

	Exercise 02
Criar um chatbot simples para atendimento ao cliente de uma pizzeria utilizando Python e a API do Google Gemini.	
Diretório de entrega: <i>ex02/</i>	
Arquivos a serem entregues: pizzabot.py , requirements.txt	
Funções ou bibliotecas permitidas: google-generativeai , os	

- Instruções:

1. Configuração do Google AI Studio:

- Acesse <https://aistudio.google.com> e faça login na sua conta Google.
- Familiarize-se com a interface do Google AI Studio.

2. Criação do chatbot no Google AI Studio:

- No Google AI Studio, crie um novo chat.
- Copie e cole o seguinte system prompt no campo apropriado:

```
Voce e o OrderBot, um servico automatizado para coletar pedidos de um restaurante de pizza.
Primeiro, voce cumprimenta o cliente, depois coleta o pedido e pergunta se e para retirada ou entrega.
Voce espera para coletar todo o pedido, depois resume e verifica uma ultima vez se o cliente deseja adicionar algo mais.
Se for uma entrega, voce pede o endereco.
Finalmente, voce coleta o pagamento.
Certifique-se de esclarecer todas as opcoes, extras e tamanhos para identificar exclusivamente o item do menu.
Voce responde de maneira curta e amigavel.
O menu inclui:
- Pizza de pepperoni R$62,56, R$48,00, R$33,60
- Pizza de queijo R$52,56, R$44,40, R$31,20
- Pizza de berinjela R$57,36, R$46,80, R$32,40
- Batatas fritas R$21,60, R$16,80
- Salada grega R$34,80
Coberturas:
- Queijo extra R$9,60
- Cogumelos R$7,20
- Linguica R$14,40
- Bacon canadense R$16,80
- Molho AI R$7,20
- Pimentoes R$4,80
Bebidas:
- Coca-cola R$14,40, R$9,60, R$4,80
- Sprite R$14,40, R$9,60, R$4,80
- Agua engarrafada R$24,00
```

- Configure a temperatura para 0.
- Teste o chatbot no Google AI Studio com algumas interações para ver como ele responde.

3. Implementação em Python (pizzabot.py):

- Obtenha uma chave de API para o Google Gemini.

- Use o código fornecido em "get code" como base para configurar o modelo Gemini em Python.
- Use o modelo "gemini-1.5-flash" do Google Gemini.
- Adicione tratamento de erros básico para lidar com problemas de conexão ou respostas inesperadas da API.
- Implemente um loop de conversação que permita múltiplas interações com o usuário.
 - * Solicite input do usuário.
 - * Envie o input para o modelo.
 - * Imprima a resposta do modelo.
 - * Continue o loop até que o usuário decida sair.
- Implemente uma função para salvar o histórico da conversa em um arquivo de texto.

- Exemplo de loop de conversação:

```
while True:
    user_input = input("You: ")
    if user_input.lower() == 'sair':
        print("Bot: Thank you for using our service. Goodbye!")
        break
    try:
        response = chat.send_message(user_input)
        print("Bot:", response.text)
    except Exception as e:
        print(f"Error: {e}")
```

- Teste e Refinamento:

- Teste seu chatbot com diferentes cenários de pedidos.
- Refine as respostas ajustando o system prompt se necessário.

- Exemplo de uso:

```
$ python pizzabot.py
Bot: Ola! Bem-vindo a nossa pizzeria. O que voce gostaria de pedir hoje?
Voce: Eu quero uma pizza de pepperoni grande.
Bot: Otima escolha! Uma pizza de pepperoni grande custa R$62,56. Gostaria de adicionar alguma cobertura extra ou bebida?
Voce: Sim, quero adicionar queijo extra e uma Coca-Cola media.
Bot: Perfeito! Adicionei queijo extra (R$9,60) e uma Coca-Cola media (R$9,60) ao seu pedido. Algo mais?
Voce: Nao, e so isso.
Bot: Certo! Seu pedido e:
1x Pizza de pepperoni grande: R$62,56
1x Queijo extra: R$9,60
1x Coca-Cola media: R$9,60
Total: R$81,76
E para entrega ou retirada?
```

- Dicas:

- Use try/except para lidar com possíveis erros da API.

- Considere adicionar uma opção para o usuário reiniciar o pedido ou sair do chat.