

Carrera: Gestión en redes y Telecomunicaciones

Asignatura: Algoritmos Fundamentales EXAC40

MSc. José Alberto Torres Ortega

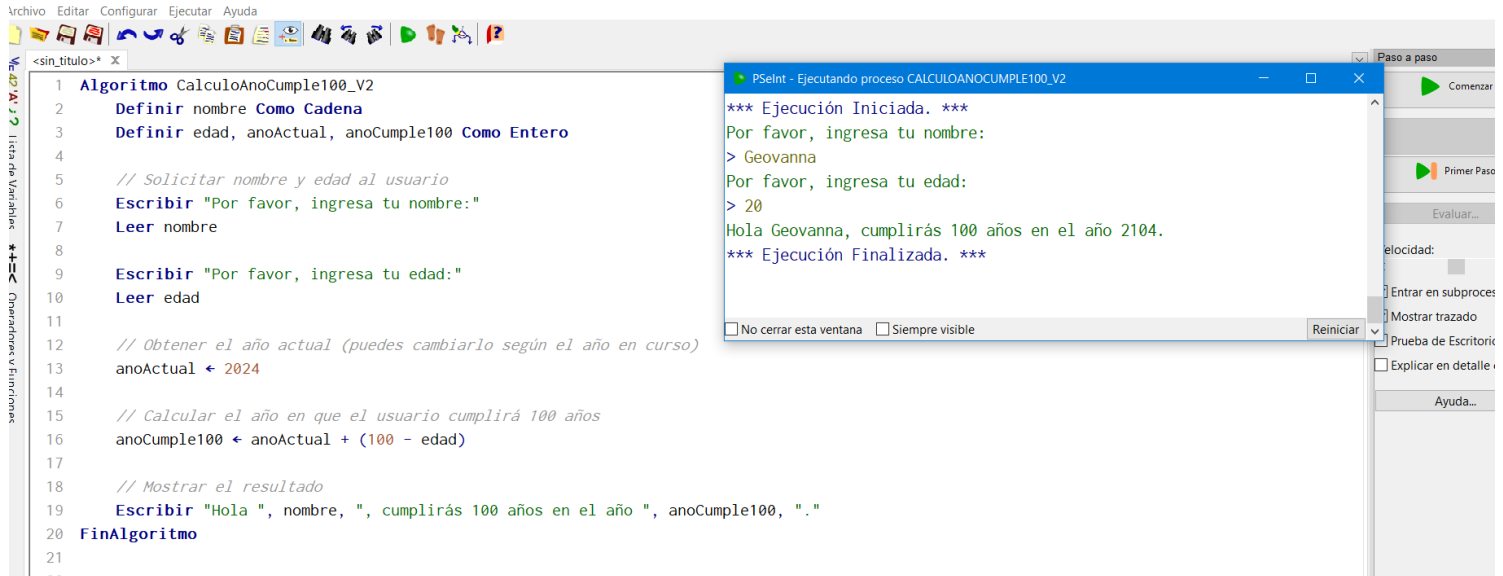
Nombre: Geovanna Pincay Arreaga

Fecha de Entrega: 1 de sep. de 24

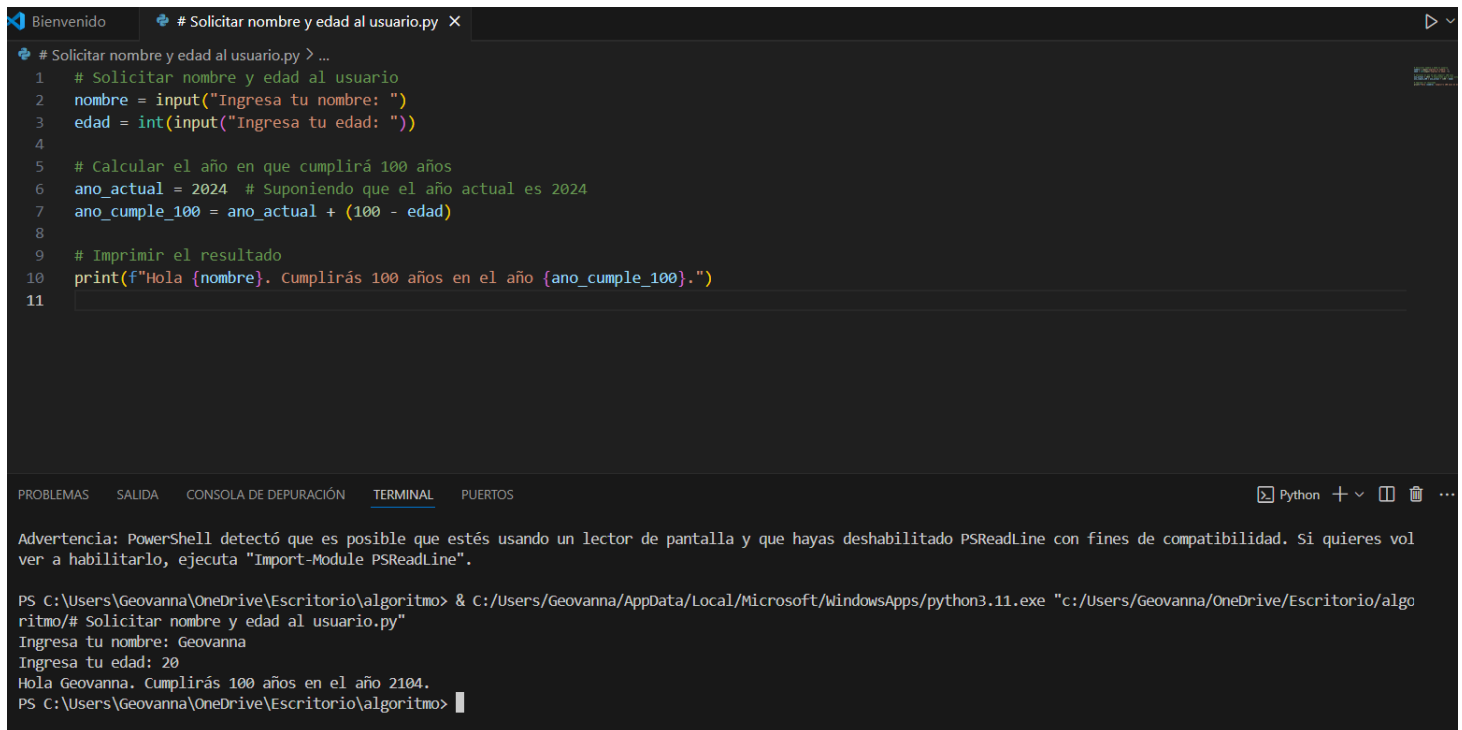
Objetivo: Fortalecer la comprensión y aplicación de los conceptos de estructura principal, tipos de datos, constantes, variables, identificadores, y operaciones básicas en PSEINT y Python. Esta tarea contribuirá al desarrollo de habilidades necesarias para abordar con éxito la Tarea Experimental propuesta.

Instrucciones para el Trabajo Autónomo:

- **Estructura Principal de un Programa:**
 - Escribe un programa sencillo en PSEINT y Python que realice las siguientes operaciones:
 - Solicitar al usuario su nombre y edad.
 - Calcular el año en el que el usuario cumplirá 100 años.
 - Imprimir el resultado en pantalla, siguiendo la sintaxis y estructura específica de cada lenguaje.



PSEINT

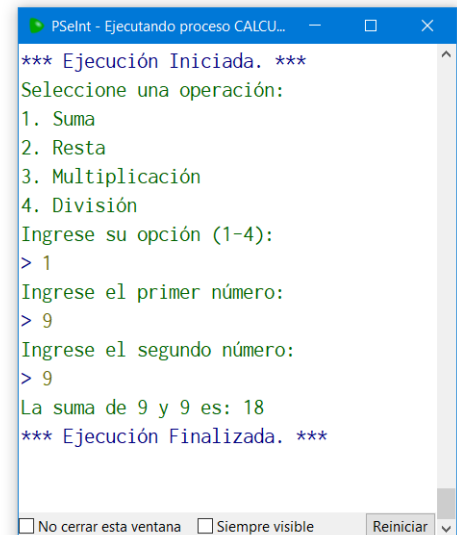


Visual Code

- **Tipos de Datos, Constantes y Variables:**

- En los dos lenguajes, declara diferentes tipos de variables (entero, flotante, cadena de texto) y constantes.
- Crea un programa que use estas variables y constantes para realizar una serie de cálculos básicos (suma, resta, multiplicación, división).
- Asegúrate de utilizar identificadores descriptivos para todas las variables y constantes.

```
1 Algoritmo CalculadoraConMenu
2 // Declaración de variables
3 Definir num1, num2, resultado Como Real
4 Definir opcion Como Entero
5
6 // Mostrar menú de opciones al usuario
7 Escribir "Seleccione una operación:"
8 Escribir "1. Suma"
9 Escribir "2. Resta"
10 Escribir "3. Multiplicación"
11 Escribir "4. División"
12 Escribir "Ingrese su opción (1-4):"
13 Leer opcion
14
15 // Solicitar al usuario que ingrese los números para la operación
16 Escribir "Ingrese el primer número:"
17 Leer num1
18 Escribir "Ingrese el segundo número:"
19 Leer num2
20
21 // Realizar la operación según la opción seleccionada
22 Segun opcion Hacer
23     1:
24         resultado ← num1 + num2
25         Escribir "La suma de ", num1, " y ", num2, " es: ", resultado
26     2:
27         resultado ← num1 - num2
28         Escribir "La resta de ", num1, " menos ", num2, " es: ", resultado
29     3:
30         resultado ← num1 * num2
31         Escribir "La multiplicación de ", num1, " por ", num2, " es: ", resultado
32     4:
33         Si num2 ≠ 0 Entonces
34             resultado ← num1 / num2
35             Escribir "La división de ", num1, " entre ", num2, " es: ", resultado
36         SiNo
37             Escribir "Error: No se puede dividir entre cero."
38         FinSi
39 De Otro Modo:
40     Escribir "Opción no válida. Por favor, ingrese una opción entre 1 y 4."
41 FinSegun
42 FinAlgoritmo
43
```



```
*** Ejecución Iniciada. ***
Seleccione una operación:
1. Suma
2. Resta
3. Multiplicación
4. División
Ingrese su opción (1-4):
> 1
Ingrese el primer número:
> 9
Ingrese el segundo número:
> 9
La suma de 9 y 9 es: 18
*** Ejecución Finalizada. ***
```

☐ No cerrar esta ventana ☐ Siempre visible

```
import math.py > ...

2  numeros = [12, 45, 67, 23, 89, 34, 56] # Lista de números
3
4  # Cálculo del promedio
5  suma_numeros = sum(numeros)
6  cantidad_numeros = len(numeros)
7  promedio = suma_numeros / cantidad_numeros
8
9  # Determinación del número máximo y mínimo
10 numero_maximo = max(numeros)
11 numero_minimo = min(numeros)
12
13 # Mostrar resultados
14 print(f"Lista de números: {numeros}")
15 print(f"Promedio de los números: {promedio:.2f}")
16 print(f"Número máximo: {numero_maximo}")
17 print(f"Número mínimo: {numero_minimo}")
18 |
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS Python + v

Advertencia: PowerShell detectó que es posible que estés usando un lector de pantalla y que hayas deshabilitado PSReadLine con fines de compatibilidad. Si quieres volver a habilitarlo, ejecuta "Import-Module PSReadLine".

PS C:\Users\Geovanna\OneDrive\Escritorio\algoritmo> & C:/Users/Geovanna/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/Geovanna/OneDrive/Escritorio/algoritmo/import math.py"

Lista de números: [12, 45, 67, 23, 89, 34, 56]
Promedio de los números: 46.57
Número máximo: 89
Número mínimo: 12

PS C:\Users\Geovanna\OneDrive\Escritorio\algoritmo>

• Operaciones Básicas y Expresiones:

- Implementa un programa en los dos lenguajes que realice las siguientes operaciones:
 - Solicitar dos números al usuario.
 - Realizar las cuatro operaciones aritméticas básicas (suma, resta, multiplicación y división).
 - Utilizar operadores lógicos para comparar los resultados y mostrar mensajes en pantalla dependiendo de la comparación (por ejemplo, si la suma es mayor que la multiplicación).

Algoritmo OperacionesBasicas

```
// Declaración de variables
Definir num1, num2, suma, resta, multiplicacion, division Como Real

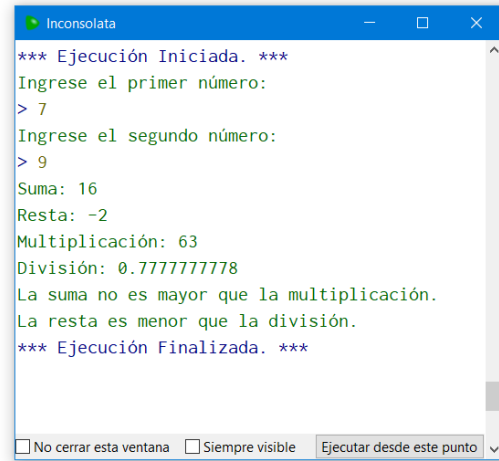
// Solicitar dos números al usuario
Escribir "Ingrese el primer número:"
Leer num1
Escribir "Ingrese el segundo número:"
Leer num2

// Realizar las cuatro operaciones aritméticas básicas
suma ← num1 + num2
resta ← num1 - num2
multiplicacion ← num1 * num2
Si num2 ≠ 0 Entonces
    division ← num1 / num2
SiNo
    division ← 999999999 // Usar un valor alto para representar infinito
FinSi

// Mostrar resultados de las operaciones
Escribir "Suma: ", suma
Escribir "Resta: ", resta
Escribir "Multiplicación: ", multiplicacion
Escribir "División: ", division

// Comparaciones y mensajes
Si suma > multiplicacion Entonces
    Escribir "La suma es mayor que la multiplicación."
SiNo
    Escribir "La suma no es mayor que la multiplicación."
FinSi

Si resta < division Entonces
    Escribir "La resta es menor que la división."
SiNo
    Escribir "La resta no es menor que la división."
FinSi
FinAlgoritmo
```



```
1 # Solicitar dos números al usuario
2 num1 = float(input("Ingrese el primer número: "))
3 num2 = float(input("Ingrese el segundo número: "))
4
5 # Realizar las cuatro operaciones aritméticas básicas
6 suma = num1 + num2
7 resta = num1 - num2
8 multiplicacion = num1 * num2
9 # Evitar división por cero
10 if num2 != 0:
11     division = num1 / num2
12 else:
13     division = float('inf') # Representa la división por cero como infinito
14
15 # Mostrar resultados de las operaciones
16 print(f"Suma: {suma}")
17 print(f"Resta: {resta}")
18 print(f"Multiplicación: {multiplicacion}")
19 print(f"División: {division}")
20
21 # Comparaciones y mensajes
22 if suma > multiplicacion:
23     print("La suma es mayor que la multiplicación.")
24 else:
25     print("La suma no es mayor que la multiplicación.")
26
27 if resta < division:
28     print("La resta es menor que la división.")
29 else:
30     print("La resta no es menor que la división.")
31
```

```
La resta es menor que la división.
PS C:\Users\Geovanna\OneDrive\Escritorio\algoritmo> & C:/Users/Geovanna/OneDrive/Escritorio/algoritmo/citar dos números al usuario.py
Ingrese el primer número: 9
Ingrese el segundo número: 8
Suma: 17.0
Resta: 1.0
Multiplicación: 72.0
División: 1.125
La suma no es mayor que la multiplicación.
La resta es menor que la división.
PS C:\Users\Geovanna\OneDrive\Escritorio\algoritmo>
```

Entregable:

- Sube los tres programas (PSEINT y Python) en un repositorio compartido.
- Realiza un breve informe explicando la elección de tipos de datos, el uso de constantes y variables, y cómo las operaciones y estructuras fueron implementadas en cada lenguaje.
- El informe deberá redactarse en formato PDF, y el nombre del documento debe tener el siguiente formato: *AUTÓNOMO2_apellido_nombre.pdf*

Realiza un breve informe explicando la elección de tipos de datos, el uso de constantes y variables, y cómo las operaciones y estructuras fueron implementadas en cada lenguaje.

En el mundo de la programación, el uso efectivo de tipos de datos, constantes, variables y estructuras de control es esencial para el desarrollo de aplicaciones precisas y eficientes. En este ensayo, exploraremos cómo dos lenguajes de programación populares, Python y PSeInt, abordan la implementación de un programa que realiza operaciones aritméticas básicas y compara los resultados. Analizaremos las elecciones de tipos de datos, el uso de constantes y variables, y la implementación de operaciones y estructuras en cada lenguaje, proporcionando una visión integral de cómo estos conceptos se traducen en código funcional.

Elección de Tipos de Datos

Python es un lenguaje versátil que permite una gran flexibilidad en el manejo de tipos de datos. En el contexto del programa para realizar operaciones aritméticas básicas, se opta por el

tipo `float` para las variables involucradas. El tipo `float` es adecuado porque puede representar números decimales, lo que es crucial para manejar el resultado de operaciones como la división, que no siempre produce un número entero. Por ejemplo, al calcular la división de dos números, el resultado puede ser un número decimal, y el tipo `float` asegura que este resultado se maneje con precisión.

En contraste, PSeInt, una herramienta diseñada para la enseñanza de algoritmos y programación, utiliza el tipo `Real` para manejar números decimales. Al igual que en Python, el tipo `Real` en PSeInt es esencial para representar resultados que no son enteros, asegurando que los cálculos se realicen correctamente. Ambos lenguajes, por lo tanto, hacen uso de tipos de datos que permiten una representación precisa de números reales, lo que es fundamental para realizar cálculos aritméticos precisos.

Uso de Constantes y Variables

El uso de constantes y variables en programación es crucial para almacenar y manipular datos de manera efectiva. En Python, las variables `num1` y `num2` se utilizan para almacenar los números ingresados por el usuario, mientras que `las variables suma, resta, multiplicacion, y division` almacenan los resultados de las respectivas operaciones aritméticas. Aunque en el ejemplo no se definen constantes explícitas, se utiliza la función `float('inf')` para manejar la división por cero, lo que actúa como una representación simbólica de infinito.

En PSeInt, el enfoque es similar. Se utilizan variables para almacenar los números ingresados (`num1` y `num2`) y para guardar los resultados de las operaciones (suma, resta, multiplicación, división). Aunque PSeInt no emplea constantes de la misma manera que Python, se utiliza un valor alto (`999999999`) para representar infinito en el caso de la división por cero. Este enfoque refleja cómo las variables pueden adaptarse para manejar situaciones especiales y representar conceptos abstractos como infinito en un contexto programático.

Implementación de Operaciones y Estructuras

La implementación de operaciones aritméticas en Python y PSeInt demuestra cómo los lenguajes de programación manejan cálculos y estructuras de control. En Python, se utilizan operadores aritméticos estándar (`+`, `-`, `*`, `/`) para realizar las operaciones básicas. La estructura condicional `if-else` se emplea para manejar la división por cero, una situación que no se puede resolver con una simple operación matemática. En este caso, si el divisor es cero, se asigna a `division` el valor `float('inf')`, que representa infinito. Esta solución asegura que el programa pueda manejar casos de división por cero sin interrumpir su ejecución.

En PSeInt, las operaciones aritméticas se realizan utilizando operadores similares. Sin embargo, el manejo de la división por cero se aborda mediante una estructura condicional `Si`, que verifica si el divisor es cero antes de realizar la operación. Si el divisor es `cero`, se asigna un valor alto a la variable `division`, actuando como una representación simbólica de infinito. Esta solución es efectiva para evitar errores de ejecución y permitir que el programa continúe funcionando correctamente.

Las comparaciones entre los resultados de las operaciones se manejan de manera similar en ambos lenguajes. En Python, las estructuras **if-else** se utilizan para comparar los resultados de las operaciones y mostrar mensajes apropiados basados en estas comparaciones. Por ejemplo, se compara si la suma es mayor que la multiplicación y si la resta es menor que la división. En PSeInt, la estructura Si cumple una función similar, permitiendo la ejecución de diferentes bloques de código basados en las condiciones evaluadas.

Conclusión

La implementación de un programa para realizar operaciones aritméticas básicas en Python y PSeInt revela cómo cada lenguaje maneja los conceptos de tipos de datos, constantes y variables, así como la implementación de operaciones y estructuras de control. Ambos lenguajes muestran una aproximación similar en cuanto a la representación de números reales y el manejo de errores como la división por cero, aunque utilizan diferentes técnicas y sintaxis para lograr estos objetivos. La elección de tipos de datos adecuados y la correcta utilización de variables y estructuras de control son fundamentales para el desarrollo de programas precisos y funcionales. A través de este análisis, se aprecia la flexibilidad y adaptabilidad de los lenguajes de programación para abordar problemas comunes y garantizar la robustez en la ejecución de cálculos y comparaciones.