Proposta de Arquitetura para o Chatbot da Dra. Jô

# 1. Objetivo e Escopo

O objetivo do chatbot é automatizar o atendimento ao cliente da Dra. Jô, permitindo que os usuários realizem agendamentos, tirem dúvidas frequentes e recebam suporte em tempo real. Isso reduz a carga de trabalho dos atendentes e melhora a experiência do usuário, fornecendo respostas rápidas e precisas.

# 2. Escolha do Chatbot: IBM Watson Assistant

A solução proposta utiliza o IBM Watson Assistant para criar um chatbot avançado, capaz de interpretar a linguagem natural dos usuários, lidar com interações complexas e fornecer respostas adequadas. A ferramenta de IA e NLP (Processamento de Linguagem Natural) do Watson Assistant é a principal responsável pelo processamento das mensagens dos usuários.

# 3. Camadas da Arquitetura

## 3.1. Interface de Usuário (Front-end)

Plataformas: O chatbot será acessível pelo site da Dra. Jô através de um widget de chat, bem como por plataformas de mensagens como WhatsApp e Facebook Messenger.

Função: A interface permite que os usuários enviem perguntas ao chatbot e recebam respostas em tempo real, facilitando a interação em um ambiente familiar.

Tecnologia: O Watson Assistant possui integrações nativas com essas plataformas, facilitando a implementação.

## 3.2. Motor de Processamento de Linguagem Natural (NLP) - IBM Watson Assistant

Função: Interpretar as mensagens dos usuários, identificar intenções e fornecer respostas baseadas em dados de treinamento.

Treinamento: São usados exemplos reais e sintéticos para treinar o chatbot a lidar com agendamentos, suporte técnico, e perguntas frequentes. O chatbot melhora conforme interage com mais usuários.

Ajustes contínuos: O modelo de IA é ajustado periodicamente para melhorar a precisão das respostas e garantir uma experiência de uso eficiente.

## 3.3. Back-end (Processamento e Integração com Sistemas)

Lógica de Negócio: O back-end é responsável pela integração com sistemas internos (ex.: agendamentos, CRM) e processamento das respostas.

Tecnologia: O back-end será implementado com Python utilizando AWS Lambda para executar funções serverless, garantindo eficiência e escalabilidade. As funções tratam a lógica de negócios e integram o chatbot com os sistemas.

Banco de Dados: O MongoDB será usado como banco de dados para armazenar informações sobre os usuários e histórico de interações.

Função: Processar as mensagens e garantir que o chatbot responda com informações em tempo real, integrando-se a serviços de agendamento e CRM via API REST.

## 3.4. Infraestrutura em Nuvem - AWS

Hospedagem: Toda a solução será hospedada na AWS Cloud, utilizando o API Gateway para expor as APIs REST e AWS Lambda para rodar o back-end.

Armazenamento Opcional: A solução pode usar o AWS S3 para armazenar conteúdos estáticos, como documentos ou respostas frequentes.

Segurança e Escalabilidade: O Watson Assistant e o MongoDB se comunicam com segurança através de API REST, garantindo que os dados sejam mantidos seguros e o sistema escalável.

# 4. Descrição dos Elementos da Arquitetura

Interface de Usuário: Conectada ao Watson Assistant por meio de APIs, seja em um site, WhatsApp, ou Messenger.

Motor de NLP: O Watson Assistant processa as entradas do usuário, entendendo intenções e respondendo com base em dados de IA treinados.

Back-end: Utilizando AWS Lambda para funções serverless, o back-end integra os dados do chatbot com os sistemas internos.

Banco de Dados: O MongoDB armazena dados dos usuários e interações.

Infraestrutura em Nuvem: A solução usa o AWS API Gateway para comunicação e AWS S3 para armazenamento opcional de dados estáticos, com escalabilidade gerenciada pelo AWS Lambda.

# 5. Previsão de Custos

| Componente | Custo Mensal |  
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|  
| IBM Watson Assistant | Plano gratuito disponível para até 10.000 mensagens mensais. Após esse limite, o custo é estimado em $0,0025 por mensagem. |  
| AWS Lambda (Aplicação Python) | O plano gratuito do AWS Lambda oferece até 1 milhão de requisições mensais. Custos adicionais dependem do número de requisições e tempo de execução. |  
| MongoDB (Banco de Dados) | A camada gratuita do MongoDB cobre até 512 MB de armazenamento. Acima disso, o custo é baseado no armazenamento e uso. |  
| API Gateway (AWS) | O API Gateway possui um plano gratuito com 1 milhão de chamadas HTTP/mês. Custos adicionais são calculados com base no número de chamadas. |  
| AWS S3 (Armazenamento) | Disponível gratuitamente para até 5 GB de armazenamento. Custos adicionais são baseados no volume de dados armazenados e acessos. |

# 6. Resultados Esperados

Melhorias na Empresa:  
- Automatização de Processos: O chatbot automatiza o atendimento e agendamentos, reduzindo a carga de trabalho humano e permitindo respostas 24/7.  
- Escalabilidade: A arquitetura suporta o crescimento da empresa, permitindo o aumento da demanda sem degradação de desempenho.  
- Satisfação do Cliente: Respostas rápidas e consistentes elevam a satisfação e fidelização dos clientes.

Valores Agregados ao Modelo de Negócio:  
- Eficiência Operacional: Com o chatbot lidando com perguntas frequentes, a equipe pode focar em atividades mais complexas.  
- Aumento de Receitas: Com mais facilidade para agendamentos e suporte rápido, a empresa deve converter mais leads e aumentar a retenção de clientes.  
- Melhoria no Atendimento: O chatbot coleta dados valiosos, que podem ser usados para ajustar os produtos e serviços.

Possíveis Desafios:  
- Necessidade de Ajustes Futuros: Inicialmente, podem ocorrer erros nas respostas, exigindo ajustes contínuos no modelo de IA.  
- Custo de Escalabilidade: Aumentar a capacidade pode trazer custos extras, especialmente em momentos de pico, exigindo uma boa gestão financeira.