



# Programación y Laboratorio II

## Clase 19

Programación multi-hilo  
y concurrencia

*Prof. Mauricio Cerizza*

## Concurrencia

- ¿Qué es concurrencia?
- Uso de concurrencia en aplicaciones

## Programación multi-hilo

- ¿Qué es un hilo?
- ¿Qué es un proceso?
- ¿Qué es la programación multi-hilo?
- Programación sincrónica vs paralela
- Ciclo de vida de un hilo



01.

Concurrencia



# ¿Qué es CONCURRENCIA?

En programación hablamos de **concurrency** cuando se ejecuta más de una tarea al mismo tiempo.

Esta habilidad es útil cuando necesitamos que una aplicación haga alguna cosa mientras está trabajando en algo más.

# Uso de concurrencia en aplicaciones

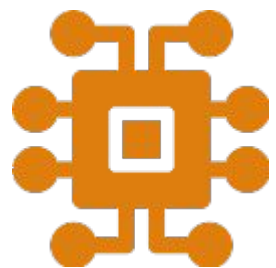
Permite que:

- Los usuarios finales puedan interactuar con la interfaz de usuario de manera no-bloqueante.
- Un servidor pueda atender varias peticiones en simultáneo y no afectar los tiempos de respuesta ante períodos de alta demanda.
- Realizar tareas de cómputo complejas de manera más rápida y haciendo un uso más eficaz los recursos de la computadora.



# 02.

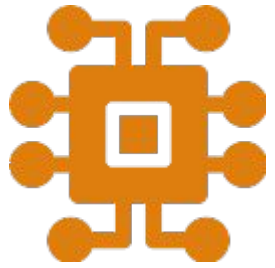
## Programación multi-hilo



# ¿Qué es un HILO?

Un **hilo (thread)**, también llamado **hebra** o **subproceso**, es la unidad básica a la que un sistema operativo asigna tiempo de procesamiento.

Son los encargados de ejecutar nuestro código sentencia a sentencia.

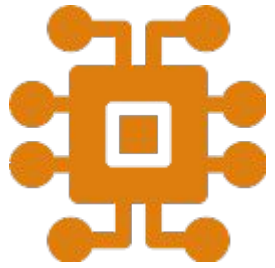


# ¿Qué es un PROCESO?

Un **proceso** es un programa en ejecución que tiene asignados recursos tales como memoria e hilos.

Todos los hilos de un mismo proceso comparten los mismos recursos asignados por el sistema operativo.





# ¿Qué es la PROGRAMACIÓN MULTI-HILO?

Por defecto, cada proceso tiene un único hilo.

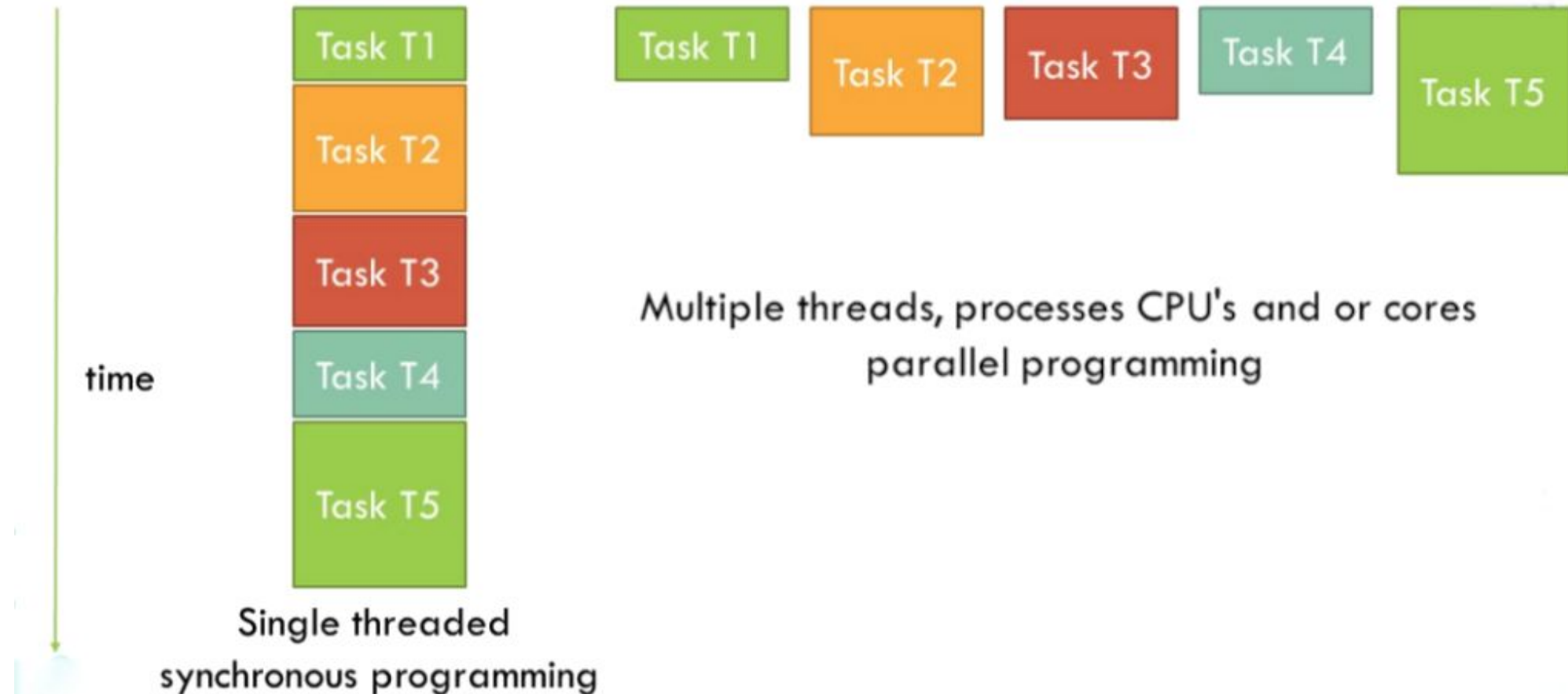
La **programación multi-hilo (multithreaded programming)** permite que un proceso se ejecute sobre múltiples hilos y cada uno de esos hilos esté realizando una tarea distinta en paralelo.

*La **programación en paralelo (parallel programming)** es un sub-tipo de programación multi-hilo.*

*Se utiliza para dividir una gran carga de trabajo en partes independientes y ejecutarlas en paralelo, maximizando el uso de los núcleos de la CPU.*

# Programación sincrónica vs paralela

## SYNCHRONOUS VS PARALLEL PROGRAMMING



# Ciclo de vida de un hilo

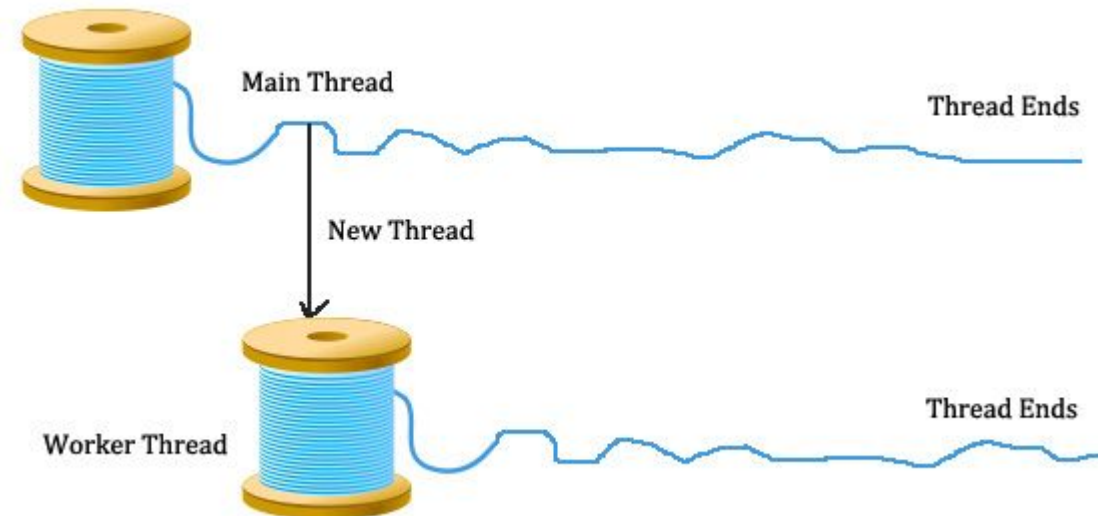
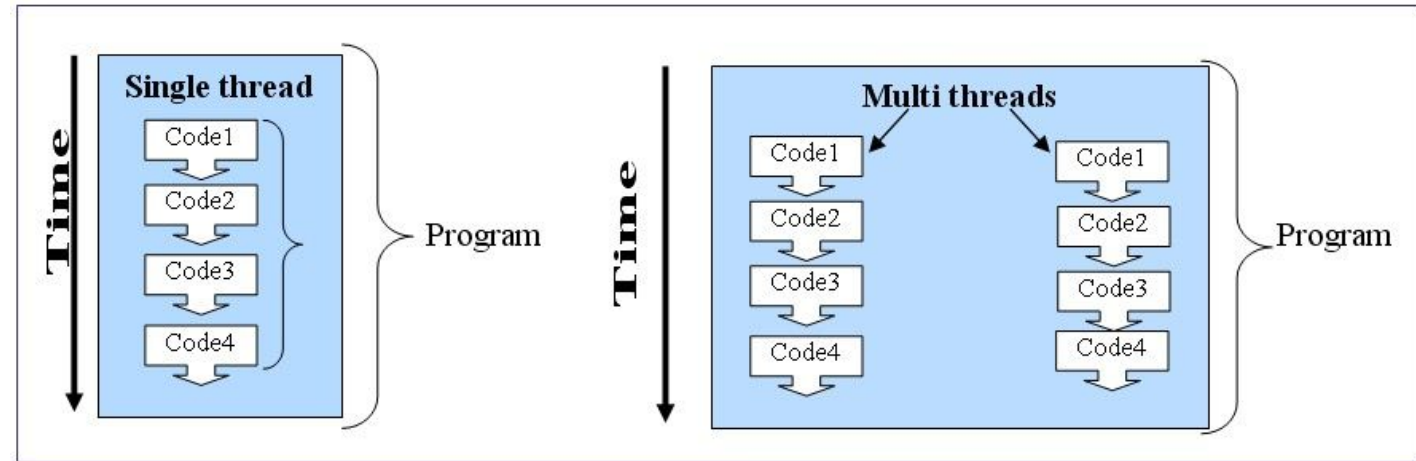
Se llama **hilo principal (main thread)** al primer hilo sobre el que se empezó a ejecutar la aplicación.

A partir de un hilo se pueden abrir nuevos **hilos secundarios**.

Un hilo secundario va a tener su propia **pila de ejecución**, independiente de la de origen.

El proceso existirá mientras al menos uno de sus hilos de ejecución siga activo.

Cuando **todos** los hilos de ejecución finalizan, el proceso no existe más y todos sus recursos son liberados.



# Ejercicios



- Ejercicio I01 - El relojero
- Ejercicio I02 - Simulador de atención a clientes

[https://codeutnfra.github.io/programacion\\_2\\_laboratorio\\_2\\_apuntes/](https://codeutnfra.github.io/programacion_2_laboratorio_2_apuntes/)