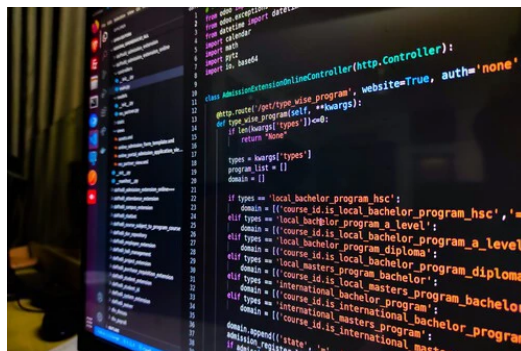


**Universidad Americana**  
**Facultad de Ingeniería y Arquitectura**

---



**Paradigma de Resolución de Problemas: “Divide y Vencerás”**



**Materia:**

Metodología y programación estructurada

**Elaborado por:**

Kimberly Zapata Espinoza  
Arellys Brigitte Obando Castillo  
Jonathan Josué Rivera Guido

**Docente**

Lcda. María del Socorro Herrera Vallejos

**Fecha 11/10/2024**

## I. Precedentes Históricos

El método "Divide y Vencerás", entendido como "una lista ordenada de objetos para facilitar su búsqueda", tiene sus orígenes en la antigua Babilonia, alrededor del 200 a.C. Sin embargo, su formulación en el contexto de la programación y los algoritmos se remonta a 1946, cuando John Mauchly publicó un artículo en el que describía su aplicación en la informática. Mauchly fue pionero en el desarrollo de técnicas que se han vuelto fundamentales en la programación moderna, resaltando la importancia de la modularidad y la estructura en la resolución de problemas computacionales.

Este paradigma es la base de numerosos algoritmos eficientes aplicables a una amplia variedad de problemas. En su acepción más amplia, "Divide y Vencerás" trasciende la mera técnica de diseño de algoritmos, considerándose incluso una filosofía general para abordar problemas en diversos campos.

## II. Principio de Rubik

El principio de Rubik establece que "un problema complejo se resuelve más fácilmente fragmentándolo en problemas cada vez más pequeños y, por lo tanto, más simples." Cada tarea o incidencia conlleva una responsabilidad única y resulta más fácil de explicar. Este enfoque implica separar un problema general en fragmentos manejables.

En el contexto de la programación, se trata de descomponer un problema difícil en partes más simples hasta que su resolución sea evidente. Finalmente, las soluciones a cada subproblema se combinan para resolver el problema original.

## III. Condiciones para la Aplicación del Método

Para que la aplicación del método "Divide y Vencerás" sea conveniente, se deben considerar las siguientes condiciones:

- **Eficiencia en la Descomposición y Combinación:** Las operaciones de dividir y combinar deben ser suficientemente eficientes, garantizando que los subproblemas sean más simples que el problema original.
- **Número Limitado de Subproblemas:** La cantidad de subproblemas generados debe ser pequeña, de modo que, tras un número finito de subdivisiones, se alcance un subproblema que pueda resolverse directamente ya que si un subproblema no es más pequeño que el original, podrías terminar en un ciclo infinito puesto que nunca estarías reduciendo la complejidad del problema.
- **Tamaño Similar de los Subproblemas:** Los subproblemas deben ser aproximadamente del mismo tamaño y no deben solaparse. Es decir, si los subproblemas se solapan, significa que pueden estar resolviendo las mismas sub-tareas, lo que no es propio de Divide y Vencerás, ya que lo ideal es que los subproblemas sean independientes entre sí, y que resolver uno no implique volver a resolver partes de otros ya que puede llevar a redundancia y a un aumento innecesario en la complejidad del algoritmo.

## IV. Etapas del Método

- ❖ **Dividir el Problema:** El primer paso consiste en plantear el problema de tal manera que se pueda descomponer en  $k$  subproblemas del mismo tipo, pero de menor tamaño. Si el tamaño de la entrada es  $n$ , se debe dividir en  $k$  subproblemas, cada

uno con una entrada de tamaño  $nk$  (donde  $1 \leq k \leq n$  y  $0 \leq nk < n$ ). Esta fase se conoce como división. Es fundamental equilibrar la carga entre los subproblemas para evitar "anomalías de funcionamiento".

- ❖ **Resolver los subproblemas:** En esta etapa, se resuelven de manera independiente todos los subproblemas, ya sea directamente (si son elementales) o de forma recursiva.
- ❖ **Combinar las soluciones:** El hecho de que el tamaño de los subproblemas sea estrictamente menor que el tamaño original del problema nos garantiza la convergencia hacia los casos elementales, también denominados casos base, que son los problemas más simples que pueden resolverse directamente y luego combinarse para obtener la solución final.

Hay que tener en cuenta que la forma en que se dividen los problemas suele ser la parte más compleja del algoritmo. Por lo tanto, el modelo puede ofrecer soluciones que son simples, pero no necesariamente las más óptimas.

## V. Consideraciones sobre Recursión

La técnica de "Divide y Vencerás" a menudo se implementa de manera recursiva, lo que conlleva a heredar sus ventajas y desventajas.

Resolver un problema mediante recursión implica que una función se llama a sí misma para repetir el proceso hasta llegar a la solución, esto puede provocar problemas de eficiencia y limitaciones de memoria.

Por un lado, la recursión ofrece un diseño simple, claro, robusto y elegante, lo que facilita la legibilidad y el mantenimiento del código. Sin embargo, puede resultar en un mayor tiempo de ejecución y un consumo más elevado de memoria en comparación con las soluciones iterativas, que utilizan estructuras de control como bucles.

## VI. Ejemplos Clásicos

Algunos algoritmos representativos del diseño "Divide y Vencerás" incluyen:

- **Algoritmos de ordenación:**
  - **Mergesort:** Este algoritmo divide una lista en dos mitades, las ordena recursivamente y luego combina las dos mitades ya ordenadas. Es un algoritmo con complejidad de tiempo  $O(n \log n)$  y es estable, lo que significa que mantiene el orden relativo de los elementos iguales.
  - **Quicksort:** Divide una lista en dos partes basándose en un pivote. Los elementos menores que el pivote se colocan en una parte y los mayores en la otra. A diferencia de Mergesort, Quicksort realiza la mayor parte del trabajo durante la división.
- **Multiplicación de números grandes:**
  - **Algoritmo de Karatsuba:** Multiplicación de dos números grandes. Divide los números en partes más pequeñas, calcula las multiplicaciones parciales, y luego las combina para obtener el producto final.
- **Análisis sintáctico:**

- **Análisis sintáctico top-down:** Utiliza el principio de divide y vencerás al descomponer una oración en sus componentes sintácticos. Uno de los enfoques más comunes en este ámbito es el análisis sintáctico recursivo, en el que se descompone una estructura gramatical en reglas más pequeñas hasta que se alcanza un resultado final.

## **VII. Aplicaciones reales y modernas**

Su uso va más allá de algoritmos clásicos como Mergesort o Quicksort. Algunos ejemplos son:

- **Inteligencia Artificial y Aprendizaje Automático:** En el reconocimiento facial, CNN divide la imagen de un rostro en regiones más pequeñas para identificar ojos, nariz y boca antes de combinarlas para identificar a la persona.
- **Optimización y Simulación:** Optimización de rutas en logística para empresas de envíos como UPS, DHL o FedEx, minimizando distancias y tiempos de entrega al dividir y seleccionar rutas más óptimas.
- **Procesamiento de Imágenes y Gráficos:** En el cine, el renderizado de una escena con iluminación compleja, sombras y reflejos se divide en bloques para acelerar el proceso de generación de la imagen final.

## **VIII. Conclusión:**

El paradigma de "Divide y Vencerás" es una estrategia fundamental en la resolución de problemas complejos, tanto en informática como en otros campos. Esto, debido a su facilidad para descomponer un problema inicial en subproblemas más pequeños y manejables, facilitando su resolución. Esta metodología no solo optimiza el proceso al reducir la complejidad paso a paso, sino que también permite alcanzar soluciones más eficientes. De este modo, el uso de esta técnica se ha convertido en un método clave para afrontar grandes desafíos de forma sistemática y eficiente, logrando el objetivo de brindar soluciones y brindar la comprensión del código en la práctica.