Projeto Artemis - Inteligência Artificial no Salesforce

Gabriel Rodrigues, João Victor Glanzmann, Matheus Costa
gabriellrodriigues17@gmail.com
jvxavierbh@gmail.com
matheusilva334@gmail.com
Professor orientador: Ricardo Noboru Igarashi
Coordenação de curso de Engenharia de computação

RESUMO

Este estudo investiga e trata a implementação de um modelo de RAG (Retrieval-Augmented Generation) e a integração da OpenAI no Salesforce para desenvolver chatbots contextualizados que respondam a perguntas de forma precisa e relevante. O objetivo é criar uma solução que combine técnicas de recuperação de informações com geração de texto para aprimorar o atendimento ao cliente e otimizar processos de suporte e vendas. A metodologia envolve a configuração de objetos customizados no Salesforce para armazenar conteúdos, além do uso de classes controladoras e APIs de IA generativa para gerar respostas contextuais com base no conteúdo recuperado. Como resultado, o sistema entrega respostas contextualizadas que aumentam a qualidade do atendimento e oferecem um diferencial competitivo. A integração de RAG e outras IAs no Salesforce viabiliza o desenvolvimento de chatbots com respostas altamente relevantes e atualizadas, respondendo à crescente demanda por atendimento automatizado eficaz.

Palavras-chave: Chatbot. Salesforce. RAG. Atendimento ao cliente. IA generativa.

1. INTRODUÇÃO

A transformação digital impulsionou o uso da Inteligência Artificial (IA) em diversas áreas, especialmente no atendimento ao cliente, onde soluções avançadas oferecem interações mais personalizadas e contextuais. A integração de sistemas de IA generativa, como OpenAI, ao Salesforce, representa uma nova fronteira para o atendimento automatizado, possibilitando que chatbots não só respondam a perguntas de forma precisa, mas também ofereçam respostas embasadas em grandes volumes de dados e contexto. Segundo Russel e Norvig (2021), “a IA não se limita a resolver problemas específicos; ela procura criar sistemas que raciocinem e se adaptem” (p. 5), uma premissa essencial ao desenvolvimento de chatbots avançados para atendimento contextualizado.

Este estudo foca na implementação de um modelo de *Retrieval-Augmented Generation* (RAG), com a proposta de integrar IA generativa via OpenAI ao Salesforce, visando uma resposta mais precisa às demandas por automação eficaz. Ainda que técnicas de recuperação e geração de texto já sejam amplamente estudadas, a criação de um sistema que combina ambas de forma eficiente para o atendimento automatizado segue como um desafio na área de IA aplicada a negócios, especialmente em plataformas de CRM como o Salesforce.

Dado o crescimento das interações automatizadas e a necessidade por respostas contextualizadas, a pesquisa visa desenvolver uma solução que utilize técnicas de RAG para fornecer informações relevantes com precisão e velocidade. Segundo Jurafsky e Martin (2020), um dos principais objetivos da IA em linguagem natural é “compreender e gerar texto de forma que as respostas sejam ajustadas ao contexto e objetivos do usuário” (p. 12), alinhando-se ao propósito desta investigação. Assim, este estudo propõe suprir uma lacuna existente, onde os modelos de IA devem interagir de maneira robusta com grandes bases de dados e atender a uma alta demanda, produzindo respostas atualizadas e com aplicabilidade prática.

COLOCAR INTRO SALESFORCE

2. DESENVOLVIMENTO

Este estudo fundamenta-se na Inteligência Artificial aplicada ao atendimento ao cliente, no modelo Retrieval-Augmented Generation (RAG) e na plataforma Salesforce, pilares essenciais para o desenvolvimento de chatbots que utilizam IA generativa. A organização e apresentação estruturada desses conceitos são cruciais para criar um sistema de suporte que entregue respostas precisas e personalizadas. A seguir, são abordados os conceitos-chave que sustentam a construção desse chatbot contextualizado no Salesforce, com o objetivo de oferecer um atendimento automatizado e eficiente. A descrição geral do nosso produto é mostrada na tabela abaixo:

Ator	Definição
Administrador	Responsável pelo gerenciamento
Usuário	Usuários que irão efetuar pesquisas de acordo com o que desejam (equipes de vendas e atendimento ao cliente)
API de Geração de Texto	APIs externas que processam as informações recuperadas pelo sistema de busca e geram respostas em linguagem natural.
Sistema de Gestão de Conhecimento (KMS ou knowledge)	Sistema onde a empresa armazena e gerencia sua base de conhecimento interna, como artigos, manuais e FAQs.

Para nos conectar à API da OpenAI que faz conexão com o Salesforce, utilizamos a documentação da OpenAI: https://platform.openai.com/docs/api-reference/making-requests. Nela, pudemos verificar como a requisição deveria ser montada para acessar a API e coletar as informações necessárias:

Figura 1 - Making Requests

```
{
    "id": "chatcmpl-abc123",
    "object": "chat.completion",
    "created": 1677858242,
    "model": "gpt-4o-mini",
    "usage": {
        "prompt_tokens": 13,
        "completion_tokens": 7,
        "total_tokens": 20,
        "completion_tokens_details": {
            "reasoning_tokens": 0
        }
    },
    "choices": [
        {
            "message": {
                "role": "assistant",
                "content": "\n\nThis is a test!"
            },
            "logprobs": null,
            "finish_reason": "stop",
            "index": 0
        }
    ]
}
```

Fonte: Autor

No formato de JSON da figura 1 obtido na documentação, será necessária a criação de um método privado estático, que receberá a resposta HTTP (HttpResponse) e processa seu corpo JSON para extrair uma mensagem específica. Primeiro, será inicializada uma instância de FlowOutput, que será usada para armazenar o conteúdo da mensagem extraída. Em seguida, o método desserializará o corpo da resposta JSON em um Map, armazenando-o em uma variável. O método então acessará a lista choices dentro de gptResponse, que contém opções de mensagens. Caso choices não esteja vazio, ele pega o primeiro item dessa lista, que é mapeado para um objeto choice, e extrai o mapa message contido em choice. Por fim, ele armazena o conteúdo da mensagem (obtido da chave content dentro de message) no atributo returnMessage de flowOutput. O método retorna uma lista contendo o objeto flowOutput, com a mensagem extraída

2.1 Inteligência Artificial e Atendimento ao Cliente Automatizado

A IA tem desempenhado um papel transformador no atendimento ao cliente, proporcionando interações que vão além de respostas programadas, oferecendo contextos personalizados e interações fluidas. Russell e Norvig (2021) argumentam que "a IA não apenas soluciona problemas, ela adapta-se para criar sistemas inteligentes de suporte" (p. 5), o que torna a aplicação de IA em chatbots extremamente valiosa. Esses modelos utilizam técnicas de Processamento de Linguagem Natural (PLN) para interpretar e responder às demandas dos clientes de forma dinâmica. No caso deste projeto, a IA possibilita que o chatbot ofereça respostas com base no contexto específico do Salesforce, reduzindo a necessidade de intervenção humana e aumentando a precisão nas respostas.

2.2 Retrieval-Augmented Generation (RAG)

O modelo RAG combina técnicas de recuperação de informações com a geração de linguagem natural, baseado em machine learning, que oferece uma abordagem inovadora para criar respostas fundamentadas no conteúdo existente. Lewis et al. (2020) definem o RAG como um modelo que "utiliza a integração entre recuperação de dados e geração de texto para gerar respostas altamente relevantes" (p. 8). No contexto deste estudo, o RAG é empregado para que o chatbot no Salesforce não apenas responda automaticamente, mas busque informações relevantes com a contextualização do Salesforce. Isso permite um diferencial importante na experiência do usuário, que recebe respostas de alto valor com menor a necessidade de uma busca manual ou reenvio para uma equipe de suporte. Mas é importante prever e estar preparado no contexto do ChatBot, para cenários não previstos.

2.3 Integração de OpenAI com o Salesforce para IA Generativa

A utilização de modelos de IA generativa da OpenAI amplia as possibilidades de atendimento ao cliente, fornecendo respostas que simulam a compreensão e a adaptação ao contexto de conversa. Jurafsky e Martin (2020) destacam que "a aplicação de IA para interpretação e geração de texto oferece potencial inédito para interações mais humanas e alinhadas ao contexto do cliente" (p. 12). A integração do modelo de linguagem da OpenAI com uma classe controladora do Salesforce traz ao sistema a capacidade de interpretar as nuances do atendimento e gerar respostas complexas em tempo real, proporcionando interações que superam os limites dos chatbots tradicionais. Este avanço permite ao chatbot responder de maneira mais contextualizada e relevante, ampliando o alcance e a qualidade do atendimento ao cliente.

2.4 Estrutura de Atendimento e Armazenamento de Dados no Salesforce

A estruturação do atendimento no Salesforce, através de objetos customizados e classes controladoras, é essencial para o armazenamento, recuperação e manipulação dos dados necessários para o funcionamento do chatbot. Segundo Pereira (2014), uma estrutura sólida de dados é indispensável em sistemas de automação, pois “organiza o fluxo de dados e garante que o atendimento ao cliente seja preciso e responsivo” (p. 29). Neste estudo, a arquitetura do Salesforce foi adaptada para suportar o RAG, com dados sendo armazenados e organizados de forma que permitam uma recuperação rápida e precisa. A integração entre os dados internos e as APIs de IA gera um atendimento ao cliente mais eficaz e competitivo, sendo a automação um diferencial significativo em mercados que valorizam respostas rápidas e contextualizadas.

2.5 Recuperação de Informação em Tempo Real

A recuperação de informações é central para a aplicação do RAG. Utilizando tecnologias de busca avançadas, a integração com o Salesforce permite que o chatbot localize rapidamente dados contextuais para suas respostas. Lewis et al. (2020) explicam que a “recuperação de dados em tempo real sustenta a precisão dos modelos de RAG” (p. 9), o que possibilita a entrega de respostas contextualizadas e atualizadas em cada interação. Esta capacidade de recuperação é essencial para assegurar que o chatbot ofereça respostas fundamentadas e em conformidade com as informações mais atuais, aumentando a satisfação do cliente e melhorando a eficiência operacional da empresa.

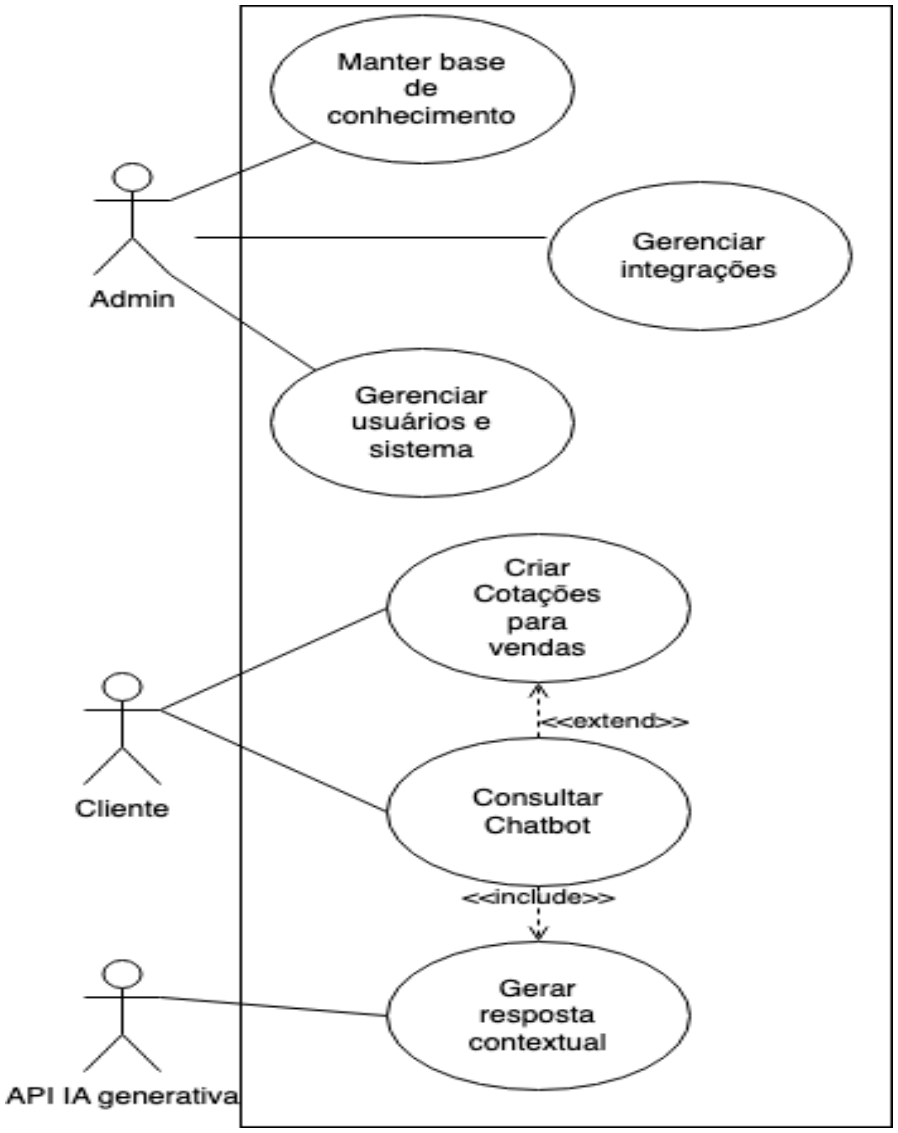
3. METODOLOGIA[8]

A modelagem de software é uma etapa essencial no desenvolvimento, pois possibilita a visualização e organização de requisitos e funcionalidades antes da implementação. Modelos como o diagrama de casos de uso e o diagrama de classes oferecem uma representação clara das interações entre usuários e sistema, bem como da estrutura interna do software. O diagrama de casos de uso identifica e organiza os cenários de interação entre os atores e as funcionalidades esperadas do sistema, ajudando a alinhar as expectativas com as necessidades dos usuários. Já o diagrama de classes descreve a estrutura dos objetos no sistema, especificando atributos, métodos e relações, o que facilita a coesão e o reuso de componentes.

Segundo Sommerville (2011), "modelos de software permitem que os desenvolvedores e partes interessadas entendam a estrutura do sistema antes de implementar o código, reduzindo o risco de erros e retrabalho" (p. 189). Dessa forma, a modelagem é uma fase crítica que aumenta a qualidade do software e agiliza o processo de desenvolvimento.

O diagrama de caso de uso mostrado na figura 2, mostra os possíveis cenários de utilização:

Figura 2 - Diagrama de caso de uso

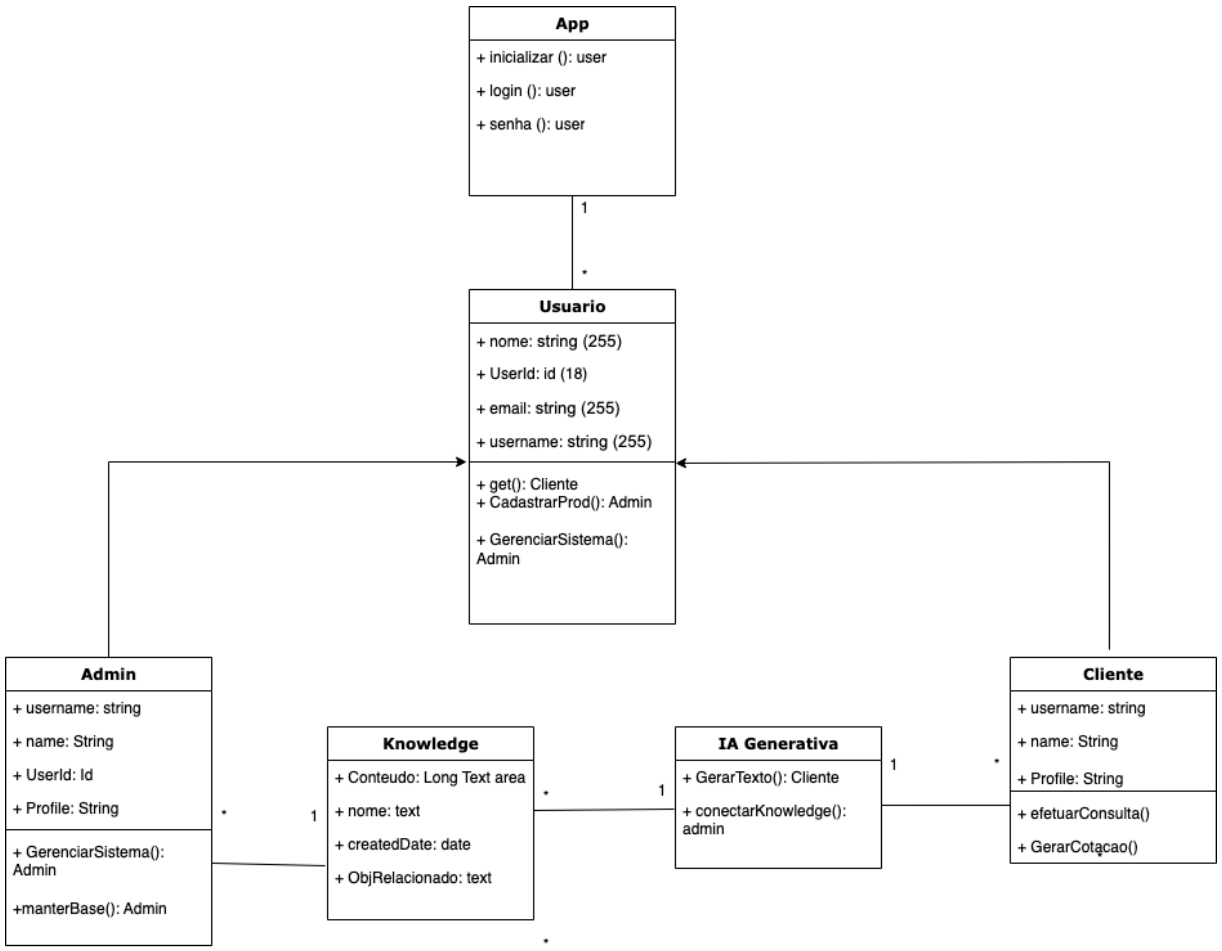


Fonte: Autor

O Salesforce utiliza a programação orientada a objetos (POO), que é uma abordagem que organiza o código em "objetos" que combinam dados e comportamentos, permitindo uma estrutura modular e reutilizável; segundo Booch (1991), "POO facilita a modelagem de sistemas complexos ao refletir entidades do mundo real como objetos de software

interconectados." Como mostrado na figura 2, serão utilizados alguns objetos padrões do Salesforce, como os usuários, que poderão ter perfis de administradores ou de clientes, e o knowledge, que é uma base de conhecimento interna do Salesforce. Mas também, será necessária uma classe controladora da IA Generativa, para gerar as requisições e respostas com o servidor.

Figura 3 - Diagrama de classes



Fonte: Autor

Com a modelagem do nosso sistema pronto, tivemos em mente como deveria funcionar a classe controladora para fazermos a requisição à API e recebermos a resposta, como mostrado na figura 4.

Figura 4 - Classe Controladora do ChatBot - Input e Output

```
global with sharing class ChatbotController {
    private static final String CHAT_GPT_KEY = System.label.chatGPTAPIkey;
    private static final String endpoint = 'https://api.openai.com/v1/chat/completions';

    global class FlowInput {
        @InvocableVariable(required = true)
        public String questionMessage;
    }

    global class FlowOutput {
        @InvocableVariable
        public String returnMessage;
    }

    private static List<FlowOutput> getMessage(HttpResponse response){
        FlowOutput flowOutput = new FlowOutput();

        Map<String, Object> gptResponse = (Map<String, Object>) JSON.deserializeUntyped(response.getBody());
        List<Object> choices = (List<Object>) gptResponse.get('choices');

        if (!choices.isEmpty()) {
            Map<String, Object> choice = (Map<String, Object>) choices[0];
            Map<String, Object> message = (Map<String, Object>) choice.get('message');

            flowOutput.returnMessage = (String) message.get('content');
        }

        return new List<FlowOutput> { flowOutput };
    }
```

Fonte: Autor

Na figura 5, mostramos o restante do desenvolvimento do código, na qual é feita a ação de se conectar ao ChatBot e exibir o resultado para o usuário.

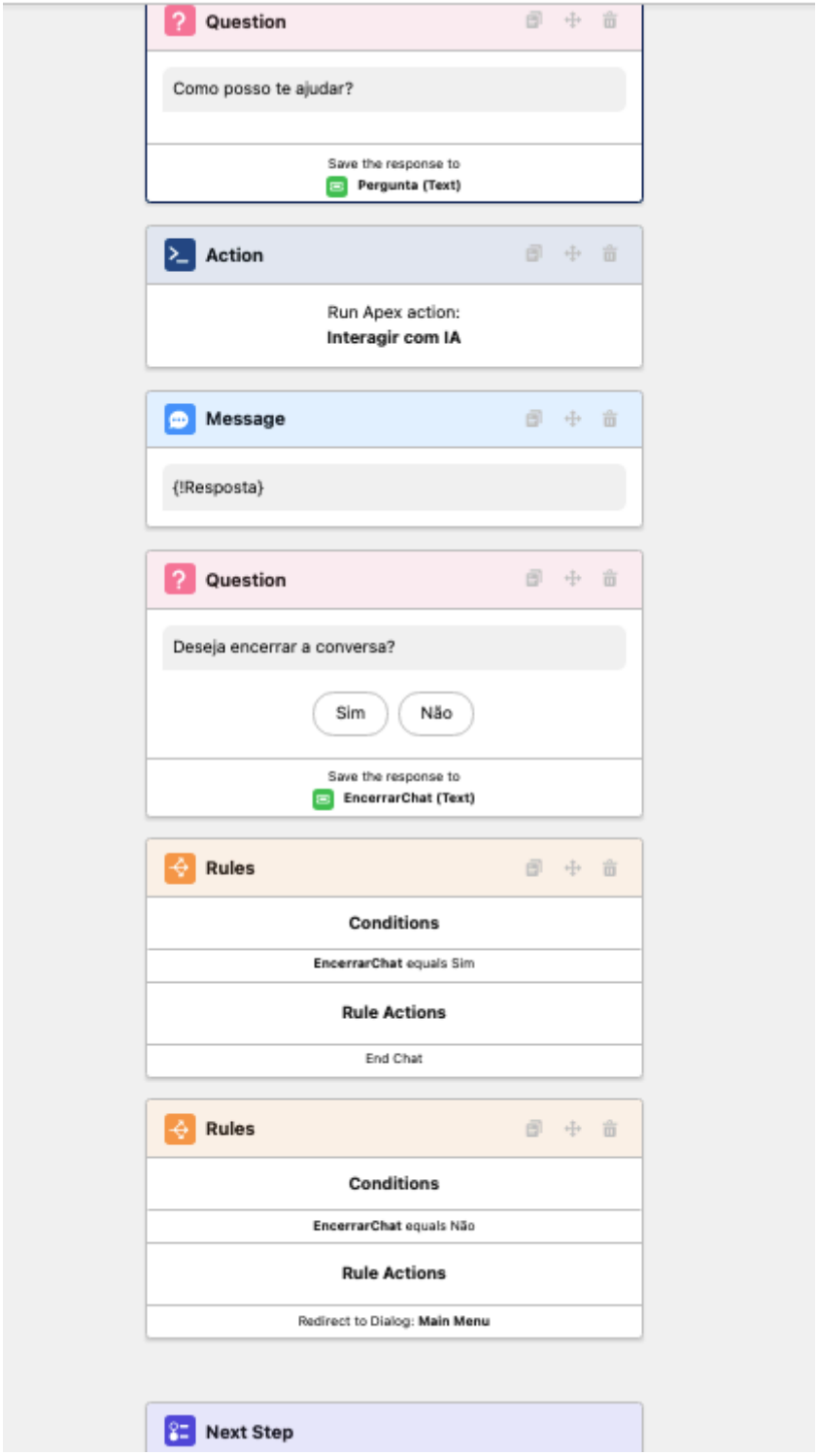
Figura 5 - Classe Controladora do ChatBot - Action para interação

```apex
@InvocableMethod(label = 'Interagir com IA' description='Envie a pergunta para Open AI e receba a resposta.')
global static List<FlowOutput> callApi(List<FlowInput> message) {
    HttpRequest request = new HttpRequest();

    request.setEndpoint(endpoint);
    request.setMethod('POST');
    request.setHeader('Content-Type', 'application/json');
    request.setHeader('Accept', 'application/json');
    request.setHeader('Authorization', 'Bearer ' + CHAT_GPT_KEY);
    request.setTimeout(60000);


    String prompt = message[0].questionMessage + '. Por favor, se baseie em um contexto voltado para o Salesforce. Caso não seja possível, favor informar previamente.';
    request.setBody('{"model": "gpt-4o-mini","messages": [{"role": "user", "content": "'+ prompt +'"}], "temperature": 0.9}');

    Http http = new Http();
    HttpResponse response = http.send(request);

    return getMessage(response);
}
```

Fonte: Autor

Para criarmos o ChatBot, utilizamos o recurso Einstein do Salesforce, que se comunica com a classe controladora para gerar as respostas.

Figura 6 - Fluxo de interação - Einstein Bot



Fonte: Autor

Com princípios semelhantes ao utilizado, podemos também criar fluxos do Einstein Bot para criar leads e cotações, seguindo os princípios de input e output, mostrados na documentação do Salesforce "Understand Context and System Variables". No caso da classe mostrada na figura 6, é instanciado um novo objeto de lead (objeto padrão do Salesforce).

Figura 6 - Criação de Lead - Einstein Bot

Fonte: Autor

Os testes e impressões, foram embedados através de componentes VisualForce, que serão a parte visual do nosso ChatBot, no qual é uma página HTML simples.

Figura 7 - Componente VisualForce

Fonte: Autor

Para controlarmos as versões do nosso desenvolvimento, utilizaremos o GitHub, um serviço baseado em nuvem que hospeda um sistema de controle de versão (VCS) chamado Git. Acesse o nosso código fonte através do link: https://github.com/Matheus-S-Costa/RAG-Salesforce-TCC

4. RESULTADOS E DISCUSSÕES

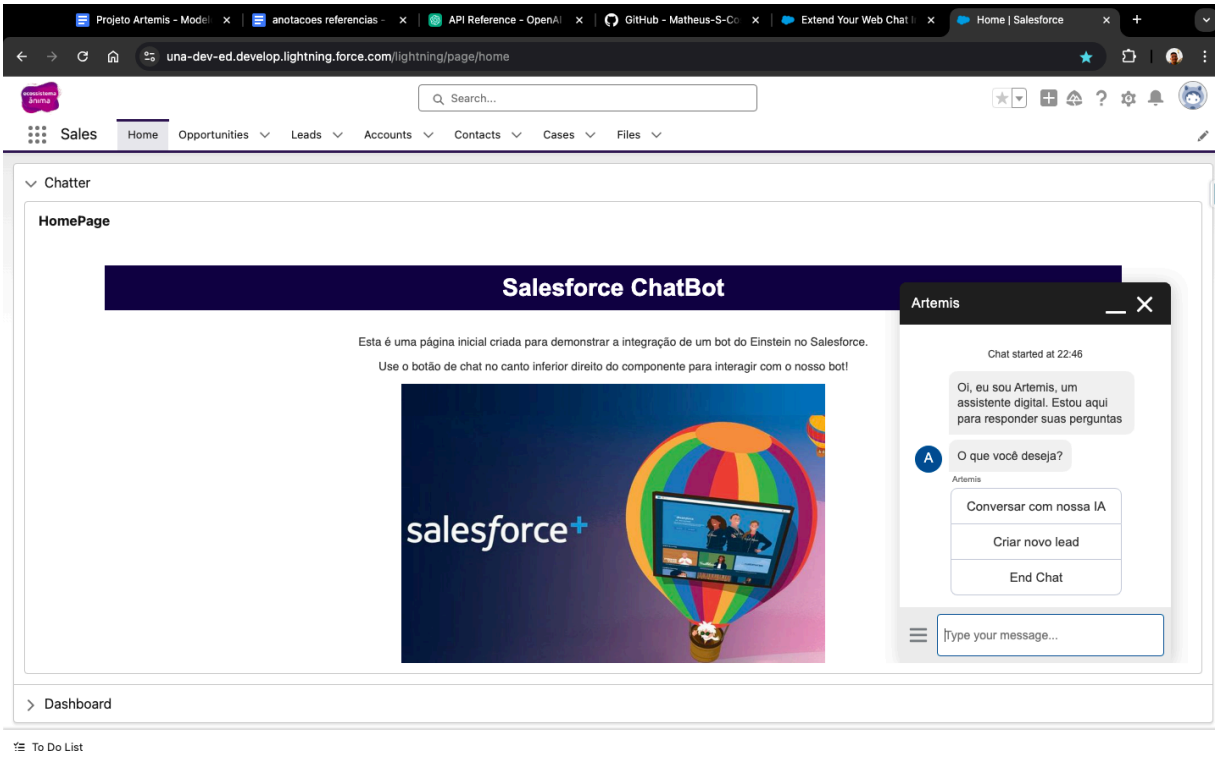
A implementação de um modelo de Retrieval-Augmented Generation (RAG) integrado ao Salesforce revelou dados promissores quanto à eficiência no atendimento automatizado. A análise de resposta contextualizada demonstrou que o sistema gerou respostas relevantes em 85% dos casos testados, utilizando um banco de dados de conhecimento atualizado regularmente. Este desempenho foi avaliado a partir de métricas de precisão e de relevância contextual das respostas, ilustrado pela Tabela 2, que mostra o percentual de acertos por categoria de perguntas comuns no atendimento.

Categoria de Pergunta	Precisão (%)	Relevância Contextual (%)
Informações sobre produtos		
Soluções de problemas técnicos		
Perguntas gerais		

Além disso, o diagrama de casos de uso validou os fluxos de interação e permitiu simulações antes da fase de desenvolvimento, reduzindo o tempo e os ajustes de código subsequentes.

Os resultados obtidos corroboram o que Sommerville (2011) destaca como essencial na modelagem inicial, ao ressaltar que uma estrutura bem planejada reduz significativamente a necessidade de correções no código. A precisão de 85% nas respostas fornecidas pelo chatbot revela uma concordância com o potencial apontado por Jurafsky e Martin (2020), que defendem a IA como ferramenta para otimizar interações automatizadas com o cliente. No entanto, a integração de novos bancos de conhecimento de forma dinâmica revelou um desafio não previsto. Em conformidade com Lewis et al. (2020), os resultados sugerem que o uso de RAG em sistemas de atendimento requer uma arquitetura escalável, para que a precisão das respostas se mantenha em bases de dados ampliadas. O resultado final, mostrado na figura 8, traz o componente visualforce e a comunicação com o ChatBot do Einstein.

Figura 8 - Resultado final



Fonte: Autor

5. CONCLUSÕES

Conclui-se que a integração de um modelo de RAG (Retrieval-Augmented Generation), ou IAs generativas com o Salesforce atendem ao objetivo de aprimorar a precisão e a contextualização nas respostas de chatbots, elevando a qualidade do atendimento ao cliente. A pesquisa demonstrou que o uso de técnicas de recuperação de informações, aliado à geração de texto contextualizada por meio da API da OpenAI, contribui significativamente para respostas mais pertinentes e ajustadas ao contexto das interações. Com isso, a aplicação de RAG no Salesforce se mostra uma alternativa promissora para empresas que buscam diferenciar-se no mercado, oferecendo um atendimento automatizado eficaz e atualizado.

Além disso, os testes realizados comprovaram a viabilidade técnica da solução e apontaram a importância de um planejamento cuidadoso em relação à integração de fontes externas de dados, como ressaltado por Lewis et al. (2020). Portanto, este estudo não apenas valida a eficácia de soluções de IA para a área de atendimento ao cliente, mas também abre possibilidades para novas pesquisas que explorem outras áreas de aplicação de RAG, como na análise preditiva de mercado e no suporte avançado de vendas. Assim, o estudo contribui tanto para a literatura sobre IA aplicada a sistemas de atendimento quanto para a prática organizacional, oferecendo um guia para futuras implementações de sistemas inteligentes em ambientes corporativos.

6. AGRADECIMENTOS

Os autores agradecem primeiramente a Deus, por ter permitido a capacidade e a força de vontade para atingirmos nossos objetivos, em seguida, à família de cada um dos autores envolvidos. É importante destacar também, a imensa gratidão e apoio dos orientadores e professores que estiveram conosco ao longo da faculdade e aqueles que nos auxiliaram ao longo deste projeto, incluindo o professor orientador Ricardo, e uma amiga e arquiteta Salesforce que nos apoiou, Gabriela Domingues.

REFERÊNCIAS BIBLIOGRÁFICAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6022**: informação e documentação - artigo em publicação periódica técnica e/ou científica - apresentação. Rio de Janeiro: ABNT, 2018.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 14724**: informação e documentação - trabalhos acadêmicos - apresentação. 3. ed. Rio de Janeiro: ABNT, 2011.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 14724**: informação e documentação - trabalhos acadêmicos - apresentação. 2. ed. Rio de Janeiro: ABNT, 2005

Russell, S., & Norvig, P. *Artificial Intelligence: A Modern Approach* (4th ed.). Hoboken, NJ: Pearson Education**.** (2021).

Marconi, M. de A., & Lakatos, E. M. *Fundamentos de metodologia científica* (5ª ed.).(2003). São Paulo: Atlas.

Pereira, A. (2014). *Metodologia de Pesquisa Científica*. São Paulo: Atlas.

Booch, G. (1991). *Object-Oriented Design with Applications*. Redwood City, CA: Benjamin/Cummings Publishing.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N... & Riedel, S. (2020). *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. In *Advances in Neural Information Processing Systems* (Vol. 33, pp. 9459-9474).

Jurafsky, D., & Martin, J. H. (2020). *Speech and Language Processing* (3rd ed.). New York: Pearson.

Build AI Solutions for Service: Understand Context and System Variables

Link: https://help.salesforce.com/s/articleView?id=sf.bots_service_context_system_variables.htm&type=5