**04 - Incluindo ID dos alunos**

No mundo real, as escolas geralmente guardam, além dos nomes, mais informações dos alunos e alunas que lá estudam. Seria interessante incluirmos na nossa tabela alguns desses dados, como um e-mail de cadastro e um código de identificação. Antes de prosseguirmos, vamos organizar melhor nosso projeto adicionando uma caixa de texto com o título "Criando os nomes" (precedido por uma cerquilha #) antes da importação do Pandas, criando uma seção.

#Criando os nomesCOPIAR CÓDIGO

Toda criação de seções seguirá o mesmo procedimento!

Após todos os códigos que escrevemos até agora, adicionaremos um novo texto, agora com o título "Incluindo ID dos alunos". Dessa forma, poderemos utilizar as setas no canto inferior esquerdo para minimizarmos determinadas seções.

A ideia agora é gerarmos um ID para nossos alunos e alunas, algo que podemos fazer de forma aleatória utilizando a biblioteca Numpy do Python, que por convenção recebe o apelido np.

import numpy as npCOPIAR CÓDIGO

Utilizaremos a função random.seed() com o parâmetro 123 pára gerarmos sempre a mesma sequência de números aleatórios.

import numpy as np

np.random.seed(123)COPIAR CÓDIGO

Criaremos uma variável total\_alunos que receberá a chamada de len(nomes), de modo a termos um registro de quantos alunos temos na escola.

total\_alunos = len(nomes)

total\_alunosCOPIAR CÓDIGO

400

Atualmente, nossos alunos possuem uma identificação de acordo com a sua posição no conjunto. Por exemplo, se pegarmos uma amostra de 3 nomes, teremos:

nomes.sample(3)COPIAR CÓDIGO

|  | **nome** |
| --- | --- |
| 34 | SEBASTIAO |
| 42 | MARLENE |
| 103 | MAICON |

O "34" é a posição do nome "Sebastião", mas queremos que os IDs sejam aleatórios, compreendendo entre 1 a 400. Como aprendemos em cursos anteriores, é possível criar novas colunas em um dataframe passando um novo nome entre colchetes. Nesse caso, criaremos a coluna "id\_aluno".

A nova coluna receberá a chamada de np.random.permutation(), uma função do Pandas que distribui números de forma aleatória. Para ela, passaremos como parâmetro o total\_alunos e somaremos 1.

nomes["id\_aluno"] = np.random.permutation(total\_alunos) + 1COPIAR CÓDIGO

Isso nos trará a nova coluna, id\_aluno, que compreende números de 1 a 400 distribuídos aleatoriamente

|  | **nome** | **id\_aluno** |
| --- | --- | --- |
| 46 | FABIANA | 373 |
| 159 | SELMA | 29 |
| 3 | ANTONIA | 383 |

Com nomes.sample(5), conseguiremos novas amostras:

|  | **nome** | **id\_aluno** |
| --- | --- | --- |
| 113 | VICENTE | 27 |
| 160 | MOACIR | 25 |
| 146 | WILIAM | 226 |
| 34 | SEBASTIAO | 270 |
| 61 | CICERO | 335 |

|  | **nome** | **id\_aluno** |
| --- | --- | --- |
| 92 | DIOGO | 345 |
| 166 | KELLY | 45 |
| 98 | THAIS | 12 |
| 142 | HUGO | 19 |
| 184 | MARILIA | 252 |

Repare que os IDs são diferentes da posição de cada nome no dataframe, como no exemplo da "Kelly", que figura na posição 166, mas cujo ID é 45.

Quando um aluno se matricula em um curso, além de um ID, as escolas costumam pedir também um e-mail para envio de informações, mensalidades, certificados e assim por diante. Para implementarmos isso em nosso projeto, criaremos a lista dominios que receberá dois valores: `@dominiodoemmail.com.bre@servicodoemail.com`.

dominios = ['@dominiodoemmail.com.br', '@servicodoemail.com']COPIAR CÓDIGO

A ideia é juntarmos os nomes dos nossos alunos a um desses domínios. Criaremos então, em nosso dataframe, as novas colunas "dominio" e "email".

A coluna nomes['dominio'] irá receber, randomicamente, algum dos domínios da nossa lista, algo que faremos com a função np.random.choice() para qual passaremos, como parâmetros, os dominios e o total\_alunos.

dominios = ['@dominiodoemmail.com.br', '@servicodoemail.com']

nomes['dominio'] = np.random.choice(dominios, total\_alunos)COPIAR CÓDIGO

Ao executarmos um nomes.sample(5), veremos que a nova coluna foi adicionada.

|  | **nome** | **id\_aluno** | **dominio** |
| --- | --- | --- | --- |
| 63 | FABIANA | 81 | @dominiodoemmail.com.br |
| 185 | DIEGO | 38 | @dominiodoemmail.com.br |
| 144 | NILTON | 311 | @servicodoemail.com |
| 120 | MOISES | 398 | @servicodoemail.com |
| 101 | GISELE | 282 | @servicodoemail.com |

Agora queremos juntar os nomes aos seus respectivos domínios, formando os endereços de e-mail. Para isso, criaremos a coluna nomes['email'] que receberá a concatenação de nomes.nome com nomes.dominio. Como queremos concatenar strings, usaremos a função str.cat(). Além disso, desejamos que os nomes estejam em caixa baixa (letras minúsculas), o que conseguiremos com str.lower().

nomes['email'] = nomes.nome.str.cat(nomes.dominio).str.lower()COPIAR CÓDIGO

Trazendo uma amostra de 5 nomes, teremos:

|  | **nome** | **id\_aluno** | **dominio** | **email** |
| --- | --- | --- | --- | --- |
| 75 | NELSON | 48 | @dominiodoemmail.com.br | nelson@dominiodoemmail.com.br |
| 18 | RAIMUNDO | 267 | @servicodoemail.com | raimundo@servicodoemail.com |
| 120 | JOSIANE | 179 | @servicodoemail.com | josiane@servicodoemail.com |
| 74 | ALAN | 123 | @dominiodoemmail.com.br | alan@dominiodoemmail.com.br |
| 71 | MARTA | 326 | @dominiodoemmail.com.br | marta@dominiodoemmail.com.br |

Nosso dataframe agora já tem características mais interessantes para o registro de alunos em uma escola, não acha?