Live de Python #54

Concorrência usando futures - Multiprocessamento #4

APOIE O CANAL
apoia.se/livedepython

Nome:
Eduardo Mendes

Instituição:
Unicamp / Diebold Nixdorf

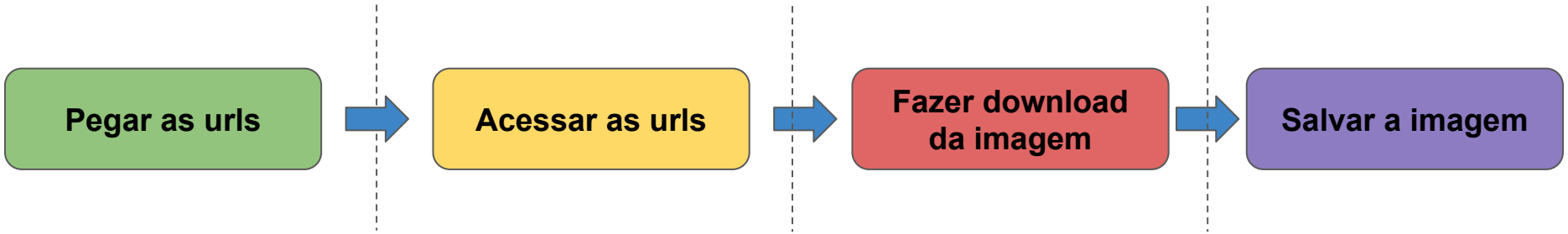
Contatos:
{facebook, github, gist
instagram, linkedin,
telegram, twitter}/dunossauro

Roteiro

- Relembrando o problema
- Analisando soluções passadas
  - Síncrona
  - Threads
  - Processes
- Voltando aos pools
- Entendendo concurrence
  - Executor objects
  - ThreadPools
  - ProcessPools
  - Future objects

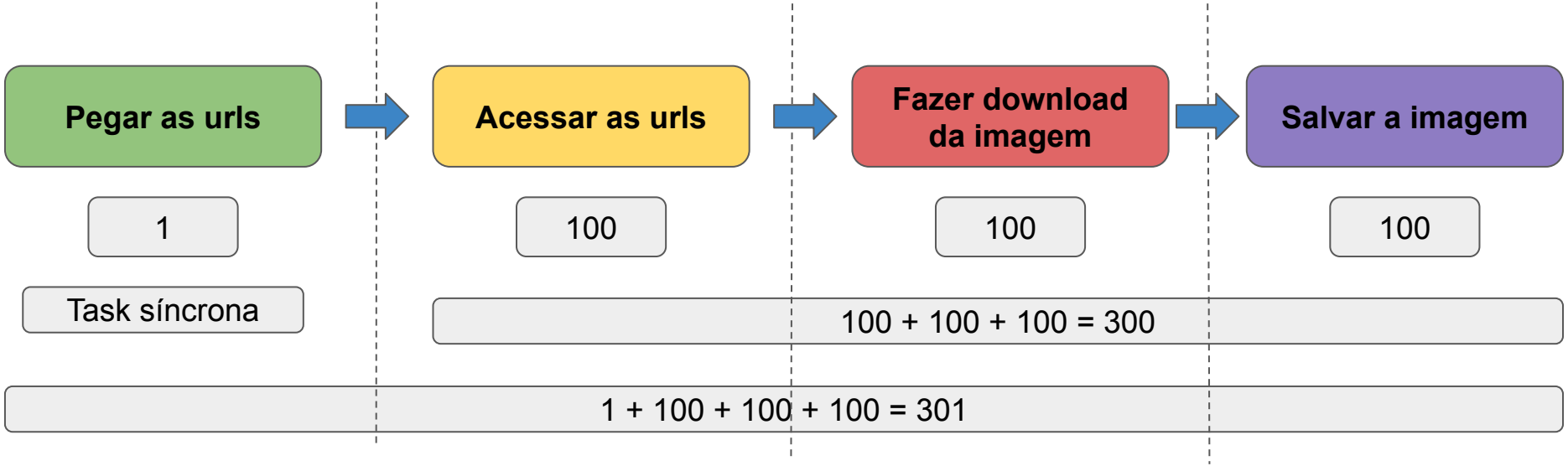
Relembrando o problema

Fazer o download de um sprite dos primeiros 100 pokémons da pokeapi

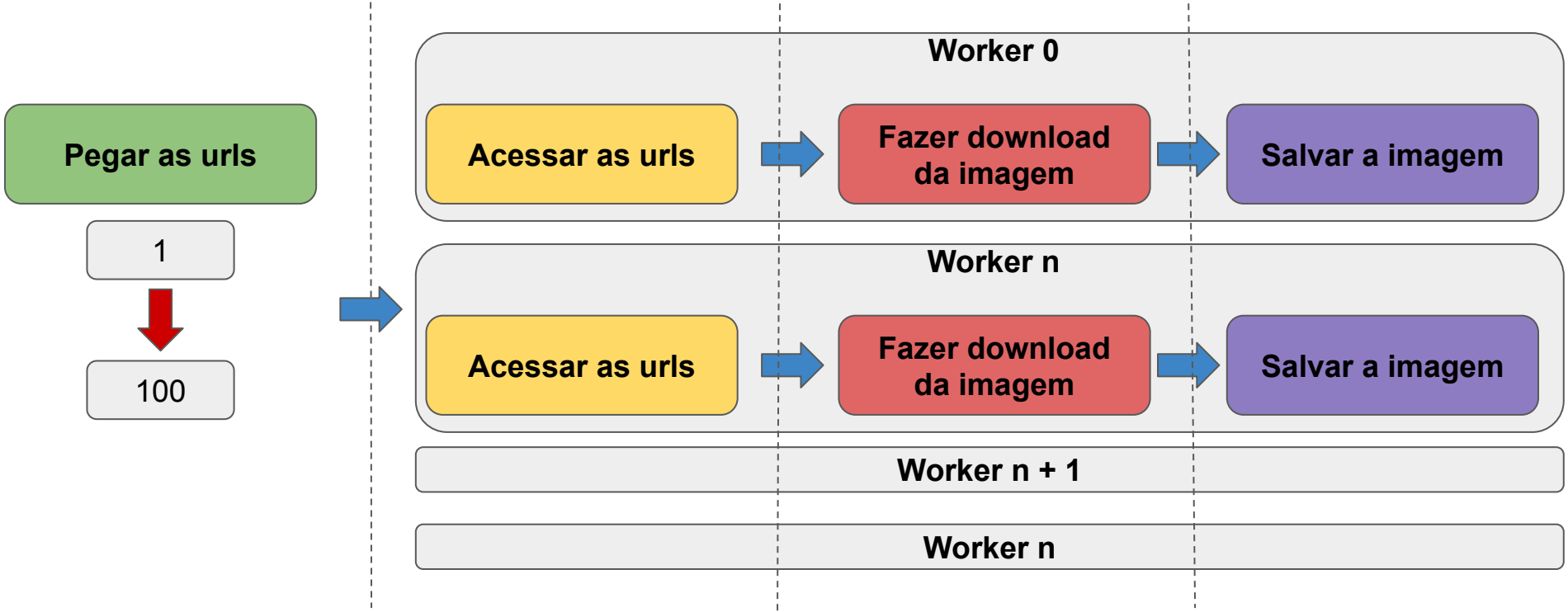


Decomposição

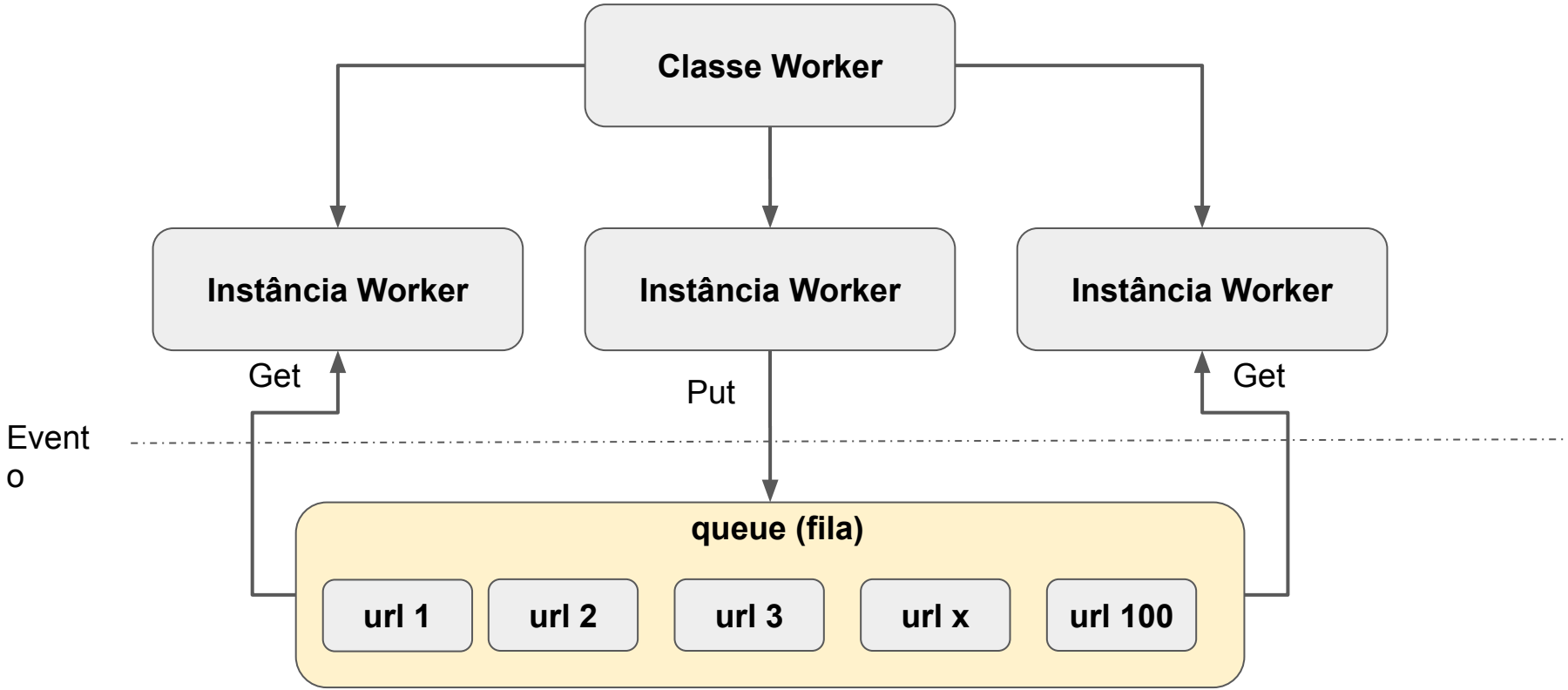
A decomposição inside em pegar um gargalo do processamento e decompor ele em tarefas menores. Vamos chamar tarefas de ‘tasks’ e coisas que resolvem as tasks de workers.



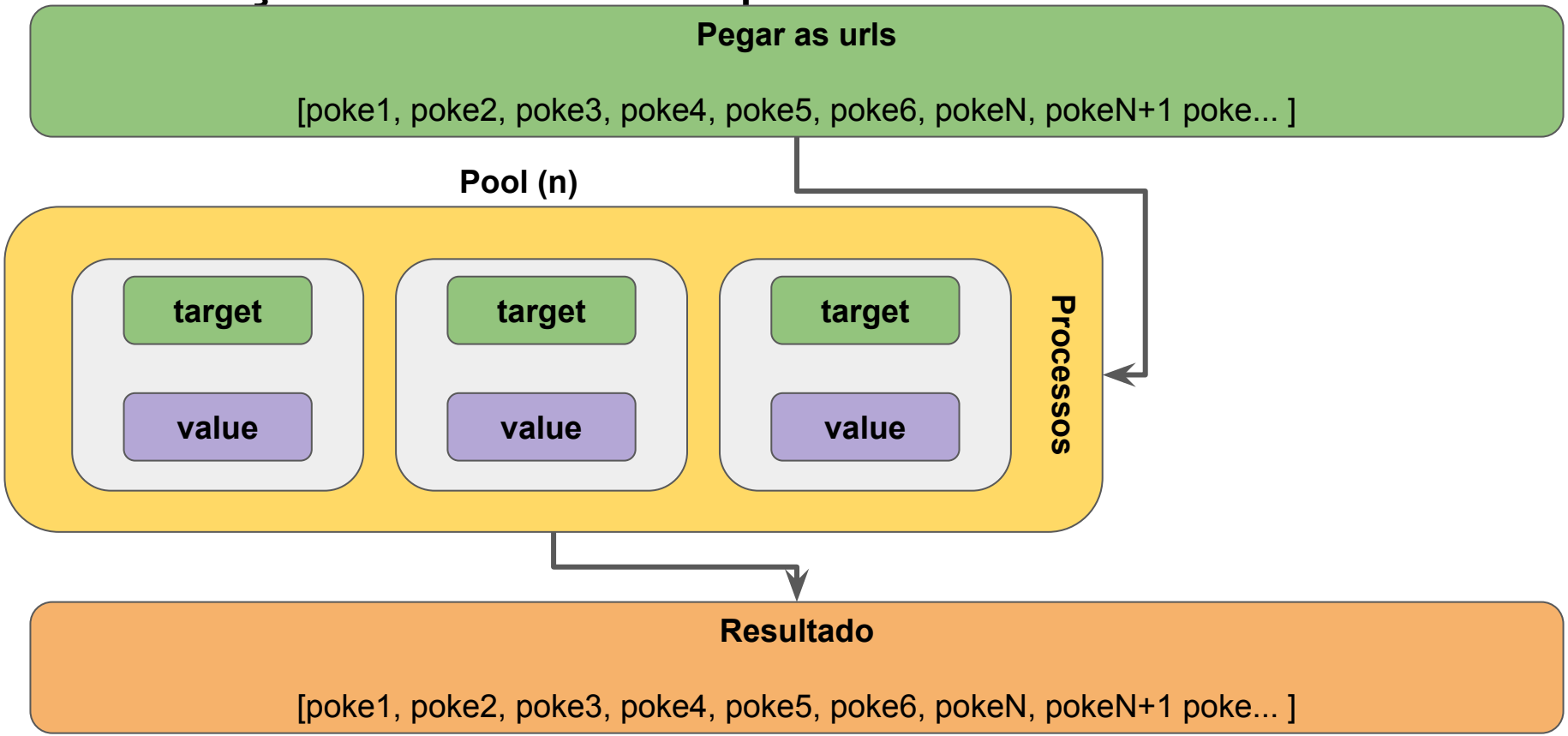
Decomposição



Solução com threads



Resolução com Pool de processos



concurrent.futures

https://docs.python.org/3/library/concurrent.futures.html

Executor (classe abstrata)

```python
from typing import Generator
from abc import abstractmethod, ABC
from concurrent.futures import Future

class Executor(ABC):
    """Uma implementação hipotética."""
    @abstractmethod
    def submit(fn, *args, **kwargs):
        return Future

    @abstractmethod
    def map(func, *iterables, timeout=None, chunksize=1):
        return Generator()

    @abstractmethod
    def shutdown(wait=True):
        pass
```

XOXO

Dúvidas?

Nome:
Eduardo Mendes

Instituição:
Unicamp / Diebold Nixdorf

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