

**Universidad Tecnológica  
de Santiago UTESA  
Facultad Ingeniería En  
Sistemas**



**Matéria:**

Algoritmos paralelos

**Tema:**

**Investigación de conceptos sobre Algoritmos paralelos**

**Grupo:**

INF-025-001

**Nombre:**

Bryan Reyes 1-17-0760

**Dirigido a:**

Docente:

Ing. Ivan Mendoza

**26 de enero del 2025**  
**Santiago, República Dominicana**

**Realizar la investigación y exposición en clases.**

**Entregar Investigación por esta vía.**

**Investigar:**

Algoritmos

Paralelo

Algoritmos paralelos

Programación paralela

Programación concurrente

Paralelismo

Hilos

Lenguajes o Frameworks de programación que usan paralelismo o programación concurrente

Programación Reactiva (Concepto, Lenguajes o Frameworks)

MVVM

SOLID

Arquitectura Hexagonal

## **Algoritmos**

Un algoritmo es un conjunto de instrucciones bien definidas y ordenadas que resuelven un problema específico o realizan una tarea. Los algoritmos son esenciales en la programación, ya que permiten estructurar soluciones eficientes y reutilizables. Se clasifican en categorías como de búsqueda, ordenamiento, optimización, entre otros.

## **Paralelo**

El término "paralelo" hace referencia a la ejecución simultánea de múltiples tareas. En computación, se aplica en el procesamiento paralelo, donde varias operaciones se realizan al mismo tiempo para mejorar el rendimiento, particularmente en sistemas con múltiples procesadores o núcleos.

## **Algoritmos paralelos**

Los algoritmos paralelos están diseñados para dividir un problema en subproblemas independientes que pueden ejecutarse simultáneamente en diferentes procesadores o núcleos. Esto reduce significativamente el tiempo de ejecución y es especialmente útil en tareas como simulaciones, aprendizaje automático y análisis de datos a gran escala.

## **Programación paralela**

La programación paralela es una técnica que permite a los programadores escribir código que se ejecute en paralelo. Se enfoca en dividir tareas en múltiples hilos o procesos para que trabajen simultáneamente. Es común en aplicaciones de alto rendimiento y sistemas distribuidos, utilizando APIs como OpenMP, MPI o lenguajes como CUDA para GPUs.

## **Programación concurrente**

La programación concurrente gestiona múltiples tareas que pueden interactuar entre sí y ejecutarse de manera intercalada en un solo procesador o simultáneamente en varios. Es útil para manejar aplicaciones que requieren multitarea, como servidores web, y utiliza herramientas como semáforos, monitores y colas para gestionar la sincronización.

## **Paralelismo**

El paralelismo es una forma avanzada de ejecución simultánea que aprovecha la arquitectura del hardware para procesar múltiples tareas al mismo tiempo. Se divide en paralelismo a nivel de datos, tareas e instrucciones, y es fundamental en sistemas modernos, desde supercomputadoras hasta dispositivos móviles.

## Hilos

Los hilos (threads) son las unidades más pequeñas de ejecución dentro de un proceso. Permiten dividir una aplicación en tareas concurrentes que comparten recursos como memoria, aumentando la eficiencia en operaciones intensivas. Su implementación varía según el lenguaje de programación y el sistema operativo.

## Lenguajes o Frameworks de programación que usan paralelismo o programación concurrente

Lenguajes como **C**, **C++**, y **Java** incluyen soporte nativo para paralelismo y concurrencia mediante bibliotecas o APIs como `pthread` o `java.util.concurrent`. Frameworks como **MPI**, **OpenMP**, y **CUDA** son populares para programación paralela en entornos científicos e industriales. Para concurrencia, **Node.js** y **Akka** son muy utilizados en aplicaciones web y distribuidas.

## Programación Reactiva (Concepto, Lenguajes o Frameworks)

La programación reactiva se basa en manejar flujos de datos y propagación de cambios. En lugar de procesar tareas secuencialmente, reacciona a eventos o cambios en tiempo real. Frameworks como **RxJava**, **Reactor**, y **Angular** implementan este enfoque, siendo especialmente útiles en sistemas que manejan eventos asíncronos y datos en streaming.

## MVVM

MVVM (Model-View-ViewModel) es un patrón arquitectónico que organiza el código dividiendo la lógica de negocio (Model), la interfaz gráfica (View) y un intermediario que conecta ambos (ViewModel). Es popular en el desarrollo de aplicaciones modernas, especialmente en plataformas como **WPF**, **Xamarin**, y **SwiftUI**.

## SOLID

SOLID es un conjunto de cinco principios de diseño orientado a objetos que promueven la creación de software más mantenible y escalable. Los principios incluyen responsabilidad única, abierto/cerrado, sustitución de Liskov, segregación de interfaces y dependencia inversa, ofreciendo una guía clara para evitar código complejo y acoplado.

## Arquitectura Hexagonal

La arquitectura hexagonal, también conocida como arquitectura de puertos y adaptadores, separa la lógica de negocio de las dependencias externas (como bases de datos o APIs). Este enfoque mejora la capacidad de prueba, la escalabilidad y la flexibilidad del software, permitiendo cambios en los componentes externos sin afectar el núcleo del sistema.