UNIVERSIDAD RURAL DE GUATEMALA

FACULTAD DE INGENIERÍA
Ingeniería en Sistemas con especialización en IA y Ciberseguridad

Práctica 3: programación modular (unidad 6)

Dividir el trabajo: procedimientos y funciones

Algoritmos • FE729 • Sesión 3 de 4

Ing. Mynor A. Cifuentes

Universidad Rural de Guatemala • Laboratorio presencial • 30 de septiembre de 2026

Agenda de hoy

Los programas crecen; hoy aprendemos a partirlos en piezas con nombre



APERTURA

10 min	Repaso de la sesión 2
10 min	Módulos: procedimientos y funciones

Hoy se trabaja la Práctica 3 del manual: unidad 6 del programa (programación modular).

NOTA

¿Ya subió `Practica_02` a su repositorio de GitHub? Si no, hágalo durante el receso.

PRÁCTICA 3

30 min	**A.** Esquema: partir el problema en tres
35 min	**B.** Un procedimiento que muestra un título
15 min	Receso
45 min	**C.** Una función que calcula el promedio
50 min	**D.** Por valor y por referencia
25 min	Evaluación en Socrative
20 min	Dudas, respaldo y cierre

01 PARTE
ALGORITMOS

Repaso de la sesión 2

¿Lo recuerdan?

Responda en voz alta antes de ver la respuesta

No.	Pregunta	Respuesta
1	¿Con qué palabra se cierra un `Si`?	
2	¿Qué ciclo se ejecuta al menos una vez?	
3	¿Para qué sirve el -1 en `PromedioConCentinela`?	
4	¿Cuál es el primer índice de un arreglo en PSeInt?	
5	¿Qué pasa si falta `Leer nota` dentro del `Mientras`?	

LO QUE SIGUE

Ayer `CincoNotas` ya tenía más de 20 líneas y hacía tres cosas: leer, sumar y mostrar.

Hoy cada una de esas tareas tendrá **su propio nombre**.

¿Lo recuerdan?

Responda en voz alta antes de ver la respuesta

No.	Pregunta	Respuesta
1	¿Con qué palabra se cierra un `Si`?	`FinSi`
2	¿Qué ciclo se ejecuta al menos una vez?	
3	¿Para qué sirve el -1 en `PromedioConCentinela`?	
4	¿Cuál es el primer índice de un arreglo en PSeInt?	
5	¿Qué pasa si falta `Leer nota` dentro del `Mientras`?	

LO QUE SIGUE

Ayer `CincoNotas` ya tenía más de 20 líneas y hacía tres cosas: leer, sumar y mostrar.

Hoy cada una de esas tareas tendrá **su propio nombre**.

¿Lo recuerdan?

Responda en voz alta antes de ver la respuesta

No.	Pregunta	Respuesta
1	¿Con qué palabra se cierra un `Si`?	`FinSi`
2	¿Qué ciclo se ejecuta al menos una vez?	`Repetir … Hasta Que`
3	¿Para qué sirve el -1 en `PromedioConCentinela`?	
4	¿Cuál es el primer índice de un arreglo en PSeInt?	
5	¿Qué pasa si falta `Leer nota` dentro del `Mientras`?	

LO QUE SIGUE

Ayer `CincoNotas` ya tenía más de 20 líneas y hacía tres cosas: leer, sumar y mostrar.

Hoy cada una de esas tareas tendrá **su propio nombre**.

¿Lo recuerdan?

Responda en voz alta antes de ver la respuesta

No.	Pregunta	Respuesta
1	¿Con qué palabra se cierra un `Si`?	`FinSi`
2	¿Qué ciclo se ejecuta al menos una vez?	`Repetir … Hasta Que`
3	¿Para qué sirve el -1 en `PromedioConCentinela`?	Avisa que ya no hay datos
4	¿Cuál es el primer índice de un arreglo en PSeInt?	
5	¿Qué pasa si falta `Leer nota` dentro del `Mientras`?	

LO QUE SIGUE

Ayer `CincoNotas` ya tenía más de 20 líneas y hacía tres cosas: leer, sumar y mostrar.

Hoy cada una de esas tareas tendrá **su propio nombre**.

¿Lo recuerdan?

Responda en voz alta antes de ver la respuesta

No.	Pregunta	Respuesta
1	¿Con qué palabra se cierra un `Si`?	`FinSi`
2	¿Qué ciclo se ejecuta al menos una vez?	`Repetir … Hasta Que`
3	¿Para qué sirve el -1 en `PromedioConCentinela`?	Avisa que ya no hay datos
4	¿Cuál es el primer índice de un arreglo en PSeInt?	`1`: de `notas[1]` a `notas[5]`
5	¿Qué pasa si falta `Leer nota` dentro del `Mientras`?	

LO QUE SIGUE

Ayer `CincoNotas` ya tenía más de 20 líneas y hacía tres cosas: leer, sumar y mostrar.

Hoy cada una de esas tareas tendrá **su propio nombre**.

¿Lo recuerdan?

Responda en voz alta antes de ver la respuesta

No.	Pregunta	Respuesta
1	¿Con qué palabra se cierra un `Si`?	`FinSi`
2	¿Qué ciclo se ejecuta al menos una vez?	`Repetir … Hasta Que`
3	¿Para qué sirve el -1 en `PromedioConCentinela`?	Avisa que ya no hay datos
4	¿Cuál es el primer índice de un arreglo en PSeInt?	`1`: de `notas[1]` a `notas[5]`
5	¿Qué pasa si falta `Leer nota` dentro del `Mientras`?	Ciclo infinito

LO QUE SIGUE

Ayer `CincoNotas` ya tenía más de 20 líneas y hacía tres cosas: leer, sumar y mostrar.

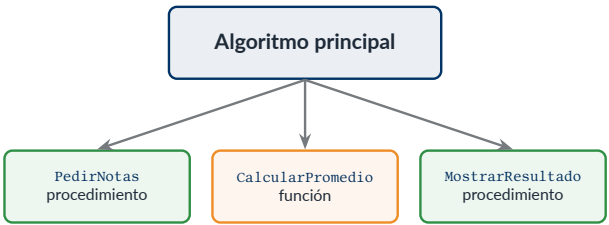
Hoy cada una de esas tareas tendrá **su propio nombre**.

02 PARTE

ALGORITMOS

Módulos: procedimientos y funciones

Diseño descendente: del problema grande a las tareas chicas



Problema: leer tres notas, calcular el promedio y mostrar el resultado

Primero se piensa el problema completo; después se baja a tareas pequeñas. Cada tarea es un **módulo** que hace **una sola cosa** y tiene un nombre claro.

ANALOGÍA

En una cocina, el jefe no hace todo: uno corta, otro cocina, otro sirve. El jefe solo **da las órdenes en el orden correcto**. El algoritmo principal es el jefe.

NOTA

Ventajas: el principal queda corto, cada módulo se prueba por separado y se puede usar varias veces.

Procedimiento y función

La diferencia está en si devuelven un valor

	Procedimiento (`SubProceso`)	Función (`Funcion`)
¿Devuelve un valor?	No: *hace* algo	Sí, uno: *calcula* algo
¿Cómo se llama?	Como una instrucción, en su propia línea	Dentro de una asignación o de un `Escribir`
Ejemplo de hoy	`MostrarTitulo`	`CalcularPromedio`

Se llama un procedimiento

```
1 MostrarTitulo
2 MostrarResultado(resultado)
```

Se usa el valor de una función

```
1 resultado <- CalcularPromedio(a, b, c)
2 Escribir CalcularPromedio(a, b, c)
```

RECOMENDACIÓN

Pista para elegir: si la tarea termina con "...y me da un número", es una función. Si termina con "...y lo muestra", es un procedimiento.

Cómo se escriben en PSeInt

Los módulos van fuera del algoritmo, arriba de `Algoritmo`

Procedimiento

```
SubProceso Nombre(parametros)
  Definir variablesLocales Como ...
  instrucciones
FinSubProceso
```

Función

```
Funcion res <- Nombre(parametros)
  Definir res Como Real
  res <- ...
FinFuncion
```

REGLAS

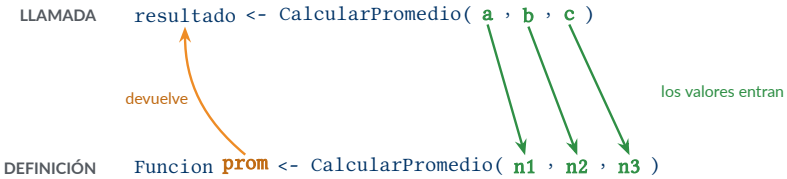
- Se **escriben** fuera del algoritmo principal y se **llaman** desde adentro con su nombre.
- Si nadie los llama, **no se ejecutan**.
- En la función, `res` es la variable que se devuelve: lo último que valga es el resultado.
- Cada módulo declara sus propias variables con `Definir`.

ADVERTENCIA

No llame a un módulo `Mostrar` ni `Leer`: son palabras de PSeInt. Use nombres como `MostrarTitulo`.

Parámetros: los datos entran por los paréntesis

Se copian por posición: el primero con el primero



Argumentos: los valores que se envían al llamar (a, b, c).

Parámetros: los nombres que los reciben dentro del módulo (n1, n2, n3).

Los nombres pueden ser distintos. Lo que importa es el **orden** y la **cantidad**.

Por valor y por referencia

¿El módulo recibe una copia o la variable original?

POR VALOR: UNA FOTOCOPIA



al terminar, `numero` sigue en **10**

ANALOGÍA

Le presta a un compañero una fotocopia de su tarea. Si él la raya, su cuaderno queda intacto.

POR REFERENCIA: EL ORIGINAL



al terminar, `numero` queda en **99**

ANALOGÍA

Le presta su cuaderno original. Lo que él escriba ahí, usted lo encontrará cuando se lo devuelva.

NOTA

En PSeInt se escribe después del nombre del parámetro: `SubProceso Cambiar(dato Por Referencia)`. Si no se indica nada, las variables simples se pasan por valor.

03 PARTE
ALGORITMOS

Práctica 3: Dividir el trabajo

Objetivos y materiales

OBJETIVOS

- Dibujar un esquema simple: algoritmo principal y tres módulos.
- Copiar y llamar un procedimiento desde el algoritmo principal.
- Copiar y usar una función que devuelve el promedio.
- Observar la diferencia entre pasar un dato por valor y por referencia.

MATERIALES Y EQUIPO

- Computadora con **PSeInt** y acceso a Socrative.
- Cuaderno para el esquema de módulos.

Guarde todo en la carpeta `Practica_03`: tres archivos `P03_*.psc`.

NOTA

Hoy no se traduce a C++ ni se usa recursividad: se trabaja con ejemplos ya escritos.

Parte A. Esquema: partir el problema en tres

En el cuaderno

30 min

> **Problema.** Se van a leer tres notas, calcular su promedio y mostrar un mensaje con el resultado.

1. Dibuje tres cajas bajo el título *Algoritmo principal*.
2. Únalas con flechas que bajen desde el principal.
3. Complete la tabla: tipo y qué hace cada módulo.

RECOMENDACIÓN

Si al nombrar un módulo necesita la palabra “y” (por ejemplo *calcular y mostrar*), casi siempre son dos módulos.

Módulo	¿Qué hace?
`PedirNotas` procedimiento	
`CalcularPromedio` función	
`MostrarResultado` procedimiento	

Parte A. Esquema: partir el problema en tres

En el cuaderno

30 min

> **Problema.** Se van a leer tres notas, calcular su promedio y mostrar un mensaje con el resultado.

1. Dibuje tres cajas bajo el título *Algoritmo principal*.
2. Únalas con flechas que bajen desde el principal.
3. Complete la tabla: tipo y qué hace cada módulo.

RECOMENDACIÓN

Si al nombrar un módulo necesita la palabra “y” (por ejemplo *calcular y mostrar*), casi siempre son dos módulos.

Módulo	¿Qué hace?
`PedirNotas` procedimiento	Lee las tres notas del teclado
`CalcularPromedio` función	
`MostrarResultado` procedimiento	

Parte A. Esquema: partir el problema en tres

En el cuaderno

30 min

> **Problema.** Se van a leer tres notas, calcular su promedio y mostrar un mensaje con el resultado.

1. Dibuje tres cajas bajo el título *Algoritmo principal*.
2. Únalas con flechas que bajen desde el principal.
3. Complete la tabla: tipo y qué hace cada módulo.

RECOMENDACIÓN

Si al nombrar un módulo necesita la palabra “y” (por ejemplo *calcular y mostrar*), casi siempre son dos módulos.

Módulo	¿Qué hace?
`PedirNotas` procedimiento	Lee las tres notas del teclado
`CalcularPromedio` función	Recibe tres notas y devuelve su promedio
`MostrarResultado` procedimiento	

Parte A. Esquema: partir el problema en tres

En el cuaderno

30 min

> **Problema.** Se van a leer tres notas, calcular su promedio y mostrar un mensaje con el resultado.

1. Dibuje tres cajas bajo el título *Algoritmo principal*.
2. Únalas con flechas que bajen desde el principal.
3. Complete la tabla: tipo y qué hace cada módulo.

RECOMENDACIÓN

Si al nombrar un módulo necesita la palabra “y” (por ejemplo *calcular y mostrar*), casi siempre son dos módulos.

Módulo	¿Qué hace?
`PedirNotas` procedimiento	Lee las tres notas del teclado
`CalcularPromedio` función	Recibe tres notas y devuelve su promedio
`MostrarResultado` procedimiento	Escribe el promedio en pantalla

NOTA

Solo `CalcularPromedio` es función: es la única que **entrega un número** a quien la llama.

Parte B. Un procedimiento que muestra un título

Archivo `P03_titulo.psc`

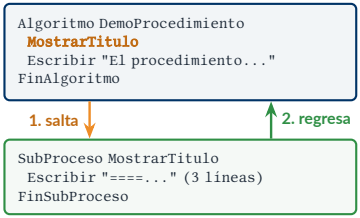
35 min

```
1  SubProceso MostrarTitulo
2    Escribir "============================="
3    Escribir "  Laboratorio de Algoritmos  "
4    Escribir "============================="
5  FinSubProceso
6
7  Algoritmo DemoProcedimiento
8    MostrarTitulo
9    Escribir "El procedimiento ya termino."
10 FinAlgoritmo
```

1. Cree `P03_titulo.psc`, copie el algoritmo y ejecútelo.
2. Cambie el texto del título por el nombre de su curso y vuelva a ejecutar.

PRUEBE

Escriba `MostrarTitulo` dos veces. ¿Cuántas veces sale el título? Ahora bórrelo. ¿Sale?



NOTA

Al terminar, sigue en la línea siguiente a la llamada.

ADVERTENCIA

Sin parámetros no lleva paréntesis. Con `()` vacíos, PSeInt marca “falta el nombre de argumento”.

Parte C. Una función que calcula el promedio

Archivo `P03_promedio.psc`

45 min

```
Funcion prom <- CalcularPromedio(n1, n2, n3)
  Definir prom Como Real
  prom <- (n1 + n2 + n3) / 3
FinFuncion

SubProceso MostrarResultado(promedio)
  Escribir "El promedio es: ", promedio
FinSubProceso

Algoritmo DemoFuncion
  Definir a, b, c, resultado Como Real

  Escribir "Tres notas:"
  Leer a, b, c
  resultado <- CalcularPromedio(a, b, c)
  MostrarResultado(resultado)
FinAlgoritmo
```

1. Cree `P03_promedio.psc` y copie el algoritmo.
2. Ejecútelo con `80`, `70` y `90`. El promedio debe ser `80`.
3. En el cuaderno escriba la línea que **guarda** el valor que devuelve la función.

OBSERVE

- Hay una función y un procedimiento en el mismo archivo.
- El algoritmo principal solo tiene 4 instrucciones útiles.

Parte C. Una función que calcula el promedio

Archivo `P03_promedio.psc`

45 min

```
Funcion prom <- CalcularPromedio(n1, n2, n3)
  Definir prom Como Real
  prom <- (n1 + n2 + n3) / 3
FinFuncion

SubProceso MostrarResultado(promedio)
  Escribir "El promedio es: ", promedio
FinSubProceso

Algoritmo DemoFuncion
  Definir a, b, c, resultado Como Real

  Escribir "Tres notas:"
  Leer a, b, c
  resultado <- CalcularPromedio(a, b, c)
  MostrarResultado(resultado)
FinAlgoritmo
```

1. Cree `P03_promedio.psc` y copie el algoritmo.
2. Ejecútelo con `80`, `70` y `90`. El promedio debe ser `80`.
3. En el cuaderno escriba la línea que **guarda** el valor que devuelve la función.

> Es la línea 15: `resultado <- CalcularPromedio(a, b, c)`. La función calcula y entrega; el principal guarda.

OBSERVE

- Hay una función y un procedimiento en el mismo archivo.
- El algoritmo principal solo tiene 4 instrucciones útiles.

Seguimiento de la parte C

El recorrido de los datos con 80, 70 y 90

Paso	Dónde	Qué pasa
1	Principal	`a = 80, b = 70, c = 90`
2	Llamada	
3	`CalcularPromedio`	
4	Regreso al principal	
5	`MostrarResultado`	

RECOMENDACIÓN

Pruebe también con `90`, `85` y `95`: el promedio debe ser `90`.

RETO

Agregue una función `Mayor(n1, n2, n3)` que devuelva la nota más alta y muéstrela con `Escribir`.

Seguimiento de la parte C

El recorrido de los datos con 80, 70 y 90

Paso	Dónde	Qué pasa
1	Principal	`a = 80, b = 70, c = 90`
2	Llamada	`n1 = 80, n2 = 70, n3 = 90`
3	`CalcularPromedio`	
4	Regreso al principal	
5	`MostrarResultado`	

RECOMENDACIÓN

Pruebe también con `90`, `85` y `95`: el promedio debe ser `90`.

RETO

Agregue una función `Mayor(n1, n2, n3)` que devuelva la nota más alta y muéstrela con `Escribir`.

Seguimiento de la parte C

El recorrido de los datos con 80, 70 y 90

Paso	Dónde	Qué pasa
1	Principal	`a = 80, b = 70, c = 90`
2	Llamada	`n1 = 80, n2 = 70, n3 = 90`
3	`CalcularPromedio`	`prom = 240 / 3 = 80`
4	Regreso al principal	
5	`MostrarResultado`	

RECOMENDACIÓN

Pruebe también con `90`, `85` y `95`: el promedio debe ser `90`.

RETO

Agregue una función `Mayor(n1, n2, n3)` que devuelva la nota más alta y muéstrela con `Escribir`.

Seguimiento de la parte C

El recorrido de los datos con 80, 70 y 90

Paso	Dónde	Qué pasa
1	Principal	`a = 80, b = 70, c = 90`
2	Llamada	`n1 = 80, n2 = 70, n3 = 90`
3	`CalcularPromedio`	`prom = 240 / 3 = 80`
4	Regreso al principal	`resultado = 80`
5	`MostrarResultado`	

RECOMENDACIÓN

Pruebe también con `90`, `85` y `95`: el promedio debe ser `90`.

RETO

Agregue una función `Mayor(n1, n2, n3)` que devuelva la nota más alta y muéstrela con `Escribir`.

Seguimiento de la parte C

El recorrido de los datos con 80, 70 y 90

Paso	Dónde	Qué pasa
1	Principal	`a = 80, b = 70, c = 90`
2	Llamada	`n1 = 80, n2 = 70, n3 = 90`
3	`CalcularPromedio`	`prom = 240 / 3 = 80`
4	Regreso al principal	`resultado = 80`
5	`MostrarResultado`	`promedio = 80`; escribe

```
Tres notas:
> 80
> 70
> 90
El promedio es: 80
```

RECOMENDACIÓN

Pruebe también con `90`, `85` y `95`: el promedio debe ser `90`.

RETO

Agregue una función `Mayor(n1, n2, n3)` que devuelva la nota más alta y muéstrela con `Escribir`.

Parte D. Por valor y por referencia

Archivo `P03_parametros.psc`

50 min

```
1  SubProceso PorValor(dato Por Valor)
2    dato <- 99
3    Escribir "Dentro de PorValor, dato vale ", dato
4  FinSubProceso
5
6  SubProceso PorReferencia(dato Por Referencia)
7    dato <- 99
8    Escribir "Dentro de PorReferencia, dato vale ", dato
9  FinSubProceso
10
11 Algoritmo DemoParametros
12   Definir numero Como Entero
13
14   numero <- 10
15   Escribir "Antes: ", numero
16   PorValor(numero)
17   Escribir "Despues de PorValor: ", numero
18
19   numero <- 10
20   PorReferencia(numero)
21   Escribir "Despues de PorReferencia: ", numero
22 FinAlgoritmo
```

1. Cree `P03_parametros.psc` y copie el algoritmo.
2. **Antes de ejecutar**, anote qué valor cree que tendrá `numero` después de cada llamada.
3. Ejecútelo una vez y compare.

FÍJESE EN

- Los dos procedimientos hacen **exactamente lo mismo**: `dato <- 99`.
- Solo cambia una palabra en el encabezado: `Valor` o `Referencia`.
- La línea 19 vuelve a poner `numero` en 10 para empezar igual.

Parte D. Lo que se observó

Anote los valores en la tabla del manual

Momento	Valor de numero
Antes de llamar	10
Después de `PorValor`	
Después de `PorReferencia`	

NOTA

Si no obtuvo 10 y 99, compare su archivo con el del manual línea por línea.

Parte D. Lo que se observó

Anote los valores en la tabla del manual

Momento	Valor de numero
Antes de llamar	10
Después de `PorValor`	**10**
Después de `PorReferencia`	

Por valor: dentro del módulo `dato` sí vale 99, pero era una copia. Al regresar, `numero` sigue en 10.

NOTA

Si no obtuvo 10 y 99, compare su archivo con el del manual línea por línea.

Parte D. Lo que se observó

Anote los valores en la tabla del manual

Momento	Valor de numero
Antes de llamar	10
Después de `PorValor`	**10**
Después de `PorReferencia`	**99**

```
Antes: 10
Dentro de PorValor, dato vale 99
Despues de PorValor: 10
Dentro de PorReferencia, dato vale 99
Despues de PorReferencia: 99
```

Por valor: dentro del módulo `dato` sí vale 99, pero era una copia. Al regresar, `numero` sigue en 10.

Por referencia: `dato` era otro nombre para `numero`. El cambio a 99 se queda.

NOTA

Si no obtuvo 10 y 99, compare su archivo con el del manual línea por línea.

¿Cuándo usar cada forma?

Una regla sencilla para las próximas prácticas

POR VALOR

Cuando el módulo solo necesita **leer** el dato.

Ejemplo: `MostrarResultado(promedio)` solo muestra; no debe cambiar nada.

POR REFERENCIA

Cuando el módulo debe **llenar o cambiar** una variable del principal.

Ejemplo: `PedirNota(nota Por Referencia)` guarda en la variable del principal lo que escribe el usuario.

RECOMENDACIÓN

Por valor es la opción segura: el módulo no puede estropear sus variables por accidente.

NOTA

Mañana, en el proyecto del menú, `CargarNotas` recibirá el arreglo y lo llenará. Los arreglos se pasan por referencia.

Si PSeInt marca un error

Los errores más comunes con módulos

ERRORES FRECUENTES

- Escribir el módulo **dentro** de `Algoritmo … FinAlgoritmo`.
- Olvidar `FinSubProceso` o `FinFuncion`.
- Llamar con **otra cantidad** de datos: la función pide 3 y se envían 2.
- Llamar la función sin guardar ni mostrar su valor: el resultado se pierde.
- Usar dentro del módulo una variable del principal que no se pasó como parámetro.
- Llamar al módulo `Mostrar`, `Leer` u otra palabra de PSeInt.

RECOMENDACIÓN

Use la **ejecución paso a paso**: verá cómo el programa salta al módulo y regresa a la línea siguiente.

NOTA

Cada módulo tiene sus **variables locales**: `prom` solo existe dentro de `CalcularPromedio`. Por eso los datos se pasan como parámetros.

Evaluación en Socrative

25 min Individual • 40 puntos

EVALUACIÓN EN LÍNEA (SOCRATIVE)

1. Abra `socrative.com` en el navegador o en la aplicación.
2. Elija **Iniciar sesión** y luego **Estudiante**.
3. Escriba el nombre de la sala.
4. Ingrese su **nombre completo**, tal como aparece en la lista.

NOMBRE DE LA SALA



NOTA

Las preguntas combinan la teoría de hoy (procedimientos, funciones y parámetros) con lo que usted ejecutó en PSeInt.

Resumen de hoy

- **Diseño descendente:** partir el problema en módulos que hacen una sola cosa.
- Un `SubProceso` *hace* algo; una `Funcion` *devuelve* un valor.
- Los módulos se escriben fuera del algoritmo y se ejecutan solo cuando se llaman.
- Los datos entran por los parámetros, en el mismo orden en que se envían.
- **Por valor:** copia; el original no cambia.
- **Por referencia:** el módulo trabaja sobre la variable original.

RECOMENDACIÓN

Antes de salir: copie la carpeta `Practica_03` en su USB o en la nube y súbala a su repositorio `Algoritmos-FE729`.

Próxima sesión

Última clase: proyecto integrador, un sistema de notas

LO QUE VIENE

- Un programa que junta las tres sesiones y ocupa casi toda la clase.
- Un menú con `Segun` dentro de un `Repetir`.
- Un arreglo de cinco notas con validación, categorías y estadísticas.
- Funciones y procedimientos como los de hoy.

PARA MAÑANA

1. Leer la Práctica 4 del manual.
2. Tener a mano sus archivos `P02_*.psc` y `P03_*.psc`.
3. Subir `Practica_03` a GitHub.

Esta tarde, en **Electrónica Digital**, un circuito también recibirá una entrada de selección y decidirá qué dato pasa: el multiplexor.

UNIVERSIDAD RURAL DE GUATEMALA

¡Gracias!

¿Preguntas?

github.com/MynorCifuentes