

os

```
Directorio actual → cwd = os.path.dirname(os.path.realpath(__file__))

os.getpid() # Id de proceso
os.getppid() # Id del proceso padre
os.exit() # Termina el proceso y devuelve un código al SO
```

subprocess:

```
proc = subprocess.Popen('notepad.exe fichero1.txt')
    subproc = subprocess.Popen([<comando y opciones>], stdout=subprocess.PIPE,
    stderr=subprocess.PIPE)
    (standardout, standarderr) = subproc.communicate()
```

psutil:

```
p.kill() / p.suspend() / p.resume()
p.name() / p.exe() / p.nice()
p.openfiles()
p.parent()
```

multiprocessing

clase Process

```
p = multiprocessing.Process(target=f, args=<tupla con parámetros>
p.start()
p.join()
```

clase Queue

```
miCola = multiprocessing.Queue(maxsize=Tamaño máximo)
miCola.empty()
miCola.full()
miCola.put(elemento)
Elemento = miCola.get()
```

threading

```
t = threading.Thread(target=hiloFun)
t = threading.Thread(target=hiloFun, args=(3,))
t = threading.Thread(target=hiloFun, kwargs = {'n':3})
t.start()
t.join()
t.name
t.is_alive()
threading.current_thread()
threading.active_count()
threading.enumerate()
```

Exclusión Mutua

```
ml = threading.Lock()
ml.acquire()
ml.release()
ml.locked()
```

```
mys = threading.Semaphore(4)
mys.acquire()
mys.release(n)
```

Colas

```
import queue
miCola = queue.Queue()
miPila = queue.LifoQueue()
miColaPrio = queue.PriorityQueue()
miColaSimple = queue.SimpleQueue()
```

```
mic.put(elem)
mic.get()
mic.qsize()
mic.empty()
Mic.full()
```