

DESENVOLVIMENTO DE CHATBOT PARA CONSULTA DE INFORMAÇÕES ACADÊMICAS DO ALUNO

DEVELOPMENT OF CHATBOT TO FETCH STUDENTS' ACADEMIC INFORMATION

DESARROLLO DE CHATBOT PARA CONSULTA DE INFORMACIÓN ACADEMICA DE ESTUDIANTES

Celso Gil Larussa Filho¹, João Henrique Gião Borges², Carlos Henrique Macias Porta³, Fabiana Florian⁴

e3122364

<https://doi.org/10.47820/recima21.v3i12.2364>

PUBLICADO: 12/2022

RESUMO

O *chatbot* é um programa de computador que interage com usuários por meio da linguagem natural, processando palavras, textos e comandos. No âmbito acadêmico, o ecossistema de consulta de informações de alunos entre centros de ensino é bastante similar, neles são disponibilizados portais em que o aluno pode realizar consultas de dados acadêmicos, tais como, boletim escolar, disciplinas matriculadas e telefones úteis. O objetivo deste artigo é desenvolver um *chatbot* baseado em regras para auxiliar os alunos na consulta dessas informações. O software foi desenvolvido utilizando Banco de Dados, API, Linguagem de programação *Python*, *Visual Studio Code* e o aplicativo de mensagens *Telegram*. Dentre as conclusões, se aponta a vantagem de disponibilizar um novo portal utilizando *chatbot* para consultar informações do aluno, sendo uma plataforma de fácil e rápido acesso, trazendo informações atualizadas a cada linha de texto digitada.

PALAVRAS-CHAVE: *Chatbot. Instituição de Ensino. Aluno. Telegram. Python.*

ABSTRACT

A chatbot is a computer program that interacts with users through language. Processing words, texts, and commands. In the academic sphere, the ecosystem for consulting student information between teaching centers is quite similar, they provide portals where students can consult academic data, such as school reports, registered subjects, and useful telephone numbers. The purpose of this article is to develop a rules-based chatbot to assist students in querying this information. The software was developed using Database, API, Python programming language, Visual Studio Code, and the Telegram messaging application. Among the conclusions, the advantage of providing a new portal using chatbot to consult student information is pointed out, being a platform of easy and quick access, bringing updated information to each line of text typed.

KEYWORDS: *Chatbot. Educational Institution. Student. Telegram. Python.*

RESUMEN

Un chatbot es un programa informático que interactúa con los usuarios a través del lenguaje natural, procesando palabras, textos y comandos. En el ámbito académico, el ecosistema de consulta de información de los alumnos entre centros docentes es bastante similar, estos brindan portales donde los alumnos pueden consultar datos académicos, como boletines escolares, asignaturas matriculadas y teléfonos de interés. El propósito de este artículo es desarrollar un chatbot basado en reglas para

¹ Graduando Bacharel em Sistemas de Informação pela Universidade de Araraquara - Uniara

² Licenciado em Computação pelo Centro Universitário Claretiano. Bacharel em Química pela Universidade Estadual Paulista Júlio de Mesquita Filho, mestre em Química pela Universidade Federal de São Carlos e doutor em Química Orgânica pela Universidade de São Paulo. Docente Curso de Sistemas de Informação da Universidade de Araraquara - UNIARA. Araraquara-SP.

³ Bacharel em Sistemas de Informação com ênfase em Data Science e Gestão de Negócios.

⁴ Graduada em Ciências Econômicas, Mestre em Desenvolvimento Territorial e Meio Ambiente pela Universidade de Araraquara (UNIARA). Doutora em Alimentos e Nutrição pela Faculdade de Ciências Farmacêuticas da Universidade Estadual Paulista "Júlio de Mesquita Filho" (FCFAR/UNESP) Bolsista Capes. Coordenadora de Trabalhos de Conclusão de Curso (TCC) das Engenharias Elétrica, Civil, Computação e Sistemas de Informação.



RECIMA21 - REVISTA CIENTÍFICA MULTIDISCIPLINAR ISSN 2675-6218

DESENVOLVIMENTO DE CHATBOT PARA CONSULTA DE INFORMAÇÕES ACADÊMICAS DO ALUNO
Celso Gil Larussa Filho, João Henrique Gião Borges, Carlos Henrique Macias Porta, Fabiana Florian

ayudar a los estudiantes a consultar esta información. El software fue desarrollado usando Base de Datos, API, lenguaje de programación Python, Visual Studio Code y la aplicación de mensajería Telegram. Entre las conclusiones, se destaca la ventaja de brindar un nuevo portal utilizando chatbot para consultar la información de los estudiantes, siendo una plataforma de fácil y rápido acceso, trayendo información actualizada a cada línea de texto tecleada.

PALABRAS CLAVE: Chatbot. Institucion de Enseñanza. Alumno. Telegram. Python.

INTRODUÇÃO

A consulta de informações acadêmicas dos alunos geralmente é realizada pelo portal da instituição de ensino, esse portal pode ser acessado por meio de *websites* ou aplicativos, neles são disponibilizados a tela de acesso ao sistema (*login*), que solicita as credenciais de acesso do aluno (usuário e senha) utilizados na autenticação. Após a autenticação do aluno via portal da instituição, as informações acadêmicas relacionadas ao seu âmbito escolar estão disponíveis para consulta, dentre elas, datas de avaliações, registro de notas, registro de faltas, horário de aulas, calendário escolar etc. É fundamental que o aluno tenha acesso às informações acadêmicas da sua instituição de ensino, pois ele irá consultá-las regularmente para auxiliá-lo em seu período escolar. Neste cenário, podemos encurtar e facilitar o processo de consulta das informações adotando a tecnologia de *chatbots*. O *chatbot* é um *software* que interage com usuários utilizando a linguagem natural (SHAWAR; ATWELL, 2007). Dessa forma, podemos reduzir o processo de autenticação, que é realizado de forma repetitiva no portal da instituição e permite que as solicitações sejam feitas diretamente em um aplicativo de mensagem enviando as solicitações via texto (linguagem natural).

O *chatbot* pode ser construído baseado em regras, onde são definidos comandos que o usuário pode solicitar, ou seja, ele compreende somente comandos que já foram definidos no seu desenvolvimento, caso o usuário digite alguma palavra ou comando fora do seu contexto o *chatbot* não processará a solicitação. Por outro lado, podemos aplicar conceitos da inteligência artificial (IA) no *chatbot* para o tornar “mais inteligente”, dessa forma, ele poderá aprender enquanto interage com o usuário, respondendo conforme o contexto da conversa, ou seja, toda vez que o chatbot interage sobre um determinado assunto ele armazena e analisa os dados fornecidos pelos usuários. (SHAWAR; ATWELL, 2007).

Segundo uma pesquisa apresentada no Mapa do Ecossistema Brasileiro de Bots em agosto de 2021, foram entrevistadas 96 empresas que desenvolvem *chatbots* e constatou-se que em 2020, 101 mil *chatbots* foram inseridos no mercado, já em 2021 esse número subiu para 216 mil, sendo desenvolvidos 115 mil novos *chatbots* em relação ao ano anterior, demonstrando que 21% deles atuam na área do Varejo, 17% no Setor Financeiro, 16% em Telecomunicações e somente 4% é voltado a Educação (PAIVA, 2021). Esses dados demonstram que o atendimento ao cliente é a principal função de um *chatbot*, podendo solucionar dúvidas interruptamente e atender várias pessoas ao mesmo tempo, 24 horas por dia, 7 dias na semana, se torna uma ótima solução para sanar dúvidas ou realizar serviços em relação a empresa, tais como, horário de funcionamento, canais de atendimento, rastreamento de produtos, data de vencimento de faturas, abertura de chamados, dentre outros. Ainda

RECIMA21 - REVISTA CIENTÍFICA MULTIDISCIPLINAR ISSN 2675-6218

DESENVOLVIMENTO DE CHATBOT PARA CONSULTA DE INFORMAÇÕES ACADÊMICAS DO ALUNO
Celso Gil Larussa Filho, João Henrique Gião Borges, Carlos Henrique Macias Porta, Fabiana Florian

de acordo com Paiva (2021), os canais mais utilizados para o desenvolvimento de *chatbots* são: WhatsApp, com 94% dos bots produzidos; Sites na Web, com 78%; Facebook Messenger, com 72%; Telegram, com 51%.

Uma pesquisa realizada sobre popularidade dos principais aplicativos de mensagens apresentada em fevereiro de 2022 mostra que entre 2.107 internautas que possuem *smartphone*, o *WhatsApp* está em 99% dos aparelhos dos entrevistados e 85% dos internautas o utilizam diariamente; *Instagram* em 82% dos aparelhos e 61% de uso diário; *Facebook Messenger* em 71% dos aparelhos e 19% de uso diário; *Telegram* em 60% dos aparelhos e 25% de uso diário (PAIVA, 2022).

Os dados percentuais acima demonstram a expansão dos *chatbots* para atendimento em diversas áreas e canais, principalmente utilizado em aplicativos de mensagem. Podemos afirmar que a maioria dos aparelhos móveis possuem pelo menos um aplicativo de mensagem, sendo utilizado diariamente pelo usuário.

No cenário acadêmico, a tecnologia de *chatbot*, disponibilizada via aplicativos de mensagens torna-se uma ótima alternativa para a instituição de ensino atender as demandas de consultas repetitivas feitas pelo aluno. Já no caso do aluno, ele terá todas as consultas realizadas armazenadas no *chat* do aplicativo, não havendo necessidade de realizar uma nova consulta quando for necessário verificar a mesma informação posteriormente, como é feita via portal web. Ao utilizar o *chatbot* a instituição de ensino disponibilizará ao aluno um novo portal para consulta de informações de forma rápida e prática.

O aplicativo de mensagens escolhido para desenvolvimento do *chatbot* é o *Telegram*, dentre as vantagens, destaca-se: Permite e auxilia a criação de robôs dentro da própria plataforma; disponibiliza uma API que permite utilizar mensagens como gatilho para serviços; é utilizado diariamente por vários usuários e é utilizado por empresas na criação de *chatbots*. Portanto, indaga-se: quais são as etapas para o desenvolvimento de um *chatbot* que auxilie o aluno a realizar consultas de informações via aplicativo de mensagens *Telegram*.

Parte-se da hipótese que será necessário desenvolver uma aplicação que se comunique com o aplicativo de mensagens *Telegram* para processar as interações do aluno. Além disso, deverá ser definido um sistema de gerenciamento de banco de dados (SGBD) para armazenar as informações acadêmicas e a criação de uma API REST para inserir e consultar os dados necessários. Dentro deste contexto, este artigo propõe desenvolver um *chatbot* baseado em regras em que o aluno irá interagir via aplicativo de mensagens *Telegram* para consultar informações vinculadas ao seu curso, tais como, registro de notas e disciplinas matriculadas.

O desenvolvimento do chatbot proposto foi realizado via ferramentas de desenvolvimento de software, sendo eles: *SQLite*; *FastAPI*; *Python*; *Visual Studio Code*, *SQLModel*, *pyTelegramBotAPI*; *Telegram*. As etapas para o desenvolvimento da pesquisa são: Desenvolver um banco de dados para armazenar informações do aluno, matrícula, professor, disciplina e resultado de provas, utilizando *SQLite*; Desenvolver uma API (Interface de Programação de Aplicações) para cadastrar, ler, alterar e deletar as informações do banco de dados utilizando *FastAPI*; Desenvolver uma aplicação de *chatbot*



RECIMA21 - REVISTA CIENTÍFICA MULTIDISCIPLINAR ISSN 2675-6218

DESENVOLVIMENTO DE CHATBOT PARA CONSULTA DE INFORMAÇÕES ACADÊMICAS DO ALUNO
Celso Gil Larussa Filho, João Henrique Gião Borges, Carlos Henrique Macias Porta, Fabiana Florian

utilizando a biblioteca pyTelegramBotAPI e o aplicativo de troca de mensagens *Telegram* para exibir as informações ao aluno.

1 REVISÃO BIBLIOGRÁFICA

1.1 CHATBOT

A definição diz que o chatbot é um software que processa a entrada de linguagem natural de um usuário e a resposta é gerada com base no contexto da conversa, retornando ao usuário, o processamento da mensagem pode ser definido com regras preestabelecidas ou com a aplicação da inteligência artificial. Geralmente a interação com o usuário é feita através de uma interface de texto (KHAN; DAS, 2018). Segundo Moraes e Souza (2015), o significado da palavra *chatbot* pode ser explicado dividindo a palavra em duas: “chat”, que vem da abreviação *chatter* (pessoa que conversa), e logo após, “bot”, que vem da abreviação *robot* (robô), ou seja, um robô em forma de software que conversa com as pessoas.

Em 1950, Alan Turing, indagava em seu artigo sobre máquinas da computação e inteligência a seguinte pergunta: “As máquinas podem pensar?”, para responder essa pergunta, definiu o Teste de Turing, que determinava critérios para diferenciar a conversa de humanos e máquinas. Nesse teste participavam dois seres humanos e uma máquina, sendo o objetivo do teste que um dos seres humanos descobrisse por meio de determinadas perguntas quem seria a máquina e quem seria o ser humano. A intenção era da máquina se passar por humano, fazendo com que o terceiro jogador não consiga distinguir se aquele determinado jogador que está respondendo as perguntas é uma máquina ou um humano. Nesse caso, poderíamos afirmar que a máquina pensa, pelo simples fato dela imitar comportamentos humanos (TURING, 2010).

O primeiro *chatbot* surgiu no MIT, desenvolvido por Joseph Weizenbaum, em 1966. ELIZA, como é chamado, é o primeiro *chatbot* a simular uma interação humana. A intenção era simples, imitar uma psicoterapeuta, realizando perguntas de acordo com os termos inseridos pelos usuários durante a conversa (WEIZENBAUM, 1983).

Em 1972 foi desenvolvido o *chatbot* PARRY, na universidade de Stanford, pelo cientista da computação e psiquiatra Kenneth Mark Colby. Enquanto ELIZA tentava simular um médico psiquiatra, PARRY utilizava a estratégia inversa, se passando por um paciente esquizofrênico que tentava provocar controvérsias com a intenção de obter respostas mais elaboradas dos usuários que interagiam com ela (ZEMČÍK, 2019).

Hoje, empresas como Apple, Microsoft, Amazon, IBM e Google desenvolvem ferramentas para que tornem os *chatbots* mais “inteligentes”, permitindo aperfeiçoar a conversa entre humanos e máquinas (SCHLICHT, 2016).

1.2 TELEGRAM

Lançado inicialmente para o sistema iOS da Apple em 14 de agosto de 2013 e chegando ao Android no final de outubro do mesmo ano, o aplicativo de mensagens surge com a proposta de enviar mensagens, fotos, vídeos e arquivos de qualquer formato, totalmente gratuito, de forma rápida e

segura, sendo permitido realizar chamadas de áudio e vídeo e podendo ser acessado de diversos canais ao mesmo tempo, tais como, celulares, tablets, notebooks (TELEGRAM, 2022).

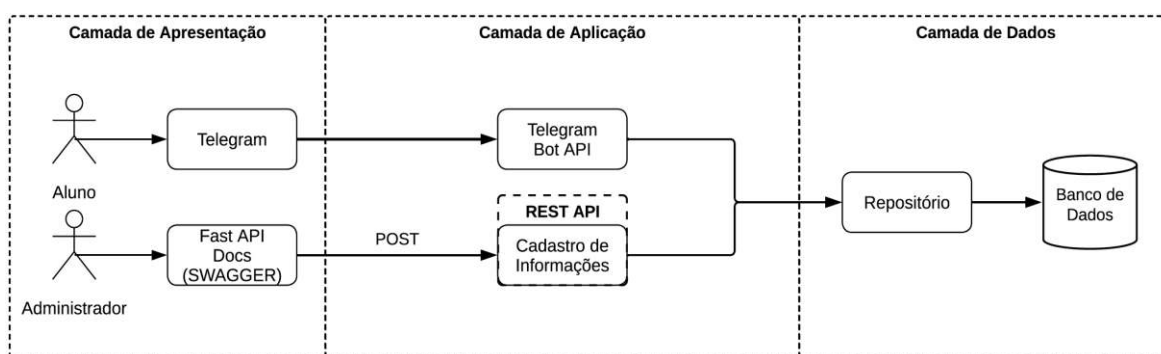
Por ter uma infraestrutura de múltiplos *data centers* e *criptografia* de ponta a ponta, o *Telegram* se torna uma plataforma com alta disponibilidade e segura, tanto para usuários quanto para os desenvolvedores. Os desenvolvedores podem utilizar a *API* aberta que o *Telegram* disponibiliza para desenvolver diversas ferramentas, tais como, automação de processos, criação de *chatbots*, ferramenta de vendas, de pagamentos etc. (TELEGRAM, 2022).

Os *bots* são aplicativos de terceiros executados dentro do *Telegram*, os usuários interagem enviando mensagens, comandos, imagens ou áudios, essas mensagens serão processadas e retornarão uma resposta conforme configurado pelo desenvolvedor. Para criar um *bot* será necessário falar com o *BotFather*, que é uma aplicação do próprio *Telegram*, essa ferramenta auxilia os usuários na criação de um novo *bot*, após a primeira interação, o usuário deve seguir alguns simples passos de configuração para gerar o *token* de autenticação, o *token* será uma identidade para o seu *bot* recém-criado fazer requisições a *API* do *Telegram* (TELEGRAM, 2022).

2 DESENVOLVIMENTO

Para o desenvolvimento deste projeto, foram definidas as tecnologias e ferramentas a serem utilizadas, sendo elas: *Telegram* e *FastAPI Swagger* na camada de Apresentação; *TelegramBotAPI* e *FastAPI* na camada de Aplicação; *SQLite* na camada de Dados; *Python* e *Visual Studio Code* para o desenvolvimento.

Figura 1 – Arquitetura do Projeto



Fonte: Própria.

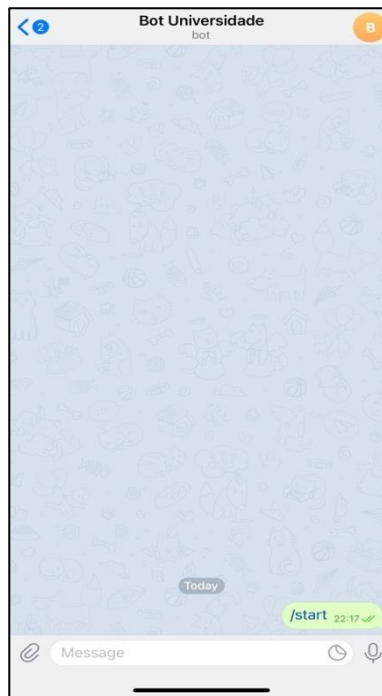
2.1 TELEGRAM

O *Telegram* é utilizado como plataforma de texto para a interação do aluno com o *chatbot*, ele será responsável pela troca de mensagens entre as entidades, pois nele o aluno irá inserir os comandos e a aplicação ficará responsável de retornar as informações solicitadas, apresentando-as no chat do *Telegram*.

RECIMA21 - REVISTA CIENTÍFICA MULTIDISCIPLINAR ISSN 2675-6218

DESENVOLVIMENTO DE CHATBOT PARA CONSULTA DE INFORMAÇÕES ACADÊMICAS DO ALUNO
Celso Gil Larussa Filho, João Henrique Gião Borges, Carlos Henrique Macias Porta, Fabiana Florian

Figura 2 – Tela de mensagens do *Telegram*

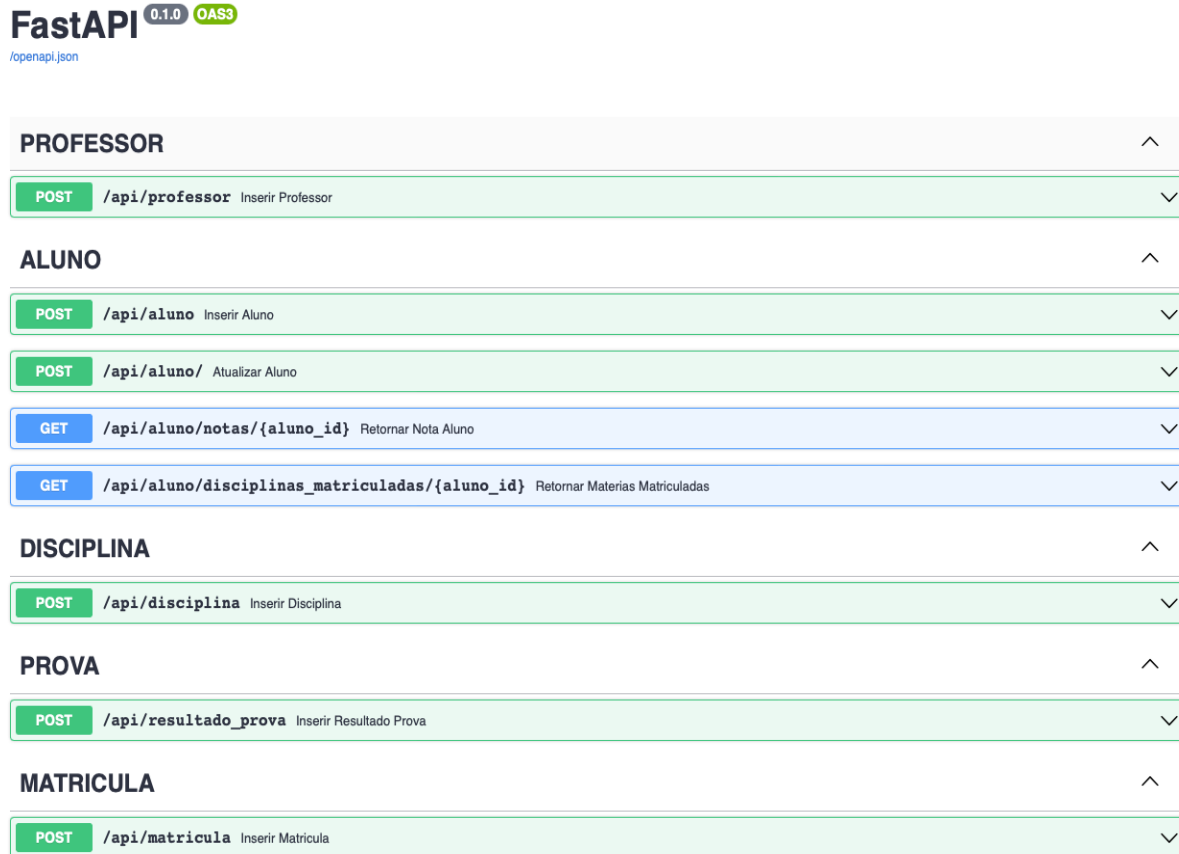


Fonte: Própria.

2.2 FASTAPI DOCS

O FastAPI Docs é um *endpoint* que utiliza o *framework Swagger*, este que por sua vez é encarregado de gerar a documentação de uma *API REST*, mapeando os modelos de dados e funcionalidades presentes na mesma. Ele disponibiliza uma interface web para realizar teste de métodos e verificar os tipos de dados que serão retornados.

Figura 3 – FastAPI Docs

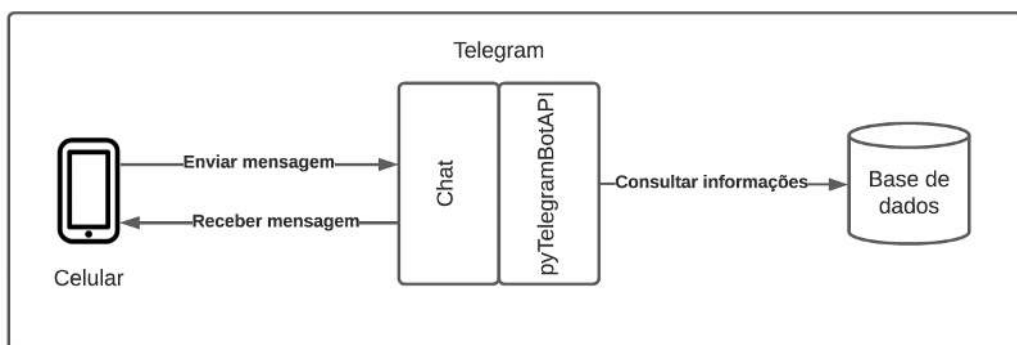


Fonte: Própria.

2.3 PYTELEGRAMBOTAPI

O *pyTelegramBotAPI* é uma implementação da API do *Telegram* que interage com o aplicativo de mensagens. Desenvolvida na linguagem de programação *Python*, é utilizado na criação de *bots* que interagem com o aplicativo de mensagens. O *pyTelegramBotAPI* utiliza manipuladores de mensagens (*handlers*) baseadas em filtros para executar funções.

Figura 4 – Funcionamento do *pyTelegramBotAPI*



Fonte: Própria.

Quando uma mensagem é enviada pelo usuário o *handler* verifica se o conteúdo da mensagem é compatível com a regra definida no filtro, se o conteúdo for compatível a função em que o *handler* está definido é executada, caso contrário o próximo *handler* será verificado e o processo de verificação de filtros se repete. Na figura 5 definimos dois *handlers* com os filtros /começar (linha 1), /contato (linha 7) e por último temos o *handler* da linha 13 que é responsável por processar mensagens que não se encaixam nos filtros anteriores.

Figura 5 – *Handlers*

```
1 @bot.message_handler(commands=['começar'])
2 def enviar_boas_vindas(mensagem):
3     retorno = "Olá, seja bem vindo!"
4     bot.reply_to(mensagem, retorno)
5
6
7 @bot.message_handler(commands=['contato'])
8 def enviar_telefone_de_contato(mensagem):
9     retorno = "O nosso telefone de contato é: (00) 00000-0000"
10    bot.reply_to(mensagem, retorno)
11
12
13 @bot.message_handler(func=lambda mensagem: True)
14 def comando_invalido(mensagem):
15     retorno = "Desculpe, não entendi o seu comando. Digite um comando válido!"
16     bot.reply_to(mensagem, retorno)
```

Fonte: Própria.

2.4 FRAMEWORK FASTAPI

O desenvolvimento da *API REST* foi realizado utilizando o *Framework FastAPI*, construído na linguagem *Python*, de código aberto e lançado em novembro de 2018 por Sebastián Ramírez, é uma ferramenta para desenvolvimento de *APIs*. Dentre as vantagens deste *framework* destaca-se a alta performance, sendo considerado um dos frameworks mais rápidos disponíveis atualmente, de fácil usabilidade, robusto e baseado em padrões da *OpenAPI Specification (OAS)* (FASTAPI, 2022).

A *OpenAPI Specification* define uma interface padrão independente de linguagem para *APIs REST* que permite humanos e computadores entendam os recursos disponíveis do serviço sem acesso ao código-fonte, documentação ou através da inspeção de tráfego de rede. Quando devidamente definido, um consumidor pode entender e interagir com o serviço remoto com uma quantidade mínima de lógica de implementação (SWAGGER, 2022).

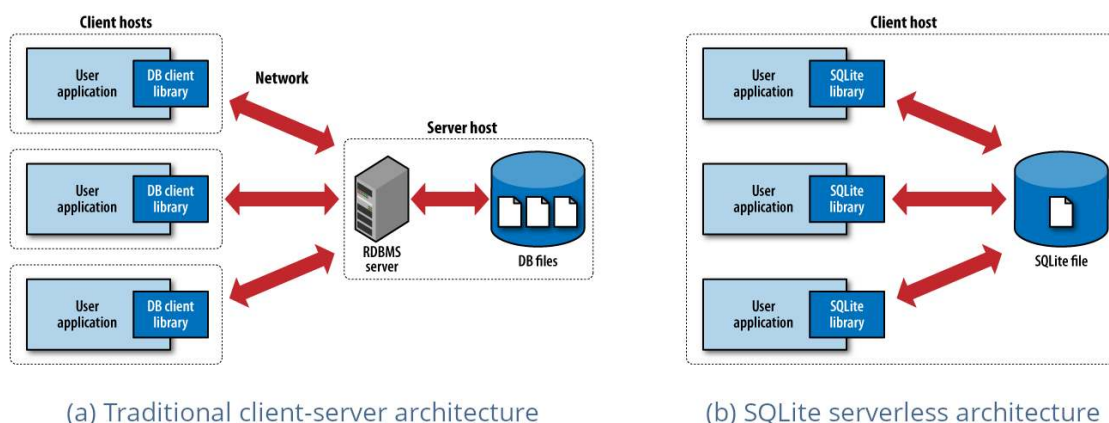
A *API REST* é responsável pela comunicação entre sistemas, nesse caso em específico permitirá que comandos enviados via *FastAPI Docs* possam ser encaminhados ao repositório que se

comunicará com o banco de dados. A Figura 3 demonstra as entidades e *endpoints* definidos para inserir e consultar dados.

2.5 SGBD SQLITE

O sistema de gerenciamento do banco de dados escolhido foi o *SQLite*, pois ele é um mecanismo de banco de dados *SQL* autônomo, ou seja, não necessita de um servidor (DBMS) para realizar suas operações, se tornando ideal para aplicações de pequeno e médio porte. O *SQLite* utiliza apenas um único arquivo de disco de formato multiplataforma para armazenar as informações, podendo ser copiado para sistemas de 32 e 64 bits sem apresentar problemas de compatibilidade (SQLite, 2022). A imagem abaixo demonstra a diferença entre as comunicações de gerenciadores de banco de dados que utilizam a tradicional arquitetura cliente-servidor (Figura 6-a) para a arquitetura sem servidor utilizada no *SQLite* (Figura 6-b).

Figura 6 – Como o SQLite é diferente dos bancos de dados tradicionais?



Fonte: Kreibich (2010, seção 1.1)

A comunicação do *SQLite* foi realizada adotando a técnica de Mapeamento Objeto Relacional (ORM). O ORM permite aproximar o paradigma de desenvolvimento de aplicações orientadas a objetos ao paradigma do banco de dados relacional (Figura 7).

Figura 7 – O que é um ORM?



Fonte: Rubabe (2020, ORM for Python)

O banco de dados relacional trabalha com tabelas e relações entre elas para representar modelos da vida real. Dentro das tabelas podemos ter uma ou mais colunas e a unidade utilizada para representação no modelo relacional é de “linhas” (Figura 8).

Figura 8 – Tabela Pessoa

ID	NOME	IDADE	SEXO
1	ANA	23	F
2	MARIA	34	F
3	JOÃO	14	M

Fonte: Própria.

O paradigma orientado a objetos trabalha de modo diferente, nele nós temos diversos elementos como classes, propriedades, visibilidade, herança e interfaces. A unidade utilizada na orientação a objetos é o “objeto” pois é a representação de algo do mundo real, seja abstrato ou concreto (Figura 9).

RECIMA21 - REVISTA CIENTÍFICA MULTIDISCIPLINAR ISSN 2675-6218

DESENVOLVIMENTO DE CHATBOT PARA CONSULTA DE INFORMAÇÕES ACADÊMICAS DO ALUNO
Celso Gil Larussa Filho, João Henrique Gião Borges, Carlos Henrique Macias Porta, Fabiana Florian

Figura 9 – Objetos Pessoa

```
import Pessoa

[
    Pessoa(id=1, nome="ANA", idade=23, sexo="F"),
    Pessoa(id=2, nome="MARIA", idade=34, sexo="F"),
    Pessoa(id=3, nome="JOÃO", idade=14, sexo="M"),
]
```

Fonte: Própria.

A biblioteca utilizada para realizar a comunicação entre os paradigmas foi o *SQLModel*. O *SQLModel* é uma ferramenta para interagir com banco de dados SQL utilizando a técnica de ORM a partir de código *Python*. Ele foi projetado para ser intuitivo, fácil de usar, altamente compatível e robusto *SQLModel* (2022). A Figura 10 demonstra a utilização de código orientado a objetos e funções do *SQLModel* para realizar operações de inserção, busca, atualização e exclusão no banco de dados.

Figura 10 – Instruções utilizando *SQLModel*

```
from sqlmodel import select
from sqlmodel import Session, create_engine
from model import Pessoa

pessoa = Pessoa(id=1, nome='Lucas', idade=21, sexo='M')

#Insert
with Session(engine) as session:
    session.add(pessoa)
    session.commit()

#Select
with Session(engine) as session:
    statement = select(Pessoa).where(Pessoa.id == 1)
    pessoas = session.exec(statement).one()

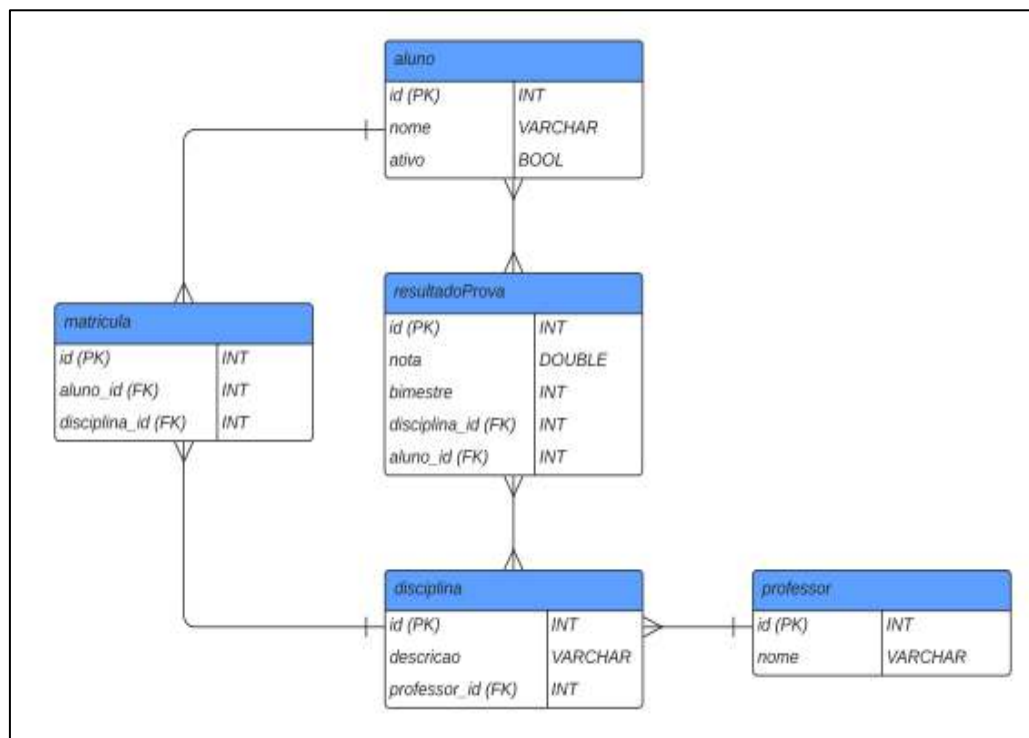
#Update
with Session(engine) as session:
    statement = select(Pessoa).where(Pessoa.id == 1)
    pessoa = session.exec(statement).one()
    pessoa.idade = 23
    session.add(pessoa)
    session.commit()

#Delete
with Session(engine) as session:
    statement = select(Pessoa).where(Pessoa.id == 1)
    pessoa = session.exec(statement).one()
    session.delete(pessoa)
    session.commit()
```

Fonte: Própria.

Para o projeto foram definidas cinco entidades baseadas na modelagem de dados, sendo elas: Aluno (para representar as informações do aluno), Disciplina (para representar as informações da disciplina e relacionar o professor responsável por ela), Matrícula (para relacionar a disciplina em que o aluno está matriculado), Professor (para representar as informações do professor) e ResultadoProva (para representar as informações sobre nota de avaliação, bimestre e relacionar o aluno). A Figura 11 demonstra a representação gráfica do Diagrama Entidade e Relacionamento (DER) utilizado no banco de dados.

Figura 11 – Diagrama Entidade e Relacionamento (DER)



Fonte: Própria.

2.6 LINGUAGEM DE PROGRAMAÇÃO E IDE

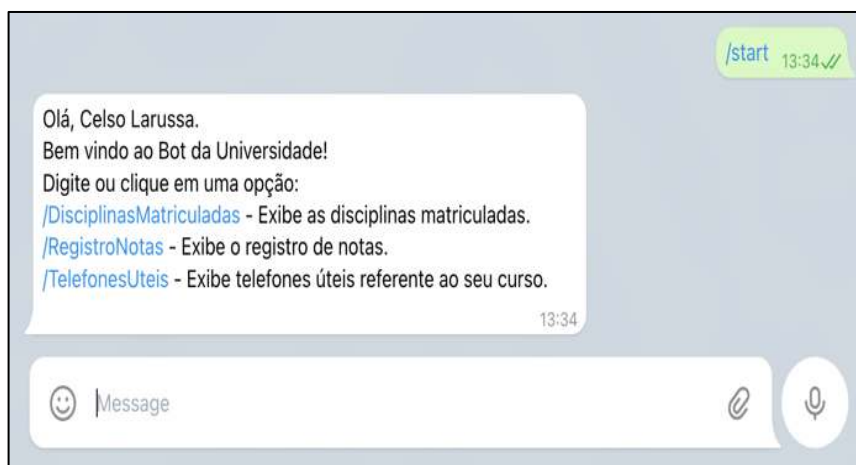
O projeto foi desenvolvido na linguagem de programação chamada *Python*, de alto nível, modular, multiplataforma e orientado a objetos, foi criada no início de 1990 por Guido Van Rossum. Essa linguagem foi escolhida para desenvolver o projeto por ter uma *sintaxe* simples, de fácil compreensão e por contar com uma vasta biblioteca destinada ao desenvolvimento de aplicações.

Este projeto foi desenvolvido utilizando o Ambiente de Desenvolvimento Integrado (IDE) *Visual Studio Code*, editor de código de código aberto criado pela Microsoft. Possui enorme coleção de extensões e atalhos que podem auxiliar o desenvolvimento de projetos.

3 RESULTADOS

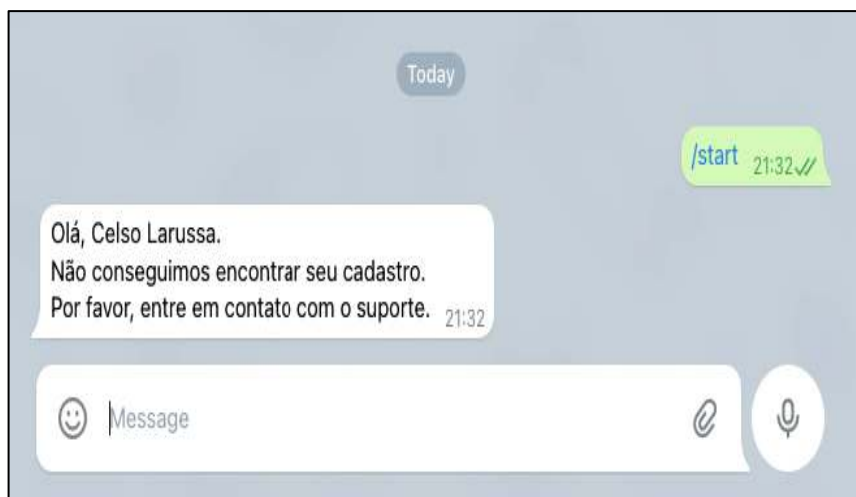
A tela inicial (Figura 12, Figura 13) do *chatbot* no aplicativo de mensagens *Telegram* apresenta a primeira interação do usuário, o gatilho para iniciar a conversa é dado pelo comando `/start`. A validação do usuário é feita pelo identificador único (ID) que cada usuário utiliza dentro da plataforma do *Telegram*, percebe-se que o bot já realizou a validação do usuário (Figura 12), condizendo que este usuário tem um cadastro no banco de dados, já na figura 13 a mensagem enviada pelo chatbot demonstra que o usuário não está devidamente cadastrado. Outro ponto a destacar é que o *chatbot* identifica e utiliza o nome que o usuário escolheu ao configurar o *Telegram* para cumprimentá-lo, sendo neste caso, Celso Larussa.

Figura 12 – Usuário cadastrado



Fonte: Própria.

Figura 13 – Usuário não cadastrado



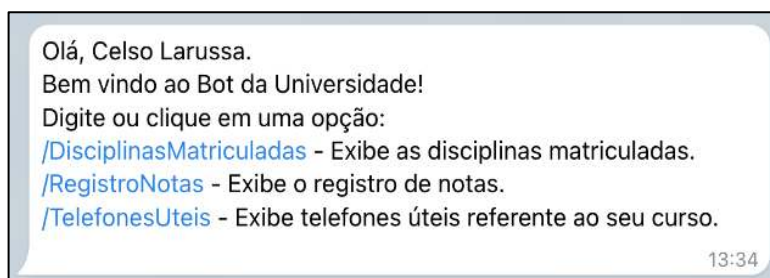
Fonte: Própria.

Dentro do escopo da mensagem de boas-vindas são listados três comandos para consulta (Figura 14), sendo eles, disciplinas matriculadas (/DisciplinasMatriculadas), registro de notas (/RegistroNotas) e telefones úteis (/TelefonesUteis), o comando escolhido poderá ser digitado ou selecionado com o mouse dentro da plataforma *Telegram*.

RECIMA21 - REVISTA CIENTÍFICA MULTIDISCIPLINAR ISSN 2675-6218

DESENVOLVIMENTO DE CHATBOT PARA CONSULTA DE INFORMAÇÕES ACADÊMICAS DO ALUNO
Celso Gil Larussa Filho, João Henrique Gião Borges, Carlos Henrique Macias Porta, Fabiana Florian

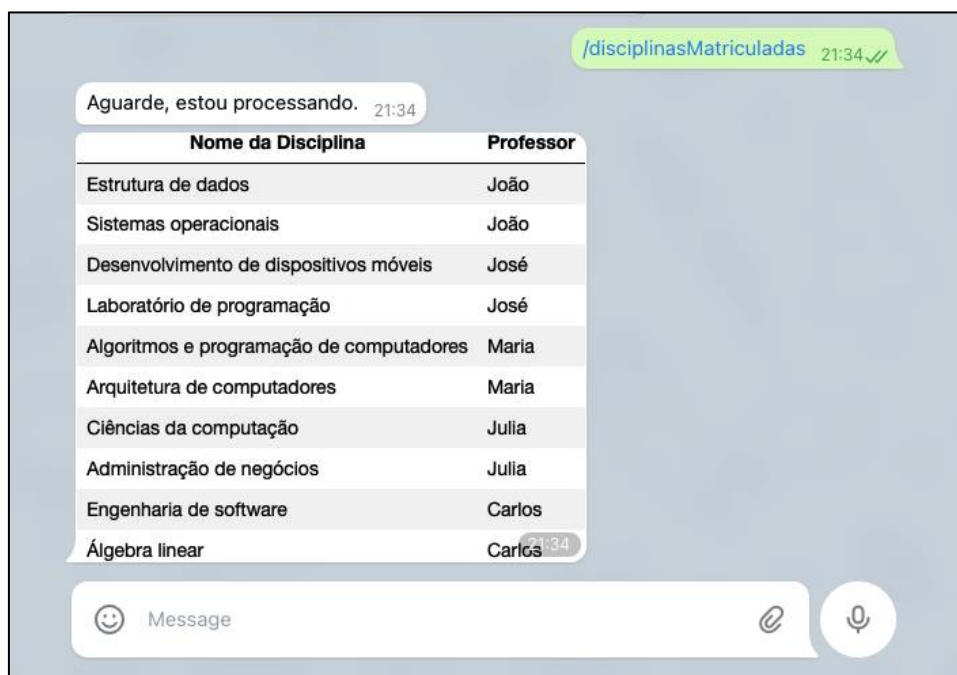
Figura 14 – Menu *chatbot*



Fonte: Própria.

Dentre as opções listadas no menu (Figura 14), o usuário pode selecionar o campo que corresponde ao seu interesse de consulta. Caso o comando escolhido seja a consulta de disciplinas matriculadas ([/DisciplinasMatriculadas](#)), será retornado ao usuário uma imagem com uma tabela contendo as disciplinas em que o Aluno estiver matriculado e o nome dos professores que a lecionam (Figura 15).

Figura 15 – Tabela de disciplinas matriculadas



Fonte: Própria.

Se o usuário escolher o comando de registro de notas ([/RegistroNotas](#)), será retornado uma imagem com uma tabela contendo o nome da disciplina e as notas em seus respectivos bimestres (figura 16).

RECIMA21 - REVISTA CIENTÍFICA MULTIDISCIPLINAR ISSN 2675-6218

DESENVOLVIMENTO DE CHATBOT PARA CONSULTA DE INFORMAÇÕES ACADÊMICAS DO ALUNO
Celso Gil Larussa Filho, João Henrique Gião Borges, Carlos Henrique Macias Porta, Fabiana Florian

Figura 16 – Tabela de registro de notas



/RegistroNotas 22:01 ✓✓

Aguarde, estou processando. 22:01

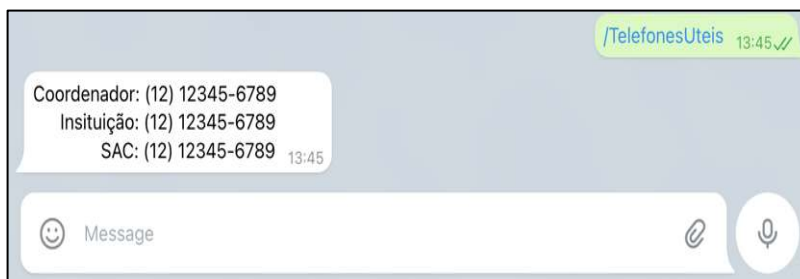
Nome da Disciplina	1° Bim.	2° Bim.	3° Bim.	4° Bim.
Estrutura de dados	7.30	8.80	8.50	5.30
Sistemas operacionais	7.60	9.40	9.70	9.30
Desenvolvimento de dispositivos móveis	5.60	2.30	5.30	10.00
Laboratório de programação	7.30	6.80	7.90	7.00
Algoritmos e programação de computadores	8.50	7.70	8.90	10.00
Arquitetura de computadores	8.00	6.20	7.40	8.00
Ciências da computação	3.20	6.50	4.50	-
Administração de negócios	6.50	3.20	3.50	-
Engenharia de software	7.90	6.90	10.00	-
Álgebra linear	4.30	6.70	6.90	22:01

Message

Fonte: Própria.

Outro comando disponível é o de telefones úteis (/TelefonesUteis), quando selecionado retorna números de telefone vinculados ao curso e sua instituição de ensino do aluno (Figura 17).

Figura 17 – Telefones Úteis



/TelefonesUteis 13:45 ✓✓

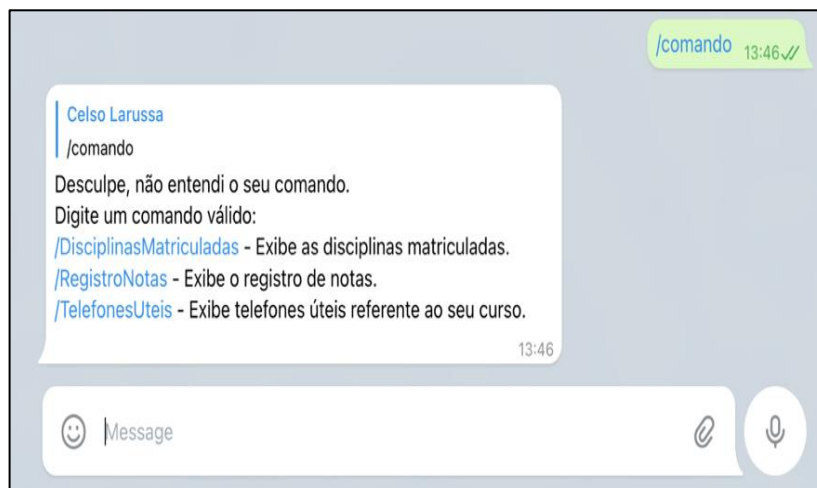
Coordenador: (12) 12345-6789
 Insituição: (12) 12345-6789
 SAC: (12) 12345-6789 13:45

Message

Fonte: Própria.

Se o usuário escolher um comando diferente dos listados significa que este será um comando inválido, a tratativa para esse caso será retornar uma mensagem informando que o comando solicitado é invalido e listando novamente os comandos disponíveis. A Figura 18 demonstra uma interação utilizando um comando inválido.

Figura 18 – Demonstração de comando inválido



Fonte: Própria.

4 CONCLUSÃO

Nos últimos anos, os *chatbots* vêm impactando positivamente diversos setores ao redor do mundo. O setor da educação é atingido pela crescente popularidade deste tipo de ferramenta de comunicação utilizada para agilizar suas interações, fornecendo suporte acadêmico. Além disso, o *chatbot* pode ajudar os centros de ensino a fornecerem um tempo de resposta mais rápido aos seus alunos, garantindo que as suas perguntas sejam respondidas em minutos ou segundos. Outras vantagens são: fornecimento de suporte 24 horas por dia, 7 dias por semana, melhorando a experiência do aluno com a instituição e eliminando processos repetitivos, podendo realocar funcionários em outras demandas ou setores, reduzindo os custos.

O desenvolvimento do *chatbot* para consulta de informações acadêmicas via aplicativos de mensagens demonstra ser uma excelente oportunidade para disponibilização de um novo canal de atendimento para a instituição de ensino, tendo em vista que a maior parte da população Brasileira utiliza aplicativo de mensagens, e que a tecnologia para desenvolvimento de *chatbots* estão evoluindo rapidamente para entregar *chatbots* mais inteligentes para seus clientes, visando interações melhores e mais completas.

O *chatbot* proposto surgiu com a necessidade de criar um canal de atendimento para que os alunos conseguissem realizar consultas de forma fácil e rápida. Dessa forma, quando implantamos um *chatbot* voltado ao atendimento acadêmico disponibilizamos uma nova forma de consulta para os alunos, permitindo que façam consultas rápidas com apenas uma linha de texto. Desse modo, o aluno consegue informações atualizadas a cada busca realizada.

Concluindo, este trabalho apresentou as etapas para o desenvolvimento de um *chatbot* para consulta de informações acadêmicas, tais como, disciplinas matriculadas e registro de notas, utilizando as seguintes tecnologias de desenvolvimento de software de código aberto: Linguagem de programação Python; Editor de código Visual Studio Code; Banco de dados Relacional *SQLite*;

Biblioteca *SQLModel*; Framework FastAPI; Biblioteca pyTelegramBotAPI; aplicativo de mensagens Telegram.

REFERÊNCIAS

FASTAPI. **FastAPI docs**. [S. l.]: Fastapi, 2022. Disponível em: <https://fastapi.tiangolo.com/pt/>. Acesso em: 26 maio 2022.

KHAN, Rashid; DAS, Anik. **Build Better Chatbots: A Complete Guide to Getting Started with Chatbots**. Bangalore: Apress, 2018. p. 9.

KREIBICH, Jay A. **Using SQLite**. [S. l.]: O'Reilly Media, Inc. 2010. Disponível em: <https://www.oreilly.com/library/view/using-sqlite/9781449394592/>. Acesso em: 16 nov. 2022.

PAIVA, Fernando. Mapa do Ecossistema Brasileiro de Bots 2021. **Mobile Time**, p. 3-5, ago. 2021. Disponível em: <https://www.mobiletime.com.br/pesquisas/mapa-do-ecossistema-brasileiro-de-bots-2021/>. Acesso em: 01 maio 2022.

PAIVA, Fernando. Mensageria no Brasil Fevereiro de 2022. **Mobile time**, p. 4-5, fev. 2022. Disponível em: <https://www.mobiletime.com.br/pesquisas/mensageria-no-brasil-fevereiro-de-2022/>. Acesso em: 10 de abril de 2022.

RUBABE, Sabanova. ORM for Python. **Medium**, out. 2020. Disponível em: <https://medium.com/pragmatech/orm-for-python-b63cfbc39e7f>. Acesso em: 16 nov. 2022.

SCHLICHT, Matt. The Complete Beginner's Guide to Chatbots: Everything you need to know. **Revista Chatbots**, 2016. Disponível em: <https://chatbotsmagazine.com/the-complete-beginner-s-guide-to-chatbots-8280b7b906ca>. Acesso em: 12 abr. 2022.

SHAWAR, B. A.; ATWELL, E. Chatbots: are they really use-ful? **Journal for Language Technology and Computational Linguistics**, v. 22, p. 29–49, 2007. Disponível em: [http://jcl.org/content/5-allissues/19-Heft1-2007/Bayan Abu-Shawar and Eric Atwell.pdf](http://jcl.org/content/5-allissues/19-Heft1-2007/Bayan%20Abu-Shawar%20and%20Eric%20Atwell.pdf).

SOUZA, L. S.; MORAES, Silvia Maria Wanderley. Construção automática de uma base AIML para chatbot: um estudo baseado na extração de informações a partir de FAQs. **Anais...** do XII ENIAC, p. 137-141, 2015.

SQLite. **About SQLite**. [S. l.: s. n.], 2022. Disponível em <https://www.sqlite.org/about.html>. Acesso em 11 out. 2022.

SQLModel, **SQLModel**. [S. l.: s. n.], 2022. Disponível em <https://sqlmodel.tiangolo.com>. Acesso em 14 out. 2022.

SWAGGER, **OpenAPI Specification**. [S. l.: s. n.], 2022. Disponível em: <https://swagger.io/specification/>. Acesso em: 10 out. 2022.

TELEGRAM. **Telegram FAQ**. [S. l.]: Telegram, 2022. Disponível em: <https://telegram.org/faq>. Acesso em: 15 maio 2022.

TURING, A. M. Maquinário computacional e inteligência. In: BONJOUR, L.; BAKER, A. (Org.) **Filosofia: textos fundamentais comentados**. São Paulo: Artmed, 2010. p. 227-231.

WEIZENBAUM, Joseph; ELIZA, Mdash. a Computer Program for the Study of Natural Language Communication Between Man and Machine. **Commun ACM**, New York, v. 9, n. 1, p. 36–45, jan. 1966. Disponível em: <http://doi.acm.org/10.1145/365153.365168>.



RECIMA21 - REVISTA CIENTÍFICA MULTIDISCIPLINAR
ISSN 2675-6218

DESENVOLVIMENTO DE CHATBOT PARA CONSULTA DE INFORMAÇÕES ACADÊMICAS DO ALUNO
Celso Gil Larussa Filho, João Henrique Gião Borges, Carlos Henrique Macias Porta, Fabiana Florian

ZEMČÍK, Tomáš. A Brief History of Chatbots. **DEStech Transactions on Computer Science and Engineering**, 2019. Disponível em:
https://www.researchgate.net/publication/336734161_A_Brief_History_of_Chatbots.