

Writeup – Misc: “~~100~~ (unicode magic)”

Objetivo

Recuperar la flag a partir del único artefacto entregado: un `README.md` que contiene aparentemente **solo un emoji**:

- ~~100~~

A simple vista no hay nada más, pero el truco está en que el archivo incluye **caracteres invisibles Unicode**.

Idea / Pista clave

Muchos retos “misc” esconden información usando **Unicode Variation Selectors**:

- Son caracteres “invisibles” que modifican cómo se renderiza un carácter previo (por ejemplo un emoji).
- En texto plano quedan pegados al final del emoji y no se ven.
- Se suelen usar como canal encubierto para **codificar bytes**.

En este reto, el emoji ~~100~~ funciona como **portador**, y después vienen una serie de Variation Selectors invisibles que codifican la flag.

Identificación (cómo darse cuenta)

Formas típicas de verlo:

Opción 1: Ver los codepoints con Python

Leyendo el archivo en UTF-8 y mostrando `ord()` (o `hex(ord())`) aparecen caracteres con rangos como:

- U+FE00..U+FE0F
- U+E0100..U+E01EF

Eso delata Variation Selectors.

Opción 2: Herramientas CLI (opcional)

Con utilidades como `xxd`, `hexdump -C` o editores que muestran Unicode invisible, se ve que el archivo NO es solo el emoji.

Cómo se decodifica

Los Variation Selectors se mapean a valores numéricos (0..255), que luego se interpretan como bytes ASCII:

Mapeo usado

- `U+FE00..U+FE0F` → valores 0..15
- `U+E0100..U+E01EF` → valores 16..255

La conversión para la segunda familia:

- `valor = codepoint - 0xE0100 + 16`

Luego, cada `valor` se convierte a carácter con `chr(valor)` y se concatena.

Script de solución (Python)

Guarda este script como `solve.py` en la misma carpeta del `README.md`

```
#!/usr/bin/env python3

def decode_variation_selectors(text: str) -> str:
    if not text:
        return ""

    # El primer carácter es el emoji portador (💯). Lo ignoramos.
    payload = text[1:]

    vals = []
    for ch in payload:
        cp = ord(ch)

        # Variation Selectors (básicos)
        if 0xFE00 <= cp <= 0xFE0F:
            vals.append(cp - 0xFE00)
```

```

# Variation Selectors suplementarios (los que más se usan para
esconder bytes)
elif 0xE0100 <= cp <= 0xE01EF:
    vals.append(cp - 0xE0100 + 16)

# Si aparece cualquier otro carácter, normalmente se ignora.
# (En algunos retos se valida o se trata como separador.)
else:
    pass

return "".join(chr(v) for v in vals)

def main():
    with open("README.md", "r", encoding="utf-8") as f:
        s = f.read().strip()

    decoded = decode_variation_selectors(s)
    print(decoded)

if __name__ == "__main__":
    main()

```

Finalmente la flag es:

ENO{EM0J1S_UN1C0D3_1S_MAG1C}