



Lab. Programación Concurrente

7° Programación

Valenzuela/Varela



Programación concurrente

Repasando y ampliando conceptos

Programación concurrente

Si bien parecen no ser difíciles los conceptos, repasemos con una imagen que integre todo lo visto hasta ahora...



En este punto vamos a hacer una ligera separación de conceptos para que no quepan dudas a futuro.

Concurrencia no es lo mismo que **Paralelismo**.

Aprender a programar concurrentemente nos va a ayudar a que nuestros programas sean paralelizables.

WHAAAAAT!!!!



Veamos esas diferencias en un cuadro, y anticipemos que nosotros solo vamos a trabajar con programas concurrentes.

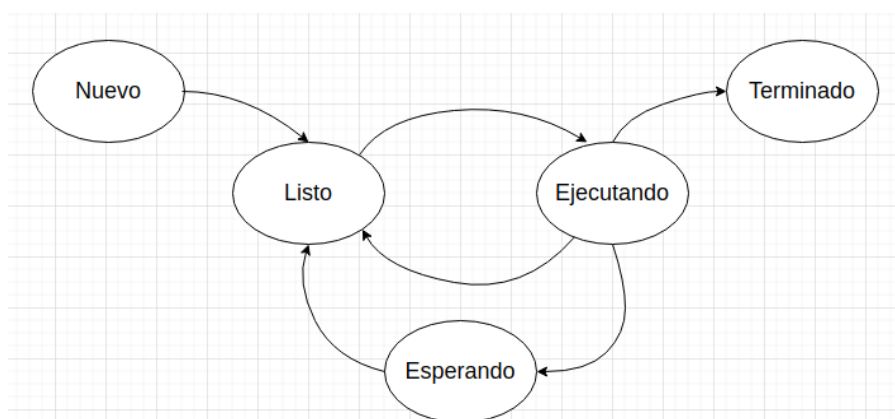
Concurrencia

- Soporta 2 o más acciones en progreso (un procesador)
- Procesos que se ejecutan de forma independiente
- Lidia con muchas cosas al mismo tiempo
- Trabaja principalmente sobre la estructura

Paralelismo

- Soporta 2 o más acciones ejecutándose simultáneamente (más de un procesador)
- Procesos que se ejecutan simultáneamente pueden estar o no relacionados
- Hacer muchas cosas al mismo tiempo
- Trabaja principalmente sobre la ejecución

Otro punto importante a resaltar de lo que dijimos en la teórica anterior es que un programa no es un proceso, sino que este último se está ejecutando y agregamos que tiene **estados de ejecución**. Spoiler Alert: Esto va a ser muy importante más adelante





Lab. Programación Concurrente

7° Programación

Valenzuela/Varela



Llamaremos **Nuevo** al estado del proceso que se está creando.

Cuando termina de crearse cambia de estado y llamaremos a este nuevo estado **Listo**.

En este momento el scheduler lo puede tomar para empezar a procesarlo. Una vez procesado pasará al nuevo estado llamado **Ejecutando**.

Pos ejecución del proceso pueden suceder tres casos. El primero y más fácil es cuando termina de ejecutarse y pasa al estado **Terminado**. El segundo caso es que el proceso necesite esperar algún dato o algo que le permita volver a la cola de procesos Listos para ser ejecutados por el scheduler nuevamente. A este nuevo estado lo llamaremos **Esperando**. El último caso es que el proceso sea ejecutado parcialmente y vuelve a la cola de Listo. Ejemplo de esto: el scheduler decidió darle tiempo a otro proceso. En este último caso el proceso volverá al estado **Listo**.

Si llegaron hasta acá y no salieron corriendo es una buena señal. Les dejamos un tiempo para asimilar todo esto y un par de ejercicios o preguntas en la práctica

2