

C O R U T I N A S

```
async def hola():  
    return "Hola"  
  
foo() # devuelve una corutina  
await foo() # devuelve "Hola"
```

⚠️ no puede esperar más de una vez ⚠️
⚠️ si no se espera, no comienza ⚠️

F U T U R O S

```
f = asyncio  
    .get_running_loop()  
    .create_future()  
f.set_result(...) # set_exception()
```

👷 de bajo nivel! 👷
👷 todo a mano! 👷
⚡ puede esperar más de una vez ⚡

T A R E A S

```
# el argumento es una corutina  
t = await asyncio.create_task(foo())  
t.add_done_callback(fn_normal)  
t.done() # booleano  
t.cancelled() # booleano  
t.exception() # None | Exception  
t.result() # el resultado
```

⚡ empieza a ejecutarse de una, no espera ⚡
⚡ puede esperar más de una vez ⚡

PYCON 
C O L O M B I A 2025



GeoPozo

G A T H E R

```
await asyncio.gather( # devuelve una lista  
    tarea1,  
    tarea2,  
    ...,  
    return_exceptions=True,  
)
```

T A S K G R O U P

```
with asyncio.TaskGroup() as tg:  
    tg.create_task(...)
```

🐍 python 3.11 🐍
👉 se cancela toda tarea del grupo si hay un error 👉



Este pocillo es código-abierto!
github.com/geopozo/pycon-co-2025