**Module main()**

Declare Integer n, k, res, opcion

Display “Menú de opciones”

Display “Opción 1. Sumatoria”

Display “Opción 2. Sumatoria con fórmula”

Display “Opción 3. Sumatoria desde k”

Display “Opción 4. Productoria”

Display “Opción 5. Productoria desde k”

Display “Indique la acción que desea realizar”

Input opcion

case opcion:

1:

Display “Ingrese el valor de n: ”

Input n

set res = call sumatoria(n)

break

2:

Display “Ingrese el valor de n: ”

Input n

set res = call sumatoria\_formula(n)

break

3:

Display “Ingrese el valor de n: ”

Input n

Display “Ingrese el valor de k: ”

Input k

set res = call sumatoria\_k(n, k)

break

4:

Display “Ingrese el valor de n: ”

Input n

set res = call productoria(n)

break

5:

Display “Ingrese el valor de n: ”

Input n

Display “Ingrese el valor de k: ”

Input k

set res = call productoria\_k(n, k)

break

default:

“No existe esa opción”

EndCase

Display “El resultado es: ”, res

**EndModule**

**Module sumatoria(Integer n)**

Declare Integer i

Declare Integer res

For i=1 To n

res = res + i

i=i+1

EndFor

return res

**EndModule**

**Module sumatoria\_formula(Integer n)**

Declare Integer res, i

set res = n \* (n + 1)/2

return res

**EndModule**

**Module sumatoria\_k(Integer n, Integer k)**

Declare Integer res, i

For i=k To n

res = res + i

i=i+1

EndFor

return res

**EndModule**

**Module productoria(Integer n)**

Declare Integer res, i

set res = 1

For i=1 To n

res = res \* i

i=i+1

EndFor

return res

**EndModule**

**Module productoria\_k(Integer n, Integer k)**

Declare Integer i, res

set res = 1

For i=k To n

res = res \* i

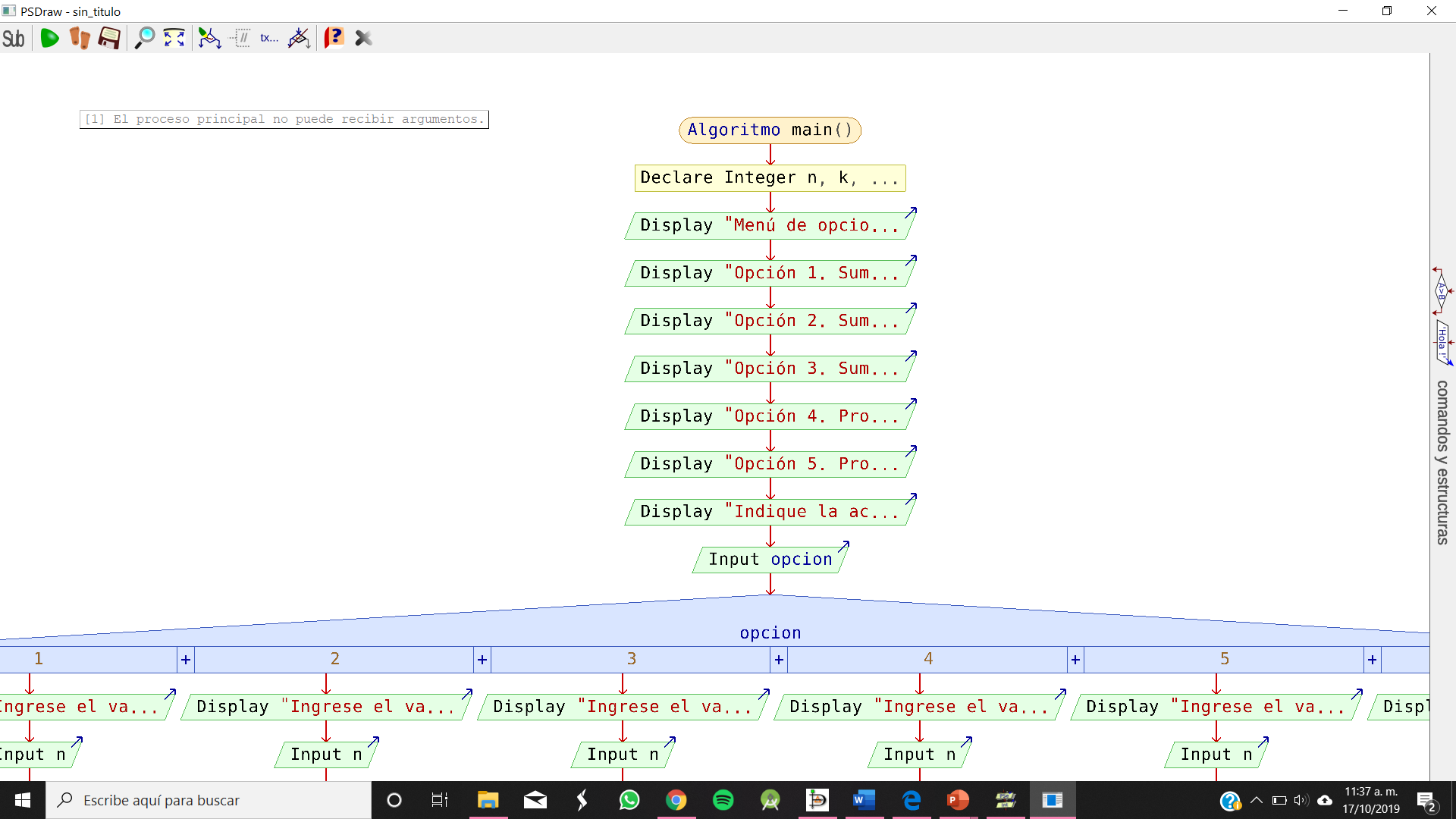
i=i+1

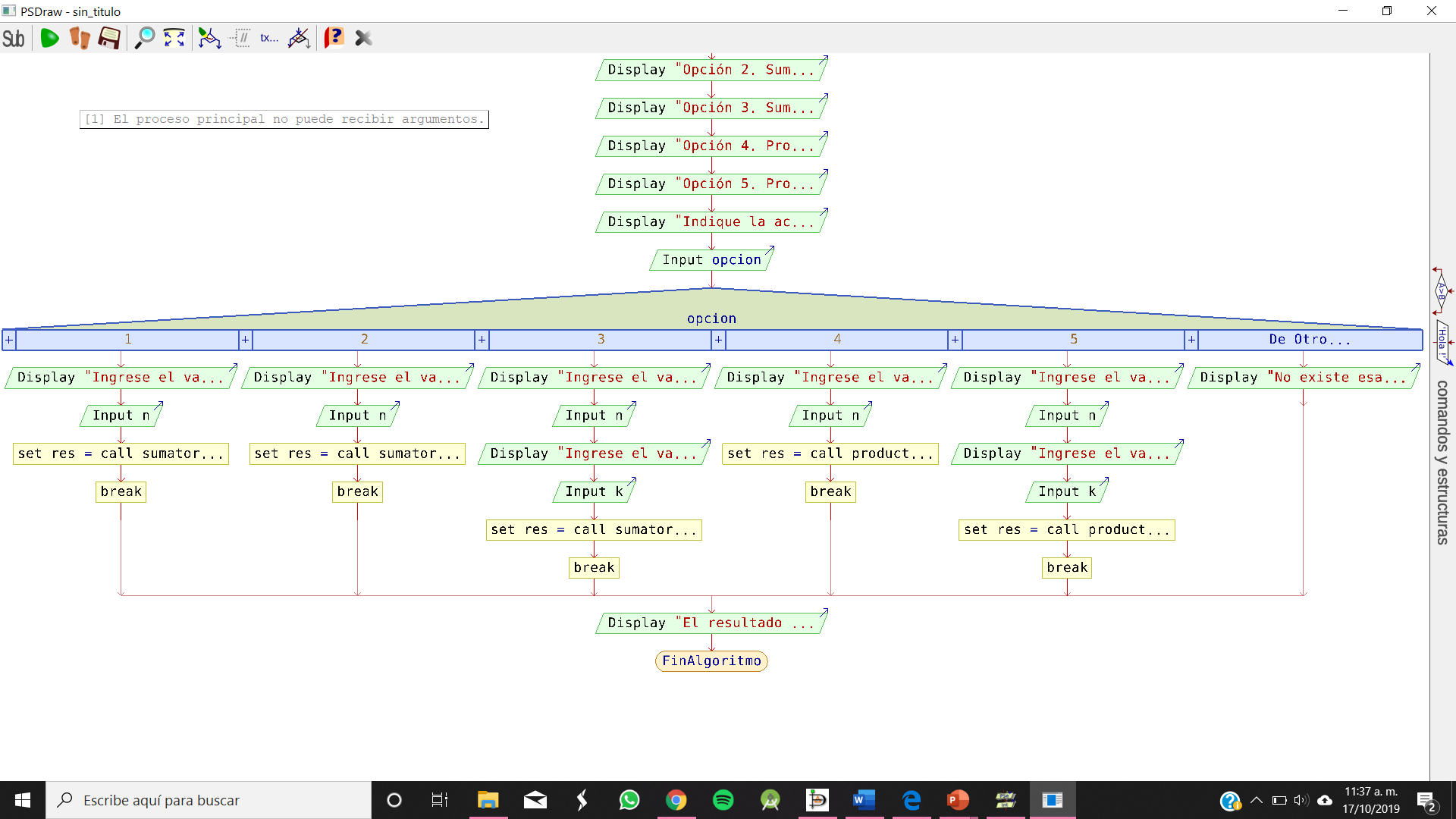
EndFor

return res

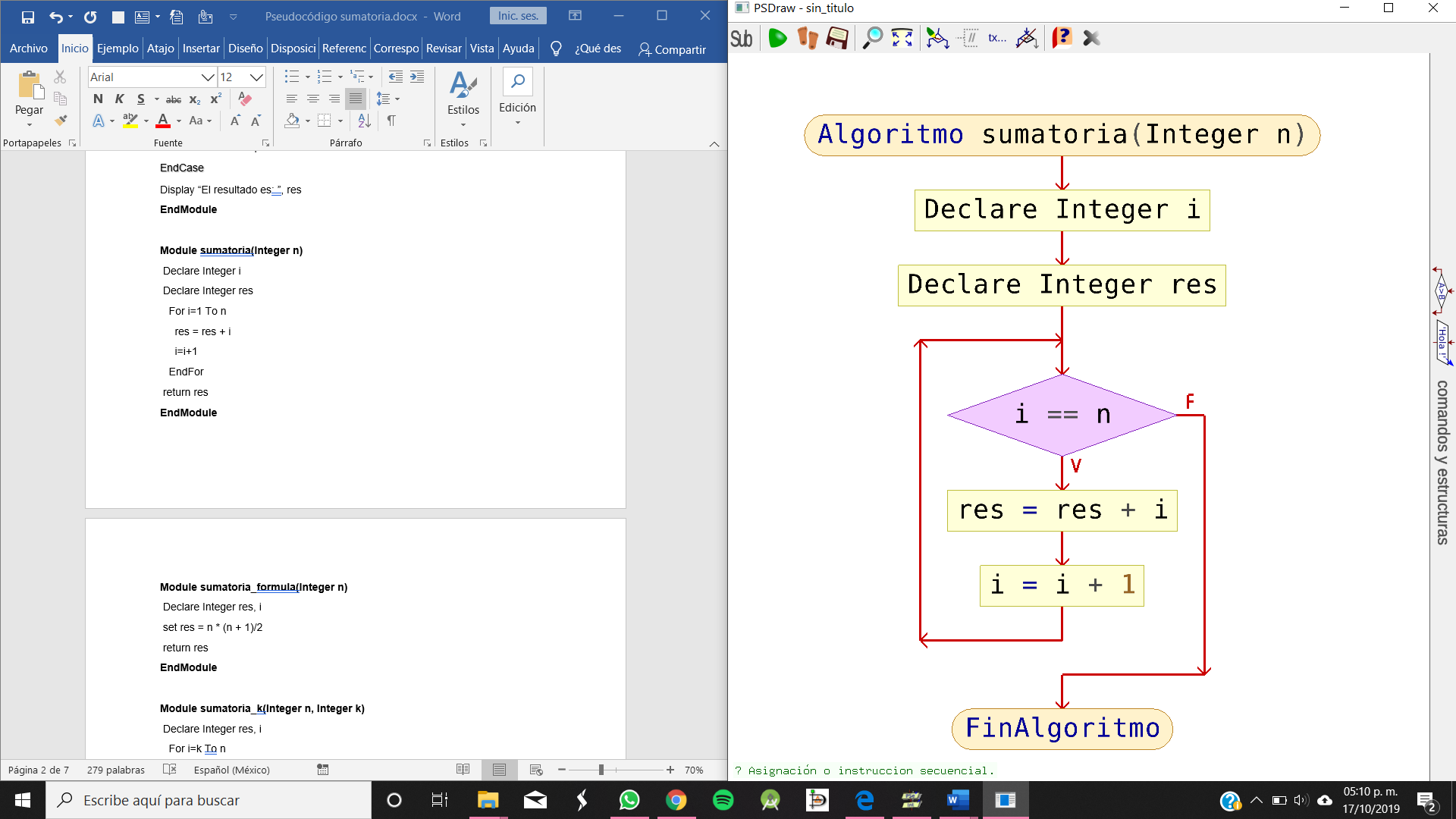
**EndModule**

**Diagrama de flujo-Módulo main**





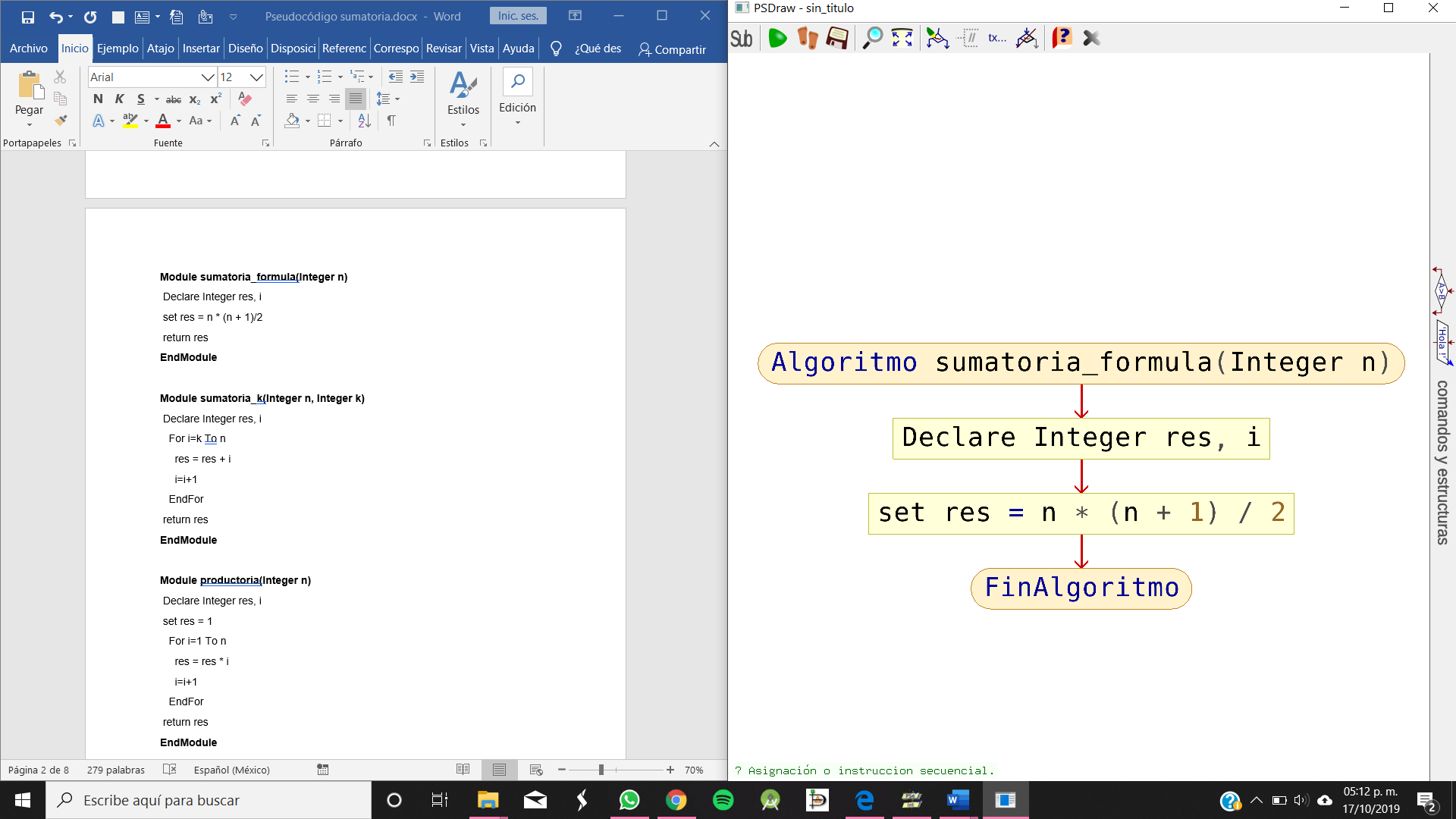
**Diagrama de flujo-Módulo sumatoria**



**Module**

**Return** res

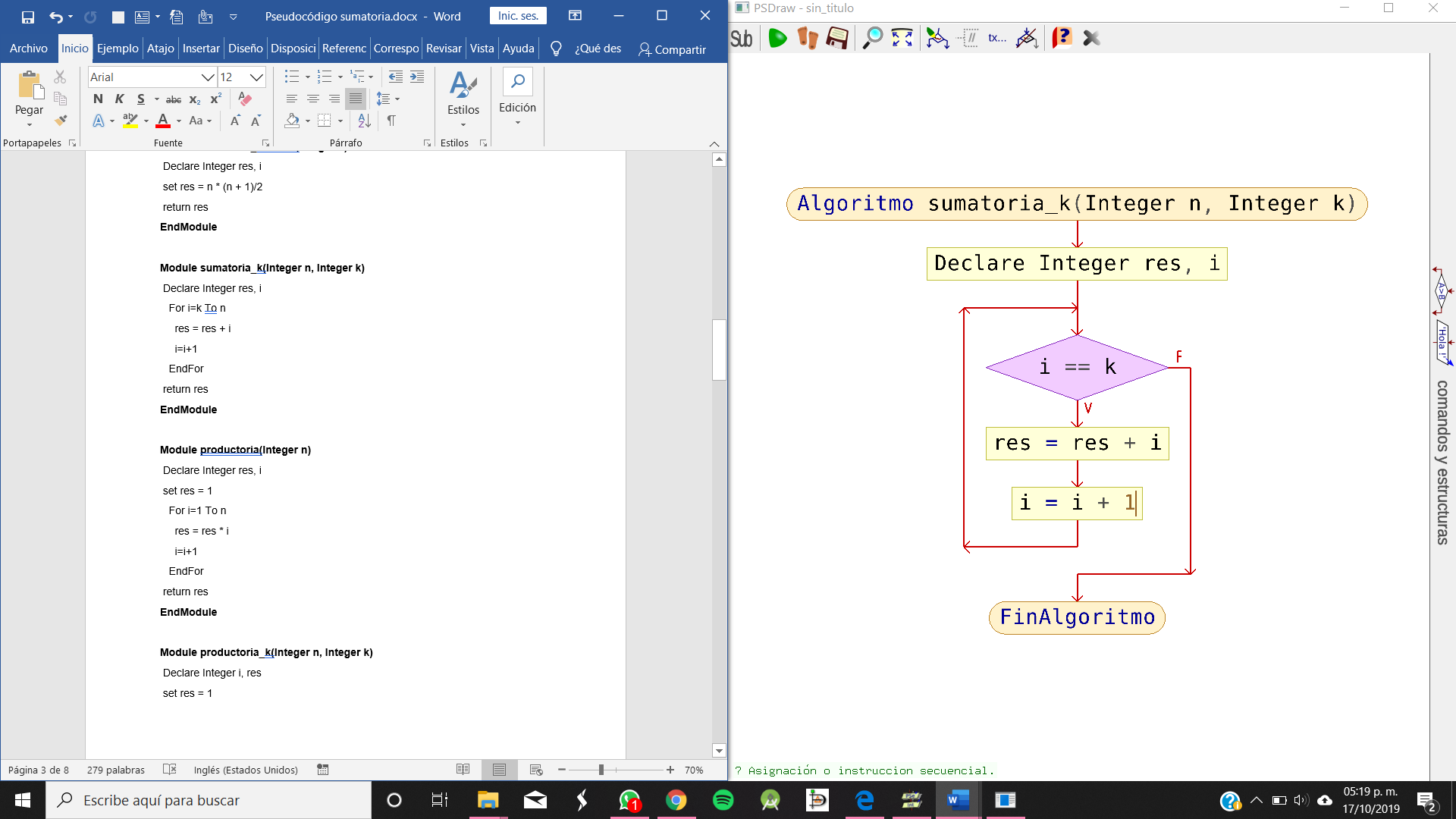
**Diagrama de flujo-Módulo sumatoria\_formula**



**Module**

**Return** res

**Diagrama de flujo-Módulo sumatoria\_k**



**Module**

**Return** res

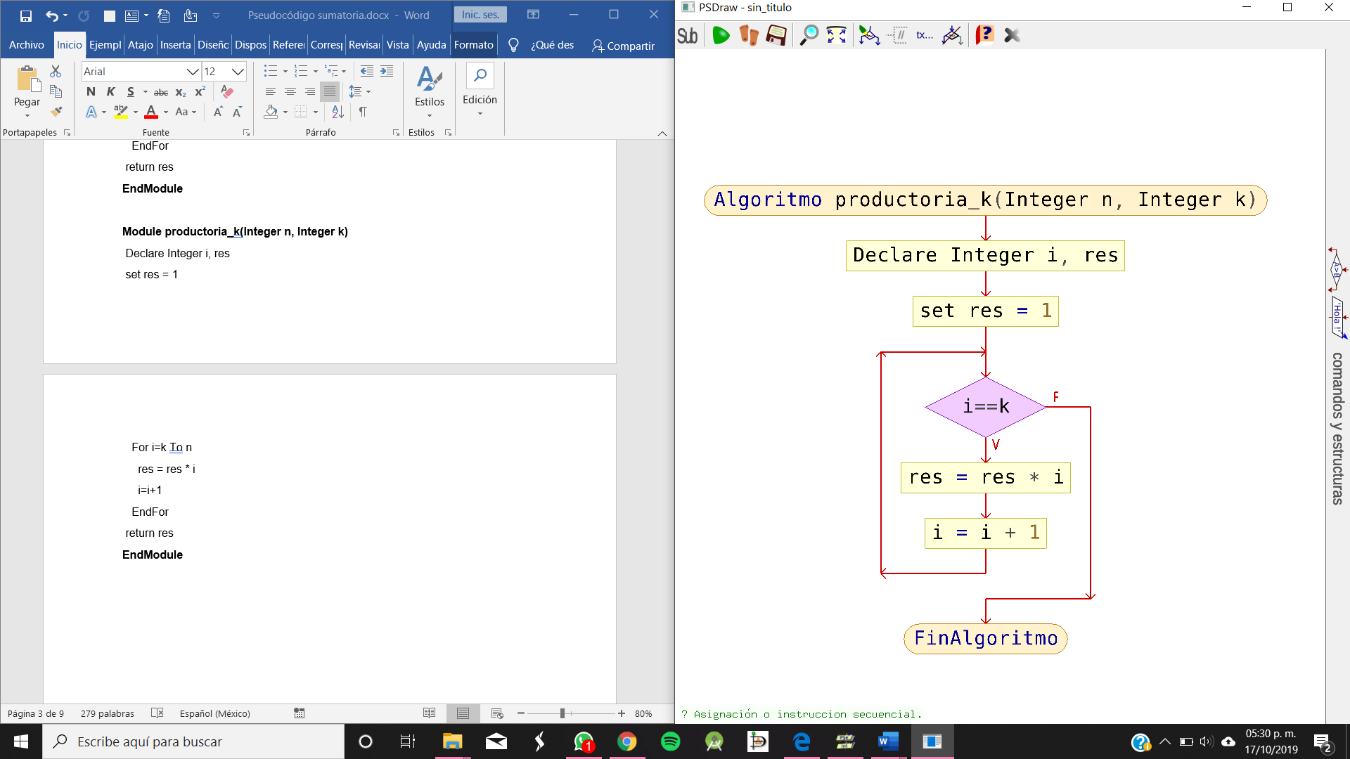
**Diagrama de flujo-Módulo productoria**



**Module**

**Return** res

**Diagrama de flujo-Módulo productoria\_k**



**Module**

**Return** res

**Module main()**

Declare Integer n

Declare Integer k

Display “Introduce el valor de n: ”

Input n

Display “Introduce el valor de k: ”

Input k

If k>n Then

Display “Error: n debe ser mayor o igual a k”

Else

Display “combinacion(n, k)”, call combinacion(n, k)

EndIf

**EndModule**

**Module combinacion(Integer n, Integer k)**

Declare Integer res

set res = call factorial(n)/(call factorial(k)\*call factorial(n-k))

return res

**EndModule**

**Module factorial(Integer n)**

Declare Integer fn = 1

Declare Integer cont

For cont=1 to n

fn = fn\*cont

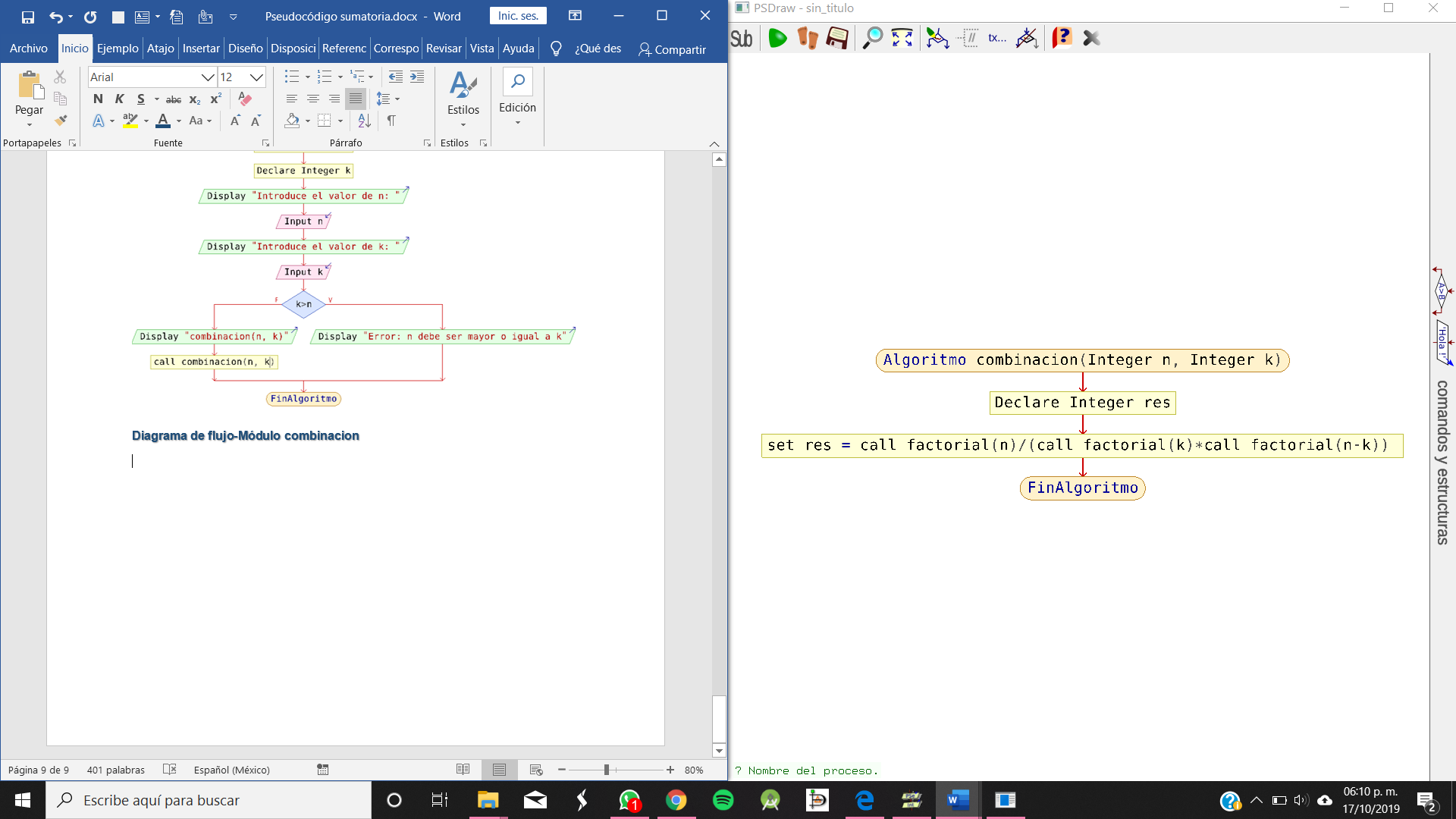
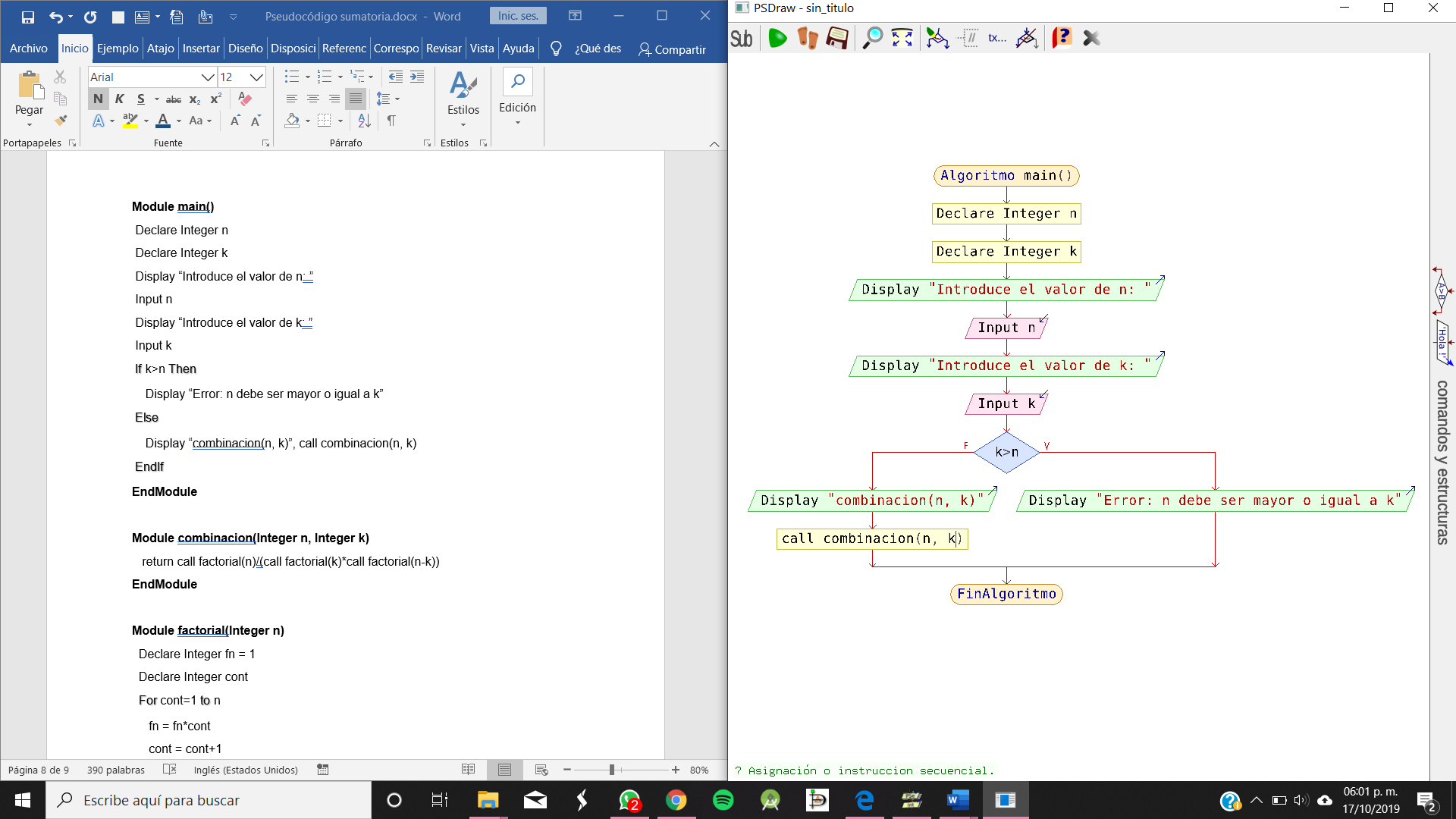
cont = cont+1

EndFor

return fn

**EndModule**

**Diagrama de flujo-Módulo main**

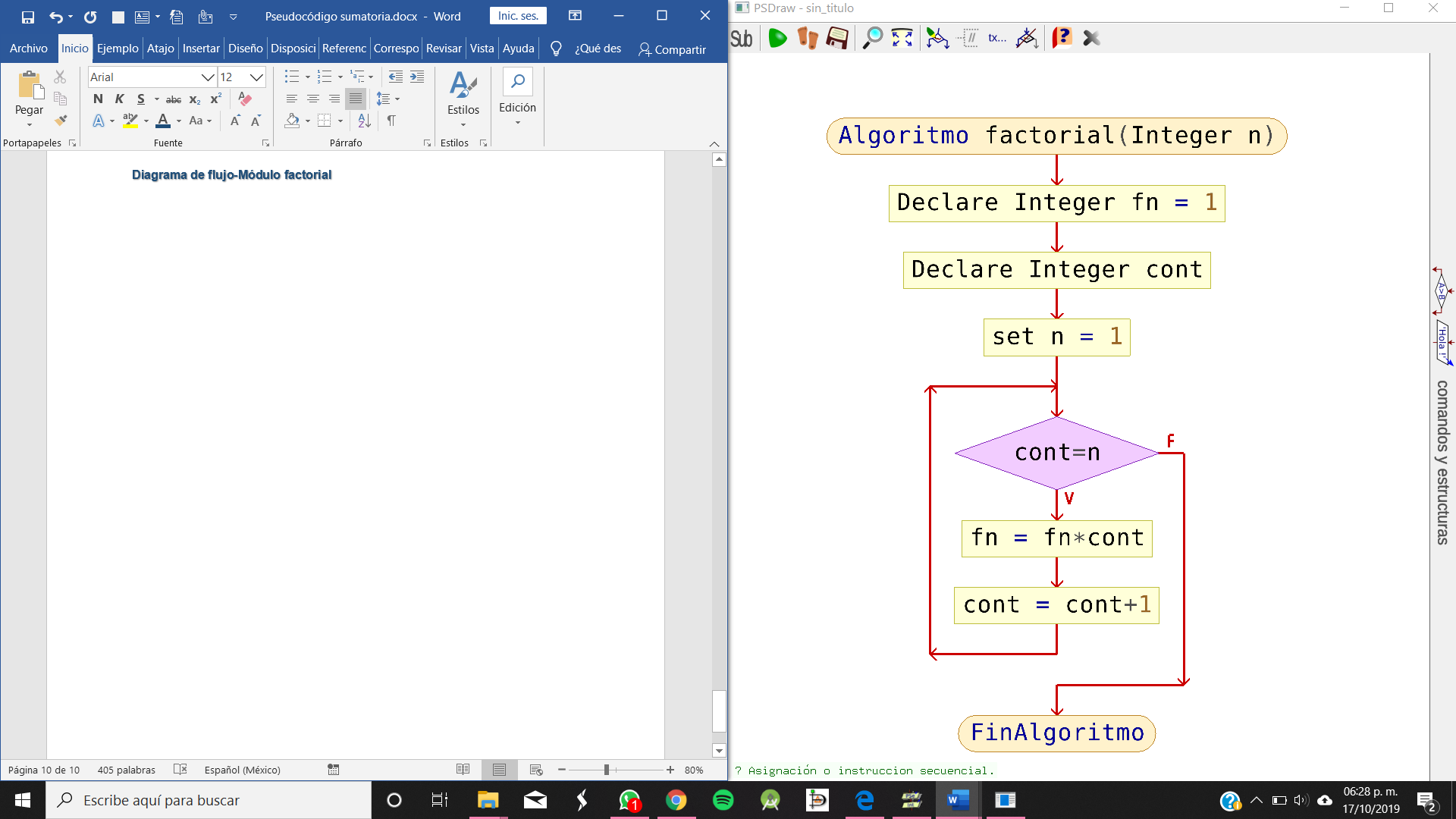


**Module**

**Return** res

**Diagrama de flujo-Módulo combinacion**

**Diagrama de flujo-Módulo factorial**



**Module**

**Return** fn