

Instituto Tecnológico de San Juan del Río



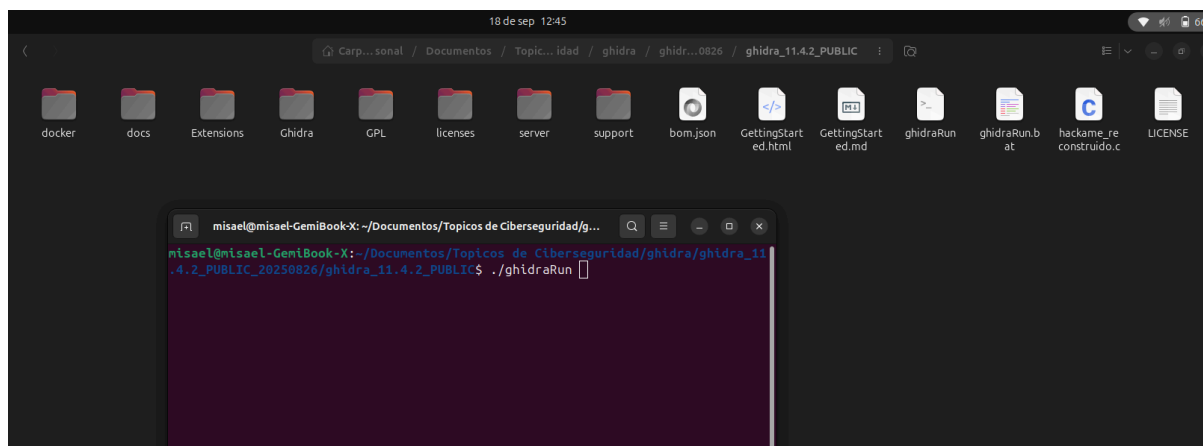
Temas de Ciber Seguridad

Ghidra código fuente

PERIODO

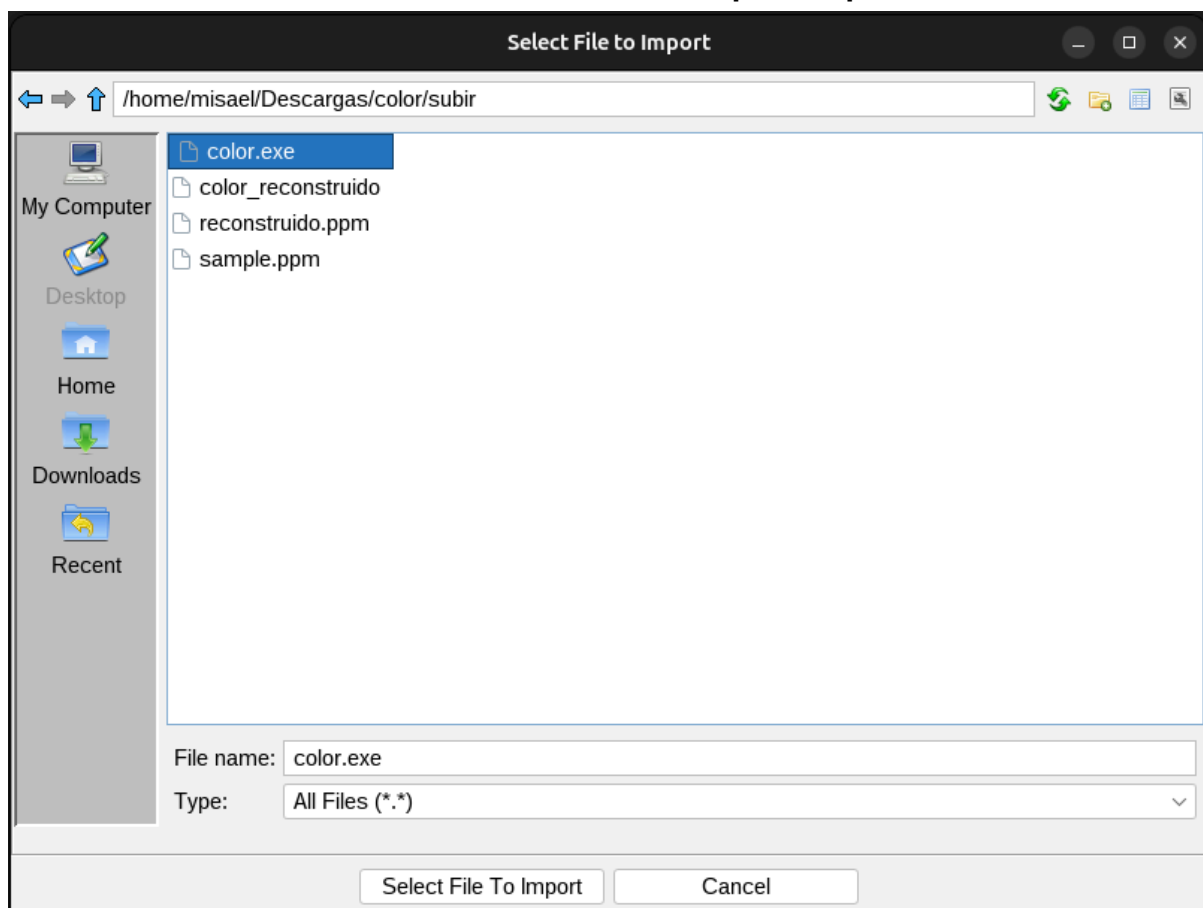
AGOSTO- DICIEMBRE 2025

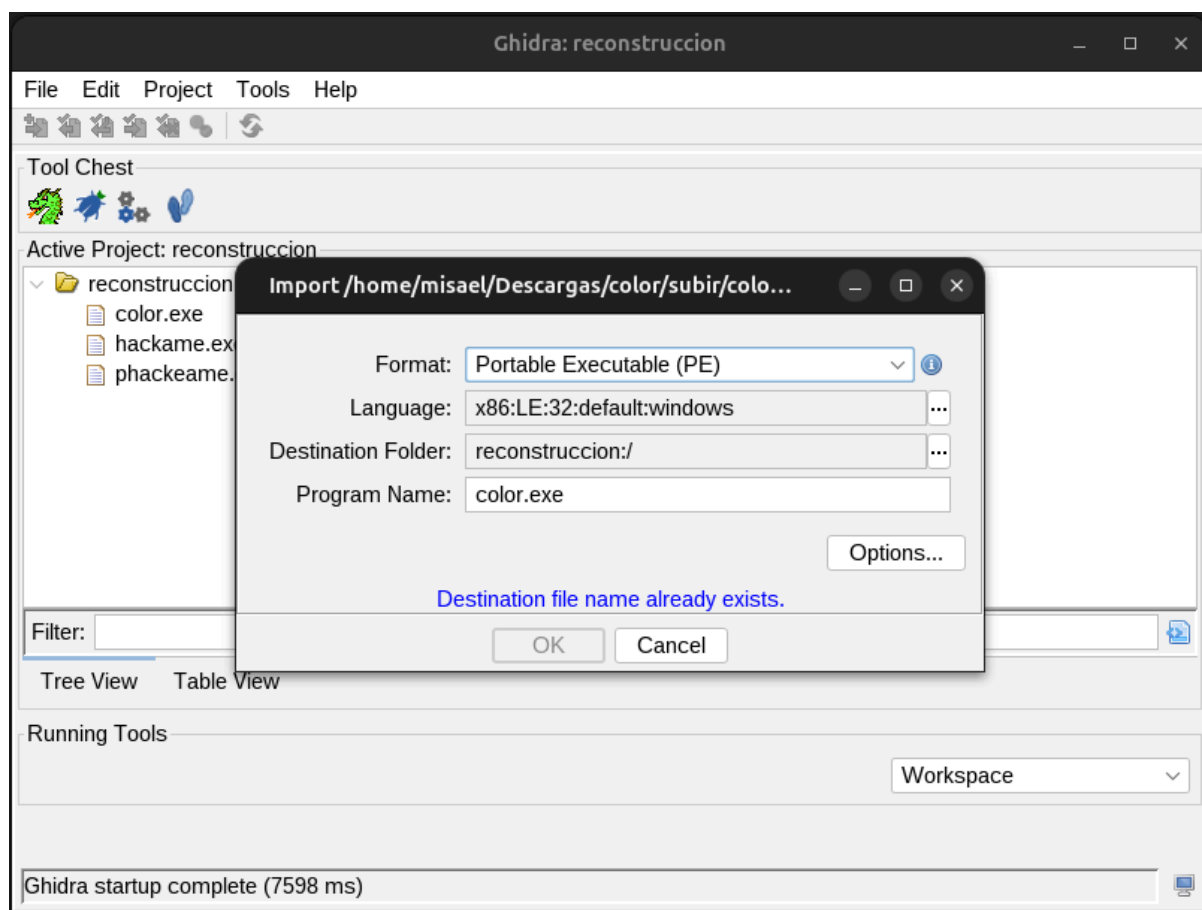




abrimos ghidra con el comando **./ghidraRun**

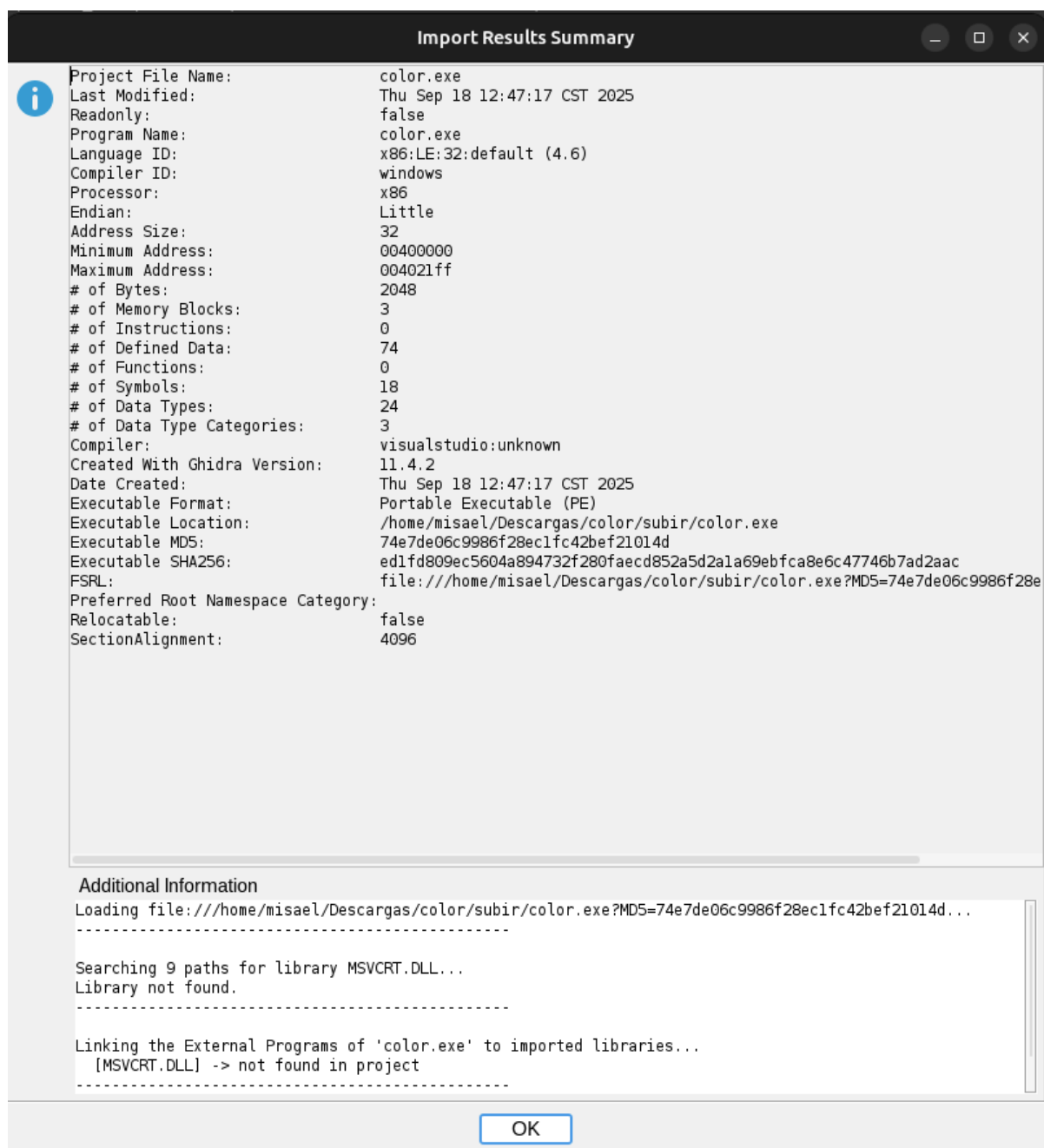
Inicio de Ghidra mediante el comando **./ghidraRun** en terminal de Linux, mostrando la ventana de selección de archivos para importar.



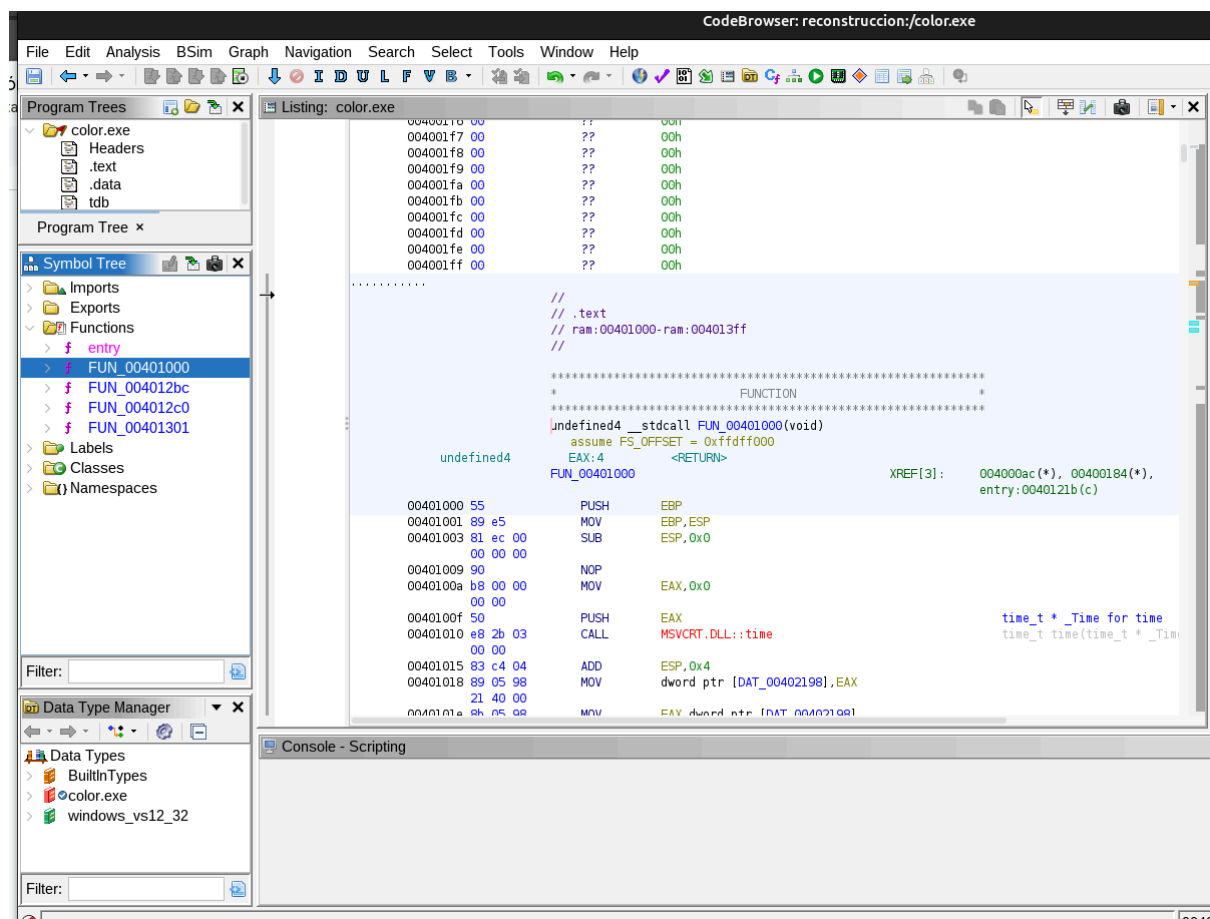
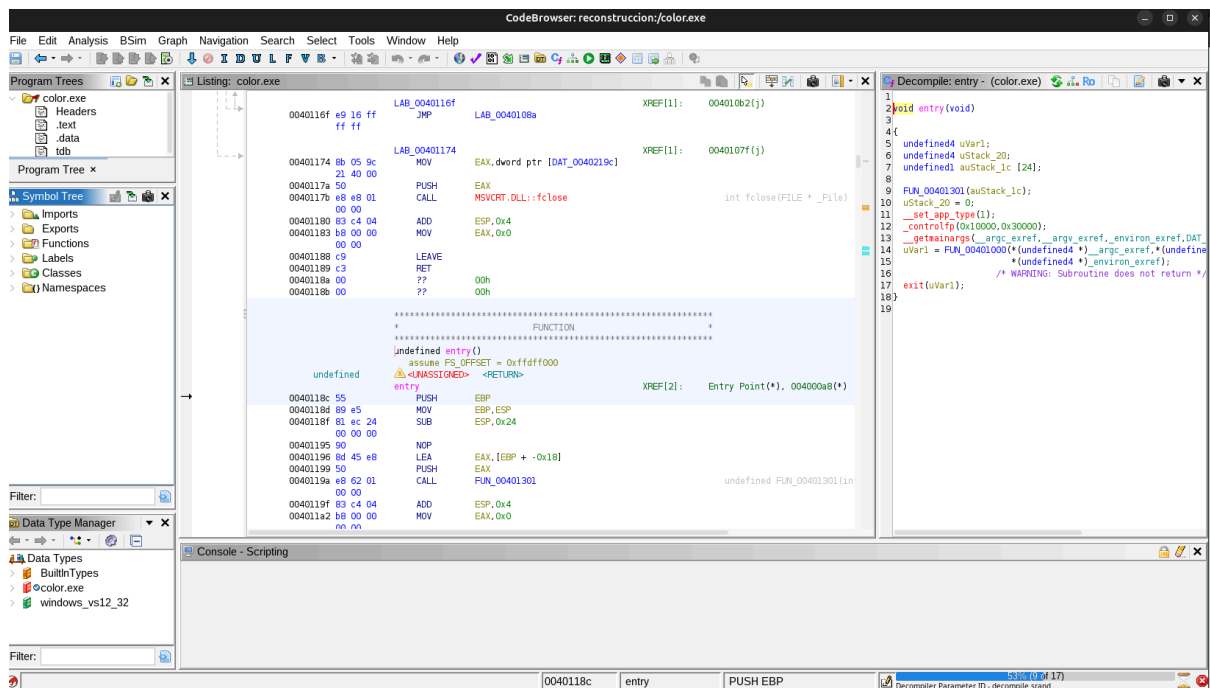


Interfaz gráfica de Ghidra mostrando:

- Proyecto activo: "reconstruction"
- Archivos importados: **color.exe**, **hackame.exe**, **phackeame**
- Opciones de importación para el archivo **color.exe**
- Formato: Portable Executable (PE) para Windows
- Lenguaje: x86 32-bit Little Endian

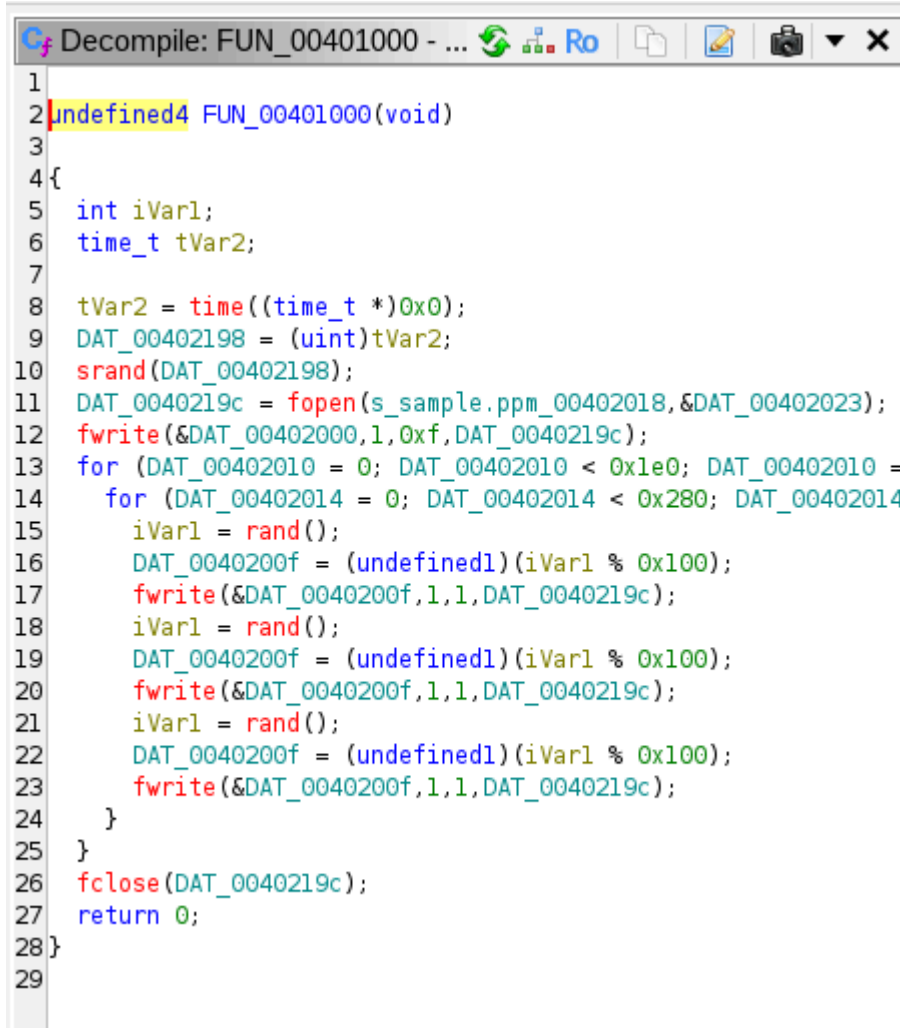


- **Información del ejecutable:** 2048 bytes, formato PE
- **Metadatos:** Compilador Visual Studio, checksum MD5
- **Advertencia:** No se encontró la librería MSVCRT.DLL (normal para análisis)



Descripción: Encabezado que indica el inicio del código fuente descompilado de la función **FUN_00401000**.

codigo fuente:



```
1
2 undefined4 FUN_00401000(void)
3
4 {
5     int iVar1;
6     time_t tVar2;
7
8     tVar2 = time((time_t *)0x0);
9     DAT_00402198 = (uint)tVar2;
10    srand(DAT_00402198);
11    DAT_0040219c = fopen(s_sample.ppm_00402018,&DAT_00402023);
12    fwrite(&DAT_00402000,1,0xf,DAT_0040219c);
13    for (DAT_00402010 = 0; DAT_00402010 < 0x1e0; DAT_00402010 =
14        for (DAT_00402014 = 0; DAT_00402014 < 0x280; DAT_00402014
15            iVar1 = rand();
16            DAT_0040200f = (undefined1)(iVar1 % 0x100);
17            fwrite(&DAT_0040200f,1,1,DAT_0040219c);
18            iVar1 = rand();
19            DAT_0040200f = (undefined1)(iVar1 % 0x100);
20            fwrite(&DAT_0040200f,1,1,DAT_0040219c);
21            iVar1 = rand();
22            DAT_0040200f = (undefined1)(iVar1 % 0x100);
23            fwrite(&DAT_0040200f,1,1,DAT_0040219c);
24        }
25    }
26    fclose(DAT_0040219c);
27    return 0;
28 }
29
```

Descripción: ¡IMAGEN MÁS IMPORTANTE! Código fuente reconstruido por Ghidra que muestra:

- Función principal **FUN_00401000** que genera imágenes PPM
- Uso de **time()** y **srand()** para números aleatorios
- Apertura de archivo **sample.ppm** en modo escritura binaria
- Escritura del header PPM (15 bytes)
- Bucles anidados para generar píxeles RGB aleatorios (640x480)
- Cierre adecuado del archivo