

## Conversor ASM a Binario (MIPS)

Este programa es una herramienta desarrollada en Python para convertir instrucciones en ensamblador MIPS a su representación binaria de 32 bits. Incluye una interfaz gráfica simple usando Tkinter que permite cargar archivos .ASM, convertir su contenido a binario, visualizar el resultado y guardarlo en un archivo de texto.

### Características

- Soporta instrucciones MIPS de tipo **R**, **I** y **J**.
- Traduce instrucciones como ADD, LW, BEQ, J, entre muchas otras.
- Soporta notación de registros por número (\$4) o por alias (\$A0, \$T1, etc.).
- Muestra errores detallados si las instrucciones no son válidas.
- Guarda la salida binaria en formato de 8 bits por línea, hasta 1000 líneas (8×1000 bits).
- Interfaz gráfica amigable construida con tkinter.

### Instrucciones Soportadas

- **Tipo R:** ADD, SUB, SLT, OR, AND, NOP
- **Tipo I:** SW, LW, ADDI, SUBI, SLTI, ANDI, ORI, BEQ, BNE, BGTZ
- **Tipo J:** J

### Requisitos

- Python 3.x
- Tkinter (ya viene incluido con la mayoría de distribuciones de Python)

### Cómo usar

1. Clona o descarga este repositorio.
2. Ejecuta el script principal:
3. En la ventana que se abre:
  - Usa **"Cargar ASM"** para seleccionar un archivo .ASM con instrucciones MIPS.
  - Haz clic en **"Convertir a Binario"** para traducir el contenido.
  - Revisa la salida binaria en la sección inferior.

- Guarda la salida si lo deseas con "**Guardar Binario**".

### **Formato del archivo ASM**

Cada línea debe contener una única instrucción MIPS válida. Se permite usar comentarios al inicio de línea con // o ;. Ejemplos válidos:

asm

ADD \$t0, \$t1, \$t2

LW \$s1, 16(\$s0)

BEQ \$s0, \$s1, -4

J 1024

NOP

### **Manejo de errores**

- Se detectan errores de sintaxis o registros inválidos.
- Los errores se muestran en una ventana emergente indicando la línea afectada.