# Apunte teórico Taller de Programación

*Programación concurrente*

#### ¿Que es la programación concurrente?¿Que permite?¿En que influye?¿Donde podemos encontrarla?

Es un concepto clave en la Ciencia de la computación, se encarga de ejecutar múltiples actividades en simultáneo, permite la interacción de distintos procesos al mismo tiempo, influye en el desarrollo de hardware, sistemas operativos y multiprocesadores.

Se utiliza en sistemas operativos, smartphones, páginas de reservas

Es la característica de los sistemas que indica que múltiples procesos /tareas pueden ser ejecutados al mismo tiempo y pueden cooperar y coordinarse para cumplir la función del sistema

#### ¿Qué escenarios de programación concurrente pueden darse?

Programa secuencial, Paralelismo o Heterogeneidad

#### ¿Los procesadores acompañaron a los sistemas concurrentes?

Si, para poder explotar este hardware es necesario programar procesos concurrentes

#### ¿Que es la Comunicación?¿Que es la sincronización?

Los procesos concurrentes tendrán la necesidad de comunicarse información. Además será necesario en ocasiones detener un proceso hasta que se produzca un determinado suceso o se den ciertas condiciones, esto es la sincronización

Los lenguajes de programación deben proporcionarnos mecanismos de comunicación y sincronización.

#### ¿Por qué elementos está compuesta la comunicación?¿Qué condiciones deben cumplirse para que pueda darse?

La comunicación está compuesta por el Pasaje de Mensajes y la Memoria Compartida

* Es necesario establecer un canal lógico o físico para transmitir información entre procesos.
* El lenguaje debe proveer un protocolo adecuado
* Los procesos deben saber cuando tienen lenguajes para leer y cuando deben transmitir mensajes.

#### ¿De qué formas puede darse el pasaje de mensajes?

Los mensajes pueden ser Enviados o Recibidos y esto puede darse de forma Sincrónica o Asincrónica

#### ¿Que es la memoria compartida?¿Cómo interactúan los procesos con ella?

Es un modo eficaz de pasar datos entre procesos o actuar coordinadamente sobre ellos.

Dado que no pueden operar simultaneamente sobre ella se “protege” y “libera” el acceso a la memoria

#### ¿Cómo se logra “proteger” y “liberar” memoria?

La solución más elemental es utilizar una variable de control que habilite o no el acceso de un proceso a la memoria compartida.

Este método clásico se denomina semáforo: Dado un recurso compartido, si está libre entonces -> bloqueo, lo uso, libero. Sino vuelvo al paso anterior.