

Trabajo Práctico N° 1

Semana de Integración I

Alumnos - Grupo n° 4

Caballero, Julieta – Bongiorno, Santiago Martin - Beisel, Alan Elias – Barrutia, Milagros

Tecnicatura Universitaria en Programación - Universidad Tecnológica Nacional.

Matemática

Docente Titular

Vanina Durrutty

Docente Tutor

Augusto Bertuzzi Gaspari

23 de abril de 2025

Trabajo Práctico N°01

Consigna

Conversión de Números:

Desarrollen un programa que convierta números decimales a binarios y, de forma opcional, también de binario a decimal.

Extensión: Validar la entrada y mostrar mensajes de error ante datos incorrectos.

Desarrollo

[GitHub - Tp grupal de Matemática y Programación I](#)

Este programa fue diseñado con el objetivo de convertir números entre los sistemas numéricos decimal y binario. El usuario tiene la posibilidad de elegir entre tres opciones: convertir un número decimal a binario, convertir un número binario a decimal, o salir del programa. Además, incluye un menú interactivo y validaciones que garantizan que el usuario ingrese valores válidos antes de realizar cualquier conversión.

Al principio del código y en todas las opciones se implementará un bucle **while True**, que es un bucle infinito. Esto significa que el programa se ejecutará hasta que el usuario decida salir. Lo usamos para que el programa no finalice cuando se convierta un número o haya un error, y tener la opción de terminar cuando se elige la opción de salir.

```
while True:
    print("Elija una opción (1, 2 o 3):")
    print("1. Pasar de decimal a binario")
    print("2. Pasar de binario a decimal")
    print("3. Salir del programa")

    eleccion = input()
```

Dentro de este bucle, el programa muestra un menú con tres opciones que el usuario puede elegir:

- **Opción 1:** Convertir de decimal a binario.
- **Opción 2:** Convertir de binario a decimal.
- **Opción 3:** Salir del programa.

El valor ingresado por el usuario se almacena en una variable llamada **eleccion**, y el flujo del programa se adapta según la opción seleccionada.

Opción 1: Convertir de decimal a binario

```
if eleccion == '1':
    while True:
        numero_decimal = input("Ingresá un número decimal positivo: ")

        if numero_decimal != "":
            for num in numero_decimal:
                if num not in '0123456789':
                    print("Número decimal inválido! Coloque números del 0 al 9.")
                    break
            else:
                decimal = int(numero_decimal)
                binario = bin(decimal)[2:]
                print("El número", decimal, "en binario es:", binario)
                print()
                break
```

En esta opción solicita al usuario un número decimal positivo. Primero, se verifica que el campo no esté vacío. Luego, se hace una validación con un **ciclo for** para asegurarse de que todos los caracteres del número ingresado sean dígitos válidos (del 0 al 9). En caso de error, el programa muestra un mensaje y vuelve a pedir el número. Si todo está bien, el programa convierte el número de texto a número entero con la función **int()**, y luego lo convierte a binario usando la función **bin()**. Esta función devuelve el número binario con un prefijo '0b', y para eliminar este prefijo usamos el método de *slicing* [2:] que corta esos dos primeros caracteres. El resultado final se muestra en pantalla, y luego se sale del bucle con **break**, lo que permite regresar al menú principal para elegir otra opción.

Opción 2: Convertir de binario a decimal

```
elif eleccion == '2':
    while True:
        numero_binario = input("Ingresá un número binario (0 y 1): ")

        if numero_binario != "":
            for num in numero_binario:
                if num != '0' and num != '1':
                    print("Número binario inválido! Coloque solo 0 y 1.")
                    break
            else:
                decimal = int(numero_binario, 2)
                print("El número binario", numero_binario, "en decimal es:", decimal)
                print()
                break
```

La opción 2 funciona de manera similar a la opción anterior. Se valida que el número binario ingresado no esté vacío, y luego se recorre cada carácter para verificar que sea un 0 o un 1. Si se encuentra algún otro valor, se muestra un mensaje de error y el programa pide de nuevo el número binario. Una vez validado el número binario, se convierte a decimal utilizando la función **int()**, que en este caso recibe dos argumentos: el número binario como texto y el número 2, que indica que el número está en base 2. De esta manera, el programa convierte el número binario a decimal. El resultado se imprime en la pantalla, y luego el programa regresa al menú principal.

Opción 3: Salir del programa

```
elif eleccion == '3':  
    print("Saludos!")  
    break
```

La opción 3 le permite al usuario salir del programa. Si se elige esta opción, el programa muestra un mensaje de despedida y termina el bucle principal mediante un **break**, lo que finaliza la ejecución del programa.

```
else:  
    print("Opción no válida! Elegí 1, 2 o 3.")  
    print()
```

Por último, si el usuario ingresa una opción que no es 1, 2 o 3 el programa lo detecta y le muestra un mensaje indicando que la opción no es válida. Luego, vuelve a mostrar el menú para que el usuario pueda elegir una opción correcta

Implementación de la IA

Utilizamos la IA para implementar el **while True**, ya que confeccionando el código nos surgió la idea de que el programa no finalice cuando se convierta un número o haya un error, sino cuando el usuario desea.

La IA nos sugirió utilizar ese método, por lo que consultamos sobre el uso del break ya que en los libros sobre “código limpio” no es una buena práctica. Esto fue lo que nos contestó:

*“¡Muy buena observación! Y tenés razón en lo que decís: según principios de **código limpio** (como los de Robert C. Martin en Clean Code), el uso de break, continue o return en medio de estructuras de control puede **dificultar la legibilidad y hacer que el flujo del programa sea más difícil de seguir**, especialmente en funciones largas o con lógica compleja. Tenés razón en desconfiar del uso del break, y es genial que lo señales. En casos chicos o didácticos como este, puede usarse sin mayor problema. Pero en producción o proyectos grandes, sí conviene evitarlo y optar por estructuras más declarativas y legibles.”*