



escola  
britânica de  
artes criativas  
& tecnologia

---

## Módulo | Análise de Dados: Coleta de Dados I

Caderno de Aula

Professor [André Perez](#)

---

### ▼ Tópicos

1. Arquivos CSV;
2. Arquivos Texto;
3. Arquivos Excel.

---

### ▼ Aulas

#### ▼ 0. Estruturas de dados

- **Não estruturado:** texto, imagem, áudio, etc.
- **Semi estruturado:** html, json, etc.
- **Estruturado:** tabelas, planilhas, etc.

#### ▼ 1. Arquivos CSV

##### ▼ 1.1. Formato

Um arquivo **csv** é um tipo de arquivo de **texto** com uma estrutura específica (**estruturado**) para

- **Linhas** são separadas pelo caracter de nova linha ' \n ', normalmente a primeira coluna é o cabeçalho (*header*);
- **Colunas** por um separador: ' , ' (mais comum), ' ; ', etc.

É um tipo de arquivo muito utilizado (talvez o mais utilizado) para armazenar dados no mundo analítico.

### Arquivo CSV: banco.csv

```
%%writefile banco.csv
age,job,marital,education,default,balance,housing,loan
30,unemployed,married,primary,no,1787,no,no
33,services,married,secondary,no,4789,yes,yes
35,management,single,tertiary,no,1350,yes,no
30,management,married,tertiary,no,1476,yes,yes
59,blue-collar,married,secondary,no,0,yes,no
35,management,single,tertiary,no,747,no,no
36,self-employed,married,tertiary,no,307,yes,no
39,technician,married,secondary,no,147,yes,no
41,entrepreneur,married,tertiary,no,221,yes,no
43,services,married,primary,no,-88,yes,yes
```

**Exemplo:** Extraindo os valores da primeira coluna (idade).

```
idades = []

with open(file='./banco.csv', mode='r', encoding='utf8') as arquivo:
    cabecalho = arquivo.readline().split(sep=',')
    indice_idade = cabecalho.index('age')
    linha = arquivo.readline()
    while linha:
        idade = linha.split(sep=',')[indice_idade]
        idades.append(idade)
        linha = arquivo.readline()

print(idades)
```

**Exemplo:** Tipo dos dados.

```
tipos_idades = set(map(lambda idade: type(idade), idades))
print(tipos_idades)
```

**Exemplo:** Média das idades.

```
from functools import reduce
```

```
soma_idades = reduce(lambda idade_a, idade_b: idade_a + idade_b, map(lambda idade:
qtd_idades = len(idades)

media_idades = soma_idades / qtd_idades
print(f"A média das idades é de {media_idades}.")
```

## ▼ 1.2. Pacote CSV

Pacote nativo do Python que facilita a leitura de arquivos no formato CSV.

```
import csv

saldos = None

with open(file='./banco.csv', mode='r', encoding='utf8') as arquivo:
    leitor_csv_iter = csv.reader(arquivo, delimiter=',')
    cabecalho = next(leitor_csv_iter)
    indice_saldo = cabecalho.index('balance')
    saldos = [linha[indice_saldo] for linha in leitor_csv_iter]

print(saldos)
```

**Exemplo:** Média dos saldos.

```
from functools import reduce

soma_saldos = reduce(lambda saldo_a, saldo_b: saldo_a + saldo_b, map(lambda saldo:
qtd_saldos = len(saldos)

media_saldos = soma_saldos / qtd_saldos
print(f"A média dos saldos é de {media_saldos}.")
```

## ▼ 2. Arquivos de Texto

### ▼ 2.1. Formato

Um arquivo **texto** é um tipo de arquivo de **texto** sem uma estrutura definida (**não estruturado**).

**Arquivo TXT:** nubank.txt

```
%%writefile nubank.txt
Como você prefere falar com a gente?
```

### E-mail

Tem alguma dúvida? Podemos te ajudar pelo nosso canal de email.  
meajuda@nubank.com.br

### Telefone

Você pode ligar para o 0800 do Nubank a qualquer hora através do número abaixo.  
0800 608 6236

### Chat

Precisa de uma ajuda agora? Entre em contato com nosso atendimento através do chat  
Basta abrir o chat no app.

### Siga o @Nubank

Saiba das novidades e receba dicas na nossas redes sociais e também na NuCommunity

### Imprensa

Reunimos todas as informações para você aqui.  
press@nu.bank

### Ouvidoria

Já conversou conosco e mesmo assim não  
conseguiu resolver o que precisava? Nossa  
Ouvidoria pode avaliar seu caso.  
0800 887 0463  
ouvidoria@nubank.com.br

Atendemos em dias úteis das 9h às 18h  
(horário de São Paulo/SP).

### Parcerias

Se você tem uma proposta de patrocínio, parceria  
ou publicidade, fale conosco por aqui: marketing@nubank.com.br

### **Exemplo:** Extrair e-mails de um arquivo de texto.

- Extrair as linhas do arquivo.

```
with open(file='./nubank.txt', mode='r', encoding='utf8') as arquivo:  
    linhas = arquivo.readlines()
```

```
print(linhas)
```

- Limpar as linhas do caracter de nova linha '\n'

```
linhas = filter(lambda linha: linha != '\n', linhas)  
linhas = map(lambda linha: linha.strip(), linhas)  
linhas = list(linhas)
```

```
print(linhas)
```

- Extrair linhas com o texto '.com'

```
linhas_com_email = filter(lambda linha: '.com' in linha, linhas)
linhas_com_email = list(linhas_com_email)

print(linhas_com_email)
```

- Extrair emails das linhas com o texto '.com'

```
emails_extraidos = []

for linha_com_email in linhas_com_email:

    palavras = linha_com_email.split(sep=' ')
    emails = filter(lambda palavra: '@' in palavra, palavras)
    emails_extraidos = emails_extraidos + list(emails)

print(emails_extraidos)
```

- E o [press@nu.bank](mailto:press@nu.bank)? :(

## ▼ 2.2. Regex

É um algoritmo de busca de padrões em strings e é implementado nativamente em diversas linguagens de programação. Você pode ler mais sobre regex neste [link](#) e testar seu regex na ferramenta online deste [link](#).

```
import re

lista_padroes = re.findall('<string de busca>', texto)
```

**Exemplo:** Extrair e-mails de um arquivo de texto.

- String de busca.

Para encontrar emails no arquivo de texto, vamos utilizar string de busca '\S+@\S+', onde:

- \S+ encontra um sequencia de caracteres sem espaço;
- @ encontra o caracter '@'
- \S+ encontra um sequencia de caracteres sem espaço.

- Código de extração.

```
import re

with open(file='./nubank.txt', mode='r', encoding='utf8') as arquivo:
    texto = arquivo.read()

emails_extraidos = re.findall('\S+@\S+', texto)
print(emails_extraidos)
```

- Código para salvar em um arquivo csv.

```
import csv

with open(file='./nubank.csv', mode='w', encoding='utf8') as arquivo:
    escritor_csv = csv.writer(arquivo, delimiter=';')
    escritor_csv.writerow(['email']) + list(map(lambda email_extraido: [email Extr
```

**Exemplo:** Extrair perfil de redes sociais.

```
import re

with open(file='./nubank.txt', mode='r', encoding='utf8') as arquivo:
    texto = arquivo.read()

perfil_extraidos = re.findall('@\S+', texto)
perfil_extraidos = filter(lambda perfil: '.' not in perfil, perfil_extraidos)
perfil_extraidos = list(perfil_extraidos)

print(perfil_extraidos)
```

## ▼ 3. Arquivos Excel

### ▼ 3.1. Formato

Um arquivo tabular nativo do Windows, sistema operacional da Microsoft.

**Arquivo Excel:** banco.xlsx

	A	B	C	D	E	F	G	H	
1	age	job	marital	education	default	balance	housing	loan	
2	30	unemployed	married	primary	no	1787	no	no	
3	33	services	married	secondary	no	4789	yes	yes	
4	35	management	single	tertiary	no	1350	yes	no	
5	30	management	married	tertiary	no	1476	yes	yes	
6	59	blue-collar	married	secondary	no	0	yes	no	
7	35	management	single	tertiary	no	747	no	no	
8	36	self-employed	married	tertiary	no	307	yes	no	
9	39	technician	married	secondary	no	147	yes	no	
10	41	entrepreneur	married	tertiary	no	221	yes	no	
11	43	services	married	primary	no	-88	yes	yes	
12									
13									

- Download

```
!wget --show-progress --continue -O ./banco.xlsx https://raw.githubusercontent.com
```

## ▼ 2.2. Pacote openpyxl

Pacote Python para interagir com planilhas excel. A documentação pode ser encontrada neste [link](#).

```
!pip install openpyxl
```

**Exemplo:** Média dos saldos.

```
from openpyxl import load_workbook

planilhas = load_workbook(filename='banco.xlsx')
planilha = planilhas.active

saldos = []

cabecalho = next(planilha.values)
indice_saldo = cabecalho.index('balance')
saldos = [linha[indice_saldo] for linha in planilha.values if linha[indice_saldo]]

print(saldos)
```

**Exemplo:** Tipo dos dados.

```
print(set(map(lambda saldo: type(saldo), saldos)))
```

**Exemplo:** Média dos saldos.

```
from functools import reduce

soma_saldos = reduce(lambda saldo_a, saldo_b: saldo_a + saldo_b, saldos)
qtd_saldos = len(saldos)

media_saldos = soma_saldos / qtd_saldos
print(f"A média dos saldos é de {media_saldos}.")
```

---

✓ 0s conclusão: 20:17

