

# MODELO DE EJECUCIÓN DE PROGRAMAS

AUTOR: WALTER MADRIGAL CHAVES

NOVIEMBRE: 2020



## Contenido

INTRODUCCIÓN.....	2
PROCESOS DE RESOLUCIÓN DE PROBLEMAS .....	3
ANÁLISIS PARA IDENTIFICAR EL PROBLEMA.....	4
DISEÑAR ALTERNATIVAS DE SOLUCIÓN.....	5
ELEGIR UNA ALTERNATIVA DE SOLUCIÓN.....	5
DESARROLLAR /IMPLEMENTAR LA SOLUCIÓN.....	5
Algoritmos .....	6
Diagrama de flujo .....	6
Codificación del algoritmo .....	7
EVALUAR LA SOLUCIÓN .....	8
PATRÓN DE DISEÑO .....	8
TÉCNICAS DE RESOLUCIÓN DE PROBLEMAS: DIVIDE Y VENCERÁS .....	9
Ventajas: .....	11
CONCLUSIONES Y RECOMENDACIONES .....	12
REFERENCIAS BIBLIOGRÁFICAS.....	13



## INTRODUCCIÓN

Es muy común que en el transcurso de nuestras vidas nos enfrentemos a diferentes tipos de problemas, estos pueden diferir en profundidad, temática y entorno involucrado, sin embargo, todos convergen en que la forma de resolverlos puede mejorar si se divide en pequeños problemas.

El mundo tecnológico y en especial el apartado de programación de sistemas no es ajeno a los problemas, es más, se podría decir que el nacimiento de la programación es para resolver problemas. Por lo que se debe esquematizar este proceso mediante técnicas establecidas que harán el trabajo más fácil.

En la siguiente lectura se analizarán algunas etapas que se recomiendan seguir en la búsqueda de soluciones a problemas. Estas ayudarán en primera instancia a marcar el rumbo a seguir, para posteriormente medir las posibles variables que interfieran, implementar soluciones y por último realizar las pruebas correspondientes.

## PROCESOS DE RESOLUCIÓN DE PROBLEMAS

Para el sitio web de la Real Academia Española un problema es:

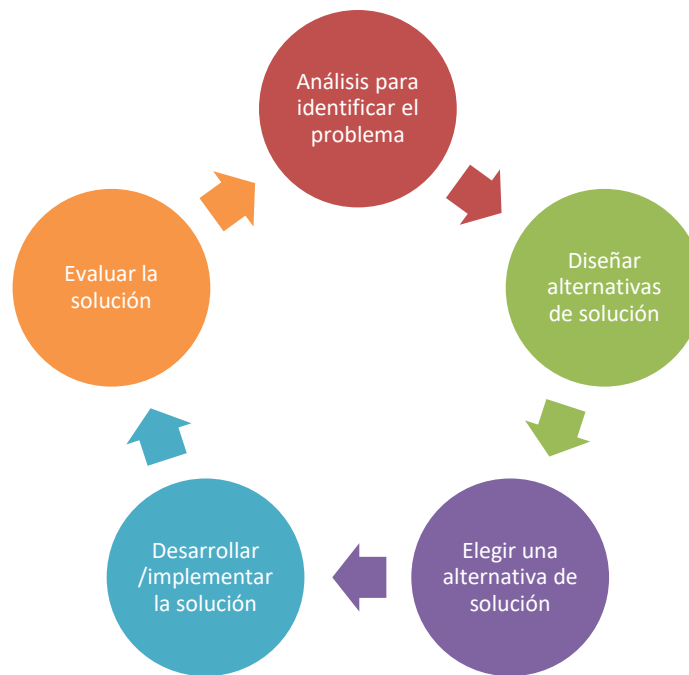
- Una cuestión que se trata de aclarar.
- Proposición o dificultad de solución dudosa.
- Conjunto de hechos o circunstancias que dificultan la consecución de algún fin.
- Disgusto, preocupación.
- Planteamiento de una situación cuya respuesta desconocida debe obtenerse a través de métodos científicos.

Claramente los problemas tienen diferentes enfoques según sea su naturaleza o su entorno; los sociales, económicos y culturales son algunos ejemplos de ellos. En esta lectura se analizará la metodología para solventar aquellos problemas cuya solución dependen de un resultado tecnológico o computacional.

La concepción de un proyecto suele surgir a partir de una necesidad que puede mapearse como un problema. Realizar un mantenimiento, una modificación, una mejora, un reemplazo o una creación de software, son algunos de los ejemplos de generadores de problemas.

La metodología de solución de problemas tiene como propósito estandarizar el proceso de búsqueda de soluciones, esta metodología cuenta con cinco etapas: identificar el problema, diseñar alternativas de solución, elegir la mejor alternativa, desarrollar la alternativa de solución y la evaluación.

Imagen 1 Fases de la metodología para resolver problemas



Fuente: Elaboración propia

## ANÁLISIS PARA IDENTIFICAR EL PROBLEMA

Esta primera etapa es muy importante es aquí en donde se concibe la idea, donde se plantea el objetivo y se plasma el rumbo a seguir, una concepción errónea significa un proyecto fallido. En esta etapa se deben tener claros algunos temas importantes como recursos disponibles, antecedentes, entorno, políticas institucionales, necesidad real, tiempos, categoría del problema a resolver (alto-medio-bajo), entre otros.

Técnicamente y dando el enfoque en desarrollo de sistemas, lo que se hace en este primer encuentro es delimitar el problema, definiendo los datos de entrada que serán procesados para obtener una salida o resultado. Es importante tener claro lo que el sistema tiene que hacer, cuales datos con su respectivo formato necesita para realizar las tareas, de donde proviene la información de entrada, como se obtienen esos datos y cómo interactúan entre sí para obtener el resultado.

Siempre se debe estudiar y analizar las posibles opciones que se den ante un cambio de escenario o ante la materialización de un riesgo. Los riesgos no pueden pasarse por alto, deben contar con mecanismos de mitigación.

## **DISEÑAR ALTERNATIVAS DE SOLUCIÓN**

No siempre la primera idea es la mejor, lo ideal es plasmar una serie de alternativas que convergen en la solución del problema. Hay que definir que riesgos conlleva cada una y su viabilidad, algunas veces las soluciones más sencillas son las adecuadas.

## **ELEGIR UNA ALTERNATIVA DE SOLUCIÓN**

Después de obtener una lista completa de alternativas, es el momento de analizarlas, ver sus pros y contras para tomar decisiones, se debe seleccionar la que se considere la mejor para obtener el resultado esperado. La elección de la alternativa impactara en el desarrollo de la solución, por lo que es necesario hacer un análisis adecuado de cada una y optar por la más adecuada, sencilla o viable.

Debido a la importancia para el rumbo del proyecto esta decisión debe ser colegiada e involucrar a varios personajes como lo son los usuarios expertos, equipo solicitante y dueños de los recursos.

## **DESARROLLAR /IMPLEMENTAR LA SOLUCIÓN**

Es la etapa más laboriosa, en donde se plasma lo planificado en las fases anteriores. Cuando se trata de un programa informático es necesario ingresar los datos de entrada para que realice el proceso y se obtenga el resultado o salida esperado.

Se utilizan dos herramientas muy importantes para el desarrollo de sistemas que son: los algoritmos y diagramas de flujo.



## Algoritmos

Los algoritmos son un grupo de ordenes finitas y consecutivas que llevan a cabo una serie de procesos, con el fin dar respuesta a determinados problemas, permiten al programador resolver el problema antes de escribirlo en un lenguaje de programación entendible por la computadora.

Para desempeñar su objetivo y concretar credibilidad en su resultado, los algoritmos deben cumplir con ciertas características fundamentales como lo son:

- Deben ser secuenciales, sus instrucciones deben procesarse una después de la otra.
- Precisos, deben ser objetivos al resolver el problema.
- Ordenados.
- Finitos, deben tener un número determinado de pasos.
- Concretos, deben mostrar un resultado al problema resuelto.

## Diagrama de flujo





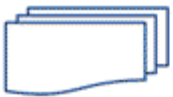





Los diagramas de flujo son un conjunto de imágenes que tienen significados específicos, sirven para realizar una representación gráfica de algoritmos o procesos, se utilizan para una mejor visibilidad de los procesos, entrada y salidas en algoritmo.

Debe cumplir con ciertas características:

- Ya existe un stock de imágenes o símbolos que se pueden utilizar, no se debe innovar incluyendo nuevos elementos.
- Todos los símbolos han de estar conectados.
- A un símbolo de proceso pueden llegarle varias líneas.

- A un símbolo de decisión pueden llegarle varias líneas, pero sólo saldrán dos (Si o No, Verdadero o Falso).
- A un símbolo de inicio nunca le llegan líneas.
- De un símbolo de fin no parte ninguna línea.

Imagen 2 Elementos base del diagrama de flujo

SÍMBOLO	SIGNIFICADO	SÍMBOLO	SIGNIFICADO
	<b>Terminal:</b> Indica el inicio o la terminación del flujo del proceso.		<b>Actividad:</b> Representa la actividad llevada a cabo en el proceso.
	<b>Decisión:</b> Señala un punto en el flujo donde se produce una bifurcación del tipo "Sí" – "No".		<b>Documento:</b> Documento utilizado en el proceso.
	<b>Multidocumento:</b> Refiere un conjunto de documentos. Por ejemplo, un expediente.		<b>Inspección / Firma:</b> Aplicado en aquellas acciones que requieren de supervisión.
	<b>Conector de un Proceso:</b> Conexión o enlace con otro proceso, en el que continúa el diagrama de flujo. Por ejemplo, un subproceso.		<b>Archivo:</b> Se utiliza para reflejar la acción de archivo de un documento o expediente.
	<b>Base de Datos:</b> Empleado para representar la grabación de datos.		<b>Línea de Flujo:</b> Indica el sentido del flujo del proceso.

Fuente: <https://www.aiteco.com/>

## Codificación del algoritmo

En esta sección es cuando se traslada las sentencias realizadas en el algoritmo al lenguaje de programación elegido. Hay que recordar que todos los lenguajes tienen su estructura gramatical específica, por lo que no se puede establecer un esquema unificado para entender este punto.

El programador debe conocer la sintaxis específica del lenguaje con la finalidad de que este proceso sea más expedito y evitar errores en la programación.



## **EVALUAR LA SOLUCIÓN**

Esta es la etapa de control de calidad, aquí se somete al objeto o sistema construido a distintas pruebas, mediante las cuales se comprueba que funciona y que responde adecuadamente a la definición del problema. Lo importante en este punto es la retroalimentación que se puede generar para mejorar el sistema creado.

## **PATRÓN DE DISEÑO**

Los patrones de diseño son formas estandarizadas que se aplican para resolver problemas comunes en el desarrollo de software. Para que un posible patrón sea considerado como tal, debe tener ciertas características. La primera es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reutilizable, lo que significa que es aplicable a diferentes problemas y distintas circunstancias.

Aplicar esta técnica correctamente genera muchas ventajas a los desarrolladores, las siguientes son algunas de ellas:

- Estandarizar la resolución de ciertos problemas reiterativos.
- Se evita el uso innecesario de tiempo y recursos, haciendo cosas que ya están hechas.
- Crear un catálogo de problemas y soluciones.
- Concentrar y simplificar el aprendizaje de las buenas prácticas.
- Proporcionar un vocabulario común entre desarrolladores.

Tabla 1 Tipos de patrones

Tipo de Patrón	Funcionalidad
<b>Creacionales</b>	Estos patrones buscan despreocupar al sistema de cómo sus objetos son creados o compuestos. Encapsula la lógica de la creación de objetos, con esto oculta los detalles de cada uno. Algunos ejemplos son: Factory Method, Prototype, Builder, Abstract Factory
<b>Comportamiento</b>	Este tipo de patrones ayuda a definir la forma en que los objetos interactúan entre sí. Algunos de los más conocidos son: Command, Observer, Strategy, entre otros.
<b>Estructurales</b>	Este tipo de patrones definen la forma en la que los objetos se componen. Algunos ejemplos son: Facade, Decorator, Composite, entre otros

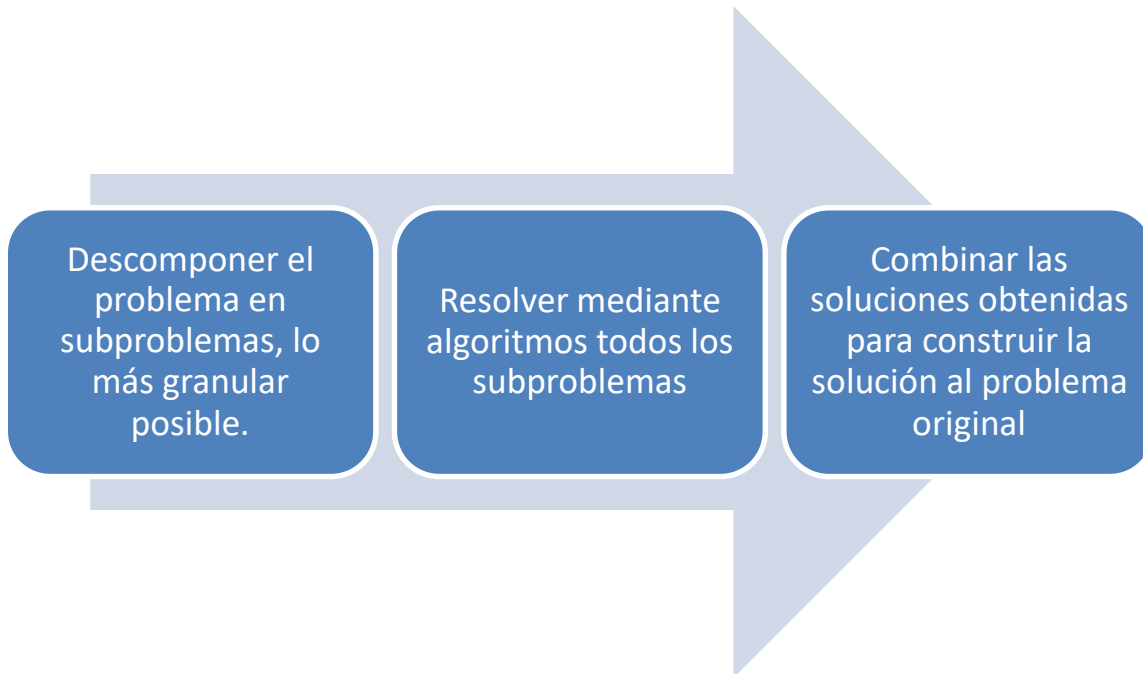
Fuente: Elaboración propia

## TÉCNICAS DE RESOLUCIÓN DE PROBLEMAS: DIVIDE Y VENCERÁS

El divide y vencerás es una técnica que puede ser utilizada en todas las áreas de nuestra vida. El mundo computacional no escapa a esta tendencia, en donde más que una técnica se considera un paradigma en la programación, consiste en realizar una segmentación del problema original en varios subproblemas y mediante la programación modular, crear algoritmos que solucionen esos subproblemas.

Los tres pasos para aplicar esta técnica son:

*Imagen 3 Pasos de la técnica: Divide y vencerás*

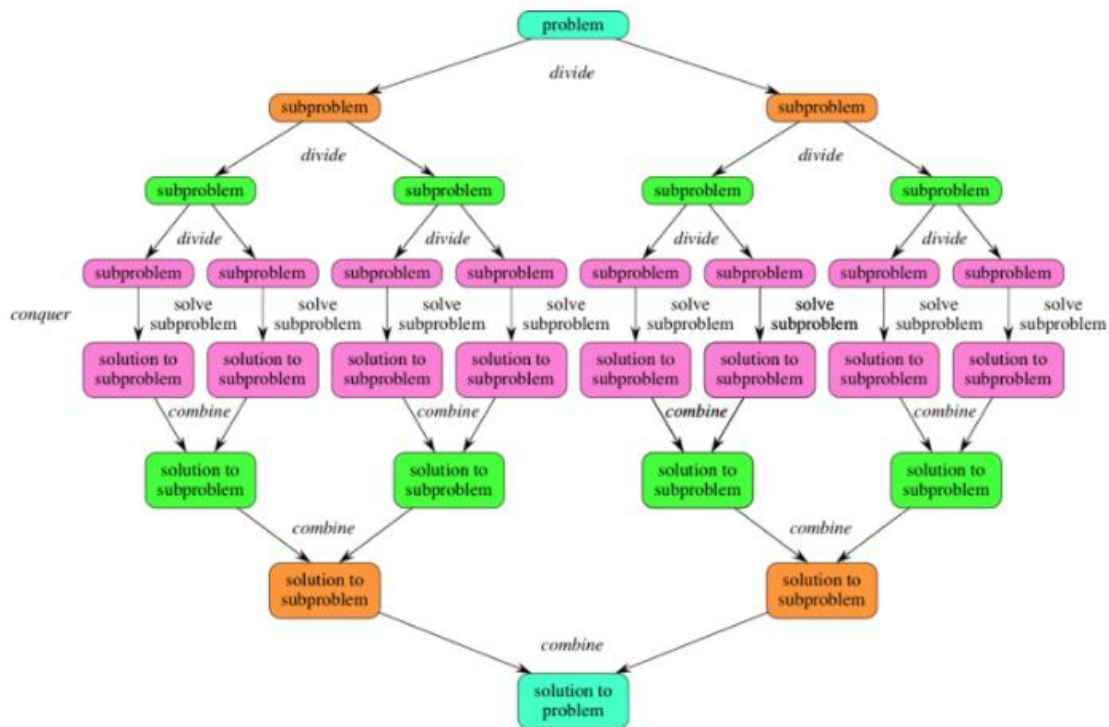


Fuente: Elaboración propia

Para que esta técnica se pueda aplicar es necesario:

- El problema original debe poder dividirse fácilmente en dos o más subproblemas.
- Los subproblemas resultantes deben de ser del mismo tipo que el problema original, pero con una resolución más sencilla.
- La solución de un subproblema debe obtenerse independientemente de los otros.
- Necesitamos un método base para resolver los problemas de tamaño pequeño.

Imagen 4 Forma general de la técnica Divide y vencerás



Fuente: <https://www.aiteco.com/>

La imagen cuatro, muestra de una forma más clara la aplicación de la técnica divide y vencerás, en primera instancia está el problema original, que se divide en dos subproblemas, estos a su vez se van granulando en más subproblemas. Al final se unen las soluciones para resolver el problema principal.

### Ventajas:

- Permite resolver problemas complejos.
- Optimización de recursos.
- Eficiencia del algoritmo, los algoritmos pequeños son más eficientes.
- Acceso a memoria, los algoritmos que siguen el paradigma divide y vencerás, tienden naturalmente a hacer un uso eficiente de las memorias cachés.
- Control del redondeo: un algoritmo de divide y vencerás podría dar resultados más exactos.



## CONCLUSIONES Y RECOMENDACIONES

- En el proceso de solución de problemas es indispensable abarcar las cinco etapas mencionadas en esta lectura, obviar alguna aumenta las posibilidades de un fracaso en el proyecto.
- La primera etapa es la más importante, marcar el rumbo a seguir no es fácil, pero es necesario, durante todo el proceso que conlleva la búsqueda de la solución hay que estar pendientes de los plasmado en esta primera etapa para no perder el objetivo del proyecto.
- El uso de algoritmos permite expresar, de una forma clara, la manera en que un problema debe ser resuelto. Los elementos que lo componen son característicos de la resolución de problemas con computadora.
- La elección de una alternativa debe ser cuidadosa, se debe analizar todas las variables y evitar sorpresas a medio camino. Apóyense en los que conocen el giro de negocio o los conocedores del entorno, suelen ser apoyos muy importantes.
- Aunque sea un poco tedioso, realice los algoritmos y diagramas de flujo necesarios para resolver el problema. Esto ayuda a prevenir errores al momento de la codificación.
- La ejercitación es la única herramienta para poder comprender y descubrir la verdadera potencialidad de las estructuras de control. Resulta fundamental alcanzar un total entendimiento del funcionamiento de estas estructuras para lograr expresar soluciones más complejas que los ejemplos aquí planteados.
- Los problemas migran de acuerdo al entorno en donde se ubican, es por esto que la retroalimentación es muy importante, en muchas ocasiones a las soluciones hay que realizarle las mejoras que el medio indique.

## REFERENCIAS BIBLIOGRÁFICAS

- Alteco Consultores*. (13 de 11 de 2020). Obtenido de <https://www.aiteco.com/diagrama-de-flujo/>
- García Vidal, G. (2012). *El proceso de solución de problemas* .
- Kölling , M., & Barnes, D. (2017). *Programación orientada a objetos con Java usando BlueJ (6a. ed.)*. Pearson Educación.
- Ramírez Langarica, A. (2016). *Informática 2* . Pearson Educación .
- Real Academia Española. (11 de 12 de 2020). *Diccionario de la lengua española*. Obtenido de <https://dle.rae.es>



[www.usanmarcos.ac.cr](http://www.usanmarcos.ac.cr)

San José, Costa Rica