



Universidade Federal
do Rio de Janeiro

Escola Politécnica

PIPA BOT: UMA SOLUÇÃO CHATBOT PARA O PROJETO PIPA UFRJ

Lucas Santos de Paula

Projeto de Graduação apresentado ao Curso de Engenharia de Computação e Informação da Escola Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro.

Orientadores: Guilherme Horta Travassos
Taísa Guidini Gonçalves

Rio de Janeiro
Agosto de 2019

PIPA BOT: UMA SOLUÇÃO CHATBOT PARA O PROJETO PIPA UFRJ

Lucas Santos de Paula

PROJETO DE GRADUAÇÃO SUBMETIDO AO CORPO DOCENTE DO CURSO DE ENGENHARIA DE COMPUTAÇÃO E INFORMAÇÃO DA ESCOLA POLITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO DE COMPUTAÇÃO.

Examinado por:

Prof. Nome do Primeiro Examinador Sobrenome, D.Sc.

Prof. Nome do Segundo Examinador Sobrenome, Ph.D.

Prof. Nome do Terceiro Examinador Sobrenome, D.Sc.

Prof. Nome do Quarto Examinador Sobrenome, Ph.D.

Prof. Nome do Quinto Examinador Sobrenome, Ph.D.

RIO DE JANEIRO, RJ – BRASIL

AGOSTO DE 2019

de Paula, Lucas Santos

PIPA Bot: uma solução chatbot para o projeto PIPA
UFRJ/Lucas Santos de Paula. – Rio de Janeiro: UFRJ/
Escola Politécnica, 2019.

XIII, 25 p.: il.; 29, 7cm.

Orientadores: Guilherme Horta Travassos

Taísa Guidini Gonçalves

Projeto de Graduação – UFRJ/ Escola Politécnica/
Curso de Engenharia de Computação e Informação, 2019.

Referências Bibliográficas: p. 24 – 24.

1. Chatbot. 2. Lean Inception. 3. Engenharia
de Software. I. Travassos, Guilherme Horta *et al.* II.
Universidade Federal do Rio de Janeiro, Escola Politécnica,
Curso de Engenharia de Computação e Informação. III.
Título.

*A Paulo Cardoso, Hercília
Cardoso e Maria Alves (in
memoriam).*

Agradecimentos

Gostaria de agradecer a todos.

Resumo do Projeto de Graduação apresentado à Escola Politécnica/ UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro de Computação.

PIPA BOT: UMA SOLUÇÃO CHATBOT PARA O PROJETO PIPA UFRJ

Lucas Santos de Paula

Agosto/2019

Orientadores: Guilherme Horta Travassos

Taísa Guidini Gonçalves

Curso: Engenharia de Computação e Informação

Chatbots são sistemas de software que fazem uso de processamento de linguagem natural para estabelecer diálogos com pessoas como se fossem humanos. Podem atuar como um canal facilitador para a comunicação entre humanos e entidades como empresas, projetos sociais e de pesquisa, hospitais, etc. Esse trabalho apresenta o desenvolvimento do PipaBot, o *chatbot* do Projeto Infância e Poluentes Ambientais (PIPA-UFRJ), que deverá atender cerca de duas mil mães participantes.

O PipaBot foi desenvolvido em duas plataformas: através do aplicativo Messenger do Facebook e também no *website* do PIPA. Através dele é possível obter informações sobre o projeto e seus objetivos, e também realizar uma identificação como paciente do projeto, que libera acesso a informações como consultas e resultados de exames médicos.

A PipaBot foi validado segundo o modelo de aceitação da tecnologia (TAM - *Technological Acceptance Model*), com a avaliação de 13 usuários sendo divididos em dois grupos principais: profissionais da saúde (pesquisadores do projeto PIPA-UFRJ), e profissionais de computação (pesquisadores da COPPE-UFRJ) tendo resultados satisfatórios os quais serão apresentados com mais detalhes neste trabalho.

Abstract of Undergraduate Project presented to POLI/UFRJ as a partial fulfillment of the requirements for the degree of Engineer.

PIPA BOT: A CHATBOT SOLUTION FOR PIPA UFRJ PROJECT

Lucas Santos de Paula

August/2019

Advisors: Guilherme Horta Travassos

Taísa Guidini Gonçalves

Course: Computer Engineering

In this work, we present ...

Sumário

Lista de Figuras	x
Lista de Tabelas	xi
Lista de Símbolos	xii
Lista de Abreviaturas	xiii
1 Introdução	1
1.1 Motivação	1
1.2 Objetivos	2
1.3 Metodologia	3
1.4 Organização do Trabalho	3
2 Revisão da Literatura	5
2.1 NLP - Processamento de Linguagem Natural	5
2.2 Chatbot baseado em regras	6
2.3 Chatbots com inteligência artificial	6
2.4 Uso do chatbot	6
2.5 Uso do chatbot na área médica	7
3 Tecnologias utilizadas na construção de chatbots	8
3.1 Ferramentas e Frameworks	8
3.1.1 Chatfuel	8
3.1.2 Botpress	8
3.2 Canais	10
3.2.1 Facebook Messenger	10
3.2.2 Telegram	11
3.2.3 Slack	11
3.3 Provedores de inteligência artificial e NLP	11
3.3.1 Wit.ai	11
3.3.2 IBM Watson	12

3.3.3	LUIS.ai	12
4	Desenvolvimento do chatbot	13
4.1	Motivação	13
4.1.1	Contexto: o projeto PIPA UFRJ	13
4.1.2	Tecnologias utilizadas no PIPABOT	13
4.2	Ambientes de desenvolvimento	15
4.3	A solução	16
4.3.1	Arquitetura	16
4.3.2	Modelos	17
4.3.3	MVP 1	19
4.3.4	MVP 2	19
4.3.5	Avaliação do produto	19
5	Conclusão	23
5.1	Contribuições	23
5.2	Limitações	23
5.3	Trabalhos futuros	23
5.4	Considerações finais	23
	Referências Bibliográficas	24
A	Algumas Demonstrações	25

Lista de Figuras

1.1	Lean Startup	3
2.1	Interpretação de uma frase com processamento de linguagem natural	5
3.1	<i>Choice</i> - Componente de respostas através de opções de múltipla escolha	9
3.2	<i>HITL</i> - Interface de troca de mensagens no painel do Botpress	10
3.3	<i>Actions</i> - Bloco com uma função javascript que é executada após o envio da mensagem.	10
4.1	Arquitetura do NodeJS	14
4.2	PipaBot - Organização do versionamento do código	15
4.3	Arquitetura do PipaBot	16
4.4	Casos de Uso do Usuário	18
4.5	Casos de Uso do Canal (Messenger e Web)	18
4.6	Casos de Uso do Bot	18
4.7	Organização do Paradigma GQM aplicado ao PipaBot	20

Lista de Tabelas

4.1	GQM - Objetivo 1	20
4.2	GQM - Objetivo 2	20
4.3	GQM - Objetivo 3	21
4.4	GQM - Objetivo 4	21
4.5	GQM - Questões para avaliação de grau de concordância	21
4.6	GQM - Métricas	22

Lista de Símbolos

\emptyset	Conjunto vazio, p. 1
\mathbb{R}	Conjunto dos números reais, p. 1

Lista de Abreviaturas

COPPE	Instituto Alberto Luiz Coimbra de Pós-graduação e Pesquisa de Engenharia, p. 1
POLI-UFRJ	Escola Politécnica da Universidade Federal do Rio de Janeiro, p. 1

Capítulo 1

Introdução

1.1 Motivação

As redes sociais tem o papel de conectar pessoas, estreitando assim os laços humanos. Graças a elas podemos nos comunicar com pessoas que estão do outro lado do globo em tempo real. O sucesso das redes sociais é tão grande, que apenas no Brasil, temos cerca de 130 milhões de usuários mensais só no Facebook. Segundo dados da pesquisa 143^a Pesquisa CNT/MDA¹, 82% dos entrevistados afirmam que fazem uso de seus *smartphones* para acesso à redes sociais. Além disso, os resultados da 29^a Pesquisa Anual de Administração e Uso de Tecnologia da Informação nas Empresas, realizada pela FGV, mostra que o Brasil já ultrapassou a marca de 1 aparelho *smartphones* por habitante².

O tamanho sucesso das redes sociais e dos *smartphones*, permitiu que os antigos sistemas de informação, que, em algumas décadas atrás necessitavam de grandes computadores para serem executados, agora são acessíveis por um dispositivo de mão. Jogos, bancos, redes sociais, e até mesmo redes de *fast-food* desenvolveram seus aplicativos e marcaram presença nos *smartphones* da população. Estima-se que apenas na Play Store (loja de aplicativos do Google), cerca de 2,6 milhões de aplicativos estejam disponíveis para *download*³.

Contudo, estudos mais recentes indicam que apenas cinco aplicativos são responsáveis por cerca de 84% do uso de aplicativos não nativos do smartphone⁴. Esses aplicativos variam de pessoa para pessoa, mas em geral, incluem aplicativos de redes sociais ou de mensagens instantâneas. Esse cenário justifica o aumento no número de chatbots disponíveis em plataformas de mensagens como Messenger nos

¹<https://www.cnt.org.br/agencia-cnt/confira-resultados-pesquisa-cnt-md>

²<https://eaesp.fgv.br/sites/eaesp.fgv.br/files/pesti2018gvciappt.pdf>

³<https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>

⁴<https://techcrunch.com/2015/06/22/consumers-spend-85-of-time-on-smartphones-in-apps-but-only-5-apps-see-heavy-use/>

últimos anos.

Chatbots são sistemas computacionais que simulam uma conversas com pessoas e tornam a interação homem-máquina mais natural[1]. São comumente chamados de assistentes virtuais, agentes virtuais ou simplesmente *bots*.

Ao criar um *chatbot* para ser utilizado em canais já consolidados como Messenger, Telegram, Slack e WhatsApp, estamos utilizando uma infraestrutura já consolidada de um aplicativo para executar tarefas que um usuário faria em um outro aplicativo, o que dispensa a instalação e ocupação da memória do aparelho do usuário. Além disso, *chatbots* também podem ser incluídos em *sites* oferecendo aos visitantes um canal rápido de atendimento enquanto navega na internet.

Outro ponto é que o formato de comunicação através de troca de mensagens oferecido pelos *chatbots* é mais humano do que a interação em um aplicativo convencional. O usuário tem uma dúvida e, ao invés de navegar entre menus e telas de um aplicativo, ele simplesmente envia seu questionamento e é prontamente respondido pelo *bot*.

Do ponto de vista de quem oferece a aplicação, ter um *chatbot* ao invés de um aplicativo significa não ter que competir para captar usuários em uma loja de aplicativos, que por sua vez é repleta de outros aplicativos que são potenciais concorrentes. Além disso, o custo de desenvolvimento tende a ser inferior, o que também é uma vantagem. Ao criar um *chatbot*, criamos um único serviço que poderá estar disponível em um ou mais canais de comunicação, utilizando a mesma estrutura. Quando criamos aplicativos, especialmente aqueles que estarão disponíveis em várias plataformas, gastamos tempo pensando na experiência do usuário para dois ou mais sistemas diferentes e, por fim desenvolvendo dois aplicativos com estruturas diferentes.

1.2 Objetivos

O objetivo principal desse trabalho é construir um *chatbot* tendo como base boas práticas de engenharia de software. O *bot* deverá ser disponibilizado em duas plataformas: *web*, para ser incorporado ao Portal do PIPA (site institucional do projeto, onde os pacientes cadastrados podem consultar informações e exames) e *Messenger*. No que diz respeito a funcionalidades, ele deverá ser capaz de dar informações sobre o projeto para o público geral e poderá acessar os usuários cadastrados no portal PIPA, podendo vincular um paciente ao usuário do *bot*. Aos pacientes que realizaram esse vínculo, será permitido obter informações de consultas e resultados de exames através do *chatbot*.

Para que o *chatbot* acesse a base de pacientes do PIPA, deverá ser criado também um novo *endpoint* a API já existente no Portal. Esse *endpoint* será protegido com

uma tecnologia de *token* para garantir que apenas solicitações autorizadas tenham acesso a dados do projeto.

1.3 Metodologia

A metodologia utilizada para o desenvolvimento desse projeto segue a linha "Construir - Medir - Aprender" herdada do *Lean Inception*, cuja finalidade é construir versões incrementais do Mínimo Produto Viável (MVP) a cada ciclo.

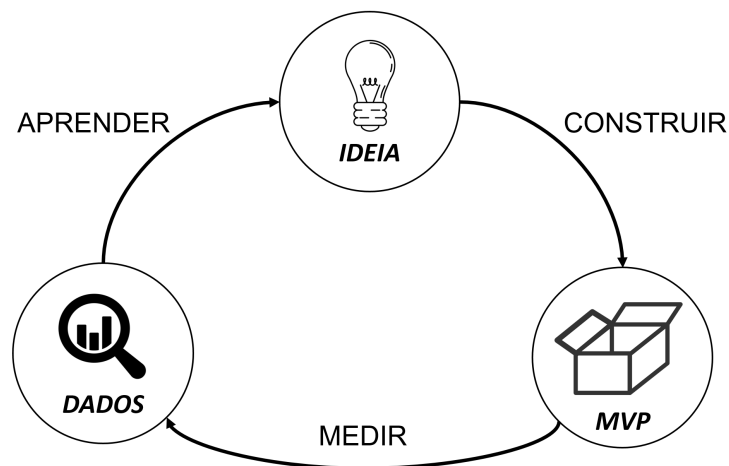


Figura 1.1: Lean Startup

Cada ciclo parte de uma ideia na qual o MVP é concebido (CONSTRUIR). Esse produto é colocado no mundo real, para ser validado pelos *stakeholders* (MEDIR). Com o resultado dessas avaliações é necessário identificar, de acordo com a situação, se a próxima etapa será um incremento, pivotamento ou uma nova concepção do zero (APRENDER).

1.4 Organização do Trabalho

Nos próximos capítulos serão apresentados conceitos, ferramentas e os resultados obtidos com esse trabalho, organizados da seguinte forma:

O capítulo 2 apresentará uma revisão da literatura, trazendo diversos conceitos a respeito de *chatbots*, como seus tipos e também possíveis usos.

O capítulo 3 enumerará uma série de ferramentas, *frameworks* e canais que podem ser utilizados para se construir *chatbots*.

O capítulo 4 apresentará o contexto do projeto PIPA Bot, bem como as escolhas do projeto e suas devidas justificativas, ambientes de desenvolvimento, arquitetura do *chatbot* e as versões do MVP construídas.

O capítulo 5 apresentará a conclusão desse projeto, com o MVP construído no último ciclo de desenvolvimento, limitações encontradas ao longo das etapas e também possíveis melhorias para trabalhos futuros.

Capítulo 2

Revisão da Literatura

2.1 NLP - Processamento de Linguagem Natural

O processamento de linguagem natural é uma vertente específica da Inteligência Artificial que utiliza conhecimentos da língua e de comunicação para melhorar a interação homem-máquina. É possível sintetizar o conceito de NLP como "a habilidade de um computador em processar a mesma linguagem que os humanos usam no cotidiano[1]. Linguística, semântica, teoria da comunicação e processamento de sinais são algumas das áreas que auxiliam no processamento de linguagem natural.

Um sistema conversacional é capaz de detectar *intents* e *entities*. O intent é o objetivo ou propósito da interação do usuário com o chatbot, enquanto uma entity é uma característica que adiciona valor a um intent[2] como demonstrada na figura 2.1.

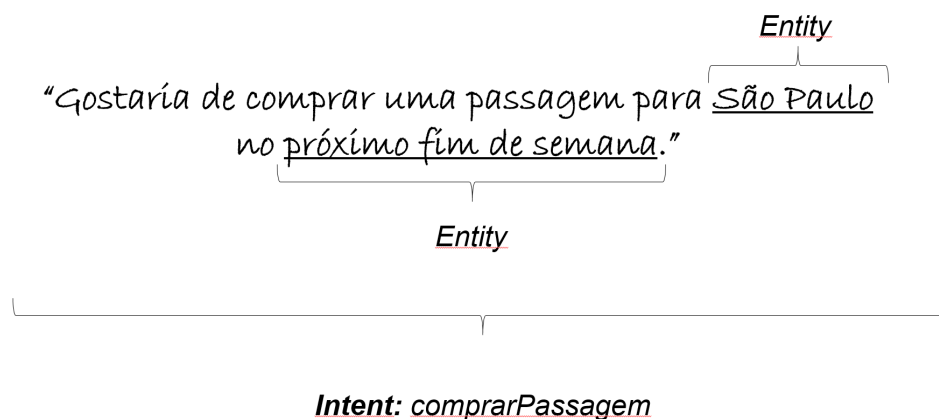


Figura 2.1: Interpretação de uma frase com processamento de linguagem natural

A frase tem como objetivo (*intent*) comprar uma passagem. Na frase, é possível identificar palavras que agregam para o objetivo de compra de passagem, como o

destino (São Paulo) e data de partida (próximo fim de semana).

2.2 Chatbot baseado em regras

Chatbots baseados em regras são um tipo particular de *bots* onde os fluxos de conversa são bem definidos e com opções mais limitadas de interação.

A LEGO¹, fabricante de brinquedos infantis, possui um chatbot baseado em regras disponível na sua página do Messenger que é responsável por fornecer sugestões de presentes aos usuários baseado em um filtro de busca. O usuário fornece informações como localização, idade da criança e orçamento disponível e o *bot* exibe as opções disponíveis.

2.3 Chatbots com inteligência artificial

Chatbots com inteligência artificial são aqueles que conseguem extrair informações de *intents* e *entities* de um texto enviado pelo usuário através de algoritmos de *machine learning*, e também tomar decisões com base nos dados extraídos e o contexto da conversa.

Em geral, os chatbots com inteligência artificial são treinados previamente para conhecer uma série de respostas para os *intents* que ele irá atender. Sendo assim, ele não será capaz de responder a situações no qual ele não seja treinado.

2.4 Uso do chatbot

A aplicabilidade dos chatbots é, em geral, a mesma que qualquer outro sistema computacional, pois trata-se de uma interface de interação que permite execução de comandos de máquina e integração com outros tipos de sistema de software, como bancos de dados, por exemplo.

O avanço no desenvolvimento das plataformas de bots, liderado principalmente pelo Facebook tornou possível a criação de chatbots que resolvem problemas reais da população, como é o caso do PrefeitoBot[3]. O PrefeitoBot é um bot para Messenger que traz informações sobre a prefeitura da cidade do Rio de Janeiro, como secretarias e horários de funcionamento. Além disso, também oferece um canal onde os usuários podem reportar ocorrências na cidade, como tombamento de árvores, alagamentos, poluição sonora, entre outras situações que o canal 1746 da cidade do Rio de Janeiro oferece atendimento.

¹<https://www.messenger.com/t/LEGO>

Chatbots também estão presentes no setor de varejo. As Casas Bahia lançou em 2017 o Bahianinho², o chatbot da empresa responsável pelo atendimento dos consumidores através do *Messenger*. Na época, sua principal função era enviar ofertas da Black Friday para os usuários, de acordo com suas preferências.

No mesmo ano, o Rock in Rio criou o Roque³. O bot do festival interagiu com mais de 77 mil usuários, trocando cerca de 3 milhões de mensagens⁴ ao longo dos 7 dias de evento. Roque tirava dúvidas dos usuários, enviava notícias e prestava suporte aos participantes do festival e também aos que assistiam de casa.

2.5 Uso do chatbot na área médica

Os *chatbots* podem desempenhar um papel estratégico na área médica. Em geral, por desempenhar um atendimento automatizado aos usuários que o utilizam, os *chatbots* podem ser responsáveis por ser um canal disponível 24 horas por dia, podendo atender cada usuário de forma personalizada, oferecendo suporte a diversas situações da área da saúde como marcação de consultas, lembretes de medicação, dúvidas sobre doenças, etc.

Hoje é possível encontrar diversos *chatbots* aplicados a área médica como o *Help-Care* [4], que auxilia no tratamento de doenças crônicas como diabetes, colesterol alto e hipertensão arterial, e o *MediBot* [5], que oferece informações sobre medicamentos.

²<https://www.facebook.com/CasasBahia/>

³<https://take.net/blog/take-notes/case-take-chatbot-no-rock-in-rio/>

⁴<https://take.net/blog/chatbots/cases-de-chatbots-famosos/>

Capítulo 3

Tecnologias utilizadas na construção de chatbots

3.1 Ferramentas e Frameworks

3.1.1 Chatfuel

Chatfuel é o *framework* mais utilizado na criação de *bots*. Através de uma plataforma totalmente online, o serviço permite a criação de *chatbots* através de um painel. Segundo dados da própria companhia, cerca de 46% dos *chatbots* desenvolvidos para o Facebook Messenger¹ são criados com a plataforma.

Com o Chatfuel é possível criar *bots* baseados em regras ou com inteligência baseada em palavras-chave, com treinamento feito diretamente pelo painel de administração.

O plano gratuito permite até 1000 usuários assinantes e exibe uma propaganda no *chatbot* indicando que ele foi desenvolvido com a ferramenta.

3.1.2 Botpress

O Botpress é um *framework open-source* de criação de bots desenvolvido em JavaScript que contém toda a infraestrutura necessária para produzir chatbot sem a necessidade de escrever muito código. Executado com o Node.js e disponível no repositório do NPM, o Botpress é focado na simplicidade e na intuitividade para o desenvolvimento de *chatbots*, oferecendo um painel interativo para criação de fluxos de conversa e um simulador para testes, o que permite a construção de um *chatbot* completo de forma bem mais rápida. Além disso, também é possível adicionar novas funcionalidades através da instalação de módulos.

¹Disponível em <https://chatfuel.com/>

Os módulos do Botpress são componentes que não fazem parte do Core principal do *framework*, mas que podem ser instalados para adicionar novas funcionalidades. Os módulos se dividem em três categorias: módulos de canais, módulos de skills e módulos funcionais.

Os módulos de canais são os componentes que permitem com que o *chatbot* envie e receba mensagens de uma plataforma específica, como Facebook Messenger, Telegram, etc. Para que isso aconteça, o Botpress Core implementa um mecanismo de enfileiramento que processa as mensagens que chegam e que saem, sequencialmente. Caso haja alguma falha nesse processo (um erro de envio, por exemplo) é feita uma nova tentativa de processamento da mensagem, antes de gerar um erro.

Skills são componentes que podem ser incluídos nos fluxos de conversa. Dessa forma, os módulos de skills são aqueles módulos que instalam tais componentes. Um exemplo de módulo de skill, por exemplo, é o **múltipla escolha**.

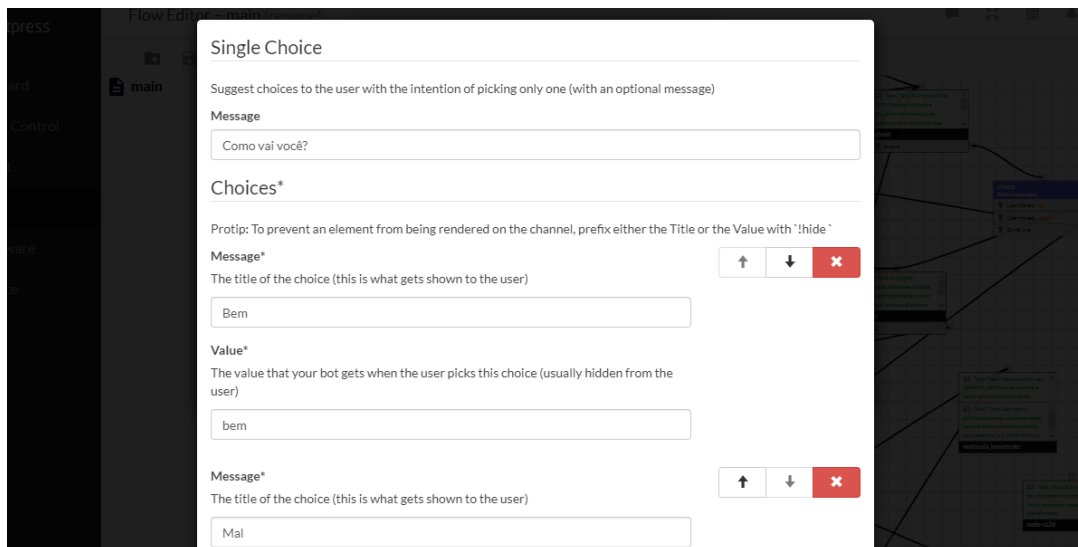


Figura 3.1: *Choice* - Componente de respostas através de opções de múltipla escolha

Diferentemente dos módulos de skill e de canais, que estendem as funcionalidades já existentes do Botpress, os módulos funcionais são aqueles que incluem funcionalidades novas. Um exemplo de módulo funcional é o HITL (Human in the loop). Com esse módulo, é possível pausar o *chatbot* em conversas específicas, permitindo um humano interagir com o usuário diretamente pelo painel, como mostrado na figura 3.2.

Por fim, o Botpress também permite criar *actions* (ações). Essas ações são trechos de código escritos em JavaScript que, diferente dos módulos, só são executadas quando invocados por algum bloco do fluxo da conversa, como na figura 3.3.

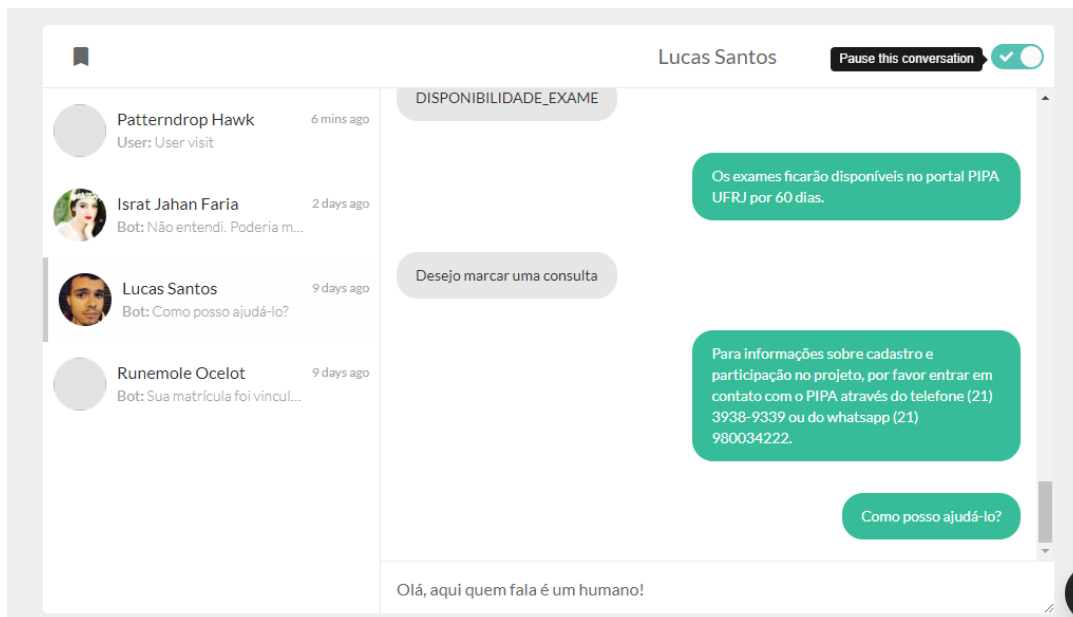


Figura 3.2: *HITL* - Interface de troca de mensagens no painel do Botpress

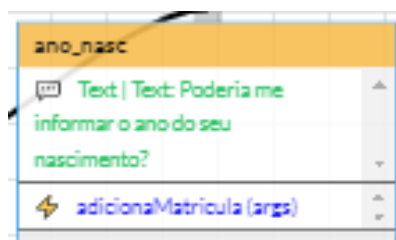


Figura 3.3: *Actions* - Bloco com uma função javascript que é executada após o envio da mensagem.

3.2 Canais

3.2.1 Facebook Messenger

É o aplicativo de mensagens desenvolvido originalmente para ser a plataforma de chat do Facebook. Com ele, os usuários podem se comunicar com outros usuários da plataforma através de textos, chamada de voz ou vídeo e também compartilhar diversos tipos de mídia e arquivos.

Em 2016 o Facebook anunciou uma plataforma de bots para o Messenger que recebe constantes melhorias desde então. Hoje, é possível que usuários façam **assinaturas** para receber conteúdos diretamente dos bots que ele conversa. Por padrão, os bots apenas respondem a mensagens de usuários mas com uma assinatura é possível com que o bot proativamente envie a mensagem.

3.2.2 Telegram

É um aplicativo de troca de mensagens multiplataforma, podendo ser acessado através de smartphones, computadores, ou pela interface web. Assim como o Messenger, é possível compartilhar arquivos de mídia com outros usuários. Por sua infraestrutura ser baseada em nuvem, é possível acessar as mídias de uma conversa de qualquer lugar. Além disso, todos os clients do Telegram são open source, e o serviço disponibiliza APIs para desenvolvedores criarem suas aplicações.

No Telegram, bots são um tipo especial de conta que não requer um número telefônico para ser criado. Os usuários interagem enviando mensagens e comandos pela conversa ou adicionando o bot a grupos. As mensagens são armazenadas nos servidores do Telegram até que o serviço que controle o bot leia e processe as mensagens. Os bots não podem enviar mensagens diretamente para qualquer usuário; é preciso que o usuário inicie uma conversa ou que ele seja adicionado a um grupo.

3.2.3 Slack

Slack é uma ferramenta colaborativa que tem como objetivo reunir pessoas, informações e as ferramentas certas para desenvolver algum tipo de trabalho. É largamente usado como ferramenta de comunicação em empresas.

Na plataforma, bots são um tipo de aplicativo que interage com o usuário através da conversação. Ele recebe exatamente os mesmos acessos que uma aplicação comum do Slack (inclusive, ao adicionarmos um bot, adicionamos uma integração - que é limitado pelo plano gratuito do Slack), com diferença que ele se torna um usuário como um outro qualquer do espaço colaborativo. É possível mencioná-lo em conversas, é possível mandar mensagens diretas, enviar arquivos e adicionar em grupos. Além disso, bots tem permissões para abrir uma nova conversa com algum usuário caso seja programado para isso.

3.3 Provedores de inteligência artificial e NLP

3.3.1 Wit.ai

Wit.ai é um provedor de inteligência artificial e processamento de linguagem natural gratuito criado pelo Facebook para ser integrado a diversos tipos de aplicações de software como aplicativos, wearables e bots². Suporta cerca de 50 linguagens diferentes.

²<https://wit.ai/>

3.3.2 IBM Watson

IBM Watson é uma suíte completa de ferramentas voltadas para inteligência artificial. Desenvolvida pela IBM, tem como principal destaque a precisão na detecção de intents mesmo com uma base de treino pequena. Com o Watson Assistant é possível desenvolver gratuitamente um *chatbot* sendo limitado a 10.000 chamadas de API por mês³.

3.3.3 LUIS.ai

É um serviço baseado em machine learning criado pela Microsoft focado em fornecer processamento de linguagem natural para bots, apps e dispositivos IoT. Desenhado para reconhecer informações de valor em mensagens de texto, o LUIS.ai é capaz de se integrar a outras ferramentas que compõem a suíte de aplicativos do Azure, como o Azure Bot Service⁴. Atualmente suporta cerca de 13 linguagens diferentes⁵.

³<https://www.ibm.com/watson/how-to-build-a-chatbot>

⁴<https://www.luis.ai/home>

⁵<https://docs.microsoft.com/pt-br/azure/cognitive-services/luis/luis-language-support>

Capítulo 4

Desenvolvimento do chatbot

4.1 Motivação

4.1.1 Contexto: o projeto PIPA UFRJ

O Projeto Infância e Poluentes Ambientais – PIPA UFRJ é um estudo epidemiológico denominado "Estudo longitudinal dos efeitos da exposição a poluentes ambientais sobre a saúde infantil - Coorte dos bebês". Este estudo tem como proposta fornecer informação que permita a investigação e análise dos efeitos dos poluentes ambientais especificamente: metais (chumbo, mercúrio, cádmio e arsênio), agrotóxicos e plastificantes, sobre o desenvolvimento das crianças, desde o período de gestação e nascimento, até os 4 anos de idade.

Em 2017 iniciou-se a fase de estudo piloto do projeto, avaliando a exposição da mãe e seu filho até os 6 meses de idade, além de suas informações sociodemográficas. Nessa fase as metodologias e estratégias propostas estão sendo testadas e validadas, a fim de aprimorar o estudo. Como resultado desse projeto, vai ser possível propor medidas preventivas e de controle, o que melhoraria a qualidade de vida da população.

O projeto é promovido pela Universidade Federal do Rio de Janeiro (UFRJ) realizado pela Maternidade Escola, pela Faculdade de Medicina e pelo Instituto de Estudos em Saúde Coletiva (IESC/UFRJ) e conta com diversos parceiros como a Fundação Oswaldo Cruz (FIOCRUZ) e o Centro de Estudos da Saúde do Trabalhador e Ecologia Humana (CESTEH).

4.1.2 Tecnologias utilizadas no PIPABOT

Node.JS

Node.js é um interpretador de código JavaScript construído sobre a máquina virtual Javascript V8 do Google, cujo foco é executar aplicações baseadas em rede no lado

do servidor.

A principal característica que diferencia o Node.js de outras tecnologias *server-side* como o PHP, é o fato de sua execução ser *single-thread*, ou seja, apenas uma thread é responsável por executar a aplicação, enquanto que outras linguagens a execução é *multi-thread*.

Os modelos tradicionais de aplicações web criam novas threads para cada requisição recebida, o que demanda recursos computacionais como memória RAM, por exemplo. Como esses recursos são limitados, haverá um número máximo de threads que poderão ser criadas até que os recursos sejam liberados novamente.

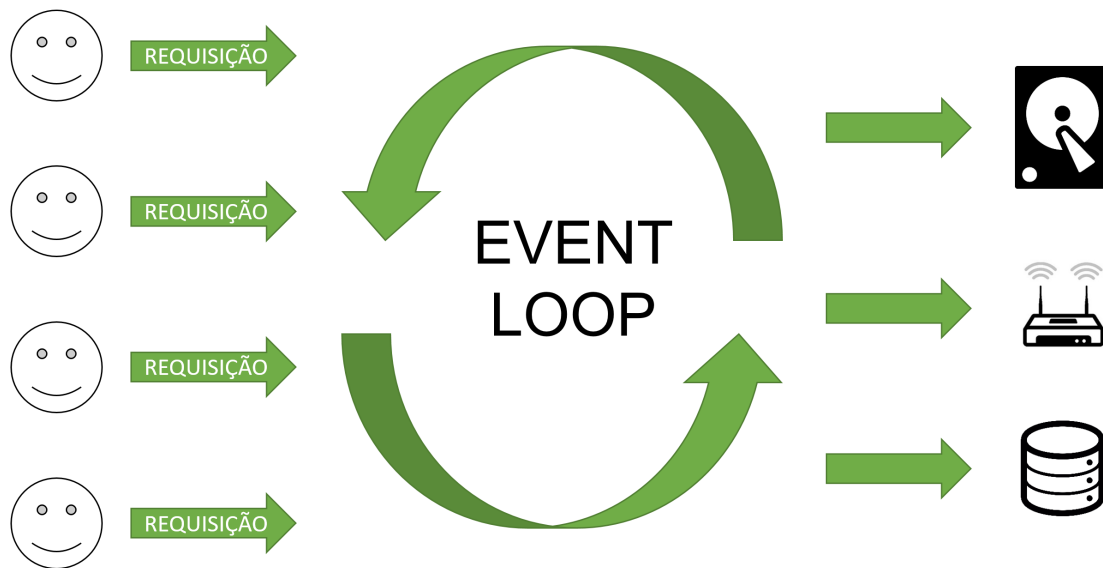


Figura 4.1: Arquitetura do NodeJS

No Node.js as requisições são tratadas por uma única thread, chamada de Event Loop. A thread é executada esperando eventos para tratar. Quando chega uma nova requisição, um evento é criado.

Para tratar a concorrência de requisições, o Node.js faz uso de chamadas de E/S não-bloqueantes. Assim, a sua única thread não fica esperando as demais chamadas já realizadas serem concluídas para continuar sua execução.

Graças a sua arquitetura, o Node.js consegue tratar um número maior de requisições concorrentes do que uma aplicação no modelo tradicional, se mostrando como uma das plataformas mais escaláveis da atualidade.

NPM

Node Package Manager, ou NPM, é o gerenciador de pacotes do Node.js. Nele é possível encontrar componentes open-source que agilizam o desenvolvimento de aplicações como conectores para bancos de dados, servidores web completos, entre

outros. Além disso, o NPM também faz o gerenciamento de dependências de um projeto.

Ao instalar um componente com o comando `npm install <componente>` em um projeto do Node.js, o utilitário adiciona o pacote e a versão utilizada e suas dependências em um arquivo chamado `package.json`; Desse modo é possível restaurar todas as dependências do projeto, caso seja necessário.

Git

Git é um sistema de versionamento de código distribuído que possui diversas ferramentas úteis para o desenvolvimento de um sistema. Sua escolha se deu pelo fato de ser open source, e por permitir criar ramificações (branches) de código a partir de um ponto do desenvolvimento, permitindo criar diferentes versões independentes entre os ramos¹. Através do uso de branches, foi criado um Git Flow para o PipaBot, como na figura 4.2.

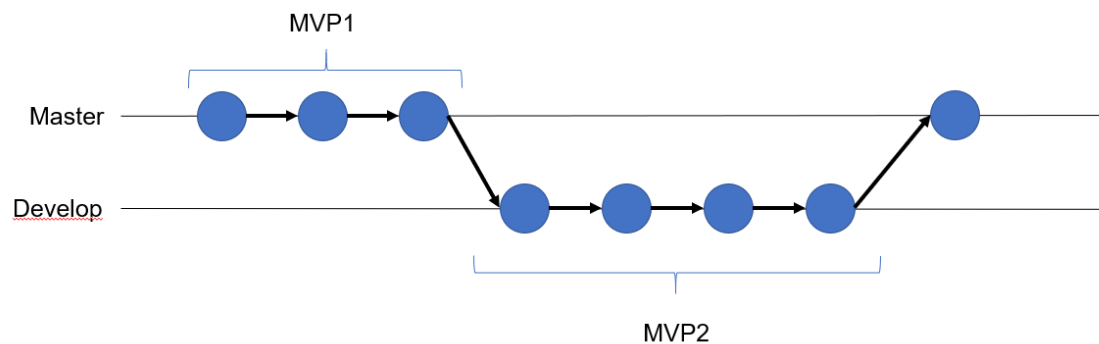


Figura 4.2: PipaBot - Organização do versionamento do código

Após o desenvolvimento da primeira versão do PipaBot, foi criada uma nova branch de desenvolvimento da próxima interação do ciclo Construir-Medir-Aprender. Isso foi feito para que a branch principal do repositório (master) possuísse um produto pronto para ser executado. Ao término da segunda versão do MVP, foi feito um merge da branch de desenvolvimento com a master.

4.2 Ambientes de desenvolvimento

O PipaBot foi desenvolvido em uma estrutura formada por dois ambientes: um ambiente de desenvolvimento e um ambiente de homologação.

O ambiente de desenvolvimento é executado no próprio computador onde as implementações são feitas e é acessível apenas localmente. É composto de um servidor

¹<https://git-scm.com/about>

HTTP do Botpress, que gera uma instância do PipaBot, tornando possível testar funcionalidades através de um chat na URL `http://localhost:3000/s/chat`.

O ambiente de homologação é composto por dois servidores HTTP e um servidor de banco de dados MySQL, todos localizados na nuvem e recebem as versões incrementais do *bot* a cada ajuste funcional, para que sejam testadas e validadas pelos *stakeholders*.

Um dos servidores HTTP é o Botpress, contendo a última release do ambiente de desenvolvimento. O outro servidor HTTP executa uma instância do Wordpress, para que seja permitido simular o Portal do PIPA e validar a versão web do PipaBot. O servidor MySQL é utilizado pelo Wordpress para armazenar os dados dos usuários.

4.3 A solução

4.3.1 Arquitetura

A arquitetura do PipaBot é composta por duas camadas, o *front-end* e o *back-end*. O *front-end* é toda a interface utilizada pelo usuário para interagir com o bot (comumente chamado de canais): o portal do PIPA através da versão web, ou pelo aplicativo do Messenger em dispositivos mobile ou PC. Enquanto o *back-end* é onde se encontra o processamento do *bot*. É nele onde as mensagens são classificadas e as respostas, escolhidas. Além disso, o back-end também é responsável pela integração dos dados dos pacientes do Portal do PIPA com o *bot*. O esquema detalhado da arquitetura pode ser visto na figura 4.3.

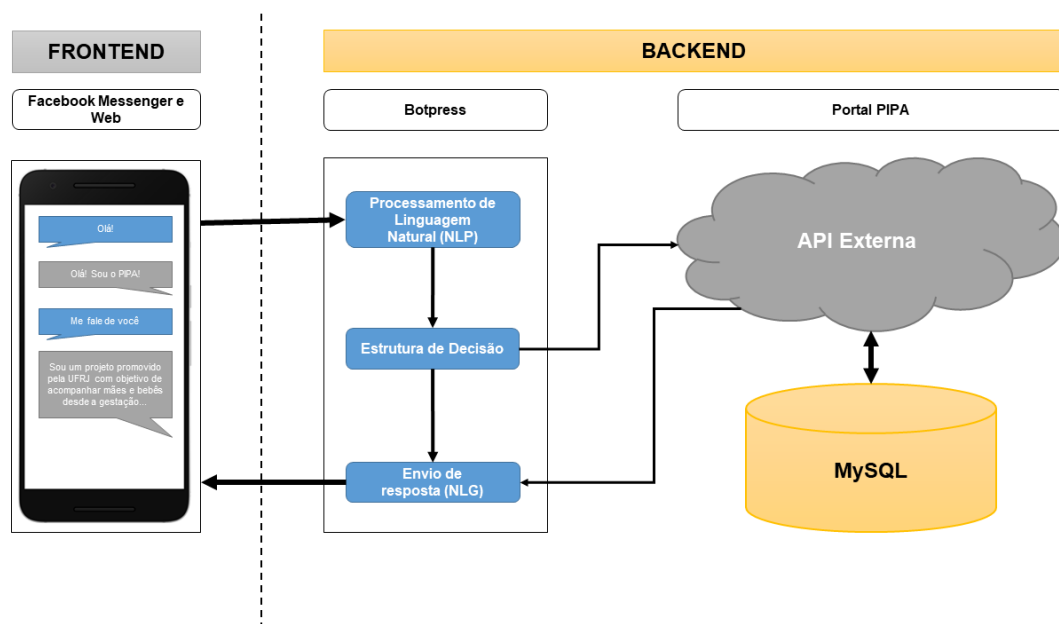


Figura 4.3: Arquitetura do PipaBot

Como framework de criação de bots, foi escolhido o **Botpress**, por possuir, em uma única solução, funções como processamento de linguagem natural, criador de fluxos de conversa, estruturas de decisão (permitindo a criação de chatbots baseados em regras ou em inteligência artificial, através de classificação de intents) e suportar diversos canais de comunicação para um mesmo *bot*, além de ser uma estrutura onde todo o código fica sob domínio do administrador do *bot*, o que é importante principalmente para implementações de funcionalidades específicas do *PipaBot* como o acesso à base de usuários, por exemplo.

Para integrar o PipaBot ao Portal do PIPA, que é desenvolvido utilizando o sistema de gestão de conteúdo Wordpress (construído em PHP e MySQL), foi utilizada a própria API REST do Wordpress. Essa API implementa endpoints que permitem acessar o CRUD (*Create, Read, Update e Delete* - Operações de leitura e escrita de dados do sistema) de diversas tabelas que guardam o conteúdo do Portal no banco de dados, inclusive de usuários. Contudo, por padrão, as rotas de leitura de usuários apenas retornam usuários que já fizeram alguma publicação no Portal, o que não era o caso dos pacientes, que nem possuem permissão de executar tal ação. Então, foram criados três novos *endpoints*, que só podem ser acessado através de autenticação através da tecnologia JWT (*JSON Web Token*), que utiliza o perfil de um usuário administrador do Portal. Esses *endpoints* são responsáveis por (I) verificar se um usuário existe através do seu CPF e ano de nascimento, e (II) por adicionar um identificador do usuário do *chatbot* no cadastro do Portal e (III) por verificar se um usuário do bot está atrelado a um cadastro de paciente no Portal - através do identificador cadastrado no *endpoint* II.

Para tornar a instalação dos recursos do PipaBot mais ágil e manutenível, todos as implementações dos *endpoints* foram encapsulados em um arquivo PHP que é reconhecido pela instalação do Wordpress como um plugin, que pode ser ativado ou desativado a qualquer momento através do painel do administrativo do Portal.

4.3.2 Modelos

Identificação	RF01		
Casos de Uso Relacionados	TESTE		
Descrição	Lorem ipsum damet...		
Prioridade	<input type="checkbox"/> Essencial	<input type="checkbox"/> Importante	<input type="checkbox"/> Desejável

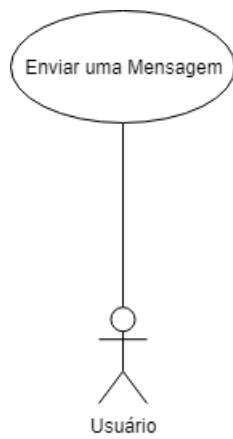


Figura 4.4: Casos de Uso do Usuário

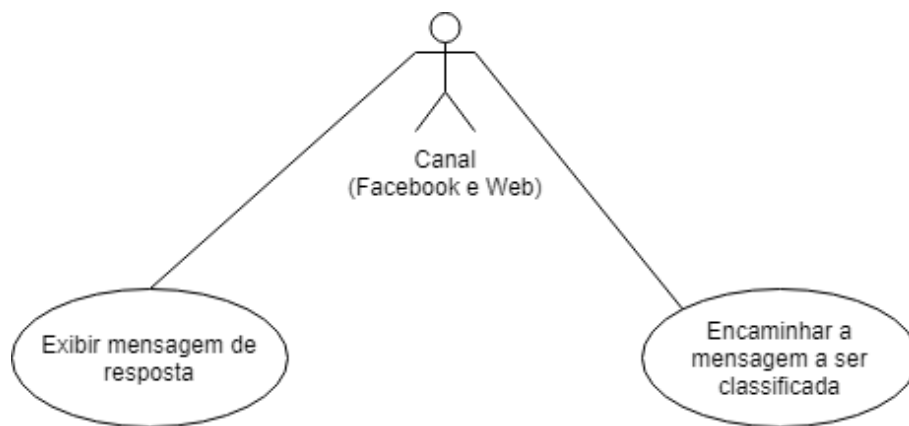


Figura 4.5: Casos de Uso do Canal (Messenger e Web)

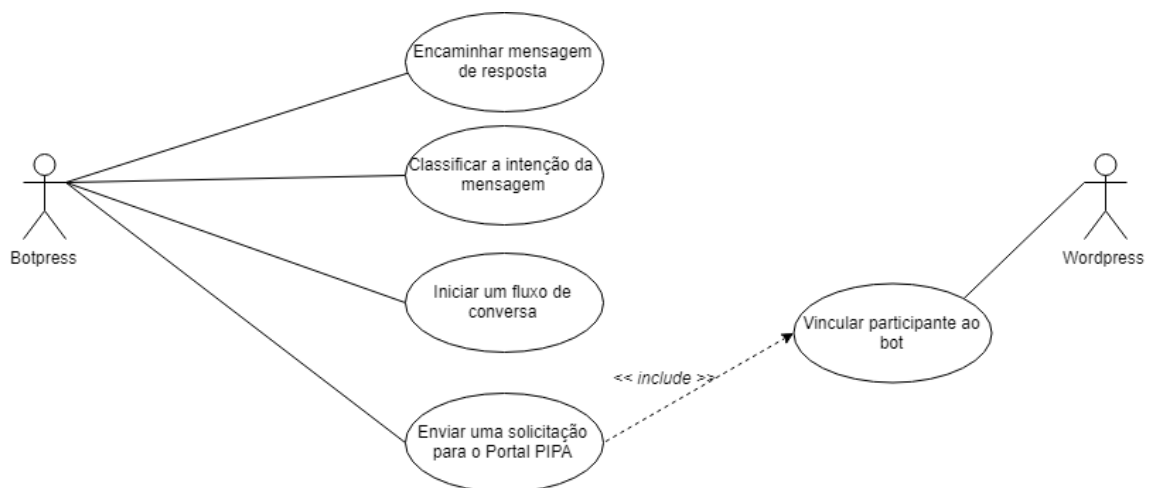


Figura 4.6: Casos de Uso do Bot

4.3.3 MVP 1

Tendo como principal problema a falta de engajamento das mães com o projeto e buscando criar um novo canal de informação sobre o PIPA, o PipaBot teve a sua primeira versão orientada a regras. Para garantir que o usuário navegasse através dos fluxos de conversa, o *bot* exibia botões, de modo que o usuário não desviasse a conversa para um outro assunto no meio de um fluxo. Essa versão era capaz de dar informações sobre o funcionamento e objetivo do projeto, e vincular usuários a um cadastro no Portal do PIPA.

O principal objetivo da primeira versão era apresentar aos pesquisadores do projeto o que um *chatbot* poderia ser capaz de fazer, demonstrando integrações com o sistema do PIPA e elementos de interação, visando novas ideias para a versão seguinte.

4.3.4 MVP 2

A segunda versão do PipaBot trouxe diversas melhorias. Foram criados diálogos exclusivos para pacientes do projeto como informações sobre consultas e exames além de ter também o treinamento para diversos *intents*, que eram reconhecidos através da tecnologia de NLP do Botpress. Cada *intent* levava a um fluxo de conversa diferente cuja interação poderia ser feita não somente por botões, mas textualmente, dando mais naturalidade ao diálogo.

4.3.5 Avaliação do produto

Para avaliar o PipaBot foi utilizado o modelo TAM (*Technology Acceptance Model*)[6] que procura determinar os aspectos de utilidade e facilidade de uso de tecnologias. Para isso, o TAM se baseia nos conceitos de **percepção de utilidade** e **percepção de facilidade de uso**, métricas que avaliam, respectivamente, o quanto o usuário acredita que a tecnologia possa melhorar seu desempenho e o quanto ele acredita que, ao utilizar a tecnologia, ele possa ficar livre de esforço físico e mental.

Para elaboração do TAM, foi utilizado o paradigma GQM (*Goal/Question/Metric*), ilustrado na figura 4.7, que consiste em descrever os objetivos (*Goals*), a partir dos objetivos elaborar um conjunto de questões (*Questions*) e, então, métricas (*metrics*) para medir as respostas. Os objetivos do PipaBot estão definidos nas tabelas abaixo.

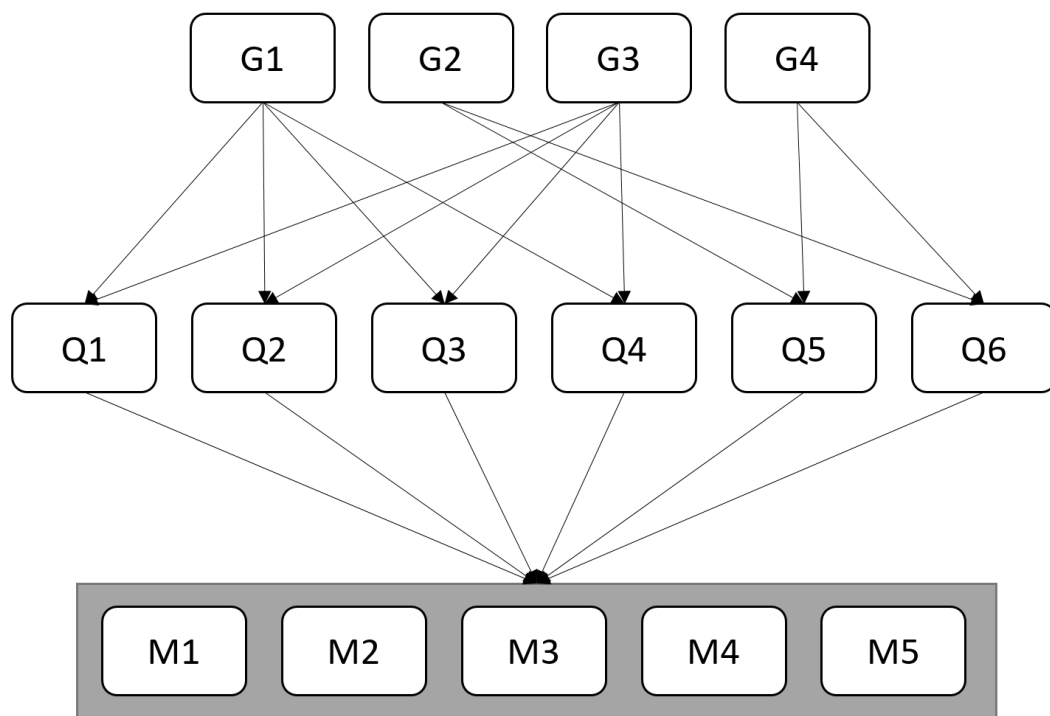


Figura 4.7: Organização do Paradigma GQM aplicado ao PipaBot

Analisar	O PipaBot (chatbot do PIPA)
Com o propósito de	Caracterizar
Com respeito a	Facilidade de uso (número de mensagens e tempo de resposta)
Sob o ponto de vista de	Profissionais de computação
No contexto de	Profissionais de computação (estudantes de graduação e pós graduação) realizando tarefas de busca de informação no PipaBot.

Tabela 4.1: GQM - Objetivo 1

Analisar	O PipaBot (chatbot do PIPA)
Com o propósito de	Compreender
Com respeito a	Utilidade (pertinência e adequação do conteúdo da resposta)
Sob o ponto de vista de	Profissionais de computação
No contexto de	Estudantes de graduação e pós-graduação do sexo feminino realizando tarefas de busca de informação no PipaBot.

Tabela 4.2: GQM - Objetivo 2

Foi elaborado um conjunto de 6 questões abordando os objetivos G1, G2, G3 e

Analisar	O PipaBot (chatbot do PIPA)
Com o propósito de	Caracterizar
Com respeito a	Facilidade de uso (número de mensagens e tempo de resposta)
Sob o ponto de vista de	Profissionais de saúde
No contexto de	Profissionais de saúde (estudantes de pós-graduação) realizando tarefas de busca de informação no PipaBot.

Tabela 4.3: GQM - Objetivo 3

Analisar	O PipaBot (chatbot do PIPA)
Com o propósito de	Compreender
Com respeito a	Utilidade (pertinência, adequação e utilidade do conteúdo da resposta)
Sob o ponto de vista de	Profissionais de saúde
No contexto de	Estudantes do sexo feminino de pós-graduação na área da saúde realizando tarefas de busca de informação no PipaBot.

Tabela 4.4: GQM - Objetivo 4

G4, visando capturar a percepção de facilidade de uso e utilidade na visão dos usuários. Para medir esses dados foi utilizada a escala *Likert*, no qual os participantes especificam seu nível de concordância com uma afirmação. As questões e métricas estão exibidas nas tabelas 4.5 e 4.6 respectivamente. Junto as métricas foi disponibilizado um espaço para que os participantes pudessem expressar seus comentários sobre cada questão.

Q1	O PipaBot é fácil de usar
Q2	O PipaBot responde rapidamente aos meus questionamentos
Q3	Eu posso obter a informação que desejo com poucas perguntas
Q4	Os elementos de navegação (menus e botões) do PipaBot facilitam a interação ao longo da conversa
Q5	O PipaBot fornece respostas pertinentes aos meus questionamentos
Q6	O PipaBot fornece respostas adequadas aos meus questionamentos

Tabela 4.5: GQM - Questões para avaliação de grau de concordância

...

M1	Concordo Totalmente
M2	Concordo Parcialmente
M3	Indiferente
M4	Discordo Parcialmente
M5	Discordo Totalmente

Tabela 4.6: GQM - Métricas

Capítulo 5

Conclusão

5.1 Contribuições

5.2 Limitações

As principais limitações do PipaBot são:

1. O PipaBot não consegue classificar *entities* dada uma limitação da versão do Botpress.
2. Fluxos que requerem acesso a fontes externas como consultas estão direcionando para um telefone de contato dada a não existência de um sistema de agendamento eletrônico no qual o PipaBot pudesse se integrar.

5.3 Trabalhos futuros

5.4 Considerações finais

Referências Bibliográficas

- [1] E MARIO HENRIQUE ADANIYA, V. H. A. “Desenvolvimento e aplicações de Chatbot”, *Revista Terra & Cultura: Cadernos de Ensino e Pesquisa*, v. 34, n. esp., 2018. ISSN: 2596-2809. Disponível em: <<http://periodicos.unifil.br/index.php/Revistateste/article/view/297>>.
- [2] JAIN, M., KOTA, R., KUMAR, P., et al. “Convey: Exploring the use of a context view for chatbots”. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, p. 468. ACM, 2018.
- [3] DE PAULA, L., ANGRADE, M., ROLIM, L., et al. “A revolução da comunicação cidadão-gestor público através da troca de mensagens automatizada”. In: *Anais do XIV Encontro Nacional de Engenharia e Desenvolvimento Social*, 2017. Disponível em: <<http://www.eneds.net/anais/index.php/edicoes/eneds2017/paper/view/537>>.
- [4] OLIVEIRA, N., COSTA, A., ARAUJO, D., et al. “HelpCare: Um Protótipo de ChatBot para o Auxílio do Tratamento de Doenças Crônicas”. In: *Anais do XIX Simpósio Brasileiro de Computação Aplicada à Saúde*, pp. 282–287, Porto Alegre, RS, Brasil, 2019. SBC. Disponível em: <<https://portaldeconteudo.sbc.org.br/index.php/sbcas/article/view/6263>>.
- [5] AVILA, C., ROLIM, T., DA SILVA, J. W., et al. “MediBot: Um chatbot para consulta de riscos e informações sobre medicamentos”. In: *Anais Estendidos do XIX Simpósio Brasileiro de Computação Aplicada à Saúde*, pp. 1–6, Porto Alegre, RS, Brasil, 2019. SBC. Disponível em: <https://sol.sbc.org.br/index.php/sbcas_estendido/article/view/6275>.
- [6] DAVIS, F. D. “User acceptance of information technology: system characteristics, user perceptions and behavioral impacts”, *International journal of man-machine studies*, v. 38, n. 3, pp. 475–487, 1993.

Apêndice A

Algumas Demonstrações