



UNIVERSIDAD AUTÓNOMA TOMÁS FRÍAS
CARRERA DE INGENIERÍA DE SISTEMAS

ESTUDIANTE: Univ. Rafael Antonio Patricio Ayllón

MATERIA: Arquitectura de Computadoras

SIGLA: SIS-522

DOCENTE: Ing. Gustavo Puita

PRACTICA N°9

AUXILIAR: Univ. Aldrin Roger Pérez Miranda

GRUPO: 1

Responda a las siguientes preguntas de MANERA CONCISA

1) ¿Qué es el 'stack' en el contexto del lenguaje ensamblador y cómo se utiliza?

R.- En el contexto del lenguaje ensamblador, el 'stack' es una estructura de datos de tipo LIFO (Last In, First Out) utilizada para almacenar información temporalmente. Es fundamental para la gestión de subrutinas (funciones), almacenamiento de variables locales, y manejo de registros.

Uso del stack:

1. Almacenamiento de Retornos: Cuando se llama a una subrutina, la dirección de retorno (la dirección a la que se debe regresar después de ejecutar la subrutina) se guarda en el stack.
2. Paso de Parámetros: Los parámetros de una función pueden pasarse a través del stack.
3. Variables Locales: Las variables locales de una función se almacenan en el stack.
4. Preservación de Registros: Los registros que necesitan ser preservados entre llamadas a subrutinas se guardan en el stack.

Operaciones Básicas:

- PUSH: Inserta (apila) un valor en la cima del stack.
- POP: Extrae (desapila) el valor de la cima del stack.

2) Describe un escenario práctico donde el uso de ensamblador sería más ventajoso que el uso de un lenguaje de alto nivel.

R.- En situaciones donde se requiere un control muy preciso del hardware, como en el desarrollo de controladores de dispositivos específicos o en optimizaciones extremas de rendimiento, el uso de ensamblador puede ser más ventajoso que un lenguaje de alto nivel.

3) Explique cada línea del siguiente código del lenguaje ensamblador y diga que es lo que se está haciendo

```
MOV AX, 5      ; Línea 1
MOV BX, 10     ; Línea 2
ADD AX, BX     ; Línea 3
MOV CX, AX     ; Línea 4
```

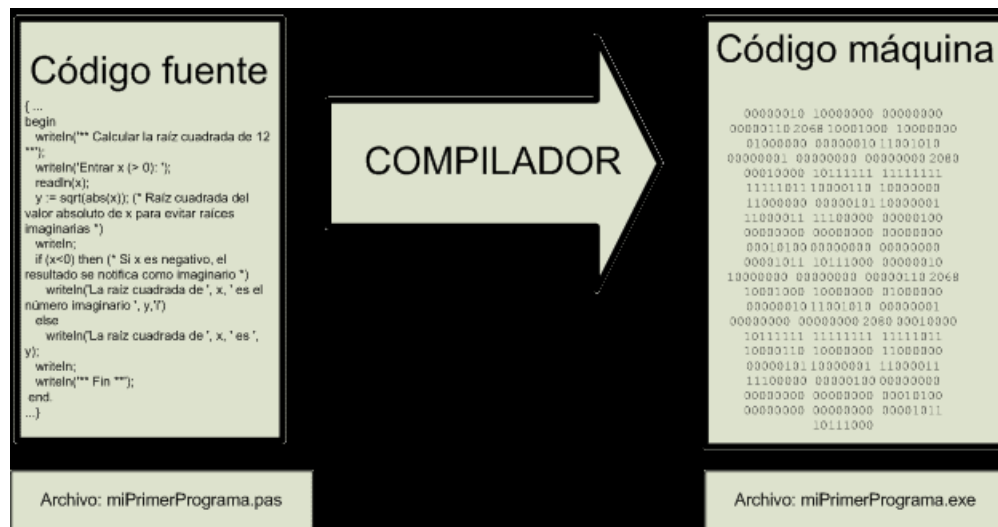
R.- Línea 1: Asigna 5 al registro AX.

Línea 2: Asigna 10 al registro BX.

Línea 3: Suma los valores en AX y BX, y guarda el resultado en AX.

Línea 4: Copia el resultado de la suma de AX a CX.

4) Explique detalladamente cómo funcionan los compiladores



Un compilador traduce todo el programa antes de su ejecución, lo que da como resultado un archivo ejecutable. Funciona en varias etapas:

- **Análisis léxico:** el compilador divide el código fuente en una serie de tokens o símbolos que representan las palabras clave, identificadores, operadores y otros elementos del lenguaje de programación.
- **Análisis sintáctico:** el compilador verifica la estructura del código fuente y su conformidad con la gramática del lenguaje de programación.

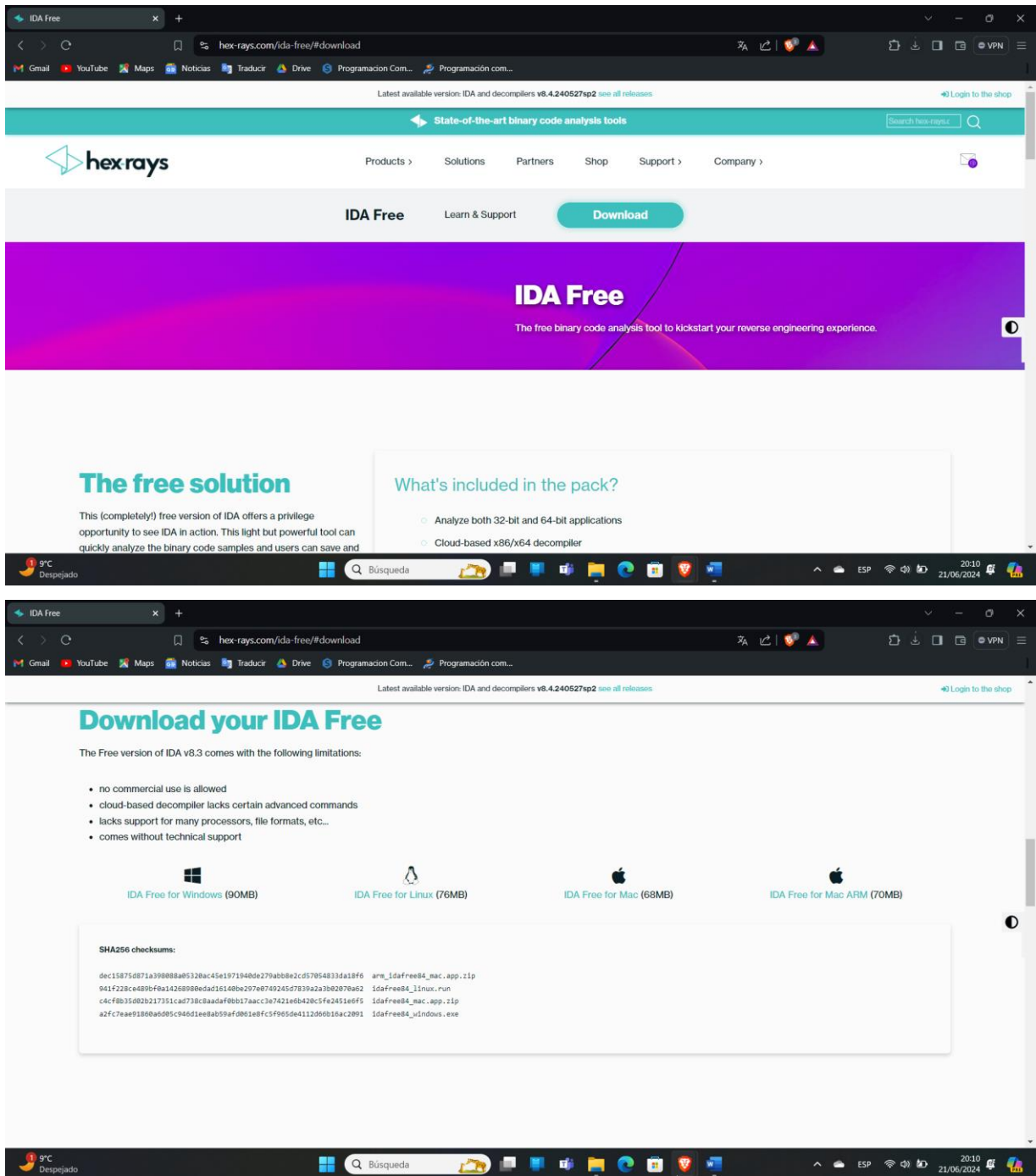
- **Análisis semántico:** este verifica que las instrucciones del programa sean semánticamente correctas y coherentes, y realiza verificaciones como la asignación de tipos y la resolución de nombres.
- **Generación de código intermedio:** este representa el programa de manera más abstracta y se puede optimizar antes de generar el código objeto.
- **Optimización de código:** puede realizar una serie de optimizaciones en el código intermedio o en el código objeto para mejorar su eficiencia, como la eliminación de código redundante o la reorganización de las instrucciones para minimizar la cantidad de ciclos de procesador requeridos.
- **Generación de código objeto:** el compilador finalmente genera el código objeto que se puede ejecutar en la computadora.
- **Vinculación y carga:** si el programa utiliza bibliotecas externas, el compilador puede vincularlas al código objeto generado y generar un archivo ejecutable que se carga en la memoria de la computadora para su ejecución.

5) Realizar capturas de pantalla del siguiente procedimiento:

IDA: Es una de las herramientas más conocidas y potentes para el análisis de código binario y desensamblado. En este laboratorio se instalará IDA FREE pero también se tiene la versión de paga IDA PRO

Paso 1:

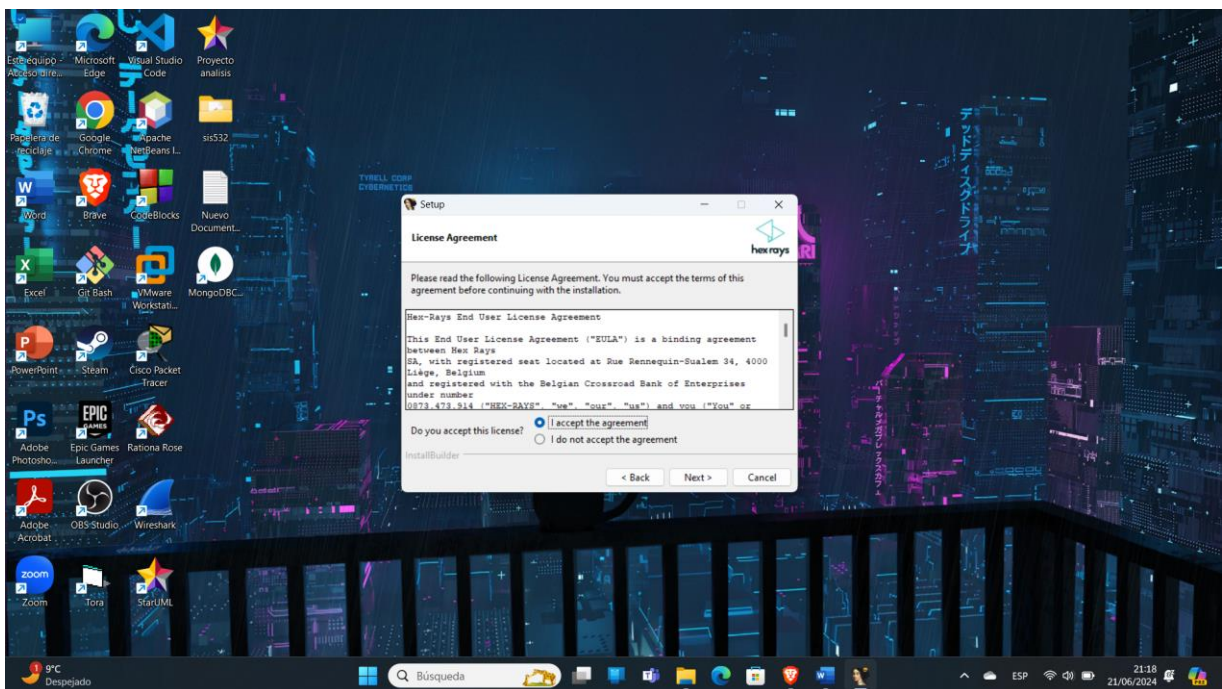
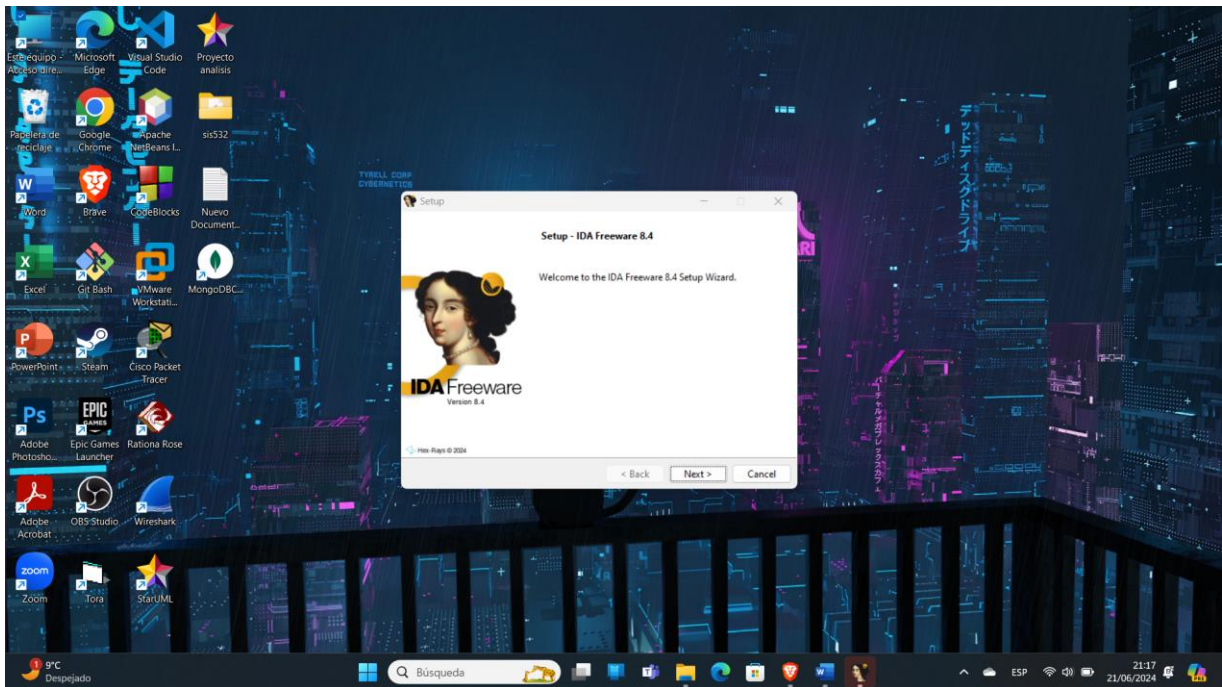
Descargar el software IDA FREE el cual lo podrá a hacer del siguiente enlace: <https://hex-rays.com/ida-free/>

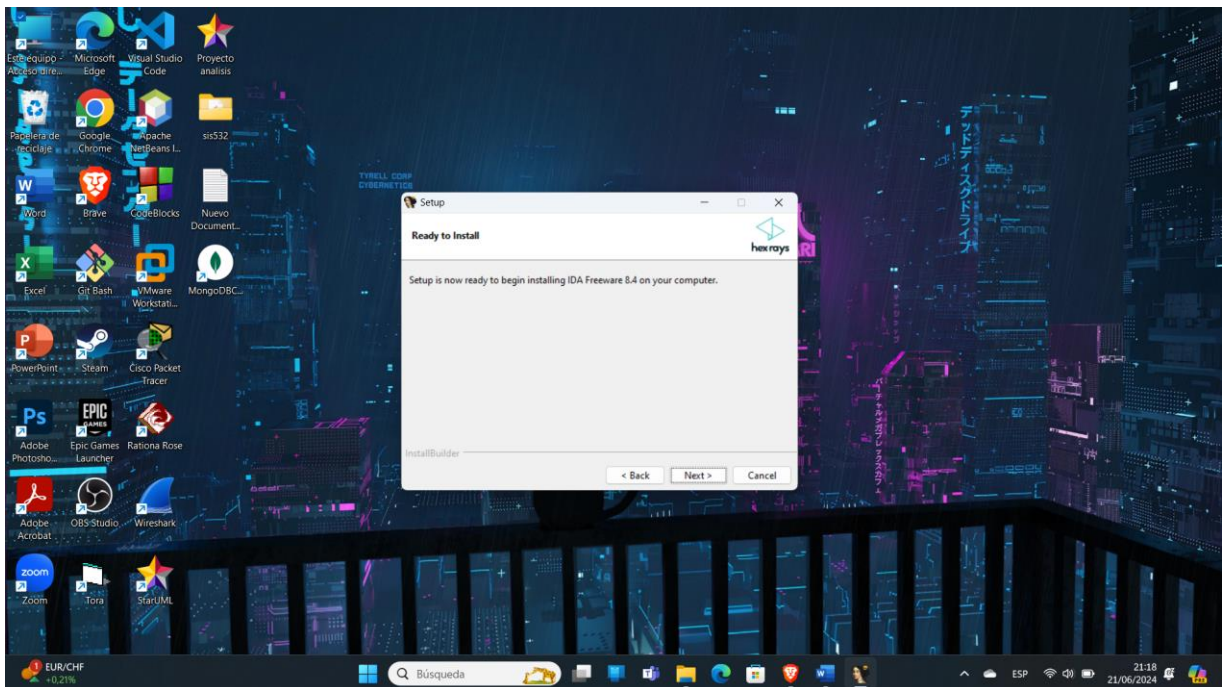
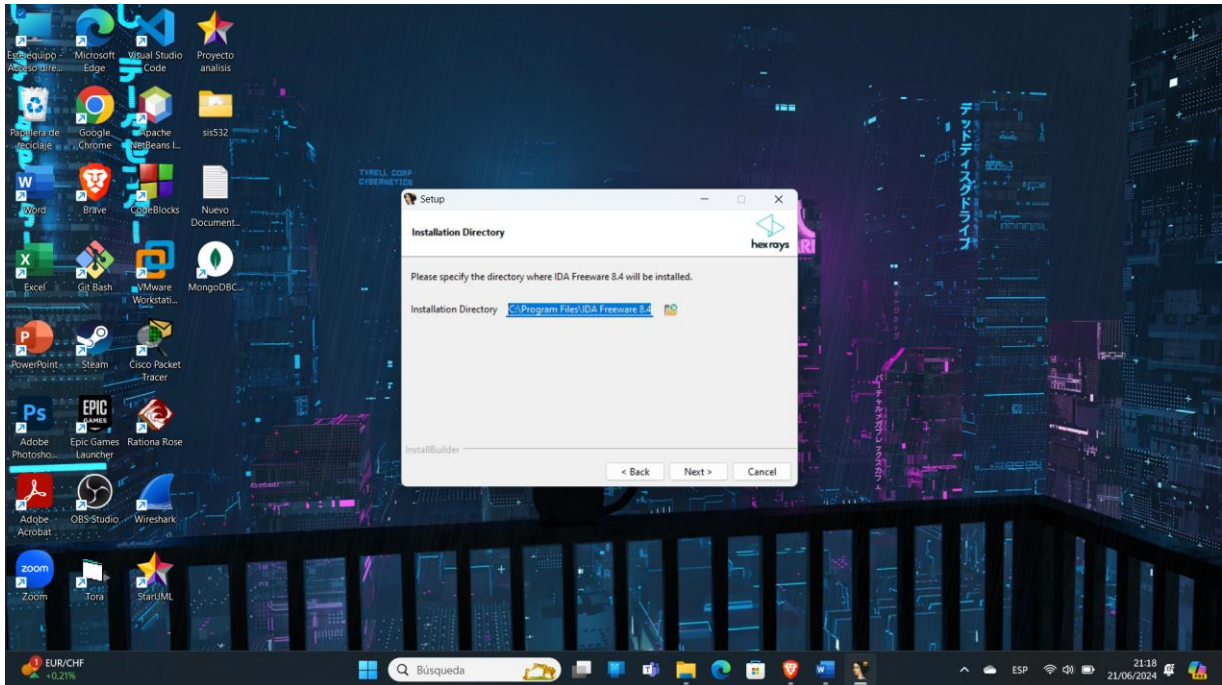


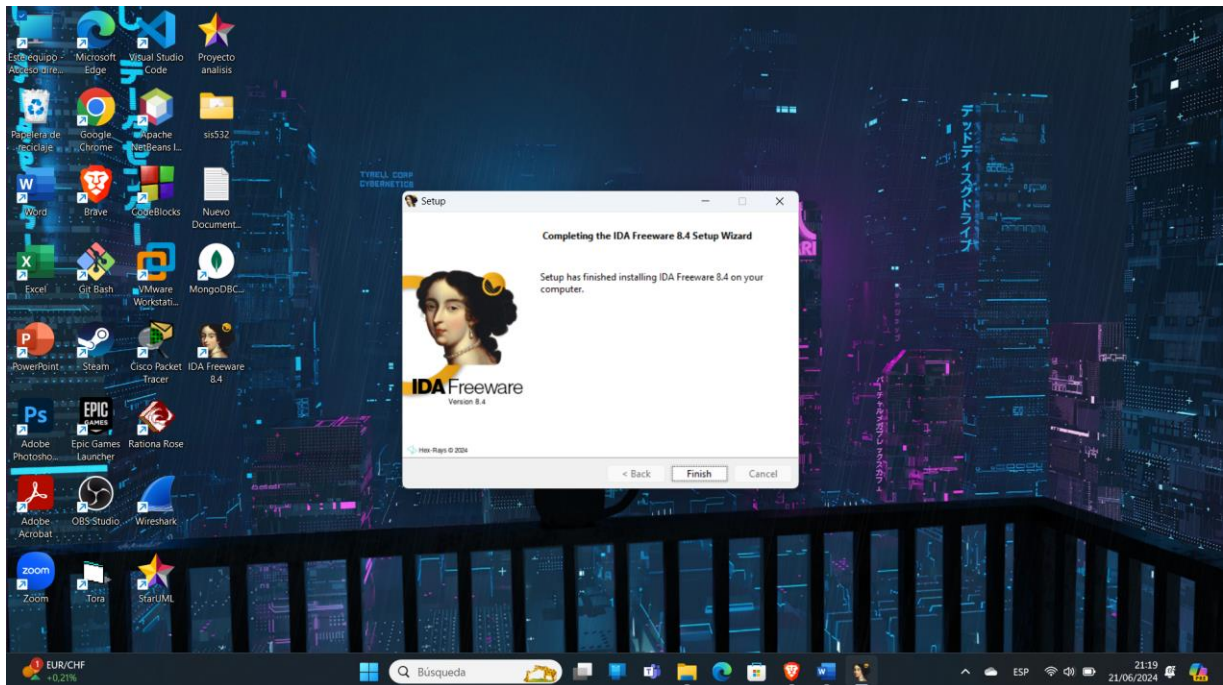
Paso 2:

Instalación

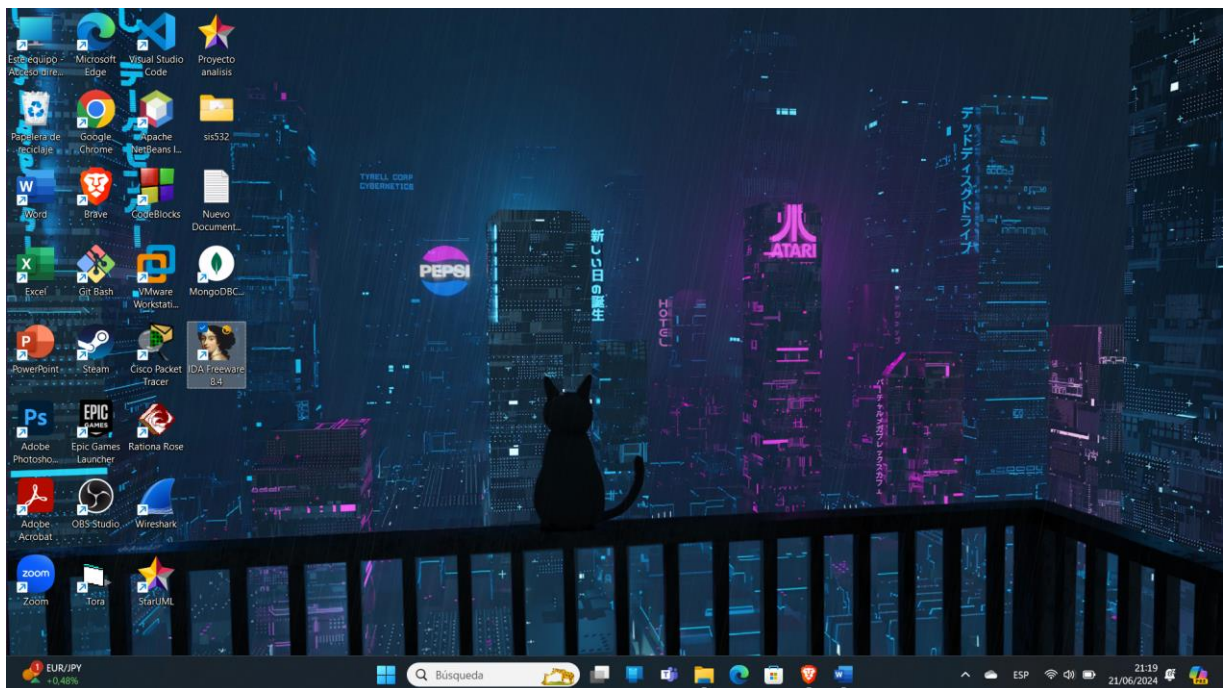
Seleccionar Next





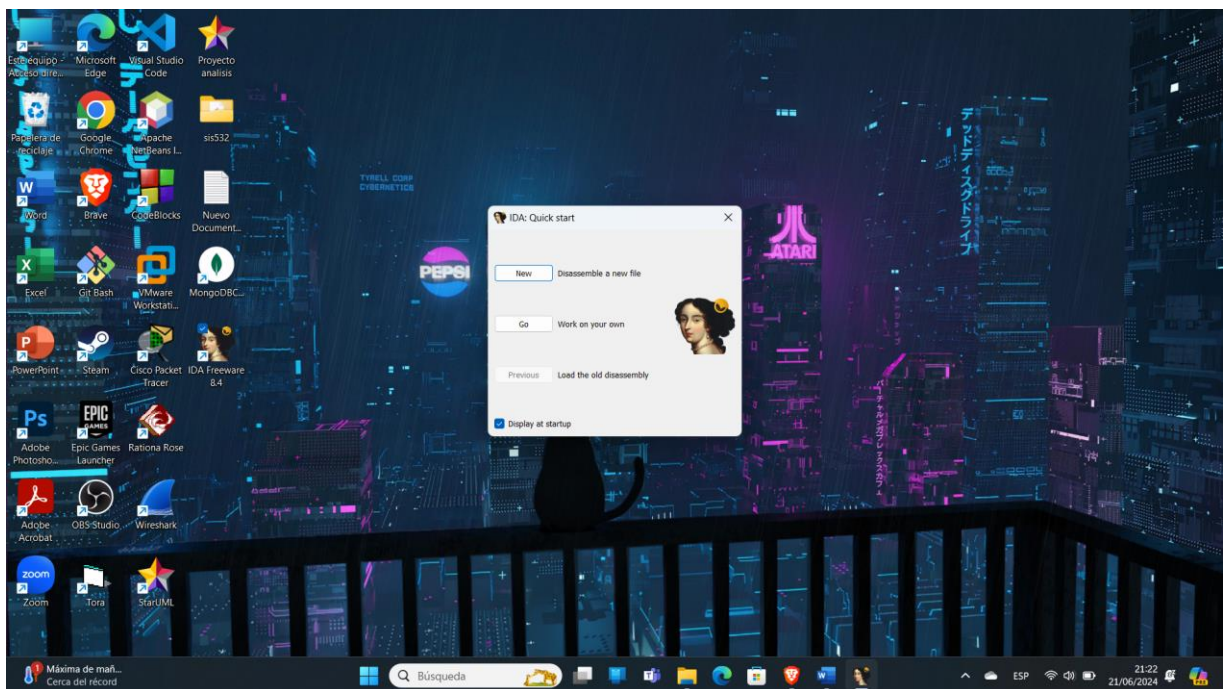
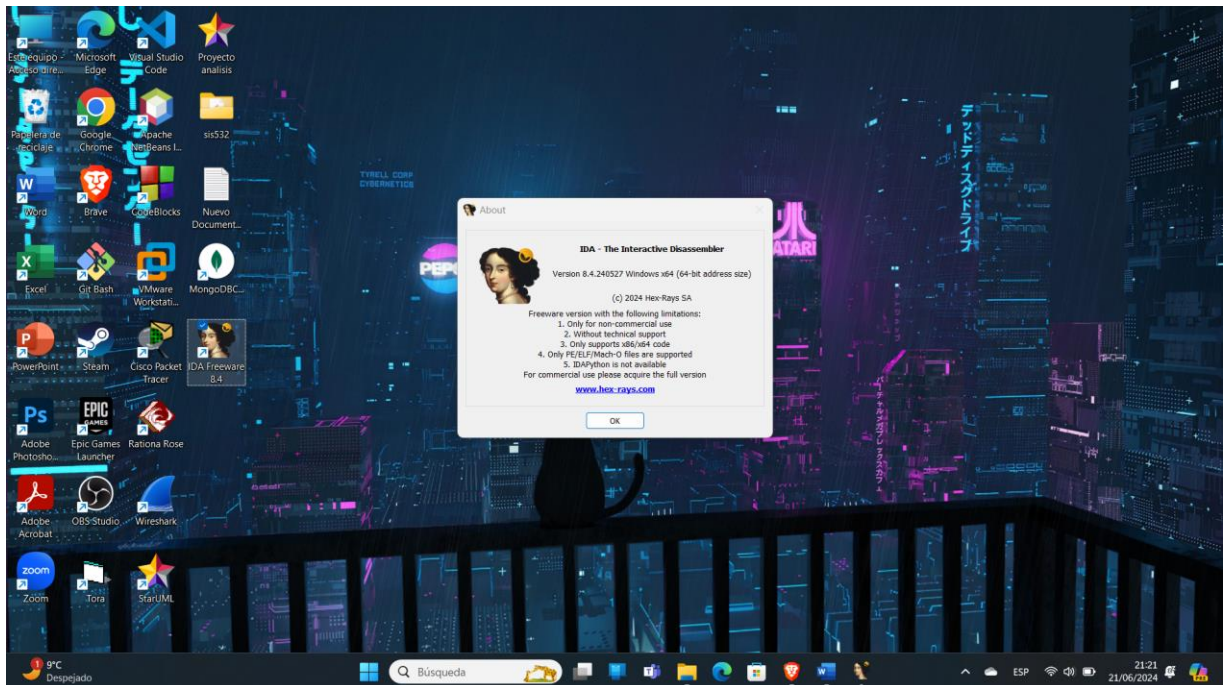


Una vez descargado e instalado deberán abrir el ejecutable .exe

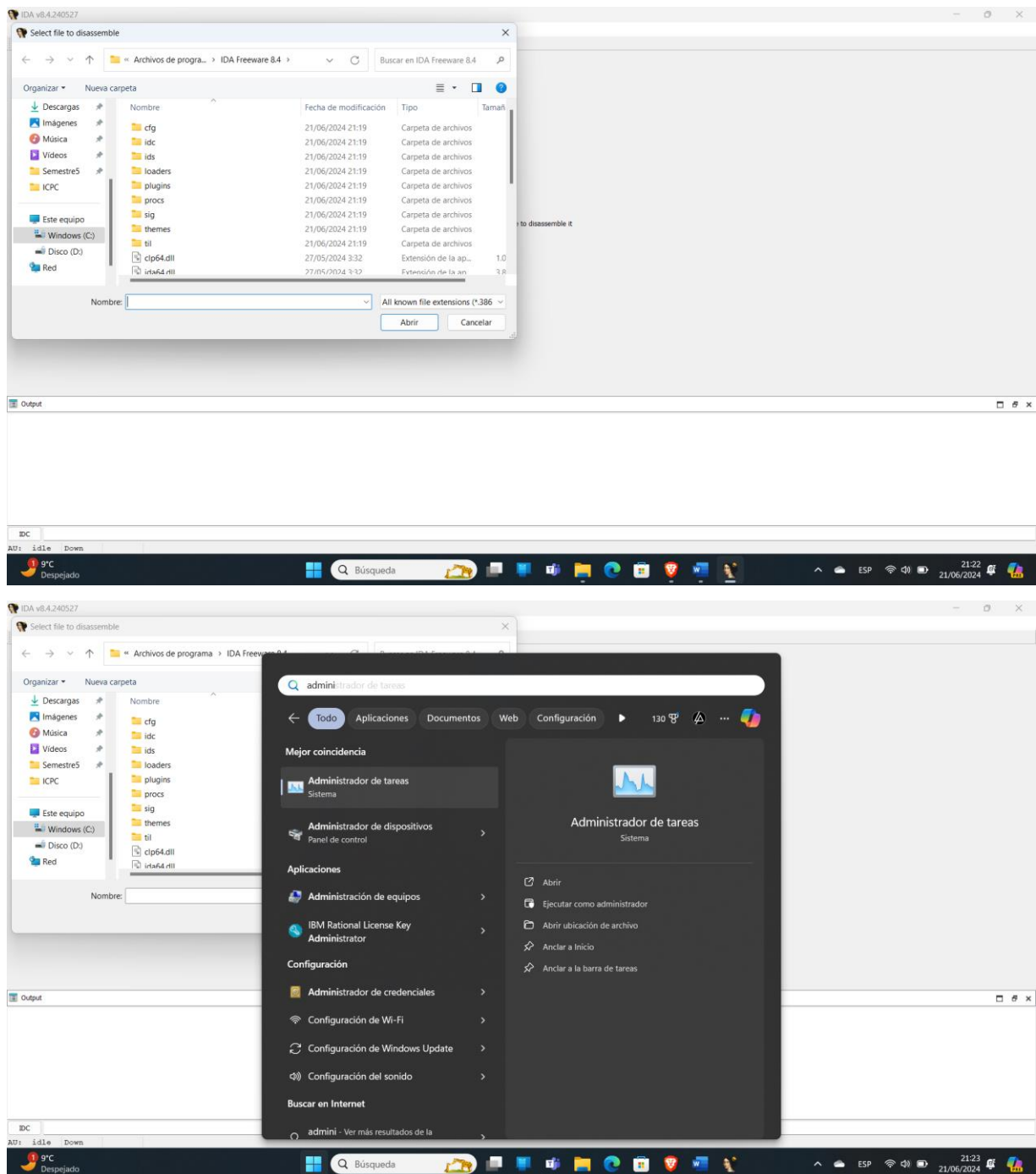


Paso 3:

Procederemos a abrir un servicio de Windows

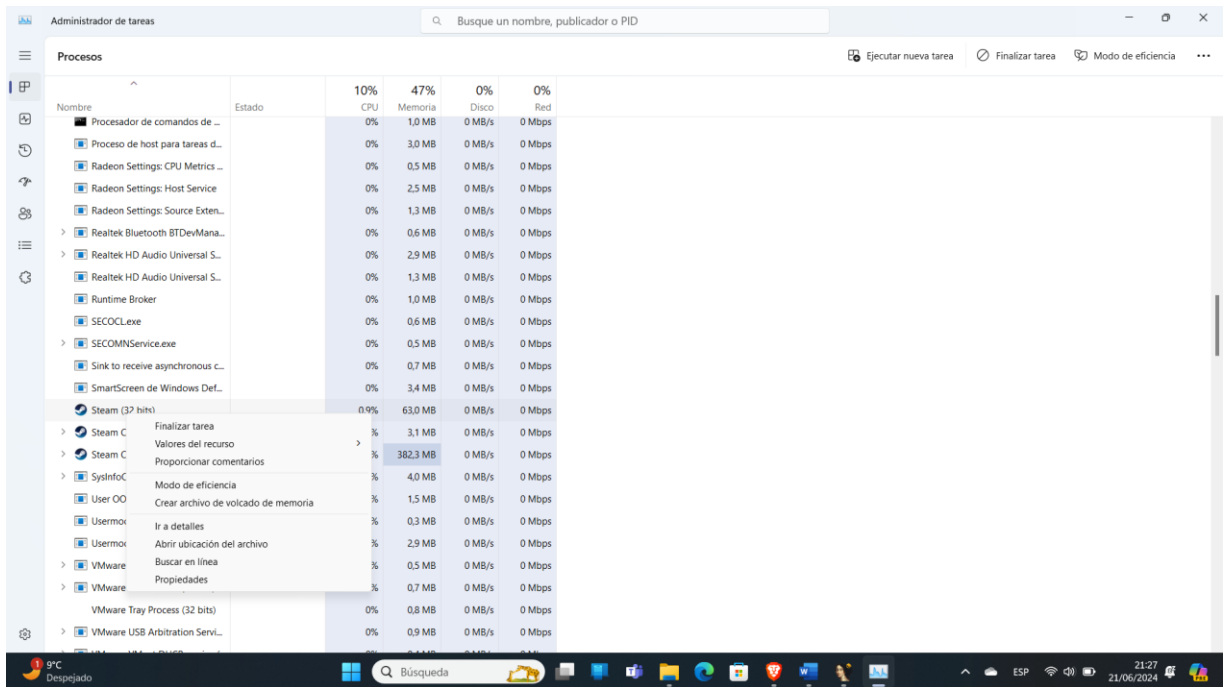


Lo que deberá hacer ahora es seleccionar “New”

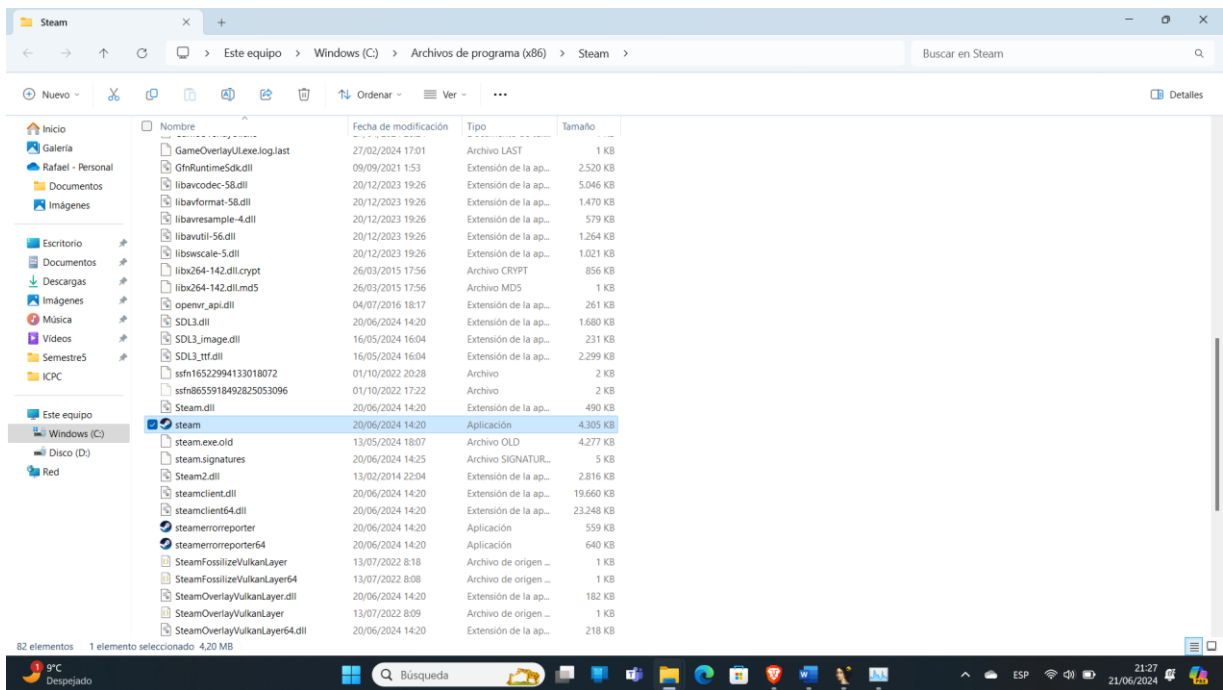


Ahora deberá seleccionar algún servicio de su administrador de tareas, primeramente, vamos a abrir el administrador de tareas

Ahora en la pestaña procesos deberá buscar cualquier servicio que se este ejecutando en tiempo real, y hacer un clic izquierdo sobre el servicio que le interesará ver el código ensamblador de este y después con un clic derecho seleccionar “Abrir ubicación del archivo”

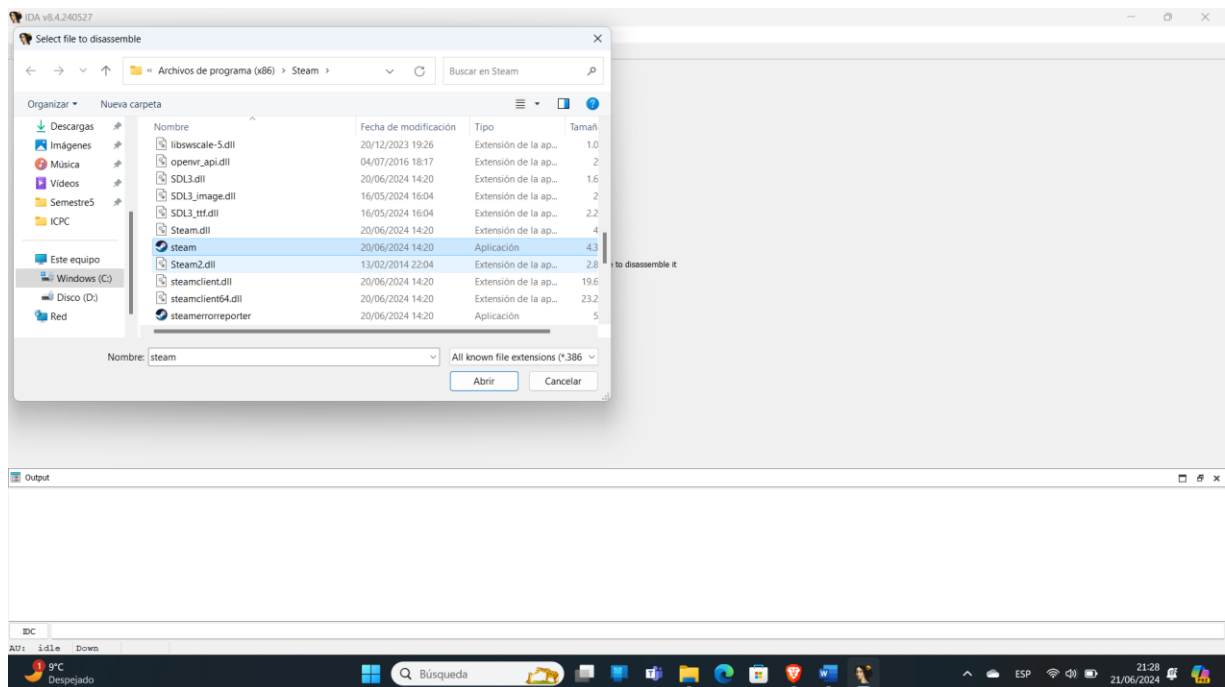


Una vez hecho esto se abrirá la ubicación del servicio

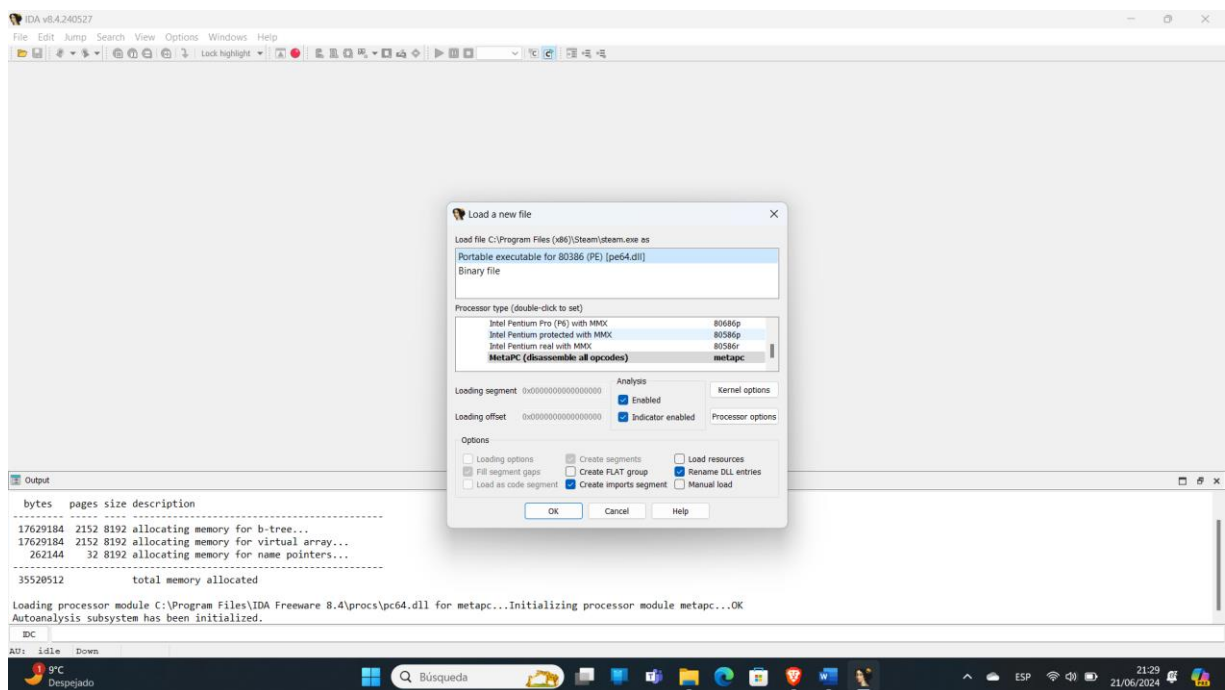


Ahora se deberá copiar la ruta en donde esta este servicio el cual es en este caso

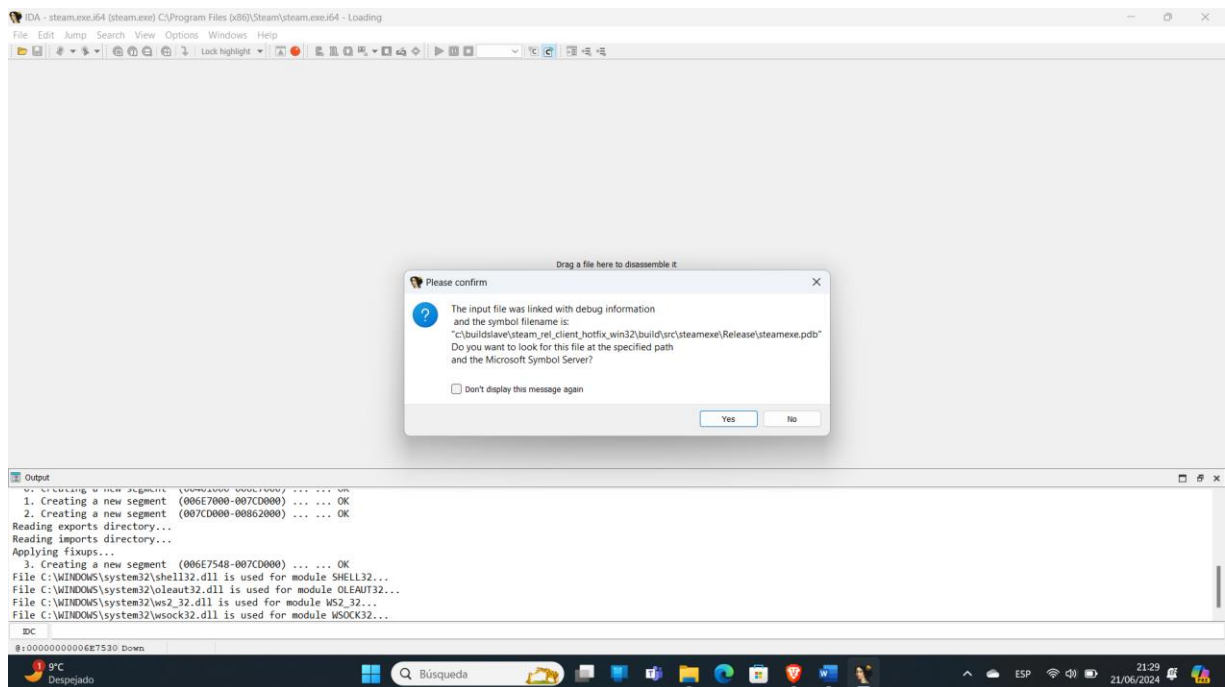
Una vez copiada esta ruta se deberá colocar en la ventana donde IDA nos pidió que se debe añadir un servicio a analizar



Una vez que coloquemos en guardar procederemos a desensamblar el servicio en este caso el “steam” tardará dependiendo el tamaño de servicio a analizar. Dejaremos todo por definido y colocamos “ok”

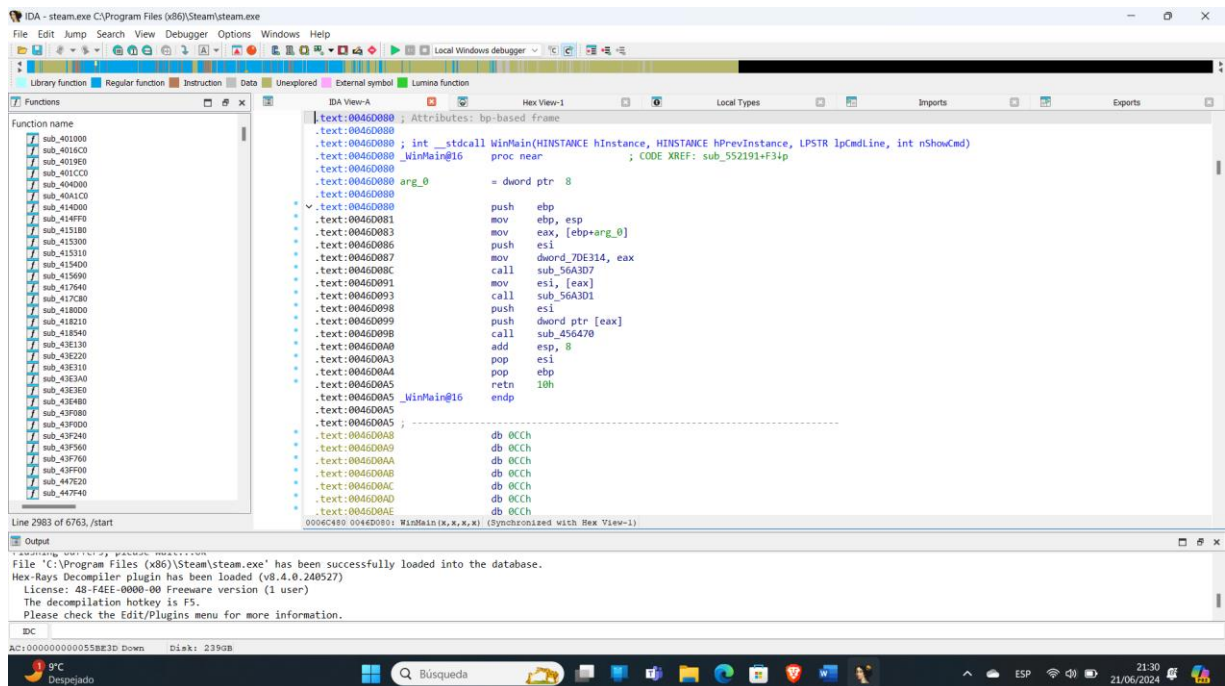


Colocaremos “no”

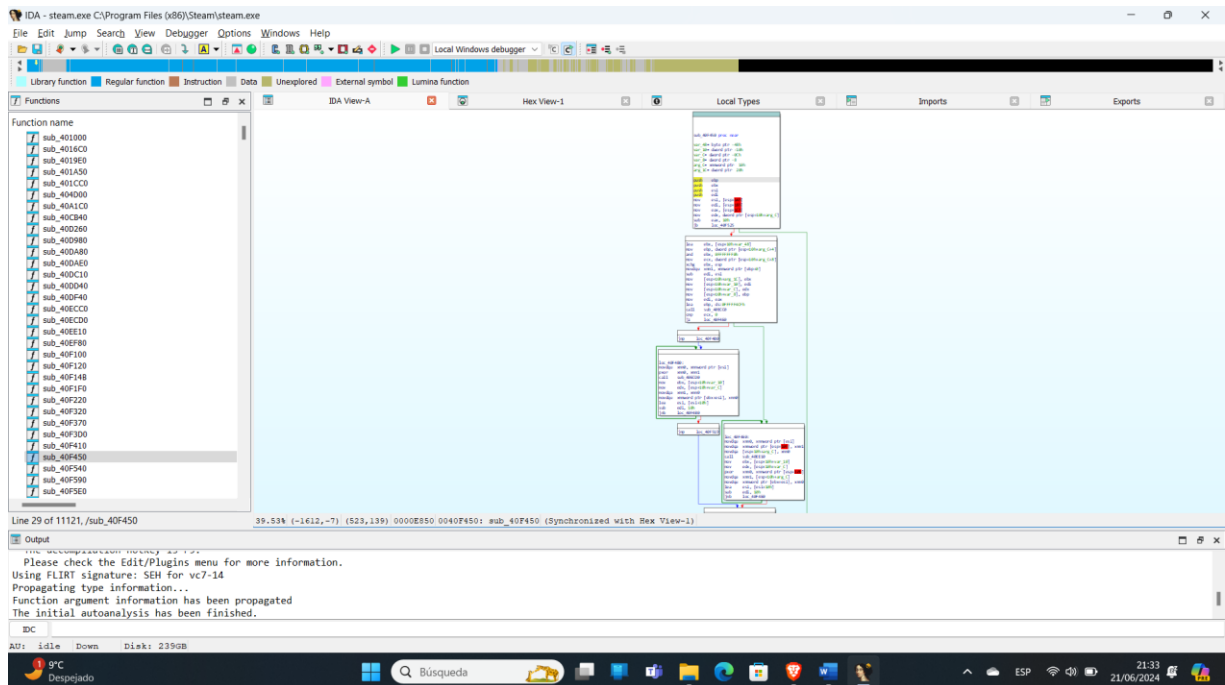


Paso 4:

Finalmente, se podrá ver código Assembler del servicio que hemos desensamblado



Como se puede ver aquí se tiene como una estructura de tablas



Aquí mismo se puede ver código Assembler

