


<div>UNIVERSIDAD AUTÓNOMA “TOMAS FRÍAS”</div> <div>CARRERA DE INGENIERÍA DE SISTEMAS</div>				
Materia:	Arquitectura de computadoras (SIS-522)			
Docente:	Ing. Gustavo A. Puita Choque			N° Práctica
Auxiliar:	Univ. Aldrin Roger Perez Miranda			
20/11/2024	Fecha publicación			9
06/12/2024	Fecha de entrega			
Grupo:	1	Sede	Potosí	

Responda las siguientes preguntas de **MANERA CONCISA**

LAS RESPUESTAS DE MANERA DIGITAL en formato .pdf

1) ¿Qué es el 'stack' en el contexto del lenguaje ensamblador y cómo se utiliza? **(10 pts)**

R.- En ensamblador, el **stack** es una estructura **LIFO** usada para almacenar datos temporales, direcciones de retorno y pasar parámetros. ✓

- **PUSH:** Inserta un valor en la pila.
 - **POP:** Extrae el valor superior. ✓
- El registro **ESP** apunta al tope de la pila.

2) Describe un escenario práctico donde el uso de ensamblador sería más ventajoso que el uso de un lenguaje de alto nivel. **(10 pts)**

R.- Es ventajoso usar ensamblador cuando se necesita:

- **Máximo control** sobre el hardware.
- **Optimización extrema** en rendimiento o tamaño del código.
- **Acceso directo** a registros o instrucciones específicas del procesador.
- **Programación en sistemas embebidos** con recursos limitados.
- **Depuración a bajo nivel** o análisis de código malicioso (ingeniería inversa).

Se tenía que "describir" el escenario



3) Explique cada línea del siguiente código del lenguaje ensamblador y diga que es lo que se está haciendo **(20 pts)**

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

R.-

MOV AX, 5; 5: Carga el valor **5** en el registro **AX**. ✓

MOV BX, 10; 10: Carga el valor **10** en el registro **BX**. ✓

ADD AX, BX; Suma el valor de **BX** al de **AX** y almacena el resultado en **AX** (**AX** = **AX** + **BX**, es decir, **AX** = **15**). ✓

MOV CX, AX; Copia el valor de **AX** (15) al registro **CX**. ✓

4) Explique detalladamente cómo funcionan los compiladores (10 pts)



R.- Un **compilador** traduce el código fuente de un lenguaje de alto nivel a código máquina ejecutable. Primero, analiza el código dividiéndolo en **tokens** (análisis léxico), verifica su estructura **gramatical** (análisis sintáctico) y comprueba su **coherencia lógica** (análisis semántico). Luego, **optimiza** el código para mejorar su rendimiento, genera un **código intermedio** y finalmente lo convierte en **código máquina**. Por último, el **enlazador** combina módulos y bibliotecas para producir el ejecutable final. ✗

5) Realizar sus propias capturas de pantalla del siguiente procedimiento:

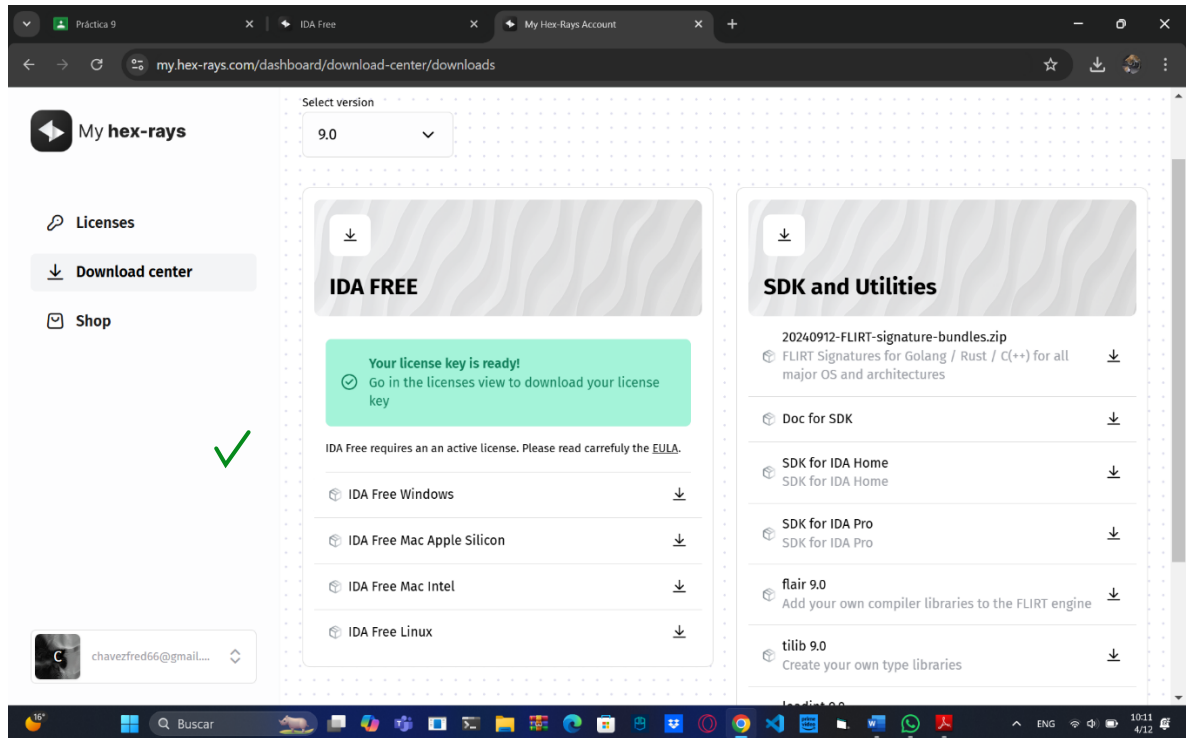
(50 pts)

EL PROCEDIMIENTO LO DEBE HACER COMO UN LABORATORIO PASO A PASO Y EXPLICAR QUE ES LO QUE SE ESTA HACIENDO CON SU RESPECTIVA CAPTURA USTED DEBE SELECCIONAR CUALQUIER SERVICIO DE SU PREFERENCIA

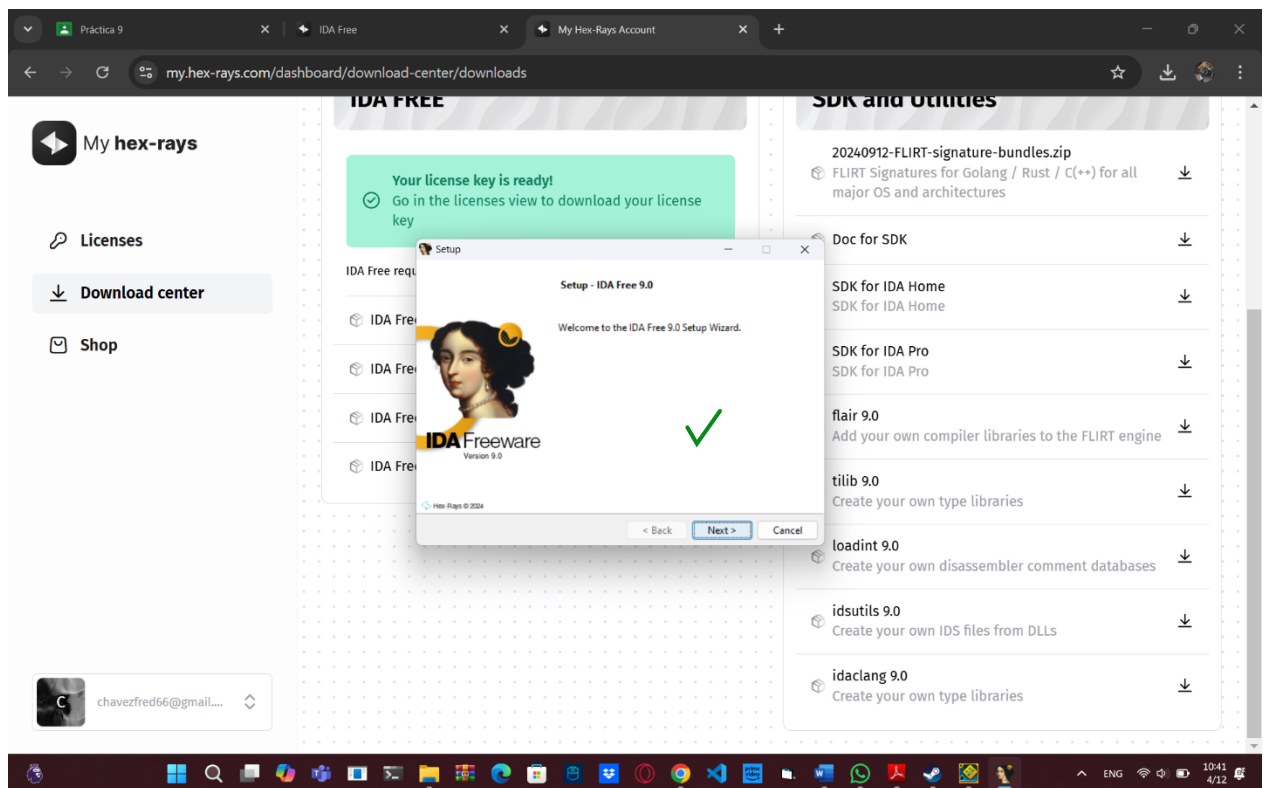
IDA: Es una de las herramientas más conocidas y potentes para el análisis de código binario y desensamblado. En este laboratorio se instalará IDA FREE pero también se tiene la versión de paga IDA PRO

R.- Realización de la práctica:

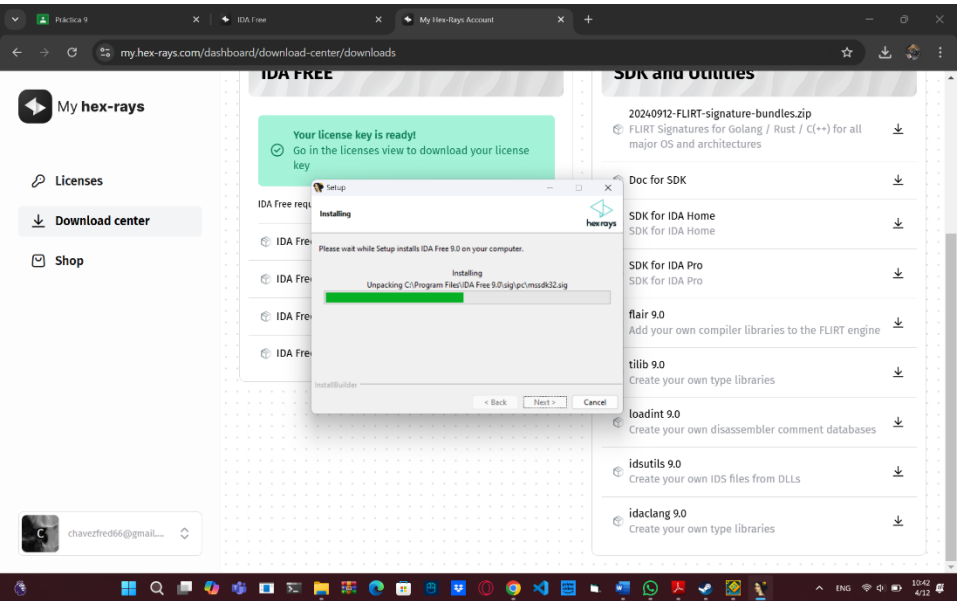
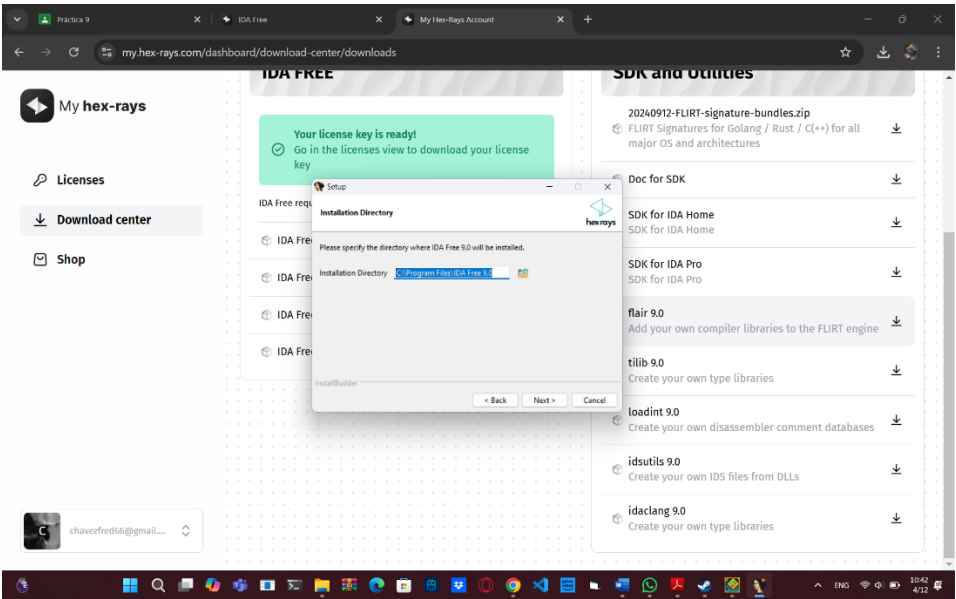
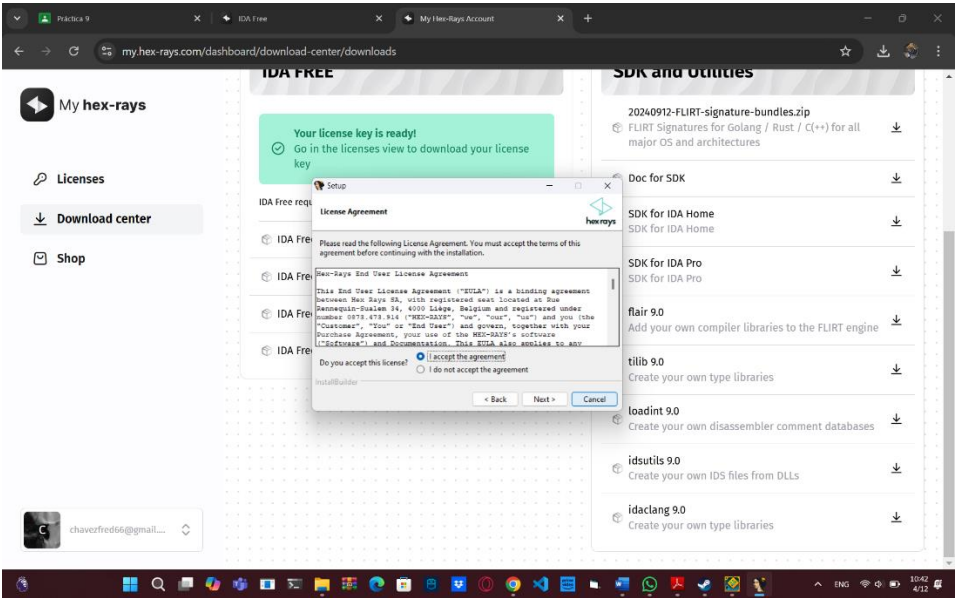
Buscamos el paquete IDA en el link proporcionado en la práctica, buscamos para la versión de Windows, también descargamos la key de activación(licencia).



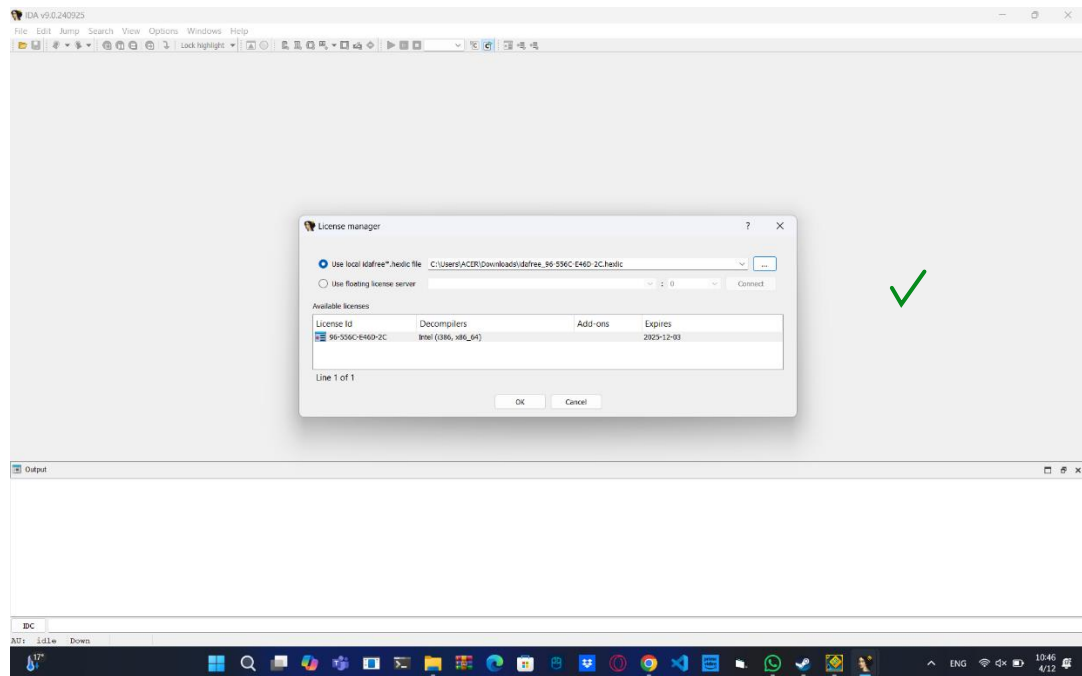
Esperamos hasta que se descargue y ejecutamos el programa:



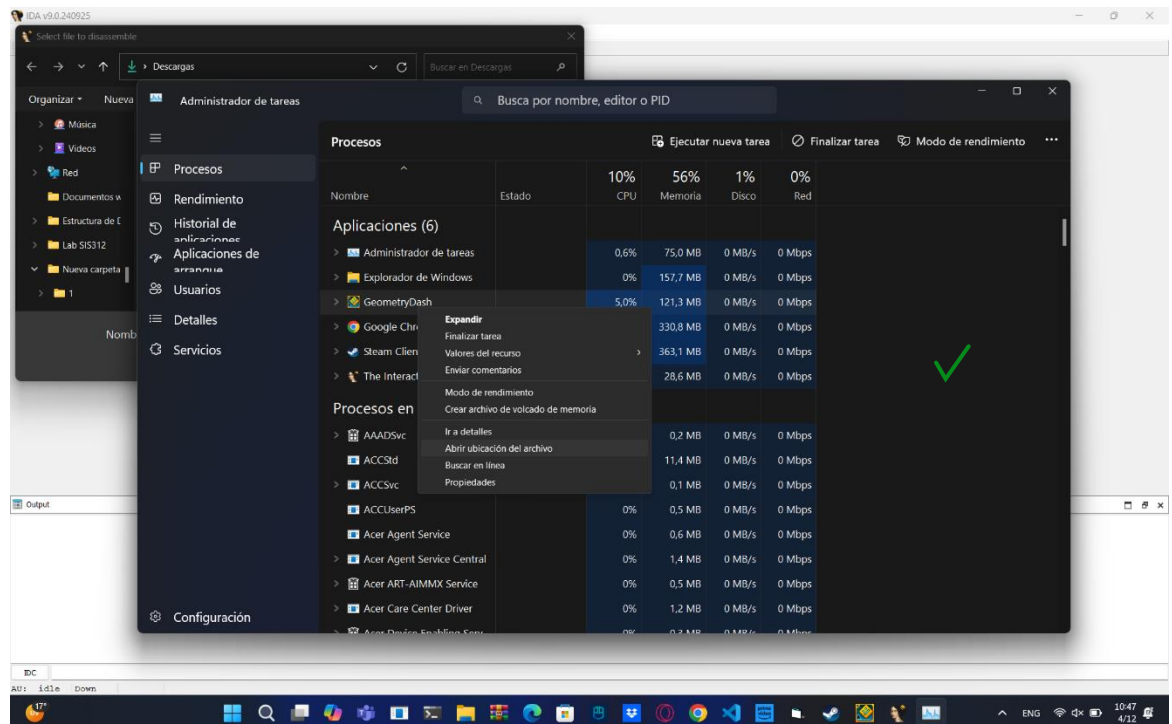
Instalamos:



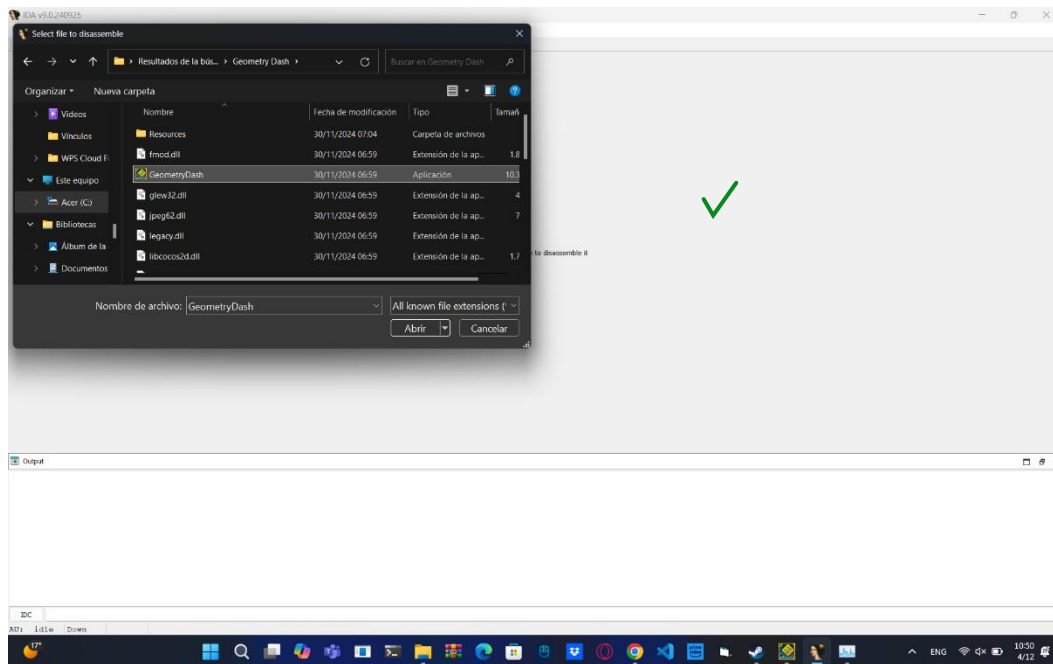
Una vez instalado abrimos el programa, nos pedirá la key o licencia, la cual es un archivo .hexlic, la cual debemos seleccionar desde el lugar donde lo descargamos



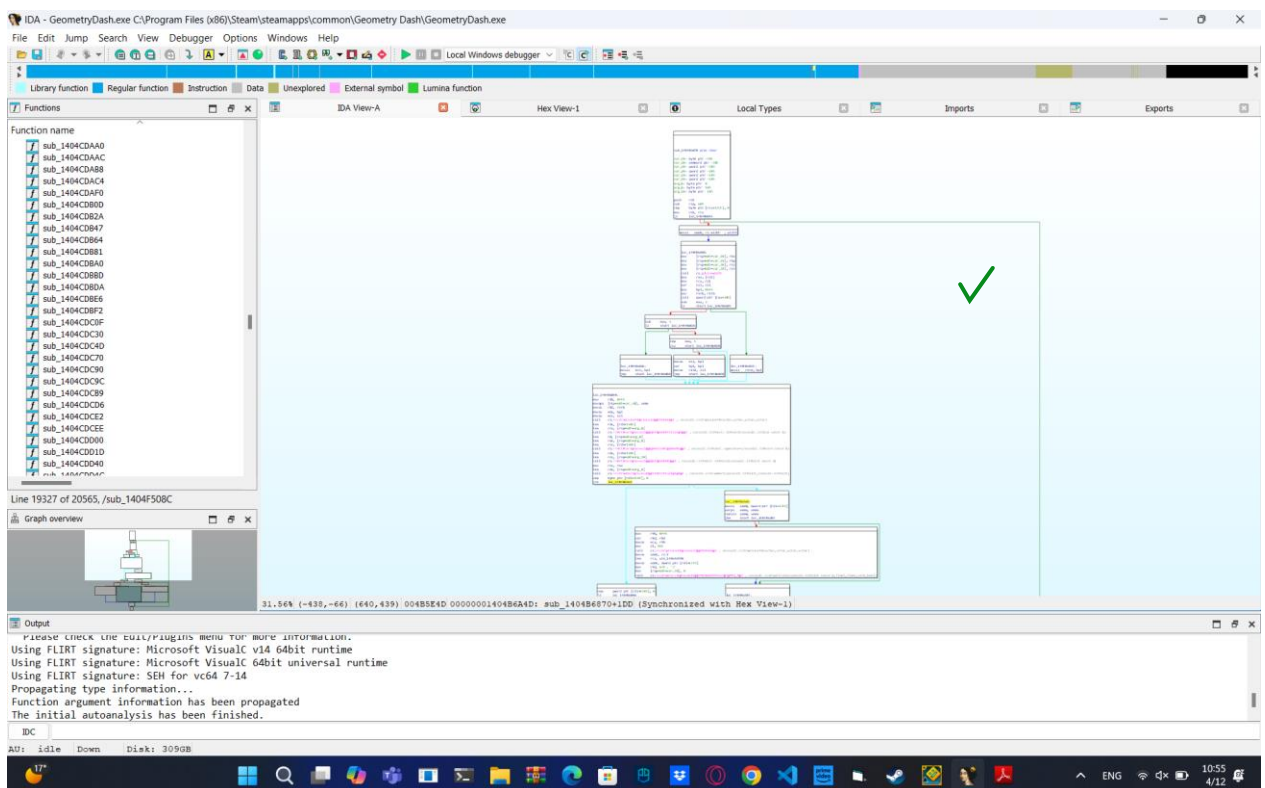
Abrimos el administrador de tareas, escogemos cualquier aplicación que deseemos analizar, debemos obtener su direccion del archivo para encontrarlo mas fácilmente con el IDA, en mi caso yo escogi Geometry Dash...



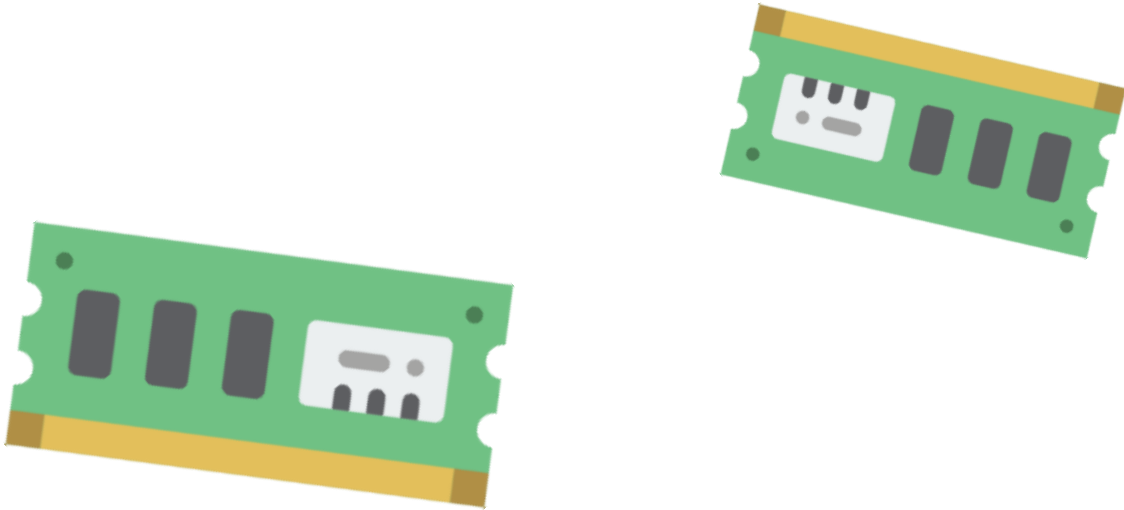
En el IDA buscamos el archivo ejecutable del programa seleccionado para desensamblarlo, con ayuda de la dirección proporcionada por el administrador de tareas.



Abrimos Geometry Dash con el programa IDA, en mi caso saque capturas a la estructura de tablas del programa



errores resultará en una penalización de -25 puntos.



LA PRESENTE PRACTICA SE DEBERA PRESENTAR EN EL SIGUIENTE FORMATO:

ApellidosPaternos_ApellidosMaternos_Nombres.pdf

Ejemplo:

Fernandez_Taboada_Mario_Jose.pdf

Y se deberá SUBIR A UN REPOSITORIO DE GITHUB

PASOS PARA SUBIR A UN REPOSITORIO DE GITHUB:

```
git init
git add README.md
git commit -m "first commit" git branch -M main git remote add
origin https://github.com/RgameplayP/Ejemplo.git git push -u
origin main
```

Para una mayor comodidad revisar el siguiente enlace: <https://youtu.be/mq-CDUwHe8Y?si=W7oZMmRakocS2EHv>

SU REPOSITORIO DE GITHUB QUE DEBE SER CREADA DE MANERA

"PUBLICA", y colocar como nombre del repositorio "Practica_09"

Revisar el ejemplo para entender cómo se debe subir el archivo .pdf (imagen)

PRACTICA_01_ARQUITECTURA DE COMPUTADORAS

Nombre: Pedrito Fernandez Carvajal
C.I.13343783



GITHUB:

Name: Pedrito_Fernandez_Carvajal Pedrito

Enlace_Lab_2: https://github.com/Pedrito/Practica_01.git

