



Aula 14 – Arquivos

Professor Rodrigo Maciel

Tipos de arquivos suportados pelo Python

- Em Python existem 2 tipos de arquivos principalmente:

- Binário

- Texto

```
0010001110110010101100101011010110010000001001100011101000110
0101100101011010110010000001001100011101000110010000101110001
0010101101011001000000100110001110100011001000010111000100011
0010011000111010001100100001011100010001110110010101100101011
0011001000010111000100011101100101011001010110101100100000010
0010000101110001000111011001010110010101101011001000000100110
0001000111011001010110010101101011001000000100110001110100011
1011001010110010101101011001000000100110001110100011001000010
0101011001010110101100100000010011000111010001100100001011100
0101011010110010000001001100011101000110010000101110001000111
1001000000100110001110100011001000010111000100011101100101011
0000100110001110100011001000010111000100011101100101011001010
1101000110010000101110001000111011001010110010101101011001000
1100100001011100010001110110010101100101011010110010000001001
1101100101011001010110101100100000010011000111010001100100001
0010101101011001000000100110001110100011001000010111000100011
1011001000000100110001110100011001000010111000100011101100101
0001110100011001000010111000100011101100101011001010110101100
0011001000010111000100011101100101011001010110101100100000010
0001000111011001010110010101101011001000000100110001110100011
0110010101100101011010110010000001001100011101000110010000101
0101011010110010000001001100011101000110010000101110001000111
0010000001001100011101000110010000101110001000111011001010110
```

Arreglar Facebook, el reto de Zuckerberg en 2018

Mark Zuckerberg es un hombre de costumbres. Cada año asume un reto. En 2017 se propuso visitar los estados de su país que le faltaban para llegar a los 50 iniciales y conocer mejor su sociedad. Antes había creado un club de lectura, escogió 25 libros en una iniciativa que tuvo gran aceptación. También se hizo runner, corría una milla al día. Entre los retos superados está construir un sistema de inteligencia artificial para su hogar o aprender mandarín, el idioma nativo de su mujer Priscilla Chan.

En 2018 ha decidido afrontar un desafío con impacto global, arreglar Facebook. Cada vez son más las voces que acusan al servicio de ser una pérdida de tiempo por su atención constante y por su escasa efectividad para atajar las noticias falsas. Con la trama rusa y la difusión de propaganda terrorista dentro de su servicio como agujeros principales. Zuckerberg lo reconoce en un post en su muro: "Tenemos mucho que hacer. Tanto para proteger nuestra comunidad contra el acoso y el odio, como de las interferencias de países. Queremos que el tiempo en Facebook esté bien empleado".

Incluso reconoce que a veces la interpretación de las normas no es la correcta: "No hemos sido capaces de prevenir algunas situaciones. Y cometemos muchos errores al ejecutar nuestras reglas, tratando de evitar un mal uso de las herramientas. Si tenemos éxito terminaremos el año mucho mejor que comenzamos y con la trayectoria marcada".

Es demasiado temprano para esperar cambios inmediatos, pero sí asume que la concentración de poder en su servicio no es positiva y genera desconfianza generalizada hacia el sector tecnológico. "Muchos llegamos a la tecnología porque pensamos que era una fuerza descentralizadora, una manera de poner poder en más manos. Con el auge de un pequeño número de grandes compañías tecnológicas -y con los gobiernos usando herramientas para vigilar a sus ciudadanos- mucha gente considera que la tecnología ha terminado por centralizar el poder", argumenta.

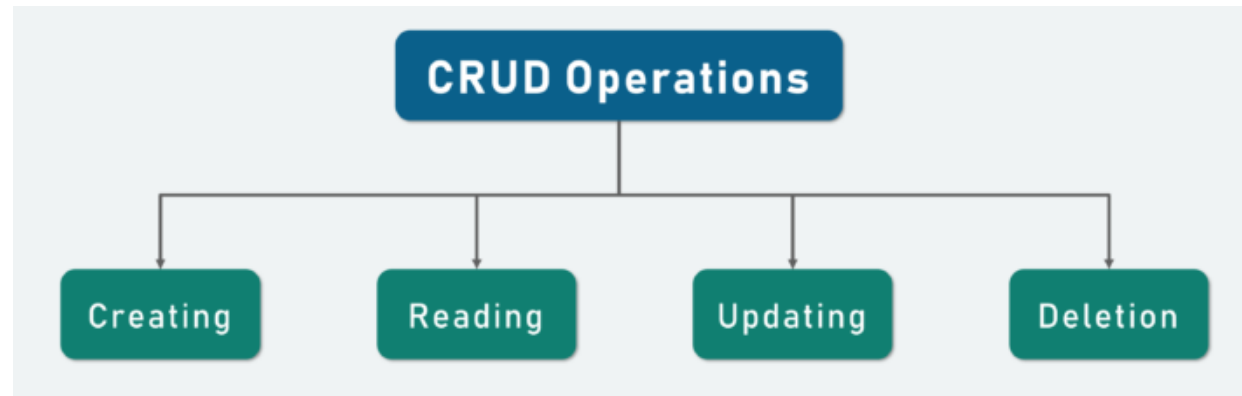
El visionario, por primera vez, se pronuncia sobre uno de los temas más candentes, la criptomoneda: "Tanto la encriptación como las criptomonedas son dos tendencias a tener en cuenta, que toman el poder de sistemas centralizados, pero corren el riesgo de ser difíciles de controlar. Me interesa saber más de ello y estudiar el impacto positivo y negativo de estas tecnologías para saber cómo usarlas mejor en nuestros servicios".

Zuckerberg, que pone así fin a su baja por paternidad tras el nacimiento de su segunda hija, tiende una mano a los usuarios: "Estoy deseando aprender y ponerme a resolver estos problemas todos juntos".

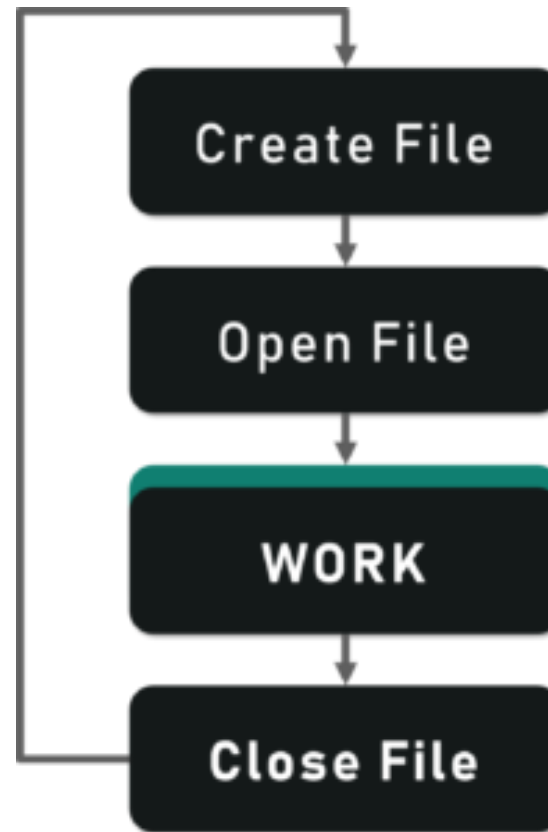
Fonte: El País - Espanha (adaptado)

Operações de arquivo usando Python (CRUD)

- Quais são as operações de arquivo que você geralmente pode executar?
 - Criar
 - Ler
 - Atualizar
 - Deletar



Operação base para manipulação de arquivo



Criando e abrindo arquivos em Python

- Para criar arquivos (e, conseqüentemente, abrí-los), utilizamos o método `open()`. Este método irá abrir o arquivo que passarmos como parâmetro com um determinado modo de uso (que também será passado como parâmetro).

Character	Meaning
'r'	Leitura
'w'	Escrita. Substitui o conteúdo do arquivo existente
'x'	Escrita. Retorna um erro caso o arquivo já exista
'a'	Escrita. Insere os novos dados no final do arquivo
'b'	Modo binário
't'	Modo de texto (padrão)
'+'	Atualizar. Tanto leitura quanto escrita

```
1 arquivo = open("usuarios.txt", "a")
```



Escrevendo dados em arquivos

- Para isso, a linguagem fornece dois métodos:
 - O método `write()` que recebe uma string como parâmetro e a insere no arquivo;
 - O método é o `writelines()` que recebe um objeto iterável (seja uma lista, uma tupla, um dicionário, etc).

Escrevendo dados em arquivos - write()

- Esse método recebe uma string como parâmetro e a insere no arquivo:

```
1 arquivo = open("usuarios.txt", "a")  
2 arquivo.write('Yuri,28,24/05/22 \n')  
3 arquivo.close()
```

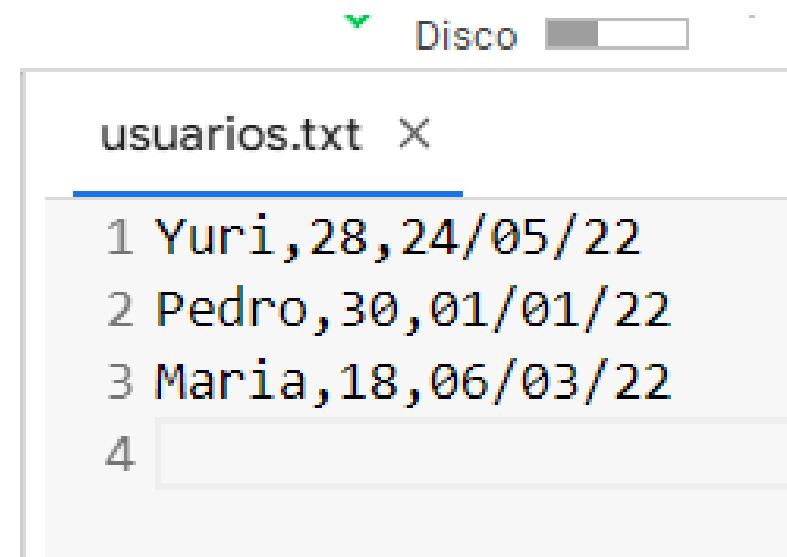
usuarios.txt X

```
1 Yuri,28,24/05/22  
2
```

Escrevendo dados em arquivos - writelines()

- Recebe um objeto iterável (seja uma lista, uma tupla, um dicionário, etc). Com este método, várias linhas poderão ser inseridas no arquivo, diferente do método write() que só recebe uma string por vez.

```
1 arquivo = open("usuarios.txt", "a")
2
3 usuarios = list()
4 usuarios.append('Yuri,28,24/05/22 \n')
5 usuarios.append('Pedro,30,01/01/22 \n')
6 usuarios.append('Maria,18,06/03/22 \n')
7
8 arquivo.writelines(usuarios)
9 arquivo.close()
```





Lendo dados de arquivos

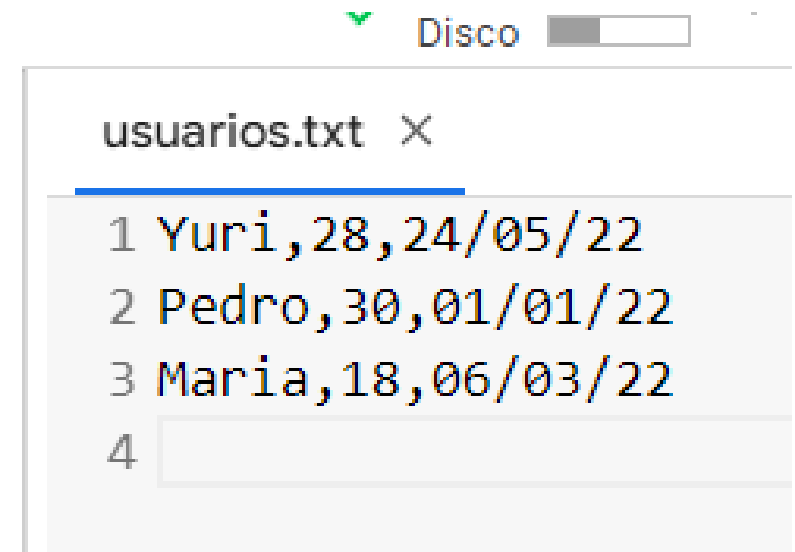
- Para isso, existem dois métodos:
 - O `readline()` que irá ler uma quantidade N de caracteres da primeira linha passadas como parâmetro.
 - O `readlines()` que irá retornar todas as linhas do arquivo.

Lendo dados de arquivos - readline()

- Irá ler uma quantidade N de caracteres da primeira linha passadas como parâmetro:

```
1 arquivo = open("usuarios.txt", "r")  
2 print(arquivo.readline(5))  
3 arquivo.close()
```

Yuri,



Lendo dados de arquivos - readlines()

- Este método irá retornar todas as linhas do arquivo:

```
1 arquivo = open("usuarios.txt", "r")  
2 print(arquivo.readlines())  
3 arquivo.close()
```

```
['Yuri,28,24/05/22 \n', 'Pedro,30,01/01/22 \n', 'Maria,18,06/03/22 \n']
```

usuarios.txt X

```
1 Yuri,28,24/05/22  
2 Pedro,30,01/01/22  
3 Maria,18,06/03/22  
4
```

Buscando e tratando informações

```
1 arquivo = open("usuarios.txt", "r")
2 lista_usuarios = arquivo.readlines()
3 arquivo.close()
```

```
1 for usuario in lista_usuarios:
2     print(usuario)
3     nome, idade, datacadastro = usuario.split(',')
4     print('Lendo o usuario {} com idade {} e cadastrado em {}'.format(nome, idade, datacadastro))
```

Yuri,28,24/05/22

Lendo o usuario Yuri com idade 28 e cadastrado em 24/05/22

Pedro,18,01/01/22

Lendo o usuario Pedro com idade 18 e cadastrado em 01/01/22

Maria,18,06/03/22

Lendo o usuario Maria com idade 18 e cadastrado em 06/03/22

Realizando uma busca com filtro

```
1 arquivo = open("usuarios.txt", "r")
2 lista_usuarios = arquivo.readlines()
3 arquivo.close()
```

```
1 lista_busca = list()
2 for usuario in lista_usuarios:
3     nome, idade, datacadastro = usuario.split(',')
4
5     if idade == '18':
6         lista_busca.append(usuario)
7
8 print(lista_busca)
9
```

```
['Pedro,18,01/01/22 \n', 'Maria,18,06/03/22 \n']
```



Praticando Python

- Faça um algoritmo que solicite do usuário 10 informações sobre alunos (nome, média nota, ano letivo). Você deverá salvar essas informações em um arquivo .txt.
- Faça um algoritmo que busque os alunos (no arquivo .txt) com média maior igual 7 e apresente suas respectivas informações.
- Faça um algoritmo que busque os alunos (no arquivo .txt) com média negativa e faça a alteração por 0 no arquivo .txt.

Praticando Python

- Faça um menu de cadastro de aluno:
 - 1 – Cadastrar alunos (nome, média, ano letivo)
 - 2 – Listar alunos aprovados (média ≥ 7)
 - 3 – Listar alunos reprovados (média < 7)
 - 4 – Ajustar notas negativas (modificar para 0)

Essa atividade deverá ser realizada utilizando funções (ex: `funcao_cadasta()`, `funcao_lista_aprovado()`, `funcao_lista_reprovado()`, `função_ajustar_notas()`).

Essa atividade deverá salvar as informações em um arquivo `.txt`. Você deverá criar informações, salvar, alterar e buscar no arquivo `.txt`.