THREADING

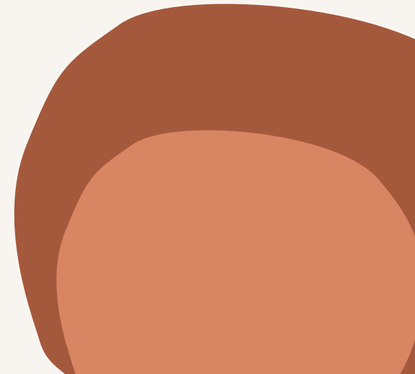
Julio Huerta – Belén Osse – Pedro Ríos

¿Threads?

¿Qué es eso?

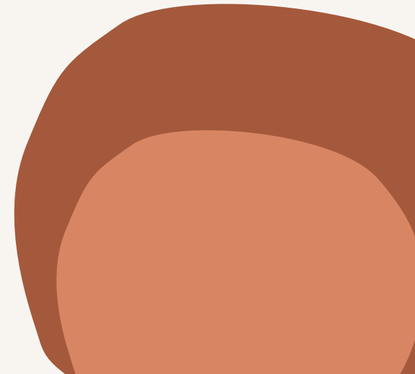
¿Qué es un thread?

Secuencias de instrucciones que pueden ser ejecutadas de forma “paralela”. Permiten a una aplicación realizar varias tareas “a la vez”.



¿Cuándo usar threads?

- Aplicaciones multiusuario
- Funciones independientes se ejecuten en paralelo
- Interfaces que necesiten respuestas
- Entre otras



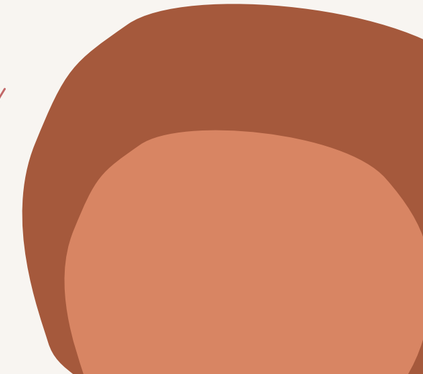
¿Cómo uso los threads?

¡Abróchense los cinturones!

¿Cómo uso los threads?

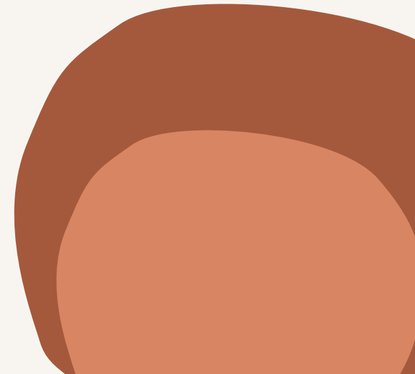
```
import treading
```

Importante!



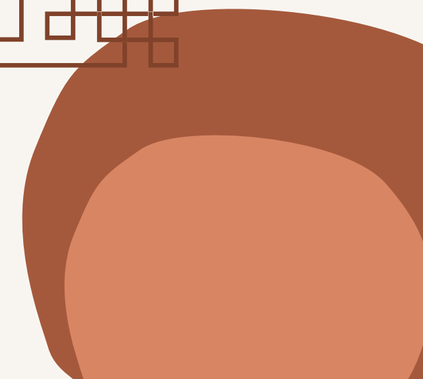
¿Cómo uso los threads?

treading.Thread()



treading.Thread()

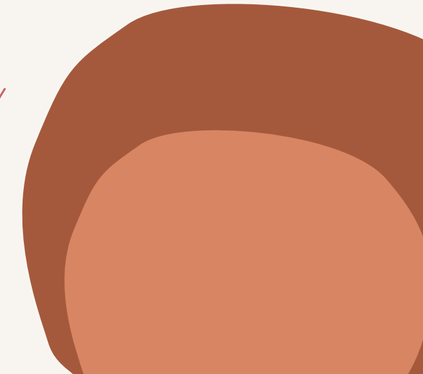
- target
- args
- kwargs
- name
- daemon



¿Cómo uso los threads?

```
hilo = treading.Thread(target = función)
```

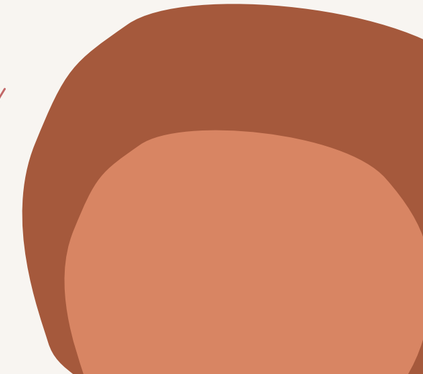
Importante!



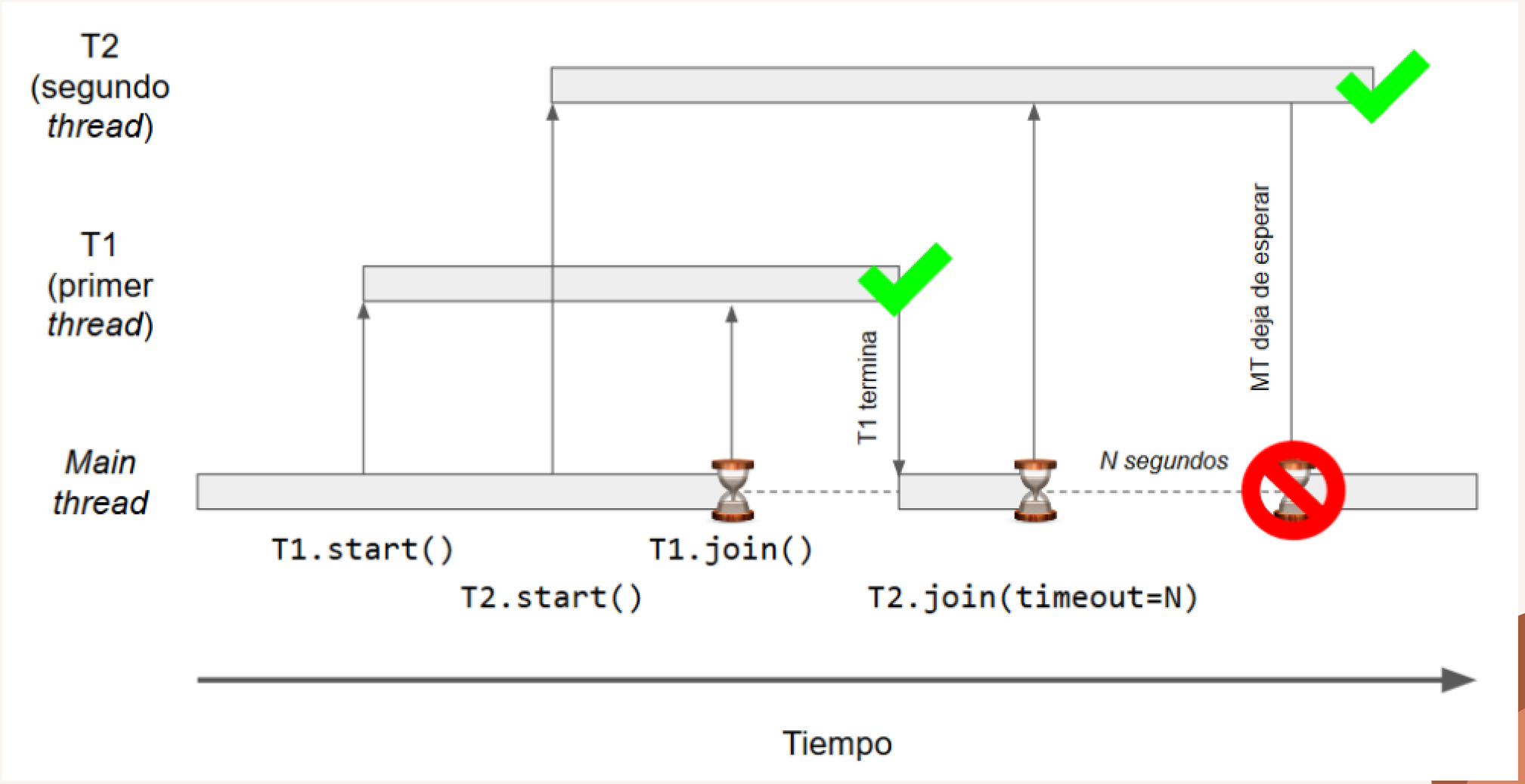
hilo = treading.Thread(target = función)

- start()
- is_alive()
- join()

Importante!



¿Cómo funciona el método join?



Veamos un ejemplo

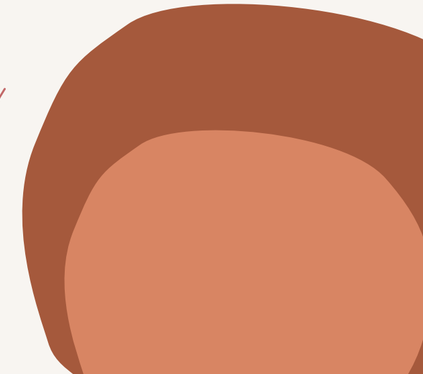
Archivo: creación_threads.py

¡Abróchense los cinturones!

Threads como clases

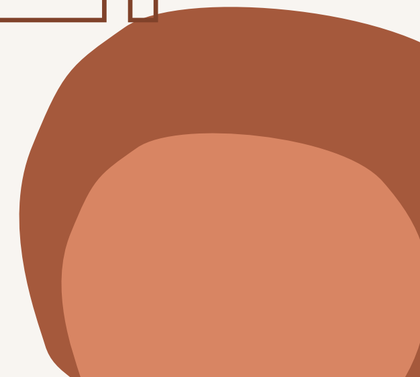
```
hilo = treading.Thread(target = función)
```

Importante!



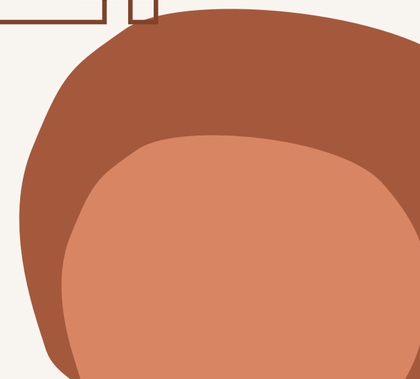
Threads como clases

```
def ClaseThread(threading.Thread):
    def __init__(self, parámetros):
        ...
    def run(self):
        ...
```



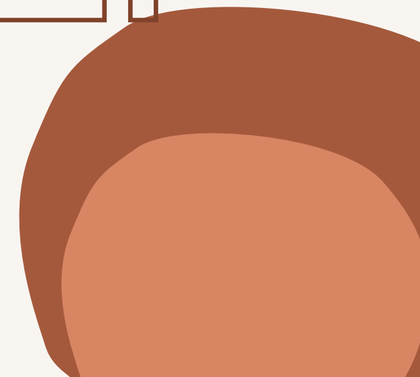
Threads como clases

hilo_clase = ClaseThread(parámetros)



Threads como clases

```
hilo_clase = Clase(parámetros)
hilo_clase.start()
```



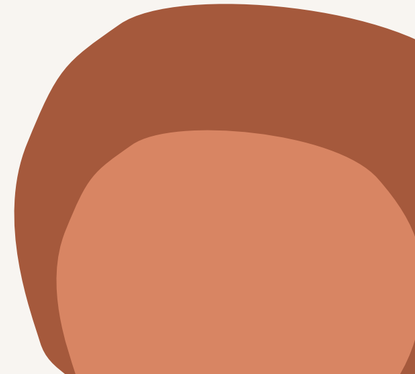
Veamos un ejemplo

Archivo: ejemplo_join.py

¡Abróchense los cinturones!

¿Cómo uso los timers?

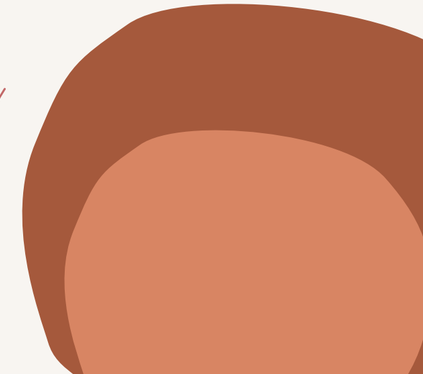
```
hilo_timer = treading.Timer(tiempo, función, args)
```



hilo_timer = treading.Timer(tiempo, función, args)

- start()
- cancel()

Importante!

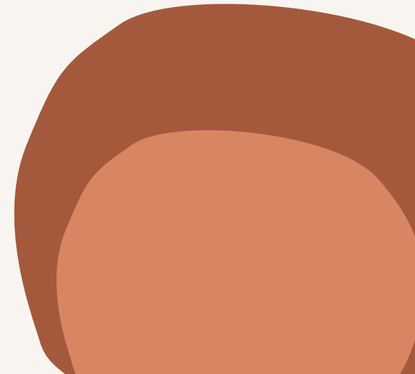


¿Locks?

¿Qué es eso?

¿Qué es un lock?

Los Loks son una primitiva de sincronización provista por la clase Lock de la librería Threading.
Se utilizan para que un thread pueda estar en una misma sección crítica a la vez.

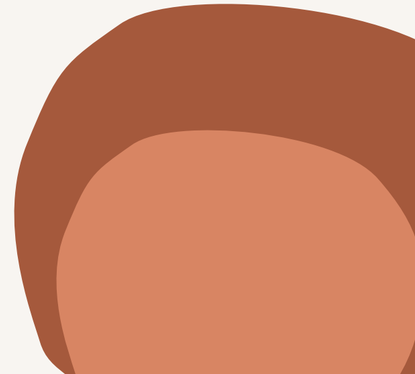


¿Cómo uso los locks?

¡Abróchense los cinturones!

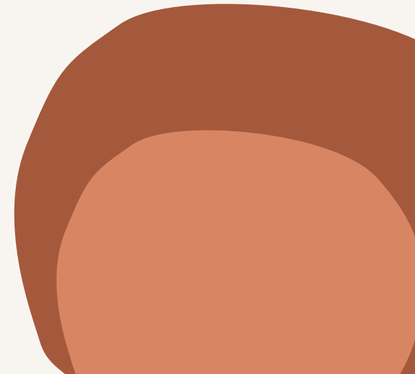
¿Cómo uso los locks?

```
lock_global = treading.Lock()
```



lock_global = treading.Lock()

- acquire()
- release()
- with



Veamos un ejemplo

Archivo: concurrencia.py

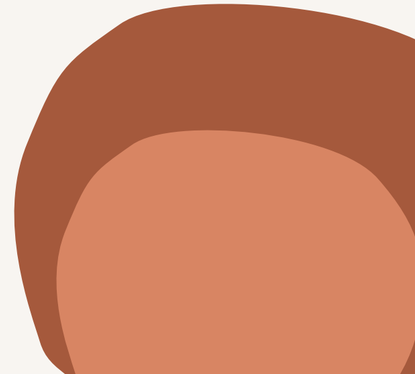
¡Abróchense los cinturones!

¿Eventos?

¿Qué es eso?

¿Qué es un evento?

Un evento es uno de los mecanismos de comunicación entre threads: un thread hace una señal, y otros threads esperan a que esa señal ocurra.
Los Event tienen un flag interno, que toma el valor True cuando la señal está activa, y False cuando no.

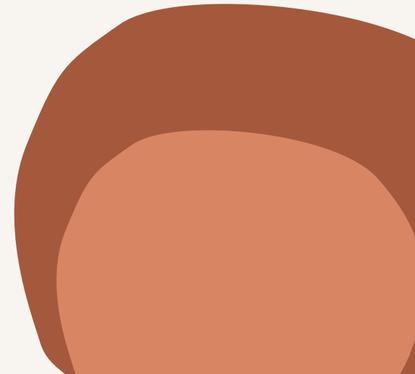


¿Cómo uso los eventos?

¡Abróchense los cinturones!

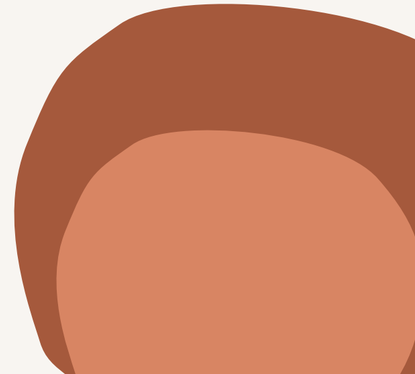
¿Cómo uso los eventos?

```
evento = treading.Event()
```



evento = treading.Event()

- set()
- wait()
- clear()



Veamos un ejemplo

Archivo: senales.py

¡Abróchense los cinturones!

Actividad

Simularemos una actividad del ramo. Para ello, contaremos con tres clases, la actividad, los alumnos, y los ayudantes.
La actividad funcionará en un Thread, y cada alumno representará un Thread también. De esta manera, cuando inicie la actividad y los alumnos comiencen a realizarla, tendremos varios Threads funcionando paralelamente.
Iniciaremos la actividad, en donde implementaremos Threads y Timers y ayudaremos a los alumnos a responder sus dudas. También tendremos cuidado con las zonas críticas del problema (los ayudantes no pueden atender a más de una persona a la vez). Todo esto lo veremos en el siguiente código!

¡A practicar!

Éxitoooo