|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | ING. KARINA GARCIA MORALES |
| *Asignatura:* | FUNDAMENTOS DE PROGRAMACION |
| *Grupo:* | 1121 |
| *No de Práctica(s):* | PRACTICA No. 3 |
| *Integrante(s):* | DAVID SAN JUAN CRUZ |
|  |  |
|  |  |
| *Semestre:* | 2019 – I |
| *Fecha de entrega:* | 05 - 09 - 2018 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

**TÍTULO DE LA PRACTICA:**

Solución de problemas y algoritmos.

**OBJETIVOS:**

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

**DESARROLLO DE LA PRACTICA:**

**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

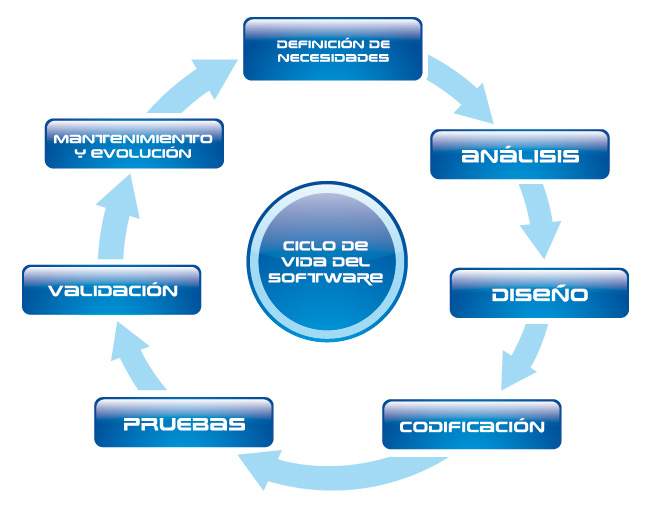
métodos abarcan una amplia gama de tareas.

**Ciclo de vida del software**

La ISO (International Organization for Standarization) en su norma 12207 define al ciclo

de vida de un software como:

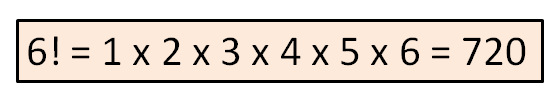
Un marco de referencia que contiene las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando desde la definición hasta la finalización de su uso.



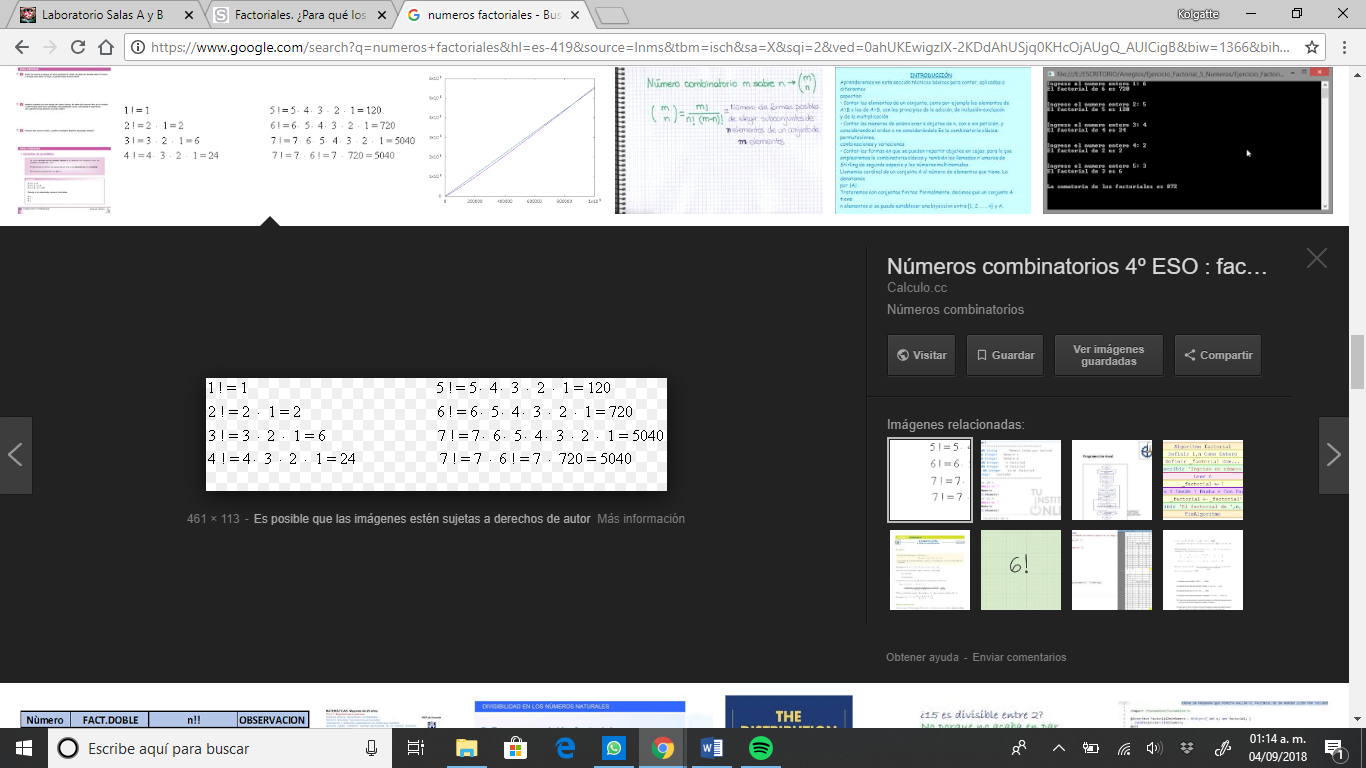
**Número Factorial**

La función factorial se representa con un signo de exclamación “**!”** detrás de un número. Esta exclamación quiere decir que hay que multiplicar todos los números enteros positivos que hay entre ese número y el 1.

**Por ejemplo:**



A este número**6!** le llamamos generalmente **“6 factorial”,** aunque también es correcto decir **“factorial de 6”.**



**LA TEORIA DE LA COMPUTABILIDAD**

La teoría de la computabilidad es la parte de la computación que estudia los problemas de decisión que puede ser resueltos con un algoritmo o equivalentemente con la llamada máquina de Turing.

El propósito inicial de la teoría de la computabilidad es hacer precisa la noción intuitiva de función calculable, esto es, una función cuyos valores pueden ser calculados de forma automática o efectiva mediante un algoritmo. Así podemos obtener una compresión más clara de esta idea intuitiva; y solo de esta forma podemos explorar matemáticamente el concepto de computabilidad y los conceptos relacionadas en ella, tales como decibilidad, etc. Surge así una teoría que n producirá resultados positivos y negativos.

**CARACTERISTICAS DE LOS ALGORITMOS**

Un algoritmo se define como un conjunto de reglas, expresadas en un lenguaje específico,

para realizar alguna tarea en general, es decir, un conjunto de pasos, procedimientos o

acciones que permiten alcanzar un resultado o resolver un problema. Estas reglas o pasos

pueden ser aplicados un número ilimitado de veces sobre una situación particular. 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

**SOLUCIÓN DEL PRIMER PROBLEMA CON SU CORRESPONDIENTE ALGORITMO Y PRUEBA DE ESCRITORIO.**

PROBLEMA: Determinar si un número dado es positivo o negativo.

RESTRICCIONES: El número no puede ser cero.

DATODS DE ENTRADA: Número real.

DATOS DE SALIDA: La validación de si el número es positivo.

DOMINIO: Todos los números reales.

SOLUCIÓN:

1 Inicio

2 Leer un número real

3 Si el número es 0 ve al final

4 Si el número es diferente de 0 se valida dos opciones:

4.1 Mayor a 0 es positivo, entonces imprime pantalla

4.2 Menor que 0 es negativo entonces imprime pantalla

5 Imprime pantalla

6 Final

|  |  |  |
| --- | --- | --- |
| **ITERACIÓN** | **X** | **SALIDA** |
| 1 | 5 | El número 5 es positivo |
| 2 | 0 | Termina mi programa |
| 3 | -10 | El número 10 es negativo |

**SOLUCIÓN DEL SEGUNDO PROBLEMA CON SU CORRESPONDIENTE ALGORITMO Y PRUEBA DE ESCRITORIO**

PROBLEMA: 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

SOLUCIÓN

1 Inicio

2 Ingresar un número real X

3 Ingresar otro número real Y

4 Si al ingresar el Segundo número real, resultan ser iguales, regresa al punto 3

5 Si el resultan ser diferentes entonces las condiciones son:

5.1 X,Y….X>Y

5.2 X,Y….Y>X

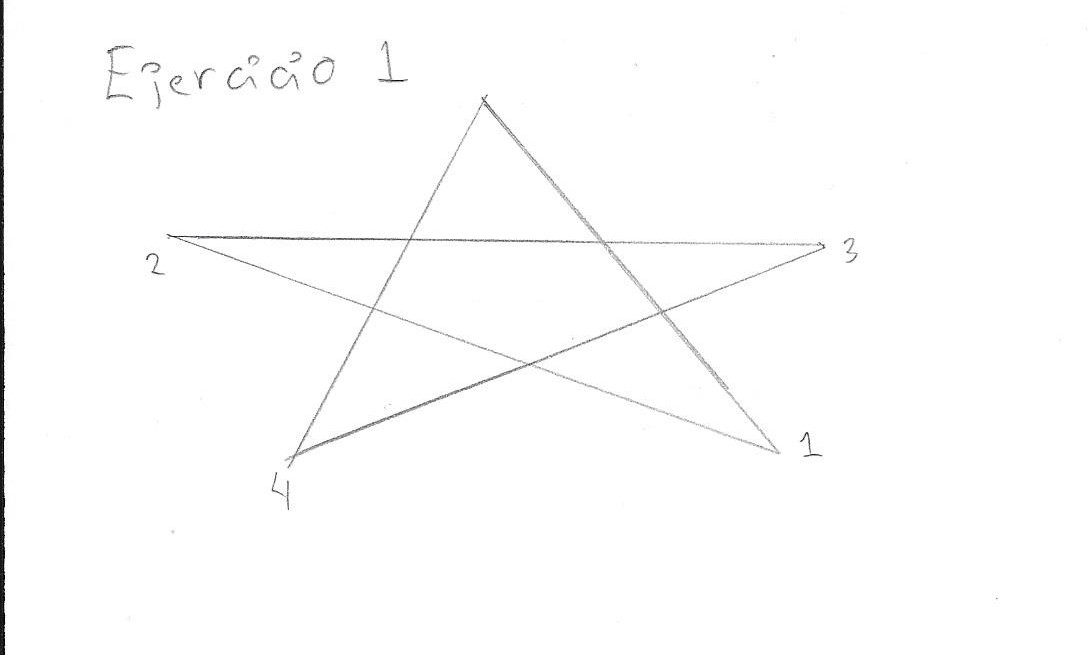
6 Fin

|  |  |  |  |
| --- | --- | --- | --- |
| **ITERACIÓN** | **X** | **Y** | **SALIDA** |
| 1 | 5 | 2 | 5 > 2 |
| 2 | 10 | 10 | --- |
| 3 | 8 | 15 | 8 < 15 |

**ALGORITMO FIGURA 1**

Resultado final del ejercicio de algoritmo número 1.

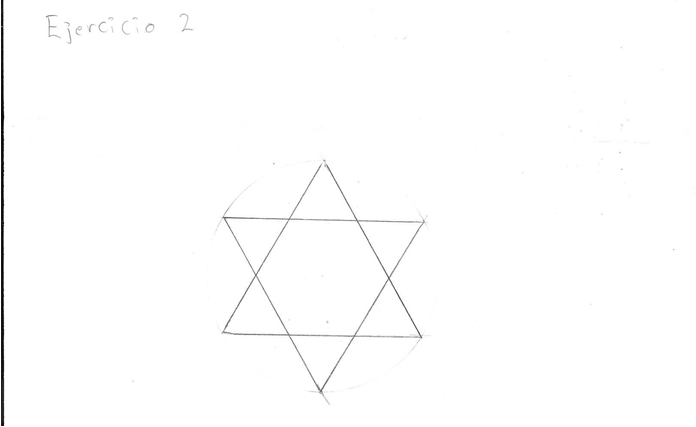
Se forma una estrella con 5 puntas.



**ALGORITMO FIGURA 2**

Resultado final del ejercicio 2 de algoritmo número 2.

Se forma una Estrella con 6 puntas.

****

**EJERCICIOS (TAREA)**

1.- Calcular el volumen de un cilindro a partir del radio de la base y la altura. (Hacer uso de la fórmula V = π (r2) (h)

Análisis.

Entrada: Radio, π, h

Salida: V

Diseño.

1 Inicio

2 Introducir R y h

3 Leer R y h

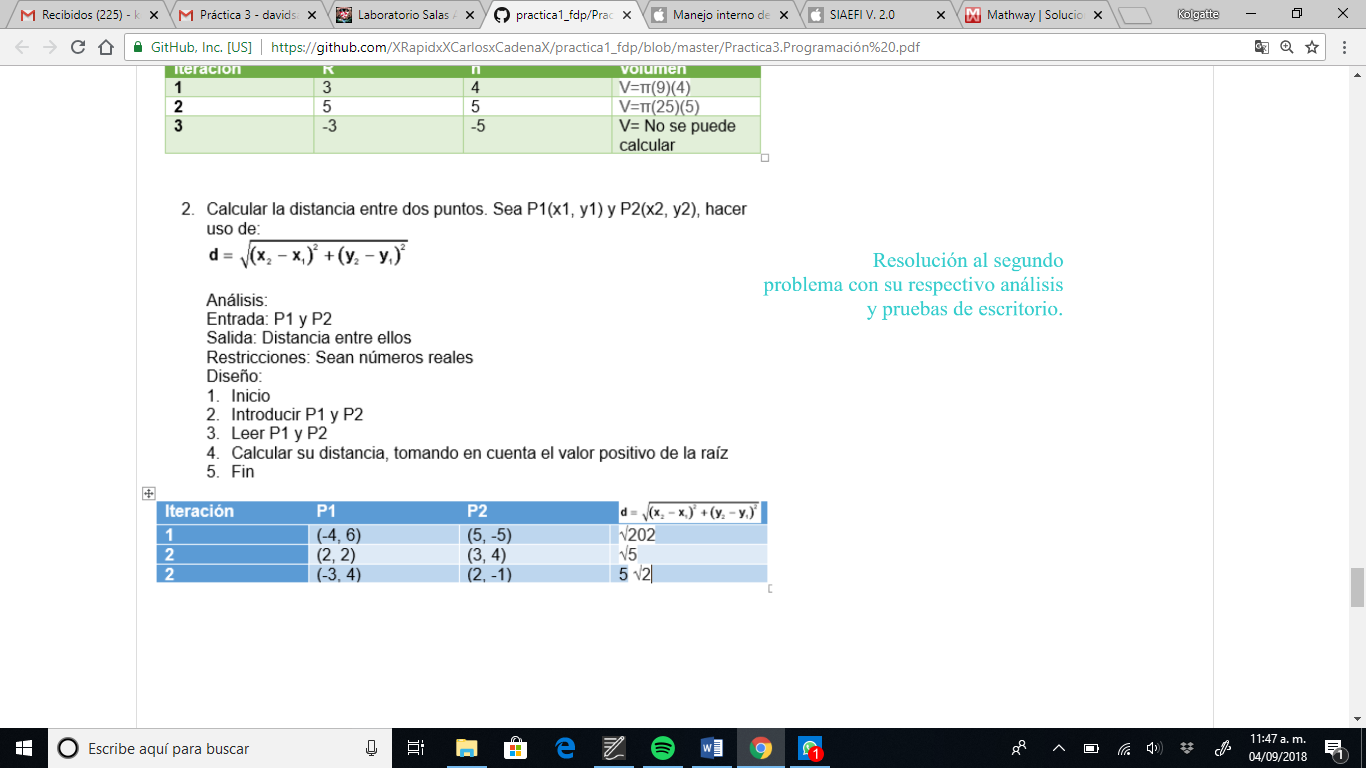
4 Si R>0 y h >0 entonces pasa al punto 6

5 Si R<o y h<0 no se puede calcular, introducir otros valores y regresar al punto 2 o ir al paso 6

6 Fin

|  |  |  |  |
| --- | --- | --- | --- |
| **ITERACIÓN** | **R** | **h** | **VOLUMEN** |
| 1 | 3 | 5 | V= π (9)(25) |
| 2 | 7 | 7 | V= π (49)(49) |
| 3 | -2 | -4 | V= No se puede |

2.- Calcular la distancia entre dos puntos. (Sea P1 (a1, b1) y P2 (a2, b2), hacer uso de:



Análisis.

Entrada: P1 y P2

Salida: Distancia entre ellos

Restricciones: Sean números reales

Diseño.

1 Inicio

2 Introducir P1 y P2

3 Leer P1 y P2

4 Calcular su distancia tomando en cuenta el valor positivo de la raíz

5 Fin

|  |  |  |  |
| --- | --- | --- | --- |
| **ITERACIÓN** | **P1** | **P2** |  |
| 1 | (-1, 5) | (-1, -6) | 11 |
| 2 | (0, 4) | (-1, 3) |  |
| 3 | (2, 2) | (3, 4) |  |

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.

Entrada: 2 números enteros

Salida: Verificar si son divisibles, no existe su resultado o es infinito

Restricciones: No hay, dado que en la salida pide que su resultado no exista, tal es el caso de denominador 0

Diseño:

1 Inicio

2 Escribir 2 números enteros

3 Leer los números dados

4 Hacer división de estos números.

5 Imprimir el resultado y en cada caso realizar lo siguiente:

5.1 Divisible: mostrar el resultado

5.2 Cuando el divisor sea 0 mostrar no existe

6 Fin

|  |  |  |  |
| --- | --- | --- | --- |
| **ITERACIÓN** | **PRIMER NUMERO** | **SEGUNDO NUMERO** | **RESULTADO** |
| 1 | 4 | 2 | “Divisible” 2 |
| 2 | 15 | 0 | “No existe” |
| 3 | 10 | 5 | “Divisible” 2 |

4.- Leer un número y verificar si un número es par o impar.

Análisis.

Entrada: Un número

Salida: Decir si es par o impar

Restricciones: Que el número pertenezca a los enteros

Diseño.

1 Inicio

2 Dar un número entero

3 Leer el número dado

4 Dividir el número entre 2 y en cada caso hacer lo siguiente:

4.1 Si al dividir entre 2 su residuo es 0 entonces es “par”

4.2 Si al dividir entre 2 su residuo es diferente a 0 entonces es “impar”

5 Fin

|  |  |  |  |
| --- | --- | --- | --- |
| **ITERACIÓN** | **NÚMERO** | **DIVIDIR ENTRE 2** | **SALIDA** |
| 1 | 4 | 4/2=2 residuo 0 | “Par” |
| 2 | 17 | 17/2= 8.5 residuo !=0 | “Impar” |
| 3 | 10 | 10/2 = 5 residuo 0 | “Par” |

**CONCLUSIONES**

* Se puso en práctica el desarrollo de los algoritmos y tener más claro la construcción de uno y su desarrollo.
* Los diferentes tipos de una construcción o indicaciones de un algoritmo
* Aprendí lo de los números factoriales, números que desconocía, una enseñanza nueva simple.
* El algoritmo número 5 se me complico para desarrollarlo no le entendi por completo y por ende no esta mostrado en la practica.

**REFERENCIAS**

<https://www.google.com/search?source=hp&ei=kh>

<https://www.smartick.es/blog/matematicas/recursos-didacticos/factoriales/>

<https://www.google.com/search?q=numeros+factoriales&hl>=

<https://www.ecured.cu/Archivo:Cicloevida1.jpg>

<https://www.google.com/search?biw=1366&bih=662&>

<http://lcp02.fi-b.unam.mx/>