

# Live de Python #55

Concorrência com asyncio - multiprocessing #5

APOIE O CANAL

**[apoia.se/livedepython](https://apoia.se/livedepython)**



**Nome:**

Eduardo Mendes

**Instituição:**

Unicamp / Diebold Nixdorf

**Contatos:**

{facebook, github, gist  
instagram, linkedin,  
telegram, twitter}/dunossauro

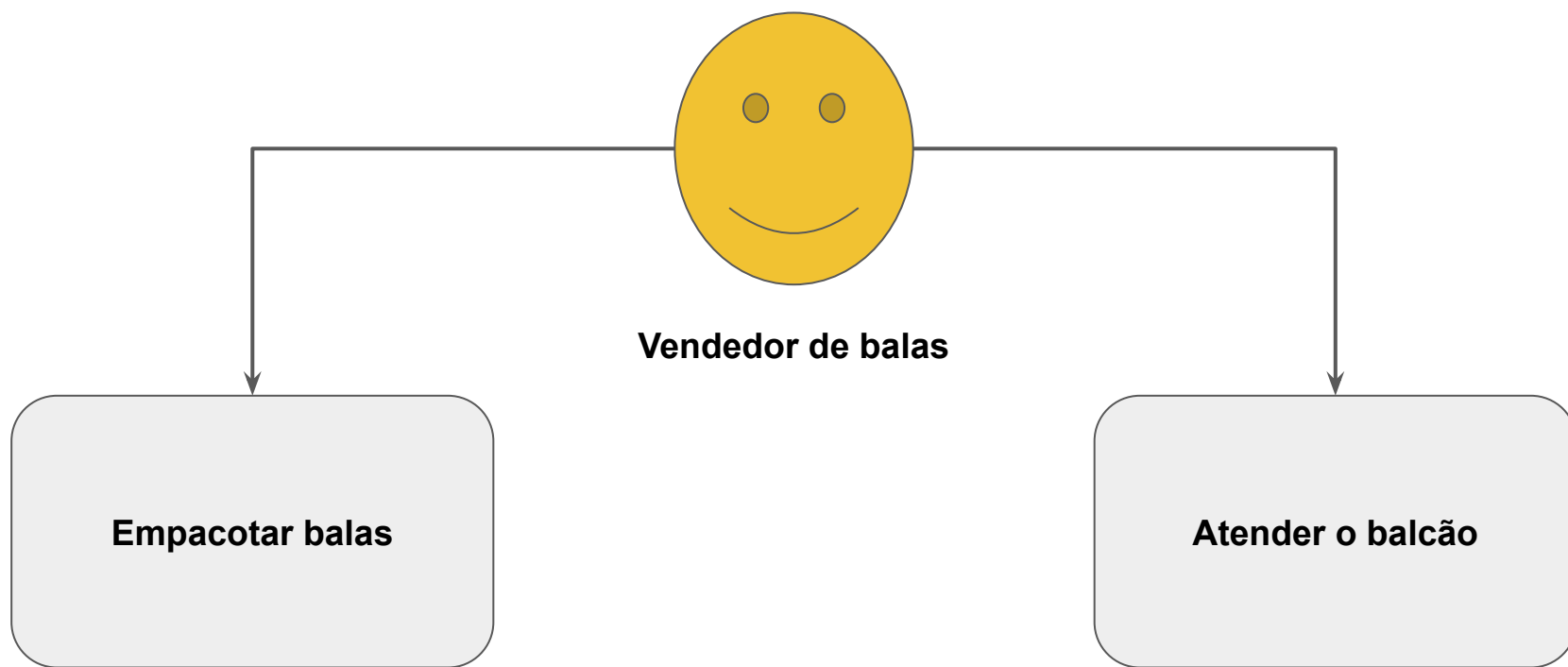
# Roteiro

- O que é uma tarefa assíncrona?
- Implementando as coisas de maneira simples (decoradores)
- Entendendo a nova sintaxe



**Vendedor de balas**





```
def empacotar_bala():  
    print('Empacotando balas')  
    # TODO
```

```
def atender_balcao():  
    print("Atendendo o balcão")  
    # TODO
```



## Vendedor de balas

O vendedor empacota balas e as coloca na vitrine, caso entre algum cliente, ele atende o cliente, caso não haja ele volta a empacotar balas.

O vendedor de balas sabe **gerenciar** suas tarefas de maneira **assíncrona** . Ou seja, ele está em um loop, toda vez que terminar de empacotar vai verificar se há alguém para atender.

**Isso é uma ROTINA**





**Vendedor de balas**

O vendedor empacota balas e as coloca na vitrine, caso entre algum cliente, ele atende o cliente, caso não haja ele volta a empacotar balas.

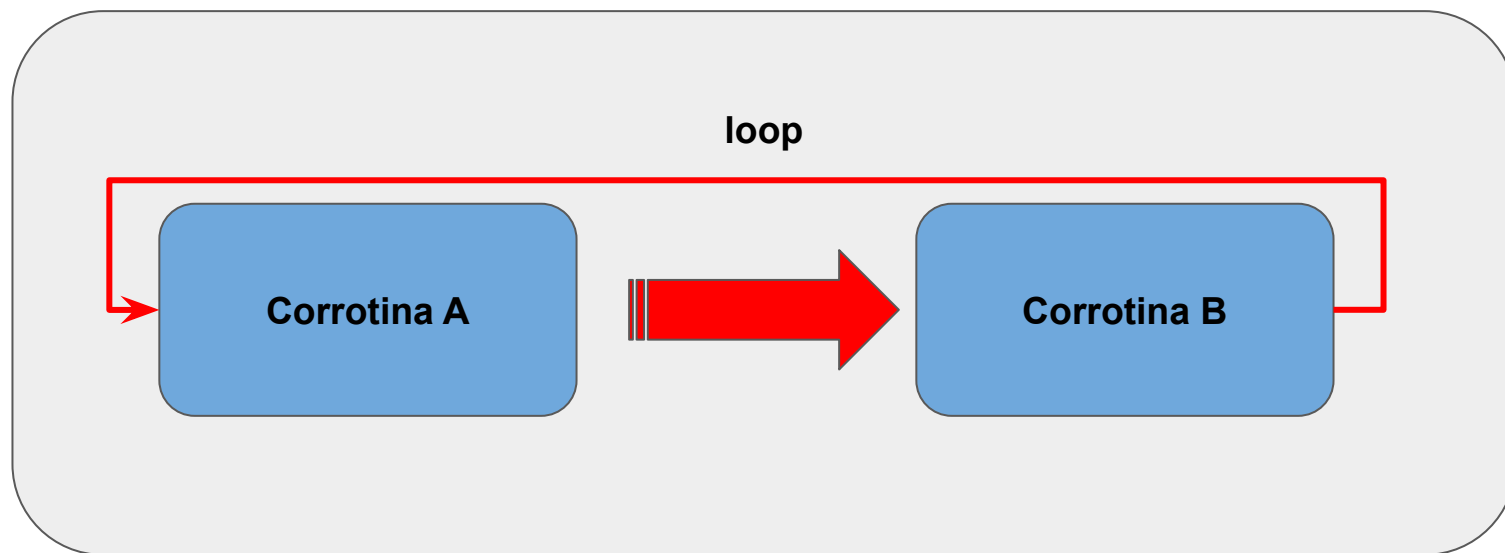
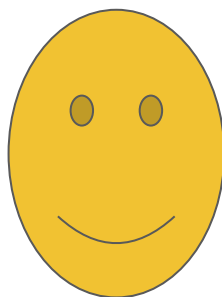
O vendedor de balas sabe **gerenciar** suas tarefas de maneira **assíncrona** . Ou seja, ele está em um loop, toda vez que terminar de empacotar vai verificar se há alguém para atender.

**Isso é uma ROTINA**

**Um evento em loop**

# Entendendo o loop

Quando o vendedor está empacotando balas, sempre antes disso e olha se não tem ninguém pra atender. Ou seja:



# Glossário (vocabulário mais básico possível)

Se você entende do assunto, pode achar essa parte uma aberração, **CALMA**

**Corrotinas:** São funções assíncronas (ou seja, tem um decorador ou começam com `async`)

```
import asyncio

async def ping_server(ip):
    pass

@asyncio.coroutine
def ping_server(ip):
    pass
```

# Glossário (vocabulário mais básico possível)

Se você entende do assunto, pode achar essa parte uma aberração, **CALMA**

**yield from / await:** É uma maneira de consumir uma corrotina. Lembre-se, para consumir uma corrotina, você precisa ser uma corrotina

```
@asyncio.coroutine
def parser_do_ping(path):
    yield from ping_server('127.0.0.1')

async def parser_do_ping(path):
    await ping_server('127.0.0.1')
```

# Glossário (vocabulário mais básico possível)

Se você entende do assunto, pode achar essa parte uma aberração, **CALMA**

**event\_loop:** loop de execução de tarefas/corrotinas

```
@asyncio.coroutine
def parser_do_ping(path):
    yield from ping_server('127.0.0.1')

async def parser_do_ping(path):
    await ping_server('127.0.0.1')
```

CODE !!!

# XOXO



Dúvidas?

**Nome:**

Eduardo Mendes

**Instituição:**

Unicamp / Diebold Nixdorf

**Contatos:**

{facebook, github, gist  
instagram, linkedin,  
telegram, twitter}/dunossauro