**03 - Saída em diferentes formatos**

Nós criamos vários dataframes diferentes por meio de leituras possíveis do Pandas, como a função read\_json(), que utilizamos para buscar os alunos, e o read\_html(), com a qual conseguimos os cursos. A partir disso, juntamos tabelas e criamos dataframes auxiliares que nos trouxeram mais informações, por exemplo matriculas\_por\_curso, que mostra a quantidade de alunos matriculados em cada curso, ou matriculas, mostrando que alunos estão inscritos em quais cursos.

Nosso objetivo agora é exportar o dataframe matriculas\_por\_curso para um novo formato, como o CSV, um dos formatos mais comuns em análise de dados. Para iniciarmos essa nova etapa do projeto, criaremos a seção "Saída em diferentes formatos".

A exportação de um dataframe para um arquivo CSV pode ser feita por meio da função to\_csv(), passando como parâmetro, entre aspas simples, o nome que será atribuído ao arquivo - nesse caso, matriculas\_por\_curso.csv. Além dele, passaremos o parâmetro index=False para determinarmos que o índice da tabela não será exportado.

matriculas\_por\_curso.to\_csv('matriculas\_por\_curso.csv', index=False)COPIAR CÓDIGO

Ao executarmos, teremos a impressão de que nada aconteceu. Entretanto, o arquivo matriculas\_por\_curso.csv estará disponível para download na aba "Files" à esquerda da tela. Como aprendemos nos outros cursos de Pandas aqui da plataforma, tal arquivo pode ser lido a partir da função pd.read\_csv().

pd.read\_csv('matriculas\_por\_curso.csv')COPIAR CÓDIGO

|  | **quantidade\_de\_alunos** | **nome\_do\_curso** |
| --- | --- | --- |
| 0 | 14 | Lógica de programação |
| 1 | 2 | Java para Web |
| 2 | 47 | C# para Web |
| 3 | 34 | Ruby on Rails |
| 4 | 1 | Cursos de Python |

Começamos esse projeto lendo um arquivo JSON, e também é possível criarmos um arquivo desse tipo. Para isso, criaremos a variável matriculas\_json recebendo a chamada de matriculas\_por\_curso.to\_json(). Ao exibirmos o conteúdo da nova variável, veremos nossos dados no formato escolhido.

matriculas\_json = matriculas\_por\_curso.to\_json()

matriculas\_jsonCOPIAR CÓDIGO

{"quantidade\_de\_alunos":{"1":14,"2":2,"3":47,"4":34,"5":1,"6":81,"7":28, ...},"nome\_do\_curso":{"1":"L\u00f3gica de programa\u00e7\u00e3o","2":"Java para Web","3":"C# para Web", ...}}

Também lemos uma tabela a partir de uma página HTML usando a função read\_html(). Para transformarmos o dataframe matriculas\_por\_curso em HTML, usaremos a função to\_html. Armazenaremos o seu retorno em uma nova variável matriculas\_html.

matriculas\_html = matriculas\_por\_curso.to\_html()

matriculas\_htmlCOPIAR CÓDIGO

Será trabalhoso lermos o conteúdo dessa variável (uma <table>), pois ele será exibido em uma única linha. Uma maneira de melhorarmos essa exibição é imprimirmos o conteúdo na tela com o comando print().

print(matriculas\_html)COPIAR CÓDIGO

Como retorno, teremos uma tabela formatada como um código HTML padrão.

<table border="1" class="dataframe">

<thead>

<tr style="text-align: right;">

<th></th>

<th>quantidade\_de\_alunos</th>

<th>nome\_do\_curso</th>

</tr>

<tr>

<th>id\_curso</th>

<th></th>

<th></th>

</tr>

</thead>

<tbody>

<tr>

<th>1</th>

<td>58</td>

<td>Lógica de programação</td>

</tr>

<tr>

<th>2</th>

<td>31</td>

<td>Java para Web</td>

</tr>

<tr>

<th>3</th>

<td>53</td>

<td>C# para Web</td>

</tr>

<tr>

<th>4</th>

<td>4</td>

<td>Ruby on Rails</td>

</tr>

<tr>

<th>5</th>

<td>12</td>

<td>Cursos de Python</td>

</tr>

<tr>

<th>6</th>

<td>74</td>

<td>PHP com MySql</td>

</tr>

<tr>

<th>7</th>

<td>32</td>

<td>.NET para web</td>

</tr>

<tr>

<th>8</th>

<td>18</td>

<td>Novas integrações com Java</td>

</tr>

<tr>

<th>9</th>

<td>59</td>

<td>TDD com Java</td>

</tr>

<tr>

<th>10</th>

<td>39</td>

<td>Código limpo com C#</td>

</tr>

<tr>

<th>11</th>

<td>61</td>

<td>Preparatório para certificação Java</td>

</tr>

<tr>

<th>12</th>

<td>65</td>

<td>Hardware básico</td>

</tr>

<tr>

<th>13</th>

<td>14</td>

<td>Persistência com .NET</td>

</tr>

<tr>

<th>14</th>

<td>27</td>

<td>Desenvolvendo jogos</td>

</tr>

<tr>

<th>15</th>

<td>46</td>

<td>Análise de dados</td>

</tr>

<tr>

<th>16</th>

<td>43</td>

<td>Estatística básica</td>

</tr>

<tr>

<th>17</th>

<td>56</td>

<td>Internet das coisas</td>

</tr>

<tr>

<th>18</th>

<td>53</td>

<td>Programação funcional</td>

</tr>

<tr>

<th>19</th>

<td>74</td>

<td>Boas práticas em Java</td>

</tr>

<tr>

<th>20</th>

<td>35</td>

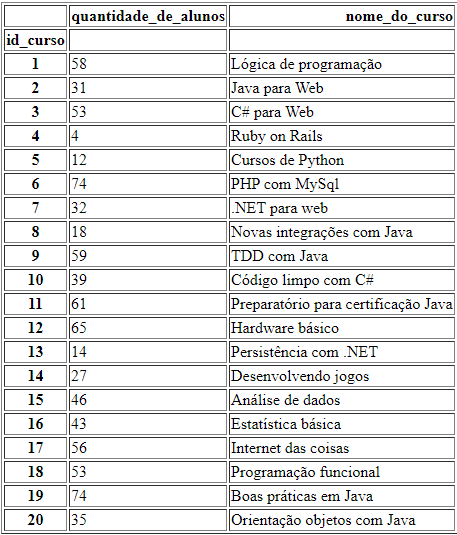
<td>Orientação objetos com Java</td>

</tr>

</tbody>

</table>COPIAR CÓDIGO

Para visualizarmos esse conteúdo em uma página, abriremos um editor de texto (como o Visual Studio Code), colaremos todo o código e salvaremos o arquivo como matriculas.html. Ao abrirmos o arquivo em um navegador, como o Chrome ou o Firefox, a tabela será exibida, com todos os cursos e quantidade de alunos.



[Exemplo de tabela em HTML](https://caelum-online-public.s3.amazonaws.com/1077-pandasIO/Transcri%C3%A7%C3%A3o/imagens/matriculas.html).

Com isso, aprendemos a exportar um dataframe para três formatos diferentes: CSV, JSON e HTML!