



# UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



*FACULTAD DE INGENIERÍA*

*MECÁNICA Y ELÉCTRICA*

**ACTIVIDAD FUNDAMENTAL #2**

**EQUIPO #2**

| Nombre                          | Matricula | Carrera |
|---------------------------------|-----------|---------|
| Natividad Aron De León Ramírez  | 1855134   | IAS     |
| Emiliano García Montemayor      | 2003905   | ITS     |
| Alonso Ramírez Páez             | 2127873   | ITS     |
| Rocío Guadalupe Sánchez Medrano | 1959446   | IAS     |
| Carlos Gabriel Beas González    | 1940892   | ITS     |
| Javier López Pérez              | 2127884   | ITS     |
| Daniel Aharon Sánchez González  | 1967943   | ITS     |

**Materia:** Sistemas Operativos

**Ing.** Norma Edith Marín Martínez

**Hora:** N1-N3 | **Aula:** 1303 | **Grupo:** 002

**Semestre:** enero - junio 2024

---

## INTRODUCCIÓN

---

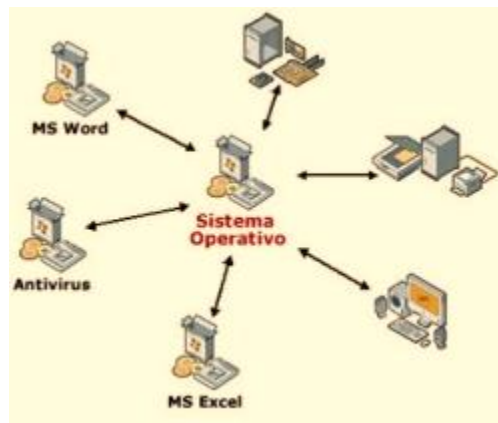
La concurrencia en sistemas operativos es un concepto fundamental que se refiere a la capacidad de ejecutar múltiples tareas de manera simultánea. En un entorno computacional, la concurrencia se logra permitiendo que varios procesos o subprocesos se ejecuten de manera independiente y concurrente, compartiendo eficientemente los recursos del sistema, como la CPU, la memoria y los dispositivos de entrada/salida.

El objetivo principal de la concurrencia en sistemas operativos es mejorar la eficiencia y el rendimiento del sistema al permitir la ejecución simultánea de múltiples tareas. Esto es especialmente crucial en entornos modernos donde las aplicaciones y servicios demandan un alto grado de interactividad y respuesta en tiempo real. La concurrencia puede manifestarse a través de diversos mecanismos, como la multitarea, la multiprogramación y el uso de hilos (threads), permitiendo que diferentes partes de un programa o varios programas se ejecuten de manera concurrente.

En términos prácticos, la concurrencia facilita la gestión eficiente de recursos, mejora la utilización de la capacidad de procesamiento y contribuye a la creación de sistemas más robustos y adaptables. Sin embargo, su implementación adecuada presenta desafíos, como la sincronización de recursos compartidos y la prevención de condiciones de carrera, que requieren estrategias y mecanismos específicos para garantizar la coherencia y la integridad de los datos. En resumen, la concurrencia en sistemas operativos es esencial para optimizar el rendimiento y la capacidad de respuesta de los sistemas informáticos en entornos modernos y dinámicos.

## ¿Qué es la 'Concurrencia'?

La concurrencia es una propiedad de los sistemas en la cual los procesos de un cómputo se hacen simultáneamente, y pueden interactuar entre ellos, es decir son procesados al mismo tiempo, de manera que, para ejecutar uno de ellos, no hace falta que se haya ejecutado otro. En otras palabras, la concurrencia es simplemente la ejecución de varias tareas al mismo tiempo.



## Tipos de procesos concurrentes

**1. Proceso independiente:** Es aquel que ejecuta sin requerir la ayuda o cooperación de otros procesos. Un claro ejemplo de procesos independientes son los diferentes shells que se ejecutan de forma simultánea en un sistema. En español es todo aquel proceso que trabaja por si solo y que no requiere de ayuda de otro para funcionar adecuadamente.

**2. Procesos son cooperantes:** Son aquellos que están diseñados para trabajar conjuntamente en alguna actividad, para lo que deben ser capaces de comunicarse e interactuar entre ellos.

Mientras tanto, los **modelos de programación concurrente** hacen referencia a las técnicas de programación que son utilizadas para expresar la concurrencia entre tareas y solución de los problemas de comunicación y sincronización entre procesos.

Existen varios modelos de programación concurrente, cada uno con sus propias características y enfoques para abordar la ejecución simultánea de tareas. Algunos de los modelos más comunes son:

- **Procesos e Hilos (Threads):** Este modelo implica la ejecución concurrente de procesos independientes o hilos dentro de un programa. Los procesos pueden compartir información a través de la comunicación interprocesos (IPC), mientras que los hilos comparten el mismo espacio de memoria, facilitando la comunicación a través de variables compartidas. Este enfoque es común en sistemas operativos modernos.
- **Programación basada en Eventos:** En este modelo, el flujo de control del programa se guía por eventos y las correspondientes respuestas asociadas. Los eventos pueden ser desencadenados por acciones del usuario, entrada/salida, o eventos internos del sistema. La programación basada en eventos es frecuentemente utilizada en interfaces de usuario y sistemas que requieren manejo eficiente de interrupciones.
- **Modelo Actores:** En este modelo, los actores son entidades independientes que encapsulan estado y comportamiento. Los actores se comunican entre sí enviándose mensajes y pueden ejecutarse de manera concurrente. Este enfoque es utilizado en sistemas distribuidos y en lenguajes como Erlang.
- **Paralelismo de Datos:** Se centra en dividir grandes conjuntos de datos en fragmentos más pequeños y procesarlos de manera simultánea. Cada fragmento de datos se procesa de manera independiente, lo que permite un alto grado de paralelismo y es común en aplicaciones que involucran operaciones intensivas en datos.
- **Modelo de Memoria Compartida:** Implica la compartición de memoria entre hilos o procesos, lo que permite la comunicación a través de variables compartidas. Sin embargo, se deben implementar mecanismos de sincronización, como semáforos y mutex, para evitar problemas de concurrencia, como condiciones de carrera.

Y continuación, una tabla comparativa sobre la programación concurrente y la programación paralela:

| <b>Programación Concurrente</b>  | <b>Programación Paralela</b>                                     |
|--|--|
| Soporta dos o más acciones en progreso.  | Soporta dos o más acciones ejecutándose simultáneamente.         |
| Procesos que se ejecutan de manera independiente   | Procesos que se ejecutan simultáneamente (tal vez relacionados). |
| Lidiar con muchas cosas al tiempo y sobre la estructura Hacer muchas cosas al tiempo y sobre la ejecución. | Hacer muchas cosas al tiempo y sobre la ejecución.               |

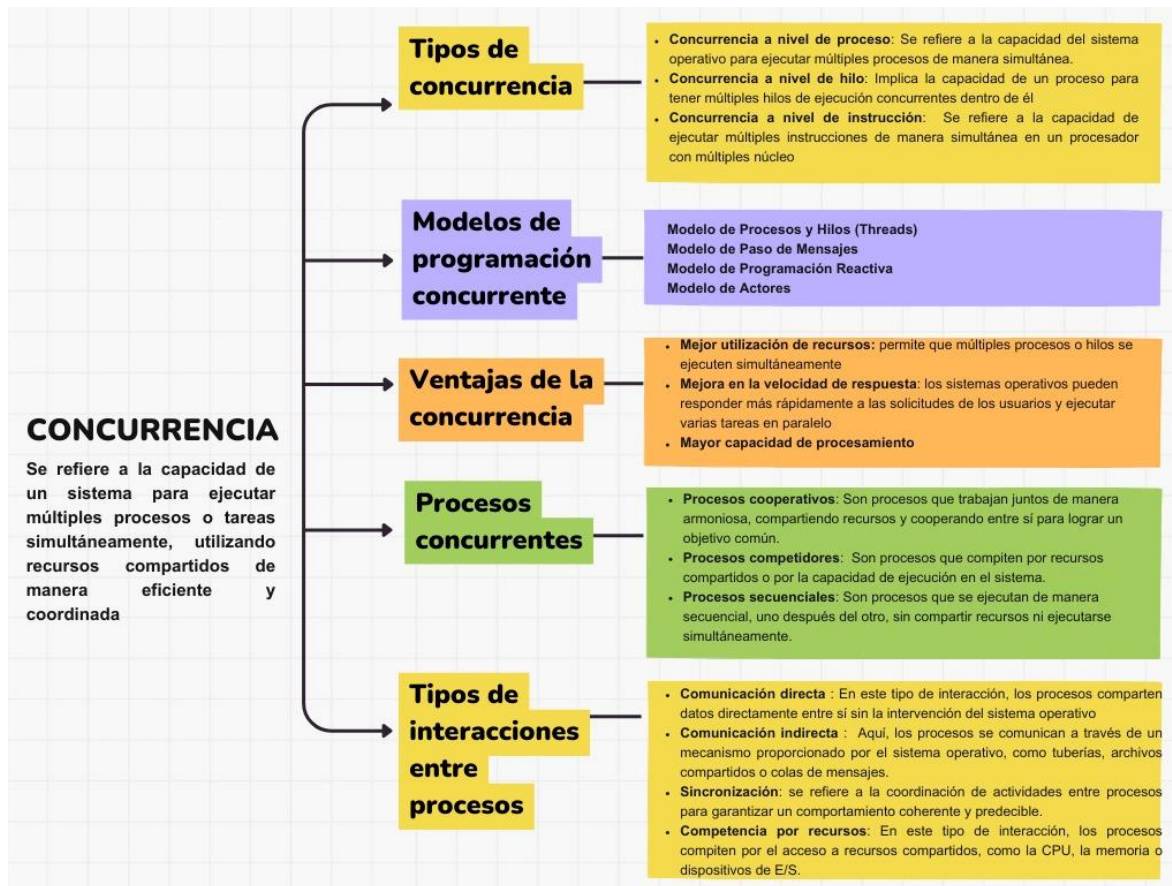
### **Ventajas de la ejecución de concurrencia**

**1. Mejor velocidad de respuesta:** los subprocesos múltiples permiten ejecutar múltiples operaciones en forma simultánea. Esto trae como consecuencia mayor velocidad de ejecución del programa. Es decir que todas las operaciones que lleva a cabo el usuario serán procesadas a la vez. Todo ello permite que la productividad del usuario sea más valiosa. Además de este modo se asegura que todos los recursos de la computadora disponibles están siendo utilizados de forma inteligente.

**2. Mejor utilización de los recursos del microprocesador:** Es sabido que con cada año que pasa, los procesadores incluyen más núcleos, y por lo tanto poder de procesamiento y velocidad. En el caso que un programa sea de un solo proceso, entonces usará un solo núcleo.

**3. Más fácil de desarrollar:** la programación concurrente es considerada como el inicio del desarrollo de Java, ofreciendo un modelo de programación consistente. En este sentido, aquellos programadores que utilizan Java en sus desarrollos pueden centrarse más en la programación y no en cómo utilizar de la mejor manera el multiproceso.

## Cuadro sinóptico con los conceptos vistos



## CONCLUSIÓN

Cómo conclusión general de nuestro equipo, sabemos que la concurrencia es la ejecución de varias tareas a la vez, así mejorando la eficiencia y el rendimiento de los sistemas informáticos. Facilita la gestión eficiente de recursos al compartir la capacidad de procesamiento, la memoria y los dispositivos de entrada/salida entre procesos o hilos. La concurrencia es crucial en entornos modernos, donde la interactividad y la respuesta en tiempo real son fundamentales.

Sin embargo, su implementación presenta desafíos, como la sincronización de recursos compartidos y la prevención de condiciones de carrera, que requieren estrategias específicas para garantizar la coherencia y la integridad de los datos. Los modelos de programación concurrente, como procesos, hilos, programación basada en eventos y otros, ofrecen enfoques variados para abordar la concurrencia.

Finalizando y para complementar, la concurrencia en el ámbito de los sistemas operativos contribuye a la creación de sistemas más robustos, adaptables y eficientes al permitir la ejecución simultánea de tareas, optimizando así la utilización de recursos y mejorando la capacidad de respuesta de los sistemas informáticos en entornos dinámicos y exigentes.

---

## BIBLIOGRAFÍA

---

- Gestión de la memoria. (2020). Plone site. <https://elpuig.xeill.net/Members/vcarceler/c1/didactica/apuntes/ud3/na9>
- Marker, G. (2022, 24 julio). ¿Qué es programación concurrente? Tecnología + Informática. [https://www.tecnologia-informatica.com/que-es-programacion-concurrente/#Ventajas\\_de\\_la\\_programaci%C3%B3n\\_concurrente%C2%A0](https://www.tecnologia-informatica.com/que-es-programacion-concurrente/#Ventajas_de_la_programaci%C3%B3n_concurrente%C2%A0)
- U. (2018, 16 diciembre). Concurrencia y Secuencialidad (Sistemas Operativos). StudentPlace. <https://studentplace98.blogspot.com/2018/09/concurrencia-ysecuencialidad-sistemas.html>
- F. (2011). Introducción a los Sistemas Operativos de Red. profefape.webnode.es. <https://profefape.webnode.es/quinto/introduccion-a-los-sistemas-operativos-de-red/>
- Lewis, S. (2021, 12 mayo). Sistema operativo de red (NOS). ComputerWeekly.es. <https://www.computerweekly.com/es/definicion/Sistema-operativo-de-red-NOS>
- Silberschatz, A., Gagne, G., & Galvin, P. B. (2002). Fundamentos de sistemas operativos (7ma Edición). Mc Graw Hill.