

07

Mão à obra!

Vamos começar criando o arquivo `testa_concorrencia.py`. Nesse arquivo, vamos abrir o mesmo arquivo (`contatos-escrita.py`) duas vezes no modo de leitura, isto é, vamos atribuir o retorno da função `open()` em duas variáveis diferentes:

```
arquivo1 = open('dados/contatos-escrita.csv', encoding='latin_1', mode='w')
arquivo2 = open('dados/contatos-escrita.csv', encoding='latin_1', mode='w')
```

Então, vamos criar dois contatos e pedir para o Python inserir cada deles um no arquivo:

```
arquivo1 = open('dados/contatos-escrita.csv', encoding='latin_1', mode='w')
arquivo2 = open('dados/contatos-escrita.csv', encoding='latin_1', mode='w')
```

```
contato_carol = '11,Carol,carol@carol.com.br\n'
contato_andreza = '12,Andreza,andreza@andreza.com.br\n'

arquivo1.write(contato_carol)
arquivo2.write(contato_andreza)
```

Qual será o dado que aparecerá no arquivo?

Ao executar o programa vemos que só o contato da `Andreza` aparece. Isso ocorre porque, no momento que o programa é encerrado, um pouco antes de acabar o processo, o Python irá fechar os arquivos invocando o método `close()`. Primeiro, ele fechou o `arquivo1` e, em seguida, o `arquivo2`, mas esta era a ordem correta?

E se o `arquivo1` fosse aberto no modo `a`, o correto seria fechá-lo depois que a variável `arquivo2` invocasse o `close()`?

Por isso, é sempre importante liberar o recurso que estamos utilizando. A boa prática diz que: **sempre que abrir um arquivo feche-o**. Ou seja, só devemos manter o arquivo "aberto" enquanto estamos trabalhando com ele, e ao terminar devemos liberar seu recurso para que outras pessoas possam utilizá-lo.

Podemos fechar os arquivos utilizando um bloco `try/finally`, mas o Python já nos deixa uma palavra reservada chamada `with`. Com ela conseguimos gerenciar o contexto de abertura do arquivo. Isto é, o arquivo só permanece aberto dentro do escopo do bloco, e, após isso, o contexto do arquivo é fechado, ou seja, o método `close()` é invocado.

Podemos, então, falar para o Python abrir os dois arquivos utilizando o `with`. Vamos mudar o modo de abertura do `arquivo2` para `a` e tentar escrever uma nova linha após o bloco do comando `with`:

```
contato_carol = '11,Carol,carol@carol.com.br\n'
contato_andreza = '12,Andreza,andreza@andreza.com.br\n'

with open('dados/contatos-escrita.csv', encoding='latin_1', mode='w') as arquivo1:
    arquivo1.write(contato_carol)

with open('dados/contatos-escrita.csv', encoding='latin_1', mode='a') as arquivo2:
    arquivo2.write(contato_andreza)
```

```
arquivo2.write('Nova linha')
```

Vemos que nos é lançada uma exceção falando que o arquivo está fechado, logo, não é possível escrever nada nele.

Falando em exceções, vimos que existem algumas exceções comuns que acontecem quanto estamos trabalhando com arquivos. Por exemplo, o arquivo pode não ser encontrado em disco ou então, não termos permissão de escrita ou leitura em determinado arquivo.

Portanto, vamos alterar o arquivo principal, colocando o tratamento de exceções e utilizando o `with` para gerenciar o contexto de abertura do arquivo. Ao final, o arquivo fica parecido com este:

```
try:
    with open('dados/contatos.csv', encoding='latin_1') as arquivo_contatos:
        for linha in arquivo_contatos:
            print(linha, end='')
except FileNotFoundError:
    print('Arquivo não encontrado')
except PermissionError:
    print('Sem permissão de escrita')
```