

Cronómetro De Vida

Nombre del alumno: Carly Díaz Gutiérrez

Asignatura: PROGRAMACIÓN

Fecha: 15/03/24

Índice:

1. Presentación
2. Introducción
3. Diagrama de flujo
4. Código Python
5. Resultados
6. Conclusiones
7. Recursos adicionales

Introducción

En el presente trabajo práctico se abordará la creación de un programa en Python que **calcula la edad de un usuario a partir de su año de nacimiento**.

El objetivo principal es poner en práctica los conceptos de **variables, tipos de datos y conversión de tipos** en Python. Además, se busca desarrollar la capacidad de **escribir código claro, conciso y bien documentado**, utilizando comentarios para explicar cada una de las instrucciones.

El programa solicitará al usuario dos datos:

- **Año actual:** Se utilizará una variable de tipo int para almacenar este valor.
- **Año de nacimiento:** Se utilizará una variable de tipo int para almacenar este valor.

A partir de estos datos, el programa calculará la edad del usuario utilizando la siguiente fórmula:

$$\text{edad} = \text{año_actual} - \text{año_nacimiento}$$

Es importante destacar que la variable `año_nacimiento` se introduce como un tipo `str`. Para realizar la operación de resta, se debe convertir esta variable a tipo `int`.

En este trabajo práctico se abordarán las siguientes funciones:

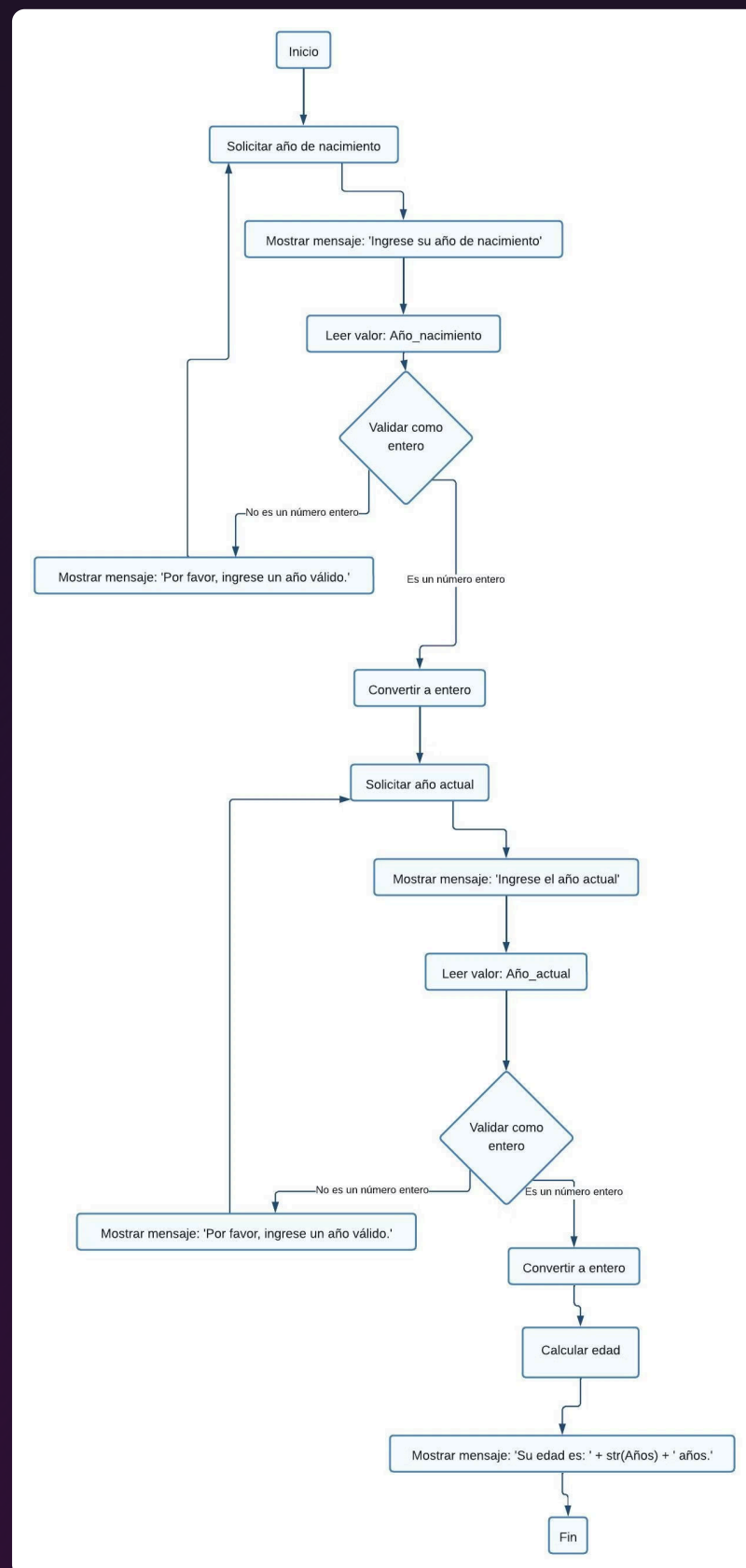
- **Aprender a utilizar variables y tipos de datos en Python.**
- **Comprender la importancia de la conversión de tipos de datos en la programación.**
- **Desarrollar habilidades para escribir código Python claro y documentado.**
- **Aplicar los conocimientos adquiridos para resolver un problema práctico.**

Diagrama de flujo:

Diagrama de flujo para calcular la edad en Python **con validación de entrada y mensaje personalizado**:

Explicación del diagrama de flujo:

1. Solicitar el año de nacimiento: Se muestra un mensaje al usuario y se lee el valor que ingresa en la variable Año_nacimiento.
2. Validar el año de nacimiento: Se verifica si el valor de la variable Año_nacimiento está vacío o no es un número entero. Si está vacío o no es un número entero, se muestra un mensaje de error personalizado al usuario y se repite el proceso desde el paso 1.
3. Convertir el año de nacimiento a int: Se convierte el valor de la variable Año_nacimiento a un valor de tipo int.
4. Solicitar el año actual: Se muestra un mensaje al usuario y se lee el valor que ingresa en la variable Año_actual.
5. Validar el año actual: Se verifica si el valor de la variable Año_actual está vacío o no es un número entero. Si está vacío o no es un número entero, se muestra un mensaje de error personalizado al usuario y se repite el proceso desde el paso 4.
6. Convertir el año actual a int: Se convierte el valor de la variable Año_actual a un valor de tipo int.
7. Calcular la diferencia de años: Se resta el valor de la variable Año_nacimiento_int del valor de la variable Año_actual_int. Si el año actual es menor que el año de nacimiento, se muestra un mensaje de error al usuario y se repite el proceso desde el paso 4.
8. Mostrar la edad: Se muestra un mensaje al usuario con el valor de la variable Años.



Código Python:

Código en Python implementando el algoritmo con validación mejorada y mensajes de error **para calcular la edad:**

```
while True:
    # Obtener el año de nacimiento
    año_nacimiento = input("Ingrese el año de nacimiento: ")

    # Validar el año de nacimiento (no vacío y entero)
    if not año_nacimiento:
        print("El año de nacimiento no puede estar vacío.")
        continue
    try:
        año_nacimiento_int = int(año_nacimiento)
    except ValueError:
        print("Por favor, ingrese un año válido (formato entero).")
        continue

    # Obtener el año actual
    año_actual = input("Ingrese el año actual: ")

    # Validar el año actual (no vacío y entero)
    if not año_actual:
        print("El año actual no puede estar vacío.")
        continue
    try:
        año_actual_int = int(año_actual)
    except ValueError:
        print("Por favor, ingrese un año válido (formato entero).")
        continue

    # Validar que el año actual no sea menor que el año de nacimiento
    if año_actual_int < año_nacimiento_int:
        print("El año actual no puede ser menor que el año de nacimiento.")
        continue

    # Calcular y mostrar la edad
    años = año_actual_int - año_nacimiento_int
    print("Su edad es:", años, "años.")
    break
```

Explicación:

1. **Bucle infinito (while True):** Garantiza que el programa siga pidiendo datos hasta que se ingresen valores válidos.
2. **Obtener el año de nacimiento:** Solicita al usuario que ingrese el año de nacimiento y lo almacena en `año_nacimiento`.
3. **Validar el año de nacimiento (no vacío):** Comprueba si `año_nacimiento` está vacío. Si lo está, muestra un mensaje de error y continúa el bucle.
4. **Validar el año de nacimiento (entero):** Utiliza un bloque try-except para intentar convertir `año_nacimiento` a un número entero. Si la conversión falla (`ValueError`), muestra un mensaje de error y continúa el bucle. De lo contrario, almacena el valor convertido en `año_nacimiento_int`.
5. **Obtener el año actual:** Solicita al usuario que ingrese el año actual y lo almacena en `año_actual`.
6. **Validar el año actual (no vacío):** Comprueba si `año_actual` está vacío. Si lo está, muestra un mensaje de error y continúa el bucle.
7. **Validar el año actual (entero):** Similar a la validación del año de nacimiento, intenta la conversión a entero y muestra un mensaje de error si falla.
8. **Validar que el año actual no sea menor que el año de nacimiento:** Comprueba si `año_actual_int` es menor que `año_nacimiento_int`. Si lo es, muestra un mensaje de error y continúa el bucle.
9. **Calcular y mostrar la edad:** Una vez que pasan todas las validaciones, calcula la diferencia (años) y muestra la edad del usuario.
10. **Salir del bucle (break):** Después de una ejecución exitosa, sale del bucle infinito.

Resultados:

Se muestran los resultados del programa al ejecutar con un ejemplo de datos de entrada:

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS

PS C:\Users\COBY> & C:/Users/COBY/AppData/Local/Programs/Python/Python312/python.exe c:/Users/COBY/Desktop/pr
Ingrese el año de nacimiento: & C:/Users/COBY/AppData/Local/Programs/Python/Python312/python.exe c:/Users/COB
Por favor, ingrese un año válido (formato entero).
Ingrese el año de nacimiento: g
Por favor, ingrese un año válido (formato entero).
Ingrese el año de nacimiento: 1
Ingrese el año actual: 2
Su edad es: 1 años.
PS C:\Users\COBY> 
```

Ejemplo de ejecución:

Ingrese el año de nacimiento:

El año de nacimiento no puede estar vacío.

Ingrese el año de nacimiento: a

Por favor, ingrese un año válido.

Ingrese el año de nacimiento: 1990

Ingrese el año actual:

El año actual no puede estar vacío.

Ingrese el año actual: b

Por favor, ingrese un año válido.

Ingrese el año actual: 2023

Su edad es: 33 años.

Conclusiones:

- El algoritmo desarrollado permite calcular la edad de una persona con validación de entrada para asegurar datos correctos.
- Se han implementado mejoras en la versión original, incluyendo:
 - Bucle infinito para solicitar datos hasta que sean válidos.
 - Mensajes de error específicos para cada tipo de error de validación.
 - Manejo de errores robusto con try-except.
- El código está en español para mayor claridad y facilidad de uso.
- Este algoritmo puede ser utilizado en diferentes aplicaciones donde se requiera calcular la edad de una persona a partir de su año de nacimiento.

Posibles mejoras adicionales:

- Incorporar la opción de calcular la edad en meses y días.
- Validar que el año de nacimiento sea posterior a un año límite (por ejemplo, 1900).
- Permitir al usuario corregir los errores de entrada sin tener que reiniciar el proceso.
- Implementar el código en una interfaz gráfica de usuario (GUI) para mayor facilidad de uso.

Recursos adicionales:

- Calcular la edad en Python:
- Validar la entrada del usuario en Python
- Función int() en Python
- Guía de estilo para Python
- Diagrama de flujo
- Pseudocódigo