UNIVERSIDAD RURAL DE GUATEMALA

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

Práctica 2: estructuras de control y arreglos (unidades 4 y 5)

Decisiones, repeticiones y una lista de notas

Algoritmos • FE729 • Sesión 2 de 4

Ing. Mynor A. Cifuentes

Universidad Rural de Guatemala • Laboratorio presencial • 29 de septiembre de 2026

Agenda de hoy

Hasta ahora el algoritmo hacía siempre lo mismo; hoy aprende a decidir y a repetir

APERTURA

10 min	Repaso de la sesión 1 y GitHub
10 min	Estructuras de control y arreglos

Hoy se trabaja la Práctica 2 del manual: unidades 4 y 5 del programa (estructuras selectivas, repetitivas y arreglos).

NOTA

Tenga a mano el archivo `P01_nota.psc` de ayer: la condición de aprobado vuelve a aparecer hoy.

PRÁCTICA 2

35 min	**A.** Decidir: aprobado o reprobado
35 min	**B.** Varias decisiones: la categoría
15 min	Receso
45 min	**C.** Repetir con `Mientras`
45 min	**D.** Cinco notas en un arreglo
25 min	Evaluación en Socrative
20 min	Dudas, respaldo y cierre

01 PARTE
ALGORITMOS

Repaso de la sesión 1

¿Lo recuerdan?

Responda en voz alta antes de ver la respuesta

No.	Pregunta	Respuesta
1	¿Cómo se declara una variable con decimales en PSeInt?	
2	¿Cómo se escribe el cociente entero de 17 entre 5?	
3	¿Cuánto vale `17 MOD 5`?	
4	Con `a = 5` y `b = 10`: ¿`a > b` O `b > a`?	
5	¿Qué etapa compara la pantalla con el cálculo a mano?	

LO QUE SIGUE

Ayer calculamos

```
aprobado <- notaFinal
  >= NOTA_MINIMA
```

un valor `Verdadero` o `Falso`.

Hoy ese valor servirá para **decidir qué mensaje mostrar**.

¿Lo recuerdan?

Responda en voz alta antes de ver la respuesta

No.	Pregunta	Respuesta
1	¿Cómo se declara una variable con decimales en PSeInt?	`Definir nota Como Real`
2	¿Cómo se escribe el cociente entero de 17 entre 5?	
3	¿Cuánto vale `17 MOD 5`?	
4	Con `a = 5` y `b = 10`: ¿`a > b` O `b > a`?	
5	¿Qué etapa compara la pantalla con el cálculo a mano?	

LO QUE SIGUE

Ayer calculamos `aprobado <- notaFinal >= NOTA_MINIMA` un valor `Verdadero` o `Falso`.

Hoy ese valor servirá para **decidir qué mensaje mostrar**.

¿Lo recuerdan?

Responda en voz alta antes de ver la respuesta

No.	Pregunta	Respuesta
1	¿Cómo se declara una variable con decimales en PSeInt?	`Definir nota Como Real`
2	¿Cómo se escribe el cociente entero de 17 entre 5?	`trunc(17 / 5)` vale `3`
3	¿Cuánto vale `17 MOD 5`?	
4	Con `a = 5` y `b = 10`: ¿`a > b` O `b > a`?	
5	¿Qué etapa compara la pantalla con el cálculo a mano?	

LO QUE SIGUE

Ayer calculamos `aprobado <- notaFinal >= NOTA_MINIMA` un valor `Verdadero` o `Falso`.

Hoy ese valor servirá para **decidir qué mensaje mostrar**.

¿Lo recuerdan?

Responda en voz alta antes de ver la respuesta

No.	Pregunta	Respuesta
1	¿Cómo se declara una variable con decimales en PSeInt?	`Definir nota Como Real`
2	¿Cómo se escribe el cociente entero de 17 entre 5?	`trunc(17 / 5)` vale `3`
3	¿Cuánto vale `17 MOD 5`?	`2`, el residuo
4	Con `a = 5` y `b = 10`: ¿`a > b` O `b > a`?	
5	¿Qué etapa compara la pantalla con el cálculo a mano?	

LO QUE SIGUE

Ayer calculamos `aprobado <- notaFinal >= NOTA_MINIMA` un valor `Verdadero` o `Falso`.

Hoy ese valor servirá para **decidir qué mensaje mostrar**.

¿Lo recuerdan?

Responda en voz alta antes de ver la respuesta

No.	Pregunta	Respuesta
1	¿Cómo se declara una variable con decimales en PSeInt?	`Definir nota Como Real`
2	¿Cómo se escribe el cociente entero de 17 entre 5?	`trunc(17 / 5)` vale `3`
3	¿Cuánto vale `17 MOD 5`?	`2`, el residuo
4	Con `a = 5` y `b = 10`: ¿`a > b O b > a`?	`Verdadero`: basta una
5	¿Qué etapa compara la pantalla con el cálculo a mano?	

LO QUE SIGUE

Ayer calculamos `aprobado <- notaFinal >= NOTA_MINIMA` un valor `Verdadero` o `Falso`.

Hoy ese valor servirá para **decidir qué mensaje mostrar**.

¿Lo recuerdan?

Responda en voz alta antes de ver la respuesta

No.	Pregunta	Respuesta
1	¿Cómo se declara una variable con decimales en PSeInt?	`Definir nota Como Real`
2	¿Cómo se escribe el cociente entero de 17 entre 5?	`trunc(17 / 5)` vale `3`
3	¿Cuánto vale `17 MOD 5`?	`2`, el residuo
4	Con `a = 5` y `b = 10`: ¿`a > b` O `b > a`?	`Verdadero`: basta una
5	¿Qué etapa compara la pantalla con el cálculo a mano?	La **prueba**

LO QUE SIGUE

Ayer calculamos `aprobado <- notaFinal >= NOTA_MINIMA` un valor `Verdadero` o `Falso`.

Hoy ese valor servirá para **decidir qué mensaje mostrar**.

Su repositorio en GitHub

Cinco minutos para dejarlo listo

1. Entre a `github.com` con su cuenta.
2. Abra su repositorio `Algoritmos-FE729`. Si no existe: **New repository**, público, con `README`.
3. **Add file** → **Upload files**: suba la carpeta `Practica_01`.
4. Escriba un mensaje corto, por ejemplo `Practica 1: primer algoritmo`, y pulse **Commit changes**.

¿POR QUÉ HACERLO CADA SESIÓN?

- Su trabajo queda respaldado aunque pierda la USB.
- Al final del curso tendrá un **portafolio** que puede mostrar.
- Así se trabaja en la industria: cada cambio queda registrado.

RECOMENDACIÓN

Al final de hoy subirá también la carpeta `Practica_02`.

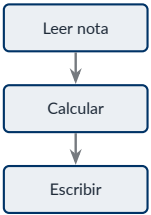
02 PARTE

ALGORITMOS

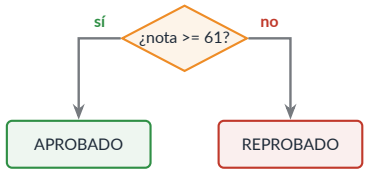
Estructuras de control

Tres formas de ordenar las instrucciones

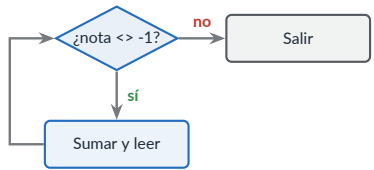
SECUENCIAL



SELECTIVA



REPETITIVA



Secuencial: una instrucción tras otra. Es lo que hicimos ayer.

Selectiva: elige un camino según una condición: `Si`, `Segun`.

Repetitiva: repite un bloque: `Mientras`, `Repetir`, `Para`.

Decidir con `Si`

La condición siempre vale `Verdadero` o `Falso`

Selectiva doble: dos caminos

```
Si notaFinal >= 61 Entonces
  Escribir "APROBADO"
SiNo
  Escribir "REPROBADO"
FinSi
```

Selectiva simple: un camino o nada

```
Si notaFinal >= 90 Entonces
  Escribir "Felicitaciones"
FinSi
```

REGLAS DE ESCRITURA

- `Entonces` va al final de la línea del `Si`.
- Todo `Si` se cierra con `FinSi`.
- `SiNo` es opcional: úselo solo si hay un segundo camino.
- Sangría de dos espacios dentro de cada bloque.

ADVERTENCIA

En la condición se compara con `=`; para guardar un valor se usa `<-`. `Si nota <- 61` es un error.

Varias decisiones: Si encadenado o Segun

Rangos: Si dentro de SiNo

```
Si nota >= 90 Entonces
  categoria <- "Sobresaliente"
SiNo
  Si nota >= 76 Entonces
    categoria <- "Notable"
  SiNo
    ...
  FinSi
FinSi
```

Se revisa de arriba hacia abajo: si la nota es 92, ya no se pregunta por 76 ni por 61.

Valores sueltos: Segun

```
Segun dia Hacer
  1:
    Escribir "Lunes"
  6, 7:
    Escribir "Fin de semana"
  De Otro Modo:
    Escribir "Valor invalido"
FinSegun
```

NOTA

Segun sirve para valores exactos (1, 2, 3...), **no para rangos** como “de 61 a 75”.

Tres ciclos para repetir

Ciclo	Pregunta la condición	Se usa cuando	Forma en PSeInt
`Mientras`	Antes de entrar	No se sabe si habrá que repetir	`Mientras cond Hacer` … `FinMientras`
`Repetir`	Al final	Debe hacerse al menos una vez (un menú, pedir de nuevo un dato)	`Repetir` … `Hasta Que cond`
`Para`	Lleva un contador	Ya se sabe cuántas vueltas: 5 notas, 10 filas	`Para i <- 1 Hasta 5 Con Paso 1 Hacer` … `FinPara`

ANALOGÍA

`Mientras`: "mientras haya fila, atienda". `Para`: "dé 5 vueltas a la cancha". `Repetir`: "pruebe la sopa hasta que esté bien".

ADVERTENCIA

Ciclo infinito: si dentro del ciclo nada cambia la condición, nunca termina. En PSeInt se detiene con el botón de **parar**.

Contador, acumulador y centinela

Tres ideas que aparecen en casi todos los ciclos

CONTADOR

Suma de uno en uno. Dice **cuántos**.

```
cantidad <- cantidad + 1
```

ACUMULADOR

Suma valores distintos. Dice **cuánto en total**.

```
suma <- suma + nota
```

CENTINELA

Un valor que **no es un dato**: es la señal de "ya terminé".

```
CENTINELA <- -1
```

Se lee	suma	cantidad
(inicio)	0	0
80		
90		
−1		

NOTA

Contadores y acumuladores **empiezan en 0** antes del ciclo. Si no se inicializan, el resultado no es confiable.

Contador, acumulador y centinela

Tres ideas que aparecen en casi todos los ciclos

CONTADOR

Suma de uno en uno. Dice **cuántos**.

```
cantidad <- cantidad + 1
```

ACUMULADOR

Suma valores distintos. Dice **cuánto en total**.

```
suma <- suma + nota
```

CENTINELA

Un valor que **no es un dato**: es la señal de "ya terminé".

```
CENTINELA <- -1
```

Se lee	suma	cantidad
(inicio)	0	0
80	80	1
90		
−1		

NOTA

Contadores y acumuladores **empiezan en 0** antes del ciclo. Si no se inicializan, el resultado no es confiable.

Contador, acumulador y centinela

Tres ideas que aparecen en casi todos los ciclos

CONTADOR

Suma de uno en uno. Dice **cuántos**.

```
cantidad <- cantidad + 1
```

ACUMULADOR

Suma valores distintos. Dice **cuánto en total**.

```
suma <- suma + nota
```

CENTINELA

Un valor que **no es un dato**: es la señal de "ya terminé".

```
CENTINELA <- -1
```

Se lee	suma	cantidad
(inicio)	0	0
80	80	1
90	170	2
−1		

NOTA

Contadores y acumuladores **empiezan en 0** antes del ciclo. Si no se inicializan, el resultado no es confiable.

Contador, acumulador y centinela

Tres ideas que aparecen en casi todos los ciclos

CONTADOR

Suma de uno en uno. Dice **cuántos**.

```
cantidad <- cantidad + 1
```

ACUMULADOR

Suma valores distintos. Dice **cuánto en total**.

```
suma <- suma + nota
```

CENTINELA

Un valor que **no es un dato**: es la señal de "ya terminé".

```
CENTINELA <- -1
```

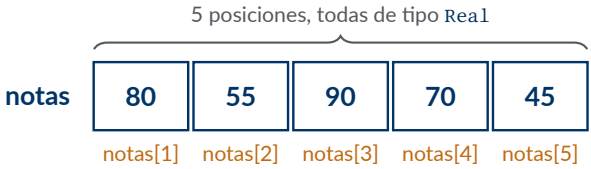
Se lee	suma	cantidad
(inicio)	0	0
80	80	1
90	170	2
-1	170	2

NOTA

Contadores y acumuladores **empiezan en 0** antes del ciclo. Si no se inicializan, el resultado no es confiable.

El -1 no se suma: al leerlo, el ciclo termina. Promedio $= 170/2 = 85$.

Arreglos: una lista con un solo nombre




```
1 Definir notas Como Real
2 Dimension notas[5]
3 notas[3] <- 90
4 Escribir notas[3]
```

¿POR QUÉ UN ARREGLO?

Cinco variables sueltas (`n1`, `n2`...) no se pueden recorrer con un ciclo. Con un arreglo, `notas[i]` cambia de caja en cada vuelta del `Para`.

ADVERTENCIA

En PSeInt (perfil Flexible) el primer índice es `1`. `notas[0]` y `notas[6]` **no existen**: el programa se detiene con un error.

03 PARTE

ALGORITMOS

Práctica 2: Decisiones, repeticiones y una lista de notas

Objetivos y materiales

OBJETIVOS

- Usar `Si` / `SiNo` para mostrar APROBADO o REPROBADO.
- Asignar una categoría de nota con varias decisiones encadenadas.
- Repetir una lectura con `Mientras` hasta que el usuario termine.
- Llenar un arreglo de cinco notas con `Para` y calcular el promedio.

MATERIALES Y EQUIPO

- Computadora con **PSeInt** y acceso a Socrative.
- Cuaderno para la tabla de seguimiento.
- El archivo `P01_nota.psc`, si todavía lo tiene.

Guarde todo en la carpeta `Practica_02`: cuatro archivos `P02_*.psc`.

Parte A. Decidir: aprobado o reprobado

Archivo `P02_aprobado.psc`

35 min

```
Algoritmo EvaluarEstudiante
  Definir NOTA_MINIMA Como Entero
  Definir notaFinal Como Real
  NOTA_MINIMA <- 61

  Escribir "Ingrese la nota final:"
  Leer notaFinal

  Si notaFinal >= NOTA_MINIMA Entonces
    Escribir "APROBADO"
  SiNo
    Escribir "REPROBADO"
  FinSi

  Si notaFinal >= 90 Entonces
    Escribir "Felicitaciones"
  FinSi
FinAlgoritmo
```

Nota	Mensajes que aparecen
45	
61	
95	

NOTA

Son dos `Si` independientes: el segundo se revisa siempre, pase lo que pase en el primero.

Parte A. Decidir: aprobado o reprobado

Archivo `P02_aprobado.psc`

35 min

```
Algoritmo EvaluarEstudiante
  Definir NOTA_MINIMA Como Entero
  Definir notaFinal Como Real
  NOTA_MINIMA <- 61

  Escribir "Ingrese la nota final:"
  Leer notaFinal

  Si notaFinal >= NOTA_MINIMA Entonces
    Escribir "APROBADO"
  SiNo
    Escribir "REPROBADO"
  FinSi

  Si notaFinal >= 90 Entonces
    Escribir "Felicitaciones"
  FinSi
FinAlgoritmo
```

Nota	Mensajes que aparecen
45	REPROBADO
61	
95	

NOTA

Son dos `Si` independientes: el segundo se revisa siempre, pase lo que pase en el primero.

Parte A. Decidir: aprobado o reprobado

Archivo `P02_aprobado.psc`

35 min

```
Algoritmo EvaluarEstudiante
  Definir NOTA_MINIMA Como Entero
  Definir notaFinal Como Real
  NOTA_MINIMA <- 61

  Escribir "Ingrese la nota final:"
  Leer notaFinal

  Si notaFinal >= NOTA_MINIMA Entonces
    Escribir "APROBADO"
  SiNo
    Escribir "REPROBADO"
  FinSi

  Si notaFinal >= 90 Entonces
    Escribir "Felicitaciones"
  FinSi
FinAlgoritmo
```

Nota	Mensajes que aparecen
45	REPROBADO
61	APROBADO
95	

Con `61` dice APROBADO porque la condición es `>=`: 61 es mayor **o igual** que 61. No felicita: 61 no llega a 90.

NOTA

Son dos `Si` independientes: el segundo se revisa siempre, pase lo que pase en el primero.

Parte A. Decidir: aprobado o reprobado

Archivo `P02_aprobado.psc`

35 min

```
Algoritmo EvaluarEstudiante
  Definir NOTA_MINIMA Como Entero
  Definir notaFinal Como Real
  NOTA_MINIMA <- 61

  Escribir "Ingrese la nota final:"
  Leer notaFinal

  Si notaFinal >= NOTA_MINIMA Entonces
    Escribir "APROBADO"
  SiNo
    Escribir "REPROBADO"
  FinSi

  Si notaFinal >= 90 Entonces
    Escribir "Felicitaciones"
  FinSi
FinAlgoritmo
```

Nota	Mensajes que aparecen
45	REPROBADO
61	APROBADO
95	APROBADO y Felicitaciones

Con `61` dice APROBADO porque la condición es `>=`: 61 es mayor **o igual** que 61. No felicita: 61 no llega a 90.

NOTA

Son dos `Si` independientes: el segundo se revisa siempre, pase lo que pase en el primero.

Parte B. Varias decisiones: la categoría de la nota

Archivo `P02_categoria.psc`

35 min

```
Algoritmo CategoriaDeNota
  Definir nota Como Real
  Definir categoria Como Cadena

  Escribir "Nota (0 a 100):"
  Leer nota

  Si nota >= 90 Entonces
    categoria <- "Sobresaliente"
  SiNo
    Si nota >= 76 Entonces
      categoria <- "Notable"
    SiNo
      Si nota >= 61 Entonces
        categoria <- "Satisfactorio"
      SiNo
        categoria <- "Insuficiente"
      FinSi
    FinSi
  FinSi

  Escribir "Categoria: ", categoria
FinAlgoritmo
```

Rango	Categoría	Prueba
90 a 100	Sobresaliente	
76 a 89	Notable	
61 a 75	Satisfactorio	
0 a 60	Insuficiente	

Pruebe con `50`, `70`, `80` y `92`: cada una debe caer en una categoría distinta.

RECOMENDACIÓN

Pruebe también los límites: `60`, `61`, `75`, `76` y `90`.

Parte B. Varias decisiones: la categoría de la nota

Archivo `P02_categoria.psc`

35 min

```
Algoritmo CategoriaDeNota
  Definir nota Como Real
  Definir categoria Como Cadena

  Escribir "Nota (0 a 100):"
  Leer nota

  Si nota >= 90 Entonces
    categoria <- "Sobresaliente"
  SiNo
    Si nota >= 76 Entonces
      categoria <- "Notable"
    SiNo
      Si nota >= 61 Entonces
        categoria <- "Satisfactorio"
      SiNo
        categoria <- "Insuficiente"
      FinSi
    FinSi
  FinSi

  Escribir "Categoria: ", categoria
FinAlgoritmo
```

Rango	Categoría	Prueba
90 a 100	Sobresaliente	92
76 a 89	Notable	80
61 a 75	Satisfactorio	70
0 a 60	Insuficiente	50

Pruebe con 50, 70, 80 y 92: cada una debe caer en una categoría distinta.

RECOMENDACIÓN

Pruebe también los límites: 60, 61, 75, 76 y 90.

Opcional: Segun para el nombre del día

Si le sobra tiempo en la parte B

```
1  Algoritmo NombreDelDia
2    Definir dia Como Entero
3
4    Escribir "Numero del dia (1 a 7):"
5    Leer dia
6
7    Segun dia Hacer
8      1:
9        Escribir "Lunes"
10     2:
11       Escribir "Martes"
12     3:
13       Escribir "Miercoles"
14     4:
15       Escribir "Jueves"
16     5:
17       Escribir "Viernes"
18     6, 7:
19       Escribir "Fin de semana"
20     De Otro Modo:
21       Escribir "Valor invalido"
22   FinSegun
23 FinAlgoritmo
```

Entrada	Resultado
1	
6	
9	

CÓMO SE LEE

- Cada caso es un valor seguido de dos puntos.
- `6, 7:` agrupa dos valores con la misma acción.
- `De Otro Modo:` atrapa todo lo demás, como el 9.

Opcional: Segun para el nombre del día

Si le sobra tiempo en la parte B

```
Algoritmo NombreDelDia
  Definir dia Como Entero

  Escribir "Numero del dia (1 a 7):"
  Leer dia

  Segun dia Hacer
    1:
      Escribir "Lunes"
    2:
      Escribir "Martes"
    3:
      Escribir "Miercoles"
    4:
      Escribir "Jueves"
    5:
      Escribir "Viernes"
    6, 7:
      Escribir "Fin de semana"
    De Otro Modo:
      Escribir "Valor invalido"
  FinSegun
FinAlgoritmo
```

Entrada	Resultado
1	Lunes
6	Fin de semana
9	Valor invalido

CÓMO SE LEE

- Cada caso es un valor seguido de dos puntos.
- `6, 7:` agrupa dos valores con la misma acción.
- `De Otro Modo:` atrapa todo lo demás, como el 9.

Parte C. Repetir con `Mientras`

Archivo `P02_promedio.psc`

45 min

```
1  Algoritmo PromedioConCentinela
2    Definir CENTINELA, cantidad Como Entero
3    Definir nota, suma, promedio Como Real
4    CENTINELA <- -1
5
6    suma <- 0
7    cantidad <- 0
8
9    Escribir "Ingrese notas. Escriba -1 para terminar."
10   Leer nota
11
12   Mientras nota <> CENTINELA Hacer
13     suma <- suma + nota
14     cantidad <- cantidad + 1
15     Leer nota
16   FinMientras
17
18   Si cantidad > 0 Entonces
19     promedio <- suma / cantidad
20     Escribir "Cantidad: ", cantidad
21     Escribir "Promedio: ", promedio
22   SiNo
23     Escribir "No se ingresaron notas"
24   FinSi
25 FinAlgoritmo
```

LÍNEA POR LÍNEA

- Líneas 6–7: el acumulador y el contador empiezan en 0.
- Línea 10: se lee la **primera** nota antes del ciclo.
- Línea 15: se lee la **siguiente**; sin ella, el ciclo no termina.
- Línea 18: el `Si` evita dividir entre 0 si la primera nota ya es -1.

ADVERTENCIA

Dentro del `Mientras` **siempre** debe volver a leerse `nota`.

Seguimiento con 80, 90, 70, -1

Anote los valores después de sumar cada nota

Después de	nota	suma	cantidad
Empezar		0	0
Primera nota	80		
Segunda nota	90		
Tercera nota	70		

¿POR QUÉ UNA TABLA?

Seguir las variables a mano, fila por fila, es la forma más segura de entender un ciclo y de encontrar errores. Se llama **prueba de escritorio**.

RECOMENDACIÓN

Pruebe también con -1 como primera nota: debe decir `No se ingresaron notas`.

Seguimiento con 80, 90, 70, -1

Anote los valores después de sumar cada nota

Después de	nota	suma	cantidad
Empezar	80	0	0
Primera nota	80		
Segunda nota	90		
Tercera nota	70		

¿POR QUÉ UNA TABLA?

Seguir las variables a mano, fila por fila, es la forma más segura de entender un ciclo y de encontrar errores. Se llama **prueba de escritorio**.

RECOMENDACIÓN

Pruebe también con -1 como primera nota: debe decir `No se ingresaron notas`.

Seguimiento con 80, 90, 70, -1

Anote los valores después de sumar cada nota

Después de	nota	suma	cantidad
Empezar	80	0	0
Primera nota	80	80	1
Segunda nota	90		
Tercera nota	70		

¿POR QUÉ UNA TABLA?

Seguir las variables a mano, fila por fila, es la forma más segura de entender un ciclo y de encontrar errores. Se llama **prueba de escritorio**.

RECOMENDACIÓN

Pruebe también con -1 como primera nota: debe decir `No se ingresaron notas`.

Seguimiento con 80, 90, 70, -1

Anote los valores después de sumar cada nota

Después de	nota	suma	cantidad
Empezar	80	0	0
Primera nota	80	80	1
Segunda nota	90	170	2
Tercera nota	70		

¿POR QUÉ UNA TABLA?

Seguir las variables a mano, fila por fila, es la forma más segura de entender un ciclo y de encontrar errores. Se llama **prueba de escritorio**.

RECOMENDACIÓN

Pruebe también con -1 como primera nota: debe decir `No se ingresaron notas`.

Seguimiento con 80, 90, 70, -1

Anote los valores después de sumar cada nota

Después de	nota	suma	cantidad
Empezar	80	0	0
Primera nota	80	80	1
Segunda nota	90	170	2
Tercera nota	70	240	3

¿POR QUÉ UNA TABLA?

Seguir las variables a mano, fila por fila, es la forma más segura de entender un ciclo y de encontrar errores. Se llama **prueba de escritorio**.

RECOMENDACIÓN

Pruebe también con -1 como primera nota: debe decir `No se ingresaron notas`.

Seguimiento con 80, 90, 70, -1

Anote los valores después de sumar cada nota

Después de	nota	suma	cantidad
Empezar	80	0	0
Primera nota	80	80	1
Segunda nota	90	170	2
Tercera nota	70	240	3

Al leer -1 la condición `nota <> CENTINELA` es `Falso` y el ciclo termina. Pantalla: `Cantidad: 3` y `Promedio: 80` ($240/3$).

¿POR QUÉ UNA TABLA?

Seguir las variables a mano, fila por fila, es la forma más segura de entender un ciclo y de encontrar errores. Se llama **prueba de escritorio**.

RECOMENDACIÓN

Pruebe también con -1 como primera nota: debe decir `No se ingresaron notas`.

Parte D. Cinco notas en un arreglo

Archivo `P02_arreglo.psc`

45 min

```
1  Algoritmo CincoNotas
2    Definir CANTIDAD, NOTA_MINIMA, i, aprobados Como Entero
3    Definir suma, promedio Como Real
4    Definir notas Como Real
5    Dimension notas[5]
6
7    CANTIDAD <- 5
8    NOTA_MINIMA <- 61
9    suma <- 0
10   aprobados <- 0
11
12   Para i <- 1 Hasta CANTIDAD Con Paso 1 Hacer
13     Escribir "Nota ", i, ":"
14     Leer notas[i]
15     suma <- suma + notas[i]
16     Si notas[i] >= NOTA_MINIMA Entonces
17       aprobados <- aprobados + 1
18     FinSi
19   FinPara
20
21   promedio <- suma / CANTIDAD
22
23   Escribir "Promedio: ", promedio
24   Escribir "Aprobados: ", aprobados, " de ", CANTIDAD
25 FinAlgoritmo
```

QUÉ COMBINA ESTE ALGORITMO

- Un **arreglo** de 5 posiciones (líneas 4–5).
- Un ciclo `Para`: `i` va de 1 a 5.
- Un **acumulador** (`suma`) y un **contador** (`aprobados`).
- Un `Si` dentro del ciclo.

RECOMENDACIÓN

En cada vuelta, `i` es el número de la caja que se está usando.

Probar el arreglo

Ejemplo con 80, 55, 90, 70 y 45; después use sus propias notas

i	notas[i]	suma	¿>= 61?	aprobados
1	80			
2	55			
3	90			
4	70			
5	45			

NOTA

Anote sus cinco notas y el promedio en la tabla del manual. Verifique el promedio a mano antes de confiar en la pantalla.

RETO

Muestre también la nota **más alta**. Pista: una variable `mayor` que empieza con `notas[1]` y un `Si` dentro del ciclo.

Probar el arreglo

Ejemplo con 80, 55, 90, 70 y 45; después use sus propias notas

i	notas[i]	suma	¿>= 61?	aprobados
1	80	80	sí	1
2	55			
3	90			
4	70			
5	45			

NOTA

Anote sus cinco notas y el promedio en la tabla del manual. Verifique el promedio a mano antes de confiar en la pantalla.

RETO

Muestre también la nota **más alta**. Pista: una variable `mayor` que empieza con `notas[1]` y un `Si` dentro del ciclo.

Probar el arreglo

Ejemplo con 80, 55, 90, 70 y 45; después use sus propias notas

i	notas[i]	suma	¿>= 61?	aprobados
1	80	80	sí	1
2	55	135	no	1
3	90			
4	70			
5	45			

NOTA

Anote sus cinco notas y el promedio en la tabla del manual. Verifique el promedio a mano antes de confiar en la pantalla.

RETO

Muestre también la nota **más alta**. Pista: una variable `mayor` que empieza con `notas[1]` y un `Si` dentro del ciclo.

Probar el arreglo

Ejemplo con 80, 55, 90, 70 y 45; después use sus propias notas

i	notas[i]	suma	¿>= 61?	aprobados
1	80	80	sí	1
2	55	135	no	1
3	90	225	sí	2
4	70			
5	45			

NOTA

Anote sus cinco notas y el promedio en la tabla del manual. Verifique el promedio a mano antes de confiar en la pantalla.

RETO

Muestre también la nota **más alta**. Pista: una variable `mayor` que empieza con `notas[1]` y un `Si` dentro del ciclo.

Probar el arreglo

Ejemplo con 80, 55, 90, 70 y 45; después use sus propias notas

i	notas[i]	suma	¿>= 61?	aprobados
1	80	80	sí	1
2	55	135	no	1
3	90	225	sí	2
4	70	295	sí	3
5	45			

NOTA

Anote sus cinco notas y el promedio en la tabla del manual. Verifique el promedio a mano antes de confiar en la pantalla.

RETO

Muestre también la nota **más alta**. Pista: una variable `mayor` que empieza con `notas[1]` y un `Si` dentro del ciclo.

Probar el arreglo

Ejemplo con 80, 55, 90, 70 y 45; después use sus propias notas

i	notas[i]	suma	¿>= 61?	aprobados
1	80	80	sí	1
2	55	135	no	1
3	90	225	sí	2
4	70	295	sí	3
5	45	340	no	3

NOTA

Anote sus cinco notas y el promedio en la tabla del manual. Verifique el promedio a mano antes de confiar en la pantalla.

RETO

Muestre también la nota **más alta**. Pista: una variable `mayor` que empieza con `notas[1]` y un `Si` dentro del ciclo.

Probar el arreglo

Ejemplo con 80, 55, 90, 70 y 45; después use sus propias notas

i	notas[i]	suma	¿>= 61?	aprobados
1	80	80	sí	1
2	55	135	no	1
3	90	225	sí	2
4	70	295	sí	3
5	45	340	no	3

Pantalla: `Promedio: 68` (340/5) y `Aprobados: 3 de 5`.

NOTA

Anote sus cinco notas y el promedio en la tabla del manual. Verifique el promedio a mano antes de confiar en la pantalla.

RETO

Muestre también la nota **más alta**. Pista: una variable `mayor` que empieza con `notas[1]` y un `Si` dentro del ciclo.

Si PSeInt marca un error

Los errores más comunes de hoy

ERRORES FRECUENTES

- Falta `Entonces`, `Hacer` o el cierre: `FinSi`, `FinMientras`, `FinPara`, `FinSegun`.
- Usar `<-` en una condición: `Si nota <- 61`.
- Olvidar `Leer nota` dentro del `Mientras`: ciclo infinito.
- Pedir `notas[0]` o `notas[6]`.
- No inicializar `suma` o `cantidad` en 0.
- Falta `Dimension` antes de usar el arreglo.

RECOMENDACIÓN

Use la **ejecución paso a paso** de PSeInt: muestra qué línea se ejecuta y cuánto vale cada variable. Es la prueba de escritorio hecha por la computadora.

DIAGRAMA DE FLUJO

Dibuje el diagrama de `P02_promedio.psc` con el botón de PSeInt y ubique el rombo del `Mientras` y la flecha que regresa.

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.

NOTA

Las preguntas combinan la teoría de hoy (`Si`, `Segun`, ciclos y arreglos) con lo que usted ejecutó en PSeInt.

NOMBRE DE LA SALA



Resumen de hoy

- `Si`…`Entonces`…`SiNo`…`FinSi` elige entre dos caminos; sin `SiNo`, entre hacer algo o nada.
- Para rangos se encadenan `Si`; para valores exactos, `Segun`.
- `Mientras` pregunta antes; `Repetir`, al final; `Para` cuenta las vueltas.
- Contador: `c <- c + 1`. Acumulador: `s <- s + x`. Ambos empiezan en 0.
- El **centinela** marca el fin de los datos y no se procesa.
- Un **arreglo** guarda varios datos con un nombre; en PSeInt se numera desde 1.

RECOMENDACIÓN

Antes de salir: copie la carpeta `Practica_02` en su USB o en la nube y súbala a su repositorio `Algoritmos-FE729`.

Próxima sesión

Práctica 3: dividir el trabajo con procedimientos y funciones

LO QUE VIENE

- Partir un problema en módulos: **diseño descendente**.
- `SubProceso`: un bloque que *hace* algo.
- `Funcion`: un bloque que *devuelve* un valor.
- Pasar datos `Por Valor` y `Por Referencia`.

PARA MAÑANA

1. Leer la Práctica 3 del manual.
2. Terminar el reto de la nota más alta, si no lo logró hoy.
3. Subir `Practica_02` a GitHub.

> Esta tarde, en **Electrónica Digital**, un circuito también repetirá algo sin detenerse: el `555` hará destellar un LED.

UNIVERSIDAD RURAL DE GUATEMALA

¡Gracias!

¿Preguntas?

github.com/MynorCifuentes