

**Contenido**

[**Objetivo.** 3](#_Toc524997432)

[**Introducción.** 3](#_Toc524997433)

[**Estructuras de control de flujo.** 4](#_Toc524997434)

[**Estructura de control secuencial.** 5](#_Toc524997435)

[**Estructuras de control condicionales (o selectivas)** 5](#_Toc524997436)

[**Estructuras de control iterativas o repetitivas** 7](#_Toc524997437)

[**Funciones** 8](#_Toc524997438)

[**Hacer un programa para que el usuario pida el sabor de un helado e imprimir resultado. 12**](#_Toc524997439)

[**Ejercicio del examen 14**](#_Toc524997440)

[**Funciones 14**](#_Toc524997441)

[**1.-Calculadora para 2 variables (+-\*/) hacer mediante llamadas a función 14**](#_Toc524997442)

[**2.-Menù 3 deportes 18**](#_Toc524997443)

[**3.- tablas de multiplicar del 1 al 10, el usuario proporciona el valor a calcular 19**](#_Toc524997444)

# **Objetivo.**

Elaborar diagramas de flujo que representen soluciones algorítmicas vistas como una serie de acciones que comprendan un proceso.

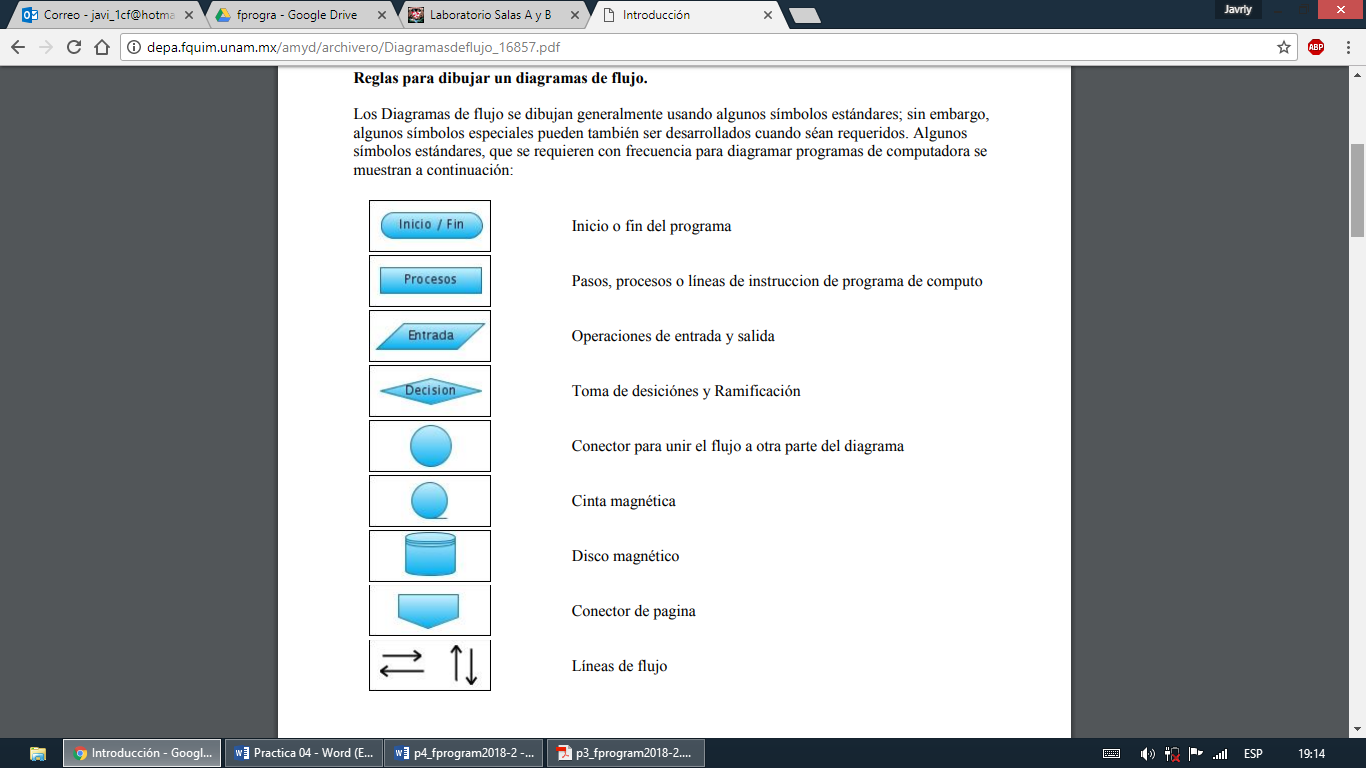
# **Introducción.**

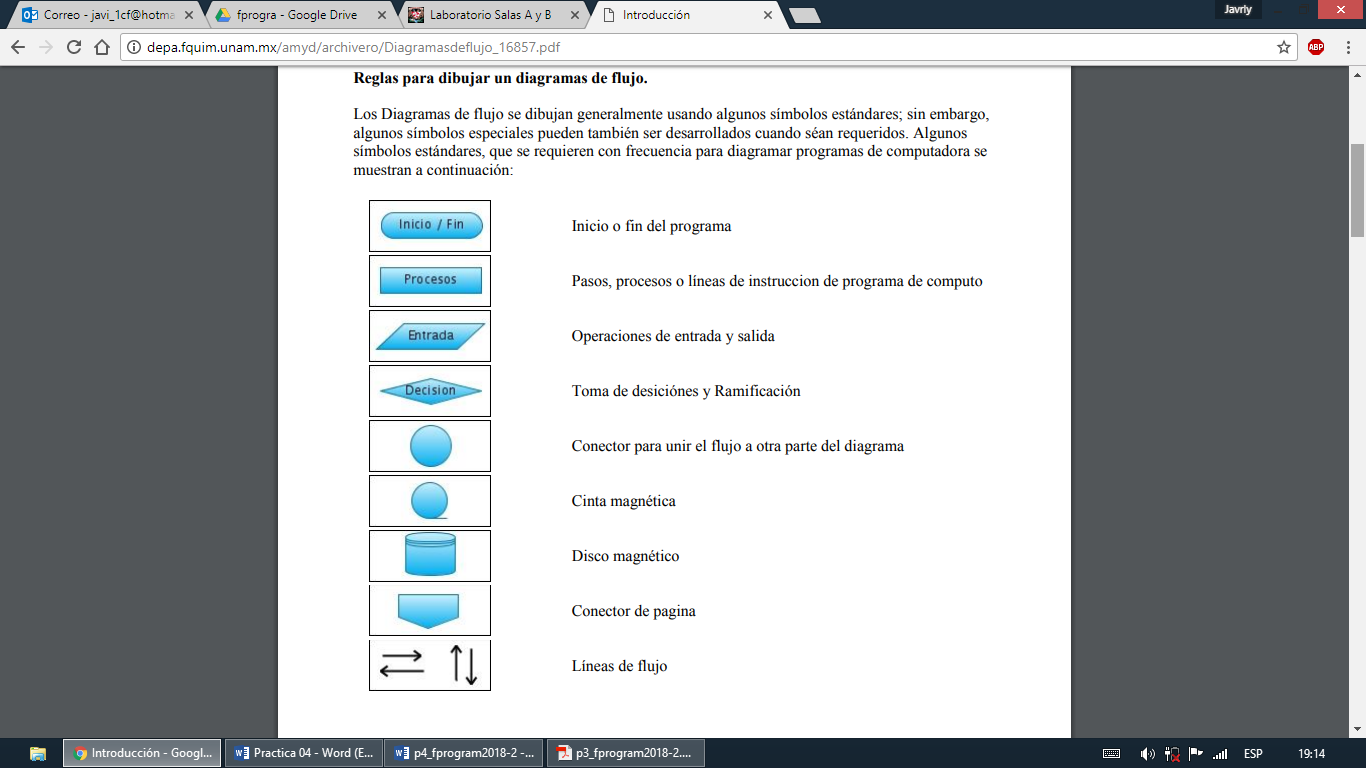
Los diagramas de flujo son una manera de representar visualmente el flujo de datos a través de sistemas de tratamiento de información. Los diagramas de flujo describen que operaciones y en que secuencia se requieren para solucionar un problema dado.

Un diagrama de flujo u organigrama es una representación diagramática que ilustra la secuencia de las operaciones que se realizarán para conseguir la solución de un problema. Los diagramas de flujo se dibujan generalmente antes de comenzar a programar el código frente a la computadora. Los diagramas de flujo facilitan la comunicación entre los programadores y la gente del negocio. Estos diagramas de flujo desempeñan un papel vital en la programación de un problema y facilitan la comprensión de problemas complicados y sobre todo muy largos. Una vez que se dibuja el diagrama de flujo, llega a ser fácil escribir el programa en cualquier idioma de alto nivel. Vemos a menudo cómo los diagramas de flujo nos dan ventaja al momento de explicar el programa a otros.

Por lo tanto, está correcto decir que un diagrama de flujo es una necesidad para la documentación mejor de un programa complejo.

Reglas para dibujar un diagrama de flujo.

Los Diagramas de flujo se dibujan generalmente usando algunos símbolos estándares; sin embargo, algunos símbolos especiales pueden también ser desarrollados cuando sean requeridos. Algunos símbolos estándares, que se requieren con frecuencia para diagramar programas de computadora se muestran a continuación:



Reglas para la creación de Diagramas

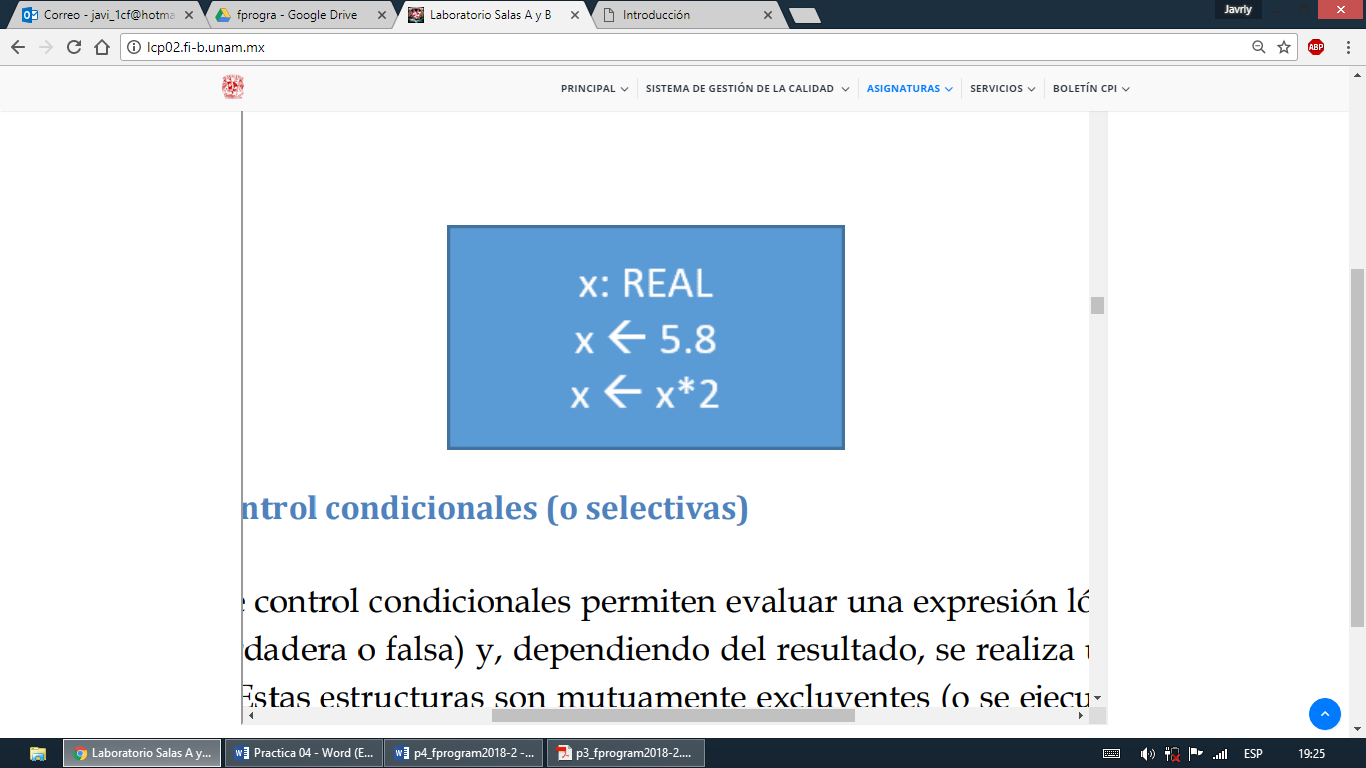
* + - 1. Los Diagramas de flujo deben escribirse de arriba hacia abajo, y/o de izquierda a derecha.
      2. Los símbolos se unen con líneas, las cuales tienen en la punta una flecha que indica la dirección que fluye la información procesos, se deben de utilizar solamente líneas de flujo horizontal o verticales (nunca diagonales).
      3. Se debe evitar el cruce de líneas, para lo cual se quisiera separar el flujo del diagrama a un sitio distinto, se pudiera realizar utilizando los conectores. Se debe tener en cuenta que solo se vana utilizar conectores cuando sea estrictamente necesario.
      4. No deben quedar líneas de flujo sin conectar
      5. Todo texto escrito dentro de un símbolo debe ser legible, preciso, evitando el uso de muchas palabras.
      6. Todos los símbolos pueden tener más de una línea de entrada, a excepción del símbolo final.
      7. Solo los símbolos de decisión pueden y deben tener mas de una línea de flujo de salida.

# **Estructuras de control de flujo.**

Permiten la ejecución condicional y la repetición de un conjunto de instrucciones.

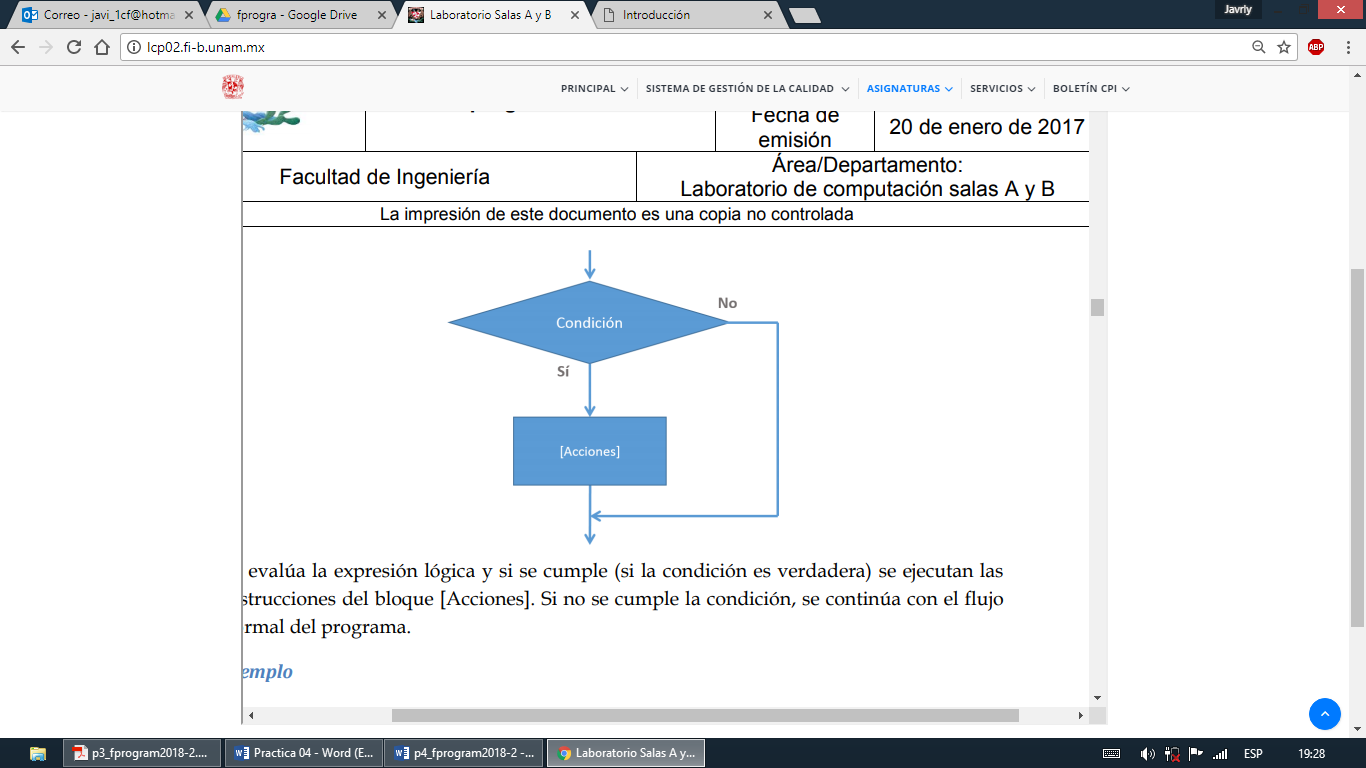
Existen 3 estructuras de control: secuencial, condicional y repetitivas o iterativas.

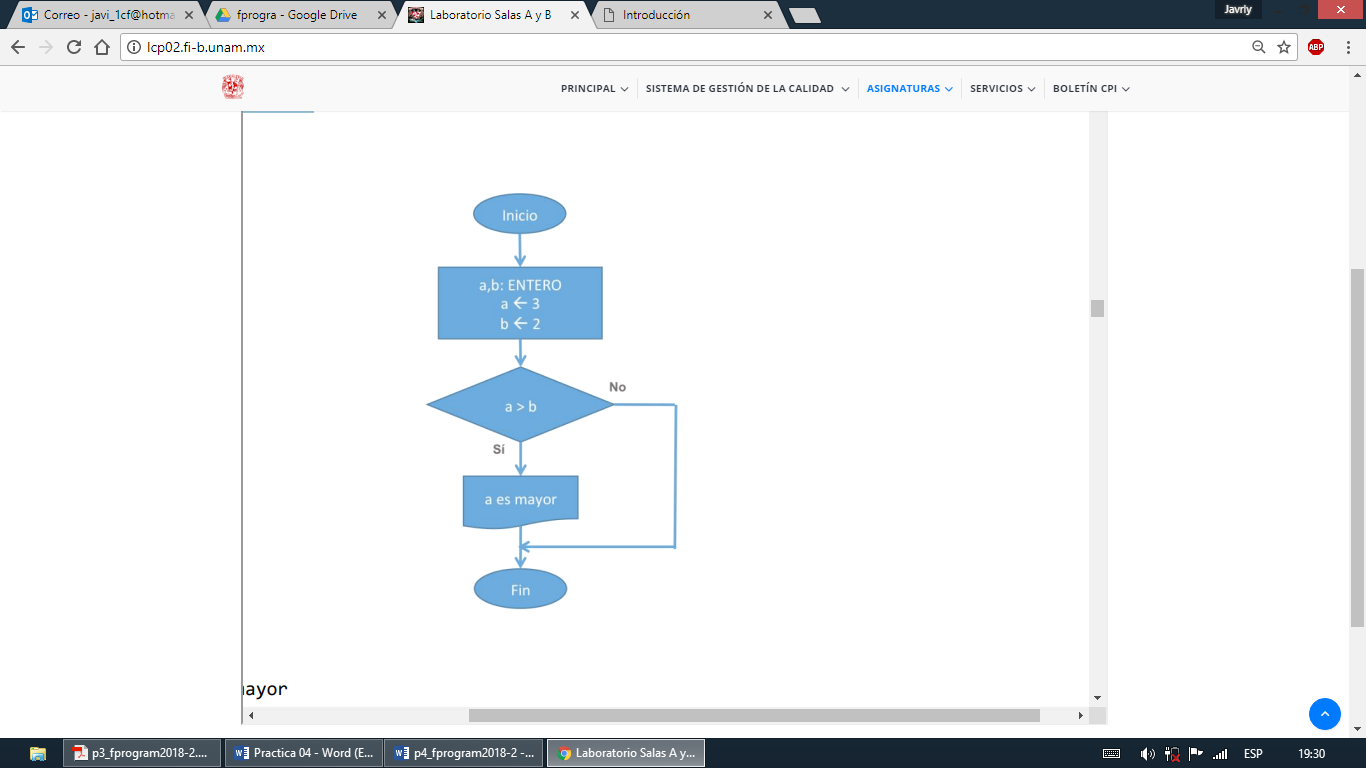
## **Estructura de control secuencial.**

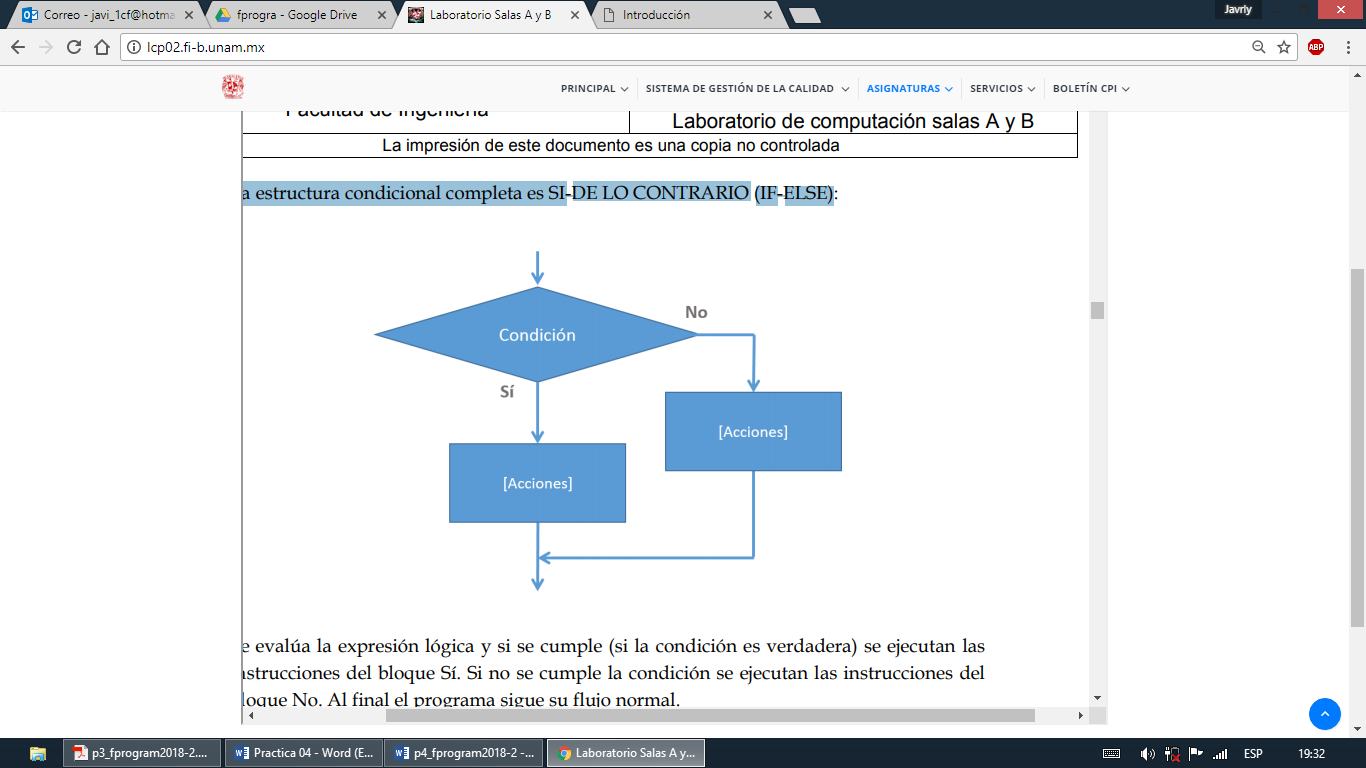
 Las estructuras de control secuenciales son las sentencias o declaraciones que se realizan una a continuación de otra en el orden en el que están escritas.

## **Estructuras de control condicionales (o selectivas)**

Las estructuras de control condicionales permiten evaluar una expresión lógica (condición que puede ser verdadera o falsa) y, dependiendo del resultado, se realiza uno u otro flujo de instrucciones. Estas estructuras son mutuamente excluyentes (o se ejecuta una acción o se ejecuta la otra).

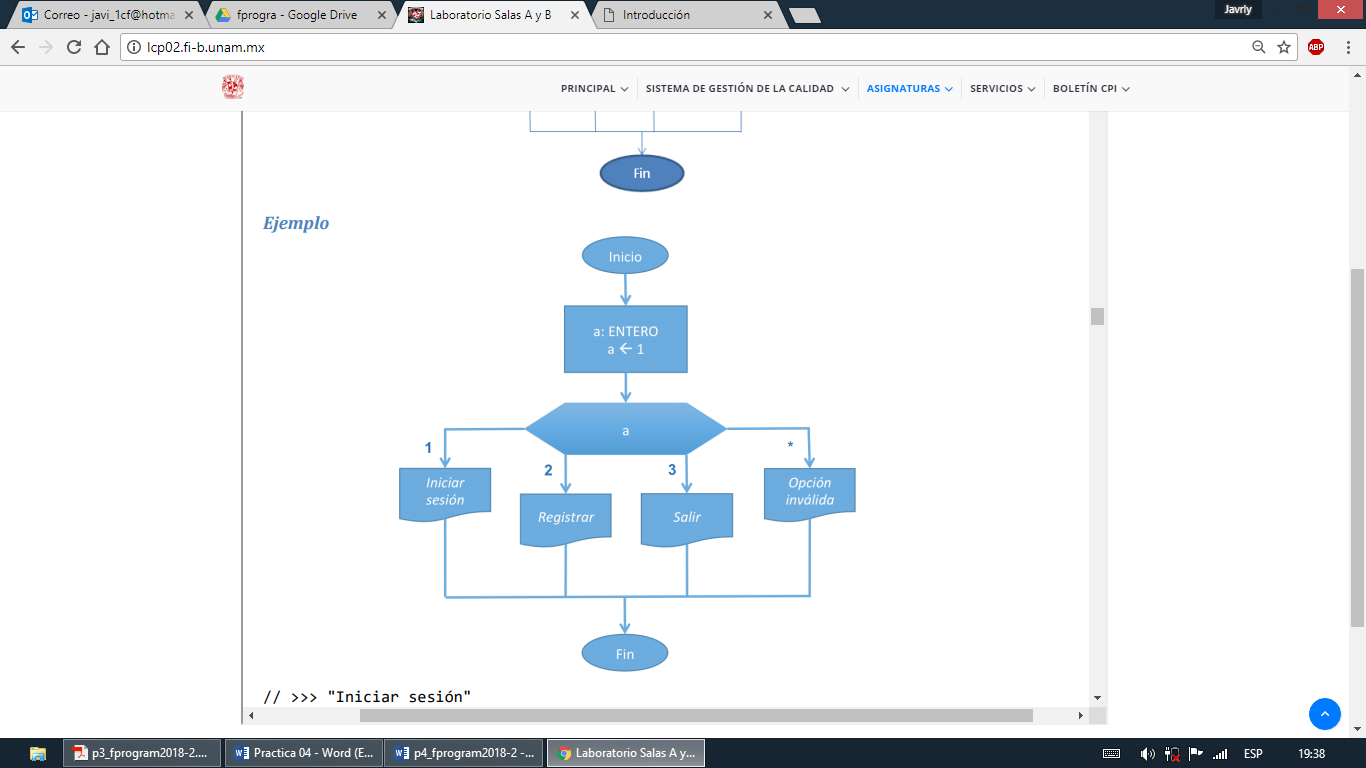
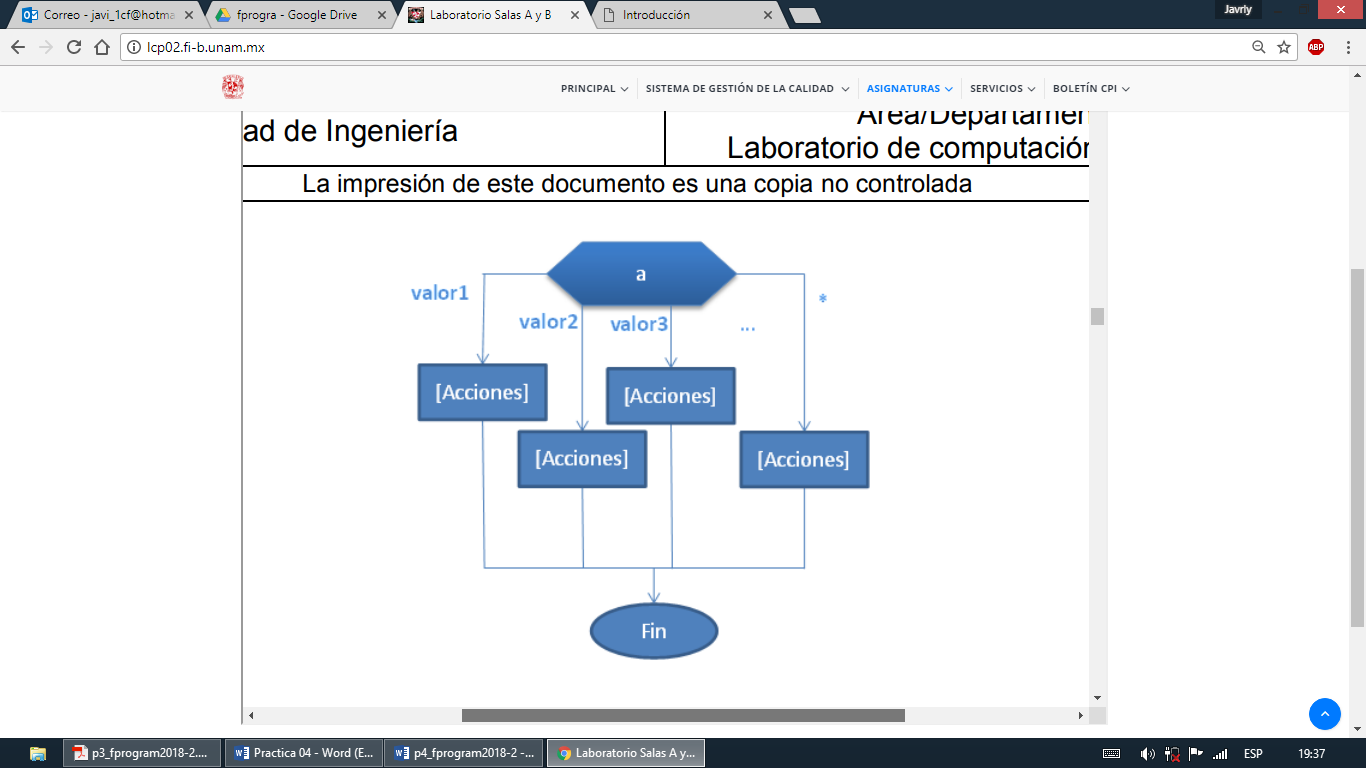
La estructura de control de flujo más simple es la estructura condicional SI (IF), su sintaxis es la siguiente:

Se evalúa la expresión lógica y si se cumple (si la condición es verdadera) se ejecutan las instrucciones del bloque [Acciones]. Si no se cumple la condición, se continúa con el flujo normal del programa.

La estructura condicional completa es SI-DE LO CONTRARIO (IF-ELSE).

Se evalúa la expresión lógica y si se cumple (si la condición es verdadera) se ejecutan las instrucciones del bloque Sí. Si no se cumple la condición se ejecutan las instrucciones del bloque No. Al final el programa sigue su flujo normal.

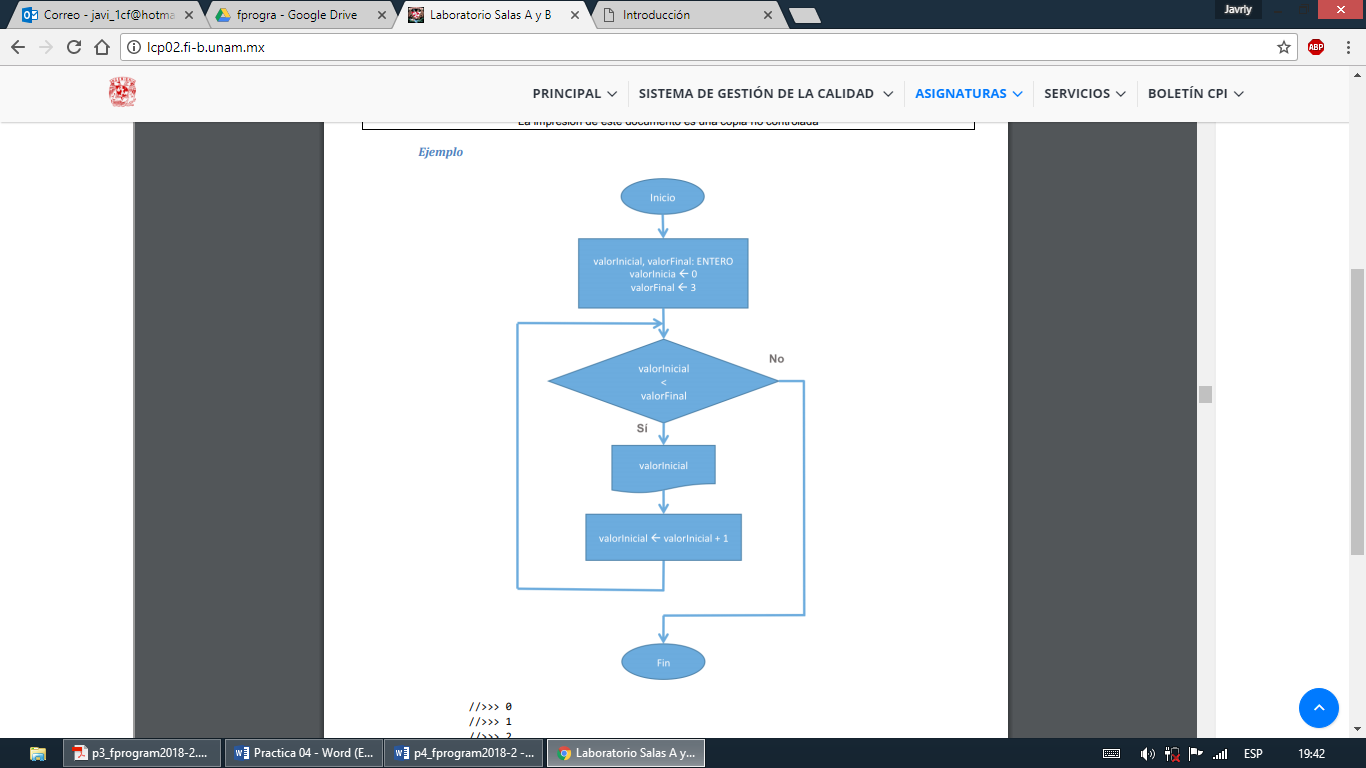
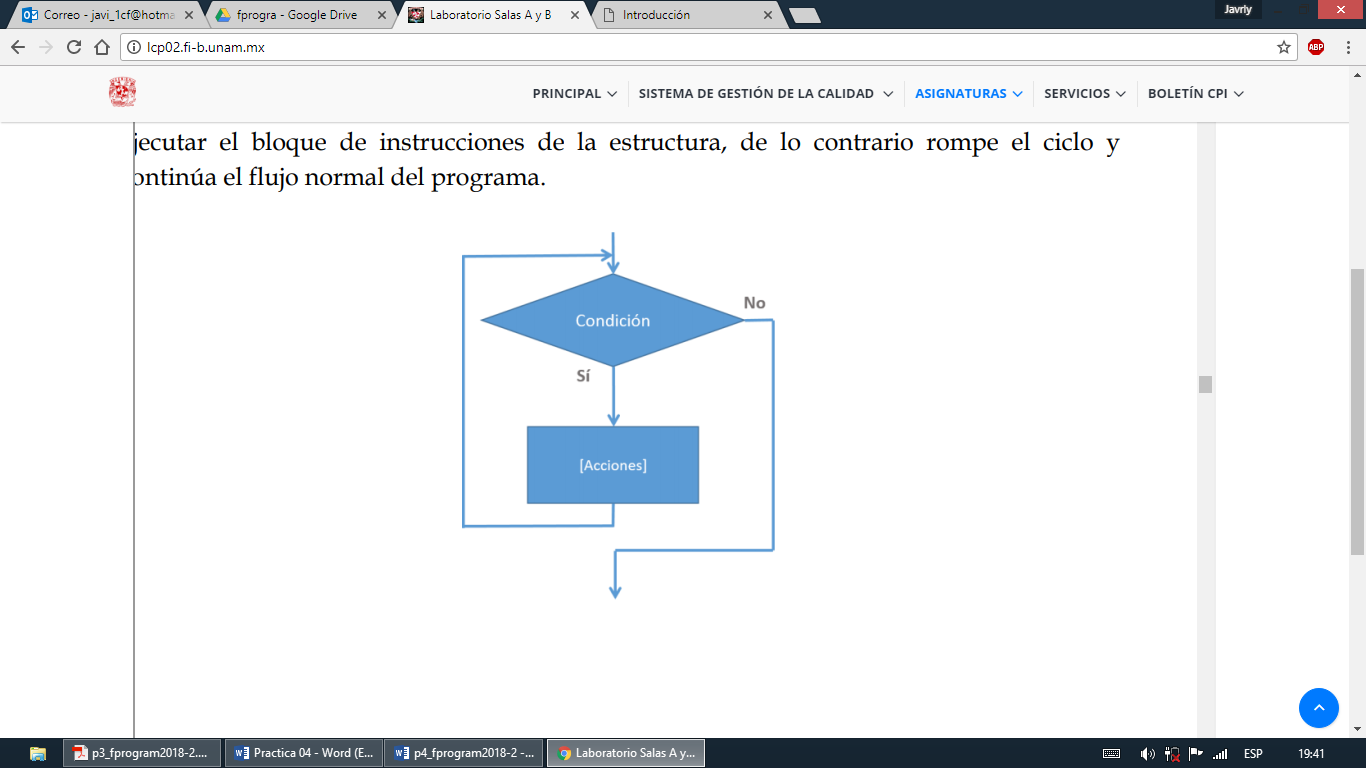
La estructura condicional SELECCIONAR-CASO valida el valor de la variable que está en el hexágono y comprueba si es igual al valor que está definido en cada caso (líneas que emanan del hexágono). Si la variable no tiene el valor de algún caso se va a la instrucción por defecto (\*).



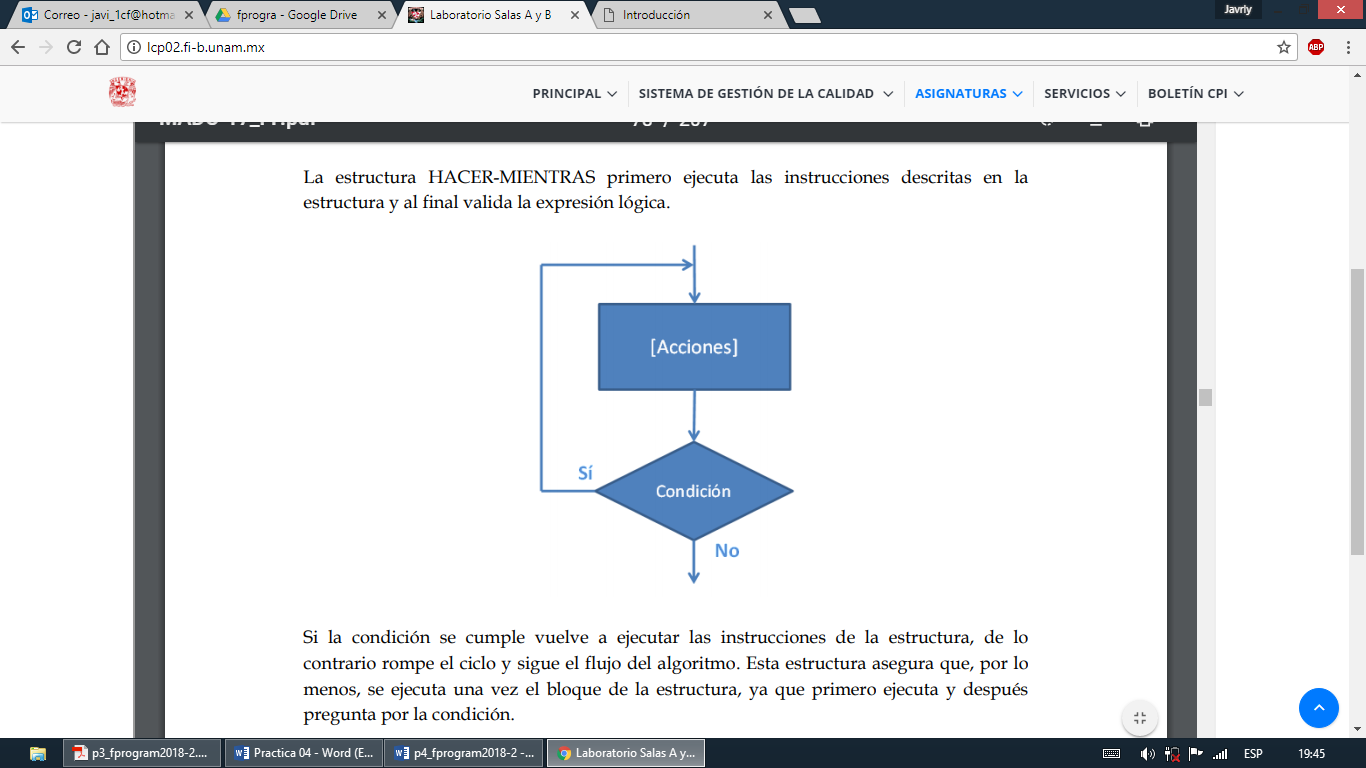
## **Estructuras de control iterativas o repetitivas**

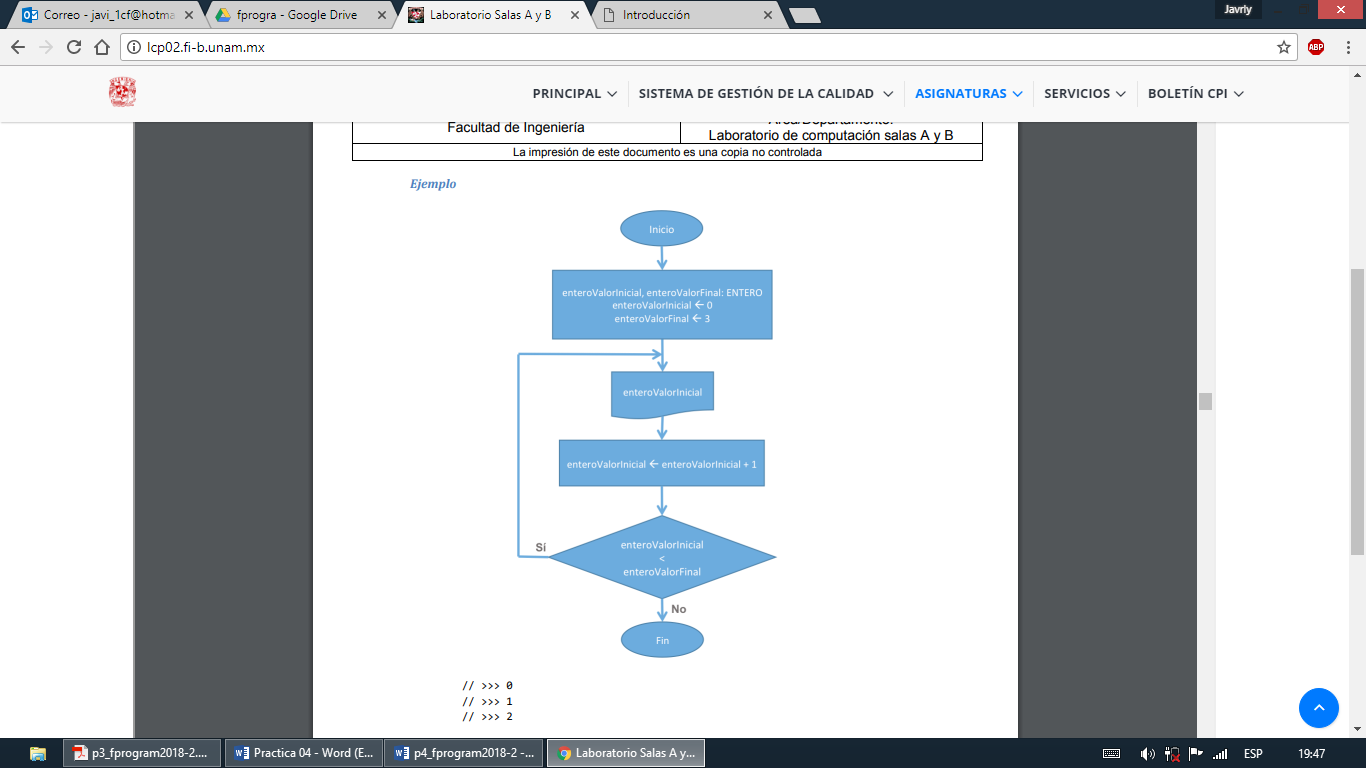
Las estructuras de control de flujo iterativo o repetitivo (también llamadas cíclicas) permiten ejecutar una serie de instrucciones mientras se cumpla la expresión lógica. Existen dos tipos de expresiones cíclicas MIENTRAS y HACER- MIENTRAS.

La estructura MIENTRAS primero valida la condición y si ésta es verdadera procede a ejecutar el bloque de instrucciones de la estructura, de lo contrario rompe el ciclo y continúa el flujo normal del programa.



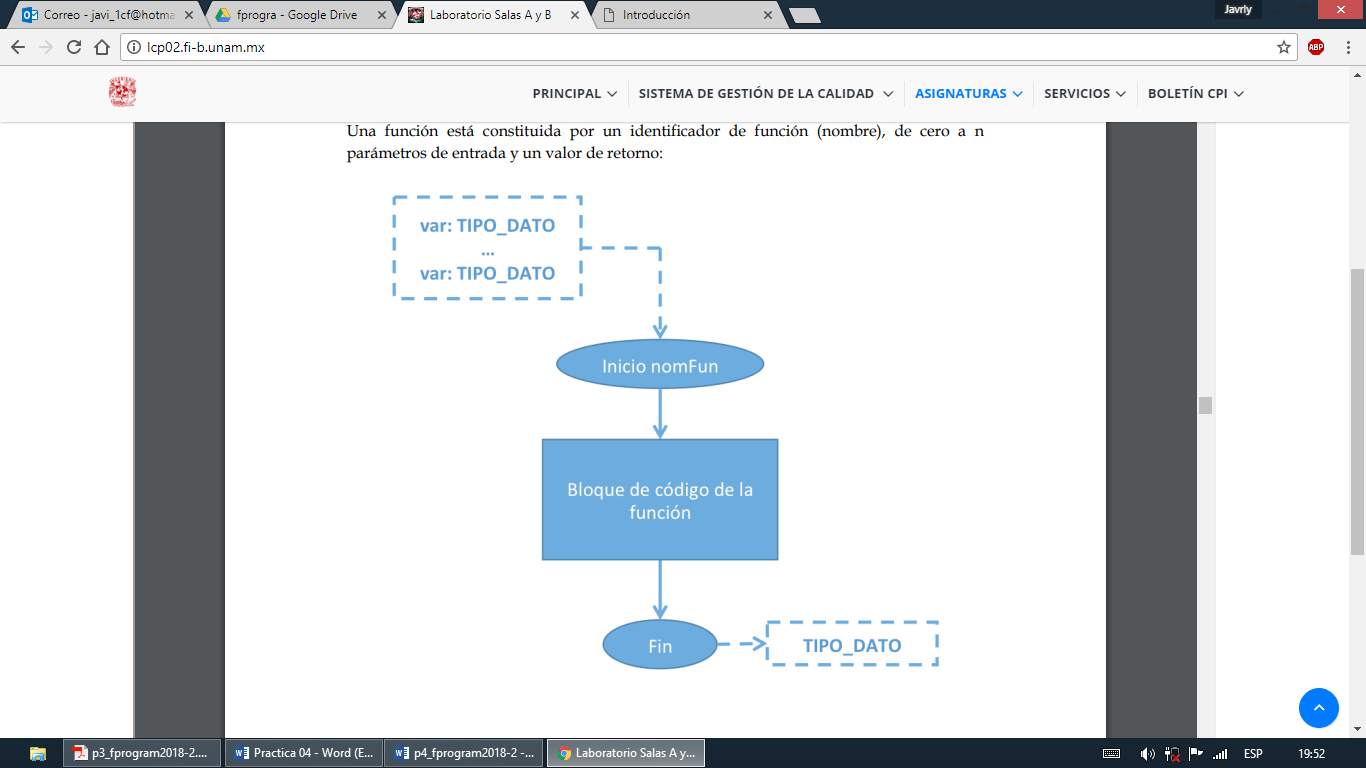
La estructura HACER-MIENTRAS primero ejecuta las instrucciones descritas en la estructura y al final valida la expresión lógica.

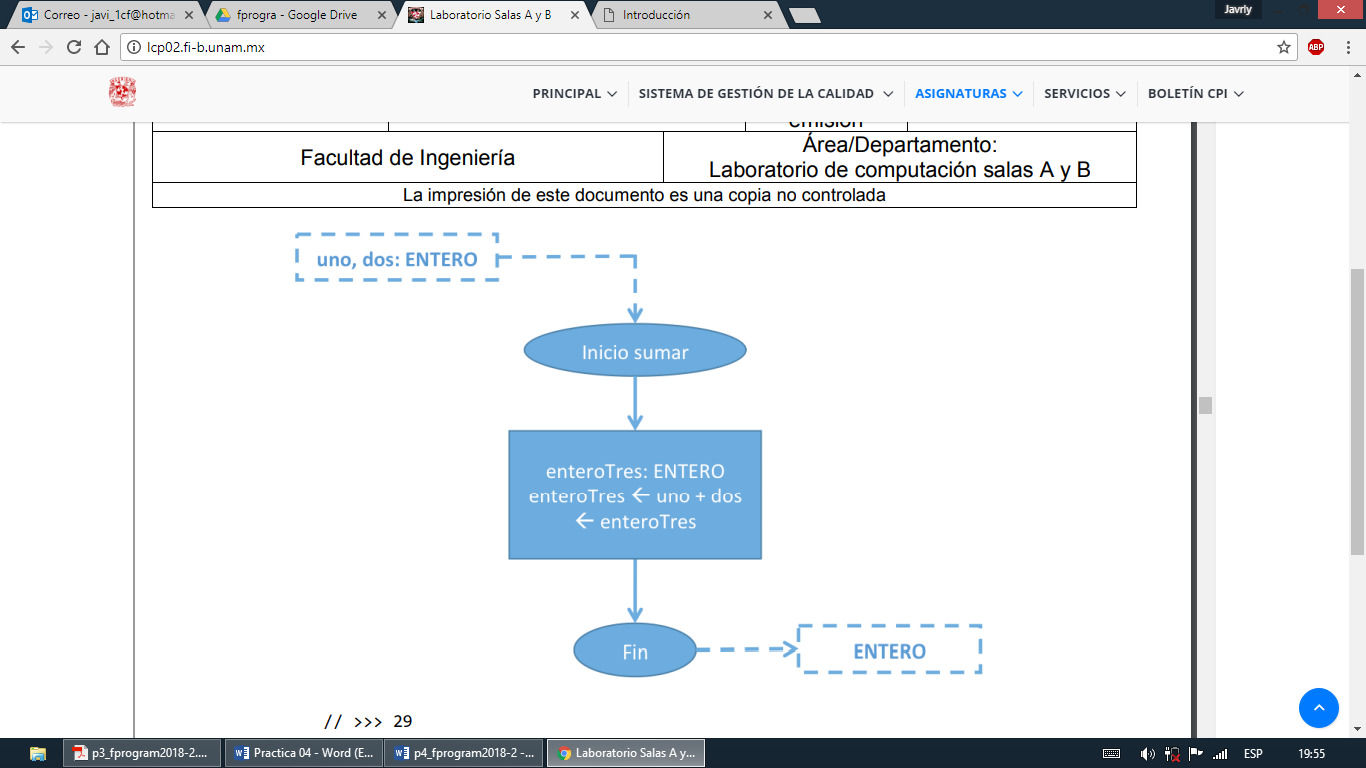


Si la condición se cumple vuelve a ejecutar las instrucciones de la estructura, de lo contrario rompe el ciclo y sigue el flujo del algoritmo. Esta estructura asegura que, por lo menos, se ejecuta una vez el bloque de la estructura, ya que primero ejecuta y después pregunta por la condición.

## **Funciones**

Cuando la solución de un problema es muy compleja se suele ocupar el diseño descendente (divide y vencerás). Este diseño implica la división de un problema en varios subprocesos más sencillos que juntos forman la solución completa. A estos subprocesos se les llaman módulos o funciones.

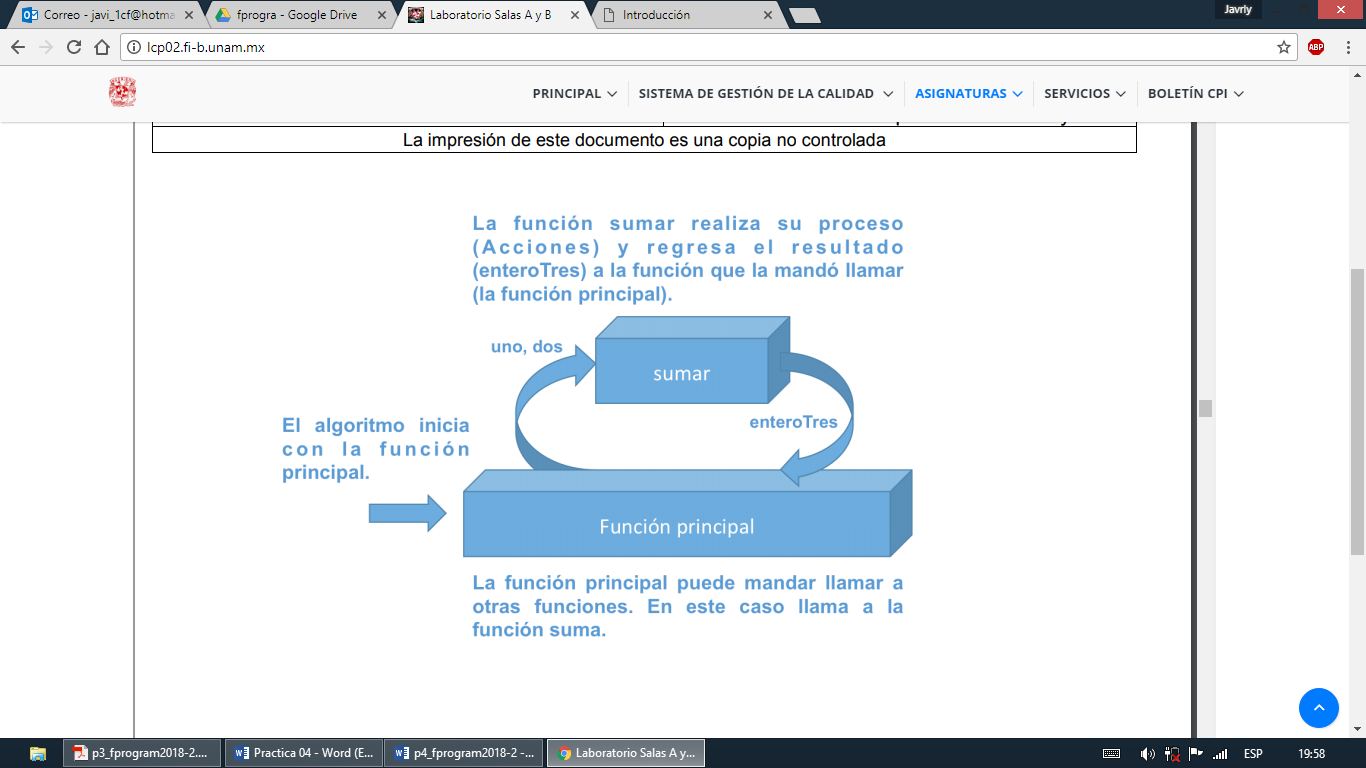
Una función está constituida por un identificador de función (nombre), de cero a n parámetros de entrada y un valor de retorno:

**nomFun** es el nombre con el que llama a la función. Las funciones pueden o no recibir algún parámetro (tipo de dato) como entrada, si la función recibe alguno se debe incluir en el recuadro inicial (el que apunta al nombre de la función). Todas las funciones pueden regresar un valor al final de su ejecución (un resultado) para ello se debe definir el dominio del conjunto de salida (tipo de dato).

**Descripción**

La primera función que se ejecuta es 'principal', ahí se crean las variables (uno y dos) y, posteriormente, se manda llamar a la función 'sumar'. La función 'sumar' recibe como parámetros dos valores enteros y devuelve como resultado un valor de tipo entero, que es la suma de los valores que se enviaron como parámetro.

Para la función 'principal' los pasos que realiza la función 'sumar' son transparentes, es decir, solo manda a llamar a la función y espera el parámetro de retorno.

La siguiente figura permite analizar la función a través del tiempo. El algoritmo inicia con la función principal, dentro de esta función se hace una llamada a una función externa (sumar). Sumar realiza su proceso (ejecuta su algoritmo) y devuelve un valor a la función principal, la cual sigue su flujo hasta que su estructura secuencial llega a su fin.

Desarrollo:

|  |  |
| --- | --- |
| Figura | Uso |
|  | Inicio fin |
| “HOLA MUNDO” | imprimir o salida de datos |
| X | ingresa variable, Lectura de variable; no texto |
| 1 | Conector en la misma hoja |
| X+2.4 | Operaciones y asignaciones, no texto |
| X<0  si  no | Condicional, no texto |
| 1 | Conexión de paginas |
|  | llamada función |

Estructura de control secuencial:

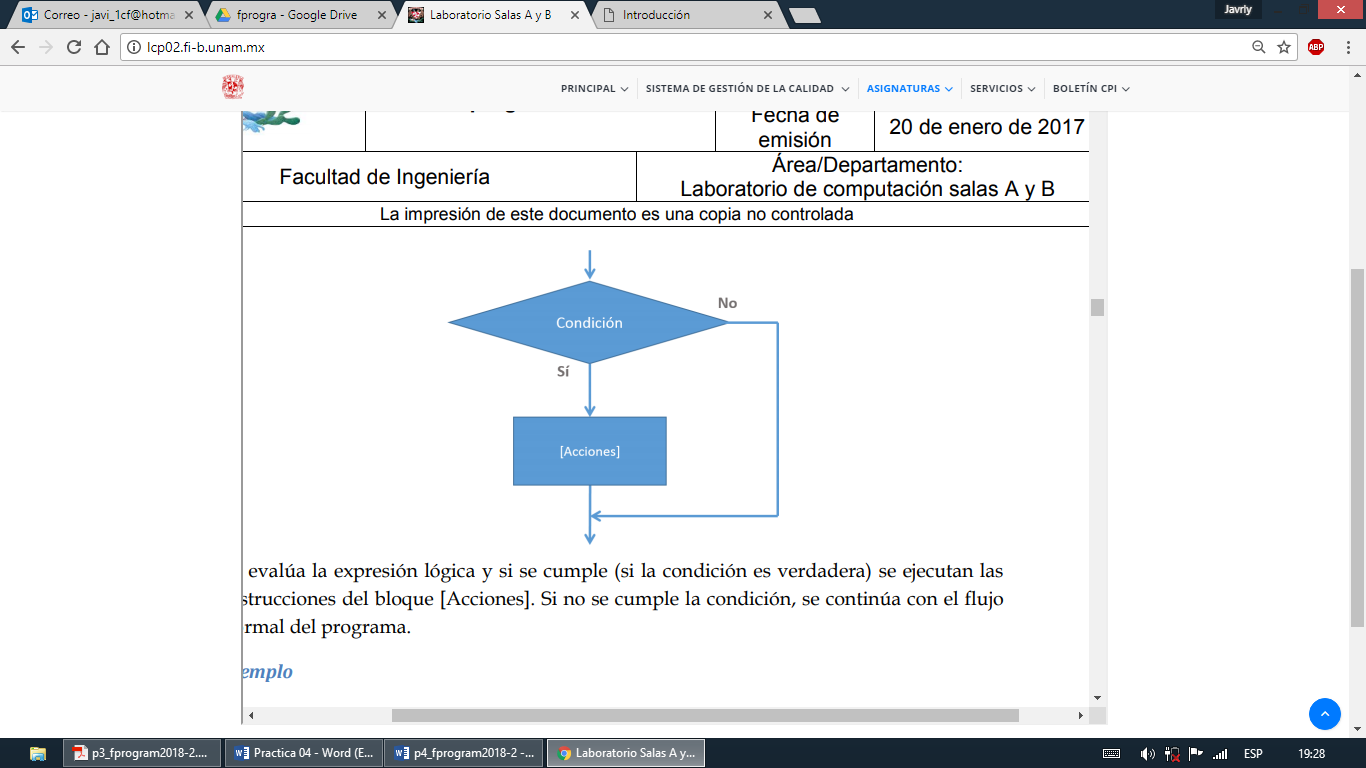
Muestra los procesos a seguir uno tras otro.

Estructura de control Iterativa:

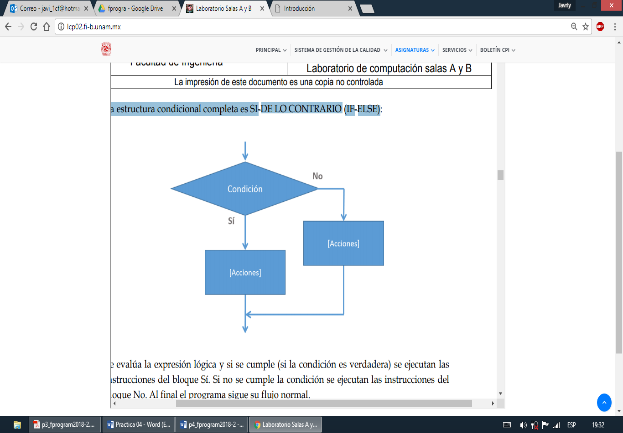
Indica un ciclo o una repetición dentro de una secuencia.

Estructura condicional o de selección:

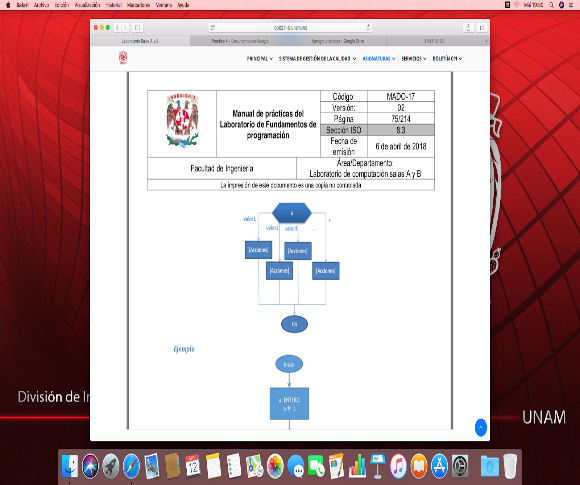
condicional simple (Mientras)



condicional completo (Hacer mientras)

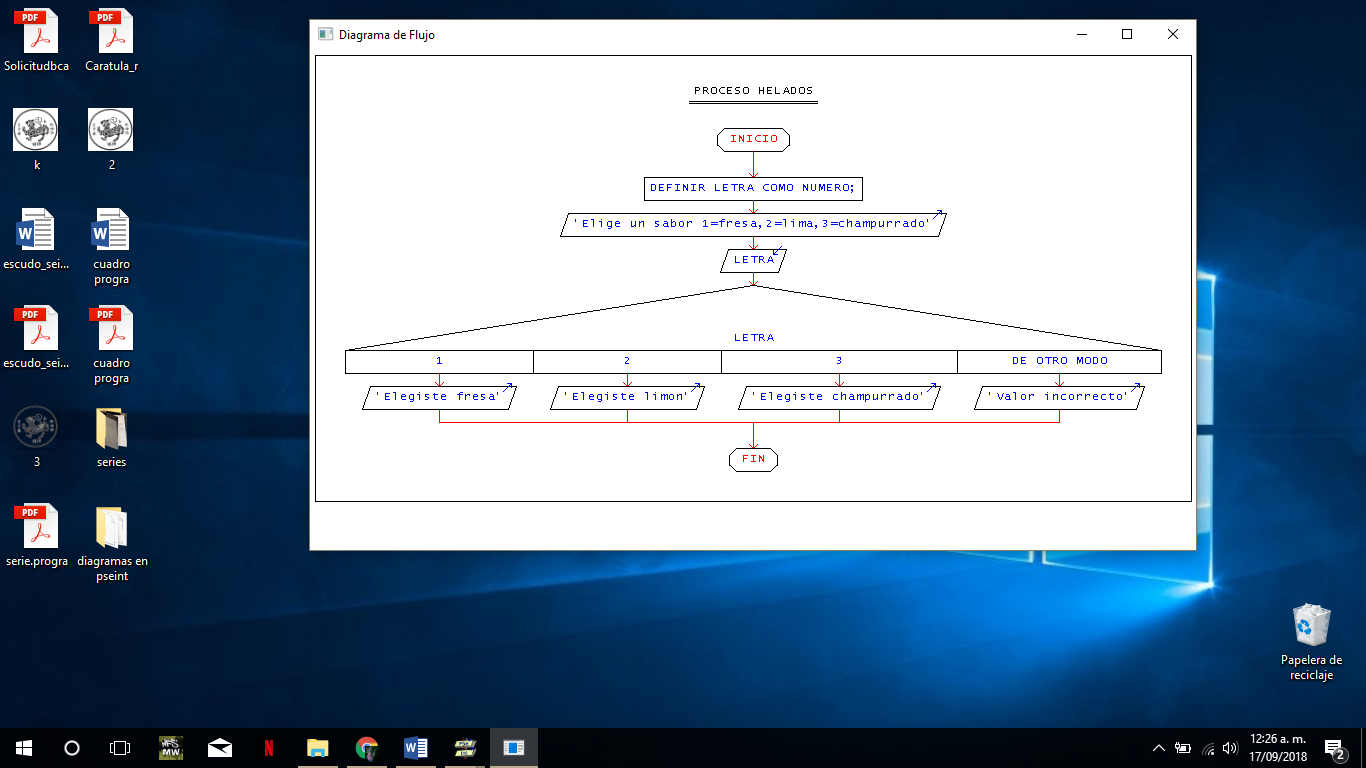


Condicional múltiple:

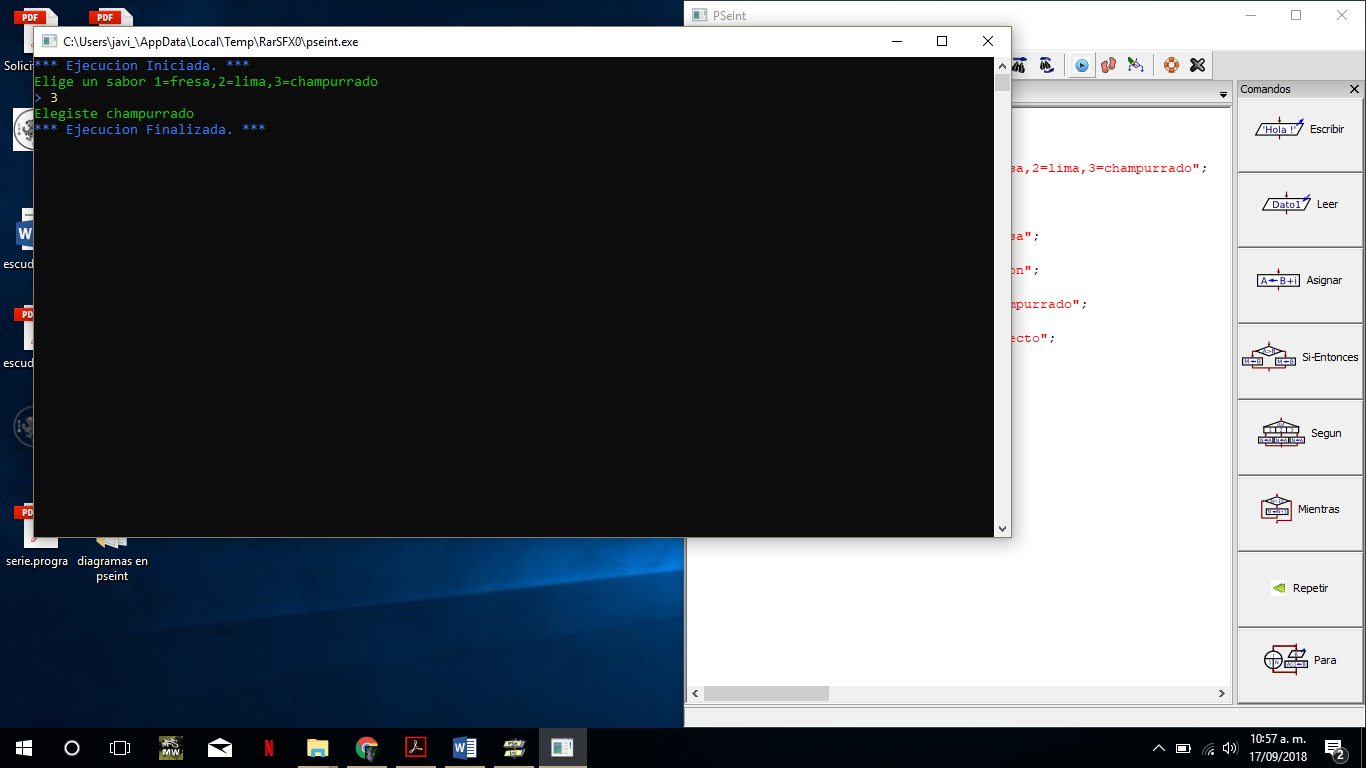


Ejemplo:

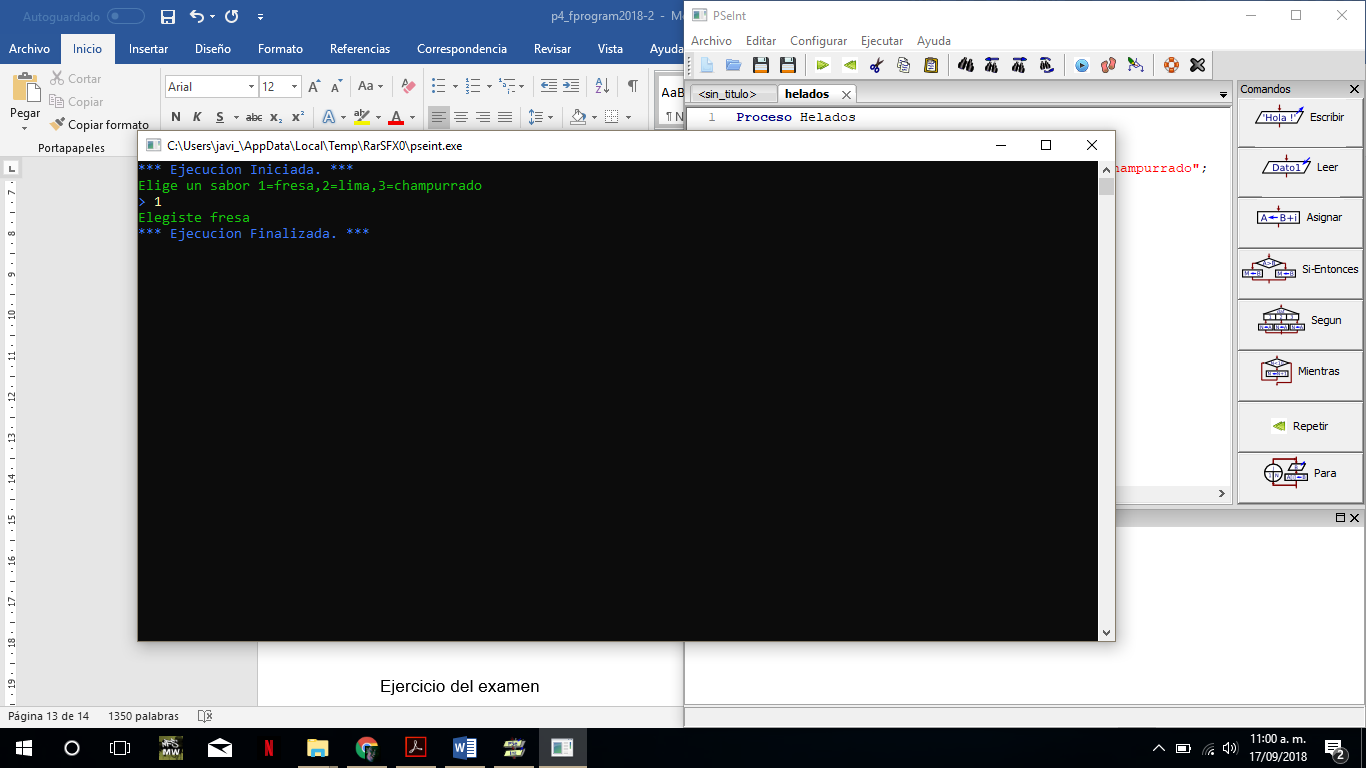
# Hacer un programa para que el usuario pida el sabor de un helado e imprimir resultado.



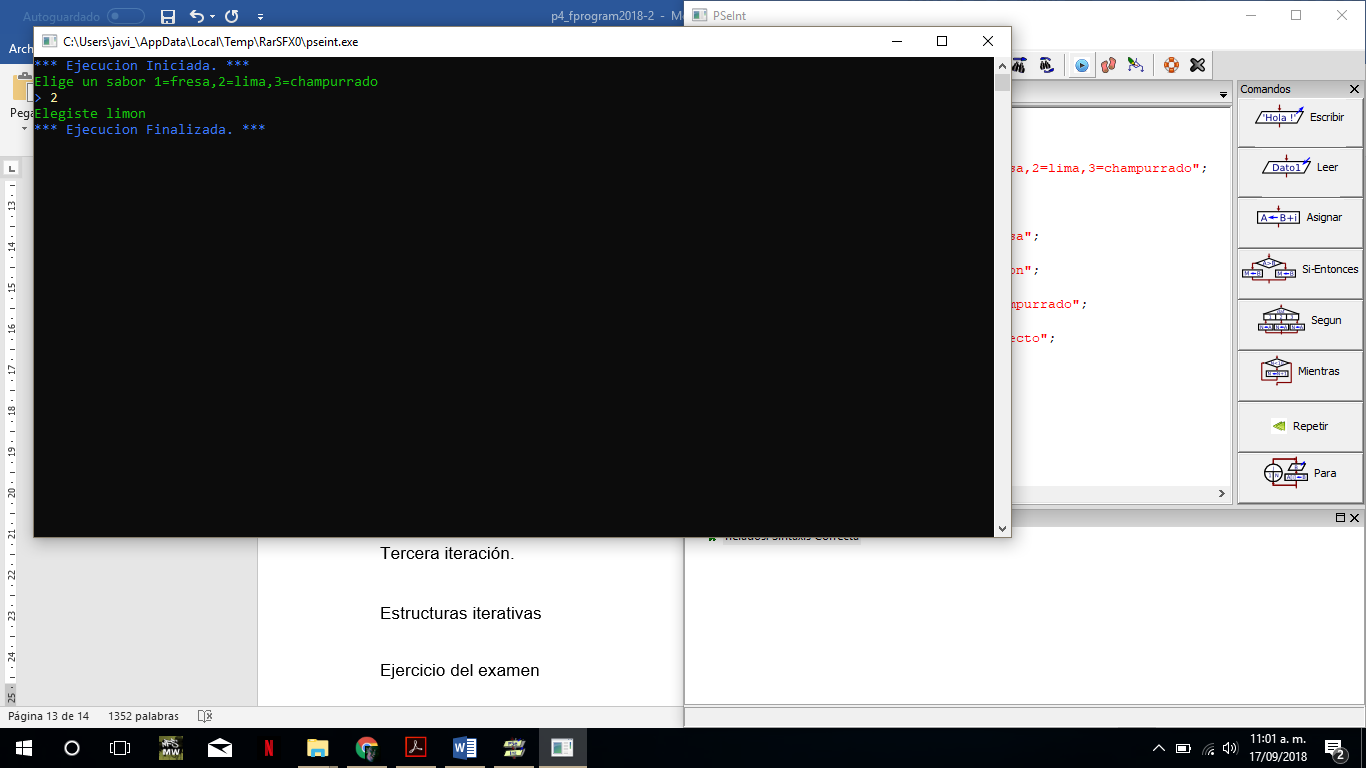
Primera iteranción.



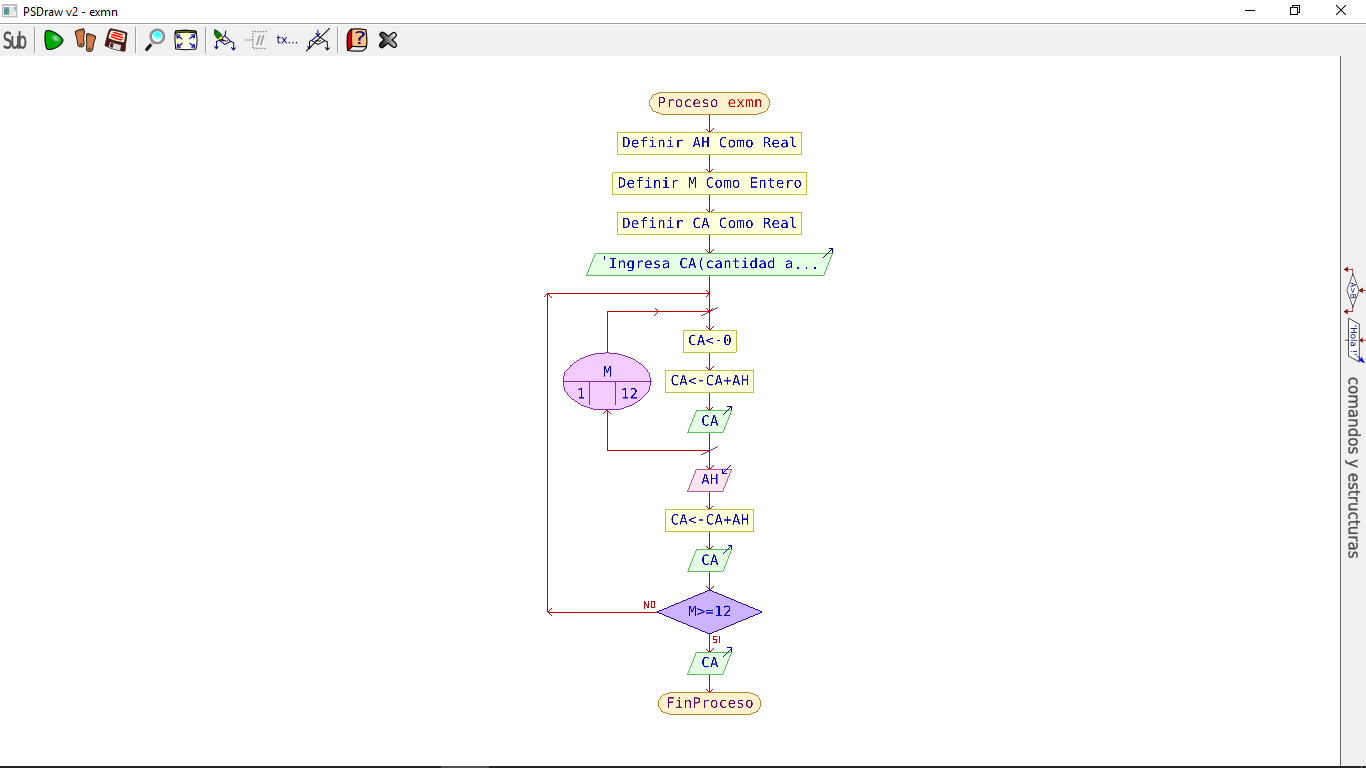
Segunda iteración.



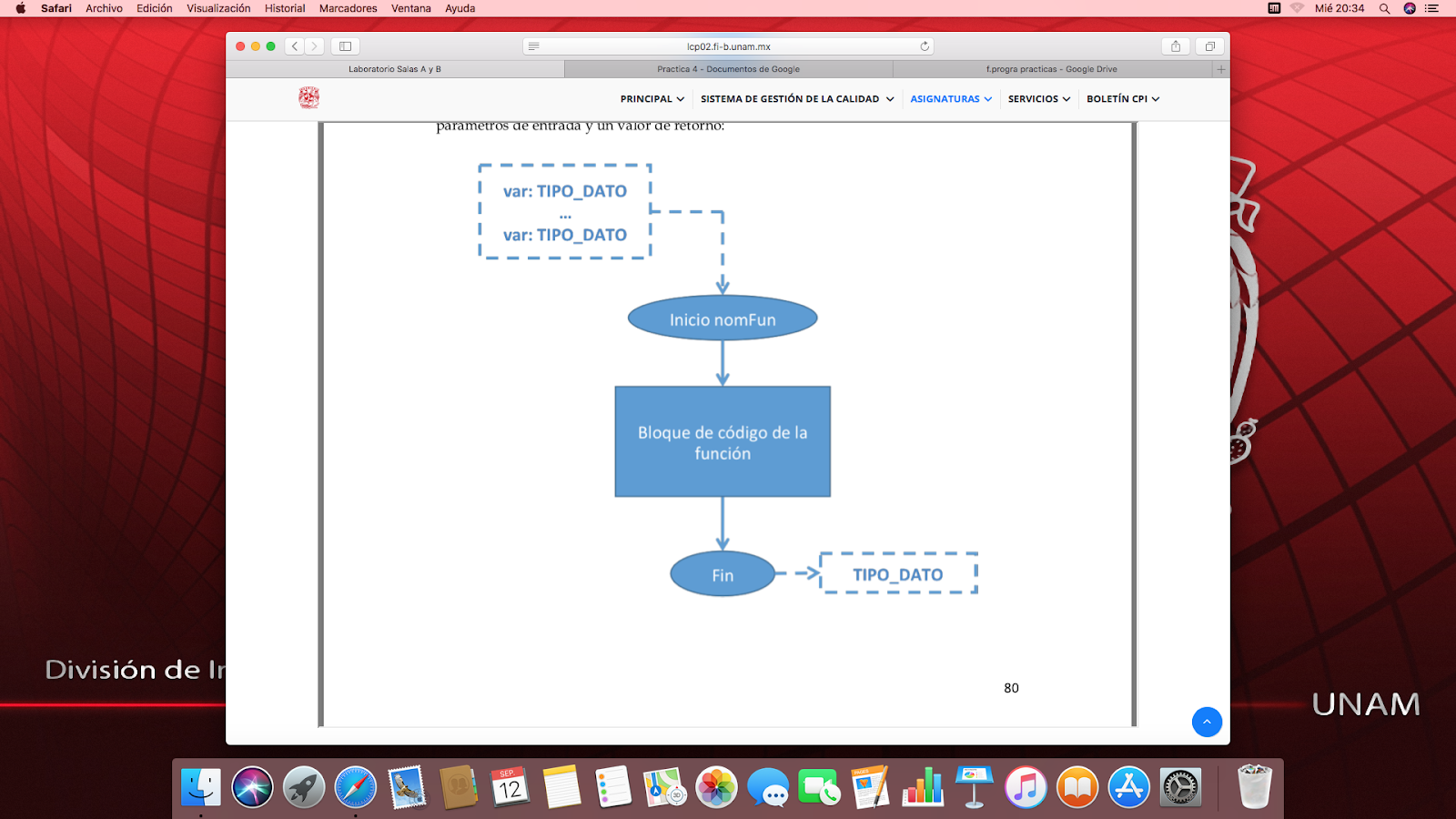
Tercera iteración.



# Ejercicio del examen

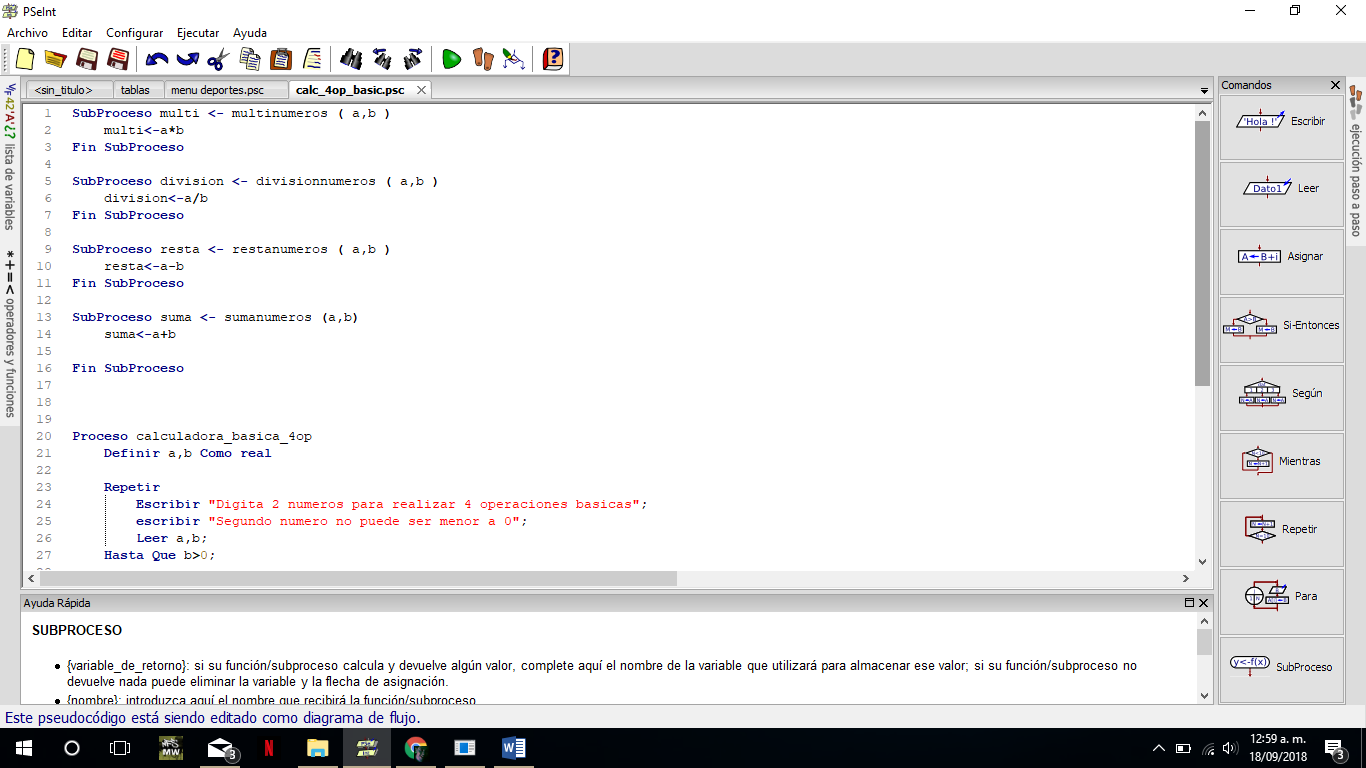


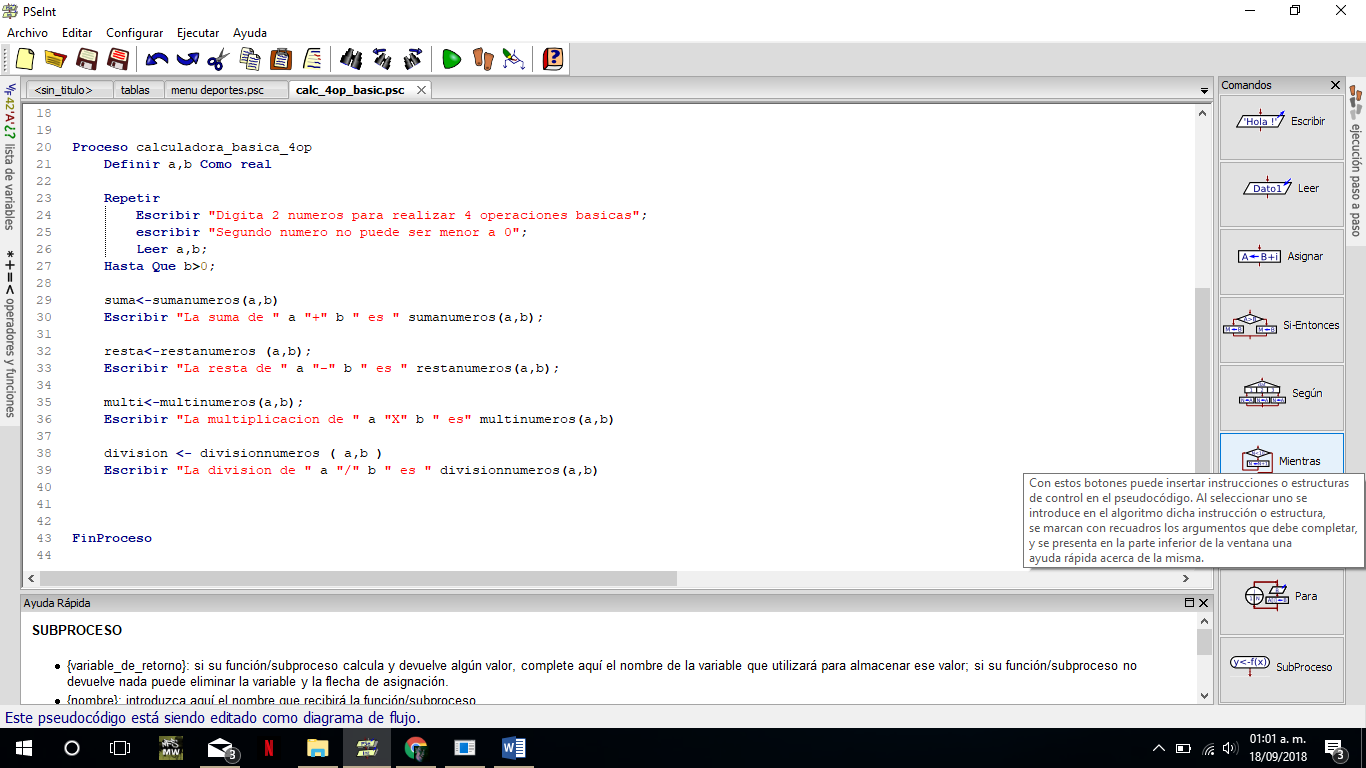
Funciones

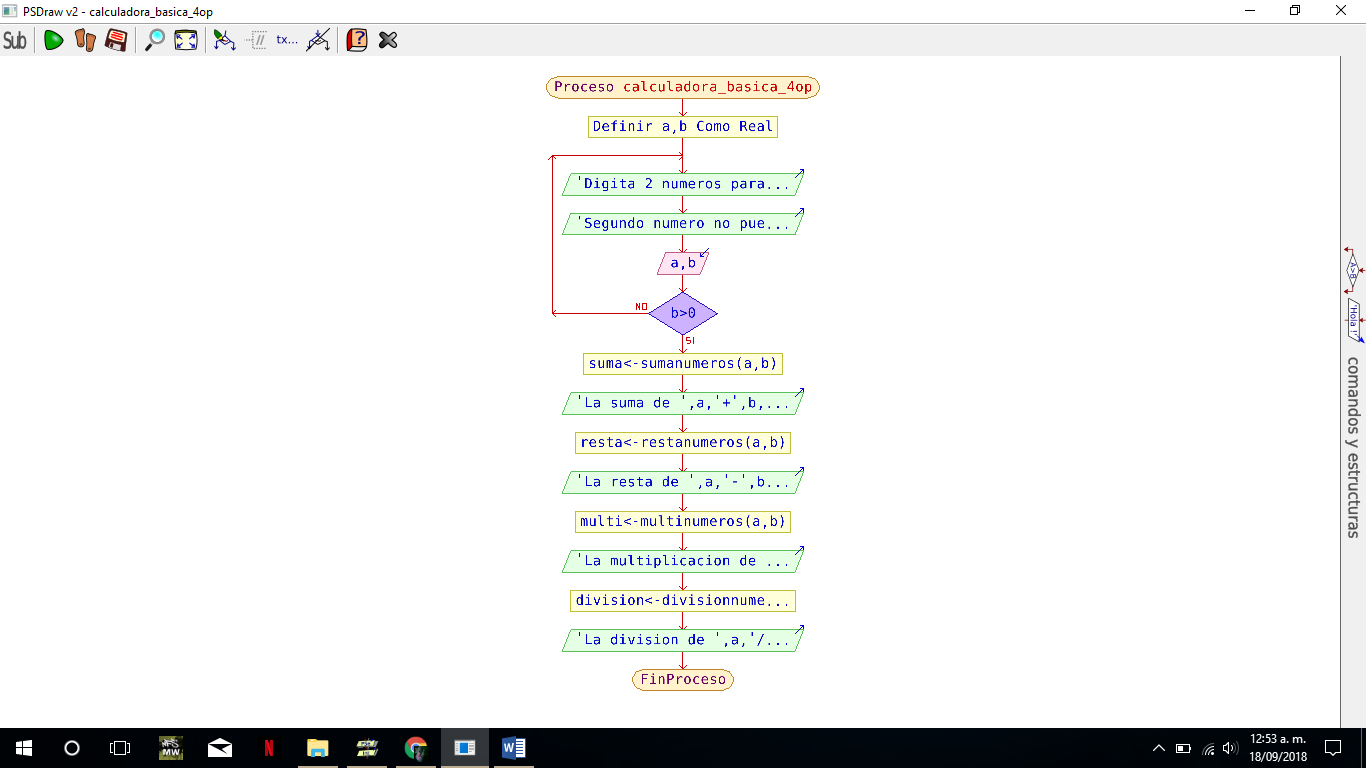


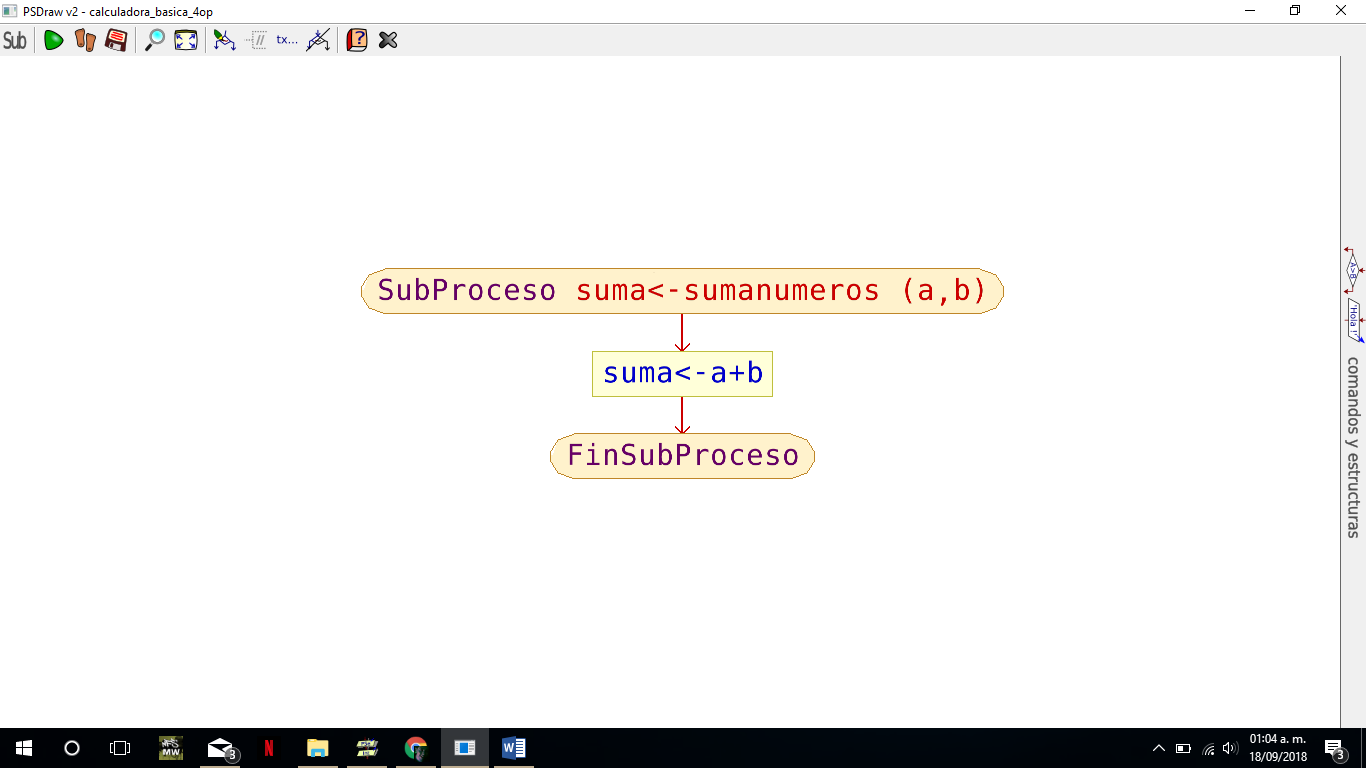
# 1.-Calculadora para 2 variables (+-\*/) hacer mediante llamadas a función

Anexo seudocódigo, ya que ocupe “pseint” para elaborar los diagramas de flujo y en el no se muestran las figuras de llamadas a función como anterior mente están explicadas.

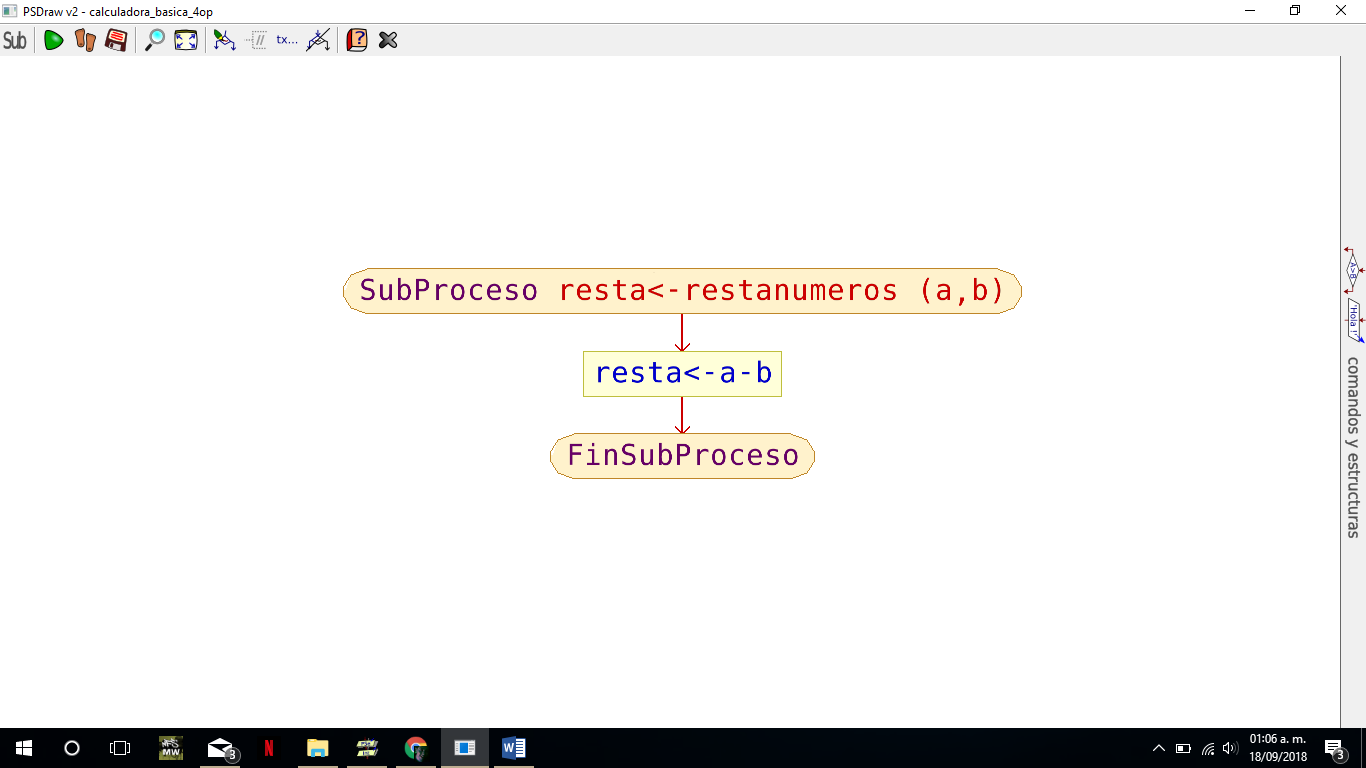


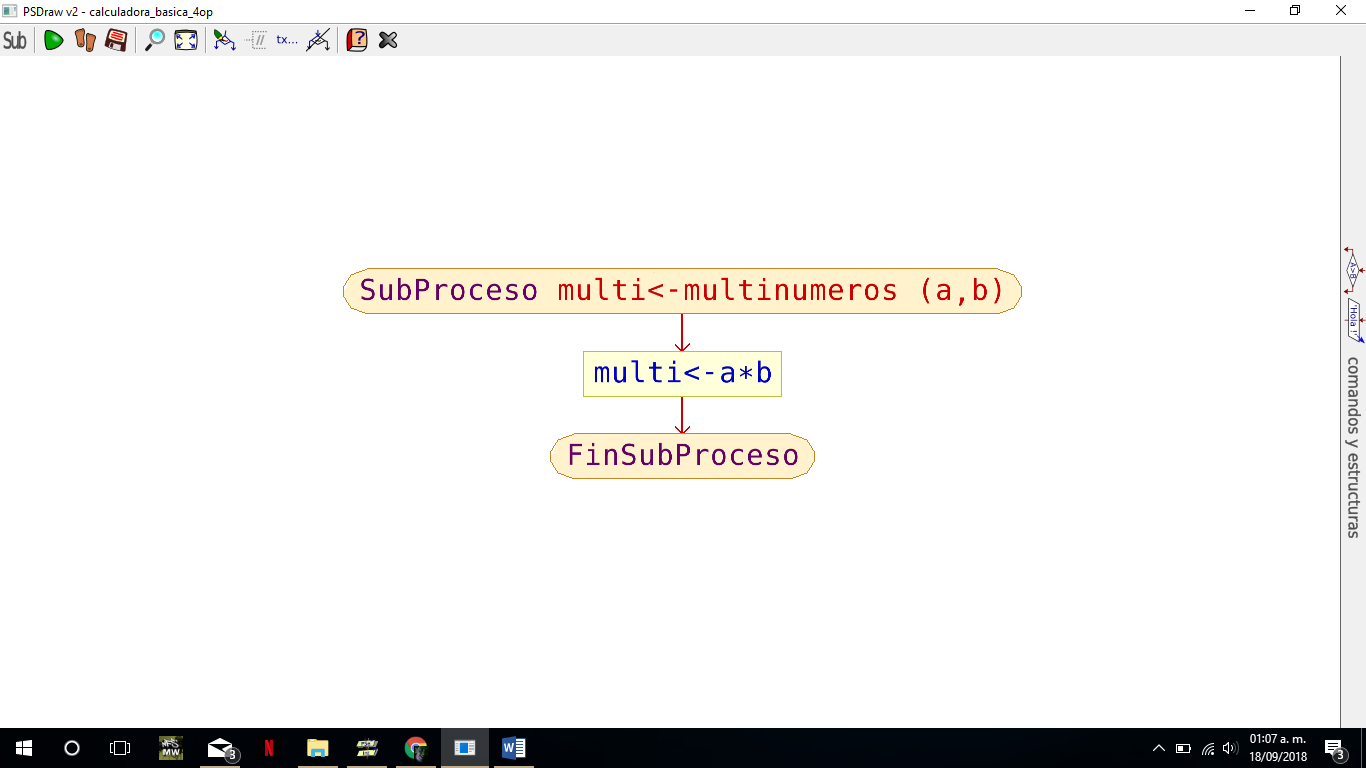


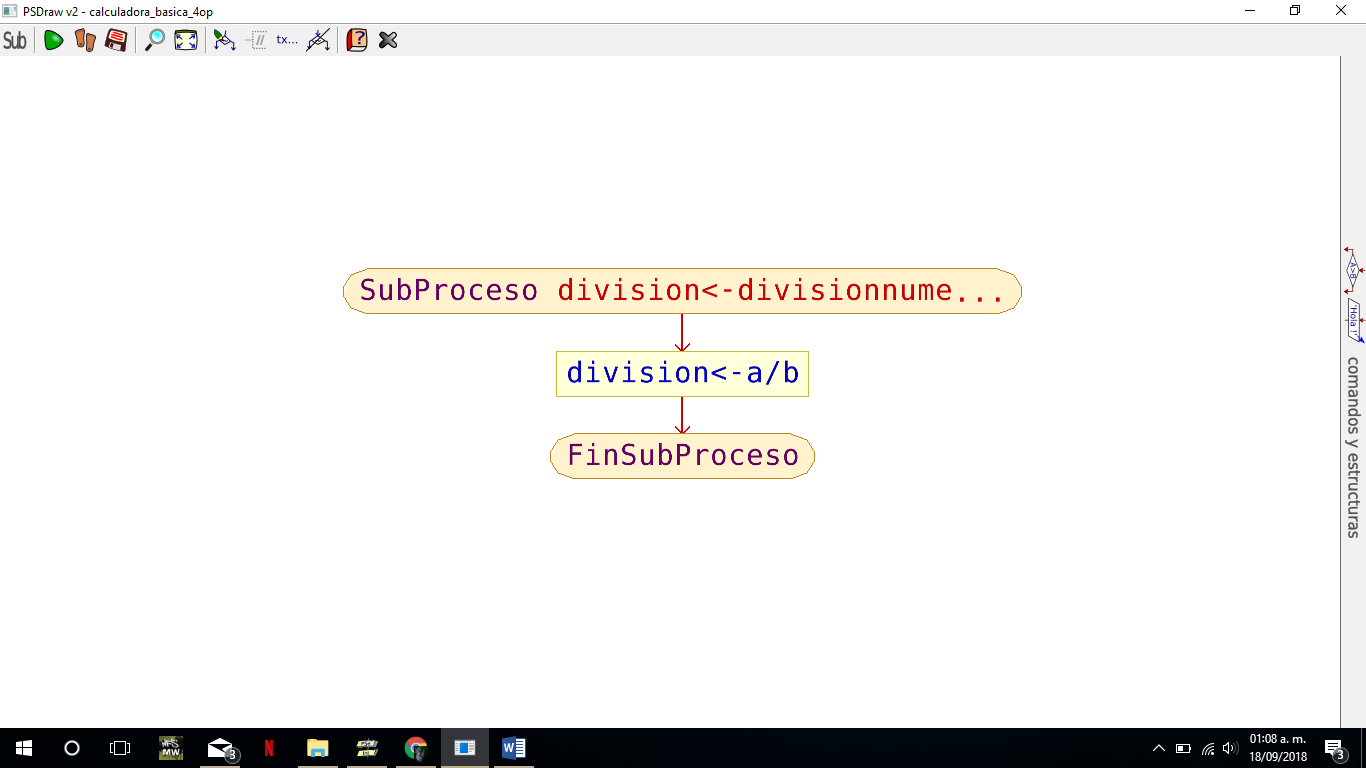
Sub proceso suma



Sub proceso resta

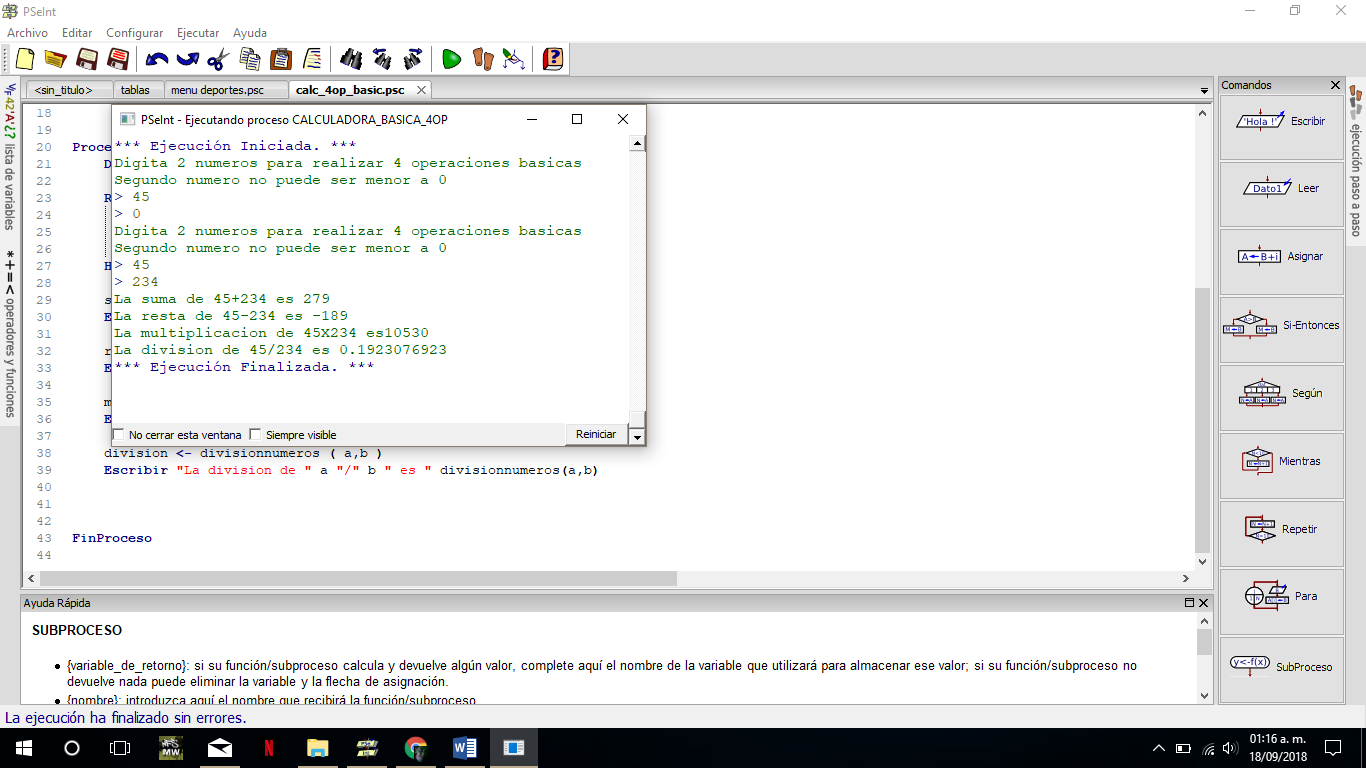
Sub proceso multiplicación

Sub proceso división

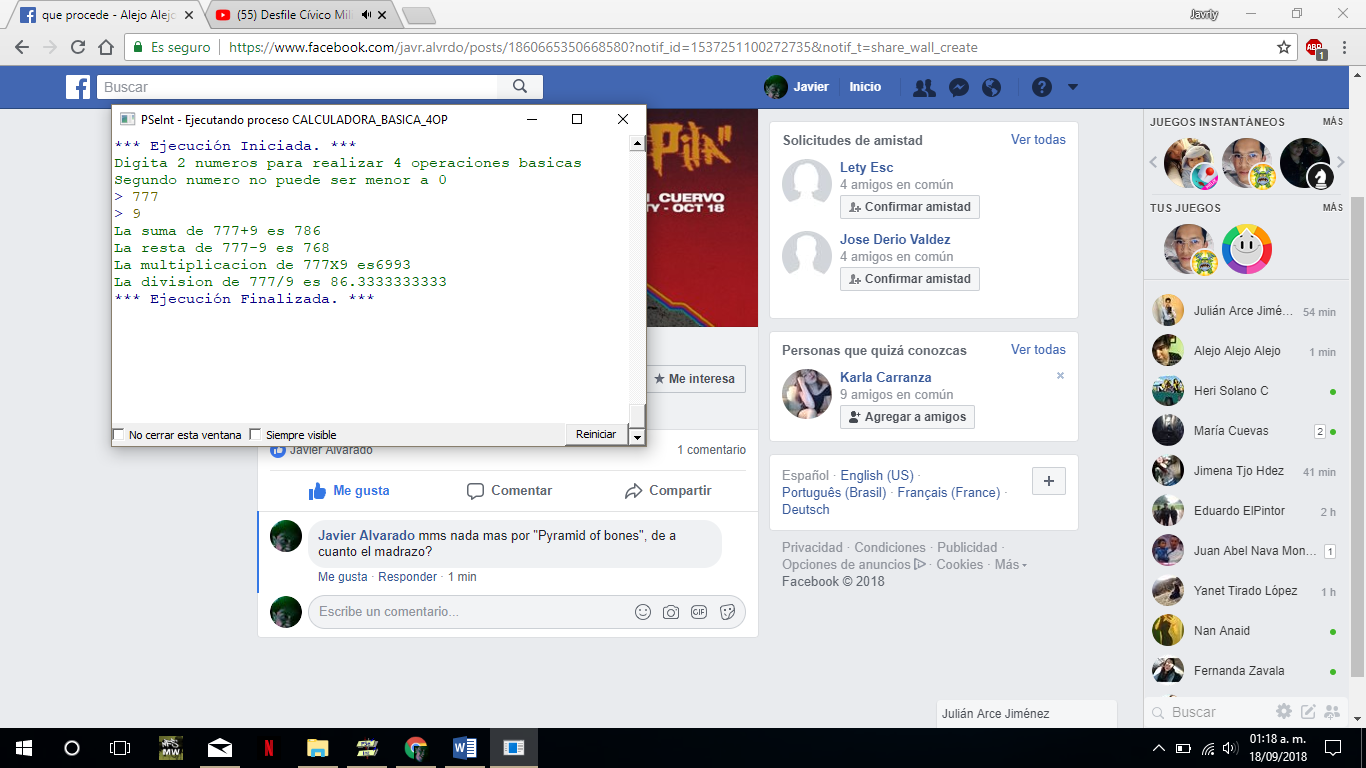


Pruebas de escritorio

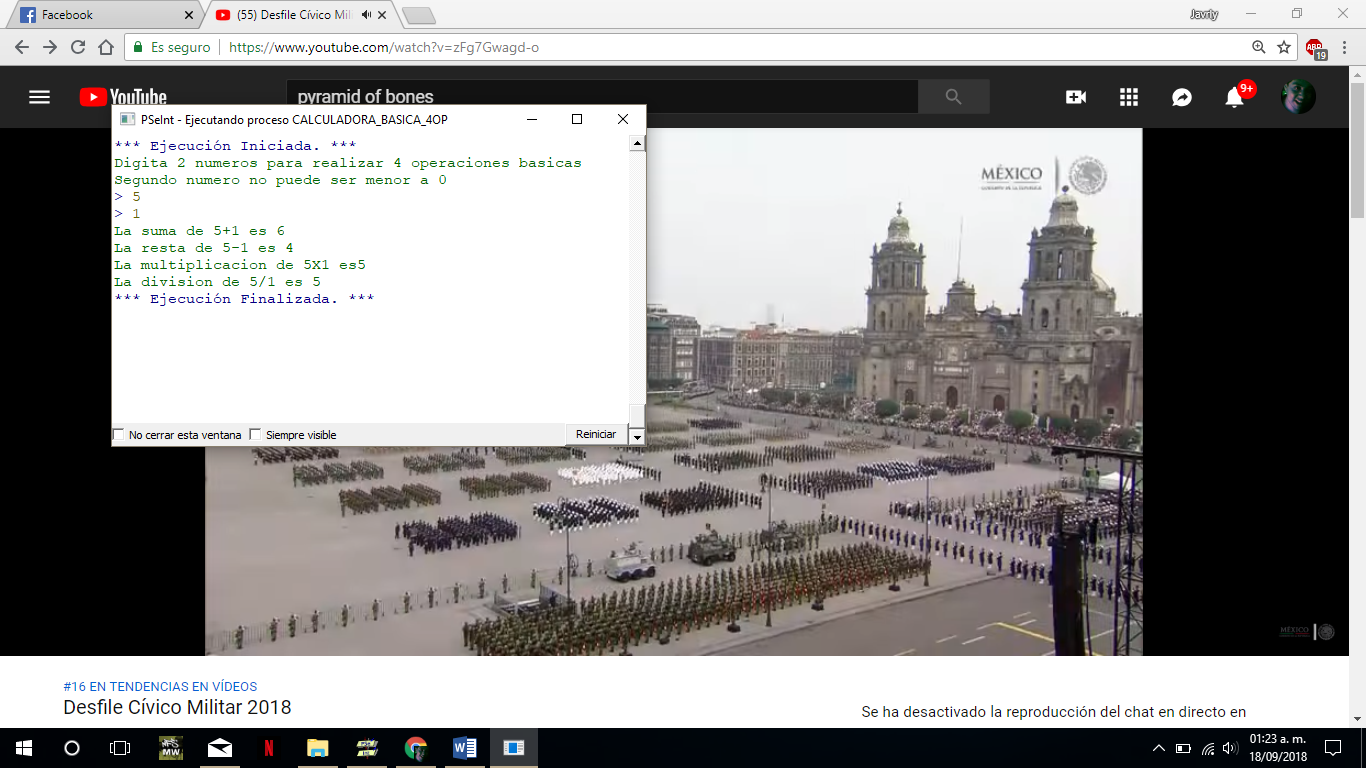
Primera iteración



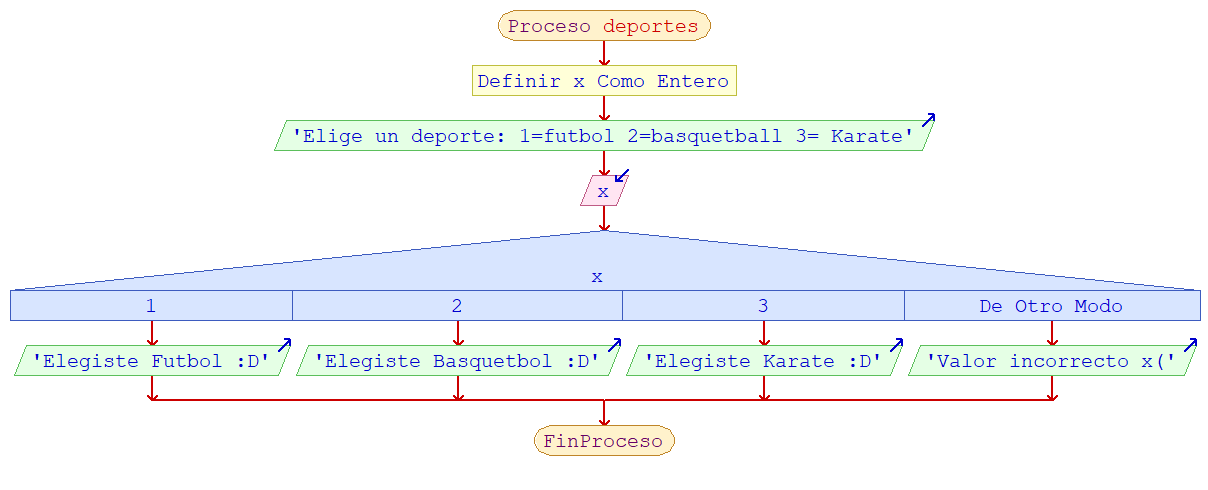
Segunda iteración



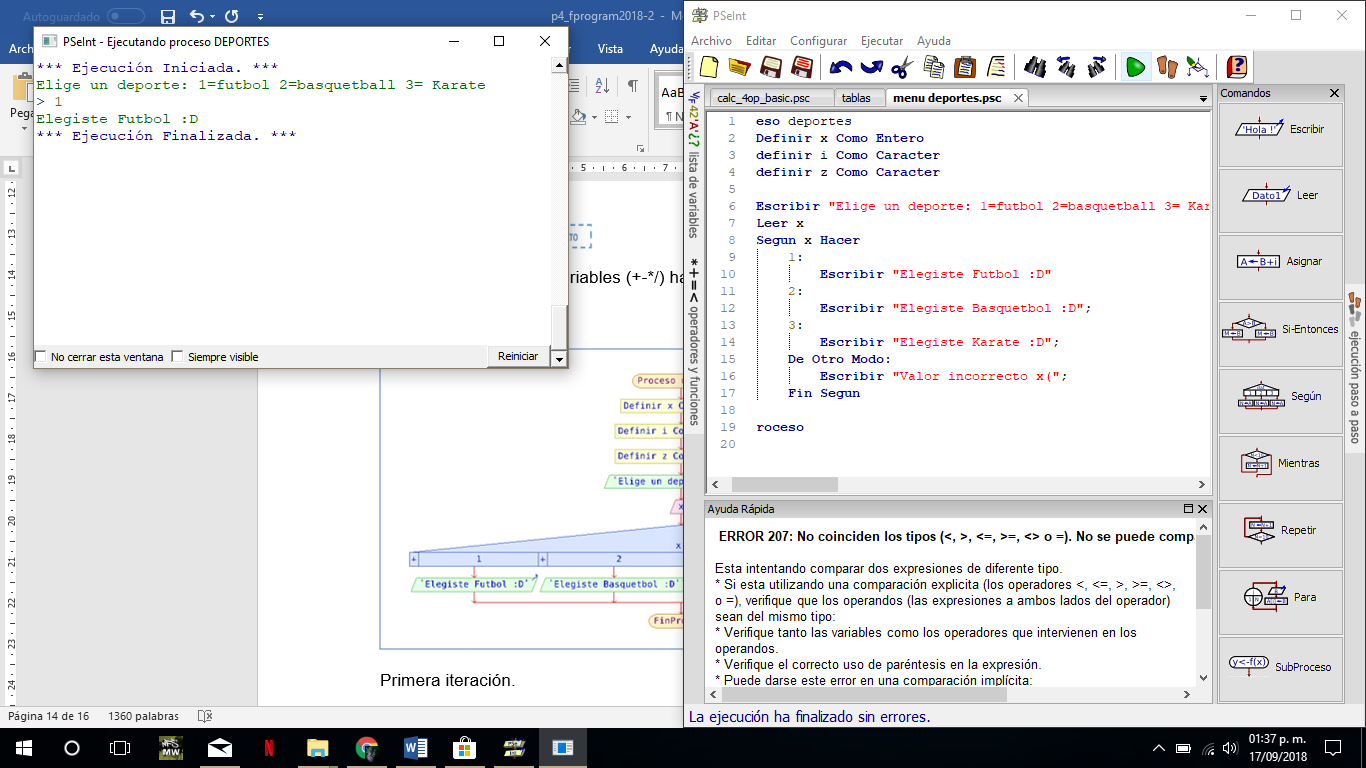
Tercera iteracia



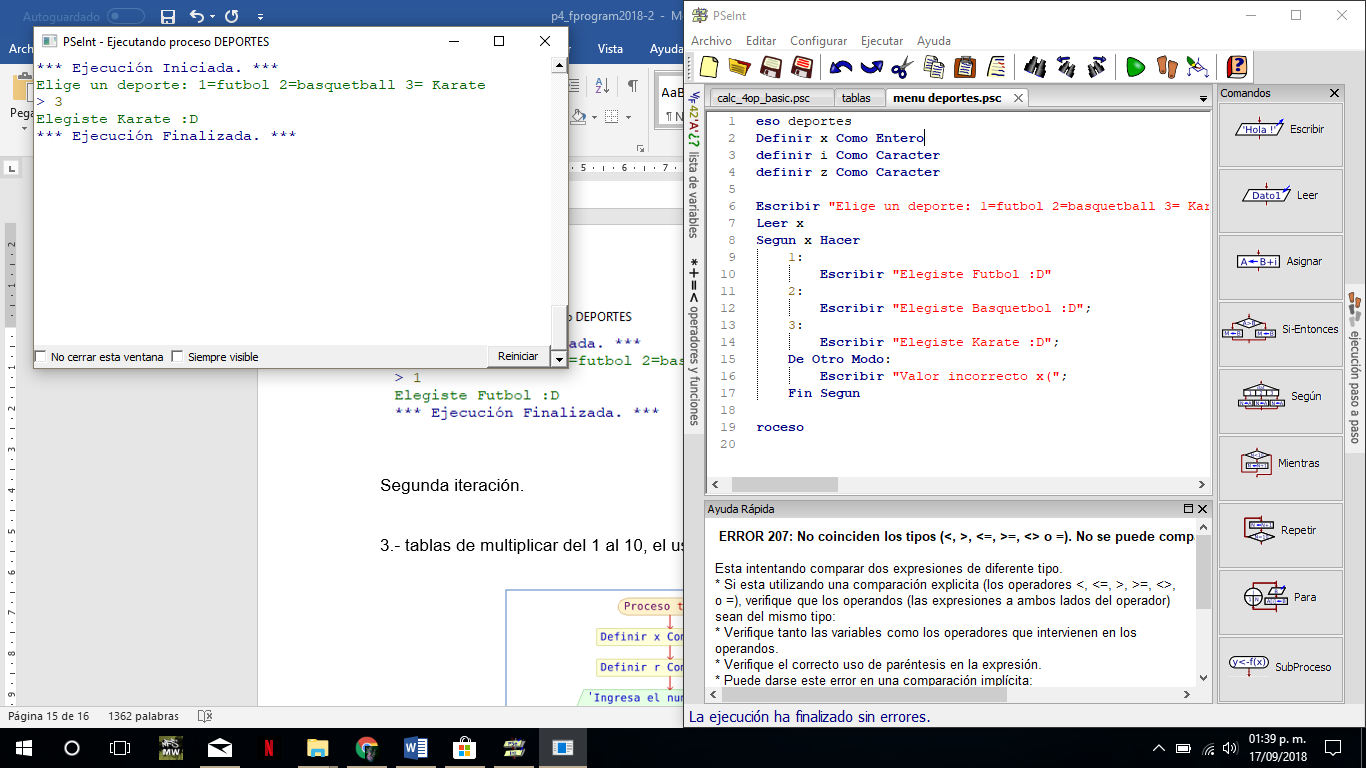
# 2.-Menù 3 deportes



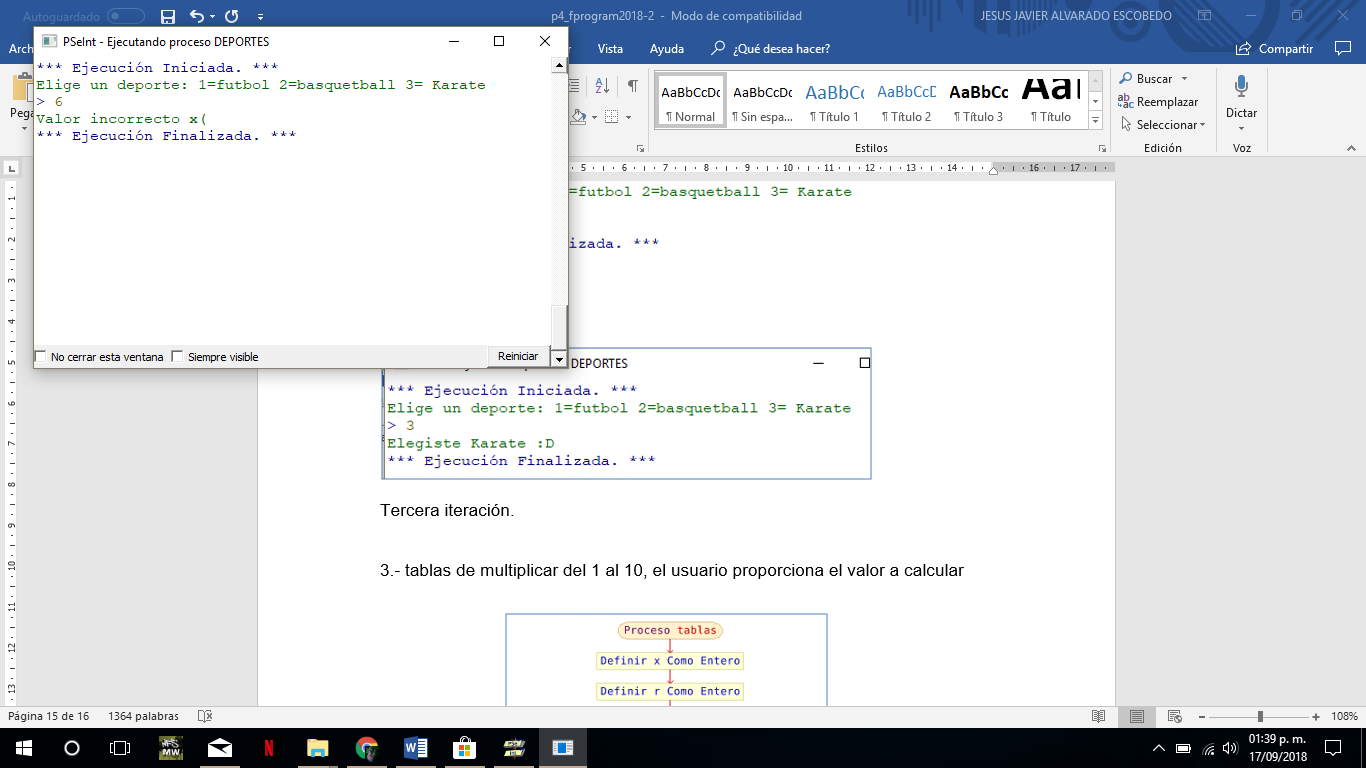
Primera iteración.



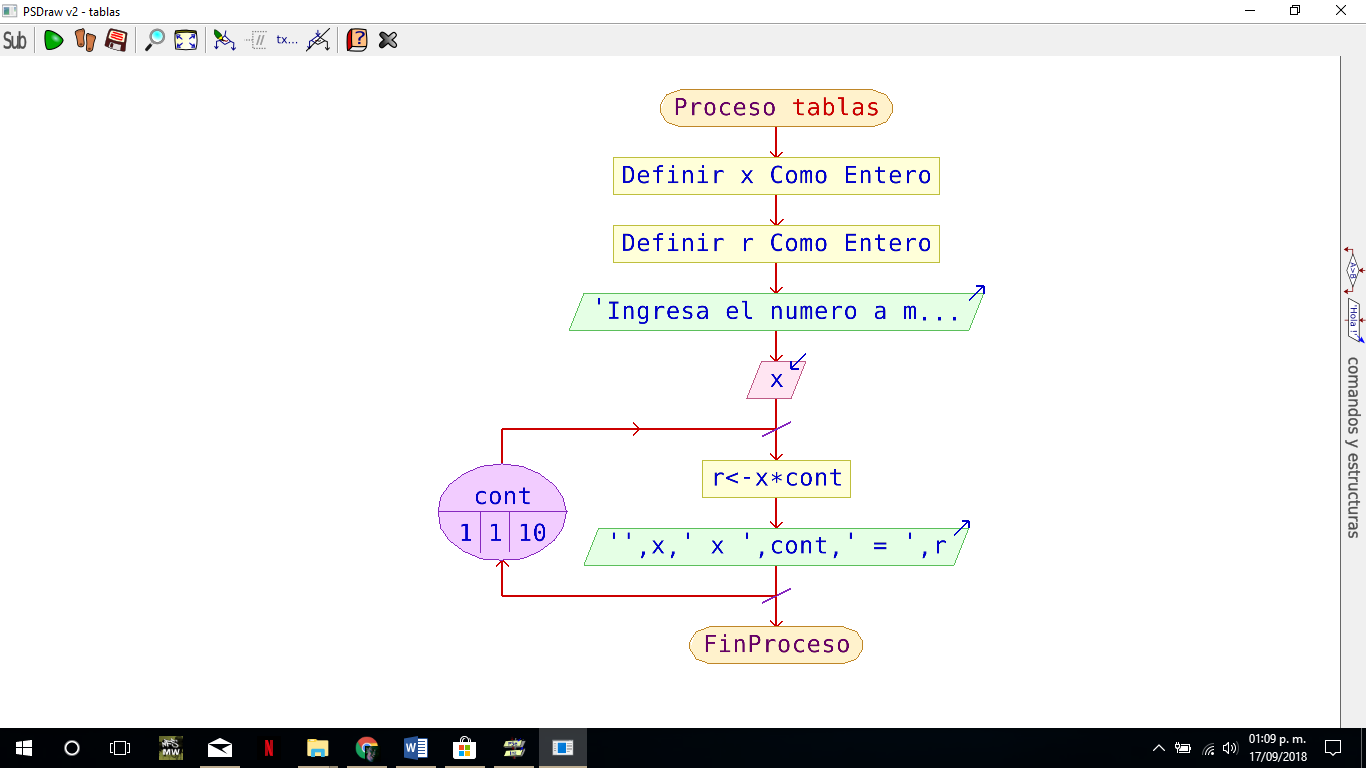
Segunda iteración.

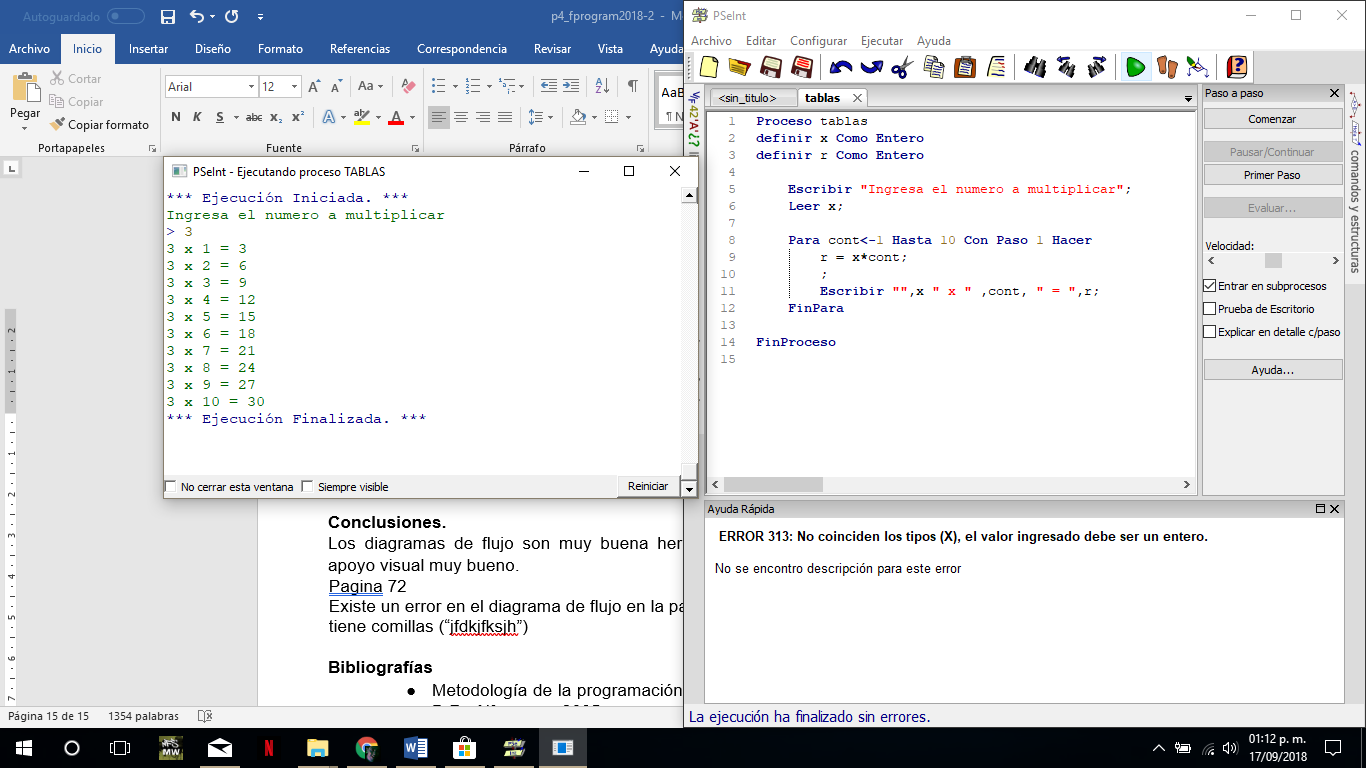


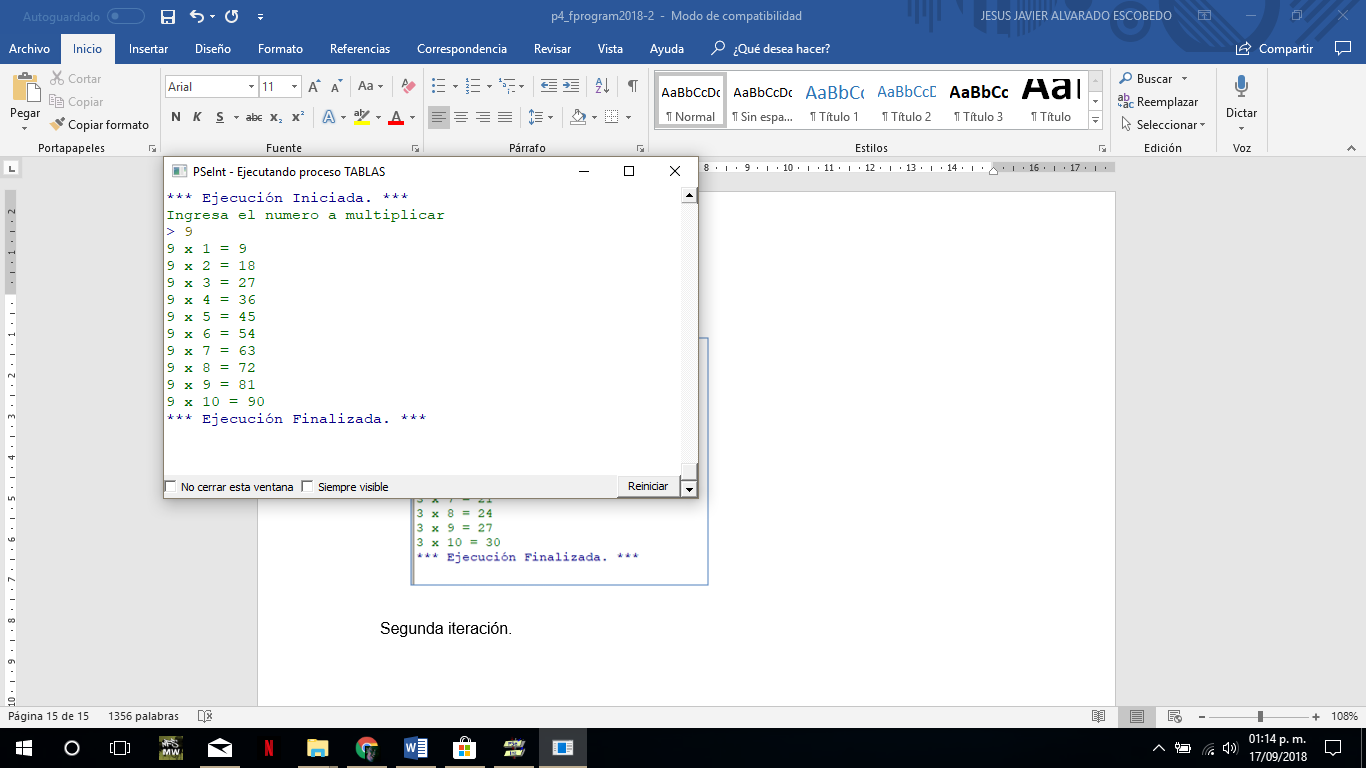
Tercera iteración.



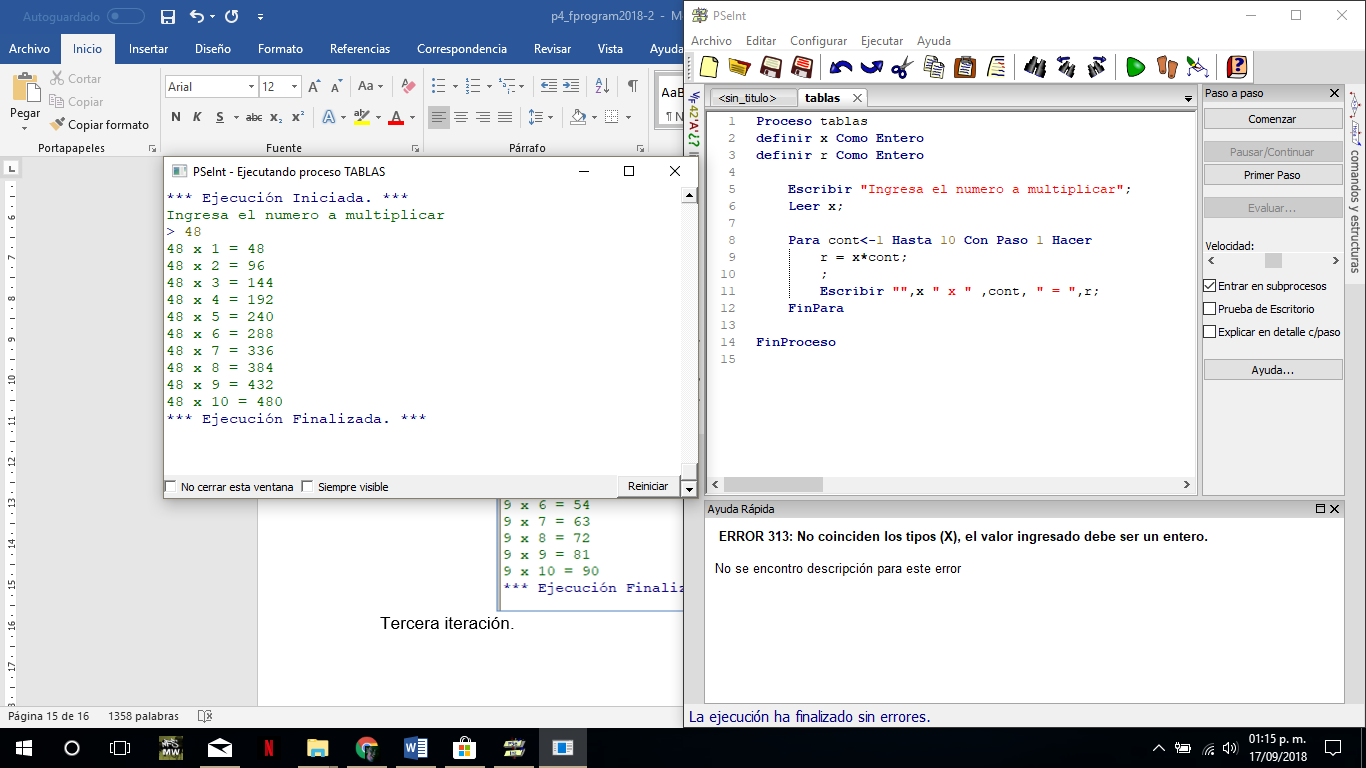
# 3.- tablas de multiplicar del 1 al 10, el usuario proporciona el valor a calcular



Primera iteración.

Segunda iteración.

Tercera iteración.



**Conclusiones.**

Los diagramas de flujo son muy buena herramienta, además de facilitar y ser un apoyo visual muy bueno.

Pagina 72

Existe un error en el diagrama de flujo en la parte de impresión de pantalla el texto no tiene comillas (“jfdkjfksjh”)

**Bibliografías**

* Metodología de la programación. Osvaldo Cairó, tercera edición, México D.F., Alfaomega 2005.
* http://depa.fquim.unam.mx/amyd/archivero/Diagramasdeflujo\_16857.pdf