



Universidade Federal
do Rio de Janeiro

Escola Politécnica

PIPABOT: UMA SOLUÇÃO CHATBOT PARA O 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

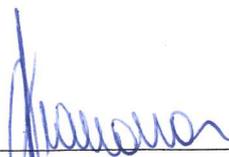
Rio de Janeiro
Setembro de 2019

PIPABOT: UMA SOLUÇÃO CHATBOT PARA O 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.

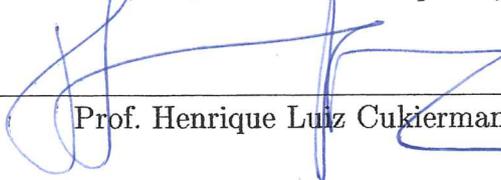
Examinado por:



Prof. Guilherme Horta Travassos, D.Sc.



Prof. Taisa Guidini Gonçalves, Ph.D.



Prof. Henrique Luiz Cukierman, D.Sc.



Prof. Thatiana Verônica Rodrigues de Barcellos Fernandes, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
SETEMBRO DE 2019

de Paula, Lucas Santos

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

XIV, 71 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. 37 – 38.

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 primeiramente a Deus por todas as oportunidades que me foram concedidas.

Aos meus pais, Elizabeth e Breno, ao meu irmão, Igor, e ao pequeno Arthur, meu afilhado, por todo apoio que me deram ao longo desses anos. Sem dúvida vocês foram os principais responsáveis por me manter de pé mesmo com os diversos tropeços da vida.

Ao meu orientador, Guilherme Horta Travassos, por todo conhecimento que me foi passado ao longo da graduação e também durante a construção deste trabalho. Agradeço pela oportunidade de desenvolver este projeto sob sua orientação e também por toda dedicação e paciência.

A minha co-orientadora, Taísa Guidini Gonçalves, por toda disponibilidade, dedicação e paciência, que foram fundamentais para a construção deste trabalho.

Aos pesquisadores do PIPA UFRJ e COPPE UFRJ por todo apoio e disponibilidade para que esse projeto fosse realizado. Gostaria de agradecer também ao Instituto de Estudos de Saúde Coletiva (IESC/UFRJ) pela bolsa oferecida ao longo dos meses de desenvolvimento deste trabalho.

Aos demais membros da equipe PrefeitoBot - Lucas Rolim, Matheus Guedes, Rafael Damasceno e Rodrigo Carvalho - projeto que serviu de inspiração para este trabalho.

A Danielle Mello, pelo apoio incondicional e por ser peça fundamental na minha vida.

Aos meus amigos e amigas, em especial, Ana Carolina Ramos, Andréa Doreste, Bruno Dias, Felipe Barros, Felipe Villela, Gustavo Machado, Heitor Muniz, Igor Amaral, João Pedro Moreira, Juliano Marinho, Kamilla Moreira, Lean Marcheni, Leonardo de Mello, Lucas Abreu, Lucas Soares, Luís Felipe Vianna, Marcos Filho, Marcelo Santos, Maria Eduarda Nascimento, Pedro Bicalho e Rayanny Silva. Sou imensamente grato por ter todos vocês na minha vida.

A todos os meus colegas de trabalho por todo apoio, paciência e compreensão ao longo do desenvolvimento deste projeto.

A toda sociedade brasileira, por ter custeado todos esses anos de estudo. Espero poder retribuir, ajudando na construção de uma sociedade melhor.

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

PIPABOT: UMA SOLUÇÃO CHATBOT PARA O PIPA UFRJ

Lucas Santos de Paula

Setembro/2019

Orientadores: Guilherme Horta Travassos

Táisa Guidini Gonçalves

Curso: Engenharia de Computação e Informação

Chatbots são sistemas de software que atuam como canal facilitador para a comunicação entre humanos e entidades (empresas, projetos sociais e de pesquisa, figuras públicas, entre outros) utilizando técnicas de processamento de linguagem natural para processar as mensagens recebidas. Esse trabalho apresenta o PipaBot, um *chatbot* aplicado ao contexto do Projeto Infância e Poluentes Ambientais (PIPA UFRJ). É esperado que o *bot* atenda cerca de duas mil mães participantes do projeto tirando dúvidas e fornecendo informações sobre o projeto.

O PipaBot foi desenvolvido para duas plataformas: o aplicativo *Messenger* do *Facebook* e o *website* do PIPA. Através dele é possível obter informações sobre o projeto e seus objetivos, e também realizar uma identificação como participante do projeto, que libera acesso a informações como consultas e resultados de exames médicos.

O *chatbot* foi avaliado segundo o modelo de aceitação da tecnologia (TAM - *Technological Acceptance Model*), por 13 usuários divididos em dois grupos: profissionais da saúde, e profissionais de computação. Os dados coletados apresentam apenas respostas positivas para os questionamentos feitos, entretanto foram sugeridas diversas melhorias que são discutidas neste trabalho.

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

PIPABOT: A CHATBOT SOLUTION FOR PIPA UFRJ

Lucas Santos de Paula

September/2019

Advisors: Guilherme Horta Travassos
Taísa Guidini Gonçalves

Course: Computer Engineering

Chatbots are software systems that act as a facilitating channel for communication between humans and entities (companies, social and research projects, public figures, among others) using natural language processing techniques to process incoming messages. This work presents PipaBot, a chatbot applied to the context of the Childhood and Environmental Pollutants Project (PIPA UFRJ). The bot is expected to serve about 2,000 project participants by answering questions and providing information about the project.

PipaBot was developed for two platforms: Facebook Messenger app and PIPA website. Through it, it is possible to obtain information about the project and its objectives, as well as identify as a project participant, which gives access to information such as medical appointments and medical examination results.

PipaBot was evaluated according to the Technological Acceptance Model (TAM) by 13 users divided into two groups: healthcare professionals and computer professionals. The collected data have only positive answers to the questions asked, however several improvements have been suggested which are discussed in this work.

Sumário

Lista de Figuras	xi
Lista de Tabelas	xiii
Lista de Abreviaturas	xiv
1 Introdução	1
1.1 Motivação	1
1.2 Objetivo	3
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 de chatbots	7
2.5 Uso de chatbots na área médica	8
2.6 Síntese e Conclusão	8
3 Tecnologias utilizadas na construção de chatbots	10
3.1 Frameworks	10
3.1.1 Chatfuel	10
3.1.2 Botpress	10
3.2 Canais	13
3.2.1 Facebook Messenger	13
3.2.2 Telegram	13
3.2.3 Slack	13
3.2.4 Web	14
3.3 Provedores de inteligência artificial e NLP	14
3.3.1 Wit.ai	14
3.3.2 IBM Watson	15

3.3.3	LUIS.ai	15
3.3.4	Botpress	15
3.4	Síntese e Conclusão	15
4	Desenvolvimento do PipaBot	17
4.1	Motivação e Contexto	17
4.2	Tecnologias utilizadas no PipaBot	18
4.2.1	Node.JS	18
4.2.2	NPM	19
4.2.3	Git	19
4.3	Ambientes de desenvolvimento	20
4.4	A solução PipaBot	20
4.4.1	Requisitos do Sistema	20
4.4.2	Arquitetura	20
4.4.3	Documentação do PipaBot	22
4.4.4	MPV 1	24
4.4.5	MPV 2	25
4.4.6	Testes Funcionais	26
4.4.7	Avaliação do PipaBot	27
4.5	Síntese e Conclusão	34
5	Conclusão	35
5.1	Contribuições	35
5.2	Limitações	35
5.3	Trabalhos futuros	36
5.4	Considerações finais	36
	Referências Bibliográficas	37
	A Documento de Requisitos	39
	B Descrição dos Casos de Uso	48
	C Diagramas de Atividades	54
	D Testes Funcionais	57
D.1	Obter informações sobre o PIPA	57
D.2	Obter informações sobre como criar um login no Portal do PIPA	58
D.3	Vincular um usuário do chatbot à sua conta no Portal do PIPA	59
D.4	Obter informações sobre exames	60
D.5	Obter informações sobre consultas	61

E	Formulário de Avaliação	63
F	Roteiro de Avaliação	68

Lista de Figuras

1.1	Lean Startup (adaptado de [1])	3
2.1	Interpretação de uma frase com processamento de linguagem natural	5
2.2	Fluxograma de um <i>chatbot</i> baseado em regras	6
3.1	<i>Choice</i> - Componente de respostas através de opções de múltipla escolha	11
3.2	<i>HITL</i> - Interface de troca de mensagens no painel do <i>Botpress</i>	12
3.3	<i>Actions</i> - Bloco com uma função <i>javascript</i> que é executada após o envio da mensagem.	12
3.4	Chatbot do site Resultados Digitais	14
4.1	Arquitetura do NodeJS - Tratamento de múltiplas requisições	18
4.2	PipaBot - Organização do versionamento do código	19
4.3	Arquitetura do PipaBot	21
4.4	Diagrama de Casos de Uso do PipaBot	22
4.5	Exemplo de diagrama de atividades	24
4.6	Fluxo de cadastro do participante PIPA para utilizar o <i>chatbot</i>	25
4.7	Fluxo de informações sobre o projeto, classificado através da tecnologia de NLP.	26
4.8	Resultado do Teste Funcional 1	27
4.9	Avaliação de Facilidade de Uso - Grupo de Profissionais de Computação	30
4.10	Avaliação de Utilidade - Grupo de Profissionais de Computação . . .	31
4.11	Avaliação de Facilidade de Uso - Grupo de Profissionais de Saúde . .	32
4.12	Avaliação de Utilidade - Grupo de Profissionais de Saúde	32
C.1	Diagrama de atividades para o fluxo de conversa Sobre o Projeto . . .	54
C.2	Diagrama de atividades para o fluxo de conversa de Exames	55
C.3	Diagrama de atividades para o fluxo de conversa de Cadastro	56
D.1	Resultado do TF01	58
D.2	Resultado do TF02	59
D.3	Resultado do TF03	60

D.4	Resultado do TF04	61
D.5	Resultado do TF05	62

Lista de Tabelas

4.1	Exemplo de Descrição de Caso de uso	23
4.2	Testes funcionais	26
4.3	Exemplo de teste funcional	27
4.4	GQM - Objetivo G1	28
4.5	GQM - Objetivo G2	28
4.6	GQM - Questões	29
4.7	GQM - Opções de resposta	29
4.8	Respostas do Grupo de Profissionais de Computação	30
4.9	Comentários - Profissionais de Computação	31
4.10	Respostas do Grupo de Profissionais de Saúde	32
4.11	Comentários - Profissionais de Saúde	33
B.1	Caso de Uso 1 - Enviar Mensagem	48
B.2	Caso de Uso 2 - Enviar mensagem de resposta	49
B.3	Caso de Uso 3 - Encaminhar a mensagem a ser classificada	49
B.4	Caso de Uso 4 - Encaminhar mensagem de resposta	50
B.5	Caso de Uso 5 - Classificar a intenção da mensagem	50
B.6	Caso de Uso 6 - Iniciar um fluxo de conversa	51
B.7	Caso de Uso 7 - Enviar uma solicitação para o Portal PIPA	52
B.8	Caso de Uso 8 - Vincular participante ao <i>bot</i>	53
D.1	Teste Funcional 1 - Fluxo de informações sobre o PIPA	57
D.2	Teste Funcional 2 - Fluxo de criação de login no portal	58
D.3	Teste Funcional 3 - Fluxo de cadastro do participante no <i>chatbot</i>	59
D.4	Teste Funcional 4 - Fluxo de exames	60
D.5	Teste Funcional 5 - Fluxo de consultas	61

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*¹. Segundo dados da 143ª Pesquisa CNT/MDA² realizada em fevereiro de 2019, 82% dos entrevistados afirmam que fazem uso de seus *smartphones* para acesso à redes sociais. Além disso, os resultados da 29ª Pesquisa Anual de Administração e Uso de Tecnologia da Informação nas Empresas, realizada pela FGV em 2018, mostra que o Brasil já ultrapassou a marca de um aparelho *smartphone* por habitante³.

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*⁴.

Contudo, estudos realizados pela *Forrester Research* em 2015 indicam que apenas cinco aplicativos são responsáveis por cerca de 84% do uso de aplicativos não nativos do *smartphone*⁵. Esses aplicativos variam de pessoa para pessoa, mas em geral, incluem aplicativos de redes sociais ou de mensagens instantâneas. Esse cenário

¹<https://exame.abril.com.br/negocios/dino/62-da-populacao-brasileira-esta-ativa-nas-redes-sociais/>

²<https://www.cnt.org.br/agencia-cnt/confira-resultados-pesquisa-cnt-md>

³<https://eaesp.fgv.br/sites/eaesp.fgv.br/files/pesti2018gvciappt.pdf>

⁴<https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>

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

justifica o aumento no número de *chatbots* disponíveis em plataformas de mensagens como *Messenger* nos últimos anos.

Chatbots são sistemas computacionais que simulam uma conversa com pessoas e tornam a interação homem-máquina mais natural[2]. 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.

O formato de comunicação através de troca de mensagens oferecido pelos *chatbots* é mais natural 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 ao *bot* que responde prontamente.

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.

Observando o cenário de baixa eficiência do canal de atendimento à situações emergenciais da cidade do Rio de Janeiro (Através do telefone ou aplicativo conhecido como 1746) e também quanto a unificação de informações sobre a prefeitura da cidade, foi desenvolvido, em caráter experimental, um assistente virtual chamado PrefeitoBot[3]. A ferramenta, além de oferecer informações sobre as secretarias, funcionamento e atuação da prefeitura, disponibilizava também um meio de reportar incidentes na cidade como queda de postes de luz, alagamentos, poluição sonora, entre outros. Esse experimento permitiu avaliar o cenário tecnológico descrito neste capítulo, e concluir que o cenário na construção de *chatbots* é favorável, podendo encontrar inclusive assistentes aplicados à educação[4], vendas de produtos[5], saúde ([6], [7] e [8]), dentre outros domínios de aplicação.

1.2 Objetivo

O objetivo principal desse trabalho é construir um *chatbot* tendo como base boas práticas de engenharia de software[9]. O *bot* deverá ser disponibilizado para o PIPA UFRJ (Projeto Infância e Poluentes Ambientais) em duas plataformas: *Messenger* e *web*.

1.3 Metodologia

A metodologia utilizada para o desenvolvimento desse projeto segue a linha **Construir - Medir - Aprender** do *Lean Inception*[1], cuja finalidade é construir versões incrementais do Mínimo Produto Viável (MPV) a cada ciclo.

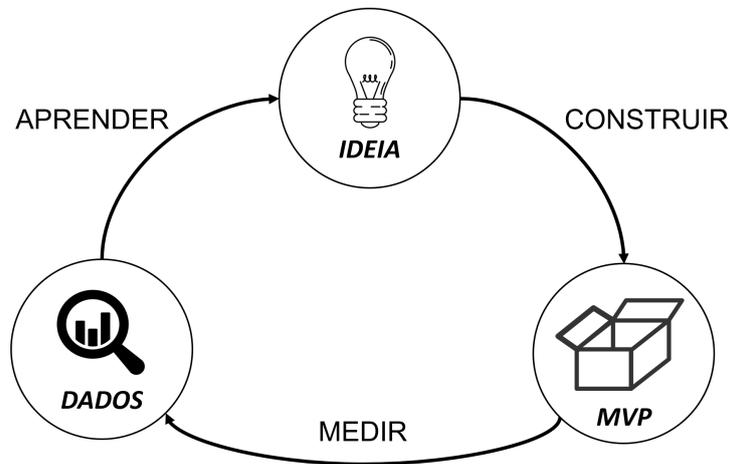


Figura 1.1: Lean Startup (adaptado de [1])

Cada ciclo parte de uma ideia na qual o MPV 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 na construção de *chatbots*.

O capítulo 4 apresentará o contexto do PipaBot, bem como as escolhas do projeto e suas devidas justificativas, os ambientes de desenvolvimento, a arquitetura do PipaBot e as versões do MPV construídas.

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

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[11]. Em geral, a conversa transcorre com cliques em botões, mantendo uma ordem nas interações, o que garante maior controle sobre a conversa.

As regras executam, em geral, uma operação booleana que implica em uma determinada ação, como na figura 2.2.

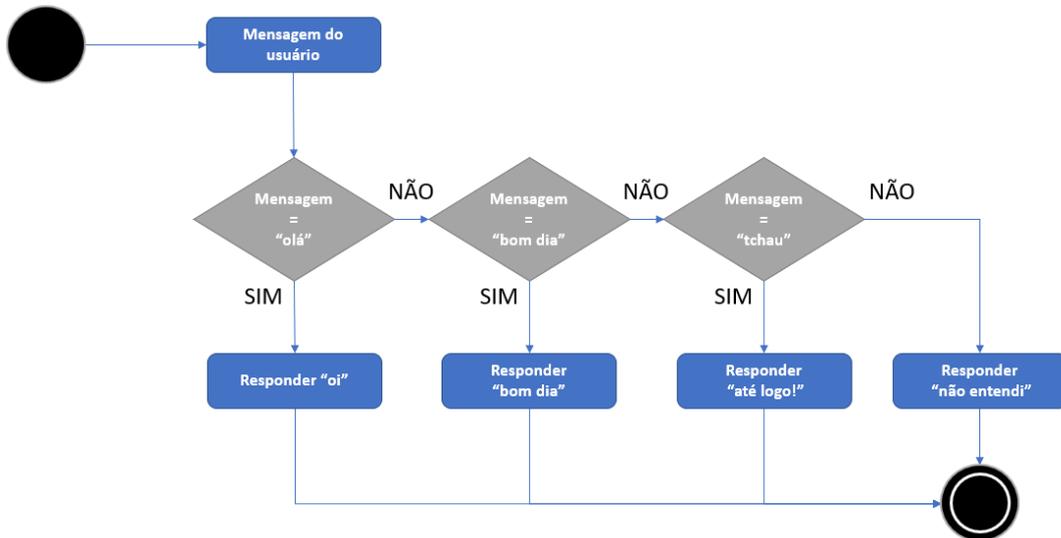


Figura 2.2: Fluxograma de um *chatbot* baseado em regras

A LEGO⁶, 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 intenções e *entities* de um texto enviado pelo usuário através de algoritmos de *machine learning*, e também tomar decisões com base nos dados extraídos e no contexto da conversa[11].

Os grandes avanços obtidos no desenvolvimento do *hardware*, em particular nas placas gráficas (GPU - *Graphics Processing Unit*) tornou possível um aprofunda-

⁶<https://www.messenger.com/t/LEGO>

mento no estudo de técnicas mais avançadas de inteligência artificial, como *machine learning*⁷.

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

Para promover o lançamento do programa *Genius* o canal *National Geographic* criou um *chatbot*⁸ que utiliza a inteligência artificial para criar um diálogo entre o usuário e o físico Albert Einstein. O *bot* era treinado com informações sobre os tópicos de estudo de Einstein, como a relatividade. Além disso, o *bot* também respondia informações pessoais e profissionais em tom de humor.

2.4 Uso de chatbots

A aplicabilidade dos *chatbots* é, em geral, a mesma que qualquer outro sistema computacional, pois trata-se de uma interface de interação que permite a execução de comandos de máquina e a 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 o *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.

Chatbots também estão presentes no setor de varejo. As Casas Bahia lançou em 2017 o Bahianinho⁹, 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.

Ainda em 2017 o Rock in Rio criou o Roque¹⁰. O *bot* do festival interagiu com mais de 77 mil usuários, trocando cerca de 3 milhões de mensagens¹¹ 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.

⁷<http://www.cienciaedados.com/gpu-e-deep-learning/>

⁸<https://www.impactbnd.com/blog/marketing-chatbot-examples>

⁹<https://www.facebook.com/CasasBahia/>

¹⁰<https://take.net/blog/take-notes/case-take-chatbot-no-rock-in-rio/>

¹¹<https://take.net/blog/chatbots/cases-de-chatbots-famosos/>

2.5 Uso de chatbots na área médica

Os *chatbots* podem desempenhar um papel estratégico na área médica. Em geral, estão disponíveis 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.

O AGEbot[6] é um *chatbot* que atende a área de saúde, mais precisamente respondendo dúvidas sobre epilepsia. Sua base de conhecimento é composta por cerca de 158 perguntas.

Desenvolvido com a ferramenta *Pandorabots*, o AGEbot é acessível através de uma interface web que se integra com o interpretador AIML (*Artificial Intelligence Markup Language*) do *Pandorabots*. O interpretador, então, busca na base de conhecimento a melhor mensagem para responder o usuário.

Para responder os questionamentos a respeito de medicamentos e seus riscos foi criado o MediBot[7]. Implementado em *JavaScript* com a tecnologia *NodeJS*, o *bot* atende os usuários através do aplicativo *Telegram* e também em uma interface *web*.

O MediBot possui dois modos de operação: respostas rápidas e interativo. No primeiro modo, o *bot* classifica a intenção do usuário através de palavras chave da mensagem, enquanto que no modo interativo o usuário navega entre opções e fluxos de conversa, que são exibidas de acordo com o contexto do diálogo.

Por fim, existem também *chatbots* como o HelpCare[8] que auxiliam no tratamento de doenças crônicas como a diabetes, controle de colesterol, hipertensão arterial. Buscando criar um protótipo que atendesse aos cenários das doenças citadas, o *bot* foi desenvolvido utilizando a ferramenta *IBM Watson*.

Dentre as principais funcionalidades do HelpCare encontram-se a retirada de dúvidas, lembretes de medicações, auxílio na dieta, listagem de hospitais próximos, entre outras.

Os exemplos de *bots* citados anteriormente indicam que esse tipo de ferramenta é bem útil para o atendimento do setor médico, dada a sua praticidade de uso e a grande quantidade de conhecimento existente que pode ser incorporado em um *chatbot*.

2.6 Síntese e Conclusão

Os *chatbots* podem ser caracterizados em dois tipos: baseado em **regras** ou **inteligência artificial**, e suas definições foram apresentadas neste capítulo. A construção de *chatbots* envolve análise de palavras e contextos para criar uma comunicação entre o usuário e o *bot*. O processamento de linguagem natural ou NLP (*Natural Language Processing*) é o conjunto de técnicas que identifica padrões, intenções e

palavras chave que permitem com que o *chatbot* escolha a resposta adequada de acordo com seu tipo.

Este capítulo também apresentou diversas aplicações de *chatbots* em diversas áreas como entretenimento, serviço público, vendas a varejo, música, e, em especial a área médica, contexto no qual este trabalho está inserido.

No próximo capítulo serão apresentados alguns *frameworks* para a criação de *chatbots*, canais no qual um *chatbot* pode ser disponibilizado e serviços que oferecem processamento de linguagem natural.

Capítulo 3

Tecnologias utilizadas na construção de chatbots

3.1 Frameworks

A criação de chatbots pode ser facilitada com a utilização de *frameworks* desenvolvidos para este fim. Estes *frameworks* integram interfaces para criação de fluxos, processamento de linguagem natural e outras ferramentas, em uma única solução, além de oferecerem suporte para diversos canais de comunicação.

3.1.1 Chatfuel

*Chatfuel*¹² 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 Messenger* 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. *LEGO*, *Coca Cola*, *Adidas*, *Green Peace* e *Nissan* criaram *chatbots* utilizando a plataforma.

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

¹²Disponível em <https://chatfuel.com/>

repositório do NPM (*Node Package Manager*), o *Botpress* é focado na simplicidade e na intuição para o desenvolvimento de *chatbots*. Ele oferece 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.

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).

Single Choice

Suggest choices to the user with the intention of picking only one (with an optional message)

Message

Choices*

Protip: To prevent an element from being rendered on the channel, prefix either the Title or the Value with `!hide`

Message* ↑ ↓ ×

The title of the choice (this is what gets shown to the user)

Value*

The value that your bot gets when the user picks this choice (usually hidden from the user)

Message* ↑ ↓ ×

The title of the choice (this is what gets shown to the user)

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.

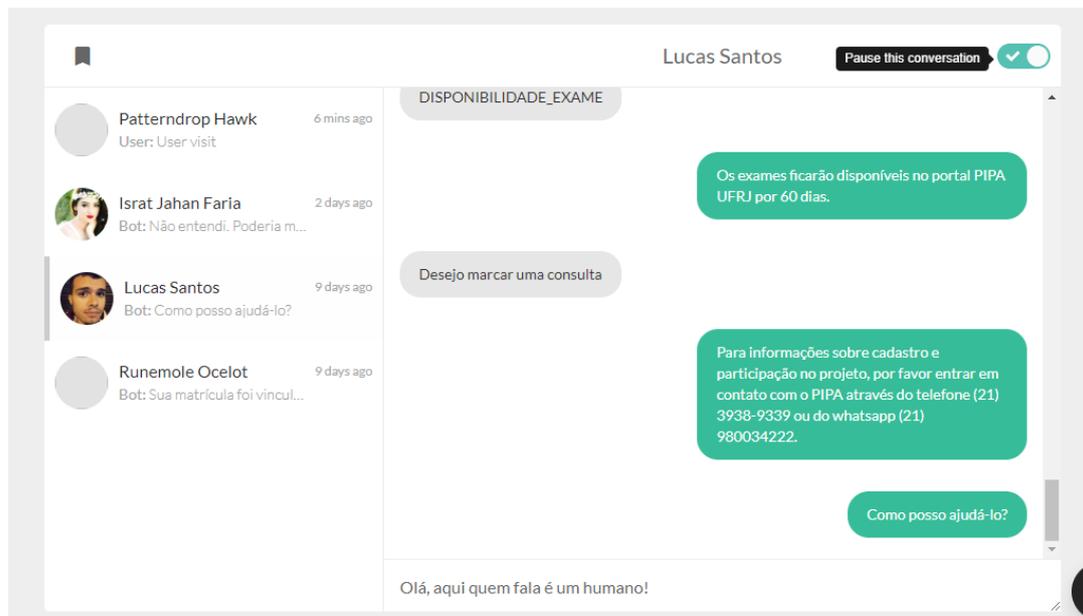


Figura 3.2: *HITL* - Interface de troca de mensagens no painel do *Botpress*

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.

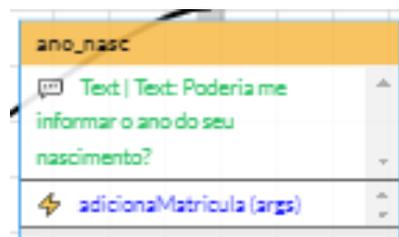


Figura 3.3: *Actions* - Bloco com uma função *javascript* que é executada após o envio da mensagem.

3.2 Canais

3.2.1 Facebook Messenger

É o aplicativo de mensagens desenvolvido originalmente para ser a plataforma de *chat* do *Facebook*¹³. 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 então¹⁴. 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 que ele envie mensagens de forma proativa.

3.2.2 Telegram

Aplicativo de troca de mensagens multiplataforma, que pode ser acessado através de *smartphones*, computadores, ou através de uma interface *web*¹⁵. 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 uma API (Application Program Interface) 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 controla 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 e informações para desenvolver algum tipo de trabalho¹⁶. Ela é largamente usada 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

¹³<https://www.messenger.com>

¹⁴<https://techcrunch.com/2016/04/12/agents-on-messenger/>

¹⁵<https://telegram.org/>

¹⁶<https://slack.com/>

do *Slack*. Inclusive, ao adicionarmos um *bot*, adicionamos uma integração - que é limitada pelo plano gratuito do *Slack*. A diferença é 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.

3.2.4 Web

Todo *website* também é capaz de hospedar um *chatbot*. A abordagem de exibir uma interface de troca de mensagens em *sites* é comum quando se trata de falar com o atendimento ao cliente, que em geral, é feito por humanos. Entretanto, é possível automatizar esse atendimento utilizando-se de uma arquitetura baseada em cliente-servidor. A interface do *bot* envia requisições para o servidor que hospeda o *bot*, e, ao receber uma resposta, exibe na tela do usuário.

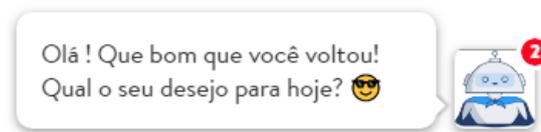


Figura 3.4: Chatbot do site Resultados Digitais

Um exemplo de site que apresenta um *chatbot* é o site da empresa Resultados Digitais¹⁷ (figura 3.4), que atua no setor de *marketing* digital.

3.3 Provedores de inteligência artificial e NLP

Existem hoje no mercado alguns serviços especializados em inteligência artificial que se integram com serviços de *chatbot*, por muita das vezes oferecendo resultados melhores no processamento de linguagem natural. Nesta seção serão apresentados alguns desses serviços.

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

¹⁷<https://resultadosdigitais.com.br>

de software como aplicativos, *wearables* e *bots*¹⁸. Suporta cerca de 50 linguagens diferentes e não existe um limite de chamadas de API¹⁹.

3.3.2 IBM Watson

IBM Watson é uma suíte completa de ferramentas voltadas para inteligência artificial. Desenvolvida pela IBM (*International Business Machines Corporation*), 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 mês²⁰.

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 (*internet of things*). 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 Service*²¹. Atualmente suporta cerca de 13 linguagens diferentes²² e é limitada a 10.000 chamadas de API gratuitas por mês²³.

3.3.4 Botpress

O *framework Botpress* apresentado na seção 3.1.2 além de suportar os provedores mencionados anteriormente, também possui um sistema de processamento de linguagem natural integrado²⁴, que possibilita o cadastro de *intents* e *entities*.

Ele tem uma unidade de NLP no mesmo servidor do *bot*, não possui limitação de requisições por janela de tempo, e é flexível quanto a implementação de componentes.

3.4 Síntese e Conclusão

Este capítulo apresentou algumas tecnologias que podem ser utilizadas para a criação de chatbots. As tecnologias foram divididas em três categorias: *frameworks*, canais e provedores de inteligência artificial.

Frameworks são um conjunto de ferramentas que integram as funcionalidades de um *chatbot*. Os canais são os meios no qual um *chatbot* pode ser disponibilizado para

¹⁸<https://wit.ai/>

¹⁹<https://wit.ai/faq>

²⁰<https://www.ibm.com/watson/how-to-build-a-chatbot>

²¹<https://www.luis.ai/home>

²²<https://docs.microsoft.com/pt-br/azure/cognitive-services/luis/luis-language-support>

²³<https://chatbotsmagazine.com/worlds-biggest-ai-engines-comparison-46e421413ab>

²⁴https://botpress.io/docs/10.50/getting_started/trivia_nlu/

o público, podendo ser um site ou até mesmo dentro de um aplicativo de conversas como o *Facebook Messenger*. Provedores de inteligência artificial e NLP são serviços próprios que possuem uma base treinada em diversos idiomas, que são capazes de classificar, em geral com melhores resultados, as características de um texto.

Para a construção do PipaBot foram utilizadas as tecnologias *Botpress* como *framework* de criação de *bots* e de processamento de linguagem natural. O *Facebook Messenger* foi escolhido como um dos canais onde o PipaBot é disponibilizado porque o *Facebook* é a segunda rede social mais utilizada no Brasil (59%), perdendo apenas para o *YouTube* (60%)²⁵. O *Facebook* possui cerca de 130 milhões de usuários ativos por mês e anualmente sua base de usuários cresce 7% ao ano. Cerca de 92% dos usuários do *Facebook* acessa a rede social através de aplicativos móveis. Por essa razão, o *Messenger* é visto como o canal mais acessível para a maioria dos usuários. Contudo, também é possível utilizar o PipaBot através do Portal PIPA atendendo àqueles usuários que não possuem acesso ao Facebook.

No próximo capítulo será apresentada a motivação e o contexto que levaram a construção do PipaBot, além de outras tecnologias que foram utilizadas no desenvolvimento. Além disso, será apresentado, também, o ambiente de desenvolvimento do sistema, a solução sistêmica proposta e a avaliação da tecnologia.

²⁵<https://exame.abril.com.br/negocios/dino/62-da-populacao-brasileira-esta-ativa-nas-redes-sociais/>

Capítulo 4

Desenvolvimento do PipaBot

4.1 Motivação e Contexto

Os *chatbots* são usados em diversas áreas, por exemplo na área médica[6][7][8]. Tendo em vista a necessidade do PIPA UFRJ de ter um canal de comunicação com os participantes do projeto, e uso efetivo de *chatbots* para este fim, o PipaBot foi proposto.

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).

4.2 Tecnologias utilizadas no PipaBot

Esta seção apresenta as tecnologias *Node.JS*, NPM e Git, que foram utilizadas ao longo do desenvolvimento do PipaBot.

4.2.1 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. No PipaBot, o *Node.js* é o componente que executa o *framework* Botpress, responsável pela construção dos fluxos de conversa e dos reconhecimentos de *intents*.

A principal característica que diferencia o *Node.js* de outras tecnologias *server-side* como o PHP (Hypertext Processor)- uma das linguagens de programação para web mais utilizada no mundo²⁶ - é o fato de sua execução ser *single-thread*, ou seja, apenas uma *thread* é responsável por executar a aplicação, enquanto que para 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.

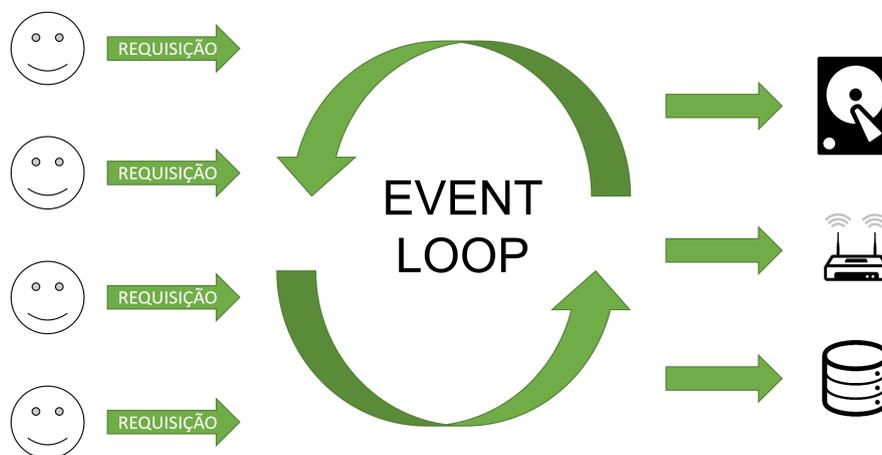


Figura 4.1: Arquitetura do NodeJS - Tratamento de múltiplas requisições

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

²⁶<https://www.devmedia.com.br/top-10-linguagens-de-programacao-mais-usadas-no-mercado/39635>

já realizadas serem concluídas para continuar sua execução (figura 4.1).

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 atualidade²⁷.

4.2.2 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, razão pelo qual é utilizado no PipaBot.

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.

4.2.3 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 ramos²⁸. Através do uso de *branches*, foi criado um *Git Flow* para o PipaBot, como apresentado na figura 4.2.

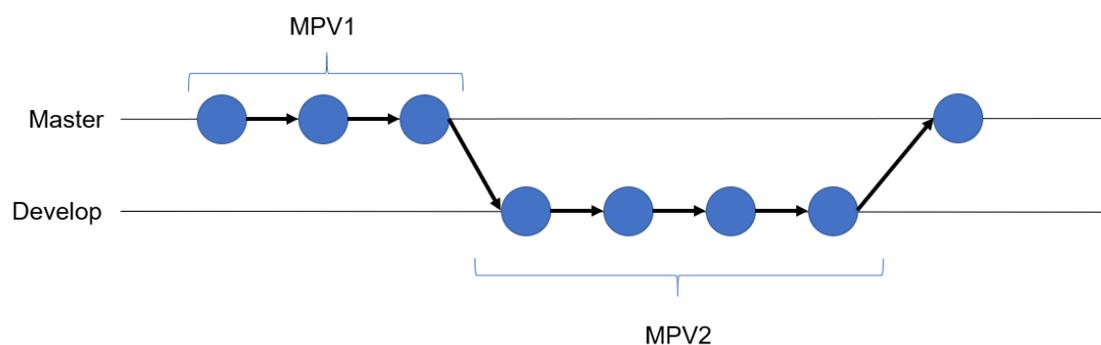


Figura 4.2: PipaBot - Organização do versionamento do código

Após o desenvolvimento da primeira versão do PipaBot, foi criada uma nova *branch* de desenvolvimento da próxima interação do ciclo Construir-Medir-Aprender.

²⁷<https://www.journaldev.com/7462/node-js-architecture-single-threaded-event-loop>

²⁸<https://git-scm.com/about>

Isso foi feito para que a *branch* principal do repositório (*master*) possuísse um produto pronto para ser executado. Ao término da segunda versão do MPV, foi feito um merge da *branch* de desenvolvimento com a *master*.

4.3 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 HTTP (*Hyper Text Transfer Protocol*) 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.

4.4 A solução PipaBot

4.4.1 Requisitos do Sistema

Os requisitos do sistema foram levantados através de reuniões com os pesquisadores do PIPA UFRJ - *stakeholders* do projeto. Suas expectativas foram analisadas através de critérios como viabilidade, privacidade e infraestrutura técnica já existente no projeto, e descritas em quatro requisitos funcionais e sete não funcionais, que caracterizaram, assim, o escopo do PipaBot. O documento de requisitos completo está disponível no apêndice A.

4.4.2 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 o aplicativo do *Messenger* do *Facebook* em *smartphones*, *tablets*, *laptops*, *desktop*, etc.

Enquanto que *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 participantes do PIPA com o *bot*. O esquema detalhado da arquitetura pode ser visto na figura 4.3.

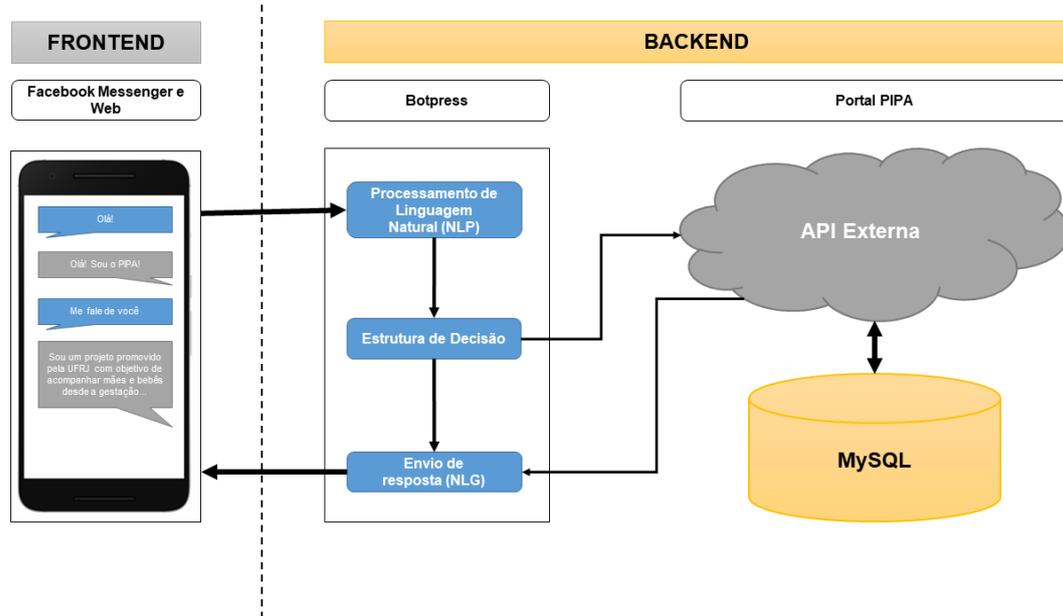


Figura 4.3: Arquitetura do PipaBot

O *core* do PipaBot utiliza o *framework* *Botpress*, por trazer, em uma única solução, funções como processamento de linguagem natural e criador de fluxos de conversa através de interface gráfica. Além disso, o *Botpress* também é capaz de classificar *intents* e suporta diversos canais (dentre os quais, o *Messenger* e *Web*, utilizados no projeto) através de um único ambiente. Por ser um *framework* desenvolvido em *NodeJS*, todo 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 do PIPA 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 participantes, que nem possuem permissão de executar tal ação. Então, foram criados três novos *endpoints*, que só podem ser acessados através de autenticação usando a 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 do participante no Portal - através do identificador cadastrado no *endpoint* (ii).

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

4.4.3 Documentação do PipaBot

A documentação do sistema que envolve o PipaBot é composta pelo documento de requisitos (Apêndice A), os modelos UML (*Unified Modeling Language*)[12] - diagrama de casos de uso (descrições) e diagrama de atividades - e documento de testes.

Diagrama de Casos de Uso

Os casos de uso do PipaBot (figura 4.4) representam principalmente a integração entre as tecnologias do *bot*, os canais (*Messenger* e *Web*) e o Portal PIPA.

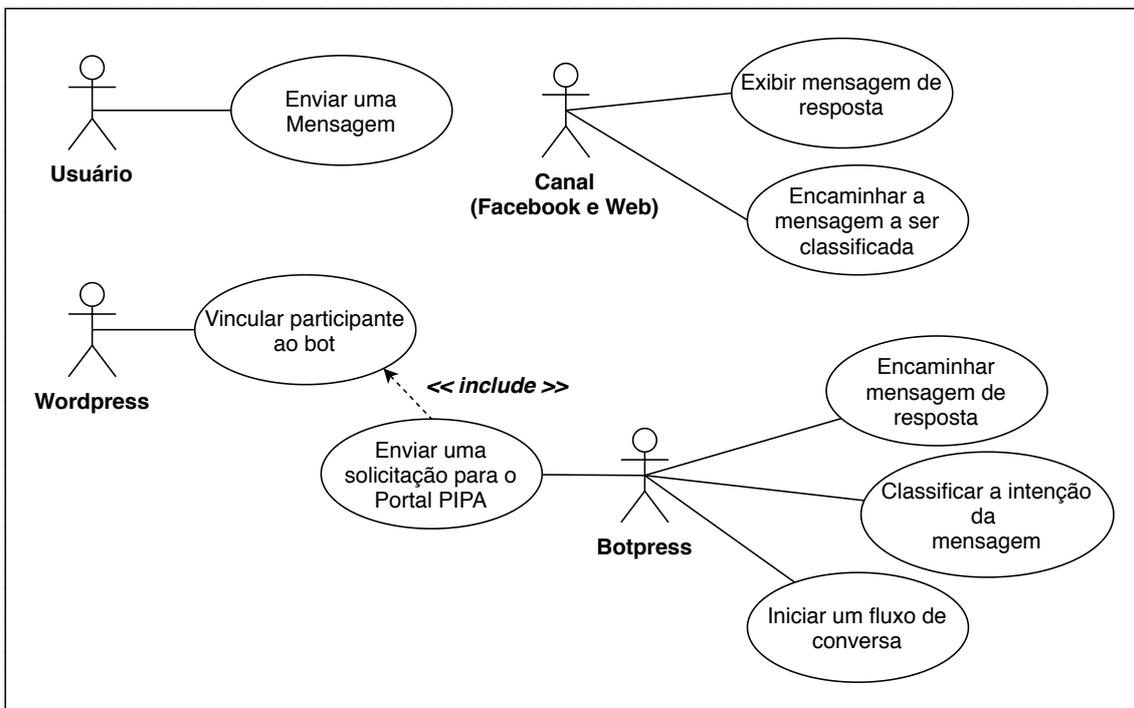


Figura 4.4: Diagrama de Casos de Uso do PipaBot

A descrição do caso de uso UC01 é apresentada na tabela 4.1 a título de exemplo. As descrições de todos os casos de uso é apresentada no Apêndice B.

Caso de Uso	UC01 - Enviar Mensagem
Atores	Usuário
Prioridade	Alta
Pré-condições	1. O usuário está conectado à internet. 2. O servidor do canal (<i>Messenger</i> ou <i>Web</i>) está <i>online</i>
Frequência de uso	Alta
Criticalidade	Alta
Condição de Entrada	O usuário enviou uma mensagem através de uma das interfaces do <i>bot</i> (<i>Messenger</i> ou <i>Web</i>)
Fluxo Principal	1. O usuário envia uma mensagem usando uma das interfaces do PipaBot 2. O canal recebe a mensagem
Fluxo Alternativo	
Pós-condições	A mensagem foi encaminhada para o canal (<i>Messenger</i> ou <i>Web</i>)
Regras de Negócio	

Tabela 4.1: Exemplo de Descrição de Caso de uso

Diagrama de Atividades

Buscando representar o comportamento dos diversos atores do sistema ao longo de uma conversa, foram criados diagramas de atividades para alguns fluxos de conversa (apêndice C). A figura 4.5 apresenta o diagrama para o fluxo de conversa sobre o PIPA.

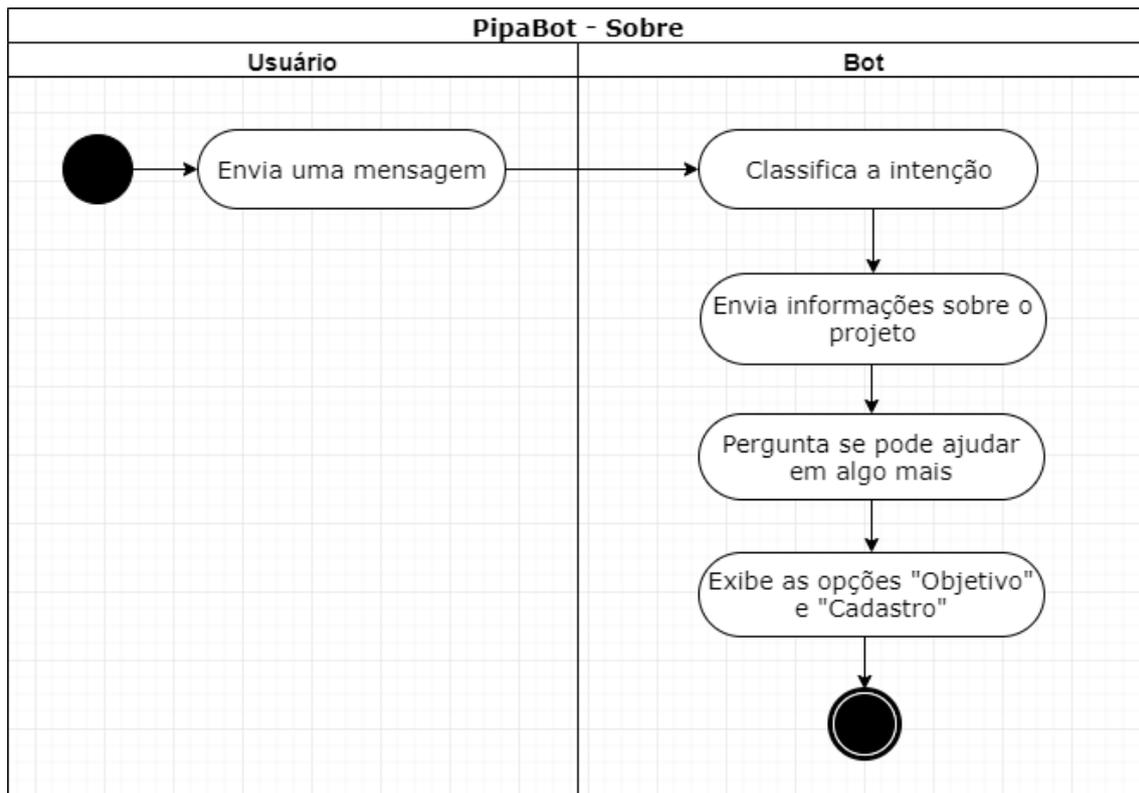


Figura 4.5: Exemplo de diagrama de atividades

4.4.4 MPV 1

Após a identificação do problema, os requisitos foram identificados e a primeira versão do PipaBot foi 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 (figura 4.6).

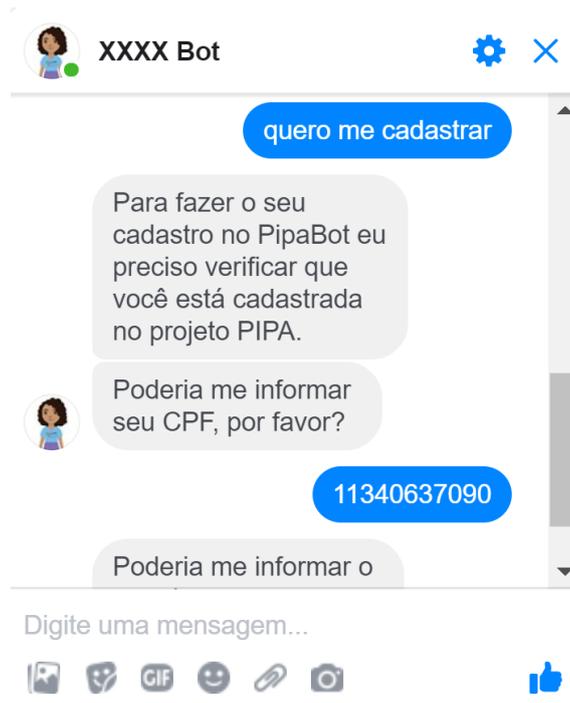


Figura 4.6: Fluxo de cadastro do participante PIPA para utilizar o *chatbot*.

O principal objetivo da primeira versão era apresentar aos pesquisadores do PIPA UFRJ o que um *chatbot* é capaz de fazer. Esta versão gerou novas ideias e novos objetivos para o próximo MPV.

4.4.5 MPV 2

A segunda versão do PipaBot apresenta diversas melhorias. Foram criados diálogos exclusivos para participantes 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* leva a um fluxo de conversa diferente cuja interação poderia ser feita não somente por botões, mas textualmente, dando mais naturalidade ao diálogo (figura 4.7).

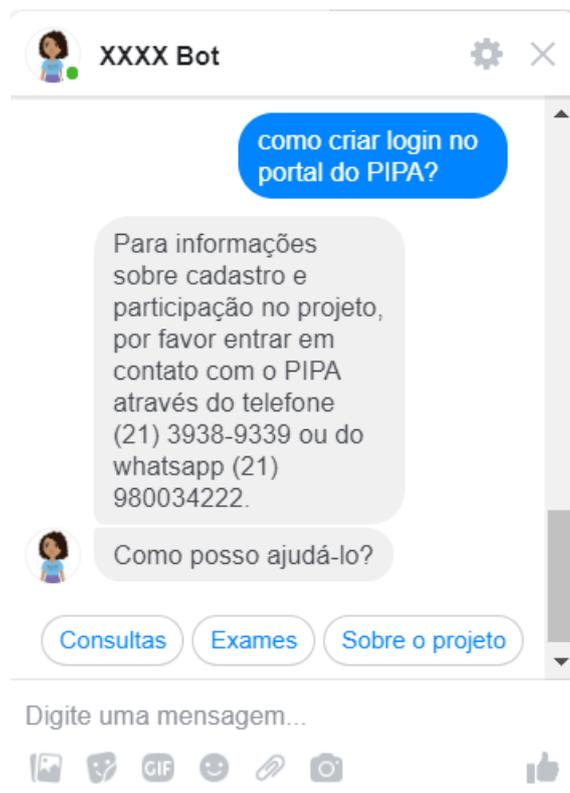


Figura 4.7: Fluxo de informações sobre o projeto, classificado através da tecnologia de NLP.

Todos os fluxos e opções de interação implementados no MPV 2 estão disponíveis em um vídeo de demonstração que pode ser acessado através da URL <https://www.youtube.com/watch?v=nybTVdPuJyw>.

4.4.6 Testes Funcionais

De modo a verificar o funcionamento dos fluxos e das integrações do PipaBot, foi realizado uma sequência de testes funcionais seguindo os cenários descritos na tabela 4.2.

Id	Teste	Resultado
01	Obter informações sobre o PIPA	Sucesso
02	Obter informações sobre como criar um login no portal do PIPA	Sucesso
03	Vincular um usuário do <i>chatbot</i> à sua conta no Portal do PIPA	Sucesso
04	Obter informações sobre exames	Sucesso
05	Obter informações sobre consultas	Sucesso

Tabela 4.2: Testes funcionais

Um exemplo de detalhamento de cenário de teste está apresentado na tabela 4.3. Os demais cenários são apresentados no apêndice D.

Criticidade	Alta
Localização	PipaBot (<i>Messenger</i> ou <i>Web</i>)
Objeto de Teste	Fluxo de informações sobre o PIPA
Caso de Teste	Obter informações sobre o PIPA
Pré-condição	O servidor do PipaBot e o canal (<i>Messenger</i> ou <i>Web</i>) estão online
Frequência de uso	Alta
Procedimento	1. Enviar uma mensagem para o PipaBot perguntando informações do projeto e/ou seus objetivos.
Resultado Esperado	Mensagens contendo informações sobre o projeto e/ou seus objetivos.

Tabela 4.3: Exemplo de teste funcional

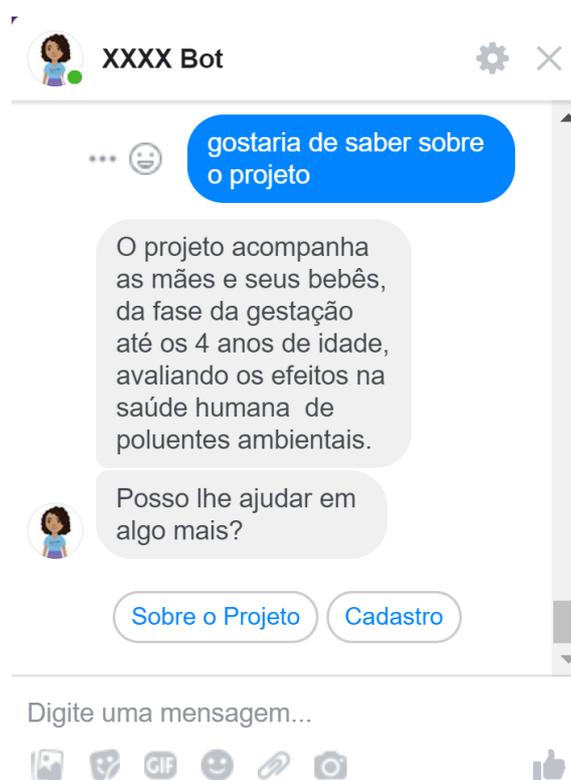


Figura 4.8: Resultado do Teste Funcional 1

4.4.7 Avaliação do PipaBot

Para avaliar o PipaBot foi utilizado o TAM (*Technology Acceptance Model*)[13] que procura determinar os aspectos de utilidade e facilidade de uso de tecnologias. Neste

trabalho **percepção de utilidade** é o quanto o usuário acredita que a tecnologia possa melhorar seu desempenho; e **percepção de facilidade de uso** é 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*)[14], 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.

Foram identificados dois principais grupos que participariam da avaliação do PipaBot: profissionais da saúde e os profissionais de computação. Com os grupos definidos, foram definidos dois objetivos para a avaliação do PipaBot, apresentados nas tabelas 4.4 e 4.5.

Analisar	O PipaBot
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 e saúde
No contexto de	Profissionais de computação e saúde realizando diálogos com o PipaBot

Tabela 4.4: GQM - Objetivo G1

Analisar	O PipaBot
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 e saúde
No contexto de	Profissionais de computação e saúde realizando diálogos com o PipaBot

Tabela 4.5: GQM - Objetivo G2

Foi elaborado um conjunto de seis questões abordando os objetivos G1 e G2, no que diz respeito a percepção de facilidade de uso e utilidade. Para capturar os dados foi utilizada uma escala *Likert*, no qual os participantes manifestam o seu nível de concordância com uma afirmação. As questões estão exibidas na tabela 4.6. Também foi disponibilizado um campo para que os participantes pudessem expressar seus comentários sobre cada questão. O formulário aplicado está apresentado no Apêndice E.

Identificador	Questão	Objetivo
Q1	O PipaBot é fácil de usar	G1
Q2	O PipaBot responde rapidamente aos meus questionamentos	G1
Q3	Eu posso obter a informação que desejo com poucas perguntas	G1
Q4	Os elementos de navegação (menus e botões) do PipaBot facilitam a interação ao longo da conversa	G1
Q5	O PipaBot fornece respostas pertinentes aos meus questionamentos	G2
Q6	O PipaBot fornece respostas adequadas aos meus questionamentos	G2

Tabela 4.6: GQM - Questões

Opções de Resposta
Concordo Totalmente
Concordo Parcialmente
Indiferente
Discordo Parcialmente
Discordo Totalmente

Tabela 4.7: GQM - Opções de resposta

Antes de iniciar a avaliação, os grupos receberam um roteiro (Apêndice F) que continha cinco cenários a serem testados:

1. Informações sobre o PIPA UFRJ
2. Cadastro no Portal PIPA UFRJ
3. Cadastro no PipaBot
4. Informações sobre exames
5. Informações sobre consultas médicas

Após a execução do roteiro, os participantes responderam as questões da tabela 4.6. O grupo de **profissionais de computação** foi composto por seis participantes do gênero masculino; quatro deles eram alunos de graduação (66,7%) e dois alunos de mestrado (33,3%); Cinco participantes possuíam faixa etária entre 21 e 30 anos e um entre 31 e 40 anos. O grupo de **profissionais da saúde** contou com sete participantes do gênero feminino; das sete, três eram alunas de graduação, três de mestrado e uma de doutorado. Com relação a faixa etária, uma participante possuía

menos de 21 anos, enquanto cinco participantes estavam entre 21 e 30 anos e uma estava entre 41 e 50 anos.

Os resultados obtidos²⁹ são apresentados nas tabelas 4.8 (profissionais de computação) e 4.10 (profissionais de saúde).

	Q1	Q2	Q3	Q4	Q5	Q6	Total
Concordo Totalmente	1	6	4	5	1	2	19
Concordo Parcialmente	5	0	2	1	5	4	17
Indiferente	-	-	-	0	-	-	0
Discordo Parcialmente	0	0	0		0	0	0
Discordo Totalmente	0	0	0	0	0	0	0

Tabela 4.8: Respostas do Grupo de Profissionais de Computação

Analisando os dados dos profissionais da computação, nota-se que as questões Q1 e Q5 tiveram o maior índice (83%) das respostas marcadas como "concordo parcialmente". Separando as respostas nas classificações Facilidade de Uso (Figura 4.9) - que compreende as questões de Q1 a Q4 e Utilidade (Figura 4.10) - Q5 e Q6 - podemos observar que o grupo ficou dividido entre as duas respostas mais positivas para o primeiro indicador, enquanto que 75% das avaliações indicam que os avaliadores não concordam totalmente com a utilidade do PipaBot.

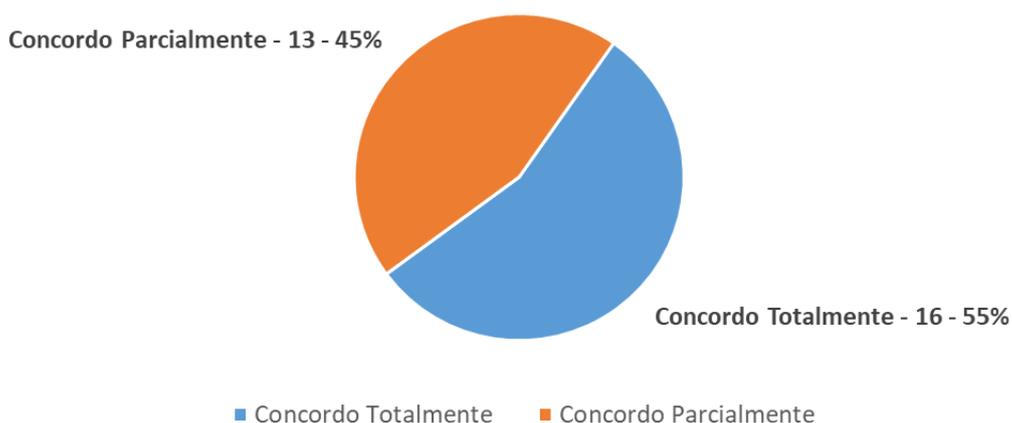


Figura 4.9: Avaliação de Facilidade de Uso - Grupo de Profissionais de Computação

²⁹ Algumas questões não possuíam a opção "indiferente".

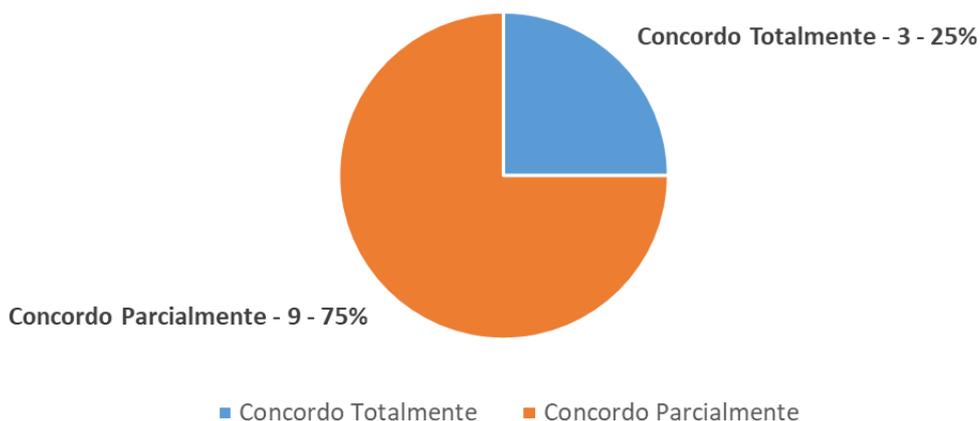


Figura 4.10: Avaliação de Utilidade - Grupo de Profissionais de Computação

A próxima etapa foi fazer a análise dos comentários (tabela 4.9). De maneira geral, a expectativa dos usuários no fluxo de consultas era poder fazer um agendamento de uma consulta ou obter uma lista de consultas marcadas, o que pode ter gerado certa frustração, justificando o grande número de respostas do tipo *Concordo Parcialmente*.

Questão	Comentário
Q1	"Especificamente a resposta sobre consultas que informa um número para ligar não é muito interessante para o usuário"
Q5	"Tirando a pergunta sobre consultas, acho que o resto ficou bem interessante"
Q5	"o legal seria integrar com os sistemas de consultas e exames e informar informações pelo próprio bot ao inves de responder com uma mensagem informado o usuario para ir em outro site"
Q5	"Não faz muito sentido o chat me direcionar para outro chat (whatsapp) para receber informações. Parece que perdi tempo. Poderia ser direcionado desde o início".

Tabela 4.9: Comentários - Profissionais de Computação

Em contrapartida, para o grupo de profissionais da saúde notamos que em todas as questões o indicador *Concordo Totalmente* foi maioria, representando, inclusive, 76% do total das respostas do questionário.

	Q1	Q2	Q3	Q4	Q5	Q6	Total
Concordo Totalmente	6	5	5	7	4	5	32
Concordo Parcialmente	1	2	2	0	3	2	10
Indiferente	-	-	-	0	-	-	0
Discordo Parcialmente	0	0	0	0	0	0	0
Discordo Totalmente	0	0	0	0	0	0	0

Tabela 4.10: Respostas do Grupo de Profissionais de Saúde

Fazendo a análise por classificação das questões, podemos ver na figura 4.11 que de maneira geral os pesquisadores do PIPA acharam a tecnologia fácil. Entretanto, quando olhamos para a utilidade (4.12), vemos que o número de participantes que concorda parcialmente sofreu um aumento.

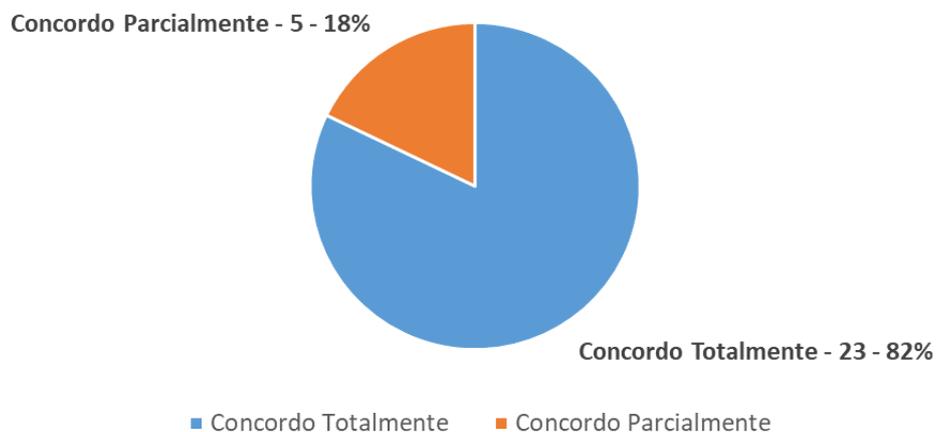


Figura 4.11: Avaliação de Facilidade de Uso - Grupo de Profissionais de Saúde

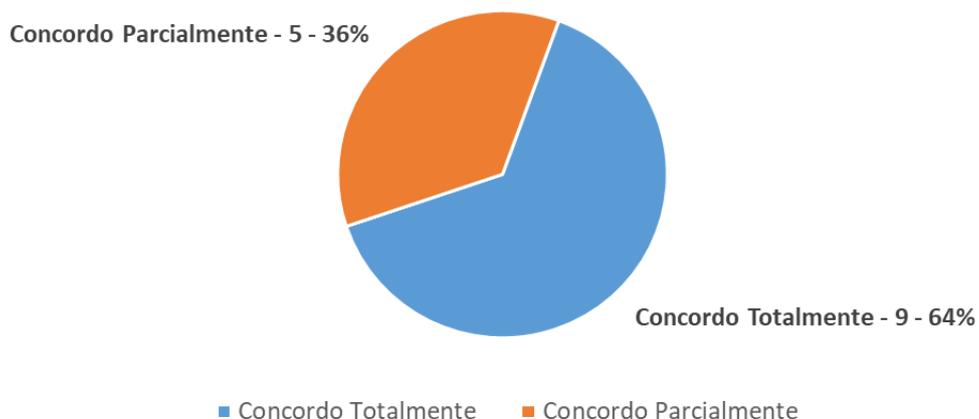


Figura 4.12: Avaliação de Utilidade - Grupo de Profissionais de Saúde

Com relação aos comentários feitos pelas profissionais da saúde (tabela 4.10), o principal ponto abordado foi a alimentação da base de conhecimento do PipaBot. Em geral, a expectativa era que houvessem outros fluxos de conversa, o que pode ter afetado a percepção de utilidade para a ferramenta.

Questão	Comentário
Q1	"Muito fácil de usar, com respostas claras. Porém, enriqueceria muito a adição de novas possibilidades de perguntas".
Q1	"Prático para informações básicas, mas limitado para algumas perguntas e me fornece dados sobre passo a passos sobre o que preciso fazer".
Q2	"Concordo, porém quando questiono sobre a importância da minha participação nesse projeto, o bot não tem uma resposta. Algumas mães ficam com certo "medo"de fornecer/participar"
Q2	"Necessita de mais alimentação no "banco de perguntas"para acessar respostas já existentes, mas funciona muito rápido".
Q5	"Poderiam existir mais explicações sobre o que é a Maternidade Escola, pois quando é feita essa pergunta o PipaBot diz que não entendeu a pergunta/dúvida".
Q5	"É uma questão de ajuste olhar as perguntas mais frequentes que ainda não tem resposta".
Q6	"O PipaBot sempre fornece informações, as vezes as respostas não são correspondentes as perguntas. Utilizar as respostas desse formulário como oportunidade de gerar novas questões"

Tabela 4.11: Comentários - Profissionais de Saúde

Considerações finais da avaliação

Ao olhar os resultados de ambos os grupos de forma conjunta, podemos concluir que os resultados são satisfatórios. Contudo, é importante ressaltar dois pontos:

1. O fato do grupo de profissionais da saúde ter tido maior índice de respostas **Concordo Totalmente** poderia representar uma ameaça à validade do estudo devido a expectativa depositada em uma implantação tecnológica no projeto. Para minimizar esse risco, a avaliação também foi executada com profissionais de computação, cujos resultados foram menos animadores.
2. Os cenários apresentados no roteiro eram de fácil entendimento e possivelmente não geraram frustrações. Portanto, este ponto também pode ser considerado uma ameaça à validade do estudo.

Como futuramente serão desenvolvidas novas ações no PipaBot para atender a demanda dos comentários, é importante que se realize uma nova avaliação da tecnologia.

4.5 Síntese e Conclusão

Este capítulo apresentou o contexto do PIPA - Projeto Infância e Poluentes Ambientais - no qual o PipaBot está inserido. Além disso, foram apresentadas algumas tecnologias de suporte e o ambiente de desenvolvimento do PipaBot.

A solução sistêmica proposta pelo PipaBot foi descrita através do documento de requisitos, diagrama de atividades e casos de uso e testes funcionais. A avaliação do PipaBot seguiu o Modelo de Aceitação de Tecnologia (TAM - *Technology Acceptance Model*), e foi executada por dois grupos de usuários: profissionais da saúde e profissionais de computação. Os resultados aprovaram a tecnologia, contudo foram apontados alguns pontos que podem comprometer a validade do estudo.

No próximo capítulo será feita a conclusão deste trabalho, apresentando as contribuições que foram feitas, as limitações da solução e potenciais melhorias a serem feitas nas próximas versões.

Capítulo 5

Conclusão

5.1 Contribuições

Implementar uma solução tecnológica no contexto do PIPA UFRJ permitiu avaliar como ferramentas com propósitos diferentes podem estar se comunicando e trocando dados entre si, ao conectarmos o Portal do PIPA ao PipaBot.

Além disso, para criar uma solução *chatbot* que atendesse também através de redes sociais, foi necessário avaliar uma série de fatores como segurança e privacidade da informação dos participantes. O Portal PIPA já permite visualização dos resultados de exames online. Entretanto, esse arquivo não deveria ser enviado através de uma conversa com o PipaBot, pelo fato de conter dados sensíveis do participante, o que violaria sua privacidade ao submeter tais informações para um servidor que não seja de propriedade do PIPA.

Por fim, este trabalho permitiu expandir os horizontes de todos os pesquisadores do PIPA, COPPE e POLI na busca por soluções tecnológicas multidisciplinares.

5.2 Limitações

As principais limitações do PipaBot são:

1. Fluxos que requerem acesso a fontes externas como consultas médicas são direcionados para um telefone de contato do PIPA UFRJ, pois o mesmo não possui um sistema de agendamento eletrônico no qual o PipaBot pudesse ser integrado.
2. A versão do *Messenger* possui uma limitação de chamadas de API imposta pelo *Facebook* que é definida por 200 chamadas de API por usuário único do *bot* por hora, não limitando o acesso de um usuário a 200 chamadas. Ou seja, se existem dez usuários únicos em uma janela de tempo de uma hora, significa que podem ser feitas até 2000 chamadas de API pela aplicação no total.

5.3 Trabalhos futuros

As próximas evoluções do PipaBot consistem em atender as principais demandas levantadas pelos avaliadores na seção 4.4.7 como (i) a criação de novos fluxos de conversa e (ii) a integração com um sistema de agendamento eletrônico por meio do qual os próprios usuários pudessem planejar suas consultas de forma muito mais rápida e direta.

Além disso, é de interesse dos pesquisadores do PIPA que o PipaBot seja capaz de enviar mensagens para os usuários de forma automática em casos de lembretes de consulta, por exemplo.

5.4 Considerações finais

Em um mundo cada vez mais acelerado, as pessoas buscam maneiras mais rápidas de se obter respostas para diversas questões do dia a dia. *Chatbots* mostram-se capazes de ser a solução sistêmica para essa situação por apresentar uma interface mais amigável, todas as funcionalidades que eram encontradas em sites e aplicativos convencionais. Contudo, é importante ressaltar que, ao utilizar-se de plataformas de terceiros há alguns pontos a serem levados em consideração.

Canais como *Facebook* armazenam arquivos e dados de mensagens trocadas pelos usuários; dessa forma, a troca de informações sensíveis, como foi o caso do PipaBot em seu fluxo de exames, deve ser contornada de maneira que apenas o responsável pelo exame (PIPA) e o dono da informação (participante) tenham acesso.

Independente da questão da privacidade, implementar um *chatbot* através de redes sociais é a melhor opção devido a grande aderência do público e a fácil acessibilidade aos aplicativos dessas redes (*Messenger*, *Telegram* e *WhatsApp*, por exemplo). Acredita-se, portanto, que o desafio maior seja identificar os pontos de comprometimento da privacidade.

Além disso, uma outra discussão importante que deve ser considerada ao utilizar um *chatbot* deve ser o quanto pode-se acreditar que um *bot* poderia substituir um atendimento humano. Creditar a um robô toda a confiança para efetuar vendas, fazer atendimentos, entregar exames, fazer lembretes, seja lá qual for a atividade, é acreditar que a inteligência por trás do sistema já conhece todas as possibilidades do negócio. A inteligência artificial hoje é sim capaz de se readaptar e ir aprendendo conforme seu uso. Entretanto, é preciso rever as necessidades e os cenários em que um *chatbot*, de fato, faça sentido.

Referências Bibliográficas

- [1] CAROLI, P. *Direto ao ponto: criando produtos de forma enxuta*. Editora Casa do Código, 2015.
- [2] AQUINO, V. H., ADANIYA, M. H. “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>>.
- [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] HELLER, B., PROCTOR, M., MAH, D., et al. “Freudbot: An investigation of chatbot technology in distance education”. In: *EdMedia+ Innovate Learning*, pp. 3913–3918. Association for the Advancement of Computing in Education (AACE), 2005.
- [5] PRADANA, A., SING, G. O., KUMAR, Y. “SamBot-Intelligent Conversational Bot for Interactive Marketing with Consumer-centric Approach”, *International Journal of Computer Information Systems and Industrial Management Applications*. ISSN, pp. 2150–7988, 2014.
- [6] FOSSATTI, M. C., RABELLO, R. D. S., MARCHI, A. C. B. D. “AGEbot: um chatterbot em AIML voltado para responder questões sobre Epilepsia”. In: *Anais do XXXI Congresso da Sociedade Brasileira de Computação*. sn, 2011.
- [7] 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>.

- [8] 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>>.
- [9] PFLEEGER, S. L. *Engenharia de software: teoria e prática*. Prentice Hall, 2004.
- [10] 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.
- [11] FADHIL, A., GABRIELLI, S. “Addressing challenges in promoting healthy lifestyles: the al-chatbot approach”. In: *Proceedings of the 11th EAI international conference on pervasive computing Technologies for Healthcare*, pp. 261–265. ACM, 2017.
- [12] BEZERRA, E. *Princípios de Análise e Projeto de Sistema com UML*, v. 3. Elsevier Brasil, 2016.
- [13] 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.
- [14] VAN SOLINGEN, R., BASILI, V., CALDIERA, G., et al. “Goal question metric (gqm) approach”, *Encyclopedia of software engineering*, 2002.

Apêndice A

Documento de Requisitos

Universidade Federal do Rio de Janeiro
Engenharia de Computação e Informação
Lucas Santos de Paula <lucasdepaula@poli.ufrj.br>
Taisa Guidini Gonçalves <taisa@cos.ufrj.br>
Guilherme Horta Travassos <ght@cos.ufrj.br>

Documento de Requisitos

Chatbot PIPA

Universidade Federal do Rio de Janeiro	Versão: 3.0
Especificação de Requisitos de Software	Data: 10/04/2019
Chatbot PIPA	

Histórico de Revisões

Data	Versão	Descrição	Autor
02/02/2019	1.0	Inclusão dos Requisitos funcionais	Lucas Santos de Paula
10/04/2019	2.0	Revisão dos requisitos e inclusão dos novos requisitos levantados	Lucas Santos de Paula
01/07/2019	3.0	Revisão dos requisitos não funcionais e mapeamento dos casos de uso.	Lucas Santos de Paula

Universidade Federal do Rio de Janeiro	Versão: 3.0
Especificação de Requisitos de Software	Data: 10/04/2019
Chatbot PIPA	

Tabela de Conteúdo

Histórico de Revisões	1
Tabela de Conteúdo	2
1. Introdução	3
1.1 Propósito	3
1.2 Escopo	3
1.3 Público-Alvo	3
1.4 Definições, Acrônimos e Abreviações	3
1.5 Referências	3
2. Visão Geral do Sistema	4
2.1 Classes e característica dos usuários	4
2.2 Premissas	4
2.3 Restrições	4
3. Requisitos Funcionais	4
4. Requisitos Não-Funcionais	4
4.1 Usabilidade	4
4.2 Confiabilidade	5
4.3 Desempenho	5
4.4 Reusabilidade	5
4.5 Segurança	5
4.6 Acessibilidade	5
9. Mapeamento de Requisitos com Casos de Uso	5

Universidade Federal do Rio de Janeiro	Versão: 3.0
Especificação de Requisitos de Software	Data: 10/04/2019
Chatbot PIPA	

Especificação de Requisitos do Chatbot

1. Introdução

1.1 Propósito

Esse documento especifica os requisitos contemplados pelo PipaBot, que integrará o sistema de atendimento aos participantes do projeto, fornecendo um canal alternativo de comunicação do paciente com o projeto, melhorando o acompanhamento das atividades diárias.

1.2 Escopo

Esse documento descreve os requisitos funcionais e não funcionais do PipaBot, que permite com que usuários comuns tenham acesso a informações públicas do projeto como objetivos e telefones, e uma participante do projeto tenha acesso a facilidades como resultado de exames e informações para marcações de consultas desde que sua conta no cadastro do PIPA esteja vinculada ao chat.

1.3 Público-Alvo

Mães, desde a fase da gestação até os quatro anos de idade do filho e demais usuários do Facebook Messenger.

1.4 Definições, Acrônimos e Abreviações

PIPA - Projeto Infância e Poluentes Ambientais
SSL - Secure Sockets Layer

1.5 Referências

<http://www.projetopipaufjrj.iesc.ufrj.br/pt/>

Universidade Federal do Rio de Janeiro	Versão: 3.0
Especificação de Requisitos de Software	Data: 10/04/2019
Chatbot PIPA	

2. Visão Geral do Sistema

2.1 Classes e característica dos usuários

A ferramenta pode ser utilizada por qualquer usuário do Facebook Messenger para buscar informações sobre o projeto.

Mães pré-cadastradas no sistema do PIPA podem vincular sua conta ao Facebook Messenger ou pelo portal do PIPA para terem acesso aos canais de resultados de exames e de consultas.

2.2 Premissas

Existe uma base de usuários na qual o chatbot do Facebook Messenger e do portal podem autenticar os usuários que conversam com ele.

2.3 Restrições

O chatbot não pode ter acesso a informações sensíveis dos usuários como exames, por exemplo.

3. Requisitos Funcionais

RF1 - O chatbot deve ser capaz de fornecer informações sobre o projeto e seus objetivos.

RF2 - O chatbot deve permitir que um usuário do Facebook Messenger ou da versão web se autentique com seu cadastro do Portal do PIPA.

RF3 - O chatbot deve dar informações de consultas apenas para usuários cadastrados

RF4 - O chatbot deve dar informações sobre exames apenas para usuários cadastrados, direcionando-os para a página de login do site do PIPA.

4. Requisitos Não-Funcionais

4.1 Usabilidade

RNF1 - O chatbot deve ser capaz de se comunicar de forma simples e intuitiva, oferecendo opções de interface como botões e menus para o usuário.

Universidade Federal do Rio de Janeiro	Versão: 3.0
Especificação de Requisitos de Software	Data: 10/04/2019
Chatbot PIPA	

4.2 Confiabilidade

RNF2 - O chatbot deve ser capaz de classificar as mensagens que ele recebe, para entender qual é a intenção do usuário, e então responder de maneira adequada.

4.3 Desempenho

RNF3 - O tempo de resposta das mensagens enviadas ao PipaBot deve ser descrito, no máximo, na ordem de segundos.

4.4 Reusabilidade

RNF4 - A base de usuários que são utilizados para o cadastro do chatbot é a base utilizada pelo Wordpress no Portal PIPA

4.5 Segurança

RNF5 - O chatbot precisa de um certificado SSL para integrar-se ao Facebook, a fim de garantir a troca segura de mensagens.

RNF6 - O chatbot não deve ter acesso a informações sensíveis do usuário, como os arquivos dos exames.

4.6 Acessibilidade

RNF7 - Deve ser possível se comunicar com o PipaBot utilizando os mecanismos de acessibilidade disponíveis no aplicativo do Messenger ou no browser.

Universidade Federal do Rio de Janeiro	Versão: 3.0
Especificação de Requisitos de Software	Data: 10/04/2019
Chatbot PIPA	

9. Mapeamento de Requisitos com Casos de Uso

Identificação	RF01		
Casos de uso relacionados	UC01 - Enviar uma Mensagem UC02 - Enviar mensagem de resposta UC03 - Encaminhar a mensagem a ser classificada UC04 - Encaminhar mensagem de resposta UC05 - Classificar a intenção da mensagem UC06 - Iniciar fluxo de conversa		
Descrição	Ao pedir uma informação sobre o projeto, o PipaBot deve responder com as informações sobre o funcionamento do Projeto. Caso a informação seja sobre o objetivo do projeto, o mesmo deve ser informado.		
Prioridade	<input checked="" type="checkbox"/> Essencial	<input type="checkbox"/> Importante	<input type="checkbox"/> Desejável

Identificação	RF02		
Casos de uso relacionados	UC01 - Enviar uma Mensagem UC02 - Enviar mensagem de resposta UC03 - Encaminhar a mensagem a ser classificada UC04 - Encaminhar mensagem de resposta UC05 - Classificar a intenção da mensagem UC06 - Iniciar fluxo de conversa UC07 - Enviar uma solicitação para o Portal PIPA UC08 - Vincular participante ao bot		
Descrição	O usuário deve ser capaz de se identificar como participante do projeto		
Prioridade	<input checked="" type="checkbox"/> Essencial	<input type="checkbox"/> Importante	<input type="checkbox"/> Desejável

Universidade Federal do Rio de Janeiro	Versão: 3.0
Especificação de Requisitos de Software	Data: 10/04/2019
Chatbot PIPA	

Identificação	RF03		
Casos de uso relacionados	UC01 - Enviar uma Mensagem UC02 - Enviar mensagem de resposta UC03 - Encaminhar a mensagem a ser classificada UC04 - Encaminhar mensagem de resposta UC05 - Classificar a intenção da mensagem UC06 - Iniciar fluxo de conversa		
Descrição	Apenas usuários que realizaram a identificação do RF02 podem ter acesso ao telefone de marcação de consultas. Caso o usuário não tenha sido identificado ainda, o bot deverá guiá-lo no processo de login.		
Prioridade	<input checked="" type="checkbox"/> Essencial	<input type="checkbox"/> Importante	<input type="checkbox"/> Desejável

Identificação	RF04		
Casos de uso relacionados	UC01 - Enviar uma Mensagem UC02 - Enviar mensagem de resposta UC03 - Encaminhar a mensagem a ser classificada UC04 - Encaminhar mensagem de resposta UC05 - Classificar a intenção da mensagem UC06 - Iniciar fluxo de conversa		
Descrição	.Apenas usuários que realizaram a identificação do RF02 podem ter acesso às informações que o bot fornece a respeito de exames, tais como link para acesso online, disponibilidade e resultados impressos.		
Prioridade	<input checked="" type="checkbox"/> Essencial	<input type="checkbox"/> Importante	<input type="checkbox"/> Desejável

Apêndice B

Descrição dos Casos de Uso

Caso de Uso	UC01 - Enviar Mensagem
Atores	Usuário
Prioridade	Alta
Pré-condições	1. O usuário está conectado à internet. 2. O servidor do canal (<i>Messenger</i> ou <i>Web</i>) está <i>online</i>
Frequência de uso	Alta
Criticalidade	Alta
Condição de Entrada	O usuário enviou uma mensagem através de uma das interfaces do <i>bot</i> (<i>Messenger</i> ou <i>Web</i>)
Fluxo Principal	1. O usuário envia uma mensagem usando uma das interfaces do PipaBot 2. O canal recebe a mensagem
Fluxo Alternativo	
Pós-condições	A mensagem foi encaminhada para o canal (<i>Messenger</i> ou <i>Web</i>)
Regras de Negócio	

Tabela B.1: Caso de Uso 1 - Enviar Mensagem

Caso de Uso	UC02 - Enviar mensagem de resposta
Atores	Canal (<i>Facebook</i> e <i>Web</i>)
Prioridade	Alta
Pré-condições	
Frequência de uso	Alta
Criticalidade	Alta
Condição de Entrada	O canal recebeu uma mensagem do <i>Botpress</i> que deve ser encaminhada para o usuário
Fluxo Principal	1. O canal recebe a mensagem de resposta do <i>Botpress</i> 2. O canal encaminha a mensagem para o usuário
Fluxo Alternativo	
Pós-condições	
Regras de Negócio	

Tabela B.2: Caso de Uso 2 - Enviar mensagem de resposta

Caso de Uso	UC03 - Encaminhar a mensagem a ser classificada
Atores	Canal (<i>Facebook</i> ou <i>Web</i>)
Prioridade	Alta
Pré-condições	[<i>Facebook Messenger</i>] - ter configurado o app no painel do <i>Facebook for Developers</i> , indicando o <i>host</i> do <i>bot</i> .
Frequência de uso	Alta
Criticalidade	Alta
Condição de Entrada	O canal recebeu uma mensagem do usuário com destino ao <i>PipaBot</i>
Fluxo Principal	1. O canal recebe uma mensagem do usuário 2. O canal encaminha a mensagem para o <i>Botpress</i>
Fluxo Alternativo	
Pós-condições	
Regras de Negócio	

Tabela B.3: Caso de Uso 3 - Encaminhar a mensagem a ser classificada

Caso de Uso	UC04 - Encaminhar mensagem de resposta
Atores	<i>Botpress</i>
Prioridade	Alta
Pré-condições	[<i>Facebook Messenger</i>] - ter preenchido o arquivo de configuração do <i>Messenger</i> com as informações de <i>appSecret</i> , <i>ApplicationId</i> , <i>accessToken</i> , <i>verifyToken</i> e <i>hostname</i>
Frequência de uso	Alta
Criticalidade	Alta
Condição de Entrada	O <i>Botpress</i> escolheu a mensagem de resposta mais adequada
Fluxo Principal	<ol style="list-style-type: none"> 1. O <i>Botpress</i> escolhe a mensagem de resposta 2. O <i>Botpress</i> encaminha a mensagem endereçada ao usuário para o canal pertinente
Fluxo Alternativo	
Pós-condições	
Regras de Negócio	

Tabela B.4: Caso de Uso 4 - Encaminhar mensagem de resposta

Caso de Uso	UC05 - Classificar a intenção da mensagem
Atores	<i>Botpress</i>
Prioridade	Alta
Pré-condições	Ter cadastrado, através do painel de administração do <i>Botpress</i> , uma base de <i>intents</i> os quais o <i>PipaBot</i> deve reconhecer
Frequência de uso	Alta
Criticalidade	Alta
Condição de Entrada	O <i>Botpress</i> recebeu uma mensagem de um canal
Fluxo Principal	<ol style="list-style-type: none"> 1. O <i>Botpress</i> recebe uma mensagem para classificar 2. O <i>Botpress</i> classifica a mensagem de acordo com sua base de conhecimento 3. O <i>Botpress</i> encaminha a mensagem para seu núcleo de estrutura de decisão
Fluxo Alternativo	
Pós-condições	
Regras de Negócio	

Tabela B.5: Caso de Uso 5 - Classificar a intenção da mensagem

Caso de Uso	UC06 - Iniciar um fluxo de conversa
Atores	<i>Botpress</i>
Prioridade	Alta
Pré-condições	Verificar se o usuário do <i>chatbot</i> está vinculado a algum participante do PIPA
Frequência de uso	Alta
Criticalidade	Alta
Condição de Entrada	O <i>Botpress</i> classificou uma mensagem com um <i>intent</i> que é reconhecido pela base de questões
Fluxo Principal	<ol style="list-style-type: none"> 1. O núcleo de decisão do <i>Botpress</i> recebe uma mensagem classificada 2. O núcleo busca o fluxo adequado para o contexto 3. O núcleo encaminha o fluxo para a unidade de Envio de Respostas
Fluxo Alternativo	
Pós-condições	
Regras de Negócio	

Tabela B.6: Caso de Uso 6 - Iniciar um fluxo de conversa

Caso de Uso	UC07 - Enviar uma solicitação para o Portal PIPA
Atores	<i>Botpress</i>
Prioridade	Média
Pré-condições	
Frequência de uso	Eventual
Criticalidade	Média
Condição de Entrada	o <i>Botpress</i> está executando o fluxo de cadastro no PipaBot
Fluxo Principal	<ol style="list-style-type: none"> 1. O fluxo da conversa verifica se o usuário já está cadastrado [FA1] 2. O <i>Botpress</i> solicita informações de CPF e ano de nascimento do usuário. 3. O <i>Botpress</i> solicita ao <i>Wordpress</i> a verificação de existência de um determinado participante através de seu CPF. [FA2] 4. O <i>Wordpress</i> retorna os dados do participante. 5. O <i>Botpress</i> compara o ano de nascimento informado pelo usuário e o retornado pelo <i>Wordpress</i> [FA3]. 6. O <i>Botpress</i> pergunta ao usuário se ele deseja confirmar o cadastro.
Fluxo Alternativo	<p>[FA1]</p> <ol style="list-style-type: none"> 1. O <i>Botpress</i> identifica que o usuário já está cadastrado. 2. O <i>Botpress</i> então envia uma mensagem para o usuário informando que ele já está cadastrado no sistema. <p>[FA2]</p> <ol style="list-style-type: none"> 1. O <i>Wordpress</i> não encontra o usuário solicitado 2. O <i>Botpress</i> informa que o usuário não foi encontrado na base de usuários. <p>[FA3]</p> <ol style="list-style-type: none"> 1. O <i>Botpress</i> informa que o usuário não foi encontrado na base de usuários.
Pós-condições	O <i>Botpress</i> aguarda uma confirmação do usuário.
Regras de Negócio	

Tabela B.7: Caso de Uso 7 - Enviar uma solicitação para o Portal PIPA

Caso de Uso	UC08 - Vincular participante ao <i>bot</i>
Atores	<i>Wordpress</i>
Prioridade	Média
Pré-condições	O <i>Botpress</i> aguarda uma confirmação do usuário para efetivar o cadastro.
Frequência de uso	Eventual
Criticalidade	Média
Condição de Entrada	Foi recebida uma requisição de cadastro
Fluxo Principal	<ol style="list-style-type: none"> 1. O <i>Botpress</i> recebe a confirmação do usuário para prosseguir com o cadastro [FA1] 2. O <i>Botpress</i> envia a requisição de confirmação para o <i>Wordpress</i> 3. O <i>Wordpress</i> recebe uma requisição de criação de vínculo 4. O <i>Wordpress</i>, então, adiciona o ID do <i>chatbot</i> ao cadastro do paciente 5. o <i>Botpress</i> salva o status de login do usuário
Fluxo Alternativo	<p>[FA1]</p> <ol style="list-style-type: none"> 1. O <i>Botpress</i> envia uma mensagem para usuário informando que o cadastro não foi realizado.
Pós-condições	<ol style="list-style-type: none"> 1. O registro do usuário na base do PIPA passa a guardar o identificador do usuário do <i>chatbot</i>. 2. O <i>Botpress</i> salva em seu estado o <i>status</i> de login do usuário.
Regras de Negócio	

Tabela B.8: Caso de Uso 8 - Vincular participante ao *bot*

Apêndice C

Diagramas de Atividades

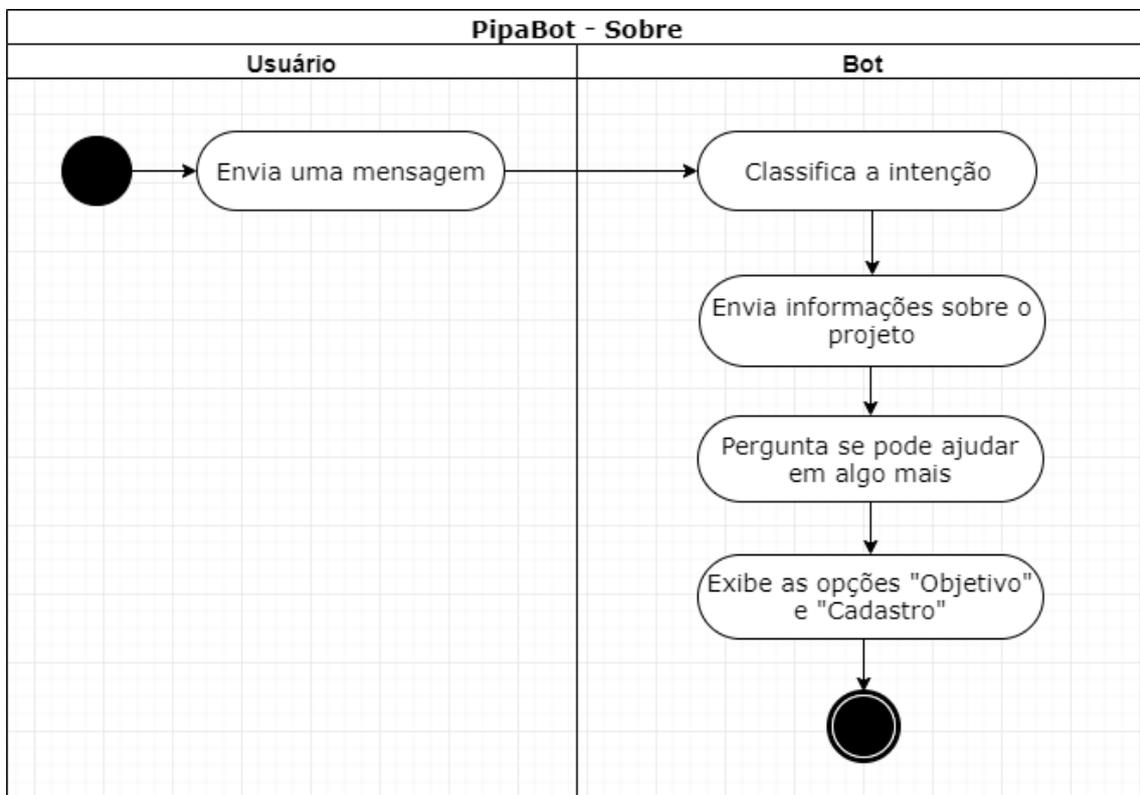


Figura C.1: Diagrama de atividades para o fluxo de conversa Sobre o Projeto

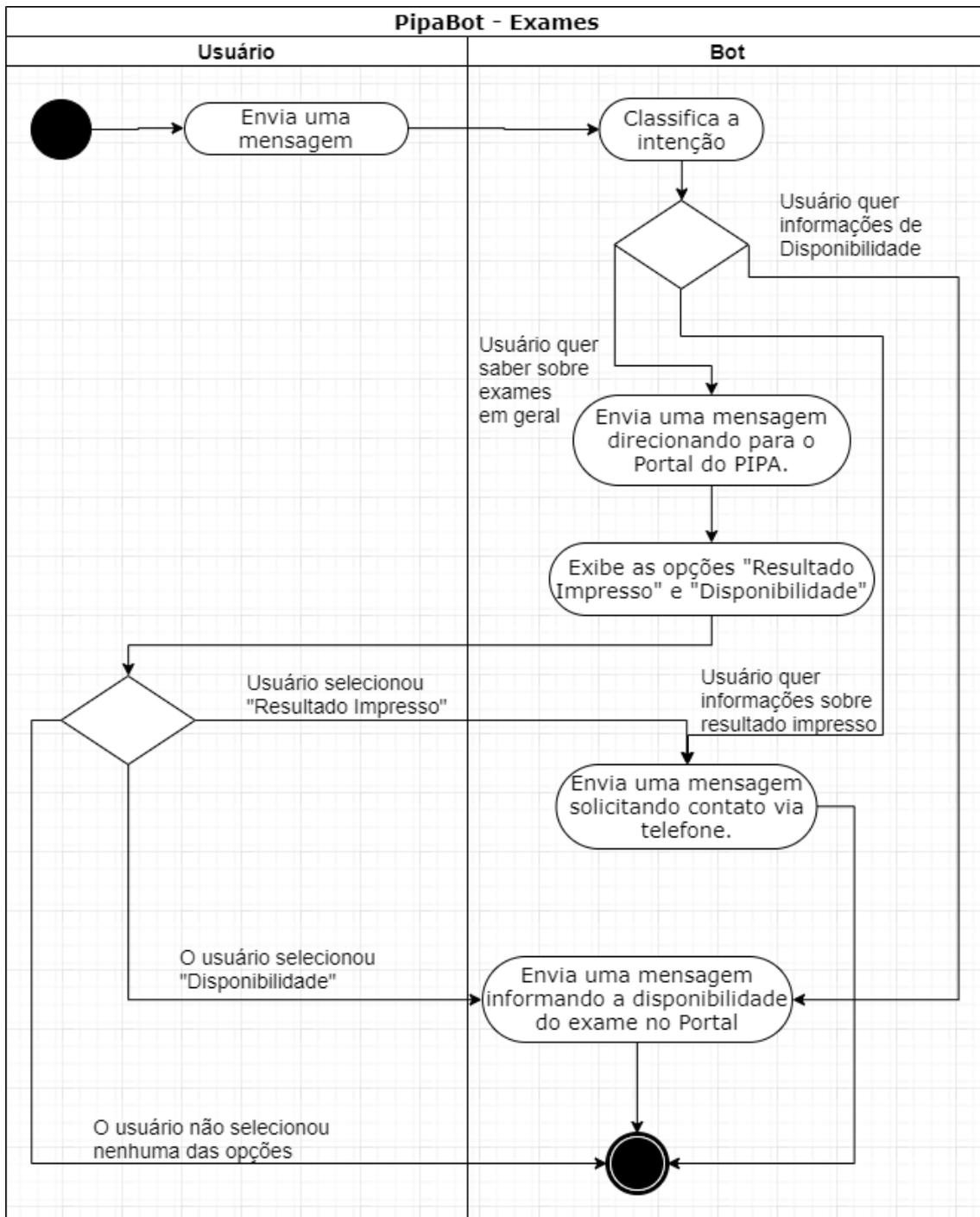


Figura C.2: Diagrama de atividades para o fluxo de conversa de Exames

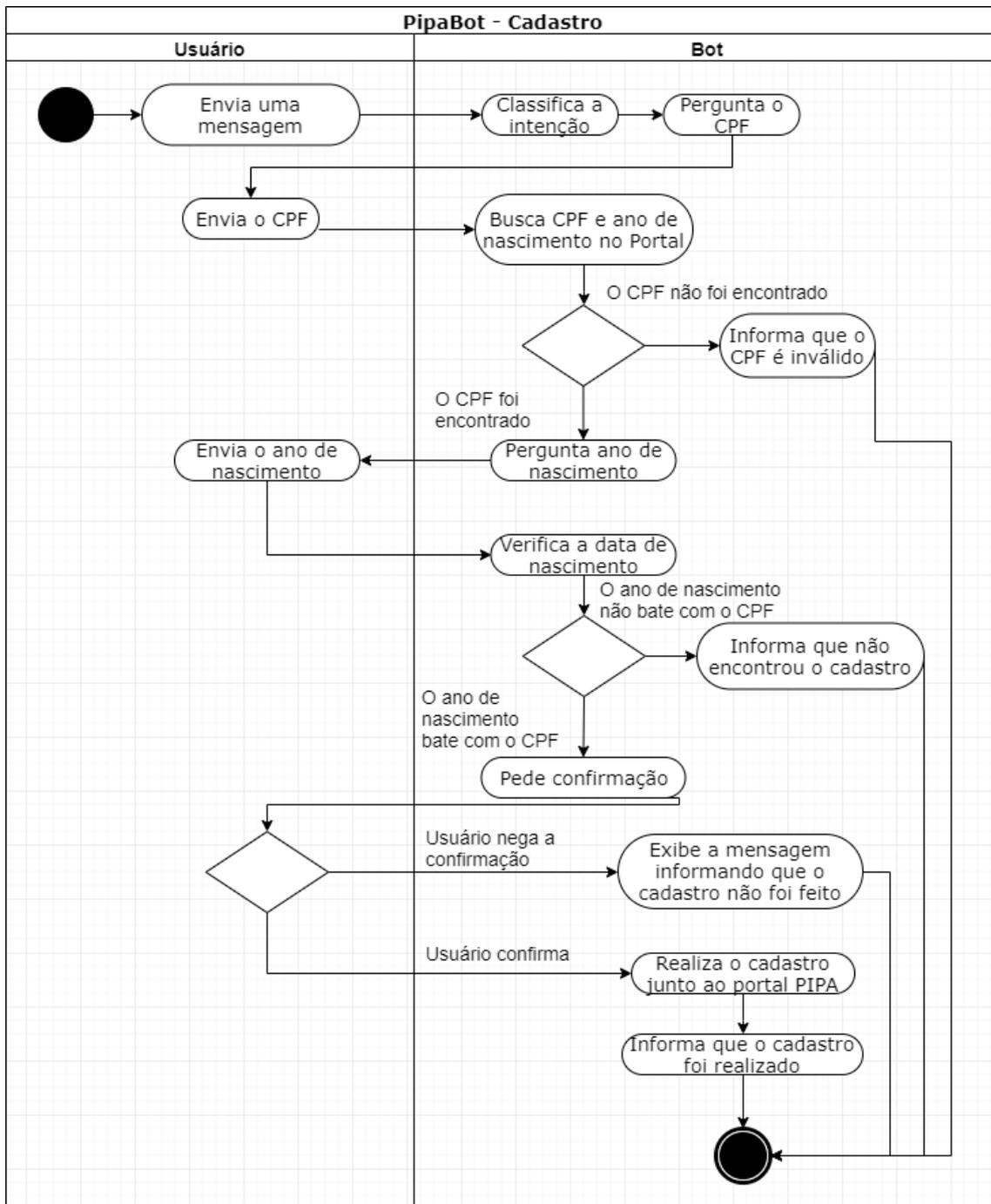


Figura C.3: Diagrama de atividades para o fluxo de conversa de Cadastro

Apêndice D

Testes Funcionais

Neste apêndice estão descritos os testes de software especificados na seção 4.4.6 deste trabalho.

D.1 Obter informações sobre o PIPA

Criticidade	Alta
Localização	PipaBot (<i>Messenger</i> ou <i>Web</i>)
Objeto de Teste	Fluxo de informações sobre o PIPA
Caso de Teste	Obter informações sobre o PIPA
Pré-condição	O servidor do PipaBot e o canal (<i>Messenger</i> ou <i>Web</i>) estão online
Frequência de uso	Alta
Procedimento	1. Enviar uma mensagem para o PipaBot perguntando informações do projeto e/ou seus objetivos.
Resultado Esperado	Mensagens contendo informações sobre o projeto e/ou seus objetivos.

Tabela D.1: Teste Funcional 1 - Fluxo de informações sobre o PIPA

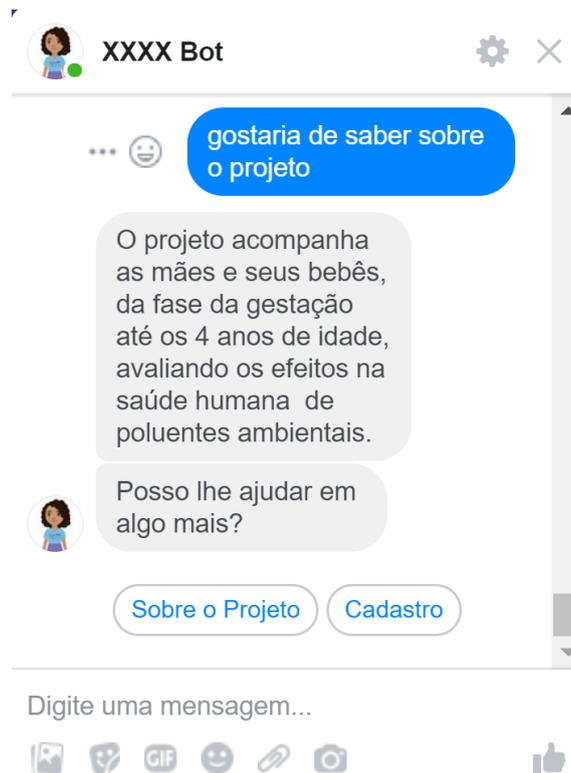


Figura D.1: Resultado do TF01

D.2 Obter informações sobre como criar um login no Portal do PIPA

Criticidade	Alta
Localização	PipaBot (<i>Messenger</i> ou <i>Web</i>)
Objeto de Teste	Fluxo de criação de login no portal
Caso de Teste	Obter informações sobre como criar um login no portal do PIPA
Pré-condição	O servidor do PipaBot e o canal (<i>Messenger</i> ou <i>Web</i>) estão online
Frequência de uso	Alta
Procedimento	1. Perguntar ao PipaBot sobre como é possível criar um login no Portal do PIPA
Resultado Esperado	O PipaBot deverá responder informando o telefone do responsável pela criação do login.

Tabela D.2: Teste Funcional 2 - Fluxo de criação de login no portal

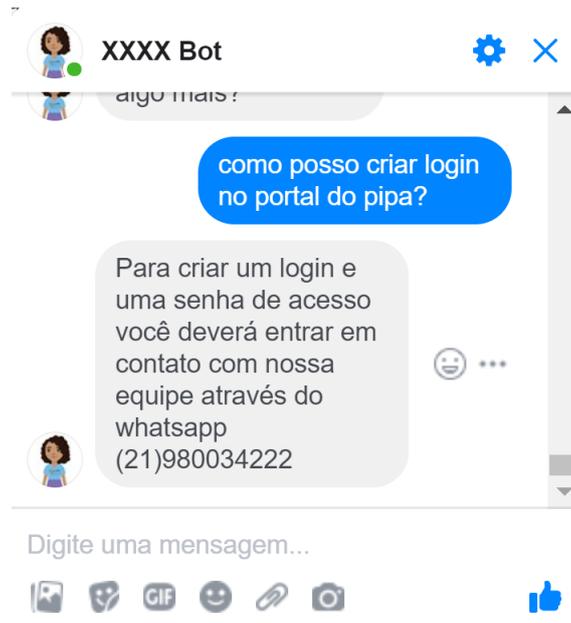


Figura D.2: Resultado do TF02

D.3 Vincular um usuário do chatbot à sua conta no Portal do PIPA

Criticidade	Alta
Localização	PipaBot (<i>Messenger</i> ou <i>Web</i>)
Objeto de Teste	Fluxo de cadastro do participante no <i>chatbot</i>
Caso de Teste	Vincular um usuário do chatbot à sua conta no Portal do PIPA
Pré-condição	O servidor do PipaBot, o Portal do PIPA e o canal (<i>Messenger</i> ou <i>Web</i>) estão online
Frequência de uso	Média
Procedimento	<ol style="list-style-type: none"> 1. Enviar uma mensagem para o PipaBot solicitando o cadastro no canal. 2. Digitar seu CPF 3. Digitar o ano do seu nascimento 4. Ao solicitar a confirmação, clicar no botão ou enviar um "Sim"
Resultado Esperado	O usuário receberá uma informação de que está cadastrado e os fluxos de consultas e exames estarão desbloqueados.

Tabela D.3: Teste Funcional 3 - Fluxo de cadastro do participante no *chatbot*

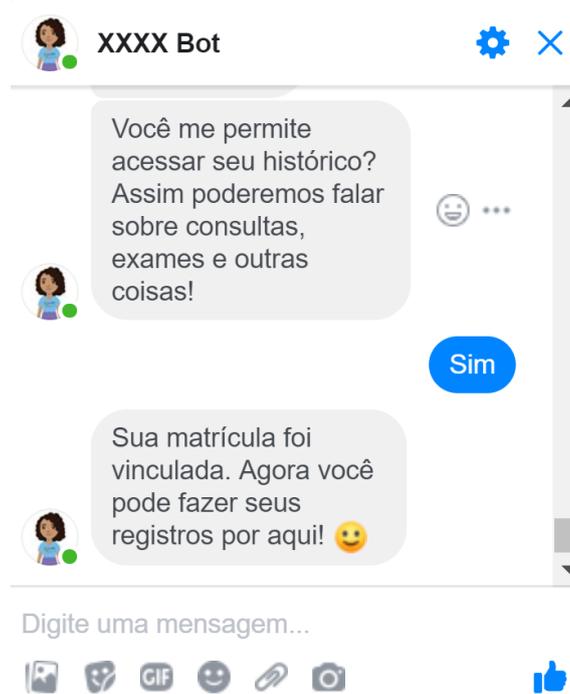


Figura D.3: Resultado do TF03

D.4 Obter informações sobre exames

Criticidade	Alta
Localização	PipaBot (<i>Messenger</i> ou <i>Web</i>)
Objeto de Teste	Fluxo de exames
Caso de Teste	Obter informações sobre exames
Pré-condição	O servidor do PipaBot e o canal (<i>Messenger</i> ou <i>Web</i>) estão online
Frequência de uso	Alta
Procedimento	Enviar uma mensagem perguntando informações sobre exames como: resultados, agendamento, etc.
Resultado Esperado	O PipaBot enviar um link para o login no Portal do PIPA onde é possível consultar os exames. Além disso, lhe fornecerá, através do clique em botões, informações como resultados impressos e disponibilidade de retirada dos exames.

Tabela D.4: Teste Funcional 4 - Fluxo de exames

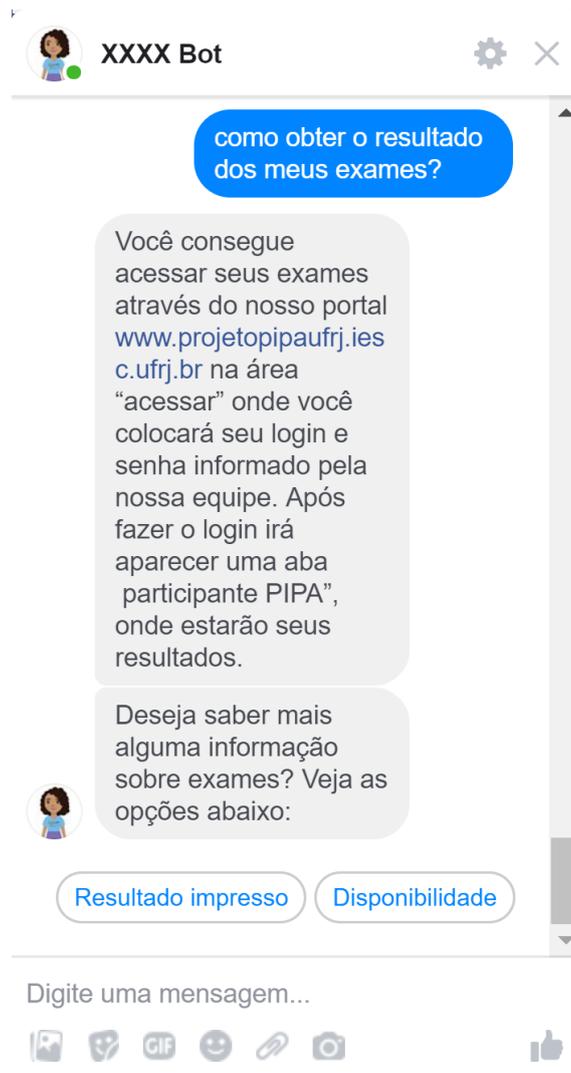


Figura D.4: Resultado do TF04

D.5 Obter informações sobre consultas

Criticidade	Alta
Localização	PipaBot (<i>Messenger</i> ou <i>Web</i>)
Objeto de Teste	Fluxo de consultas
Caso de Teste	Obter informações sobre consultas
Pré-condição	O servidor do PipaBot e o canal (<i>Messenger</i> ou <i>Web</i>) estão online
Frequência de uso	Alta
Procedimento	Enviar uma mensagem perguntando sobre marcação de consultas.
Resultado Esperado	O PipaBot irá informar o telefone de contato para agendamento e marcação de consultas.

Tabela D.5: Teste Funcional 5 - Fluxo de consultas

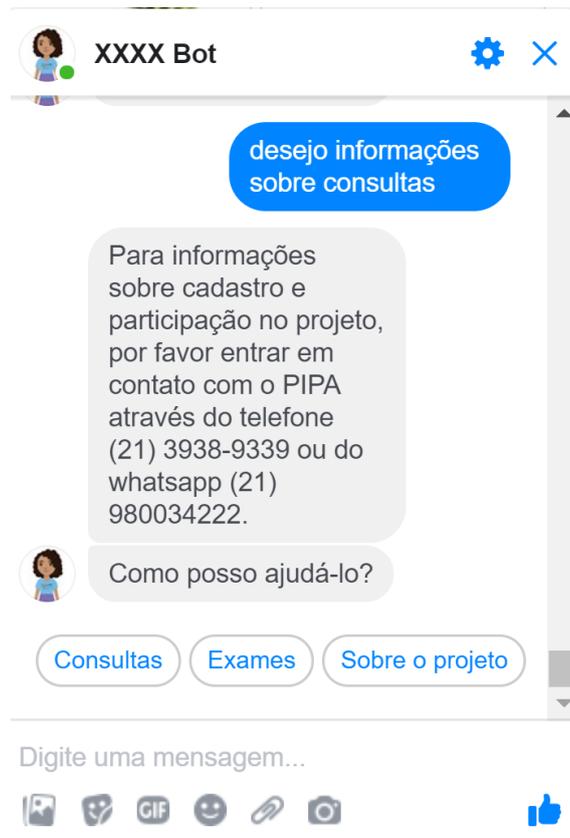


Figura D.5: Resultado do TF05

Apêndice E

Formulário de Avaliação

Avaliação do PipaBot

Esse formulário visa apoiar a avaliação da tecnologia PipaBot implementada para o projeto PIPA-UFRJ. O objetivo é avaliar a facilidade de uso e a utilidade do PipaBot seguindo o modelo TAM (Technology Acceptance Model). Esta avaliação faz parte de um projeto final de graduação de Engenharia de Computação e Informação da UFRJ.

As respostas coletadas através deste formulário são anônimas.

Qualquer dúvida quanto ao preenchimento desse formulário ou quanto ao projeto PipaBot, entrar em contato com Lucas de Paula (lucasdepaula@poli.ufrj.br).

Muito obrigado por cooperar com esta pesquisa,
Lucas de Paula, Taísa Guidini e Guilherme Travassos

Para prosseguir, clique no botão abaixo.

***Obrigatório**

Dados Sociodemográficos

1. Sexo *

Marcar apenas uma oval.

- Masculino
 Feminino

2. Formação *

Marcar apenas uma oval.

- Pós-Doutorado
 Aluno de Doutorado
 Aluno de Mestrado
 Aluno de Graduação

3. Faixa Etária *

Marcar apenas uma oval.

- Menos de 21 anos
 De 21 a 30 anos
 De 31 a 40 anos
 De 41 a 50 anos
 De 51 A 60 anos
 Mais de 60 anos

Facilidade de Uso

As próximas perguntas tem como objetivo avaliar a Facilidade de Uso do PipaBot com relação ao número de mensagens enviadas para obter uma determinada informação e o tempo de resposta.

As respostas são objetivas, e dispostas em uma escala que vai de "Concordo Totalmente" até "Discordo Totalmente".

Após cada questão, é possível deixar comentários pertinentes ao questionamento.

4. O PipaBot é fácil de usar *

Marcar apenas uma oval.

- Concordo Totalmente
- Concordo Parcialmente
- Discordo Parcialmente
- Discordo Totalmente

5. Comentários

6. O PipaBot responde rapidamente aos meus questionamentos *

Marcar apenas uma oval.

- Concordo Totalmente
- Concordo Parcialmente
- Discordo Parcialmente
- Discordo Totalmente

7. Comentários

8. Eu posso obter a informação que desejo com poucas perguntas *

Marcar apenas uma oval.

- Concordo Totalmente
- Concordo Parcialmente
- Discordo Parcialmente
- Discordo Totalmente

9. Comentários

10. Os elementos de navegação (menus e botões) do PipaBot facilitam a interação ao longo da conversa *

Marcar apenas uma oval.

- Concordo Totalmente
- Concordo Parcialmente
- Indiferente
- Discordo Parcialmente
- Discordo Totalmente

11. Comentários

Utilidade

As próximas perguntas tem como objetivo avaliar a Utilidade do PipaBot com relação a pertinência e adequação das respostas recebidas.

As respostas são objetivas, e dispostas em uma escala que vai de "Concordo Totalmente" até "Discordo Totalmente".

Após cada questão, é possível deixar comentários pertinentes ao questionamento.

12. O PipaBot fornece respostas pertinentes aos meus questionamentos *

Marcar apenas uma oval.

- Concordo Totalmente
- Concordo Parcialmente
- Discordo Parcialmente
- Discordo Totalmente

13. Comentários

14. O PipaBot fornece respostas adequadas aos meus questionamentos *

Marcar apenas uma oval.

- Concordo Totalmente
- Concordo Parcialmente
- Discordo Parcialmente
- Discordo Totalmente

15. Comentários

Powered by
 Google Forms

Apêndice F

Roteiro de Avaliação

**Universidade Federal do Rio de Janeiro
Engenharia de Computação e Informação**

Roteiro de avaliação do Chatbot PIPA

Lucas Santos de Paula <*lucasdepaula@poli.ufrj.br*>
Taisa Guidini Gonçalves <*taisa@cos.ufrj.br*>
Guilherme Horta Travassos <*ght@cos.ufrj.br*>

Julho/2019

1. Introdução

O PipaBot é um chatbot (software desenvolvido para simular um correspondente humano em uma conversa) que está inserido no contexto do Projeto PIPA UFRJ. A tecnologia é capaz de fornecer informações ligadas aos objetivos do projeto, login, exames e consultas médicas.

2. Objetivo

Esse roteiro tem como objetivo guiar o usuário na avaliação do PipaBot segundo o modelo de avaliação da tecnologia (do inglês Technology Acceptance Model - TAM). Este modelo tem como foco a avaliação da facilidade de uso e da utilidade da tecnologia (PipaBot) considerando a perspectiva e expectativa do usuário.

3. Características a serem observadas

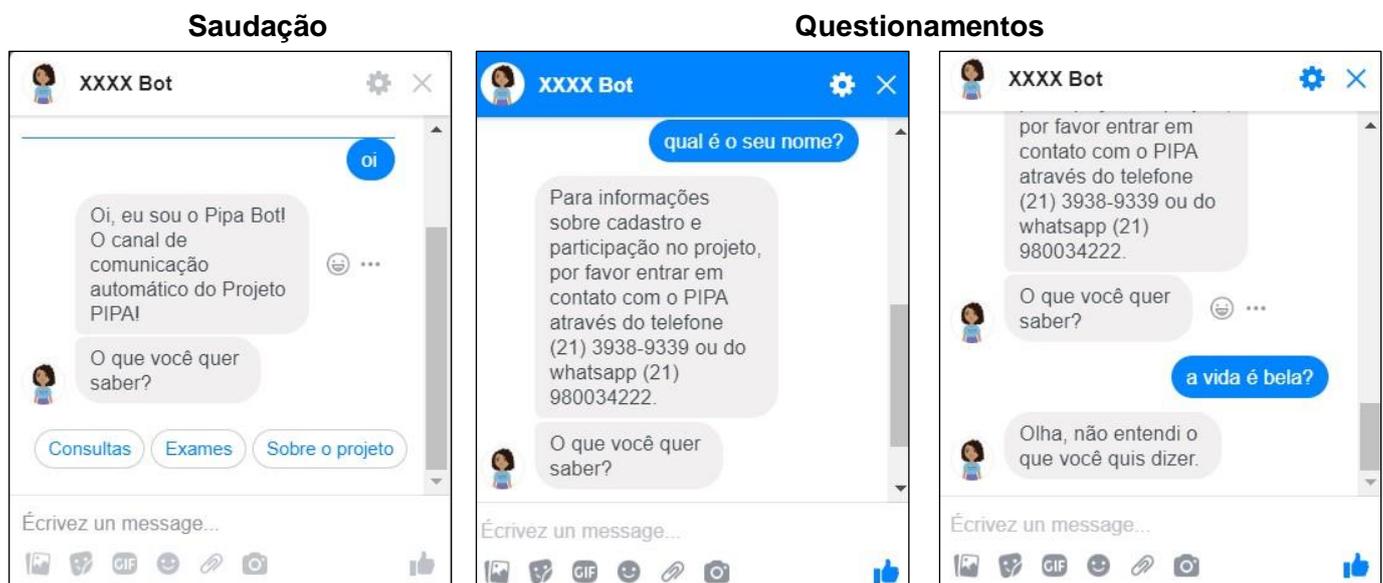
As características a serem observadas nessa avaliação são **facilidade de uso** e **utilidade**.

Caracteriza-se **facilidade de uso** como o número de mensagens e tempo de resposta necessários para se obter uma determinada informação. Assim, espera-se que o usuário informe sua percepção de facilidade frente a experiência de uso com o PipaBot.

A **utilidade** é caracterizada pela pertinência e adequação das respostas oferecidas pelo chatbot ao longo do diálogo. O usuário deve informar sua percepção de utilidade frente as suas expectativas em relação ao PipaBot. Atente que não estamos em buscas de elogios, mas de informações que nos levem a construir uma solução tecnológica que atenda a sua expectativa!

4. Cenários

Esta seção apresenta alguns possíveis cenários de uso que serão avaliados. O objetivo é que o usuário dialogue com o chatbot buscando obter as informações descritas nos cenários (os quais esperamos que representem situações de interesse em comum).



4.1. Sobre o projeto PIPA

Neste cenário o usuário dialoga com o chatbot com o objetivo de obter informações gerais a respeito do projeto (por exemplo, o objetivo do projeto, o objeto de estudo do projeto, como participar do projeto, etc.). O objetivo é obter informações!

4.2. Cadastro no portal do PIPA

Neste cenário o usuário dialoga com o chatbot com o objetivo de obter informações relacionadas a criação de um cadastro (login) no Portal do PIPA. Este cadastro permite que uma participante do projeto tenha acesso aos resultados dos seus exames, por exemplo. O objetivo é obter informações!

4.3. Cadastro no PipaBot

Neste cenário o usuário dialoga com o chatbot com o objetivo de cadastrar o seu perfil junto ao chatbot. Neste cenário o chatbot solicitará o CPF e ano de nascimento do usuário. Um usuário com perfil cadastrado pode acessar informações dentro do chatbot que são restritas as participantes do projeto (informações de exames e consultas, por exemplo). O objetivo é efetuar o cadastro!

4.4 Exames

Neste cenário o usuário cadastrado no PipaBot dialoga com o chatbot com o objetivo de obter informações sobre exames (por exemplo, disponibilidade dos exames, resultado dos exames, onde retirar/pegar os exames prontos, etc.). O objetivo é obter informações!

4.5 Consultas

Neste cenário o usuário cadastrado no PipaBot dialoga com o chatbot com o objetivo de obter informações sobre consultas médicas (por exemplo, data da próxima consulta, como marcar ou remarcar uma consulta, etc.). O objetivo é obter informações!