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

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | GARCIA MORALES KARINA |
| *Asignatura:* | FUNDAMENTOS DE PROGRAMACIÓN |
| *Grupo:* | 1121 |
| *No de Práctica(s):* | PRÁCTICA 3 |
| *Integrante(s):* | CADENA MARTÍNEZ CARLOS DAVID |
|  |  |
| *No. de Equipo de cómputo empleado* |  |
| *Semestre:* | 2019-1 |
| *Fecha de entrega:* | 5/Sep/2018 |
| *Obervaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_

PRACTICA 3: SOLUCIÓN DE PROBLEMAS Y ALGORITMOS

**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.

**Actividades**:

A partir del enunciado de un problema, identificar el conjunto de entrada y el conjunto de salida.

Elaborar un algoritmo que resuelva un problema determinado (dado por el profesor), identificando los módulos de entrada, de procesamiento y de salida.

**Introducción**

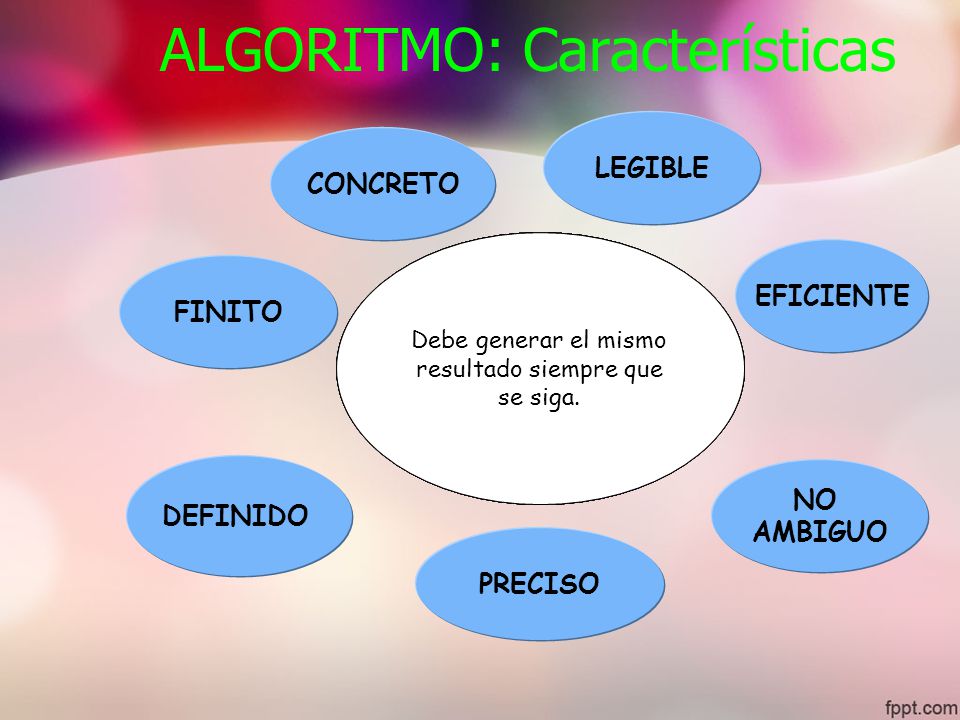
Un problema informático se puede definir como el conjunto de instancias al cual corresponde un conjunto de soluciones, junto con una relación que asocia para cada instancia del problema un subconjunto de soluciones (posiblemente vacío). Para poder solucionar un problema nos apoyamos en la Ingeniería 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:**



**CARACTERISTICAS DE LOS ALGORITMOS**  
Las características fundamentales que debe cumplir todo algoritmo son:

* Un algoritmo debe ser preciso e indicar el orden de realización de cada paso.
* Un algoritmo debe estar definido. Si se sigue un algoritmo dos veces, se debe obtener el mismo resultado cada vez.
* Un algoritmo debe ser finito. el algoritmo se debe terminar en algún momento; o sea, debe tener un número finito de pasos.
* Un algoritmo debe ser legibles: El texto que lo describe debe ser claro, tal que permita entenderlo y leerlo fácilmente.

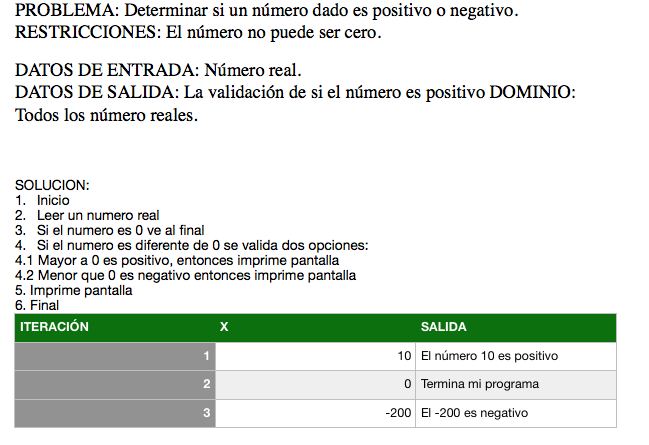


**TEORIA DE LA COMPUTABILIDAD:**

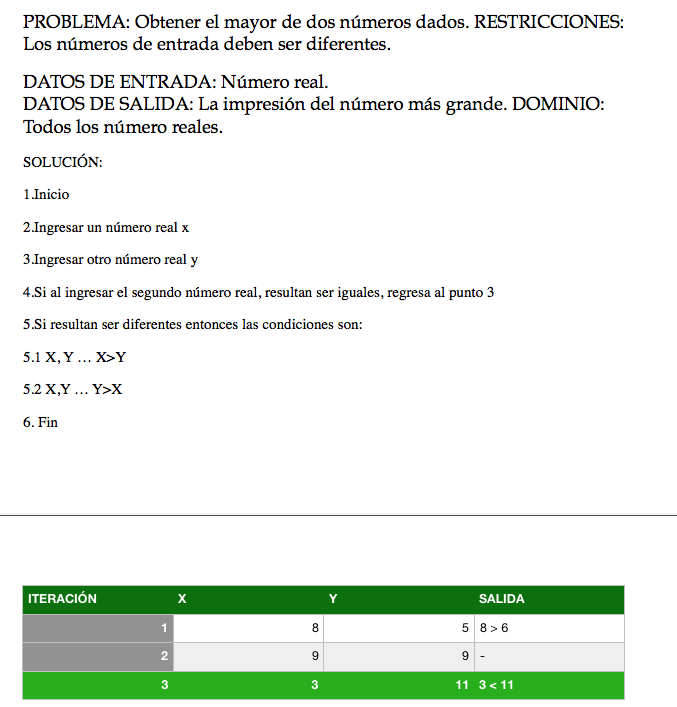
La Teoría de la computabilidad es la parte de la [computación](https://es.wikipedia.org/wiki/Teor%C3%ADa_de_la_computaci%C3%B3n) que estudia los [problemas de decisión](https://es.wikipedia.org/wiki/Problema_de_decisi%C3%B3n) que pueden ser resueltos con un [algoritmo](https://es.wikipedia.org/wiki/Algoritmo) o equivalentemente con la llamada [máquina de Turing](https://es.wikipedia.org/wiki/M%C3%A1quina_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 comprensión más clara de esta idea intuitiva; y solo de esta forma podemos explorar matemáticamente el concepto de computabilidad y los conceptos relacionadas con ella, tales como decibilidad, etc... Surge así una teoría que producirá resultados positivos y negativos (estamos pensando en resultados de no computabilidad o de indecibilidad).

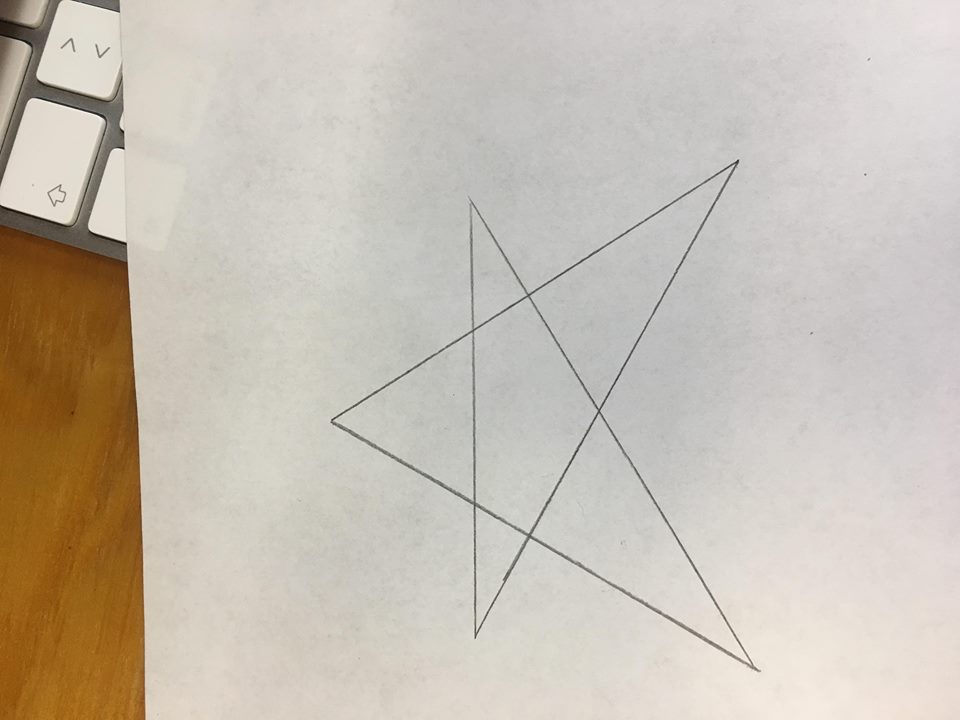


**DESARROLLO DE LA PRÁCTICA**

Solución del primer problema  
con su correspondiente algoritmo y  
prueba de escritorio.

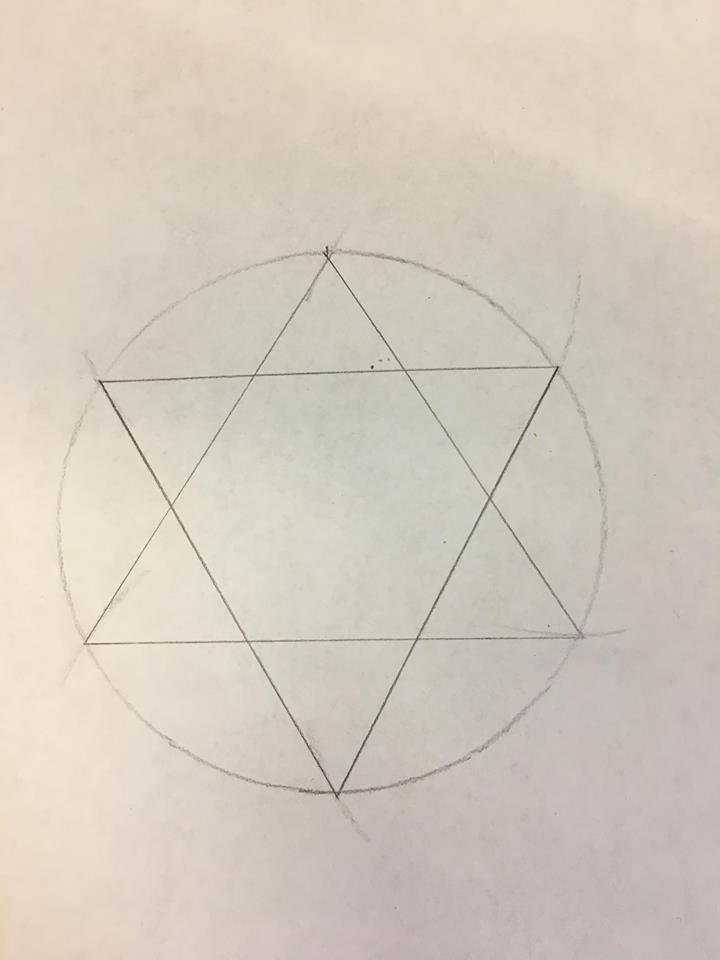


Solución al segundo problema con su correspondiente  
algoritmo y prueba de escritorio.

 Resultado final, tras seguir el algoritmo

del ejercicio número 1.

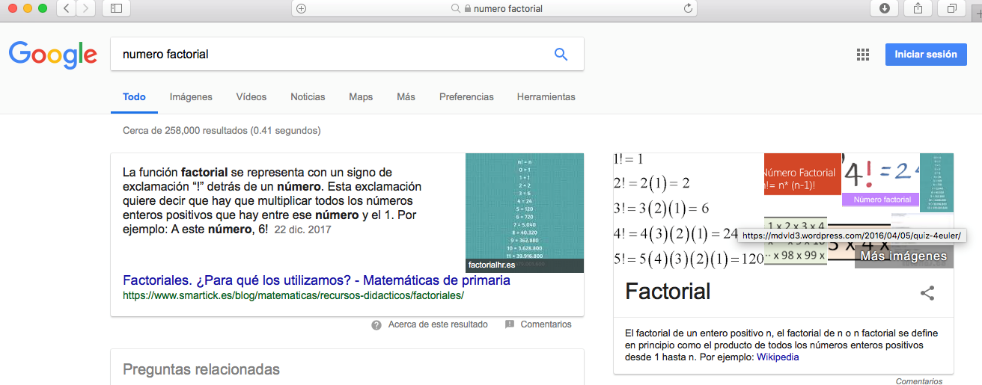
Se formó una estrella con 5 puntas.



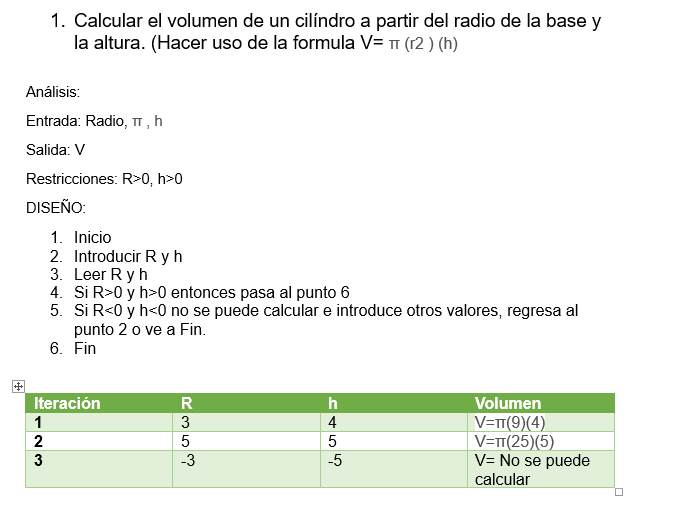
Resultado del ejercicio 2 resuelto en clase

con base al algoritmo, se formó una estrella

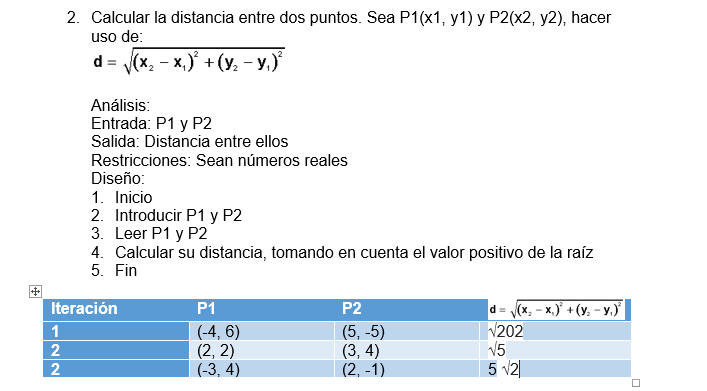
con 6 puntas.



Búsqueda de lo qué es un número factorial.

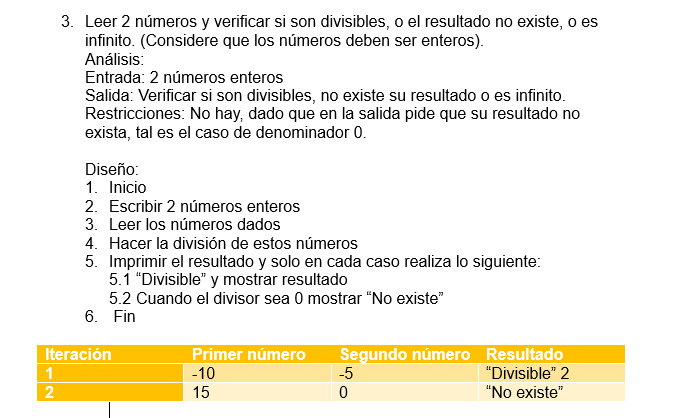
**EJERCICIOS DE TAREA:**

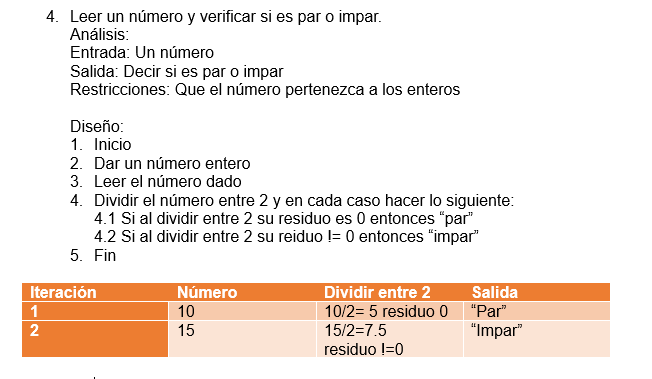
Realización del algoritmo para este primer problema.



Resolución al segundo problema con su respectivo análisis

y pruebas de escritorio.

 Resolución del tercer problema con su respectivo algoritmo y prueba de escritorio.



Resolución del cuarto problema con su algoritmo

su prueba de escritorio.

**CONCLUSIONES:**

* Realmente me resultó una práctica un poco compleja dado que al principio me costó organizar los algoritmos para cada problema.
* También no pude resolver el último ejercicio, por más que traté en encontrar una forma para poder resolverlo.
* Por último puedo decir que con esta práctica aprendí y reforcé la construcción de un algoritmo.

**BIBLIOGRAFÍA:**

* <http://decsai.ugr.es/~castro/MCII/node2.html>
* <https://es.wikipedia.org/wiki/Teor%C3%ADa_de_la_computabilidad>
* <https://aulatec.wordpress.com/caracteristicas-y-propiedades-de-los-algoritmos/>