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

Laboratorios de computación

|  |  |
| --- | --- |
| *Profesor:* | Ing. Claudia Rodriguez Espino. |
| *Asignatura:* | Fundamentos de programación |
| *Grupo:* | 03 |
| *No de Práctica(s):* | 03 |
| *Integrante(s):* | Castañeda Garcés Santiago |
| *Semestre:* | 2018-2 |
| *Fecha de entrega:* | 02-02-2018 |
| *Observaciones:* |  |

salas A y B

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

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.

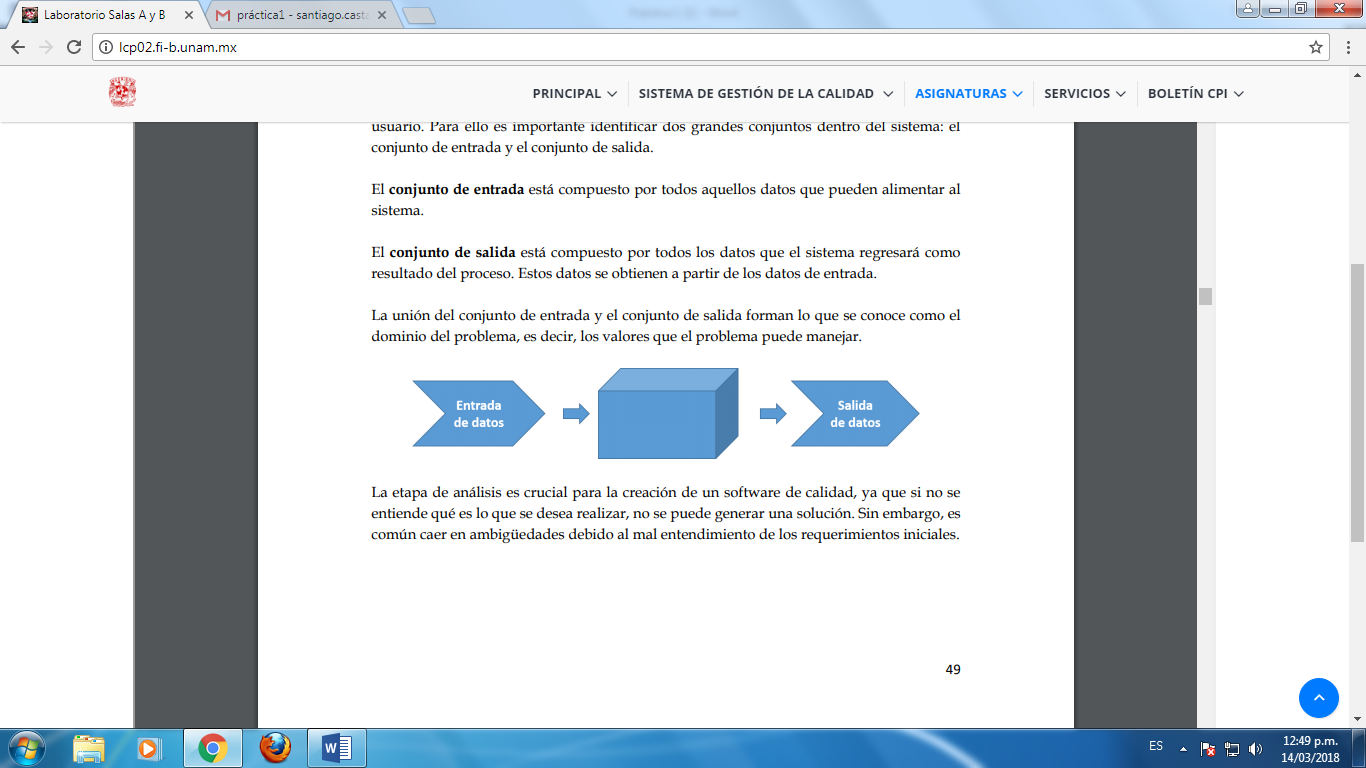
• Pruebas y mantenimiento (validación y verificación).

**Solución de problemas**

El análisis es el proceso para averiguar qué es lo que requiere el usuario del sistema de software (análisis de requisitos). Esta etapa permite definir las necesidades de forma clara y concisa (especificación de requisitos).

El conjunto de entrada está compuesto por todos aquellos datos que pueden alimentar al sistema.

El conjunto de salida está compuesto por todos los datos que el sistema regresará como resultado del proceso. Estos datos se obtienen a partir de los datos de entrada.



**Ejemplo**

\*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 número reales.

**SOLUCIÓN:**

1. Solicitar un número real.

2. Si el número ingresado es cero, se regresa al punto 1.

3. Si el número ingresado es diferente de cero, se validan las siguientes condiciones:

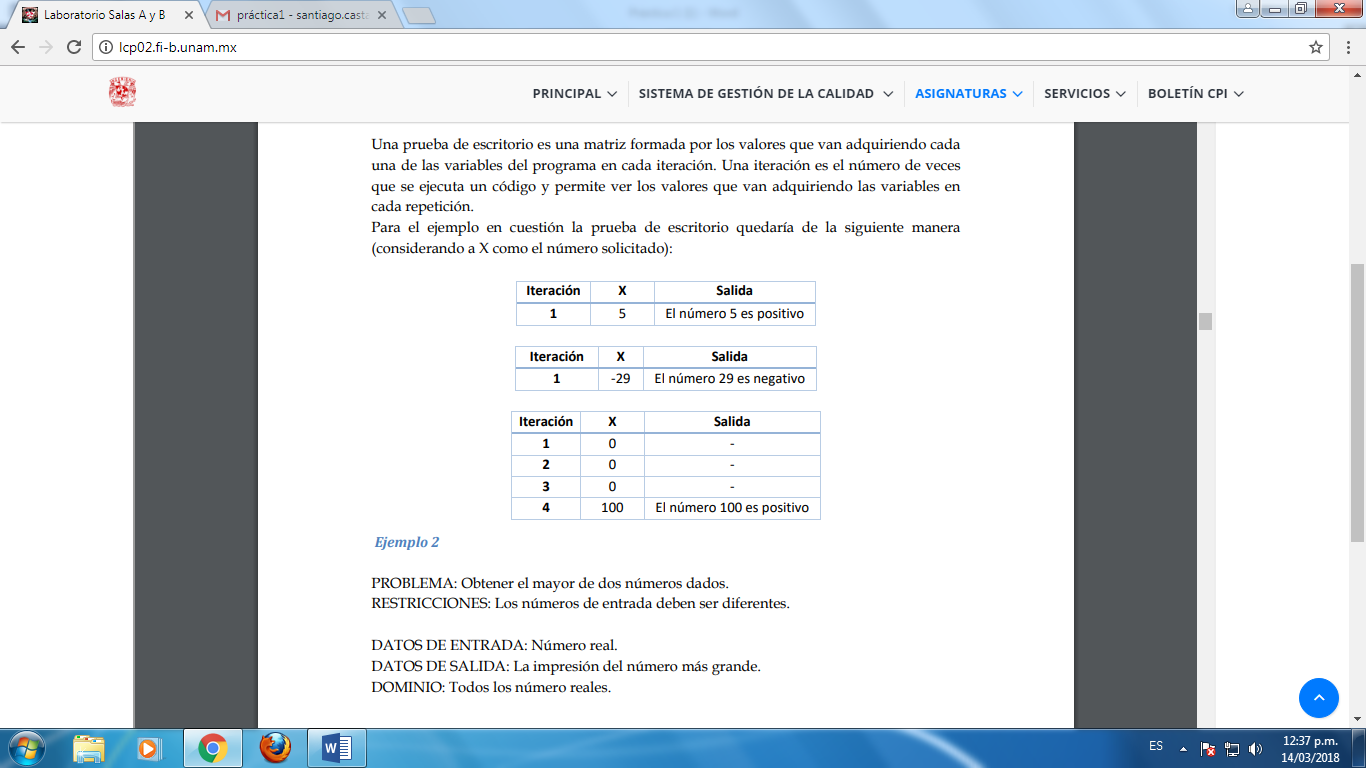
3.1 Si el número ingresado es mayor a 0 se puede afirmar que el número es positivo.

3.2 Si el número ingresado es menor a 0 se puede afirmar que el número es negativo.

**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.

Para el ejemplo en cuestión la prueba de escritorio quedaría de la siguiente manera (considerando a X como el número solicitado):

Desarrollo

**Ejercicio 1**

\*PROBLEMA: Determinar el resultado de: y=x2- 4x+20.

\*RESTRICCIONES: x tiene que ser mayos a dos.

\*DATOS DE ENTRADA: Número real.

\*DATOS DE SALIDA: El resultado de y según, y= x2- 4x+20.

\*DOMINIO: Todos los números reales.

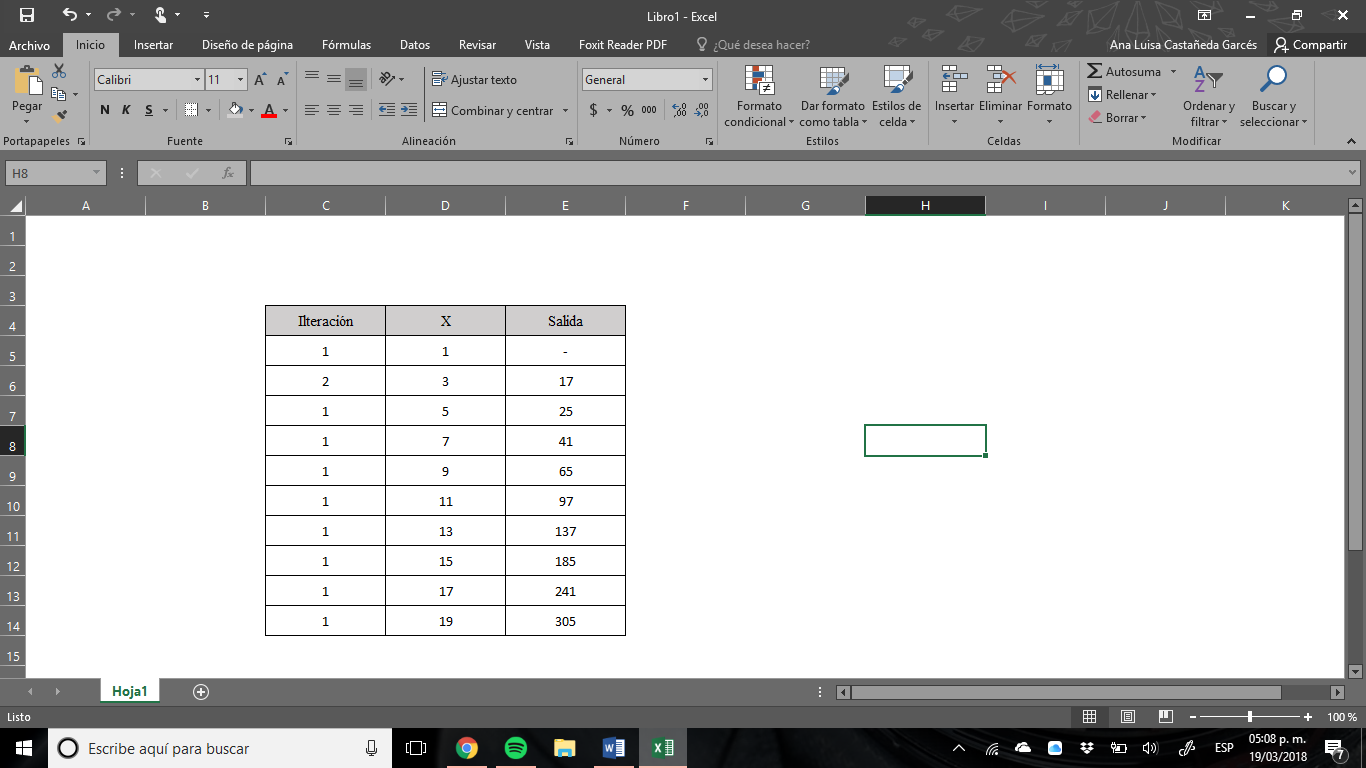
**SOLUCIÓN:**

1. Solicitar un número real x mayor a 2.

2. Si el número ingresado x es menor o igual a dos, se regresa al punto 1.

3. Si el número ingresado x es mayor a dos, se realiza la siguiente operación: x2 -4x+20.

4.- Se muestra el resultado de la operación.

 Prueba de escritorio (X es el número entero del que se calculará x2 -4x+20):

**Ejercicio 2**

\*PROBLEMA: Determinar el resultado de: y=3x28x+2.

\*RESTRICCIONES: x tiene que ser menor a dos.

\*DATOS DE ENTRADA: Número real.

\*DATOS DE SALIDA: El resultado de y, según y=3x28x+2.

\*DOMINIO: Todos los números reales.

**SOLUCIÓN:**

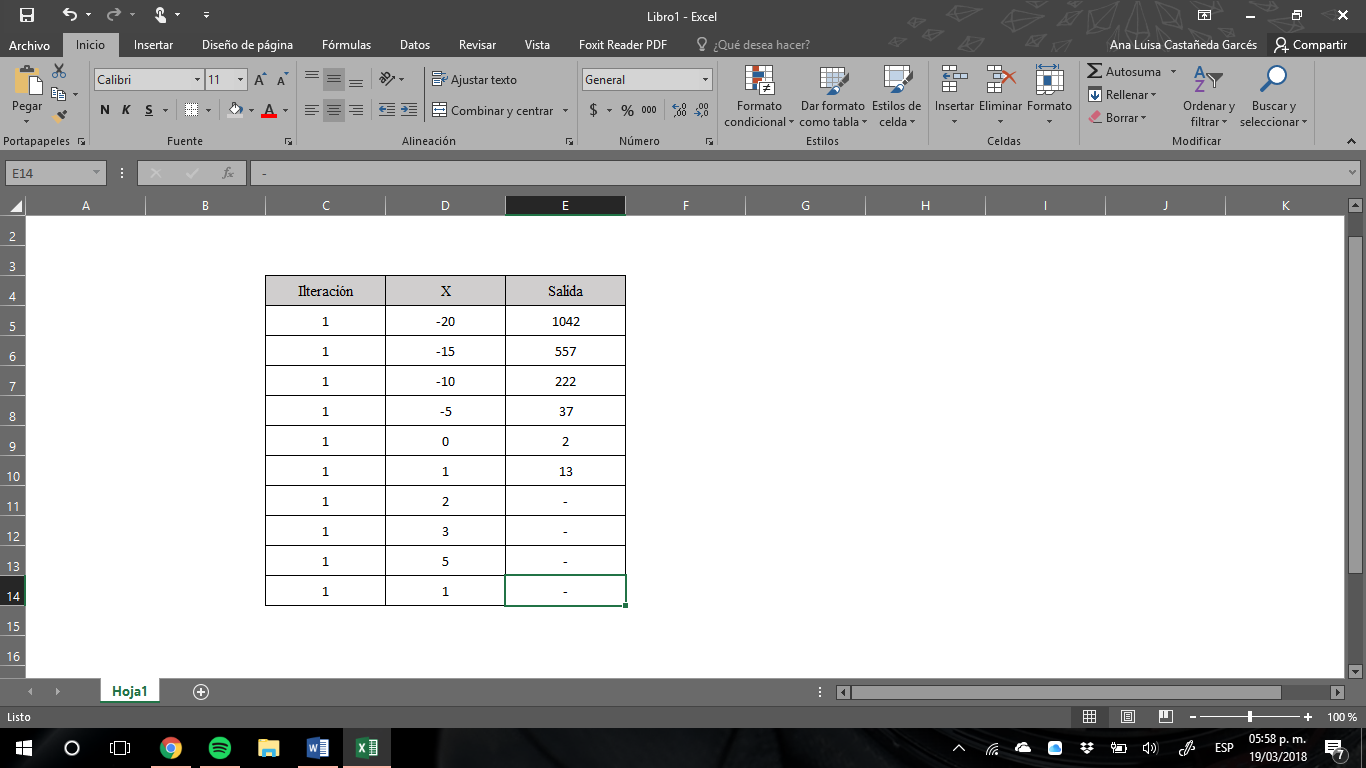
1. Solicitar un número real x menor a 2.

2. Si el número ingresado x es mayor o igual a dos, se regresa al punto 1.

3. Si el número ingresado x es menor a dos, se realiza la siguiente operación: 3x28x+2.

4.- Se muestra el resultado de la operación.

Prueba de escritorio (X es el número entero del que se calculará 3x28x+2):



**Ejercicio 3**

\*PROBLEMA: Determinar el área de un circulo.

\*RESTRICCIONES: r sea un número entero.

\*DATOS DE ENTRADA: Número real.

\*DATOS DE SALIDA: El resultado de r, según (3.1416) \*r2.

\*DOMINIO: Todos los números reales.

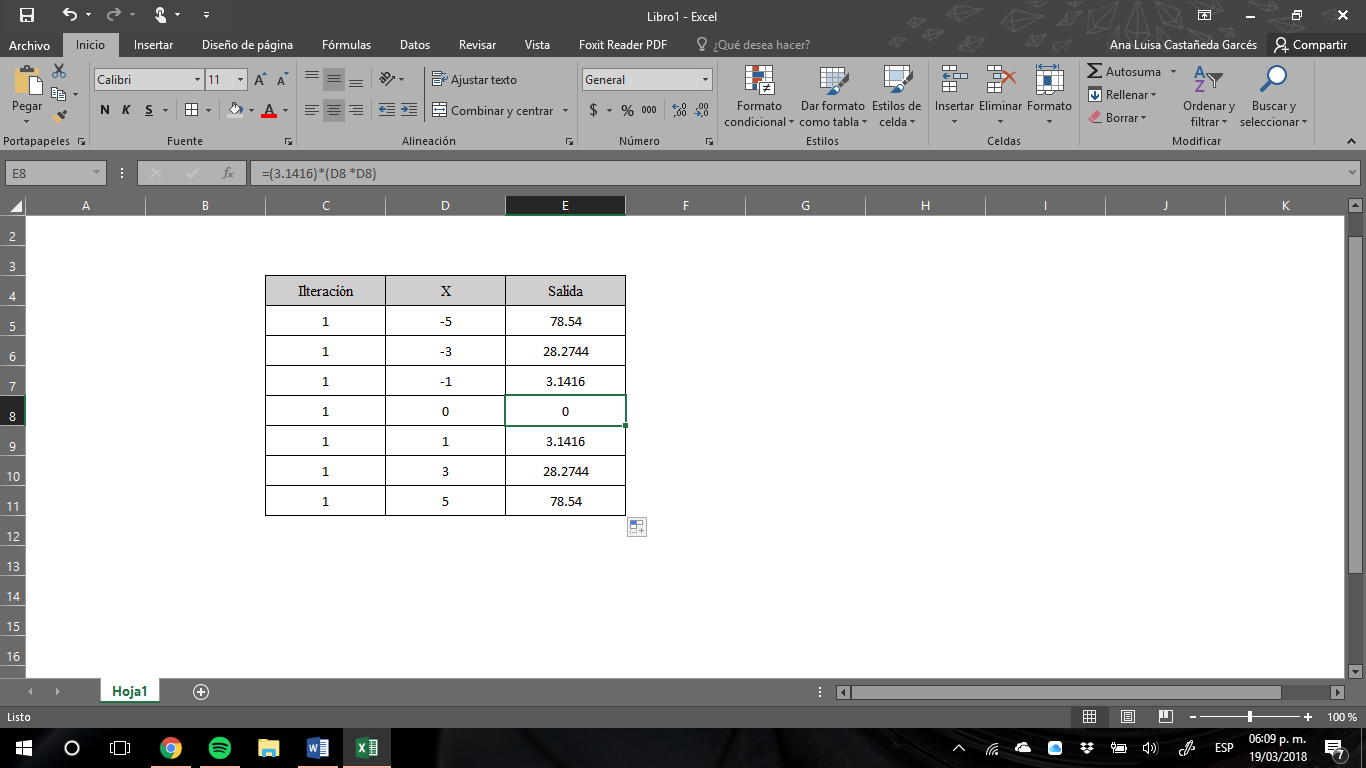
**SOLUCIÓN:**

1. Solicitar un número real r.

2. Realiza la siguiente operación: (3.1416) \*r2.

3.- Se muestra el resultado de la operación.

Prueba de escritorio (r es el número entero del que se calculará (3.1416) \*r2):



**Ejercicio 4**

\*PROBLEMA: Determinar el resultado de: y= .

\*RESTRICCIONES: [()[a].

\*DATOS DE ENTRADA: Número real.

\*DATOS DE SALIDA: El resultado de y, según

\*DOMINIO: Todos los números reales.

**SOLUCIÓN:**

1. Solicitar un número real a, b, c.

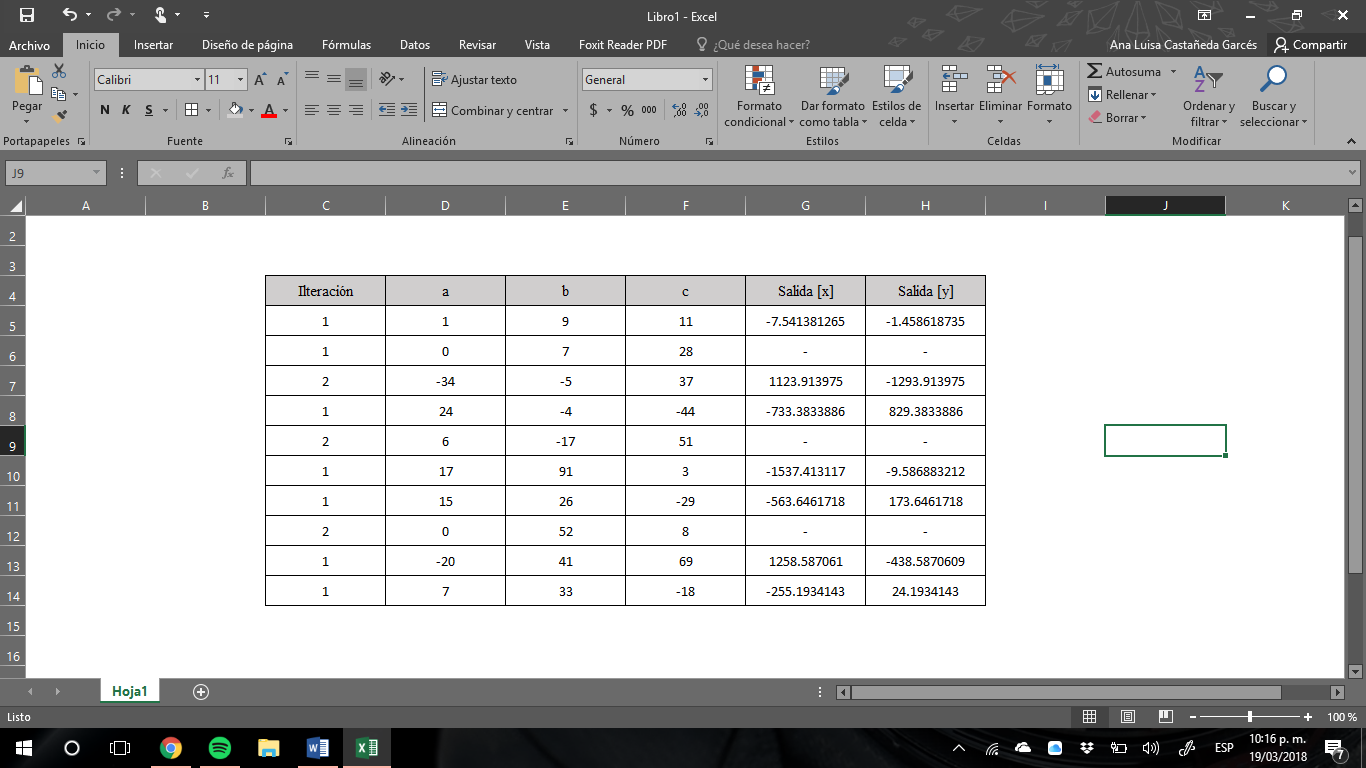
2. Si en la operación de los números ingresados: [()] se regresa al punto 1.

2.1. Si en la operación de los números ingresados: [a] se regresa al punto 1.

3. Si el cumplen los puntos 2 y 2.1 se realiza la siguiente operación: x= y se crea la variable x.

3.1. Si el cumplen los puntos 2 y 2.1 se realiza la siguiente operación: y= y se crea la variable y.

4.- Se muestra el resultado de las variables x, y.

 Prueba de escritorio [x es el número que se calculará : ( x=), y es el número que se calcula: (y=) ],

Conclusiones

Se cumplieron los objetivos porque pudimos elaborar algoritmos correctos y eficientes para la solución de problemas siguiendo las etapas de Análisis y Diseño pertenecientes al Ciclo de vida del software.

También me ayudó a repasar mis conocimientos en Excel y con este tipo de problemas te hace pensar mucho analíticamente si estas escribiendo y/o pidiendo lo que en realidad quieres, porque la computadora va a hacer lo que le digas, la cuestión es cómo lo escribes a la computadora.