

**GERADOR DE
RESPOSTAS DE
IMAGEM
USANDO
GEMINI**



O QUE É?

O aplicativo é uma solução web simples que recebe imagens via API, especificamente para realizar OCR (Reconhecimento Óptico de Caracteres) e gerar respostas com base no texto extraído das imagens via API do GEMINI.



API

DESENVOLVIDO EM C#

```
[Route("api/[controller]")]
[ApiController]
0 referências
public class ImageController : ControllerBase
{
    [HttpPost("convert-image")]
    0 referências
    public async Task<IActionResult> ConvertImage(IFormFile image)
    {
        if (image == null || image.Length == 0)
            return BadRequest("Envie uma imagem valida.");

        string textFromImage;
        try
        {
            using (var stream = new MemoryStream())
            {
                await image.CopyToAsync(stream);
                var ocrEngine = new TesseractEngine(@".\\tessdata", "eng", EngineMode.Default);
                var img = Pix.LoadFromMemory(stream.ToArray());
                var page = ocrEngine.Process(img);
                textFromImage = page.GetText();
            }
        }
        catch (TesseractException ex)
        {
            return StatusCode(500, $"Tesseract falhou: {ex.Message}");
        }

        var responseText = await GetResponseFromGemini(textFromImage);
        return Ok(responseText);
    }
}
```

API

DESENVOLVIDO EM C#

```
1 referência
private static async Task<string> GetResponseFromGemini(string text)
{
    var apiKey = "AIzaSyDOGihRSUqZTjkir05QJyTPoT1MRMjLN7o";
    var client = new RestClient("https://generativelanguage.googleapis.com/v1beta/models/gemini-1.5-flash-");
    var request = new RestRequest("", Method.Post);
    request.AddHeader("Content-Type", "application/json");

    request.AddQueryParameter("key", apiKey);

    request.AddJsonBody(new
    {
        contents = new[]
        {
            new
            {
                parts = new[]
                {
                    new { text }
                }
            }
        }
    });

    var response = await client.ExecuteAsync(request);

    if (response.IsSuccessfull)
    {
        var jsonResponse = System.Text.Json.JsonSerializer.Deserialize<GeminiResponse>(response.Content);
        return jsonResponse?.candidates[0]?.content?.parts[0]?.text ?? "";
    }
    else
        return $"Error: {response.StatusDescription} - {response.Content}";
}
```

```
1 referência
public class GeminiResponse
{
    0 referências
    public Candidate[] candidates { get; set; }

    1 referência
    public class Candidate
    {
        1 referência
        public Content content { get; set; }

        1 referência
        public class Content
        {
            1 referência
            public Part[] parts { get; set; }

            1 referência
            public class Part
            {
                1 referência
                public string text { get; set; }
            }
        }
    }
}
```

HTML

```
<html /> <html /> <body /> <script />
<!DOCTYPE html>
<html lang="pt">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Upload de Imagem</title>
  <style>...
</style>
</head>
<body>
  <div class="container">
    <h2>Gerador de Respsotas</h2>
    <input type="file" id="inputImagem" accept="image/*">
    <button id="botaoUpload">Enviar Imagem</button>
    <p id="mensagemResposta"></p>
  </div>
  <script>...
</script>
</body>
</html>
```

CSS

```
<!DOCTYPE html>
<html lang="pt">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Upload de Imagem</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f4f4f4;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
      margin: 0;
    }

    .container {
      background-color: #fff;
      padding: 20px;
      border-radius: 10px;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
      text-align: center;
      width: 300px;
    }

    h2 {
      margin-bottom: 20px;
    }

    input[type="file"] {
      margin-bottom: 20px;
    }

    button {
      background-color: #007bff;
      color: white;
      border: none;
      padding: 10px 20px;
      cursor: pointer;
      border-radius: 5px;
      font-size: 16px;
    }

    button:hover {
      background-color: #0056b3;
    }
  </style>
</head>
<body>
  <div class="container">
    <h2>Carregar Imagem</h2>
    <input type="file">
    <button>Enviar</button>
  </div>
</body>
</html>
```

```
#mensagemResposta {
  margin-top: 20px;
  font-weight: bold;
  background-color: #e7f1ff;
  border-left: 4px solid #007bff;
  padding: 10px;
  border-radius: 5px;
  text-align: left;
  white-space: pre-wrap;
}
```

JAVASCRIPT

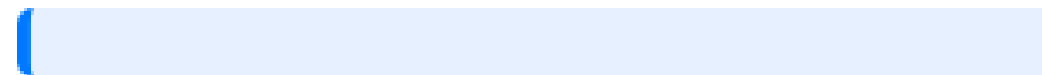
```
<> index.html X
<> index.html > html > body > script
1  <!DOCTYPE html>
2  <html lang="pt">
3  > <head> ...
60 </head>
61 <body>
62 > <div class="container"> ...
67 </div>
68
69 <script>
70   document.getElementById("botaoUpload").addEventListener("click", async function () {
71     const inputImagem = document.getElementById("inputImagem");
72     const arquivo = inputImagem.files[0];
73     const mensagemResposta = document.getElementById("mensagemResposta");
74
75     if (!arquivo) {
76       mensagemResposta.textContent = "Selecione uma imagem.";
77       return;
78     }
79
80     const formData = new FormData();
81     formData.append("image", arquivo);
82
83     try {
84       const resposta = await fetch("https://localhost:7196/api/Image/convert-image", {
85         method: "POST",
86         body: formData,
87       });
88
89       if (resposta.ok) {
90         const dados = await resposta.text();
91         mensagemResposta.textContent = "Resposta: " + dados;
92       } else {
93         mensagemResposta.textContent = "Falha no upload: " + resposta.statusText;
94       }
95     } catch (erro) {
96       mensagemResposta.textContent = "Erro: " + erro.message;
97     }
98   });
99 </script>
100 </body>
101 </html>
```

TELA INICIAL

Gerador de Respsotas

Nenhum arquivo escolhido

Enviar Imagem



TELA COM ERRO

Gerador de Respsotas

Escolher arquivo Nenhum arquivo escolhido

Enviar Imagem

Selecione uma imagem.

TELA COM RESPOSTA

Gerador de Respsotas

Escolher arquivo Sem título.png

Enviar Imagem

Resposta: Para determinar o valor de O (Ordem de Crescimento) da função $f(n) = 3n^3 + n$, seguimos estas etapas:

1. ****Identificar o termo dominante:****
O termo dominante é o termo que cresce mais rapidamente à medida que n aumenta. Neste caso, o termo dominante é $3n^3$, pois cresce mais rapidamente que n quando n se torna grande.

2. ****Descartar coeficientes e termos de ordem inferior:**** O coeficiente 3 e o termo n são irrelevantes para determinar o valor de O quando n se torna muito grande.

3. ****Expressar a complexidade usando a notação O :** Portanto, a complexidade da função $f(n) = 3n^3 + n$ é $O(n^3)$.

****Conclusão:**** O valor de O para a expressão de complexidade $f(n) = 3n^3 + n$ é **$O(n^3)$** . Isso significa que a função cresce como n^3 quando n se torna muito grande.

OBRIGADO!

ALUNO: VITOR FACCIO
RA: 04721-016