

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

PAULO SÉRGIO SOUZA LIMA

CHATBOT INSTRUTOR DE INVESTIMENTOS CONSERVADORES

Uberlândia/MG
2021

PAULO SÉRGIO SOUZA LIMA

CHATBOT INSTRUTOR DE INVESTIMENTOS CONSERVADORES

Trabalho de conclusão de curso apresentado como parte dos requisitos para obtenção do grau de Bacharel em Engenharia Elétrica da Universidade Federal de Uberlândia.

Orientador:
Prof. Dr. Fábio Vincenzi Romualdo da Silva

Uberlândia/MG
2021

AGRADECIMENTOS

Gostaria de agradecer primeiro a Deus, por me dar o dom da vida e possibilitar que eu desfrute de minhas conquistas.

Agradeço aos meus pais, Iris e Sérgio, por me apoiarem e por me proporcionar a chance de conseguir me formar, o que facilitou que eu alcançasse esse objetivo, e também a minha irmã, Bianca, por me auxiliar e servir de exemplo para mim.

Agradeço a minha namorada, Carol, por sempre estar ao meu lado, me amparando, me apoiando e me animando a continuar nessa caminhada.

Agradeço às minhas avós, Maria Inês e Maria Efigênia, por toda preocupação nesse período e também por contribuírem para que eu pudesse alcançar esse sonho.

Agradeço a todos meus amigos por proporcionarem ótimos momentos e por me incentivarem a continuar na graduação.

Agradeço aos meus colegas de graduação e aos meus professores, por compartilharem seus conhecimentos comigo e me guiarem a conquista da formação.

RESUMO

O ato de automatizar processos tem a intenção de fazer com que os mesmos executem sem o auxílio de um humano, mas também para que estes processos se tornem mais assertivos e rápidos. O trabalho propõe automatizar o processo de fornecimento de informações sobre investimentos conservadores, utilizando para isso do conceito de chatbots. Será feito um levantamento de todas as tecnologias necessárias para que seja possível construir um chatbot e disponibilizá-lo para os usuários na internet. Com isso será criado o chatbot, implementando sua lógica de identificação de mensagens e busca de respostas, bem como interfaces de controle para fornecer informações para o mesmo e controle de usuários, possibilitando assim que possam ser criados diversos tipos de chatbots que forneçam informações sobre o assunto desejado de acordo com o usuário que prover as informações. Ao fim do trabalho será apresentada uma plataforma pronta para receber informações sobre o assunto desejado, bem como um chatbot que fornecerá informações sobre investimentos conservadores.

Palavras-chave: Investimentos conservadores, dinheiro, investir, chatbot, Python.

ABSTRACT

The act of automating processes is intended to make them run without the help of a human, but also to make these processes more assertive and faster. The work offers to automate the process of providing information about conservative investments, using for this the concept of chatbots. A survey will be made of all the necessary specifications so that it is possible to build a chatbot and make it available to users on the internet. With this, the chatbot will be created, implementing its message identification logic and search for answers, as well as control interfaces to provide information for it and user control, thus enabling the creation of different types of chatbots that provide information about the desired subject according to the user who taster like information. At the end of the work, a platform will be presented ready to receive information about the desired subject, as well as a chatbot that will provide information about conservative investments.

Key-words: conservative investments, money, investing, chatbot, Python.

LISTA DE FIGURAS

Figura 1 - Porcentagem da população que investe em cada país	14
Figura 2 - Divisão dos tipos de investidores na B3 em 2021	15
Figura 3 - Onde o brasileiro investe	16
Figura 4 - Diagrama de acesso dos tipos de usuários	22
Figura 5 - Arquivos iniciais na criação do projeto	24
Figura 6 - Modelo de dados aplicativo usuarios.....	25
Figura 7 - Modelo de dados aplicativo perguntas	26
Figura 8 - Modelo de dados aplicativo captura	26
Figura 9 - Rotas url aplicativo usuarios.....	27
Figura 10 - Rotas url aplicativo perguntas	28
Figura 11 - Rota url aplicativo home	28
Figura 12 - Rota url aplicativo captura	29
Figura 13 - Rotas das requisições até o template final	29
Figura 14 - Templates html do aplicativo usuario	30
Figura 15 - Templates html acessados pelo aplicativo perguntas	31
Figura 16 - Template html acessado pelo aplicativo home	31
Figura 17 - Template html acessado pelo aplicativo captura	32
Figura 18 - Acessos dos usuários.....	33
Figura 19 - Funções que o usuário colaborador tem acesso	34
Figura 20 - Fluxo da requisição e resposta ao usuário do chatbot	35
Figura 21 - Mensagem na linha de comando após iniciar o chatbot.....	36
Figura 22 - Tela home da aplicação	37
Figura 23 - Tela de login para usuário administrador ou colaborador.....	38
Figura 24 - Tela inicial logado com usuario administrador.....	38
Figura 25 - Tela para inserção de novos usuários colaboradores	39
Figura 26 - Tela inicial logado com usuário colaborador	39
Figura 27 - Tela de inserção para a resposta a certa mensagem enviada	40
Figura 28 - Tela de edição de mensagem e resposta já cadastradas	40
Figura 29 - Tela com informações capturadas de usuários do chatbot	41
Figura 30 - Tela com dados de um usuário do chatbot.....	41
Figura 31 - Tela da API de integração	42
Figura 32 - Tela do chatbot de finanças.....	42

Figura 33 - Exemplo 1 de resposta do chatbot	43
Figura 34 - Exemplo 2 de resposta do chatbot	44
Figura 35 - Exemplo 3 de resposta do chatbot	45

Sumário

1 INTRODUÇÃO.....	10
1.1 Justificativa.....	12
1.2 Objetivos	12
2 REFERENCIAL TEÓRICO	13
2.1 Educação Financeira.....	13
2.1.1 Renda Ativa x Renda Passiva	16
2.1.2 Investimentos de Renda Fixa x Investimentos de Renda Variável.....	16
2.1.3 Tipos de Investimento	17
2.1.4 Corretoras de Valores	18
2.2 Chatbot.....	19
2.3 Python	19
2.4 Django.....	20
2.5 HTML.....	21
3 DESENVOLVIMENTO	22
3.1 Primeiras Configurações	22
3.2 Criação de Aplicativos e Tabelas	24
3.3 Modelo de dados.....	25
3.3.1 Modelo de dados aplicativo usuarios.....	25
3.3.2 Modelo de dados aplicativo perguntas	25
3.3.3 Modelo de dados aplicativo captura.....	26
3.4 Rotas URL.....	27
3.4.1 URLs aplicativo usuarios	27
3.4.2 URLs aplicativo perguntas.....	28
3.4.3 URLs aplicativo home.....	28
3.4.4 URLs aplicativo captura.....	28

3.5	Templates.....	29
3.5.1	Templates do aplicativo usuarios	30
3.5.2	Templates do aplicativo perguntas.....	30
3.5.3	Templates do aplicativo home.....	31
3.5.4	Templates do aplicativo captura	31
3.6	Views.....	32
3.6.1	View do aplicativo usuarios	32
3.6.2	View aplicativo perguntas.....	33
3.7	Teste chatbot e cadastro de perguntas e respostas.....	35
4	RESULTADOS E DISCUSSÕES.....	37
4.1	Home.....	37
4.2	Login.....	38
4.3	Login/administrador	38
4.4	Login/usuário colaborador	39
4.5	Cadastro de novas perguntas / editar / excluir	39
4.6	Capturas.....	41
4.7	API.....	41
4.8	Chatbot.....	42
5	CONCLUSÃO	47
6	Referências.....	48

1 INTRODUÇÃO

O Brasil não proporciona educação financeira para alunos do ensino básico e médio. No ensino universitário, na maior parte dos cursos esse assunto também não é abordado. A falta de conhecimento acaba prejudicando a maioria esmagadora dos brasileiros, pois não sabem administrar o próprio dinheiro, não possuem noção da importância do controle orçamentário, dos impactos e a necessidade de se evitar o endividamento, dos efeitos positivos de construir uma reserva financeira, a médio e longo prazo e não sabem como incorporar o hábito de investir ao seu dia a dia de forma orgânica.

Universidades, principalmente em países desenvolvidos, já implementaram cursos de engenharia em Indústria 4.0 (SACKEY, 2016). A implementação desses cursos é baseada em equipes multidisciplinares em várias tecnologias e técnicas, onde os alunos de graduação devem aprofundar e fortalecer suas habilidades (KHAN, 2018)(MABKHOT, 2018).

A Alemanha oferece, na INTERNATIONALE HOCHSCHULE, um curso de BACHAREL EM ENGENHARIA INDUSTRIAL 4.0 no estilo “Estude onde quiser. 100 % online ou diretamente no local ou em mais de 28 unidades universitárias IU em toda a Alemanha”(HOCHSCHULE,2018). Além dessas facilidades, a universidade alemã oferece mais 12 centros de estudos e exames na Áustria e Suíça.

Algumas das disciplinas do curso são: Gestão da Cadeia de Abastecimento; Economia e Mercado; Contabilidade de Custos e de Desempenho; Financiamento e Investimento.

Profissionais com formações e conhecimentos diversos se tornam interessantes para as empresas visto que uma única pessoa domina várias áreas e consegue integra-las, com isso profissionais que tenham conhecimento de administração, trabalho colaborativo, gestão da cadeia de abastecimento, **financiamento e investimento**, saber analisar os dados coletados de modo a otimizar a produção, saber gerenciar um projeto, simular um negócio, dentre outros assuntos, são profissionais mais visados no mercado de tecnologia.(UNOPAR,2019)

Outras universidades da Alemanha, bem como outros países da Europa, Estados Unidos e Ásia também se adaptaram para atender as novas tendências da Indústria 4.0.

O Brasil não proporciona nenhuma formação financeira básica para a população e ainda não implantou cursos de graduação que atendam as novas tendências mundiais da Indústria 4.0.(LEWGOY,2021)

Além, disso novas ferramentas de aprendizado estão sendo cada vez mais utilizadas, pois cada geração prefere aprender de uma determinada maneira. Atualmente, os aplicativos de mensagens são mais utilizados do que as redes sociais. O fato de que as pessoas passam mais tempo em aplicativos de mensagens do que em redes sociais já foi comprovado. Em 2015, o número de usuários mensais dos quatro principais aplicativos de mensagens do mundo WhatsApp, Messenger, WeChat e Viber - ultrapassou as quatro principais plataformas de mídia social - Facebook, Twitter, Instagram e Google+ (INSIDER'S,2016).

Muitos trabalhos vêm sendo publicados com a intenção de utilizar chatbots na educação. A Universidade Estácio lançou um chatbot para ingressantes do Ensino Médio. É o Rota de Carreiras (Facebook Messenger), um chatbot que pretende mostrar aos futuros alunos o dia a dia que terão em suas carreiras. A Professora Elektra foi um dos primeiros chatbots na educação brasileira. Ele foi desenvolvido em 2002, por pesquisadores da Universidade Federal do Rio Grande do Sul (UFRGS). (LA CRUZ, 2019)

Em Bangladesh foi desenvolvido um chatbot denominado Doly que se comunica na língua Bengali. Seu algoritmo de IA identifica a dúvida e toma decisão da melhor resposta (que deve ser instantânea e na língua Bengali).

Outro trabalho usou um chatbot denominado Edubot para auxiliar os alunos do primeiro ano em uma matéria de programação de um curso de engenharia. Quando os usuários fazem uma pergunta ao EduBot, o algoritmo de IA busca uma resposta em uma base de dados. Se não encontrar uma resposta, o EduBot responde ao aluno que não encontrou uma resposta para a pergunta e que o professor já foi avisado. O chatbot envia um email para o professor com a pergunta feita pelo aluno. O professor pode respondê-la para o aluno por meio de email e, caso a pergunta seja relevante,

acrescentá-la juntamente com sua resposta no banco de dados do chatbot, melhorando assim a capacidade de resposta dele.

Nesse sentido, esse trabalho propõem a implementação de um chatbot instrutor de investimentos conservadores.

1.1 Justificativa

Atualmente o engenheiro de controle e automação necessita de conhecimento de diversas áreas para que possa atuar na Indústria 4.0. Ter conhecimento somente na área técnica não proporciona uma visão geral de como a indústria opera. Limitando-se dessa maneira a possibilidade de o profissional reduzir custos e elevar a qualidade do produto, visto que não compreende o *modus operandi* de outras áreas relacionadas a gestão da cadeia de abastecimento; economia e mercado; contabilidade de custos e de desempenho; financiamento e investimento. Este trabalho não tem a pretensão de abordar todos assuntos relacionados a investimentos. A intenção é fornecer informações básicas de investimento de renda fixa. Todavia, o chatbot proposto poderá ser programado para instruir sobre outros assuntos em trabalhos futuros.

1.2 Objetivos

Esse trabalho propõe a implementação de um chatbot Instrutor de Investimentos Conservadores. O chatbot poderá ser utilizado por qualquer pessoa que tiver interesse de aprender sobre investimentos, proverá conhecimentos básicos para que se comece a investir, possibilitando assim que o usuário escolha qual caminho de investimento mais lhe agrada. Ele ficará disponível no site www.sirius.guru e poderá ser acessado por qualquer pessoa que tiver interesse em aprender sobre investimentos conservadores.

2 REFERENCIAL TEÓRICO

2.1 Educação Financeira

Educação financeira é o nome dado a um tipo de ensino que tem como objetivo instruir as pessoas a como gerir o próprio dinheiro, visando bem-estar de vida futuro que seja duradouro, e a realização de sonhos. A educação financeira vai além de ensinamentos propriamente financeiros, mas também atua em âmbito emocional, pois é necessário obter-se um controle pessoal sobre com que gastar o dinheiro que se possui, aprendendo a diferenciar gastos necessários e desnecessários, a identificar boas oportunidades financeiras em bens ou produtos, e a manter o controle em momentos difíceis do mercado.

Um dos pontos mais importantes na educação financeira é conscientizar a pessoa para que desenvolva o hábito de gastar menos do que ganha. Outro ponto importante é investir de modo que se consiga rentabilidade acima da inflação, pois, caso contrário o capital investido irá diminuir com o passar do tempo. Nesse sentido, aplicar o dinheiro na poupança é melhor do que deixar o dinheiro parado em casa ou na conta bancária, porém isso torna o indivíduo um poupador e não um investidor, visto que a rentabilidade da poupança sempre fica abaixo da inflação.

Possuir a tão sonhada independência ou estabilidade financeira, adquirir um imóvel, um automóvel, abrir uma empresa, viajar para o lugar desejado, ou até mesmo aposentar mais cedo sem abrir mão de uma renda mensal, coisas que são possíveis de se obter desde que se planeje bem, ponderando as escolhas do presente em pró de resultados futuros.

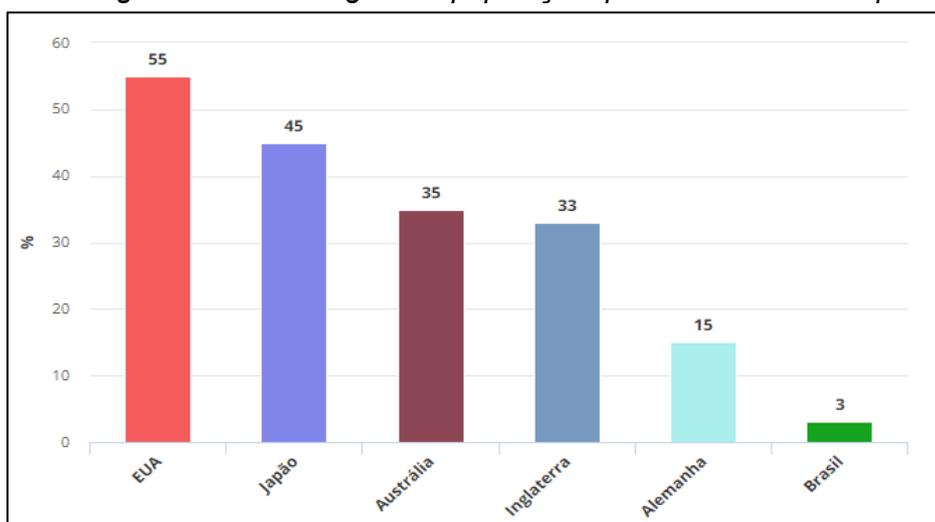
Gastar dinheiro é inevitável, a sobrevivência do ser humano depende de gastos com moradia, alimentação, higiene pessoal, gastos que são considerados primários e que são ditos gastos fixos, afinal todos os meses tem se a certeza que a conta de água será cobrada, que serão feitas compras no mercado, portanto esses são alguns dos gastos para apenas sobreviver. O primeiro objetivo de uma pessoa é possuir capital para custear as contas mensais, e não deixar que outros gastos atrapalhem isso. A partir do momento que a pessoa consegue pagar seus custos mensais e ainda sobra alguma quantia de dinheiro, é preciso saber aplicá-la.

Uma das barreiras para se tornar um investidor é o consumismo. Comprar uma roupa, um calçado, algum bem que não é necessário no momento. Isso não significa que a pessoa não possa gastar com coisas de qualidade, comprar presentes ou com coisas que proporcionem bem estar. Significa que deve desenvolver o hábito do bom senso na hora de gastar sua renda ativa. Esse tipo de renda é aquele em que a pessoa deve trabalhar para obtê-la.

O ato de investir, é um dos principais assuntos de educação financeira, pois são os investimentos que podem trazer estabilidade para a vida do profissional e de seu negócio, e mesmo para a vida pessoal. Saber como aplicar o dinheiro, onde ele pode ter um bom rendimento. Existem diferentes tipos de investimento, com rentabilidade diferente e vários graus de riscos. A questão dos riscos é o que mais gera medo nas pessoas. Geralmente a palavra investimento já vem atrelada ao ato de arriscar e correr o risco de perder o dinheiro. Todavia, esse pensamento é fruto de desinformação e falsas impressões espalhadas pela sociedade.

Segundo reportagem do site valor investe, do grupo globo, apenas 3% dos brasileiros investiram em ações no ano de 2020 (GOEKING, 2021). Nos EUA 55% dos americanos investem em ações, o que coloca os Estados Unidos no topo da lista dos países que possuem maior porcentagem de investidores. Em segundo lugar encontra-se o Japão, com 45%, conforme gráfico apresentado na Figura 1.

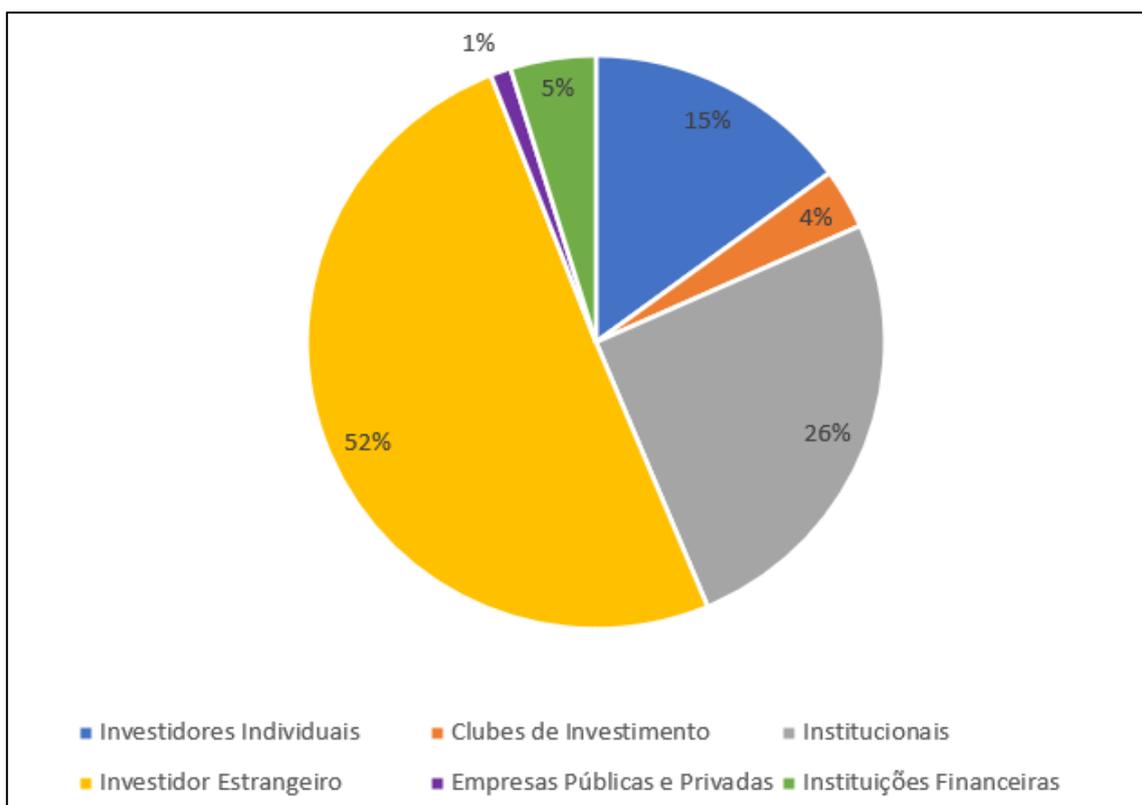
Figura 1 - Porcentagem da população que investe em cada país



Fonte: Gallup, JPX, Statistics Bureau of Japan, Australian Securities Exchange, Finder, Deutsches Aktieninstitut, Anbima. Organização: XP (2020)

O Brasil é o melhor lugar para se investir, e o pior lugar para se fazer uma dívida, devido aos altos juros que o país possui, porém o brasileiro geralmente pouco investe e muito se endivida. Mas se o Brasil é o melhor lugar para se investir, e os brasileiros não investem, quem está investindo? Os estrangeiros! Isso ocorre por causa da grande rentabilidade dos títulos do país.

Figura 2 - Divisão dos tipos de investidores na B3 em 2021



Fonte: Autoria própria

A B3 é a bolsa de investimentos do Brasil. O gráfico mostrado na Figura 2 foi retirado do próprio site da B3, ele evidencia que 52% dos investimentos feitos no país são feitos por pessoas estrangeiras, valor referente à apenas pessoas estrangeiras, sem levar em consideração empresas ou instituições financeiras do exterior (B3, 2021).

A imagem apresentada na Figura 3 mostra onde os brasileiros concentraram seus investimentos no ano de 2019. A maioria optou pela poupança (ANBIMA, 2019). O índice de investimentos em poupança elevado está relacionado ao pouco conhecimento do brasileiro sobre investimentos de renda fixa.

Figura 3 - Onde o brasileiro investe



Fonte: ANBIMA(2019)

2.1.1 Renda Ativa x Renda Passiva

Existem dois tipos de renda. Um que se trabalha ativamente para consegui-la, e o outro onde é adquirida sem atividade. Quando se fala em emprego, profissão, esses são os meios de se conseguir renda ativa, pois para adquiri-la o indivíduo precisa se manter realizando algum tipo de atividade. Já quando se fala em renda conquistada a partir de aluguéis de imóveis, rendimentos de investimentos, dividendos de ações, direitos autorais de livros, dentre outras atividades em que a pessoa recebe sem a necessidade de realizar uma atividade, é denominada de renda passiva.

Visando bem-estar e qualidade de vida, conquistar fontes de renda passiva é imprescindível, porque em algum momento da vida o cidadão se aposenta. Nesse sentido, uma das maneiras de se adquirir renda passiva é através dos investimentos.

2.1.2 Investimentos de Renda Fixa x Investimentos de Renda Variável

O investimento de renda fixa, pode ser prefixado, onde no ato do investimento já se sabe o valor que ele retornará na data de retirada, ou pós-fixados, onde o rendimento está atrelado a alguma taxa básica de juros do país, e se descobre o valor rendido apenas no ato da retirada. Investimento de renda fixa, mesmo que pós-

fixados, são considerados seguros, devido à ausência de inúmeros fatores que possam influenciar o rendimento.

Investimentos de renda variável não é possível prever o retorno, podendo até mesmo não retornar ou causar algum tipo de prejuízo, pois estes estão atrelados a muitos fatores externos, economia, política, ativos de empresas, portanto há dificuldade de prever o rendimento da aplicação. Apesar de mais arriscado, chama atenção de investidores devido a possibilidade de retornos maiores que a renda fixa.

2.1.3 Tipos de Investimento

A seguir será apresentada características de alguns tipos de investimento.

Tesouro Direto: investimento do tipo de renda fixa, criado através de um programa do governo que visava captar dinheiro para financiar e custear suas despesas. Neste modo, o indivíduo empresta dinheiro para o país, e em troca, o valor aplicado renderá juros enquanto estiver em posse da instituição. Aplicações no tesouro direto possuem data de vencimento, valor mínimo para se investir, porém possibilita retirada a qualquer momento, rendendo diariamente.

Certificado de Depósito Bancário: investimento do tipo renda fixa, com características semelhantes ao tesouro direto, porém com a mudança de que o dinheiro será emprestado para bancos. Considerada modalidade de baixo risco, possui prazo de vencimento, porém em alguns casos não é possível retirar o valor adquirido a qualquer momento.

Ações: investimento de renda variável, nesta modalidade há a compra de frações de uma empresa, onde o patrimônio adquirido varia junto com a performance da empresa, podendo aumentar caso a empresa tenha bons resultados, ou diminuir caso a empresa não tenha sucesso. Há um grande risco nessa modalidade, para investir é preciso estudar e avaliar cuidadosamente as opções e o desempenho da empresa que irá se investir.

Existem diversos tipos de investimentos, porém os citados a cima são alguns dos mais comuns e que podem ser rentáveis. Para realizar os investimentos existem alguns passos a se seguir, geralmente o primeiro deles é escolher a corretora de

valores para isso, onde a mesma irá apresentar os recursos para tornar o investimento possível.

2.1.4 Corretoras de Valores

Corretoras de valores ou também conhecidas como corretoras de investimentos, são as responsáveis por intermediar a compra e venda de ativos financeiros, pois apresentam uma liberação para atuar nessa posição. As corretoras em geral podem atuar como auxiliadoras dos investidores, pois podem oferecer serviços de assessoria de investimentos, recursos que facilitem identificar os rendimentos, e sempre antes de tudo obtêm o perfil de investidor do cliente em questão, pois será isso que irá nortear o próprio investidor a que tipos de investimentos escolher de acordo com suas características pessoais, e possibilitará a corretora a fornecer produtos que se encaixem em seu perfil. Para quem quer começar a investir, é preciso escolher um bom investimento e uma boa corretora para intermediar os ativos aplicados.

Analisando os conceitos passados, é possível concluir que investir não é algo fácil, e que leva tempo, assim como a educação financeira demandará possuir certas informações, trabalhar com o psicológico, além de outros diversos fatores, porém levando em consideração o que isso pode proporcionar, são conceitos que precisam ser passados para frente, que essas informações cheguem as pessoas de forma clara e sucinta, para que possam atingir um nível de vida satisfatório.

A grande aliada da educação e do mundo de investimentos é a tecnologia, pois quando se fala em educação, a tecnologia possibilita que as informações cheguem de forma fácil e rápida, educando assim as pessoas que as leem, e quando se trata de investimento, a tecnologia ajuda a projetar ganhos através da programação de recursos para isso, identificar bons produtos, e também fornecer informações referentes de forma simplificada. Aliar a educação financeira, investimentos e tecnologia, aumenta o potencial de que as 3 juntas ajudem a sociedade a caminhar para rumos melhores, e possibilita a construção de ferramentas iterativas como os chatbots, que é uma tecnologia que interage com seus usuários como se fosse um

humano, para que as informações corretas cheguem as pessoas, de uma forma que as agrade.

2.2 Chatbot

Chatbot é a denominação dada a um software capaz de conversar com uma pessoa de maneira tão natural quanto o diálogo de dois humanos, respondendo na maior parte das vezes de forma assertiva.

Os aplicativos de mensagens são os meios de comunicação mais usados no mundo. É cada vez mais difícil jovens irem em busca de informações detalhadas, com leituras longas, essa é uma das principais características das gerações Z e Alfa. Embarcar chatbots em estruturas de aplicativo de mensagem se torna uma boa alternativa pelo fato de seus usuários muitas vezes nem suspeitarem que estão conversando com um software.

Os chatbots começaram a ser usados para fim comerciais para esclarecer dúvidas dos clientes bem como na venda de determinado produto.

Na educação eles podem ser equipados com banco de dados contendo vários assuntos e com algoritmos de inteligência artificial para detectar falhas na formação do aluno.

Nesse trabalho o chatbot será implementado usando a linguagem de programação Python, em conjunto com recursos de banco de dados e recursos *web*.

2.3 Python

A linguagem python está entre as 10 mais usadas no mundo. É uma linguagem de programação simples e de fácil compreensão. Consegue atender desde demandas simples até as complexas, podendo ser integrada com várias API's, bibliotecas e linguagens do mercado, o que o torna sempre uma boa opção para solucionar um problema ou demanda de tecnologia. Python possibilita a criação de scripts de automação, que é um dos maiores objetivos da tecnologia, e isso trabalhando com bibliotecas nativas (GABRIEL OLIVEIRA,2021).

É utilizada também para desenvolvimento web, podendo atuar em sites simples e até páginas que demandam ações de marketing, que trabalhem com ciência de dados, possuindo uma variedade de *frameworks* que podem ser integrados a ele, como os mais populares Django e Flask.

Quando há a necessidade de se resolver um problema de tecnologia, que demande automação, seja de baixa ou alta complexidade, que demande integração com a *web*, a linguagem python se mostrará sempre como uma das melhores opções.

2.4 Django

Django é *framework web* para a linguagem Python. A construção de aplicações na web pode ser complexa, devido a problemas de segurança, requisição e resposta de páginas, rotas e caminhos, tudo isso precisa ser bem configurado para que a aplicação funcione, o Django facilita essas questões. O framework Django é usado em aplicações como Instagram, YouTube, Spotify, site da NASA, etc (BYLEARN,2020).

O Django funciona como um orquestrador de processos, protegendo a aplicação, facilitando a configuração de rotas, construção de páginas, fazendo todo o trabalho pesado por trás da configuração de uma aplicação *web*, deixando o programador livre para construção da aplicação em si, possuindo ainda o banco de dados sqlite embarcado. O Django foi criado inicialmente para sustentar um site jornalístico, que necessitava de velocidade e segurança para a publicação de reportagens, porém em 2005 já se tornou um código aberto, é dito como um *framework* para perfeccionistas que possuem prazo (TIAGO,2020).

O Django é organizado em pequenas aplicações internas que implementam pedaços específicos, utilizando do método MVT (*Models, Views, Templates*). *Models*, ou modelos, são os dados usados na solução. Referencia-se o pedaço da declaração dos campos que seriam criados no banco de dados, e à medida que a aplicação executa, são alimentados. Nas *views* são extraídas as informações das requisições, é feito o processamento necessário, e então é fornecida a resposta. *Template* é o que será apresentado ao usuário, são arquivos de texto estilizados com linguagem html, css e javascript, com partes estáticas e variáveis que apresentam o que é fornecido pelas *views*.

2.5 HTML

Html é a sigla para *HiperText Markup Language*, no português, Linguagem de Marcação de HiperTexto. Html é a linguagem utilizada para desenvolver as páginas que estão na internet, ela que possibilita definir, marcar, posicionar toda a estrutura dentro de um site, indo de simples títulos a até imagens, dimensões entre outros.

Os hipertextos, citado no nome da linguagem, representa a conexão de vários elementos dentro de um site, é o que faz a comunicação de dados e recursos dentro do site se interligarem, e permitir uma comunicação organizada e muitas vezes intuitiva no site.

A linguagem é repleta de *tags* que servem para referenciar o que é cada coisa dentro da página, títulos, textos, imagens, formatação, tudo no html possui uma tag para referencia-lo. Html é a base para o site, o esqueleto, o que irá sustentar e organizar as informações dentro dele, portanto se há a necessidade de fazer uma aplicação na internet é necessário aprende html.

Em conjunto com html pode se trabalhar com ferramentas como o css, que é a sigla para *Cascading Style Sheets*, que traduzindo é folha de estilo composta por camadas. Com css é possível construir interfaces única a um site, pois ele atua na definição de como os elementos serão exibidos e os estilizando. Para criação de um site estilizado e visualmente organizado, combinar os recursos de html e css são indispensáveis, e na maioria dos sites presentes na internet, esses recursos estão presentes.

Como html é dito como esqueleto da página, ele pode ainda conter em seu corpo outras linguagens como o javascript e o próprio python, portanto é totalmente possível construir uma aplicação na web que utilize recurso e ferramentas dessas outras linguagens, basta fazer a integração.

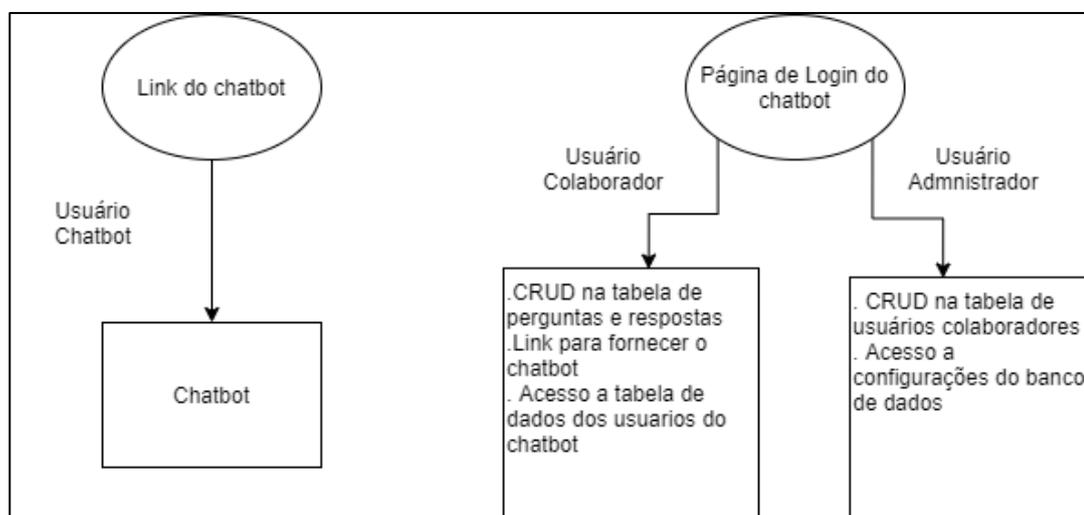
A partir dos recursos citados, é possível organiza-los e integrá-los para que se chegue ao objetivo traçado que é a criação do chatbot de educação de financeira, portanto, integrando-os e fazendo a programação correta, a aplicação irá funcionar como desejado.

3 DESENVOLVIMENTO

Entendido o escopo e quais recursos terá a aplicação, o que será desenvolvido é um chatbot, utilizando a linguagem python pura para programá-lo, em conjunto com o *framework web* Django, integrando com script html e recursos de css para que seja possível hospedá-lo em um site, onde o próprio será capaz de fornecer informações para o usuário sobre investimentos conservadores, para que no futuro ele por conta própria decida qual caminho seguir.

É preciso entender que o sistema contará com 3 tipos de usuários. Usuário administrador que será o responsável por controlar os usuários colaboradores (criadores de chatbot), vale lembrar que o usuário administrador é o mesmo usado para criar recursos durante a programação. Usuário colaborador é o usuário que irá configurar as mensagens e respostas de um chatbot específico seu, podendo possuir vários usuários colaboradores com vários chatbots que fornecem respostas diferentes. Usuário do chatbot, que é o usuário que irá mandar mensagens e receber as respostas programadas do chatbot em uso.

Figura 4 - Diagrama de acesso dos tipos de usuários



Fonte: Autoria própria

O esquema do sistema seguirá o diagrama da Figura 4, com cada usuário tendo acesso aos respectivos recursos, portanto a criação precisa prover tais recursos para serem executados.

3.1 Primeiras Configurações

Para iniciar foi preciso instalar o python na máquina que foi desenvolvida a aplicação, pois como a máquina utiliza o sistema operacional Windows, o python não

vem instalado nativamente, para *download* da linguagem basta acessar o site oficial do python.

Após a instalação do python, é necessário instalar o *framework* Django. Para isso basta abrir o terminal de comando do sistema operacional utilizado, e digitar o comando “pip install Django”. Vale lembrar que o Django traz nativo consigo o banco de dado sqlite, que será utilizado pelo chatbot para armazenar as perguntas e respostas.

O próprio Django possui recursos que auxiliam na criação do chatbot. Para começar a criar um projeto foi realizado o comando “django-admin startproject chatbot” onde chatbot foi o nome escolhido para o projeto. Com isso já é criado o ambiente inicial do projeto, com uma pasta com nome chatbot, que será a “pasta mãe” do projeto, com um *script* Python chamado manage.py, que no futuro será usado para fazer com que a aplicação rode, e uma pasta administrativa também de nome chatbot com alguns arquivos de configuração para que seja possível hospedar o site em algum domínio. É necessário também executar o comando “python manage.py createsuperuser” que criara o usuário administrador da aplicação. Com esse login que será feito o primeiro acesso, e com ele que será possível manipular os diversos usuários que contribuem com a criação das perguntas e respostas que o chatbot provê.

Para realizar a criação dos *scripts* extra necessários, e também para ajudar na programação, foi utilizado o programa VS code, mas essa escolha foi opcional, realizar a criação dos *scripts* em qualquer outro interpretador ou mesmo no bloco de notas não é problema, os interpretadores apenas servem para ajudar quanto a indentação do código, e a visualizar melhor os recursos utilizados marcando-os com cores diferentes e etc.

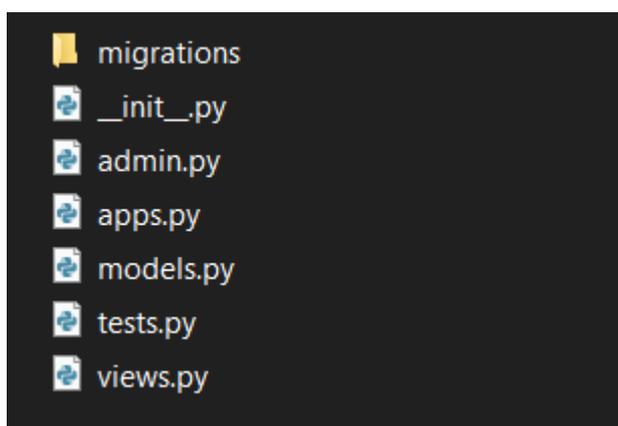
Um pacote a ser instalado é o unidecode, que é uma biblioteca do python que ajuda na padronização de variáveis de texto, retirando acentuações e pontuações. Esse pacote em conjunto com outros recursos que serão programados na linguagem python, ajudará na padronização das mensagens recebidas a fim de prover uma inteligência artificial e identificação de linguagem natural assertiva. A partir disto começou-se a criação das pequenas aplicações que, separadamente, resolviam uma parte do processo.

3.2 Criação de Aplicativos e Tabelas

Os recursos presentes no projeto, são considerados como aplicativos do Django, portanto para isso, necessitaremos de um aplicativo que represente a home da nossa página, que é onde o usuário que irá cadastrar as perguntas faça seu *login*, um aplicativo para representar os usuários e seus *login* no sistema, para o controle de quem vai manipular as perguntas e respostas, um aplicativo referente a captura de informações de quem está utilizando o chatbot, e um aplicativo para referenciar as perguntas feitas por quem utiliza o chatbot e qual será a resposta dada, para trata-las.

Para instalar esses diferentes aplicativos, basta digitar o comando no terminal de comando da máquina, lembrando de estar dentro da pasta do projeto, “python manage.py startapp (nome do app)” para todos os aplicativos citados acima, isso irá criar pastas internas no projeto com os nomes dos aplicativos digitados, contendo inicialmente os arquivos mostrados na Figura 5.

Figura 5 - Arquivos iniciais na criação do projeto



Fonte: Autoria própria

A primeira configuração a se fazer é acessar a pasta chatbot de configuração, e escrever no arquivo urls.py, os caminhos das pastas de cada aplicativo. Quando as requisições são trocadas, o chatbot precisa saber qual o caminho seguir. Posteriormente será criado arquivo de urls.py para todas as pastas das aplicações, para que os aplicativos possam trocar requisições entre eles. A troca de requisições é facilitada pelo Django, ele fornece a configuração interna de rotas de acordo com o que for especificado nos *scripts* urls.py.

3.3 Modelo de dados

O modelo de dados de cada um dos aplicativos são armazenados dentro do arquivo `models.py`, que está localizado dentro da respectiva pasta.

Os aplicativos que possuem modelos de dados são: o da captura, referente a captura de dados de quem está utilizando o chatbot, o aplicativo das perguntas, que é usado pelas perguntas e respostas do chatbot, e o de usuários que é responsável por controlar os usuários autorizados a criar perguntas e respostas.

São criadas no banco tabelas com os nomes das aplicações, com os campos sendo os dados declarados no `models.py` da respectiva aplicação.

3.3.1 Modelo de dados aplicativo usuarios

O modelo de dados do aplicativo usuários está apresentado na Figura 6, há informações pessoais como nome, email, usuário e senha, e dois campos administrativos, `code` para referenciar cada usuário e `active` para especificar se é um usuário ativo ou não.

Figura 6 - Modelo de dados aplicativo usuarios

```
from django.db import models

# Create your models here.
class Usuario(models.Model):
    code = models.CharField(max_length=15)
    active = models.BigIntegerField()
    name = models.CharField(max_length=100)
    email = models.CharField(max_length=100)
    user = models.CharField(max_length=50)
    password = models.CharField(max_length=10)

    def __str__(self):
        return self.name
```

Fonte: Autoria própria

3.3.2 Modelo de dados aplicativo perguntas

O modelo de dados do aplicativo perguntas é mostrado na Figura 7, contendo os campos `question` e `answer`, que correspondem ao campo de armazenamento de

perguntas e respostas. O campo `code` é usado para referenciar pergunta e resposta. O campo `code_user` referencia quem é o usuário detentor daquela pergunta e resposta, e o campo `code_relation` serve para realizar uma pergunta feita no momento, com outra feita anteriormente.

Figura 7 - Modelo de dados aplicativo perguntas

```
from django.db import models

# Create your models here.
class Pergunta(models.Model):
    code = models.CharField(max_length=15)
    code_user = models.CharField(max_length=15)
    active = models.BigIntegerField()
    code_relation = models.CharField(max_length=15)
    question = models.CharField(max_length=500)
    answer = models.CharField(max_length=500)

    def __str__(self):
        return self.question
```

Fonte: Autoria própria

3.3.3 Modelo de dados aplicativo captura

O modelo de dados do aplicativo captura está mostrado na Figura 8, nele estão os dados que serão armazenados caso seja identificada uma mensagem contendo dados pessoais fornecidos por algum usuário do chatbot.

Figura 8 - Modelo de dados aplicativo captura

```
from django.db import models

# Create your models here.
class Captura(models.Model):
    code = models.CharField(max_length=15)
    code_user = models.CharField(max_length=15)
    active = models.BigIntegerField()
    name = models.CharField(max_length=100)
    age = models.IntegerField()
    sex = models.CharField(max_length=10)
    email = models.CharField(max_length=100)
    cellphone = models.CharField(max_length=20)
    phone = models.CharField(max_length=20)
    cep = models.CharField(max_length=10)
    state = models.CharField(max_length=50)
    city = models.CharField(max_length=100)
    neighborhood = models.CharField(max_length=100)
    address = models.CharField(max_length=100)
    number = models.CharField(max_length=5)
    cpf = models.CharField(max_length=15)
    cnpj = models.CharField(max_length=15)

    def __str__(self):
        return self.code
```

Fonte: Autoria própria

Após a criação de todos os modelos de dados do chatbot, é preciso executar um comando no terminal para que sejam criados esses campos no banco de dados. São geradas tabelas para cada um dos apps. Os comandos para criação da classe de tabelas são: “Python manage.py makemigrations” e “Python manage.py migrate”. Isso irá criar arquivos de característica serial na pasta migrations de cada app, referenciado cada um com sua tabela no banco.

3.4 Rotas URL

URLs são utilizadas para configurar como os aplicativos e recursos serão localizados, informando o caminho dentro das pastas do projeto. No arquivo urls.py da pasta administrativa é feita a distribuição para os arquivos urls.py de cada aplicação, referenciando que dada uma rota inicial, o processamento será direcionado para alguma das aplicações. Foi configurado para no início das rotas possuir o nome da aplicação que a mesma chamará, por exemplo, dada a rota inicial “/usuarios” o processamento será direcionado ao urls.py da pasta do aplicativo usuarios.

3.4.1 URLs aplicativo usuarios

A Figura 9 mostra como ficou a codificação do arquivo, mostrando os caminhos criados e quais funções são chamadas a partir das rotas do aplicativo de usuários. Lembrando que as ações de controle de usuário podem ser feitas apenas pelo usuário administrador.

Figura 9 - Rotas url aplicativo usuarios

```
urlpatterns = [
    path('login/', login),
    path('entrar/', entrar),
    path('', usuarios),
    path('usuario/<int:code>', usuario),
    path('novo/', novo),
    path('salvarNovo/', salvarNovo),
    path('edicao/<int:id>', edicao),
    path('salvarEdicao/', salvarEdicao),
    path('delecao/<int:id>', delecao),
    path('salvarDelecao/', salvarDelecao),
]
```

Fonte: Autoria própria

3.4.2 URLs aplicativo perguntas

No arquivo de rotas do aplicativo perguntas, está referenciado as respectivas rotas de manipulação das perguntas e respostas cadastradas, como mostrado na Figura 10, bem como os dados que cada rota chama. Também nesse arquivo contém a rota do chatbot em si, que será a rota requisitada todas as vezes que um usuário acessar o chatbot.

Figura 10 - Rotas url aplicativo perguntas

```
urlpatterns = [
    path('<int:code_user>/', perguntas),
    path('pergunta/<int:id>/', pergunta),
    path('novo/<int:code_user>/', novo),
    path('salvarNovo/', salvarNovo),
    path('edicao/<int:id>/', edicao),
    path('salvarEdicao/', salvarEdicao),
    path('delecao/<int:id>/', delecao),
    path('salvarDelecao/', salvarDelecao),
    path('chatbot/<int:code_user>/', chatbot),
    path('questao/<int:code_user>/<int:code_before>/<str:question>/', questao),
    path('api/<int:code_user>/', api)
```

Fonte: Autoria própria

3.4.3 URLs aplicativo home

O arquivo de rotas do aplicativo home chama apenas uma rota vazia, por ser a própria página inicial de controle do chatbot.

Figura 11 - Rota url aplicativo home

```
urlpatterns = [
    path('', home)
```

Fonte: Autoria própria

3.4.4 URLs aplicativo captura

No arquivo de rotas do aplicativo captura será referenciada uma rota contendo os dados dos usuários que já forneceram dados, com isso referencia-se a cada usuário com um código único, como mostrado na Figura 12.

Figura 12 - Rota url aplicativo captura

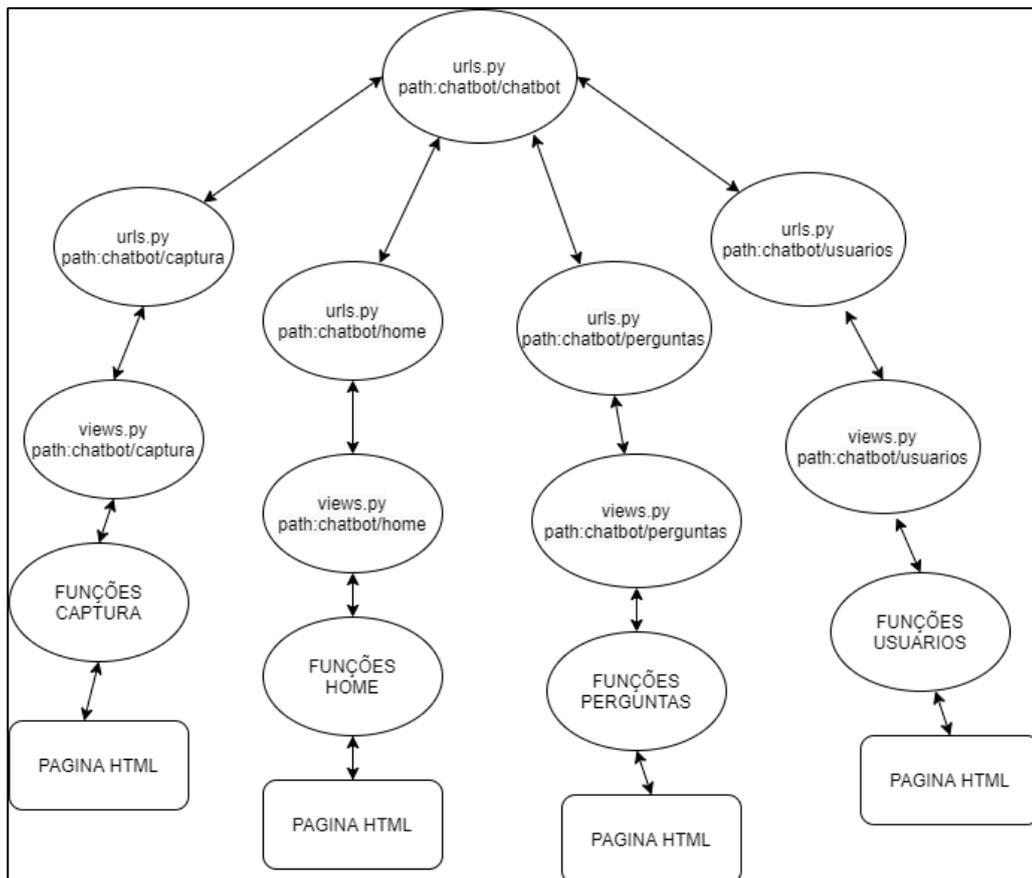
```
urlpatterns = [  
    path('<int:code_user>/', capturas)  
]
```

Fonte: Autoria própria

3.5 Templates

Os *templates* se referem a criação das páginas html que são utilizados para interação com o usuário. Para cada rota criada é preciso criar um *template* html. A Figura 13 apresenta como é a comunicação feita entre os arquivos.

Figura 13 - Rotas das requisições até o template final



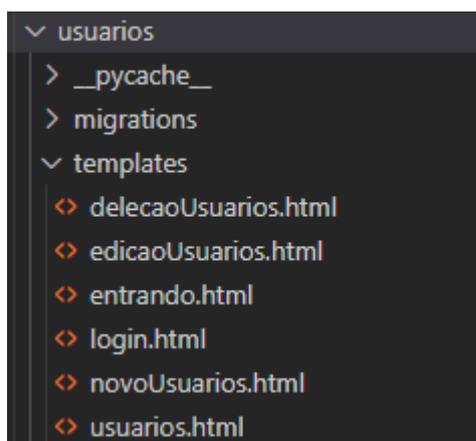
Fonte: Autoria própria

Dada a requisição de uma rota feita em alguma das páginas html, o processo procura o início desta rota no arquivo urls.py da pasta administrativa, nesse arquivo, de acordo com a rota inicial, será feita a distribuição do processamento para os aplicativos criados. Identifica-se então o aplicativo, e é procurado o restante da rota no urls.py da pasta do aplicativo em questão, nesse arquivo, a rota estará chamando uma função a ser processada, função essa por sua vez que deverá estar localizada no arquivo view.py do aplicativo. É acessada a função requisitada, feito o processamento que pode ser requisitar ou fornecer dados ao banco, e ao fim do processamento a função devolve uma resposta que pode ser renderizar uma nova página ou fornecer novos dados a uma mesma página.

3.5.1 Templates do aplicativo usuarios

A pasta de *templates* do aplicativo usuarios, mostrada na Figura 14, possui um arquivo denominado usuarios.html que é usado para exibir os usuários cadastrados. A página novoUsuarios.html é utilizada para cadastrar novos usuários, a edicaoUsuarios.html e delecaoUsuarios.htm para editar e deletar usuários e são acessadas somente pelo administrador do sistema. As páginas login.html e entrando.html são páginas utilizadas para todos os usuários que contribuem na manutenção e atualização do chatbot.

Figura 14 - Templates html do aplicativo usuario



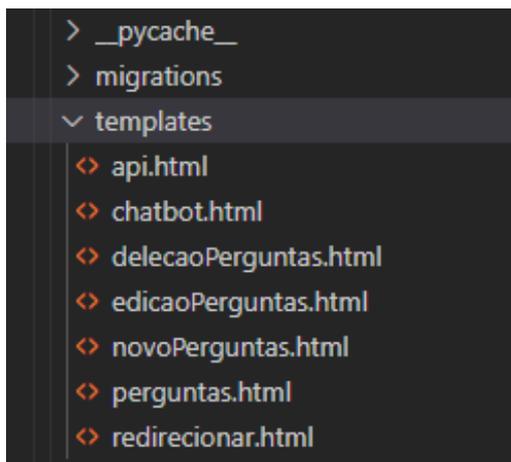
Fonte: Autoria própria

3.5.2 Templates do aplicativo perguntas

A Figura 15 mostra as páginas do aplicativo de perguntas. A página chatbot.html é a página do próprio chatbot, nela foi feita a programação para que o chatbot apresente uma estrutura de um aplicativo de mensagens. A página api.html

contém o código que inicializa a execução do chatbot, `delecaoPerguntas.html`, `edicaoPerguntas.html` e `novoPerguntas.html` são as páginas responsáveis por deletar, editar e criar novas perguntas, respectivamente. O arquivo `perguntas.html` contém as perguntas e `redirecionar.html` redireciona a página que lista as perguntas, funciona como se fosse um *refresh*.

Figura 15 - Templates html acessados pelo aplicativo perguntas

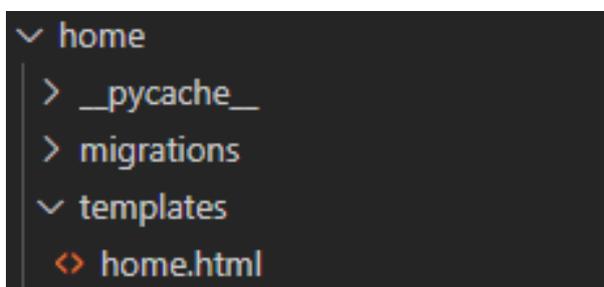


Fonte: Autoria própria

3.5.3 Templates do aplicativo home

A Figura 16 exibe o arquivo `home.html` que é a página de entrada para usuários de controle do chatbot.

Figura 16 - Template html acessado pelo aplicativo home

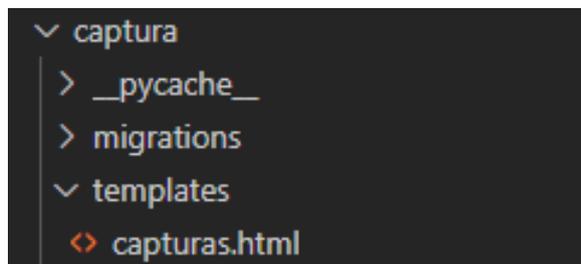


Fonte: Autoria própria

3.5.4 Templates do aplicativo captura

A Figura 17 mostra a pasta *templates* do aplicativo de captura, ela possui o arquivo `captura.html` cuja função é apresentar os dados que já foram capturados de usuários do chatbot.

Figura 17 - Template html acessado pelo aplicativo captura



Fonte: Autoria própria

3.6 Views

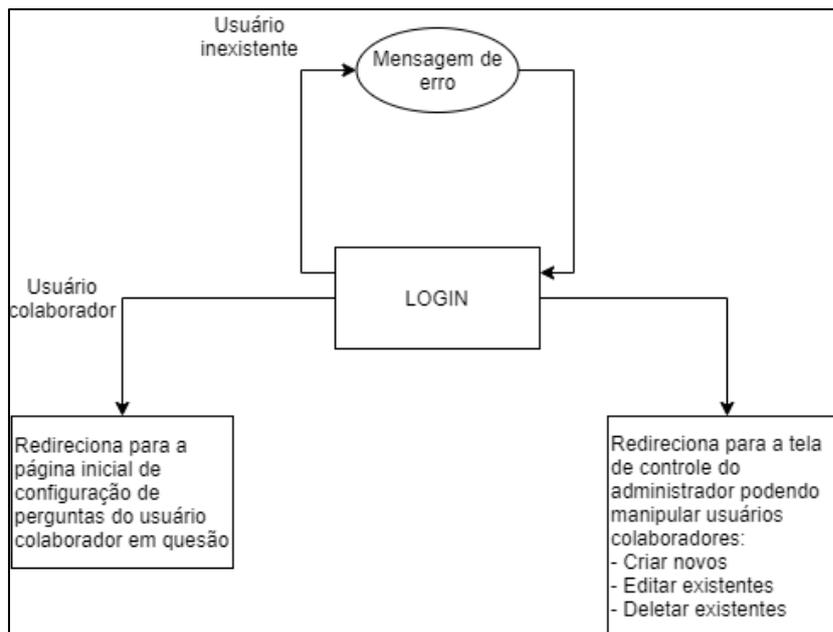
Nas *views* está toda a lógica das funções que o chatbot realiza, desde a interpretação de linguagem natural, até a abertura das páginas html.

Quanto maior o número ou mais complexas forem as funções, maiores e mais complexas de programar são as *views*. A lógica implementada nas *views* dos aplicativos de perguntas e usuarios foi um pouco mais complexa, devido a grande quantidade de funções e ações que precisavam ser feitas.

3.6.1 View do aplicativo usuarios

Na *view* do aplicativo usuarios estão as funções de manipulações de usuários colaboradores, é nela que o administrador do sistema os cadastra e manipula, e identificações de usuários administrador e colaborador. A Figura 18 mostra o diagrama de operação da tela de *login*, e o que será feito após o processamento. A parte de *login* é de responsabilidade da *view* do aplicativo usuarios. Nela há três caminhos possíveis, e para isso é buscado no banco de usuários cadastrados se o usuário e senha digitados estão corretos. O primeiro é a tentativa de entrada com usuário inexistente, e gera uma mensagem de erro, caso o usuário não esteja cadastrado. O segundo é o *login* com um usuário colaborador, nesse caso ele será redirecionado a sua tela inicial de perguntas, onde poderá acessar suas perguntas cadastradas para editar, apagar ou criar novas perguntas. A terceira opção ocorre quando o usuário é o administrador, nesse caso poderá acessar, criar, editar e deletar usuários colaboradores, que são funções também criadas na *view* do aplicativo em questão.

Figura 18 - Acessos dos usuários



Fonte: Autoria própria

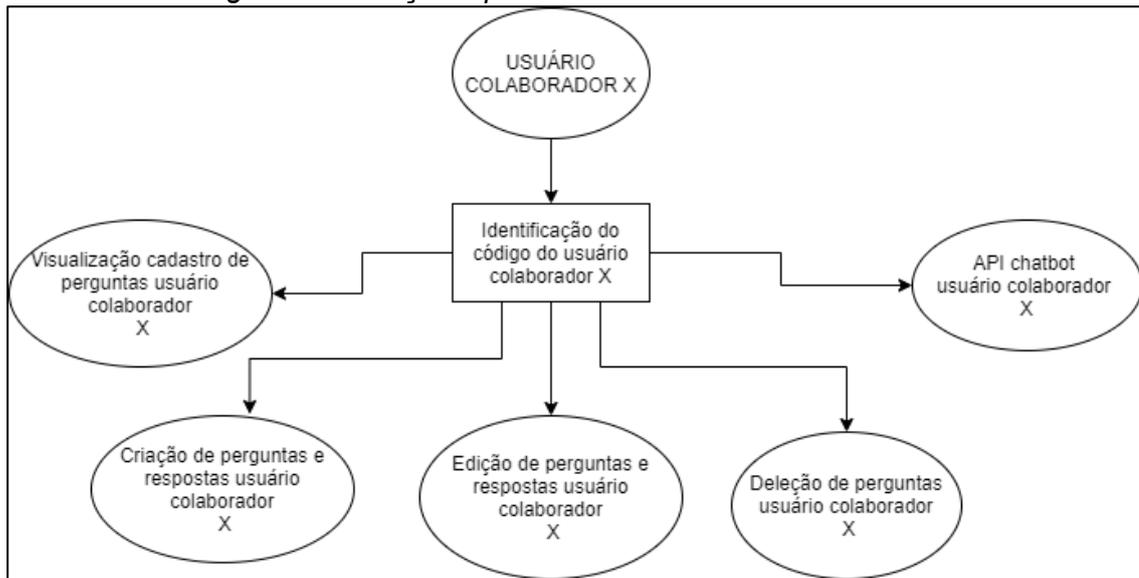
3.6.2 View aplicativo perguntas

Na *view* do aplicativo perguntas é feito o processamento de linguagem natural das mensagens enviadas pelo usuário e o algoritmo determina qual pergunta está relacionada com uma determinada resposta, portanto nela está a lógica do chatbot. Também realiza o controle de criar, editar e deletar mensagens e respostas a serem escolhidas para fornecer ao usuário. As funções são as mesmas descritas que cada rota no `urls.py` da aplicação solicita.

Dentro da *view* de perguntas existe recursos relacionados a usuários colaboradores, e recursos relacionados a inteligência do chatbot. Para recursos relacionados a usuários colaboradores, como a criação ou deleção de perguntas, é necessário que as funções sejam capazes de referenciar a perguntas e respostas do usuário em questão através do seu código de usuário, e a partir disso realizar as ações na tabela que guarda tais informações.

A Figura 19 mostra as funções disponíveis quando um usuário colaborador está logado e for consumir a *view* de perguntas. É necessário identificação seu código de usuário para ter acesso as suas perguntas.

Figura 19 - Funções que o usuário colaborador tem acesso



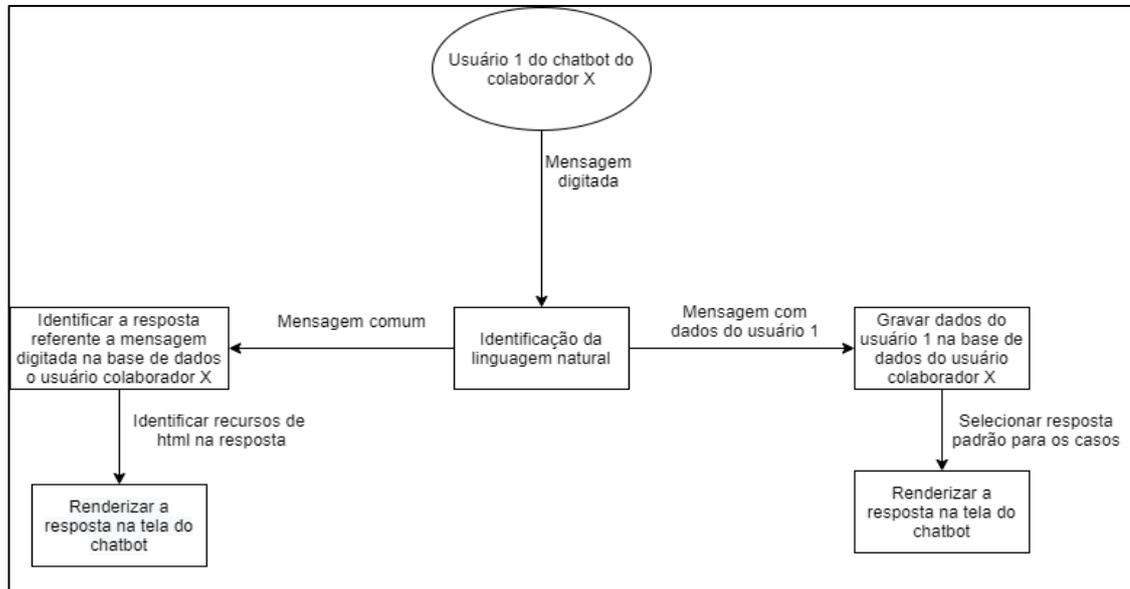
Fonte: Autoria própria

A Figura 20 mostra como funciona o caminho para identificação da mensagem digitada ao chatbot. Primeiramente o algoritmo identifica a informação digitada, através da identificação de linguagem natural, retirando possíveis abreviações, erros de português e acentuação que podem conter na mensagem (recurso da biblioteca unidecode). Após isso é feita uma identificação de que tipo de mensagem foi digitada. Caso sejam informações pessoais, como email, telefone, cpf, dentre outras, o chatbot realiza a coleta e armazena essas informações na tabela do usuário dono do chatbot. Se for uma mensagem comum inicia-se a busca por uma resposta, para isso a mensagem digitada é transformada em uma lista com as palavras da mensagem, então são percorridas todas as mensagens cadastradas pelo usuário dono do chatbot, fazendo a contagem de quantas palavras da mensagem digitada existem dentro da mensagem cadastrada, a princípio o chatbot não terá a função de entender sinônimos. Após percorrer todas as mensagens cadastradas, aquela que possuir o maior número de palavras iguais as da mensagem digitada, é dita como a mensagem identificada, então a partir disso é fornecida a resposta cadastrada para a tela do usuário do chatbot.

Também foi programado um recurso para ajudar na identificação da resposta para a mensagem atual, que leva em consideração a mensagem digitada anteriormente. Isso serve para referenciar que o que está sendo dito tem relação com algo já falado, ou seja, serve para ajudar a manter um diálogo que a partir de mesmas

mensagens digitadas, como a mensagem “sim” que pode ser dita para várias coisas, segue diferentes caminhos de acordo com o que foi dito ou perguntado anteriormente

Figura 20 - Fluxo da requisição e resposta ao usuário do chatbot



Fonte: Autoria própria

Caso a mensagem digitada não esteja na base de dados, o chatbot retorna a resposta que mais se assemelha a pergunta realizada.

Foram abordados nos tópicos anteriores a criação dos modelos de dados, alocação no banco, criação das URLs para identificar qual rota seguir, criação dos templates das páginas html que implementam as chamadas e criação das funções, o arquivo de views que são os responsáveis pelas ações dentro da lógica do chatbot. Os itens seguintes são dedicados a testes.

3.7 Teste chatbot e cadastro de perguntas e respostas

Para inicializar o chatbot utilizando o *prompt* de comando, utiliza-se o comando “python manage.py runserver”, conforme exemplificado na Figura 21.

Figura 21 - Mensagem na linha de comando após iniciar o chatbot

```
chatbot>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
September 23, 2021 - 17:34:36
Django version 3.2.7, using settings 'chatbot.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Fonte: Autoria própria

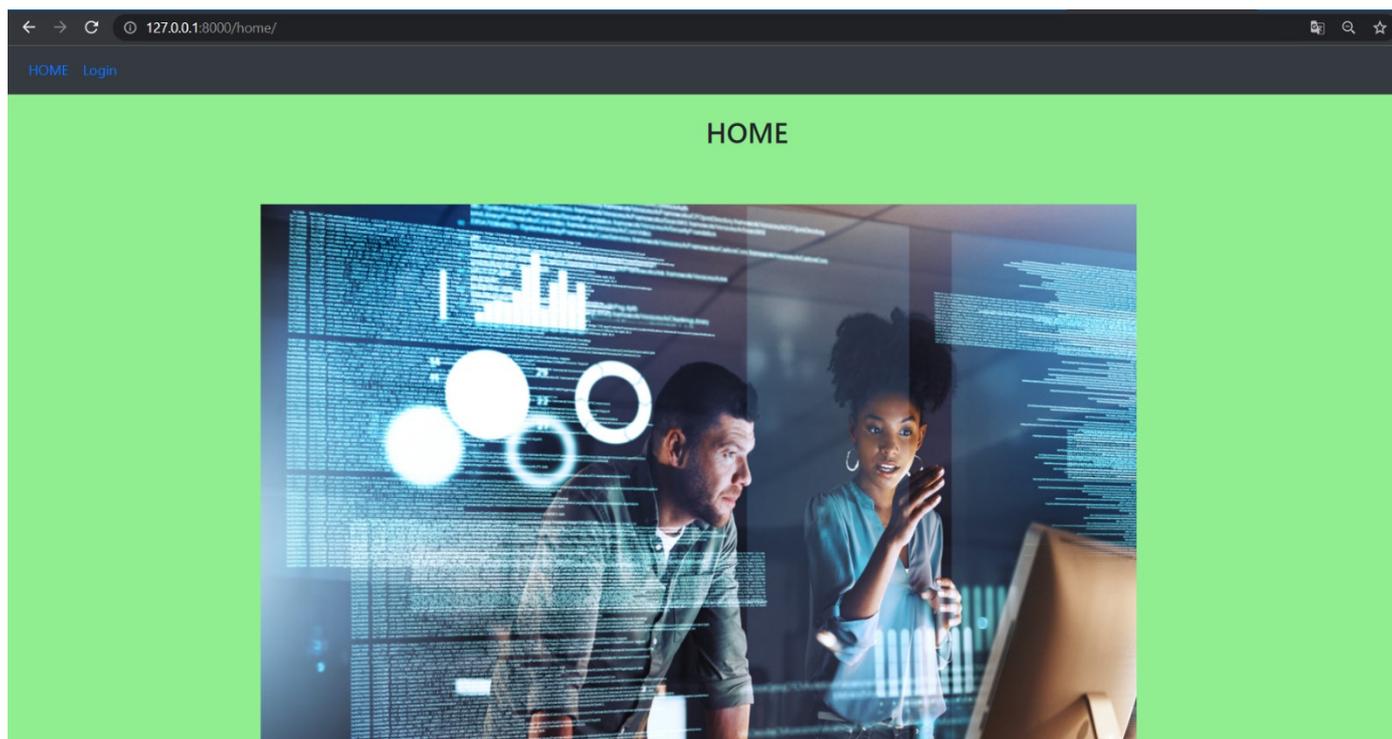
A interface do chatbot pode ser acessada localmente no browser por meio da url <http://127.0.0.1:8000/>.

4 RESULTADOS E DISCUSSÕES

4.1 Home

A Figura 22 mostra a tela inicial, página home, de controle do chatbot. No canto superior esquerdo existe a opção de *Login*, caso a pessoa seja colaboradora ou administradora.

Figura 22 - Tela home da aplicação

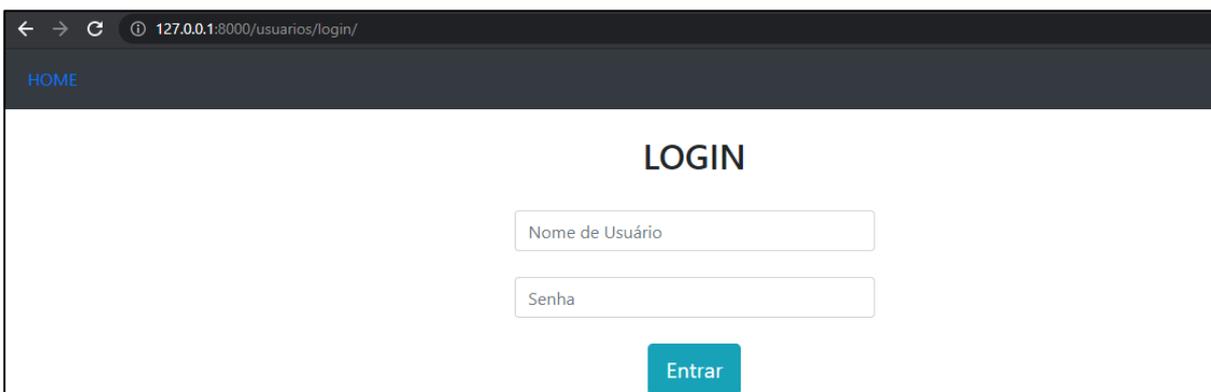


Fonte: Autoria própria

4.2 Login

A Figura 23 mostra a interface de Login, apresentando os campos que deveram ser preenchidos para usuários de controle se conectarem, caso os dados passados não estejam na base de usuários cadastrados, é apresentada uma mensagem de erro.

Figura 23 - Tela de login para usuário administrador ou colaborador



HOME

LOGIN

Nome de Usuário

Senha

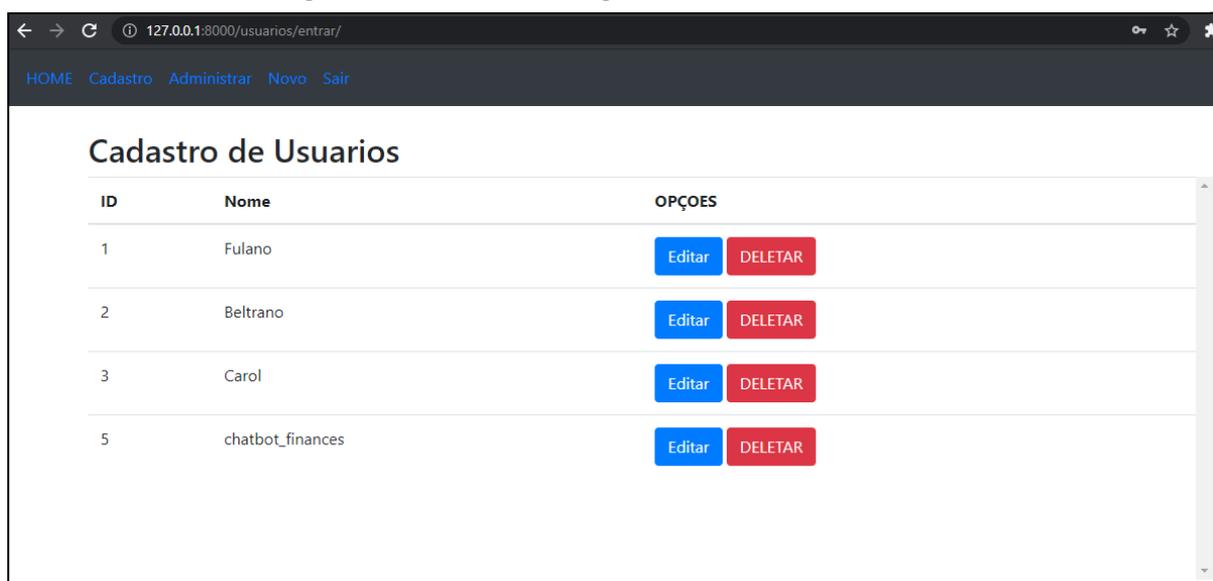
Entrar

Fonte: Autoria própria

4.3 Login/administrador

A Figura 24 mostra a interface inicial após o Login bem-sucedido com usuário administrador. Caso existam colaboradores cadastrados eles serão exibidos, caso contrário aparecerá uma tela vazia. Na barra superior, o administrador possui a opção de voltar para a página home, acessar a interface de cadastro ou sair do sistema.

Figura 24 - Tela inicial logado com usuário administrador



HOME Cadastro Administrar Novo Sair

Cadastro de Usuarios

ID	Nome	OPÇÕES
1	Fulano	Editar DELETAR
2	Beltrano	Editar DELETAR
3	Carol	Editar DELETAR
5	chatbot_finances	Editar DELETAR

Fonte: Autoria própria

Clicando em “Novo” na barra de navegação, é apresentada a tela igual à da Figura 25, onde será preenchido com os dados do novo usuário colaborador. Caso na tela inicial se clicar em editar ou deletar, a interface mostrada é semelhante à da Figura 25 com a diferença de que será apresentado os dados do usuário em questão, podendo editá-los ou deletá-los

Figura 25 - Tela para inserção de novos usuários colaboradores

The screenshot shows a web browser window with the URL 127.0.0.1:8000/usuarios/novo/. The navigation bar contains links for HOME, Cadastro, Administrar, Novo, and Sair. The main content area is titled 'Inserção de Usuários' and contains the following form elements:

- Nome:
- E-MAIL:
- Usuario:
- Senha:
- Usuario Ativo:
- Buttons: Cancelar (grey), Salvar (blue)

Fonte: Autoria própria

4.4 Login/usuário colaborador

A Figura 26 mostra a página inicial apresentada caso o login seja feito por um usuário colaborador, apresentando primeiramente a tela com suas perguntas cadastradas

Figura 26 - Tela inicial logado com usuário colaborador

The screenshot shows a web browser window with the URL 127.0.0.1:8000/perguntas/1010960562810/. The navigation bar contains links for HOME, Cadastro, Chatbot, Novo, Capturas, API, and Sair. The main content area is titled 'Cadastro de Perguntas e Respostas' and displays a table with the following data:

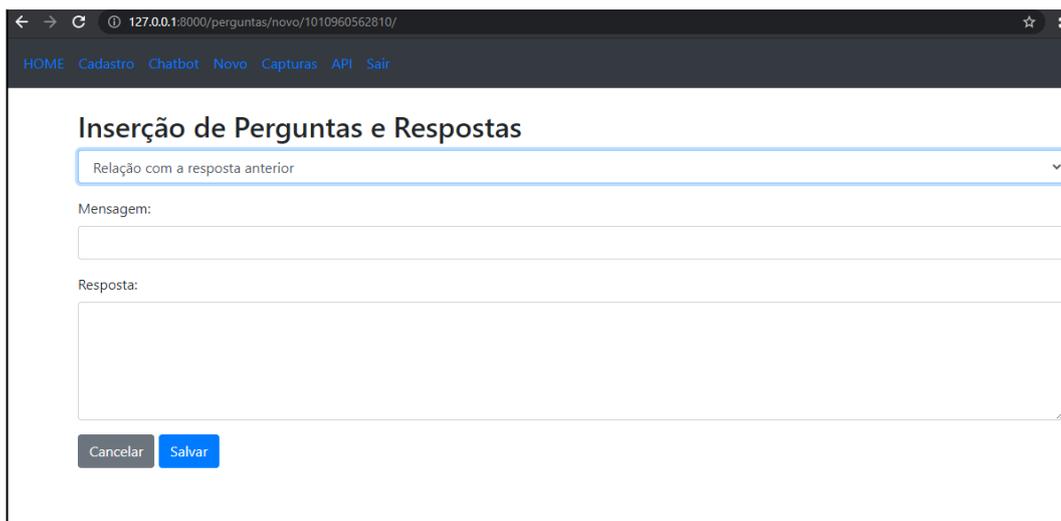
ID	Pergunta	OPÇÕES
13	Como guardar dinheiro?	Editar DELETAR
14	Quando devo começar a economizar?	Editar DELETAR
15	Investir é muito arriscado?	Editar DELETAR
16	Como começar um investimento?	Editar DELETAR
17	Qual o valor certo para começar a investir?	Editar DELETAR
19	Poupança é um bom investimento?	Editar DELETAR

Fonte: Autoria própria

4.5 Cadastro de novas perguntas / editar / excluir

A Figura 27 exibe a tela para adição de perguntas, ou mensagens, e respostas que é apresentada ao se clicar em “Novo” na barra de navegação. O campo “Relação com resposta anterior” é usado para definir se a pergunta terá alguma relação com outra pergunta já cadastrada.

Figura 27 - Tela de inserção para a resposta a certa mensagem enviada

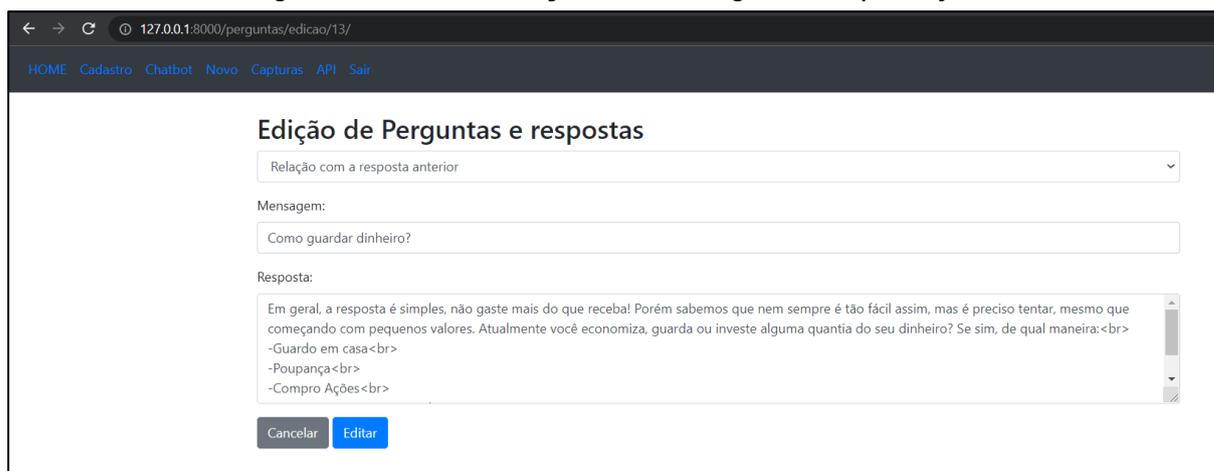


The screenshot shows a web browser window with the URL 127.0.0.1:8000/perguntas/novo/1010960562810/. The navigation bar includes links for HOME, Cadastro, Chatbot, Novo, Capturas, API, and Sair. The main content area is titled 'Inserção de Perguntas e Respostas'. It features a dropdown menu for 'Relação com a resposta anterior', a text input field for 'Mensagem:', and a larger text area for 'Resposta:'. At the bottom, there are 'Cancelar' and 'Salvar' buttons.

Fonte: Autoria própria

A Figura 28 exibe a interface de edição das perguntas cadastradas caso se clique na opção “editar” na tela inicial. Caso se clique em “deletar” a página apresentada será semelhante a Figura 28 porém com botão e função de deletar.

Figura 28 - Tela de edição de mensagem e resposta já cadastradas



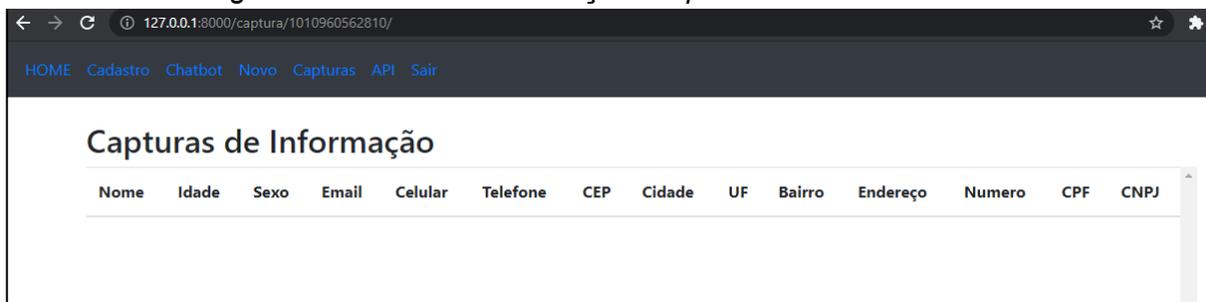
The screenshot shows a web browser window with the URL 127.0.0.1:8000/perguntas/edicao/13/. The navigation bar is the same as in Figure 27. The main content area is titled 'Edição de Perguntas e respostas'. It features a dropdown menu for 'Relação com a resposta anterior', a text input field for 'Mensagem:' containing 'Como guardar dinheiro?', and a larger text area for 'Resposta:' containing the text: 'Em geral, a resposta é simples, não gaste mais do que receba! Porém sabemos que nem sempre é tão fácil assim, mas é preciso tentar, mesmo que começando com pequenos valores. Atualmente você economiza, guarda ou investe alguma quantia do seu dinheiro? Se sim, de qual maneira:
 -Guardo em casa
 -Poupança
 -Compro Ações
'. At the bottom, there are 'Cancelar' and 'Editar' buttons.

Fonte: Autoria própria

4.6 Capturas

A Figura 29 demonstra a tela de captura de informação referente aos dados digitados pelo usuário do chatbot, como nome, número de telefone, endereço, etc.

Figura 29 - Tela com informações capturadas de usuários do chatbot



The screenshot shows a web browser window with the URL `127.0.0.1:8000/captura/1010960562810/`. The navigation menu includes `HOME`, `Cadastro`, `Chatbot`, `Novo`, `Capturas`, `API`, and `Sair`. The main content area is titled `Capturas de Informação` and contains an empty table with the following headers: `Nome`, `Idade`, `Sexo`, `Email`, `Celular`, `Telefone`, `CEP`, `Cidade`, `UF`, `Bairro`, `Endereço`, `Numero`, `CPF`, and `CNPJ`.

Fonte: Autoria própria

Caso o chatbot não tenha capturado nada, a tabela aparecerá em branco como representado na Figura 30.

Figura 30 - Tela com dados de um usuário do chatbot



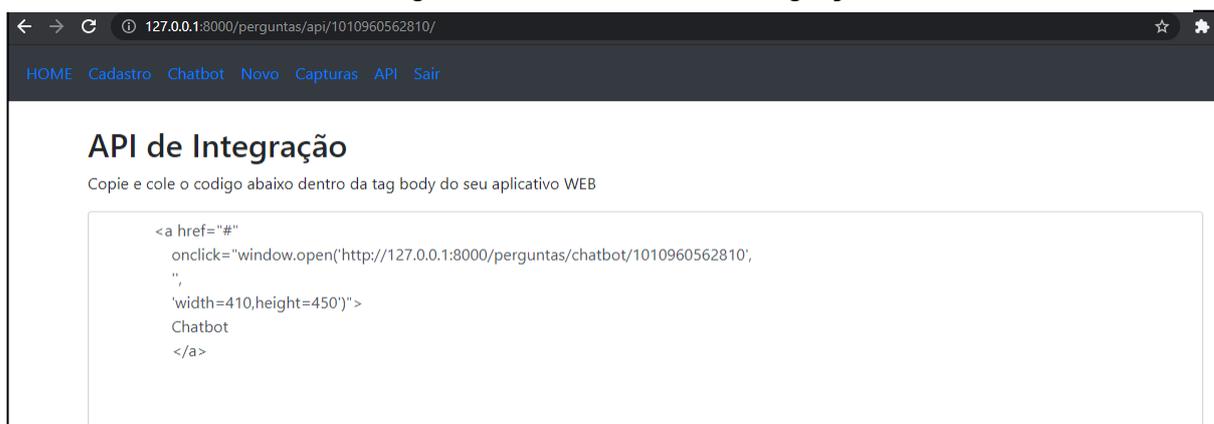
The screenshot shows the same web browser window as Figure 29, but now the table contains one row of data. The headers are: `Nome`, `Idade`, `Sexo`, `Email`, `Celular`, `Telefone`, `CEP`, `Cidade`, `UF`, `Bairro`, and `Endereç`. The data row contains: `PAULO`, `22`, `M`, `paulo@email.com`, `992554822`, `3433333333`, `38412288`, `UBERLÂNDIA`, `MG`, `JARDIM DAS PALMEIRAS`, and `JARDIM`.

Fonte: Autoria própria

4.7 API

A página API se referencia a página que será apresentado o código para o usuário colaborador poder oferecer o chatbot por ele criado para outras pessoas através de uma página na web. Como mostrado na Figura 31, a página apresenta o código que basta ser copiado e colado no arquivo html de alguma página na internet, que estará fornecendo seu chatbot. Vale ressaltar que o caminho inicial está `127.0.0.1:800`, pois a imagem foi retirada da tela enquanto a aplicação executava localmente, caso executasse em um site ou servidor, essa rota seria substituída pela rota padrão do local hospedado.

Figura 31 - Tela da API de integração

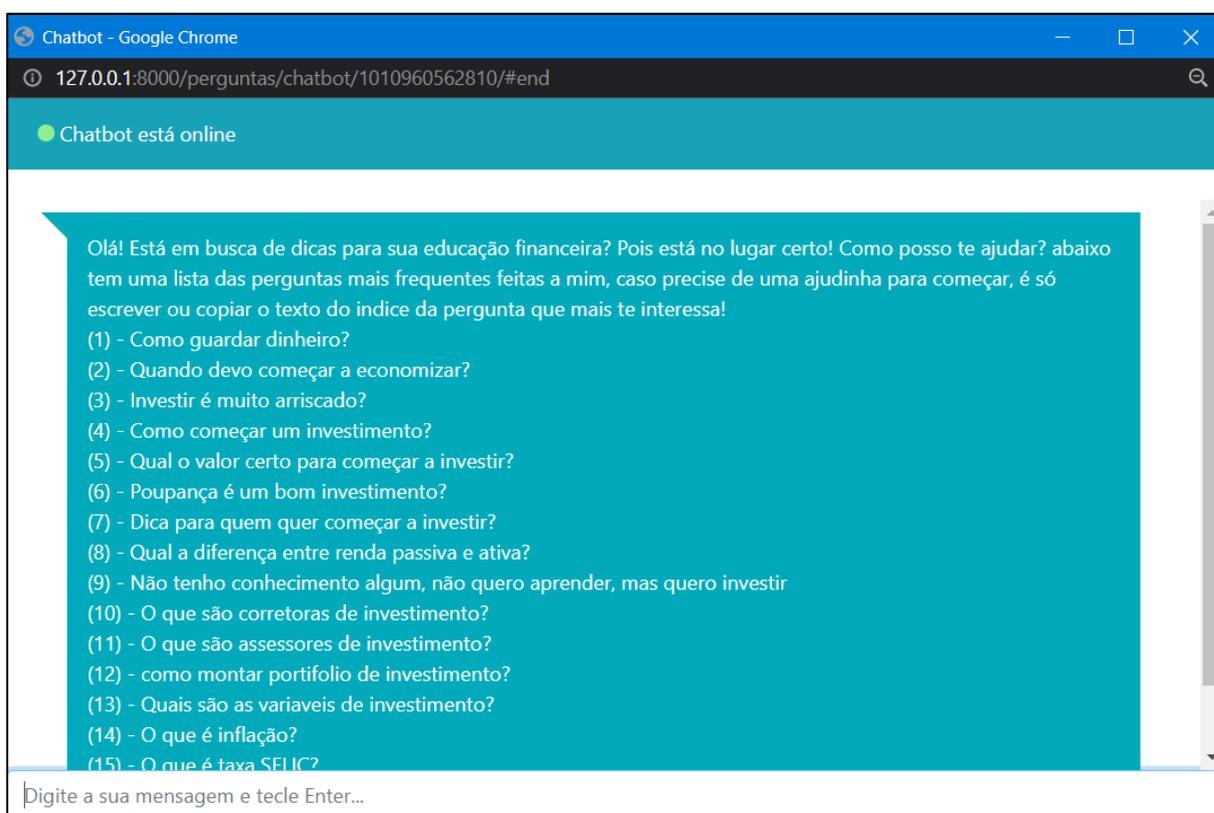


Fonte: Autoria própria

4.8 Chatbot

A Figura 32 mostra a interface do chatbot. A mensagem inicial são alguns tópicos de exemplo que o usuário pode explorar.

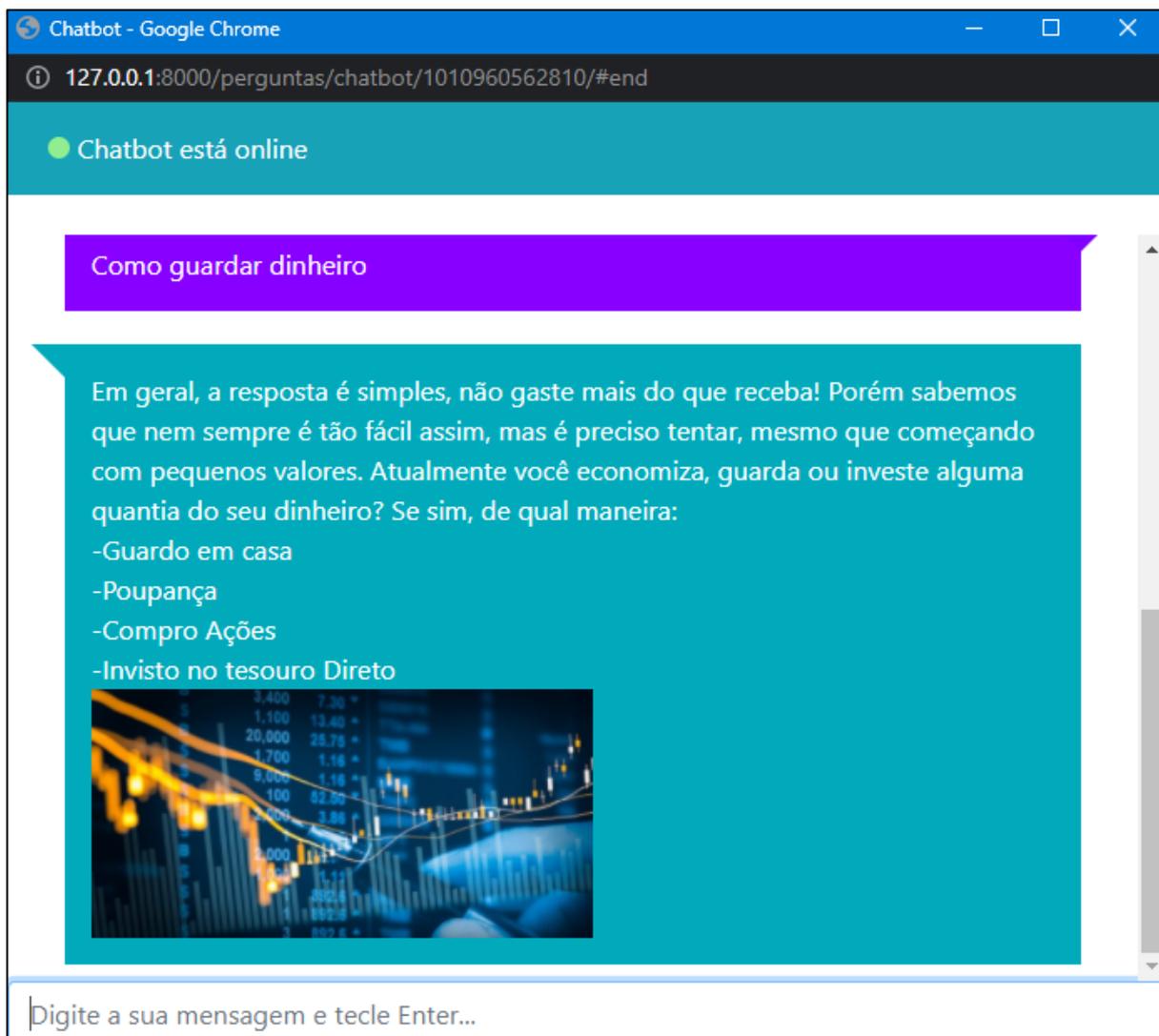
Figura 32 - Tela do chatbot de finanças



Fonte: Autoria própria

O usuário digita sua mensagem no campo com a marca d'água “Digite sua mensagem e tecla Enter”, a partir do momento que a mensagem for digitada e a tecla Enter apertada, a mensagem é enviada para o processo de identificação descrito na view do aplicativo de perguntas, e então a resposta é fornecida. A Figura 33 apresenta uma resposta dada a uma mensagem digitada.

Figura 33 - Exemplo 1 de resposta do chatbot



Fonte: Autoria própria

O chatbot também possui recursos de mostrar imagens. Comandos HTML e tags podem ser digitados junto com as respostas fornecidas, que o chatbot é capaz de interpretá-los.

Com a mensagem digitada da maneira que foi cadastrada pelo usuário dono do chatbot em questão, ou mesmo com as mesmas palavras, porém em ordem diferente, a identificação é bem sucedida, porém o algoritmo não é capaz de entender

sinônimos das palavras. Para a identificação de sinônimos seria necessário integrar a solução com alguma API do mercado que consiga prover a funcionalidade, o que pode ser uma solução para o futuro.

Outro ponto identificado foi o fato de que quanto maior o número de palavras, tanto digitadas pelo usuário, como o número de palavras presente na mensagem configurada pelo dono do chatbot, maior a chance de o algoritmo referenciar a mensagem de forma correta, visto que a comparação é feita utilizando o número de palavras digitadas existentes na frase cadastrada pelo usuário colaborador.

Na *view* do aplicativo de perguntas foi explicado sobre a função de relacionar a mensagem atual com uma mensagem anterior, visando possibilitar rumos e relações diferentes mesmo que sejam digitadas mensagens comuns como a mensagem “sim”, comparando as Figuras 34 e 35 é possível entender o funcionamento.

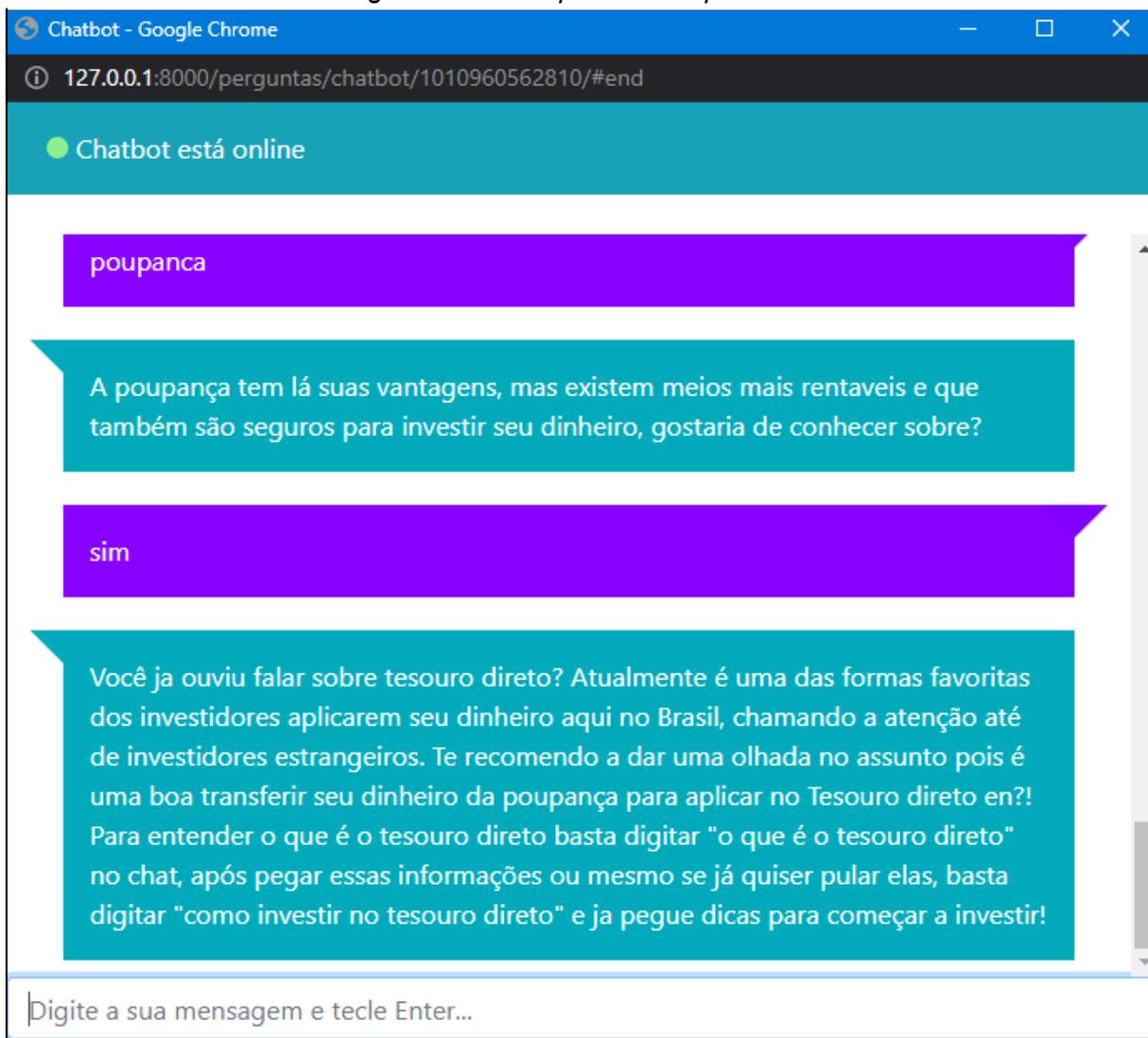
Figura 34 - Exemplo 2 de resposta do chatbot



Fonte: Autoria própria

A Figura 35 exibe uma mensagem enviada após o usuário responder “sim” para uma pergunta feita ao chatbot. Porém a resposta “sim” pode ser dada para diversas outras perguntas que podem estar cadastradas, com isso se relaciona ela a mensagem digitada anterior a ela. Caso seja cadastrada outra mensagem “sim” mas com relação a uma mensagem anterior diferente da citada na Figura 34, a resposta será diferente, assim como mostrado na Figura 35.

Figura 35 - Exemplo 3 de resposta do chatbot



Fonte: Autoria própria

O recurso implementado ajudou a fornecer inteligência para o chatbot, e foi um recurso implementado com o decorrer dos testes realizados. Com a referência única a mensagem digitada, mensagens menores que poderiam prover inúmeras respostas seguintes diferentes, se prendiam a apenas uma resposta seguinte, com isso o recurso se mostrou útil para que as expressões e respostas padrões de forma avulsa pudessem ser compreendidas de acordo com o contexto inserido.

Com os resultados apresentados devido ao modo que foi criado, o chatbot se apresenta como uma boa ferramenta de ensino devido a lógica inserida no mesmo, a partir dos resultados das requisições feitas com as mensagens digitadas, a inteligência artificial se mostrou capaz de identificar o que foi passado. A evolução do projeto fica por conta da identificação de linguagem natural, visto que na língua portuguesa existem diversos sinônimos para uma mesma palavra. Como o chatbot faz uma comparação palavra a palavra, ele não está preparado para identificar sinônimos, com isso uma estratégia interessante a se seguir é fornecer pro usuário as mensagens e dúvidas que ele possa digitar para que o que venha a ser digitado esteja de acordo com o cadastrado.

A criação se mostra adequada para dúvidas que seguem um padrão, que podem ser recorrentes para diversas pessoas, e através do acesso e da digitação da mensagem, ela receba a resposta adequada. O chatbot consegue manter diálogos mais longos de acordo com o construído, porém para uma maior assertividade o ideal seria construir um mecanismo de identificação mais robusto, porém isso demandaria mais recursos externos ao python puro.

5 CONCLUSÃO

A tecnologia avança e com isso o mundo sofre mudanças, mudanças essas que refletem nos hábitos das pessoas, na maneira de se comunicarem, na maneira de aprenderem. Entendendo o mundo que se vive, e a intenção inicial de ensinar educação financeira e os recursos utilizados, o chatbot construído se mostrou condizente com o proposto. A solução se mostrou capaz de identificar e comparar as mensagens digitadas com a base de dados fornecida na configuração do chatbot, porém com algumas limitações devido aos limites ferramentais.

Para que a solução fosse ainda mais assertiva, capaz de entender sinônimos, entender todas as palavras mesmo que digitadas de forma errada, seria necessário integrar novos recursos a solução, o que por um lado pode ser útil visando identificar diálogos ainda mais informais, porém pode ser uma falha no projeto depender de recursos externos que possam ser desativados no futuro.

Os chatbots por vezes são utilizados para manter um diálogo com uma pessoa, ou simplesmente fornecer informação à dúvidas que são recorrentes nas pessoas, o chatbot criado se mostra voltado a sanar dúvidas pontuais baseadas em mensagens que seguem um padrão, e apesar de possuir recursos para configurar informações necessárias para que seja mantido um diálogo mais longo, dependendo dos caminhos tomados a precisão das respostas podem diminuir.

Com o objetivo traçado de construir um chatbot capaz de ensinar educação financeira, os resultados obtidos foram satisfatórios, a integração de tecnologias como a linguagem python, recurso de banco de dados e recursos *web*, apresentaram um ótimo e fiel resultado ao que foi proposto. O chatbot construído possibilita não somente conversas sobre finanças, mas sobre qualquer coisa, pode ser utilizado por usuários das mais diversas finalidades e hospedado em qualquer aplicação web. Isso abre margem para que a aplicação cresça e desenvolva novas funcionalidades, como cálculos precisos, geração de gráficos em tempo real, se mostrando uma aplicação com boa escalabilidade, preparada para resolver o problema da educação financeira da sociedade e com boa margem de evolução para outras áreas.

6 Referências

S. M. Sackey, A. Bester, *Industrial engineering curriculum in Industry 4.0 in a South African context.*, S. Afr. J. Ind. Eng., vol. 27, pp. 101–114, 2016. Acesso em: <https://doi.org/10.7166/27-4-1579>

R. Khan, S. U. Khan, *Design and implementation of an automated network monitoring and reporting back system.*, J. Ind. Inf. Integr., vol. 9, pp. 24–34, 2018. Acesso em: <https://doi.org/10.1016/j.jii.2017.11.001>

M. M. Mabkhot, A. Al-Ahmari, B. Salah, H. Alkhalefah, *Requirements of the smart factory system: A survey and perspective.*, Machines, vol. 23, pp. 1-22, 2018. Acesso em: <https://doi.org/10.3390/machines6020023>

INTERNATIONALE HOCHSCHULE *Com seu diploma em Engenharia Industrial Indústria 4.0, você está pronto para a digitalização de ramos inteiros da indústria.* Acesso em 20 de Junho de 2021, em https://www.iu.de/kombistudium/lp/bachelor/wirtschaftsingenieurwesen-industrie-40/?clickref=1101lhwtGX4&utm_source=private-network&utm_medium=koop&utm_campaign=partner1100l37277

Profissional Interdisciplinar: Por que é mais valorizado?. UNOPAR, 2019. Acesso em 19 de Setembro de 2021, em: <https://blog.unopar.com.br/profissional-interdisciplinar/>

LEWGOY, Julia, Educação financeira nas escolas fica pra trás em meio a democratização de investimentos, *ValorInveste*, 2021. Acesso em 19 de Setembro de 2021, em: <https://valorinveste.globo.com/educacao-financiera/noticia/2021/06/22/educacao-financiera-nas-escolas-fica-para-tras-em-meio-a-democratizacao-de-investimentos.ghtml>

Business Insider's premium research service. THE MESSAGING APPS REPORT: Messaging apps are now bigger than social networks. Acesso em 15 de Fevereiro de 2021, em: <https://www.businessinsider.com/the-messaging-app-report-2015-11>

LA CRUZ, Fernanda. Chatbots na educação: um novo projeto de aprender e ensinar, 2019. Acesso em 18 de Setembro de 2021, em : <https://cesu.cps.sp.gov.br/chatbots-na-educacao-um-novo-jeito-de-aprender-e-ensinar/>

GOEKING, Weruska. Apenas 3% dos brasileiros investiram em 2020 e média aplicada caiu 31 %, *ValorInveste*, 2021. Acesso em 19 de Setembro de 2021, em: <https://valorinveste.globo.com/mercados/renda-variavel/bolsas-e->

indices/noticia/2021/01/15/apenas-3percent-dos-brasileiros-investiram-em-aco-es-em-2020-e-media-aplicada-caiu-31percent.gHTML

Participação dos Investidores na bolsa brasileira, *B3*, 2021. Acesso em 19 de Setembro de 2021, em: https://www.b3.com.br/pt_br/market-data-e-indices/servicos-de-dados/market-data/consultas/mercado-a-vista/participacao-dos-investidores/volume-total-acumulado/

Raio X do investidor brasileiro, *ANBIMA*, 2019. Acesso em 19 de Setembro de 2021, em: https://www.anbima.com.br/pt_br/especial/raio-x-do-investidor-2019.htm

OLIVEIRA, Gabriel. 11 linguagens de programação em alta no mercado em 2021, *Digital Innovation One*, 2021. Acesso em 20 de Setembro de 2021, em: <https://digitalinnovation.one/artigos/11-linguagens-de-programacao-em-alta-no-mercado-em-2021>

Top 5 aplicativos feitos com Django, *ByLearn*, 2020. Acesso em 20 de Setembro de 2021, em: <https://dojo.bylearn.com.br/Python/5-aplicativos-feitos-com-django/>

TIAGO, O que é Django e como usar no desenvolvimento web, *Mundo Devops*, 2020. Acesso em 20 de Setembro de 2021, em: <https://mundodevops.com/blog/django-Python/>