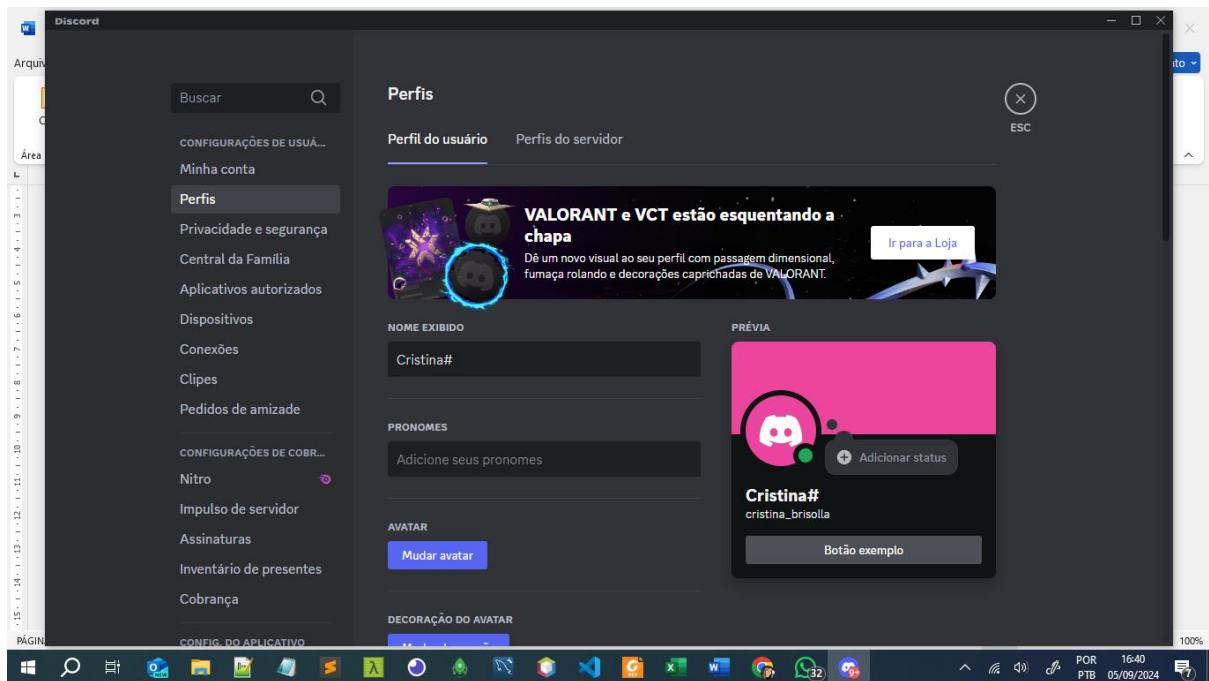
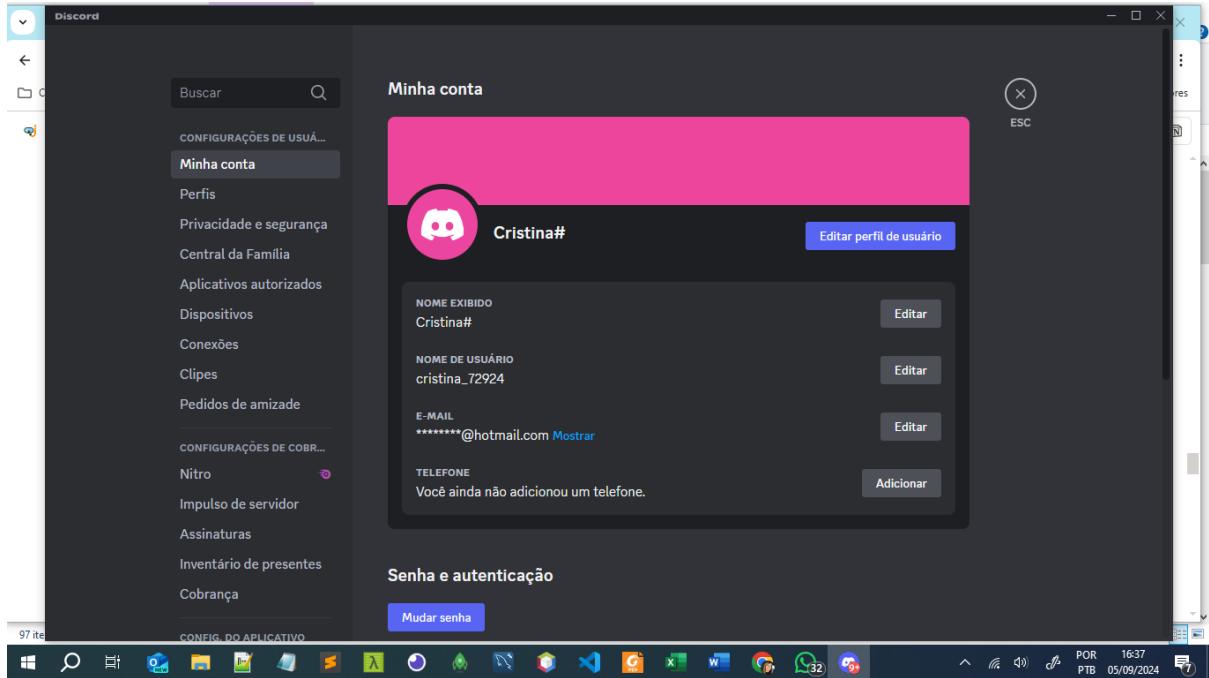


Meu Readme para estudo

Programação e Google Gemini



LINK PARA INSTRUÇÕES CSS – ALURA

<https://www.alura.com.br/artigos/css>

Vamos criar uma página para consulta de atletas.

Criamos o código para a atleta Rebeca Andrade e pedimos para o Gemini criar o código para mais duas atletas (ao invés de fazermos manualmente)

Passamos essa orientação:

tenho o seguinte código html

```
... <div class="item-resultado">
```

```
<h2><a href="#" target="_blank"></a> Rebeca Andrade</h2>
```

```
<p class="descricao-meta ">Rebeca Andrade é uma ginasta brasileira que conquistou o coração dos brasileiros com suas performances incríveis. Campeã olímpica e multimedalhista, ela se tornou um símbolo de superação e determinação no esporte.</p>
```

```
<a href="https://pt.wikipedia.org/wiki/Rebeca_Andrade" target="_blank"> Mais Informações</a>
```

```
</div>
```

```
...
```

Agora preciso criar mais duas atletas: a Rayssa Leal do skate e a Beatriz Souza do judô

Resposta do Gemini:

Vamos lá, criar mais atletas!

Entendendo o código:

Você já tem um bloco HTML que representa uma atleta, com:

- **Um título (h2):** Contém um link para um perfil (ainda sem o nome da atleta).
- **Uma descrição:** Um parágrafo com informações sobre a atleta.
- **Um link externo:** Para mais detalhes sobre a atleta.

Duplicando e adaptando:

Para adicionar as outras duas atletas, vamos duplicar esse bloco e fazer as devidas alterações:

HTML

```
<div class="item-resultado">
```

```
<h2><a href="#" target="_blank">Rebeca Andrade</a></h2>
```

<p class="descricao-meta">Rebeca Andrade é uma ginasta brasileira que conquistou o coração dos brasileiros com suas performances incríveis. Campeã olímpica e multimedalhista, ela se tornou um símbolo de superação e determinação no esporte.</p>

Mais Informações

</div>

<div class="item-resultado">

<h2>Rayssa Leal</h2>

<p class="descricao-meta">Rayssa Leal, a Fadinha, é uma skatista brasileira que conquistou o mundo com seu talento e carisma. Aos poucos anos, ela se tornou a mais jovem medalhista olímpica da história do Brasil.</p>

Mais Informações

</div>

<div class="item-resultado">

<h2>Beatriz Souza</h2>

<p class="descricao-meta">Beatriz Souza é uma judoca brasileira que conquistou a medalha de ouro nas Olimpíadas de Tóquio. Sua história de superação e dedicação a inspirou milhões de pessoas.</p>

Mais Informações

</div>

Use o código [com cuidado](#).

O que foi feito:

- **Duplicamos o bloco:** Criamos duas novas divs com a mesma estrutura da primeira.
- **Alteramos os nomes:** Substituímos "Rebeca Andrade" pelos nomes das outras atletas.
- **Adaptamos as descrições:** Escrevemos novas descrições, destacando as principais conquistas e características de cada atleta.
- **Atualizamos os links:** Substituímos os links para as páginas da Wikipedia das respectivas atletas.

Observações:

- **Personalização:** Você pode personalizar ainda mais, adicionando fotos, datas de nascimento, ou outras informações relevantes.
- **Organização:** Se você tiver muitas atletas, pode considerar usar uma estrutura de dados (como JSON) para armazenar as informações e gerar o HTML dinamicamente.

- **Estilização:** Aplique estilos CSS para deixar a lista de atletas com uma aparência mais profissional e atraente.

Exemplo de estilização básica (CSS):

CSS

```
.item-resultado {  
    border: 1px solid #ccc;  
    padding: 10px;  
    margin-bottom: 20px;  
}
```

```
.descricao-meta {  
    font-size: 14px;  
    color: #666;  
}
```

Use o código [com cuidado](#).

Com esse código, você terá uma lista com as três atletas, cada uma com suas informações básicas.

Quer adicionar mais alguma funcionalidade? Podemos explorar opções como:

- **Ordenação:** Ordenar as atletas por nome, esporte ou data de nascimento.
- **Filtros:** Permitir que o usuário filtre as atletas por esporte.
- **Modais:** Exibir mais detalhes sobre cada atleta ao clicar no nome.

Qualquer dúvida, é só perguntar!

Agora desejamos incluir uma instrução em java script

Pedimos para o Gemini fazer o código

Resposta

Criando seu primeiro "Olá, mundo!" em JavaScript

Com certeza! Criar um "Olá, mundo!" é o primeiro passo de qualquer programador. Vamos te guiar nesse processo:

Entendendo o básico:

- **HTML:** É a estrutura da página, como um esqueleto.
- **CSS:** É responsável pelo estilo da página, como cores e layout.
- **JavaScript:** É a linguagem de programação que adiciona interatividade à página.

Criando a página HTML:

1. **Crie um novo arquivo:** Salve-o com a extensão .html (por exemplo, index.html).
2. **Adicione a estrutura básica HTML:**

HTML

```
<!DOCTYPE html>

<html>
  <head>
    <title>Olá, mundo!</title>
  </head>
  <body>
    <script>
      // Aqui vai o seu código JavaScript
    </script>
  </body>
</html>
```

Use o código [com cuidado](#).

Escrevendo o JavaScript:

Dentro da tag <script> você vai inserir o código JavaScript. Para exibir a mensagem "Olá, mundo!", podemos usar o método alert():

HTML

```
<!DOCTYPE html>

<html>
```

```
<head>
    <title>Olá, mundo!</title>
</head>
<body>
    <script>
        alert("Olá, mundo!");
    </script>
</body>
</html>
```

Use o código [com cuidado](#).

Explicando o código:

- **<script>**: Indica que o conteúdo dentro desta tag é código JavaScript.
- **alert("Olá, mundo!");**: Exibe uma caixa de diálogo com a mensagem "Olá, mundo!".

Abrir o arquivo:

- **Salve o arquivo**: Certifique-se de salvar o arquivo com a extensão .html.
- **Abra no navegador**: Clique duas vezes no arquivo ou arraste-o para a janela do seu navegador.

Ao abrir o arquivo no navegador, você verá uma caixa de diálogo com a mensagem "Olá, mundo!".

Outras formas de exibir a mensagem:

- **Usando o console**:

JavaScript

```
console.log("Olá, mundo!");
```

Use o código [com cuidado](#).

A mensagem será exibida no console do navegador, útil para depuração.

- **Escrevendo diretamente no HTML:**

HTML

```
<body>  
  <p>Olá, mundo!</p>  
</body>
```

Use o código [com cuidado](#).

A mensagem será exibida como um parágrafo na página.

Parabéns! Você acabou de criar seu primeiro programa em JavaScript.

Quer aprender mais? Podemos explorar:

- **Variáveis:** Para armazenar dados.
- **Operadores:** Para realizar cálculos.
- **Condições:** Para tomar decisões no código.
- **Laços:** Para repetir ações.
- **Funções:** Para organizar o código em blocos reutilizáveis.

Qualquer dúvida, é só perguntar!

Gostaria de ver exemplos mais complexos ou aprender sobre algum tópico específico de JavaScript?

Após aprender a criar um arquivo em java script, vamos transportá-lo para a nossa aplicação:

Criamos um arquivo em java script de nome dados.js, onde serão relacionados todos os atletas que queremos apresentar na busca.

Voltamos no arquivo criado app.js e no lugar de Olá Mundo, colocamos o nome do arquivo js criado para armazenar os atletas que aparecerão na busca.

dados.js

The screenshot shows the Microsoft Edge browser window with the following details:

- Address Bar:** CursoIA
- File Explorer (Left Side):** Shows files in the CURSOIA folder:
 - index.html
 - app.js
 - dados.js
 - style.css
 - dados.js (under CURSOIA)
 - app.js (under CURSOIA)
 - index.html (under CURSOIA)
 - Inteligência Artificial Generativa.docx
- Code Editor (Main Area):** The file "dados.js" is open, containing the following code:

```
dados.js > [object Object]
dados
1 let dados = "Rayssa Leal"
```
- Bottom Status Bar:** Ln 1, Col 26, Espaços: 4, UTF-8, CRLF, JavaScript, Port: 5500, Go Live
- Taskbar (Bottom):** Shows various pinned icons for Microsoft Office applications like Word, Excel, and Powerpoint.

Substituindo no app.js:

The screenshot shows a Microsoft Edge browser window with the following details:

- File Explorer Sidebar:** On the left, there's a sidebar titled "EXPLORADOR" containing a tree view of files. The visible items include "CURSOSIA", "css" (with "# style.css"), "JS" (with "app.js", "dados.js", and "index.html"), and a document named "Inteligência Artificial Generativa.docx".
- Code Editor Tab:** The main content area has tabs for "index.html", "JS app.js", "JS dados.js", and "# style.css". The "JS app.js" tab is active, displaying the following code:

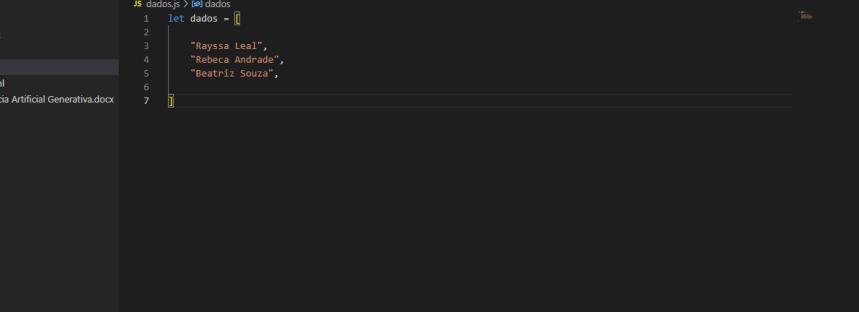
```
1  console.log(dados);
```
- Bottom Status Bar:** The status bar at the bottom shows "Ln 1, Col 20", "Espaços: 4", "UTF-8", "CRLF", "JavaScript", "Port: 5500", "Go Live", and a "PTB" icon.
- Taskbar:** The taskbar at the bottom of the screen shows icons for File Explorer, File Explorer (with 34 notifications), and other system icons.

No html precisamos incluir os 2 arquivos java. Devem ser posicionados antes do </body>:

Nesta ordem:

```
<script src="dados.js"></script>  
<script src="app.js"></script>
```

Até aqui só incluímos um atleta no nosso arquivo dados.js. Para aproveitarmos a mesma variável podemos incluir todos os atletas dentro de um colchete., formando uma LISTA em js. Fica assim:



The screenshot shows a Microsoft Edge browser window with the following details:

- Address Bar:** CursoA
- Page Content:** A 404 error page titled "Página não encontrada". It includes a link to "Mais detalhes" and a "Fazer upload de arquivo" button.
- Taskbar:** Shows the taskbar with the Edge icon highlighted, along with icons for File Explorer, Task View, and other system icons.

Obs= sempre começa no índice “0” . Aqui temos Rayssa =0, Rebeca=1, Beatriz=2

Mas no nosso projeto precisamos que sejam apresentados o nome, a descrição e o gênero do esporte. Precisaremos trabalhar com **OBJETO** em js. Incluímos os objetos dentro da lista.

Fica assim:

```
index.html ● JS app.js JS dados.js ● # style.css
CURSOSIA
css
# style.css
JS app.js
JS dados.js
index.html
Inteligência Artificial Generativa.docx

JS dados.js > [dados
1 Titulos dados =
2
3
4
5 titulo : "Rayssa Leal",
6 descricao : "Rayssa Leal, a Fadinha, é uma skatista brasileira que conquistou o mundo com seu talento e carisma",
7 link : "https://pt.wikipedia.org/wiki/Rayssa_Leal",
8
9 },
10
11 {
12
13 titulo : "Rebeca Andrade",
14 descricao : "Rebeca Andrade é uma ginasta brasileira que conquistou o coração dos brasileiros com suas performances",
15 link : "https://pt.wikipedia.org/wiki/Rebeca_Andrade",
16
17 },
18
19 {
20
21 titulo : "Beatriz Souza",
22 descricao : "Beatriz Souza é uma judoca brasileira que conquistou a medalha de ouro nas Olimpíadas de Tóquio",
23 link : "https://pt.wikipedia.org/wiki/Beatriz_Souza"
24
25 }
26
27 ]
28
29
```

Exercício:

Claro! Que tal criar uma base de conhecimento sobre seus filmes favoritos? Desafio:

Crie um arquivo filmes.js e dentro dele, declare uma lista de objetos. Cada objeto deve representar um filme e conter as seguintes informações:

Título

Sinopse

Gênero

Ano de lançamento

Depois, no seu arquivo app.js, utilize console.log para exibir o título e a sinopse de cada filme da sua lista.

Para deixar mais interessante, tente adicionar pelo menos 5 filmes diferentes e, se quiser, inclua uma nova propriedade, como "Diretor" ou "Elenco".

Divirta-se e não esqueça de compartilhar seu código no Discord! 😊🎬

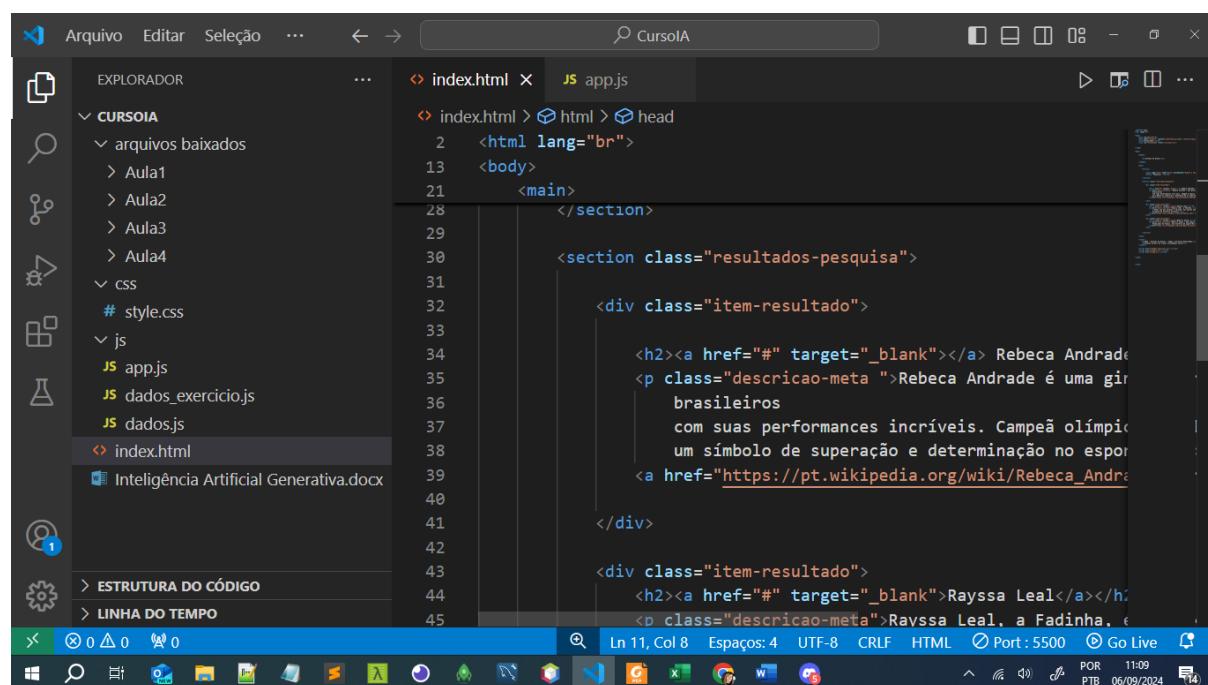
Aula 4

Como inserir as informações que estão no console para a página html

Precisamos interagir o java script com o html

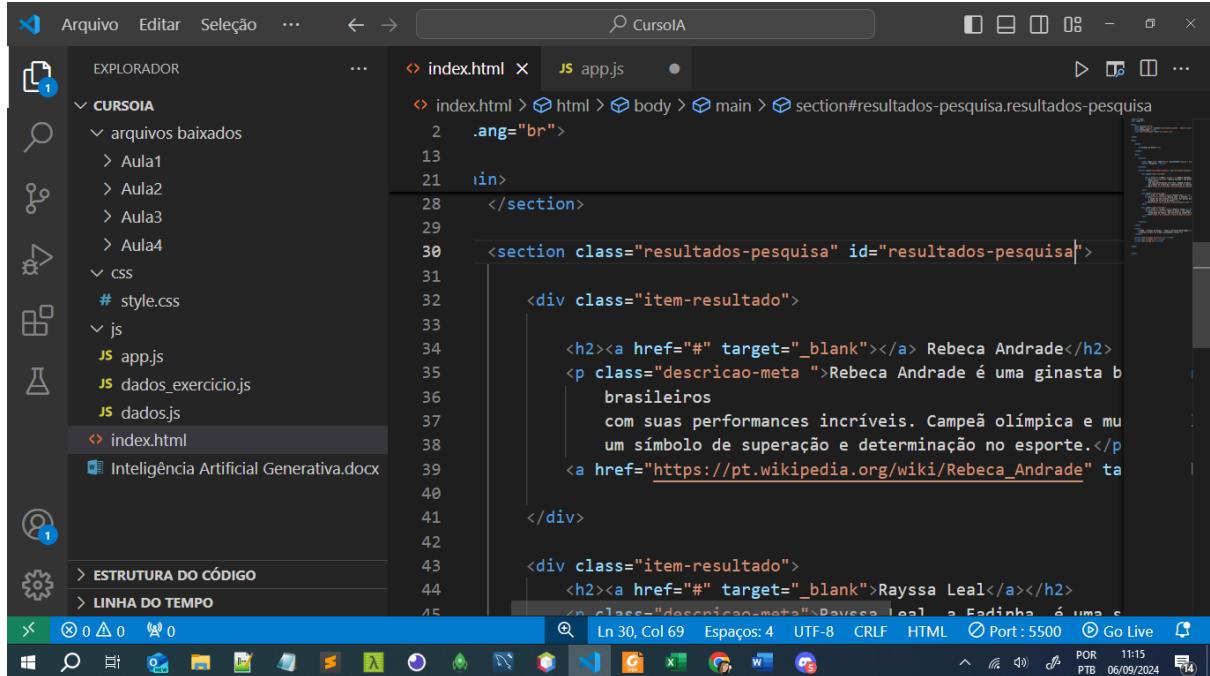
Em qual lugar vamos inserir?

O resultado da busca vem da <div> item-resultado



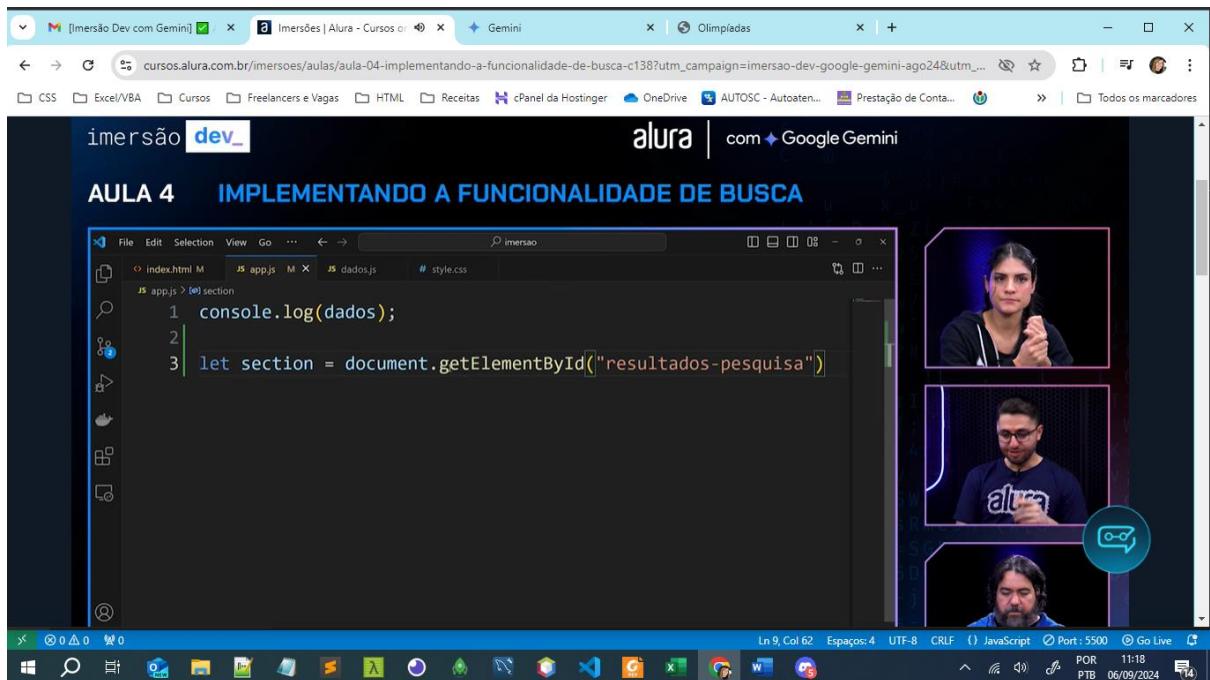
The screenshot shows the Microsoft Edge browser's developer tools open to the DOM tab. The left sidebar shows the file structure of the project 'CURSOIA'. The main pane displays the HTML code for 'index.html'. A specific div element with the class 'item-resultado' is selected and highlighted in yellow. This div contains two h2 elements, each with an anchor tag pointing to a blank target. Below each h2 is a paragraph with descriptive text about the person linked. The status bar at the bottom shows the file is 'Ln 11, Col 8' with 'Espaços: 4' and 'Port: 5500'.

Então as informações que constam do console precisam estar dentro da <section> resultados-pesquisa. Vamos precisar acrescentar um ID nessa instrução. Fica assim:



```
Arquivo Editar Seleção ... 🔍 CursoA
EXPLORADOR ...
CURSOIA
  arquivos baixados
    Aula1
    Aula2
    Aula3
    Aula4
  css
    # style.css
  js
    JS app.js
    JS dados_exercicio.js
    JS dados.js
    index.html
    Inteligência Artificial Generativa.docx
ESTRUTURA DO CÓDIGO
LINHA DO TEMPO
index.html x JS app.js
index.html > html > body > main > section#resultados-pesquisa.resultados-pesquisa
  .ang="br"
  in
  </section>
<section class="resultados-pesquisa" id="resultados-pesquisa">
  <div class="item-resultado">
    <h2><a href="#" target="_blank">Rebeca Andrade</a></h2>
    <p class="descricao-meta">Rebeca Andrade é uma ginasta brasileiros com suas performances incríveis. Campeã olímpica e um símbolo de superação e determinação no esporte.</p>
    <a href="https://pt.wikipedia.org/wiki/Rebeca_Andrade" ta
  </div>
  <div class="item-resultado">
    <h2><a href="#" target="_blank">Rayssa Leal</a></h2>
    <p class="descricao-meta">Rayssa Leal é a rainha das olimpíadas de surfe. Ela é a primeira brasileira a ganhar uma medalha de ouro na modalidade. Sua história é inspiradora e demonstra que com determinação e talento podemos alcançar grandes resultados.
  </div>
</section>
```

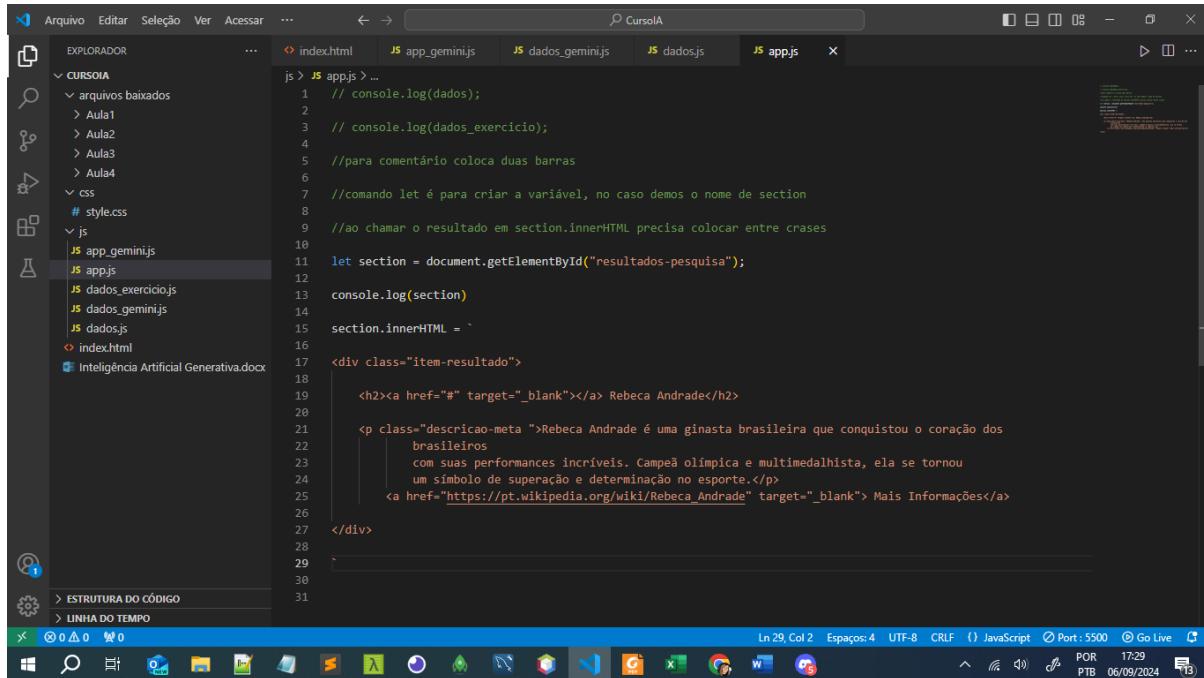
Agora precisamos voltar no app.js e criar uma variável para receber este ID. Assim:



```
[imersão Dev com Gemini] Imersões | Alura - Cursos on-line Gemini
cursos.alura.com.br/imersões/aulas/aula-04-implementando-a-funcionalidade-de-busca-c138?utm_campaign=imersao-dev-google-gemini-ago24&utm_medium=referral&utm_source=google
CSS Excel/VBA Cursos Freelancers e Vagas HTML Receitas cPanel da Hostinger OneDrive AUTOSC - Autoaten... Prestação de Conta...
imersão dev_
alura com Google Gemini
AULA 4 IMPLEMENTANDO A FUNCIONALIDADE DE BUSCA
```

```
File Edit Selection View Go ... 🔍 imersão
index.html M JS app.js M JS dados.js # style.css
JS app.js > section
  1 console.log(dados);
  2
  3 let section = document.getElementById("resultados-pesquisa")
```

Continuando, criamos ainda no app.js um comando para carregar na variável section (js) as informações que estão no html

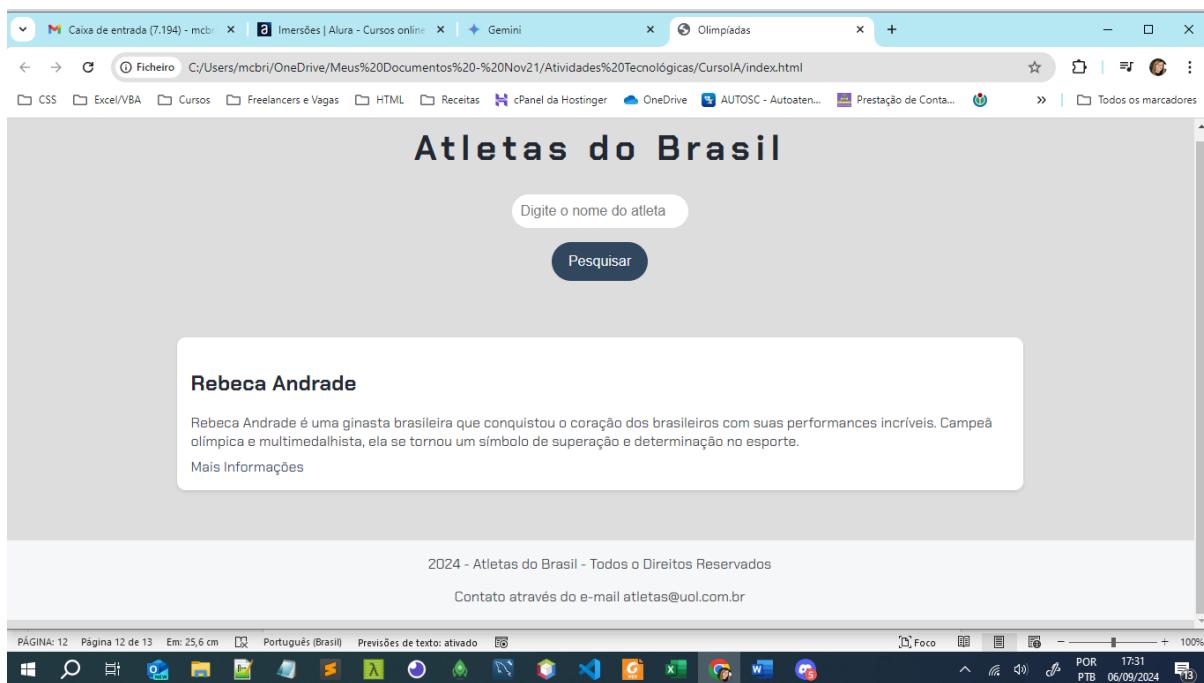


```
Arquivo Editar Seleção Ver Acessar ... ← → C:\Users\mcibri\OneDrive\Meus%20Documentos%20-%20Nov21\Atividades%20Tecnológicas\CursolA JS app.gemini.js JS dados_gemini.js JS dadosjs JS app.js x

EXPLORADOR
CURSOIA
arquivos baixados
Aula1
Aula2
Aula3
Aula4
css
# style.css
JS
JS app_gemini.js
JS appjs
JS dados_exercicio.js
JS dados_gemini.js
JS dadosjs
index.html
Inteligência Artificial Generativa.docx

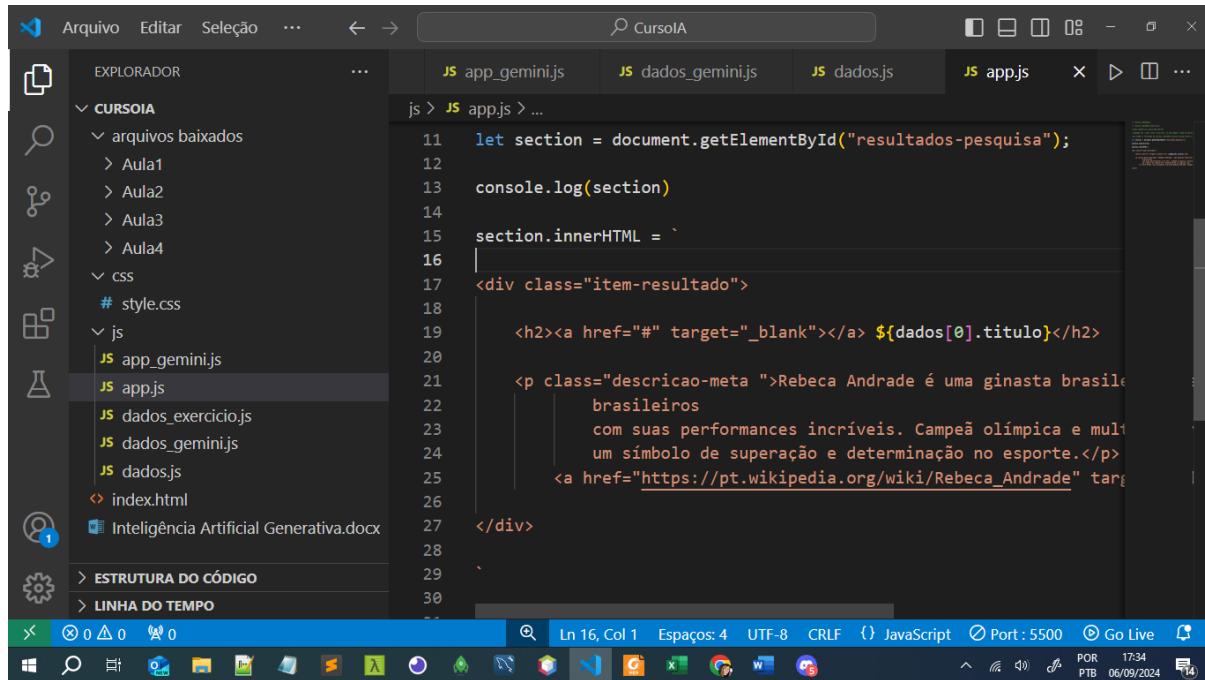
js > JS app.js > ...
1 // console.log(dados);
2
3 // console.log(dados_exercicio);
4
5 //para comentário coloca duas barras
6
7 //comando let é para criar a variável, no caso demos o nome de section
8
9 //ao chamar o resultado em section.innerHTML precisa colocar entre crases
10
11 let section = document.getElementById("resultados-pesquisa");
12
13 console.log(section)
14
15 section.innerHTML = `

<div class="item-resultado">
    <h2><a href="#" target="_blank">Rebeca Andrade</a></h2>
    <p class="descricao-metadados">Rebeca Andrade é uma ginasta brasileira que conquistou o coração dos brasileiros com suas performances incríveis. Campeã olímpica e multimedalhista, ela se tornou um símbolo de superação e determinação no esporte.</p>
    <a href="https://pt.wikipedia.org/wiki/Rebeca_Andrade" target="_blank">Mais Informações</a>
</div>
`
```



Agora já aparece no site o nome da esportista. Porém precisamos que apareça o conteúdo de uma variável que pode conter qq um dos três nomes, no nosso caso. Fica assim:

No código que veio do html , no local do nome criamos uma variável : \${dados[0].titulo}



```
Arquivo Editar Seleção ... ← → 🔍 Cursola
```

EXPLORADOR

✓ CURSOIA

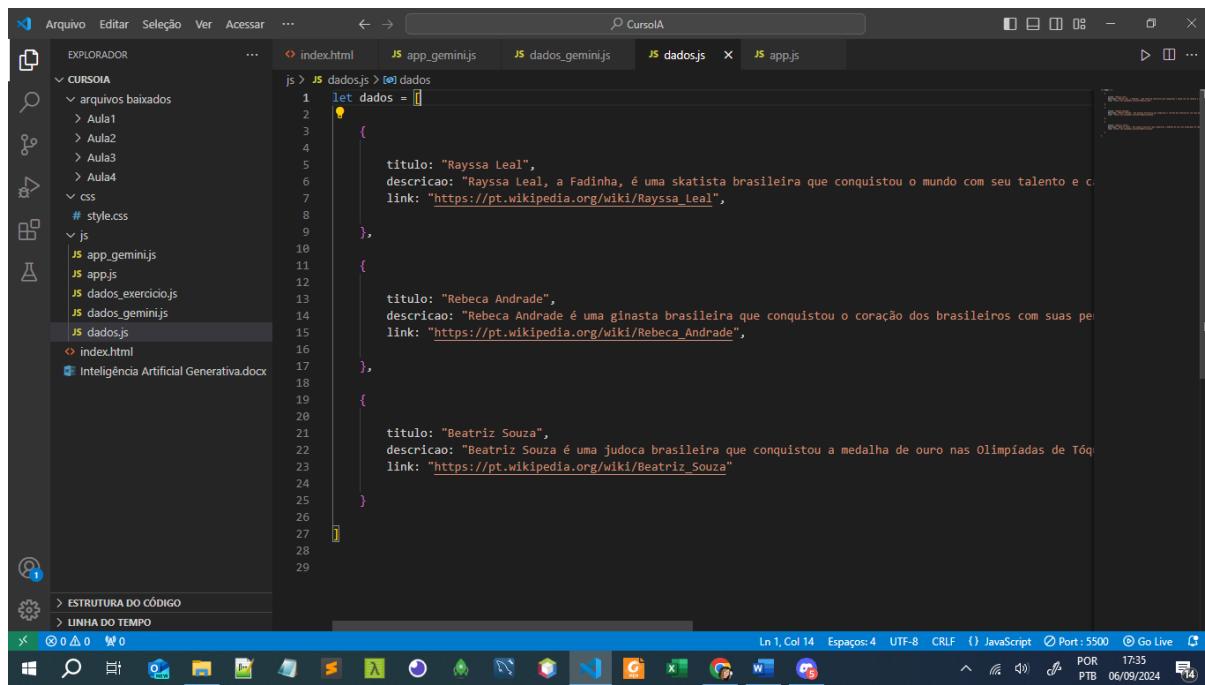
- ✓ arquivos baixados
 - > Aula1
 - > Aula2
 - > Aula3
 - > Aula4
- ✓ css
 - # style.css
- ✓ js
 - JS app_geminijs
 - JS app.js
 - JS dados_exercicio.js
 - JS dados_gemini.js
 - JS dados.js
 - index.html
- Inteligência Artificial Generativa.docx

js > JS app.js >

```
11 let section = document.getElementById("resultados-pesquisa");
12
13 console.log(section)
14
15 section.innerHTML =
16
17 <div class="item-resultado">
18
19 <h2><a href="#" target="_blank"></a> ${dados[0].titulo}</h2>
20
21 <p class="descricao-meta ">Rebeca Andrade é uma ginasta brasileira com suas performances incríveis. Campeã olímpica e multi campeã, é um símbolo de superação e determinação no esporte.</p>
22
23 <a href="https://pt.wikipedia.org/wiki/Rebeca_Andrade" target="_blank">Leia mais</a>
24
25 </div>
26
27
28
29
30
```

Ln 16, Col 1 Espaços: 4 UTF-8 CRLF () JavaScript ⚡ Port: 5500 ⚡ Go Live

Agora sim, puxou o 1º nome da instrução do arquivo dados.js



```
Arquivo Editar Seleção Ver Acessar ... ← → 🔍 Cursola
```

EXPLORADOR

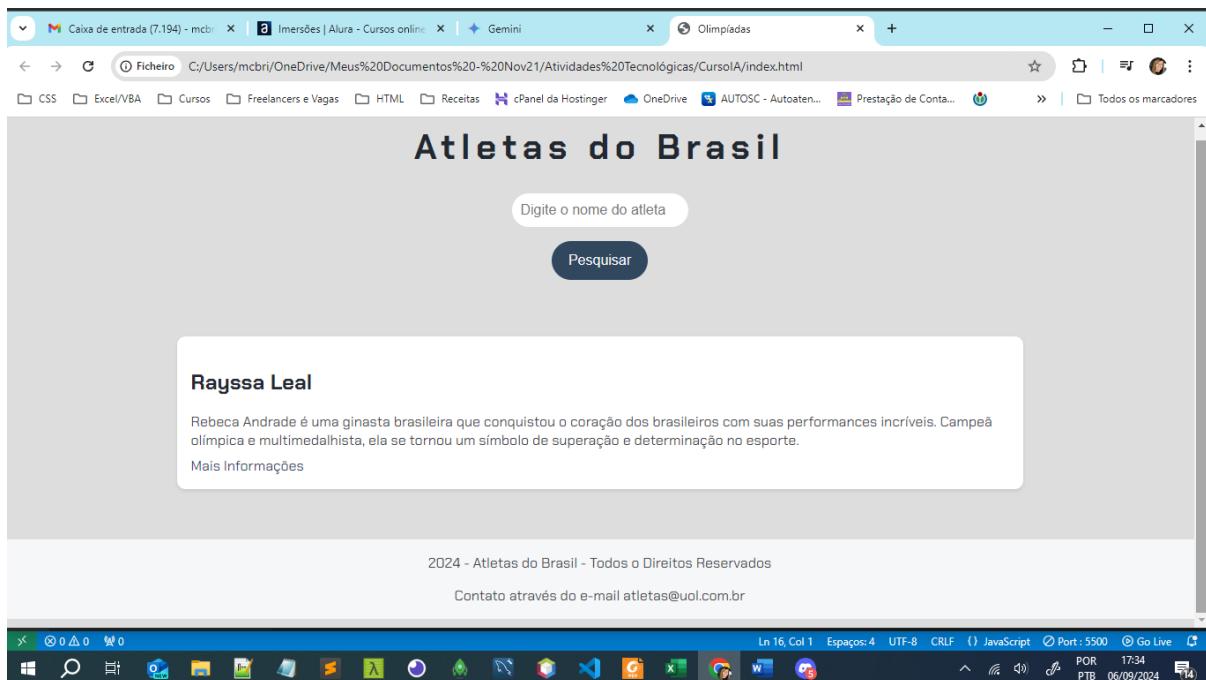
✓ CURSOIA

- ✓ arquivos baixados
 - > Aula1
 - > Aula2
 - > Aula3
 - > Aula4
- ✓ css
 - # style.css
- ✓ js
 - JS app_geminijs
 - JS app.js
 - JS dados_exercicio.js
 - JS dados_gemini.js
 - JS dados.js
 - index.html
- Inteligência Artificial Generativa.docx

js > JS dados.js > dados

```
1 let dados = [
2
3 {
4   titulo: "Rayssa Leal",
5   descricao: "Rayssa Leal, a Fadinha, é uma skatista brasileira que conquistou o mundo com seu talento e coragem.",
6   link: "https://pt.wikipedia.org/wiki/Rayssa_Leal",
7 }
8
9 {
10   titulo: "Rebeca Andrade",
11   descricao: "Rebeca Andrade é uma ginasta brasileira que conquistou o coração dos brasileiros com suas performances incríveis. Campeã olímpica e multi campeã, é um símbolo de superação e determinação no esporte.",
12   link: "https://pt.wikipedia.org/wiki/Rebeca_Andrade",
13 }
14
15 {
16   titulo: "Beatriz Souza",
17   descricao: "Beatriz Souza é uma judoca brasileira que conquistou a medalha de ouro nas Olimpíadas de Tóquio 2020 com sua técnica e força.",
18   link: "https://pt.wikipedia.org/wiki/Beatriz_Souza"
19 }
20
21
22
23
24
25
26
27 ]
```

Ln 1, Col 14 Espaços: 4 UTF-8 CRLF () JavaScript ⚡ Port: 5500 ⚡ Go Live



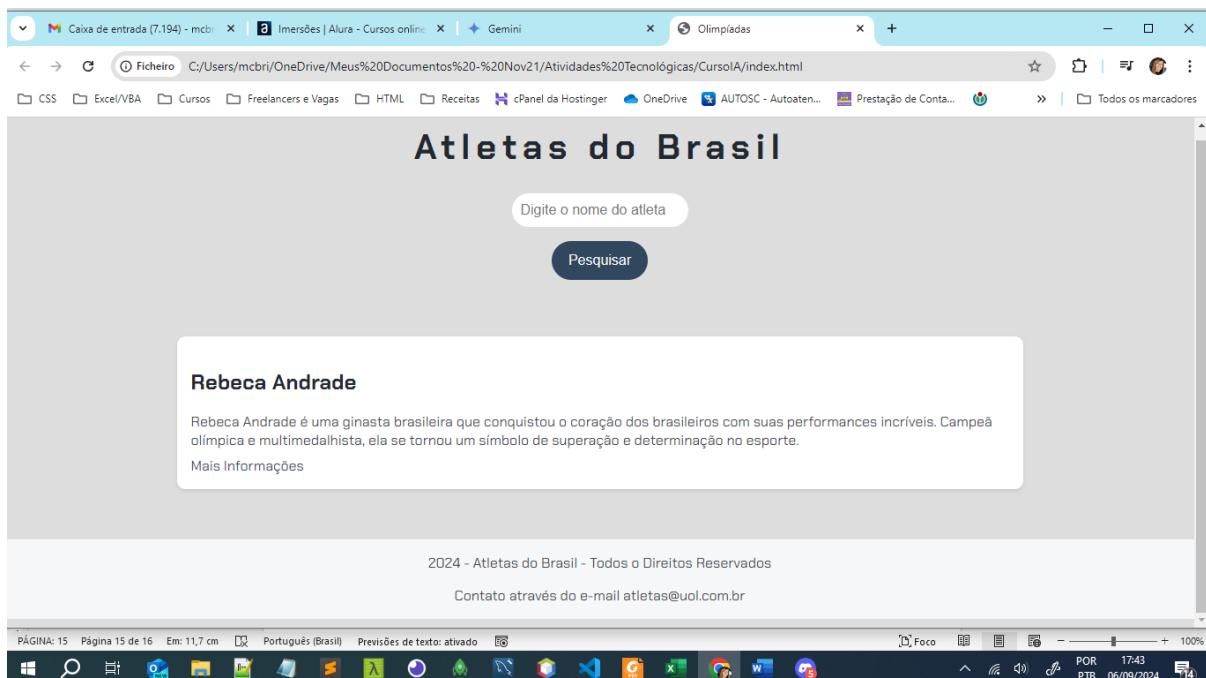
Se eu mudar para índice [1] vai para Rebeca

Entretanto a descrição não mudou junto com o nome.

Precisamos criar uma variável também para a descrição.

Assim:

```
js > JS app.js > ...
11 let section = document.getElementById("resultados-pesquisa");
12
13 console.log(section)
14
15 section.innerHTML =
16
17 <div class="item-resultado">
18
19 <h2><a href="#" target="_blank"></a> ${dados[1].titulo}</h2>
20
21 <p class="descricao-metá " ${dados[1].descricao}</p>
22
23 <a href="https://pt.wikipedia.org/wiki/Rebeca_Andrade" target="_blank"> Mais Informações</a>
24
25
26
27
28
```



Deu certo, mas precisamos alterar o índice nas duas linhas de comando, para todas as informações serem puxadas.

Agora vamos criar uma lista para apresentar todos os nomes de nosso arquivo. A lista vai fazer um looping enquanto encontrar valores para serem exibidos. Nesse caso, retiramos a parte que contém o índice, pois a instrução prevê que todos os itens sejam verificados.

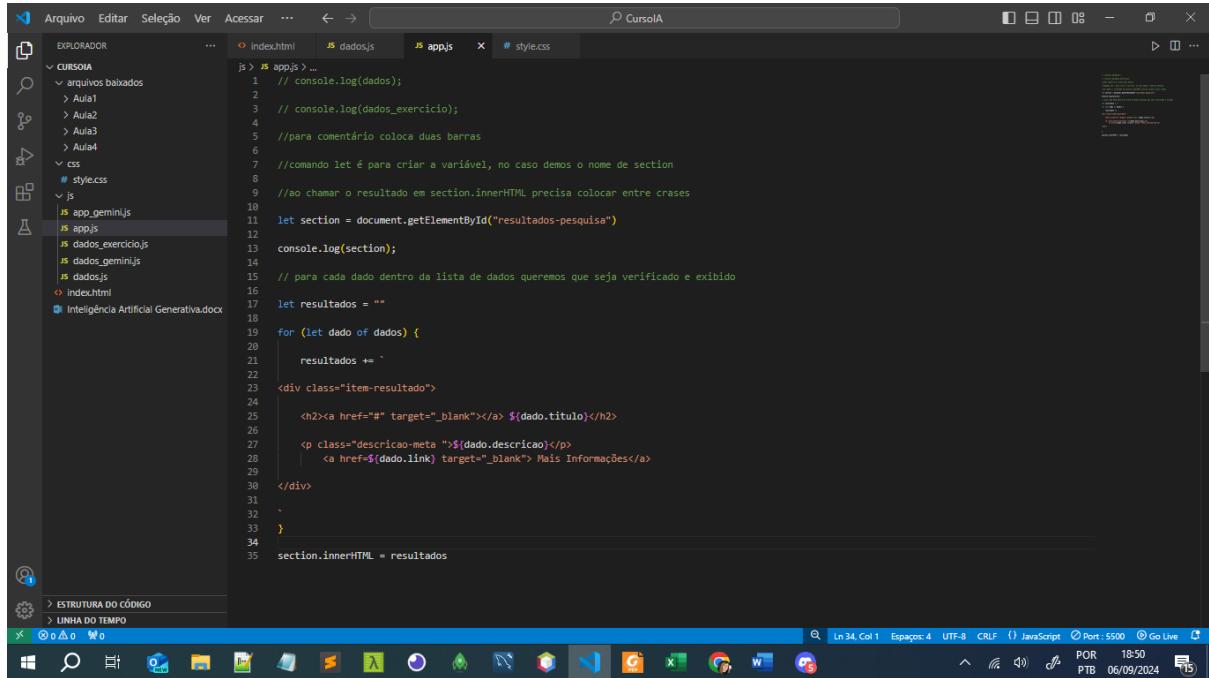
Usamos o comando `for` para a instrução(cada item é um dado da lista)

```
for (let dado of dados) {  
    section.innerHTML = `  
        <div class="item-resultado">  
            <h2><a href="#" target="_blank"></a> ${dado.titulo}</h2>  
            <p class="descricao-meta ">${dado.descricao}</p>  
            <a href=${dado.link} target="_blank"> Mais Informações</a>  
        </div>  
`  
}
```

Obs: acrescentamos o sinal + antes do = para exibir toda a lista, caso contrário só vai exibir a primeira.

Melhorando o código, criamos a variável resultados para receber todos os itens do arquivo dados.js.

Retiramos do looping a exibição, colocando após o término do for. Assim:

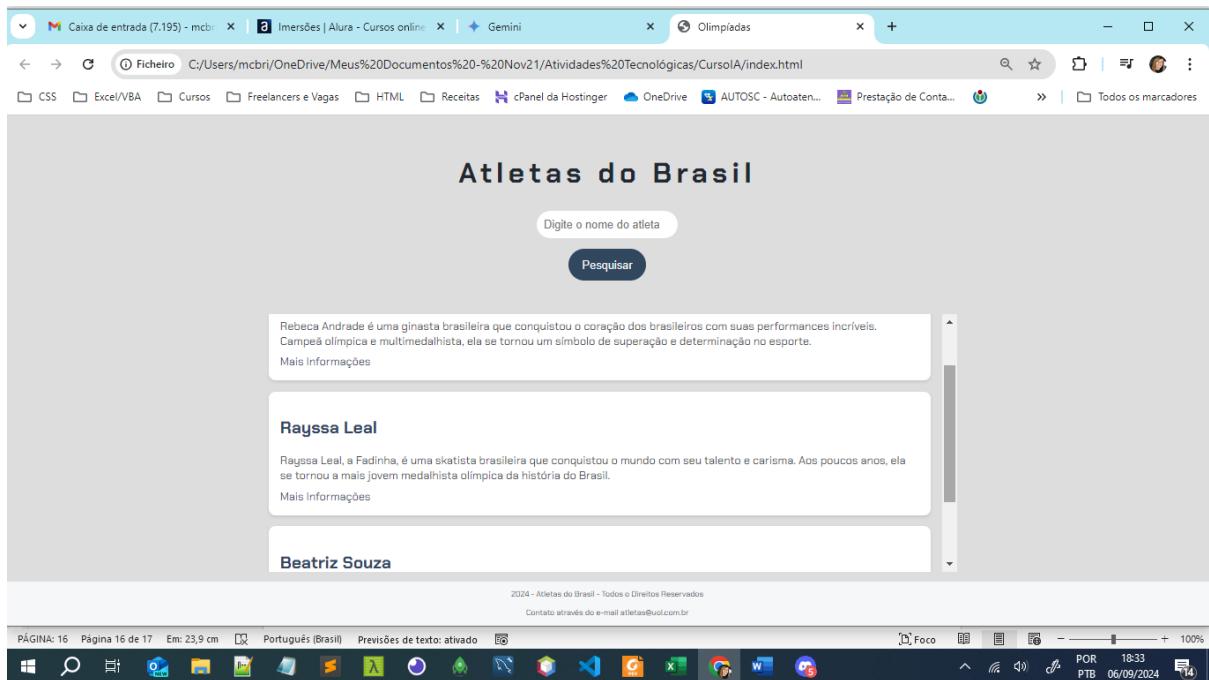


```
index.html JS dados.js JS app.js # style.css

CURSOIA
arquivos baixados
Aula1
Aula2
Aula3
Aula4
css
style.css
JS
app_gemini.js
app.js
dados_exercicio.js
dados_gemini.js
dados.js
index.html
Inteligencia Artificial Generativa.docx

js > JS app.js > ...
1 // console.log(dados);
2
3 // console.log(dados_exercicio);
4
5 //para comentário coloca duas barras
6
7 // comando let é para criar a variável, no caso demos o nome de section
8
9 // ao chamar o resultado em section.innerHTML precisa colocar entre crases
10
11 let section = document.getElementById("resultados-pesquisa")
12
13 console.log(section);
14
15 // para cada dado dentro da lista de dados queremos que seja verificado e exibido
16
17 let resultados = ""
18
19 for (let dado of dados) {
20
21     resultados += `
22
23 <div class="item-resultado">
24
25     <h2><a href="#" target="_blank"></a> ${dado.titulo}</h2>
26
27     <p class="descricao-meta ">${dado.descricao}</p>
28     | <a href=${dado.link} target="_blank"> Mais Informações</a>
29
30 </div>
31
32
33 }
34
35 section.innerHTML = resultados
```

Resultado



Atletas do Brasil

Digite o nome do atleta

Pesquisar

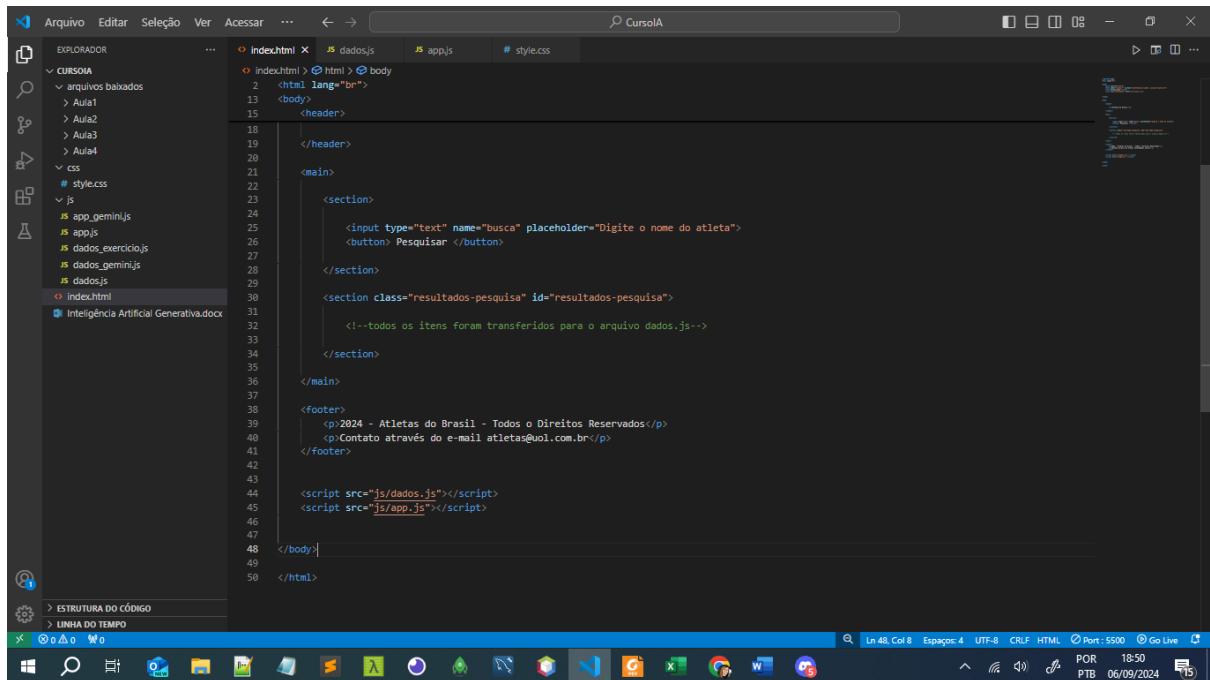
Rebeca Andrade é uma ginasta brasileira que conquistou o coração dos brasileiros com suas performances incríveis. Campeã olímpica e multimedalhista, ela se tornou um símbolo de superação e determinação no esporte.
[Mais Informações](#)

Rayssa Leal
Rayssa Leal, a Fadimba, é uma skatista brasileira que conquistou o mundo com seu talento e charme. Nos poucos anos, ela se tornou a mais jovem medalhista olímpica da história do Brasil.
[Mais Informações](#)

Beatriz Souza

2024 - Atletas do Brasil - Todos os Direitos Reservados
Contato através do e-mail atletas@uol.com.br

Dessa forma, podemos retirar todos os itens do código HTML, ficando somente no arquivo de dados.js. O HTML fica assim:



The screenshot shows the Microsoft Edge DevTools interface. The left sidebar displays the file structure under 'CURSOIA'. The main pane shows the HTML code for 'index.html' with line numbers. The code includes a header, a main section containing a search form, and a footer with copyright and contact information. Scripts are included via `<script>` tags pointing to 'dados.js', 'app.js', and 'style.css'. The bottom status bar shows the file path as 'CursolA', line 48, column 8, and other details like port 5500 and timestamp 06/09/2024.

```
<html lang="pt-BR">
  <body>
    <header>
      ...
    </header>
    <main>
      <section>
        <input type="text" name="busca" placeholder="Digite o nome do atleta" />
        <button> Pesquisar </button>
      </section>
      <section class="resultados-pesquisa" id="resultados-pesquisa">
        <!-- todos os itens foram transferidos para o arquivo dados.js -->
      </section>
    </main>
    <footer>
      <p>2024 - Atletas do Brasil - Todos os Direitos Reservados</p>
      <p>Contato através do e-mail atletas@uol.com.br</p>
    </footer>
    <script src="js/dados.js"></script>
    <script src="js/app.js"></script>
  </body>
</html>
```

Melhorando o código, queremos que apareçam os atletas somente depois de clicar no botão pesquisar; precisamos ajustar o button para que funcione.

Acrescentamos no HTML:

```
<section>
  <input type="text" name="busca" placeholder="Digite o nome do atleta" />
  <button onclick="pesquisar()"> Pesquisar </button>
</section>
```

The screenshot shows a code editor interface with two tabs open: 'index.html' and 'app.js'. The 'index.html' tab displays an HTML page with a search form. The 'app.js' tab shows a JavaScript function named 'pesquisar()' which iterates through data and generates HTML for each item.

```
Arquivo Editar Seleção Ver Acessar ... 🔍 Cursola index.html • JS app.js JS dados.js
EXPLORADOR
CURSOIA
arquivos baixados
Aula1
Aula2
Aula3
Aula4
Aula5
css
# style.css
JS
JS app_geminis.js
JS app.js
JS dados_exercicio.js
JS dados_geminis.js
JS dados.js
index.html
Inteligência Artificial Generativa.docx
ESTRUTURA DO CÓDIGO
LINHA DO TEMPO
index.html > html > body > main > section > button
2 <html lang="br">
4 <head>
11 </head>
12
13 <body>
14
15 <header>
16
17 <h1>Atletas do Brasil</h1>
18
19 </header>
20
21 <main>
22
23 <section>
24
25 <input type="text" name="busca" placeholder="Digite o nome do atleta" >
26 <button onclick="pesquisar()"> Pesquisar </button>
27
28 </section>
29
30 <section class="resultados-pesquisa" id="resultados-pesquisa">
31
32 <!--todos os itens foram transferidos para o arquivo dados.js-->
33
34
35 </main>
36
37 <footer>
38 <p>2024 - Atletas do Brasil - Todos os Direitos Reservados</p>
39 <p>Contato através do e-mail atletas@uol.com.br</p>
40
```

```
index.html > JS app.js > pesquisar()
6 // comando let é para criar uma variável, no caso demos o nome de section
7 // ao chamar o resultado em section.innerHTML precisa colocar entre crases
8
9
10 Function pesquisar() {
11
12   //console.log('ok') comando para aparecer se o comando está funcionando (Otávio)
13
14
15 let section = document.getElementById("resultados-pesquisa")
16
17 console.log(section);
18
19 // para cada dado dentro da lista de dados queremos que seja verificado e exibido
20
21 let resultados = ""
22
23 for (let dado of dados) {
24
25   resultados += `

26
27 <div class="item-resultado">
28
29   <h2><a href="#" target="_blank"></a> ${dado.titulo}</h2>
30
31   <p class="descricao-meta ">${dado.descricao}</p>
32   <a href=${dado.link} target="_blank"> Mais Informações</a>
33
34 </div>
35
36
37 }
38
39 resultados
40
41
42
43 }
```

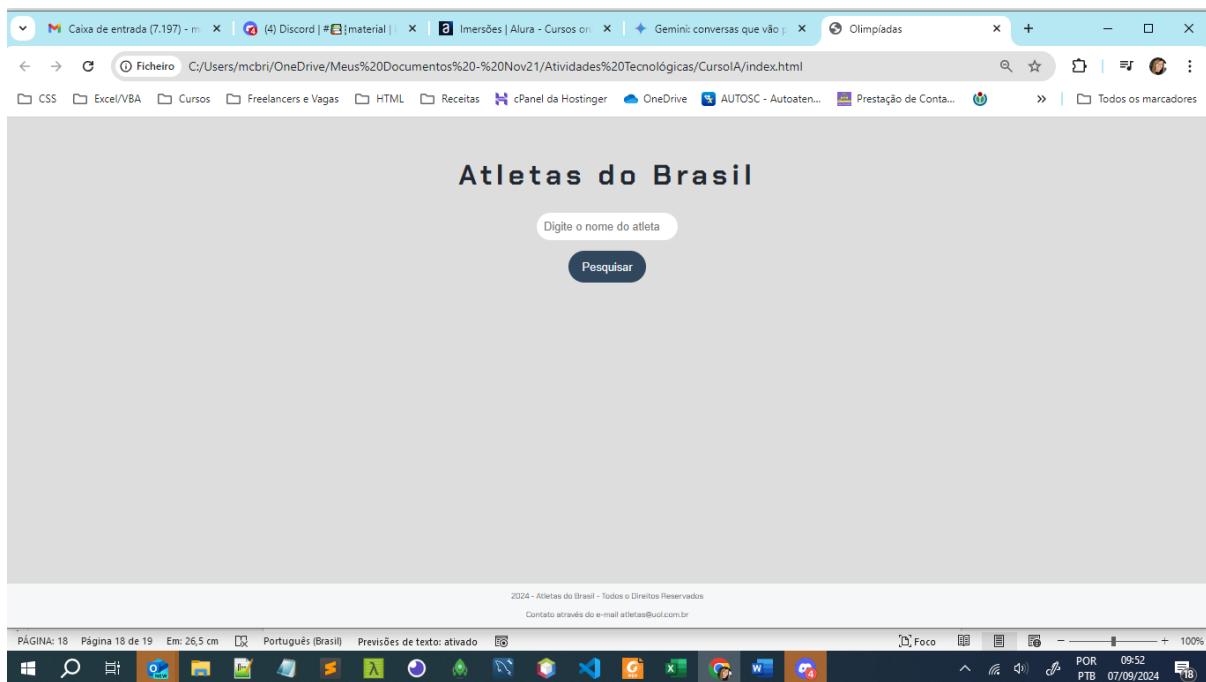
E acrescentamos no nosso app.js a function pesquisar() { englobando todo o nosso código anterior.

The screenshot shows the same code editor interface as before, but now the 'app.js' file contains the entire code from the previous screenshot, including the 'pesquisar()' function definition.

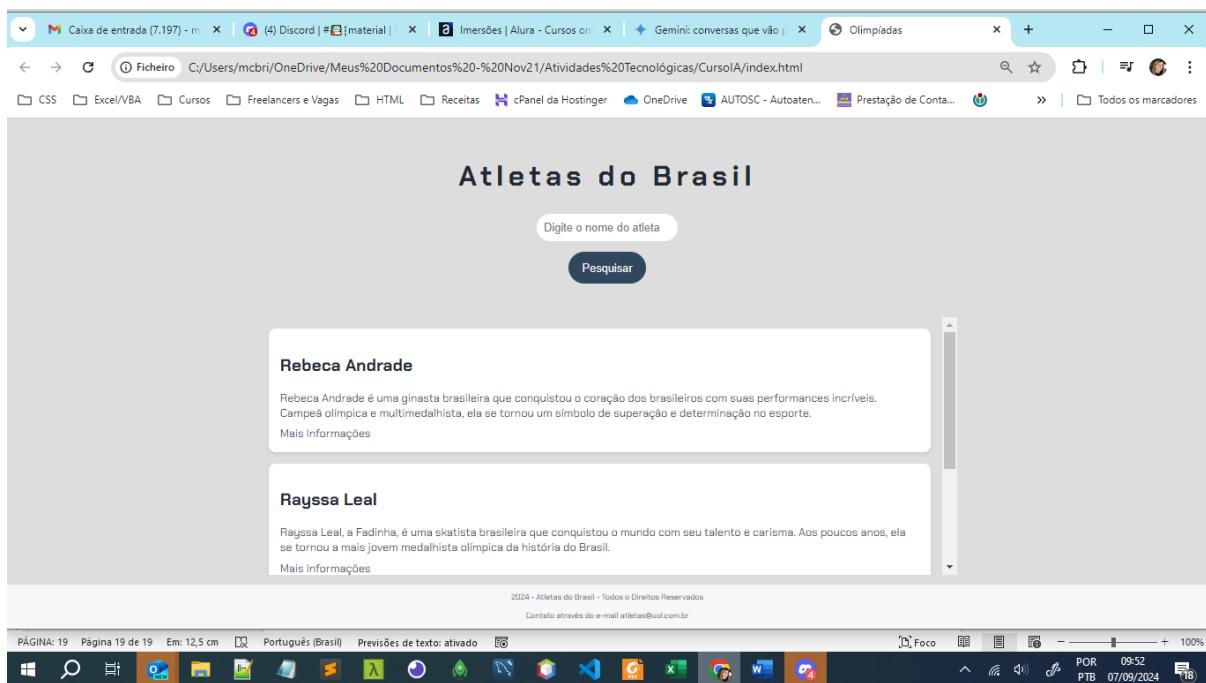
```
Arquivo Editar Seleção Ver Acessar ... 🔍 Cursola index.html • JS app.js JS dados.js
EXPLORADOR
CURSOIA
arquivos baixados
Aula1
Aula2
Aula3
Aula4
Aula5
css
# style.css
JS
JS app_geminis.js
JS app.js
JS dados_exercicio.js
JS dados_geminis.js
JS dados.js
index.html
Inteligência Artificial Generativa.docx
ESTRUTURA DO CÓDIGO
LINHA DO TEMPO
index.html > JS app.js > pesquisar()
6 // comando let é para criar uma variável, no caso demos o nome de section
7 // ao chamar o resultado em section.innerHTML precisa colocar entre crases
8
9
10 Function pesquisar() {
11
12   //console.log('ok') comando para aparecer se o comando está funcionando (Otávio)
13
14
15 let section = document.getElementById("resultados-pesquisa")
16
17 console.log(section);
18
19 // para cada dado dentro da lista de dados queremos que seja verificado e exibido
20
21 let resultados = ""
22
23 for (let dado of dados) {
24
25   resultados += `

26
27 <div class="item-resultado">
28
29   <h2><a href="#" target="_blank"></a> ${dado.titulo}</h2>
30
31   <p class="descricao-meta ">${dado.descricao}</p>
32   <a href=${dado.link} target="_blank"> Mais Informações</a>
33
34 </div>
35
36
37 }
38
39 resultados
40
41
42
43 }
```

O resultado é este



E quando clicamos em pesquisar aparecem os atletas



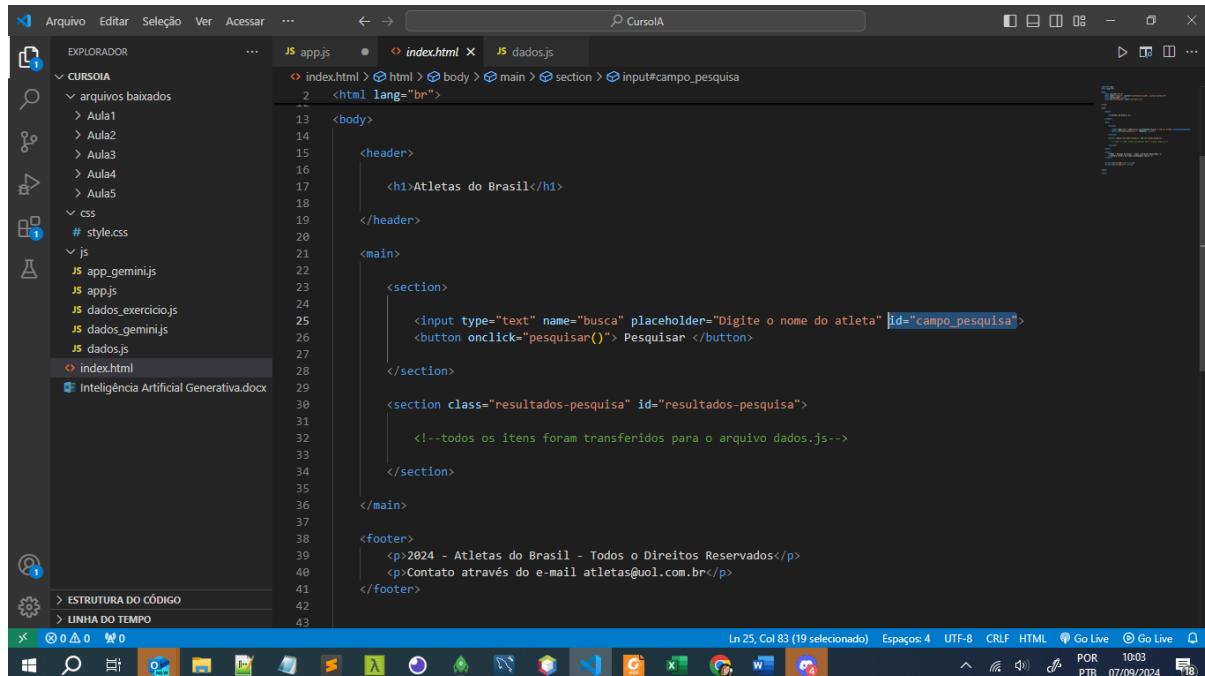
Agora vamos fazer o botão pesquisar com novas funcionalidades.

Exemplo pesquisa por nome, esporte...

Aula 5

Inicialmente vamos ao código html e incluímos um id no comando input:

```
id="campo_pesquisa"
```



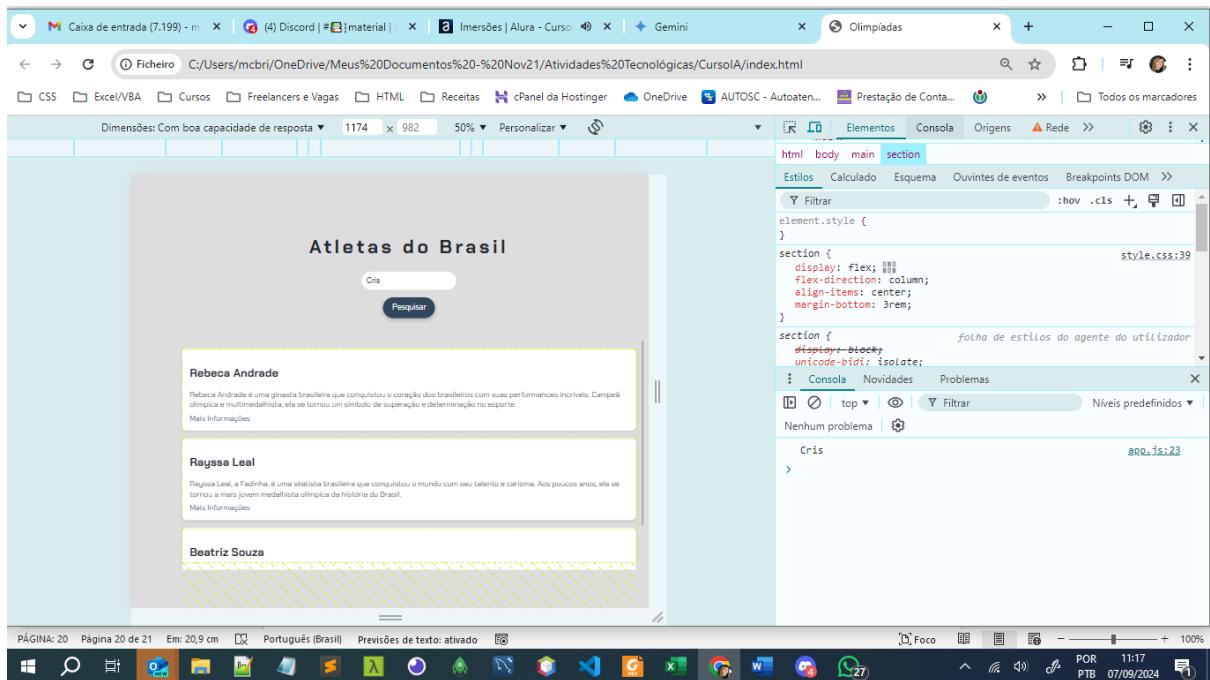
```
<html lang="pt-BR">
  <body>
    <header>
      <h1>Atletas do Brasil</h1>
    </header>
    <main>
      <section>
        <input type="text" name="busca" placeholder="Digite o nome do atleta" id="campo_pesquisa">
        <button onclick="pesquisar()">Pesquisar</button>
      </section>
      <section class="resultados-pesquisa" id="resultados-pesquisa">
        <!-- todos os itens foram transferidos para o arquivo dados.js -->
      </section>
    </main>
    <footer>
      <p>2024 - Atletas do Brasil - Todos os Direitos Reservados</p>
      <p>Contato através do e-mail atletas@uol.com.br</p>
    </footer>
  </body>
</html>
```

Vamos para o nosso app.js

Precisamos criar um novo comando para puxar as informações do id campo_pesquisa já criado.

Lembrando que no java script, em substantivos compostos o segundo elemento inicia com a letra maiúscula, ficando assim = campoPesquisa.

Dentro da function pesquisa, logo abaixo do comando que puxa o resultado da pesquisa incluído anteriormente colocamos: let campoPesquisa = document.getElementById("campo-pesquisa").value (queremos puxar o valor digitado dentro da caixa de texto)

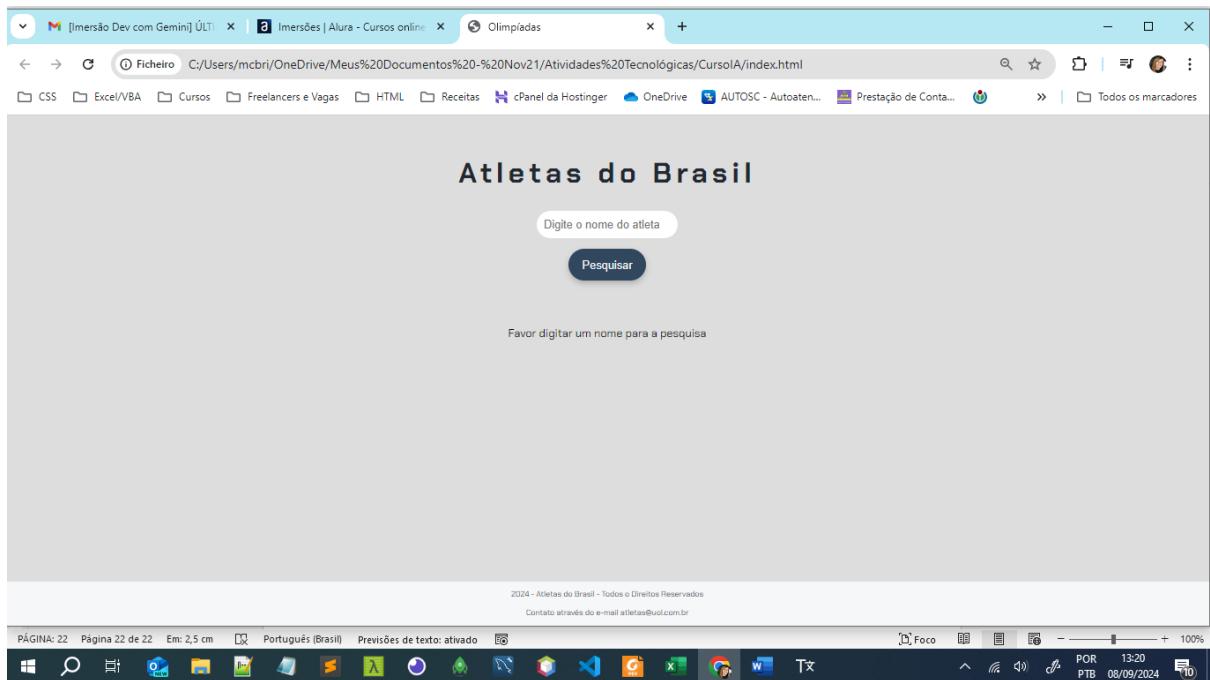


Agora precisamos comparar se o valor digitado pelo usuário existe na nossa relação de nomes de atletas. Vamos incluir a instrução na estrutura `for`, abaixo da variável `resultados`, criada para receber as informações

`dados.titulo.includes(campoPesquisa)`

```
if (campoPesquisa == "") {  
  
    section.innerHTML = "Favor digitar um nome para a pesquisa"  
  
    return  
}
```

```
function pesquisar() {  
  
    let section = document.getElementById("resultados-pesquisa");  
    let campoPesquisa = document.getElementById("campo_pesquisa").value;  
  
    if (campoPesquisa == "") {  
        section.innerHTML = "Favor digitar um nome para a pesquisa"  
        return  
    }  
  
    let resultados = "";  
  
    for (let dado of dados) {  
        if (dado.titulo.includes(campoPesquisa)) {  
            resultados += `...`;  
        }  
    }  
  
    section.innerHTML = resultados;  
}
```



Ampliando nossa pesquisa pelo tipo de esporte. Incluímos a instrução no campo abaixo

A screenshot of the Brackets IDE. The left sidebar shows a file tree with files like 'index.html', 'dados.js', 'dados_exercicio.js', 'dados_gemini.js', and 'dados.json'. The main editor pane shows the 'app.js' file with the following code:

```
function pesquisar() {
    // criamos e limpamos a variável resultados para receber os resultados
    let resultados = "";

    for (let dado of dados) {
        const mensagemDiv = document.getElementById("campo_pesquisa");

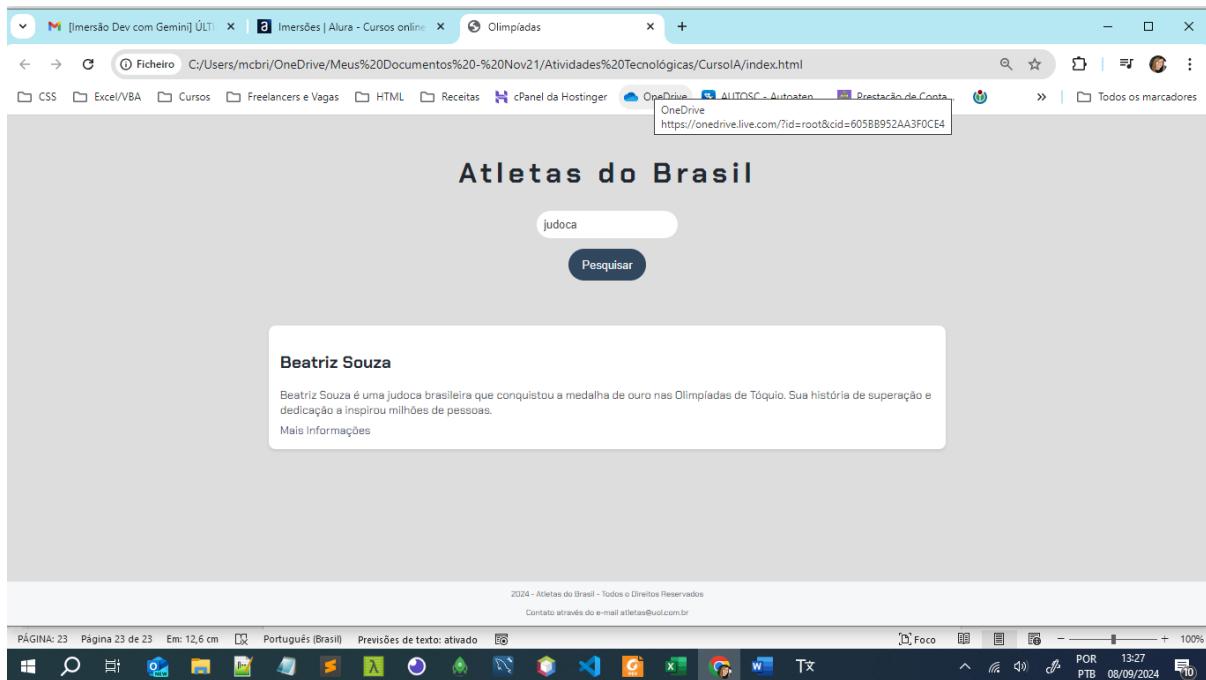
        // se o valor digitado no campo pesquisa for ok
        // então faça
        console.log(dado.titulo.includes(campoPesquisa)) //usado para testar o comando

        // no java script o ou corresponde a duas barras verticais
        if (campoPesquisa == dado.titulo || dado.descricao.includes(campoPesquisa)) {
            resultados += `

<div class="item-resultado">
    <h2><a href="#" target="_blank">${dado.titulo}</a></h2>
    <p class="descricao-meta ">${dado.descricao}</p>
    <a href=${dado.link} target="_blank">Mais Informações</a>
</div>
`;
        }
    }
}
```

The status bar at the bottom indicates the code is in JavaScript mode, has 96 selected characters, and is saved.

Resultado, consultando por esporte



Agora vamos deixar tudo minúsculo para não dar erro na hora da consulta.(de qualquer forma que digitar, a instrução vai entender) Do jeito que está, se digitarmos minúsculo, como no nosso arquivo dados.js está com a inicial maiúscula, a consulta não retornará nenhum resultado.

Fizemos algumas alterações no código, incluindo variáveis.

```

function pesquisar() {
  if (campoPesquisa == "") {
    section.innerHTML = "Favor digitar um nome para a pesquisa"
    return
  }

  // para cada dado dentro da lista de dados queremos que seja verificado e exibido
  let resultados = "";
  let titulo = "";
  let descricao = "";

  for (let dado of dados) {
    titulo = dado.titulo.toLowerCase();
    descricao = dado.descricao.toLowerCase();

    //se o valor digitado no campo pesquisa for ok
    //então faça
    console.log(descricao.includes(campoPesquisa)) //usado para testar o comando

    if (titulo.includes(campoPesquisa) || descricao.includes(campoPesquisa)) {
      resultados += `

Título: ${titulo}

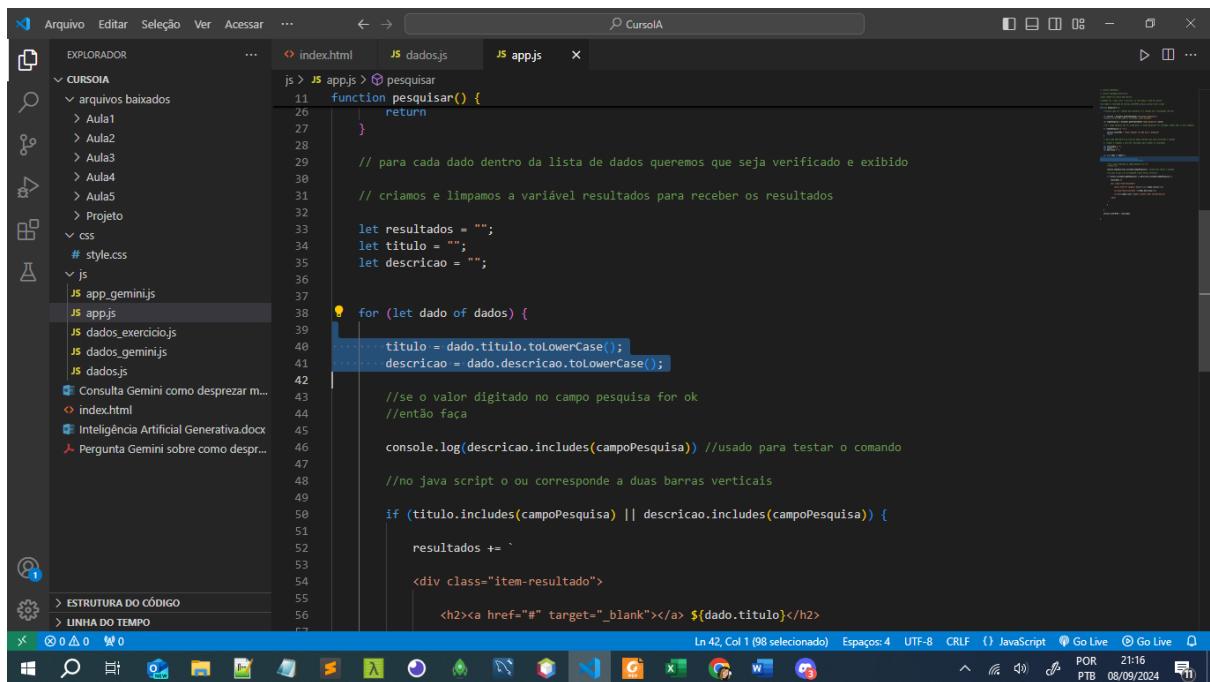


Descrição: ${descricao}

`;
    }
  }

  resultados.innerHTML = "Resultados da Pesquisa";
  section.innerHTML = resultados.innerHTML;
}

```



```
function pesquisar() {
    let resultados = "";
    let titulo = "";
    let descricao = "";

    for (let dado of dados) {
        titulo = dado.titulo.toLowerCase();
        descricao = dado.descricao.toLowerCase();

        //se o valor digitado no campo pesquisa for ok
        //então faça

        console.log(descricao.includes(campoPesquisa)) //usado para testar o comando

        //no java script o ou corresponde a duas barras verticais

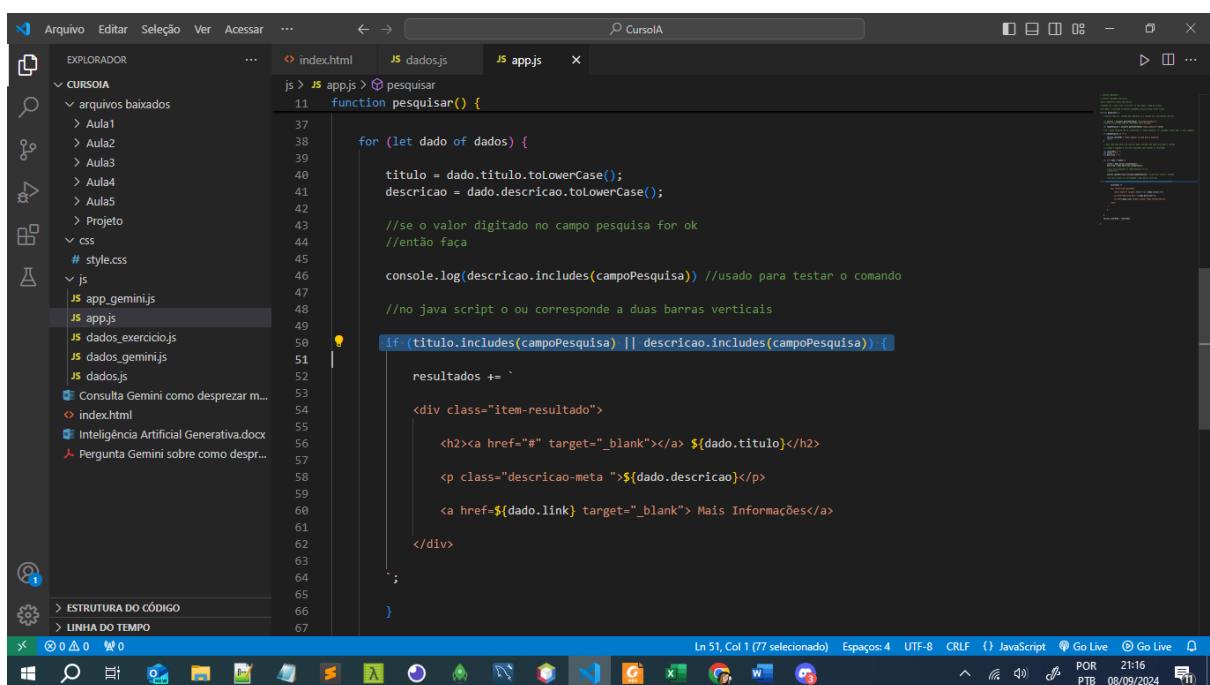
        if (titulo.includes(campoPesquisa) || descricao.includes(campoPesquisa)) {

            resultados += `

<div class="item-resultado">

<h2><a href="#" target="_blank">${dado.titulo}</h2>
<p class="descricao-meta ">${dado.descricao}</p>
<a href=${dado.link} target="_blank"> Mais Informações</a>

</div>
` ;
        }
    }
}
```



```
function pesquisar() {
    let resultados = "";
    let titulo = "";
    let descricao = "";

    for (let dado of dados) {
        titulo = dado.titulo.toLowerCase();
        descricao = dado.descricao.toLowerCase();

        //se o valor digitado no campo pesquisa for ok
        //então faça

        console.log(descricao.includes(campoPesquisa)) //usado para testar o comando

        //no java script o ou corresponde a duas barras verticais

        if (titulo.includes(campoPesquisa) || descricao.includes(campoPesquisa)) {

            resultados += `

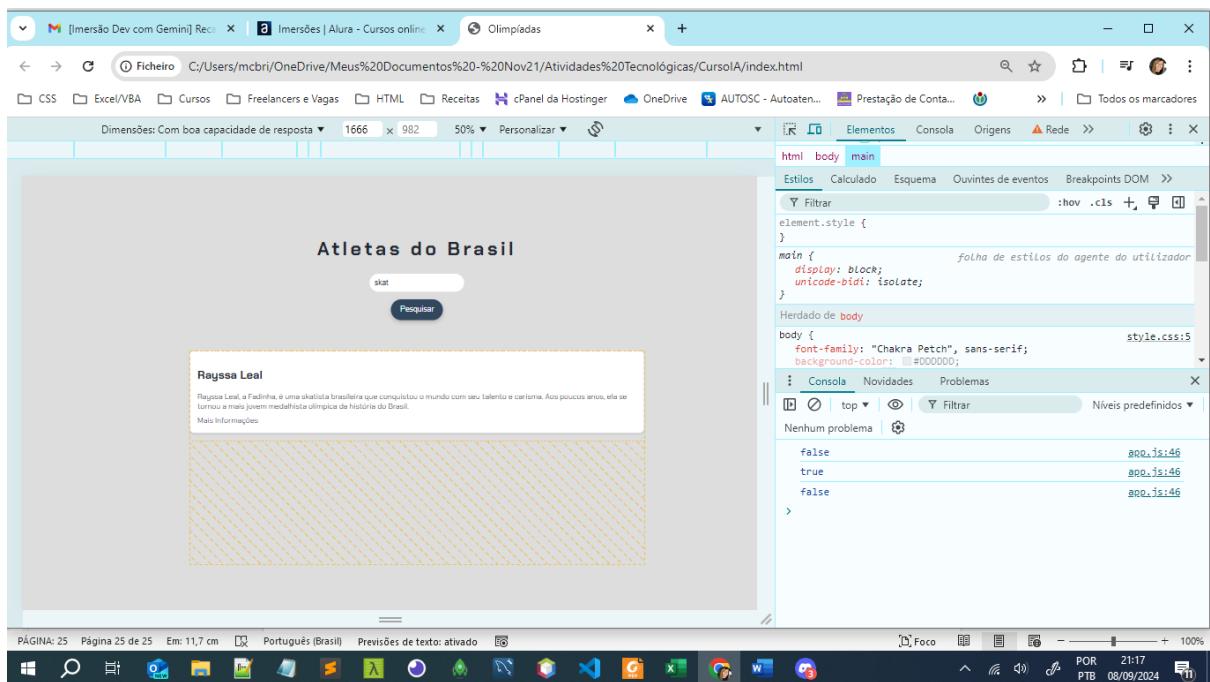
<div class="item-resultado">

<h2><a href="#" target="_blank">${dado.titulo}</h2>
<p class="descricao-meta ">${dado.descricao}</p>
<a href=${dado.link} target="_blank"> Mais Informações</a>

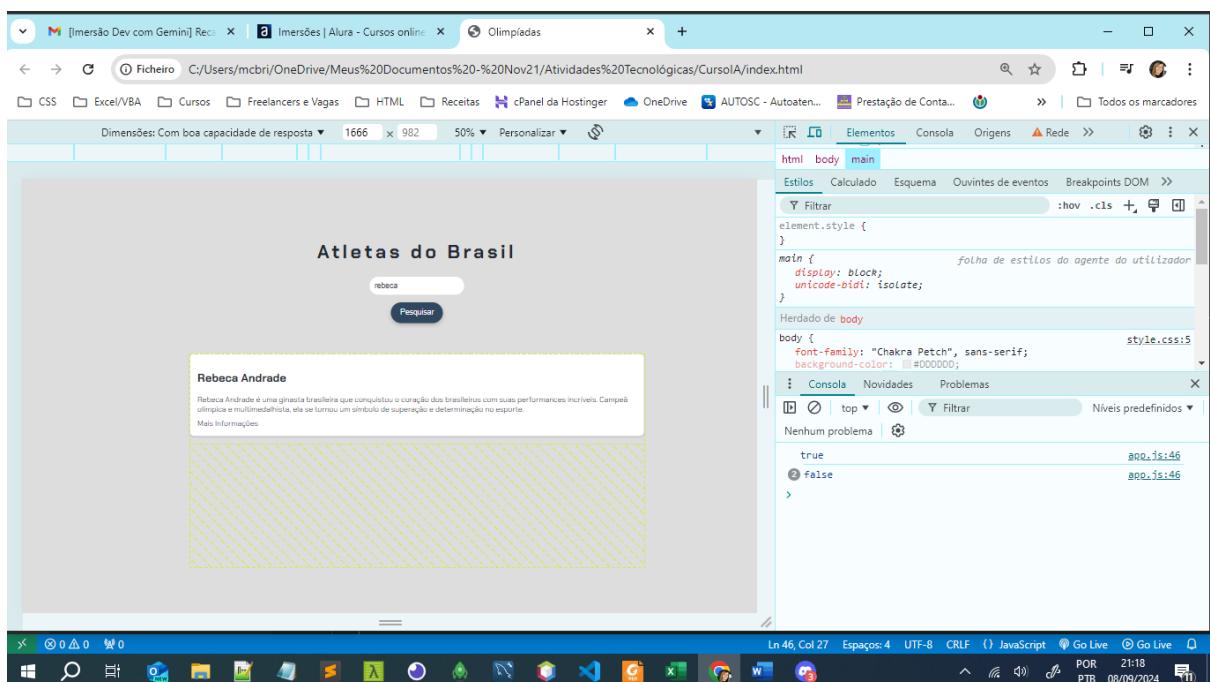
</div>
` ;
        }
    }
}
```

Resultado:

Consultando por esporte/descrição



Consultando por nome, minúsculo



Estava dando erro...consultei o Gemini.

Precisamos incluir mais uma linha no comando de declaração de variáveis. Orientação do Gemini

The screenshot shows a web browser window with the address bar pointing to gemini.google.com/app/22cbf49de7ea1b04?utm_source=website&utm_medium=referral&utm_campaign=alura_aug24. The main content area is titled "Gemini" and contains a "JavaScript" code snippet:

```
for (let dado of dados) {
    titulo = dado.titulo.toLowerCase();
    descricao = dado.descricao.toLowerCase();

    // Convertendo a string de pesquisa para minúsculas
    campoPesquisa = campoPesquisa.toLowerCase();

    // ... restante do seu código ...
}
```

Below the code, a message says "Use o código [com cuidado](#)". A "Explicação:" section is present, followed by a text input field with placeholder "Digite uma pergunta ou comando". A note at the bottom states: "Gemini pode apresentar informações imprecisas, inclusive sobre pessoas. Por isso, cheque as respostas. Sua privacidade e os apps do Gemini".

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The title bar includes icons for file operations, a search bar with "CursolA", and status indicators like "Ln 42, Col 53" and "JavaScript". The left sidebar shows a file tree with "CURSOIA" folder containing "arquivos baixados" (Aula1, Aula2, Aula3, Aula4, Aula5, Projeto), "css" (style.css), and "js" (app_geminis, app.js, dados_exercicio.js, dados_geminis, dados.js). The main editor pane displays the "app.js" file content:

```
function pesquisar() {
    // para cada dado dentro da lista de dados queremos que seja verificado e exibido
    let resultados = "";
    let titulo = "";
    let descricao = "";

    for (let dado of dados) {
        titulo = dado.titulo.toLowerCase();
        descricao = dado.descricao.toLowerCase();
        campoPesquisa = campoPesquisa.toLowerCase();

        // se o valor digitado no campo pesquisa for ok
        // então faça
        console.log(titulo.includes(campoPesquisa)) //usado para testar o comando
        //no java script o ou corresponde a duas barras verticais

        if (titulo.includes(campoPesquisa) || descricao.includes(campoPesquisa)) {
            resultados += `

<div class="itemResultado">
    <h2><a href="#" target="_blank"></a> ${dado.titulo}</h2>

```

The status bar at the bottom shows "Ln 42, Col 53 (45 selecionado)" and "JavaScript".

Agora, digitando de qualquer forma, sempre encontra o resultado.

Podemos também, criar umas tags no arquivo de dados.js para facilitar a busca . Assim:

The screenshot shows a Microsoft Edge browser window with the following details:

- Address Bar:** CursolA
- Left Sidebar (Explorador):** CURSOIA, arquivos baixados (Aula1, Aula2, Aula3, Aula4, Aula5, Projeto), css (# style.css), js (app_gemini.js, app.js, dados_exercicio.js, dados_gemini.js, dados.js).
- Content Area:** A code editor showing JavaScript code for `dados.js` and `app.js`.
- Bottom Taskbar:** Shows various pinned icons including Microsoft Edge, File Explorer, Task View, and others.

The code in `dados.js` is as follows:

```
js > JS dados.js > [dados]
1 let dados = [
2
3
4
5   titulo: "Rebeca Andrade",
6   descricao: "Rebeca Andrade é uma ginasta brasileira que conquistou o coração dos brasileiros com suas pe
7   link: "https://pt.wikipedia.org/wiki/Rebeca_Andrade",
8   tags: "ginastica, solo",
9
10 ]
11
12
13
14   titulo: "Rayssa Leal",
15   descricao: "Rayssa Leal, a Fadinha, é uma skatista brasileira que conquistou o mundo com seu talento e c
16   link: "https://pt.wikipedia.org/wiki/Rayssa_Leal",
17   tags: "skat, skatista, skate",
18
19 }
20
21
22
23   titulo: "Beatriz Souza",
24   descricao: "Beatriz Souza é uma judoca brasileira que conquistou a medalha de ouro nas Olimpíadas de Tóq
25   link: "https://pt.wikipedia.org/wiki/Beatriz_Souza",
26   tags: "judo, judô",
27
28 }
29
30
31 ]
32 ]
```

The status bar at the bottom indicates: Ln 10, Col 7 (344 selecionado) | Espaços: 4 | UTF-8 | CRLF | JavaScript | Go Live | Go Live | POR PTB 21:49 08/09/2024

E alterando o app.js

The screenshot shows a Microsoft Edge browser window with the following details:

- Address Bar:** CursolA
- Left Sidebar (Explorador):** CURSOIA, arquivos baixados (Aula1, Aula2, Aula3, Aula4, Aula5, Projeto), css (# style.css), js (app_gemini.js, app.js, dados_exercicio.js, dados_gemini.js, dados.js).
- Content Area:** A code editor showing JavaScript code for `app.js`.
- Bottom Taskbar:** Shows various pinned icons including Microsoft Edge, File Explorer, Task View, and others.

The code in `app.js` is as follows:

```
js > JS app.js > [pesquisar]
11 function pesquisar() {
29   // para cada dado dentro da lista de dados queremos que seja verificado e exibido
30
31   // criamos e limpamos a variável resultados para receber os resultados
32
33   let resultados = "";
34   let titulo = "";
35   let descricao = "";
36   let tags = "";
37
38
39   for (let dado of dados) {
40
41     let resultados = "";
42     let titulo = dado.titulo.toLowerCase();
43     let descricao = dado.descricao.toLowerCase();
44     let campoPesquisa = campoPesquisa.toLowerCase();
45     let tags = dado.tags.toLowerCase();
46
47     // se o valor digitado no campo pesquisa for ok
48     // então faça
49
50     console.log(descricao.includes(campoPesquisa)) // usado para testar o comando
51
52     // no java script o ou corresponde a duas barras verticais
53
54     if (titulo.includes(campoPesquisa) || descricao.includes(campoPesquisa) || tags.includes(campoPesquisa))
55
56       resultados += `

57         <div class="item-resultado">
58           <div>${dado.titulo}</div>
59           <div>${dado.descricao}</div>
60           <div>${dado.tags}</div>
61         </div>
62       `;
63
64     }
65
66   }
67
68   resultados += `

69     <div class="resultado">
70       <div>${resultados}</div>
71     </div>
72   `;
73
74   resultados += `

75     <div class="campo-pesquisa">
76       <input type="text" value="Search" placeholder="Search" />
77     </div>
78   `;
79
80   resultados += `

81     <div class="botao-pesquisar">
82       <button>Search</button>
83     </div>
84   `;
85
86   resultados += `

87     <div class="resultado">
88       <div>${resultados}</div>
89     </div>
90   `;
91
92   resultados += `

93     <div class="resultado">
94       <div>${resultados}</div>
95     </div>
96   `;
97
98   resultados += `

99     <div class="resultado">
100       <div>${resultados}</div>
101     </div>
102   `;
103
104   resultados += `

105     <div class="resultado">
106       <div>${resultados}</div>
107     </div>
108   `;
109
110   resultados += `

111     <div class="resultado">
112       <div>${resultados}</div>
113     </div>
114   `;
115
116   resultados += `

117     <div class="resultado">
118       <div>${resultados}</div>
119     </div>
120   `;
121
122   resultados += `

123     <div class="resultado">
124       <div>${resultados}</div>
125     </div>
126   `;
127
128   resultados += `

129     <div class="resultado">
130       <div>${resultados}</div>
131     </div>
132   `;
133
134   resultados += `

135     <div class="resultado">
136       <div>${resultados}</div>
137     </div>
138   `;
139
140   resultados += `

141     <div class="resultado">
142       <div>${resultados}</div>
143     </div>
144   `;
145
146   resultados += `

147     <div class="resultado">
148       <div>${resultados}</div>
149     </div>
150   `;
151
152   resultados += `

153     <div class="resultado">
154       <div>${resultados}</div>
155     </div>
156   `;
157
158   resultados += `

159     <div class="resultado">
160       <div>${resultados}</div>
161     </div>
162   `;
163
164   resultados += `

165     <div class="resultado">
166       <div>${resultados}</div>
167     </div>
168   `;
169
170   resultados += `

171     <div class="resultado">
172       <div>${resultados}</div>
173     </div>
174   `;
175
176   resultados += `

177     <div class="resultado">
178       <div>${resultados}</div>
179     </div>
180   `;
181
182   resultados += `

183     <div class="resultado">
184       <div>${resultados}</div>
185     </div>
186   `;
187
188   resultados += `

189     <div class="resultado">
190       <div>${resultados}</div>
191     </div>
192   `;
193
194   resultados += `

195     <div class="resultado">
196       <div>${resultados}</div>
197     </div>
198   `;
199
200   resultados += `

201     <div class="resultado">
202       <div>${resultados}</div>
203     </div>
204   `;
205
206   resultados += `

207     <div class="resultado">
208       <div>${resultados}</div>
209     </div>
210   `;
211
212   resultados += `

213     <div class="resultado">
214       <div>${resultados}</div>
215     </div>
216   `;
217
218   resultados += `

219     <div class="resultado">
220       <div>${resultados}</div>
221     </div>
222   `;
223
224   resultados += `

225     <div class="resultado">
226       <div>${resultados}</div>
227     </div>
228   `;
229
230   resultados += `

231     <div class="resultado">
232       <div>${resultados}</div>
233     </div>
234   `;
235
236   resultados += `

237     <div class="resultado">
238       <div>${resultados}</div>
239     </div>
240   `;
241
242   resultados += `

243     <div class="resultado">
244       <div>${resultados}</div>
245     </div>
246   `;
247
248   resultados += `

249     <div class="resultado">
250       <div>${resultados}</div>
251     </div>
252   `;
253
254   resultados += `

255     <div class="resultado">
256       <div>${resultados}</div>
257     </div>
258   `;
259
260   resultados += `

261     <div class="resultado">
262       <div>${resultados}</div>
263     </div>
264   `;
265
266   resultados += `

267     <div class="resultado">
268       <div>${resultados}</div>
269     </div>
270   `;
271
272   resultados += `

273     <div class="resultado">
274       <div>${resultados}</div>
275     </div>
276   `;
277
278   resultados += `

279     <div class="resultado">
280       <div>${resultados}</div>
281     </div>
282   `;
283
284   resultados += `

285     <div class="resultado">
286       <div>${resultados}</div>
287     </div>
288   `;
289
290   resultados += `

291     <div class="resultado">
292       <div>${resultados}</div>
293     </div>
294   `;
295
296   resultados += `

297     <div class="resultado">
298       <div>${resultados}</div>
299     </div>
300   `;
301
302   resultados += `

303     <div class="resultado">
304       <div>${resultados}</div>
305     </div>
306   `;
307
308   resultados += `

309     <div class="resultado">
310       <div>${resultados}</div>
311     </div>
312   `;
313
314   resultados += `

315     <div class="resultado">
316       <div>${resultados}</div>
317     </div>
318   `;
319
320   resultados += `

321     <div class="resultado">
322       <div>${resultados}</div>
323     </div>
324   `;
325
326   resultados += `

327     <div class="resultado">
328       <div>${resultados}</div>
329     </div>
330   `;
331
332   resultados += `

333     <div class="resultado">
334       <div>${resultados}</div>
335     </div>
336   `;
337
338   resultados += `

339     <div class="resultado">
340       <div>${resultados}</div>
341     </div>
342   `;
343
344   resultados += `

345     <div class="resultado">
346       <div>${resultados}</div>
347     </div>
348   `;
349
350   resultados += `

351     <div class="resultado">
352       <div>${resultados}</div>
353     </div>
354   `;
355
356   resultados += `

357     <div class="resultado">
358       <div>${resultados}</div>
359     </div>
360   `;
361
362   resultados += `

363     <div class="resultado">
364       <div>${resultados}</div>
365     </div>
366   `;
367
368   resultados += `

369     <div class="resultado">
370       <div>${resultados}</div>
371     </div>
372   `;
373
374   resultados += `

375     <div class="resultado">
376       <div>${resultados}</div>
377     </div>
378   `;
379
380   resultados += `

381     <div class="resultado">
382       <div>${resultados}</div>
383     </div>
384   `;
385
386   resultados += `

387     <div class="resultado">
388       <div>${resultados}</div>
389     </div>
390   `;
391
392   resultados += `

393     <div class="resultado">
394       <div>${resultados}</div>
395     </div>
396   `;
397
398   resultados += `

399     <div class="resultado">
400       <div>${resultados}</div>
401     </div>
402   `;
403
404   resultados += `

405     <div class="resultado">
406       <div>${resultados}</div>
407     </div>
408   `;
409
410   resultados += `

411     <div class="resultado">
412       <div>${resultados}</div>
413     </div>
414   `;
415
416   resultados += `

417     <div class="resultado">
418       <div>${resultados}</div>
419     </div>
420   `;
421
422   resultados += `

423     <div class="resultado">
424       <div>${resultados}</div>
425     </div>
426   `;
427
428   resultados += `

429     <div class="resultado">
430       <div>${resultados}</div>
431     </div>
432   `;
433
434   resultados += `

435     <div class="resultado">
436       <div>${resultados}</div>
437     </div>
438   `;
439
440   resultados += `

441     <div class="resultado">
442       <div>${resultados}</div>
443     </div>
444   `;
445
446   resultados += `

447     <div class="resultado">
448       <div>${resultados}</div>
449     </div>
450   `;
451
452   resultados += `

453     <div class="resultado">
454       <div>${resultados}</div>
455     </div>
456   `;
457
458   resultados += `

459     <div class="resultado">
460       <div>${resultados}</div>
461     </div>
462   `;
463
464   resultados += `

465     <div class="resultado">
466       <div>${resultados}</div>
467     </div>
468   `;
469
470   resultados += `

471     <div class="resultado">
472       <div>${resultados}</div>
473     </div>
474   `;
475
476   resultados += `

477     <div class="resultado">
478       <div>${resultados}</div>
479     </div>
480   `;
481
482   resultados += `

483     <div class="resultado">
484       <div>${resultados}</div>
485     </div>
486   `;
487
488   resultados += `

489     <div class="resultado">
490       <div>${resultados}</div>
491     </div>
492   `;
493
494   resultados += `

495     <div class="resultado">
496       <div>${resultados}</div>
497     </div>
498   `;
499
500   resultados += `

501     <div class="resultado">
502       <div>${resultados}</div>
503     </div>
504   `;
505
506   resultados += `

507     <div class="resultado">
508       <div>${resultados}</div>
509     </div>
510   `;
511
512   resultados += `

513     <div class="resultado">
514       <div>${resultados}</div>
515     </div>
516   `;
517
518   resultados += `

519     <div class="resultado">
520       <div>${resultados}</div>
521     </div>
522   `;
523
524   resultados += `

525     <div class="resultado">
526       <div>${resultados}</div>
527     </div>
528   `;
529
530   resultados += `

531     <div class="resultado">
532       <div>${resultados}</div>
533     </div>
534   `;
535
536   resultados += `

537     <div class="resultado">
538       <div>${resultados}</div>
539     </div>
540   `;
541
542   resultados += `

543     <div class="resultado">
544       <div>${resultados}</div>
545     </div>
546   `;
547
548   resultados += `

549     <div class="resultado">
550       <div>${resultados}</div>
551     </div>
552   `;
553
554   resultados += `

555     <div class="resultado">
556       <div>${resultados}</div>
557     </div>
558   `;
559
560   resultados += `

561     <div class="resultado">
562       <div>${resultados}</div>
563     </div>
564   `;
565
566   resultados += `

567     <div class="resultado">
568       <div>${resultados}</div>
569     </div>
570   `;
571
572   resultados += `

573     <div class="resultado">
574       <div>${resultados}</div>
575     </div>
576   `;
577
578   resultados += `

579     <div class="resultado">
580       <div>${resultados}</div>
581     </div>
582   `;
583
584   resultados += `

585     <div class="resultado">
586       <div>${resultados}</div>
587     </div>
588   `;
589
590   resultados += `

591     <div class="resultado">
592       <div>${resultados}</div>
593     </div>
594   `;
595
596   resultados += `

597     <div class="resultado">
598       <div>${resultados}</div>
599     </div>
600   `;
601
602   resultados += `

603     <div class="resultado">
604       <div>${resultados}</div>
605     </div>
606   `;
607
608   resultados += `

609     <div class="resultado">
610       <div>${resultados}</div>
611     </div>
612   `;
613
614   resultados += `

615     <div class="resultado">
616       <div>${resultados}</div>
617     </div>
618   `;
619
620   resultados += `

621     <div class="resultado">
622       <div>${resultados}</div>
623     </div>
624   `;
625
626   resultados += `

627     <div class="resultado">
628       <div>${resultados}</div>
629     </div>
630   `;
631
632   resultados += `

633     <div class="resultado">
634       <div>${resultados}</div>
635     </div>
636   `;
637
638   resultados += `

639     <div class="resultado">
640       <div>${resultados}</div>
641     </div>
642   `;
643
644   resultados += `

645     <div class="resultado">
646       <div>${resultados}</div>
647     </div>
648   `;
649
650   resultados += `

651     <div class="resultado">
652       <div>${resultados}</div>
653     </div>
654   `;
655
656   resultados += `

657     <div class="resultado">
658       <div>${resultados}</div>
659     </div>
660   `;
661
662   resultados += `

663     <div class="resultado">
664       <div>${resultados}</div>
665     </div>
666   `;
667
668   resultados += `

669     <div class="resultado">
670       <div>${resultados}</div>
671     </div>
672   `;
673
674   resultados += `

675     <div class="resultado">
676       <div>${resultados}</div>
677     </div>
678   `;
679
680   resultados += `

681     <div class="resultado">
682       <div>${resultados}</div>
683     </div>
684   `;
685
686   resultados += `

687     <div class="resultado">
688       <div>${resultados}</div>
689     </div>
690   `;
691
692   resultados += `

693     <div class="resultado">
694       <div>${resultados}</div>
695     </div>
696   `;
697
698   resultados += `

699     <div class="resultado">
700       <div>${resultados}</div>
701     </div>
702   `;
703
704   resultados += `

705     <div class="resultado">
706       <div>${resultados}</div>
707     </div>
708   `;
709
710   resultados += `

711     <div class="resultado">
712       <div>${resultados}</div>
713     </div>
714   `;
715
716   resultados += `

717     <div class="resultado">
718       <div>${resultados}</div>
719     </div>
720   `;
721
722   resultados += `

723     <div class="resultado">
724       <div>${resultados}</div>
725     </div>
726   `;
727
728   resultados += `

729     <div class="resultado">
730       <div>${resultados}</div>
731     </div>
732   `;
733
734   resultados += `

735     <div class="resultado">
736       <div>${resultados}</div>
737     </div>
738   `;
739
740   resultados += `

741     <div class="resultado">
742       <div>${resultados}</div>
743     </div>
744   `;
745
746   resultados += `

747     <div class="resultado">
748       <div>${resultados}</div>
749     </div>
750   `;
751
752   resultados += `

753     <div class="resultado">
754       <div>${resultados}</div>
755     </div>
756   `;
757
758   resultados += `

759     <div class="resultado">
760       <div>${resultados}</div>
761     </div>
762   `;
763
764   resultados += `

765     <div class="resultado">
766       <div>${resultados}</div>
767     </div>
768   `;
769
770   resultados += `

771     <div class="resultado">
772       <div>${resultados}</div>
773     </div>
774   `;
775
776   resultados += `

777     <div class="resultado">
778       <div>${resultados}</div>
779     </div>
780   `;
781
782   resultados += `

783     <div class="resultado">
784       <div>${resultados}</div>
785     </div>
786   `;
787
788   resultados += `

789     <div class="resultado">
790       <div>${resultados}</div>
791     </div>
792   `;
793
794   resultados += `

795     <div class="resultado">
796       <div>${resultados}</div>
797     </div>
798   `;
799
800   resultados += `

801     <div class="resultado">
802       <div>${resultados}</div>
803     </div>
804   `;
805
806   resultados += `

807     <div class="resultado">
808       <div>${resultados}</div>
809     </div>
810   `;
811
812   resultados += `

813     <div class="resultado">
814       <div>${resultados}</div>
815     </div>
816   `;
817
818   resultados += `

819     <div class="resultado">
820       <div>${resultados}</div>
821     </div>
822   `;
823
824   resultados += `

825     <div class="resultado">
826       <div>${resultados}</div>
827     </div>
828   `;
829
830   resultados += `

831     <div class="resultado">
832       <div>${resultados}</div>
833     </div>
834   `;
835
836   resultados += `

837     <div class="resultado">
838       <div>${resultados}</div>
839     </div>
840   `;
841
842   resultados += `

843     <div class="resultado">
844       <div>${resultados}</div>
845     </div>
846   `;
847
848   resultados += `

849     <div class="resultado">
850       <div>${resultados}</div>
851     </div>
852   `;
853
854   resultados += `

855     <div class="resultado">
856       <div>${resultados}</div>
857     </div>
858   `;
859
860   resultados += `

861     <div class="resultado">
862       <div>${resultados}</div>
863     </div>
864   `;
865
866   resultados += `

867     <div class="resultado">
868       <div>${resultados}</div>
869     </div>
870   `;
871
872   resultados += `

873     <div class="resultado">
874       <div>${resultados}</div>
875     </div>
876   `;
877
878   resultados += `

879     <div class="resultado">
880       <div>${resultados}</div>
881     </div>
882   `;
883
884   resultados += `

885     <div class="resultado">
886       <div>${resultados}</div>
887     </div>
888   `;
889
890   resultados += `

891     <div class="resultado">
892       <div>${resultados}</div>
893     </div>
894   `;
895
896   resultados += `

897     <div class="resultado">
898       <div>${resultados}</div>
899     </div>
900   `;
901
902   resultados += `

903     <div class="resultado">
904       <div>${resultados}</div>
905     </div>
906   `;
907
908   resultados += `

909     <div class="resultado">
910       <div>${resultados}</div>
911     </div>
912   `;
913
914   resultados += `

915     <div class="resultado">
916       <div>${resultados}</div>
917     </div>
918   `;
919
920   resultados += `

921     <div class="resultado">
922       <div>${resultados}</div>
923     </div>
924   `;
925
926   resultados += `

927     <div class="resultado">
928       <div>${resultados}</div>
929     </div>
930   `;
931
932   resultados += `

933     <div class="resultado">
934       <div>${resultados}</div>
935     </div>
936   `;
937
938   resultados += `

939     <div class="resultado">
940       <div>${resultados}</div>
941     </div>
942   `;
943
944   resultados += `

945     <div class="resultado">
946       <div>${resultados}</div>
947     </div>
948   `;
949
950   resultados += `

951     <div class="resultado">
952       <div>${resultados}</div>
953     </div>
954   `;
955
956   resultados += `

957     <div class="resultado">
958       <div>${resultados}</div>
959     </div>
960   `;
961
962   resultados += `

963     <div class="resultado">
964       <div>${resultados}</div>
965     </div>
966   `;
967
968   resultados += `

969     <div class="resultado">
970       <div>${resultados}</div>
971     </div>
972   `;
973
974   resultados += `

975     <div class="resultado">
976       <div>${resultados}</div>
977     </div>
978   `;
979
980   resultados += `

981     <div class="resultado">
982       <div>${resultados}</div>
983     </div>
984   `;
985
986   resultados += `

987     <div class="resultado">
988       <div>${resultados}</div>
989     </div>
990   `;
991
992   resultados += `

993     <div class="resultado">
994       <div>${resultados}</div>
995     </div>
996   `;
997
998   resultados += `

999     <div class="resultado">
1000       <div>${resultados}</div>
1001     </div>
1002   `;
1003
1004   resultados += `

1005     <div class="resultado">
1006       <div>${resultados}</div>
1007     </div>
1008   `;
1009
1010   resultados += `

1011     <div class="resultado">
1012       <div>${resultados}</div>
1013     </div>
1014   `;
1015
1016   resultados += `

1017     <div class="resultado">
1018       <div>${resultados}</div>
1019     </div>
1020   `;
1021
1022   resultados += `

1023     <div class="resultado">
1024       <div>${resultados}</div>
1025     </div>
1026   `;
1027
1028   resultados += `

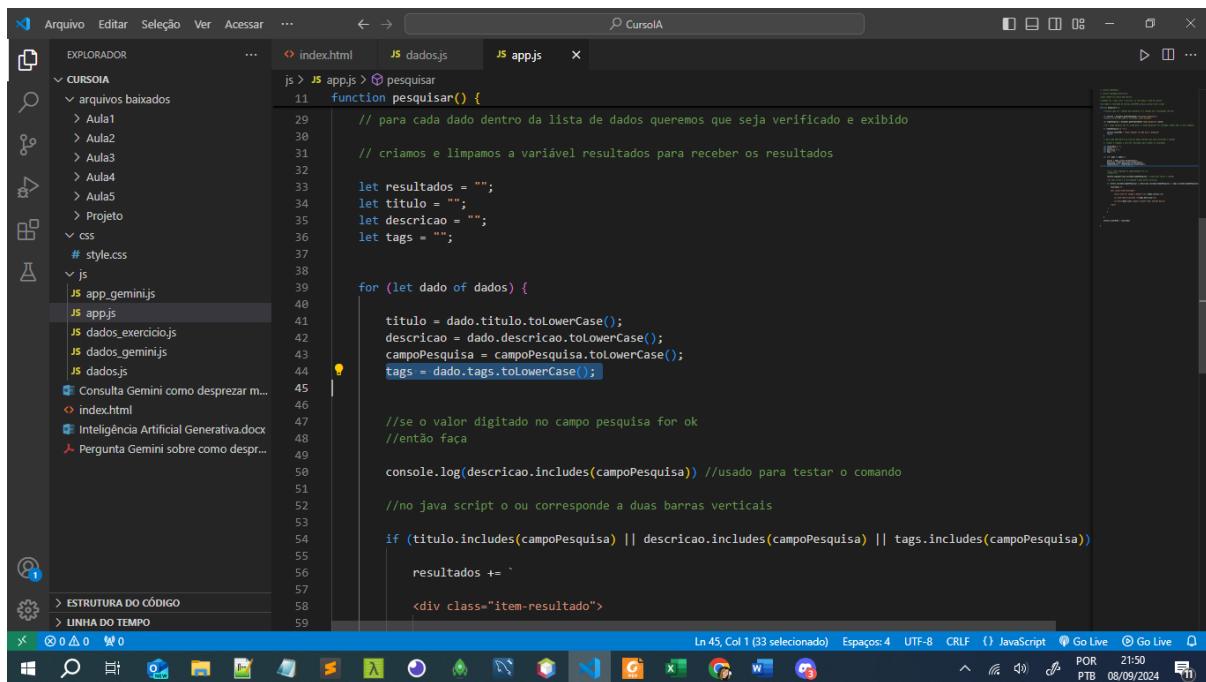
1029     <div class="resultado">
1030       <div>${resultados}</div>
1031     </div>
1032   `;
1033
1034   resultados += `

1035     <div class="resultado">
1036       <div>${resultados}</div>
1037     </div>
1038   `;
1039
1040   resultados += `

1041     <div class="resultado">
1042       <div>${resultados}</div>
1043     </div>
1044   `;
1045
1046   resultados += `

1047     <div class="resultado">
1048       <div>${resultados}</div>
1049     </div>
1050   `;
1051
1052   resultados += `

1053     <div class="resultado">
1054       <div>${resultados}</div>
1055    
```



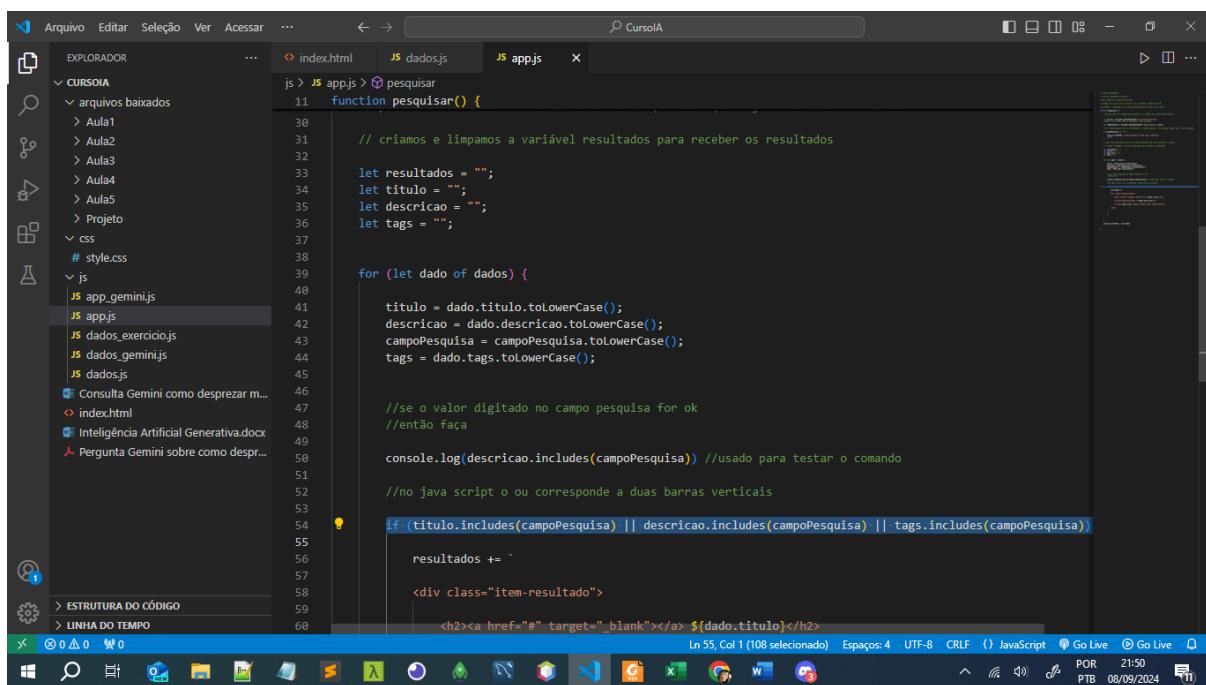
```
function pesquisar() {
    let resultados = "";
    let titulo = "";
    let descricao = "";
    let tags = "";

    for (let dado of dados) {
        titulo = dado.titulo.toLowerCase();
        descricao = dado.descricao.toLowerCase();
        campoPesquisa = campoPesquisa.toLowerCase();
        tags = dado.tags.toLowerCase();

        // se o valor digitado no campo pesquisa for ok
        // então faça
        console.log(descricao.includes(campoPesquisa)) //usado para testar o comando

        //no java script o ou corresponde a duas barras verticais
        if (titulo.includes(campoPesquisa) || descricao.includes(campoPesquisa) || tags.includes(campoPesquisa))

            resultados += `
```



```
function pesquisar() {
    let resultados = "";
    let titulo = "";
    let descricao = "";
    let tags = "";

    for (let dado of dados) {
        titulo = dado.titulo.toLowerCase();
        descricao = dado.descricao.toLowerCase();
        campoPesquisa = campoPesquisa.toLowerCase();
        tags = dado.tags.toLowerCase();

        // se o valor digitado no campo pesquisa for ok
        // então faça
        console.log(descricao.includes(campoPesquisa)) //usado para testar o comando

        //no java script o ou corresponde a duas barras verticais
        if (titulo.includes(campoPesquisa) || descricao.includes(campoPesquisa) || tags.includes(campoPesquisa))

            resultados += `
```

Agora vamos colocar uma condição para que se o usuário digitar uma coisa que não tem nada a ver, por exemplo, laranja,, o sistema gere uma mensagem

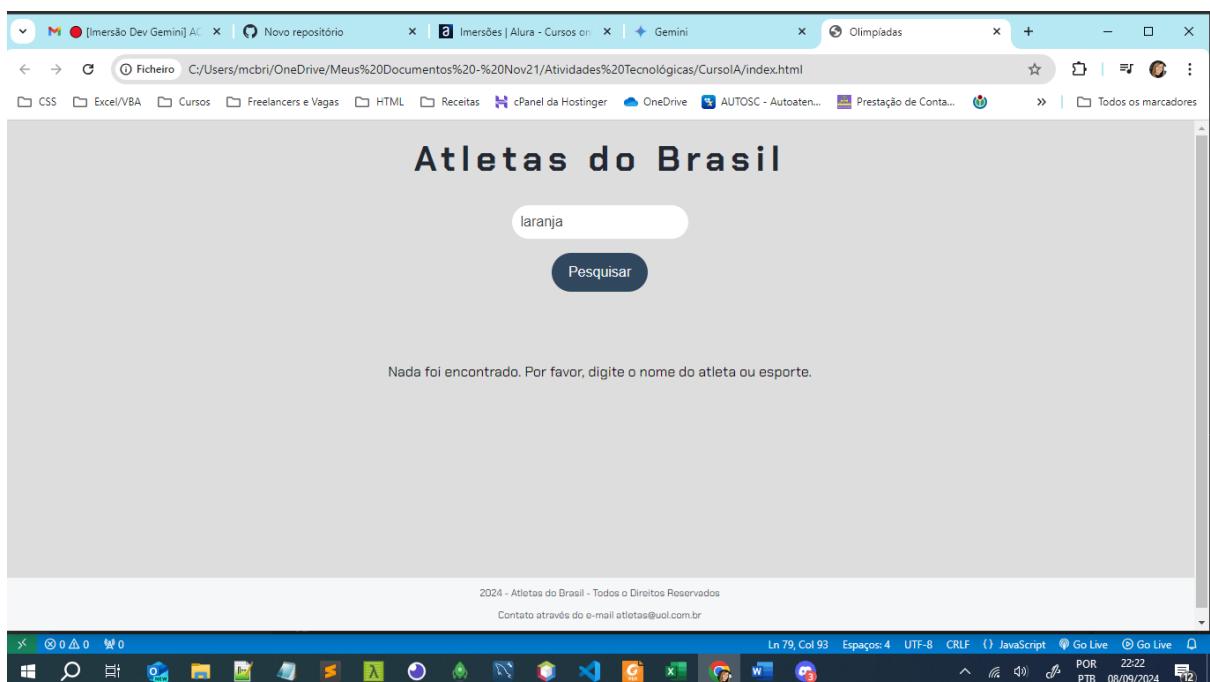
Acrescentamos a instrução no final, após o fechamento do for.

The screenshot shows a code editor interface with several tabs open. The active tab is 'app.js', which contains the following code:

```
11     function pesquisar() {
66         </div>
67         ...
68     }
69 }
70 }
71 }
72 }
73 }
74 }
75 // resultado significa nenhum resultado
76 if (!resultados) {
77     resultados = "<p> Nada foi encontrado. Por favor, digite o nome do atleta ou esporte.</p>";
78 }
79
80 // atribui o resultado encontrado ao HTML
81
82
83
84
85
86 section.innerHTML = resultados
87
88
89
90
91
92 }
```

The code editor's sidebar shows a file tree with 'CURSOIA' selected, containing files like 'Aula1', 'Aula2', 'Aula3', 'Aula4', 'Aula5', 'Projeto', 'css', '#style.css', 'js', 'app_gemini.js', 'app.js', 'dados_exercicio.js', 'dados_gemini.js', 'dados.js', and 'Consult Gemini como desprezar m...'. The status bar at the bottom shows 'Ln 79, Col 93' and other system information.

Resultado:



Agora vamos carregar o projeto para o GitHub.

Criei o repositório: [Introdução ao Google Gemini e desenvolvimento web](#)