


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

### Qué es el 'stack' en el contexto del lenguaje ensamblador y cómo se utiliza

El stack (pila) en el contexto del lenguaje ensamblador es una estructura de datos que funcionaba bajo el principio de Last In, First Out (LIFO),

PUSH: Inserta un valor en el stack.

POP: Extrae un valor del stack.

CALL y RET: Usan el stack para almacenar la dirección de retorno cuando se llama a una subrutina.

ESP (Stack Pointer): Registro que apunta al tope del stack.



### Escenario práctico donde el ensamblador es más ventajoso que un lenguaje de alto nivel

El ensamblador es preferido cuando se necesita control absoluto sobre el hardware o un alto rendimiento.

Escenario Diseño de sistemas embebidos:

Se tenía que "decribir" ✗

En microcontroladores de recursos limitados (como los usados en sensores IoT), el ensamblador permite optimizar el uso de memoria y ciclos de reloj. Esto es crítico para garantizar la eficiencia y el funcionamiento en tiempo real.

### Explicación del código ensamblador

El código realiza operaciones aritméticas y transfiere valores entre

registros: MOV AX, 5 ; Línea 1

Función: Mueve el valor inmediato 5 al registro AX.

Propósito: Inicializa el registro AX con el valor 5.



MOV BX, 10 ; Línea 2

Función: Mueve el valor inmediato 10 al registro BX.



Propósito: Inicializa el registro BX con el valor 10.

AX, BX ; Línea 3

Función: Suma el contenido de BX al contenido de AX y almacena el resultado en AX.



Propósito: Realiza la operación  $AX = AX + BX$ , que en este caso es  $5 + 10 = 15$ .

MOV CX, AX ; Línea 4

Función: Copia el contenido de AX al registro CX.



Propósito: Almacena el resultado de la suma (15) en CX.

### Explicación detallada de cómo funcionan los compiladores

Se tenía que explicar a "detalle"

Un compilador es un programa que traduce el código fuente escrito en un lenguaje de alto nivel (como C, Python o Java) a un lenguaje de bajo nivel (como lenguaje ensamblador o código máquina) que pueda ser ejecutado por un procesador.



### Etapas principales del funcionamiento de un compilador:

Análisis Léxico:

Análisis Sintáctico:

Análisis Semántico:

Generación de Código Intermedio:

Optimización del Código:

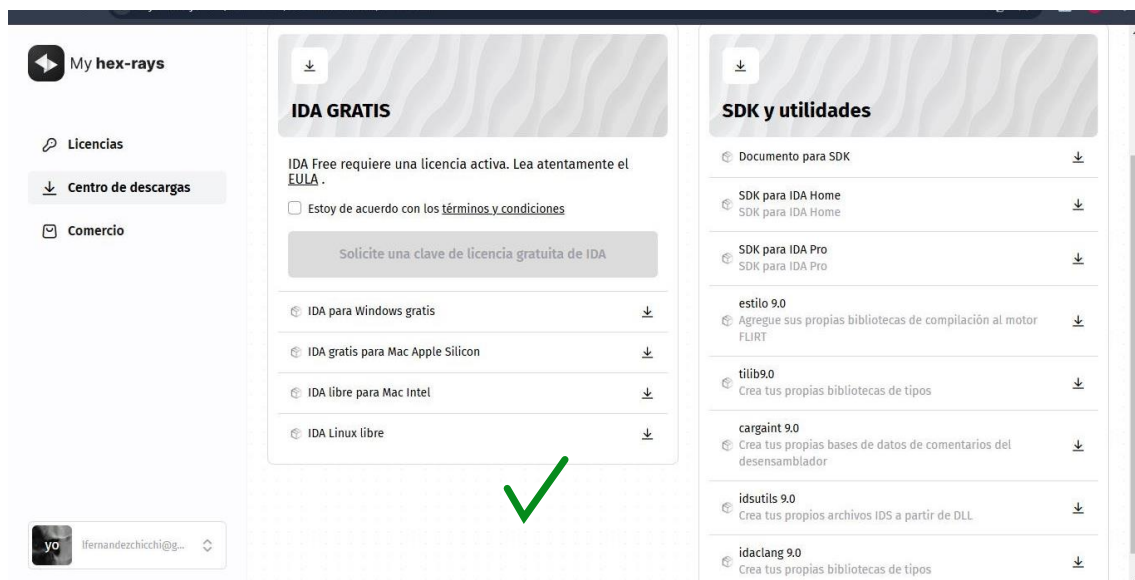
Generación de Código de Máquina:

Ensamblado y Enlace:

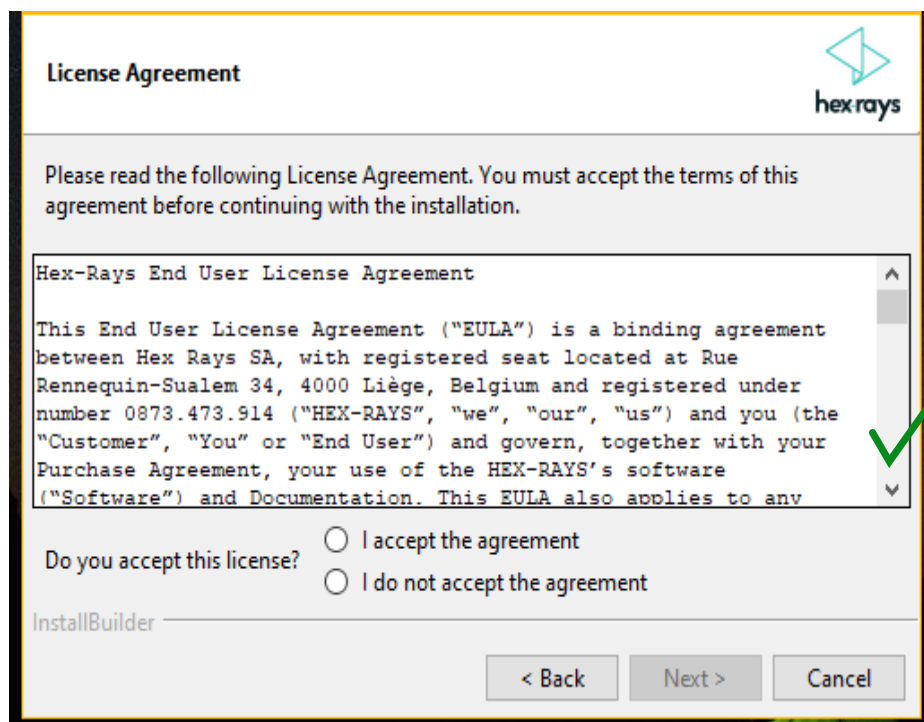
Realizar sus propias capturas de pantalla del siguiente procedimiento:

(50 pts.)

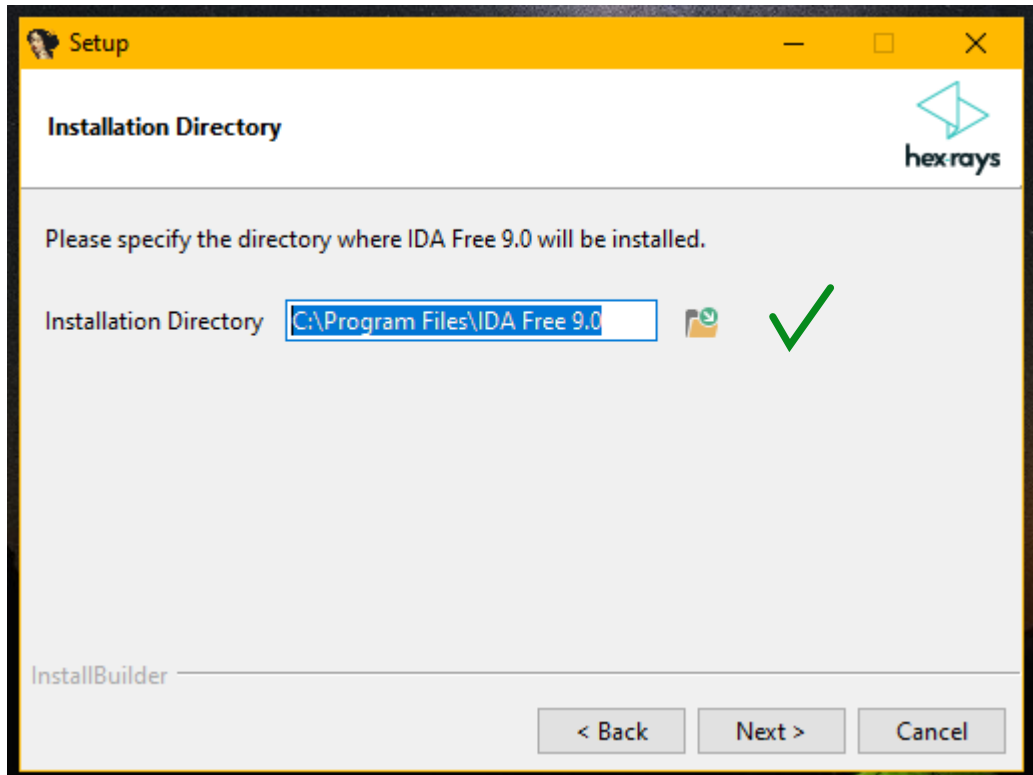
Descargar desde su página web y activar la licencia .



## Instalar



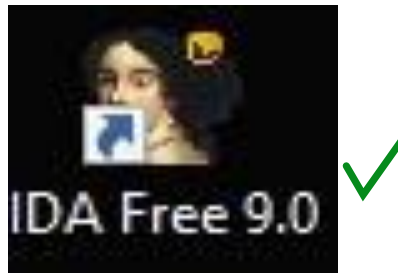
Selecciona la carpeta donde se instalara y después



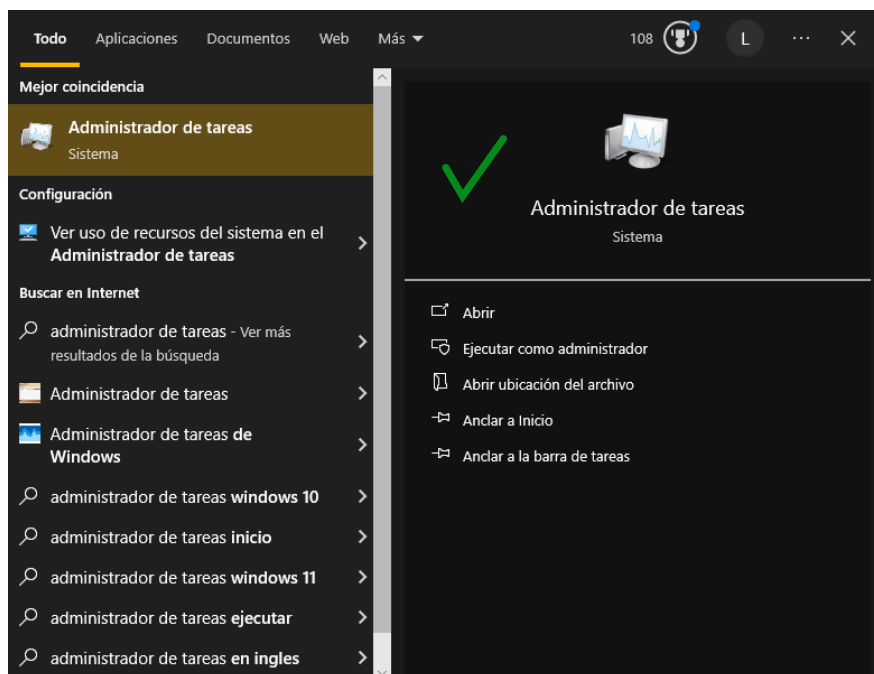
la ruta

finist





Ejecutamos el Administrador de tareas



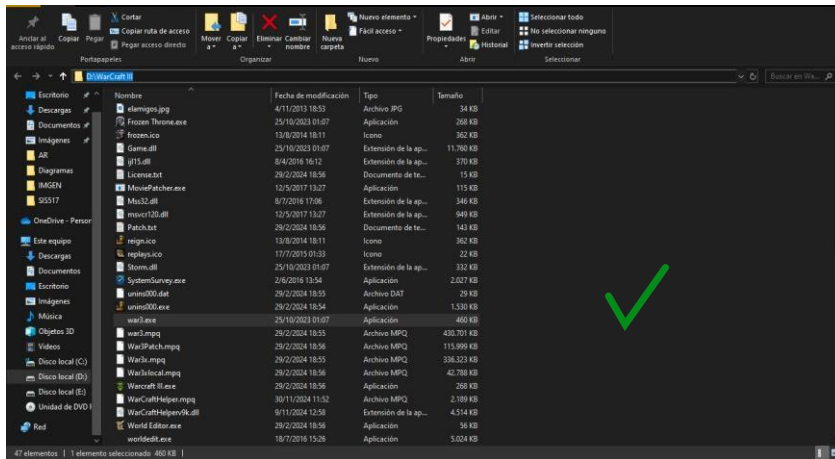
Luego ejecutamos una aplicación cualquiera

## Ubicamos en el archivo Click derecho en el APP

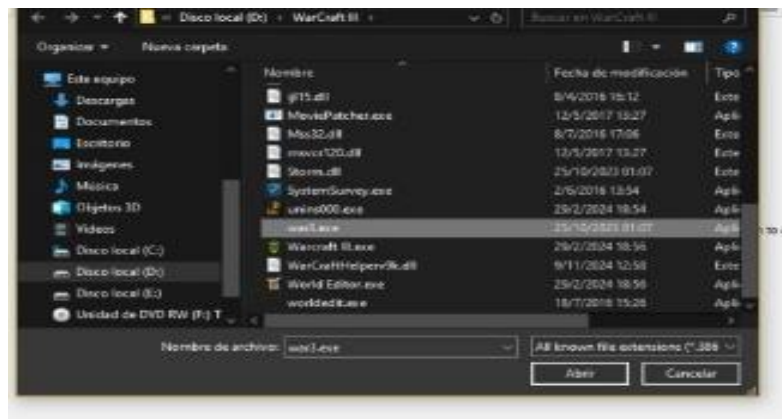
Archivo Opciones Vista					
Procesos Rendimiento Historial de aplicaciones Inicio Usuarios Detalles Servicios					
Nombre	Estado	33% CPU	54% Memoria	9% Disco	0% Red
Aplicaciones (7)					
> Administrador de tareas		1,8%	20,1 MB	0 MB/s	0 Mbps
> Explorador de Windows (5)		1,9%	45,6 MB	0 MB/s	0 Mbps
> Google Chrome (18)		0,6%	1.104,1 MB	0,1 MB/s	0 Mbps
> Microsoft Word (2)		0%	61,6 MB	0 MB/s	0 Mbps
> Reproductor multimedia (2)		0,4%	107,6 MB	0 MB/s	0 Mbps
> The Interactive Disassembler		0%	35,8 MB	0 MB/s	0 Mbps



la ruta del archivo



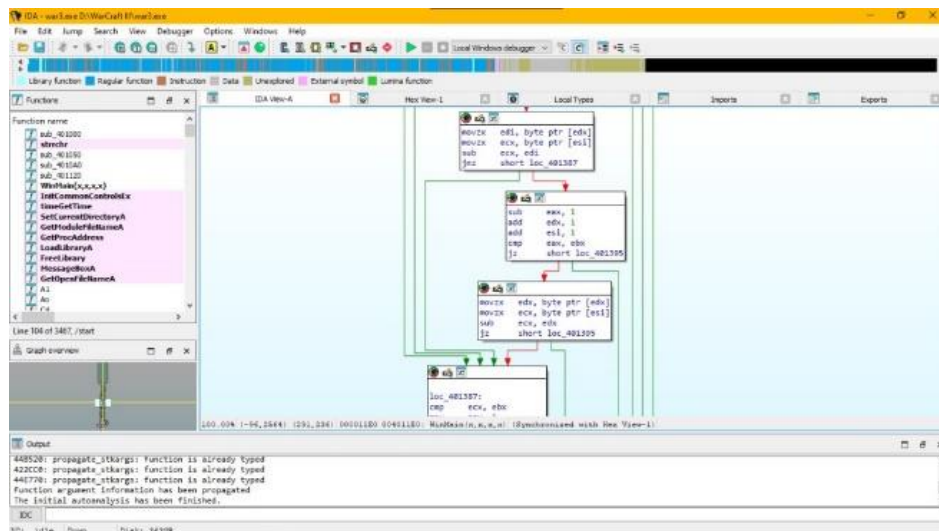
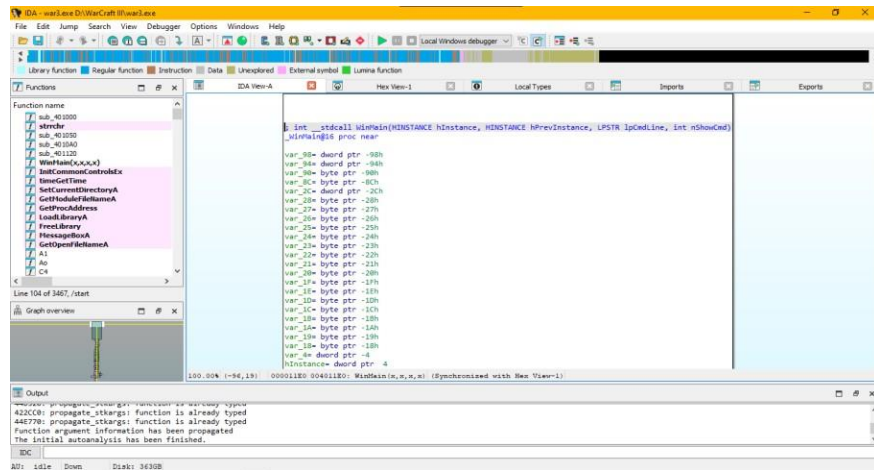
En IDA Free pagamos la ruta que copiamos



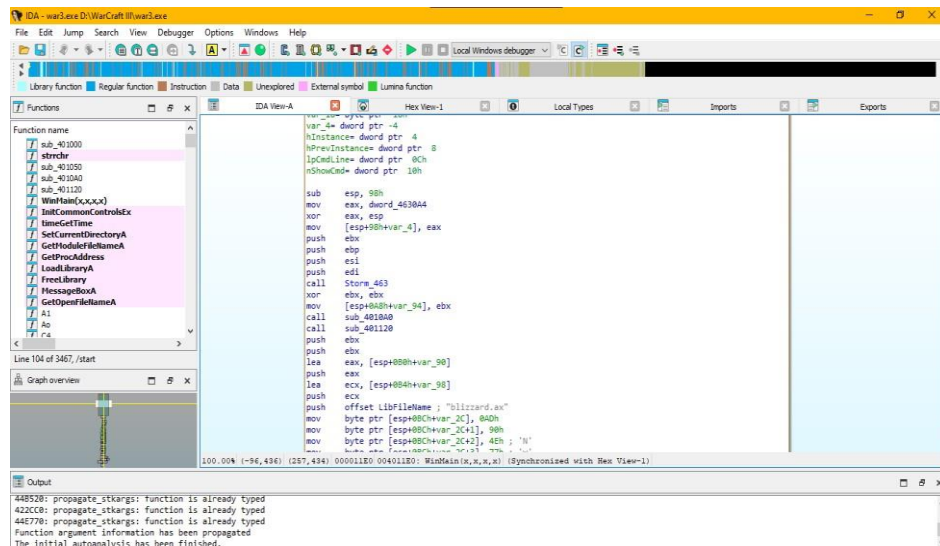




**Ya se puede ver el código de Assenmbler del servicio que  
hemos desensamblado**



## Código Assembler



Falta un poco mas de explicacion de lo que se hace en cada captura y del codigo assembler que se muestra