Enterprise Challenge - Sprint 1

[FIAP] Graduação - 1TIAOR - 2024/2 – Grupo 33

Matheus Augusto Rodrigues Maia - RM560683
Alex da Silva Lima - RM559784
Johnatan Sousa Macedo Loriano - RM559546
Bruno Henrique Nielsen Conter - RM560518
Fabio Santos Cardoso - RM560479

Objetivo

Detalhar a escolha do projeto de chatbot para o Enterprise Challenge da FIAP, bem como arquitetura de solução, recursos previstos e eventuais custos.

Escopo

Desenvolvimento de chatbot para interação com cliente da empresa via canal digital. É esperado que ele apoie o cliente em dúvidas dicas sobre processos da empresa, aprimore o atendimento e reduza o fluxo de atendimento, consequentemente.

Premissas da arquitetura

A arquitetura está focada em um modelo isonômico e agnóstico em termos de recursos, visando entregar o mesmo nível de qualidade, capacidade e operacionalidade, independente do local onde será provisionado, seja na nuvem, seja localmente.

Além da independência de ambiente, a arquitetura procurar abstrair a complexidade de fluxos multicanais (utilizando os mesmos componentes de diferentes formas), promover escala sob demanda e distribuir cargas de trabalho de forma eficiente e igualitárias.

Tipo de Chatbot

Optamos por utilizar o framework LangChain pela flexibilidade de mudança, tanto no escopo como no modelo de apoio, sendo possível anexar em um modelo LLM público ou, eventualmente, em um modelo LLM privado.

O framework possui como grande diferencial sua capacidade de integrar modelos de linguagem com fluxos de dados e ferramentas externas, proporcionando uma experiência de conversa mais dinâmica e sofisticada. Ao invés de seguir a abordagem tradicional baseada em regras fixas e respostas predefinidas, o LangChain permite criar chatbots baseados em IA e Processamento de Linguagem Natural (NLP), que conseguem compreender e interagir de maneira mais fluida e natural com os usuários.

Um dos principais recursos do LangChain é a sua flexibilidade, que permite trabalhar com diversas fontes de dados, APIs externas e fluxos de trabalho, o que possibilita o aprimoramento contínuo do chatbot.

Por ser um recurso que se adapta à diversos modelos, torna-se também uma ferramenta boa para um MVP, dado que não precisa de treino e torna mais clara e objetiva a implementação, dispensando fluxos de dados complexos.

Resumo de Operação

O fluxo considera acessar um repositório central de dados através de um endpoints REST local chamados de “Agentes de Consulta”. Este repositório será atualizado frequentemente (inicialmente sob demanda) por outros recursos chamado “Agente de Coleta”. Ambos trabalharão para manter o repositório atualizado e a solução funcional.

A instância do LangChain, quando consultada, realizará uma contextualização da pergunta utilizando a fonte interna e então realizará uma consulta no modelo LLM anexado na solução.

Tecnologias Envolvidas

Optamos por utilizar recursos open source com grande adoção da comunidade, apoio de players de mercado relevantes e possibilidade de execução gerenciada, a depender do provedor escolhido.

Kubernetes

Escolhido para suportar a solução, o Kubernetes foi criado originalmente pela Google e desenvolvido por uma equipe interna da empresa como parte de sua infraestrutura de gerenciamento de containers. A ferramenta hoje é referência em ambientes de computação elástica e ambientes com arquiteturas complexas e flexíveis.

Envoy Proxy

Escolhido para distribuir as requisições e cargas de trabalho entre as diversas instâncias do LangChain como API, O Envoy Proxy foi criado pela Lyft como uma solução de proxy de alta performance e open-source para microsserviços. Seu grande propósito é fornecer uma camada de comunicação entre serviços que facilita a observabilidade, segurança e roteamento dinâmico em arquiteturas distribuídas.

Dapr (Distributed Application Runtime)

Tido como um recurso chave para a viabilidade da arquitetura, foi criado pela Microsoft e lançado em 2019. Seu grande propósito é simplificar o desenvolvimento de aplicativos distribuídos, fornecendo uma série de abstrações e APIs para lidar com desafios comuns, como comunicação entre serviços, gerenciamento de estado, pub/sub, segurança e escala.

Possui autoscaler compatível com o Kubernetes que permite se utilizar de diversos critérios para escala de recursos de computação.

É uma ótima ferramenta porque permite utilizá-la com propósitos diferentes e com linguagens diferentes, quando necessário. Um exemplo prático está na própria arquitetura, onde será utilizado tanto como API, quanto como agente de alimentação do repositório de dados central, fornecendo recursos de gatilhos e busca de dados, além de permitir criar qualquer tipo de aplicação.

Redis

Ferramenta escolhida como repositório de dados central, seu grande propósito é atuar como um armazém de dados em memória de alto desempenho, que oferece suporte a diferentes tipos de dados, como strings, listas, sets e hashes. O Redis é amplamente utilizado como cache e banco de dados para melhorar a performance de sistemas, proporcionando acesso ultrarrápido a dados frequentemente acessados, e como fila de mensagens para gerenciamento de tarefas assíncronas em arquiteturas distribuídas. É conhecido por sua velocidade, simplicidade e escalabilidade.

O LangChain possui recursos para trabalhar diretamente com o Redis, inclusive fornecendo métodos específicos para ler e gravar dados especiais.

Web App

A Solução vista como principal entrega uma API, mas um web app é considerado também, neste caso para interação simples, pois não acreditamos quer será colocada à disposição do cliente final, sendo utilizada para testes internos.

Custos estimados

Para o MVP no Kubernetes estão sendo considerados nós de baixa prioridade (SPOT) nos agentes de consulta, utilizando fluxos de chamada stateful, além da abstração criada pelo Envoy Proxy.

Justamente pela escolha de recursos de computação baixo custo, espera-se uma replicação de nós acentuada visando impactar o mínimo possível a disponibilidade do serviço.

A estimativa média é de que nós de baixa prioridade comuns (Pesquisado no Microsoft Azure) custem em média 15% do valor total de um recurso comum, embora apresente riscos de disponibilidade que pode impactar a aplicação como um todo, quando não dimensionado corretamente.

Estimativa (Microsoft Azure- East Us 2, 13/11/2024)

Recurso	CPU	RAM	SPOT	Quantidade	Valor (Mensal) R$
Kubernetes	4	16	Não	1	861,54
Dapr - Agente de Coleta	2	8	Sim	4	258,312
Dapr - Agende Consulta	2	8	Sim	4	258,312
Redis	4	16	Não	1	861,54
Envoy Proxy	2	8	Sim	4	430,53
Web APP	2	8	Sim	1	64,578

Total Estimado: R$ 2.734,81

Resultado Esperado

Espera-se que o chatbot seja um direcionador de clientes, reduza o tempo de navegação deles nos canais digitais da empresa e aumente a experiencia final do cliente com a empresa.

Desenho da arquitetura