**UNVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

**FACULTAD DE INGENIERÍA**

**LABORATORIO DE DOCENCIA**

**LABORATORIOS DE COMPUTACIÓN SALAS A Y B**

**“ALGORITMOS”**

***PROFESORA****:* ING. KARINA GARCÍA MORALES

***ASIGNATURA****:* FUNDAMENTOS DE PROGRAMACIÓN

***GRUPO****:* 1121

***PRÁCTICA****:* 3

**ALUMNO**: CABRERA LUNA SEBASTIÁN

**SEMESTRE: 2018**

**FECHA DE ENTREGA: 4-septiembre-2018**

**CALIFICACIÓN: \_\_\_\_\_\_\_**

**Objetivo:**

A partir del análisis del problema (conjuntos de entrada y salida), elaborar algoritmos que permitan resolver el problema planteado.

**Desarrollo:**

En la práctica analizaremos problemas y elaboraremos algoritmos para resolver los problemas planteados.

**Conceptos:**

**Algoritmo**:   
Un algoritmo es 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 características de un algoritmo son las siguientes:

-Un algoritmo debe ser preciso, es decir, llegar a la solución en el menor tiempo

posible y sin ambigüedades.

-También debe ser determinista, es decir, a partir de un conjunto de datos idénticos de entrada, debe arrojar siempre los mismos resultados a la salida.

-El que un proceso sea computable implica que, en algún momento, el proceso va a llegar a su fin. Un algoritmo debe ser finito, por tanto, en algún momento debe terminar, lo que al mismo tiempo implica que el dominio del problema debe estar acotado.

**Ciclo de vida del software:**



El algoritmo se encuentra en la parte de diseño del ciclo de vida del software

Un algoritmo consta de 3 módulos básicos: módulo de **entrada**, módulo de **procesamiento** y módulo de **salida**.



**Módulo de entrada:**

Representa los datos que requieren para resolver el problema. Estos datos se pueden solicitar al usuario, leer de un archivo, consultar de una base de datos, etc.

**Módulo de procesamiento de datos:**

Representa las operaciones necesarias para obtener

un resultado a partir de los datos de entrada.

**Módulo de salida:**

Permite mostrar los resultados obtenidos a partir del módulo de procesamiento de datos. Los resultados pueden mostrarse en diversos sitios: en la pantalla, en un archivo, en una base de datos, etc.

**Prueba de escritorio:**

El diseño de la solución de un problema implica la creación del algoritmo y la validación del mismo. La validación se suele realizar mediante una prueba de escritorio.

Una prueba de escritorio es una matriz formada por los valores que van adquiriendo cada una de las variables del programa en cada iteración. Una iteración es el número de veces que se ejecuta un código y permite ver los valores que van adquiriendo las variables en cada repetición.

**Ejercicios en clase:**

**Ejercicio 1**

PROBLEMA: Seguir el algoritmo para obtener una figura

ENTRADA: Hoja tamaño carta en limpio, regla y lápiz.

SALIDA: Figura correcta.

***Algoritmo***

1. Dibuja una V invertida. Empieza desde el lado izquierdo, sube, y baja hacia el lado derecho, no levantes el lápiz.

2. Ahora dibuja una línea en ángulo ascendente hacia la izquierda. Debe cruzar la primera línea más o menos a 1/3 de la altura. Todavía no levantes el lápiz del

papel.

3. Ahora, dibuja una línea horizontal hacia la derecha. Debe cruzar la V invertida más o menos a 2/3 de la altura total. Sigue sin levantar el lápiz.

4. Dibuja una línea en un ángulo descendente hasta el punto de inicio. Las líneas deben unirse.

5. Ahora ya puedes levantar el lápiz del papel. Has terminado la estrella de 5puntas.

**Ejercicio 2**

PROBLEMA: Seguir el algoritmo para obtener una figura

ENTRADA: Hoja tamaño carta en limpio, regla y lápiz.

SALIDA: Figura correcta.

***Algoritmo***

1. Empieza dibujando un círculo con un compás. Coloca un lápiz en el compás.

Coloca la punta del compás en el centro de una hoja de papel.

2. Ahora gira el compás, mientras mantienes la punta apoyada en el papel. El lápiz dibujará un círculo perfecto alrededor de la punta del compás.

3. Marca un punto en la parte superior del círculo con el lápiz. Ahora, coloca la punta del compás en la marca. No cambies el radio del compás con que hiciste el círculo.

4. Gira el compás para hacer una marca en el propio círculo hacia la izquierda. Haz una marca también en el lado derecho.

5. Ahora, coloca la punta del compás en uno de los puntos. Recuerda no cambiar el radio del compás. Haz otra marca en el círculo.

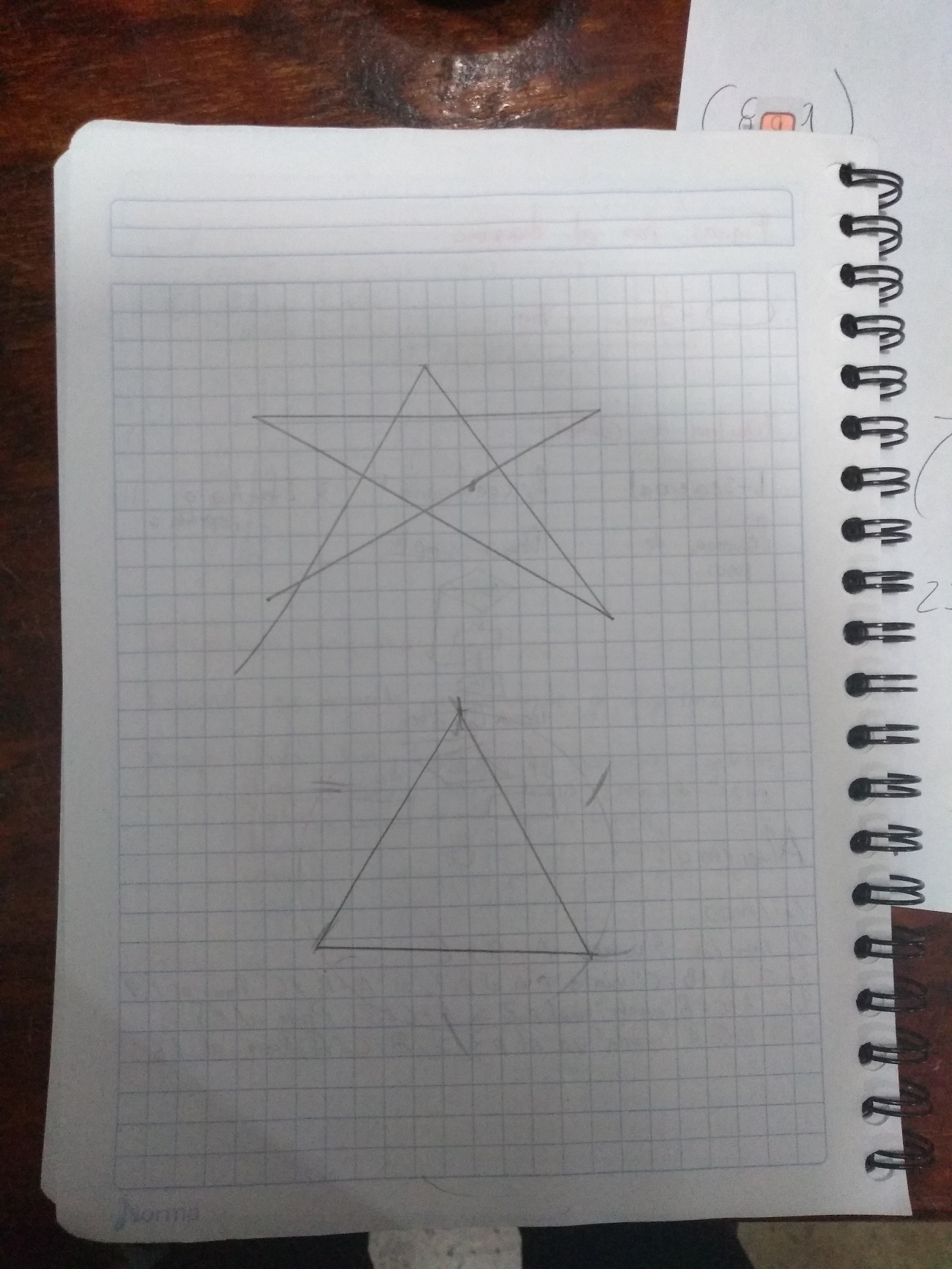
6. Continúa moviendo la punta del compás a las otras marcas, y continúa hasta que tengas 6 marcas a la misma distancia unas de otras. Ahora, yapuedes dejar tu compás a un lado.

7. Usa una regla para crear un triángulo que empiece en la marca superior del círculo. Coloca el lápiz en la marca superior. Ahora dibuja una línea hasta la segunda marca por la izquierda. Dibuja otra línea, ahora hacia la derecha, saltándote la marca de la parte más baja. Complementa el triángulo con una línea hacia la marca superior. Así completarás el triángulo.

3. Ahora, dibuja una línea horizontal hacia la derecha. Debe cruzar la V invertida más o menos a 2/3 de la altura total. Sigue sin levantar el lápiz.

4. Dibuja una línea en un ángulo descendente hasta el punto de inicio. Las líneas deben unirse.

5. Ahora ya puedes levantar el lápiz del papel. Has terminado la estrella de 5 puntas.



***Ejemplo 1***

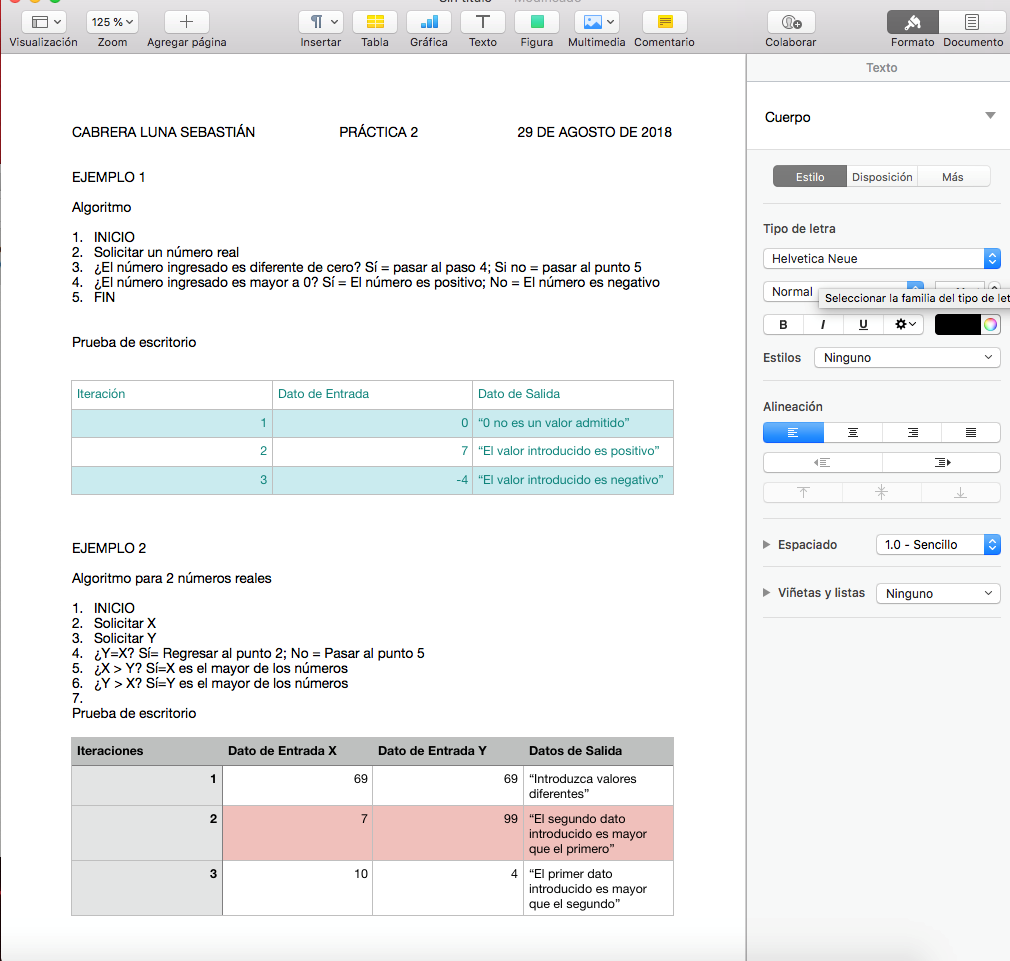
**PROBLEMA**: Determinar si un número dado es positivo o negativo.

**RESTRICCIONES**: El número no puede ser cero.

**DATOS DE ENTRADA**: Número real.

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

**DOMINIO**: Todos los números reales.

****

***Ejemplo 2***

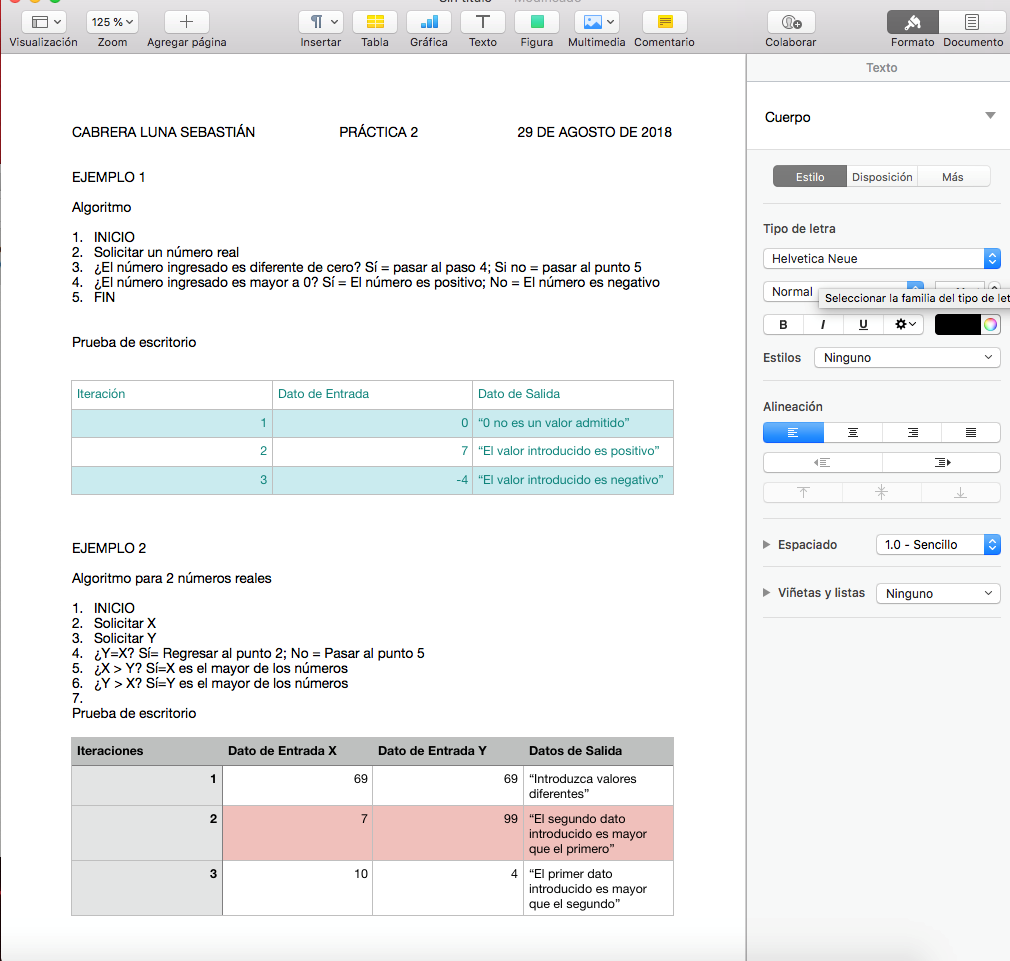
**PROBLEMA**: Obtener el mayor de dos números dados.

**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.

****

***Ejemplo 3***

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

**RESTRICCIONES**: El número de entrada debe ser entero y no puede ser negativo.

**DATOS DE ENTRADA**: Número entero.

**DATOS DE SALIDA**: La impresión del factorial del número.

**DOMINIO**: Todos los números naturales positivos.

Algoritmo para obtener factorial

1. INICIO
2. Ingresar x
3. Dividir x entre el número mismo
4. Seguir la serie n+1 empezando por el 1 y acabar hasta el número x
5. Multiplicar n+1 hasta x
6. Imprimir resultado
7. FIN

|  |  |  |
| --- | --- | --- |
| **ITERACION** | **DATO DE ENTRADA** | **DATO DE SALIDA** |
| **1** | **5** | **120** |
| **2** | **3** | **6** |
| **3** | **6** | **720** |

**Ejercicios de tarea**

1. ***Calcular el volumen de un cilindro a partir del radio de la base y la altura. (Hacer uso de la fórmula V r h2 = π).***

**Análisis**

**Entrada:** Radio de la base (R) y la altura (A), π=3.1416

**Salida:** Volumen

**Restricciones:** Ninguna

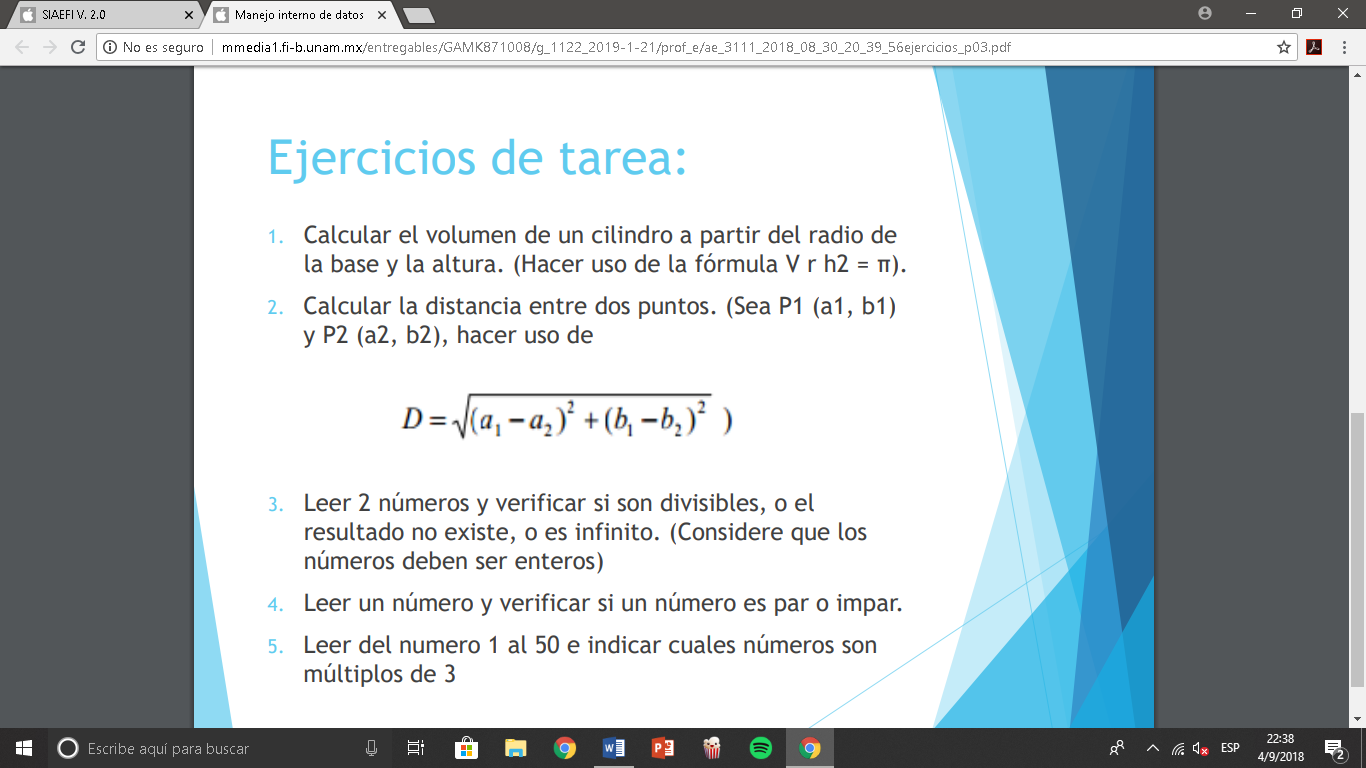
**Metodología:** V r h2 = π

**Construcción, diseño**

1. Inicio
2. Escribir R y A
3. Leer R y A
4. Aplicar fórmula Vrh2=π
5. Imprimir resultado
6. FIN

|  |  |  |  |
| --- | --- | --- | --- |
| ITERACIÓN | ENTRADA | SALIDA | IMPRIMIR |
| 1 | R= 5; A=8 | 628.3185 | “El volumen es 628.3185” |
| 2 | R=20; A=10 | 12,566.3706 | “El volumen es 12,566.3706” |
| 3 | R=7; A=4 | 615.7521 | “El volumen es 615.7521” |

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

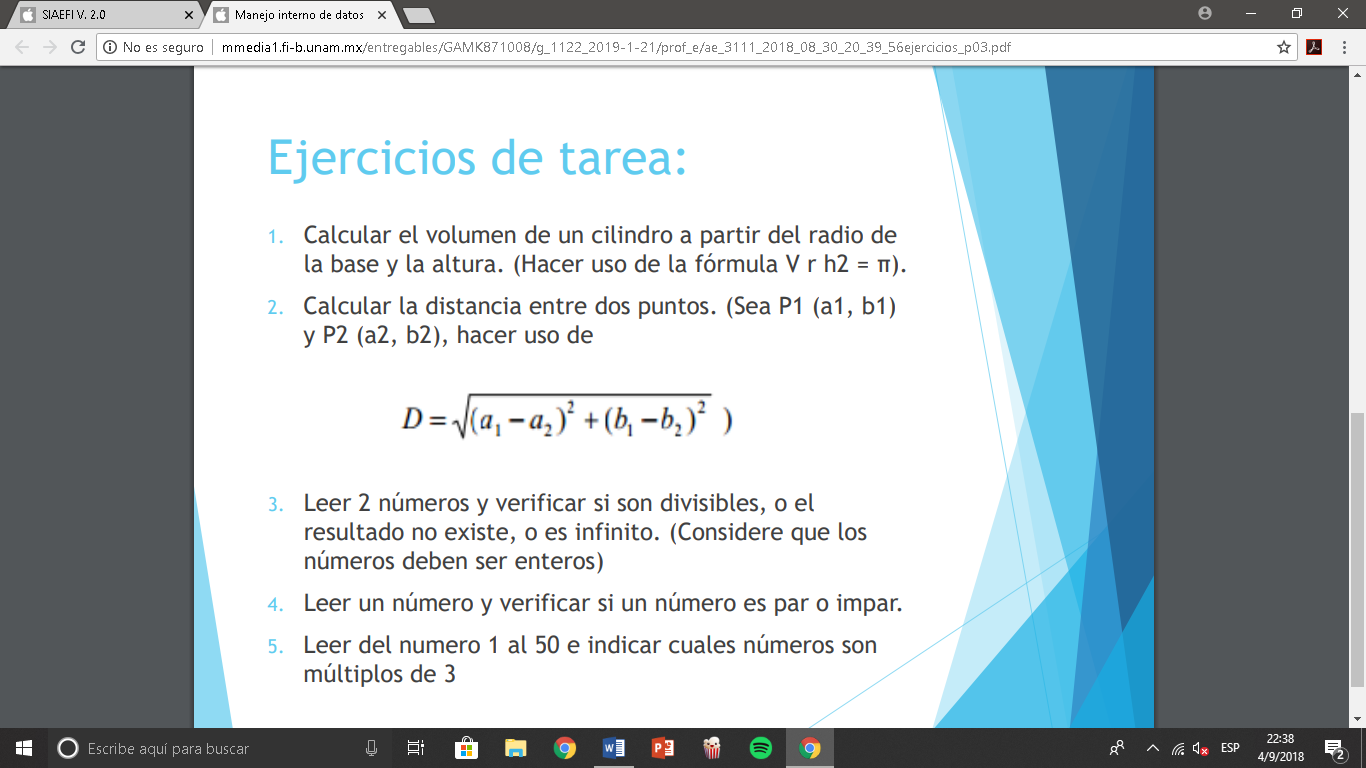


**Análisis**

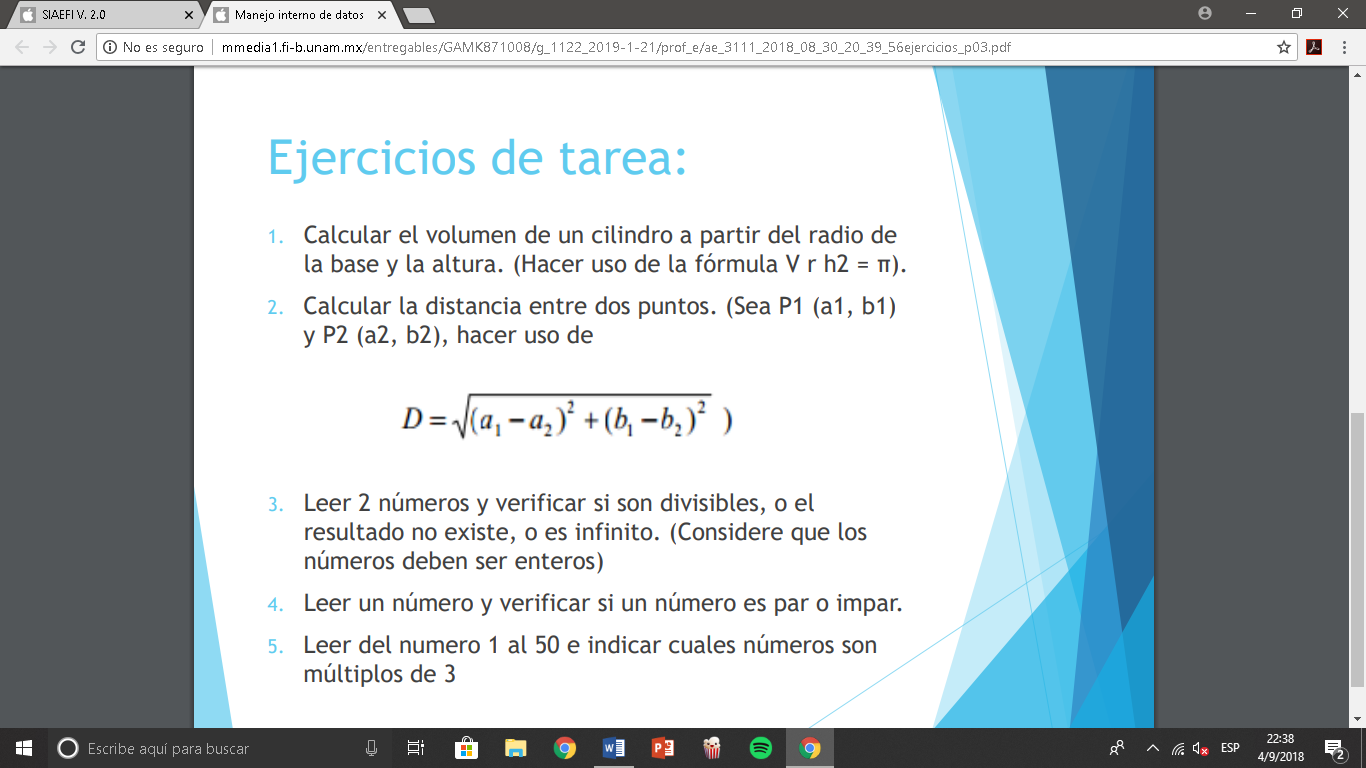
**Entrada:** A1, A2, B1 y B2

**Salida:** Distancia

**Restricciones:** Ninguna

**Metodología:** 

**Construcción y diseño**

1. INICIO
2. Leer A1, A2, B1 y B2
3. Aplicar 
4. Imprimir resultado “La distancia es x”
5. FIN

|  |  |  |  |
| --- | --- | --- | --- |
| ITERACIÓN | ENTRADA | SALIDA | IMPRIMIR |
| 1 | 8,9 y 2,3 | 8.48 | “La distancia es 8.48” |
| 2 | 2,9 y 7,3 | 7.81 | “La distancia es 7.81” |
| 3 | 4,3 y 9,7 | 6.40 | “La distancia es 6.40” |

1. ***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**: X, Y

**Salida**: Saber si son divisibles

**Restricciones**: X, Y >=0

**Metodología**: X/Y

**Construcción, diseño**

1. INICIO
2. Leer X, Y
3. Aplicar X/Y
4. Si Y>=0 imprimir resultado, Su Y=0 Imprimir “El resultado es infinito”
5. FIN

|  |  |  |  |
| --- | --- | --- | --- |
| ITERACIÓN | ENTRADA | SALIDA | IMPRIMIR |
| 1 | 100, 50 | 2 | “El resultado es 2” |
| 2 | 9,0 | ∞ | “El resultado es ∞” |
| 3 | 3,1 | 3 | “El resultado es 3” |

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

**Análisis**

**Entrada:** X

**Salida:** Saber si es par o impar

**Restricciones:** X>=0

**Metodología:** X/2

**Construcción, Diseño**

1. INICIO
2. Leer X
3. Aplicar X/2, SI X=Z Imprimir “El número es par”, Si X=Q Imprimir “El número es impar”
4. FIN

|  |  |  |  |
| --- | --- | --- | --- |
| ITERACIÓN | ENTRADA | SALIDA | IMPRIMIR |
| 1 | 46 | 23 | “El número es par” |
| 2 | 97 | 32.3 | “El número es impar” |
| 3 | 200 | 100 | “El número es par” |

1. ***Leer del número 1 al 50 e indicar cuales números son múltiplos de 3***

**Análisis**

**Entrada:** Números del 1 al 50

**Salida:** Saber si son múltiplos de 3

**Restricciones:** Ninguna

**Metodología:** N/3

**Construcción, Diseño**

1. INICIO
2. Leer números del 1 al 50
3. Aplicar N/3 para cada número
4. SI N=Z Imprimir “El número es múltiplo de 3”, Si X=Q Imprimir “El número no es múltiplo de 3”
5. FIN

|  |  |  |  |
| --- | --- | --- | --- |
| ITERACIÓN | ENTRADA | SALIDA | IMPRIMIR |
| 1 | 9 | 3 | “El número es múltiplo de 3” |
| 2 | 45 | 15 | “El número es múltiplo de 3” |
| 3 | 10 | 3.33333 | “El número no es múltiplo de 3” |

**Referencias**

* Raghu Singh (1995). International Standard ISO/IEC 12207 Software Life Cycle

Processes. Agosto 23 de 1996, de ISO/IEC. Consulta: Junio de 2015. Disponible en:

<http://www.abelia.com/docs/12207cpt.pdf>

- Carlos Guadalupe (2013). Aseguramiento de la calidad del software (SQA). [Figura

1].Consulta: Junio de 2015. Disponible en:

https://www.mindmeister.com/es/273953719/aseguramiento-de-la-calidad

delsoftware-sqa

* Andrea S. (2014). Ingeniería de Software. [Figura 2]. Consulta: Junio de 2015.

Disponible en: <http://ing-software-verano2014.blogspot.mx>

* Michael Littman. (2012). Intro to Algorithms: Social Network Analysis. Consulta: Junio

de 2015, de Udacity. Disponible en: https://www.udacity.com/course/viewer#!/ccs215/

l-48747095/m-48691609