagem por meio de conversas. Com eles, já é possível fazer compras, obter informações e até pedir uma pizza. Satya Nadella, CEO da Microsoft, disse em uma conferência que “os *bots* são os novos *apps*”. E ele não está sozinho nessa. “Por que eu preciso de um *app* de previsão do tempo se posso perguntar para um bot qual a previsão? Todos os *apps* de uso muito específico vão perder espaço ou até serão extintos”, segundo Richard Chaves, diretor de novas tecnologias e inovação da Microsoft Brasil. (PRADO, 2016).

De acordo com uma pesquisa feita pela NIELSEN e encomendada pelo *Facebook*, *apps* de mensagens instantâneas devem crescer 50% nos próximos dois anos. Na pesquisa, 53% dos questionados afirmam que existe mais possibilidade de comprarem produtos ou serviços de negócios com os quais eles possam trocar mensagens (MULLER, 2016). Com a marca de 1,2 bilhão de usuários ativos por mês, atingida em junho de 2017, e *api* específicas para *chatbots*, o *Facebook Messenger* se tornou uma ferramenta promissora para atendimento a clientes.

Seguindo a proposta de usabilidade dos *chatbots* e análises de mercado, este projeto propõe desenvolver um canal de atendimento automatizado, integrado a um sistema de gestão de pedidos que permita maior disponibilidade do serviço de entrega de comida. Trazendo uma experiência prática e funcional, juntamente com a comodidade do uso do popular *Facebook Messenger* para efetuar os pedidos, eliminando assim a necessidade de utilização de um aplicativo voltado apenas a pedidos, tornando o acesso multiplataforma já que o *chat* também pode ser acessível de qualquer navegador que possibilite o *login* na rede social.

2 DEFINIÇÃO DO PROBLEMA

Buscando aproveitar melhor seu tempo, muitas pessoas acabam consumindo serviços que lhes agreguem praticidade ao dia a dia. Pedir comida no conforto de casa ou no trabalho é um hábito cada vez mais comum entre consumidores brasileiros. Uma pesquisa feita pelo Broadcast, conduzido pelo *delivery* Ifood, mostrou que 56% das pessoas fazem pedidos semanais. Nesse número, 14% realizam pedidos mais de duas vezes por semana; 18% duas vezes; e 24% fazem um pedido por semana. Ainda dentre os entrevistados pela pesquisa foi constatado que dentre os motivos para fazerem pedidos por meios digitais, destacam-se a facilidade de uso (64%), agilidade do cardápio (55%) e não precisar falar com um atendente (51%). (NEWS, 2016). Segundo a Associação Brasileira de Bares e Restaurantes (ABRASEL), só em 2015 o mercado de *delivery* totalizou um aumento de R\$ 1 bilhão em relação ao ano anterior, boa parte desse crescimento foi alavancado pela tecnologia, que traz comodidade para quem compra.

Diante desse cenário, a aplicação de novas tecnologias na relação com o cliente, é essencial para melhorar o atendimento e conquistar sua preferência.

Seguindo as tendências tecnológicas, o atendimento por *chatbots* já é uma realidade no Brasil. A onda nascida na China com o WeChat, da Tencent, foi importada para o Ocidente pelo *Facebook*, e vem sendo aplicada por diversas empresas para usos diversos como via de relacionamento com clientes e vendas.

No Brasil o uso do *chatbot* pelo *Facebook Messenger* se torna uma das melhores alternativas para o uso da tecnologia, “Temos um problema de conexão com a Internet e de limitação de dispositivos móveis, em tanta memória e processador. Não dá para ter vários aplicativos no celular. O *app* de bate-papo é o mais leve e fácil de conectar”, explica Gustavo Sperb, do Superplayer. “Acostumamos a usar a interface de mensagem. É mais rápido e intuitivo. E tem uma inteligência artificial que vai aprendendo o gosto do usuário e entregando o que ele gosta.” (GOMES, 2016).

Este projeto tem como foco trazer uma interface de comunicação natural para pedidos de entrega de comida, com a implementação de um canal de atendimento por *chat* de mensagem, de forma que a experiência de contato do cliente, seja mais familiar e produtiva possível, adicionando valor ao serviço prestado pela empresa.

Após pesquisas na Internet por softwares para gestão de entrega de comida, foram selecionados alguns dos sistemas com maior similaridade como o projeto proposto, porém, nenhum *software* apresenta um atendimento automatizado pela plataforma *Facebook Messenger* com abrangência de diversos segmentos do ramo entrega de comida. A seguir apresenta-se brevemente a descrição do sistemas e suas funcionalidades de destaque.

O IGO PIZZAS é um sistema de gestão exclusivamente para pizzarias, com ferramenta de gestão *Web* e possibilidade de se efetuar pedido por aplicativo *mobile* específico ou por robô do *Facebook Messenger*, sendo este último associado à página da empresa contratante do serviço. (IGOPIZZAS, 2017).

O FOODBOOKING é um sistema de pedidos e gestão para restaurantes, pizzarias, lanchonetes e bares. Os pedidos podem ser recebidos pelo site, página do *Facebook*, aplicativos móveis e por telefone de forma integrada ao sistema de gestão, onde também é possível efetuar o controle de compras, despesas e relatórios. (FOODBOOKING, 2017).

O IFOOD é um serviço de *delivery* de refeições acessível por um *app* gratuito, disponível para *smartphones* com *Android*, *iOS* e *Windows Phone*, ou pela *Web*. Flexível a diversos segmentos do mercado de entrega de comida, permite o *login* através da conta do *Facebook*. A busca por estabelecimentos pode ser filtrada por tipo de comida e/ou por localização, cada restaurante tem os tipos de pagamento que aceita e, por isso, fica a critério do usuário. Em alguns, há a opção de fazer o pagamento direto pelo *app*, usando cartão. (JESUS, 2016).

Para uma melhor visão dos sistemas concorrentes, foram efetuados testes e leitura da documentação disponível dos principais serviços de entrega de comida do Brasil. A pesquisa levou em consideração os recursos citados nas fontes pesquisadas dos sistemas analisados, para a criação da tabela abaixo.

Tabela 1 – Tabela comparativa de sistemas.

Características	Food Booking	Igo Pizzas	Ifood	Hunger Management
Atendimento automatizado Chatbot		X		X
Gestão de vendas	X	X	X	X
Cupom fiscal eletrônico	X			
Interface gestão web	X	X	X	X
Controle de entregadores			X	X
Acesso mobile	X	X	X	X
Acompanhamento de status do pedido		X	X	X
Integração google maps para taxa de entrega		X	X	X
Pagamento online	X		X	X
Multi segmentos	X		X	X

Fonte: Criado pelo autor do projeto, 2017.

Conforme a Tabela 1 pode se perceber que, os principais pontos de destaque do Hunger Management em comparação com os demais sistemas são o atendimento automatizado por *chatbot* em uma plataforma que permita a utilização em diversos segmentos do ramo de entrega de comida. O sistema Hunger Management não contempla funcionalidade para administrar as tributações legais da empresa, o que não torna necessário a implementação de cupom fiscal eletrônico, um diferencial no sistema Food Booking.

Sendo assim, o projeto proposto visa oferecer aos clientes um atendimento intuitivo, onde o robô possa agilizar a execução do pedido ou responder dúvidas relativas ao serviço prestado pela empresa.

3 OBJETIVOS

Apresentam- se nesta seção os objetivos gerais e específicos do projeto proposto.

3.1 OBJETIVO GERAL

O objetivo deste projeto é desenvolver um sistema *Web* para gestão de entregas de comida e um *WebService* para tratamento das informações enviadas pelo *Webhook* da plataforma *facebook messenger*, possibilitando que clientes efetuem pedidos através de atendimento automatizado por *chatbot*, permitindo rapidez e alta disponibilidade do atendimento.

3.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos do projeto são apresentados abaixo.

- a) Proporcionar uma melhor experiência usabilidade ao cliente de serviço de entrega de comida;
- b) Aumentar a disponibilidade e agilidade no atendimento;
- c) Possibilitar de forma simplificada aos gestores do negócio a extração de métricas de quantidade vendas por data, valor vendido por data, clientes com mais pedidos efetuados e extração de lista de clientes para ações de marketing.
- d) Reduzir o custo de atendentes humanos;
- e) Reduzir o custo de telefonia disponibilizando um canal adicional para atendimento.

4 ANÁLISE DE TECNOLOGIAS/FERRAMENTAS

Nesta seção estão descritas as tecnologias que estão sendo utilizadas no projeto.

4.1 LINGUAGENS DE PROGRAMAÇÃO E FRAMEWORKS

Esta subseção apresenta as linguagens de programação que estão sendo utilizadas no desenvolvimento do projeto.

4.1.1 Node.js

NODE.JS é uma plataforma em tempo de execução para desenvolvimento de aplicações *sever-side*, construída pela Google sobre o interpretador de JavaScript de código aberto V8 Javascript Engine para Chrome, com o objetivo de permitir uma maneira fácil de se desenvolver programas de rede escaláveis. Sua principal vantagem é o uso do modelo I/O direcionada a evento não bloqueante que o torna leve e eficiente, o NODE.JS vem crescendo desde sua criação em 2009, e evoluiu até a poderosa e crescentemente popular estrutura de desenvolvimento assíncrono, usada para a criação de aplicativos JavaScript altamente escalonáveis. (LHRIG, 2014).

O NODE.JS foi utilizado neste projeto para o desenvolvimento do *Web Service* responsável pela integração entre o *WebHook* da plataforma *Facebook Messenger* e o sistema *Web* de gestão de entregas e pedidos HUNGER MANAGEMENT.

O *Webhook* da plataforma *Facebook Messenger*, permite que o *WebService* receba informações sempre que houver atualizações de um determinado conjunto de tópicos e seus campos. As notificações de atualização do *Webhook* são enviadas como solicitações POST para uma URL de retorno de chamada devidamente cadastrada. As notificações podem ser leves, indicando apenas que um campo foi atualizado, ou podem incluir o valor recentemente atualizado.

4.1.2 Express framework

O Express é um Framework para aplicações *Web* do NODE.JS, mínimo e flexível, ele fornece um conjunto robusto de recursos para aplicativos *Web* e móveis. Com uma grande lista de utilitários HTTP e *middleware* disponíveis, como um sistema de *views* intuitivo em MVC, sistema de roteamento robusto e um executável para criação de aplicações (BROWN, 2014). Express Framework é um componente chave para a criação de aplicações *Web* dinâmicas junto a NODE.JS, e foi utilizado neste projeto para o desenvolvimento do *Web Service* como ferramenta facilitadora na criação das estruturas de resposta a solicitações HTTP.

4.1.3 Mongoose

Mongoose é uma biblioteca do *Nodejs* que proporciona uma solução baseada em esquemas para modelar os dados da sua aplicação. Ele possui sistema de conversão de tipos, validação, criação de consultas e *hooks* para lógica de negócios, fornecendo um mapeamento de objetos do MongoDB similar ao ORM (*Object Relational Mapping*), ou ODM (*Object Data Mapping*) no caso do Mongoose. Isso significa que o Mongoose traduz os dados do banco de dados para objetos *JavaScript* para que possam ser utilizados por sua aplicação. (ADMIN, 2016).

A biblioteca Mongoose foi utilizada para facilitar a modelagem das operações de CRUD (acrônimo de *Create*, *Read*, *Update* e *Delete* na língua Inglesa) na base MongoDB.

4.1.4 Laravel framework

É um framework PHP (linguagem de *script open source* de uso geral) (SECURE.PHP.NET, 2017) que utiliza o padrão de arquitetura MVC (Model-view-controller), utilizado para simplificar o desenvolvimento e aumentar a produtividade, que possui entre suas principais características a possibilidade de desenvolver aplicações seguras e performáticas com rapidez. O LARAVEL utiliza uma *engine* de *template* chamada BLADE, com uma gama de ferramentas para interface amigáveis e funcionais, para se comunicar com o banco de dados utiliza o ELOQUENT ORM, uma implementação simples do ActiveRecord e o COMPOSER como seu gerenciador de dependências. (OTWELL; BORGES, 2013).

LARAVEL Framework será utilizado neste projeto para o desenvolvimento do sistema Web de gestão de entregas e pedidos, HUNGER MANAGEMENT.

4.1.5 Plataforma Facebook Messenger

A plataforma *Facebook Messenger*, é formada por um conjunto de API (*Application Programming Interface*) que implementam funcionalidades para integração com *Webhook* externos, onde podem ser enviadas e recebidas informações do usuário de contato do *messenger*, em uma simples conversação. Possui *Chat Extensions* que permitem a criação de *bot* para conversação, tornando possível que múltiplas pessoas conversem com o mesmo contato ao mesmo tempo através de um *chatbot*.

A ferramenta foi escolhida devido a sua vasta documentação, quantidade de API e popularidade do aplicativo *Facebook Messenger*, que já possui mais de um bilhão de usuários. (MESSENGER, 2017).

4.2 BANCO DE DADOS

Esta subseção apresenta as ferramentas para banco de dados que estão sendo utilizadas no desenvolvimento do projeto.

4.2.1 Mysql

O MYSQL é um sistema gerenciador de banco de dados (SGBD) relacional de código aberto, que utiliza a linguagem SQL (Structure Query Language – Linguagem de Consulta Estruturada) para inserir, acessar e gerenciar o conteúdo armazenado. (PISA, 2012). Foi utilizado devido a sua simplicidade de utilização e manutenção, e por ser o banco de dados relacional disponível na hospedagem escolhida.

4.2.2 Mysql workbench

O MYSQL WORKBENCH é uma ferramenta visual para *design*, desenvolvimento e administração de base de dados MySQL. Essa ferramenta é originária do DBDesigner. A ferramenta possui as opções de abrir conexão, editar dados, editar *scripts* SQL, gerenciar conexões, novo modelo de dados, modelo de dados a partir da base (engenharia reversa), modelo de dados de um *script* SQL, criação de instâncias de servidor, importação/exportação de base, gerenciamento de segurança e gerenciamento de instâncias. (VESPA, 2010).

A ferramenta MYSQL WORKBENCH (*Community Edition* – Edição da comunidade), foi escolhida por disponibilizar as principais funcionalidades para desenvolvimento do modelo ER (Entidade relacional) do projeto proposto, de forma gratuita na versão utilizada.

4.2.3 MongoDB

O MongoDB é um banco de dados orientado a documentos de alta performance, *open source* e *schema-free*, escrito em C++. Ele é uma mistura entre os repositórios escaláveis baseados em chave/valor e a tradicional riqueza de funcionalidades dos bancos relacionais. (NASCIMENTO, 2010).

MongoDB foi utilizado para persistência dos dados utilizados para a tomada de decisões do *chatbot* e para armazenar os pedidos de forma temporária até serem finalizados e enviados a base Mysql.

4.2.4 Robomongo

O Robomongo é um *client* de consulta performático para bancos não relacionais de alta performance MongoDB. Ele possibilita configurar várias conexões e realiza as consultas de forma bem simples e intuitiva, possibilitando até mesmo vários modos de visualização e exportação das *queries* construídas no console pra te auxiliar no desenvolvimento das suas aplicações. (FIDELIS, 2016).

O Robomongo foi utilizado para gerenciamento da base MongoDB através de interface gráfica.

4.3 AMBIENTE DE DESENVOLVIMENTO

Esta subseção apresenta a ferramenta de ambiente de desenvolvimento que foi utilizada no desenvolvimento do projeto.

4.3.1 Netbeans IDE

O NETBEANS é um ambiente de desenvolvimento integrado (IDE-Integrated Development Environment) gratuito e de código aberto. O NETBEANS IDE possui suporte para instalação em diversas plataformas, como Windows, Linux, Solaris e MacOS, sendo fácil de instalar e usar, e suporta o desenvolvimento em JAVA, PHP, HTML, JAVASCRIPT, CSS, GROOVY e C++, possuindo uma grande quantidade de *plug-ins* a disposição, oferecendo aos desenvolvedores todas as ferramentas necessárias para criação de aplicativos profissionais de *desktop*, empresariais, *Web* e móveis. (OFICINA, 2008).

O NETBEANS IDE foi utilizado como ferramenta para codificação do módulo do sistema *Web* HUNGER MANAGEMENT em LARAVEL PHP e do *Web Service* em NODE.JS e EXPRESSJS, foi escolhido devido a vasta biblioteca de *plug-ins* e linguagens suportadas, além de oferecer uma interface amigável e prática ao desenvolvedor.

4.4 FERRAMENTAS DE MODELAGEM E CONTROLE DE VERSÃO

Esta subseção apresenta as ferramentas para modelagem e controle de versão que estão sendo utilizadas no desenvolvimento do projeto.

4.4.1 Astah

ASTAH Community é um *software* para modelagem UML (Unified Modeling Language – Linguagem de Modelagem Unificada) com suporte a UML2. O ASTAH apresenta uma interface amigável dividida em várias seções, cada uma com sua respectiva finalidade, permite a exportação de diagramas completos nos formatos JPEG e PNG. (LIMA, 2011). O *software* foi desenvolvido

no Japão na plataforma JAVA, o que garante sua portabilidade para qualquer plataforma que possui JVM (Máquina Virtual Java). Obteve o prêmio “Produto de Software do Ano 2006”, pela Agência de Promoção de Informação Tecnológica no Japão. Anteriormente conhecido como JUDE, ele funciona nas plataformas Windows, Mac e Linux. (STARTUPSTARS, 2015). Foi escolhido por possuir os artefatos necessários para a elaboração da documentação técnica do sistema.

4.4.2 Bizagi process modeler

BIZAGI PROCESS MODELER é uma ferramenta freeware de gestão de processos ágil, fácil de utilizar que permite desenhar, diagramar, documentar e publicar os processos utilizando o padrão BPMN (*Business Process Model and Notation*). Dirigido a profissionais que estejam vinculados ou interessados no desenvolvimento de projetos BPM (*Business Process Management*), é amplamente utilizado por Analistas de processos, Designers de processos e Desenvolvedores BPM. (LIMA, 2016).

BIZAGI PROCESS MODELER foi escolhido devido ser gratuito e contar com todas as funcionalidades necessárias para o desenvolvimento dos processos de negócio do projeto proposto.

4.4.3 Balsamiq mockup

Balsamiq Mockups é uma ferramenta de *wireframing* rápida que ajuda você a trabalhar mais rápido e de forma mais inteligente. Ele reproduz a experiência de esboçar em um quadro branco, mas usando um computador. (BALSAMIQ, 2017).

Desenvolvido pela Balsamiq Studio é uma aplicação desenvolvida na linguagem de programação *ActionScript*, que executa adobe AIR (*Adobe Integrated Runtime*). Essa aplicação é utilizada para desenvolver protótipos ou modelos (*mockups*), como as telas de um sistema *desktop*, ou sistema/páginas *web* ou *mobile*. (MALHERBI, 2016).

A ferramenta BALSAMIQ MOCKUP foi escolhida para a prototipação das telas do projeto proposto, devido a sua interface de fácil utilização, gratuidade e por apresentar as funcionalidade necessárias para a criação do *wireframes*.

4.4.4 Git / Github

GIT é um sistema de controle de versão distribuído livre e de código aberto, projetado para facilitar e controlar o desenvolvimento em equipe, permitindo que diversas pessoas possam contribuir simultaneamente. Foi desenvolvido por Linus Torvalds para o desenvolvimento do Kernel Linux, e passou posteriormente a ser adotado por muitos outros projetos. (GIT-SCM, 2017).

GITHUB é um serviço *Web* que oferece diversas funcionalidades extras aplicadas ao GIT. Permite a hospedagem gratuita de projetos pessoais facilmente visualizados e explorados pela interface para navegador. GITHUB foi escolhido para hospedar as versões do projeto devido a sua fácil integração com o NETBEANS e seu modo colaborativo pela *Web*. (SCHMITZ, 2015).

4.5 FERRAMENTAS PARA TESTE DE SOFTWARE

Esta subseção apresenta as ferramentas para teste de software que estão sendo utilizadas no desenvolvimento do projeto.

4.5.1 Postman

O POSTMAN é um client REST criado pelo programador Abhinav Asthana e está disponível na forma de extensão para o *Google Chrome*. Ele permite realizar requisições HTTP a partir de uma interface simples e intuitiva, facilitando o teste e depuração de serviços REST. (RODRIGUES, 2017).

A ferramenta POSTMAN esta sendo utilizada neste projeto para envio de dados no formato JSON ao *webservice* de forma a validar o comportamento e resposta do serviço.

5 DESCRIÇÃO DA SOLUÇÃO

Esta seção descreve os aspectos funcionais e técnicos da solução.

5.1 ASPECTOS FUNCIONAIS

O pedido pode ser realizado através qualquer dispositivo com acesso ao *Facebook Messenger* ou por ligação telefônica, não sendo necessário o download de nenhum aplicativo exclusivo para contato com a empresa. Ao agregar um *chatbot* para atendimento, a solução busca reduzir o tempo de solicitação do pedido automatizando o processo, além de um maior controle sobre as vendas da empresa. Com o sistema registrando informações de cada ação, será possível gerar relatórios precisos de forma imediata e ampliar a visão do gestor sobre o negócio.

Acessando através do seu *Facebook* a página da rede social da empresa, o cliente tem acesso para enviar uma mensagem e iniciar o atendimento, podendo visualizar horário de atendimento, efetuar um pedido ou ver pedidos em aberto. Ao ser finalizado o pedido pelo cliente, o robô registra a solicitação que é atualizada na fila de pedidos do sistema *Web* de gestão, onde é visualizada pelos responsáveis pela produção. Depois de produzido e liberado pelo atendente, é destinado ao entregador com a forma de pagamento escolhida. Se o cliente utilizou o *chatbot* para efetuar seu pedido, o sistema envia uma notificação automaticamente ao receber o status de saída para entrega.

O sistema *Web* de gestão possui três níveis de acesso, definidos conforme sua utilização no processo de venda, sendo: um nível administrador (criação de usuários, alteração de preços, cadastro de novos produtos, sabores, descontos promocionais, entregadores, etc.), um nível espectador (este possui acesso apenas á visualização de suas informações de usuário e da fila de pedidos para produção) e um nível atendente (responsável pelo atendimento e cadastro de pedidos efetuados por telefone).

5.2 ASPECTOS TÉCNICOS

O projeto proposto é constituído por dois módulos: um sistema de gestão com uma interface gráfica *Web* para usuário a ser desenvolvida em LARAVEL PHP, utilizando elementos responsivos do BOOTSTRAP nativo do *framework*, e um *Web Service* possuindo uma camada de negócio para a lógica do *chatbot*, de forma a gerenciar as requisições POST e respostas ao *WebHook* do *Facebook Messenger* e utilizando uma camada DAO (Data Access Object), permitindo separar regras de negócio das regras de acesso ao banco de dados. A tecnologia de *Webhook* utilizada pelo *Facebook Messenger* permite que o *Webservice* receba informações passivamente apenas quando o cliente interagir com o *chat*.

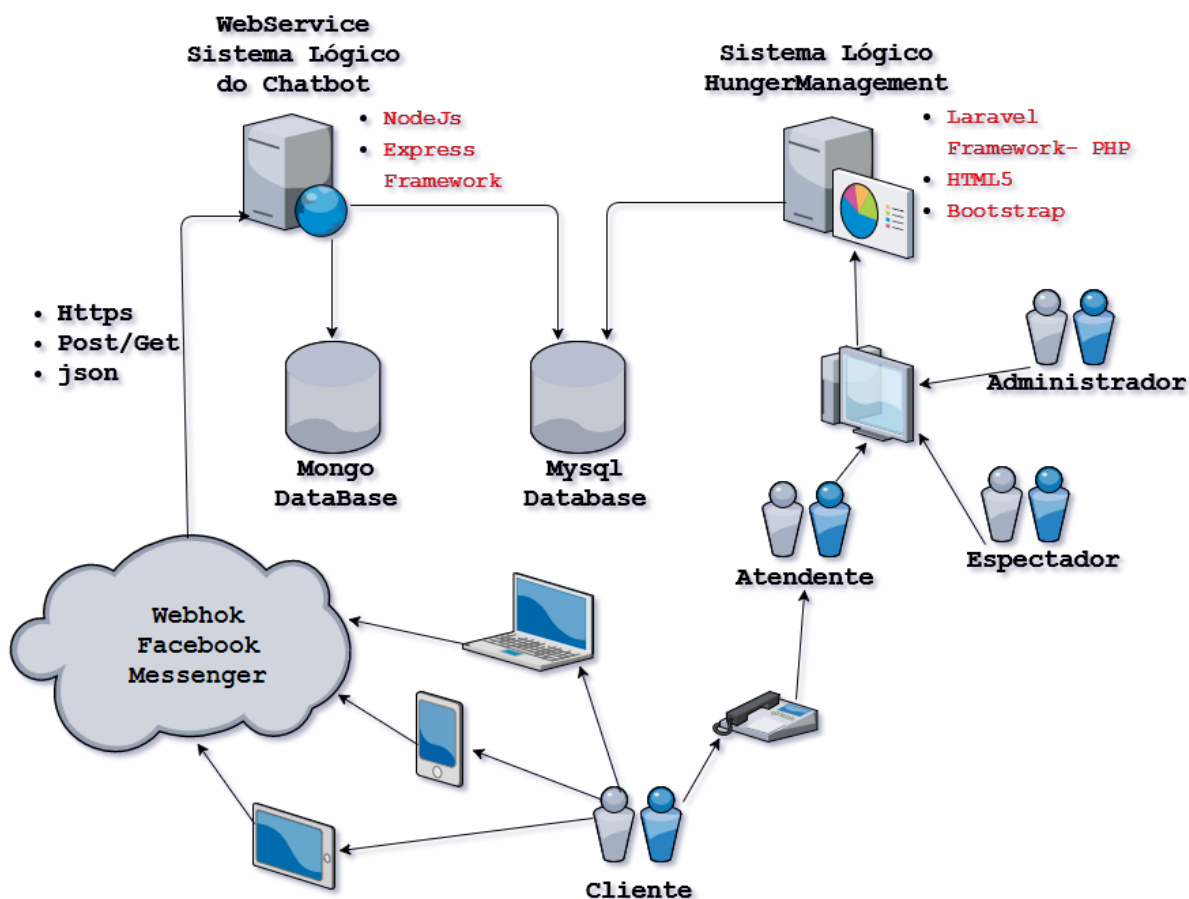
Webhooks permitem que sistemas externos recebam notificações de todos os eventos que ocorrem no sistema. Quando um evento acontece, o sistema envia uma requisição HTTP POST para a URL configurada no *webhook* com as informações relativas ao evento. Ao receber a notificação, o sistema pode executar diversas tarefas, dependendo do objetivo da integração.

Os *webhooks* também são chamados de *Callbacks* ou *Reverse API*.
(BOLETOSIMPLES, 2017).

É utilizada uma base de dados *Nosql MongoDB* para armazenar informações para tomada de decisão do *chatbot*, pedidos em andamento e expressões a serem tratadas futuramente para resposta, enquanto as informações da camada de negócio são armazenados em uma base de dados relacional *Mysql*.

A Figura 2, apresenta a ilustração com a visão técnica da solução com o objetivo de demonstrar as conexões entre os componentes.

Figura 2 – Visão técnica da solução



Fonte: Criado pelo autor do projeto, 2017.

Como se pode verificar na Figura 2, a comunicação com o *Webservice* utilizará o protocolo de transferência de hipertexto seguro HTTPS, uma implementação do protocolo HTTP sobre uma camada adicional de segurança que utiliza o protocolo SSL/TLS, enquanto a troca de dados ocorrerá através de JSON.

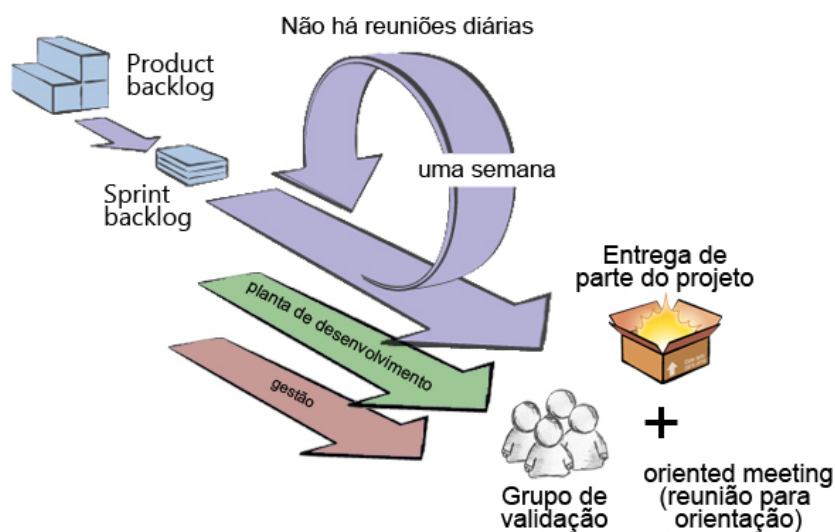
JSON é um modelo para armazenamento e transmissão de informações no formato texto. Apesar de muito simples, tem sido bastante utilizado por aplicações *Web* devido a sua capacidade de estruturar informações de uma forma bem mais compacta do que a conseguida pelo modelo XML, tornando mais rápido o parsing dessas informações. (DEV MEDIA, 2017).

6 ABORDAGEM DE DESENVOLVIMENTO

Para o desenvolvimento do projeto, esta sendo utilizada a metodologia ágil Scrum solo, elaborada para desenvolvedores que trabalham individualmente (solo). Criado em 2012, caracterizasse como um processo iterativo e incremental que une as boas práticas delineadas pelo *Personal Software Process* (PSP) e pelo Scrum. (PAGOTTO et al., 2016). A abordagem de desenvolvimento foi selecionada por, permitir a identificação de prioridades dentro do projeto, a entrega ao final de cada iteração de uma parte, e o desenvolvimento solo do projeto proposto.

Na Figura 3 pode-se ver o ciclo do Scrum que foi utilizado e adaptado para o desenvolvimento do projeto.

Figura 3 – Fluxo de processos metodologia ágil Scrum Solo



Fonte: (SCRUMSOLO, 2016)

O Scrum é um *framework* ágil para gerenciamento de projetos que se destaca por sua abordagem enxuta (*lean*) de desenvolvimento, utilizando equipes pequenas e multidisciplinares permitindo um melhor controle. O projeto é dividido em vários *sprints* (ciclos curtos de desenvolvimento) consecutivos que ocorrerão de acordo com a prioridade do *product owner* (proprietário do produto). Cada período de *sprint* é definido, geralmente, entre duas e quatro semanas, e durante esse período o *scrum team* se dedica para ter um conjunto pequeno de funcionalidades codificadas e testadas.

No início do projeto, foi efetuada uma breve análise do sistema a ser desenvolvido, obtendo-se os objetivos a serem atingidos e as necessidades do usuário, bem como as principais *user stories* a serem desenvolvidas. A partir das necessidades, foi possível escrever o *Backlog* do Produto e planejar as entregas e o desenvolvimento do sistema.

Estão sendo utilizados neste projeto os artefatos *Product Backlog* para a listagem das funcionalidades desejadas para o projeto, *User Stories* para detalhamento das necessidades do usuário que devem ser implementadas e *Sprint Backlog* que listarão as tarefas a serem selecionadas para cada *Sprint*. Foi determinado que a duração das Sprints sejam de duas semanas, tempo estimado para que sejam entregue as funcionalidades validadas e com maior qualidade a fim de se reduzir os riscos negativos. No início de cada ciclo serão efetuadas reuniões de planejamento da *Sprint*, onde serão selecionadas as *User Stories* a serem implementadas no período.

7 ARQUITETURA DO SISTEMA

Esta seção apresenta os artefatos que documentam a modelagem do sistema, ajudando a descrever a função, arquitetura e o design da aplicação.

7.1 MODELAGEM FUNCIONAL

Foram elaborados ao longo do desenvolvimento do projeto proposto, os artefatos *Product Backlog*, *User Stories* e *Sprint Backlog* para a modelagem e desenvolvimento das funcionalidades apresentadas até o momento.

7.1.1 Product backlog

O *Product Backlog* foi elaborado com base nas principais funcionalidades desejadas para o projeto proposto. Nele consta um código para identificação, uma breve descrição das funcionalidades e sua prioridade, devidamente determinada junto ao orientador do projeto, de forma a serem executadas primeiramente as tarefas necessárias para a obtenção de um produto mínimo funcional. A Tabela 2 que exibe as informações do *Product Backlog*.

Tabela 2 – *Product Backlog*.

Product Backlog			
ID	Nome	Descrição	Prio.
BL01	Efetuar pedido e visualizar informações pelo <i>chatbot Facebook Messenger</i>	Permite que os clientes possam verificar pedidos em aberto, efetuar o pedido e obter informações de horário de atendimento.	Alta
BL02	Visualizar fila de pedidos	Permite ao utilizador do módulo de gestão a visualização da fila de pedidos e seus respectivos status.	Alta
BL03	Cadastro de pedidos por atendente humano	Permite o cadastro do pedido de forma manual por um atendente humano.	Alta
BL04	Cadastro de Produtos	Permite o cadastro de produtos.	Alta
BL05	Cadastro de Sabores	Permite o cadastro de sabores dos produtos.	Alta
BL06	Cadastro de Usuário	Permite o cadastro de usuário e definição de um perfil de acesso.	Média
BL07	Autenticação por login e senha	Exige a autenticação de usuário a senha para acesso ao módulo de gestão.	Média
BL08	Cadastro de Entregadores	Permite o cadastro de entregadores.	Média
BL09	Aviso de expedição do pedido pelo <i>chatbot Facebook Messenger</i>	Permite o aviso automático da expedição do pedido em caso do mesmo ter sido efetuado pelo canal de atendimento <i>chatbot Facebook Messenger</i> .	Baixa
BL10	Métricas de vendas por canal de atendimento e período	Permite visualizar as quantidades de vendas por canal de atendimento e período.	Baixa
BL11	Integração com o Google Maps para cálculo da taxa de entrega por km	Permite o cálculo automático do frete por km de forma integrada com o Google Maps, com base no valor do km estabelecido por parâmetro.	Baixa

Fonte: Criado pelo autor do projeto, 2017.

7.1.2 User stories

As *User Stories* (em português Histórias de Usuário), descrevem os objetivos do usuário a serem atingidos através do projeto proposto. Detalham as ações desejadas, utilizando três parâmetros da necessidade do usuário: “QUEM”, “O QUÊ” e “POR QUÊ”. As *User Stories* são desenvolvidas com base na lista apresentada no *Product Backlog* do projeto. A Tabela 7.1.2 exibe de forma ordenada as *User Stories* criadas até o momento.

Tabela 3 – User Stories.

User Stories		
ID	Backlog	Descrição
US01	BL01	Como um cliente do serviço de entrega eu quero verificar o horário de atendimento através do chatbot da empresa no Facebook Messenger para verificar a disponibilidade do serviço.
US02	BL01	Como um cliente do serviço de entrega eu quero verificar os pedidos em aberto através do chatbot da empresa no Facebook Messenger para .
US03	BL01	Como um cliente do serviço de entrega eu quero efetuar um pedido de entrega de comida através do chatbot da empresa no Facebook Messenger.
US04	BL02	Como atendente usuário do sistema de gestão eu quero visualizar a fila de pedidos.
US05	BL03	Como um usuário atendente do sistema de gestão eu quero cadastrar o pedido manualmente pela tela de cadastro de pedidos do sistema informando os dados necessários.
US06	BL04	Como um usuário administrador do sistema de gestão eu quero cadastrar o produto na tela de cadastro de produtos do sistema informando os dados necessários.
US07	BL05	Como um usuário administrador do sistema de gestão eu quero cadastrar o sabor do produto na tela de cadastro de sabores de produtos do sistema informando os dados necessários.
US08	BL06	Como um usuário administrador do sistema de gestão eu quero cadastrar o usuário na tela de cadastro de usuários do sistema informando os dados necessários.
US09	BL07	Como usuário do sistema de gestão eu quero efetuar login no sistema utilizando usuário e senha cadastrados.
US10	BL08	Como usuário administrador do sistema de gestão eu quero cadastrar o entregador na tela de cadastro de entregadores do sistema informando os dados necessários.
US11	BL09	Como um cliente do serviço de entrega eu quero receber um aviso informando a saída e o entregador do pedido de comida por mim efetuado através do chatbot da empresa no Facebook Messenger.
US12	BL10	Como um usuário administrador do sistema de gestão eu quero visualizar os dados das quantidades de vendas por canal de atendimento e período através da tela de métricas do sistema.
US13	BL11	Como um usuário administrador do sistema de gestão eu quero cadastrar o valor do km para cálculo da taxa de entrega na tela de cadastro de parâmetros de negócio do sistema informando os dados necessários.

US14	BL11	Como um usuário administrador ou atendente do sistema de gestão eu quero visualizar o valor da taxa de entrega calculado com base no endereço do pedido na tela de pedidos do sistema informando os dados necessários.
US15	BL11	Como um cliente do serviço de entrega eu quero visualizar o valor da taxa de entrega calculado com base no endereço do pedido mim efetuado através do chatbot da empresa no Facebook Messenger.

Fonte: Criado pelo autor do projeto, 2017.

7.1.3 Sprint backlog

Com base no levantamento das *User Stories*, foram determinadas *Sprints* que possuem uma ou mais Histórias agrupadas. Conforme a Tabela 4, cada *Sprint* possui uma data de início e término para controle de entrega.

Tabela 4 – *Sprint Backlog*.

Sprint Backlog		
Sprint	Período	Atividades
00	12/09 a 26/09	Montagem do ambiente de desenvolvimento e elaboração do <i>Product Backlog</i> , <i>User Stories</i> e <i>Sprint Backlog</i>
01	26/09 a 10/10	Início do desenvolvimento da US03: Desenvolvimento do <i>backend</i> e criação dos templates do menu inicial, tipos de produtos, produtos, sabores e pedido.
02	10/10 a 24/10	Fim do desenvolvimento da US03: Desenvolvimento do algoritmo para tratamento das mensagens de texto e <i>postback</i> recebidas pelo <i>chatbot</i> .
03	24/10 a 07/11	Desenvolvimento da US01: Desenvolvimento do <i>backend</i> de controle de horário de atendimento do <i>chatbot</i> e do <i>backend</i> e <i>template</i> de verificação de horário para o cliente. Desenvolvimento da US02: Desenvolvimento do <i>backend</i> e do <i>template</i> para exibir as informações de pedidos em aberto ao cliente.
04	07/11 a 21/11	Desenvolvimento da US04: Criação do <i>backend</i> e <i>frontend</i> da fila de pedidos em atendimento.
05	21/11 a 05/12	Desenvolvimento da US09: Criação do <i>backend</i> e <i>frontend</i> de login de usuário do sistema de gestão.
06	05/12 a 19/12	Desenvolvimento da US05: Criação do <i>backend</i> e <i>frontend</i> para o CRUD de cadastro de pedidos manualmente.

Fonte: Criado pelo autor do projeto, 2017.

7.2 MODELAGEM DE PROCESSO DE NEGÓCIO

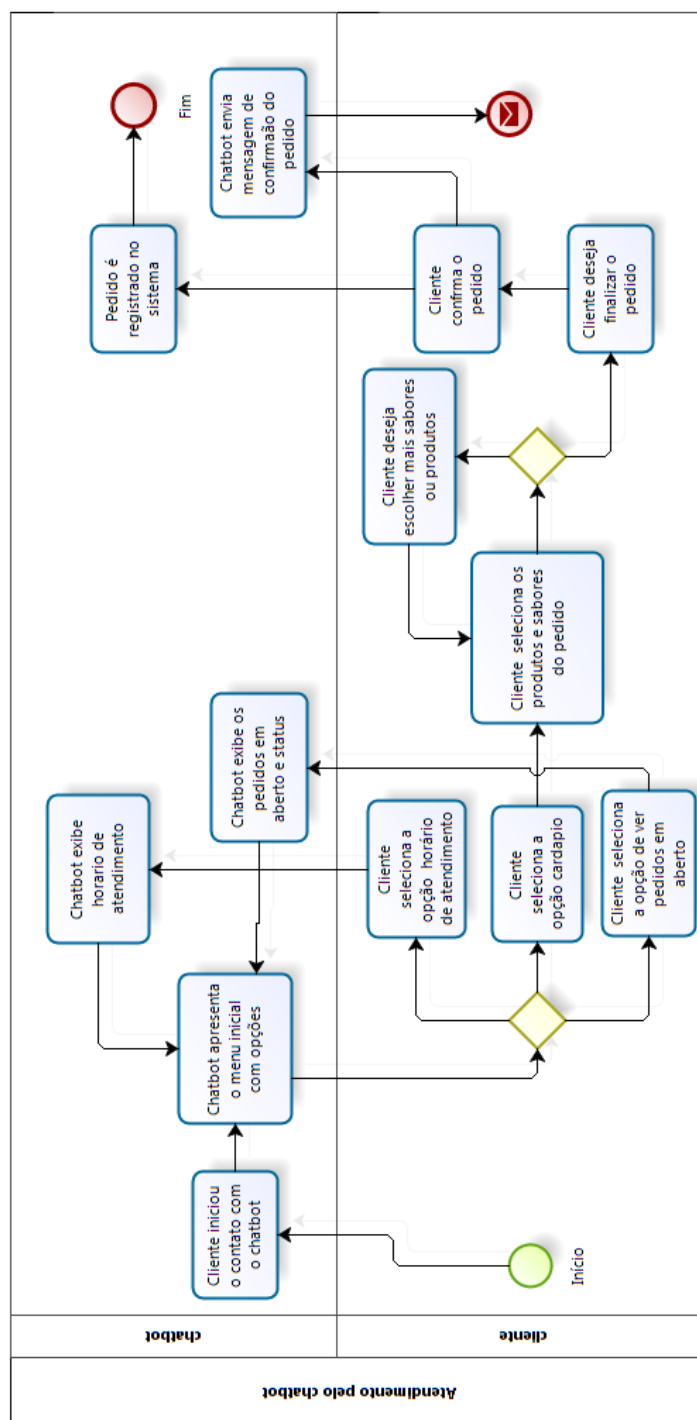
Para a modelagem de processo de negócio foi elaborada uma lista com as regras de negócio.

7.2.1 Diagramas BPMN

Para uma melhor compreensão e modelagem dos processos de negócio foram desenvolvidos diagramas BPMN (*Business Process Modeling Notation*) que descrevem a lógica passo a passo dos processos por meio de desenhos.

A Figura 4 descreve o fluxo principal de atendimento pelo *chatbot*.

Figura 4 – Diagrama BPM - Fluxo principal de atendimento do *chatbot*



Fonte: Criado pelo autor do projeto, 2017.

7.2.2 Lista de regras de negócio

Foram criadas regras de negócio para complementar as *User Stories*, a Tabela 5 lista as regras criadas até o momento.

Tabela 5 – Regras de Negócio.

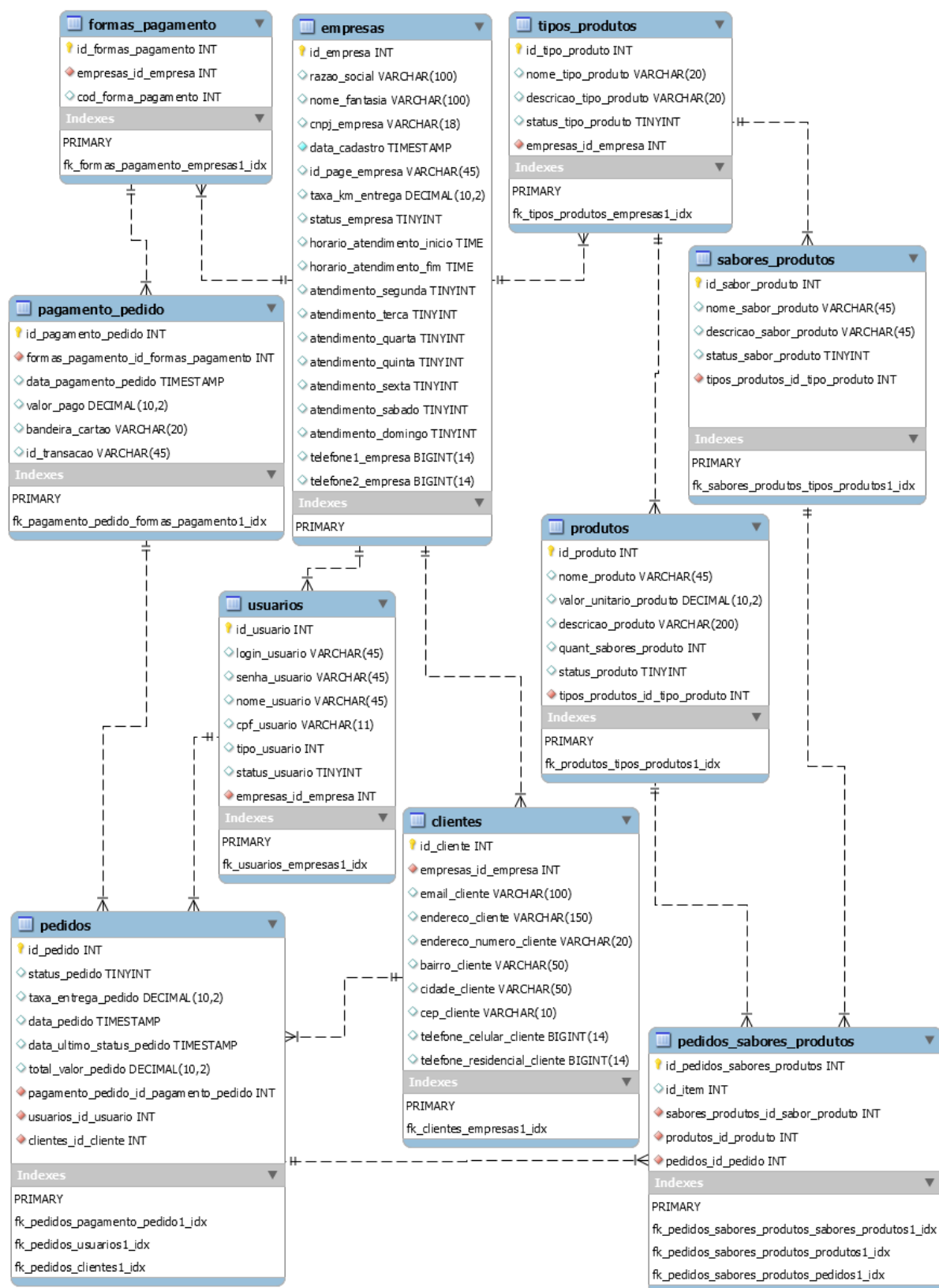
ID	Nome	Descrição
RN01	Concluir o pedido	Pedido só será concluído após confirmação pelo cliente
RN02	Status de pedido aberto	Pedidos abertos possuirão status ordenados com a seguinte descrição: 1=Pendente, 2=Em Produção, 3=Expedido e 4=Entregue
RN03	Abertura de pedido	Deverá ser selecionado pelo menos um item do catálogo para abertura do pedido
RN04	Status usuário	Usuários com status Desativado não poderão efetuar login no sistema de gestão
RN05	Situação do pedido	Pedidos realizados possuirão a sinalização de apenas uma das seguintes situações: 1=Aberto, 2=Cancelado ou 3=Finalizado
RN06	Criação de Usuário	A criação de novos usuários só poderá ser efetuada pelo usuário administrador do sistema
RN07	Horário para atendimento pelo chatbot	Cliente só poderá efetuar pedidos pelo chatbot se horário e dia forem permitidos pela empresa no sistema.

Fonte: Criado pelo autor do projeto, 2017.

7.3 MODELAGEM DE DADOS

A Figura 6 apresenta o diagrama de entidades e relacionamentos que representam o banco de dados relacional *Mysql* da aplicação.

Figura 5 – Diagrama de Entidade Relacional



Fonte: Criado pelo autor do projeto, 2017.

7.4 DICIONÁRIO DE DADOS

As Tabelas e figuras apresentadas nesta subseção, detalham a estrutura de persistência de dados desenvolvidas para o projeto até o momento. Para armazenagem dos dados foram utilizadas duas ferramentas com diferentes paradigmas de funcionamento, uma base de dados relacional *Mysql* e uma base de dados orientada a documentos *Nosql MongoDB*.

7.4.1 Relacional sql - Mysql

Esta subseção apresenta o detalhamento dos tipos de informações armazenadas nas tabelas do banco relacional *Mysql*.

Tabela 6 – Tabela Empresas.

Coluna	Tipo	Não Nulo	Valor Padrão
id_empresa	Int	Sim	Auto Incremento
razão_social	Varchar(100)	Não	
nome_fantasia	Varchar(100)	Não	
cnpj_empresa	Varchar(18)	Não	
data_cadastro	Timestamp	Sim	Current Timestamp
id_page_empresa	Varchar(45)	Não	
taxa_km_entrega	Decimal(10,2)	Não	
status_empresa	Tinyint	Não	1
horario_atendimento_fim	Time	Não	
horario_atendimento_inicio	Time	Não	
atendimento_segunda	Tinyint	Não	0
atendimento_terca	Tinyint	Não	0
atendimento_quarta	Tinyint	Não	0
atendimento_quinta	Tinyint	Não	0
atendimento_sexta	Tinyint	Não	0
atendimento_sabado	Tinyint	Não	0
atendimento_domingo	Tinyint	Não	0
telefone1_empresa	Bigint(14)	Não	
telefone2_empresa	Bigint(14)	Não	

Fonte: Criado pelo autor do projeto, 2017.

Tabela 7 – Tabela Produtos.

Coluna	Tipo	Não Nulo	Valor Padrão
id_produto	Int	Sim	Auto Incremento
nome_produto	Varchar(45)	Não	
valor_unitario_produto	Decimal(10,2)	Não	
descricao_produto	Varchar(200)	Não	
quant_sabores_produto	Int	Não	
status_produto	Tinyint	Não	1
tipos_produtos_id_tipos_produtos	Int	Não	

Fonte: Criado pelo autor do projeto, 2017.

Tabela 8 – Tabela Tipos Produtos.

Coluna	Tipo	Não Nulo	Valor Padrão
id_tipo_produto	Int	Sim	Auto Incremento
nome_tipo_produto	Varchar(20)	Não	
descricao_tipo_produto	Varchar(200)	Não	
status_tipo_produto	Tinyint	Não	1
empresas_id_empresa	Int	Não	

Fonte: Criado pelo autor do projeto, 2017.

Tabela 9 – Tabela Clientes.

Coluna	Tipo	Não Nulo	Valor Padrão
id_cliente	Int	Sim	Auto Incremento
empresas_id_empresa	Int	Não	
id_page_cliente	Varchar(45)	Não	
email_cliente	Varchar(100)	Não	
endereço_cliente	Varchar(150)	Não	
endereço_numero_cliente	Varchar(20)	Não	
bairro_cliente	Varchar(50)	Não	
cidade_cliente	Varchar(50)	Não	
cep_cliente	Varchar(10)	Não	
telefone_celular_cliente	Bigint(14)	Não	
telefone_residencial_cliente	Bigint(14)	Não	

Fonte: Criado pelo autor do projeto, 2017.

Tabela 10 – Tabela Pedidos Sabores Produtos.

Coluna	Tipo	Não Nulo	Valor Padrão
id_pedidos_sabores_produtos	Int	Sim	Auto Incremento
id_item	Int	Não	
sabores_produtos_id_sabor_produto	Int	Não	
produtos_id_produto	Int	Não	
pedidos_id_pedidos	Int	Não	

Fonte: Criado pelo autor do projeto, 2017.

Tabela 11 – Tabela Pedidos.

Coluna	Tipo	Não Nulo	Valor Padrão
id_pedido	Int	Sim	Auto Incremento
status_pedido	Tinyint	Não	
taxa_entrega_pedido	Decimal(10,2)	Não	
data_pedido	Timestamp	Não	
data_ultimo_status_pedido	Timestamp	Não	
total_valor_pedido	Decimal(10,2)	Não	
pagamento_pedido_id_pagamento_pedido	Int	Não	
usuario_id_usuario	Int	Não	
clientes_id_cliente	Int	Não	

Fonte: Criado pelo autor do projeto, 2017.

7.4.2 Nosql - MongoDB

Esta subseção apresenta o detalhamento dos tipos de informações e estruturas *JSON* armazenadas no banco de dados *Nosql MONGODB*.

Tabela 12 – Collection BotIO.

Nome	Tipo
_id	ObjectId
outCommand	Int32
inPutString	Array[String]
weight	Int32

Fonte: Criado pelo autor do projeto, 2017.

Tabela 13 – *Collection Requests*.

Nome	Tipo
_id	ObjectId
user_Id	String
page_Id	String
company_Id	Number
product_Id	Number
product_name	String
product_value	Number
date	Date
flavors_product	Array[Number]

Fonte: Criado pelo autor do projeto, 2017.

7.5 MODELAGEM DE INTERFACE GRÁFICA DO USUÁRIO

Esta subseção apresenta os *wireframes* de baixa fidelidade confeccionados até o momento para a interface do módulo de gestão web.

Figura 6 – *Wireframe* da tela de fila de pedidos

A Web Page

https://hungermanagement.com.br

Nome da Empresa Sistema de Gestão HungerManagement

+ CadastrarPedido

Cliente

N° Pedido	Status	Situação	Horario	Cliente	Pedido	Entregador
123545	Aberto	Expedido	20:00	Fulano	Pizza 45 cm + Porção Batata + Refrigerante coca-cola	Zeca..
123546	Aberto	Expedido	20:02	Huguinho	Pizza 45 cm + Porção Batata + Refrigerante coca-cola	Manuel...
123547	Aberto	Em Produção	20:02	Zézinho	Pizza 45 cm + Porção Batata + Refrigerante coca-cola	
123548	Aberto	Em produção	20:06	Luizinho	Pizza 45 cm + Porção Batata + Refrigerante coca-cola	
123549	Aberto	Em produção	20:10	Pedro	Pizza 45 cm + Porção Batata + Refrigerante coca-cola	
123550	Aberto	Pendente	20:12	Jesus	Pizza 45 cm + Porção Batata + Refrigerante coca-cola	
123551	Aberto	Pendente	20:12	João	Pizza 45 cm + Porção Batata + Refrigerante coca-cola	
123552	Aberto	Pendente	20:20	Fulano	Pizza 45 cm + Porção Batata + Refrigerante coca-cola	

Logo empresa

Nome do Usuário

Clientes

Entregadores

Administração

Sair

Fonte: Criado pelo autor do projeto, 2017.

8 FUNCIONAMENTO DO SISTEMA

Esta seção apresenta as funcionalidades do sistema desenvolvidas até o momento.

8.1 MÓDULO DE ATENDIMENTO AUTOMATIZADO CHATBOT

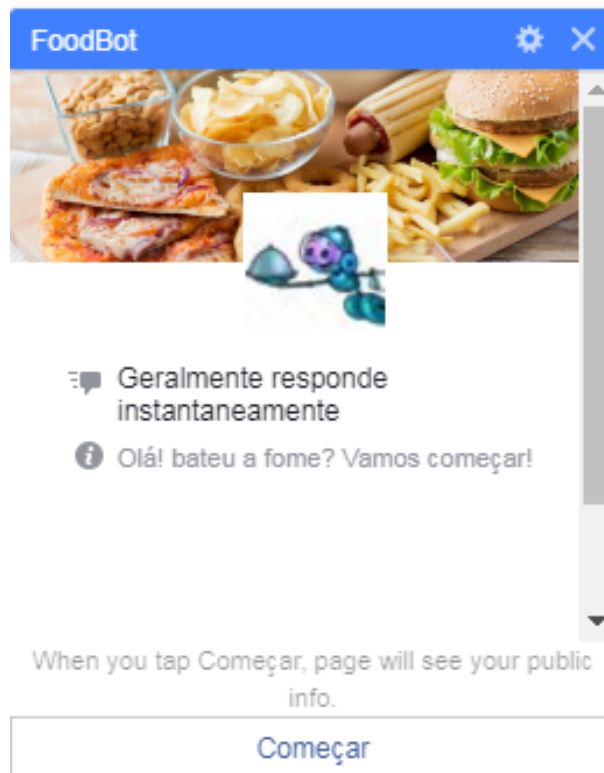
Esta subseção apresenta as interfaces de usuário elaboradas para o *chatbot* acessível pelo *Facebook Messenger*. A *api* conta com padrões de *template* pré estabelecidos, que permitem adaptações conforme a necessidade de interação com o usuário. Para a tomada de decisões do robô de atendimento foi desenvolvido um algoritmo diferenciado para os dois tipos de ações retornadas pela *api* do *Facebook Messenger*, *message* e *postback*.

- * Message: A ação do tipo *message* retorna o texto da mensagem digitada pelo remetente. Esta ação foi tratada no algoritmo para identificar palavras chave das frases e direcionar o usuário a um ponto específico no fluxo de atendimento, levando em conta o peso de importância do comando.
- * Postback: A ação do tipo *postback* é o retorno da ação sobre o botão utilizado no *template* de interface com o usuário, contendo informações como o *payload* que foi usado neste projeto para retorno de um JSON contendo informações como o comando e dados pertinentes a ação. As retornadas, direcionam o usuário a um ponto específico no fluxo de atendimento.

8.1.1 Template Inicial de atendimento do *chatbot Facebook Messenger*

A Figura 7 exibe o *template* inicial para primeira interação com o *chatbot*, possuindo uma breve mensagem de boas vindas para o usuário e um botão de ação "Começar".

Figura 7 – Template inicial *chatbot*

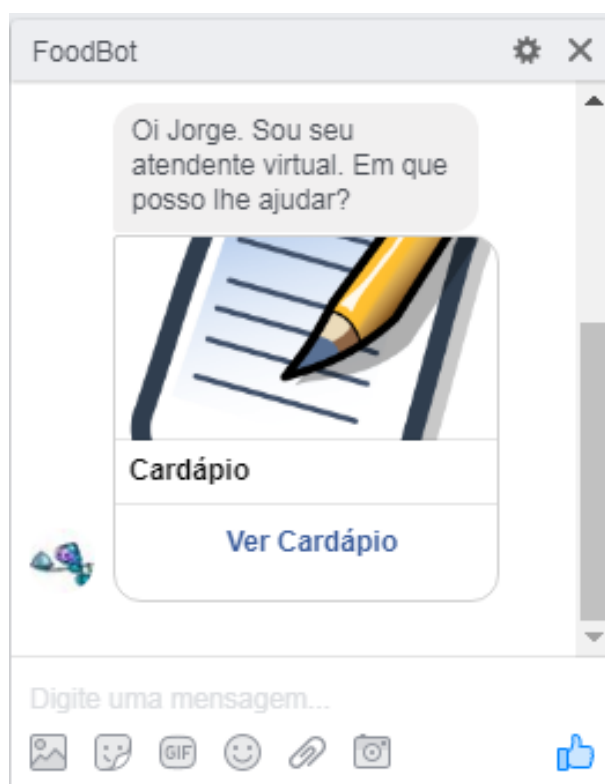


Fonte: Criado pelo autor do projeto, 2017.

8.1.2 Template de boas vindas e menu inicial do *chatbot Facebook Messenger*

A Figura 8 exibe o *template* de boas vinda com uma mensagem contendo o nome do respectivo usuário que efetuou o acesso, também exibe em primeiro momento a opção de ver cardápio. O menu inicial é acessível pelo botão começar ou através de mensagem escrita contendo alguma instrução de saudação como por exemplo: oi, boa tarde, boa noite, bom dia.

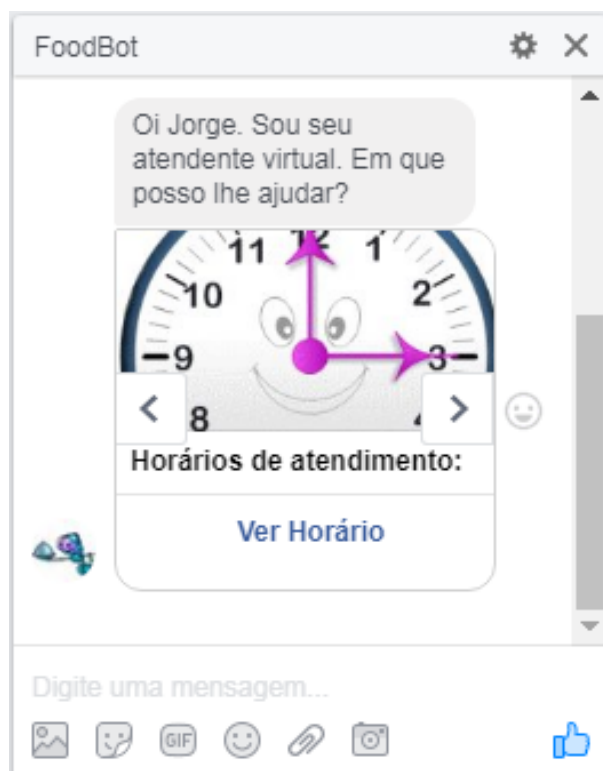
Figura 8 – Template inicial boas vindas e ver cardápio do *chatbot*



Fonte: Criado pelo autor do projeto, 2017.

A Figura 9 exibe a opção ver horário de atendimento, acessível através dos botões de rolagem horizontal.

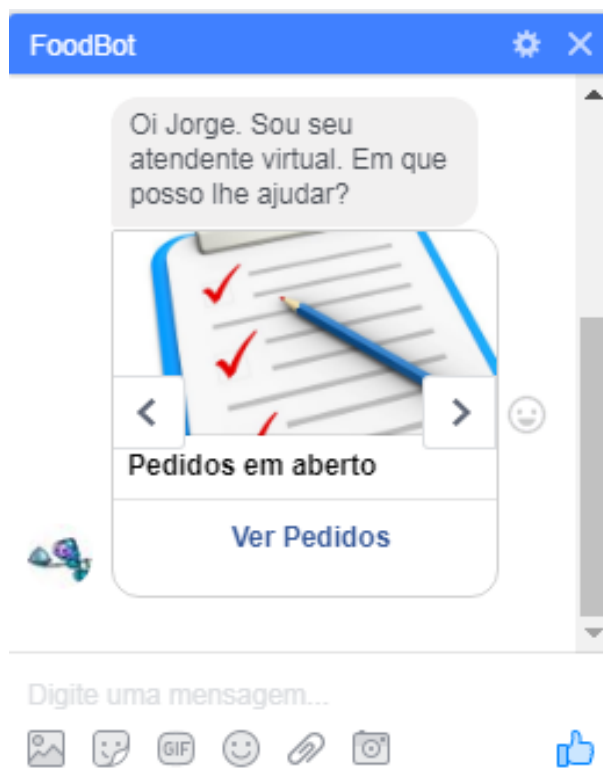
Figura 9 – Template inicial boas vindas e Horário de atendimento do *chatbot*



Fonte: Criado pelo autor do projeto, 2017.

A Figura 10 exibe a opção ver pedidos em aberto do usuário, acessível através dos botões de rolagem horizontal.

Figura 10 – Template inicial boas vindas e pedidos em aberto do *chatbot*

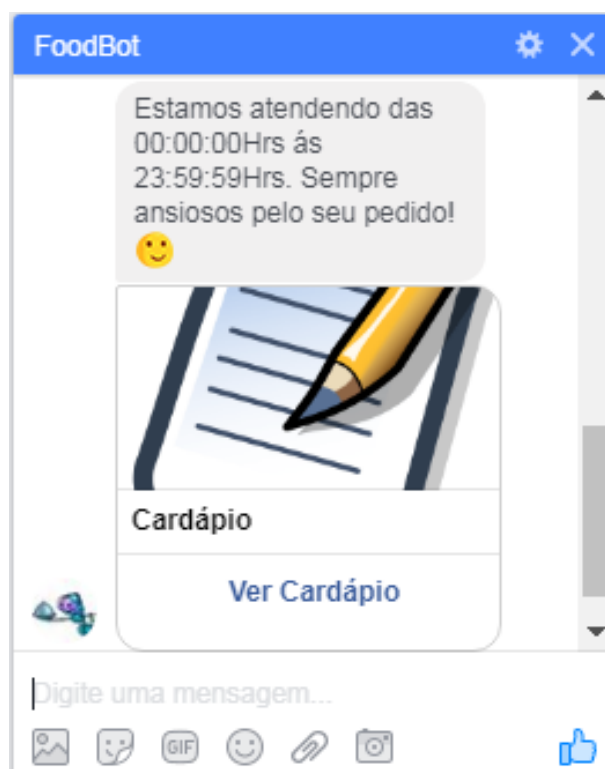


Fonte: Criado pelo autor do projeto, 2017.

8.1.3 Template de horários de atendimento do *chatbot Facebook Messenger*

A Figura 11 exibe a mensagem contendo o horário de atendimento da empresa, que é acessível pelo usuário através do menu inicial opção ver horário ou através de mensagem escrita contendo a instrução horário. Após exibir o horário de atendimento é exibido o menu inicial de atendimento.

Figura 11 – Template horários de atendimento do *chatbot*

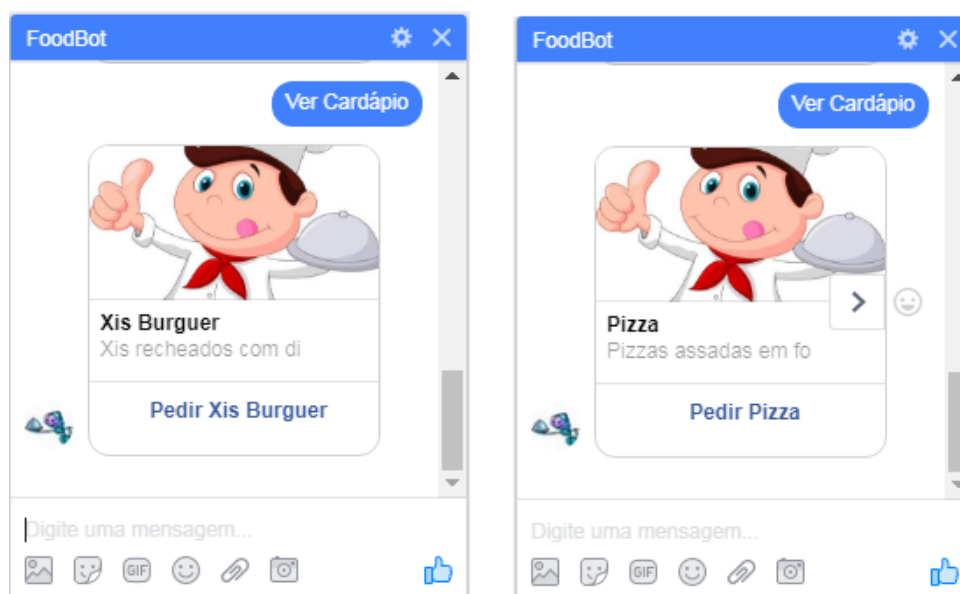


Fonte: Criado pelo autor do projeto, 2017.

8.1.4 Template de cardápio do *chatbot Facebook Messenger*

A Figura 12 exibe o *template* de cardápio contendo os tipos de produtos cadastrados para a empresa, as opções podem ser visualizados rolando o menu horizontalmente.

Figura 12 – Template de cardápio do *chatbot*

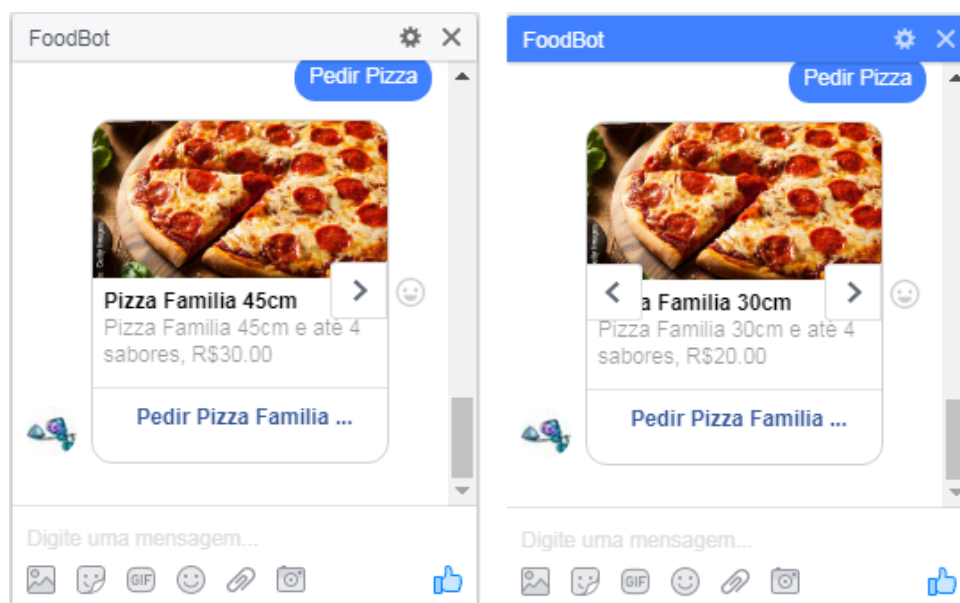


Fonte: Criado pelo autor do projeto, 2017.

8.1.5 Template de produtos do *chatbot Facebook Messenger*

A Figura 13 exibe o *template* de produtos cadastrados para a empresa, os produtos oferecidos podem ser visualizados rolando o menu horizontalmente.

Figura 13 – Template de produtos do *chatbot*



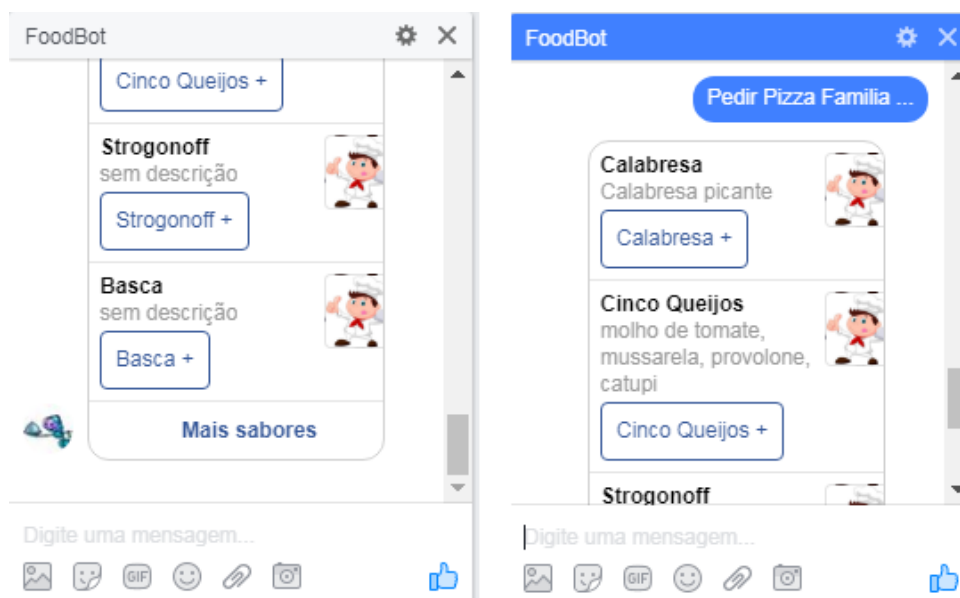
Fonte: Criado pelo autor do projeto, 2017.

8.1.6 Template de sabores do produto do *chatbot Facebook Messenger*

A Figura 14 exibe o *template* de sabores do produto selecionado conforme cadastrado para a empresa, os sabores oferecidos podem ser visualizados rolando o menu verticalmente.

Caso o usuário não encontre o sabor desejado na primeira listagem, é possível percorrer os demais sabores através do botão mais sabores.

Figura 14 – Template de sabores do produto do *chatbot*

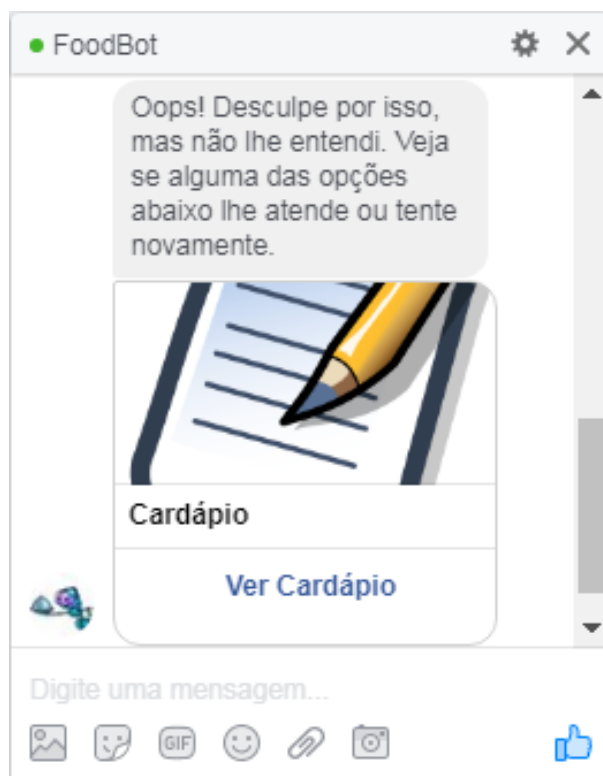


Fonte: Criado pelo autor do projeto, 2017.

8.1.7 Template de resposta para exceções do *chatbot Facebook Messenger*

A Figura 15 exibe a mensagem informando que o *chatbot* não entendeu a solicitação do usuário e sugere em seguida as opções do cardápio da empresa no *template* de cardápio. Esta resposta é utilizada se o cliente informou termos desconhecidos para o *chatbot*.

Figura 15 – Template de resposta para exceções do *chatbot*

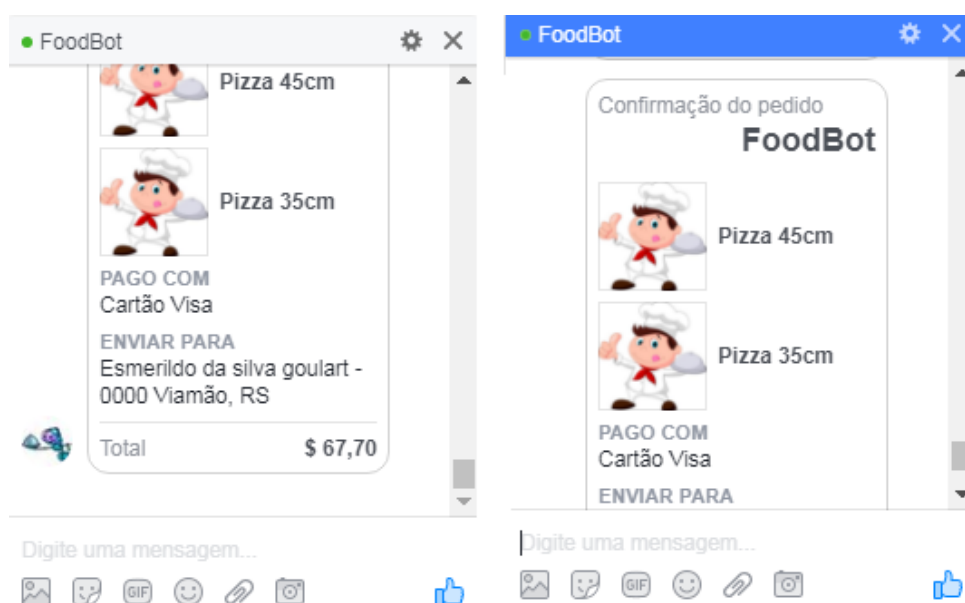


Fonte: Criado pelo autor do projeto, 2017.

8.1.8 Template de recibo do pedido do *chatbot Facebook Messenger*

A Figura 16 exibe o *template* de recibo do pedido após ser confirmado pelo cliente.

Figura 16 – Template de recibo de pedido do *chatbot*




Fonte: Criado pelo autor do projeto, 2017.


A Figura 17 exibe o *template* de recibo do pedido de forma maximizada após o cliente clicar no recibo da mensagem.

Figura 17 – Template de recibo de pedido maximizado do *chatbot*

Detalhes do pedido

FoodBot

 Pizza 45cm
\$40.30

 Pizza 35cm
\$20.30

SOLICITADO EM
Sábado, 4 de novembro 18:25

ENVIAR PARA
Jorge Abranti
Esmerildo da silva goulart - 0000
Viamão, RS 94420290

PAGO COM
Cartão Visa

RESUMO	
Subtotal	\$ 60,70
Manuseio e frete	\$ 7,00
Imposto estimado	\$ 0,00
Total	\$ 67,70

Número do pedido: 12345678902

Fonte: Criado pelo autor do projeto, 2017.

9 VALIDAÇÃO

Esta seção descreve a estratégia de validação adotada para o projeto proposto.

9.1 ESTRATÉGIA

Para validar se os objetivos do projeto estão sendo alcançados, as funcionalidades desenvolvidas passam por uma verificação ao final de cada *Sprint*, validando se as *User Stories* implementadas agregam para o cumprimento da solução proposta, avaliando sua adequação funcional com base na norma ISO/IEC 25010 presente no conjunto de modelo de qualidade da ISO/IEC 25000 também conhecida como norma SQuaRe (System and Software Quality Requirements and Evaluation).

O modelo de qualidade da ISO/IEC 25000 é uma evolução da série de normas ISO/IEC 9126 e 14598, e tem como objetivo principal fornecer uma visão geral do produto de *software*. O modelo especifica seis características de qualidade (funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade), onde estão sendo consideradas na avaliação apenas as relevantes para o *software* do projeto proposto. (BERTUOL, 2014).

A adequação funcional das *User Stories* implementadas em cada *Sprint*, avalia a capacidade da funcionalidade em atender as necessidades implícitas e explícitas, quando utilizada em situações específicas e leva em consideração as seguintes sub-características.

- a) **Completude funcional:** grau em que o conjunto de funções do produto realiza as tarefas especificadas, alcançando aos objetivos.
- b) **Correção funcional:** capacidade de fornecer resultados corretos, com o grau de precisão necessário.
- c) **Apropriação funcional:** capacidade do conjunto de funções do produto em facilitar a realização de tarefas específicas e objetivas.

Para verificar se os objetivos do projeto proposto foram alcançados, ao final do desenvolvimento da aplicação será disponibilizada uma versão para usuários e fornecedores de serviço de entrega de comida. Com base nas sub-características de usabilidade da norma ISO/IEC 25010, e compreendendo que “a aceitabilidade das aplicações Web por seus usuários depende diretamente da usabilidade da aplicação, que pode ser melhorada através da realização de inspeções” (VAZ et al., 2008), serão aplicados questões referentes a usabilidade utilizando-se do padrão de escala LIKERT para avaliação, o qual apresenta uma série de cinco proposições, das quais o inquirido deve selecionar apenas uma nos permitindo medir as atitudes e conhecer o grau de conformidade do entrevistado com qualquer afirmação proposta, e assim definir o grau de satisfação com a solução. (AMARO et al., 2004/2005).

As sub-características definidas para a avaliação de usabilidade são:

- a) **Operabilidade:** É o grau ao qual um produto possui atributos que o tornem fácil de operar e controlar.
- b) **Facilidade de uso:** É o grau em que uma interface de usuário permite a interação de forma agradável e satisfatória.

A Tabela 14 descreve os níveis de concordância a serem utilizados no questionário de satisfação.

Tabela 14 – *Escala Likert - Níveis de concordância.*

Nível	Descrição da opção
1	Discordo totalmente
2	Discordo parcialmente
3	Não concordo, nem discordo
4	Concordo parcialmente
5	Concordo totalmente

Fonte: Criado pelo autor do projeto, 2017.

10 CRONOGRAMA

Tabela 15 – Cronograma do Projeto TCC1

TRABALHO DE CONCLUSÃO 1		
Atividade	Data	Detalhamento descritivo
Documentação	21/08/2017	Entrega do formulário de aceite
Documentação	11/09/2017	Entrega da Proposta do Plano de Trabalho
Desenvolvimento Sprint 00	12/09 a 26/09	Montagem do ambiente de desenvolvimento elaboração do <i>Product Backlog</i> , <i>User Stories</i> e <i>Sprint Backlog</i> .
Desenvolvimento Sprint 01	26/09 a 10/10	Início do desenvolvimento da US03: Desenvolvimento do <i>backend</i> e criação dos templates do menu inicial, tipos de produtos, produtos, sabores e pedido.
Desenvolvimento Sprint 02	10/10 a 24/10	Fim do desenvolvimento da US03: Desenvolvimento do algoritmo para tratamento das mensagens de texto e <i>postback</i> recebidas pelo <i>chatbot</i> .
Desenvolvimento Sprint 03	24/10 a 07/11	Desenvolvimento da US01: Desenvolvimento do <i>backend</i> de controle de horário de atendimento do <i>chatbot</i> e do <i>backend</i> e <i>template</i> de verificação de horário para o cliente. Desenvolvimento da US02: Desenvolvimento do <i>backend</i> e do <i>template</i> para exibir as informações de pedidos em aberto ao cliente.
Desenvolvimento Sprint 04	07/11 a 21/11	Desenvolvimento da US04: Criação do <i>backend</i> e <i>frontend</i> da fila de pedidos em atendimento.
Desenvolvimento Sprint 05	21/11 a 05/12	Desenvolvimento da US09: Criação do <i>backend</i> e <i>frontend</i> de login de usuário do sistema de gestão.
Desenvolvimento Sprint 06	05/12 a 19/12	Desenvolvimento da US05: Criação do <i>backend</i> e <i>frontend</i> para o CRUD de cadastro de pedidos manualmente.

Fonte: Criado pelo autor do projeto, 2017.

Tabela 16 – Cronograma do Projeto TCC2

TRABALHO DE CONCLUSÃO 2		
Atividade	Data	Detalhamento descritivo
Planejamento	06/03 a 20/03	Revisão das observações da avaliação de TCC1
Desenvolvimento Sprint 07	20/03 a 03/04	Desenvolvimento da US06: Desenvolvimento do <i>backend</i> e <i>frontend</i> do CRUD de cadastro de produtos. Desenvolvimento da US07: Desenvolvimento do <i>backend</i> e <i>frontend</i> do CRUD de cadastro de sabores de tipo produto.
Desenvolvimento Sprint 08	03/04 a 17/04	Desenvolvimento da US08: Desenvolvimento do <i>backend</i> e <i>frontend</i> do CRUD de cadastro de usuários. Desenvolvimento da US10: Desenvolvimento do <i>backend</i> e <i>frontend</i> do CRUD de cadastro de entregadores.
Documentação	16/04/2018	Entrega do Relatório de Projeto Atualizado
Apresentação	23/04 a 27/04	Seminário de Andamento
Desenvolvimento Sprint 09	01/05 a 15/05	Desenvolvimento da US11: Desenvolvimento do <i>backend</i> e <i>template</i> do aviso de saída do pedido para entrega pelo <i>chatbot</i> . Desenvolvimento da US12: Desenvolvimento do <i>backend</i> e <i>frontend</i> dos relatórios de vendas por canal de atendimento. Desenvolvimento da US15: Desenvolvimento do <i>backend</i> e <i>template</i> para calculo e visualização da taxa de entrega por endereço do cliente pelo <i>chatbot</i> .
Desenvolvimento Sprint 10	15/05 a 29/05	Desenvolvimento da US13: Desenvolvimento do <i>backend</i> e <i>frontend</i> para cadastro de taxa de entrega por km. Desenvolvimento da US14: Desenvolvimento do <i>backend</i> e <i>frontend</i> do calculo da taxa de entrega com base no endereço do cliente.
Validação	29/05 a 17/06	Validação
Documentação	18/06/2018	Entrega do Relatório Final de Projeto
Apresentação	25/06 a 29/06	Banca Final de TCC
Documentação	09/07/2018	Entrega da versão final do Relatório de Projeto

Fonte: Criado pelo autor do projeto, 2017.

11 COMPONENTES REUTILIZADOS

Esta seção apresenta uma lista básica dos componentes reutilizados no projeto proposto.

BOOTSTRAP. Disponível em: <<http://getbootstrap.com/>>. Acessado em: 29 Out. 2017.

EXPRESSJS. Disponível em: <<http://expressjs.com/pt-br/>>. Acessado em: 10 Set. 2017.

LARAVEL FRAMEWORK. Disponível em: <<https://laravel.com/docs/5.0>>. Acessado em: 10 Set. 2017.

NODE.JS. Disponível em: <<https://nodejs.org/en/>>. Acessado em: 10 Set. 2017.

PLATAFORMA FACEBOOK MESSENGER. Disponível em:
<<https://developers.facebook.com/docs/messenger-platform/webhook/>>. Acessado em: 10 Set. 2017.

REFERÊNCIAS

- ADMIN. **Nodejs e MongoDB – Introdução ao Mongoose**. 2016. Disponível em: <<http://nodebr.com/nodejs-e-mongodb-introducao-ao-mongoose/>>. Acesso em: 30 Out. 2017.
- AMARO, A.; PÓVOA, A.; MACEDO, L. **A arte de fazer questionários**. [s.n.], 2004/2005. Disponível em: <<http://www.mobilizadores.org.br/wp-content/uploads/2015/03/A-arte-de-fazer-questionarios.pdf>>. Acesso em: 02 Set. 2017.
- BALSAMIQ. **Balsamiq Mockups**. 2017. Disponível em: <<https://balsamiq.com/>>. Acesso em: 20 Set. 2017.
- BERTUOL, G. Uma abordagem para avaliação da qualidade de artefatos de software. 2014. Disponível em: <<http://repositorio.ufsm.br/bitstream/handle/1/5444/BERTUOL%2C%20GELSON.pdf?sequence=1&isAllowed=y>>. Acesso em: 14 set. 2017.
- BOLETOSIMPLES. **Webhooks**. 2017. Disponível em: <<http://api.boletosimples.com.br/webhooks/>>. Acesso em: 30 Out. 2017.
- BROWN, E. **Web Development with Node and Express**. [S.l.]: O'REILLY Media, Inc., 2014.
- DEVMEDIA. **Introdução ao formato JSON**. 2017. Disponível em: <<http://www.devmedia.com.br/introducao-ao-formato-json/25275>>. Acesso em: 09 Set. 2017.
- FIDELIS, M. **Robomongo :: Um cliente muito poderoso para MongoDB**. 2016. Disponível em: <<http://www.nanoshots.com.br/2016/08/robomongo-um-cliente-muito-poderoso.html>>. Acesso em: 30 Out. 2017.
- FOODBOOKING. 2017. Disponível em: <<https://foodbooking.com.br/>>. Acesso em: 09 Set. 2017.
- GIT-SCM. **GIT - everything is local**. 2017. Disponível em: <<https://git-scm.com/>>. Acesso em: 02 Set. 2017.
- GOMES, H. S. **MEU AMIGO ROBÔ**. 2016. Disponível em: <<http://especiais.g1.globo.com/tecnologia/2016/especial-chatbots>>. Acesso em: 03 Set. 2017.
- IGOPIZZAS. 2017. Disponível em: <<https://www.igopizzas.com/>>. Acesso em: 09 Set. 2017.
- INTELIGÊNCIA, I. **Um em cada três internautas tem app de delivery de comida**. IBOPE Inteligência, 2017. Disponível em: <<http://www.ibopeinteligencia.com/noticias-e-pesquisas/um-em-cada-tres-internautas-tem-app-de-delivery-de-comida/>>. Acesso em: 03 set. 2017.
- JESUS, A. **O que é iFood? Saiba como funciona o app de delivery**. 2016. Disponível em: <<http://www.techtudo.com.br/noticias/noticia/2016/07/o-que-e-ifood-saiba-como-funciona-o-app-de-delivery.html>>. Acesso em: 09 Set. 2017.
- LHRIG, C. J. **Pro Node.js para Desenvolvedores**. [S.l.]: ditora Ciência Moderna, 2014. Tradução: Angelo Giuseppe Meira Costa (angico).
- LIMA, D. d. **Modelos softwares com Astah Community**. 2011. Disponível em: <<http://www.techtudo.com.br/tudo-sobre/astah-community.html>>. Acesso em: 01 Set. 2017.

LIMA, D. d. **Bizagi Modeler: modele processos de negócio**. 2016. Disponível em: <<http://www.techtudo.com.br/tudo-sobre/bizagi-modeler.html>>. Acesso em: 20 set. 2017.

MALHERBI, E. **Prototipação de Sistemas utilizando a Ferramenta Balsamiq Mockup**. 2016. Disponível em: <<http://www.devmedia.com.br/prototipacao-de-sistemas-utilizando-a-ferramenta-balsamiq-mockup/27232>>. Acesso em: 20 Set. 2017.

MESSANGER, F. **Plataforma do Messenger**. 2017. Disponível em: <<https://developers.facebook.com/products/messenger/m>>. Acesso em: 20 Set. 2017.

MULLER, L. **Apps de mensagens instantâneas devem crescer 50 próximos dois anos**. IBOPE Inteligência, 2016. Disponível em: <<https://www.tecmundo.com.br/mensageiros/112411-apps-mensagens-instantaneas-devem-crescer-50-proximos-dois-anos.htm>>. Acesso em: 03 set. 2017.

NASCIMENTO, J. **3 razões para usar MongoDB**. 2010. Disponível em: <<https://imasters.com.br/artigo/18334/mongodb/3-razoes-para-usar-mongodb/?trace=1519021197>>. Acesso em: 30 Out. 2017.

NEWS, F. S. **Maioria das pessoas pedem semanalmente por delivery, diz pesquisa**. FOOD SERVICE NEWS, 2016. Disponível em: <<http://www.foodservicenews.com.br/maioria-das-pessoas-pedem-semanalmente-por-delivery-diz-pesquisa>>. Acesso em: 03 set. 2017.

OFICINA, R. **O que é o NetBeans?** 2008. Disponível em: <https://www.oficinadanet.com.br/artigo/1061/o_que_e_o_netbeans>. Acesso em: 09 Set. 2017.

OTWELL, T.; BORGES, P. **Laravel: De Aprendiz a Artesão (Brazilian Portuguese)**. [S.l.]: Leanpub, 2013.

PAGOTTO, T.; FABRI, J. A.; LERARIO, A.; GONCALVES, J. A. Scrum solo: Software process for individual development. In: **2016 11th Iberian Conference on Information Systems and Technologies (CISTI)**. [S.l.: s.n.], 2016.

PISA, P. **O que é e como usar o MySql?**. FOOD SERVICE NEWS, 2012. Disponível em: <<http://www.techtudo.com.br/artigos/noticia/2012/04/o-que-e-e-como-usar-o-mysql.html>>. Acesso em: 30 Ago. 2017.

PRADO, A. L. **Chatbots vão dominar seu smartphone. Entenda por quê**. 2016. Disponível em: <<http://exame.abril.com.br/tecnologia/o-que-sao-chatbots-os-robos-que-vao-invadir-seu-smartphone/>>. Acesso em: 03 set. 2017.

REDAÇÃO, D. **6 pontos para você entender o fenômeno dos "chats bots"**. 2016. Disponível em: <<https://conteudo.startse.com.br/para-empresendedores/redacao/6-pontos-para-voce-entender-o-fenomeno-dos-chats-bots-por-quem-entende/>>. Acesso em: 29 Out. 2016.

RODRIGUES, J. **Testando serviços Web API com Postman**. 2017. Disponível em: <<http://www.linhadecodigo.com.br/artigo/3712/testando-servicos-web-api-com-postman.aspx>>. Acesso em: 30 Out. 2017.

SCHMITZ, D. **Tudo que você queria saber sobre Git e GitHub, mas tinha vergonha de perguntar.** FOOD SERVICE NEWS, 2015. Disponível em: <<https://tableless.com.br/tudo-que-voce-queria-saber-sobre-git-e-github-mas-tinha-vergonha-de-perguntar/>>. Acesso em: 02 set. 2017.

SCRUMSOLO. **SCRUMSOLO.** 2016. Disponível em: <<https://scrumsolo.wordpress.com/>>. Acesso em: 03 set. 2017.

SECURE.PHP.NET. **O que é php?** 2017. Disponível em: <https://secure.php.net/manual/pt_BR/intro-what-is.php>. Acesso em: 30 Ago. 2017.

STARTUPSTARS. **O que é Astah?** 2015. Disponível em: <<http://www.startupsstars.com/2015/10/o-que-e-o-astah-posttecnico-por-bruno-seabra>>. Acesso em: 01 set. 2017.

VAZ, V. T.; CONTE, T.; BOTT, A.; MENDES, E.; TRAVASSOS, G. H. Inspeção de usabilidade em organizações de desenvolvimento de software – uma experiência prática. 2008. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/sbqs/2008/027.pdf>>. Acesso em: 02 set. 2017.

VESPA, T. G. **MySQL Workbench.** 2010. Disponível em: <<https://thiagovespa.com.br/blog/2010/09/18/mysql-workbench/>>. Acesso em: 20 Set. 2017.