EXPLOITING Y REVERSING USANDO HERRAMIENTAS GRATUITAS

Comenzaremos un nuevo curso, el cual estará enfocado en la explotación y el reversing en Windows utilizando herramientas gratuitas y que se pueden conseguir fácilmente. (IDA FREE, Radare, Windbg, X64dbg, Ghidra, etc)

No utilizaremos ninguna herramienta paga, y la idea no es competir entre las diferentes herramientas, sino aprender a usarlas, en mi caso aprender a usar Radare y Ghidra que son herramientas que no uso en el día a día y tratare de usarlas lo mejor que pueda, con la salvedad que no soy un experto ni mucho menos en estas dos, así que si alguien ve algún error o mejor forma de usarlas, seria bueno que me avise, así voy corrigiendo y aprendiendo también.

Este curso empezara con la instalación de las herramientas, luego un poco de teoría sobre vulnerabilidades y luego ejemplos ultra sencillos de explotación e ira incrementando poco a poco la dificultad, lo que no habrá en este curso será la introducción a las instrucciones en assembler, eso está en el curso anterior de REVERSING CON IDA PRO, en los otros cursos anteriores y en muchos sitios de Internet, aquí vamos a ir directo a la explotación y el reversing, el que tenga dudas sobre las instrucciones, pueden consultar los cursos anteriores para aclararles las dudas.

Para armar mi entorno de trabajo lo primero será instalar las herramientas.

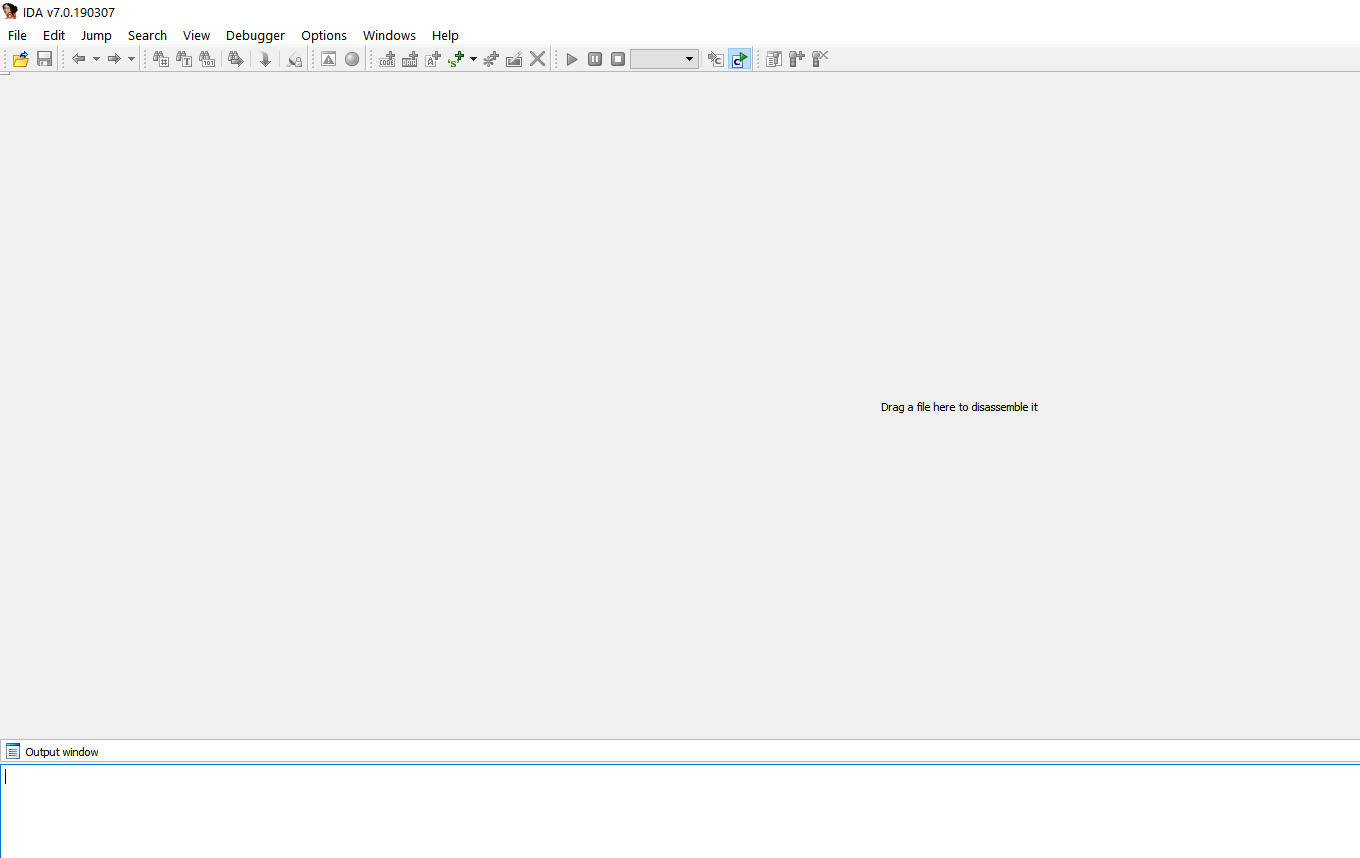
**INSTALAR IDA FREE**

Para descargarse el IDA FREE lo pueden hacer desde aquí

<https://www.hex-rays.com/products/ida/support/download_freeware.shtml>



Allí se bajan el archivo idafree70\_windows y la instalación es muy sencilla, solo siguen las instrucciones del instalador y tendrán IDA FREE funcionando en su máquina rápidamente.



**INSTALAR RADARE**

Lo siguiente será instalar radare2

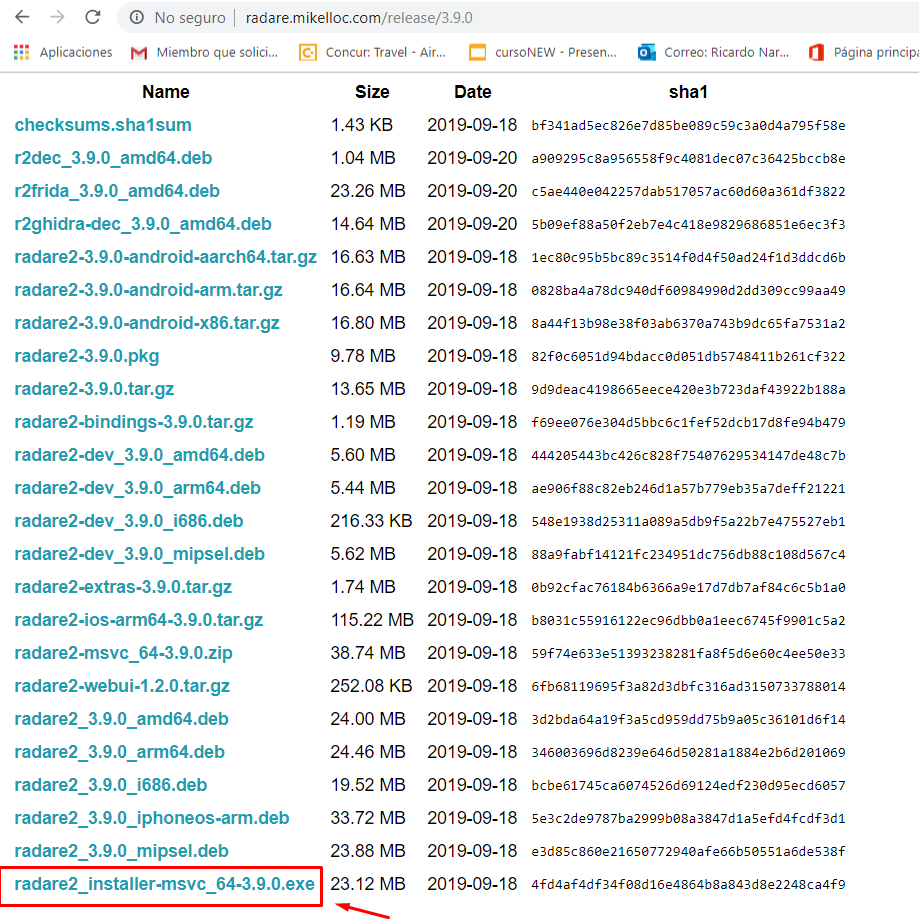
Yendo a la página de radare

<https://github.com/radareorg/radare2/releases>

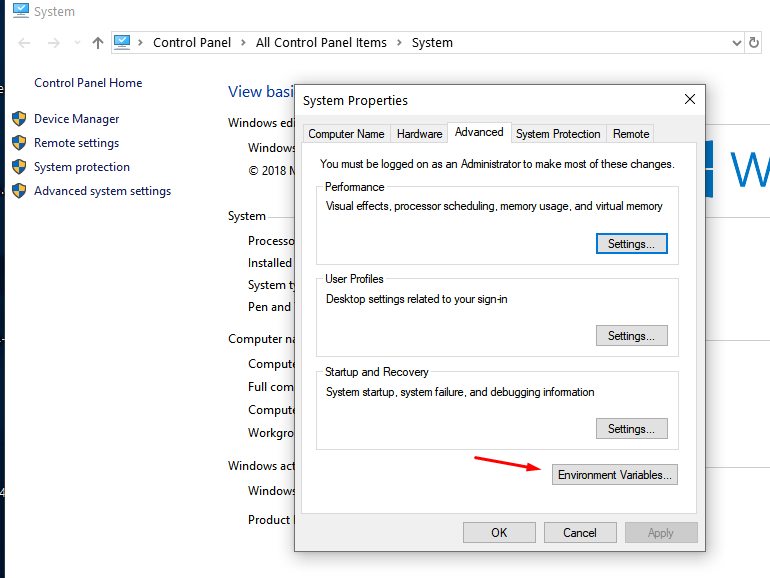
Allí entramos en el último reléase.



En este momento es el 3.9.0, más adelante será una versión mas nueva, no importa entran allí y se bajan el instalador de la versión más nueva para Windows.



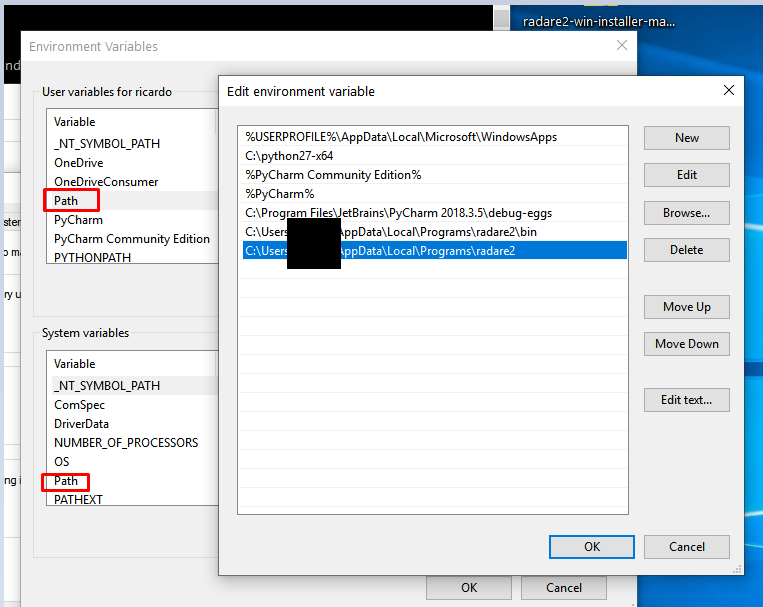
Instalarlo también es muy sencillo una vez terminado solo hay que agregar el path donde se instaló en las environment variables.



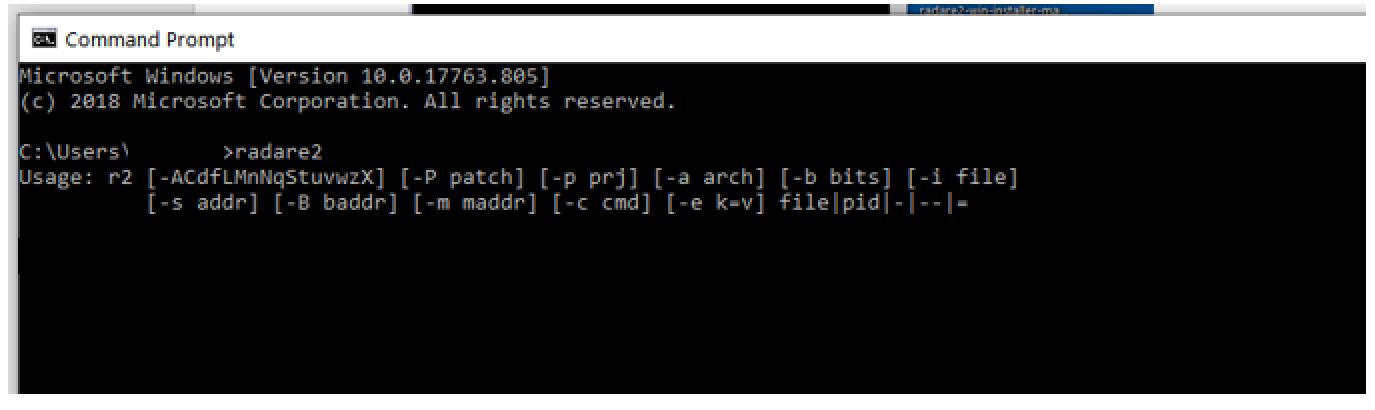
Y allí en la environment variable Path agregar

C:\Users\<nombre\_de\_usuario>\AppData\Local\Programs\radare2

C:\Users\<nombre\_de\_usuario>\AppData\Local\Programs\radare2\bin



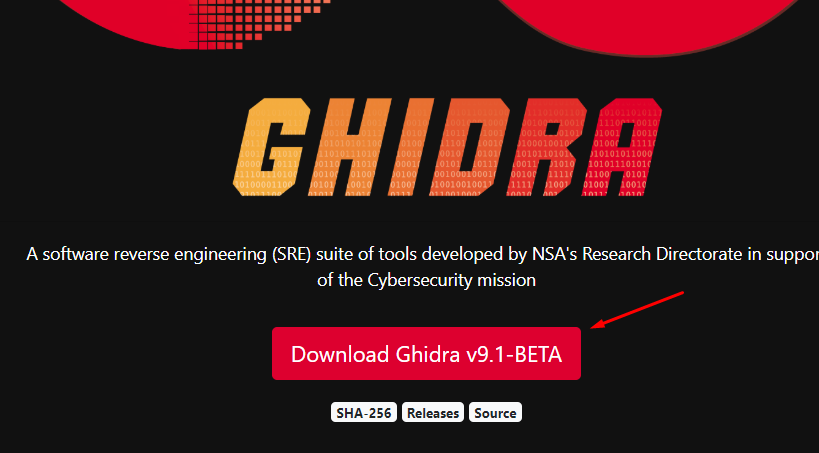
La idea es que desde un command prompt (CMD) podamos tipear radare2 y lo reconozca



**INSTALAR GHIDRA**

Instalar GHIDRA es muy sencillo también

<https://ghidra-sre.org/>



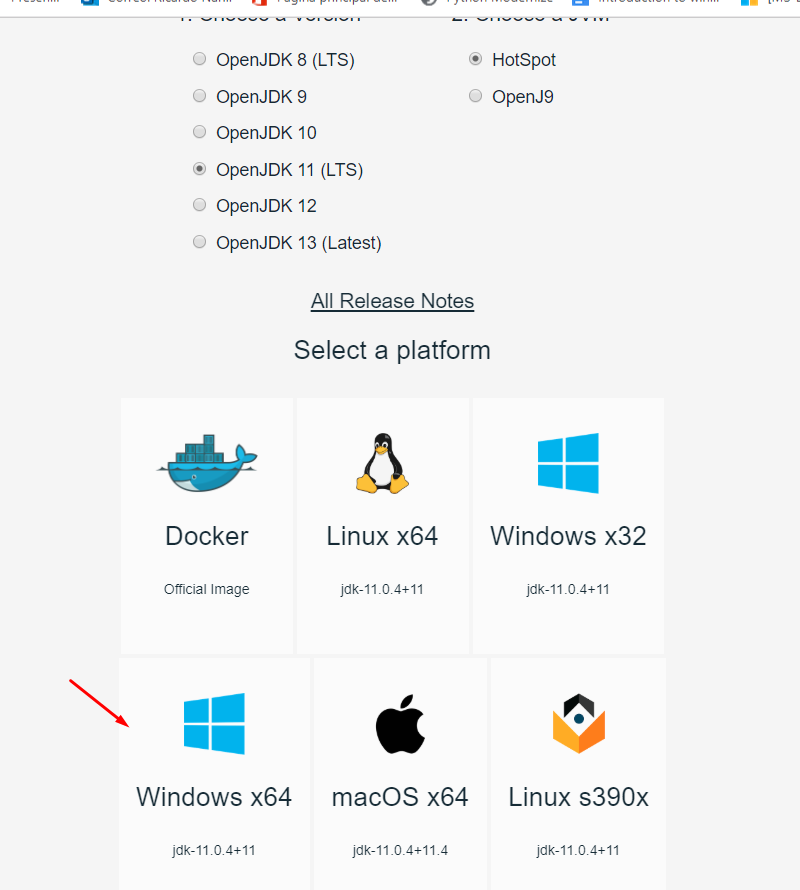
Allí se descargan el zip, en mi caso se llamaba ghidra\_9.1-BETA\_DEV\_20190923.zip y lo descomprimo en algún lugar, yo cree una maquina virtual de VMWARE, sin conexión a Internet, la cual uso solo para GHIDRA, no es que desconfíe de la NSA pero mejor, que trabaje allí.

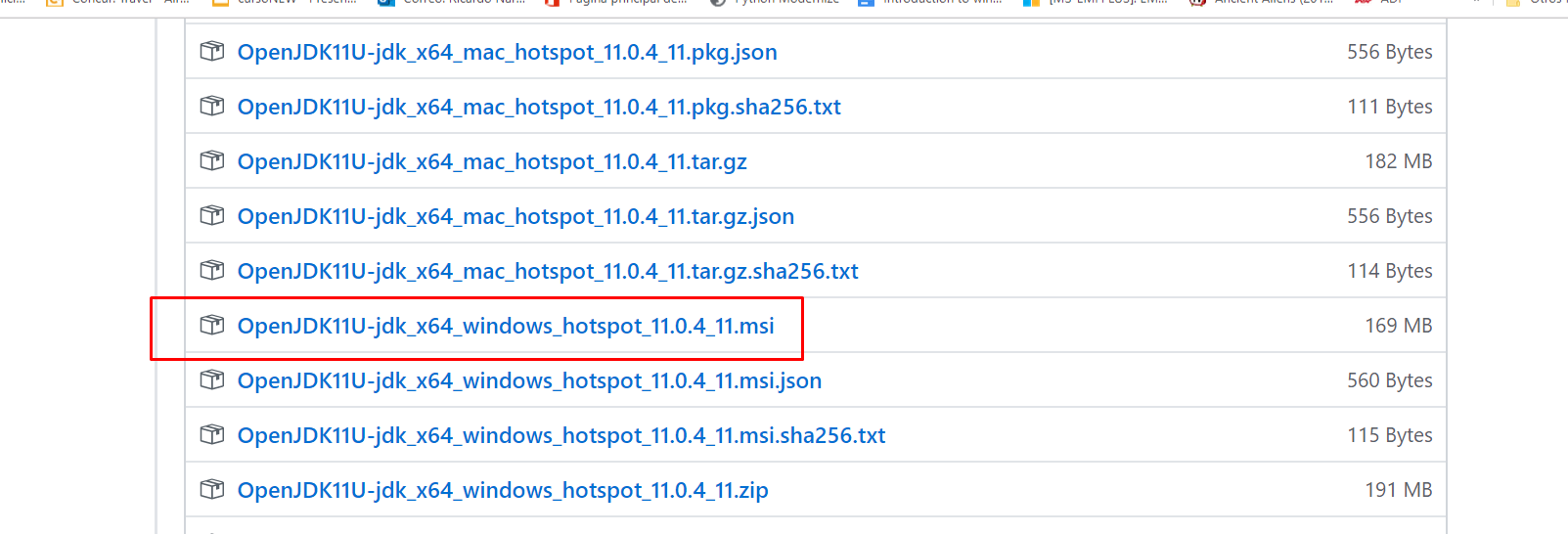
Una vez descomprimido hay que instalar un java compatible, en la pagina de Oracle se puede bajar o de otros sitios en mi caso baje este instalador y luego de terminar de instalarlo, agregar el path a donde esta el bin de java, a las environment variables.

La versión 11, en la página de Ghidra dice que es la más compatible.

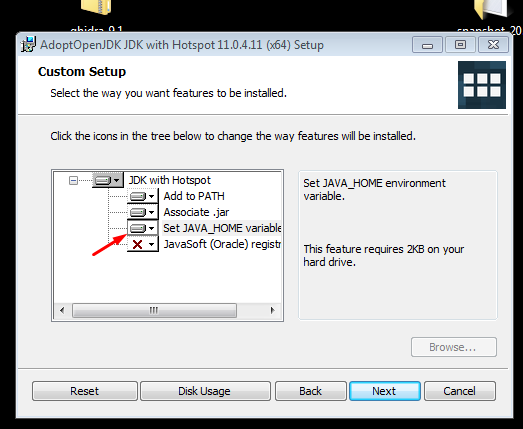
También dicen los que lo usaron que la versión 11 de OpenJDK es la que mejor va para GHIDRA.

<https://adoptopenjdk.net/releases.html?variant=openjdk11&jvmVariant=hotspot>

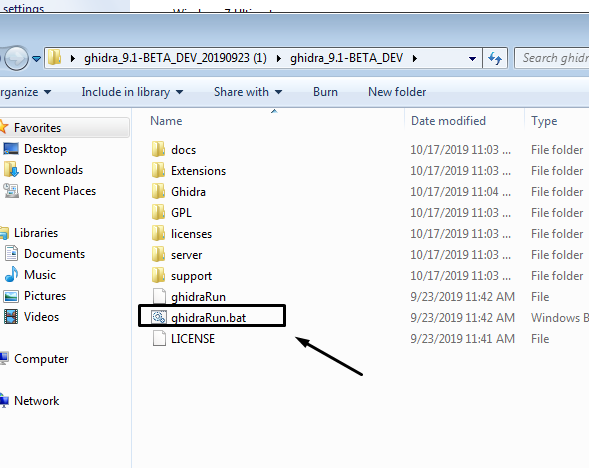




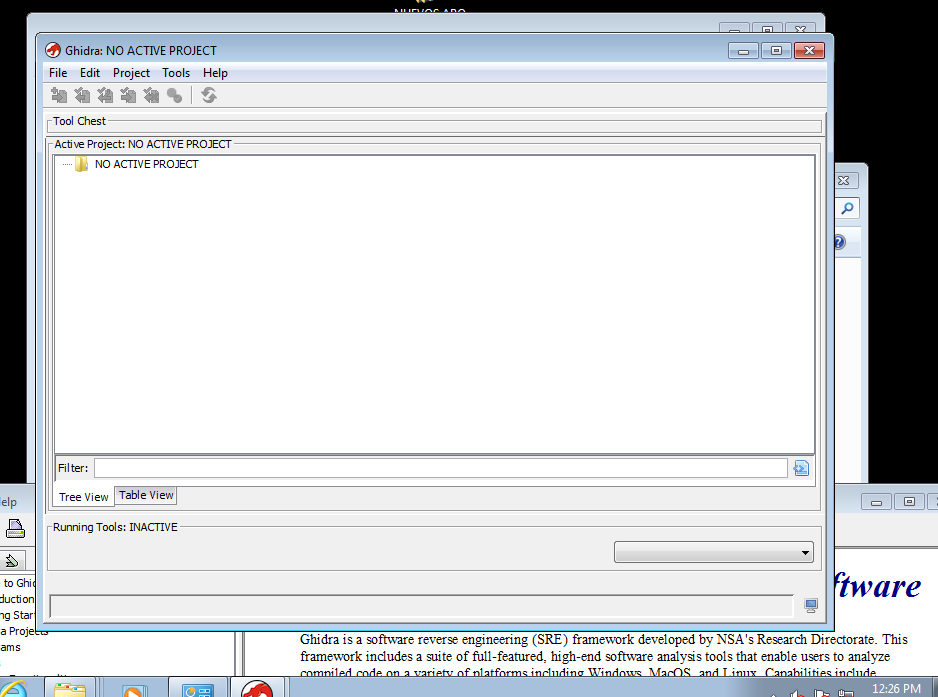
Una vez descargado el instalador tiene la opción para agregar a la environment variable Path, la ruta al binario java.exe



Una vez terminada la instalación de Java, ya se puede ejecutar.



Ejecuto ese bat y arranca.



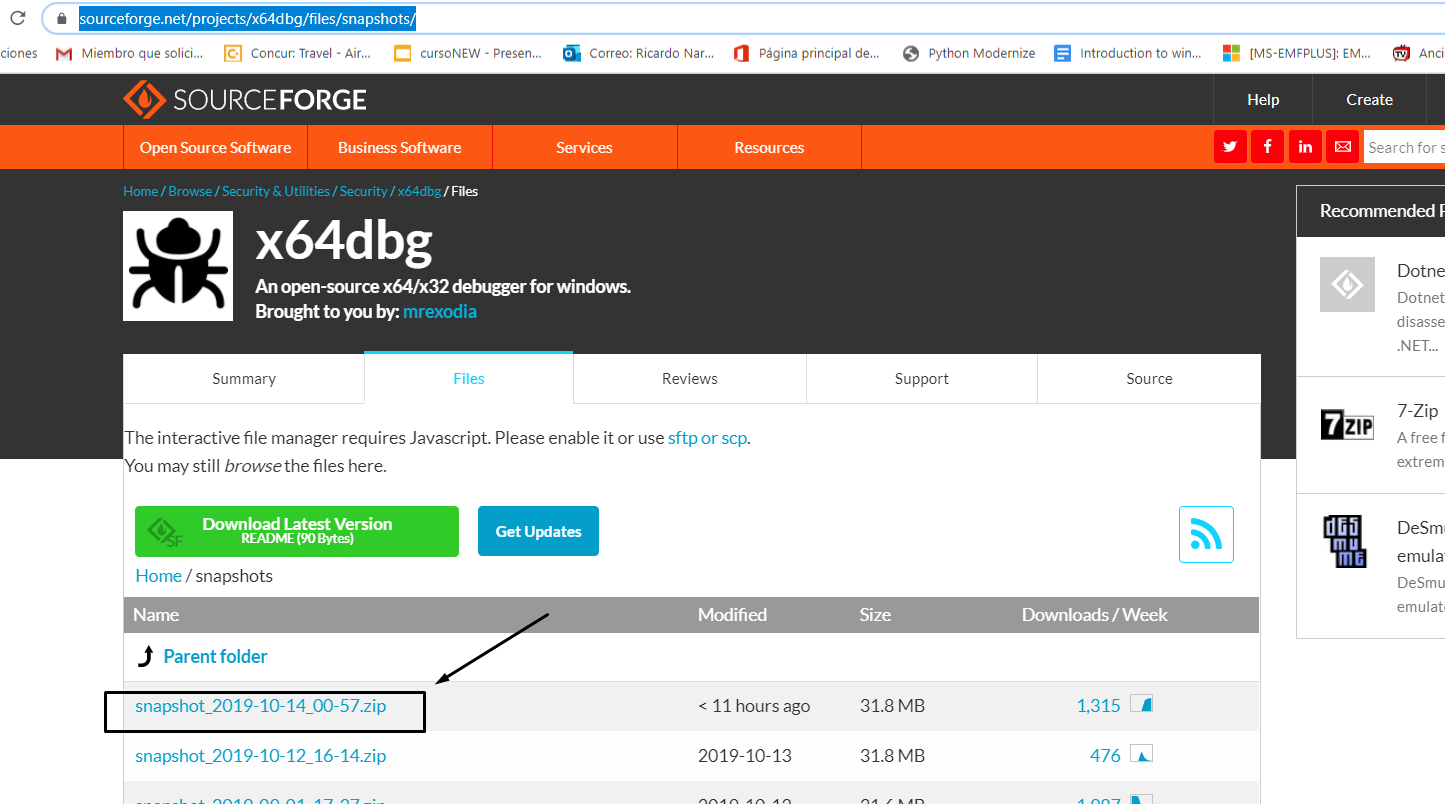
Bueno al menos ya tenemos esto listo también

**INSTALAR X64DBG**

De este programa salen nuevos snapshots casi a diario

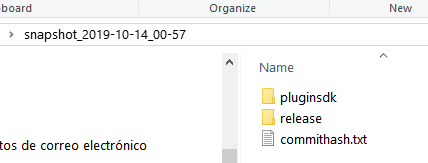
<https://sourceforge.net/projects/x64dbg/files/snapshots/>

Por supuesto bajen el mas nuevo que haya en ese momento

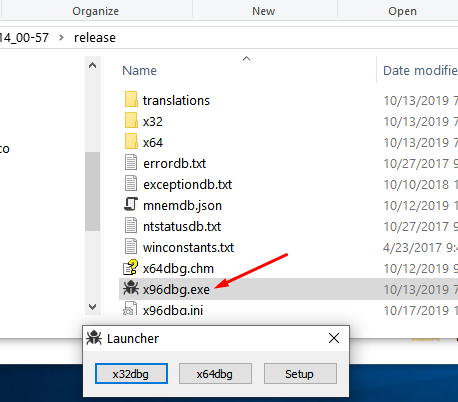


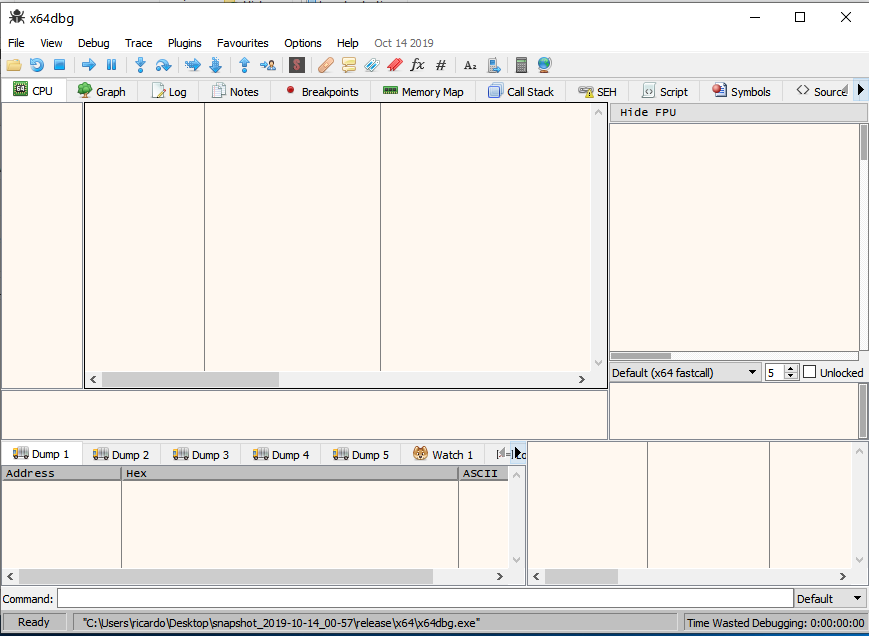
En mi caso es snapshot\_2019-10-14\_00-57.zip

Al unzipearlo vamos dentro de la carpeta reléase.



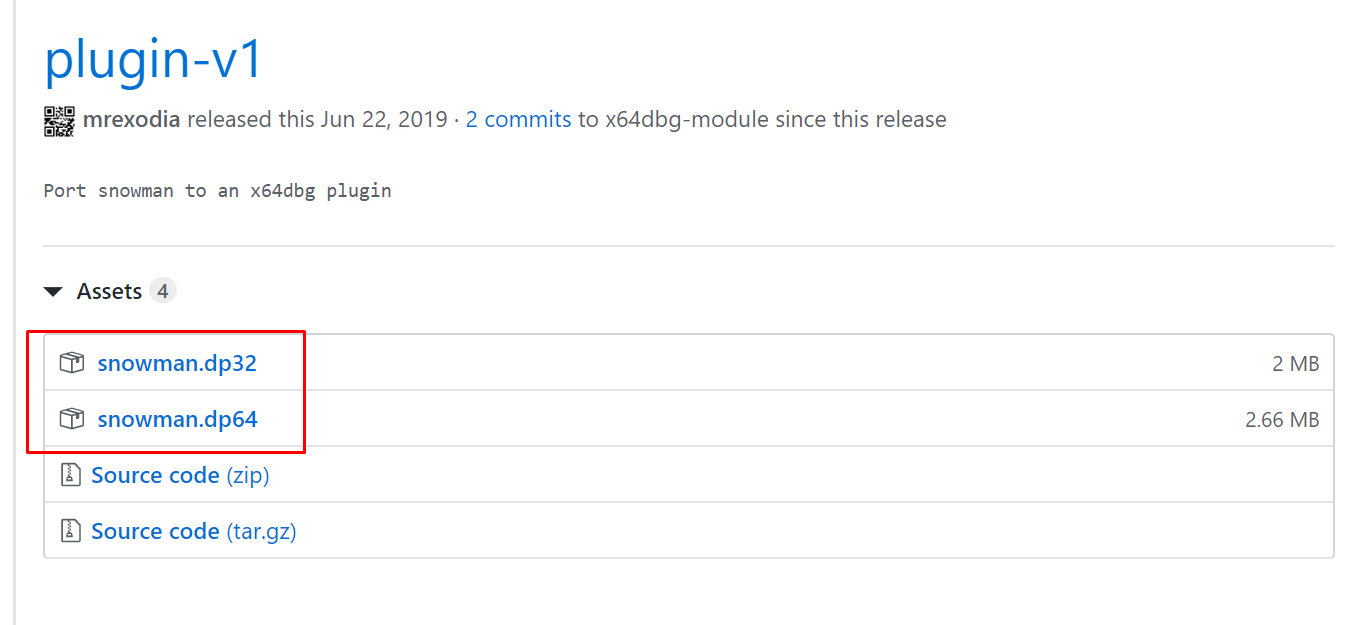
Al ejecutar como administrador, nos aparece un launcher para que elijamos la versión para 32 o 64 bits según necesitemos.



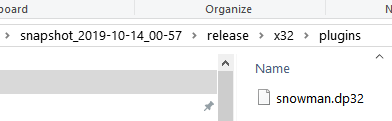


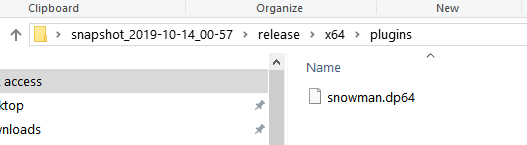
Podemos bajar el plugin SNOWMAN para decompilar, ya que el IDA FREE no trae el decompilador y agregárselos a la carpeta plugins

<https://github.com/x64dbg/snowman/releases/tag/plugin-v1>



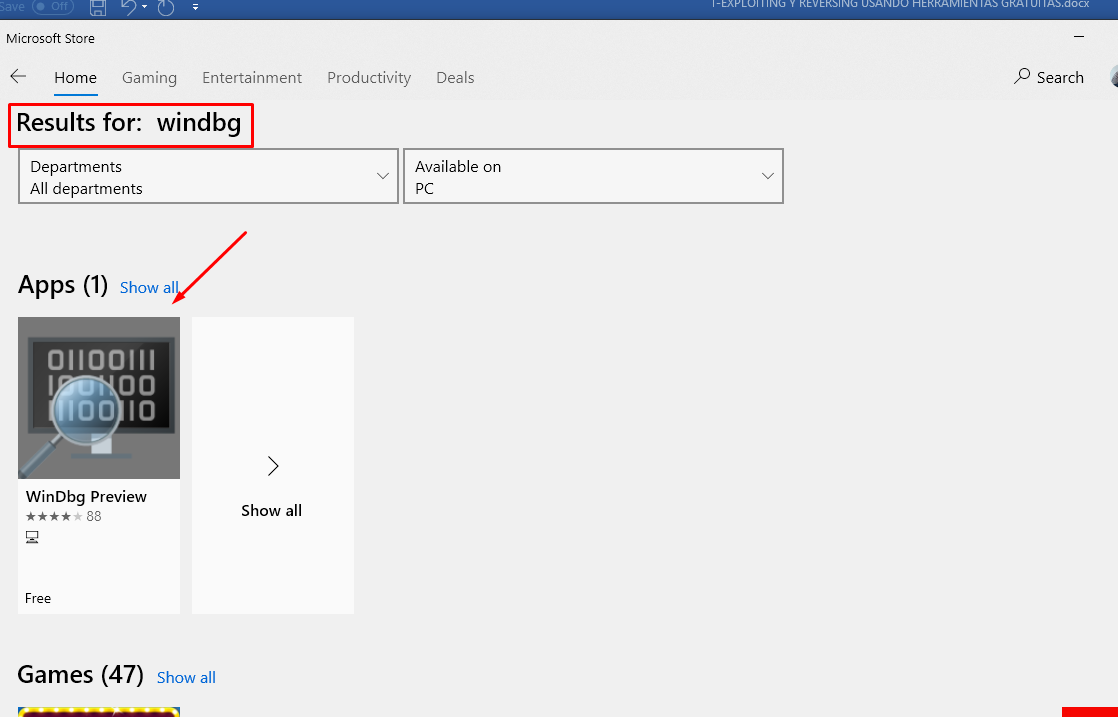
El de 32 bits lo copiamos dentro de la carpeta plugins de la versión de 32 bits y el de 64 bits en la correspondiente





**INSTALAR WINDBG**

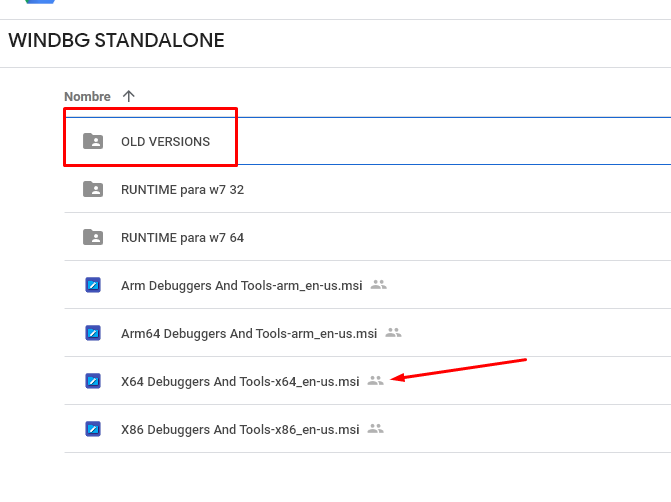
Si tienen Windows 10 instalar Windbg es ir a Microsoft STORE y buscar WINDBG e instalarlo desde allí.



Desde allí se instala automáticamente WINDBG PREVIEW que es la versión más nueva, si tienen Windows 7 tendrán que instalar una versión anterior de Windbg.

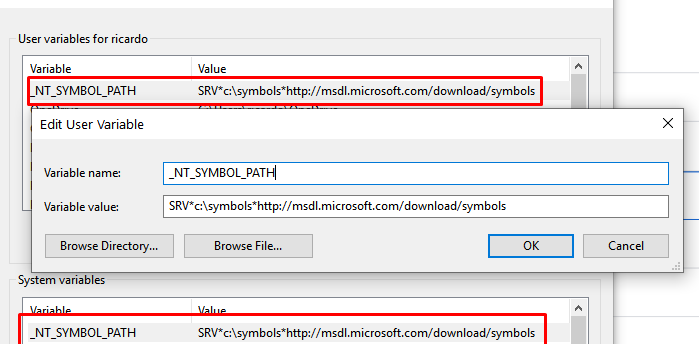
<https://drive.google.com/open?id=1UEPBecOoir-nhyRK-RHOAhRq05ZtlW1y>

Allí hay varias versiones algunas mas viejas otras no tanto,



La más nueva para Windows 7 que tengo es esa, si no les va hay en OLD VERSIONS algunas anteriores que seguro irán.

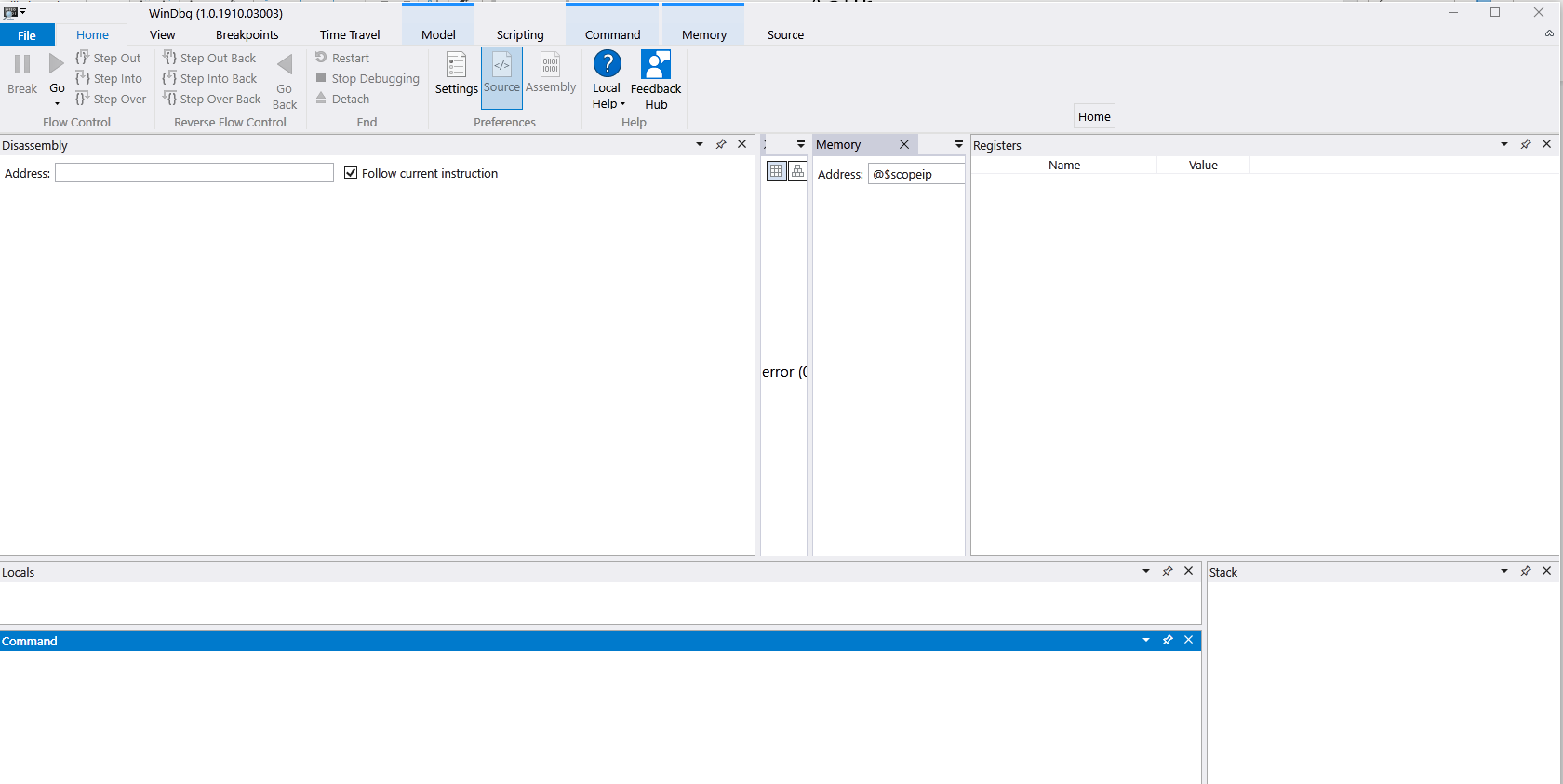
Lo siguiente es configurar los símbolos para Windbg, crear la carpeta llamada **symbols** en C:/ y luego ir a las environment variables y agregar la variable \_NT\_SYMBOL\_PATH.



Colocarle como valor

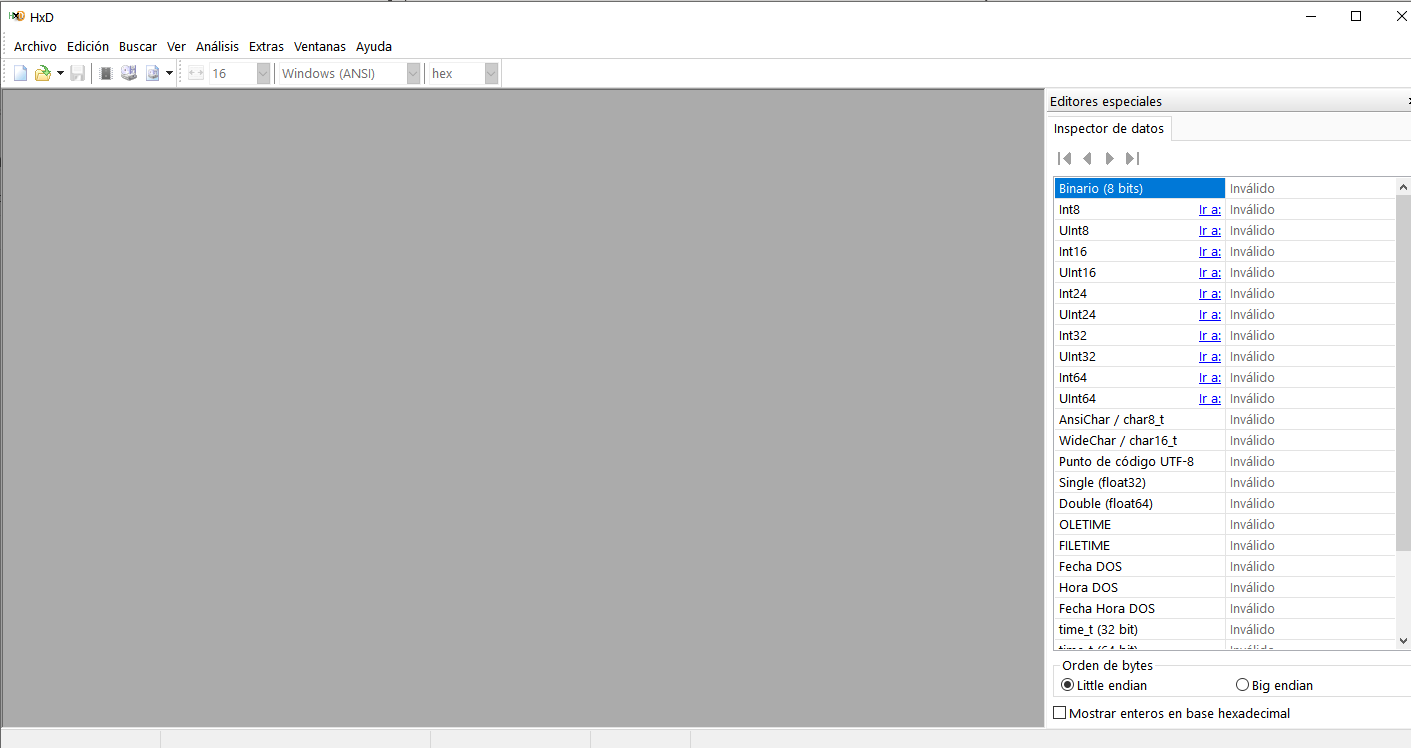
**SRV\*c:\symbols\*http://msdl.microsoft.com/download/symbols**

Con eso ya tendríamos instalado Windbg PREVIEW.



Luego se puede instalar algún editor hexadecimal como el HxD que es gratuito.

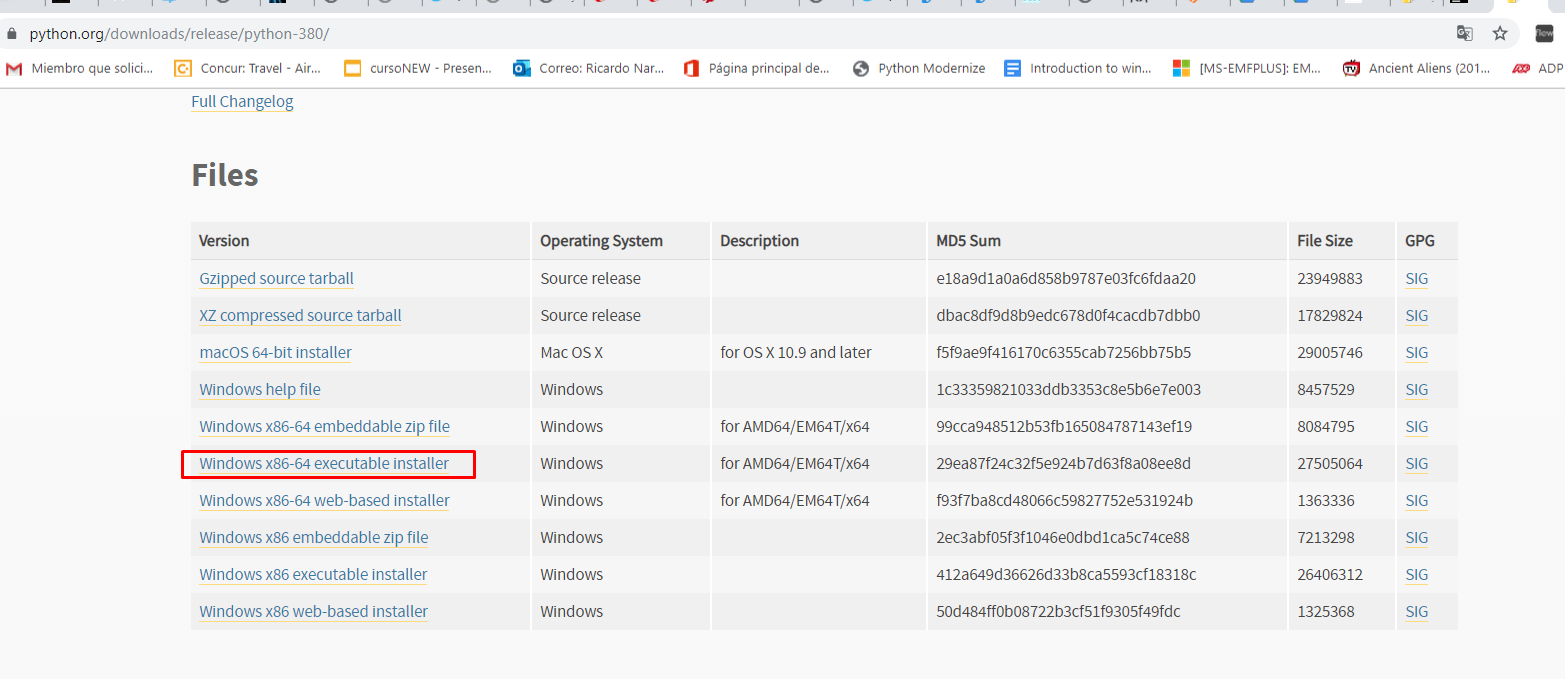
<https://mh-nexus.de/en/hxd/>



**INSTALAR PYTHON**

Instalaremos Python 3 que es el más nuevo, para poder armar los exploits para cada ejercicio.

<https://www.python.org/downloads/>



Sera la última versión que exista en el momento que la bajen, en mi caso es esta.

python-3.8.0-amd64.exe

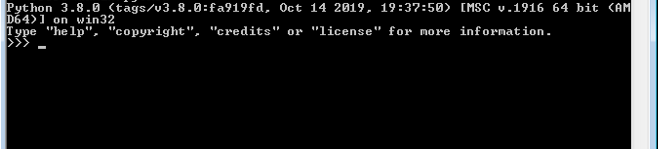
Para agregar en la environment variable PATH la ruta al ejecutable Python.exe, el instalador tiene una tilde para que lo haga automáticamente.

Normalmente el path está en

C:\Users\**XXXXX**\AppData\Local\Programs\Python\Python38

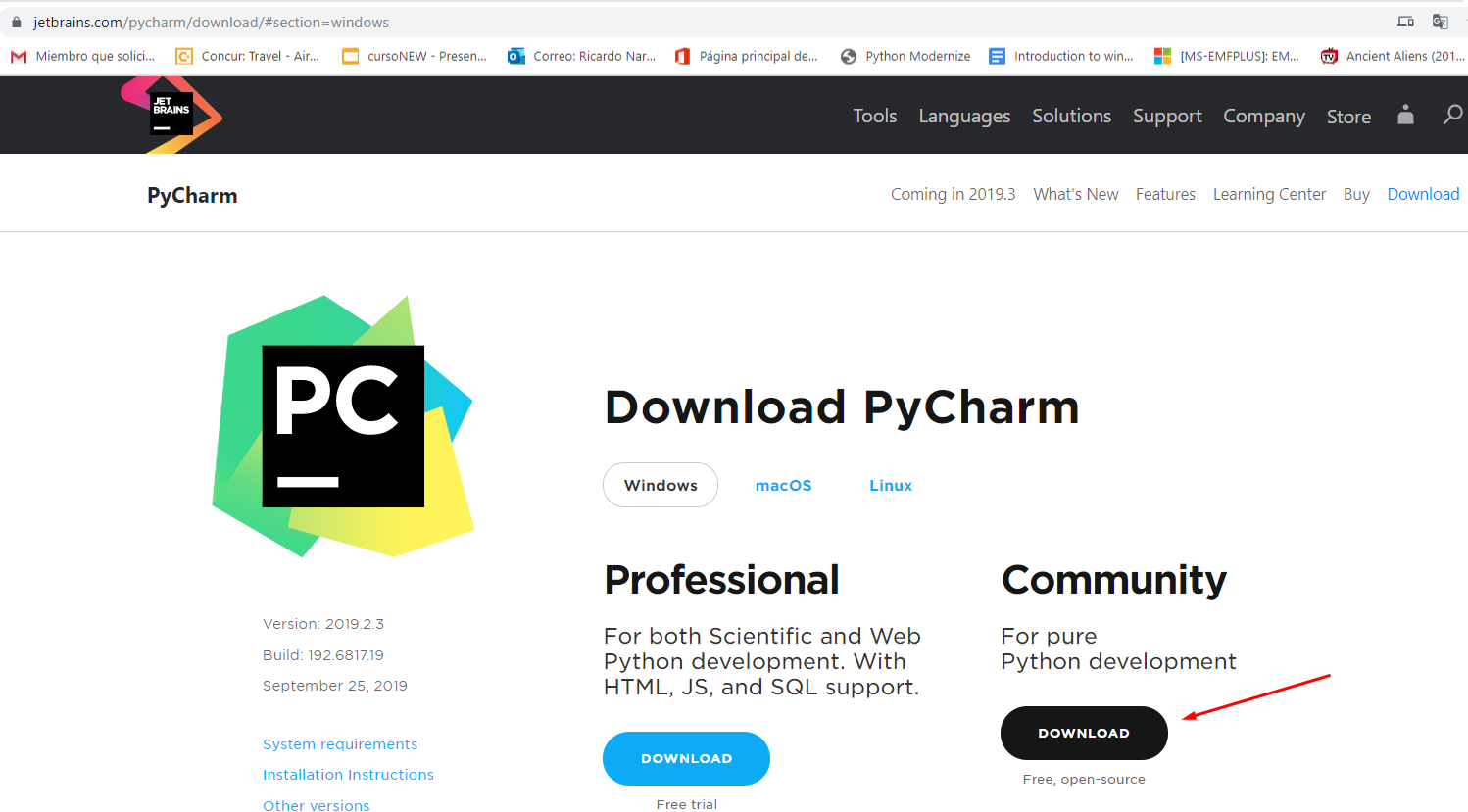
Cambiará Python38 por la versión actual.

El tema es que al tipear en la command prompt podamos ejecutar Python y lo reconozca.

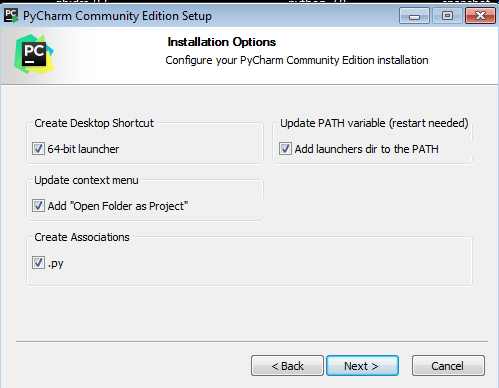


**INSTALAR PYCHARM COMMUNITY**

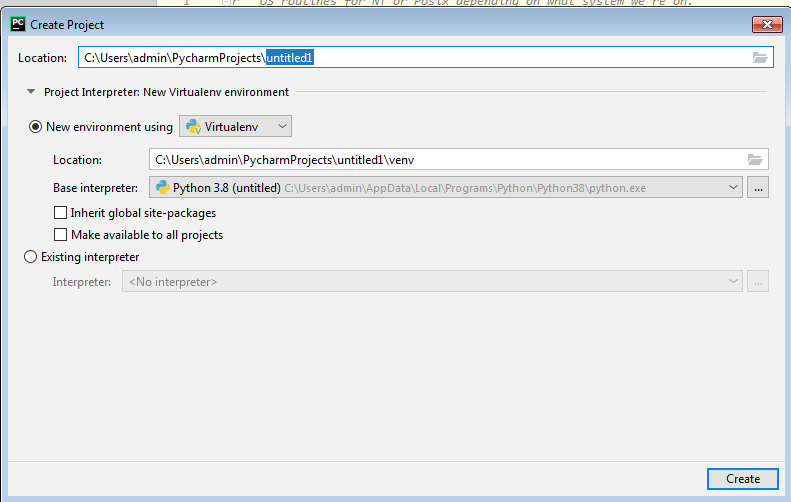
<https://www.jetbrains.com/pycharm/download/>



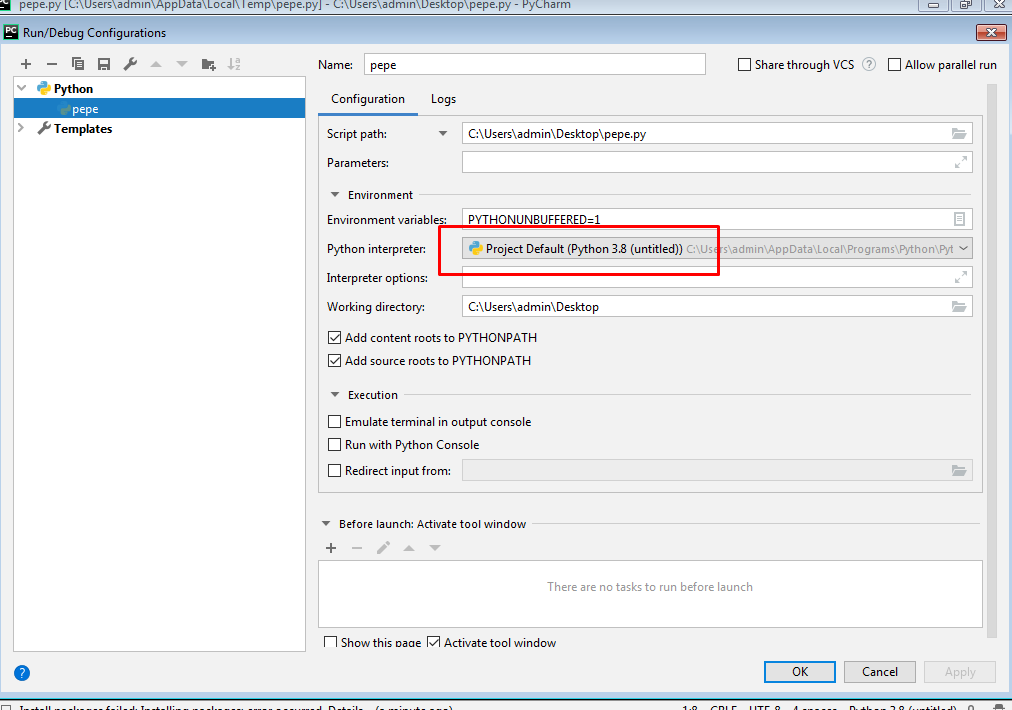
En mi caso es pycharm-community-2019.2.3.exe será la última versión que exista la que ustedes bajen en su momento.



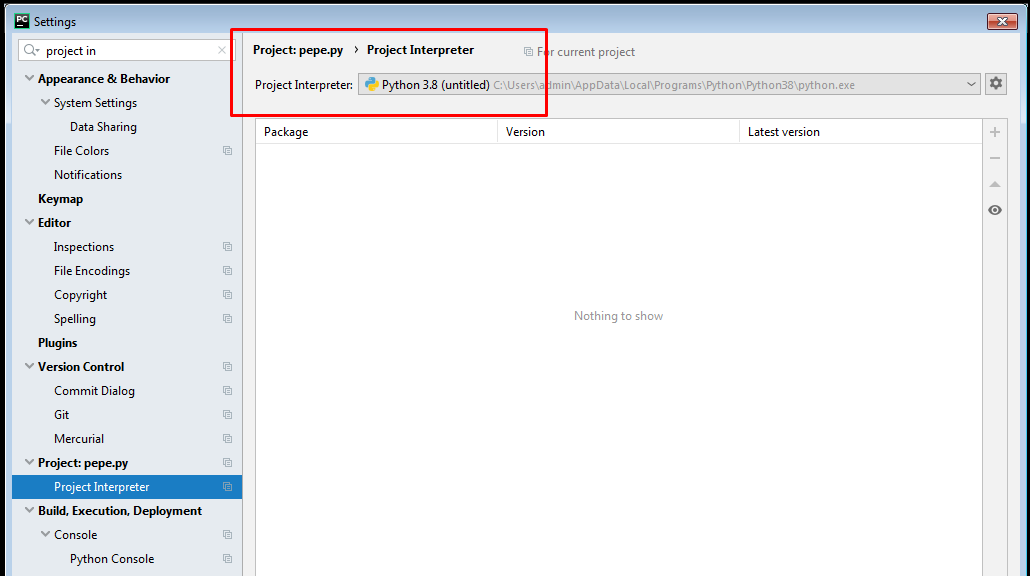
Creamos un proyecto una vez instalado.



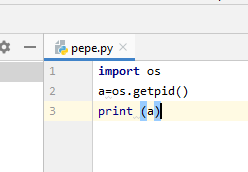
En RUN-DEBUG configuration mirar que este el interprete allí.



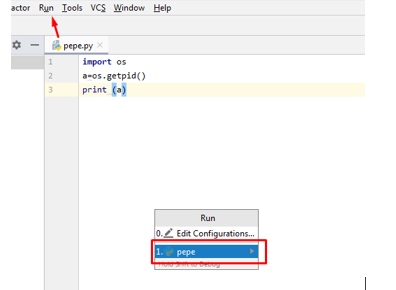
También en SETTINGS buscar PROJECT INTERPRETER y ver que este detectado Python 3.8 o el que sea en su caso.

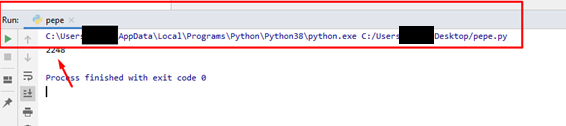


La cuestión es que si hago un pepe.txt lo renombro a pepe.py y lo arrastro allí

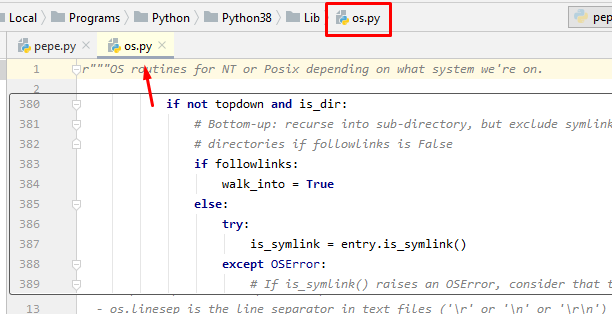


Le escribo un pedazo de código al ejecutarlo con RUN debería imprimir.





Y además debe autocompletar y si pasamos el mouse por ejemplo la palabra os, al apretar CTRL y hacer clic allí, debería abrir la librería correspondiente os.



Con eso terminamos la parte 1, con la instalación de las herramientas que vamos a utilizar, la parte 2 comenzara con buffer overflow un poco de teoría y los primeros ejercicios sencillos algunos hare yo, otros les dejare para que hagan ustedes.

Hasta la parte 2

Ricardo Narvaja (www.ricardonarvaja.info)

18/10/2019