|  |
| --- |
| Laboratorio de Computación  Salas A y B |
|  |

|  |  |
| --- | --- |
| Profesor: | GARCIA MORALES KARINA ING. |
| Asignatura: | FUNDAMENTOS DE PROGRAMACION |
| Grupo: | 1121 |
| No de Práctica(s): | 5 |
| Integrante(s): | JOSE DANIEL CALLEJAS SANDOVAL |
|  |  |
|  |  |
|  |  |
| No. de Equipo de cómputo empleado: | 26 |
| Semestre: | 1 |
| Fecha de entrega: | 25-9-2018 |
| Observaciones: |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Guía práctica de estudio 05: Pseudocódigo

OBJETIVOS:

Elaborar pseudocódigos que representen soluciones algorítmicas empleando la sintaxis y semántica adecuadas.

DESARROLLO:

¿Qué es un seudocódigo?

Un pseudocódigo es la representación escrita de un algoritmo, es decir, muestra en forma de texto los pasos a seguir para solucionar un problema. El pseudocódigo posee una sintaxis propia para poder realizar la representación del algoritmo (solución de un problema).

Características: Se hace uso de Sangrías, Palabras Reservadas, No hay numeración, tenemos Inicio y Fin, nos permite ver el paso a lenguaje de programación (antes de codificar)

Tipos de datos:

* ENTERO -> valor entero positivo y/o negativo
* REAL -> valor con punto flotante y signo
* BOOLEANO -> valor de dos estados: verdadero o falso
* CARACTER -> valor tipo carácter
* CADENA -> cadena de caracteres (Arreglos)

A: =b+c Igualdad

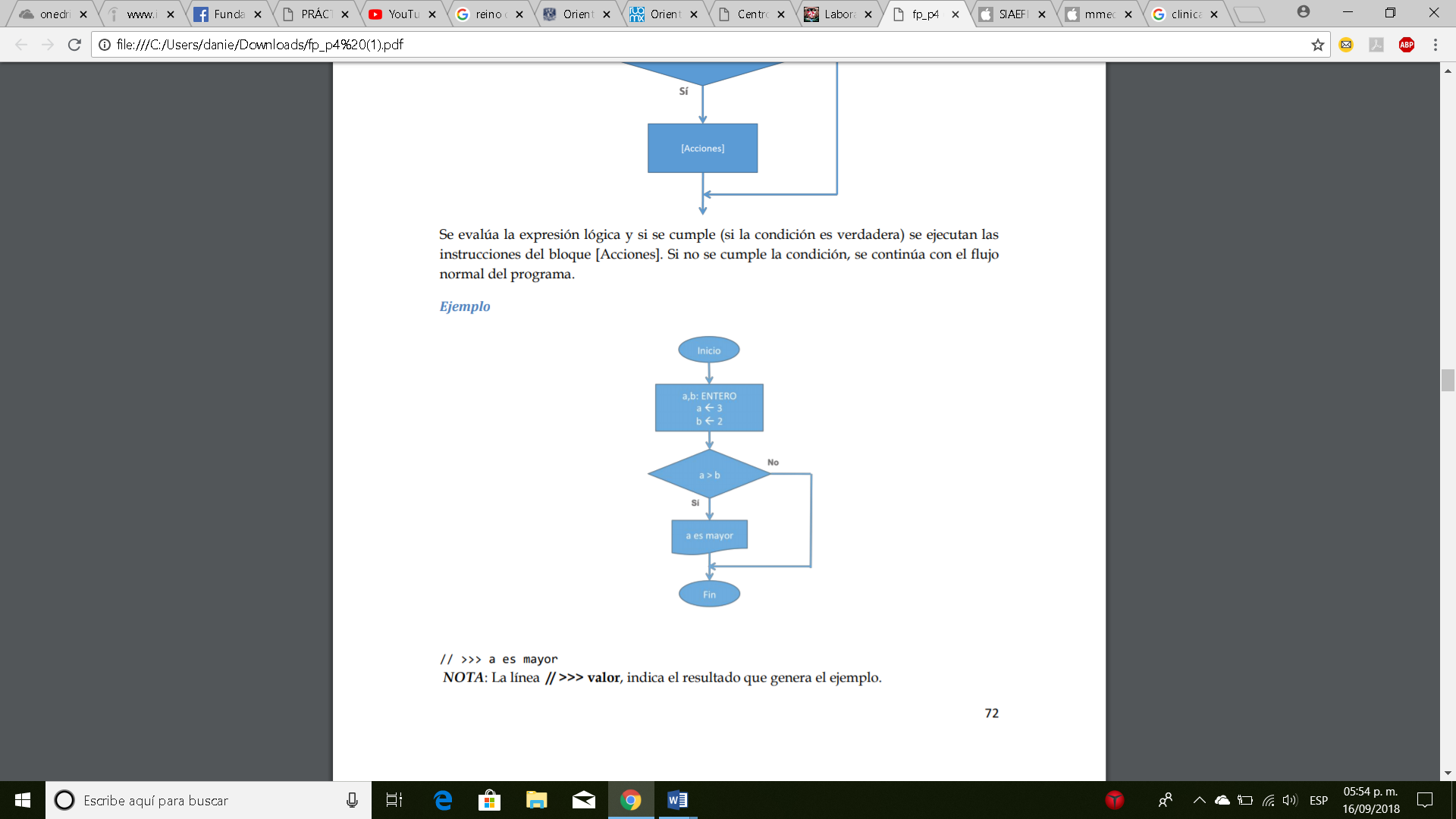
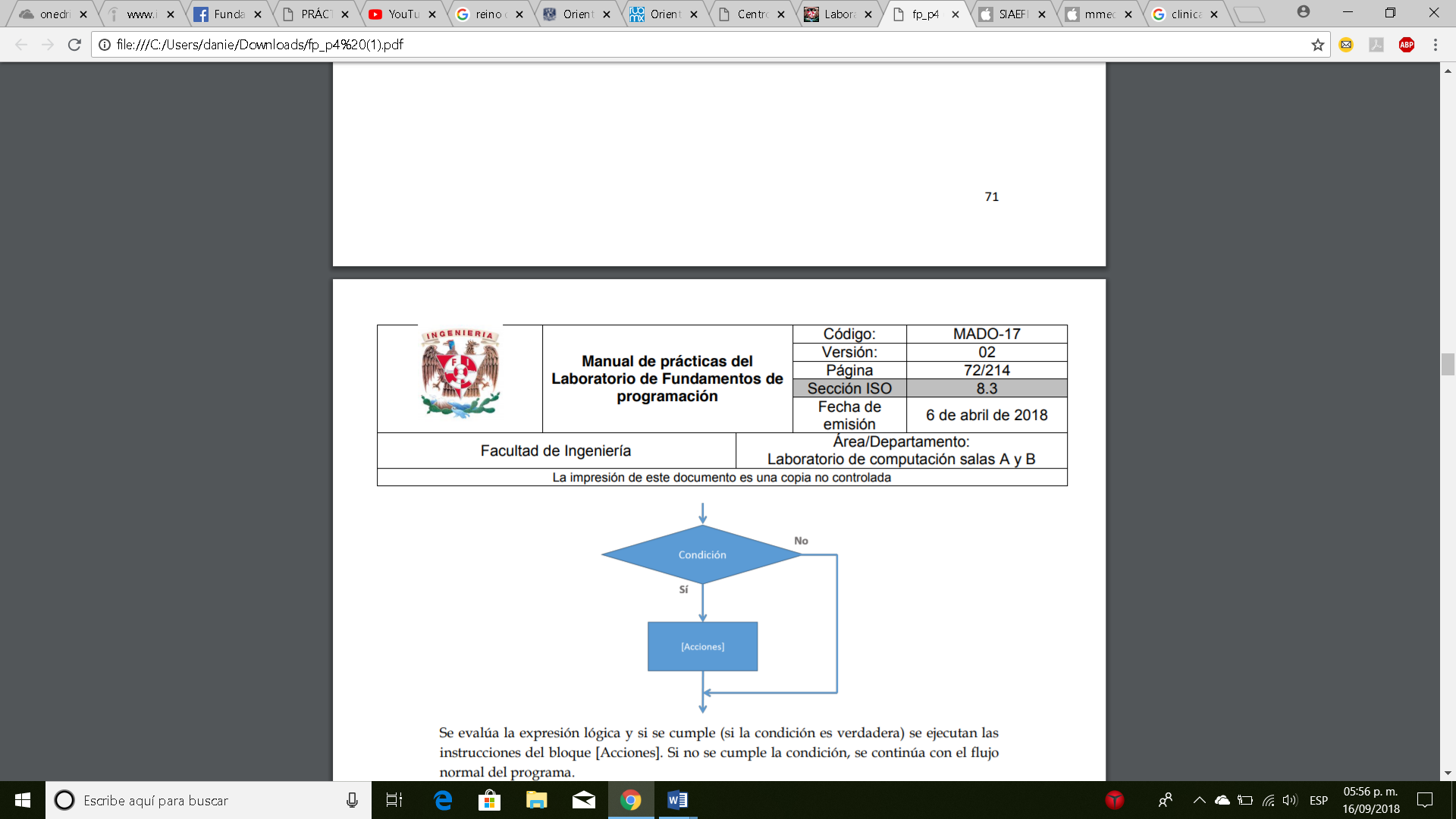
A<>b Distinto de

NOTACION DE CAMELLO: promInicial o promFinal. Se utiliza por buenas practicas. De minúscula a mayúscula.

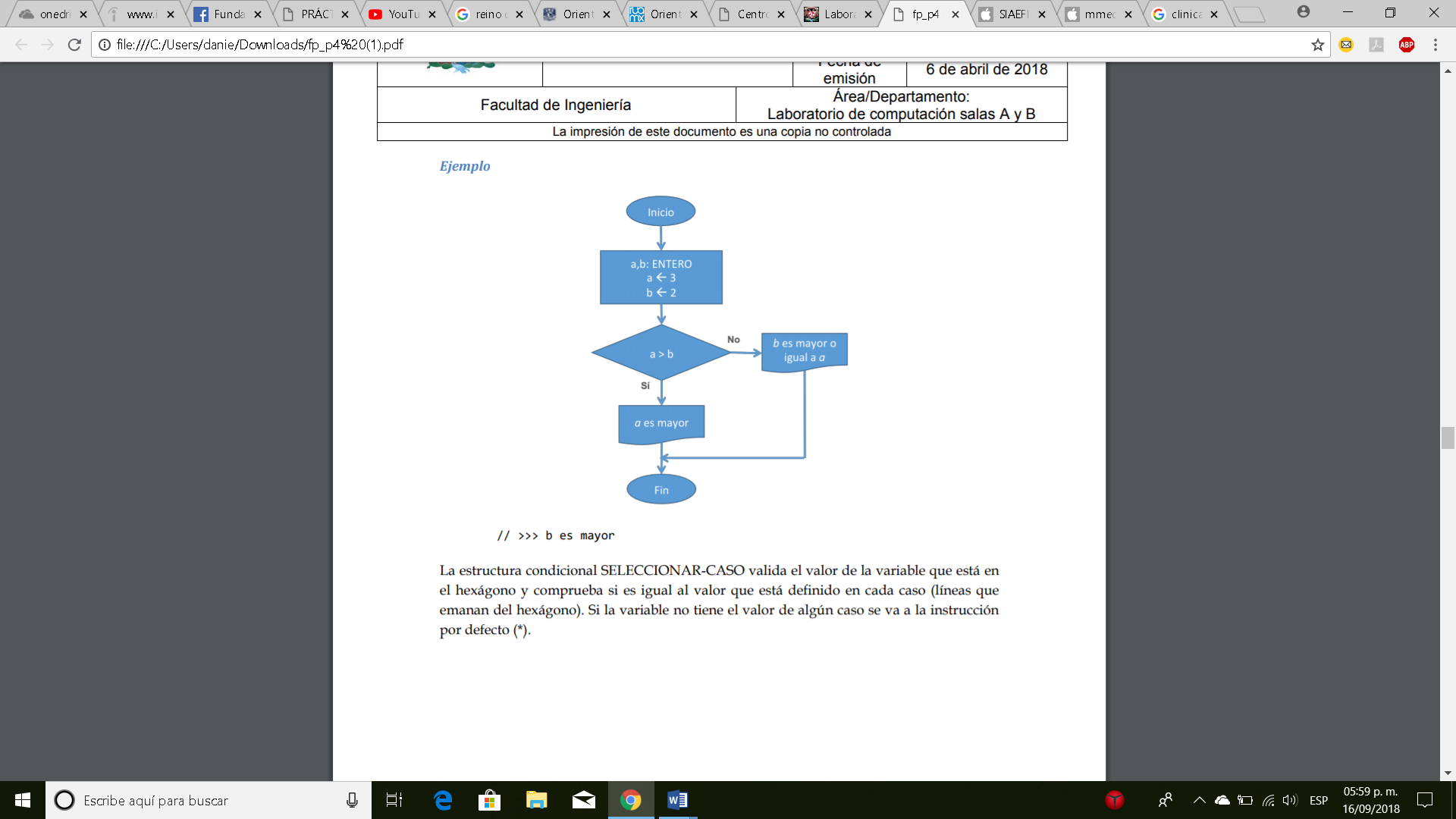
Estructura Secuencial: Inicializar una variable, declarar una variable. Se utilizan las palabras reservadas, LEER (Declarar variables) y ESCRIBIR (Mostrar texto)

Estructura Condicional: Toma de decisiones, si cumple una condición, realiza unas instrucciones.

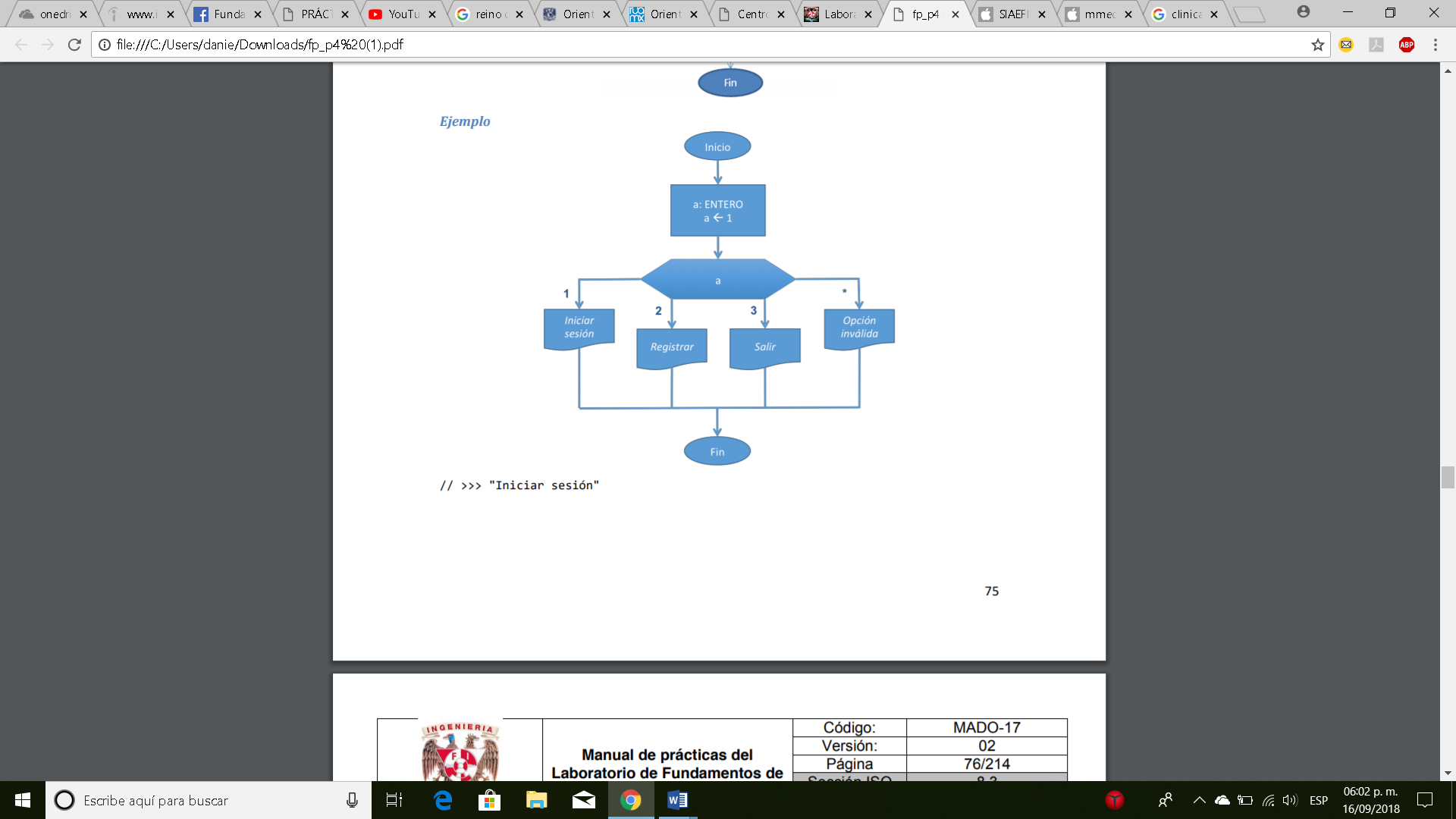
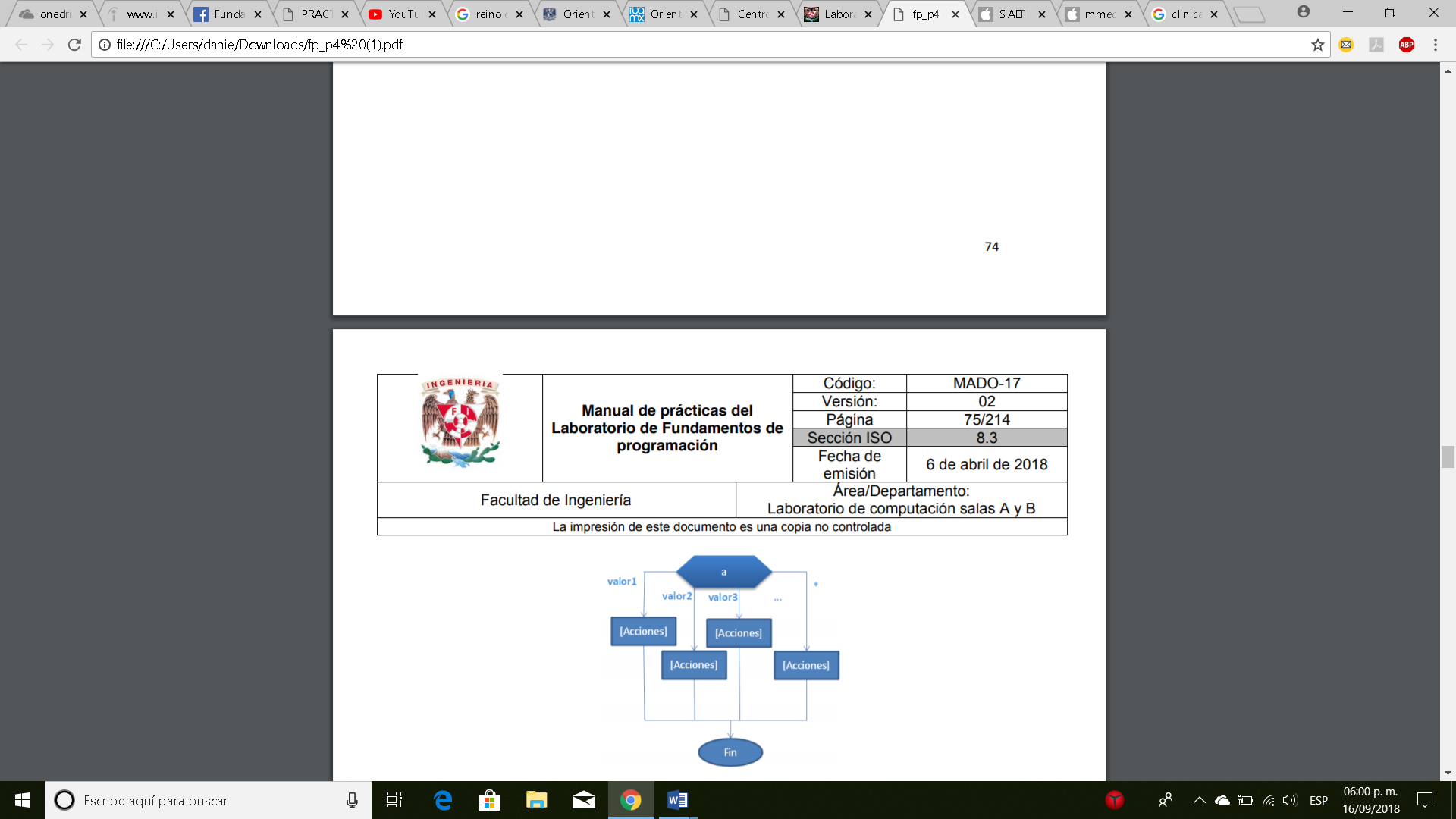
* Condicional Simple: Si cumple una condición, instrucciones, y si no, va a Fin.



* Condicional Compuesta: Si cumple una condición, realiza una instrucción, y si no realiza otra instrucción y va a fin.



* Condicional Múltiple: Seleccionar entre distintas alternativas.



EJEMPLO:

Algoritmo SABORES

    Escribir "DAME UN SABOR,1-CHOCOLATE,2-FRESA,3-LIMON"

    Leer X

    Segun X Hacer

        1:

            Escribir "CHOCOLATE"

        2:

            Escribir "FRESA"

        3:

            Escribir "LIMON"

        De Otro Modo:

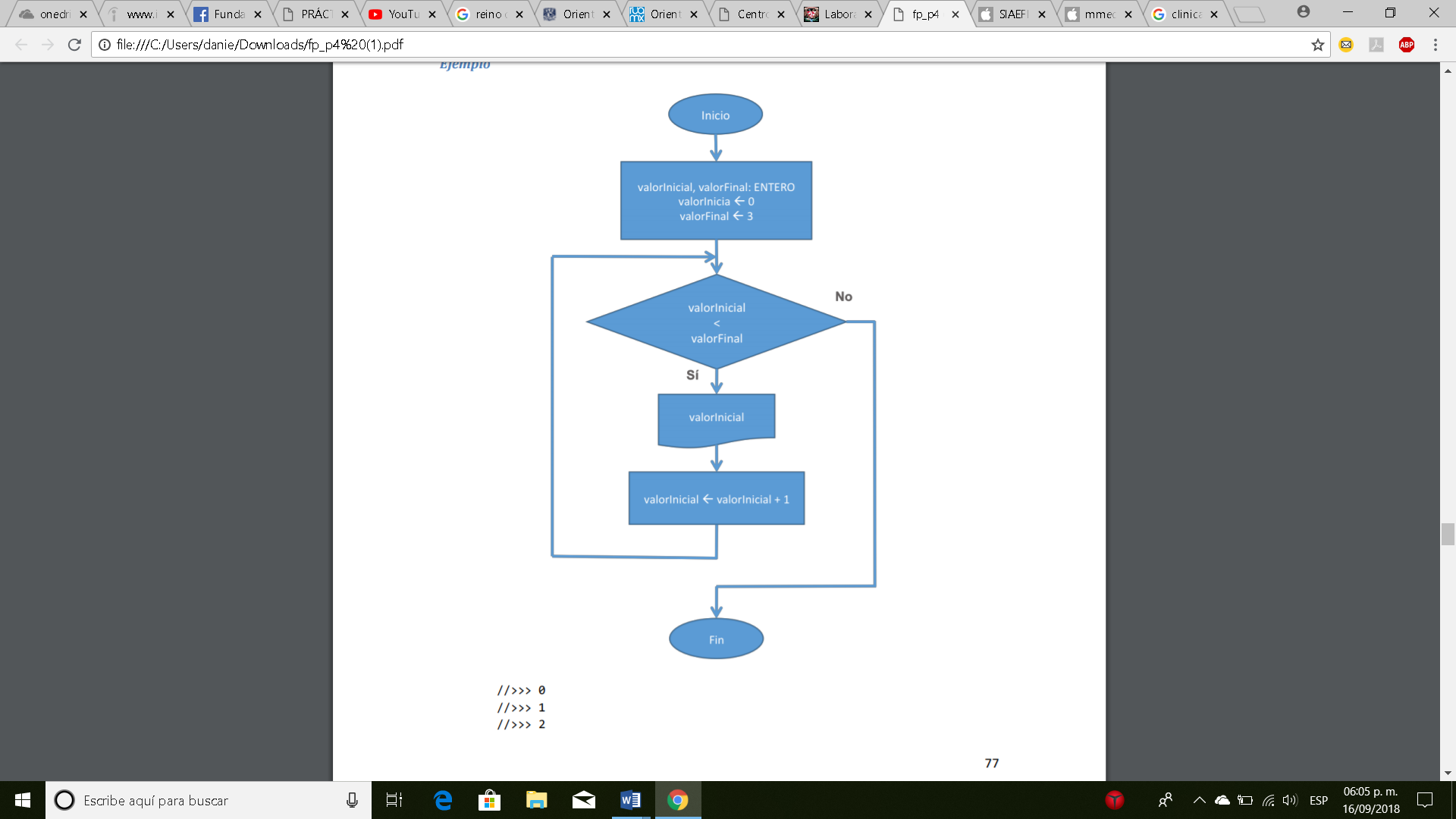
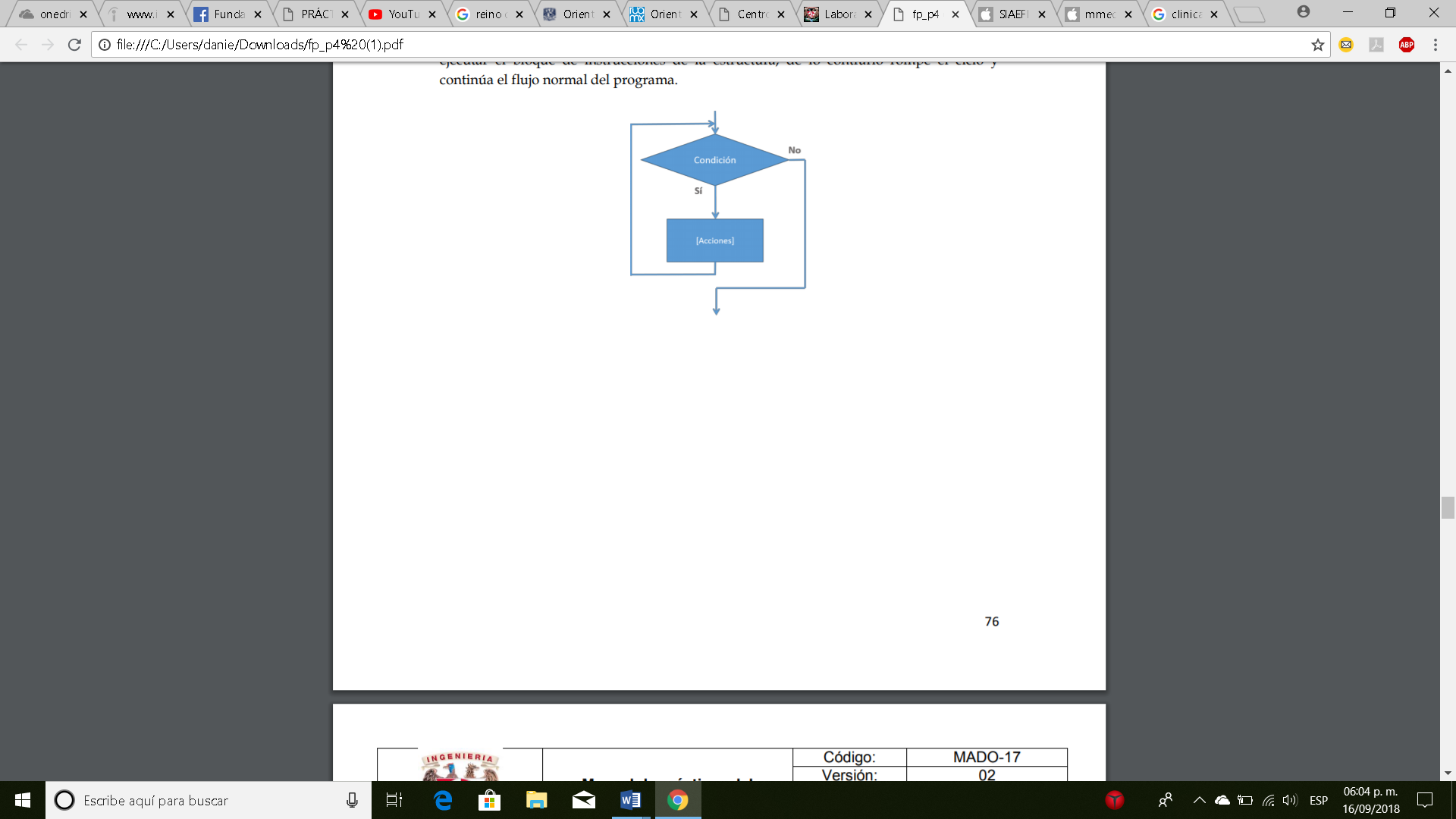
            Escribir "ERROR"

    FinSegun

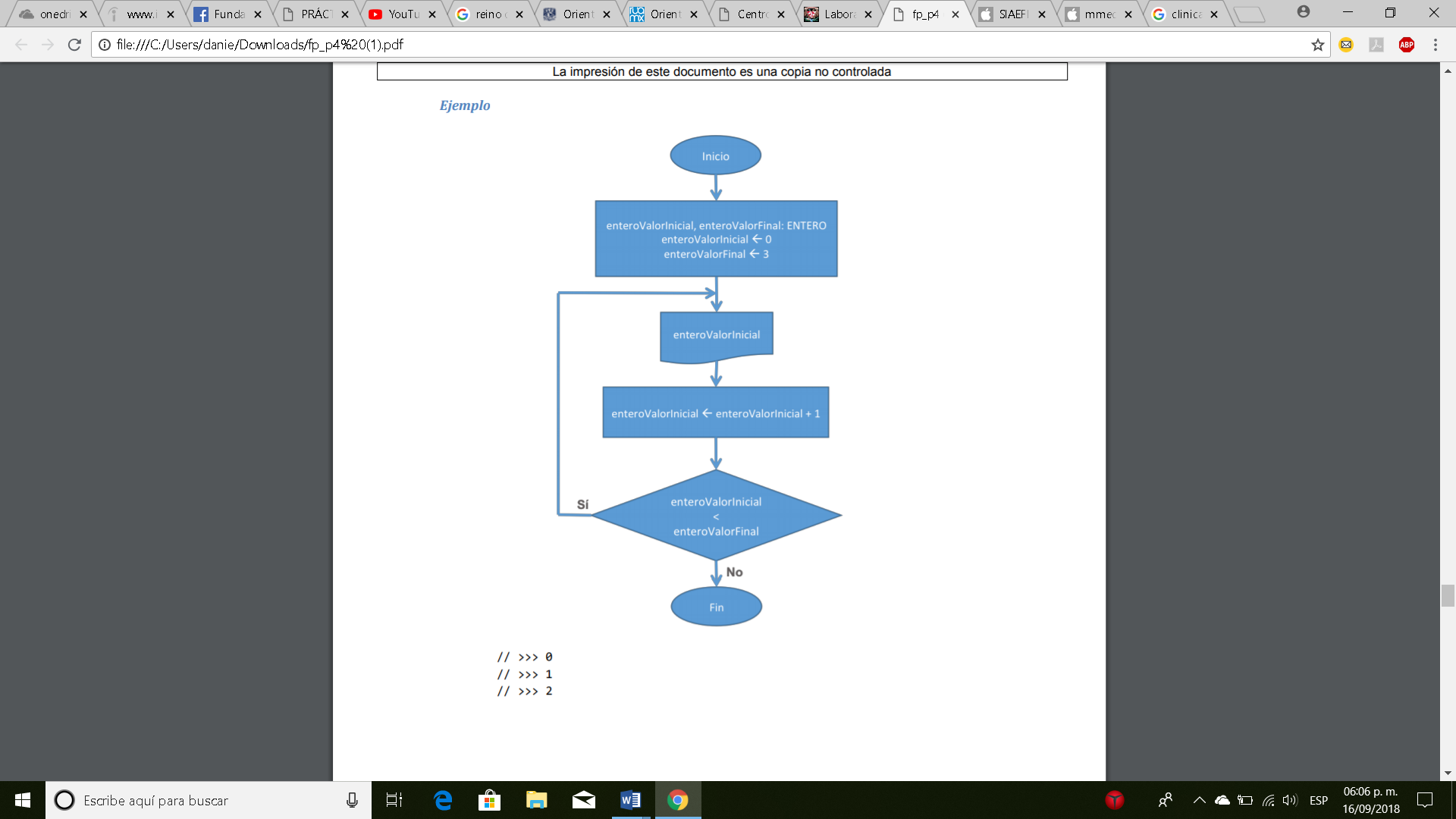
FinAlgoritmo

Estructura Iterativa: Es un ciclo, n cantidad de veces.

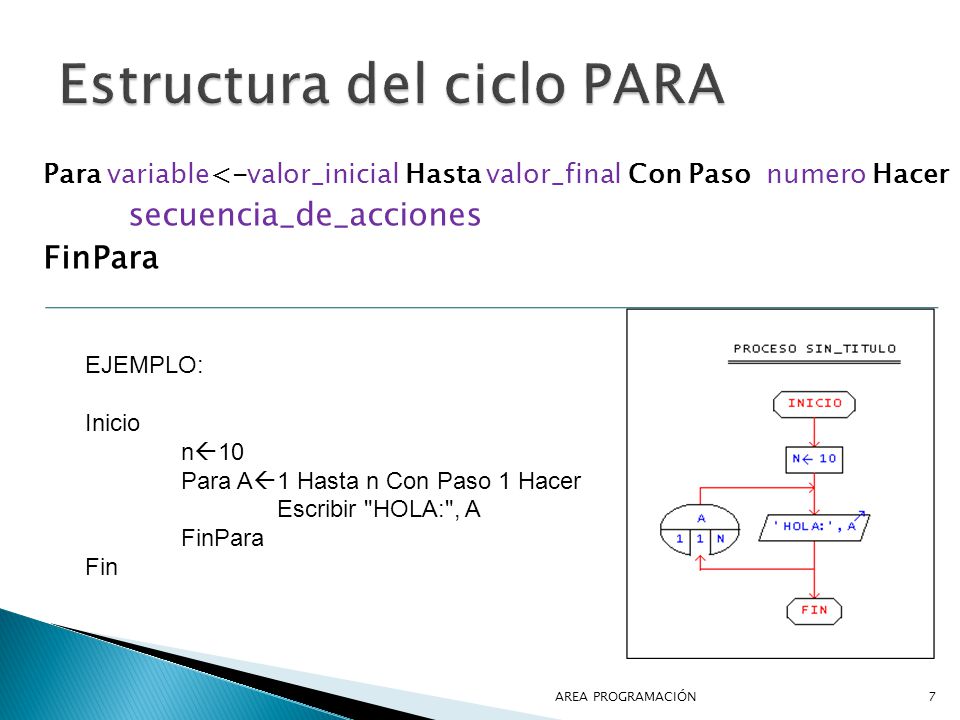
* MIENTRAS: Condición y después instrucción.



* HACER-MIENTRAS: Instrucción y después condición. Se ejecutará por lo menos 1 sola vez.

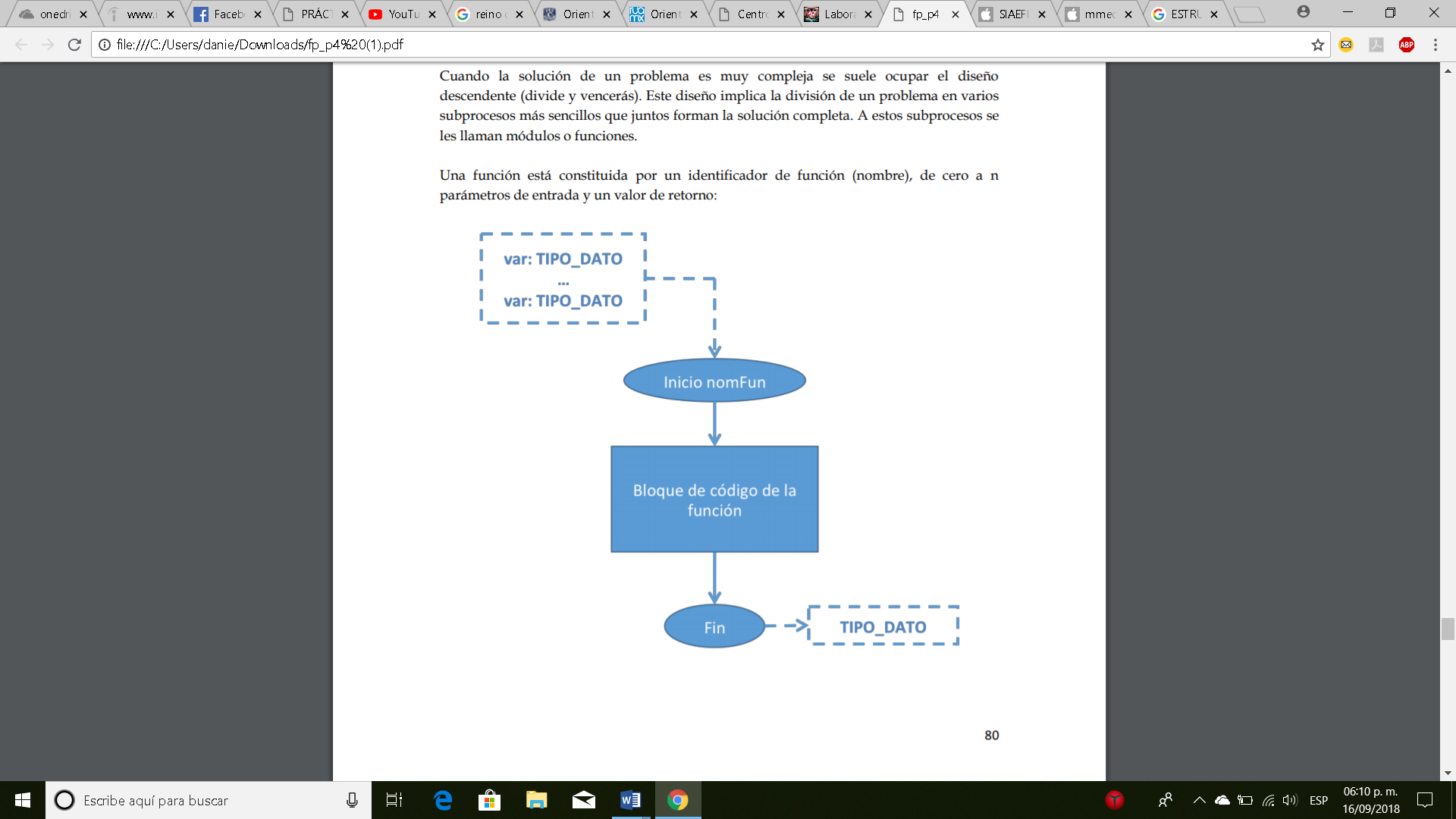


* PARA: Valor inicial, Condición, e Incrementos o Decrementos.

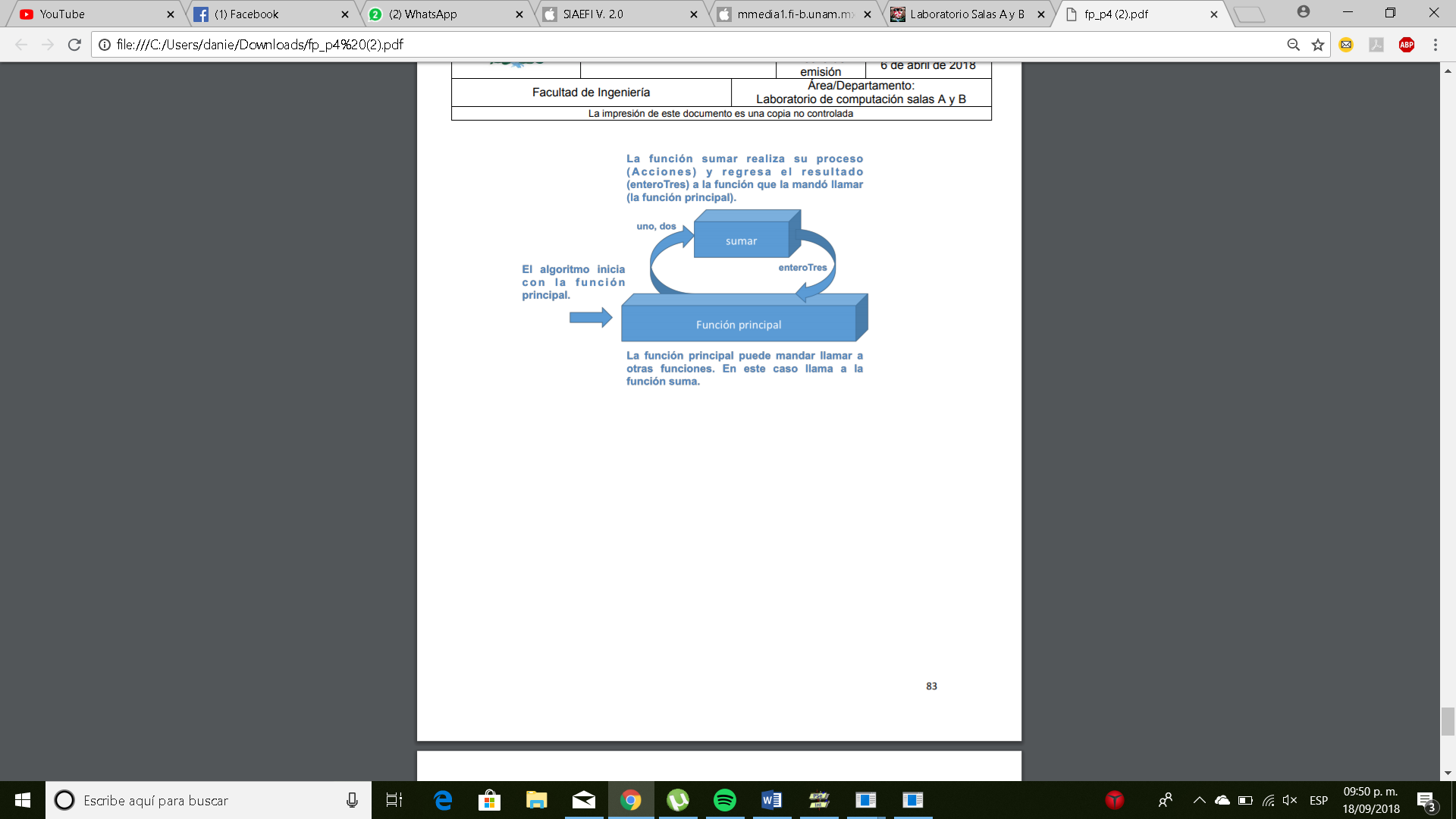
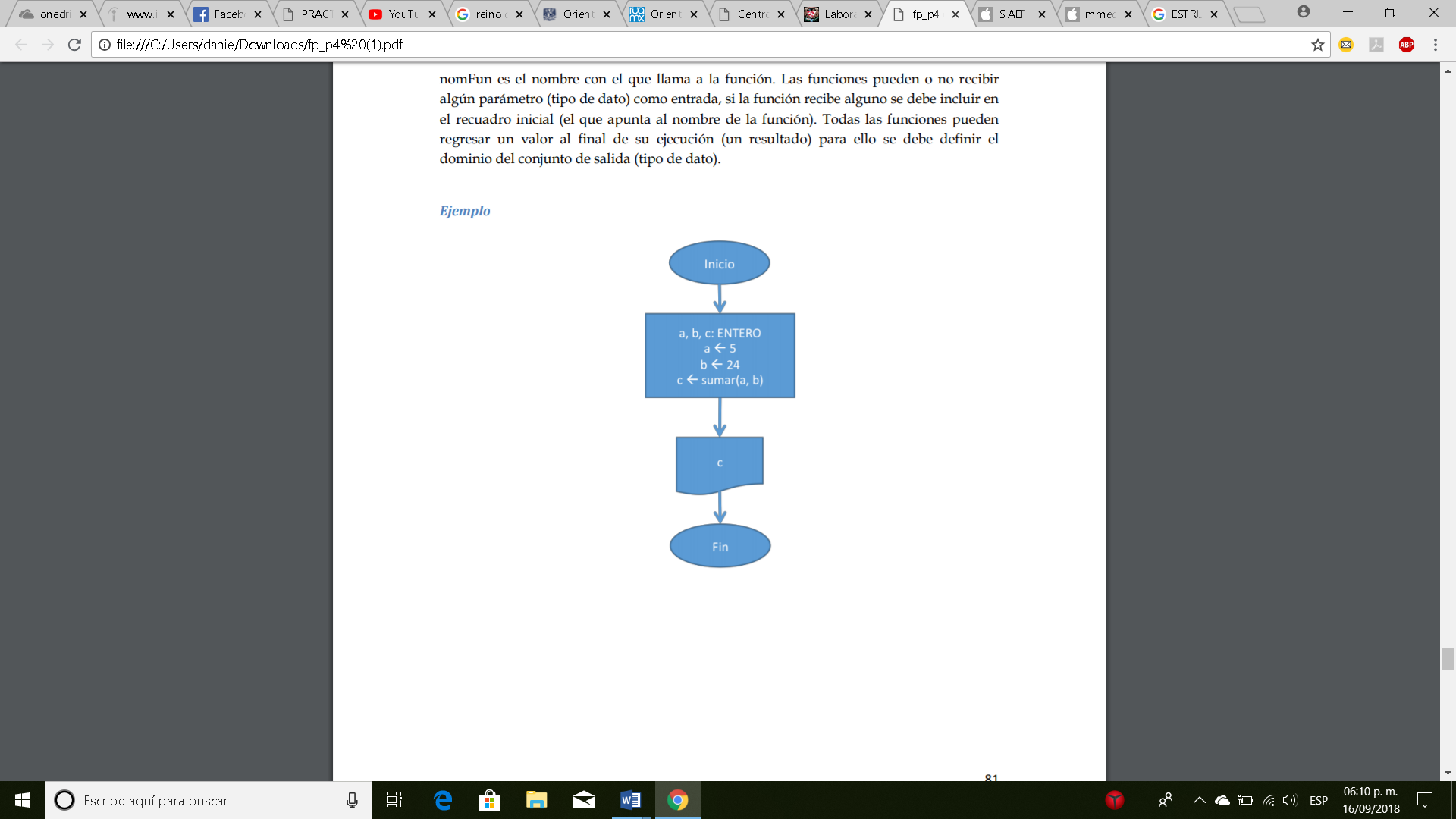


FUNCIONES

Se tiene inicio, proceso y fin (igual que cualquier diagrama), aunque muestra las variables antes de iniciar. 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).



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.



ESTRUCTURAS:

Facultad: REG

carreraAlumno:CADENA

numeroCuenta:ENTERO

nombreCompleto: CADENA

FIN REG

Alu: REGfacultad

AluCiencias: REGfacultad

AluIng: REGfacultad

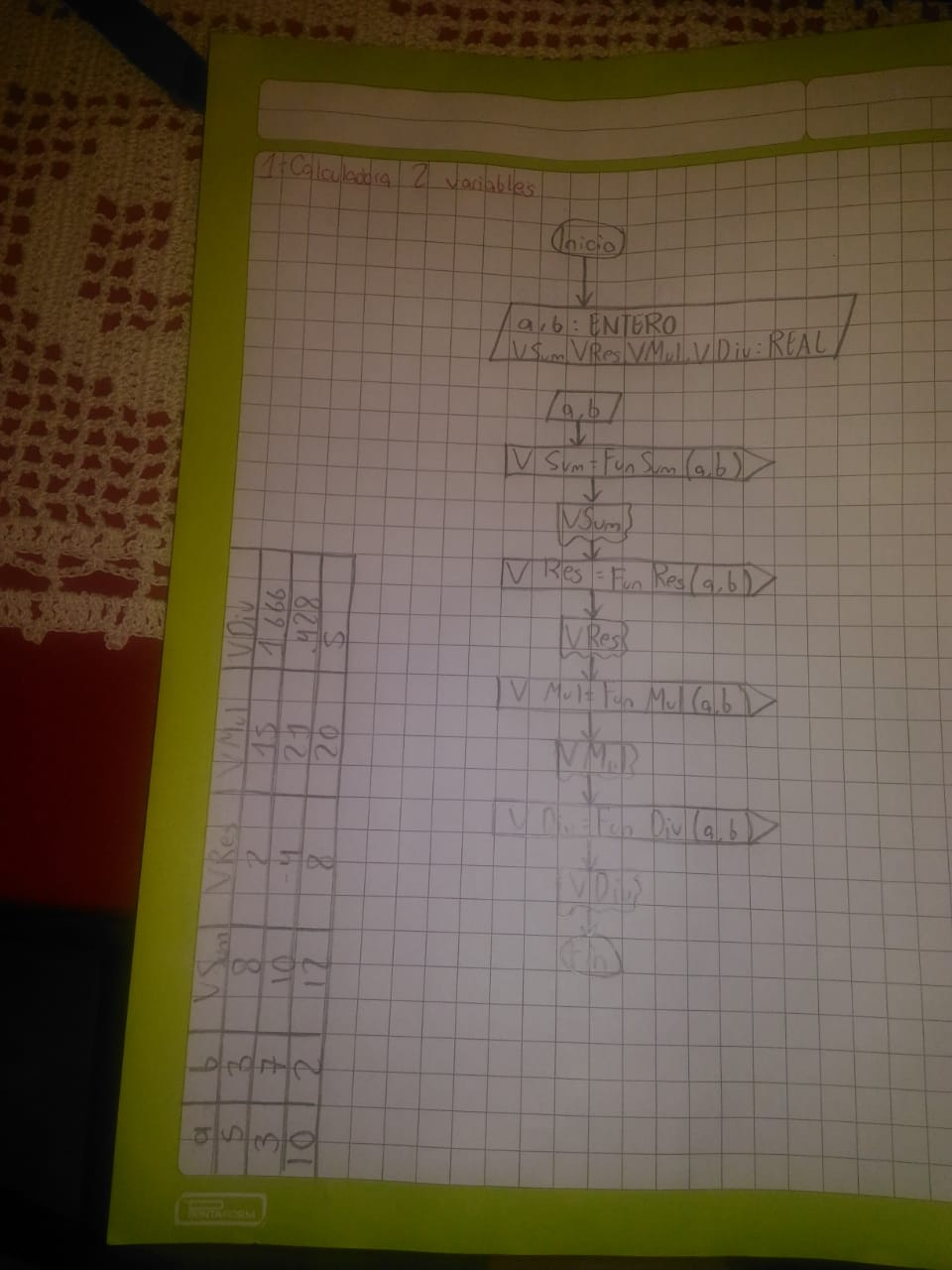
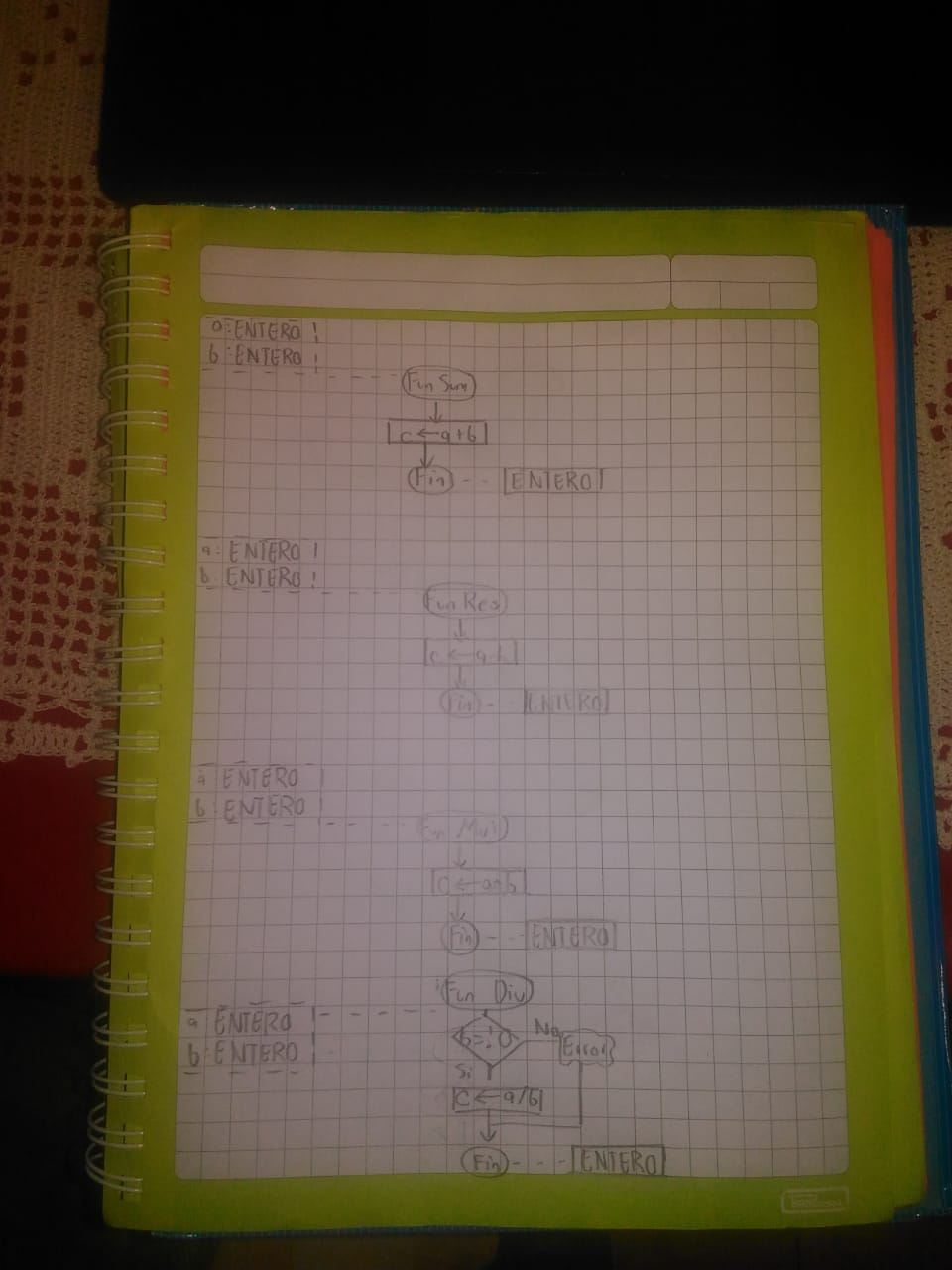
REGfacultad.carAlumno: =” Industrial”

Alu.carAlumno: =” Industrial”

AluCiencias

EJERCICIOS DE TAREA:

1. - CALCULADORA



PSEUDOCODIGO:

Inicio

FUNC Sum (a: ENTERO; b: ENTERO) RET:ENTERO

enteroTres: Entero

enteroTres:= a+b

RET enteroTres

Fin FUNC Sum

FUNC Res (a: ENTERO; b: ENTERO) RET:ENTERO

enteroTres: Entero

enteroTres:= a-b

RET enteroTres

Fin FUNC Res

FUNC Mult (a: ENTERO; b: ENTERO) RET:ENTERO

enteroTres: Entero

enteroTres:= a\*b

RET enteroTres

Fin FUNC Mult

FUNC Div (a: ENTERO; b: ENTERO) RET:ENTERO

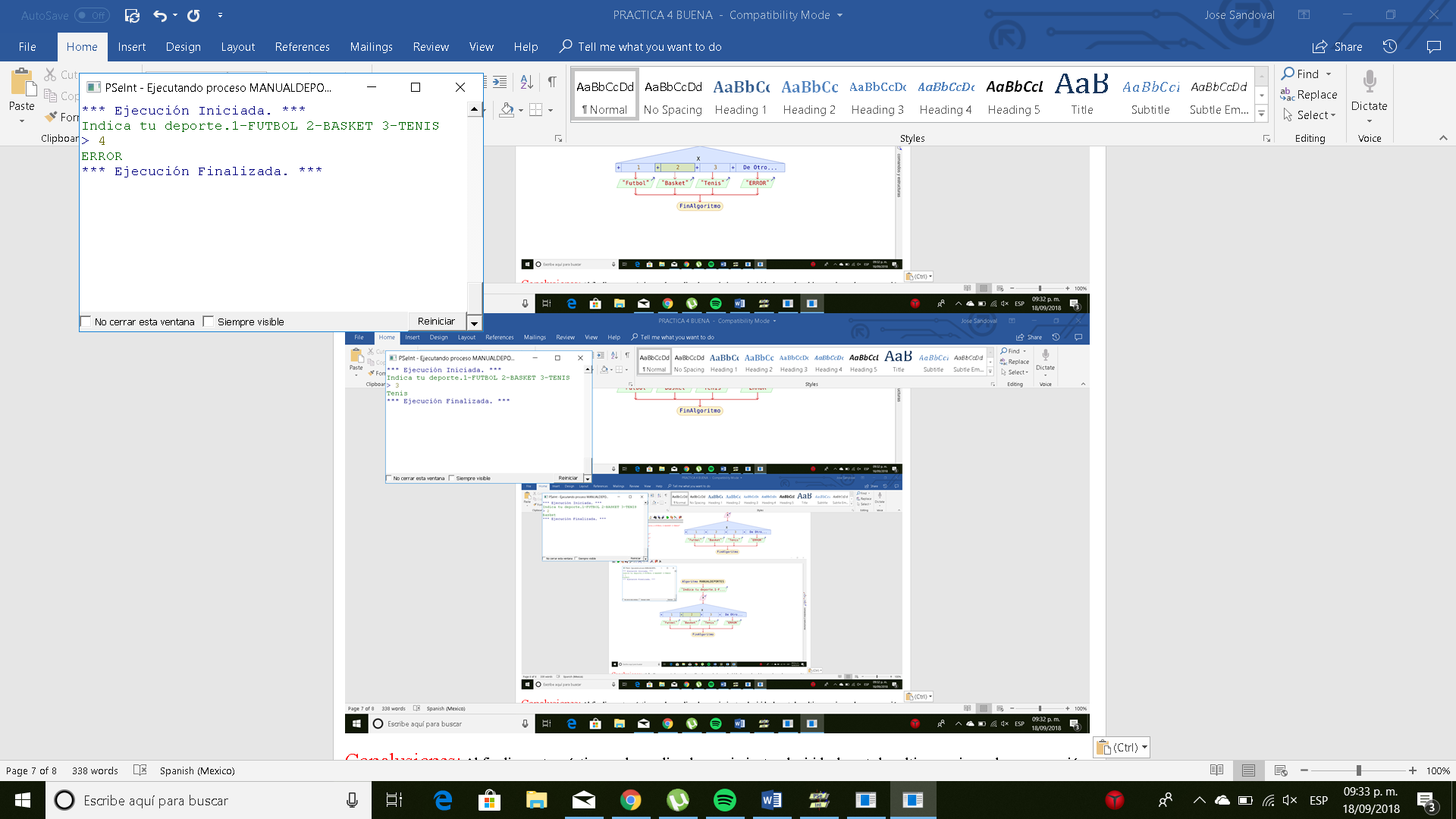
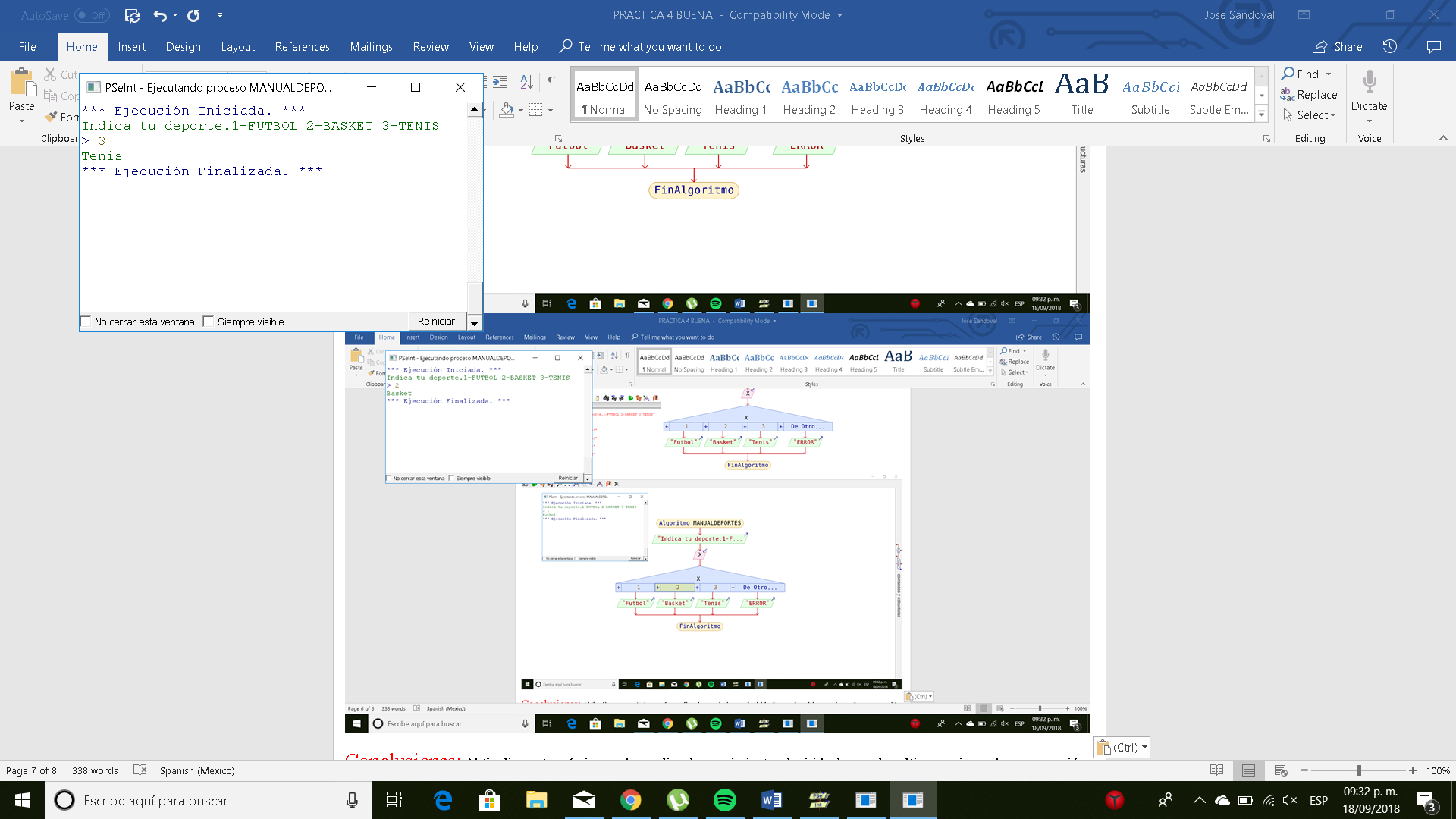
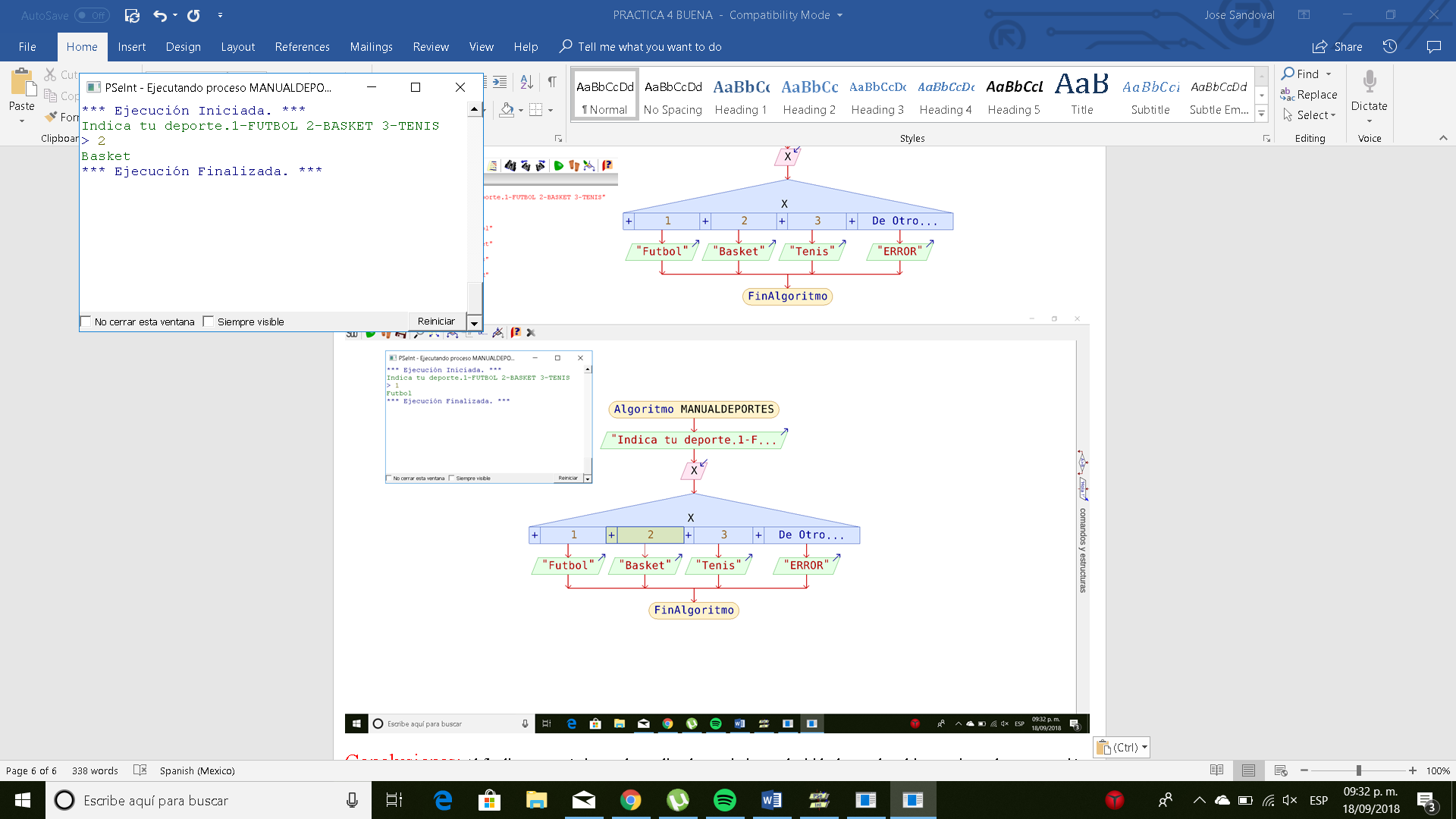
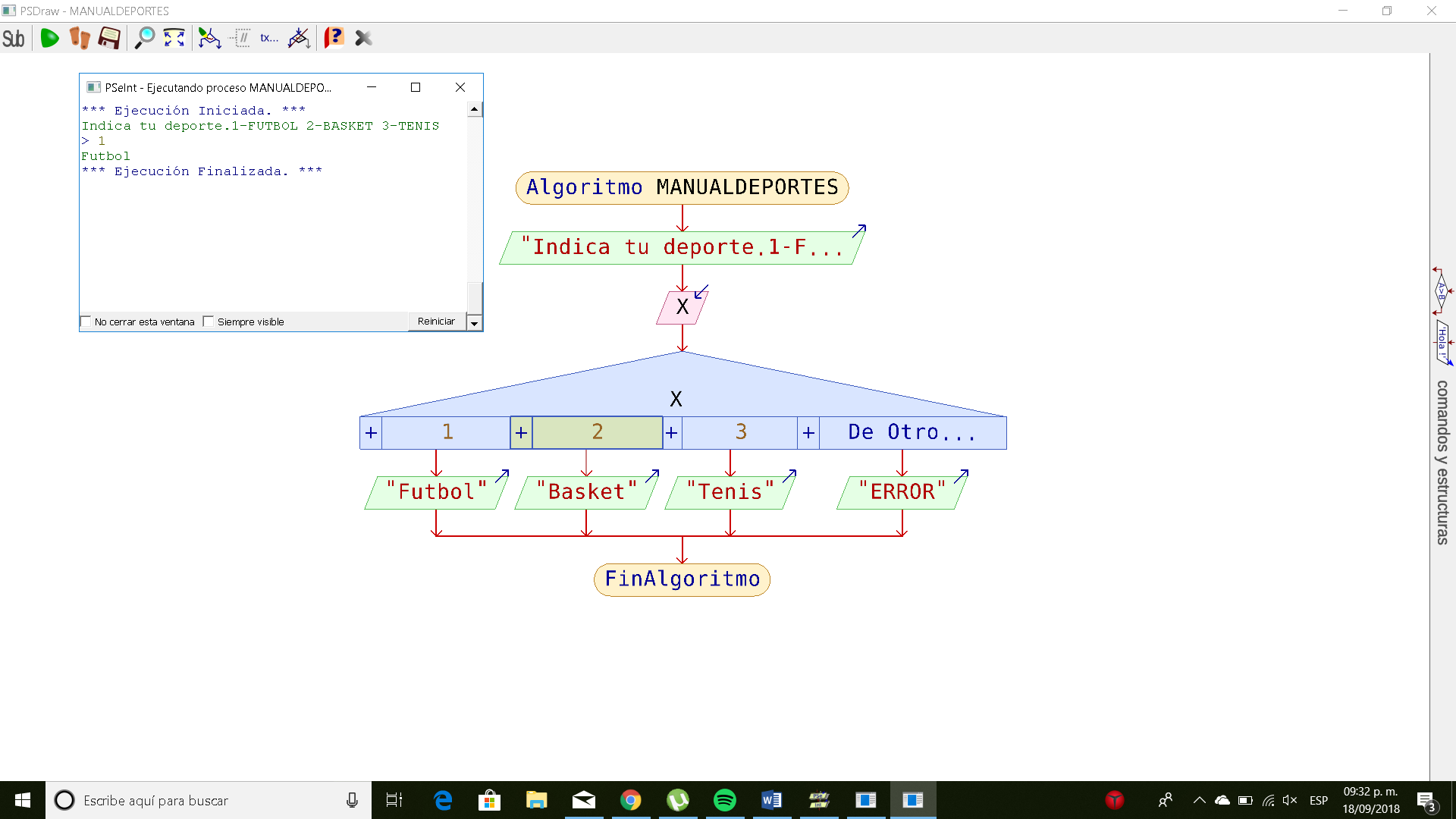
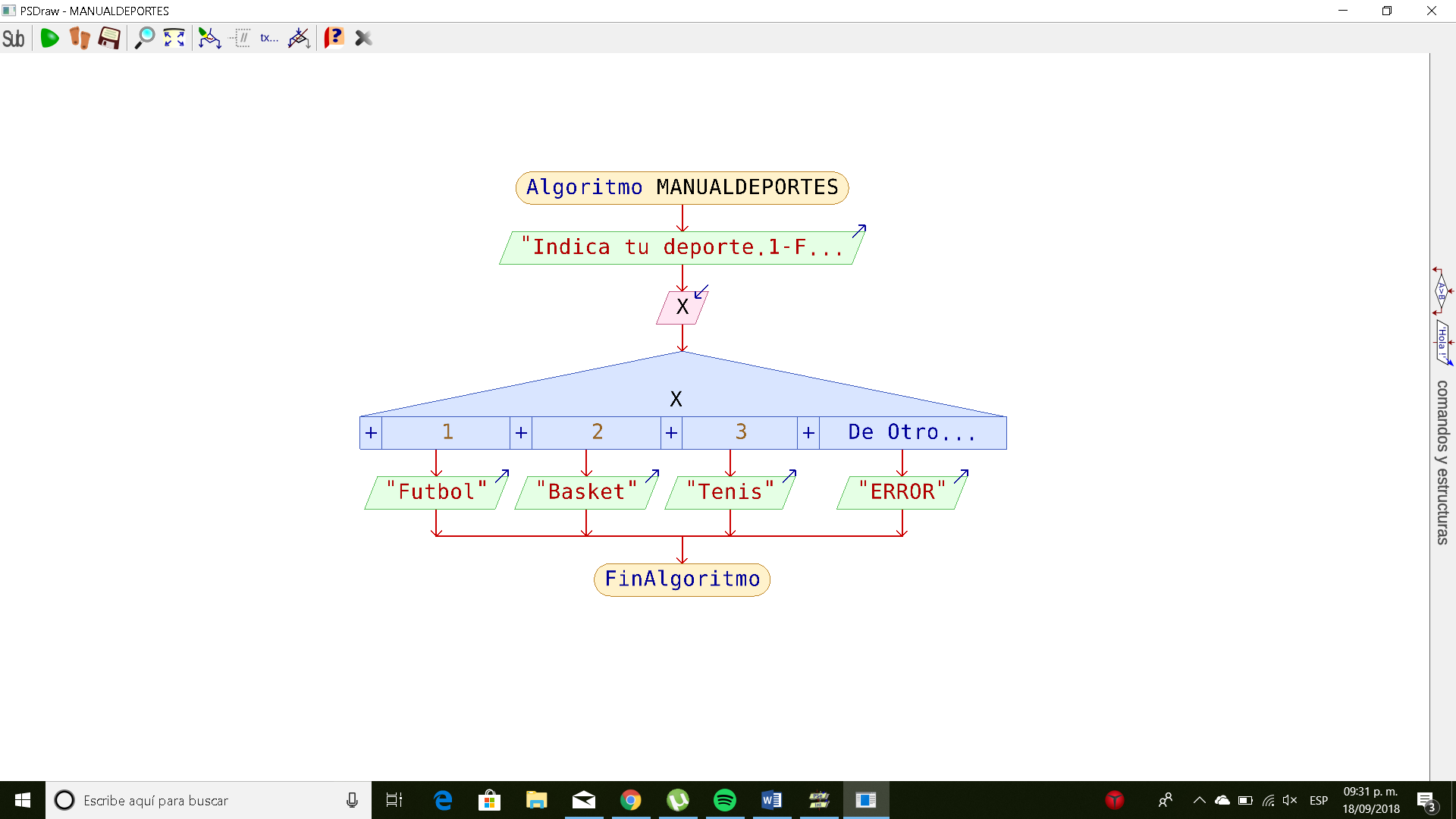
enteroTres: Entero

enteroTres:= a/b

RET enteroTres

Fin FUNC Div

2.-MANUAL DE DEPORTES



PSEUDOCODIGO:

Inicio

D: ENTERO

IMPRIMIR “DEPORTES”

IMPRIMIR “1-FUTBOL”

IMPRIMIR “2-NATACION”

IMPRIMIR “3-GOLF”

Leer D

Inicio de opción múltiple

Si D=1

IMPRIMIR “FUTBOL”

Si D=2

IMPRIMIR “NATACION”

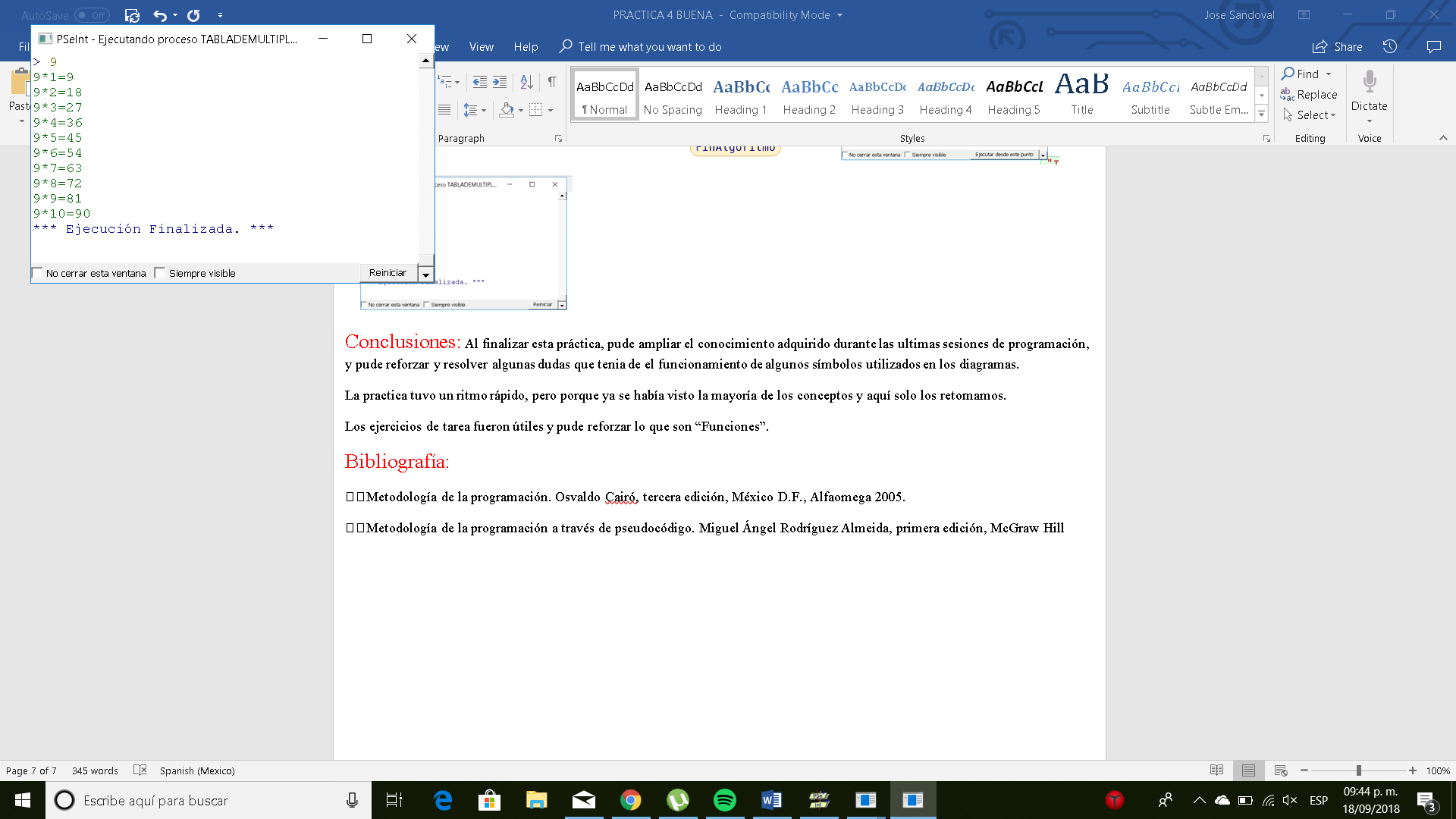
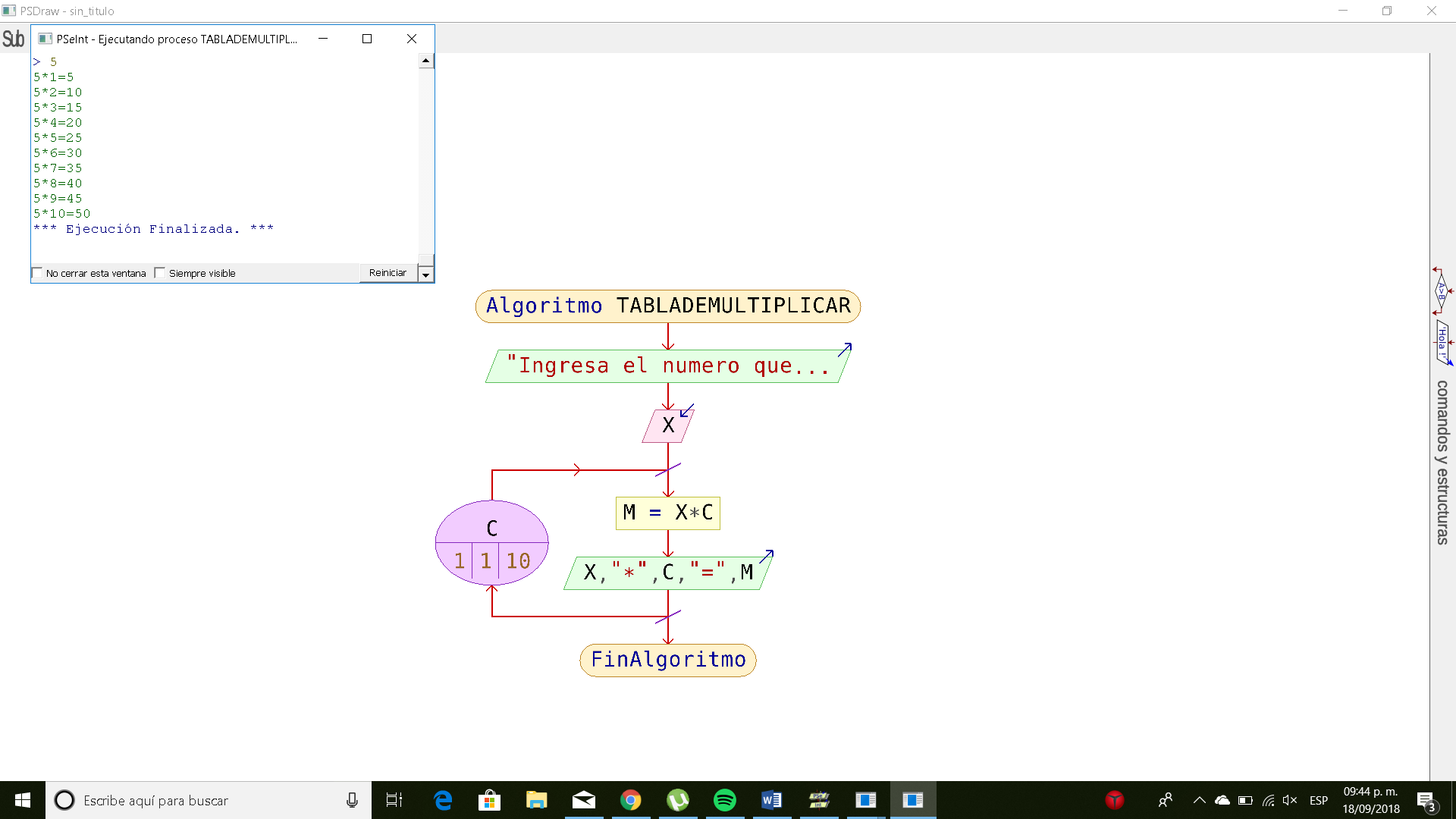
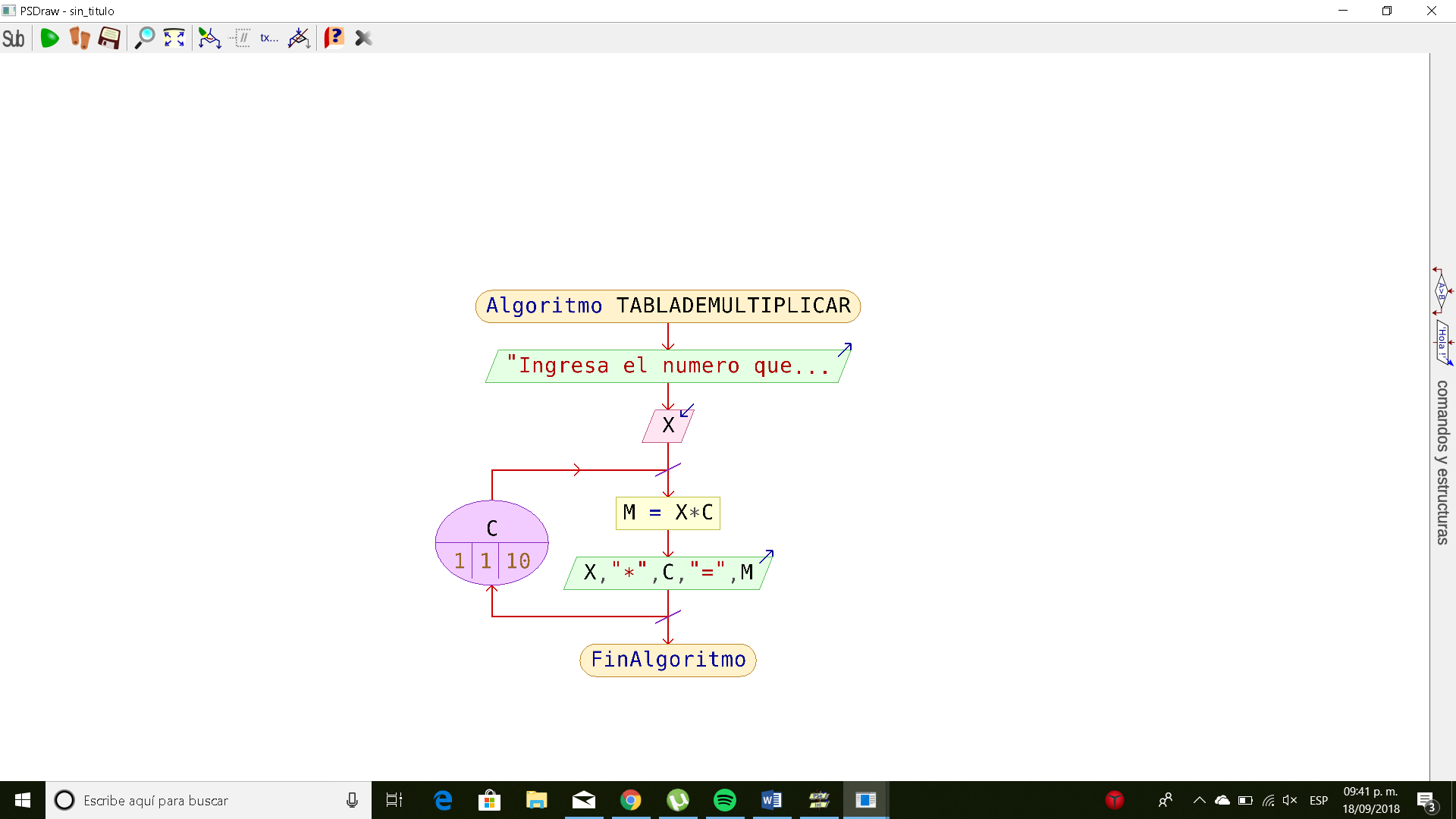
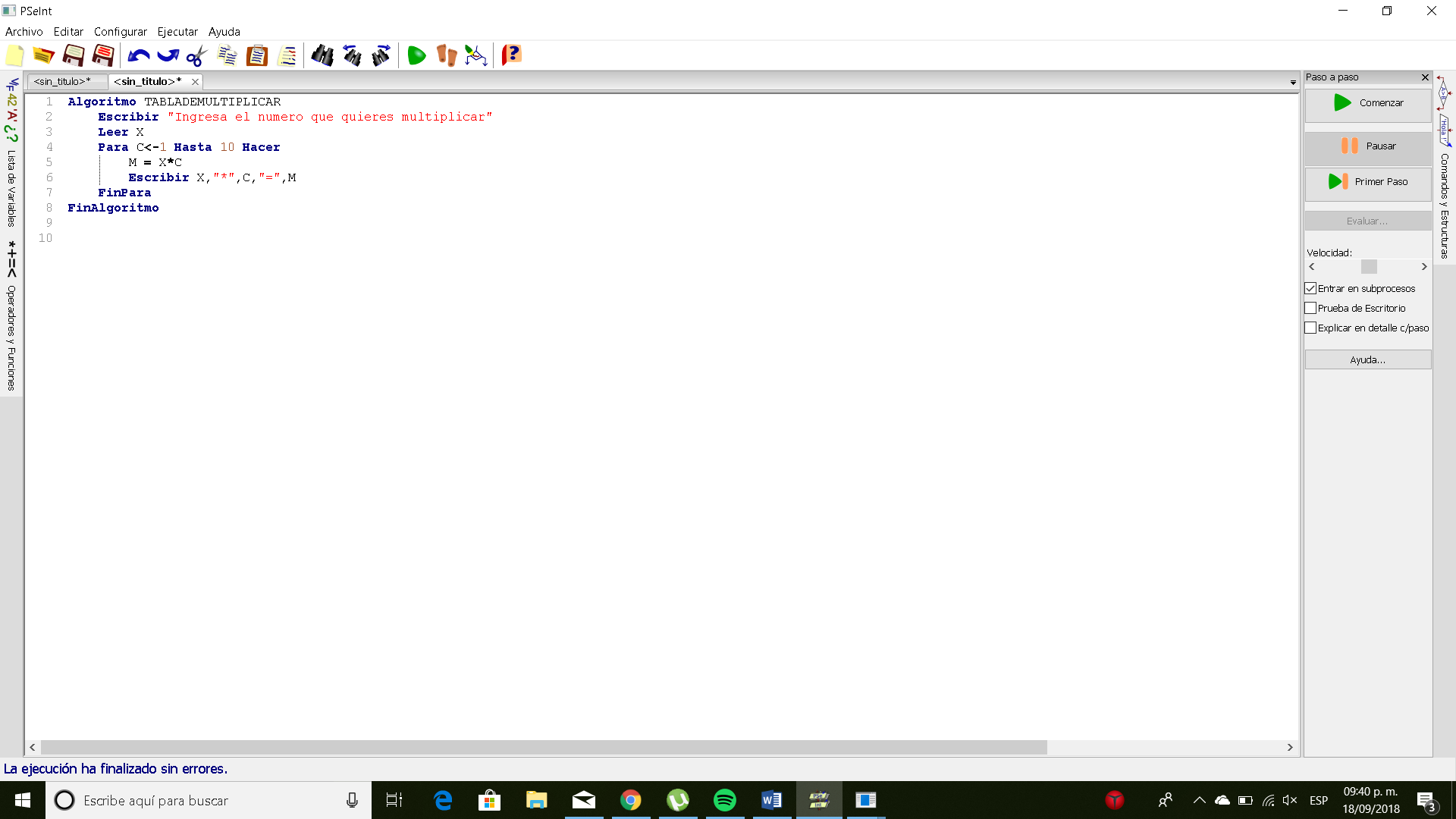
Si D=3

IMPRIMIR “GOLF”

Fin de opción múltiple

Fin

3.- TABLA DE MULTIPLICAR



PSEUDOCODIGO:

Inicio

X: ENTERO

IMPRIMIR “DAME EL NUMERO A MULTIPLICAR”

LEER X

PARA C: =1 HASTA 10 CON PASO 4 HACER

M: =X\*C

ESCRIBIR X,” \*”, C,” =”, M

FinPara

Fin

CONCLUSIONES:

* Tras haber visto los algoritmos y los diagramas de flujo, los seudocódigos, cobran mayor sentido ya que solamente son una forma diferente de representar un algoritmo que quiere ser trasladado a un código informático.
* Vi los mismos conceptos ya antes vistos, pero ahora de manera más precisa para poder llevarlos a la codificación.
* Hubiera sido provechoso que realizáramos más ejercicios dentro de clase, pero los ejercicios de tarea ayudan a retomar el conocimiento de diagramas de flujo y acostumbrarse a trasladarlos a seudocódigo.

BIBLIOGRAFIA:

* Metodología de la programación. Osvaldo Cairó, tercera edición, México D.F., Alfaomega 2005.
* Metodología de la programación a través de pseudocódigo. Miguel Ángel Rodríguez Almeida, primera edición, McGraw Hill