## **Propuesta de Investigación: Paradigma "Divide y Vencerás"**

[Propuesta de Investigación: Paradigma "Divide y Vencerás"](#_3t4rqb5mxfyf) 1

[**Introducción**](#_2b1mflza0lhf) **1**

[**Objetivos de Aprendizaje**](#_eyx7jryikns4) **1**

[**1. Investigación Teórica**](#_72x8d739t3k2) **2**

[-Definición:](#_6k6w59tkjae0) 2

[Principios Básicos:](#_4h0lgqho425r) 2

[Ejemplos Clásicos:](#_8apli0auaxec) 2

[Complejidad:](#_hp7bgafm0o6f) 3

[**2. Aplicación Práctica**](#_ki90ztaeidcs) **3**

[Selección de Problemas:](#_mwrfuvowrm82) 3

[Implementación:](#_qo5hg8tt6ol) 3

[Experimentación:](#_n9h6seqczimv) 4

[Comparación:](#_raoxgrlzg3us) 4

[**3. Presentación Grupal**](#_h3ts9l2b90v9) **4**

[● Estructura:](#_m6fa9u9rtyca) 4

[● Visualización:](#_a3i35n9t1zw5) 4

[Referencias bibliográficas](#_ibr2t9w7cvbc) 4

# **Introducción**

Esta propuesta de investigación se centra en el paradigma de resolución de problemas "Divide y Vencerás", un enfoque algorítmico fundamental en ciencias de la computación. Los estudiantes tendrán la oportunidad de explorar a profundidad este concepto, aplicarlo a diversos escenarios y comprender su impacto en la eficiencia de las soluciones computacionales.

# **Objetivos de Aprendizaje**

* **Comprender a fondo:** Asimilar los principios básicos del paradigma "Divide y Vencerás".
* **Aplicar conocimientos:** Utilizar este paradigma en la resolución de problemas reales.
* **Evaluar eficiencia:** Comparar la eficiencia de algoritmos basados en "Divide y Vencerás" con otros enfoques.
* **Desarrollar habilidades:** Fortalecer habilidades de investigación, análisis y presentación.

# **1. Investigación Teórica**

## -**Definición**:

"El paradigma 'Divide y Vencerás' consiste en resolver un problema dividiéndolo en subproblemas de un tamaño similar, resolviendo recursivamente estos subproblemas y combinando las soluciones para obtener la solución al problema original. Busca una definición precisa del paradigma en libros de texto como "Introducción a los Algoritmos" de Cormen, Leiserson, Rivest y Stein.

## **Principios Básicos:**

**Dividir:** Descomponer el problema en subproblemas más pequeños del mismo tipo.

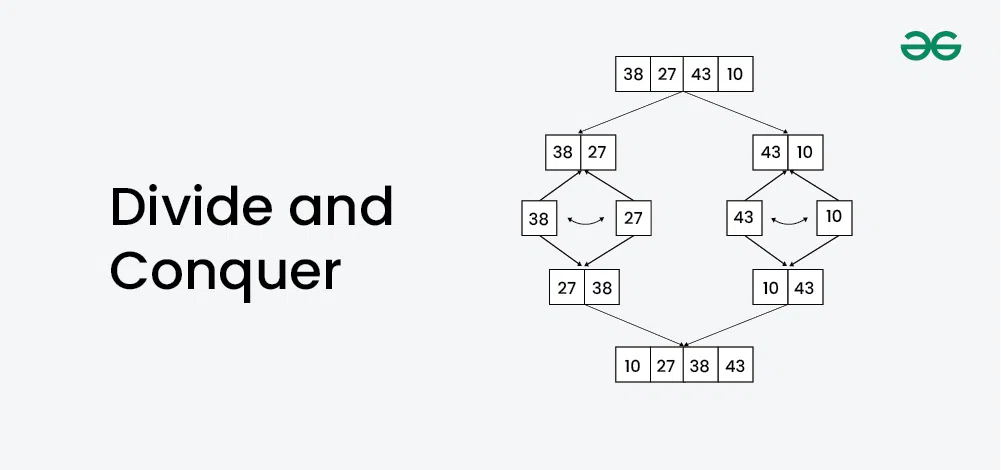
**Conquistar:** Resolver los subproblemas de forma recursiva.

**Combinar:** Combinar las soluciones de los subproblemas para obtener la solución al problema original.

## **Ejemplos Clásicos:**

* **Quicksort:** Ordena una lista dividiéndola en dos sublistas, una con elementos menores a un pivote y otra con elementos mayores, y luego ordenando recursivamente las sublistas.
* **Mergesort:** Ordena una lista dividiéndola en dos mitades, ordenando recursivamente cada mitad y luego fusionando las listas ordenadas.
* **Búsqueda Binaria:** Busca un elemento en una lista ordenada dividiendo la lista por la mitad en cada iteración.

## **Complejidad:**

* **Notación O:** Utiliza la notación O grande para expresar la complejidad de tiempo y espacio de los algoritmos.
* **Casos promedio y peor de los casos:** Analiza cómo el rendimiento del algoritmo varía en diferentes escenarios de entrada.

# **2. Aplicación Práctica**

## **Selección de Problemas:**

Identifica problemas que tengan una estructura recursiva natural, como ordenación, búsqueda, multiplicación de matrices, etc.

## **Implementación:**

Elige un lenguaje de programación (Python, C++, Java, etc.) e implementa los algoritmos.

## **Experimentación:**

Utiliza diferentes tamaños de entrada para medir el tiempo de ejecución y el consumo de memoria.

## **Comparación:**

Compara los resultados con otros algoritmos (por ejemplo, burbuja, inserción) para evaluar la eficiencia.

# **3. Presentación Grupal**

## **Estructura**:

* + Introducción al paradigma
  + Ejemplos prácticos con código y diagramas
  + Análisis de resultados (tablas, gráficos)
  + Conclusiones y discusión

## **Visualización**:

* + Utiliza diapositivas, código en vivo, animaciones para ilustrar los conceptos.

### **Referencias bibliográficas**

* **Libros de texto:**
  + Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to algorithms. MIT press.