



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorios de docencia

Laboratorio de Computación Salas A y B

Profesor(a): Karina García Morales

Asignatura: Fundamentos de Programación

Grupo: 22

No de Práctica(s): Práctica 4

Integrante(s): Carlos Eduardo González Villa

*No. de lista o
brigada:* 20

Semestre: Primer Semestre

Fecha de entrega: 10 de septiembre de 2024

Observaciones:

CALIFICACIÓN: _____

Guía práctica de estudio 04: Diagramas de flujo

Objetivo

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

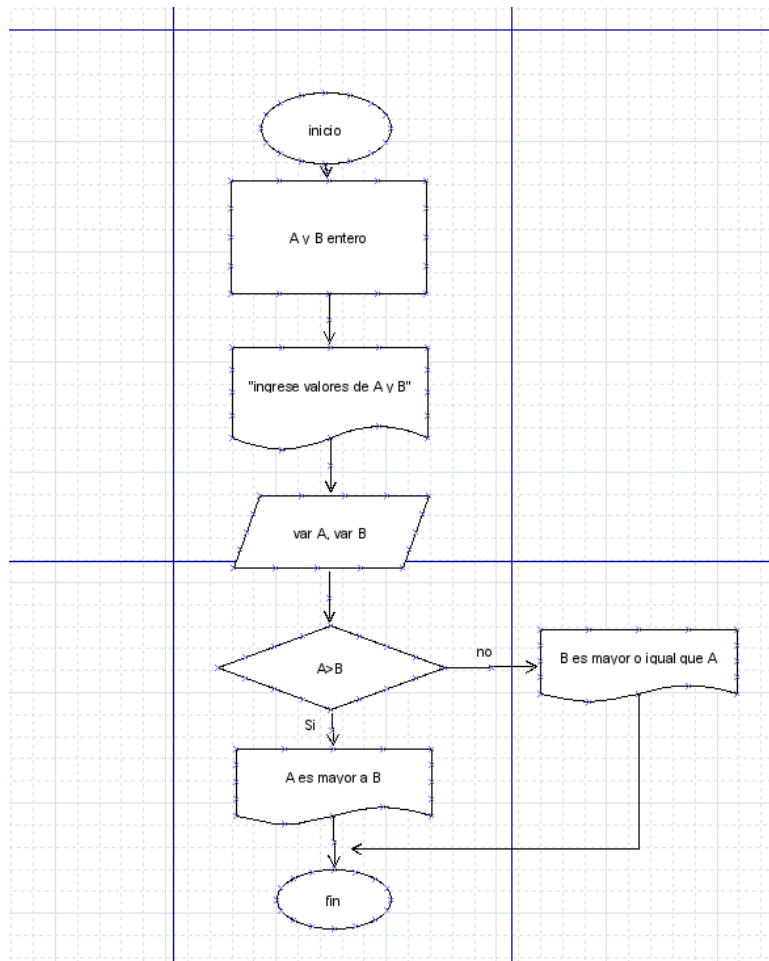
Introducción

Un diagrama de flujo es la representación gráfica de un proceso, es decir, muestra gráficamente el flujo de acciones a seguir para cumplir con una tarea específica.

Dentro de las ciencias de la computación, un diagrama de flujo es la representación gráfica de un algoritmo. La correcta construcción de estos diagramas es fundamental para la etapa de codificación, ya que, a partir del diagrama de flujo es posible codificar un programa en algún lenguaje de programación

Desarrollo

1.- Realizamos un diagrama de flujo que usa la estructura de **control condicional SI-DE LO CONTRARIO** donde, dados dos números enteros asignados a dos variables, muestra la variable que almacena el valor mayor.



2.- Realizamos un diagrama de flujo que usa la estructura de **control condicional SELECCIONAR-CASO** donde, de acuerdo con el valor de la variable a, escribe un mensaje.

En este caso seleccionando una variable seleccionábamos una canción.

```
graph TD
    Start([Proceso Cancion]) --> DefVar[Definir Var Como Entero]
    DefVar --> Msg1[Seleccione una opcion...]
    Msg1 --> Var{Var}
    Var -- 1 --> Accur[Acurrucar]
    Var -- 2 --> TuConEl[TuConEl]
    Var -- 3 --> Afuera[Afuera]
    Var -- De Otro... --> Var
    Accur --> FinProceso([FinProceso])
    TuConEl --> FinProceso
    Afuera --> FinProceso
```

3.- Realizamos un diagrama de flujo que usa la **estructura de control iterativa MIENTRAS** donde se escribe el valor de la variable valor-Inicial mientras se cumpla que el valor de valor-Inicial sea menor al valor de valor-Final

```
1 Algoritmo ValidarNumero
2 Definir numero Como Entero
3 // Solicitar el número al usuario
4 Escribir 'Ingrese un número entre 1 y 10:'
5 Leer numero
6 // Ciclo MIENTRAS que el número esté fuera del rango
7 Mientras numero<1 O numero>10 Hacer
8     Escribir 'Número fuera del rango. Intente nuevamente.'
9     Escribir 'Ingrese un número entre 1 y 10:'
10    Leer numero
11 FinMientras
12 Escribir 'Número válido ingresado: ', numero
13 FinAlgoritmo
14
```

```
graph TD
    Start([Algoritmo ValidarNumero]) --> DefVar[Definir numero Como Entero]
    DefVar --> Msg1[Ingrese un número entre 1 y 10:]
    Msg1 --> Var{numero<1 O numero>10}
    Var -- V --> LoopBody[Número fuera del rango. Intente nuevamente.  
Ingrese un número entre 1 y 10:]
    LoopBody --> Var
    Var -- F --> Msg2[Número válido ingresado: ', numero]
    Msg2 --> FinAlgoritmo([FinAlgoritmo])
```

Tarea

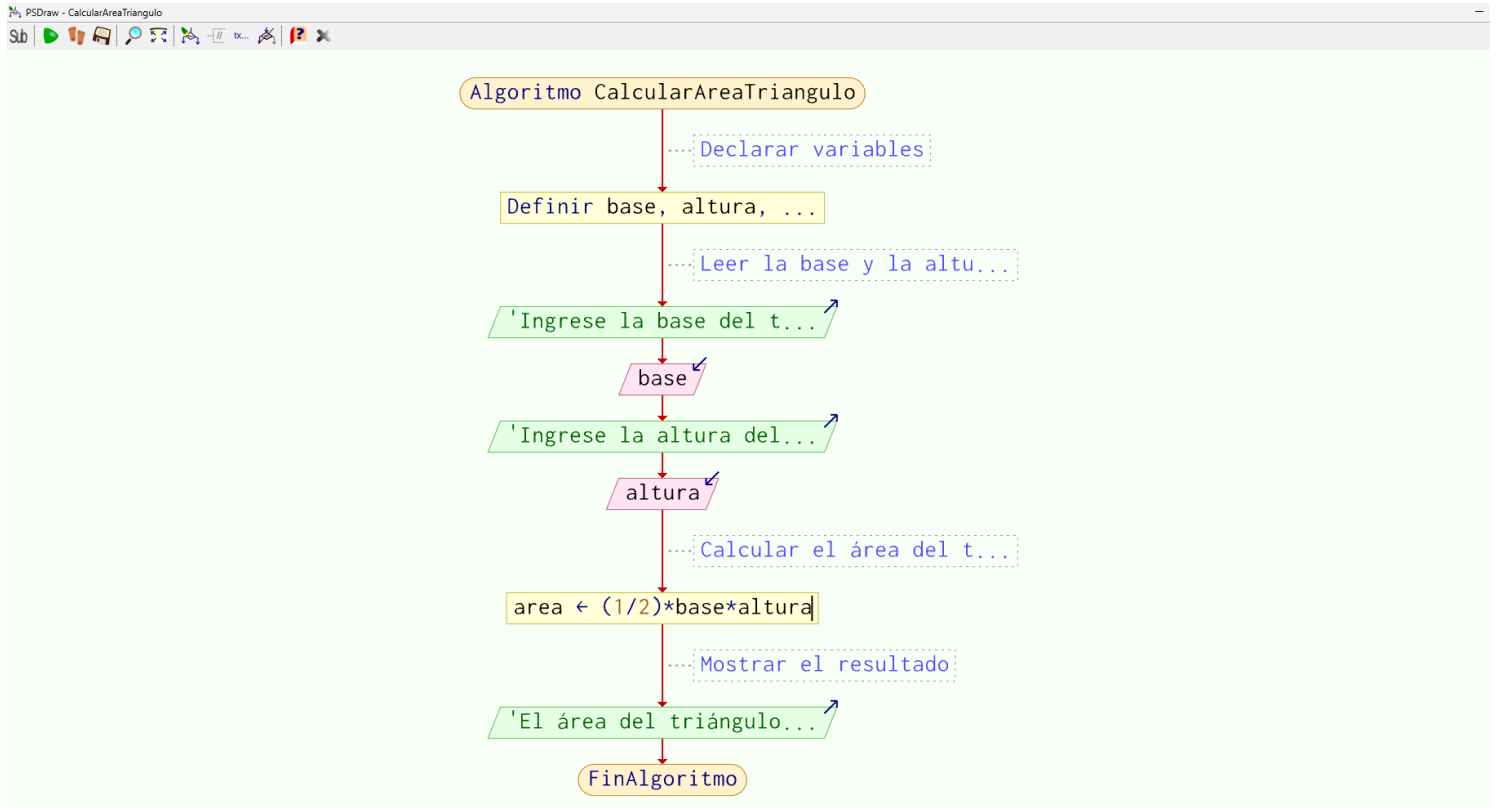
Escriba análisis, diagrama de flujo y 3 pruebas de escritorio, recuerda utilizar por lo menos un valor crítico.

1.- Calcule el área de un triángulo

Entrada de datos: Se leen los valores de la base y la altura del triángulo.

Cálculo del área: Se aplica la fórmula del área del triángulo y se guarda el resultado en la variable área.

Salida del resultado: Se muestra el valor calculado del área en pantalla.



Pruebas de escritorio

```
*** Ejecución Iniciada. ***
Ingrese la base del triángulo:
> 3
Ingrese la altura del triángulo:
> 6
El área del triángulo es:9
*** Ejecución Finalizada. ***
```

```
*** Ejecución Iniciada. ***
Ingrese la base del triángulo:
> 20
Ingrese la altura del triángulo:
> 70
El área del triángulo es:700
*** Ejecución Finalizada. ***
```

```
*** Ejecución Iniciada. ***
Ingrese la base del triángulo:
> 0
Ingrese la altura del triángulo:
> 0
El área del triángulo es:0
*** Ejecución Finalizada. ***
```

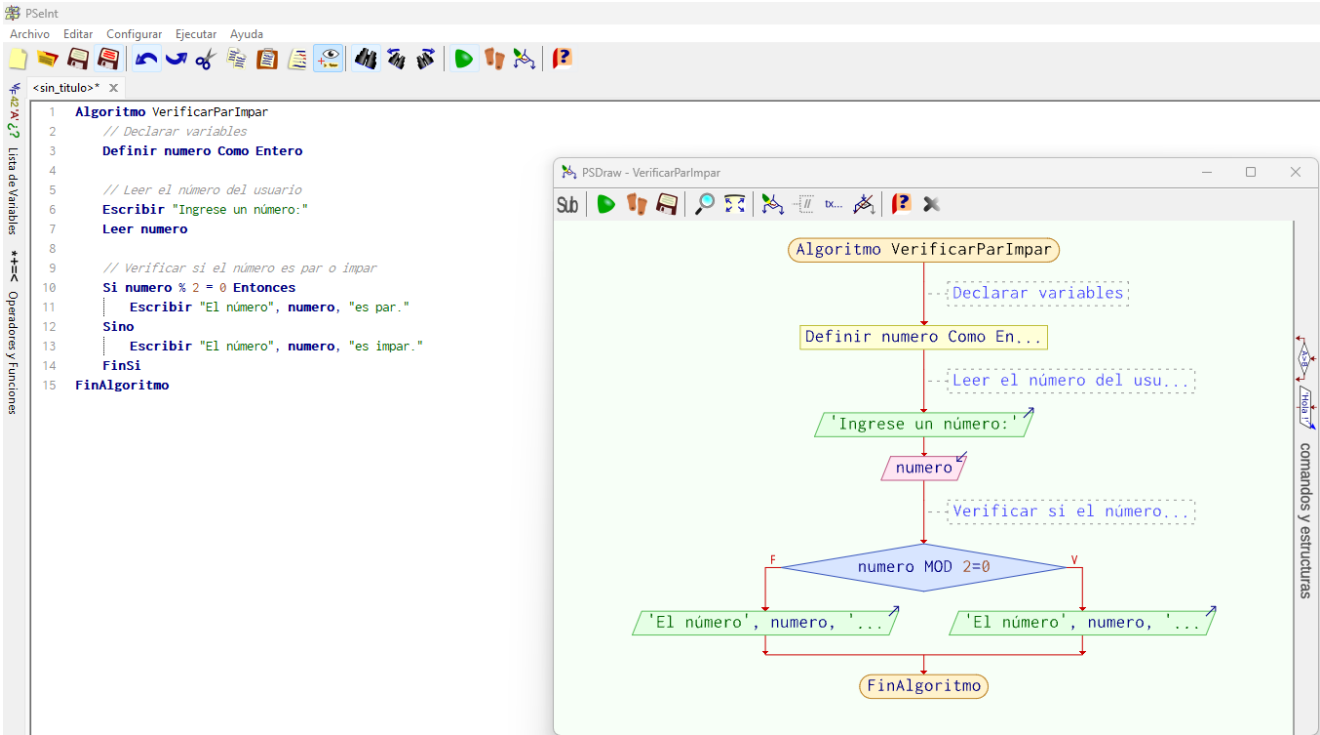
2.- Determine si un número es par (utiliza el %) indicar el número que ingresó, en caso contrario indicar el número y si es impar.

Declaración de variables: Se declara una variable número de tipo entero para almacenar el número ingresado por el usuario.

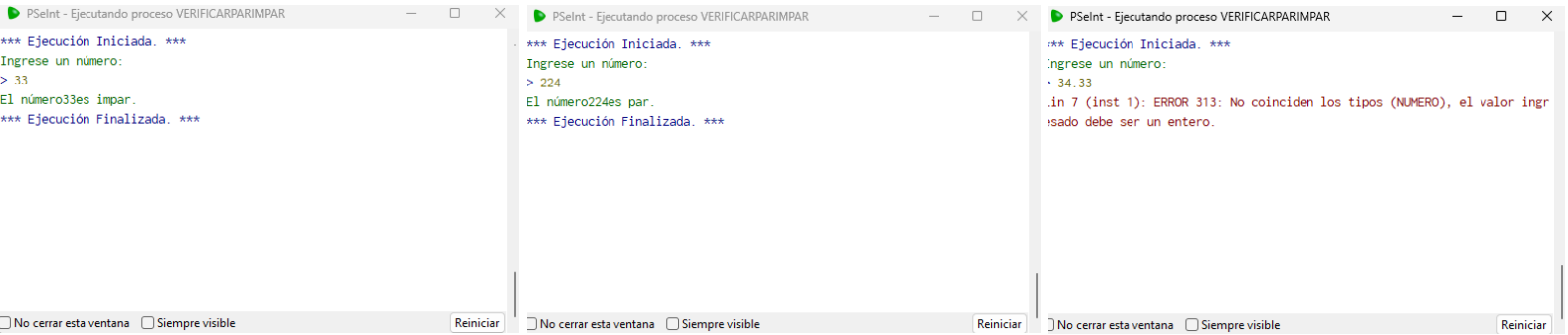
Entrada de datos: Se lee el número ingresado por el usuario.

Verificación: Se utiliza el operador módulo % para determinar si el número es par o impar. Si el resultado de numero \% 2 es 0, el número es par; en caso contrario, es impar.

Salida del resultado: Se muestra un mensaje indicando si el número es par o impar.



Pruebas de escritorio



3.- Solicitar al usuario los centímetros cúbicos para el pago de agua y debe ser mayor a cero, en caso contrario se le debe volver a pedir el valor, los rangos son: De 1 a 500 centímetros cúbicos pagará 5.45 por centímetro cúbico, de 501 a 750 centímetros cúbicos pagará 7.49 por centímetro cúbico y mayor a 750 centímetros cúbicos pagará 10.3 por centímetro cúbico, además si excede los 200 pesos se le aplicará un descuento del %10.

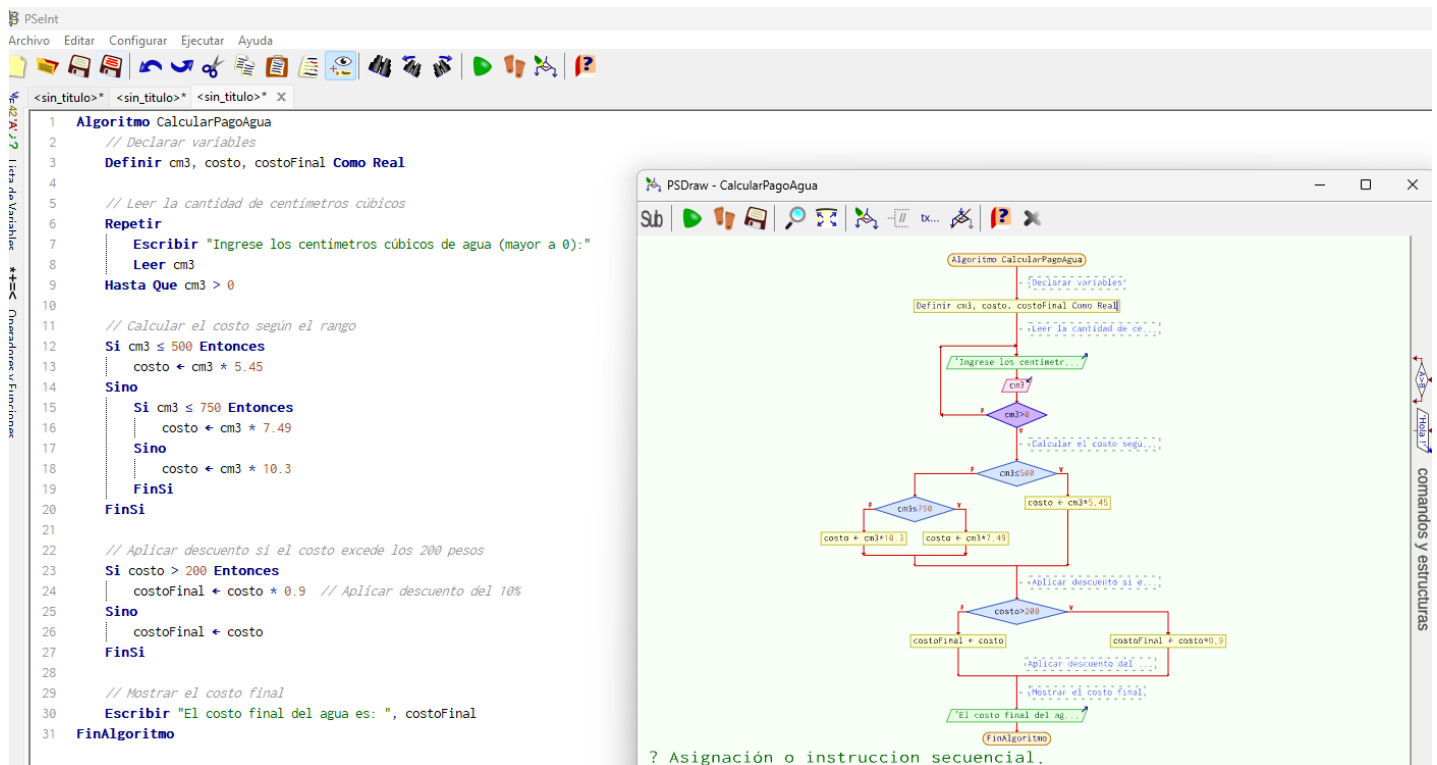
Entrada de datos: Se solicita al usuario que ingrese los centímetros cúbicos de agua y se verifica que el valor sea mayor a 0. Si el valor ingresado es menor o igual a 0, se le vuelve a pedir al usuario que ingrese un valor válido.

Cálculo del costo: Según el rango de centímetros cúbicos, se calcula el costo usando las tarifas correspondientes:

- De 1 a 500 cm³: 5.45 por cm³
- De 501 a 750 cm³: 7.49 por cm³
- Mayor a 750 cm³: 10.3 por cm³

Aplicación de descuento: Si el costo calculado excede los 200 pesos, se aplica un descuento del 10% al costo final.

Salida del resultado: Se muestra el costo final del agua después de aplicar el descuento si corresponde.



Pruebas de escritorio

The following table summarizes the test results shown in the screenshots:

| Input (cm³) | Costo Final (pesos) |
|-------------|---------------------|
| 23 | 125.35 |
| 200 | 981 |
| 233.33 | 1144.48365 |

4.- Genera un menú de los 3 ejercicios anteriores con la estructura **CONDICIONAL MÚLTIPLE** y **HACER MIENTRAS** (el usuario no sale del menú hasta que elija la opción 4 que es salir).

Objetivo: Crear un menú con tres opciones y repetirlo dos veces.

1. Menú de Opciones

- **Entrada:** Opción seleccionada por el usuario.
- **Salida:** Ejecutar la opción correspondiente.

2. Opciones del Menú

1. Área del Triángulo:

- **Entrada:** Base y altura.
- **Proceso:** $\text{Área} = (\text{base} * \text{altura}) / 2$.
- **Salida:** Mostrar el área.

2. Número Par o Impar:

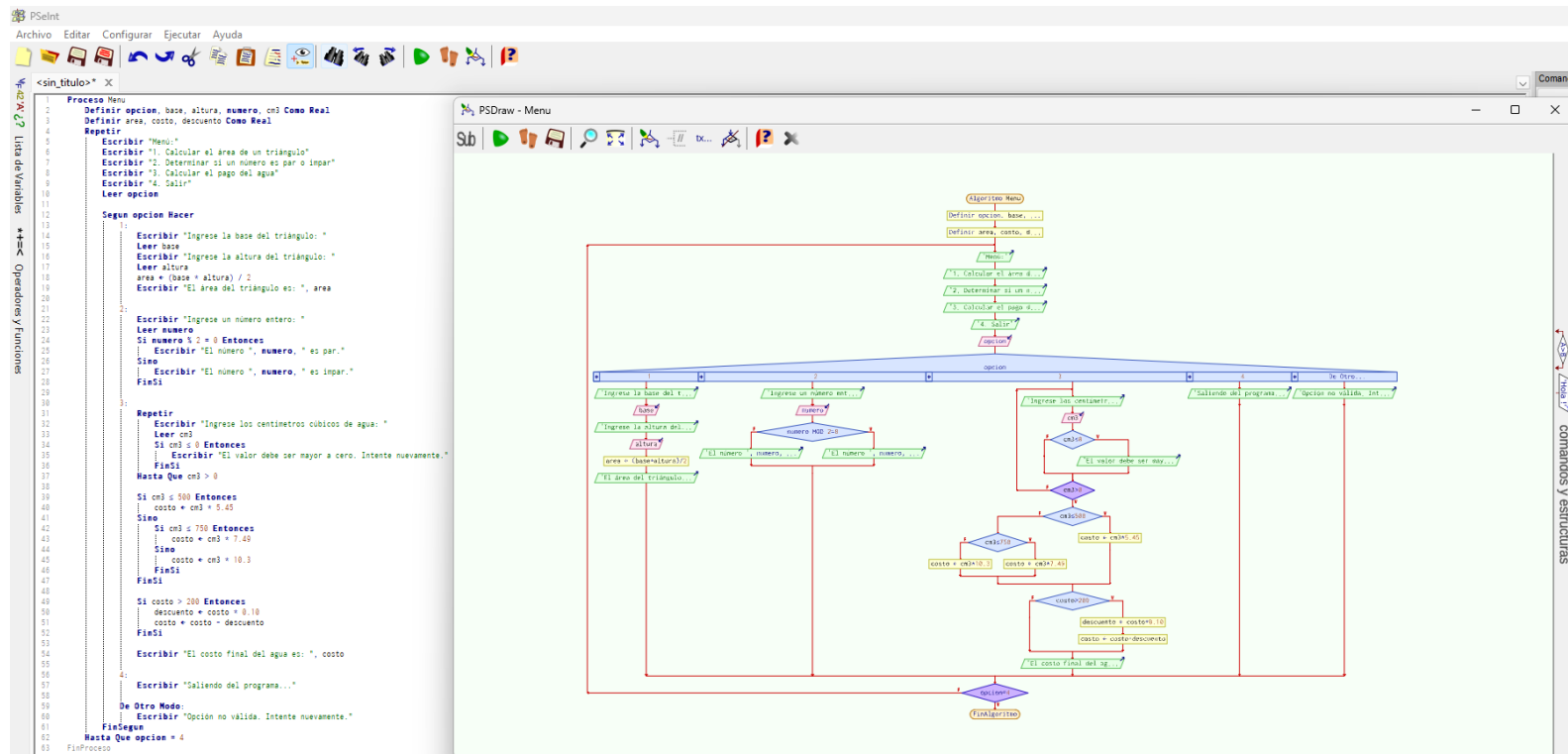
- **Entrada:** Número entero.
- **Proceso:** Verificar $\text{número} \% 2$.
- **Salida:** Indicar si el número es par o impar.

3. Pago del Agua:

- **Entrada:** Centímetros cúbicos.
- **Proceso:**
 - Validar $\text{valor} > 0$.
 - Calcular costo según el rango.
 - Aplicar 10% de descuento si el costo > 200 .
- **Salida:** Mostrar el costo final.

3. Estructura del Programa

- **Repetición:** Utilizar **Mientras** para repetir el menú dos veces.
- **Condicional Anidada:** Manejar las opciones con **Si ... Entonces ... Si no**.



Pruebas de escritorio

```
PSeInt - Ejecutando proceso MENU
2. Determinar si un número es par o impar
3. Calcular el pago del agua
4. Salir
> 1
Ingrese la base del triángulo:
> 22
Ingrese la altura del triángulo:
> 33
El área del triángulo es: 363
Menú:
1. Calcular el área de un triángulo
2. Determinar si un número es par o impar
3. Calcular el pago del agua
4. Salir
>
```

```
PSeInt - Ejecutando proceso MENU
Menú:
1. Calcular el área de un triángulo
2. Determinar si un número es par o impar
3. Calcular el pago del agua
4. Salir
> 3
Ingrese los centímetros cúbicos de agua:
> 35
El costo final del agua es: 190.75
Menú:
1. Calcular el área de un triángulo
2. Determinar si un número es par o impar
3. Calcular el pago del agua
4. Salir
> 1
```

```
PSeInt - Ejecutando proceso MENU
Menú:
1. Calcular el área de un triángulo
2. Determinar si un número es par o impar
3. Calcular el pago del agua
4. Salir
> 2
Ingrese un número entero:
> 33
El número 33 es impar.
Menú:
1. Calcular el área de un triángulo
2. Determinar si un número es par o impar
3. Calcular el pago del agua
4. Salir
>
```

```
PSeInt - Ejecutando proceso MENU
4. Salir
> 2
Ingrese un número entero:
> 33
El número 33 es impar.
Menú:
1. Calcular el área de un triángulo
2. Determinar si un número es par o impar
3. Calcular el pago del agua
4. Salir
> 4
Saliendo del programa...
*** Ejecución Finalizada. ***
[ ] No cerrar esta ventana [ ] Siempre visible [Reiniciar]
```


5.- Genera un menú de los 3 ejercicios anteriores con la estructura condicional anidada, este programa solo se debe repetir 2 veces, utiliza la estructura MIENTRAS o PARA, este programa no necesita la opción de salir.

Objetivo: Crear un menú con tres opciones y repetirlo solo dos veces.

1. Menú de Opciones

- **Entrada:** Opción seleccionada por el usuario.
- **Salida:** Ejecutar la opción correspondiente.

2. Opciones del Menú

1. Área del Triángulo:

- **Entrada:** Base y altura.
- **Proceso:** $\text{Área} = (\text{base} * \text{altura}) / 2$.
- **Salida:** Mostrar el área.

2. Número Par o Impar:

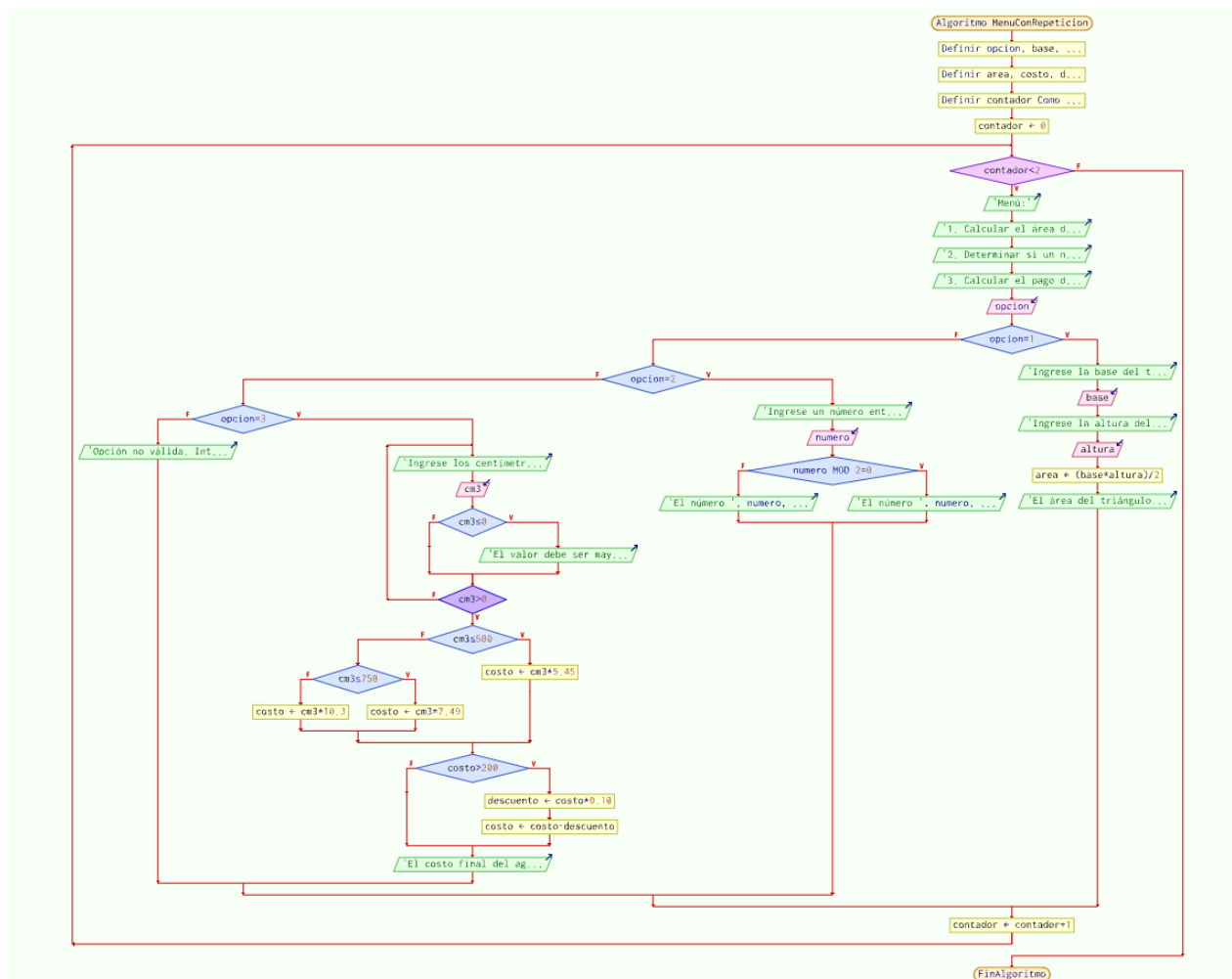
- **Entrada:** Número entero.
- **Proceso:** Verificar $\text{número} \% 2$.
- **Salida:** Indicar paridad del número.

3. Pago del Agua:

- **Entrada:** Centímetros cúbicos.
- **Proceso:**
 - Validar que el valor sea > 0 .
 - Calcular costo basado en el rango.
 - Aplicar 10% de descuento si el costo > 200 .
- **Salida:** Mostrar el costo final.

3. Estructura del Programa

- **Repetición:** Usar Mientras para repetir el menú dos veces.
- **Condicional Anidada:** Usar Si ... Entonces ... Sino para manejar las opciones.



Pruebas de escritorio

*** Ejecución Iniciada. ***

```

Menú:
1. Calcular el área de un triángulo
2. Determinar si un número es par o impar
3. Calcular el pago del agua
> 1
Ingrese la base del triángulo:
> 23
Ingrese la altura del triángulo:
> 3
El área del triángulo es: 34.5
Menú:
1. Calcular el área de un triángulo
2. Determinar si un número es par o impar
3. Calcular el pago del agua
> 3
Ingrese los centímetros cúbicos de agua:
> 33
El costo final del agua es: 179.85
*** Ejecución Finalizada. ***
  
```

*** Ejecución Iniciada. ***

```

Menú:
1. Calcular el área de un triángulo
2. Determinar si un número es par o impar
3. Calcular el pago del agua
> 2
Ingrese un número entero:
> 33
El número 33 es impar.
Menú:
1. Calcular el área de un triángulo
2. Determinar si un número es par o impar
3. Calcular el pago del agua
> 1
Ingrese la base del triángulo:
> 65
Ingrese la altura del triángulo:
> 0707
El área del triángulo es: 22977.5
*** Ejecución Finalizada. ***
  
```

*** Ejecución Iniciada. ***

```

Menú:
1. Calcular el área de un triángulo
2. Determinar si un número es par o impar
3. Calcular el pago del agua
> 3
Ingrese los centímetros cúbicos de agua:
> 54
El costo final del agua es: 264.87
Menú:
1. Calcular el área de un triángulo
2. Determinar si un número es par o impar
3. Calcular el pago del agua
> 2
Ingrese un número entero:
> 3
El número 3 es impar.
*** Ejecución Finalizada. ***
  
```

*** Ejecución Iniciada. ***

```

Menú:
1. Calcular el área de un triángulo
2. Determinar si un número es par o impar
3. Calcular el pago del agua
> 2
Ingrese un número entero:
> 3.3
  
```

Lin 23 (inst 1): ERROR 298: Los operandos para el operador MOD deben ser enteros

Conceptos

Estructuras Condicionales

1. Condicional simple (si)

Esta estructura evalúa una condición, y si es verdadera, se ejecuta un bloque de código. Si es falsa, no se ejecuta nada.

Características:

- Sólo evalúa una condición.
- Si la condición es falsa, el bloque de código es ignorado.

2. Compuesta (si - sino)

Este tipo de condicional tiene dos ramas: si la condición es verdadera, ejecuta un bloque, y si es falsa, ejecuta otro bloque alternativo.

Características:

- Útil cuando hay una acción que debe ejecutarse tanto si la condición es verdadera como si es falsa.

3. Condicional múltiple (según)

Esta estructura permite evaluar múltiples condiciones o casos posibles para una variable. Es similar a una cadena de si - sino, pero más organizada.

Características:

- Útil cuando hay múltiples valores posibles a evaluar.
- Proporciona una manera clara y legible de manejar muchos casos.

Estructuras Iterativas

1. Ciclo mientras (mientras)

Este ciclo evalúa una condición al principio de cada iteración, y si es verdadera, el ciclo se ejecuta. Si es falsa, el ciclo termina.

Características:

- Útil cuando no se sabe de antemano cuántas veces se debe ejecutar el ciclo.
- La condición se evalúa al inicio del ciclo, lo que significa que es posible que el bloque de código no se ejecute ni una sola vez si la condición es falsa desde el principio.

2. Ciclo hacer-mientras (hacer-mientras)

Este ciclo es similar al ciclo mientras, pero la condición se evalúa al final de cada iteración, lo que garantiza que el ciclo se ejecute al menos una vez.

Características:

- Útil cuando se necesita que el ciclo se ejecute al menos una vez, independientemente de la condición inicial.

3. Ciclo para (para)

Este ciclo es ideal cuando se sabe de antemano cuántas veces se debe ejecutar el código. Tiene un contador que se incrementa o disminuye en cada iteración hasta que se cumple una condición.

Características:

- Útil para contar repeticiones específicas o recorrer rangos.
- Se define un inicio, una condición final, y un incremento o decremento.

Conclusión

El uso de la aplicación PSeInt para creación de algoritmos, nos ha permitido comprender mejor cómo resolver problemas de manera lógica y ordenada. A lo largo de la práctica, se ha aprendido a descomponer problemas complejos en pasos más pequeños y manejables, lo que ha facilitado la búsqueda de soluciones a ciertos errores que podemos cometer durante la creación del algoritmo.

Bibliografía

Bustos, J. L. (2024, 25 abril). ¿Qué es la estructura iterativa en programación? KeepCoding Bootcamps. <https://keepcoding.io/blog/que-es-estructura-iterativa-programacion/>

Sancho, R. S. (s. f.). Estructuras Iterativas · ciencia-de-datos-con-r. https://rsanchezs.gitbooks.io/ciencia-de-datos-con-r/content/estructuras_control/iterativas/estructuras_iterativas.html

Estructuras condicionales. (s. f.). DesarrolloWeb.com. <https://desarrolloweb.com/articulos/2225.php>

Estructuras condicionales simples y compuestas. (s. f.). <https://www.tutorialesprogramacionya.com/javaya/detalleconcepto.php?codigo=80>

Símbolos de diagramas de flujo. (s. f.). <https://www.smartdraw.com/flowchart/simbolos-de-diagramas-de-flujo.htm>