

Westerley da Silva Reis

Sistema de Diálogo em Linguagem Natural para Serviços de Atendimento ao Cliente

Corumbá

2017

Westerley da Silva Reis

Sistema de Diálogo em Linguagem Natural para Serviços de Atendimento ao Cliente

Banca da disciplina de TCC 2, apresentada
ao Curso de Sistemas de Informação da Fun-
dação Universidade Federal de Mato Grosso
do Sul, Câmpus do Pantanal, como parte dos
requisitos para a obtenção do título de Ba-
charel em Sistemas de Informação.

Fundação Universidade Federal de Mato Grosso do Sul

Curso de Sistemas de Informação

Programa de Graduação

Orientador: Prof. Ma. Lucineide Rodrigues da Silva

Corumbá

2017

Westerley da Silva Reis

Sistema de Diálogo em Linguagem Natural para Serviços de Atendimento ao Cliente

Banca da disciplina de TCC 2, apresentada
ao Curso de Sistemas de Informação da Fun-
dação Universidade Federal de Mato Grosso
do Sul, Câmpus do Pantanal, como parte dos
requisitos para a obtenção do título de Ba-
charel em Sistemas de Informação.

Trabalho aprovado. Corumbá, 31 de março de 2017:

**Prof. Ma. Lucineide Rodrigues da
Silva**
Orientadora

Adriana Viana Postigo Paravisine
Convidado 1

Murilo Machado Oliveira
Convidado 2

Corumbá
2017

Agradecimentos

Agradeço a Deus, por fazer parte deste momento da minha vida, e à minha família por estar me apoiando todos esses anos nos momentos de dificuldades.

À minha orientadora, Lucineide Rodrigues da Silva que me direcionou durante todo o desenvolvimento deste trabalho, e aos professores que fizeram parte da minha formação acadêmica.

Por fim, agradeço a todos que contribuíram comigo nesse trajeto percorrido, à minha namorada e amigos que me ajudaram durante as atividades desenvolvidas, com as correções, teste do produto final, entre outras atividades.

Resumo

Os sistemas de diálogos são aplicações de uma das áreas de estudos do Processamento de Linguagem Natural (PLN), que possibilitam a comunicação entre usuários e máquinas por meio da língua natural, utilizando um conjunto de técnicas da Inteligência Artificial (IA) para processar a escrita e a fala do usuário. Tais sistemas, cada vez mais, tornam-se tendência nos serviços de atendimento ao cliente devido à facilidade que propõem ao usuário durante a interação. Visando esta facilidade na troca de informações, o presente trabalho propõe o desenvolvimento de um sistema de diálogo que utiliza a linguagem natural para serviços de atendimento aos acadêmicos da Fundação Universidade Federal de Mato Grosso do Sul, Câmpus do Pantanal, com objetivo de fornecer informações relacionadas às atividades desta instituição.

Palavras-chave: Processamento de Linguagem Natural. Sistemas de Diálogos. Atendimento ao Cliente.

Abstract

Dialogue systems are applications from one of the areas of Natural Language Processing studies (NLP) that enable communication between users and machines through natural language, using a set of techniques of Artificial Intelligence (AI) to process the writing and speaking of the users. Such systems become trend in customer services due to the facility they offer to the users during system interaction. Aiming this simplicity in information exchange this paper proposes the development of a dialogue system that uses natural language to students care service of the Fundação Universidade Federal de Mato Grosso do Sul (Câmpus Pantanal), for providing related information about this institution's activities.

Keywords: Natural Language Processing. Dialogue Systems. Customer Service.

Listas de ilustrações

Figura 1 – Diferentes utilidades dos <i>chatterbots</i>	15
Figura 2 – Exemplo para montagem de uma arvore sintática	21
Figura 3 – Arquitetura de um Sistema de Interpretação de Língua Natural	22
Figura 4 – Fases Principais de um Gerador de Textos	23
Figura 5 – Simulando uma conversa com o <i>chatterbot</i> Eliza	27
Figura 6 – Simulando uma conversa com o <i>chatterbot</i> A.L.I.C.E.	28
Figura 7 – Tela inicial da plataforma Microsoft Bot Framework	30
Figura 8 – Aplicação executada localmente em Node.js	31
Figura 9 – Exemplo de Chatterbot - Bing News Bot	31
Figura 10 – Plataforma Wit.ai do Facebook	33
Figura 11 – Exemplo de Chatterbot - Dinner Ideas Bot	34
Figura 12 – Atividades do desenvolvimento incremental	35
Figura 13 – API RESTful com Node.JS	40
Figura 14 – Autenticação com JWT	41
Figura 15 – Estrutura do JWT	41
Figura 16 – Arquivo de configuração - package.json	43
Figura 17 – Processo utilizado para requisição das informações	43
Figura 18 – Criação da página CPAN BOT	46
Figura 19 – Página do Facebook configurada	46
Figura 20 – Processo para criar um aplicativo	47
Figura 21 – Produto Messenger adicionado ao aplicativo	47
Figura 22 – Selecionando o botão <i>setup webhooks</i>	48
Figura 23 – Configuração do <i>webhooks</i>	48
Figura 24 – Processo de classificação de texto	49
Figura 25 – Mensagem de boas vindas com o botão começar	52
Figura 26 – Mensagem de inicialização de conversa e menu persistente	53
Figura 27 – Template estruturado no formato de carrossel	53
Figura 28 – Mensagem estruturada	54
Figura 29 – Resposta rápida	54
Figura 30 – Planos de Hospedagem de aplicativos	55
Figura 31 – Caso de Uso - Usuário	80
Figura 32 – Caso de Uso - Setor	81
Figura 33 – Caso de Uso - Telefone	81
Figura 34 – Caso de Uso - Notícia	82
Figura 35 – Caso de Uso - Curso	82
Figura 36 – Modelo de Banco de Dados - Parte I	83

Figura 37 – Modelo de Banco de Dados - Parte II	83
Figura 38 – Aplicação Web - Tela de login	84
Figura 39 – Aplicação Web - Tela de Início	84
Figura 40 – Aplicação Web - Tela de Usuários	85
Figura 41 – Aplicação Web - Tela de Departamentos	85
Figura 42 – Aplicação Web - Tela de Coordenações	86
Figura 43 – Aplicação Web - Tela de Telefones	86
Figura 44 – Aplicação Web - Tela de Notícias	87
Figura 45 – Mapa Mental Chatterbot	87
Figura 46 – Fluxograma de Interação Bot X Usuário - Parte I	88
Figura 47 – Fluxograma de Interação Bot X Usuário - Parte II	89
Figura 48 – Fluxograma de Interação Bot X Usuário - Parte III	90
Figura 49 – Fluxograma de Interação Bot X Usuário - Parte IIII	91
Figura 50 – Pergunta 1: Você utilizou o chatterbot? (22 respostas)	92
Figura 51 – Pergunta 2: Qual é a sua relação com o Câmpus do Pantanal (18 respostas)	92
Figura 52 – Pergunta 3: O atendimento por chatterbot proporcionou o acesso rápido a informação disponibilizada pelo CPAN? (18 respostas)	93
Figura 53 – Pergunta 4: O chatterbot respondeu todas as suas dúvidas? (18 respostas)	93
Figura 54 – Pergunta 7: Qual nota você daria para este serviço? Obs.: Considerando 1 como nota mínima e 5 como nota máxima de satisfação (18 respostas)	95

Lista de tabelas

Tabela 1 – Comparação entre MySQL e MongoDB	37
Tabela 2 – Classes e mensagens para treinamento do <i>bot</i>	50
Tabela 3 – Requisitos Funcionais (Aplicação Web)	61
Tabela 4 – Requisitos Funcionais (Chatterbot)	62
Tabela 5 – Requisitos Não Funcionais	63
Tabela 6 – Regras de Negócios (Aplicação Web)	64
Tabela 7 – Regras de Negócios (Chatterbot)	64
Tabela 8 – Requisito funcional RF01	65
Tabela 9 – Requisito funcional RF02	66
Tabela 10 – Requisito funcional RF03	66
Tabela 11 – Requisito funcional RF04	67
Tabela 12 – Requisito funcional RF05	67
Tabela 13 – Requisito funcional RF06	68
Tabela 14 – Requisito funcional RF07	68
Tabela 15 – Requisito funcional RF08	68
Tabela 16 – Requisito funcional RF09	69
Tabela 17 – Requisito funcional RF10	69
Tabela 18 – Requisito funcional RF11	70
Tabela 19 – Requisito funcional RF12	70
Tabela 20 – Requisito funcional RF13	71
Tabela 21 – Requisito funcional RF14	71
Tabela 22 – Requisito funcional RF15	72
Tabela 23 – Requisito funcional RF16	72
Tabela 24 – Requisito funcional RF17	73
Tabela 25 – Requisito funcional RF18	73
Tabela 26 – Requisito funcional RF19	74
Tabela 27 – Requisito funcional RF20	74
Tabela 28 – Requisito funcional RF21	75
Tabela 29 – Requisito funcional RF22	75
Tabela 30 – Requisito funcional RF23	76
Tabela 31 – Requisito funcional RF24	76
Tabela 32 – Requisito funcional RF25	76
Tabela 33 – Requisito funcional RF26	77
Tabela 34 – Requisito funcional RF27	77
Tabela 35 – Requisito funcional RF28	78
Tabela 36 – Requisito funcional RF29	78

Lista de abreviaturas e siglas

PLN	Processamento de Linguagem Natural
IA	Inteligência Artificial
API	Application Programming Interface
SO	Sistema Operacional
SDK	Software Development Kit
RI	Recuperação de Informação
EI	Extração de Informação
CEO	Chief Executive Officer
IBM	International Business Machines
MIT	Massachusetts Institute of Technology
USD	United States Dollar
JWT	Json Web Token
SSL	Secure Socket Layer
URL	Uniform Resource Locator
HTTP	Hypertext Transfer Protocol

Sumário

1	INTRODUÇÃO	13
1.1	Objetivos	13
1.2	Motivação	14
1.3	Organização do Trabalho	15
2	PROCESSAMENTO DE LINGUAGEM NATURAL	17
2.1	Níveis de Processamento	17
2.1.1	Nível Fonológico	17
2.1.2	Nível Morfológico	18
2.1.3	Nível Lexical	18
2.1.4	Nível Sintático	19
2.1.5	Nível Semântico	19
2.1.6	Nível Discursivo	20
2.1.7	Nível Pragmático	20
2.2	Arquitetura de Sistemas de PLN	20
2.2.1	Interpretação de Língua Natural	20
2.2.2	Geração de Língua Natural	22
2.3	Áreas de Aplicações	23
2.4	Desafios do PLN	25
3	SISTEMAS DE DIÁLOGOS	26
3.1	Assistentes Virtuais	26
3.2	<i>Chatterbots</i>	26
4	FRAMEWORKS PARA DESENVOLVIMENTO DE BOTS	29
4.1	Microsoft Bot Framework	29
4.2	WIT.AI	32
5	METODOLOGIA	35
5.1	Modelo Incremental	35
5.2	Ferramentas Utilizadas	36
5.2.1	JetBrains PhpStorm	36
5.2.2	Astah Community	36
5.2.3	Draw.IO	36
5.3	Mean Stack	37
5.3.1	MongoDB	37

5.3.2	Express	37
5.3.3	Angular	38
5.3.4	NodeJS	38
5.4	Rede / Segurança	39
5.4.1	Arquitetura REST	39
5.4.2	API RESTful com NodeJS	39
5.4.3	Autenticação com Json Web Token	40
6	ARQUITETURA DO SISTEMA	42
6.1	Desenvolvimento da API	42
6.2	Desenvolvimento da Aplicação Web	44
6.3	Desenvolvimento do Chatterbot	45
6.4	Hospedagem	55
7	LIMITAÇÕES E RESULTADOS	56
8	CONCLUSÕES E TRABALHOS FUTUROS	58
	REFERÊNCIAS	59
	APÊNDICE A – ENGENHARIA DE REQUISITOS	61
A.1	Identificação dos Requisitos Funcionais	61
A.1.1	Requisitos funcionais (Aplicação Web)	61
A.1.2	Requisitos funcionais (Chatterbot)	62
A.2	Identificação dos Requisitos Não Funcionais	63
A.3	Regras de Negócios	64
A.3.1	Regras de negócios (Aplicação Web)	64
A.3.2	Regras de negócios (Chatterbot)	64
A.4	Especificação dos Requisitos Funcionais	64
	APÊNDICE B – MODELOS CRIADOS	80
B.1	Diagrama de Caso de Uso	80
B.2	Modelo do Banco de Dados	82
B.3	Aplicação Web	83
B.4	Mapa Mental Chatterbot	87
B.5	Fluxograma Interação Bot X Usuário	88
	APÊNDICE C – RESULTADOS	92
C.1	Pesquisa de Satisfação	92

1 Introdução

A língua natural pode ser entendida como a linguagem utilizada para a comunicação entre os seres humanos. Com o avanço da computação, o estudo e tratamento do processamento da língua para a compreensão e geração pelos sistemas computacionais possibilitou o surgimento de diferentes abordagens para tratar dos problemas enfrentados na comunicação entre homens e máquinas em suas áreas de aplicação.

O Processamento de Linguagem Natural (PLN) consiste em sistemas computacionais que utilizam de um conjunto de técnicas para processar a escrita e a fala da linguagem humana. Esses sistemas diferem-se dos demais devido à necessidade da compreensão da língua para o processamento das informações (JURAFSKY; MARTIN, 2000).

Atualmente, há um grande número de empresas envolvidas em pesquisas na área de PLN, tais como Google¹, Facebook² e Microsoft³. Empresas que estão investindo nesse cenário, no qual a comunicação com os sistemas de inteligência artificial tornam a interação mais eficaz entre homens e máquinas como, assistentes virtuais e serviços de atendimento com robôs virtuais, permitindo que possam ser utilizados em diversos serviços e também para uso pessoal.

No âmbito comercial, há empresas que utilizam tais tecnologias como oportunidade para oferecer serviços diferenciados aos seus clientes. Assim, os clientes podem adquirir serviços por meio de aplicações inteligentes utilizadas para simular a conversa com um ser humano (*chatterbots*), possibilitando que a comunicação seja realizada de uma maneira rápida, apenas digitando ou executando comando por voz (DREHER, 2016).

1.1 Objetivos

Como objetivos gerais deste trabalho, buscou-se: compreender os sistemas de inteligência artificial e o processo utilizado para processar e gerar a língua natural nos sistemas de diálogos, por exemplo, robôs de conversação e assistentes virtuais; identificar as tecnologias do ramo de PLN que auxiliam no desenvolvimento de sistemas de diálogo (*chatterbots*); selecionar uma tecnologia e desenvolver um sistema de diálogo para atendimento à comunidade sobre as atividades desenvolvidas no Câmpus do Pantanal (UFMS) em Corumbá-MS.

Os objetivos específicos deste trabalho são:

¹ Disponível em: <<https://www.google.com.br/intl/pt-BR/about/>>.

² Disponível em: <<https://www.facebook.com/>>.

³ Disponível em: <<https://www.microsoft.com/>>.

1. Levantamento do referencial teórico sobre o PLN.
2. Identificar tecnologias de *chatterbot* que auxiliam no processo de desenvolvimento.
3. Selecionar uma tecnologia para que possa desenvolver um *chatterbot* que responda as informações de notícias, cursos, formas de ingresso, telefones e setores do CPAN.
4. Desenvolver uma aplicação web para gerenciar o conteúdo acessado pelo *chatterbot*.
5. Desenvolver um *web scraping* para buscar pelas últimas notícias cadastradas no site do CPAN, e cadastrá-la na aplicação web.
6. Treinar o *bot*, por meio de um algoritmo de classificação em aprendizado de máquina, para entendimento das mensagens na interação com o usuário.
7. Cadastrar na aplicação web as informações referente ao CPAN, para ser acessada pelo *bot* na interação com os usuários.

1.2 Motivação

Como acadêmico do Câmpus do Pantanal, foi observado a falta de um canal de comunicação que responda as constantes dúvidas dos acadêmicos e da comunidade externa relacionadas aos cursos oferecidos, ingresso na universidade, assistência estudantil, acesso à internet, entre outros recursos disponibilizados pela instituição. A obtenção das respostas para as dúvidas mencionadas é dificultada, pois não há um serviço que organize estas informações em um ambiente único.

Surgiu, desta forma, a oportunidade de prover o acesso rápido a informação por meio de um serviço diferenciado e unificado para atendimento, que contemplam as dúvidas recorrentes dos recursos disponibilizados pela instituição. Este serviço de atendimento será disponibilizado por meio de um *chatterbot*, pelo qual as pessoas podem interagir usando um dos principais meios para comunicação atualmente, o Facebook Messenger, que conta com mais de 1 bilhão de pessoas conectadas (FACEBOOK, 2017).

Grandes empresas têm investido em tecnologias que envolvem os sistemas de diálogos, com a oportunidade de interagir com aplicativos que contemplam diversas utilidades. Por exemplo, notícias, jogos de entretenimento e compras por serviços de mensagens, como mostra a Figura 1.

Figura 1 – Diferentes utilidades dos *chatterbots*



Fonte: Disponível em: <<https://goo.gl/QHf2iy>>.

Por fim, a possibilidade de integração dos *chatterbots* com os diferentes canais de comunicação permitem que os serviços prestados alcancem um número maior de pessoas, conectadas pelo Facebook Messenger, Skype, Telegram, entre outros.

1.3 Organização do Trabalho

Este trabalho está organizado em 8 Capítulos, sendo este iniciado por uma introdução sobre o Processamento de Linguagem Natural, objetivos e motivação deste trabalho. No Capítulo 2 é realizado o levantamento bibliográfico sobre o PLN, contextualização das técnicas para a compreensão da linguagem humana, as limitações que estes sistemas enfrentam no processamento da fala e escrita, e as áreas de aplicação.

No Capítulo 3 são abordados os sistemas de diálogos inteligentes, utilizados na comunicação com os humanos, tais como as/os assistentes virtuais e *chatterbots*, bem como as diferenças entre eles. No Capítulo 4 são apresentadas as tecnologias que envolvem o PLN disponibilizadas pelas empresas já citadas, a fim de facilitar o desenvolvimento destes sistemas computacionais, bem como exemplos de softwares desenvolvidos por estas tecnologias.

No Capítulo 5 é apresentada a metodologia do trabalho, desde o processo de desenvolvimento às ferramentas utilizadas. O Capítulo 6 mostra o desenvolvimento da API, aplicação web e *chatterbot* proposto para os objetivos específicos. No Capítulo 7

são apresentadas as limitações e os resultados obtidos com o projeto desenvolvido. Por fim, o Capítulo 8 apresenta as conclusões e os trabalhos futuros que podem ser realizados posteriormente.

2 Processamento de Linguagem Natural

O Processamento de Linguagem Natural (PLN) é uma das áreas de inteligência artificial (IA), consiste em um sistema computacional no qual o modo de interação depende do uso de alguma língua natural, permitindo que a comunicação seja realizada por meio de texto ou voz.

De acordo com Liddy (2001), PLN é uma variedade de técnicas computacionais para analisar e representar textos que ocorrem naturalmente em um ou mais níveis de análise linguística, tendo como finalidade alcançar a linguagem humana para uma série de tarefas ou aplicações. Ainda para Liddy (2001), no PLN podem ser aplicadas diversas técnicas para realizar um determinado tipo de análise da linguagem em diferentes idiomas de acordo com suas tipologias linguísticas.

Para que um sistema consiga processar uma língua natural é preciso que a máquina compreenda a estrutura do idioma no qual irá interagir. Na seção seguinte são descritos os níveis de processamento utilizados por sistemas para a compreensão da língua.

2.1 Níveis de Processamento

Para Liddy (2001), as técnicas de PLN para a compreensão da língua estão divididas em níveis de processamento, classificados em: fonológico, morfológico, lexical, sintático, semântico, discursivo e pragmático. Abaixo são descritos de maneira sucinta os níveis de processamento necessários para a compreensão de uma língua natural.

2.1.1 Nível Fonológico

De acordo com Liddy (2001), o nível fonológico está relacionado com a interpretação dos sons da fala por meio da pronúncia. Em PLN, as ondas sonoras são processadas para a interpretação da linguagem específica utilizada. Este tipo de processamento é utilizado em sistemas de reconhecimento de voz.

Segundo Nunes et al. (2007), o sistema sonoro pode ser representado por meio dos fonemas de uma língua natural. Em PLN, a representação e operacionalização dos fonemas são particularmente importantes para o processamento dos sons produzidos pelo falante de uma língua para determinar os paradigmas sonoros, tais como as alterações de timbre e intensidade das palavras.

2.1.2 Nível Morfológico

O nível morfológico está relacionado com à análise e compreensão dos significados das partes das palavras. Para Nunes et al. (2007), as palavras de uma língua também podem ser segmentadas em unidades mínimas em termos de seu significado (gramatical ou lexical) denominadas morfemas.

Na língua portuguesa, a morfologia distingue-se em grammatical e lexical. De acordo com Luft (2008), na morfologia lexical são tratados os problemas como origem, formação e estrutura das palavras. A morfologia grammatical ocupa-se com a classificação das palavras, categorias gramaticais (gênero, número, grau, pessoa, modo, tempo, aspecto), paradigmas flexionais, etc.

Nos exemplos citados por Nunes et al. (2007, p. 19) temos:

1. *pedra* => -a: morfema grammatical que indica os nomes terminados em “-a”.
2. *macaca* => -a: morfema grammatical que indica gênero feminino.
3. *procuramos* => -a: morfema grammatical que indica a primeira conjugação verbal;
-mos: morfema grammatical que indica primeira pessoa do plural do presente do indicativo.
4. *operação* => -ção: morfema lexical que indica evento.
5. *incerto* => in-: morfema lexical que indica negação.

A segmentação morfológica e a relação que ela implica no significado das palavras é importante para a compreensão do processamento linguístico. Nesse sentido, Liddy (2001) destaca que no PLN, da mesma forma que para os seres humanos, uma palavra desconhecida é dividida em seus morfemas constituintes a fim de compreender o seu significado.

2.1.3 Nível Lexical

O nível lexical, de acordo com Liddy (2001), trata do processo de interpretar o significado das palavras individualmente. No nível lexical são analisados os diferentes termos de uma palavra.

Para Vieira e Lima (2001, p. 12), “O léxico ou dicionário é uma estrutura fundamental para a maioria dos sistemas e aplicações. É a estrutura de dados contendo os itens lexicais e as informações correspondentes a estes itens”.

Nos exemplos de Vieira e Lima (2001) encontram-se as palavras isoladas que constituem as entradas de um léxico (como *lua*, *mel*, *casa*, *modo*) ou composições que

quando reunidas apresentam um significado específico (por exemplo: *lua de mel*, *casa de cultura ou a grosso modo*).

2.1.4 Nível Sintático

O nível sintático tem como objetivo analisar a estrutura sintática e as funções que as palavras desempenham em uma frase ou oração. Segundo Liddy (2001), este nível concentra-se na análise das palavras, de modo a descobrir sua estrutura gramatical.

Nesse sentido, Vieira e Lima (2001, p. 5) afirmam que “É através da análise sintática que se pode verificar se a concordância estabelecida pelas regras da língua está sendo obedecida”.

Para Nunes et al. (2007), as combinações de palavras para formar uma frase ou oração dotada de um sentido completo devem seguir regras estruturais bem definidas, devido essas regras determinarem, por exemplo, o emprego dos pronomes, a aplicação de crase, a realização da concordância, dentre outros.

Nos exemplos abaixo, citados por Vieira e Lima (2001), as diferentes possibilidades de interpretações em uma oração também estão relacionadas com sua estrutura.

- O homem viu o menino com o telescópio.
- Ele entrou na sala de muleta.

As diferentes interpretações na análise das orações acima são (O homem viu com o telescópio ou o menino estava com o telescópio; ele entrou de muleta ou a sala de muleta) como pode-se perceber há diferentes interpretações para as orações, contudo a ambiguidade constatada está no nível sintático.

2.1.5 Nível Semântico

O nível semântico, de acordo com Nunes et al. (2007), está relacionado ao significado das palavras em busca de alcançarem certo sentido no escopo da sentença, não apenas nas palavras como uma unidade completa, mas nas suas unidades constitutivas.

Segundo Liddy (2001), todos os níveis de processamento da língua citados contribuem para o significado, no entanto o processamento semântico determina os possíveis significados de uma sentença, concentrando-se nas interações entre os significados.

Pode-se perceber que o estudo da semântica interpretará o significado das palavras utilizadas na comunicação tanto em palavras individuais quanto em expressões ou frases da linguagem natural. Portanto, no processamento semântico serão considerados os problemas enfrentados com os múltiplos sentidos (ambiguidade) de algumas palavras, por exemplo, a palavra “banco” que pode referir-se tanto a instituição financeira, quanto ao assento.

2.1.6 Nível Discursivo

O nível discursivo, segundo Liddy (2001), concentra-se na interpretação das propriedades do texto como um todo, diferentemente da análise sintática e semântica que trabalham com orações interpretadas isoladamente.

É na análise do discurso que se estudam as relações que envolvem a língua e as situações de contexto, com foco nas questões dos fenômenos extralingüísticos, ou seja, a compreensão de certas palavras e sentenças a partir de diversos fatores e situações comunicativas em que se desenvolve o discurso, além das formas e das estruturas da língua (NUNES et al., 2007).

A análise do discurso também indica o fato das construções ideológicas presentes nos textos permitir ao falante preferir um enunciado ao invés de outro. Em PLN, esses conhecimentos são aplicados quando a ênfase do processamento são as referências anafóricas, ou seja, elementos já expressos anteriormente no discurso (ex.: *ele, isso, aquele.*) e os dêiticos, elementos de referência temporal e espacial situados no contexto em que o enunciado é produzido (ex.: *hoje, daqui a pouco, lá, ali, aqui, lá, cá.*) afirma Nunes et al. (2007).

2.1.7 Nível Pragmático

O nível pragmático, de acordo com Nunes et al. (2007), consiste em examinar uma construção linguística procurando a compreensão das palavras e sentenças presentes no ato de fala na atividade comunicativa.

Para Vieira e Lima (2001), a pragmática tem como objeto de estudo o significado de uma sentença que integra a diferença entre o significado literal da linguagem e o significado da linguagem em uso, ou seja, o contexto do falante na comunicação.

2.2 Arquitetura de Sistemas de PLN

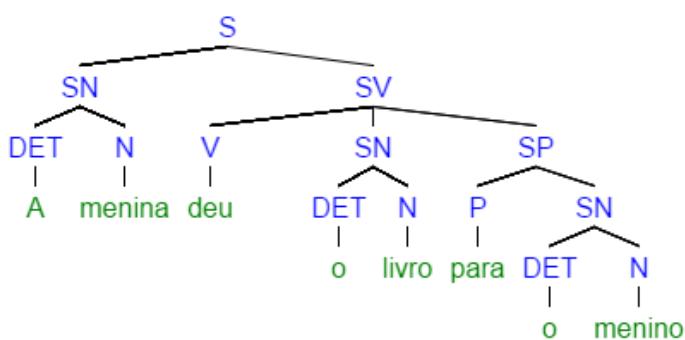
Os níveis de processamento da linguagem nos estudos de PLN seguem uma arquitetura geral para a interpretação e geração de uma língua natural, que pode ser utilizada para diferentes finalidades como, recuperação de informação e tradução automática. A seguir, são apresentadas as atividades executadas para interpretar e gerar a língua natural.

2.2.1 Interpretação de Língua Natural

Na Figura 3 são apresentados os módulos e os recursos necessários para o processamento da língua, conforme Nunes et al. (2007). Os módulos são:

- **Analisador Léxico:** É a primeira etapa do processamento linguístico que envolve a identificação e separação dos componentes do texto, também conhecida como tokenização. Ex.: [Você] [sabe] [que] [horas] [são] [?].
- **Analisador Sintático:** Consiste na construção de um esquema arbóreo com a estrutura sintática da sentença conforme a gramática da língua, a partir da separação dos componentes do texto no Analisador Léxico. A Figura 2 mostra a estrutura sintática para a oração “A menina deu o livro para o menino”.

Figura 2 – Exemplo para montagem de uma árvore sintática

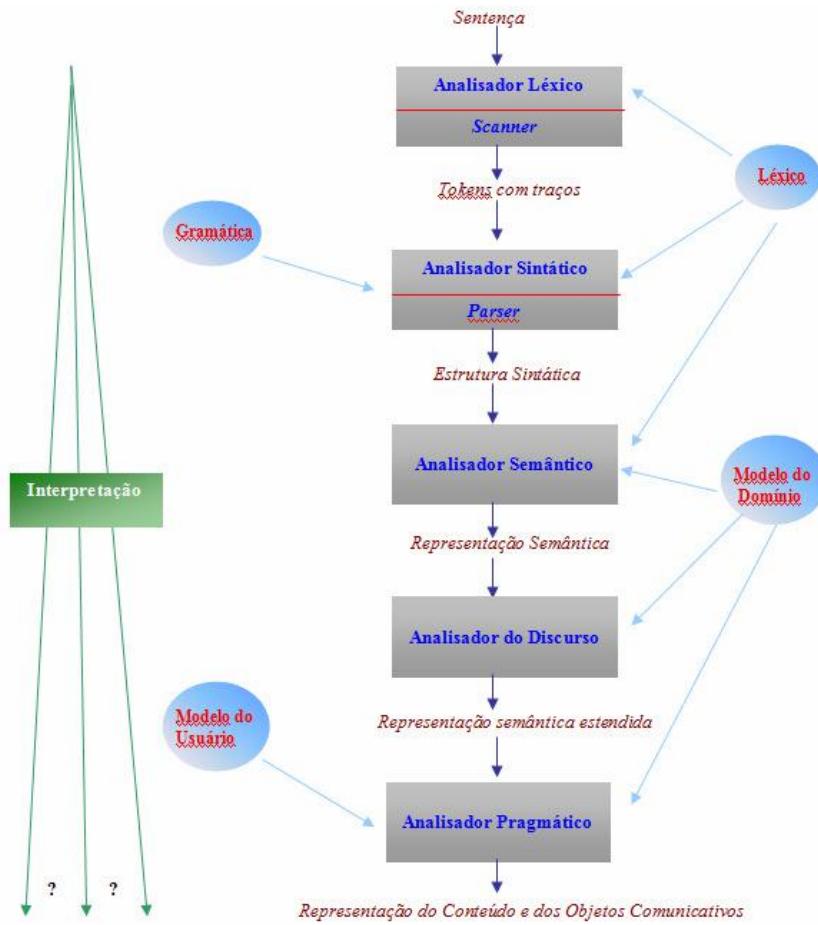


Fonte: Estrutura montada com a ferramenta disponibilizada em <<http://mshang.ca/syntree/>>

A árvore sintática montada está composta por: S - Sentença, SN - Síntagma Nominal, SV - Síntagma Verbal, SP - Síntagma Preposicional, D - Determinante, N - Nome, P - Preposição, V - Verbo.

- **Analisador Semântico:** O analisador semântico é responsável por analisar e compreender as palavras individuais de uma sentença e a relação entre elas, a partir da estrutura sintática construída no analisador sintático.
- **Analisador de Discurso:** O analisador de discurso consiste na análise do significado de uma frase que, dependendo da sentença que a antecede, pode influenciar o significado das sentenças que o segue.
- **Analisador Pragmático:** No analisador pragmático será analisado o significado real que a sentença representa. Ex.: “Você sabe que horas são?”, deve ser interpretado como a solicitação para alguém que deseja que o horário seja informado. A seguir, a Figura 3 apresenta a arquitetura de um sistema de interpretação de língua natural.

Figura 3 – Arquitetura de um Sistema de Interpretação de Língua Natural



Fonte: Nunes et al. (2007).

2.2.2 Geração de Língua Natural

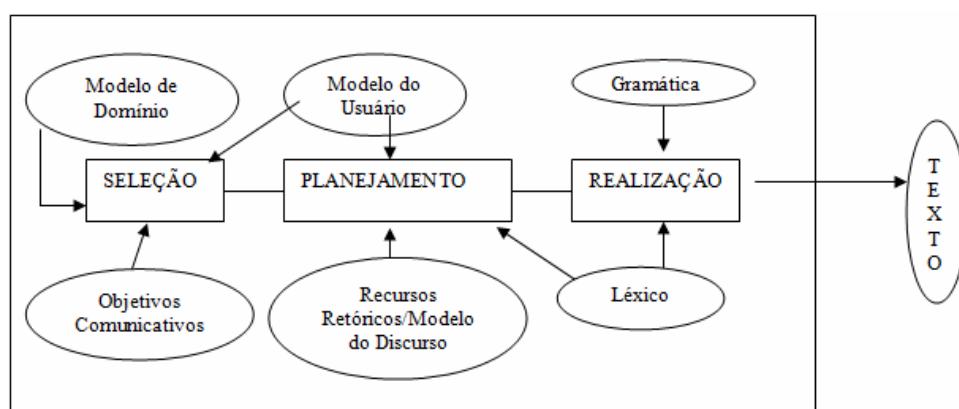
A seguir, é apresentada a arquitetura geral de um sistema, que tem como função produzir textos em língua natural a partir de um conjunto de elementos e objetivos da comunicação. Conforme Nunes et al. (2007), o processo consiste em 3 passos fundamentais durante a geração:

- 1. Seleção do conteúdo:** Este processo consiste em selecionar os itens de conhecimentos que deverão fazer parte do texto na geração de língua natural. Por exemplo, extrair de uma base de dados os itens necessários que irão compor a resposta a partir de uma pergunta de entrada.
- 2. Planejamento do texto:** Este processo consiste em planejar a comunicação, ou seja, é analisado quando será exibido o conteúdo selecionado no passo anterior, devido os conhecimentos extraídos poderem estar em bases de registros diferentes, devendo ser organizado para melhor apresentação textual do conteúdo.

3. Realização do texto: Este processo é responsável pela realização gramatical do plano de texto produzido no passo anterior. Nesta etapa, serão determinados os itens linguísticos, como a escolha adequada de vocabulário, estilo do texto, escolhas léxicas, morfológicas e sintáticas adequadas para expressar o conteúdo e a estrutura textual, entre outros.

A Figura 4 apresenta as fases principais de um gerador de textos já listadas, conforme Nunes et al. (2007).

Figura 4 – Fases Principais de um Gerador de Textos



Fonte: Nunes et al. (2007).

2.3 Áreas de Aplicações

As diferentes possibilidades das áreas de aplicações em PLN requerem uma ou mais tarefas básicas para o processamento da língua, classificadas conforme Nunes (2008):

- No **Pré-processamento**, os textos são subdivididos em unidades fonéticas, lexicais, gramaticais, semânticas ou discursivas, de acordo com o objetivo da tarefa em questão.
- **Classificar (Etiquetar)** as unidades do texto, segundo classes pertinentes às tarefas morfossintáticas, sintáticas, semânticas e discursivas.
- **Mapear representações** da língua natural para uma representação sintática, semântica ou discursiva; e dessas para interpretação e geração de língua natural.

Abaixo são descritas algumas das aplicações que utilizam das técnicas de PLN, entre elas os sistemas de Recuperação de Informação, Extração de Informação, Correção Ortográfica e Gramatical e o Reconhecimento de Voz.

Recuperação de Informação (RI)

Segundo Norving e Russell (2013), a recuperação de informação é a tarefa de encontrar documentos que são relevantes para a necessidade do usuário em obter as informações, podendo ser encontrados desde contextos pequenos até em contextos maiores como a *World Wide Web* (www).

Nesses sistemas todos os documentos relevantes, de acordo com a pesquisa realizada são recuperados, possibilitando realizar a consulta em linguagem natural. Dentre os sistemas de consulta mais conhecidos no contexto da web está o buscador da Google¹.

Extração de Informação (EI)

De acordo com Nunes (2008), os sistemas de extração de informação buscam a resposta em um ou mais documentos a partir de uma pergunta de entrada. Nesses sistemas, para ser preciso à pergunta ao documento inspecionado é necessário um processamento complexo de PLN, como etiquetação, consulta a dicionários e ontologias, aplicações heurísticas, etc.

Dos sistemas de extração de informação o IBM Watson² é o mais conhecido, utilizando um conjunto de técnicas juntamente com a recuperação de informação para buscar a resposta adequada a partir de uma interação com o usuário em linguagem natural.

Correção Ortográfica e Gramatical

Segundo Vieira e Lima (2001), os sistemas de correção ortográfica e gramatical verificam se o conjunto de palavras no texto pertence ao vocabulário da língua e se as construções gramaticais estão de acordo com as regras de concordância, por exemplo, ortografia oficial, uso da crase, colocação pronominal, concordância verbal, pontuação, uso de prefixos, e outros.

Dentre as aplicações que possuem as funcionalidades citadas acima para correção ortográfica e gramatical, está o conjunto de aplicativos para escritório Microsoft Office e entre outros pacotes com a mesma finalidade.

Reconhecimento de Voz

Segundo Vieira e Lima (2001), o reconhecimento de voz pode ser definido como sistemas que realizam a transcrição da fala em texto ou interfaces de comandos por voz. Entre os sistemas de reconhecimento de voz mais conhecidos encontram-se as assistentes virtuais presentes nos smartphones e computadores, que têm como propósito melhorar a interação entre o usuário e o aparelho por meio do reconhecimento das ações a serem executadas.

¹ Disponível em: <<https://www.google.com.br/>>.

² Disponível em: <<http://www.ibm.com/watson/>>.

2.4 Desafios do PLN

Um dos obstáculos enfrentados pelo PLN é fazer com que os sistemas compreendam a mensagem que um ser humano quer passar, pelo fato da comunicação conter muitas variações dialetais e expressões ambíguas, que interferem no conteúdo de uma mensagem.

Segundo Norving e Russell (2013), as linguagens naturais são ambíguas, podendo ter diferentes significados para uma determinada sentença. Na frase “Ela estava em minha companhia” a palavra “companhia” pode estar tanto relacionada a “presença da pessoa em si” ou “alguma empresa”.

Ainda para Norving e Russell (2013), as línguas naturais são difíceis de lidar porque estão em constantes mutações. Nas línguas naturais é necessário o conhecimento da estrutura que envolve a construção da fala e escrita em que se concentra, tornando os resultados mais difíceis no que se trata do processamento da língua.

Neste sentido, Siqueira (2005) também destaca que uma das limitações é que a linguagem humana não é exata como a linguagem de máquina. Nos exemplos “Qual é o seu nome?” ou “Como você se chama?” apesar de terem significados idênticos, as frases são sintaticamente distintas.

Outro desafio apresentado é para uma das áreas de aplicações do PLN, por exemplo, a tradução automática que segundo Norving e Russell (2013) há um custo muito alto no processamento, pelo fato de requerer o conhecimento profundo do texto e cada língua categorizar as palavras de formas diferentes.

Como visto, o processamento de linguagem natural ainda é limitado apesar de ter evoluído nos últimos anos. Em relação as expressões ambíguas que interferem no conteúdo de uma mensagem, estes podem ser resolvidos com a lógica *Fuzzy*, que segundo Cavalcante, Bezerra e Nunes (2010, p. 11), “coloca-se como o principal instrumento para uma representação mais adequada do conhecimento, isso se devendo à sua capacidade de lidar com incertezas, raciocínio aproximado, termos vagos e ambíguos”. Muitos dos desafios enfrentados ainda serão compreendidos e resolvidos devido o uso de PLN nas aplicações apresentadas serem recorrentes nos dias atuais.

No próximo capítulo são abordados os sistemas de diálogos, tais como assistentes virtuais e *chatterbots*, bem como o propósito e a diferença entre eles.

3 Sistemas de Diálogos

Neste capítulo são apresentadas as diferenças entre as assistentes virtuais e os *chatterbots*, ambos com o propósito de melhorar a interação entre humanos e máquinas por meio de serviços de ajuda, recomendações de compras e suporte nas atividades diárias realizadas.

3.1 Assistentes Virtuais

As assistentes virtuais são algoritmos¹ de inteligência artificial que tem como propósito atender às necessidades dos seres humanos, realizando suas principais atividades por meio da interação por voz com os dispositivos, sejam eles smartphones, tablets, computadores, entre outros.

Além de executar os comandos dos usuários, as assistentes virtuais são capazes de aprender conforme as atividades são atendidas, e também por meio das informações dos seus respectivos perfis, e dos algoritmos de aprendizagem de máquina.

Dentre as assistentes virtuais mais conhecidas existentes no mercado, destacam-se a Google Now, Cortana e a Siri. Todas as assistentes contendo recursos avançados de interação com o usuário, podendo responder diversas perguntas realizadas por voz no dispositivo com suporte em diferentes idiomas.

Entre os serviços e recursos disponibilizados pelas assistentes, podemos encontrar funções que organizam toda a rotina do usuário, desde as funções mais básicas como controle da agenda, alarmes, informações de notícias, previsões e outras funcionalidades.

As utilizações desses recursos disponibilizados pelas assistentes virtuais melhoram as atividades realizadas no cotidiano, pelo fato dos usuários apenas pedirem para as assistentes executarem as ações desejadas. De acordo com Reisenwitz (2016), as possibilidades de interação com as assistentes virtuais tornaram uma das tendências tecnológicas que em breve serão utilizadas pela maioria dos usuários que desejam otimizar suas atividades.

3.2 Chatterbots

Os *chatterbots* ou *chatbots* são algoritmos de computador que simulam a conversa com os humanos por meio de uma língua natural. Diferentemente das assistentes virtuais, os *bots* ‘robôs’ têm um propósito mais específico, ou seja, as informações a serem respondidas

¹ Um algoritmo é uma sequência de passos computacionais que transformam um conjunto de valores como entrada e produz um conjunto de valores como saída (CORMEN, 2002).

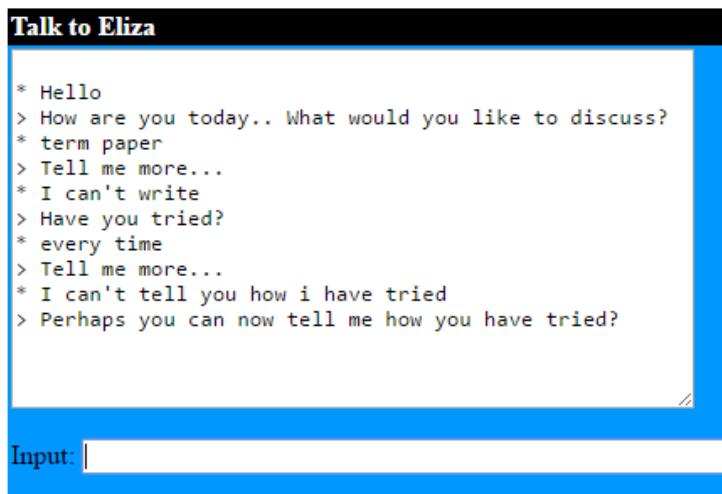
precisam estar em sua base de conhecimento para poder haver a interação entre máquinas e humanos.

Para Laven (2017), os *chatterbots* são programas de inteligência artificial que tentam simular a conversação com o objetivo de confundir um ser humano a fim de pensar que está falando com outra pessoa.

De acordo com Manfio (2014), os robôs, se comparado ao interlocutor humano, têm o diferencial de trabalhar com uma base de conhecimento bem mais precisa em relação a um assunto específico, sendo assim, não correndo o risco de esquecer ou confundir alguma informação.

Dentre os primeiros robôs de conversação popularmente conhecidos destacam-se a Eliza criada no MIT em 1966 por Joseph Weizenbaum. O robô tem como objetivo simular o diálogo entre um paciente e uma psicóloga, formulando perguntas a partir do diálogo com os pacientes. A Figura 5 mostra uma interação realizada com o robô virtual, tendo como assunto “problemas ao escrever o TCC”.

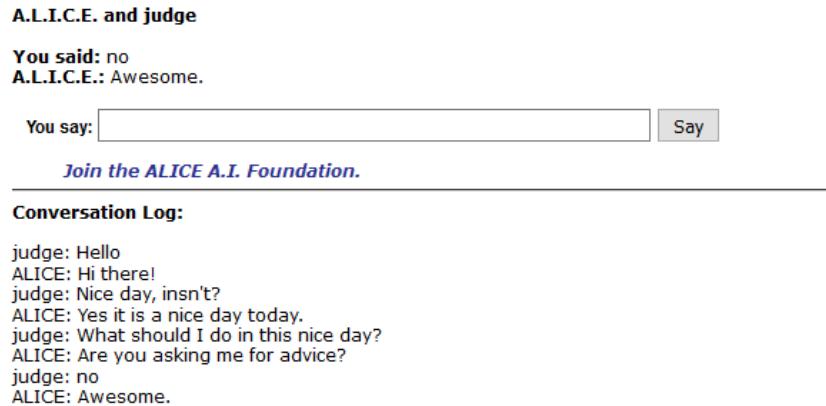
Figura 5 – Simulando uma conversa com o *chatterbot* Eliza



De acordo com Mittmann (2006), as técnicas que envolvem o *chatterbot* Eliza são: analisar a mensagem encaminhada em busca de palavras-chave, que estão associadas a um número que indica a prioridade de cada palavra. Após a associação das palavras-chave é realizado o processo de transformação e normalização, por exemplo, “eu” é transformada em “você”, “meu” para “seu”, etc. No processo de normalização, as palavras que estiverem no plural são transformadas para o singular, a fim de usar a estrutura da sentença de entrada para compor parte da resposta do *chatterbot*.

Outro robô de conversação é a A.L.I.C.E. (*Artificial Linguistic Internet Computer Entity*) criada por Richard S. Wallace em 1995, podendo simular uma conversa livre com um ser humano. A Figura 6 mostra uma conversa realizada com o robô virtual.

Figura 6 – Simulando uma conversa com o *chatterbot* A.L.I.C.E.



O *chatterbot* A.L.I.C.E. utiliza em sua base de conhecimento para criar os diálogos de interação a linguagem AIML (*Artificial Intelligence Markup Language*), uma linguagem baseada em categorias, sendo que cada uma destas categorias representa um padrão de entrada na linguagem natural, conforme A.L.I.C.E. (2017).

Devido à facilidade de obter uma informação a partir de uma interação com um robô de conversação, os *chatterbots* tornam as informações mais amigáveis e acessíveis, proporcionando maior conforto na interação com as pessoas, afirma Siqueira (2005).

No próximo capítulo são apresentados *frameworks* utilizados no desenvolvimento de *chatterbots*, disponibilizadas pelo Facebook e Microsoft para auxiliar no processo de construção de robôs virtuais.

4 Frameworks Para Desenvolvimento de Bots

Com o avanço das tecnologias, os serviços de atendimento diferenciados tornam-se essenciais para garantir a qualidade na comunicação entre empresas e clientes. Os serviços disponibilizados atualmente, tais como aplicativos, sites, redes sociais, entre outras ferramentas, permitem um maior relacionamento com os clientes nos serviços prestados, conforme Reisenwitz (2016).

Ainda para Reisenwitz (2016), empregar a tecnologia em atendimento ao cliente para tornar a experiência mais eficiente e rápida se tornará um novo padrão. Pois, a tecnologia permite economizar tempo e otimizar tarefas recorrentes.

Para o desenvolvimento do sistema de diálogo foram analisados três serviços de desenvolvimento, a Microsoft Bot Framework, WIT.AI e Messenger Platform. Tais serviços possibilitam a integração da aplicação com os diferentes canais de mensagens disponíveis atualmente, por exemplo, Facebook Messenger, Skype, Slack, Web chat, entre outros.

A utilização dos *frameworks* permitem o desenvolvimento rápido de aplicações, que fazem uso do processamento da linguagem natural, pelo fato de não ter necessidades em preocupar-se com a estrutura da língua que envolve o PLN, sendo de responsabilidade da plataforma encapsula-las, permitindo que os desenvolvedores preocupem-se com a regra de negócio da aplicação na implementação dos *bots* (WIT.AI, 2017).

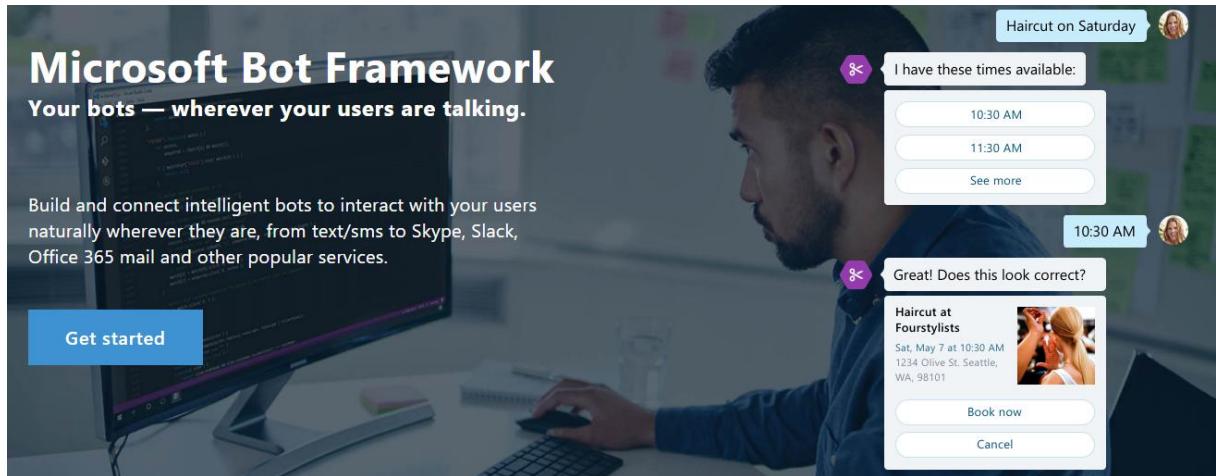
4.1 Microsoft Bot Framework

A plataforma Bot Framework¹ desenvolvida pela empresa Microsoft, como mostra a Figura 7, é uma das opções para criação de robôs virtuais para ser utilizada em qualquer tipo de finalidade. A plataforma foi inaugurada durante um evento para desenvolvedores realizado pela empresa, a conferência build² 2016.

¹ Disponível em: <<https://dev.botframework.com/>>.

² Disponível em: <<https://build.microsoft.com/>>.

Figura 7 – Tela inicial da plataforma Microsoft Bot Framework



Fonte: Microsoft (2017)

O *framework* disponibilizado pode ser utilizado gratuitamente e permite a implementação de *chatterbots* que podem ser integrados a diferentes canais de comunicação, tais como E-mail, Skype, Facebook Messenger, SMS, Telegram e outros serviços.

O Bot Framework fornece ferramentas para resolver facilmente os problemas relacionados à linguística e muito mais para desenvolvedores, por exemplo, tradução automática para mais de 30 idiomas, gerenciamento de estado de conversação e usuários, ferramentas de depuração, controle de web chat embutido (MICROSOFT, 2017).

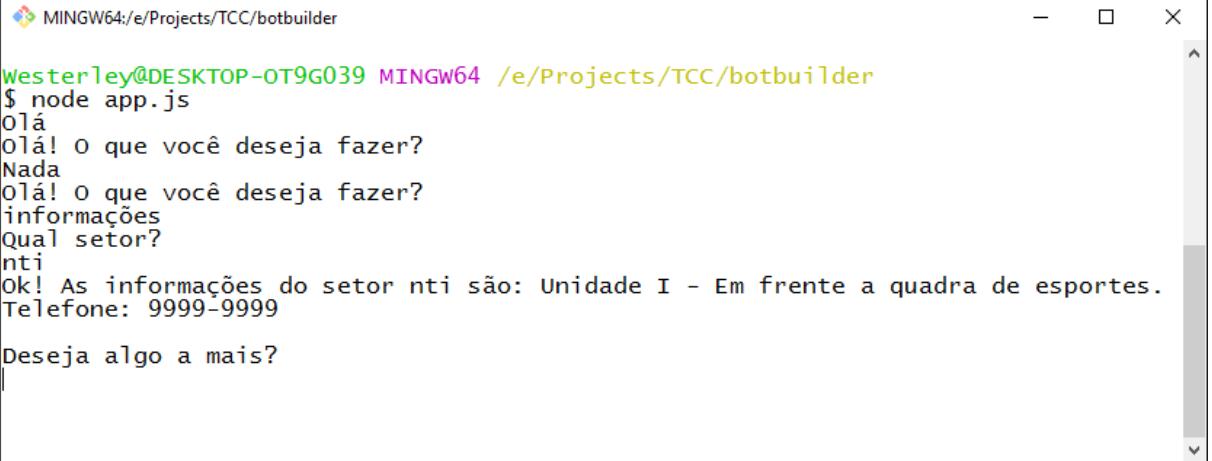
Os *chatterbots* podem ser desenvolvidos sobre duas linguagens de programação. Dentre as linguagens disponíveis estão o C# e Node.JS. O C#³ (lê-se CSharp) é uma linguagem de programação orientada a objetos criada para o desenvolvimento de aplicações. O Node.JS⁴ é uma plataforma para execução da linguagem Java Script no ambiente do servidor.

A Figura 8 mostra um exemplo de implementação utilizando o Node.js. A aplicação consiste em um *bot* que responde com informações sobre os setores do Câmpus do Pantanal de acordo com a palavra chave fornecida pelo usuário.

³ Para mais informações sobre a linguagem e suas características acesse: <<https://goo.gl/xx5xAk>>.

⁴ Para mais informações sobre a linguagem e suas características acesse: <<https://nodejs.org/en/about/>>.

Figura 8 – Aplicação executada localmente em Node.js



```

MINGW64:/e/Projects/TCC/botbuilder
westerley@DESKTOP-OT9G039 MINGW64 /e/Projects/TCC/botbuilder
$ node app.js
Olá
Olá! O que você deseja fazer?
Nada
Olá! O que você deseja fazer?
informações
Qual setor?
nti
Ok! As informações do setor nti são: Unidade I - Em frente a quadra de esportes.
Telefone: 9999-9999

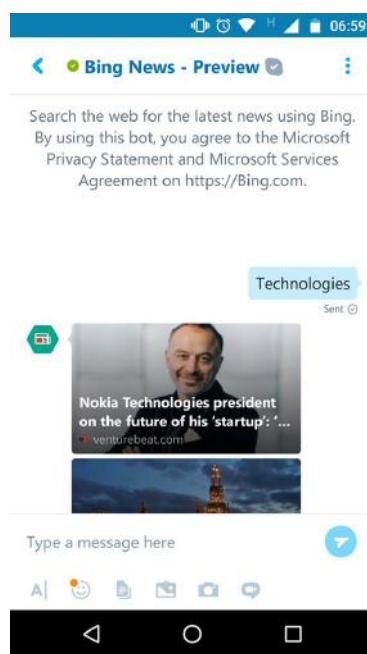
Deseja algo a mais?

```

Após implementar a lógica do *chatterbot*, o próximo passo será publicar a aplicação. A publicação consiste em hospedá-la em um servidor web e realizar as configurações para que haja a comunicação entre o servidor e os serviços de mensagens disponibilizados.

A Figura 9 apresenta uma aplicação de software, Bing News Bot, desenvolvido com esta plataforma para exibição de notícias. Com o *bot* é possível encontrar as últimas notícias a partir de tópicos digitados pelos usuários no chat integrado ao Skype. Após digitados os tópicos, o *bot* exibe três notícias com o link para redirecionamento ao site contendo a matéria completa.

Figura 9 – Exemplo de Chatterbot - Bing News Bot



No primeiro acesso ao chat, o *bot* exibe as instruções de como proceder para receber as notícias por meio do Bing News, e após digitar a palavra chave *technologies* são exibidos os links para a matéria completa do assunto pesquisado.

De acordo com Satya Nadella (Microsoft CEO), durante a apresentação na conferência Build 2016, o propósito do Bot Framework é permitir a criação de aplicações que fazem uso destas tecnologias para interagir com as empresas e serviços, podendo aproveitar o poder da linguagem humana e aplicá-las a todas as interfaces e interações computacionais, facilitando a navegação e busca por informação.

4.2 WIT.AI

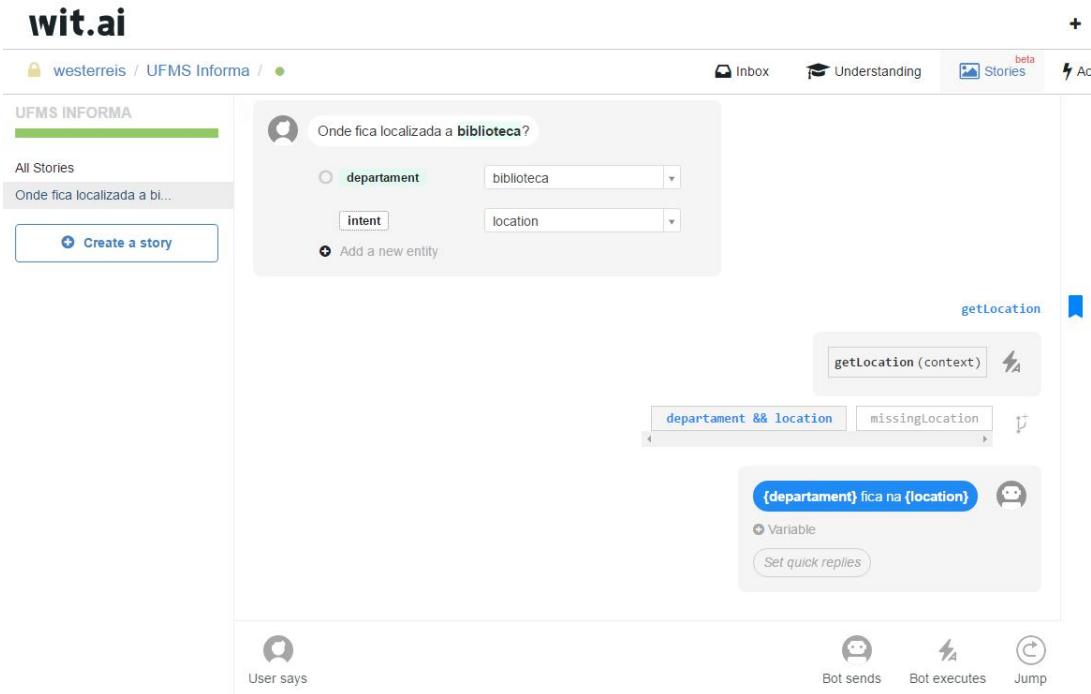
A plataforma do Facebook Wit.ai⁵ permite criar aplicações em linguagem natural por textos e comando de voz por meio do Wit console, possibilitando que desenvolvedores possam construir os *bots* a partir do console de configuração e treinamento da aplicação.

A plataforma pode ser utilizada gratuitamente e com suporte a alguns idiomas, tais como Português, Espanhol, Italiano, Russo, entre outros. A construção do *bot* é realizada por meio de mensagens que terão que ser montadas. Sendo cada mensagem uma sequência de possíveis conversas que os usuários provavelmente perguntarão ao *bot*.

Na aplicação intitulada como “UFMS/CPAN informa” como mostra a Figura 10, a primeira pergunta feita do usuário ao *bot* foi “Onde fica localizada a biblioteca?” e a resposta “{*departament*} fica na {*location*}”. Todas as mensagens devem conter uma sequência lógica para que o *bot* possa responder as perguntas adequadamente.

⁵ Disponível em: <<https://wit.ai/>>.

Figura 10 – Plataforma Wit.ai do Facebook



Fonte: Wit.ai (2017)

Após elaborar os cenários de interação, é preciso integrar a plataforma a uma das linguagens de programação disponíveis para a implementação das ações que irão ocorrer durante a interação com o robô virtual. Dentre as linguagens disponíveis estão o Node.JS, Python e Ruby. O Python⁶ é uma linguagem de programação orientada a objetos, interativa, interpretada e multiplataforma, que atende a diferentes propósitos. O Ruby⁷ é uma linguagem dinâmica, orientada a objetos, *open source* e possui uma sintaxe de fácil escrita e leitura com foco na produtividade.

Durante a elaboração dos cenários de interação com o robô virtual, os mesmos devem seguir os seguintes passos: criar as mensagens, realizar o treinamento da aplicação para a compreensão da língua natural, implementar as ações com uma das linguagens de programação disponíveis e por fim publicar a aplicação.

A publicação consiste em hospedar a aplicação em um servidor web e integrar o serviço hospedado à uma página criada no Facebook. Após a integração entre os serviços, a aplicação estará disponível para a comunicação entre os usuários do Messenger.

Os *bots* para o Facebook Messenger também podem ser desenvolvidos por meio da

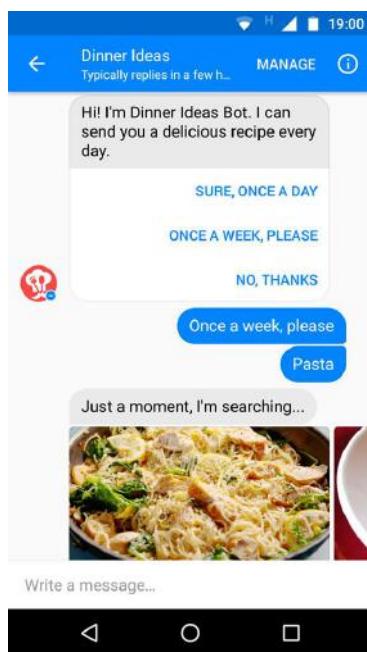
⁶ Para mais informações sobre a linguagem e suas características acesse: <<https://www.python.org/about/>>.

⁷ Para mais informações sobre a linguagem e suas características acesse: <<https://www.ruby-lang.org/en/>>.

API oficial, **Messenger Platform**, porém as técnicas que envolvem o processamento de linguagem natural devem ser implementados pelo próprio desenvolvedor, devido a API oferecer apenas os recursos essenciais para a interação com o usuário.

A Figura 11 mostra um *bot* desenvolvido para o Facebook Messenger, que fornece receitas a partir de nomes de ingredientes ou nomes de pratos. Com as informações recebidas o *bot* dinner ideas, irá enviar receitas com os ingredientes e modo de preparo em um link para acesso ao conteúdo pelo usuário. No primeiro acesso ao chat, o *bot* fornece opção para o usuário escolher se deseja receber automaticamente as receitas todos os dias, uma vez por semana ou não receber.

Figura 11 – Exemplo de Chatterbot - Dinner Ideas Bot



No próximo capítulo é apresentado o processo de desenvolvimento e as ferramentas utilizadas para desenvolver o projeto proposto.

5 Metodologia

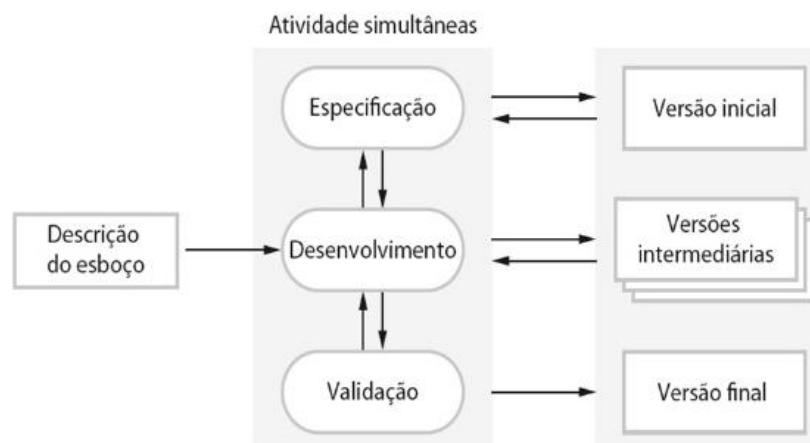
Este capítulo apresenta o processo de desenvolvimento utilizado para construir o software proposto, bem como as ferramentas utilizadas e quesitos de segurança implementados.

5.1 Modelo Incremental

O modelo incremental tem como propósito desenvolver o sistema por meio de funcionalidades com uma estratégia única, no qual as partes irão compor o sistema a cada incremento do ciclo iterativo. Segundo Sommerville (2011), o desenvolvimento incremental é baseado na ideia de implementar um sistema, por meio da criação de várias versões de acordo com os comentários dos usuários, até que o sistema final seja desenvolvido.

A Figura 12 mostra as seguintes atividades compostas pelo modelo incremental, tais como: atividades de especificação, desenvolvimento e validação.

Figura 12 – Atividades do desenvolvimento incremental



Fonte: Sommerville (2011, p. 22)

Este modelo tem como vantagens conter maior interação com o cliente, pelo fato de a cada ciclo ser fornecido *feedback* das funcionalidades entregues com as devidas melhorias a serem realizadas no próximo incremento. Este modelo é ideal quando não se tem os requisitos bem definidos no início de projeto, sendo necessário coletá-los a medida que o mesmo amadurece.

5.2 Ferramentas Utilizadas

Nesta seção são abordadas as ferramentas que serão utilizadas no projeto, incluindo as ferramentas para modelagem e codificação da aplicação proposta.

5.2.1 JetBrains PhpStorm

A ferramenta PhpStorm¹, da empresa JetBrains, consiste em um ambiente de desenvolvimento de software utilizado para a codificação de aplicações em HTML (*HyperText Markup Language*), linguagem de marcação que especifica o formato de texto exibido em um navegador, JavaScript, uma linguagem de scripts baseado no navegador, e PHP, uma linguagem de script para criação de páginas web dinâmicas no lado do servidor (DEITEL, 2008). Neste ambiente serão desenvolvidas todas as funcionalidades, tais como: a aplicação web, *chatterbot* e a API (*Application Programming Interfaces*), uma interface que expõe um conjunto de dados e funções para facilitar a interação e a troca de informações ao consumir os serviços da web (MASSÉ, 2012).

5.2.2 Astah Community

A ferramenta Astah Community² consiste em um ambiente de modelagem de diagramas UML (*Unified Modeling Language*) no qual serão apresentadas as funcionalidades do sistema de forma padronizada no ciclo de desenvolvimento de software. Dentre os diagramas existentes, podemos citar o diagrama de caso de uso, de classe, de interação, de atividades, entre outros.

Para o projeto foi modelado apenas os diagramas de casos de uso, que, segundo Engholm (2010), são utilizados na fase inicial do desenvolvimento do software, pois a modelagem irá mostrar as funcionalidades previstas para o sistema e os usuários que utilizarão das funcionalidades.

5.2.3 Draw.IO

A ferramenta Draw.IO³ é uma aplicação *online* desenvolvida pela Google para diversos tipos de modelagens, por exemplo, diagramas UML, fluxogramas, mapas mentais, *wireframe*, entre outros.

Para o projeto serão utilizados os fluxogramas e formas livres para a modelagem das interações do *bot* e apresentação da API.

¹ Disponível em: <<https://www.jetbrains.com/phpstorm/>>.

² Disponível em: <<http://astah.net/editions/community>>.

³ Disponível em: <<https://www.draw.io/>>.

5.3 Mean Stack

Mean Stack consiste em um conjunto de ferramentas para o desenvolvimento de aplicações web, no qual MEAN, significa (MongoDB, Express, Angular e NodeJS). Nas próximas seções são apresentados os componentes mencionados e os fragmentos desenvolvidos com os mesmos.

5.3.1 MongoDB

O MongoDB⁴ consiste em um banco de dados NoSQL⁵ orientado a documentos cuja estrutura de dados é composta de pares de campos e valores no formato BSON, semelhante os objetos JSON⁶ (*JavaScript Object Notation*). Segundo Edward e Sabharwal (2015), os documentos podem ter diferentes esquemas, o que significa que sua estrutura pode mudar à medida que a aplicação evolui, ou seja, não necessariamente precisam ser do mesmo padrão, contendo informações diferentes em cada documento armazenado.

A Tabela 1 mostra a comparação das termologias presentes nos bancos de dados relacionais, tendo como exemplo, o MySQL, e o banco não relacional, tendo como exemplo, o MongoDB.

Tabela 1 – Comparação entre MySQL e MongoDB

MySQL	MongoDB
Tabelas	Coleções
Linhas	Documentos
Colunas	Campos

Ainda para Edward e Sabharwal (2015), o MongoDB possui alta escalabilidade, desempenho, e disponibilidade, permitindo com que as consultas sejam realizadas mais rapidamente pelo fato de armazenar seus dados em documentos no formato BSON, no qual todos os dados relacionados estão presentes em um só lugar.

5.3.2 Express

O express⁷ é um *framework* para aplicações com Node.JS, que fornece um conjunto de recursos, por exemplo, estrutura do servidor, métodos de manipulação com as rotas da API, entre outros recursos essenciais para o desenvolvimento na plataforma. Sem o uso de um *framework* no processo de desenvolvimento do sistema, a conclusão do produto

⁴ Disponível em: <<http://www.mongodb.com/>>.

⁵ NoSQL (Not only SQL) termo relacionado a banco de dados não relacionais, onde os dados não estruturados são usados para armazenar grande quantidade de informações (EDWARD; SABHARWAL, 2015).

⁶ Formatação da informação para troca de dados. Disponível em: <<http://www.json.org/json-pt.html>>.

⁷ Disponível em: <<http://expressjs.com/>>.

final torna-se mais demorada e custosa, devido às funcionalidades principais terem que ser desenvolvidas do zero na aplicação.

5.3.3 Angular

O Angular⁸ é um *framework* JavaScript desenvolvido pela Google para o desenvolvimento de SPA⁹ (*Single Page Applications*). O objetivo do *framework* é construir aplicativos com código reutilizável para que possa ser implantado em diferentes plataformas, sendo estas, aplicações web, *mobile* e *desktop*.

Atualmente, o Angular encontra-se em sua versão número 2. Embora o *framework* seja baseado na linguagem JavaScript, seu desenvolvimento é realizado em uma outra versão do mesmo - chamada TypeScript - que, ao término, terá o código fonte final compilado para o JavaScript puro.

No desenvolvimento com o Angular, as funcionalidades do sistema são criadas por meio de componentes, que poderão ser reaproveitados para compor o sistema em sua totalidade. No projeto, o Angular foi utilizado para desenvolver o *front-end* da aplicação com todas as requisições à API para a exibição das informações ao usuário.

5.3.4 NodeJS

O Node.JS¹⁰ é uma plataforma construída sobre o motor V8¹¹ do navegador chrome para execução do JavaScript no ambiente do servidor, possibilitando o desenvolvimento de aplicações de alta escalabilidade. Nos servidores atuais, por exemplo, PHP e JAVA, para cada conexão estabelecida do usuário com o servidor é criada uma nova *thread* de execução, sobrecarregando os servidores para atender às requisições estabelecidas.

Com o ambiente Node.JS, ao invés de ser criada uma nova *thread* no servidor, é apenas disparado um evento que será executado sobre a plataforma do Node, propondo uma *thread* única para lidar com as requisições, sendo assim um modelo de arquitetura de entrada/saída e orientada a eventos, que não bloqueia as novas requisições estabelecidas, tornando leve o desenvolvimento de novas aplicações com um grande números de conexões simultâneas.

Os sistemas Node.JS lidam muito bem com aplicações em tempo real, por exemplo, chats, transações bancárias, entre outras, que precisam de tempo de resposta rápida da aplicação, oferecendo diversas vantagens ao ser utilizada para o *chatterbot* proposto. Com

⁸ Disponível em: <<https://angular.io/>>.

⁹ SPA - Conceito para aplicações que não realizam a atualização da página no navegador, ao serem requisitadas novas informações.

¹⁰ Disponível em: <<https://nodejs.org/en/>>.

¹¹ Motor V8 consiste em um interpretador criado pela Google escrito na linguagem de programação C++ para ser usado em seu navegador, mas com a possibilidade de poder ser incorporado a qualquer aplicação.

o Node.JS será criada a API que irá servir as requisições do *chatterbot* e as requisições da aplicação web.

5.4 Rede / Segurança

A seção a seguir apresenta os conceitos e a arquitetura cliente/servidor que envolvem a troca de informações no projeto proposto e os quesitos de segurança implementados.

5.4.1 Arquitetura REST

A arquitetura REST (*Representational State Transfer*) é um modelo de *Web Services*¹² baseado no protocolo HTTP para servir aplicações em rede, permitindo que diferentes aplicações possam acessar o servidor para consumir os serviços disponíveis. As requisições ao servidor pelos clientes ocorrem por meio das operações mais comumente utilizadas, são elas: GET, POST, PUT, DELETE. A arquitetura não consiste em um modelo obrigatório que deve ser seguido, mas apenas orientações para a troca de informações entre sistemas.

Neste projeto foi utilizada a arquitetura REST, para que tanto a aplicação web quanto o *bot* possam acessar as mesmas informações armazenadas no banco de dados.

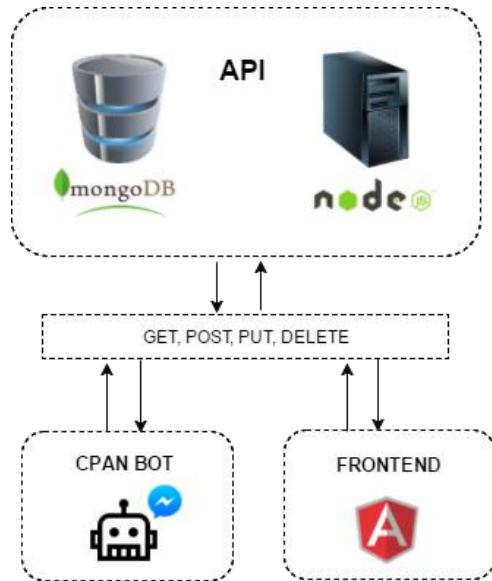
5.4.2 API RESTful com NodeJS

API RESTful trata-se de uma implementação da arquitetura REST sem distinção de linguagem de programação a ser utilizada para a disponibilização do serviço. Neste projeto, o desenvolvimento da API foi realizada na plataforma Node.JS com o armazenamento das informações no banco NoSQL MongoDB.

A Figura 13 ilustra uma API RESTful em Node.JS com mongoDB sendo utilizada para servir as informações a duas aplicações, são elas: *chatterbot* e *front-end*, ambas acessando a API por meio dos métodos disponibilizados.

¹² Serviço utilizado para realizar a integração entre os diferentes sistemas e arquiteturas, permitindo a troca de informações.

Figura 13 – API RESTful com Node.JS



Para o projeto, o *chatterbot* implementado na plataforma Node.JS e a aplicação web implementada em Angular serão responsáveis por consumir a API, no qual o acesso deverá ser realizado por meio de autorização, sendo esta concedida ao realizar o *login* na aplicação. Na seção seguinte será abordado o método de proteção utilizado para acesso às rotas da API.

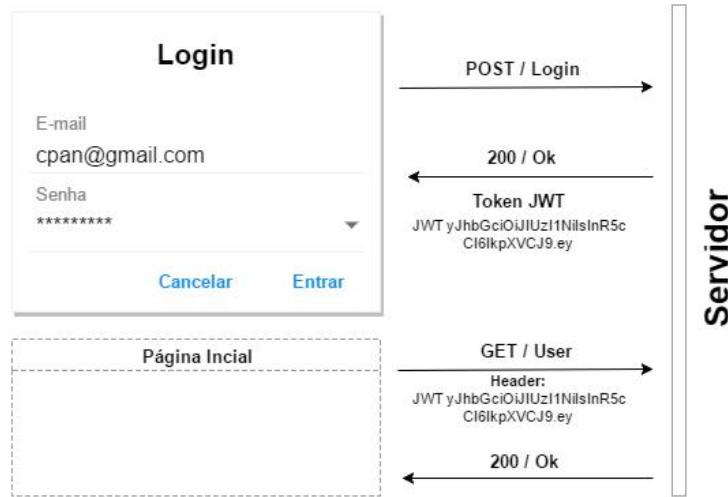
5.4.3 Autenticação com Json Web Token

Json Web Token (JWT)¹³ consiste em um método de autenticação, no qual o servidor gera um *token* e encaminha para o solicitante. O *token* gerado contém as informações essenciais para que a cada requisição realizada pelo cliente ao servidor utilize-o para acessar as informações pertencentes às rotas protegidas.

A Figura 14 ilustra o processo de requisição ao servidor, no qual ao realizar o *login* de acesso na aplicação, o servidor irá encaminhar o *token* gerado para quem o requisitou, que deve mantê-lo guardado para que a cada requisição ao servidor este seja encaminhado junto para a validação de acesso às rotas.

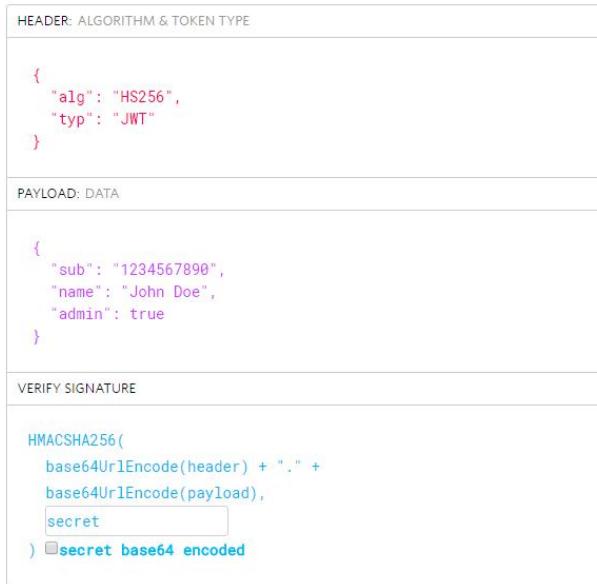
¹³ Disponível em: <<https://jwt.io/>>.

Figura 14 – Autenticação com JWT



O acesso às rotas protegidas por JWT deve ser liberado apenas caso o *token* enviado pelo cliente seja válido, e este deve ser composto pela seguinte estrutura *Header*, *Payload*, *Signature*, como mostra a Figura 15.

Figura 15 – Estrutura do JWT



Fonte: Disponível em: <<https://jwt.io/>>.

Para um *token* mais seguro, algumas medidas podem ser tomadas, por exemplo, expiração do *token* gerado, não guardar senhas no *token*, e limite de tentativas de requisições caso a verificação de acesso não seja válida.

6 Arquitetura do Sistema

Neste capítulo será abordado o processo de desenvolvimento para construir a API, aplicação web e *chatterbot* propostos.

6.1 Desenvolvimento da API

A API REST em node.js será responsável por receber todas as requisições provindas da aplicação web e *chatterbot*. Sendo a API desenvolvida a partir do *framework* express.js que possibilitará criar a aplicação de forma simples e rápida (EXPRESS.JS, 2017).

Para isso será necessário ter as seguintes pacotes instalados que podem ser baixadas do repositório oficial do node.js a partir do terminal do sistema operacional.

Passo 1: npm init

Passo 2: npm install nome_do_pacote --save

- Pacotes

- express - *Framework* web para node.js
- body-parser - Pacote utilizado para lidar com o conjunto de dados enviadas junto com as requisições HTTP.
- mongoose - Utilizado para manipular o banco de dados mongodb.
- bcrypt-nodejs - Utilizado para criptografar os dados.
- cheerio - Pacote utilizado para manipular elementos HTML das páginas web.
- request - Pacote utilizado para realizar requisições HTTP.
- morgan - Pacote utilizado para visualizar *logs* de execução.
- jsonwebtoken - Pacote utilizado para criar e validar *tokens*.
- passport - Pacote utilizado para trabalhar com estratégias de autenticação.
- passport-jwt - Pacote utilizado para trabalhar com estratégias de autenticação com json web tokens.
- natural - Pacote utilizado para trabalhar com processamento de linguagem natural.

A Figura 16 mostra o arquivo *package.json* criado na estrutura do projeto após o término da instalação dos pacotes.

Figura 16 – Arquivo de configuração - package.json

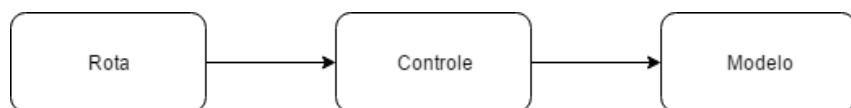
```
{
  "name": "cpanbot",
  "version": "1.0.0",
  "description": "Trabalho de Conclusão de Curso",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "author": "Westerley Reis",
  "license": "ISC",
  "dependencies": {
    "bcrypt-nodejs": "0.0.3",
    "body-parser": "^1.15.2",
    "cheerio": "^0.22.0",
    "crypto": "0.0.3",
    "express": "^4.14.0",
    "jsonwebtoken": "^7.2.1",
    "mongoose": "^4.7.3",
    "morgan": "^1.7.0",
    "passport": "^0.3.2",
    "passport-jwt": "^2.2.1",
    "request": "^2.79.0",
    "natural": "^0.4.0"
  }
}
```

O desenvolvimento da API está dividido da seguinte forma:

1. Configurações: Pasta com os arquivos de configurações para as informações do banco de dados, chave secreta para manipular o *token* e porta utilizada para rodar a aplicação.
2. Rotas: Pasta com os arquivos de rotas¹ da aplicação tais como: coordenação, departamento, início, notícias, telefones e usuários.
3. Controle: Pasta com os arquivos utilizados para manipular os dados recebidos das URLs.
4. Modelo: Pasta com a estrutura do banco de dados em mongoDB.

A Figura 17 ilustra o processo utilizado para persistir e requisitar as informações da API. O primeiro a receber a requisição é o arquivo de rotas, que encaminha os dados para o controlador, onde serão realizadas as validações e salvar no banco de dados.

Figura 17 – Processo utilizado para requisição das informações



¹ Caminho utilizado no servidor para acessar as informações por meio das requisições HTTP.

A API irá atender às requisições para as funcionalidades listadas na subseção A.1.1, estas são realizadas para o cadastro, alteração, listagem e exclusão de usuários, setores, telefones e cursos. Para a funcionalidade de notícias, a API apenas atenderá às requisições de listagem e cadastro, porém o cadastro acontece através da extração dos dados da página oficial de notícias do cpan <<https://cpan.ufms.br/noticias/>>.

A extração dos dados da página, ou *web scraping*, é realizada a partir do pacote instalado anteriormente “cheerio”, que irá realizar a raspagem dos elementos HTML presentes na página selecionados pelo desenvolvedor. Durante o processo de *web scraping* alguns problemas podem ocorrer no decorrer do tempo, por exemplo: 1) o *layout* do site sofrer mudanças e os elementos HTML selecionados saírem do contexto da extração. 2) os acessos simultâneos a página durante o processo de extração dos dados serem negados pelo servidor do site.

Para contornar o problema número dois listado acima, a extração das notícias ocorrerá apenas quando houver o cadastro de uma nova postagem no site do cpan. No processo de cadastro da notícia, o administrador da página do Câmpus do Pantanal deverá acessar a rota de notícias da API para a extração ocorrer.

Na seção seguinte serão apresentados os pacotes necessários e o processo utilizado para desenvolver a aplicação web em angular versão 2.

6.2 Desenvolvimento da Aplicação Web

Para o desenvolvimento da aplicação web em angular versão 2 foi utilizada a estrutura elaborada por Vlado Tesanovic, que pode ser encontrada a partir do repositório no github² <<https://github.com/vladotesanovic/angular2-express-starter>>. O mesmo foi utilizado por motivo das configurações necessárias para a implantação da aplicação no servidor já estarem configuradas.

A criação de um novo projeto também pode ser realizada a partir do projeto oficial no github *Quickstart Project* do angular ou por meio da ferramenta angular-cli, que pode ser encontrada no repositório oficial do node.js.

As aplicações em angular são desenvolvidas a partir de componentes, que consistem na combinação de uma classe de elementos responsáveis por controlar uma parte de um modelo HTML, ou seja, o processo consiste em escrever vários componentes que, ao final irá compor a aplicação em sua totalidade, sendo estes criados a partir da propriedade seletor (tag HTML customizada) e *template* (código HTML).

Entre os módulos nativos de dependência utilizados para o desenvolvimento da aplicação que tratam das requisições http, manipulação de formulários, entre outros

² Plataforma construída para os desenvolvedores hospedarem código fontes, com o intuito colaborativo.

módulos essenciais importados pelo angular, estão:

- Module (NgModule, FormsModule, HttpClientModule, CommonModule, RouterModule);
- Components (Router, Routes, ActivateRoute, Subscription);
- Service (Injectable, Observable, HTTP, Headers, CanActivate).

Para os módulos externos utilizados estão:

- Ng2PaginationModule - Utilizado para criar paginações;
- compodoc - Utilizado para criar a documentação dos módulos implementados.

Entre os componentes criados para o funcionamento da aplicação estão:

- Módulos (autenticação, coordenação, telefone, setor, notícias e usuário);
- Componentes (autenticação, coordenação, telefone, setor, notícias e usuário);
- Serviços (autenticação, coordenação, telefone, setor, notícias e usuário).

Na próxima seção será abordado os passos necessários para construir o *chatterbot* para o Facebook Messenger, que consiste no projeto principal proposto como trabalho de conclusão de curso.

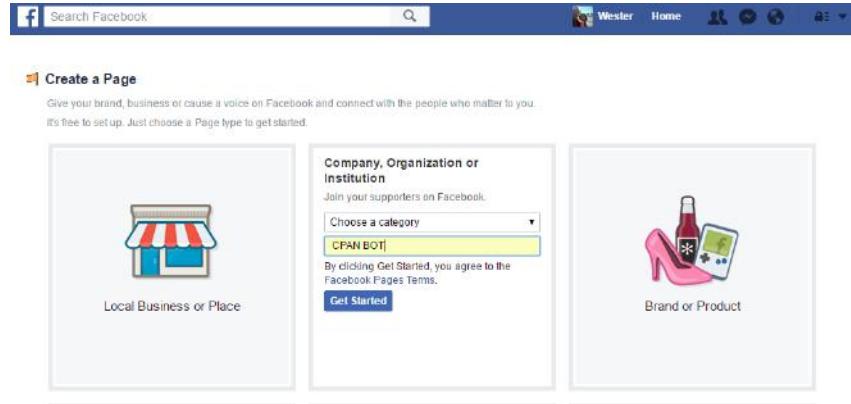
6.3 Desenvolvimento do Chatterbot

Para o desenvolvimento do *chatterbot* foi utilizado a API oficial Messenger Plataform³, que disponibiliza um guia inicial na plataforma node.js em seu repositório no github <<https://github.com/fbsamples/messenger-platform-samples>>.

O *chatterbot* deve estar vinculado a uma página do Facebook que deverá ser criada a partir de uma conta de usuário já cadastrada. A Figura 18 mostra o processo de criação da página na categoria *Company, Organization or Institution*.

³ Disponível em: <<https://developers.facebook.com/docs/messenger-platform>>.

Figura 18 – Criação da página CPAN BOT



A princípio o *chatterbot* não estará vinculada a página oficial do Câmpus do Pantanal no Facebook por motivos de aprovação⁴ e testes. Com isso, a página criada estará visível apenas para usuários de testes que foram selecionados para interagirem com o *chatterbot* desenvolvido. A Figura 20 mostra a página criada com a capa e foto de perfil configurados.

Figura 19 – Página do Facebook configurada



O próximo passo será vincular a página a um aplicativo do Facebook que poderá ser criado a partir de uma conta de desenvolvedor no site <<https://developers.facebook.com/>>. Posteriormente, deve-se adicionar o Messenger em *add product* para que possa ser configurada a comunicação do *chatterbot* com a API criada na seção 6.1.

⁴ Todos os bots criados para o Messenger precisam de aprovação pela equipe do Facebook para que possam estar disponíveis para todos os usuários.

Figura 20 – Processo para criar um aplicativo

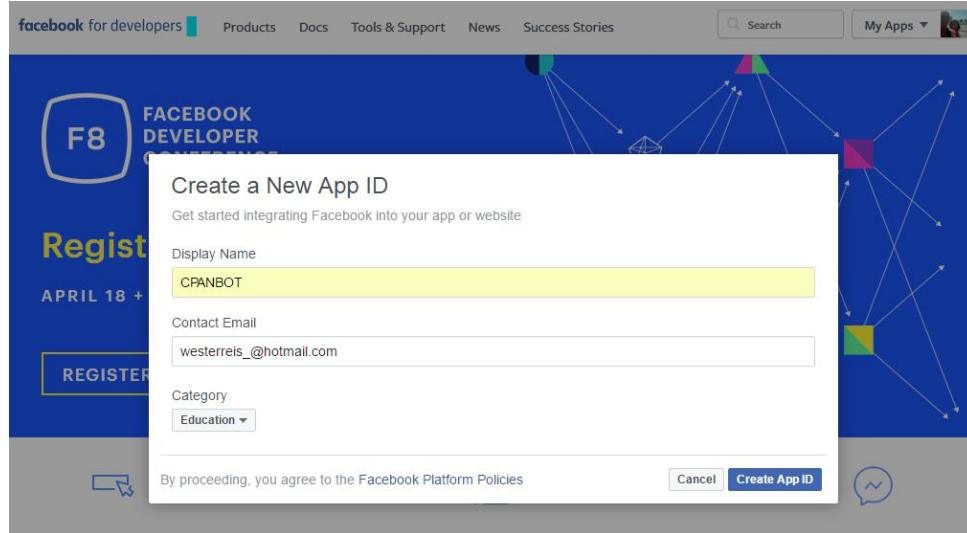
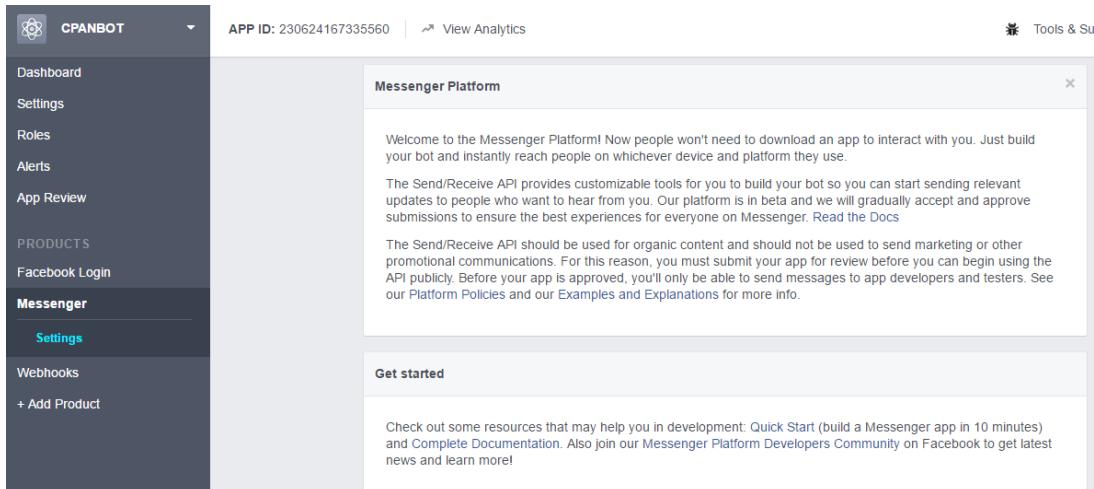
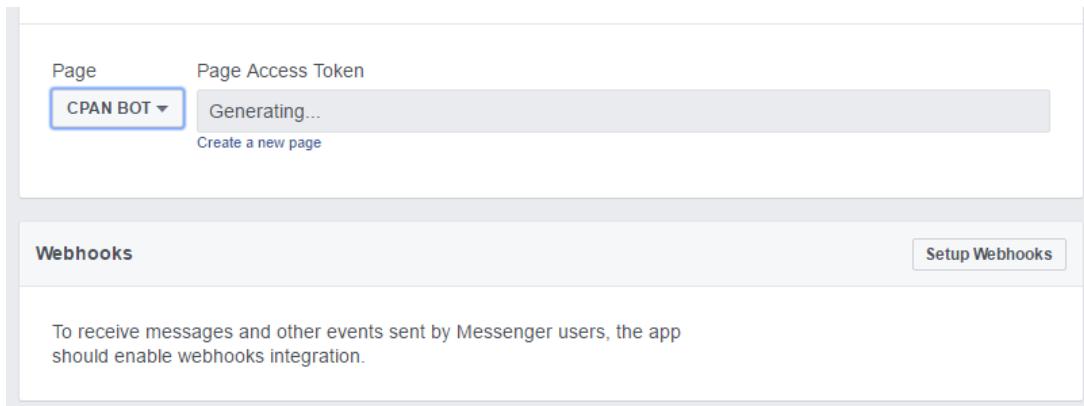


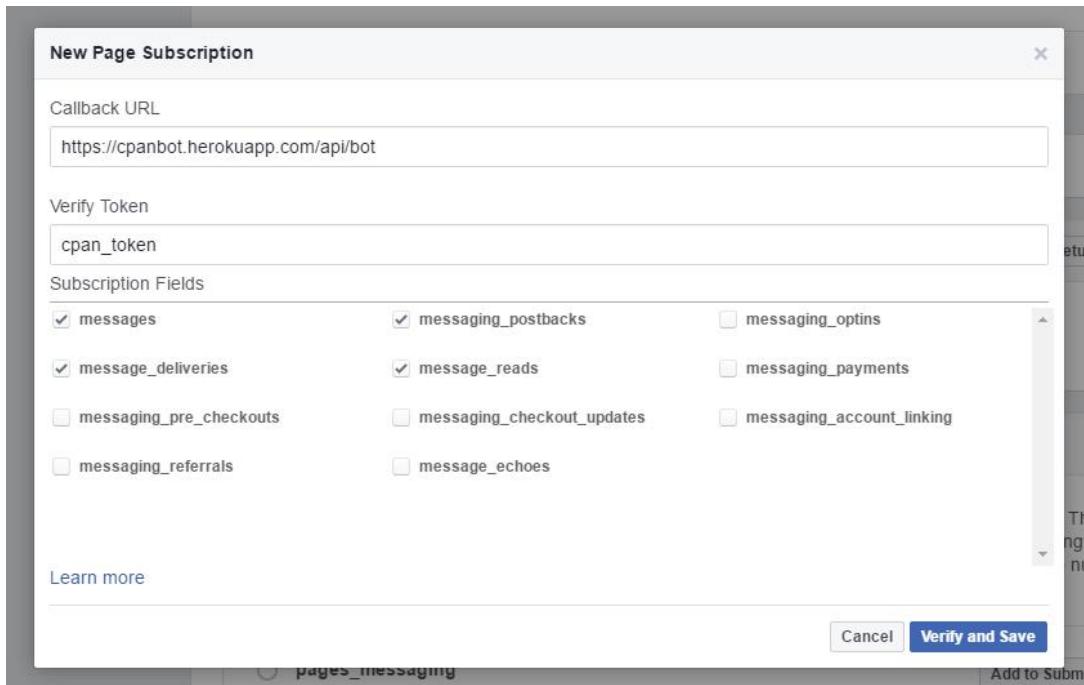
Figura 21 – Produto Messenger adicionado ao aplicativo



Após ter adicionado o produto ao aplicativo, o passo seguinte será configurar os *webhooks* para estabelecer a conexão entre o *chatterbot* e a API desenvolvida. Para configurá-la basta selecionar o botão *setup webhooks* como mostra a Figura 22.

Figura 22 – Selezionando o botão *setup webhooks*

Ao selecioná-lo, as configurações necessárias serão exibidas, como mostra a Figura 23. Entre os campos configurados estão: URL da API, *token* verificador que deve ser adicionado ao código fonte durante o desenvolvimento do *chatterbot*, e os itens que devem ser selecionado, tais como: *messages*, *message_deliveries*, *messaging_postbacks*, *message_reads*.

Figura 23 – Configuração do *webhooks*

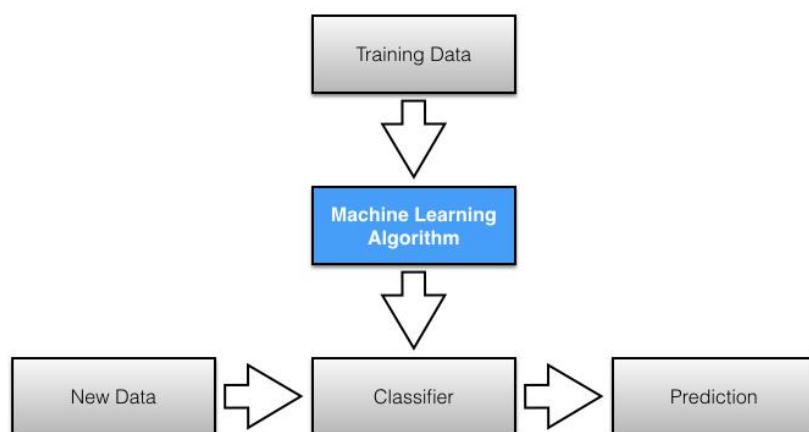
Após o processo de configuração do aplicativo, o desenvolvimento inicia-se a partir das funcionalidades descritas subseção A.1.2. Todas as mensagens enviadas ao *bot* serão recebidas pela API implementada, que encaminhará as mensagens correspondentes solicitadas.

Para que o *bot* envie a mensagem corretamente, foi utilizado o algoritmo de classificação Naive Bayes, que é baseado no teorema de probabilidade de Bayes, no qual trata da probabilidade de um evento ocorrer baseado em um conjunto de dados previamente treinados, sendo estes chamados de aprendizado supervisionado que envolve a aprendizagem de uma função a partir de exemplos de suas entradas e saídas (NORVING; RUSSELL, 2004).

De acordo com Rish (2001), o classificador Naive Bayes é um algoritmo ingênuo pois trata o conjunto de dados como recursos independentes, mostrando-se eficaz em aplicações que lidam com classificação de texto e diagnósticos médicos. Em outras palavras, Sunil (2015) um classificador Naive Bayes assume que a presença de uma característica particular em uma classe não está relacionada com a presença de qualquer outro recurso, por exemplo, todas as características para que uma fruta possa ser considerada como maçã, tais como: vermelha, redonda, e tiver cerca de 3 polegadas de diâmetro, contribuem de forma independente para a probabilidade do fruto ser uma maçã, mesmo que a existência das características sejam dependentes.

A classificação de textos tem como objetivo classificar um determinado dado com o intuito de prever a classe correta a partir de suas características. Mas para isso é preciso que um conjunto de amostras passe por treinamento por um classificador e, a partir de novos conjuntos de dados, o algoritmo determine a classe correta para o texto analisado. A Figura 24 mostra o processo de aprendizado supervisionado da técnica de classificação.

Figura 24 – Processo de classificação de texto



Fonte: Disponível em: <<https://goo.gl/iMfVPe>>.

Conforme Sunil (2015) a equação utilizada pelo algoritmo Naive Bayes para calcular a probabilidade de um evento ocorrer é dada por:

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

No qual temos:

- $P(c|x)$ é a probabilidade do evento “c” ocorrer de acordo com as amostras de treinamento “x”
- $P(c)$ é a probabilidade a priori do evento “c”
- $P(x|c)$ é a probabilidade de “x” dado “c”
- $P(x)$ é a probabilidade das amostras de treinamento

Para o *bot* proposto a classificação das mensagens foram treinadas a partir do pacote “natural” instalado na seção 6.1. Entre as mensagens e classes classificadas estão:

Tabela 2 – Classes e mensagens para treinamento do *bot*

Classes	Mensagens
notic	- noticias - novidades
inform	- informacoes - informacao
department	- setor - departamento
telefon	- telefone - tel - cel
cours	- curso - coordenacao - coordenacoes
administracao	- administracao - adm
contabeis	- ciencias contabeis - contabeis - contabilidade
direito	- direito
psicologia	- psicologia
sistemas	- sistemas de informação - sistemas de info - sistemas
biologia	- ciencias biologicas - biologia
fisica	- educacao fisica - edu fisica
geografia	- geografia
historia	- historia

letras	- letras - letras portugues ingles - letras portugues espanhol
matematica	- matematica
pedagogia	- pedagogia
bacharelado	- bacharelado
licenciatura	- licenciatura
pos	- pos graduacao - pos - mestrado - doutorado
no	- nao - tchau - bye - nada - ok
yes	- sim - yeh - desejo
initial	- oi - ola - bom dia - boa tarde - boa noite
ingresso	- forma de ingresso - quais sao as formas de ingresso - como entrar na faculdade - como faco para entrar na universidade

De acordo com a Tabela 2, todas as mensagens enviadas pelo usuário para o *bot* e as frases treinadas com o classificador não apresentam acentuação, pelo fato de ser realizado o processamento de limpeza dos dados, sendo assim evitando o tratamento de mais frases, pois há casos de o usuário escrever da forma correta **com acentuação** e casos de o usuário escrever da forma incorreta **sem acentuação**.

Para o desenvolvimento do *chatterbot* com o Messenger API, são disponibilizadas diversas funcionalidades para que a interação com o usuário torne-se mais acessível e agradável visualmente. Dentre os recursos utilizados estão:

1. Mensagem de boas vindas;
2. Botão começar;
3. Menu persistente;
4. Mensagens de textos;

5. Template estruturados;
6. Mensagens estruturadas;
7. Respostas rápidas.

Cada recursos utilizados na interação do *bot* com o usuário para atender as funcionalidades identificadas ao Câmpus do Pantanal podem ser observadas nas Figura 25, 26, 27, 28, 29

Figura 25 – Mensagem de boas vindas com o botão começar



Figura 26 – Mensagem de inicialização de conversa e menu persistente



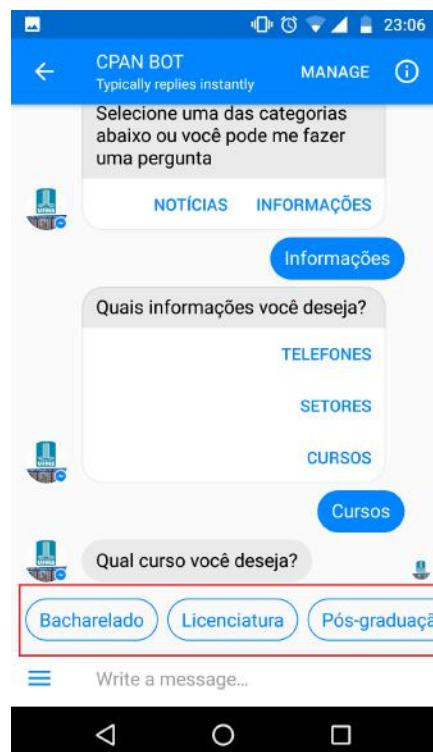
Figura 27 – Template estruturado no formato de carrossel



Figura 28 – Mensagem estruturada



Figura 29 – Resposta rápida



6.4 Hospedagem

Para a implantação da aplicação foi contratado o servidor Heroku⁵, que consiste em um ambiente para hospedagem em nuvem. O Heroku possibilita a hospedagem de forma rápida, pois utiliza o sistema de versionamento de código fonte git⁶ para realizar o upload da aplicação ao servidor.

Atualmente o Heroku possui três planos de hospedagem, como o mostra a Figura 30, os planos *Free*, *Hobby* e *Professional*.

Figura 30 – Planos de Hospedagem de aplicativos

Plano	Características	Preço
Free	Ícone: Planeta com satélite. Descrição: Ideal for experimenting with cloud applications in a limited sandbox. CORE PLATFORM FEATURES: NEVER SLEEPS USES AN ACCOUNT-BASED POOL OF FREE DYNOS CUSTOM DOMAINS 512 MB RAM 1 web/1 worker	\$0
Hobby	Ícone: Planeta com satélite. Descrição: Perfect for small scale personal projects and hobby apps. CORE PLATFORM FEATURES: NEVER SLEEPS FREE SSL FOR CUSTOM DOMAINS APPLICATION METRICS MULTIPLE WORKERS FOR MORE POWERFUL APPS 512 MB RAM 10 Process Types \$7 per dyno/month	\$7
Professional	Ícone: Hexágono com símbolos. Sub-plano: Standard (1X 2X). Descrição: Enhanced visibility, performance, and availability for powering your professional applications. ALL HOBBY FEATURES + SIMPLE HORIZONTAL SCALABILITY THRESHOLD ALERTS PREBOOT 512MB OR 1GB RAM	\$25
	Ícone: Hexágono com símbolos. Sub-plano: Performance (M L). Descrição: Superior performance when it's most critical for your super scale, high traffic apps. ALL STANDARD FEATURES + MIX WITH STANDARD 1X, 2X DYNOS DEDICATED AUTOSCALING 2.5GB OR 14GB RAM	\$500
	Descrição: \$25 - \$500 per dyno/month <small>prorated to the second</small>	\$25 - \$500

Fonte: Disponível em: <<https://www.heroku.com/>>.

Para a hospedagem da aplicação proposta foi contratado o plano *Hobby* que está no valor de U\$7 USD, aproximadamente R\$21,80 reais na cotação do dia. O Heroku também disponibiliza ao hospedar a aplicação o domínio com a criptografia SSL, que é requerida no Messenger Platform ao configurar o *webhook* na seção 6.3.

⁵ Disponível em: <<https://www.heroku.com/>>.

⁶ Disponível em: <<https://git-scm.com/>>.

7 Limitações e Resultados

Neste capítulo apresentam-se os resultados obtidos com a pesquisa e o desenvolvimento da API, aplicação web e o *chatterbot* proposto, bem como as limitações que surgiram para implementá-las.

As limitações que ocorreram inicialmente no projeto, foi em relação a modelagem do banco de dados NoSql da API, que apresenta uma arquitetura diferente dos bancos de dados relacionais tradicionais, no qual o modo em que os dados são inseridos e organizados. Na aplicação web, o desenvolvimento com o angular, que atualmente encontra-se em uma nova versão, foi dificultado no que diz respeito à coleta de informações referentes aos problemas vivenciados na etapa de desenvolvimento, principalmente relacionados à comunicação entre os componentes e autenticação com JWT, consequentemente atrasando o progresso do trabalho.

Para o *chatterbot* desenvolvido com a API oficial, Messenger Platform, as limitações identificadas foram: limite de elementos do carrossel utilizado para exibir os cursos do CPAN, que foi contornado a partir da divisão dos cursos em modalidade; versões antigas do aplicativo do Messenger não exibirem o botão “começar” utilizado para iniciar a conversa com o *bot*. Por fim, uma recente atualização da API que passou a também não exibir o botão “começar” mesmo na versão atual do aplicativo.

Apesar das barreiras que limitaram o avanço do projeto, todas as funcionalidades propostas foram atendidas. A princípio, o *chatterbot* desenvolvido não foi publicado para todos os usuários do Facebook, uma vez que o *bot* deve ser aprovado pelo Facebook antes de ser disponibilizado ao público e pelo fato que a aplicação web e a API desenvolvidas estarem disponibilizadas em uma hospedagem privada. Antes da disponibilização do *bot* ao público, o Facebook possibilita que possam ser adicionados usuários ao aplicativo para a realização de testes com o serviço disponibilizado.

Sendo assim, para realizar a análise dos resultados, uma pesquisa de satisfação foi aplicada a um grupo de 33 participantes, sendo a mesma respondida por 18 pessoas após a interação com o *bot*. Dentre os selecionados estão: acadêmicos, membros da comunidade externa, professores do curso de sistemas de informação e técnicos administrativos da instituição.

Como instrumento de coleta da pesquisa realizada, um formulário elaborado por meio do Google Forms¹ foi aplicado para mensurar a satisfação dos usuários que utilizaram o serviço disponibilizado.

¹ Disponível em: <https://www.google.com/forms/about/>

A análise realizada teve como parâmetro os seguintes itens:

1. Verificar se o usuário adicionado testou o *bot*.
2. Verificar a relação do usuário com a instituição, afim de identificar seu perfil que é classificado em 3 categorias: acadêmico, funcionário ou membro da comunidade externa.
3. Verificar se o serviço disponibilizado forneceu o acesso rápido ao conteúdo referente ao CPAN.
4. Verificar se as dúvidas do usuário foram sanadas.
5. Verificar se o usuário teria alguma sugestão a oferecer.
6. Por fim, a nota de satisfação com o serviço, considerando 1 como nota mínima e 5 como nota máxima.

Com os dados obtidos no parâmetro 1, pode-se concluir que o serviço de atendimento por meio de *chatterbot* proporcionou o acesso rápido às notícias postadas no site do CPAN e informações relacionadas à instituição. Além disso, com o parâmetro 4 na pesquisa aplicada, o serviço atendeu as dúvidas das pessoas que interagiram-se com o *bot*. Portanto, as pessoas ficaram satisfeitas com o serviço de atendimento prestado.

8 Conclusões e Trabalhos Futuros

O objetivo deste trabalho foi explorar de forma sucinta o Processamento de Linguagem Natural e realizar um levantamento de tecnologias existentes destinadas ao desenvolvimento de aplicações que envolvem os sistemas de diálogos. Tal abordagem possibilitou o desenvolvimento de um *chatterbot*, que utiliza como classificação das mensagens um algoritmo de aprendizado de máquina, possibilitando que o *bot* possa responder pelas informações de notícias, cursos, formas de ingresso, telefones e setores da instituição. Sendo também desenvolvida uma aplicação web para o gerenciamento dos conteúdos exibidos pelo *bot*.

Os estudos desenvolvidos proporcionam como trabalhos futuros a realização de novas pesquisas, que envolvam o desenvolvimento de sistemas de diálogos em linguagem natural. Por exemplo, algoritmos para o treinamento das mensagens na interação com a aplicação, sendo utilizado neste trabalho o classificador Naive Bayes. E a realização de pesquisas por novas tecnologias de processamento de linguagem natural, que surgiram no decorrer do desenvolvimento deste trabalho.

Como sugestões de trabalhos futuros para o *chatterbot* desenvolvido, algumas melhorias para serem implementadas foram propostas a partir da pesquisa realizada e observações feitas durante as interações dos usuários com o *bot*. As sugestões são listadas abaixo:

- Integrar o *chatterbot* com a página oficial do CPAN.
- Exibir informações sobre outros setores do CPAN com a possibilidade de acessá-las por meio da linguagem natural, uma alternativa para a técnica atualmente implementada, que consiste em botões clicáveis com as opções disponíveis.
- Possibilidade de treinar o *bot* a partir das mensagens que não foram corretamente respondidas, a fim de melhorar o serviço no futuro.
- Por fim, adicionar ao *bot* o calendário acadêmico do CPAN para acesso dos usuários.

Com as sugestões listadas a serem implementadas no serviço de atendimento, irá possibilitar que o *bot* consiga responder corretamente as informações não respondidas anteriormente, pois as mesmas passarão a pertencer à base de conhecimento do *chatterbot*.

Referências

- A.L.I.C.E. *An Introduction to A.L.I.C.E., the Alicebot engine, and AIML*. 2017. Disponível em: <<http://www.alicebot.org/about.html>>. Acessado em: 04 de abril de 2017. Citado na página 28.
- BUZAN, T. *Mapas Mentais*. [S.l.: s.n.], 2006. Tradução de Paulo Polzonoff Jr. Citado na página 87.
- CAVALCANTE, E. R.; BEZERRA, G.; NUNES, R. Lógica fuzz: Conceitos e aplicações. 2010. Natal - RN - Brasil. Citado na página 25.
- CORMEN, T. H. *Algoritmos: Teoria e prática*. [S.l.: s.n.], 2002. Tradução segunda edição [americana]. Citado na página 26.
- DEITEL, P. J. *Ajax, Rich Internet Applications e desenvolvimento Web para programadores*. São Paulo: [s.n.], 2008. Citado na página 36.
- DREHER, F. *A revolução dos robôs no atendimento aos clientes*. 2016. Disponível em: <<http://computerworld.com.br/revolucao-dos-robos-no-atendimento-aos-clientes>>. Acessado em: 4 de março de 2017. Citado na página 13.
- EDWARD, S.; SABHARWAL, N. *Practical MongoDB - Architecting, Developing and Administering MongoDB*. [S.l.: s.n.], 2015. Citado na página 37.
- ENGHOLM, H. *Engenharia de Software - Na prática*. [S.l.: s.n.], 2010. Citado 4 vezes nas páginas 36, 61, 63 e 64.
- EXPRESS.JS. *Express js*. 2017. Disponível em: <http://expressjs.com/>. Acessado em: 04/03/2017. Citado na página 42.
- FACEBOOK. *Messenger Platform*. 2017. Disponível em: <<https://messengerplatform.fb.com/>>. Acessado em: 04 de março de 2017. Citado na página 14.
- JURAFSKY, D.; MARTIN, J. H. *Speech and Language Processing*. [S.l.: s.n.], 2000. Citado na página 13.
- LAVEN, S. *What is a chatterbot?* 2017. Disponível em: <<http://www.simonlaven.com/>>. Acessado em: 04 de março de 2017. Citado na página 27.
- LIDDY, E. D. *Natural Language Processing*. 2. ed. [S.l.]: Marcel Decker, Inc, 2001. Citado 4 vezes nas páginas 17, 18, 19 e 20.
- LUFT, C. P. *Moderna Gramática Brasileira*. [S.l.: s.n.], 2008. Citado na página 18.
- MANFIO, E. R. Processamento de linguagem natural, robôs de conversação e linguística. v. 4, n. 1, 2014. Citado na página 27.
- MASSÉ, M. *REST API Design Rulebook*. [S.l.: s.n.], 2012. Citado na página 36.

- MICROSOFT. *Bot Framework*. 2017. Disponível em: <<https://dev.botframework.com/>>. Acessado em: 04 de março de 2017. Citado na página 30.
- MITTMANN, A. Implementação do chatterbot eliza na linguagem multiparadigma oz. 2006. Florianópolis. Citado na página 27.
- NORVING, P.; RUSSELL, S. *Inteligência Artificial*. [S.l.: s.n.], 2004. Citado na página 49.
- NORVING, P.; RUSSELL, S. *Inteligência Artificial*. 3. ed. [S.l.: s.n.], 2013. Citado 2 vezes nas páginas 24 e 25.
- NUNES, M. G. V. O processamento de línguas naturais: para quê e para quem? n. 73, maio 2008. ISSN - 0103-2585. Citado 2 vezes nas páginas 23 e 24.
- NUNES, M. G. V. et al. Introdução ao processamento das línguas naturais e algumas aplicações. ago. 2007. NILC-TR-07-10. Citado 6 vezes nas páginas 17, 18, 19, 20, 22 e 23.
- REISENWITZ, C. *Help Desk Technology Trends for 2016: Live Chat, Anticipatory Customer Service*. 2016. Disponível em: <<http://blog.capterra.com/help-desk-technology-trends-for-2016/>>. Acessado em: 04 de março de 2017. Citado 2 vezes nas páginas 26 e 29.
- RISH, I. An empirical study of the naive bayes classifier. 2001. Citado na página 49.
- SIQUEIRA, R. A. *Revista Eletrônica de Jornalismo Científico - Robôs com Inteligência Artificial*. 2005. Disponível em: <<http://www.inbot.com.br/novo/noticias/comciencia/index.php>>. Acessado em: 04 de março de 2017. Citado 2 vezes nas páginas 25 e 28.
- SOMMERVILLE, I. *Engenharia de Software*. 9º. ed. [S.l.: s.n.], 2011. Citado na página 35.
- SUNIL, R. *6 Easy Steps to Learn Naive Bayes Algorithm*. 2015. Disponível em: <<https://www.analyticsvidhya.com/blog/2015/09/naive-bayes-explained/>>. Acessado em: 21 de fevereiro de 2017. Citado na página 49.
- VIEIRA, R.; LIMA, V. L. Linguística computacional: princípios e aplicações. v. 3, 2001. Jornada de Atualização em Inteligência Artificial. Citado 4 vezes nas páginas 18, 19, 20 e 24.
- WIT.AI. *The Wit.ai Community*. 2017. Disponível em: <<https://wit.ai/>>. Acessado em: 04 de março de 2017. Citado 2 vezes nas páginas 29 e 33.

APÊNDICE A – Engenharia de Requisitos

A engenharia de requisitos é uma das etapas da engenharia de software que consiste em organizar todas as funcionalidades desejáveis para que um sistema possa funcionar. Para Engholm (2010, p. 151), “requisitos são como uma condição ou capacidade de um software que deve ser implementada por um sistema ou componente de sistema para se alcançar determinado fim”.

Nas próximas seções são apresentados os requisitos funcionais, não funcionais e regras de negócios utilizados neste projeto, tanto para a aplicação web quanto no *chatterbot*.

A.1 Identificação dos Requisitos Funcionais

Os requisitos funcionais do projeto apresentado são as funções ou ações que podem ser utilizadas pelo usuário durante a utilização da aplicação (ENGHOLM, 2010). Os requisitos apresentados estão divididos em requisitos funcionais da aplicação web e requisitos funcionais do *chatterbot*, para manter as distinções das funcionalidades de ambas as aplicações.

A.1.1 Requisitos funcionais (Aplicação Web)

Tabela 3 – Requisitos Funcionais (Aplicação Web)

ID	Descrição	Prioridade	Depende de
RF(1)	O sistema deverá permitir a autenticação de usuários na aplicação.	Alta	
RF(2)	A sistema deverá permitir o cadastro de novos usuários na aplicação.	Média	RF(1)
RF(3)	sistema deverá permitir que os usuários alterem suas informações cadastradas.	Média	RF(2)
RF(4)	O sistema deverá permitir que os usuários alterem suas senhas.	Média	RF(2)
RF(5)	O sistema deverá permitir a exclusão de usuários.	Média	RF(2)
RF(6)	O sistema deverá cadastrar automaticamente as três últimas notícias postadas no site do CPAN.	Alta	
RF(7)	O sistema deverá listar as notícias cadastradas.	Alta	RF(6)
RF(8)	O sistema deverá permitir aos usuários excluírem as notícias.	Alta	RF(1) RF(6)

RF(9)	O sistema deverá permitir aos usuários cadastrarem os telefones.	Alta	RF(1)
RF(10)	O sistema deverá permitir aos usuários editarem os telefones.	Alta	RF(9)
RF(11)	O sistema deverá permitir aos usuários excluírem os telefones.	Alta	RF(9)
RF(12)	O sistema deverá listar os telefones cadastrados.	Alta	RF(9)
RF(13)	O sistema deverá permitir aos usuários cadastrarem os setores.	Alta	RF(1)
RF(14)	O sistema deverá permitir aos usuários editarem os setores.	Alta	RF(13)
RF(15)	O sistema deverá permitir aos usuários excluírem os setores.	Alta	RF(13)
RF(16)	O sistema deverá listar os setores cadastrados.	Alta	RF(13)
RF(17)	O sistema deverá permitir o cadastro de perguntas relacionadas aos setores.	Média	RF(13)
RF(18)	O sistema deverá permitir aos usuários editarem as perguntas relacionadas aos setores.	Média	RF(17)
RF(19)	O sistema deverá permitir aos usuários excluírem as perguntas relacionadas aos setores.	Média	RF(17)
RF(20)	O sistema deverá permitir aos usuários cadastrarem os cursos do CPAN.	Alta	RF(1)
RF(21)	O sistema deverá permitir aos usuários editarem os cursos do CPAN.	Alta	RF(20)
RF(22)	O sistema deverá permitir aos usuários excluírem os cursos do CPAN.	Alta	RF(20)
RF(23)	O sistema deverá listar os cursos do CPAN.	Alta	RF(20)
RF(24)	O sistema deverá permitir o cadastro das formas de ingresso nos cursos do CPAN.	Média	RF(1)

A.1.2 Requisitos funcionais (Chatterbot)

Tabela 4 – Requisitos Funcionais (Chatterbot)

ID	Descrição	Prioridade	Depende de
RF(25)	O <i>bot</i> deverá informar as três últimas notícias postadas no site do CPAN.	Alta	RF(6)
RF(26)	O <i>bot</i> deverá mostrar as informações referente aos setores do CPAN.	Alta	RF(13)

RF(27)	O <i>bot</i> deverá mostrar as dúvidas relacionadas aos setores do CPAN.	Alta	RF(17)
RF(28)	O <i>bot</i> deverá fornecer os telefones da secretaria, coordenações e dos setores do CPAN.	Alta	RF(9)
RF(29)	O <i>bot</i> deverá mostrar as informações referente aos cursos do CPAN.	Alta	RF(20)
RF(30)	O <i>bot</i> deverá informar sobre as formas de ingresso nos cursos do CPAN.	Alta	RF(28)

A.2 Identificação dos Requisitos Não Funcionais

Os requisitos não funcionais definem como o sistema atenderá as funcionalidades do sistema, sejam elas ligadas a segurança, disponibilidade, restrições, entre outras especificações que o software deverá atender. Segundo Engholm (2010, p. 165), “os requisitos não funcionais não estão associados a funcionalidades disponibilizadas aos usuários”. Portanto, são especificações ligadas diretamente ao sistema.

Tabela 5 – Requisitos Não Funcionais

ID	Descrição	Categoria	Escopo	Prioridade
RNF(1)	O acesso ao sistema deverá ser realizado pela internet.	Portabilidade	Sistema	Alta
RNF(2)	A autenticação no sistema deverá ser realizada com json web token.	Segurança	Sistema	Alta
RNF(3)	O sistema deverá atender aos requisitos de usabilidade da IEEE.	Usabilidade	Sistema	Alta
RNF(4)	O <i>bot</i> deverá enviar as mensagens por meio de requisições à API.	Interoperabilidade	Sistema	Alta
RNF(5)	O sistema deverá possuir o protocolo HTTPS.	Segurança	Sistema	Alta
RNF(6)	O sistema deverá utilizar tecnologia para reaproveitamento dos componentes gerados.	Padrões	Sistema	Alta
RNF(7)	O e-mail para cadastro deverá ser único na base de dados.	Integridade / Segurança	Funcionalidade	Alta

A.3 Regras de Negócios

As regras de negócios descrevem as restrições ou ações a serem tomadas ao executar uma funcionalidade do sistema, indicando como o negócio irá funcionar. As regras de negócios do projeto estão separadas para a aplicação web e para o *chatterbot*.

A.3.1 Regras de negócios (Aplicação Web)

Tabela 6 – Regras de Negócios (Aplicação Web)

ID	Descrição	Prioridade
RN(1)	A autenticação no sistema deverá ser realizada pelo e-mail e senha cadastrados.	Alta
RN(2)	O usuário deverá estar autenticado para acesso a algumas funcionalidades do sistema.	Alta
RN(3)	Todos os usuários cadastrados poderão ter acesso às funcionalidades do sistema.	Alta
RN(4)	O acesso ao sistema deverá ser por meio do e-mail e senha cadastrados.	Alta
RN(5)	A senha de acesso ao sistema deverá conter no mínimo 6 caracteres do tipo alfanumérico.	Alta

A.3.2 Regras de negócios (Chatterbot)

Tabela 7 – Regras de Negócios (Chatterbot)

ID	Descrição	Prioridade
RN(4)	O bot deverá orientar os usuários quanto às ações a serem tomadas.	Alta
RN(5)	A interação do usuário com o <i>bot</i> deverá ser realizada por botões de comando e textos.	Alta
RN(6)	A língua utilizada para interação do <i>bot</i> com o usuário deverá ser a língua portuguesa..	Alta
RN(7)	O <i>bot</i> deverá utilizar das técnicas de PLN para a compreensão da língua natural do usuário.	Alta

A.4 Especificação dos Requisitos Funcionais

A especificação dos requisitos consiste em detalhar a lista dos requisitos funcionais levantados na fase inicial do projeto de maneira que esses requisitos sejam suficientes para retratar todo o comportamento funcional do sistema (ENGHOLM, 2010). Na especificação dos requisitos, os mesmos estão divididos nas seguintes etapas:

- Descrição: Descreve a funcionalidade a ser executada.

- Atores: Descreve a entidade que participa na interação com o sistema. Para este projeto foram adotados os seguintes atores:
 - Administrador: Tipo de usuário que possui acesso total ao sistema. Todos os usuários que acessam a aplicação web são do tipo administrador.
 - Usuário: Tipo de usuário comum, utilizado apenas para os usuários do Facebook.
- Pré-condições: Descreve as ações que devem ser tomadas antes da execução da funcionalidade.
- Pós-condições: Descreve o que ocorrerá após a conclusão da funcionalidade.
- Fluxo básico: Descreve o fluxo normal, esperado ou previsto da funcionalidade executada.
- Fluxo Alternativo: Descreve os desvios do fluxo básico durante sua execução.

A seguir são apresentadas as especificações dos requisitos funcionais identificados na fase inicial do projeto a fim de relatar o fluxo das atividades executadas pelo usuário e o sistema.

Tabela 8 – Requisito funcional RF01

Descrição:	O sistema deverá permitir a autenticação de usuários na aplicação.
Atores:	Administrador
Pré-condições:	- Ter cadastro no sistema.
Pós-condições:	- Usuários terão acesso ao sistema.
Fluxo de Eventos	
Fluxo Básico:	1. Os atores informam ao sistema que desejam efetuar o login. 2. Os atores informam os dados de acesso obrigatórios. 2.1 E-mail, Senha. 3. Os atores clicam no botão entrar. 4. O sistema valida os dados de acesso. 5. O sistema autentica o usuário na aplicação. 6. O sistema redireciona o usuário para a página principal.
Fluxo Alternativo:	1. (Passo 4) Validação das informações 1.1 Caso não tenha o usuário cadastrado ou a senha estiver incorreta, o sistema exibirá a mensagem “Usuário ou senha incorreta”.

Tabela 9 – Requisito funcional RF02

Descrição:	A sistema deverá permitir o cadastro de novos usuários na aplicação.
Atores:	Administrador
Pré-condições:	- Estar autenticado no sistema.
Pós-condições:	- Usuários cadastrados
Fluxo de Eventos	
Fluxo Básico:	<ol style="list-style-type: none"> 1. Os atores informam ao sistema que desejam efetuar o cadastro. 2. Os atores informam os dados obrigatórios. 2.1 Nome, E-mail, Senha, Confirmar Senha. 3. Os atores clicam no botão salvar. 4. O sistema valida os dados. 5. O sistema exibe a mensagem “Usuário salvo com sucesso”.
Fluxo Alternativo:	<ol style="list-style-type: none"> 1. (Passo 4) Validação das informações 1.1 Se os dados cadastrais obrigatórios não forem preenchidos é exibida a mensagem “Preencha este campo” 1.2 Caso já exista o e-mail cadastrado é exibida a mensagem “E-mail já existente”. 1.3 Caso não tenha o mínimo de caractere no campo senha é exibida a mensagem “O campo deve ter no mínimo 6 caracteres do tipo alfanumérico”.

Tabela 10 – Requisito funcional RF03

Descrição:	O sistema deverá permitir que os usuários alterem suas informações cadastradas.
Atores:	Administrador
Pré-condições:	- Estar autenticado no sistema.
Pós-condições:	- Informações alterada.
Fluxo de Eventos	
Fluxo Básico:	<ol style="list-style-type: none"> 1. Os atores informam ao sistema que desejam alterar o cadastro. 2. Os atores informam os dados obrigatórios. 2.1 Nome, E-mail. 3. Os atores clicam no botão salvar. 4. O sistema valida os dados. 5. O sistema exibe a mensagem “Usuário salvo com sucesso”.
Fluxo Alternativo:	<ol style="list-style-type: none"> 1. (Passo 4) Validação das informações 1.1 Se os dados cadastrais obrigatórios não forem preenchidos é exibida a mensagem “Preencha este campo” 1.2 Caso já exista o e-mail cadastrado é exibida a mensagem “E-mail já existente”.

Tabela 11 – Requisito funcional RF04

Descrição:	O sistema deverá permitir que os usuários alterem suas senhas.
Atores:	Administrador
Pré-condições:	<ul style="list-style-type: none"> - Estar autenticado no sistema. - Ter informações cadastradas.
Pós-condições:	- Senha alterada.
Fluxo de Eventos	
Fluxo Básico:	<ol style="list-style-type: none"> 1. Os atores informam ao sistema que desejam alterar a senha cadastrada. 2. Os atores informam os dados obrigatórios. 2.1 Senha, Confirmar Senha. 3. Os atores clicam no botão salvar. 4. O sistema valida os dados. 5. O sistema exibe a mensagem “Usuário salvo com sucesso”.
Fluxo Alternativo:	<ol style="list-style-type: none"> 1. (Passo 4) Validação das informações 1.1 Se os dados cadastrais obrigatórios não forem preenchidos é exibida a mensagem “Preencha este campo” 1.2 Caso não tenha o mínimo de caractere no campo senha é exibida a mensagem “O campo deve ter no mínimo 6 caracteres”.

Tabela 12 – Requisito funcional RF05

Descrição:	O sistema deverá permitir a exclusão de usuários.
Atores:	Administrador
Pré-condições:	<ul style="list-style-type: none"> - Estar autenticado no sistema. - Ter usuário cadastrado.
Pós-condições:	- Senha alterada.
Fluxo de Eventos	
Fluxo Básico:	<ol style="list-style-type: none"> 1. Os atores informam ao sistema que desejam excluir um usuário. 2. O sistema envia a mensagem “Deseja deletar o usuário?”. 3 Os atores clicam em “Ok”. 4. O sistema deleta o usuário cadastrado.
Fluxo Alternativo:	<ol style="list-style-type: none"> 1. (Passo 4) Validação das informações 1.1 Se o usuário clicar em cancelar, o sistema retorna a listar os usuários cadastrados.

Tabela 13 – Requisito funcional RF06

Descrição:	O sistema deverá cadastrar automaticamente as três últimas notícias postadas no site do CPAN.
Atores:	Administrador
Pré-condições:	- Ter notícia cadastrada no site do CPAN. - Estar autenticado no sistema.
Pós-condições:	- Notícia cadastrada.
Fluxo de Eventos	
Fluxo Básico:	1. Os atores por meio da barra de endereço no navegador, acessam o link de cadastro de notícias da API. 2. O sistema busca pelas últimas notícias e cadastrada no sistema.
Fluxo Alternativo:	1. (Passo 2) Caso ocorra um erro ao cadastrar as notícias, é exibida a mensagem “Não foi possível salvar as notícias”

Tabela 14 – Requisito funcional RF07

Descrição:	O sistema deverá listar as notícias cadastradas.
Atores:	Administrador
Pré-condições:	- Ter notícia cadastrada.
Pós-condições:	- Notícias listadas.
Fluxo de Eventos	
Fluxo Básico:	1. Os atores clicam na área relacionada a notícias. 2. O sistema exibe as notícias cadastradas.
Fluxo Alternativo:	1. (Passo 4) Exibição das notícias 1.1 Se não tiver notícias cadastradas, o sistema irá exibir a mensagem “Nenhuma notícia encontrada”.

Tabela 15 – Requisito funcional RF08

Descrição:	O sistema deverá permitir aos usuários excluírem as notícias.
Atores:	Administrador
Pré-condições:	- Estar autenticado no sistema. - Ter notícia cadastrada.
Pós-condições:	- Notícia excluída.
Fluxo de Eventos	
Fluxo Básico:	1. Os atores informam ao sistema que desejam excluir uma notícia. 2. O sistema envia a mensagem “Deseja deletar a notícia?”. 3. Os atores clicam em “Ok”. 4. O sistema deleta a notícia cadastrada.
Fluxo Alternativo:	1. (Passo 4) Mensagem do sistema 1.1 Se o usuário clicar em cancelar, o sistema retorna a listar as notícias cadastradas.

Tabela 16 – Requisito funcional RF09

Descrição:	O sistema deverá permitir aos usuários cadastrarem os telefones.
Atores:	Administrador
Pré-condições:	- Estar autenticado no sistema.
Pós-condições:	- Telefone cadastrado.
Fluxo de Eventos	<p>Fluxo Básico:</p> <ol style="list-style-type: none"> 1. Os atores informam ao sistema que desejam cadastrar um telefone. 2. Os atores informam os dados obrigatórios. 2.1 Nome, Pertence, Telefone. 3. Os atores clicam no botão salvar. 4. O sistema valida os dados informados. 5. O sistema exibe a mensagem “Telefone salvo com sucesso”. <p>Fluxo Alternativo:</p> <ol style="list-style-type: none"> 1. (Passo 4) Validação das informações 1.1 Se os dados cadastrais obrigatórios não forem preenchidos é exibida a mensagem “Preencha este campo”.

Tabela 17 – Requisito funcional RF10

Descrição:	O sistema deverá permitir aos usuários editarem os telefones.
Atores:	Administrador
Pré-condições:	- Estar autenticado no sistema.
Pós-condições:	- Telefone alterado no sistema.
Fluxo de Eventos	<p>Fluxo Básico:</p> <ol style="list-style-type: none"> 1. Os atores informam ao sistema que desejam alterar um telefone. 2. Os atores informam os dados obrigatórios. 2.1 Nome, Pertence, Telefone. 3. Os atores clicam no botão salvar. 4. O sistema valida os dados informados. 5. O sistema exibe a mensagem “Telefone salvo com sucesso”. <p>Fluxo Alternativo:</p> <ol style="list-style-type: none"> 1. (Passo 4) Validação das informações 1.1 Se os dados cadastrais obrigatórios não forem preenchidos é exibida a mensagem “Preencha este campo”.

Tabela 18 – Requisito funcional RF11

Descrição:	O sistema deverá permitir aos usuários excluírem os telefones.
Atores:	Administrador
Pré-condições:	- Estar autenticado no sistema. - Ter telefone cadastrado no sistema.
Pós-condições:	- Telefone excluído.
Fluxo de Eventos	
Fluxo Básico:	<ol style="list-style-type: none"> 1. Os atores informam ao sistema que desejam excluir um telefone. 2. O sistema envia a mensagem “Deseja deletar o telefone?”. 3. Os atores clicam em “Ok”. 4. O sistema deleta o telefone cadastrado
Fluxo Alternativo:	<ol style="list-style-type: none"> 1. (Passo 4) Mensagem do sistema 1.1 Se o usuário clicar em cancelar, o sistema retorna a listar os telefones cadastrados.

Tabela 19 – Requisito funcional RF12

Descrição:	O sistema deverá listar os telefones cadastrados.
Atores:	Administrador
Pré-condições:	- Ter telefone cadastrado no sistema.
Pós-condições:	- Telefone excluído.
Fluxo de Eventos	
Fluxo Básico:	<ol style="list-style-type: none"> 1. Os atores clicam na área relacionada aos telefones. 2. O sistema exibe os telefones cadastrados.
Fluxo Alternativo:	<ol style="list-style-type: none"> 1. (Passo 2) Exibição dos telefones 1.1 Se não tiver telefones cadastrados, o sistema irá exibir a mensagem “Nenhum telefone encontrado”

Tabela 20 – Requisito funcional RF13

Descrição:	O sistema deverá permitir aos usuários cadastrarem os setores.
Atores:	Administrador
Pré-condições:	- Estar autenticado no sistema.
Pós-condições:	- Setor cadastrado.
Fluxo de Eventos	<p>Fluxo Básico:</p> <ol style="list-style-type: none"> 1. Os atores informam ao sistema que desejam cadastrar um setor. 2. Os atores informam os dados obrigatórios. 2.1 Nome, Descrição do setor. 3. Os atores clicam no botão salvar. 4. O sistema valida os dados informados. 5. O sistema exibe a mensagem “Setor salvo com sucesso”. <p>Fluxo Alternativo:</p> <ol style="list-style-type: none"> 1. (Passo 4) Validação das informações 1.1 Se os dados cadastrais obrigatórios não forem preenchidos é exibida a mensagem “Preencha este campo”.

Tabela 21 – Requisito funcional RF14

Descrição:	O sistema deverá permitir aos usuários editarem os setores.
Atores:	Administrador
Pré-condições:	- Estar autenticado no sistema.
Pós-condições:	- Setor alterado no sistema.
Fluxo de Eventos	<p>Fluxo Básico:</p> <ol style="list-style-type: none"> 1. Os atores informam ao sistema que desejam alterar um setor. 2. Os atores informam os dados obrigatórios. 2.1 Nome, Descrição do setor. 3. Os atores clicam no botão salvar. 4. O sistema valida os dados informados. 5. O sistema exibe a mensagem “Setor salvo com sucesso”. <p>Fluxo Alternativo:</p> <ol style="list-style-type: none"> 1. (Passo 4) Validação das informações 1.1 Se os dados cadastrais obrigatórios não forem preenchidos é exibida a mensagem “Preencha este campo”.

Tabela 22 – Requisito funcional RF15

Descrição:	O sistema deverá permitir aos usuários excluírem os setores.
Atores:	Administrador
Pré-condições:	- Estar autenticado no sistema. - Ter setor cadastrado no sistema.
Pós-condições:	- Setor excluído.
Fluxo de Eventos	
Fluxo Básico:	<ol style="list-style-type: none"> 1. Os atores informam ao sistema que desejam excluir um setor. 2. O sistema envia a mensagem “Deseja deletar o setor?”. 3. Os atores clicam em “Ok”. 4. O sistema deleta o setor cadastrado.
Fluxo Alternativo:	<ol style="list-style-type: none"> 1. (Passo 2) Mensagem do sistema <p>1.1 Se o usuário clicar em cancelar, o sistema retorna a listar aos setores cadastrados.</p>

Tabela 23 – Requisito funcional RF16

Descrição:	O sistema deverá listar os setores cadastrados.
Atores:	Administrador
Pré-condições:	- Estar autenticado no sistema. - Ter setor cadastrado no sistema.
Pós-condições:	- Setor excluído.
Fluxo de Eventos	
Fluxo Básico:	<ol style="list-style-type: none"> 1. Os atores clicam na área relacionada aos setores. 2. O sistema exibe os setores cadastrados.
Fluxo Alternativo:	<ol style="list-style-type: none"> 1. (Passo 2) Exibição dos setores <p>1.1 Se não tiver setores cadastrados, o sistema irá exibir a mensagem “Nenhum setor encontrado”.</p>

Tabela 24 – Requisito funcional RF17

Descrição:	O sistema deverá permitir o cadastro de perguntas relacionadas aos setores.
Atores:	Administrador
Pré-condições:	- Estar autenticado no sistema. - Ter setor cadastrado no sistema.
Pós-condições:	- Pergunta cadastrada no sistema.
Fluxo de Eventos	
Fluxo Básico:	<ol style="list-style-type: none"> 1. Os atores informam ao sistema que desejam cadastrar perguntas relacionadas ao setores. 2. Os atores informam os dados obrigatórios. 2.1 Pergunta, Resposta. 3. Os atores clicam no botão salvar. 4. O sistema valida os dados informados. 5. O sistema exibe a mensagem “Pergunta salva com sucesso”.
Fluxo Alternativo:	<ol style="list-style-type: none"> 1. (Passo 4) Validação das informações 1.1 Se os dados cadastrais obrigatórios não forem preenchidos é exibida a mensagem “Preencha este campo”.

Tabela 25 – Requisito funcional RF18

Descrição:	O sistema deverá permitir aos usuários editarem as perguntas relacionadas aos setores.
Atores:	Administrador
Pré-condições:	- Estar autenticado no sistema. - Ter setor cadastrado no sistema. - Ter pergunta cadastrada.
Pós-condições:	- Pergunta alterada no sistema.
Fluxo de Eventos	
Fluxo Básico:	<ol style="list-style-type: none"> 1. Os atores informam ao sistema que desejam alterar uma pergunta. 2. Os atores informam os dados obrigatórios. 2.1 Pergunta, Resposta. 3. Os atores clicam no botão salvar. 4. O sistema valida os dados informados. 5. O sistema exibe a mensagem “Pergunta salva com sucesso”.
Fluxo Alternativo:	<ol style="list-style-type: none"> 1. (Passo 4) Validação das informações 1.1 Se os dados cadastrais obrigatórios não forem preenchidos é exibida a mensagem “Preencha este campo”.

Tabela 26 – Requisito funcional RF19

Descrição:	O sistema deverá permitir os usuários excluírem as perguntas relacionadas aos setores.
Atores:	Administrador
Pré-condições:	<ul style="list-style-type: none"> - Estar autenticado no sistema. - Ter setor cadastrado no sistema. - Ter pergunta cadastrada.
Pós-condições:	- Pergunta excluída.
Fluxo de Eventos	<p>Fluxo Básico:</p> <ol style="list-style-type: none"> 1. Os atores informam ao sistema que desejam excluir uma pergunta. 2. O sistema envia a mensagem “Deseja deletar a pergunta?”. 3. Os atores clicam em “Ok”. 4. O sistema deleta a pergunta cadastrada. <p>Fluxo Alternativo:</p> <ol style="list-style-type: none"> 1. (Passo 2) Mensagem do sistema 1.1 Se o usuário clicar em cancelar, o sistema retorna a listar as perguntas cadastradas.

Tabela 27 – Requisito funcional RF20

Descrição:	O sistema deverá permitir aos usuários cadastrarem os cursos do CPAN.
Atores:	Administrador
Pré-condições:	<ul style="list-style-type: none"> - Estar autenticado no sistema.
Pós-condições:	- Curso cadastrado.
Fluxo de Eventos	<p>Fluxo Básico:</p> <ol style="list-style-type: none"> 1. Os atores informam ao sistema que desejam cadastrar um curso. 2. Os atores informam os dados obrigatórios. 2.1 Nome, Modalidade, Descrição do curso, Link da página. 3. Os atores clicam no botão salvar. 4. O sistema valida os dados informados. 5. O sistema exibe a mensagem “Curso salvo com sucesso”. <p>Fluxo Alternativo:</p> <ol style="list-style-type: none"> 1. (Passo 4) Validação das informações 1.1 Se os dados cadastrais obrigatórios não forem preenchidos, é exibida a mensagem “Preencha este campo”.

Tabela 28 – Requisito funcional RF21

Descrição:	O sistema deverá permitir aos usuários editarem os cursos do CPAN.
Atores:	Administrador
Pré-condições:	- Estar autenticado no sistema. - Ter curso cadastrado no sistema.
Pós-condições:	- Curso alterado no sistema.
Fluxo de Eventos	
Fluxo Básico:	<ol style="list-style-type: none"> 1. Os atores informam ao sistema que desejam alterar um curso. 2. Os atores informam os dados obrigatórios. 2.1 Nome, Modalidade, Descrição do curso, Link da página. 3. Os atores clicam no botão salvar. 4. O sistema valida os dados informados. 5. O sistema exibe a mensagem “Curso salvo com sucesso”.
Fluxo Alternativo:	<ol style="list-style-type: none"> 1. (Passo 4) Validação das informações 1.1 Se os dados cadastrais obrigatórios não forem preenchidos, é exibida a mensagem “Preencha este campo”.

Tabela 29 – Requisito funcional RF22

Descrição:	O sistema deverá permitir aos usuários excluírem os cursos do CPAN.
Atores:	Administrador
Pré-condições:	- Estar autenticado no sistema. - Ter curso cadastrado no sistema.
Pós-condições:	- Curso excluído.
Fluxo de Eventos	
Fluxo Básico:	<ol style="list-style-type: none"> 1. Os atores informam ao sistema que desejam excluir um curso. 2. O sistema envia a mensagem “Deseja deletar o curso?”. 3. Os atores clicam em “Ok”. 4. O sistema deleta o curso cadastrado.
Fluxo Alternativo:	<ol style="list-style-type: none"> 1. (Passo 2) Mensagem do sistema 1.1 Se o usuário clicar em cancelar, o sistema retorna a listar os cursos cadastrados.

Tabela 30 – Requisito funcional RF23

Descrição:	O sistema deverá listar os cursos do CPAN.
Atores:	Administrador
Pré-condições:	- Ter curso cadastrado.
Pós-condições:	- Cursos listados.
Fluxo de Eventos	
Fluxo Básico:	<ol style="list-style-type: none"> 1. Os atores clicam na área relacionada aos cursos. 2. O sistema exibe os cursos cadastrados.
Fluxo Alternativo:	<ol style="list-style-type: none"> 1. (Passo 2) Exibição dos cursos 1.1 Se não tiver cursos cadastrados, o sistema irá exibir a mensagem “Nenhum curso encontrado”.

Tabela 31 – Requisito funcional RF24

Descrição:	O sistema deverá permitir o cadastro das formas de ingresso nos cursos do CPAN.
Atores:	Administrador
Pré-condições:	- Estar autenticado no sistema.
Pós-condições:	- Formas de ingresso cadastrada
Fluxo de Eventos	
Fluxo Básico:	<ol style="list-style-type: none"> 1. Os atores clicam no menu início. 2. O sistema exibe a opção para cadastro. 3. Os atores preenchem a descrição. 4. Os atores clicam no botão salvar. 5. O sistema valida os dados informados. 6. O sistema exibe a mensagem “Salvo com sucesso”.
Fluxo Alternativo:	<ol style="list-style-type: none"> 1. (Passo 5) Validação das informações 1.1 Se os dados cadastrais obrigatórios não forem preenchidos, é exibida a mensagem “Preencha este campo”.

Tabela 32 – Requisito funcional RF25

Descrição:	O <i>bot</i> deverá informar as três últimas notícias postadas no site do CPAN.
Atores:	Usuário
Pré-condições:	<ul style="list-style-type: none"> - Estar logado no facebook messenger. - Solicitar as notícias ao <i>bot</i>.
Pós-condições:	- Notícia exibida ao usuário.
Fluxo de Eventos	
Fluxo Básico:	<ol style="list-style-type: none"> 1. O usuário solicita as notícias por meio da linguagem natural ou por meio das opções do menu. 2. O <i>bot</i> retornará as três últimas notícias ao usuário.
Fluxo Alternativo:	<ol style="list-style-type: none"> 1. Caso ocorra um problema ao ser solicitada, a notícia é exibida a mensagem “Não foi possível buscar as notícias”.

Tabela 33 – Requisito funcional RF26

Descrição:	O <i>bot</i> deverá mostrar as informações referente aos setores do CPAN.
Atores:	Usuário
Pré-condições:	- Estar logado no facebook messenger. - Solicitar os setores do cpan ao <i>bot</i> .
Pós-condições:	- Informações dos setores exibidas ao usuário.
Fluxo de Eventos	
Fluxo Básico:	<ol style="list-style-type: none"> 1. O usuário solicita as informações dos setores por meio da linguagem natural ou por meio das opções do menu. 2. O <i>bot</i> exibe a mensagem “Qual setor você deseja?” e mostra as opções disponíveis. 3. O usuário seleciona a opção desejada. 4. O <i>bot</i> retorna as informações do setor.
Fluxo Alternativo:	<ol style="list-style-type: none"> 1. Caso ocorra um problema ao ser selecionado o setor é exibida a mensagem “Não foi possível buscar o setor”.

Tabela 34 – Requisito funcional RF27

Descrição:	O <i>bot</i> deverá mostrar as dúvidas relacionadas aos setores do CPAN.
Atores:	Usuário
Pré-condições:	- Estar logado no facebook messenger. - Solicitar mais informações sobre os setores do cpan.
Pós-condições:	- Informações dos setores exibidas ao usuário.
Fluxo de Eventos	
Fluxo Básico:	<ol style="list-style-type: none"> 1. O usuário solicita as informações dos setores por meio da linguagem natural ou por meio das opções do menu. 2. O <i>bot</i> exibe a mensagem “Qual setor você deseja?” e mostra as opções disponíveis. 3. O usuário seleciona a opção desejada. 4. O <i>bot</i> retorna as informações do setor juntamente com a mensagem “Mais informações”. 5. O usuário seleciona o botão. 6. O <i>bot</i> retorna as dúvidas relacionadas ao setor.
Fluxo Alternativo:	<ol style="list-style-type: none"> 1. Caso ocorra um problema ao ser selecionado o setor é exibida a mensagem “Não foi possível buscar o setor”. 2. Caso ocorra um problema ao ser solicitado mais informações, é exibida a mensagem “Não foi possível buscar mais informações”.

Tabela 35 – Requisito funcional RF28

Descrição:	O <i>bot</i> deverá fornecer os telefones da Secretaria, Coordenações e dos setores do CPAN.
Atores:	Usuário
Pré-condições:	- Estar logado no facebook messenger. - Solicitar os telefones ao <i>bot</i> .
Pós-condições:	- Informações dos setores exibidas ao usuário.
Fluxo de Eventos	
Fluxo Básico:	<ol style="list-style-type: none"> 1. O usuário solicita os telefones por meio da linguagem natural ou por meio das opções do menu. 2. O <i>bot</i> exibe a mensagem “Quais telefones você deseja?” e mostra as opções disponíveis. 3. O usuário seleciona a opção desejada. 4. O <i>bot</i> retorna os telefones referente a opção selecionada.
Fluxo Alternativo:	<ol style="list-style-type: none"> 1. Caso ocorra um problema ao ser solicitado o telefone, é exibida a mensagem “Não foi possível buscar os telefones”.

Tabela 36 – Requisito funcional RF29

Descrição:	O <i>bot</i> deverá mostrar as informações referente aos cursos do CPAN.
Atores:	Usuário
Pré-condições:	- Estar logado no facebook messenger. - Solicitar os cursos do cpan ao <i>bot</i> .
Pós-condições:	- Cursos exibidos ao usuário.
Fluxo de Eventos	
Fluxo Básico:	<ol style="list-style-type: none"> 1. O usuário solicita os cursos por meio da linguagem natural ou por meio das opções do menu. 2. O <i>bot</i> exibe a mensagem “Qual curso você deseja?” e mostra as opções disponíveis. 3. O usuário seleciona a opção desejada. 4. O <i>bot</i> retorna os telefones referente a opção selecionada.
Fluxo Alternativo:	<ol style="list-style-type: none"> 1. Caso ocorra um problema ao ser solicitado um curso, será exibida a mensagem “Não foi possível buscar os cursos”.

Tabela 37 – Requisito funcional RF30

Descrição:	O <i>bot</i> deverá informar sobre as formas de ingresso nos cursos do CPAN.
Atores:	Usuário
Pré-condições:	<ul style="list-style-type: none"> - Estar logado no facebook messenger. - Solicitar as formas de ingresso ao <i>bot</i>.
Pós-condições:	- Formas de Ingresso exibida ao usuário.
Fluxo de Eventos	
Fluxo Básico:	<ol style="list-style-type: none"> 1. O usuário solicita as formas de ingresso por meio da linguagem natural ou por meio das opções do menu. 2. O <i>bot</i> retorna a informação referente a opção selecionada.
Fluxo Alternativo:	<ol style="list-style-type: none"> 1. Caso ocorra um problema ao ser solicitado a forma de ingresso, é exibida a mensagem “Não foi possível buscar pela informação”.

APÊNDICE B – Modelos Criados

B.1 Diagrama de Caso de Uso

Nesta seção será apresentada a modelagem dos casos de usos da aplicação web, no qual serão apresentadas as funcionalidades da aplicação de acordo com os requisitos listados na subseção A.1.1. As ilustrações estão divididas em cinco funcionalidades relacionadas ao usuário, setores, cursos, telefones e notícias.

Figura 31 – Caso de Uso - Usuário

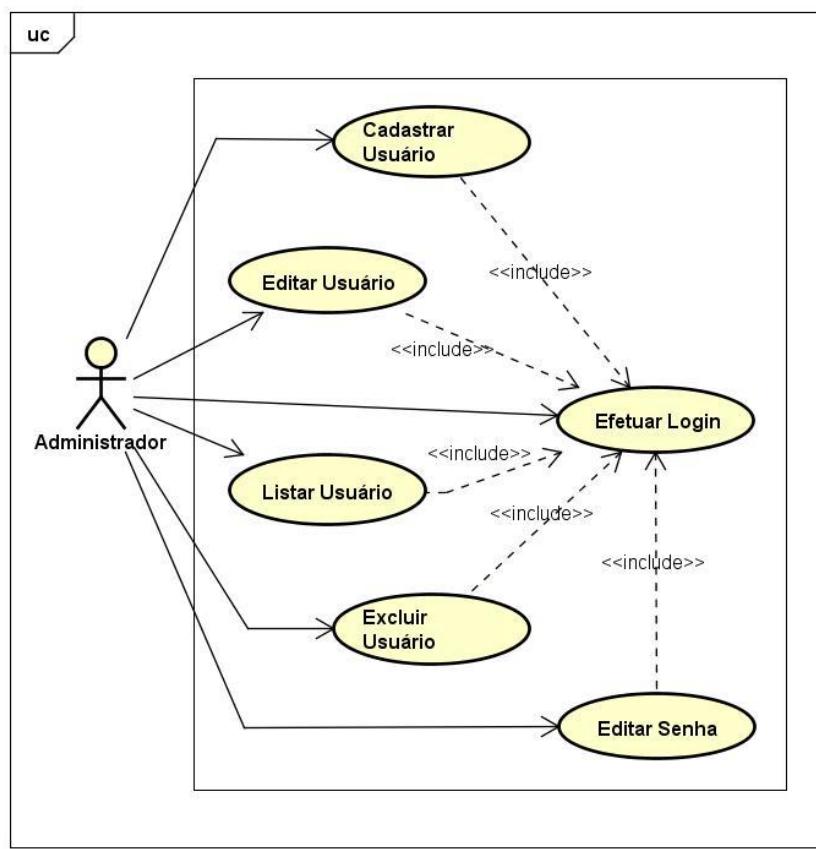
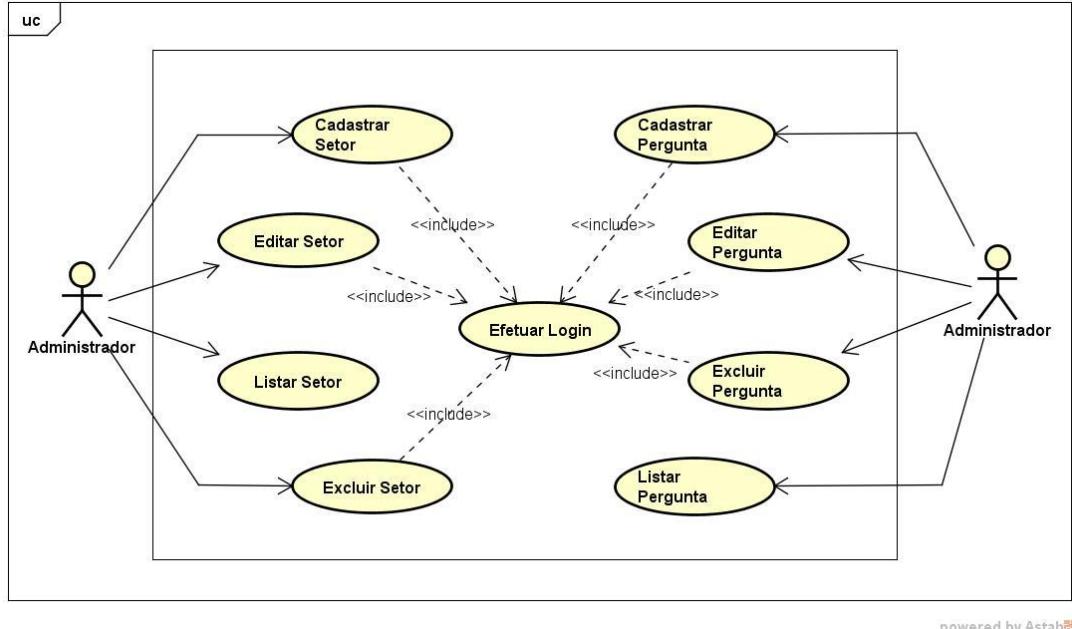
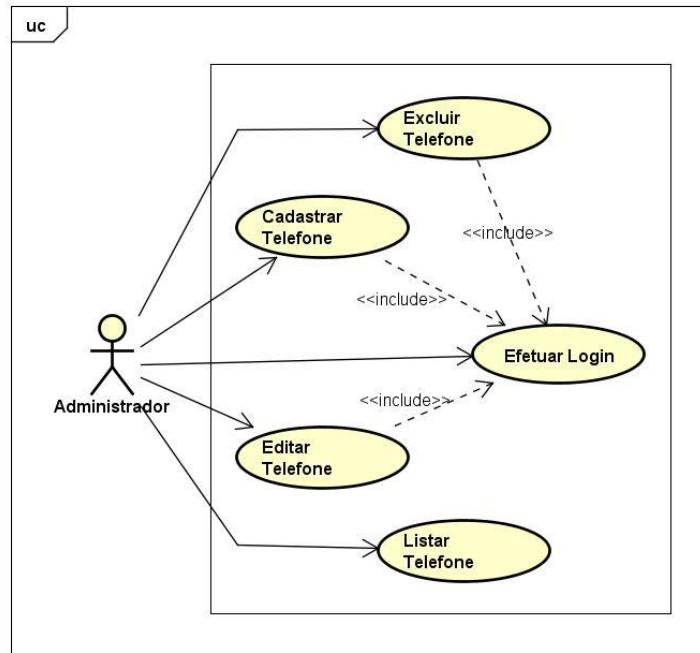


Figura 32 – Caso de Uso - Setor



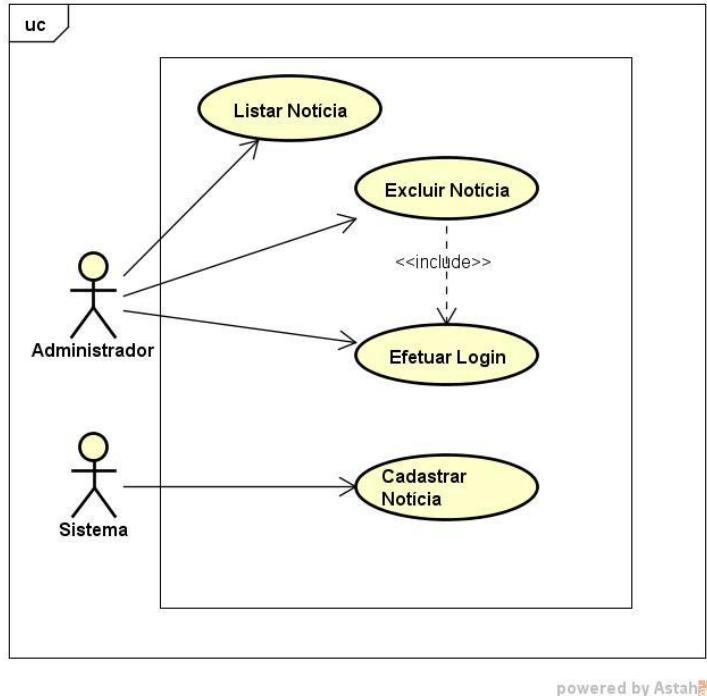
powered by Astah

Figura 33 – Caso de Uso - Telefone



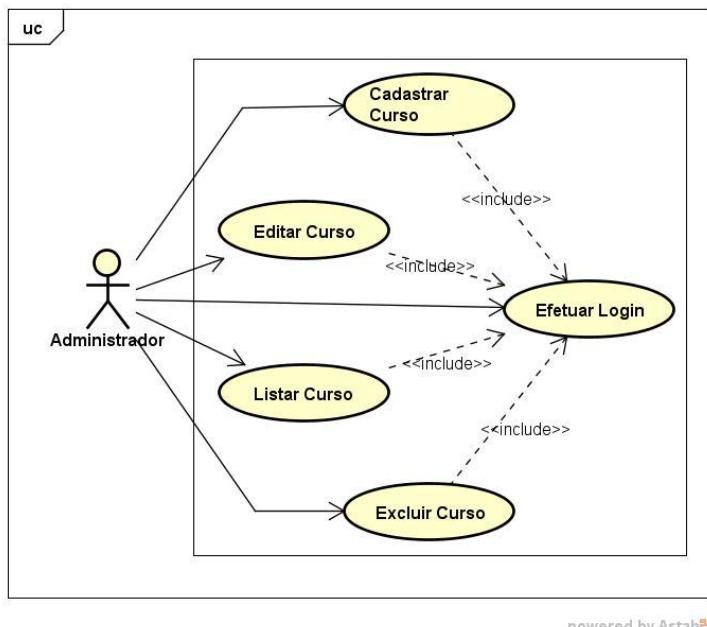
powered by Astah

Figura 34 – Caso de Uso - Notícia



powered by Astah

Figura 35 – Caso de Uso - Curso



powered by Astah

B.2 Modelo do Banco de Dados

A seção seguinte apresenta o modelo de banco de dados criado com o mongoDB, citado na subseção 5.3.1. Estes estão divididos nas coleções de usuários, informações,

notícias, início, departamentos, coordenações e telefones.

Figura 36 – Modelo de Banco de Dados - Parte I

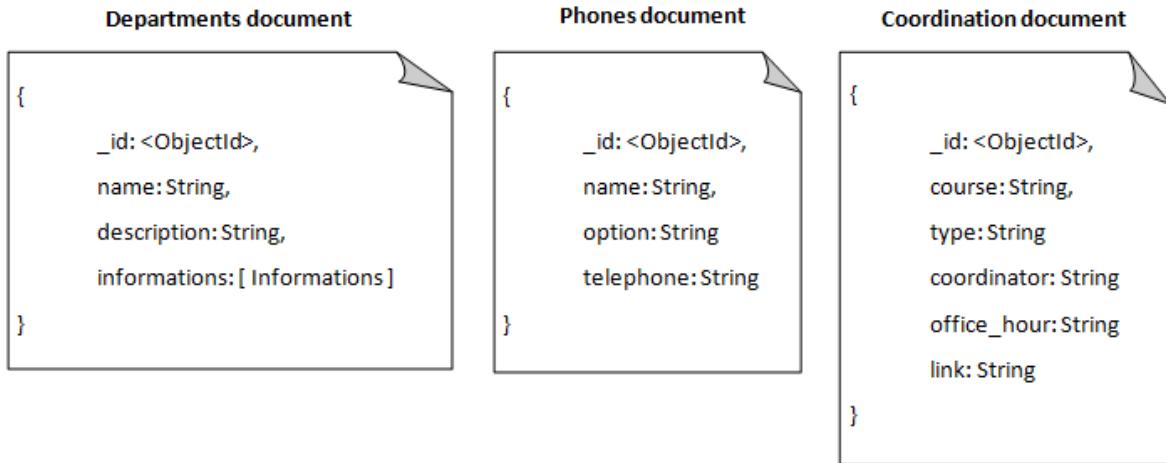
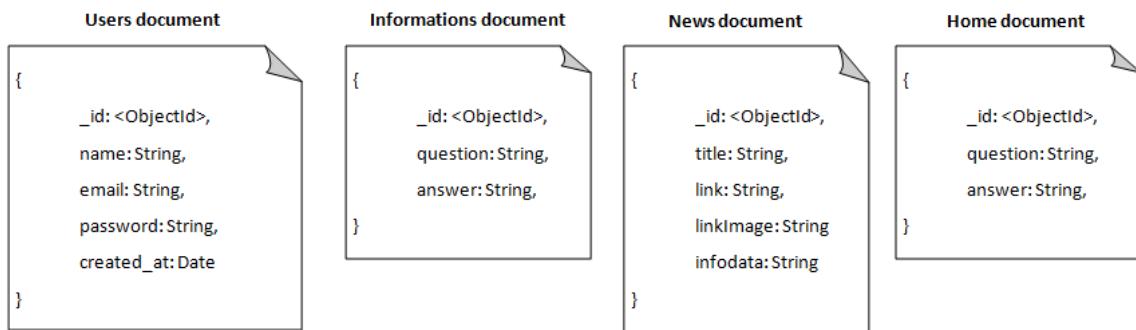


Figura 37 – Modelo de Banco de Dados - Parte II



B.3 Aplicação Web

A seção seguinte apresenta as telas da aplicação web desenvolvida.

Figura 38 – Aplicação Web - Tela de login



Figura 39 – Aplicação Web - Tela de Início

A interface de usuário da CPAN. A barra lateral esquerda é uma navegação com ícones e links: "CPAN" (ícone de casa), "Início" (ícone de casa), "Usuários" (ícone de usuário), "Departamentos" (ícone de edifício), "Coordenação" (ícone de escrivana), "Telefones" (ícone de telefone), "Notícias" (ícone de notícias) e "Sair" (ícone de saída). A área principal é intitulada "Home" e contém o link "Forma de Ingresso" e um botão com um símbolo de mais. O fundo da interface é cinza.

Figura 40 – Aplicação Web - Tela de Usuários

The screenshot shows the CPAN web application interface. On the left is a dark sidebar menu with the following items:

- CPAN
- Inicio
- Usuários
- Departamentos
- Coordenação
- Telefones
- Notícias
- Sair

The main content area is titled "Usuário" and contains a sub-header "Adicionar Usuário" with a blue "+" button. Below this is a table with two rows of user data:

Nome	E-mail	Ações
Westerley Reis	westerreis_@hotmail.com	
Lucineide Rodrigues	lucineide.silva@ufms.br	

Figura 41 – Aplicação Web - Tela de Departamentos

The screenshot shows the CPAN web application interface. On the left is a dark sidebar menu with the following items:

- CPAN
- Inicio
- Usuários
- Departamentos
- Coordenação
- Telefones
- Notícias
- Sair

The main content area is titled "Setor" and contains a sub-header "Adicionar Setor" with a blue "+" button. Below this is a table with three rows of department data:

Nome	Descrição	Ações
NTI	O NTI é o responsável pelo suporte aos assuntos relacionados à informática e à tecnologia da informação. O setor fica localizado na Unidade I - Bloco A. Telefone para contato: (67) 3234-6823	
BIBLIOTECA	A Biblioteca do CPAN "Manoel de Barros" é uma biblioteca setorial que faz parte do Sistema de Bibliotecas da UFMS. A mesma fica localizada na Unidade I - Bloco A. Telefone para contato: (67) 3234-6843	
CPAC	O CPAC lida com o apoio as ações de assistência acadêmica nos Câmpus da UFMS. O setor fica localizado na Unidade I – Bloco F – Sala 7. Telefone para contato: (67) 3234-6892	

Figura 42 – Aplicação Web - Tela de Coordenações

Curso	Coordenador(a)	Ações
SISTEMAS DE INFORMAÇÃO	PROF. MA. LUCINEIDE RODRIGUES DA SILVA	
ADMINISTRAÇÃO	PROF. ME. ANDERSON LUIS DO ESPÍRITO SANTO	
CIÊNCIAS CONTÁBEIS	PROF. ESPEC. MARTA SOUZA FERNANDES	
DIREITO	PROF. ME. MARIA ANGÉLICA BIROLI FERREIRA DA SILVA	
PSICOLOGIA	PROF. ME. ANA MARIA DE VASCONCELOS SILVA	
CIÊNCIAS BIOLÓGICAS	PROF. DRA. ALINE MACKERT DOS SANTOS	

« Previous 1 2 3 Next »

Figura 43 – Aplicação Web - Tela de Telefones

Nome	Telefone	Opção	Ações
ADMINISTRAÇÃO	(67) 3234-6847	COORDENAÇÃO	
CIÊNCIAS BIOLÓGICAS	(67) 3234-6877	COORDENAÇÃO	
CIÊNCIAS CONTÁBEIS	(67) 3234-6877	COORDENAÇÃO	
DIREITO	(67) 3234-6885	COORDENAÇÃO	
GEOGRAFIA	(67) 3234-6877	COORDENAÇÃO	
SISTEMAS DE INFORMAÇÃO	(67) 3234-6877	COORDENAÇÃO	

« Previous 1 2 3 Next »

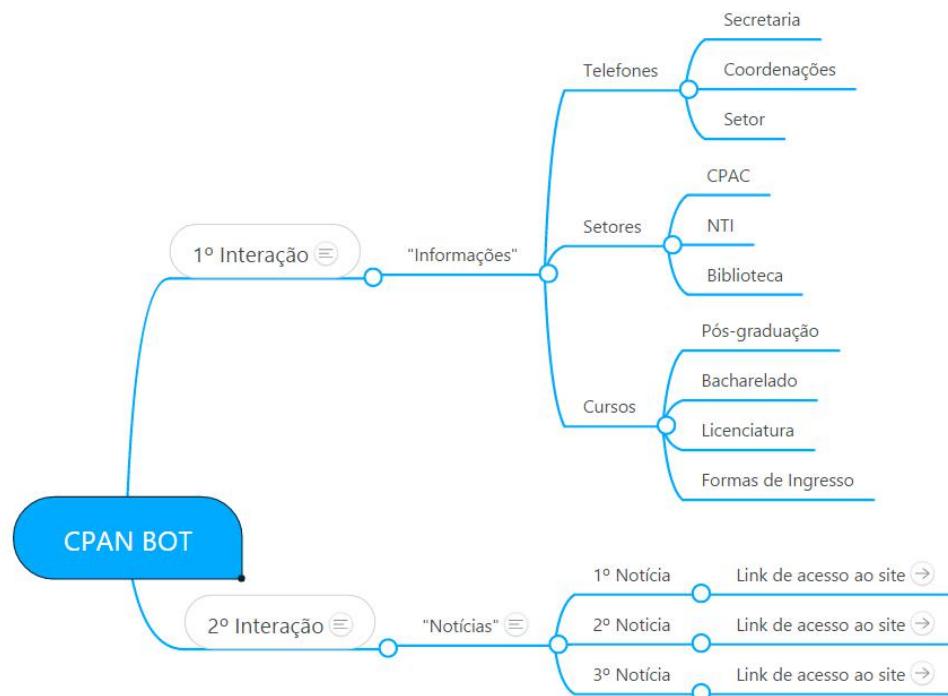
Figura 44 – Aplicação Web - Tela de Notícias

Título	Data	Ações
Campus do Pantanal da UFMS oferece 515 vagas em 12 cursos de Graduação no SISU 2017	19 de janeiro de 2017	
Comunicado CPAC/CPAN	16 de janeiro de 2017	
Lançado Edital de Movimentação Interna para 2017 1	21 de dezembro de 2016	
CPAN recebe novos aparelhos de Ar Condicionado e Bebedouros	27 de janeiro de 2017	
Divulgado edital de renovação de participação nas Ações de Assistência Estudantil para 2017	26 de janeiro de 2017	
Edital de Acompanhamento dos acadêmicos inseridos na Bolsa Permanência e Auxílio Alimentação	3 de fevereiro de 2017	

B.4 Mapa Mental Chatterbot

A seção seguinte apresenta as interações do *chatterbot* com o auxílio de mapas mentais, que segundo Buzan (2006) são um método de armazenar, organizar e priorizar informações, usando palavras-chave e imagens-chave.

Figura 45 – Mapa Mental Chatterbot



B.5 Fluxograma Interação Bot X Usuário

Esta seção apresenta os fluxogramas de interação do *bot* com o usuário, sendo estes compostos pelas notícias, setores, cursos e telefones do CPAN.

Figura 46 – Fluxograma de Interação Bot X Usuário - Parte I

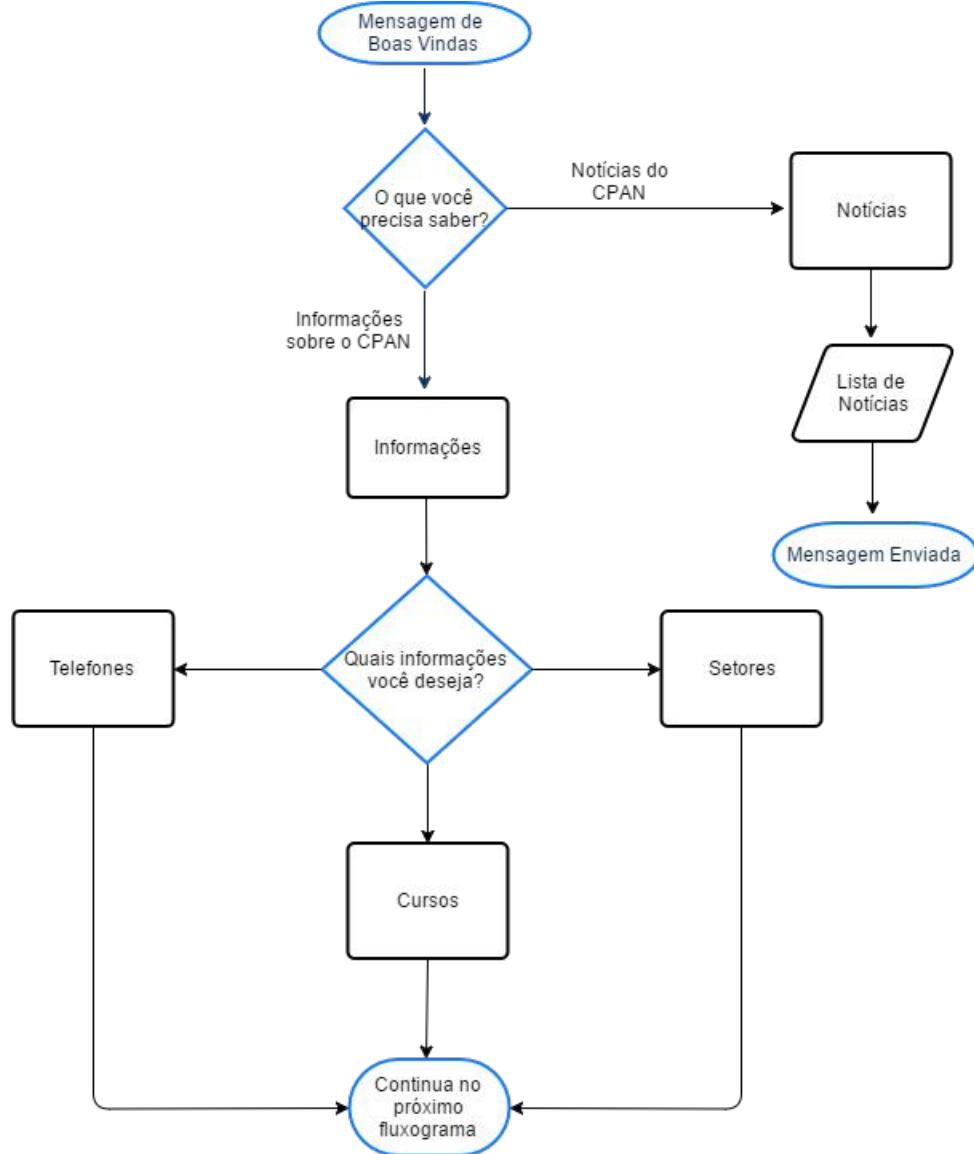


Figura 47 – Fluxograma de Interação Bot X Usuário - Parte II

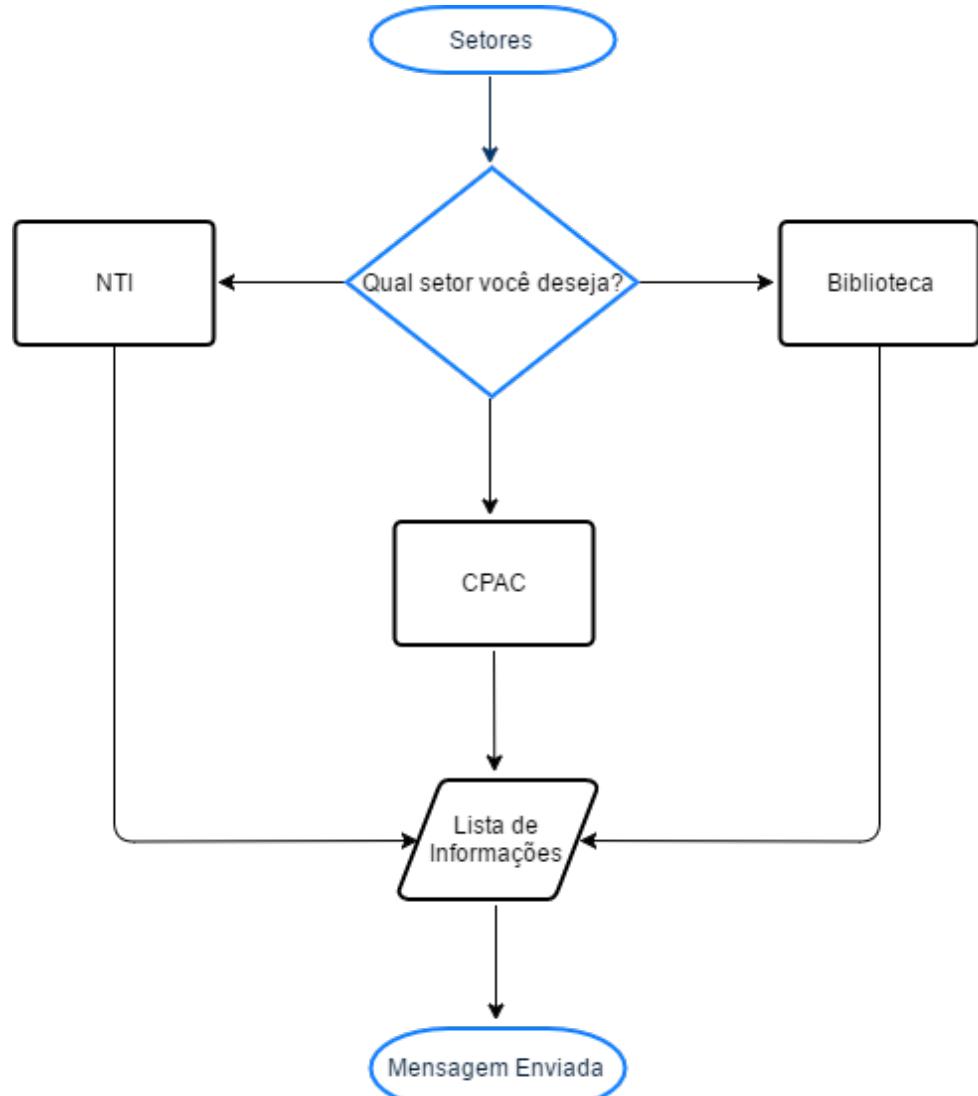


Figura 48 – Fluxograma de Interação Bot X Usuário - Parte III

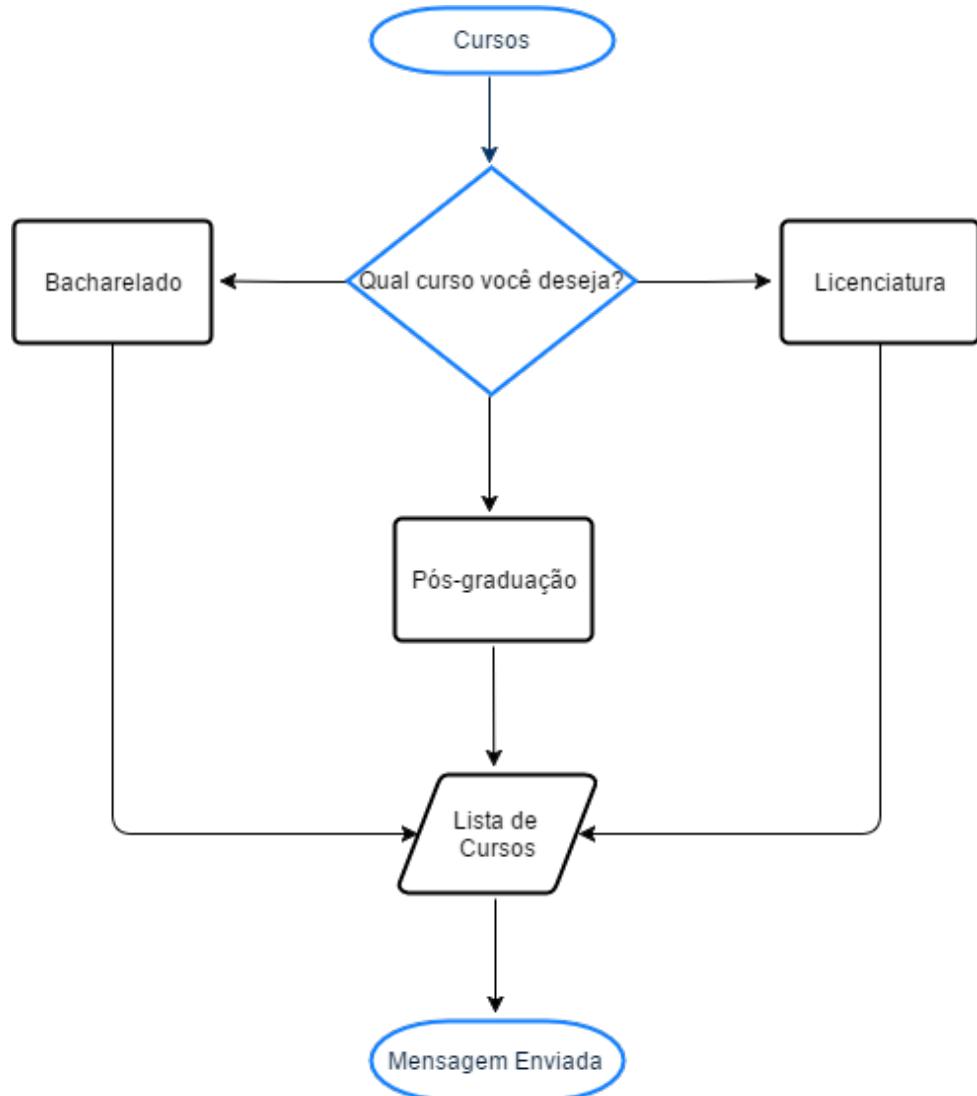
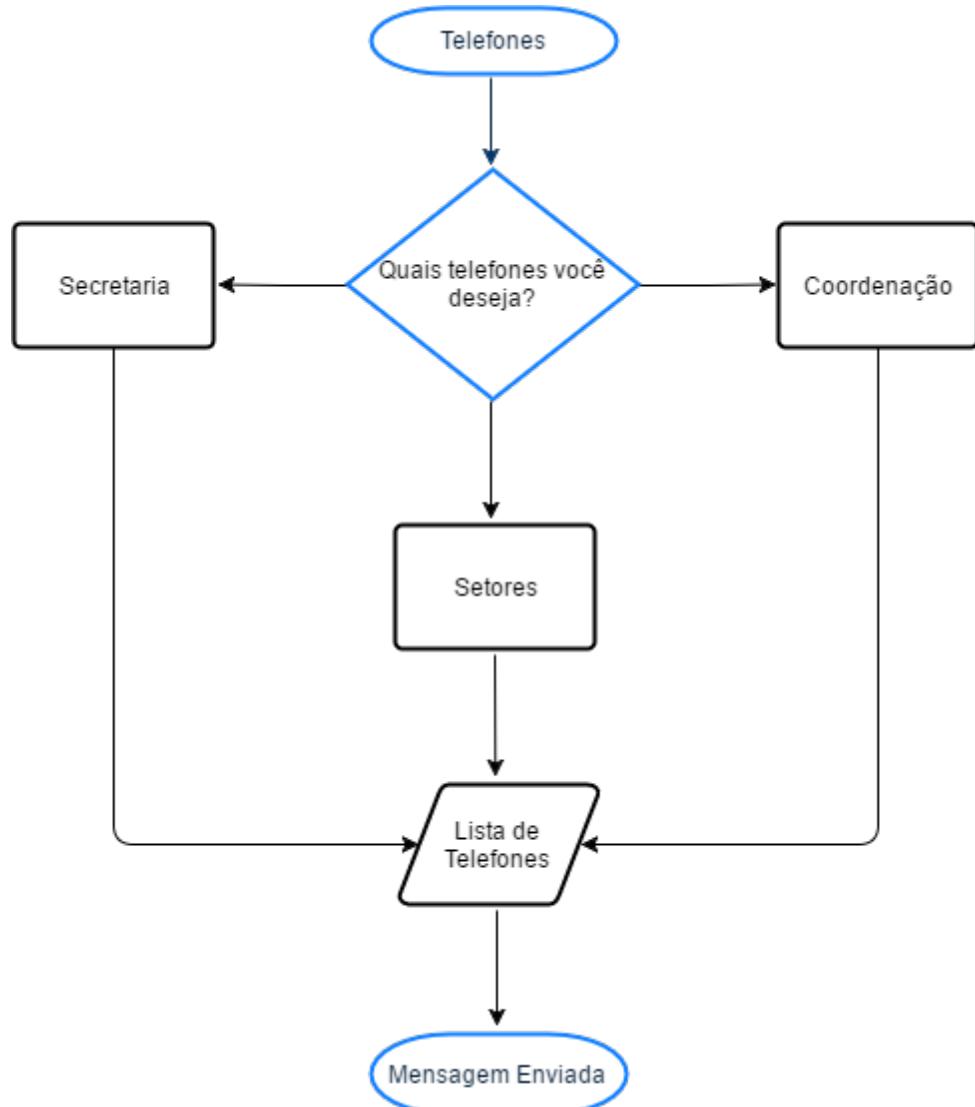


Figura 49 – Fluxograma de Interação Bot X Usuário - Parte IIII



APÊNDICE C – Resultados

C.1 Pesquisa de Satisfação

O objetivo desta pesquisa foi identificar se o atendimento ao público por meio do serviço de *chatterbot* atendeu às dúvidas relacionadas ao Câmpus do Pantanal. No total foram 33 participantes adicionados para interagir com o *bot*. No entanto, o formulário foi respondido por 18 pessoas após a interação com o *bot*. Entretanto, 4 pessoas não conseguiram abrir o serviço pelo aplicativo do Messenger por razões desconhecidas.

Figura 50 – Pergunta 1: Você utilizou o chatterbot? (22 respostas)

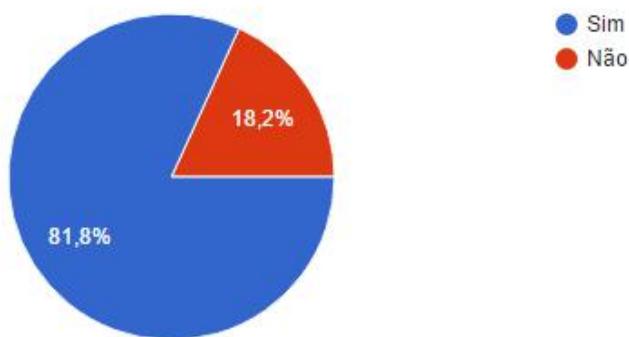


Figura 51 – Pergunta 2: Qual é a sua relação com o Câmpus do Pantanal (18 respostas)

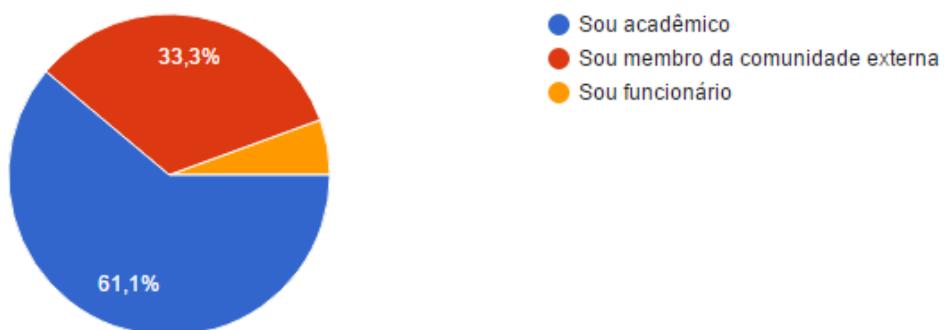


Figura 52 – Pergunta 3: O atendimento por chatterbot proporcionou o acesso rápido a informação disponibilizada pelo CPAN? (18 respostas)

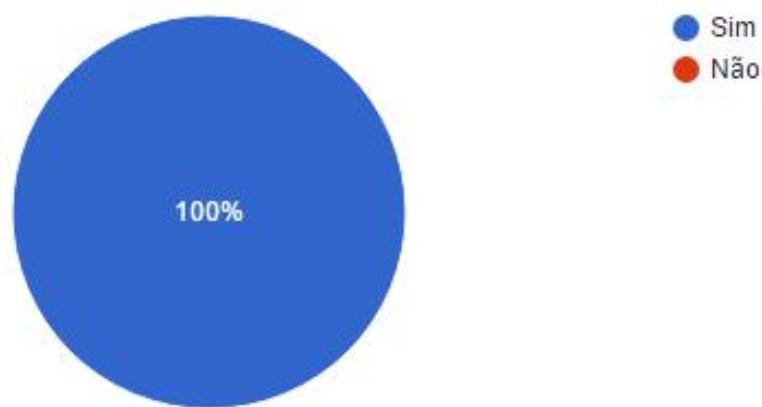
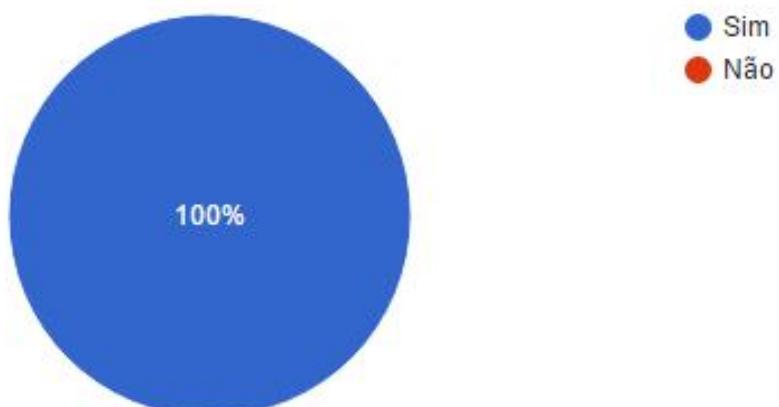


Figura 53 – Pergunta 4: O chatterbot respondeu todas as suas dúvidas? (18 respostas)



Pergunta 5: Quais dúvidas não foram atendidas? (0 respostas)

Não há respostas para esta pergunta.

**Pergunta 6: Quais melhorias você gostaria de sugerir para este serviço?
(4 respostas)**

Resposta 1 - Layout das informações

Resposta 2 - Mais perguntas sobre o câmpus

Resposta 3 - O serviço no geral apresenta bons recursos e procura sanar de forma simples eventuais dúvidas que os usuários possam ter. Entretanto, gostaria de sugerir algumas melhorias:

1. Pude observar que, para algumas frases entradas pelo usuário, o chatterbot não encontra a resposta mais apropriada. Por exemplo, quando o usuário entra a frase "coordenação do curso x". Nesta porção da aplicação, ao invés da mesma direcionar o usuário para o curso pretendido, ela mostra uma resposta referente ao ingresso acadêmico.
2. Quando o usuário está buscando por um curso em específico, por exemplo "Educação Física", o bot retorna todos os cursos que compartilham a mesma modalidade, por exemplo: bacharelado ou licenciatura, ao invés de retornar somente o curso que foi solicitado. Este ponto não chega a ser um problema crítico, tendo em vista que o usuário pode, aproveitando esta "funcionalidade", conhecer mais sobre os outros cursos oferecidos pela instituição.
3. Quando o bot não entende o que o usuário está pretendendo, seria uma melhoria se o mesmo retornasse a mensagem de "não entendi" de maneiras variadas, proporcionando uma interação mais natural e parecida com a humana.
4. Ao pedir um telefone, as seguintes categorias aparecem: Secretaria, coordenação e setores. Entretanto, as mesmas aparecem de forma clicável e, ao digitar uma das três opções, o bot não entende a mensagem, limitando o usuário a clicar em uma das três opções apresentadas. Este problema não impacta em nada o funcionamento da aplicação, porém, poderia adicionar valor ao serviço, caso a correção deste fosse aplicada.

Finalmente, a interação com o chatterbot CPAN foi realizada de maneira rápida e eficiente. A navegação na aplicação também é um ponto a favor, possibilitando que os usuários accessem um menu persistente, pelo qual algumas opções são disponibilizadas, como: acessar o site do CPAN e o menu inicial. De forma moderna e intuitiva, esta aplicação pode ser uma ferramenta importantíssima na disseminação de conteúdo da instituição e um excelente canal de comunicação.

Resposta 4 - Editais

Figura 54 – Pergunta 7: Qual nota você daria para este serviço? Obs.: Considerando 1 como nota mínima e 5 como nota máxima de satisfação (18 respostas)

