

# Criando API'S .NET utilizando apenas lógica de api's de I.A.

---

Márcio Rogério Nizzola



# Márcio R. Nizzola

---

Arquiteto de Software na CI&T

Professor na Etec de Itu

Fundador do Meetup Itu Developers

Estudando programação a partir de 1989

Vivendo de software desde 1992

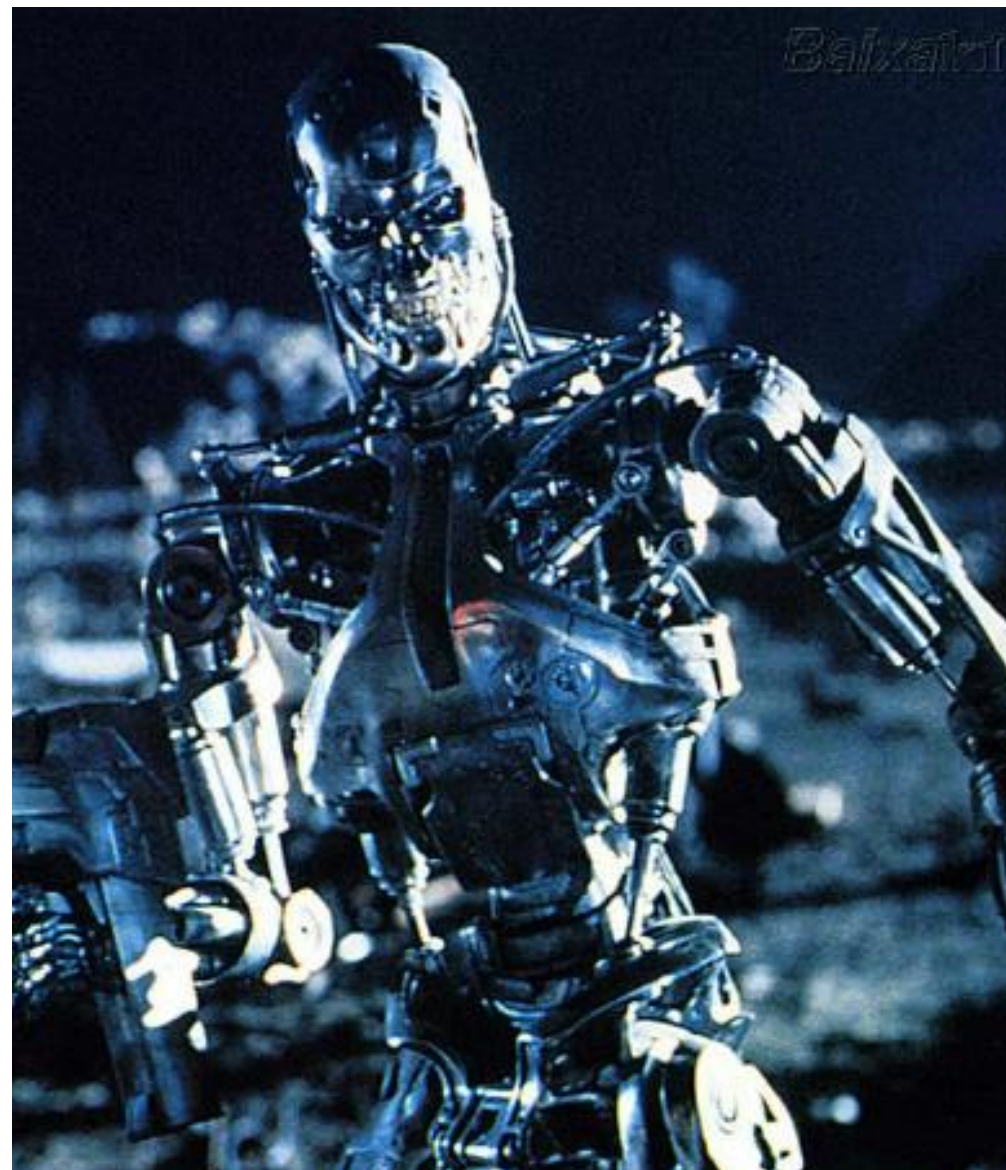
Principais tecnologias: C#, Php, Python, Javascript,  
Angular – (antigas: Clipper, C++, Delphi, Visual Basic)

[www.linkedin.com/in/nizzola](http://www.linkedin.com/in/nizzola)

# TIPOS DE INTELIGÊNCIA ARTIFICIAL

---

- IA Estreita (ou IA Fraca): Também conhecida como Inteligência Artificial Específica, refere-se a sistemas projetados para realizar tarefas específicas e limitadas dentro de um domínio predefinido. Esses sistemas são altamente especializados e não possuem capacidade de generalização. Exemplos incluem chatbots, reconhecimento de fala e sistemas de recomendação.
- IA Geral (ou IA Forte): Também conhecida como Inteligência Artificial Geral ou Superinteligência, é um nível mais avançado de IA que visa reproduzir a inteligência geral humana. Esses sistemas seriam capazes de realizar qualquer tarefa intelectual que um ser humano pudesse fazer, incluindo raciocínio abstrato, aprendizado de novos conceitos e adaptação a diferentes cenários.

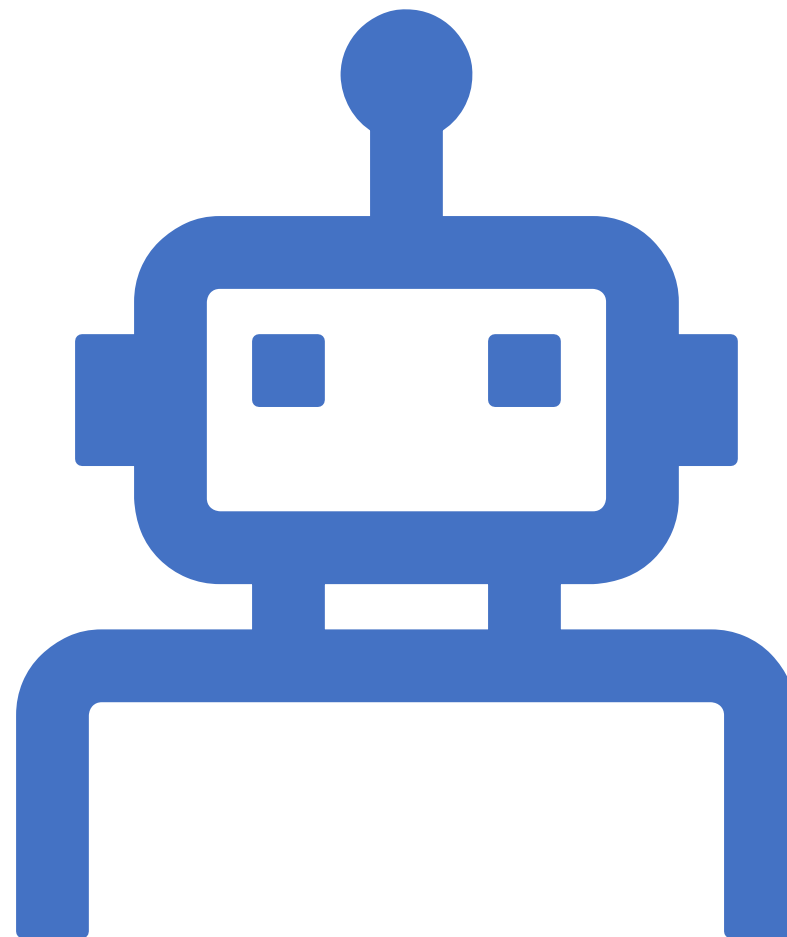


# O que é uma I.A. Generativa

---

É uma tecnologia com capacidade de aprender a padrões complexos de comportamento a partir de uma base de dados, é do tipo “Estreita”

Através de uma técnica chamada aprendizagem de máquina ("machine learning" em inglês), o IAs generativas como ChatGPT e DALL-E conseguem reproduzir conteúdos após receber treinamento.

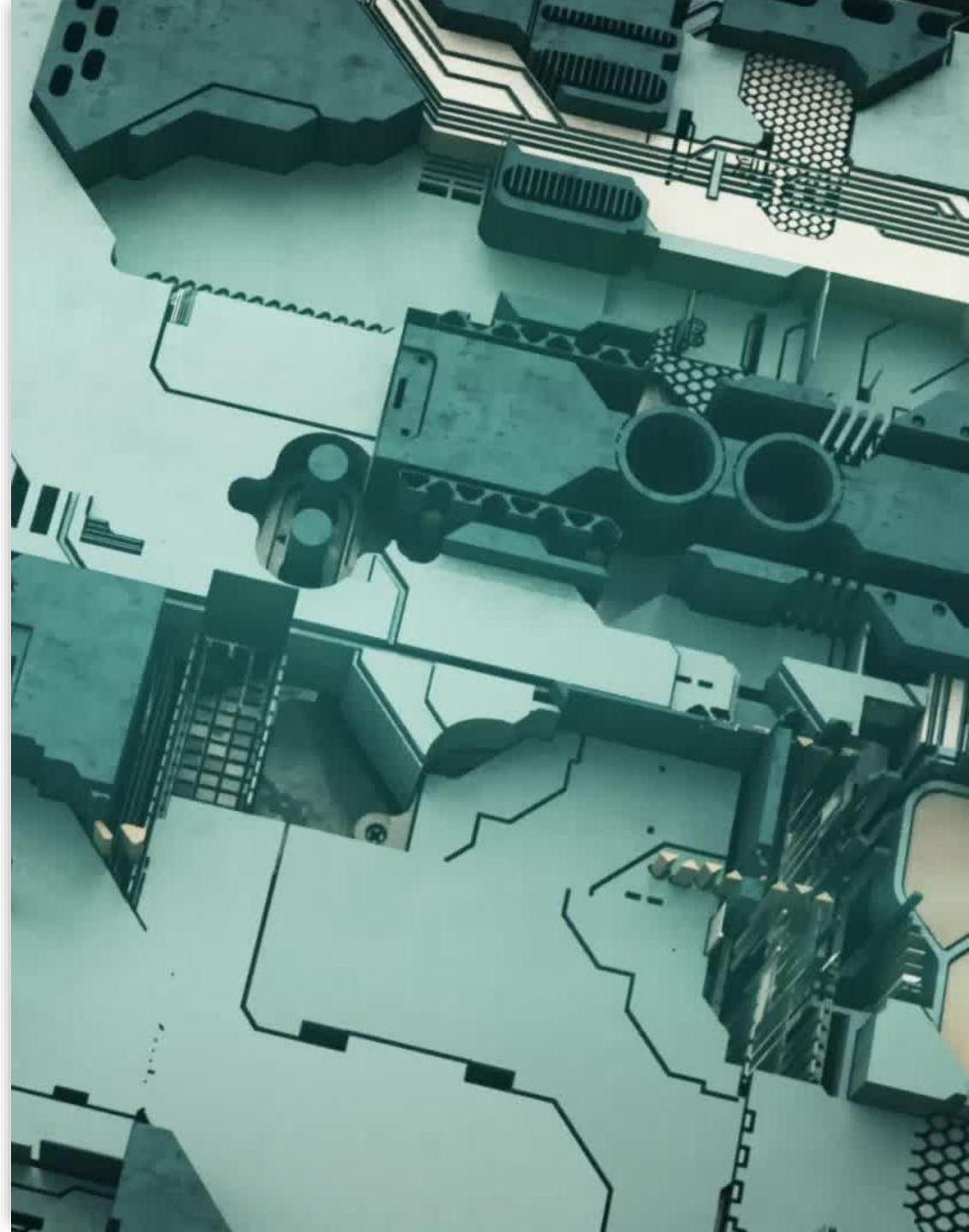




# Engenharia de prompt, o que é isso?

Termo que se refere ao processo de criação e ajuste de prompts (ou instruções) que são dados a um modelo de linguagem baseado em Inteligência Artificial, como GPT-3 ou GPT-4.

- **Chatbots e assistentes virtuais:** Ajudar a otimizar a interação entre o usuário e a IA.
- **Geração de conteúdo:** Criar prompts que incentivem a IA a gerar conteúdo de alta qualidade, como redações, scripts de vídeo, postagens de mídia social, código de programação, etc.
- **Análise de sentimentos:** Projetar prompts que ajudem a IA a analisar corretamente o sentimento em um texto.
- **Tradução de idiomas:** Criar prompts que auxiliem a IA a traduzir corretamente entre idiomas.
- **Resposta a perguntas:** Formular prompts que ajudem a IA a fornecer respostas precisas e úteis para perguntas específicas.

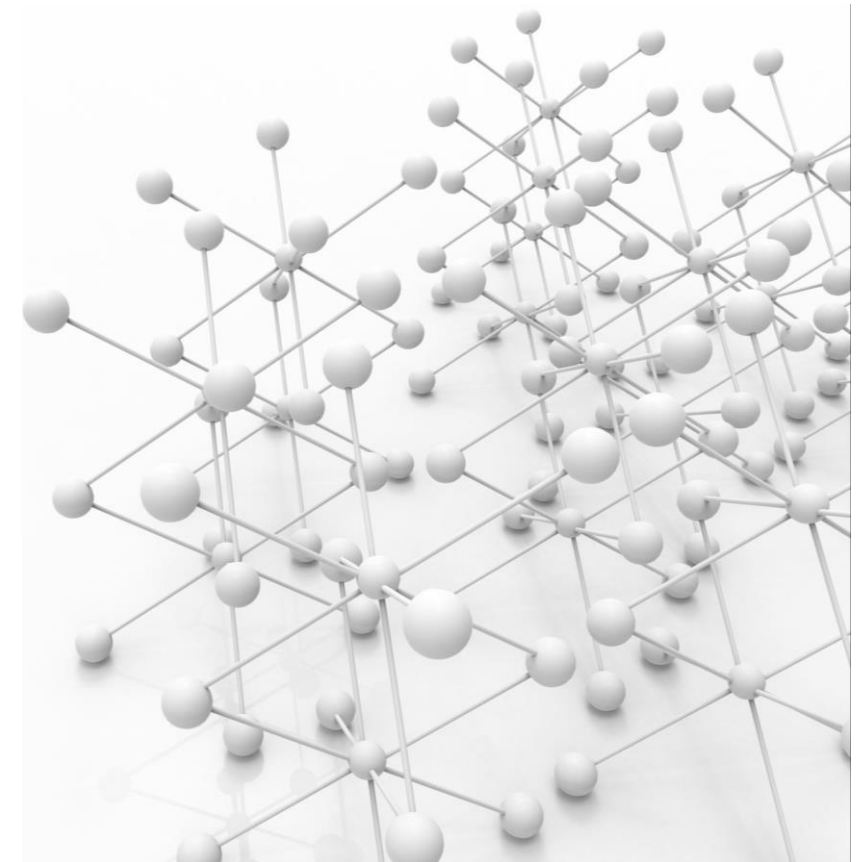


# Como funciona a comunicação - LLM

---

## Modelo LLM

- A tecnologia LLM (large language model) é uma técnica de inteligência artificial que foi desenvolvida a partir de uma série de avanços em áreas como aprendizado de máquina, processamento de linguagem natural e redes neurais artificiais.
- A criação da tecnologia LLM teve início na década de 1980, com o desenvolvimento de modelos de linguagem baseados em regras.
- Em 2015, a empresa OpenAI lançou o modelo de linguagem GPT (Generative Pre-training Transformer), que utilizava uma arquitetura baseada em transformers para gerar texto de alta qualidade.



# Integrando uma I.A. com uma aplicação

---

A OPENAI disponibiliza uma integração com sua inteligência artificial na forma de API.

Na camada gratuita versão 3.5 muito sujeita à instabilidades e acesso falho, além ter um crédito e prazo de duração limitados.

Ela está disponível para ser contratada direto pela OpenAi ou pela plataforma Azure..

Para acessar é preciso ter uma assinatura dos serviços, e seu acesso em larga escala é pago, porém a performance é bem melhor.



A close-up photograph of a red pushpin stuck into a map. The map shows various colored lines and text, though it is out of focus. The pushpin is sharp and has a classic design with a wide, flat head. The background is a soft, blurred mix of colors, suggesting an indoor setting with natural light.

# O Projeto

---

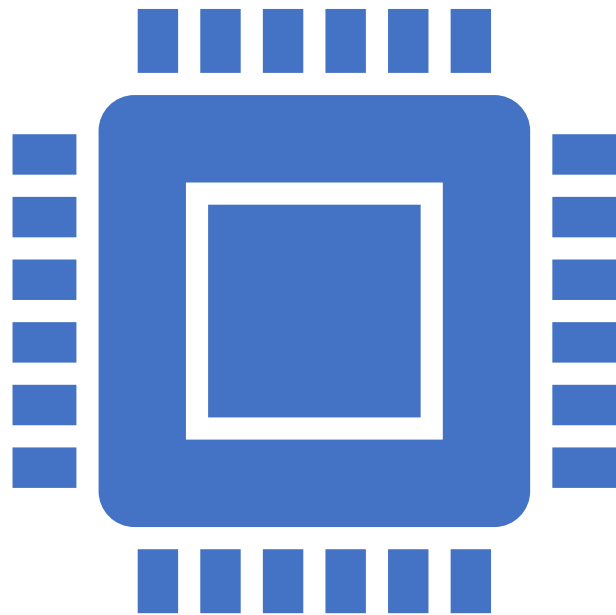
Criar uma Aplicação que elabore roteiros de viagens baseado nas informações da I.A.

1) Fornecer um roteiro de viagem para X dias para locais diversos, incluindo:

- Principais pontos turísticos
- Opções de Hospedagem
- Locais para Almoço e Jantar

2) Fornecer detalhes sobre os locais exibidos





# Montando os prompts

Como podemos configurar a lógica da aplicação?

Para comunicarmos com a I.A. e obter os melhores resultados, vamos utilizar um recurso de montar um prompt inicial que crie um modelo de resposta de acordo com o padrão que precisamos obter.

Neste cenário, utilizando a biblioteca da Microsoft Azure OpenAi podemos enviar mensagens com 3 definições diferentes:

- System – (pré-programação do sistema)
- User – (interação com o usuário)
- Assistant – (histórico de respostas anteriores da I.A.)

# Como obter JSON na resposta de uma I.A.

---

Para conseguirmos obter as respostas e categorizá-las, precisamos estabelecer um padrão, desta forma, através de prompt, forneceremos um modelo (Json Exemplo) e pediremos que as respostas da I.A. sejam no formato especificado.

Exemplo de mensagem inicial (System):

A sua resposta deverá vir em formato JSON conforme o exemplo que vou fornecer, você deverá estruturar o mesmo resultado, como neste modelo de exemplo em um array Json pronto para Deserialização, sua resposta deverá conter apenas o Json, sem nenhum comentário: { "Information": "colocar o texto da resposta", "Link": "colocar o endereço do site que referencia o local" }

# O que precisamos para transformar isso numa aplicação ?

Uma linguagem de programação para fazer requisições Http

Uma assinatura de um serviço de Inteligência Artificial

Configuração da comunicação com o serviço

Criação de prompts para comunicação com a I.A.

# Serviços que foram necessários

---

Microsoft Azure Web App

Microsoft Open Ai



# Modelo de Prompt 1

faça um roteiro de viagem de 5 dias para Grécia e conhecendo os principais pontos turísticos, indique um hotel e locais para almoçar e jantar baseados nas melhores experiências dos usuários em formato JSON conforme exemplo abaixo:você consegue estruturar o mesmo resultado neste modelo de exemplo em um array Json pronto para Deserialização :

```
{
  "dia": 1,
  "Cidade": "Veneza",
  "pontosTuristicos": [
    "Piazza San Marco",
    "Basílica de São Marcos",
    "Palácio Ducal",
    "Ponte dos Suspiros"
  ],
  "hotel": "Hotel Gritti Palace",
  "almoco": "Osteria Da Rialto",
  "jantar": "Ristorante Quadri"
}
```

```
]
```



# Modelo de Prompt 2

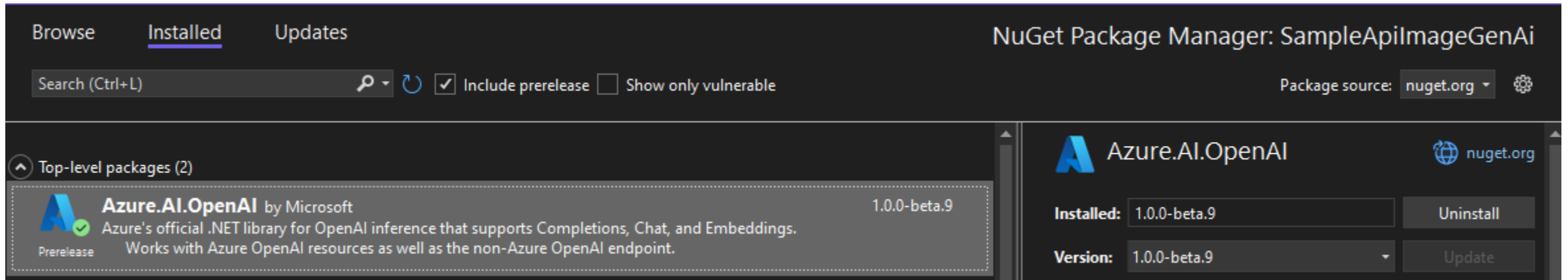
@ "você atuará como um guia de viagens, falando sobre o que há para se conhecer nos lugares que forem apontados na visão do que um turista gostaria de ver, inclusive falando de dados históricos do local e curiosidades, como data da criação do local, fatos históricos que já aconteceram, pessoas famosas que estiveram ali, citar se é cobrada entrada, se tem horários específicos e alguma restrição para crianças ou idosos A sua resposta deverá vir em formato JSON conforme o exemplo que vou fornecer, você deverá estruturar o mesmo resultado

como neste modelo de exemplo em um array Json pronto para Deserialização, sua resposta deverá conter apenas o Json, sem nenhum comentário.

: { "Information": "colocar o texto da resposta", "Link": "colocar o endereço do site que referencia o local" }

# Pacote Nuget utilizado

Biblioteca Azure Ai Openai no C# - versão 1.0.0-beta 9



Link:[learn.microsoft.com/pt-br/dotnet/api/overview/azure/ai.openai-readme?view=azure-dotnet-preview](https://learn.microsoft.com/pt-br/dotnet/api/overview/azure/ai.openai-readme?view=azure-dotnet-preview)

# 0 código

---

```
public List<DestinationModel> GetItinerary(ItineraryRequestModel request)
{
    OpenAIClient client = new(new Uri(_config.Endpoint), new AzureKeyCredential(_config.Key));

    IList<ChatMessage> messages = new List<ChatMessage>();

    messages.Add(new ChatMessage(ChatRole.System, PromptConstants.ItineraryQuestion));
    string question = $"Destino {request.Name} {request.days} dias";

    messages.Add(new ChatMessage(ChatRole.User, question));

    var chatCompletionsOptions = new ChatCompletionsOptions(messages);

    Response<ChatCompletions> response = client.GetChatCompletions(
        deploymentOrModelName: _config.DeploymentModel,
        chatCompletionsOptions);

    Console.WriteLine(response.Value.Choices[0].Message.Content);
    var jsonResponse = JsonSerializer.Deserialize<List<DestinationModel>>(response.Value.Choices[0].Message.Content);
    return jsonResponse;
}
```

# Os tipos de mensagens enviadas

SystemValue

AssistantValue

UserValue

FunctionValue

```
string nonAzureOpenAPIKey = "your-api-key-from-platform.openai.com";
var client = new OpenAIClient(nonAzureOpenAPIKey, new OpenAIClientOptions());
var chatCompletionsOptions = new ChatCompletionsOptions()
{
    DeploymentName = "gpt-3.5-turbo", // Use DeploymentName for "model" with non-Azure clients
    Messages =
    {
        new ChatMessage(ChatRole.System, "You are a helpful assistant. You will talk like a pirate."),
        new ChatMessage(ChatRole.User, "Can you help me?"),
        new ChatMessage(ChatRole.Assistant, "Arrrr! Of course, me hearty! What can I do for ye?"),
        new ChatMessage(ChatRole.User, "What's the best way to train a parrot?"),
    }
};
```



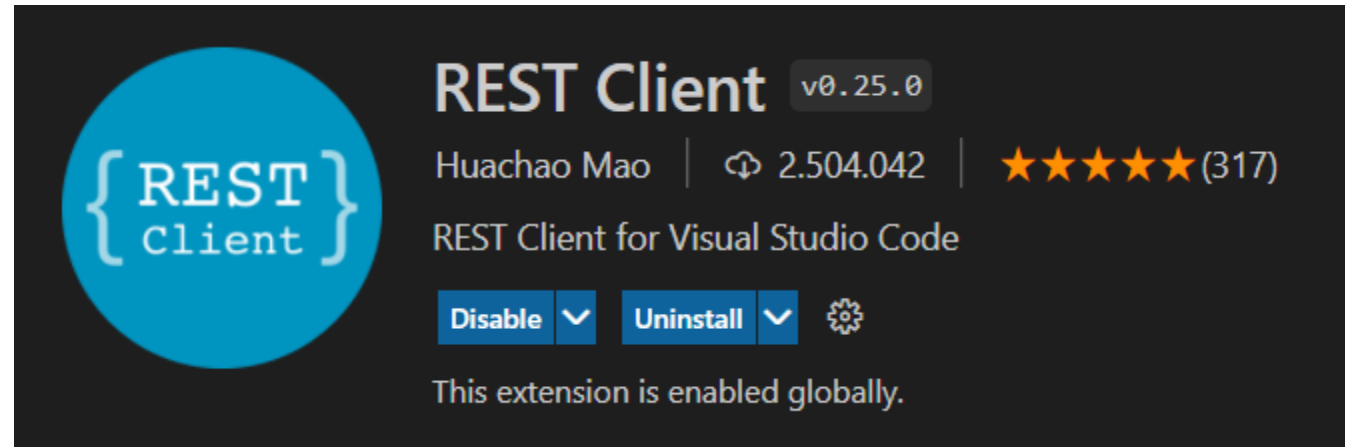
# Demonstração

---

Vai lá....Testa esse código....



# Para testar mais uma dica....



<https://marcionizzola.medium.com/esque%C3%A7a-o-postman-use-o-vscode-para-testar-api-s-2b98c004b750>

# Exemplo da execução - Itinerary

itinerary.http > ...

```
1
2  ### Obtem informações da localidade
   Send Request
3  POST https://localhost:7285/api/travel/itinerary
4  content-type: application/json
5
6  {
7    "name": "Roma",
8    "days": 7
9  }
10
11
```

```
1  HTTP/1.1 200 OK
2  Connection: close
3  Content-Type: application/json; charset=utf-8
4  Date: Fri, 01 Dec 2023 20:42:53 GMT
5  Server: Kestrel
6  Transfer-Encoding: chunked
7
8  [
9    {
10     "dia": 1,
11     "cidade": "Roma",
12     "pontosTuristicos": [
13       "Coliseu",
14       "Fórum Romano",
15       "Palatino",
16       "Arco de Constantino"
17     ],
18     "hotel": "Hotel Artemide",
19     "almoco": "Ristorante Pizzeria San Marco",
20     "jantar": "Il Pagliaccio"
21   },
22   {
23     "dia": 2,
24     "cidade": "Roma",
25     "pontosTuristicos": [
26       "Fontana di Trevi",
27       "Panteão",
28       "Piazza Navona",
29       "Castelo Sant'Angelo"
30     ],
31     "hotel": "Hotel Artemide",
32     "almoco": "Cul de Sac",
33     "jantar": "Ristorante Maccheroni"
34   },

```

# Exemplo da execução - Itinerary

```
[
  {
    "dia": 1,
    "cidade": "Roma",
    "pontosTuristicos": [
      "Coliseu",
      "Fórum Romano",
      "Arco de Constantino",
      "Panteão"
    ],
    "hotel": "Hotel de Rome",
    "almoco": "La Piazzetta",
    "jantar": "Pane e Salame"
  },
  {
    "dia": 2,
    "cidade": "Roma",
    "pontosTuristicos": [
      "Museus do Vaticano",
      "Capela Sistina",
      "Basílica de São Pedro"
    ],
    "hotel": "Holiday Inn Rome Aurelia",
    "almoco": "La Zanzara",
    "jantar": "L'Antica Roma"
  },
  {
    "dia": 3,
    "cidade": "Florença",
    "pontosTuristicos": [
      "Galleria degli Uffizi",
      "Ponte Vecchio",
      "Piazza del Duomo",
      "Catedral de Santa Maria del Fiore"
    ],
    "hotel": "Hotel Brunelleschi",
    "almoco": "Antico Noè",
    "jantar": "Il Santo Bevitore"
  },
]
```

```
{
  "dia": 4,
  "cidade": "Pisa",
  "pontosTuristicos": [
    "Torre de Pisa",
    "Catedral de Pisa",
    "Batistério de Pisa"
  ],
  "hotel": "NH Pisa",
  "almoco": "Osteria di Culegna",
  "jantar": "Trattoria Da Bruno"
},
{
  "dia": 5,
  "cidade": "Roma",
  "pontosTuristicos": [
    "Fontana di Trevi",
    "Escadaria de Espanha",
    "Piazza Navona"
  ],
  "hotel": "Grand Hotel de la Minerve",
  "almoco": "Osteria del Pegno",
  "jantar": "Hostaria da Pietro"
}
]
```

# Exemplo da execução – Place Info

place.http > POST /api/travel/placeinformation

```
1  ### place info
   Send Request
2  POST https://localhost:7285/api/travel/placeinformation
3  content-type: application/json
4
5  {
6    "placeName": "Fórum Romano",
7    "geoReference": "Roma-It"
8  }
9
10
11 ### place info
   Send Request
12 POST https://localhost:7285/api/travel/placeinformation
13 content-type: application/json
14
15 {
16   "placeName": "Orelhão de Itu",
17   "geoReference": "Itu/SP"
18 }
19
```

```
1  HTTP/1.1 200 OK
2  Connection: close
3  Content-Type: application/json; charset=utf-8
4  Date: Fri, 01 Dec 2023 20:41:35 GMT
5  Server: Kestrel
6  Transfer-Encoding: chunked
7
8  {
9    "information": "O Fórum Romano é uma antiga praça pública que foi o centr
   o da vida política e comercial em Roma durante séculos. Os turistas podem c
   aminhar por ruas de paralelepípedos e ver ruínas de templos, tribunais e ed
   ifícios governamentais, como o Arco de Tito, a Basílica de Maxêncio e a Cas
   a das Virgens Vestais. É possível também ver os túmulos de importantes figu
   ras históricas, como Julio César. Há sítios arqueológicos que permitem uma
   experiência mais interativa, como o Templo de Rômulo e a Casa de Augusto. A
   entrada no Fórum Romano está incluída no ingresso do Coliseu. É recomendado
   ter um guia para poder aproveitar a visita ao máximo, pois há muitos detalh
   es que passam despercebidos. ",
10   "link": "https://www.rome.net/roman-forum"
11 }
```

# Outro exemplo

Mantendo uma conversa sequencial

```
OpenAIClient client = new(new Uri(endpoint), new AzureKeyCredential(key));

IList<ChatMessage> messages = new List<ChatMessage>(); ← criamos uma lista

messages.Add(new ChatMessage(ChatRole.System, PromptConstants.TravelPrompt));
string question = string.Empty;

do
{
    Console.Write("Digite sua pergunta:");
    question = Console.ReadLine();
    if (question != string.Empty)
    {
        messages.Add(new ChatMessage(ChatRole.User, question)); ← enviamos a lista

        var chatCompletionsOptions = new ChatCompletionsOptions(messages);

        Response<ChatCompletions> response = client.GetChatCompletions(
            deploymentOrModelName: "nizzola35",
            chatCompletionsOptions);
        sempre adicionamos a mensagem de resposta recebida na lista também
        Console.WriteLine(response.Value.Choices[0].Message.Content);
        messages.Add(new ChatMessage(ChatRole.Assistant, response.Value.Choices[0].Message.Content)); ←
    }
} while (!String.IsNullOrEmpty(question));
```



# Considerações

Com esta abordagem você pode criar inúmeros serviços como:

- Assistente de problemas mecânicos
- Montagem de treinos para exercícios
- Receitas para cozinhar
- Resumos de Livros

# Referências

[https://github.com/Azure/azure-sdk-for-net/blob/Azure.AI.OpenAI\\_1.0.0-beta.9/sdk/openai/Azure.AI.OpenAI/README.md](https://github.com/Azure/azure-sdk-for-net/blob/Azure.AI.OpenAI_1.0.0-beta.9/sdk/openai/Azure.AI.OpenAI/README.md)

<https://learn.microsoft.com/en-us/dotnet/api/overview/azure/ai.openai-readme?view=azure-dotnet-preview>

# Contatos

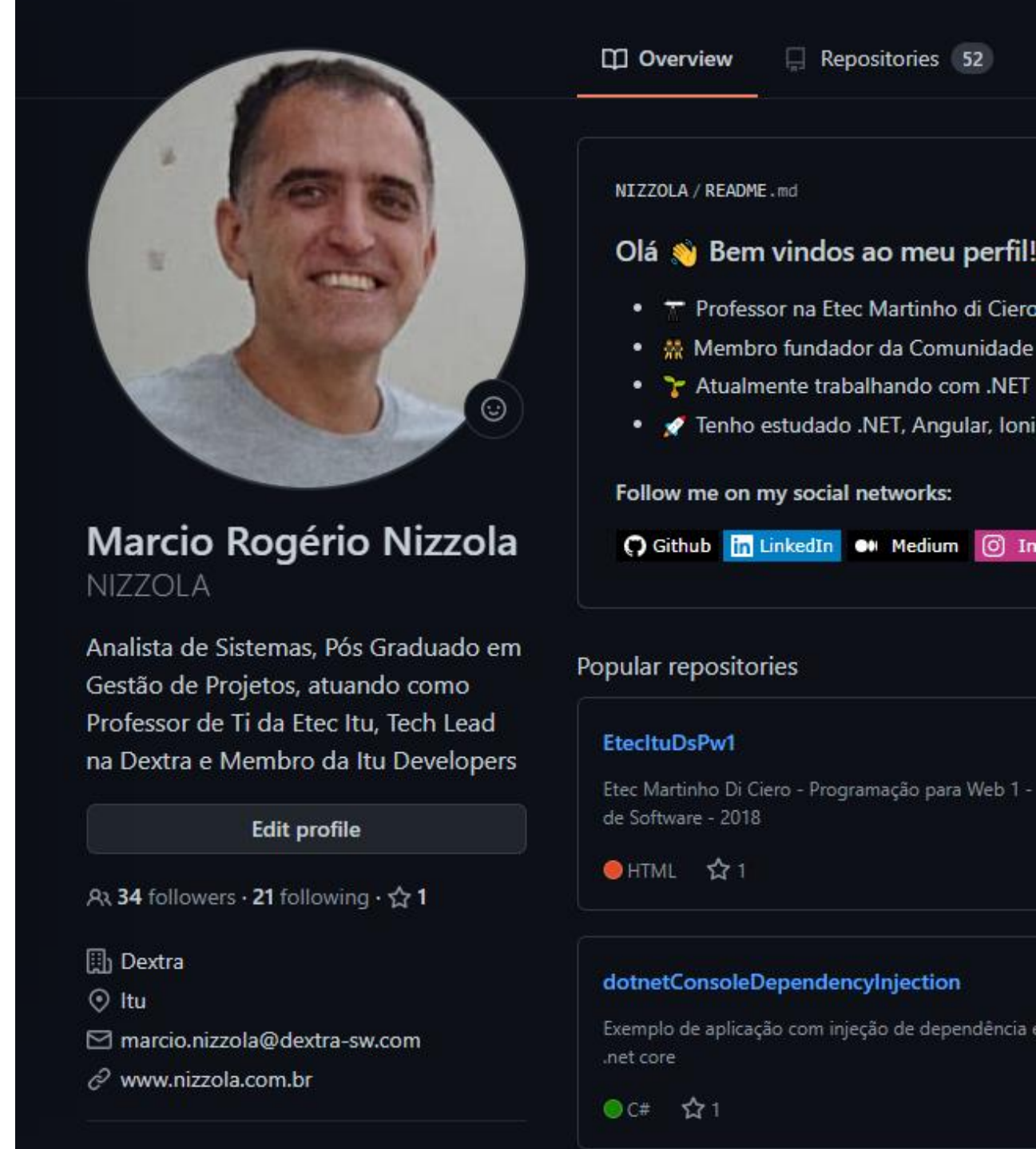
<https://linktr.ee/NIZZOLA>

<https://github.com/nizzola>

<https://www.linkedin.com/in/nizzola>

Mais de 100 artigos sobre .NET

[marcionizzola.medium.com](http://marcionizzola.medium.com)



The screenshot shows a GitHub profile for Marcio Rogério Nizzola. The profile includes a circular profile picture of a smiling man, a bio stating he is a Systems Analyst, Graduate in Project Management, and a Tech Lead at Etec Itu, and a list of social media links for GitHub, LinkedIn, Medium, and Instagram. The profile also shows 34 followers and 21 following. Below the profile, there are two repository cards: 'EtecltuDsPw1' and 'dotnetConsoleDependencyInjection'.

**Overview** Repositories 52

**NIZZOLA / README .md**

**Olá 🙌 Bem vindos ao meu perfil!**

- 🎓 Professor na Etec Martinho di Ciero
- 👥 Membro fundador da Comunidade
- 🌱 Atualmente trabalhando com .NET
- 🚀 Tenho estudado .NET, Angular, Ioni

**Follow me on my social networks:**

[Github](#) [LinkedIn](#) [Medium](#) [Instagram](#)

**Analista de Sistemas, Pós Graduado em Gestão de Projetos, atuando como Professor de Ti da Etec Itu, Tech Lead na Dextra e Membro da Itu Developers**

**Edit profile**

**34 followers · 21 following · ☆ 1**

**Dextra**  
Itu  
marcio.nizzola@dextra-sw.com  
www.nizzola.com.br

**Popular repositories**

**EtecltuDsPw1**  
Etec Martinho Di Ciero - Programação para Web 1 - de Software - 2018  
HTML ☆ 1

**dotnetConsoleDependencyInjection**  
Exemplo de aplicação com injeção de dependência .net core  
C# ☆ 1

<https://forms.gle/qvTJsQJknKM1fVNv5>



NPS Palestra

compartilhe sua opinião conosco