

UFG

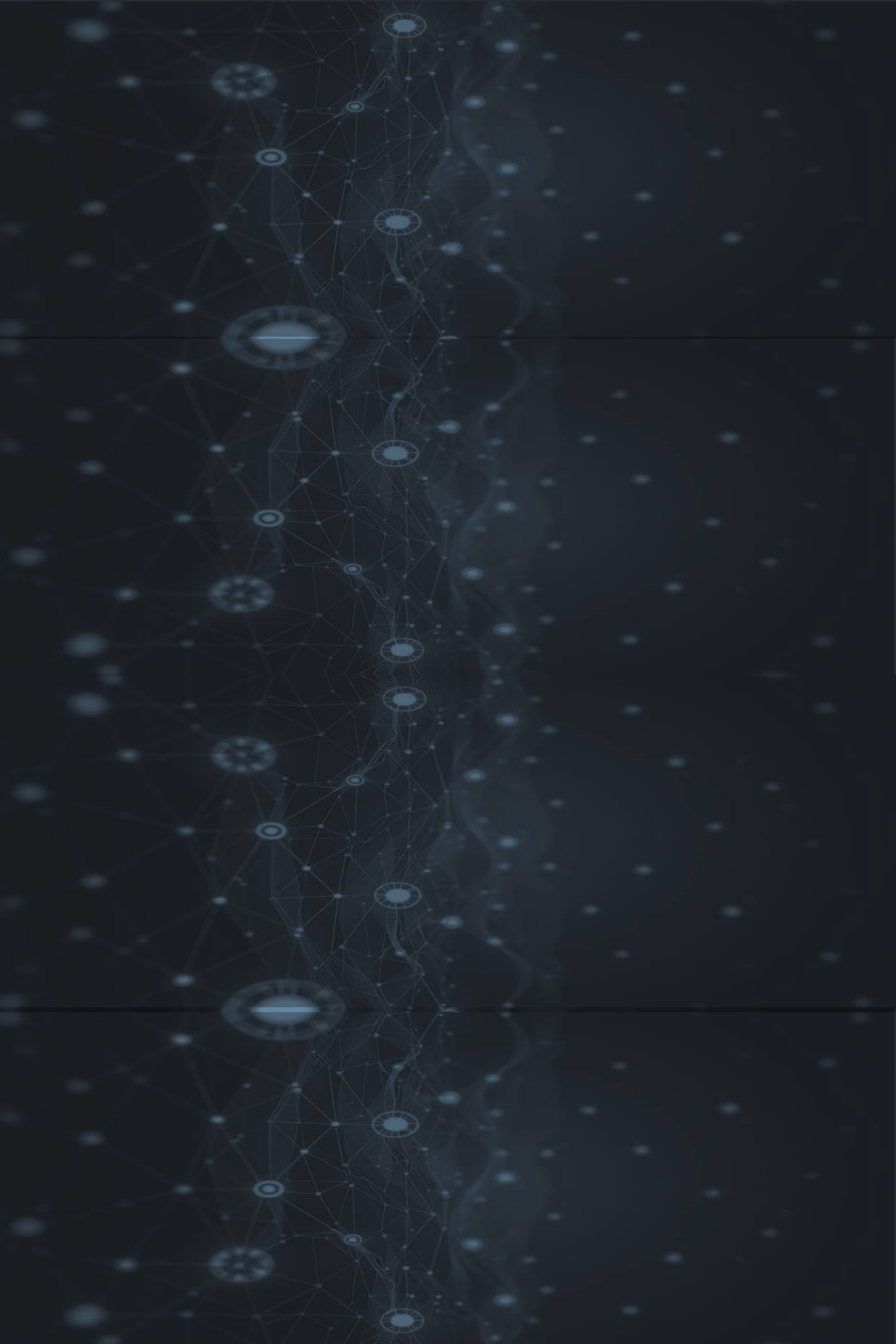
**Pós-Graduação em
Sistemas e Agentes
Inteligentes**

Disciplina

Engenharia de Prompt

Professor Sanderson Macedo

- **Marcu Loreto**
- **Leandro Lima**
- **Ricardo Kerr**



Assistente Virtual de Suporte Técnico com OpenAI e Guardrails

Esta apresentação demonstra como integrar um modelo de linguagem (LLM) da OpenAI a um assistente de suporte técnico, utilizando guardrails para validação.

Resumo e Objetivo

Objetivo

Implementar um assistente virtual de suporte técnico que utiliza a OpenAI para fornecer respostas consistentes e personalizadas aos usuários.

Funcionalidade

O sistema recebe perguntas do usuário, pesquisa informações relevantes e fornece respostas que incluem o slogan "Resetou ta Novo".

Guardrails

O código utiliza guardrails para garantir que as respostas da LLM estejam de acordo com as diretrizes e incluam o slogan definido.

Benefícios

O sistema oferece suporte técnico eficiente, automatizado e consistente, melhorando a experiência do usuário.



Configuração do Ambiente

Carregamento de Variáveis

O código utiliza a biblioteca `dotenv` para carregar variáveis de ambiente, incluindo a chave da API da OpenAI (`OPENAI_API_KEY`) armazenada em um arquivo `.env`.

Validação da Chave de API

Antes de iniciar o sistema, o código verifica se a chave foi encontrada, levantando um erro caso contrário. Isso garante que a comunicação com a API da OpenAI seja estabelecida corretamente.

Instanciação do Cliente OpenAI

Após validar a chave, um cliente OpenAI (objeto `client`) é criado, permitindo enviar requisições ao modelo de linguagem.

Retail	9	Tag Rite	Petty, Cavigert	Wegri: 120035	Wegri: 120035	Wegri: 120035
Retail	3	Tag Rite	Petty, Cavigert	Wegri: 120035	Wegri: 120035	Wegri: 120035
Retail	3	Tag Rite	Petty, Cavigert	Wegri: 120035	Wegri: 120035	Wegri: 120035
Retail	9	Tag Rite	Petty, Cavigert	Wegri: 120035	Wegri: 120035	Wegri: 120035
Retail	6	Tag Rite	Petty, Cavigert	Wegri: 120035	Wegri: 120035	Wegri: 120035
Special	0	Tag Rite	Petty, Cavigert	Wegri: 120035	Wegri: 120035	Wegri: 120035
Calad	0	Tag Rite	Petty, Cavigert	Wegri: 120035	Wegri: 120035	Wegri: 120035
Barst	9	Tag Rite	Petty, Cavigert	Wegri: 120035	Wegri: 120035	Wegri: 120035
Comm	8	Tag Rite	Petty, Cavigert	Wegri: 120035	Wegri: 120035	Wegri: 120035
Comm	0	Tag Rite	Petty, Cavigert	Wegri: 120035	Wegri: 120035	Wegri: 120035
Comm	0	Tag Rite	Petty, Cavigert	Wegri: 120035	Wegri: 120035	Wegri: 120035
Dalt	0	Tag Rite	Petty, Cavigert	Wegri: 120035	Wegri: 120035	Wegri: 120035
Gall	0	Tag Rite	Petty, Cavigert	Wegri: 120035	Wegri: 120035	Wegri: 120035
Wolain	0	Tag Rite	Petty, Cavigert	Wegri: 120035	Wegri: 120035	Wegri: 120035
Gall	0	Tag Rite	Petty, Cavigert	Wegri: 120035	Wegri: 120035	Wegri: 120035

Lógica de Negócio: Busca de Documentos

Função search_documents(query)

Esta função simula a consulta a uma base de dados local, recebendo uma string (consulta do usuário) e retornando informações relevantes com base em um conjunto de dados pré-definido.

Exemplos de Chaves

As chaves de pesquisa correspondem a problemas comuns, como "máquina não liga" ou "máquina travando", retornando soluções correspondentes.

Benefícios

Esse mecanismo permite contextualizar as respostas do assistente, fornecendo informações específicas para problemas comuns, sem necessidade de reescrever as soluções no código principal.

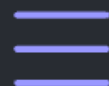


Prompt Inicial e Regras de Atendimento



Contextualização

O prompt inicial define o assistente como um especialista em suporte técnico, definindo claramente seu papel e responsabilidades.



Instruções Detalhadas

O prompt lista objetivos específicos (identificar problemas, coletar dados do cliente, incluir slogan), fornecendo diretrizes claras.



Tom de Voz

O prompt define que o atendimento deve ser claro, educado e objetivo, garantindo respostas consistentes e alinhadas aos valores da assistência.

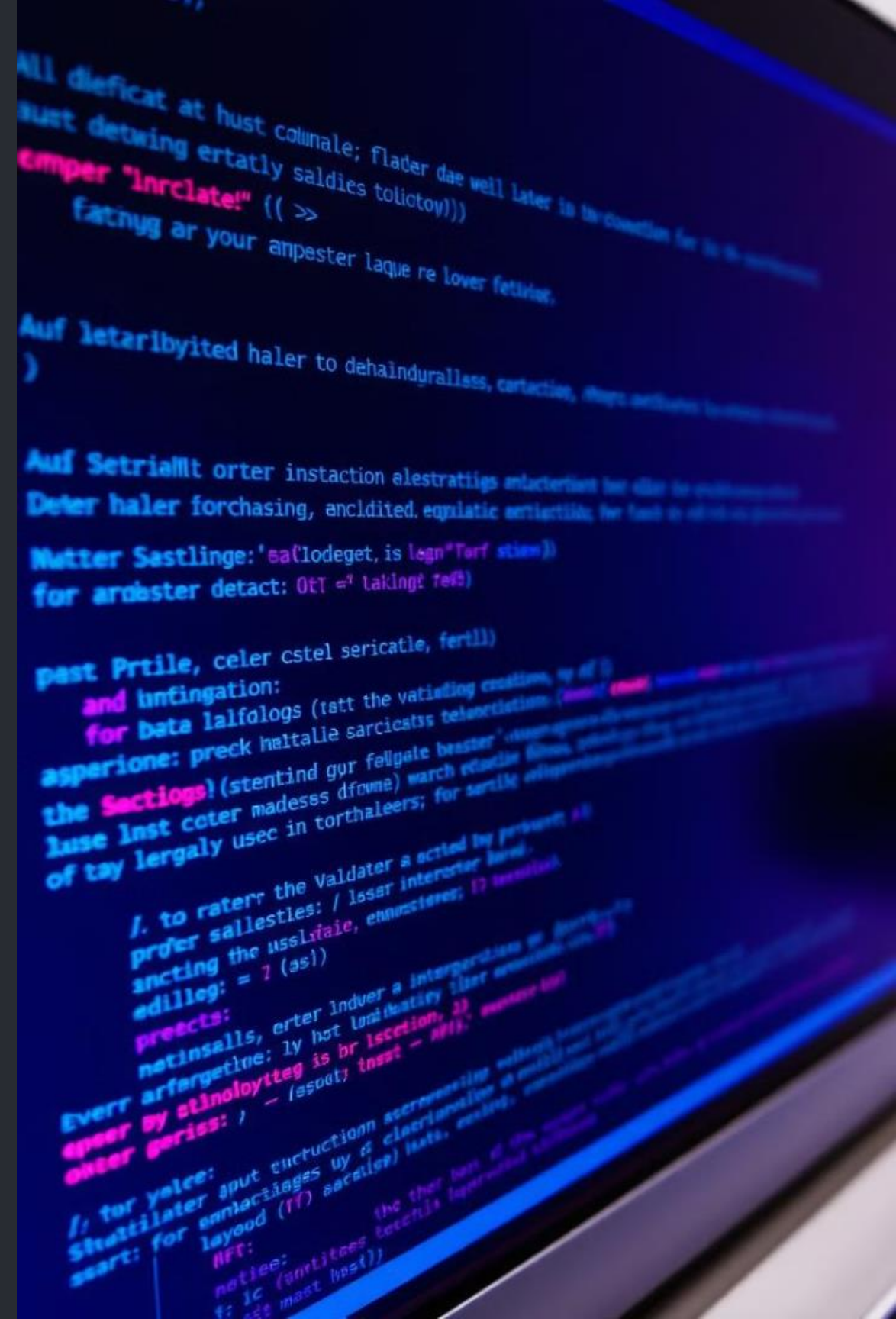


Memória Contextual

A exigência de incluir o slogan "Resetou ta Novo" e o endereço da assistência técnica é uma técnica de prompting que garante a fidelidade às diretrizes em cada resposta.

Validação da Entrada do Usuário

- 1 O código verifica se o usuário digitou uma mensagem não vazia.
- 2 Caso o **user_input** seja apenas espaços ou vazio, solicita que o usuário insira algo válido.
- 3 Garante a interação mínima com o sistema, evitando chamadas desnecessárias à API.
- 4 Melhora a experiência do usuário ao prevenir respostas irrelevantes do modelo devido a entradas inválidas.



Interação com a LLM e Guardrails

Chamadas à API OpenAI

O código utiliza `client.chat.completions.create` para gerar respostas com base no histórico de mensagens.

Validação com Guardrails

○ **Guard()** com o validador **ContainsString(substring="Resetou ta Novo")** garante que a resposta da LLM inclua o slogan definido.

Benefício da Validação

Isso garante conformidade no conteúdo, evitando que a LLM esqueça o slogan ou traga respostas em desacordo com o prompt inicial.

Demonstração do Fluxo de Execução

1

Entrada do Usuário

O usuário digita uma mensagem descrevendo seu problema.

2

Validação

O código valida se a entrada é válida (não vazia).

3

Busca de Informações

Busca informações relevantes na base simulada, adicionando-as ao contexto se necessário.

4

Resposta da LLM

O sistema envia o histórico para a LLM via API, gera uma resposta e a valida com Guardrails.



Conclusão

O código demonstra como integrar um modelo de linguagem (LLM) em um cenário de atendimento técnico.

As técnicas de prompting e o uso de validações com Guardrails asseguram a qualidade, relevância e consistência das respostas.

A abordagem garante que informações essenciais (como slogan) sejam sempre incluídas, mantendo a identidade e profissionalismo do suporte.

[https://github.com/Marcu-Loreto/Agentes Inteligentes](https://github.com/Marcu-Loreto/Agentes_Inteligentes)