



**INSTITUTO
FEDERAL**

Alagoas

INSTITUTO FEDERAL DE ALAGOAS - CAMPUS ARAPIRACA

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

JONAS SOUSA LEANDRO

MATHEUS BRENO DOS SANTOS MOURA

MARVIN: UM CHATBOT PARA AUXÍLIO À COMUNIDADE DO IFAL

ALAGOAS
DEZEMBRO DE 2022

JONAS SOUSA LEANDRO
MATHEUS BRENO DOS SANTOS MOURA

MARVIN: UM CHATBOT PARA AUXÍLIO À COMUNIDADE DO IFAL

Monografia apresentada ao curso de Sistemas de Informação do Instituto Federal de Educação, Ciência e Tecnologia de Alagoas, como requisito parcial para a obtenção do título de Bacharel em Sistemas de Informação.

Orientadora: Prof. Dra. Cledja Rolim

Instituto Federal de Alagoas - Campus Arapiraca

ALAGOAS
DEZEMBRO DE 2022



Dados Internacionais de Catalogação na Publicação
Instituto Federal de Alagoas
Campus Arapiraca

L437m

Leandro, Jonas Sousa.

Marvin: um chatbot para auxílio à comunidade do IFAL /
Jonas Sousa Leandro, Matheus Breno dos Santos Moura. –
2022.

1 PDF: il., color. (1 arquivo : 2,9 MB).

Arquivo digital no formato PDF do trabalho acadêmico com
54 folhas.

Orientação: Prof^a. Dr^a. Cledja Karina Rolim da Silva.

Trabalho de Conclusão de Curso (Graduação,
Bacharelado em Sistemas de Informação) – Instituto Federal
de Alagoas, *Campus Arapiraca*, Arapiraca, 2022.

1. Chatbot. 2. Sistemas de informação. 3. IFAL. I. Moura,
Matheus Breno dos Santos. II. Título.

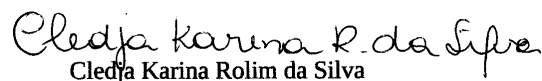
CDD: 005

JONAS SOUSA LEANDRO
MATHEUS BRENO DOS SANTOS MOURA

MARVIN: UM CHATBOT PARA AUXÍLIO À COMUNIDADE DO IFAL

Monografia apresentada ao curso de Sistemas de Informação do Instituto Federal de Educação, Ciência e Tecnologia de Alagoas, como requisito parcial para a obtenção do título de Bacharel em Sistemas de Informação.

Trabalho aprovado. Alagoas, 22 de dezembro de 2022:


Cledja Karina Rolim da Silva

Prof. Dra. Cledja Rolim
Instituto Federal de Alagoas - Campus Arapiraca



Prof. Dr. Douglas Menezes
Instituto Federal de Alagoas - Campus Arapiraca



Prof. Dr. Tarsis Marinho
Instituto Federal de Alagoas - Campus Arapiraca

ALAGOAS
DEZEMBRO DE 2022

RESUMO

A quantidade crescente de dados vem agravando o problema da procura de informação, pois, em sua maioria, são dados textuais, não estruturados e dispersos em várias fontes. Neste contexto, este trabalho propõe o uso de um *chatbot* para melhorar a experiência do usuário ao buscar por informações relacionadas ao IFAL campus Arapiraca. Este estudo descreve uma revisão bibliográfica por trabalhos semelhantes, o desenvolvimento da aplicação e o teste com os usuários. Para o desenvolvimento da aplicação, batizado de Marvin, foi realizada a leitura de projetos pedagógicos, instruções normativas, páginas na internet para o cadastramento de informações no *chatbot*. Além do estudo de frameworks e ferramentas. Os testes tiveram o objetivo de avaliar o uso e aceitação do *chatbot* criado, que de acordo com os resultados, foi bem aceito pela comunidade, sendo uma solução interessante para acesso à informação.

Palavras-chave: Chatbot; Sistemas de informação; IFAL; Informação.

ABSTRACT

The huge amount of data aggravates the search for information, because most of the data are texts, unstructured and dispersed in several sources. In this context, this work aims the use and the development of a chatbot to improve the user experience related to IFAL campus Arapiraca. This study describes a peer-review of similar work, the development of an application, and the user testing. To the development of the chatbot, it was called Marvin, it carried out the reading of pedagogical projects, normative instructions, and internet pages, besides the study of the frameworks related to chatbots. The user testing had the goal to evaluate the using and acceptance of the chatbot. And in agreement of the data results, Marvin has had a good acceptance, being an interesting solution to the information access.

Keywords: Chatbot; Information System; IFAL; Information.

LISTA DE FIGURAS

Figura 1. Protótipo das telas do Marvin em dimensões diferentes	31
Figura 2. Logotipo e avatar do Marin	33
Figura 3. Protótipo da versão desktop do Marvin	34
Figura 4. Tela inicial do Marvin	35
Figura 5. Tela de diálogo com o Chatbot	35
Figura 6. Configuração do banco de dados no código do chatbot	36
Figura 7. Comandos para chamada HTTP para a API	38
Figura 8. Consulta ao acervo da biblioteca	39
Figura 9. Resultado de uma consulta ao acervo	39
Figura 10. Chat de diálogo com botão de redirecionamento ao acervo da biblioteca.	40
Figura 11. Divulgação do Marvin nas dependências do campus	41
Figura 12. Perfil oficial do campus Arapiraca contribuindo para divulgação	42
Figura 13. Fluxograma de interação com a pesquisa de satisfação	43
Figura 14. Interações com o Marvin de outubro a novembro de 2022	45

LISTA DE TABELAS

Tabela 1. Resumo dos trabalhos relacionados à proposta deste projeto	22
Tabela 2. Quantidade de pessoas que gostaram e não gostaram Marvin segundo avaliações	42
Tabela 3. Preferência entre Marvin ou atendimento presencial no IFAL	43
Tabela 4. Buscar no Marvin é mais rápido que na internet	44
Tabela 5. Tabela 4 – Relação entre os usuários e o IFAL	44
Tabela 6. Quantidade de cada interação para os intents cadastrados	45
Tabela 7. Preferência do usuário em resposta direta ou navegar pelo menu.	47

LISTA DE ABREVIATURAS E SIGLAS

SQL	Structured Query Language
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
RPG	Role-playing game
AIML	Artificial Intelligence Markup Language
XML	Extensible Markup Language
IBM	International Business Machines
ETL	Extract Transform Load
API	Application Programming Interface
AVEAS	Ambientes Virtuais de Ensino Aprendizagem
INEP	Instituto Nacional de Estudos e Pesquisa Educacionais
PAAS	Platform as a Service
NoSQL	Not Only Structured Query Language
ACID	Atomicity, Consistency, Isolation, and Durability
REST	Representational State Transfer
HTTP	Hypertext Transfer Protocol
JSON	Representational State Transfer
BSON	JavaScript Object Notation
HTML	HyperText Markup Language
CSS	Cascading Style Sheet
ECMA	European Computer Manufacturer's Association
FAQ	Frequently Asked Questions
SIGAA	Sistema Integrado de Gestão de Atividades Acadêmicas
DOM	Document Object Model

SUMÁRIO

RESUMO	5
ABSTRACT	6
LISTA DE FIGURAS	7
LISTA DE TABELAS	8
LISTA DE ABREVIATURAS E SIGLAS	8
SUMÁRIO	10
1. INTRODUÇÃO	11
2. CHATBOT	13
3. REVISÃO DA LITERATURA	17
4. DESENVOLVIMENTO DE UM CHATBOT PARA O IFAL - CAMPUS ARAPIRACA	23
4.1 FERRAMENTAS UTILIZADAS	23
4.1.1 NoSQL	23
4.1.2 DialogFlow	25
4.1.3 JAVASCRIPT E NODE.JS	28
4.2. MARVIN	30
4.2.1 AVATAR, IDENTIDADE VISUAL E INTERFACE	32
4.2.2 Banco de Dados NoSql: Google Firestore	36
4.3 Uma tentativa frustrada: Web Scraping	37
5. AVALIAÇÃO DO MARVIN	41
6. CONCLUSÕES E TRABALHOS FUTUROS	48
7. REFERÊNCIAS	50

1. INTRODUÇÃO

A quantidade de dados que vem sendo produzida tem levado a um problema: a enorme quantidade de dados textuais não estruturados, “produzidos com o crescimento da sociedade digital, que vão de informações nas redes sociais, dados de controles médicos, genéticos, e-mails, etc. [...] O conteúdo já não é estruturado, pois tem forma e tamanho livres” (Barbieri, 2020). Estes dados estão dispersos e o processo para transformá-los em informação pode ser custoso.

A maior parte destes dados está em linguagem natural, entretanto computacionalmente, são utilizadas linguagens formais, relacionadas a área de programação/desenvolvimento de *software* para a inserção ou consulta. Coppin (2010) aponta para a necessidade de as pessoas obterem informação na própria linguagem que utilizam: “a ideia de que as pessoas deveriam ser capazes de fazer perguntas em sua própria linguagem, ou algo semelhante a isso, tem se tornado bastante popular” (Coppin, 2010).

Neste contexto, as informações administrativas e sobre os cursos do Instituto Federal de Alagoas que possam ser de interesse dos alunos e comunidade em geral estão dispersas em instruções normativas, planos pedagógicos curriculares e site institucional.

Este trabalho tem como finalidade propor um *chatbot* como *interface* de consulta para estes dados. De forma que a comunidade possa efetuar perguntas de maneira simples e obter as informações solicitadas.

O problema de pesquisa consiste em: *é possível melhorar a experiência da comunidade ao buscar por informações sobre o Instituto Federal de Alagoas Campus Arapiraca através do uso de chatbot?*

Publicado em 2021, o artigo de Silva et al (2021) expõe a proposta de um *chatbot* para auxiliar a comunidade do Instituto Federal de Brasília na busca e notificações de novas informações, tendo em vista que, por padrão, estas informações ficam disponíveis no site institucional e o usuário pode não ter acesso tempestivamente. Em testes realizados, a aplicação foi bem aceita pelos usuários, por apresentar uma forma de busca ágil e eficaz, tem potencial em agregar na resolução do problema.

A obtenção de informações básicas pelo aluno sobre o curso ou serviço prestado pela instituição normalmente é obtida através de consulta presencial ou

telefônica à secretaria ou por pesquisa na internet. Ambas as formas apresentam desvantagens, o que pode frustrar a experiência do aluno em algum momento. As ferramentas de busca na internet irão procurar em toda a rede e não especificamente no assunto e podem retornar resultados imprecisos ou incorretos. O atendimento presencial possui limitação de horário e constitui uma barreira maior para a comunidade externa. Além disso, há pouca referência na literatura do uso de *chatbots* para problemas de busca e resposta em instituições de ensino.

Este trabalho possui três etapas: na primeira etapa, houve o método de revisão da literatura, onde foi realizada a busca de instituições de ensino pública que fazem uso de *chatbots*; na segunda etapa, foi desenvolvimento do *chatbot*; e na terceira e última etapa, foi realizado um estudo de caso para avaliar a ferramenta. Cada etapa será apresentada em detalhes nos capítulos seguintes.

Os objetivos deste trabalho são:

- **Objetivo geral:**

Propor um *chatbot* para o IFAL e verificar se o uso do mesmo traz melhoria na experiência dos alunos e comunidade externa do IFAL campus Arapiraca através de testes de usuário em relação ao acesso a informações administrativas ou sobre o curso.

- **Objetivos específicos:**

- Verificar se o *chatbot* trouxe melhoria na experiência dos usuários em relação a consulta a internet ou manuais/portarias;
- Identificar as principais dificuldades dos usuários ao usar o *chatbot*; e
- Identificar as principais vantagens e desvantagens do *chatbot* em relação aos métodos tradicionais.

2.CHATBOT

O conceito *chatbot* é amplo, mas, em termos gerais, pode ser entendido como uma *interface* de interação que simula uma conversa, ou seja, as entradas são sentenças em linguagem natural que são processadas em uma saída, também em linguagem natural. Normalmente, essa estrutura é humanizada em forma de uma robô conversacional, inclusive com nome, personalidade e aparência personalizados.

Linguagens naturais são as linguagens usadas pelas pessoas para se comunicarem, por exemplo, inglês, português, etc. Elas são complexas, pouco estruturadas e, principalmente ambíguas, o que quer dizer que uma mesma sentença pode ter significados diferentes dependendo do contexto. É diferente das linguagens formais, como o SQL ou JAVA¹, onde cada sentença deve conter um comando único independente do contexto em que é executada.

Isso certamente gera um conflito, pois, apesar de computacionalmente as linguagens formais serem utilizadas, as pessoas utilizam linguagens naturais para interagir com o mundo. Por isso, o Processamento de linguagem natural, um campo da Ciência da Computação, utiliza técnica para transformar informações de linguagem natural em uma estrutura formal e vice e versa.

Coppin (2010) aponta para a importância dos computadores entenderem linguagem natural: “Hoje em dia, muitos sistemas de telefonia são capazes de compreender uma gama restrita de comandos e perguntas para auxiliar clientes de grandes centrais telefônicas, sem a necessidade de utilizar recursos humanos.” Também argumenta que a quantidade de dados textuais não estruturados são imensos, muito maiores que os estruturados, principalmente depois do que se chama de *Big Data*.

Luger (2013) expõem algumas aplicações da linguagem natural: compreensão de histórias e resposta automática a questões; uma *interface* para banco de dados; um sistema para extração de informações para a *web*, e; algoritmos de aprendizado para generalizar a informação extraída. O *chatbot*, como é um *interface* para a linguagem natural, ao longo da história vem sendo utilizado para estas e outras aplicações. Como exemplo, o jogo online *AI Dungeon*² simula um jogo de RPG (*Role*

¹ Disponível em: <https://www.java.com/pt-BR/>. Acesso em 02 de dezembro de 2022

² Disponível em: <https://aidungeon.io>. Acesso em 29 de novembro de 2022

Playing Game) e através de texto, gera um mundo ficcional em que o jogador pode interagir com entradas textuais; a cada rodada o jogador deve digitar sua ação e o jogo extrai as informações e gera novas consequências de forma a manter uma história coesa.

Em outubro de 1950, Alan Turing publica o artigo *Computing Machinery and Intelligence* onde discute se máquinas são capazes de pensar (Turing, 1950). Considerando a dificuldade em conceituar o que seria pensar, Turing propõe a reformulação da pergunta propondo um jogo: um interlocutor interage com um humano e uma máquina sem saber quem é quem, tendo acesso apenas às suas respostas sem qualquer outra característica, tal como voz ou aparência. No final, o interlocutor deve responder qual dos dois é a máquina. A pergunta final é: uma máquina poderia se passar por um ser humano, de forma que, o interlocutor não saiba distinguir? A isto, Turing chamou de *The Imitation Game* (“O Jogo da Imitação” em livre tradução). Este pensamento foi uma forte influência nos primeiros *chatbots*.

Eliza é considerada o primeiro *chatbot*, muito influenciado pelo teste de Turing. Foi desenvolvido por Joseph Weizenbaum, no *Massachusetts Institute of Technology* - MIT (Weizenbaum, 1966). Seu propósito é simular uma psicanalista que estimula a conversa e dá conselhos. Seu funcionamento é ainda muito rudimentar e tem como base identificar palavras-chave. Quando o programa não as encontra, realiza mais perguntas, até que os termos sejam identificados. Com isso, as respostas dadas pela *Eliza* pode muitas vezes não fazer sentido, por exemplo: “pode ser ilustrado da seguinte forma: se a entrada for, por exemplo, a frase: ‘*I am fine now thank you*’, a resposta pode ser algo como: ‘*How long have you been fine now thank i?*’ (Leonhardt et al, 2003).

Eliza não possui memória de suas interações passadas, nem tampouco é capaz de distinguir o contexto. É um programa simples com cerca de 200 linhas de código, dividido em dois módulos e baseado em regras (Comarella e Café, 2008). Apesar disso, a reação do público foi muito positiva.

Também muito influenciado pelo teste de Turing, foi criado o *Parry* em 1972, que simulava um paciente com esquizofrenia (Adamopoulou et Moussiades, 2020). Ele trouxe melhorias pois conseguia simular emoções através de pesos atribuídos a sentença do usuário. Apesar de suas limitações: baixa capacidade em entender as entradas dos usuários e de expressar emoções, demora nas respostas e não aprender com as conversas (Adamopoulou et Moussiades, 2020) .

Quase trinta anos depois de *Eliza*, foi criado em 1995 o *chatbot* A.L.I.C.E (*Artificial Linguistic Internet Computer Entity*) por Richard S. Wallace na *Lehigh University* (Leonhardt et al, 2003). Sua inovação está na criação da linguagem de marcação AIML (*Artificial Intelligence Markup Language*), criada para estrutura a base de conhecimento: “apresenta um conjunto de *tags* e comandos simples para implementação da base de conhecimento de um *chatterbot* e serve para analisar as mensagens enviadas pelo usuário e decidir a forma como estas mensagens devem ser respondidas” (Leonhardt et al, 2003).

A AIML foi usada para construir a ampla base de dados do *chatbot* A.L.I.C.E. Tem como base a linguagem de marcação XML³ e tem como objetivo registrar e classificar os dados de interação. No AIML, são criadas categorias (em inglês, *category*) e modelos (em inglês, *template*) que representam as possíveis respostas ao usuário. O objeto que armazena todas as categorias é chamado de “Graphmaster”, que contém as categorias em uma estrutura de árvore. A AIML tenta selecionar uma resposta apropriada para a fala do usuário através de uma busca profunda nesta estrutura (Adamopoulou et Moussiades, 2020) .

Foi desenvolvido com um modelo de aprendizagem supervisionado, o que quer dizer que as respostas do *chatbot* são validadas ou não pelos usuários, assim, ajustando o seus pesos, de forma que gradativamente melhora a precisão das respostas. Um "treinador" monitora as conversas do *chatterbot* e cria o índice novo de AIML, a fim de obter as respostas mais apropriadas. O cérebro de A.L.I.C.E. conta atualmente com mais de 45.000 elementos chamados categorias. (Comarella e Café, 2008).

A *Eliza* operava com apenas 200 palavras-chave (Adamopoulou et Moussiades, 2020). Portanto, A.L.I.C.E significou um avanço nas técnicas de criação de um *chatbot*, apesar de possuir suas limitações, tais como a de gerar novas respostas ou expressar emoção.

Outro *chatbot* importante a ser mencionado é o *SmarterChild*, criado em 2001 (Adamopoulou et Moussiades, 2020). Ele estava integrado nos programas de conversa *on-line Messengers like America Online (AOL)* e *Microsoft (MSN)*. A interação com o *SmarterChild* começava como uma conversa com qualquer outro usuário. Seu propósito era o de responder a diversas dúvidas dos usuários, tais como, o clima em uma cidade, a cotação de uma moeda, pesquisas na internet, etc.

³ Disponível em: <https://pt.wikipedia.org/wiki/XML>. Acesso em 29 de novembro de 2022.

Quando foi implantado, as ferramentas de busca na internet não eram tão sofisticadas quanto as atuais. Desta forma, o *SmarterChild* disponibiliza uma *interface* de pesquisa para o usuário.

Os assistentes virtuais popularizam os *chatbots*. Atualmente, há vários no mercado, por exemplo: a *Siri* da *Apple*⁴, *Google Assistant*⁵, *Cortana* da *Microsoft*⁶, *Alexa* da *Amazon*⁷, entre outros. Além da *interface* escrita, os usuários interagem principalmente através da fala. Eles disponibilizam uma forma mais humanizada de acessar recursos do sistema, como por exemplo, abrir um aplicativo, informar a hora, tocar uma música, etc.

Em 2011, a IBM criou o *chatbot* *Watson*⁸ (Adamopoulou et Moussiades, 2020). O surgimento do *Watson* trouxe entusiasmo, pois ele trazia um maior poder computacional e de compreensão da fala humana, principalmente quando conseguiu vencer dois campeões no *Jeopardy*⁹, um jogo em que o usuário recebe dicas e deve relacionar com uma resposta. Apesar do entusiasmo inicial, atualmente o uso do *Watson* é mais conservador, servindo como uma ferramenta para a criação de *chatbots*, principalmente para uso comercial, sendo o principal concorrente do *DialogFlow* da *Google*¹⁰.

No próximo capítulo serão mostrados os principais trabalhos relacionados ao uso e aplicação de *chatbots*, principalmente em instituições de ensino.

⁴ Disponível em: <https://www.apple.com/br/siri/>. Acesso em 29 de novembro de 2022.

⁵ Disponível em: https://assistant.google.com/intl/pt_br/. Acesso em 29 de novembro de 2022.

⁶ Disponível em:

<https://support.microsoft.com/pt-br/topic/o-que-%C3%A9-a-cortana-953e648d-5668-e017-1341-7f26f7d0f825>. Acesso em 29 de novembro de 2022.

⁷ Disponível em: <https://www.amazon.com.br/b?ie=UTF8&node=19877613011>. Acesso em 29 de novembro de 2022.

⁸ Disponível em: <https://www.ibm.com/br-pt/watson>. Acesso em 02 de dezembro de 2022.

⁹ Disponível em: <https://www.jeopardy.com/>. Acesso em 29 de novembro de 2022.

¹⁰ Disponível em: <https://dialogflow.cloud.google.com/>. Acesso em 29 de novembro de 2022.

3. REVISÃO DA LITERATURA

A primeira etapa do desenvolvimento do sistema foi o de buscar trabalhos relacionados. Para isto, utilizamos o portal de Periódicos da CAPES¹¹, na procura de artigos relacionados à proposta, de forma a identificar o público e o resultado da interação com o *chatbot*. Desta forma, formando uma base em que pudéssemos balizar o resultado desta pesquisa; e que também, gerasse outros parâmetros a serem analisados, pois como o tema é muito amplo, pois o *chatbot* é utilizado em várias situações (aprendizagem de idiomas, uso médico/psicológico, extrair informações, etc) foi refinada a pesquisa, selecionando artigos mais próximos a nossa proposta, utilizando as palavras chaves “*chatbot*” e “*chatterbot*”. Assim foi realizada a seleção de oito artigos.

Depois que Alan Turing propôs o Jogo da Imitação em 1950, os primeiros *chatbot* eram fortemente inspirados em testar se um computador poderia se passar por um ser humano. Podemos concluir com a leitura dos artigos que, atualmente, os *chatbot* são utilizados em áreas e problemas diversos e que o processo de melhoria de um *chatbot* deve ser contínuo e se apoiar no *feedback* dos usuários, pois é difícil prever como eles irão interagir com a aplicação, pois a linguagem natural é complexa e possui muitas possibilidades de estrutura e intenção.

No trabalho de Tôrres et al (2020), o ambiente de estudo é a rede de supermercados Verdemar, que possui um conjunto complexo de soluções tecnológicas implementadas internamente. E, por este motivo, precisa gerenciar possíveis falhas e melhorias para a manutenção dos sistemas. Para isso, as equipes de tecnologia da informação da organização utilizaram o *software Mantis*¹² para inserir os problemas e, como anotação, as possíveis soluções.

O *Mantis* permite a inserção de casos relacionados a um dos vários projetos. Com isso, as informações de problemas e respostas não estão centralizadas, tornando difícil encontrar soluções para situações parecidas. Era necessário encontrar uma forma de busca mais adequada e, para tanto, Tôrres et al (2020) implementaram um *chatbot*.

O problema e soluções encontradas no artigo de Tôrres et al (2020) são bem parecidos com o presente trabalho. Porém, a implementação é totalmente diferente.

¹¹ Disponível em: <https://www.periodicos.capes.gov.br/>. Acesso em: 29 de novembro de 2022.

¹² Disponível em: <https://www.mantisbt.org/>. Acesso em: 29 de novembro de 2022.

Para implementar o *chatbot*, eles fizeram um *backup* do banco de dados e através do *software Pentaho*¹³, utilizaram a técnica de ETL (Extract Transform Load, “extração, transformação e carregamento”)¹⁴ para posterior elaboração das perguntas e respostas com a API Microsoft QnA Maker, um serviço da Microsoft Azure¹⁵.

Neste trabalho as informações, por estarem salvas em um banco de dados, estavam mais estruturadas e menos dispersas. A solução encontrada foi interessante, porém necessita de alguns ajustes, pois a pergunta mostra como resposta às possíveis soluções encontradas em cada caso e, como citado no artigo, as respostas não apresentam gentileza e cortesia, reduzindo uma característica importante do *chatbot* que é a de apresentar informações de forma mais humanizada. No artigo, não foram feitos testes com os usuários. Desta forma, não demonstra a efetividade da ferramenta com a equipe de tecnologia da informação, quem realmente irá utilizar da solução.

O artigo de Illescas-Manzano et al (2021) usa o *chatbot* em aspecto negocial em relação ao funil de vendas, que é uma técnica do *marketing* que atrai clientes através de alguns níveis para, por fim, concluir a venda. Este estudo testa se um *chatbot* pode aumentar o número de *leads*, um termo no *marketing* para clientes em potencial que possuem real interesse em produtos. Foi implementado um *chatbot* com a ferramenta *ManyChat*¹⁶ e integrado com o sistema de mensagens do Facebook¹⁷. Quando os usuários entrassem na página do produto no Facebook, o *chatbot*, através do messenger, iria iniciar uma conversa, solicitando nome e e-mail. O artigo mediu o número de *leads* como a proporção dos usuários que iniciaram a conversa entre os que inseriram seus dados para contato.

Um ponto negativo do trabalho de Illescas-Manzano et al (2021) é que o *chatbot* possui uma estrutura muito simples e sua interação com o usuário é muito superficial. Como citado no artigo, a escolha pelo *ManyChat* foi apoiada pelo fato de não haver programadores na equipe, o que limitou o *chatbot* a perguntas simples. Outro ponto negativo é o número baixo de testes. Portanto, as conclusões deste artigo não podem ser generalizadas. Apesar de ser um estudo interessante, pois

¹³ Disponível em: <https://www.pentaho.com/marketplace/>; Acesso em 29 de novembro de 2022.

¹⁴ Disponível em: <https://www.oracle.com/br/integration/what-is-etl/>. Acesso em 29 de novembro de 2022.

¹⁵ Disponível em: <https://azure.microsoft.com/pt-br/>; Acesso em 30 de novembro de 2022.

¹⁶ Disponível em: <https://manychat.com/>. Acesso em 29 de novembro de 2022.

¹⁷ Disponível em: <https://pt-br.facebook.com/>. Acesso em 29 de novembro de 2022

normalmente os usuários necessitam preencher formulários para inserir seus dados no processo de vendas.

Santos Júnior et al (2021) desenvolveram um *chatbot* com objetivo de incentivar e de auxiliar o pré-natal em Unidades Básicas de Saúde, tendo como base uma pesquisa bibliográfica para delinear as perguntas e as respostas que fariam parte do escopo do agente conversacional. Conforme informado pelo artigo, embora seja fortemente recomendado o pré-natal, muitas mulheres se ausentam das consultas. Neste contexto, o agente conversacional, batizada de *GEST*, não irá substituir os profissionais e procedimentos de saúde, mas sim, conscientizar e ajudar em dúvidas pontuais. Um fator importante no uso de *chatbot* e que o estudo reafirma é privacidade e descrição dos usuários, que, muitas vezes, podem sentir vergonha em perguntar determinados assuntos a outra pessoa. Houve teste com 200 usuários através de preenchimento de formulários. Uma fragilidade do artigo é não deixar claro como e quais foram as perguntas efetuadas aos usuários e como foram as respostas.

O estudo de Barros e Guerreiro (2018) versa sobre o ensino a distância (EAD) e a dificuldade dos alunos deste sistema de ensino de receber tutoria individual, dado a escassa assistência, um problema recorrente do ensino a distância. Conforme os autores, o objetivo do estudo é responder a seguinte pergunta: “Para além dos motivos clássicos de dificuldades de apoio aos tutores, como tempo, condições técnicas e dificuldades de comunicação e habilidades de empatia pedagógica, de que forma é que um *chatbot* (Tutor Virtual) poderá facilitar e contribuir por um apoio rápido e eficiente aos estudantes?”. Um aspecto importante dos agentes conversacionais que foi mencionado é a sua disponibilidade. Um *chatbot* deve estar disponível todos os dias em todas as horas. O artigo tece um estudo interessante sobre *chatbots*, em especial, sobre o IBM Watson. Porém, acaba trazendo pouca coisa além do referencial teórico. Seria interessante que houvesse testes, trazendo, assim, dados externos que poderiam contestar ou corroborar com as premissas.

O artigo de Joveliano et al (2020) evidencia a dificuldade de pessoas com deficiências em encontrar instituições de ensino superior adequadas às suas necessidades, mostrando que muitas delas precisam mudar de cidade ou estado. Neste contexto, é trazido o conceito de educação a distância (EAD), pois, por essência, visa reduzir barreiras ao acesso à aprendizagem. No ensino a distância, é

comum o uso de Ambientes Virtuais de Ensino Aprendizagem (AVEAS), plataformas disponibilizadas via internet que auxiliam os alunos. Sobre a metodologia adotada foi realizada uma pesquisa bibliográfica previamente, para tanto, foram coletados dados do site do INEP (Instituto Nacional de Estudos e Pesquisa Educacionais)¹⁸ dos alunos matriculados em 2016 junto com os dados dos municípios. Foi utilizada a linguagem de programação *R*¹⁹, muito usada para tratamento de dados e estatística e manipulação dos dados. Foi realizado o tratamento desses dados, retirando dados inconsistentes ou ausentes, retirando colunas desnecessárias e adicionando colunas importantes. Os *chatbot* se mostraram ser uma opção interessante em relação a acessibilidade, pois a maioria deles possuem interface para texto ou para áudio, ou a conversão de um para outro (“*text to speech*” e “*speech to text*”). Como ponto negativo, o artigo não deixa claro em que cenários o *chatbot* poderia ser utilizado para este propósito.

O artigo de Ota et al (2019) trata sobre sistemas adaptativos de aprendizagem, que são soluções que utilizam a tecnologia para gerar um plano de estudo personalizado para cada aluno que tem como base suas necessidades ou preferências. Para tanto, pode fazer uso de algoritmos específicos (*learning analytics*), mecanismos de avaliação que permitem avaliar as necessidades dos alunos (Ota et al, 2019). Neste trabalho, foram desenvolvidos dois cursos de extensão, um sobre matemática e o outro sobre língua portuguesa. O estudante realiza um teste de verificação e posteriormente é gerado um plano de estudo personalizado de acordo com seu desempenho. Um *chatbot* foi integrado à plataforma de ensino para responder às principais dúvidas dos estudantes. Esta ação é citada pelo artigo como em andamento e não há mais detalhes quanto a sua aceitação. Considerando a proposta do ensino adaptativo, seria interessante que o *chatbot* fosse capaz de coletar as dúvidas dos estudantes que não fazem parte de sua base de conhecimento para posterior integração, tornando possível conhecer as dificuldades encontradas pelos alunos de forma dinâmica e promover a melhoria contínua.

No trabalho de Leonhardt et al (2003), um *chatbot* foi implementado na UFRGS²⁰ para fins educacionais com o propósito de responder a dúvidas dos

¹⁸ Disponível em: <https://www.gov.br/inep/pt-br>. Acesso em 29 de novembro de 2022.

¹⁹ Disponível em: [https://pt.wikipedia.org/wiki/R_\(linguagem_de_programa%C3%A7%C3%A3o\)](https://pt.wikipedia.org/wiki/R_(linguagem_de_programa%C3%A7%C3%A3o)). Acesso em 29 de novembro de 2022.

²⁰ Universidade Federal do Rio Grande do Sul

estudantes; no primeiro momento, em 2002, sobre dúvidas em Física para alunos, do ensino secundário, que estivessem se preparando para o vestibular e, em 2003, sobre conhecimentos de Redes de Computadores e Internet para alunos do Curso de Especialização a Distância em Informática na Educação.

O *chatbot*, batizado como *Profª Elektra*, foi criado tendo como base o *chatbot* A.L.I.C.E (Artificial Linguistic Internet Computer Entity) e utiliza a linguagem de marcação AIML (*Artificial Intelligence Markup Language*)²¹. Uma abordagem interessante foi a criação de diferentes respostas para o mesmo questionamento, escolhidas aleatoriamente, pois “se o usuário do ambiente tornou a questionar o robô sobre um mesmo tópico é porque provavelmente a resposta apresentada não satisfaz sua necessidade” (Leonhardt et al, 2003).

Através dos *logs* de interação com os usuários, pôde-se perceber que as respostas do *chatbot* eram objetivas e não estimulavam a continuidade do diálogo e que muitas das perguntas não eram compreendidas pelo *chatbot*, pois usuários tendem a se comunicar de forma diferente de acordo com a sua região (entre outros fatores). Normalmente, um agente conversacional vem com a proposta de ser mais humanizado, entregando conhecimento de forma mais natural para as pessoas e estimulando a interação. Outro ponto de observação está na imprevisibilidade em como o usuário irá interagir com o *chatbot*, desta forma é necessário realizar correções tendo como a base a intenção do usuário em relação ao questionamento.

O trabalho de Allison (2012) expõe a implantação de um *chatbot*, a *Pixel*, na Universidade de Nebraska-Lincoln em 2010. Seu propósito era de prover informações aos alunos referente ao serviço da biblioteca e outros recursos. Em relação a interação dos usuários, o artigo diz que é impossível prevê-lo. Se fez notar a quantidade de termos de baixo calão ou profanos, levando ao desenvolvimento de uma área em que o usuário teria que se desculpar, caso contrário o *chatbot* não responderia. O artigo é omissivo quanto ao resultado, não informa a quantidade de aceitação, nem o perfil de quem acessou o *chatbot*.

A Tabela 1 abaixo apresenta um resumo dos trabalhos lidos e apresentados neste capítulo, em relação a ferramenta (ou ferramentas) de desenvolvimento utilizadas, se foi aplicado numa instituição de ensino e se foram realizados testes com usuários.

²¹ Disponível em: <http://www.aiml.foundation/>. Acesso em 29 de novembro de 2022.

Tabela 1 — Resumo dos trabalhos relacionados à proposta deste projeto

Artigo	Ferramenta(s) usada no desenvolvimento	Instituição de ensino	Realizado testes com usuários
Tôrres et al (2020)	<i>Pentaho, Microsoft Azure</i>	Não	Não
Illescas-Manzano et al (2021)	<i>ManyChat</i>	Não	Sim
Santos Júnior et al (2021)	<i>DialogFlow</i>	Não	Sim
Barros e Guerreiro (2018)	Estudo Bibliográfico	Sim	Não
Joveliano et al (2020)	<i>Moodle</i>	Sim	Não
Ota et al (2019)	<i>DialogFlow</i>	Sim	Não
Leonhardt et al (2003)	A.L.I.C.E, <i>AIML</i>	Sim	Não
Allison (2012)	<i>PHP</i> ²² , <i>SQL</i> , Program-O ²³ , <i>AIML</i>	Sim	Não

²² Disponível em: <https://www.php.net/>. Acesso em 02 de novembro de 2022.

²³ Disponível em: <https://program-o.com/>. Acesso em 02 de novembro de 2022.

4. DESENVOLVIMENTO DE UM *CHATBOT* PARA O IFAL - CAMPUS ARAPIRACA

Na segunda fase, foi criado e implementado o *chatbot*. Foi utilizado o *Dialogflow*, uma solução comercial da Google para criação de chatbots. Para extrair os temas que o *chatbot* iria responder, foram estudados:

- Os planos pedagógicos curriculares dos cursos superiores de licenciatura em Letras e bacharelado em Sistemas de Informação e dos cursos técnicos subsequentes de Logística e Eletroeletrônica;
- Site institucional do Instituto Federal de Alagoas: como documentos apresentados, normas, etc;
- Assuntos como: uso da biblioteca, localização e contatos telefônicos.

Dos temas foram ainda extraídos diversos subtemas para alimentar as perguntas que o *chatbot* iria responder.

Além do *Dialogflow*, utilizamos o *Heroku*²⁴ - que é uma plataforma de serviço em nuvem, que permite implementar uma página *web* e deixá-la disponível. Assim, permitindo um amplo acesso ao *chatbot* de qualquer outro dispositivo. Escolhemos a *Heroku* pois possui opção de serviço gratuito e sem tempo de validade. Testamos também a plataforma *000WebHost*²⁵, da empresa *Hostinger*, porém, apesar de possuir um plano gratuito, este é temporário. Outras ferramentas também foram utilizadas, como *NoSQL*²⁶. Todas as ferramentas utilizadas serão apresentadas brevemente nas próximas seções.

4.1 FERRAMENTAS UTILIZADAS

4.1.1 *NoSQL*

Banco de dados *NoSQL* é um termo vago para uma ampla gama de banco de dados que não utiliza a linguagem SQL (*Structured Query Language*). Esse termo surgiu como o nome de uma reunião em São Francisco, Estados Unidos, em 11 de

²⁴ Disponível em: <https://www.heroku.com/>. Acesso em 29 de novembro de 2022.

²⁵ Disponível em: <https://br.000webhost.com/>. Acesso em 29 de novembro de 2022.

²⁶ Disponível em: <https://aws.amazon.com/pt/nosql/>. Acesso em 29 de novembro de 2022.

junho de 2009, organizada por Johan Oskarsson, o seu propósito era reunir conhecimento sobre diversos banco de dados que se destoava dos tradicionais: Voldemort, Cassandra, Dynamite, HBase, Hypertable, CouchDB, MongoDB (Fowler e Sadalage, 2013).

Os bancos NoSQL possuem uma estrutura flexível, em muitos modelos não implementando o princípio ACID (acrônimo para Atomicidade, Consistência, Isolamento e Durabilidade) dos SQLs, sem esquema, adequadas para *clusters*. Existem diversos tipos: chave e valor, documento, grafos, coluna entre outros.

Apesar da popularidade dos bancos relacionais, o seu uso implica em algumas limitações. Uma delas é a incompatibilidade de impedância. As linguagens de programação utilizam outro paradigma de programação, comumente, Programação Orientada a Objetos, que contrasta com o paradigma SQL. Ou seja, um programa é estruturado tendo como base atributos, herança, classes, objetos porém deve realizar transações com um banco de dados que utiliza outra estrutura, tais como tuplas, relacionamentos, tabelas, etc. Por um breve período, houve entusiasmo com banco de dados orientado a objetos, porém, este não teve o mesmo sucesso que o paradigma de programação. Outra ferramenta que tenta atenuar este problema são os *frameworks* de mapeamento objeto-relacional, que convertem dados relacionais para o paradigma orientado a objetos e vice e versa. Porém, é uma solução que também vem com desvantagens. É comum, por exemplo, perda de performance. Outro fator adverso está em abstrair demais o esquema relacional, o que muitas vezes torna difícil gerar transações mais complexas e que não foram mapeadas pelo *framework* e torna a aplicação muito dependente.

As aplicações *webs* trouxeram uma nova forma de integrar sistemas, além da tradicional arquitetura monolítica com um servidor central e um banco de dados relacional. Desta forma, passou-se a utilizar APIs, tendo como destaque o modelo REST (FOROUZAN, 2010), que se comunicam através do protocolo HTTP (FOROUZAN, 2010), tornando possível inserir e consultar dados complexos de forma segura e, principalmente, permitindo uma integração flexível e fácil para integrar diversas equipes. Cada API pode utilizar um conjunto de ferramentas que melhor geram resultado, o que vai ao encontro dos bancos não relacionais.

Os banco de dados não relacionais permitem uma estrutura mais complexa de dados aninhados.

Como o nome sugere, o banco de dados de documentos se estrutura em documentos, uma estrutura de dados que permite reunir informações relacionadas de forma agregada; essas informações podem estar aninhadas em listas, atributos ou, até, mesmo outros documentos. Um documento pode, por exemplo, representar uma pessoa da seguinte forma: um atributo nome, cujo valor será do tipo texto, um atributo de telefones que será uma lista e outro atributo endereço, que será outro documento aninhado. De certa forma, um banco de dados de documentos também é chave e valor, porém o valor é uma estrutura complexa. O documento pode ser do tipo *XML*, *JSON* ou *BSON*²⁷. E um conjunto de documentos relacionados é chamado de coleção.

Essa estrutura, por não depender de relacionamento, são atrativos ao uso de *clusters*, ou seja, torna flexível o aumento da capacidade de processamento de forma horizontal, em vez de aumentar cada vez mais a capacidade de um servidor, é possível dispor várias máquinas trabalhando de forma coordenada. Assim, é possível ajustar a estrutura de hardware de acordo com a demanda das aplicações demandadas.

4.1.2 *DialogFlow*

Atualmente, ferramentas de processamento de linguagem natural utilizam recursos avançados de inteligência artificial, além de outras áreas do conhecimento, tais como linguística, gramática, psicologia, entre outros. Implementar tais ferramentas do zero não é uma tarefa trivial. Por isso, é muito comum o uso de *frameworks* ou plataformas, programas desenvolvidos por outras pessoas que auxiliam no desenvolvimento de determinados programas.

O *Dialogflow* é uma plataforma da Google para desenvolvimentos de linguagem natural que pode ser usada para criar *chatbots*, assistentes virtuais ou outras *interfaces* conversacionais e integra ferramentas para a conversão de texto para fala e fala para texto, compreensão da intenção dita pelo usuário, identificação de entidades, análise de sentimento, entre outras.

Sua implementação é muito simples. Utiliza de uma plataforma *web*, abstraindo a estrutura computacional complexa. É online, ou seja, não é necessário

²⁷ Disponível em: <https://www.mongodb.com/json-and-bson>. Acesso em 01 de dezembro de 2022.

instalar qualquer programa, sendo necessário apenas realizar o cadastro e efetuar o login.

É possível escolher entre duas versões do *Dialogflow*: CX e ES. A ES oferece recursos padrões da plataforma, é recomendado para pequenas e médias empresas; já o CX oferece recursos avançados e é recomendado para grandes empresas.

O ponto de partida no *Dialogflow* é a criação de uma agente. No contexto de *chatbot* ou assistente virtual, podemos pensar em um agente realmente como a personalização da interface. Do ponto de vista técnico, é o módulo de processamento de linguagem natural que agrega as demais soluções. Foi criado tendo por objetivo entender as nuances da linguagem.

Com o agente criado, o próximo passo é criar as *intents*, ou intenções. Nesta parte, queremos captar a essência no que o usuário espera interagindo com o agente para através de sua intenção lhe prover uma resposta adequada. A *intent* é composta por alguns elementos, primeiro lhe damos um nome. Depois é necessário cadastrar diversas frases de treinamento, que são possíveis frases que o usuário pode dizer ao agente; através delas a inteligência artificial do *Dialogflow* será capaz de identificar a intenção do usuário. Não é necessário o cadastramento de muitas frases de treinamento, o *Dialogflow* recomenda pelo menos 20. Por fim, é possível cadastrar uma ou mais respostas que será emitida para o usuário. É importante destacar a diferença com a Eliza (Weizenbaum, 1966), que buscava por palavras-chaves de forma rudimentar, procedimento simplório para lidar com as relações complexas da linguagem.

Essa é a estrutura básica de uma *intent*, porém há outras características interessantes. É possível, por exemplo, configurar um contexto. Por que isso? Considere a seguinte frase: “quero comprar”. Ela não produz muito significado. Porém, imagine que antes dessa frase, foi dito ao receptor: “este produto custa 1,99 reais”. Agora faz sentido, porque entendemos o contexto. Dessa maneira, o agente é capaz de ter uma conversa mais fluida.

Outra funcionalidade interessante contida nos *intents* são as entidades e os parâmetros. Entidades basicamente são estruturas de dados para o controlar a extração de informação das entradas do usuário, podem ser entidades predefinidas do sistema, por exemplo, hora, data, endereço, ou personalizadas, por exemplo, em uma faculdade podemos criar uma entidade para representar o nome dos cursos.

Os parâmetros possuem uma forte relação com as entidades, são dados que desejamos extrair das entradas do usuário. Por exemplo, suponha que através de um *chatbot*, um usuário esteja agendando uma consulta médica. Para tanto, se faz necessário que ele informe o horário e para qual especialidade. De forma que o agente será capaz de extrair esta informação e até mesmo cadastrar em um banco de dados. No *Dialogflow* podemos configurar um parâmetro como obrigatório e cadastrar respostas personalizadas para extrair essas informações. Desta forma, o agente irá solicitar ao usuário até que ele informe.

Todos os conceitos explanados até aqui podem ser implementados com *Low Code*, ou seja não é necessário conhecimento de programação, o *Dialogflow* possui uma *interface* que permite apenas o cadastramento de dados. Entretanto, quando for necessário devolver uma resposta complexa ao usuário, que possua condicionantes internas ou externas ou que acessam bancos de dados ou APIs, por exemplo, será preciso desabilitar o modelo de resposta padrão e habilitar o *fullfilment*.

O *fullfilment* na versão ES pode ser implementado com a integração de ferramenta externa ou com um editor integrado na plataforma *web* (chamado de *inline editor*). É uma estrutura de código em *Nodejs*²⁸, um ambiente de programação que utiliza a linguagem de programação *EcmaScript*²⁹, mais conhecida como *Javascript*. É possível instalar dependências inserindo no arquivo *package.json*, que é usado para configurar o ambiente *Node.js*. E com isso, inserir ferramentas para realizar consultas externas através do protocolo HTTP, acessar banco de dados ou apenas entregar uma resposta personalizada para um determinado tipo de interação.

Outro ponto importante de destacar é a vasta quantidade de integrações que o *Dialogflow* permite com outros ambientes, que podem permitir interações sem áudio, texto ou em ambos. A documentação do *Dialogflow* as classifica como: “Integrações de telefonia incorporadas por parceiros”: *Voximplant, Twilio, SignalWire, Genesys, Avaya, AudioCodes*. “Integrações do próprio *Dialogflow*”: Aplicativo de mensagens do *Dialogflow, Dialogflow Phone Gateway, Web Demo do Dialogflow, Messenger do Facebook*, Local de trabalho no *Facebook, Google Assistente* (legado), *Hangouts Chat, LINE, Slack, Telegram*; “Integrações de código aberto

²⁸ Disponível em: <https://nodejs.org/en/>. Acesso em 29 de novembro de 2022.

²⁹ Disponível em: <https://www.ecma-international.org/>. Acesso em 29 de novembro de 2022

fornecidas pelo Google”: *Viber*, *Twitter*, Mensagens de texto do *Twilio*, Mensagens de IP do *Twilio*, *Spark (Cisco Webex)*, *Skype* e *Kik*, e; Integrações independentes.

4.1.3 JAVASCRIPT E NODE.JS

Neste trabalho, foi utilizado o *Node.js* para implementar algumas funcionalidades no *chatbot* e também para deixar a página *web* disponível. Logo, é útil termos uma breve introdução desse programa e da linguagem que ele utiliza, o *JavaScript*.

O JavaScript (Flanagan, 2013) foi criado em 1995 com o propósito de inserir funcionalidades de um programa nas páginas *web* do navegador Netscape. Com ele, as páginas HTML podem deixar de ser simples texto estático e se transformarem em verdadeiros programas.

Apesar do nome “*JavaScript*”, ela não tem nenhuma relação com a linguagem de programação *Java*, embora possuam sintaxe levemente parecida, pois ambas foram escritas com a linguagem de programação C. O nome foi escolhido por decisão de marketing, que tentava atrair um pouco da atenção destinada à linguagem *Java* (Flanagan, 2013).

Devido a marca registrada e a falta de controle de como o *JavaScript* era implementado pelos navegadores, resultado em várias versões da mesma linguagem, a Netscape resolveu enviar a linguagem para a ECMA, European Computer Manufacturer’s Association, para padronizá-la. Neste momento, a linguagem foi batizada como *ECMAScript*, porém continuou sendo chamada de *JavaScript* pela maioria das pessoas, desta maneira “*JavaScript*” e “*ECMAScript*” são basicamente sinônimos.

JavaScript é uma linguagem de alto nível, dinâmica, interpretada e não tipada, conveniente para estilos de programação orientados a objetos e funcionais. A sintaxe de *JavaScript* é derivada da linguagem *Java*, das funções de primeira classe de Scheme e da herança baseada em protótipos de Self. (Flanagan, 2013)

De alto nível significa que está mais próxima da linguagem natural que a de máquina. *JavaScript* foi criada com a intenção de ser fácil de aprender e compreender.

Na linguagem *JavaScript* não é preciso inserir o tipo de cada variável, que é inferido quando a variável recebe algum valor, por isso o termo “não tipada”.

JavaScript utiliza tipos, porém ela é fracamente tipada, quando uma variável recebe um valor de determinado tipo pode receber outro valor de tipo diferente, por isso dizemos que ela é dinâmica. Por exemplo, se recebe do tipo *String* (texto) pode receber depois do tipo *Number* (número) sem qualquer erro.

JavaScript incorpora tenuemente diversos paradigmas de programação, principalmente o Paradigma de Orientação a Objetos e o Funcional. Em relação a Orientação a Objetos, utiliza conceitos como herança e polimorfismo, mas, até o momento, não é possível utilizar métodos *privados*. Em relação ao Paradigma Funcional, utiliza funções de primeira classe e *Currying*.

“A linguagem *JavaScript* básica define uma API mínima para trabalhar com texto, arrays, datas e expressões regulares” (Flanagan, 2013). O sistema hospedeiro, ou seja, o ambiente que irá utilizar a linguagem *JavaScript* deve implementar funcionalidades de entrada e saída, além de poder incluir outras funcionalidades.

Até 2009, esse ambiente hospedeiro eram os navegadores. Porém, neste ano, foi criado o *Node.js*, um ambiente que utiliza o *JavaScript* nos servidores.

O sucesso do *Node.js* está na forma que implementa função de entrada e saída e chamadas HTTP de forma assíncrona através de um *loop* de eventos. É uma forma elegante de resolver a ociosidade do programa, pois não o interrompe enquanto aguarda uma requisição terminar.

4.2. MARVIN

Tudo começa com a criação do agente, módulo que agrupa o *chatbot*. O propósito de um *chatbot* é comunicar informações de forma mais humanizada, por isso é prática comum que possua um nome. O *chatbot* deste trabalho foi batizado de Marvin. A inspiração para o nome veio do livro O Guia do Mochileiro das Galáxias de Douglas Adams (Adams, 2010). Nele, Marvin é um robô com alta capacidade de raciocínio, mas com sensibilidade humana, que o torna depressivo e mal humorado, pois, apesar de sua capacidade, está condenado a fazer tarefas repetitivas e manuais. Desta forma, o Marvin é um arquétipo para muitos dos trabalhos modernos. Apesar da alta capacidade humana de criação, as pessoas estão condenadas a atividades repetitivas. Dito isto, as tecnologias da informação, muitas vezes, podem atenuar este problema automatizando e simplificando tarefas.

O próximo passo é a modelagem dos *intents*, onde iremos cadastrar as ações do *chatbot* de acordo com a intenção dos usuários. Para tanto, criamos algumas classificações: matrícula, avaliação, biblioteca, calendário acadêmico, quais cursos existem, atividades complementares, sobre o campus, informações gerais de um curso, extensão. Cada uma dessas classificações se tornou um *intent*. A maioria deles se desdobram em outros de acordo com a navegação dos usuários.

Para cada *intent* foram cadastradas frases de treinamento, que são frases que possivelmente os usuários podem inserir. A escolha das frases foi intuitiva, já que a inteligência artificial do *Dialogflow* é capaz de expandir o significado e inferir em outras expressões.

As respostas do *chatbot* foram extraídas manualmente através da leitura dos planos pedagógicos dos cursos superiores de Bacharelado de Sistemas de Informação e Licenciatura em Letras e dos cursos técnicos subsequentes de Logística e Eletroeletrônica. E também de instruções técnicas e site institucional. Por exemplo, as informações referente a matrícula foram encontradas em uma FAQ³⁰ no site institucional, já as informações referentes a trancamento de matrícula compulsório foram extraídas da Resolução 10/CS de 2015.

Como forma de integração do *chatbot* foi escolhido o *Dialogflow Messenger*, que apesar de estar em sua versão Beta, demonstrou estabilidade durante a produção. A proposta é de que o Marvin esteja disponível em texto de forma

³⁰ Frequently Asked Questions, ou Perguntas Feitas Frequentemente

desacoplada de uma plataforma específica, para que possa ser acessado em uma página *web*. O *Dialogflow Messenger* foi escolhido em relação a integração de mensagem tradicional porque possui um conjunto mais rico de componentes visuais.

Depois de criar e configurar o *chatbot*, é necessário torná-lo disponível para os usuários. Para tanto, foi desenvolvido um projeto *Node.js* que teria a simples tarefa de deixar a página *web* disponível. O *Dialogflow* possui um caixa de diálogo para interagir com o *chatbot* que pode ser inserida em uma página HTML, para tanto basta copiar e colar um conjunto de *tags* HTML.

Além da caixa de diálogo, a página *web* segue as características de uma *landing page*: um design simples e estático, que tem por objetivo orientar o usuário a um comportamento específico. O conceito vem da área de Marketing e normalmente está vinculado à conversão de vendas em plataformas digitais. A página orienta o usuário a utilizar a caixa de diálogo e introduz o *chatbot*, seu avatar e um breve texto.

A página também tem como princípio a responsividade. O *design* se adequa de acordo com a largura e a altura da tela. Como resultado, usuários acessando de telefones celulares, tablets ou monitores terão o tamanho da fonte e das imagens, a posição dos elementos ajustados, como mostra a Figura 1 abaixo.

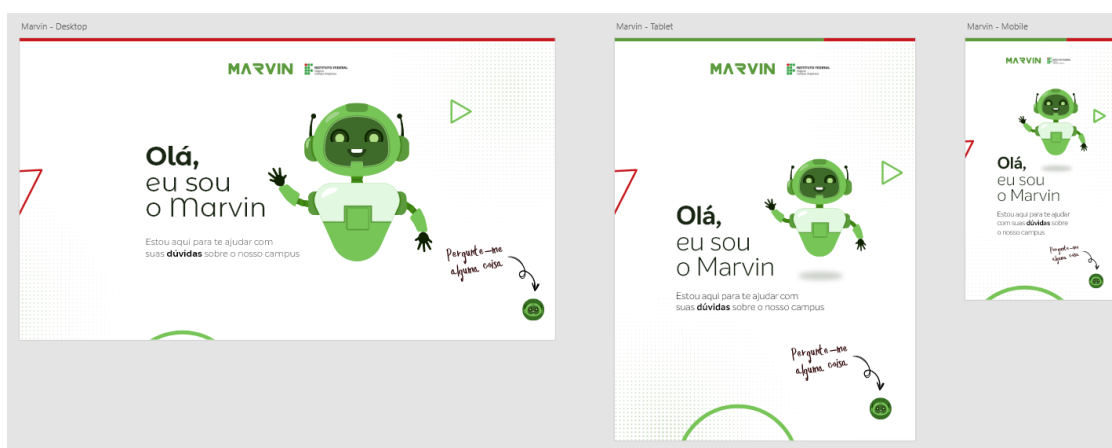


Figura 1 — Protótipo das telas do Marvin em dimensões diferentes

Para tornar a página disponível através na internet, foi utilizada a plataforma da *Heroku* para instalar o servidor criado com *Node.js*. O servidor manterá a página e seus recursos disponíveis.

Na plataforma *Heroku*, foi utilizada a integração com o *Github*³¹ para a instalação e atualização automática da página. Desta forma, todas as vezes que o repositório no *Github* for atualizado, a página é atualizada. O repositório está disponível no endereço <https://github.com/jon-sousa/bot/tree/main/public> e a página do *chatbot* está disponível no endereço <https://marvin-robo.herokuapp.com/>.

4.2.1 AVATAR, IDENTIDADE VISUAL E INTERFACE

A identidade visual merece uma atenção especial porque nela está o poder de chamar a atenção e começar a criar uma ponte para uma relação contínua com o usuário.

O Marvin possui sua identidade visual apoiada na identidade visual do IFAL. Logo, ao acessar a página do Marvin, o usuário consegue associar o *chatbot* a instituição, devido a utilização de elementos gráficos que conversam entre si e trazem referências visuais. Para o logotipo do Marvin, foi utilizado uma tipografia moderna que remete a tecnologia com traços fortes e simples para não poluir a composição.

Para o avatar, buscamos referências em *bots* disponíveis na *web* e nos inspiramos em desenhos vetoriais disponíveis em bancos de imagens onde, a partir dessas buscas, foi encontrado um avatar o qual foi alterado e adaptado a fim de atender as necessidades do nosso projeto. Dessa forma, o avatar do Marvin ganhou forma e foram criadas 8 imagens que deverão ser utilizadas durante a interação do usuário no chat. A Figura 2 mostra o logotipo e as 8 posições do avatar.

³¹ Disponível em: <https://github.com/> Acesso em 29 de novembro de 2022.

MARVIN

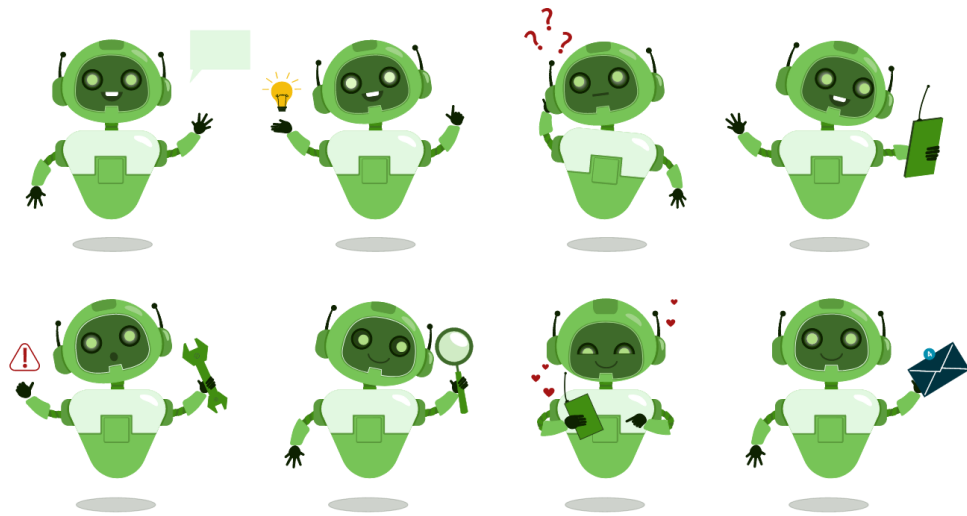


Figura 2 — Logotipo e avatar do Marvin

Antes de implementar a *interface* na página do *chatbot*, foi projetado um protótipo de alta fidelidade no software *XD*, da *Adobe Creative Cloud*³². “O *Adobe XD* é uma ferramenta de prototipação de aplicativos e sites. Basicamente ele funciona com *artboards* que podem ser ligados entre si, dando uma visão geral de navegação. O programa é dividido em duas funcionalidades: *Design* e *Protótipo*.” (SANTANA, 2016).

Essa etapa é de extrema importância para evitar retrabalho durante a implementação do código, pois nos permite enxergar como será a visualização final do produto em diversos tamanhos de tela, proporcionando uma boa experiência aos usuários independentemente do dispositivo em que será acessado. A Figura 3 mostra o protótipo na versão *Desktop* do Marvin.

³² Disponível em: <https://www.adobe.com/br/creativecloud.html>. Acesso em 29 de novembro de 2022



Figura 3 — Protótipo da versão desktop do Marvin

O objetivo da *interface* gráfica da *landing page* do Marvin foi proporcionar uma familiaridade ao usuário através de elementos que fazem referência à identidade visual do IFAL. Para isso, foram utilizadas cores da paleta, disponível no manual de comunicação. Além disso, a tipografia e outros elementos gráficos também foram incluídos cuidadosamente na composição sem descaracterizar a identidade visual da instituição para não afetar a usabilidade da *interface*.

Após o desenvolvimento do protótipo, implementamos a *interface* utilizando HTML5 e CSS3 (SILVA, 2018). HTML5 (*HyperText Markup Language*) é uma linguagem de marcação de texto utilizada para produzir páginas *web*, trata-se da versão mais recente da linguagem, trazendo diversos novos atributos em relação às versões anteriores. Na página desenvolvida para o Marvin, utilizamos o HTML5 juntamente com o CSS3 para a estilização da mesma.

Segundo Soares (2006, p. 49), CSS (*Cascading Style Sheets*), ou folhas de estilo, é uma ferramenta que possibilita personalizar toda a aparência de uma página *web*, tornando a experiência do usuário mais interessante. A estilização de uma página *web* pode ser modificada na própria página ou interativamente através de regras de estilos definidas em um arquivo separado como fizemos no caso do Marvin. A Figura 4 mostra a tela de visualização do usuário ao acessar o endereço da landing page do Marvin.



Figura 4 — Tela inicial do Marvin

Além disso, apesar da limitação de personalização do *DialogFlow*, o chat de diálogo também foi personalizado para trazer mais harmonia à página. Logo, as cores dos botões, textos e balões de fala foram personalizadas para se adaptar a identidade visual do Marvin como mostra a Figura 5.

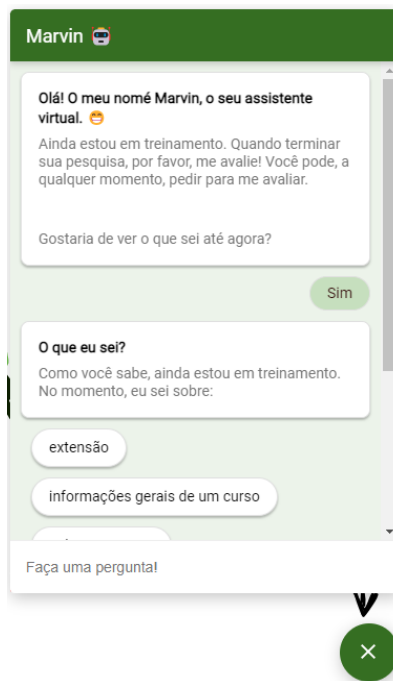


Figura 5 — Tela de diálogo com o Chatbot

4.2.2 Banco de Dados NoSql: *Google Firestore*

O banco de dados foi utilizado para salvar as respostas das avaliações do usuário. Durante a avaliação, o *Dialogflow* irá salvar em memória cada resposta do usuário através dos parâmetros de contexto, um objeto *JavaScript*. Por fim, este objeto é tratado para retirar atributos desnecessários à avaliação e há a requisição ao banco de dados.

A escolha de um banco de dados do tipo documento se deu por sua flexibilidade, pois não impõe as restrições de um banco de dados tradicional, o que permite manter o controle na aplicação. As avaliações terão quantidade de atributos variáveis, dependendo das respostas do usuário. Assim sendo, cada documento também terá uma quantidade diferente de atributos.

O formato de documento representa de forma adequada as avaliações, pois permite visualizá-las individualmente como um todo. Caso fosse utilizado um banco de dados relacional, seria necessário realizar vários “*joins*” entre diversas tabelas para visualizar cada documento.

Quando um agente no *Dialogflow* é criado, sua estrutura está inserida em um projeto *Google Cloud*. Desta forma, foi escolhido o banco de dados *Google Firestore*³³ pela sua fácil integração ao *chatbot*. O *Firestore* funciona diretamente na nuvem, sem servidor. Sua configuração e uso são simples.

```
const Firestore = require('@google-cloud/firestore');
const PROJECTID = 'marvin_rqsma';
const COLLECTION_NAME = 'avaliacoes';

const firestore = new Firestore({
  projectId: PROJECTID,
  timestampsInSnapshots: true
});
```

Figura 6 — Configuração do banco de dados no código do *chatbot*

A Figura 10 mostra a configuração do *Firestore* no código do *chatbot*, chamado pelo *Dialogflow* de *Fullfilment*, que é basicamente um projeto em *Node.js* integrado ao *Google Cloud*. É necessário incluir a biblioteca do *Firestore* como dependência no arquivo *Package.json*, o arquivo de configuração dos repositórios *Node.js*. Não é necessário inserir nenhuma credencial ou *tokens* de autenticação pois o *Firestore* já está integrado ao projeto. Para utilizar as funções de escrita,

³³ Disponível em: <https://cloud.google.com/firestore/>. Acesso em 30 de novembro de 2022

leitura, atualização ou exclusão basta utilizar o objeto criado através da *interface* provida pela biblioteca.

O Firestore possui outras vantagens, como: possuir alta disponibilidade, de 99,99% a 99,999% conforme informado em sua página *web*, possuir sincronização em tempo real e ter transações ACID (SILVA, 2021).

O *Firestore* provê um suporte *web* em que é possível consultar o uso, os documentos inseridos e criar consultar utilizando estruturas condicionais, de ordenação e contagem.

4.3 Uma tentativa frustrada: *Web Scraping*

O Sistema Integrado de Gestão de Atividades Acadêmicas (SIGAA)³⁴ permite a consulta do acervo da biblioteca e era nossa intenção integrar a consulta ao acervo na ferramenta, de forma que fosse possível perguntar ao *chatbot* se determina do livro está disponível e obter a resposta.

Porém, não há uma API ou outra fonte de pesquisa para acessar estes dados diretamente. A própria consulta no SIGAA não é fluída: há muitas entradas e é necessário primeiro consultar se o livro existe no acervo para depois detalhar se está disponível nas bibliotecas dos campi.

Uma possível solução foi usar um *web crawler*, também conhecido como *web scraping* (Mitchell, 2019). É a técnica de extrair informações de uma página HTML navegando através de sua estrutura. Foi desenvolvido com um *web crawler* com a biblioteca *Puppeteer*³⁵ do ambiente de programação *Node.js*. O *Puppeteer* é uma API de alto nível em que, através do controle do *Chromium*, possibilita selecionar atributos, textos, inserir informação, pressionar botões, entre várias outras funcionalidades. Resumidamente, o *Puppeteer* simula um navegador e disponibiliza ferramentas para extrair informações.

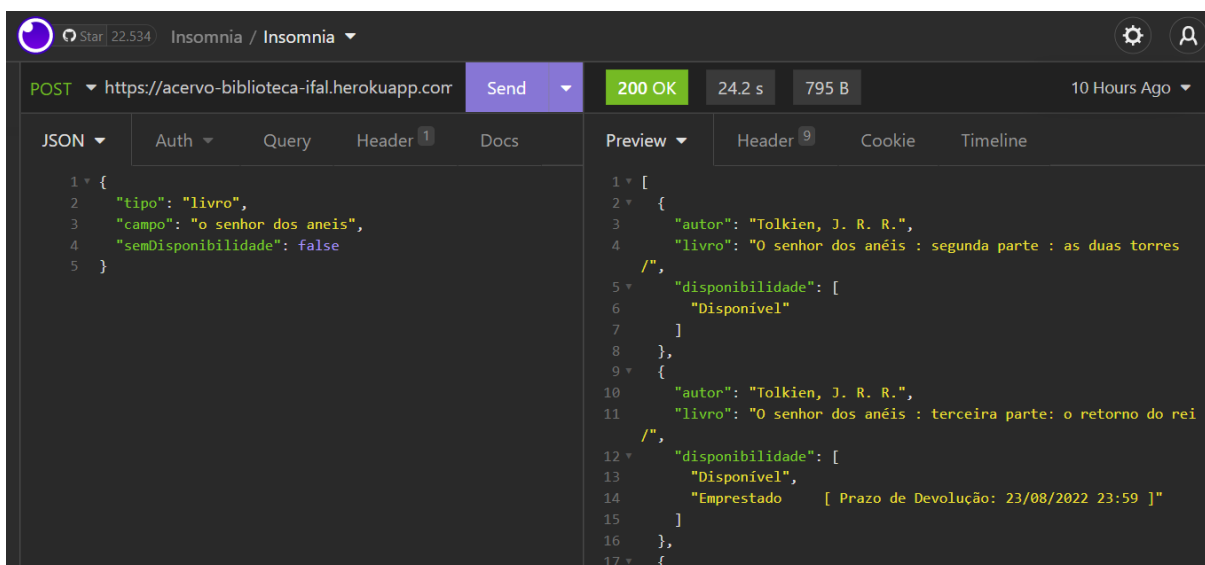
Foi desenvolvida uma API em *Node.js* que recebe através de uma requisição HTTP (método *POST*), no corpo da requisição no formato JSON, as informações de autor, livro, assunto e se deseja a informação da disponibilidade do livro. Com estas informações, o *Puppeteer* abre a página do SIGAA (endereço: <https://sigaa.ifal.edu.br/sigaa/public/biblioteca/buscaPublicaAcervo.jsf?aba=p-biblioteca>) através do navegador virtual, insere as informações nos campos de livro, autor e

³⁴ Disponível em: <https://sigaa.ifal.edu.br/>. Acesso em 29 de novembro de 2022.

³⁵ Disponível em: <https://pptr.dev/>. Acesso em 29 de novembro de 2022.

assunto, escolhe a biblioteca do Campus Arapiraca e pressiona o botão de pesquisar. Se na requisição estiver selecionado para inserir a disponibilidade. O *Puppeteer* irá pressionar o botão de detalhamento de cada livro, selecionar as informações de disponibilidade e inserir nos livros pesquisados. A cada iteração terá que pressionar o botão do livro correspondente, ir para a página de detalhamento, extrair a informação, depois voltar para a página anterior para pressionar o botão do próximo livro.

Esta API foi implantada na plataforma de serviço em nuvem *Heroku*, está disponível através do endereço <https://acervo-biblioteca-ifal.herokuapp.com> e funciona conforme o esperado. O problema é que esta tarefa de extrair informações através do próprio navegador é custosa computacionalmente, além do problema de latência de rede, o que torna a API demorada para responder, em torno de 10 segundos. O *Dialogflow* possui uma tolerância muito pequena quanto ao tempo que o usuário pode ficar esperando pela resposta, pois pode frustrar a experiência do usuário. Com isso, a chamada HTTP no *fullfiment* do agente acarretava em erro. Então, como única alternativa, quando o *chatbot* é demandado quanto ao acervo da biblioteca, apenas informa o endereço da página do SIGAA para consulta.



```
POST https://acervo-biblioteca-ifal.herokuapp.com 200 OK 24.2 s 795 B 10 Hours Ago
JSON Auth Query Header 1 Docs Preview Header 9 Cookie Timeline
1 {
2   "tipo": "livro",
3   "campo": "o senhor dos aneis",
4   "semDisponibilidade": false
5 }
1 [
2   {
3     "autor": "Tolkien, J. R. R.",
4     "livro": "O senhor dos anéis : segunda parte : as duas torres"
5   },
6   "disponibilidade": [
7     "Disponível"
8   ],
9   {
10    "autor": "Tolkien, J. R. R.",
11    "livro": "O senhor dos anéis : terceira parte: o retorno do rei"
12  },
13  "disponibilidade": [
14    "Disponível",
15    "Emprestado [ Prazo de Devolução: 23/08/2022 23:59 ]"
16  ],
17 }
```

Figura 7 — Comandos para chamada HTTP para a API

A Figura 6 mostra uma chamada HTTP para a API realizada através do *software Insomnia*, onde é enviado uma requisição do tipo POST e em seu corpo é inserido um arquivo de texto do tipo JSON. Neste arquivo, o atributo “tipo” determina se a busca é por livro, autor ou assunto, “campo” é o texto de busca e “semDisponibilidade” é um atributo booleano (do tipo verdadeiro ou falso) que irá

determinar se a API retornará a situação de cada exemplar na biblioteca do IFAL, se está disponível ou emprestado. O retorno da requisição também é um arquivo de texto do tipo JSON. A Figura 7 mostra a página de consulta ao acervo da biblioteca disponível no endereço <https://sigaa.ifal.edu.br/sigaa/public/biblioteca/buscaPublicaAcervo.jsf?aba=p-biblioteca>











TÍTULOS ENCONTRADOS (1 A 5 DE 5)				
Autor	Título	Edição	Ano	Qtd.
Tolkien, J. R. R.	O senhor dos anéis : segunda parte : as duas torres /	2. ed.	2000.	3  
Tolkien, J. R. R.	O senhor dos anéis : terceira parte: o retorno do rei /	2. ed.	2000.	6  
Tolkien, J. R. R.	O senhor dos anéis : primeira parte: a sociedade do anel /	2. ed.	2000.	3  
Tolkien, J. R. R.	O senhor dos anéis : segunda parte : as duas torres /		2002.	2  
Tolkien, J. R. R.	O senhor dos anéis /		2001.	1  
1 a 5 de 5 título(s) encontrado(s)				

Figura 8 — Consulta ao acervo da biblioteca

A Figura 8 mostra o resultado da primeira consulta.

EXEMPLAR(ES) 1 A 3 DE 3				
Código de Barras	Tipo de Material	Coleção	Status	Situação
CAMPUS ARAPIRACA				
156722	Livro	Acervo Circulante	REGULAR	Disponível
<i>Localização:</i> 823 T649s 2.ed. ex.1				
<i>Mostrar Detalhes</i>				
CAMPUS SANTANA DO IPANEMA				
SIP0324	Livro	Acervo Circulante	REGULAR	Disponível
<i>Localização:</i> 823 T649s 2.ed. ex.1				
<i>Mostrar Detalhes</i>				
CAMPUS VIÇOSA				
CVIC005	Livro	Acervo Circulante	REGULAR	Disponível
<i>Localização:</i> 823 T649s 2. ed. ex.1				
<i>Mostrar Detalhes</i>				

Figura 9 — Resultado de uma consulta ao acervo

Quando é pressionado o ícone de lupa para detalhar a disponibilidade do livro, é mostrado a situação em todos os campi e não apenas no campus selecionado anteriormente. Para conseguir acessar a disponibilidade de todos os resultados da consulta é necessário ir e voltar várias vezes em cada resultado.

A página foi construída sem utilizar identificadores dos elementos HTML, como "id", "class" ou "data attributes". E sua estrutura está inserida em tabelas (elemento HTML *Table*). Isto torna o processamento pelo *Web Scraping* ainda mais custoso, pois tem que navegar através do DOM sem qualquer atalho. São necessárias várias chamadas HTTP para retornar todos os dados solicitados. A Figura 9 mostra como o Marvin pode de forma fácil redirecionar o usuário para a consulta ao acervo da biblioteca.

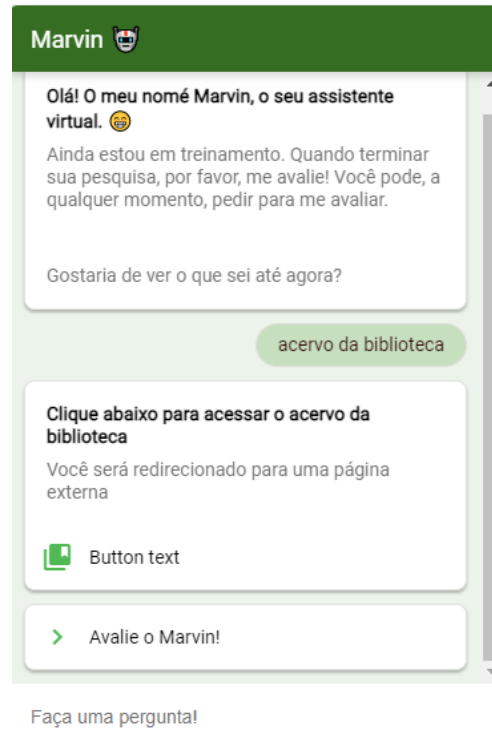


Figura 10 — Chat de diálogo com botão de redirecionamento ao acervo da biblioteca

No próximo capítulo é apresentado como foi realizada a avaliação do Marvin.

5. AVALIAÇÃO DO MARVIN

Na última etapa, os usuário utilizaram a ferramenta livremente, perguntando sobre qualquer assunto que o Marvin pudesse responder. E no final, responderam a um questionário realizado pelo próprio Marvin, já que o *Dialogflow* permite a conexão a um banco de dados. Neste questionário, o usuário responde quais as principais dificuldade e facilidade no uso da ferramenta, de forma a identificar melhora ou piora em relação ao uso de outras plataformas. Além das respostas ao questionário, analisamos quais assuntos foram mais solicitados e como houve a interação entre os usuários e o *chatbot*. A entrevista foi semi-estruturada, guiada, com questões abertas e fechadas. A forma de observação foi espontânea, ou seja, não houve interferência dos pesquisadores na comunidade.

A divulgação do Marvin se deu através de panfletos e mídias sociais. Realizamos a entrega de panfletos e colagem de cartazes nos principais murais do campus para atingir o público interno (Figura 11), além da divulgação pelas redes sociais como em grupos de *Whatsapp* e *Instagram* para atingir o público externo e interno (Figura 12).



Figura 11 — Divulgação do Marvin nas dependências do campus



Figura 12 — Perfil oficial do campus Arapiraca contribuindo para divulgação

Primeiro solicitamos que o usuário utilizasse livremente o *chatbot*. Ao final de cada resposta, um botão no próprio *chat* sugere que o avalie. Quando o usuário o deseja avaliar, o Marvin efetua algumas perguntas a fim de obter as informações referentes ao problema da pesquisa. A maioria das perguntas são do tipo *sim* ou *não*. Sempre que for dada uma resposta negativa, será solicitado o motivo da resposta. Desta forma, o trabalho pretende não apenas medir a aceitação, como também entender quais os principais pontos de frustração ou melhoria. Neste momento, o Marvin utiliza as entradas digitadas como resposta, desta maneira, se o usuário tentar acessar outro *intent*, será redirecionado para a avaliação. A avaliação ficou disponível entre os dias 20 de outubro e 11 de novembro de 2022 para o público interno ou externo do Instituto Federal de Alagoas Campus Arapiraca. 42 pessoas avaliaram o Marvin, conforme apresentado na Tabela 2, destes 41 o avaliaram de forma geral positivamente o uso do chatbot.

Tabela 2 – Quantidade de pessoas que gostaram e não gostaram do Marvin segundo avaliações

Resposta	Quantidade de pessoas
Gostaram	41
Não gostaram	1

Na Figura 13, um fluxograma mostra o curso de interações do usuário para responder o formulário de avaliação.

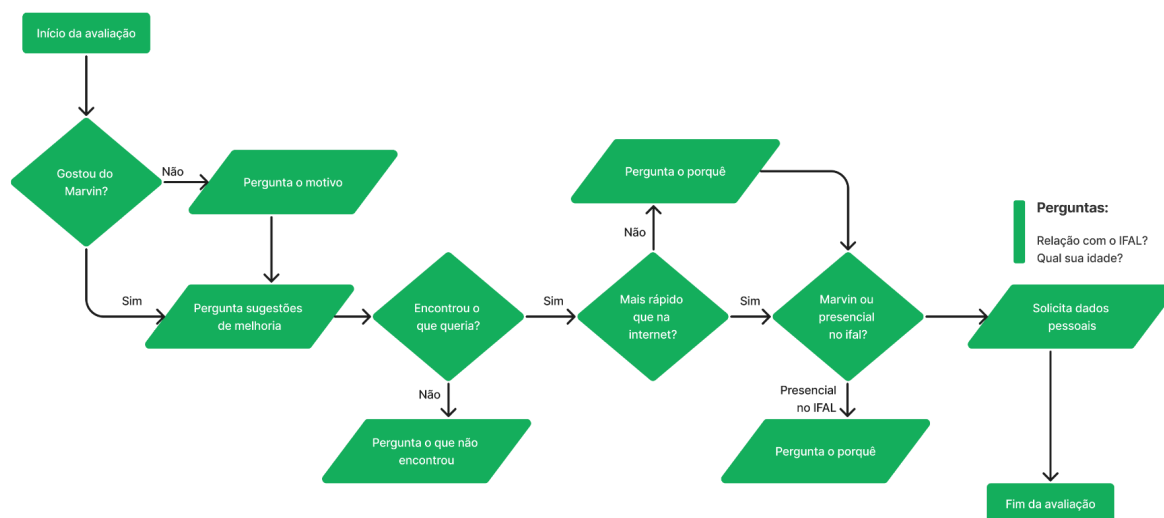


Figura 13 — Fluxograma de interação com a pesquisa de satisfação

Quando o usuário buscar por uma informação que o *chatbot* não possa compreender, o dado de pesquisa será salvo no banco de dados. Isto tem por objetivo identificar o que o usuário esperava encontrar e quais os pontos falhos da aplicação.

O uso livre do usuário sem qualquer validação em um ambiente não controlado ocasionou que uma pequena parte das avaliações apresentassem dados incorretos ou inconsistentes. Apesar de causar uma pequena perda de informações referente as perguntas da avaliação, houve um aumento na compreensão de como o usuário interage com o *chatbot*.

Das respostas, apenas uma pessoa não gostou da aplicação (Tabela 2). Como justificativa, respondeu: “tenho dúvidas”, o que sugere que o usuário possuía dúvidas mais específicas não satisfeitas pelo *chatbot*. O que corrobora a hipótese que em dúvidas mais específicas o usuário preferiria o atendimento presencial no campus.

Tabela 3 – Preferência entre Marvin ou atendimento presencial no IFAL

Resposta	Quantidade
Marvin	36
Presencial no IFAL	6

Ao ser perguntado se preferem consultar o Marvin ou o atendimento presencial no Campus em eventual dúvida, seis pessoas responderam que

prefeririam o atendimento presencial. Como justificativa, uma pessoa respondeu que prefere o contato humano, os demais deram respostas inválidas, sem contexto, o que indica que o usuário tentou realizar uma nova pergunta ao *chatbot* em vez de responder à avaliação.

Tabela 4 – Buscar no Marvin é mais rápido que na internet?

Respostas	Quantidade
Sim	34
Resposta inválida	5
Não	3

Quando perguntado se a busca no Marvin foi mais rápida que nas ferramentas de busca na internet. Três pessoas responderam que não, e cinco deram respostas inválidas.

Tabela 5 – Relação entre os usuários e o IFAL

Respostas	Quantidade
Aluno de SI	17
Não aluno	11
Integrado	7
Servidor	5
Letras	2

A Tabela 5 mostra a relação dos tipos de usuários que avaliaram a aplicação. Em maior número estão os alunos do curso superior de Sistemas de Informação. Logo em seguida, estão as pessoas que não são alunos. O Marvin nasceu com o propósito de ser uma opção aos alunos para obter informações em relação ao campus e seus cursos. Porém, está claro que o Marvin pode ser uma ótima ferramenta para o IFAL manter a comunidade externa atualizada, como por exemplo, em editais abertos, que acabam se embaralhando na página principal da instituição. Houve sugestões de melhoria relacionadas à ampliação dos assuntos e melhoria da capacidade de compreensão do *chatbot*.

A avaliação e as interações obtidas evidenciam o quão diferente é em relação às interações humanas, apesar do propósito do *chatbot* seja a de humanizar. No primeiro dia de avaliação, houve entradas inválidas em algumas respostas, causadas pelo recarregamento da página durante a avaliação. Como o *chatbot*

esperava pela resposta do usuário, quando a página era recarregada o *chatbot* aceitava o nome do *intent* inicial como resposta e pulava para a próxima resposta. Então foi necessário programar que, quando a página fosse recarregada, a avaliação voltaria para a pergunta em aberto. Esse comportamento não é esperado em uma interação entre humanos, não é comum uma pessoa simplesmente ir embora ao ser perguntada por alguma coisa.

O *Dialogflow* também fornece um conjunto de ferramentas para análise de dados. A Figura 14 mostra o gráfico com a quantidade de interações com o *chatbot* desde que começou o período de avaliações dos usuários. O pico de avaliações aconteceu em 03/11/2022 com 254 interações..

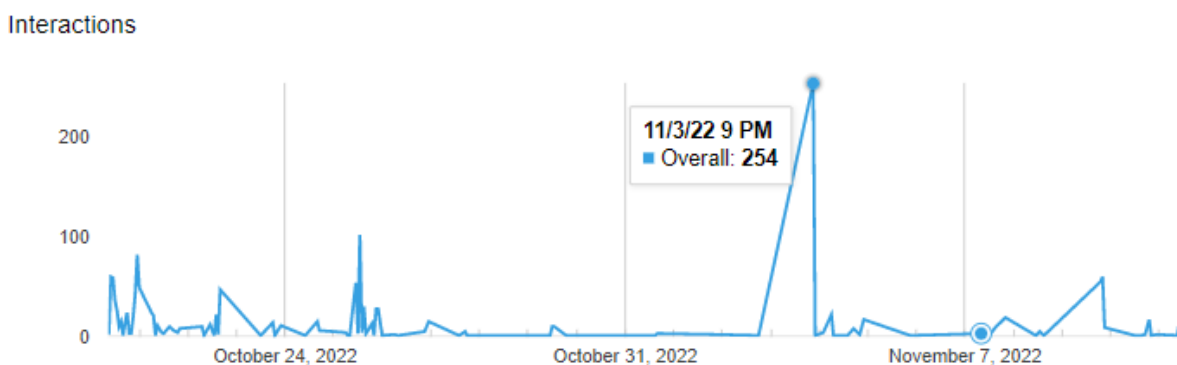


Figura 14 — Interações com o Marvin de outubro a novembro de 2022

A Tabela 6 exibe a quantidade de interações para cada *intent*. O *intent* com o sufixo “m-a” foram utilizados para a avaliação do *chatbot*. Como nomenclatura, foi utilizado o ponto para separar a classificação de forma hierárquica e o hífen, para preencher o espaço em branco entre palavras de um mesmo termo. Por exemplo, o *intent* sobre a matriz curricular do curso de Sistemas de Informação recebeu o título de “cursos.si.matriz-curricular”. Como exceção, há os *intents* “*Default Welcome Intent*”, “*Default Welcome Intent - yes*” e “*Default Fallback Intent*”, que mantiveram os nomes padrões da plataforma.

Tabela 6 – Quantidade de cada interação para os *intents* cadastrados

	Nome	Interações
0	matrícula	6
1	cursos.si.informacoes-gerais	14
2	m-a	128

3	m-a-6	86
4	m-a-9	239
5	m-a-7	44
6	m-a-2	19
7	estrutura.contato	5
8	estagio	8
9	extensao	7
10	biblioteca.biblioteca	17
11	matricula.documentos	2
12	cursos.eletronica.informacoes-gerais	1
13	m-a-4	89
14	extensao-cursos	4
15	curso.letras.atividades-complementares	1
16	biblioteca.biblioteca.biblioteca-virtual	6
17	cursos.si.matriz-curricular	1
18	Default Fallback Intent	80
19	curiosidades.matematica	1
20	atividades-complementares	3
21	m-a-3	120
22	cursos.logistica.matriz-curricular	1
23	Default Fallback Intent - yes	37
24	estrutura.atendimento	9
25	curiosidades.criador	1
26	matricula.cancelamento	2
27	calendario.calendario	24
28	index	15
29	avaliacao.prova-final	4
30	curiosidades.piada	2
31	curiosidades.sentido-da-vida	2
32	cursos.pcc	4
33	estrutura.localizacao	16
34	cursos.informacoes-gerais	8
35	m-a-8 - fallback	7

36	m-a-5	47
37	cursos.si.extensao.atividades	1
38	cursos.cursos	34
39	matricula.trancamento	1
40	m-a-8	142
41	avaliacao.avaliacao	8
42	cursos.si.atividades-complementares	4
43	estrutura-sobre.o.campus	35
44	curso.logistica.informacoes-gerais	2
45	cursos.logistica.praticas.profissionais	5
46	Default Welcome Intent	255
47	curso.letras.informacoes-gerais	1
48	biblioteca.biblioteca.acerto	9

Quando o chatbot é iniciado, o “*Default Welcome Intent*” é disparado, emitindo uma resposta de boas vindas. No final desta mensagem, é perguntado se o usuário deseja saber quais assuntos o Marvin é capaz de responder. Se obtiver resposta afirmativa, é mostrado um menu de opções. Conforme a Tabela 7, a maioria dos usuários preferiram realizar uma resposta direta, em vez de navegar pelo menu.

Tabela 7 – Preferência do usuário em resposta direta ou navegar pelo menu

Nome	Quantidade de interações
Default Welcome Intent - yes	37
Default Welcome Intent	255

6. CONCLUSÕES E TRABALHOS FUTUROS

Esse trabalho pretendeu trazer uma alternativa à comunidade do IFAL na busca de informações sobre os cursos e sobre o campus, a fim de melhorar a experiência dos mesmos ao encontrar essas informações que antes estavam disponíveis, de forma dispersa, apenas em instruções normativas, planos pedagógicos curriculares e site institucional.

Como hipótese, tínhamos que o *chatbot* traria uma melhora na experiência da comunidade do IFAL em relação a obtenção das informações devido à praticidade. E que, em comparação a busca na internet ou manuais/relatórios, o *chatbot* iria melhorar a experiência, pois a forma de busca no *chatbot* é mais adequada e fácil para os alunos. Em comparação a consulta à secretaria, assumimos que o *chatbot* traria melhorias por ser mais acessível, entretanto não suprirá a necessidade dos alunos em informações mais específicas ou pouco frequentes.

A partir dos resultados obtidos no presente trabalho, podemos afirmar que o *chatbot* Marvin alcançou resultados satisfatórios, por isso, acreditamos certamente que o Marvin consegue responder às dúvidas reais, que fazem parte da rotina dos discentes do IFAL e até mesmo de outros interessados no assunto. A quantidade de respostas afirmativas corrobora que o *chatbot* agrega na encontrabilidade de informações.

Com isso, a hipótese de que o *chatbot* traria uma melhora na experiência da comunidade do IFAL em relação a obtenção das informações devido à praticidade foi confirmada através do feedback dos próprios usuários que testaram o Marvin e responderam o formulário de avaliação disponibilizado.

Dessa forma, podemos concluir que o uso de *chatbots*, é uma alternativa viável para o problema de encontrabilidade da informação. Entretanto, é de extrema importância que melhorias sejam feitas de forma frequente para que seja possível atender às necessidades do público-alvo e, no caso do Marvin, sempre conseguir responder às dúvidas atuais dos usuários.

Como trabalhos futuros propõe-se: a integração do *chatbot* a outras plataformas, tais como, serviços de mensagens instantâneas ou redes sociais; integração com o SIGAA, a ampliação do número de assuntos que o *chatbot* é capaz de responder; e a inclusão de assuntos casuais que, apesar de não corresponder ao propósito do *chatbot*, podem aumentar o engajamento do usuário.

Também, propõe-se a criação de um formulário específico para que a comunidade possa sugerir assuntos para inserção no *chatbot* para manter a aplicação atualizada a novos assuntos e para que possa atender a diferentes demandas. Para tanto, é interessante criar elementos motivadores, como por exemplo inserir um campo na página para contribuidores ou mencioná-los no próprio *chatbot*.

7. REFERÊNCIAS

- ADAMOPOULOU, Eleni; Moussiades, Lefteris. Chatbots: History, technology, and applications. *Machine Learning with Applications*, v. 2, p. 100006, 15 de dezembro de 2019. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2666827020300062>. Acesso em: 28 set 2022.
- ADAMS, Douglas. O guia do mochileiro das galáxias. São Paulo : Arqueiro, 2010.
- ALLISON, DeeAnn. Chatbots in the Library: Is It Time?. *Library Hi Tech*. v.30.1. p.95-107. 2012. Disponível em: <https://www-emerald.ez133.periodicos.capes.gov.br/insight/content/doi/10.1108/07378831211213238/full/html>. Acesso em: 10 set. 2022.
- BARBIERI, C. Governança de dados: Práticas, conceitos e novos. 1. ed. Rio de Janeiro: Editora Alta Books, 2020.
- BARROS, Daniela Melaré Vieira; Guerreiro, Aníbal Martin. Novos Desafios Da Educação a Distância: Programação E Uso De Chatbots. *Espaço Pedagógico*. v.26.2. p.410-31. 2019. Disponível em: <http://seer.upf.br/index.php/rep/article/view/8743>. Acesso em: 10 set. 2022.
- COMARELLA, Rafaela Lunardi; Cafe, Lígia Maria Arruda. CHATTERBOT: Conceito, Características, Tipologia E Construção. *Informação & Sociedade, Estudos*. v.18.2. 2008. Disponível em: <https://periodicos.ufpb.br/ojs/index.php/ies/article/view/1758>. Acesso em: 10 set. 2022.
- COPPIN, B. Inteligência Artificial. 1. ed. Rio de Janeiro: LTC, 2010.
- FLANAGAN, David. 6.ed. JavaScript: o guia definitivo. Porto Alegre: Bookman, 2013.
- FOROUZAN, Behrouz A. Protocolo TCP/IP. [Porto Alegre, RS]: Grupo A, 2010. E-book. ISBN 9788563308689. Disponível em: <https://app.minhabiblioteca.com.br/#/books/9788563308689/>. Acesso em: 01 dez. 2022.
- FOWLER, Martin; Sadalage, Pramod J. NoSQL Essencial: Um Guia Conciso para o Mundo Emergente da Persistência Poliglota. 1. ed. São Paulo: Novatec, 2013.

- ILLESCAS-MANZANO, María D.; López, Noé Vicente; González, Nuno Afonso; Rodríguez, Carmen Cristofol. Implementation of Chatbot in Online Commerce, and Open Innovation. *Journal of Open Innovation*. v.7.2. p.125. 2021. Disponível em: <https://www.mdpi.com/2199-8531/7/2/125/htm>. Acesso em: 10 set. 2022.
- INSTITUTO FEDERAL DE ALAGOAS. RESOLUÇÃO Nº 10/CS de 20 de maio de 2015. [Aprova as alterações na Res nº 32/CS/2014]. Disponível em: <https://www2.ifal.edu.br/o-ifal/ensino/legislacao-e-normas/arquivos-legislacao/departamento-de-educacao-basica/resolucao-no-10-cs-2015-aprova-as-alteracoes-na-res-no-32-cs-2014-normas-org-didatica.pdf/view>. Acesso em: 26 nov. 2022.
- JOVELIANO, Daniel De Azevedo; Galli, Isaac Maciel; Santos Júnior, Gilson Nunes Dos; Silva, Mateus Rodrigues Almeida da; Benites, Cristiano Da Silva; Ribeiro, Francisco Carlos. Trabalhando Com a Deficiência Auditiva: Uma Proposta De Ensino a Distância Com O Uso De Chatbot. *RISTI : Revista Ibérica De Sistemas E Tecnologias De Informação*. v.29. p.135-47. 2020. Disponível em: <https://www.proquest.com/openview/fb8bfe36673b48be722692d37b7b8d05/1?q-origsite=gscholar&cbl=1006393>. Acesso em: 10 set. 2022.
- LEANDRO, Jonas Sousa; Moura, Matheus Breno dos Santos. Bot. [S. l.], 2022. Disponível em: <https://github.com/jon-sousa/bot/tree/main/public>. Acesso em: 27 nov. 2022.
- LEANDRO, Jonas Sousa; Moura, Matheus Breno dos Santos. Marvin. [S. l.], 2022. Disponível em: <http://marvin-robo.herokuapp.com/>. Acesso em: 27 nov. 2022.
- LEONHARDT, M. D.; Castro, D. D. de; Dutra, R. L. de S.; Tarouco, L. M. R. ELEKTRA: Um Chatterbot para Uso em Ambiente Educacional. *Renote, Porto Alegre*, v. 1, n. 2, 2003. Disponível em: <https://www.seer.ufrgs.br/index.php/renote/article/view/14336>. Acesso em: 10 set. 2022.
- LUGER, George F. *Inteligência Artificial*. 6. ed. Rio de Janeiro: Pearson, 2013.
- MITCHELL, R. *Web Scraping com Python: Coletando mais dados da web moderna*. 2. ed. São Paulo: Novatec Editora, 2019.
- OTA, Marcos Andrei; Trindade, Sara Dias; Araújo Júnior, Carlos Fernando; Souza, Alberto Messias Da Costa. *Aprendizagem Adaptativa Online: Uma Experiência Usando Trilhas e Chatbot para Desenvolver Competências Básicas em Língua*

Portuguesa e Matemática para o Ensino Superior. Revista de Ensino de Ciências e Matemática. v.10.4. p.56-69. 2019. Disponível em: <https://revistapos.cruzeirosul.edu.br/index.php/rencima/article/view/2455>.

Acesso em: 10 set. 2022.

SANTANA, Flávio. O futuro do design de interfaces com o Adobe XD. 06 jun. 2016. Design Culture. Disponível em: <https://designculture.com.br/o-futuro-do-design-de-interfaces-com-o-adobe-xd/>

Acesso em: 27 set. 2022.

SANTOS JUNIOR, João Benedito dos; Gomes, Julia; Dias, Julia Da Silva; Souza, Larissa Neris Oliveira de; Zanotti, Ana Carolina Nogueira; Dias, Raeli Pereira; Carvalho, Angela Borges de. Uma Proposta De Chatbot Para Apoio a Gestantes No Contexto Do Sistema De Saúde Brasileiro. RISTI : Revista Ibérica De Sistemas E Tecnologias De Informação. v.42. p.344-52. 2021. Disponível em: <https://www.proquest.com/openview/9f1bc67c09eaad9a4c4b58d713c92098/1?pq-origsite=gscholar&cbl=1006393>. Acesso em: 10 set. 2022.

SILVA, Katlen Karine Costa; TIerno, Rodrigo Ortega; BRANCHINE, Sandra Maria; VILAÇA, Douglas Santos Silva; OLIVEIRA, Fábio Henrique Monteiro. Desenvolvimento de ferramenta de chatbot como solução para a comunicação do IFB. In: ENCONTRO DE INOVAÇÃO EM SISTEMAS DE INFORMAÇÃO - SIMPÓSIO BRASILEIRO DE SISTEMAS DE INFORMAÇÃO (SBSI), 17. , 2021, On-line. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2021. p. 185-188. DOI: <https://doi.org/10.5753/sbsi.2021.15375>.

SILVA, Luiz F C.; RIVA, Aline D.; ROSA, Gabriel A.; et al. Banco de Dados Não Relacional. [Porto Alegre, RS]: Grupo A, 2021. E-book. ISBN 9786556901534. Disponível em: <https://app.minhabiblioteca.com.br/#/books/9786556901534/>. Acesso em: 28 set. 2022.

SILVA, Maurício Samy. Web Design Responsivo: aprenda a criar sites que se adaptam automaticamente a qualquer dispositivo, desde desktops até telefones celulares. São Paulo: Novatec Editora, 2018.

SOARES, Wallace. Ajax (Asynchronous JavaScript and XML): guia prático para windows. São Paulo: Érica, 2006.

Tôrres, Viviâne De Almeida; Maia, Luiz Cláudio Gomes; MUYLDER, Cristiana Fernandes de. Criação De Um Chatbot Para Atendimento De Suporte

Sistêmico De Uma Rede De Supermercados a Partir De Uma Base De Conhecimento Registrada No Mantis. A.to.Z. Curitiba. v. 8.2. p.60-68. 2020. Disponível em: <https://revistas.ufpr.br/atoz/article/view/65170>. Acesso em: 10 set. 2022.

TURING, Alan M. Computing machinery and intelligence (1950). The Essential Turing: the Ideas That Gave Birth to the Computer Age, p. 433-464, 2012. Disponível em: <http://www.cse.chalmers.se/~aikmitr/papers/Turing.pdf#page=442>

WEIZENBAUM, Joseph. ELIZA—a computer program for the study of natural language communication between man and machine. Communications of the ACM, v. 9, n. 1, p. 36-45, 1966.

