



## Introdução

A programação concorrente e paralela são ferramentas básicas para o desenvolvimento de aplicações modernas, que possam aproveitar os recursos disponíveis em com vários processadores presentes, mesmo, em sistemas embarcados e móveis.

Crie uma pasta no Git chamada 'Semana05' e desenvolva as atividades propostas nos vídeos

## Roteiro de Atividades

1. Veja o seguinte conteúdo teórico, para subsidiar o desenvolvimento das atividades:

- Comando top no Linux

[https://www.youtube.com/watch?v=w\\_5tpzbhyJU](https://www.youtube.com/watch?v=w_5tpzbhyJU)  
<https://www.youtube.com/watch?v=OJSrLp6NUNg>

- Threading vs multiprocessing in python (Veja, pelo menos, entre 5:12 e 12:47)

<https://www.youtube.com/watch?v=AZnGRKFUUoc>

- Thread

<https://www.youtube.com/watch?v=xNBMNKjpJzM>

- Multiprocessing VS Threading VS AsyncIO in Python

<https://leimao.github.io/blog/Python-Concurrency-High-Level/>

- Grok the GIL: How to write fast and thread-safe Python

<https://opensource.com/article/17/4/grok-gil>

- Demystifying Python's Async and Await Keywords (Avançado - Opicional)

[https://www.youtube.com/watch?v=F19R\\_M4Nay4](https://www.youtube.com/watch?v=F19R_M4Nay4)

2. Desenvolva as atividades propostas nos vídeos:

- Duas classes disponíveis em Python para permitir a execução de funções paralelas são a Thread e Multiprocess. Cada uma possui características de operação próprias. Para compreender essa diferença, veja o seguinte vídeo:

<https://www.youtube.com/watch?v=ecKWiaHCEKs>

a) Dentro da pasta dessa semana, crie uma subpasta chamada 'Thread' e desenvolva as atividades propostas no seguinte vídeo: <https://www.youtube.com/watch?v=IEEhzQoKtQU>

b) Dentro da pasta dessa semana, crie uma subpasta chamada 'Multiprocess' e desenvolva as atividades propostas no seguinte vídeo: [https://www.youtube.com/watch?v=fKl2JW\\_qrso](https://www.youtube.com/watch?v=fKl2JW_qrso)

O vídeo possui legendas em inglês, que podem ser traduzidas para o português.



3. A função Async é uma alternativa para a implementação de programação concorrente/paralela com Python. Antes de começar a implementação, veja:

- Async/await: Introdução à programação assíncrona em Python:

<https://medium.com/@edytarcio/async-await-introdu%C3%A7%C3%A3o-%C3%A0-programa%C3%A7%C3%A3o-ass%C3%ADncrona-em-python-fa30d077018e>

- Função Async com Python:

<https://www.youtube.com/watch?v=9rWaiJaWQs>

Dentro da pasta dessa semana, crie uma subpasta chamada 'Async' e desenvolva as atividades propostas no seguinte vídeo:

<https://www.youtube.com/watch?v=t5Bo1JegEmE>

O vídeo possui legendas em inglês, que podem ser traduzidas para o português.

### Referências Extras

- O que é o Python Global Interpreter Lock (GIL):

<https://acervolima.com/o-que-e-o-python-global-interpreter-lock-gil/>

- Asyncio (em português):

<https://www.youtube.com/watch?v=xGoEpCaachs>

<https://www.youtube.com/watch?v=jksHpfsarfc>

<https://www.youtube.com/watch?v=GST5IHIKlJE>

- Async in Python:

<https://realpython.com/async-io-python/>

- Threading Tutorial

<https://www.youtube.com/watch?v=olYdboDdGtM>

<https://www.youtube.com/watch?v=cdPZ1pJACMI>