

Contenido

[Objetivo. 3](#_Toc523868729)

[Introducción. 3](#_Toc523868730)

[**Desarrollo** 6](#_Toc523868731)

[**Conclusión** 10](#_Toc523868732)

[**Bibliografías.** 10](#_Toc523868733)

# Objetivo.

Elaborar algoritmos correctos y eficientes en la solución de problemas siguiendo las etapas de Análisis y Diseño pertenecientes al Ciclo de vida del software.

# Introducción.

Un problema informático se puede definir como el conjunto de instancias al cual corresponde un conjunto de soluciones, junto con una relación que asocia para cada instancia del problema un subconjunto de soluciones (posiblemente vacío).

Para poder solucionar un problema nos apoyamos en la Ingeniería de Software que de acuerdo a la IEEE se define como “La aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software". Por lo que el uso y establecimiento de principios de ingeniería sólidos, son básicos para obtener un software que sea económicamente fiable y funcione eficientemente.

La Ingeniería de Software provee métodos que indican cómo generar software. Estos métodos abarcan una amplia gama de tareas:

Planeación y estimación del proyecto.

Análisis de requerimientos del sistema y software.

Diseño de la estructura de datos, la arquitectura del programa y el procedimiento algorítmico.

Codificación.

Pruebas y mantenimiento (validación y verificación).

En el contexto matemático, los algoritmos son una serie de normas o leyes específicas que hace posible la ejecución de actividades, cumpliendo una serie de pasos continuos que no le originen dudas a la persona que realice dicha actividad. Los algoritmos se pueden expresar de diversas formas: lenguaje natural, lenguaje de programación, pseudocódigo y diagramas de flujo.

Características de los algoritmos

Los pasos de un algoritmo deben ser simples y exentos de ambigüedades (diferentes significados), deben seguir un orden cuidadosamente prescrito, deben ser efectivos y deben de resolver el problema en un número finito de pasos.

La resolución de problemas mediante una computadora.

Consiste en dar una adecuada formulación de pasos precisos a seguir. Si se piensa en la forma en que una persona indica a otra como resolver un problema, se verá que habitualmente se utiliza un lenguaje común y corriente para realizar la explicación, quizá entremezclado con algunas palabras técnicas. Esto es un riesgo muy grande. Los que tienen cierta experiencia al respecto saben que es difícil transmitir el mensaje y por desgracia, con mucha frecuencia se malinterpretan las instrucciones y por lo tanto se ejecuta incorrectamente la solución obteniéndose errores.

Etapas en la resolución de problemas con computadora La resolución de problemas utilizando como herramienta una computadora no se resume únicamente en la escritura de un programa, sino que se trata de una tarea más compleja. El proceso abarca todos los aspectos que van desde interpretar las necesidades del usuario hasta verificar que la respuesta brindada es correcta.

Las etapas son las siguientes:

Análisis del problema

Se analiza el problema en su contexto del mundo real. Deben obtenerse los requerimientos del usuario. El resultado de este análisis es un modelo preciso del ambiente del problema y del objetivo a resolver. Dos componentes importantes de este modelo son los datos a utilizar y las transformaciones de los mismos que llevan al objetivo.

Diseño de una solución

Una técnica de diseño en la resolución de problemas consiste en la identificación de las partes (subproblemas) que componen el problema y la manera en que se relacionan. Cada uno de estos subproblemas debe tener un objetivo específico, es decir, debe resolver una parte del problema original. La integración de las soluciones de los subproblemas es lo que permitirá obtener la solución buscada.

Especificación de algoritmos

La solución de cada subproblema debe ser especificada a través de un algoritmo.

Esta etapa busca obtener la secuencia de pasos a seguir para resolver el problema.

La elección del algoritmo adecuado es fundamental para garantizar la eficiencia de

la solución.

Escritura de programas

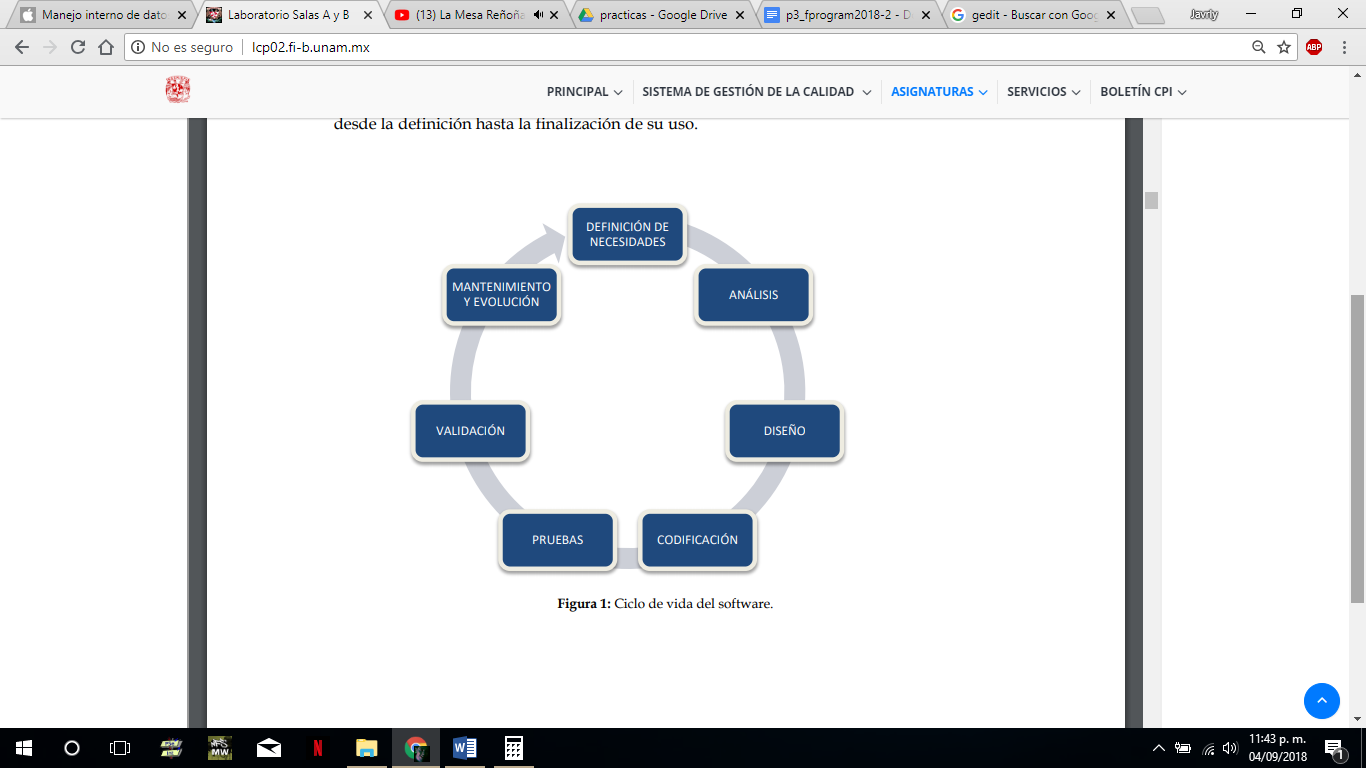
Un algoritmo es una especificación simbólica que debe convertirse en un programa real sobre un lenguaje de programación concreto. A su vez, un programa escrito en un lenguaje de programación determinado (ej: Pascal, Ada, etc) es traducido automáticamente al lenguaje de máquina de la computadora que lo va a ejecutar.

Esta traducción, denominada compilación, permite detectar y corregir los errores sintácticos que se cometan en la escritura del programa.

Verificación

Realizar una verificación exhaustiva de todas las posibles condiciones de ejecución de un sistema de software. La facilidad de verificación y la depuración de errores de funcionamiento del programa conducen a una mejor calidad del sistema y es un objetivo central de la Ingeniería de Software.

Ciclo de vida del software.

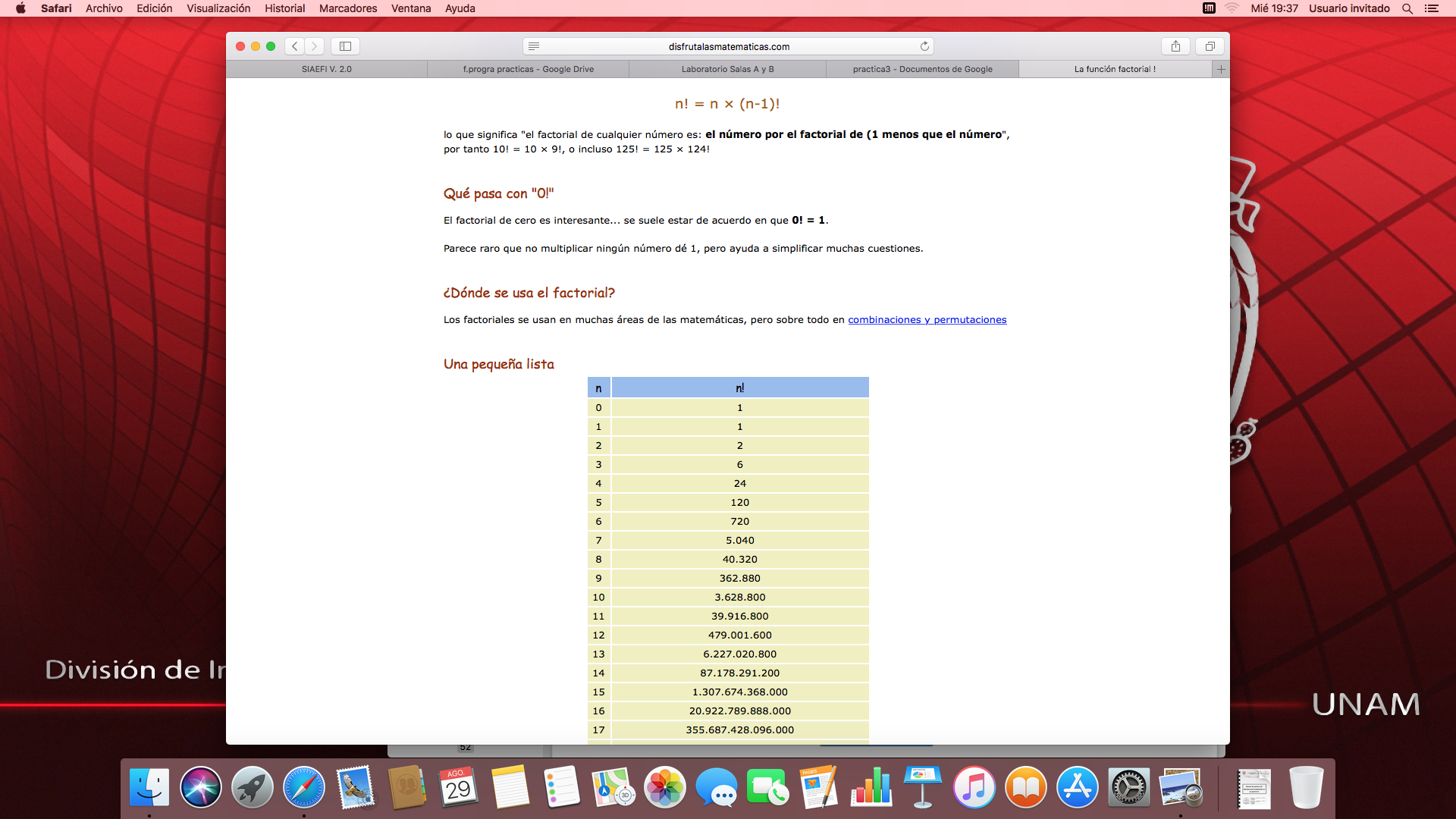


Que es la teoría de la computabilidad?

La teoría de la computabilidad es la parte de la computación que estudia los problemas de decisión que pueden ser resueltos con un algoritmo.

Las principales características con las que debe cumplir un algoritmo son:

* Preciso: Debe indicar el orden de realización de paso y no puede tener ambigüedad
* Definido: Si se sigue dos veces o más se obtiene el mismo resultado.
* Finito: Tiene fin, es decir tiene un número determinado de pasos.
* Correcto: Cumplir con el objetivo.
* Debe tener al menos una salida y esta debe de ser perceptible
* Debe ser sencillo y legible
* Eficiente: Realizarlo en el menor tiempo posible
* Eficaz: Que produzca el efecto esperado

Desarrollo como calcular el numero factorial

# **Desarrollo**

***algoritmo para verificar si un número es positivo o negativo***

DATOS DE ENTRADA: Número real.

DATOS DE SALIDA: La validación de si el número es positivo DOMINIO: Todos los números reales.

Análisis: número positivo mayores a cero, número negativo menor a cero.

Restricción: El número cero es neutro, no admitir el 0.

Variables: n=número

1. inicio
2. Pedir un número
3. leer el número
4. sí n=0 regresar al paso 1.
5. sí, n>0 imprimir pantalla “número positivo”, sino imprimir “número negativo”, ir al paso 7
6. sí, n<0 imprimir pantalla “número negativo”, sino imprimir “número positivo”, ir al paso 7
7. FIN.

|  |  |  |  |
| --- | --- | --- | --- |
| Iteración | entrada | proceso | salida |
| 1 | 22 | 22>0 | Positivo |
| 2 | 69 | 69>0 | Positivo |
| 3 | -4 | -4<0 | Negativo |

**Obtener el mayor de dos números dados.**

RESTRICCIONES: Los números de entrada deben ser diferentes.

DATOS DE ENTRADA: Número real.

DATOS DE SALIDA: La impresión del número más grande.

DOMINIO: Todos los números reales.

Análisis: dos variables y; No deben ser iguales

1. Inicio
2. Pedir un número, será variable x
3. Pedir segundo número. será variable y
4. leer x, y
5. sí x=y regresar a paso 1
6. sí x>e imprimir x” Es mayor a “y, ir paso 7
7. sí x<e imprimir y “Es mayor a, ir paso 7
8. fin

|  |  |  |  |
| --- | --- | --- | --- |
| Iteración | entrada | operación | salida |
| 1 | x=5 y=3 | x>y | 5 “es mayor a”3 |
| 2 | x=7 y=9 | x<y | 9 “es mayor a”7 |
| 3 | x=6 y=22 | x<y | 22 “es mayor a”6 |

**Obtener la factorial de un número dado.**

La factorial de un número está dado por el producto de ese número por cada uno de los números anteriores hasta llegar a 1. La factorial de 0 (0!) es 1.

**Calcular la longitud de una circunferencia a partir de su radio**.

(Usar la fórmula C = 2πr )

Análisis.

Variables: C= salida, , r = radio

Restricciones: no hay radios negativos o con valor 0

Algoritmo.

1.- Inicio

2.- Pedir radio

3.- Leer radio

4.- Realizar operación

5.-Imprimir “Radio es igual a” C

6.-Fin

|  |  |  |  |
| --- | --- | --- | --- |
| Iteración | Entrada | Operación | Salida |
| 1 | 4 |  | “Radio es igual a” 25.13 |
| 2 | 6 |  | “Radio es igual a” 37.69 |
| 3 | 8 |  | “Radio es igual a” 50.26 |

**Calcular la distancia entre dos puntos**.

(Sea P1 (a1, b1) y P2 (a2, b2), hacer uso de

Analisis

P1= a1,b1

P2= a2,b2

Variables: D= distancia entre dos puntos; A=resta de a1,a2; B=resta de b1,b2 ; R= radical

Algoritmo

1.-Inicio

2,-Pedir coordenadas de P1

3.-Pedir coordenadas de P2

4.- ,

5.- A^2 +B^2 R

6.-

7.-Imprimir D

8.-Fin

|  |  |  |  |
| --- | --- | --- | --- |
| Iteración | Entrada | Operaciones | Salida |
| 1 | P1(3,6) P2(8,2) | , *6*  403 | 6.40 |
| 2 | P1(2,4) P2(4,9) | , *4* | 5.38 |
| 3 | P1(5,5) P2(2,5) | , *5* | 3 |

**Leer 2 números y verificar si son divisibles, o el resultado no existe, o es infinito**.

(Considere que los números deben ser enteros)

Análisis.

x/0 = tiende a infinito

0/y = resultado no existe

Variables: x, y, r=resultado E=entero,

1.-Inicio

2.-Pedir numerador X

3.-Pedir denominador Y

4.- Leer x,y

5-x=0? Si, “imprimir resultado no existe”; No ir paso 5

6.-y=0? Si, tiende a infinito; No ir paso 6

7.-x/y

8.-z=E? Si, imprimir “los números son divisibles” ; No, “Los números no son divisibles”

9.-Fin

|  |  |  |  |
| --- | --- | --- | --- |
| Iteración | entrada | operaciones | Salida |
| 1 | X=2 Y=6 | 2/6=0,33333 | “los números no son divisibles” |
| 2 | X=3 Y=0 | 3/0 | “Tiende al infinito” |
| 3 | X=1 Y=8 | 1/8=0.125 | “Los números no son divisibles” |

**Leer un número y verificar si un número es par o impar.**

Análisis.

Un número es par cuando es la división entre 2 es un entero.

Variable: x=número, r=resultado, e=entero

1.-Inicio

2.-Pedir número

3.-Leer x

4.-

5.-r =e ? si, imprimir “Es par”; No, Imprimir “ Es impar”

6.-Fin

|  |  |  |  |
| --- | --- | --- | --- |
| Iteración | Entrada | Operación | Salida |
| 1 | 6 | 6/2=3 | Par |
| 2 | 2480 | 2480/2=1240 | Par |
| 3 | 25 | 25/2=12,25 | Impar |

**Leer del número 1 al 50 e indicar cuales números son múltiplos de**

**3.**

Análisis.

Un número es divisible entre 3 si al dividirlo es un entero

Variables: acum, cont, e=entero

1.-Inicio

2.-cont=0,cont ++,

3.-acum=50

4.-acum/3

5.-r=e? sí, Enteros No, no enteros

6.- Fin

|  |  |  |  |
| --- | --- | --- | --- |
| Iteración | Entrada | Operación | Salida |
| 1 | 1  2  3  4  .  .  .  50 | ½=0.5  2/2=1  3/2=1.5  4/2=2  .  .  .  50/2=25 | “No entero  “Entero”  “No entero”  “Entero”  .  .  .  “Entero” |
| 2 | 1  2  3  4  .  .  .  50 | ½=0.5  2/2=1  3/2=1.5  4/2=2  .  .  .  50/2=25 | “No entero  “Entero”  “No entero”  “Entero”  .  .  .  “Entero” |

# **Conclusión**

Los algoritmos suelen ser difíciles, pero una herramienta fuertísima para nuestro desarrollo académica, además de ser hasta cierto punto divertidos.

**Bibliografías.**

* [**http://conceptodefinicion.de/algoritmo/**](http://conceptodefinicion.de/algoritmo/)
* **http://ing.unne.edu.ar/pub/informatica/Alg\_diag.pdf**
* **http://correo.uan.edu.mx/~iavalos/FP/FP1.html**
* **http://weblidi.info.unlp.edu.ar/catedras/ingreso/Material2014/IAI/Cap1.pdf**