

MANUAL DEL SIMULADOR

La realización de este proyecto tiene como objetivo simular al microprocesador Z80, por lo que es importante seguir las instrucciones que en este manual se proporcionan.

Para ejecutar el simulador es necesario tener instalado java.

Empezaremos por descargar el JDK de la página oficial de Oracle (<http://www.oracle.com/technetwork/java/javase/downloads/index.html>) y en la sección de descargas buscamos la versión Java SE, en esta parte descargaremos el JDK. Tenemos que aceptar los términos y condiciones de licencia. Bajaremos la versión que corresponda a nuestro sistema operativo.

The screenshot shows the Oracle Technology Network Java SE Downloads page. On the left, there's a sidebar with links for Java SE, Java EE, Java ME, Java SE Support, Java SE Advanced & Suite, Java Embedded, JavaFX, Java DB, Web Tier, Java Card, Java TV, New to Java, Community, Java Magazine, and Java Advanced. The main content area has tabs for Overview, Downloads, Documentation, Community, Technologies, and Training. Under the Downloads tab, there are buttons for Latest Release, Next Release (Early Access), Embedded Use, and Previous Releases. Below these are three download buttons: Java Platform (JDK) 7u21, JavaFX 2.2.21, and JDK 7 + NetBeans. To the right, there's a sidebar for Java SDKs and Tools with links for Java SE, Java EE and Glassfish, Java ME, JavaFX, Java Card, and NetBeans IDE. Another sidebar for Java Resources includes links for New to Java?, APIs, Code Samples & Apps, Developer Training, Documentation, Java.com, Java.net, Student Developers, and Tutorials. At the bottom, there's an advertisement for Java magazine with a 'NEW!' badge and a link to get it for FREE!

Java SE Development Kit 7u21
Debe aceptar el Contrato de licencia del código binario de Oracle para Java SE para descargar el software.

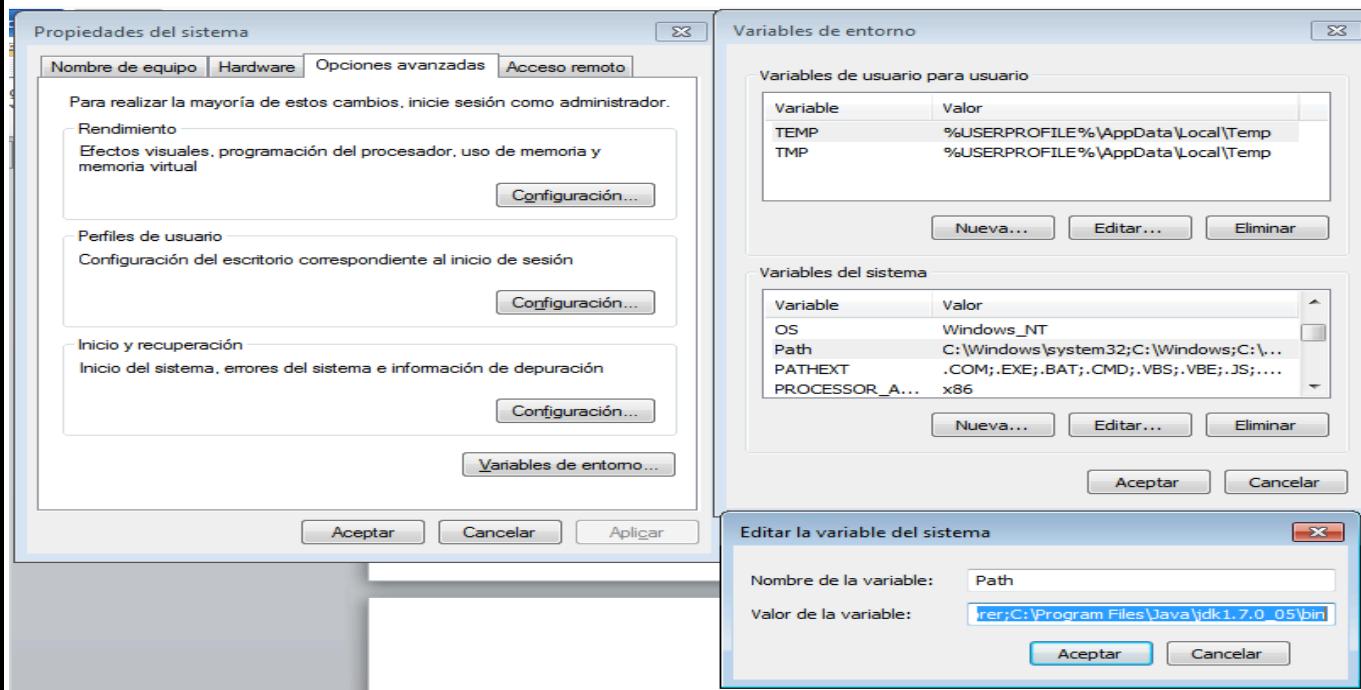
Acepta el acuerdo de licencia Acuerdo de licencia Decline

Producto / Descripción del archivo	Tamaño del archivo	Descargar
Linux ARM v6/v7 Soft Float ABI	65.09 MB	jdk-7u21-linux-arm-sfp.tar.gz
Linux x86	80.35 MB	jdk-7u21-linux-i586.rpm
Linux x86	93.06 MB	jdk-7u21-linux-i586.tar.gz
Linux x64	81.43 MB	jdk-7u21-linux-x64.rpm
Linux x64	91.81 MB	jdk-7u21-linux-x64.tar.gz
Mac OS X x64	144.18 MB	jdk-7u21-macosx-x64.dmg
Solaris x86 (SVR4 paquete)	135.84 MB	jdk-7u21-solaris-i586.tar.Z
Solaris x86	92.08 MB	jdk-7u21-solaris-i586.tar.gz
Solaris x64 (SVR4 paquete)	22.67 MB	jdk-7u21-solaris-x64.tar.Z
Solaris x64	15.02 MB	jdk-7u21-solaris-x64.tar.gz
Solaris SPARC (SVR4 paquete)	136.09 MB	jdk-7u21-solaris-sparc.tar.Z
Solaris SPARC	95.44 MB	jdk-7u21-solaris-sparc.tar.gz
Solaris SPARC de 64 bits (SVR4 paquete)	22.97 MB	jdk-7u21-solaris-sparcv9.tar.Z
Solaris SPARC de 64 bits	17.58 MB	jdk-7u21-solaris-sparcv9.tar.gz
Windows x86	88.98 MB	jdk-7u21-windows-i586.exe
Windows x64	90.57 MB	jdk-7u21-windows-x64.exe

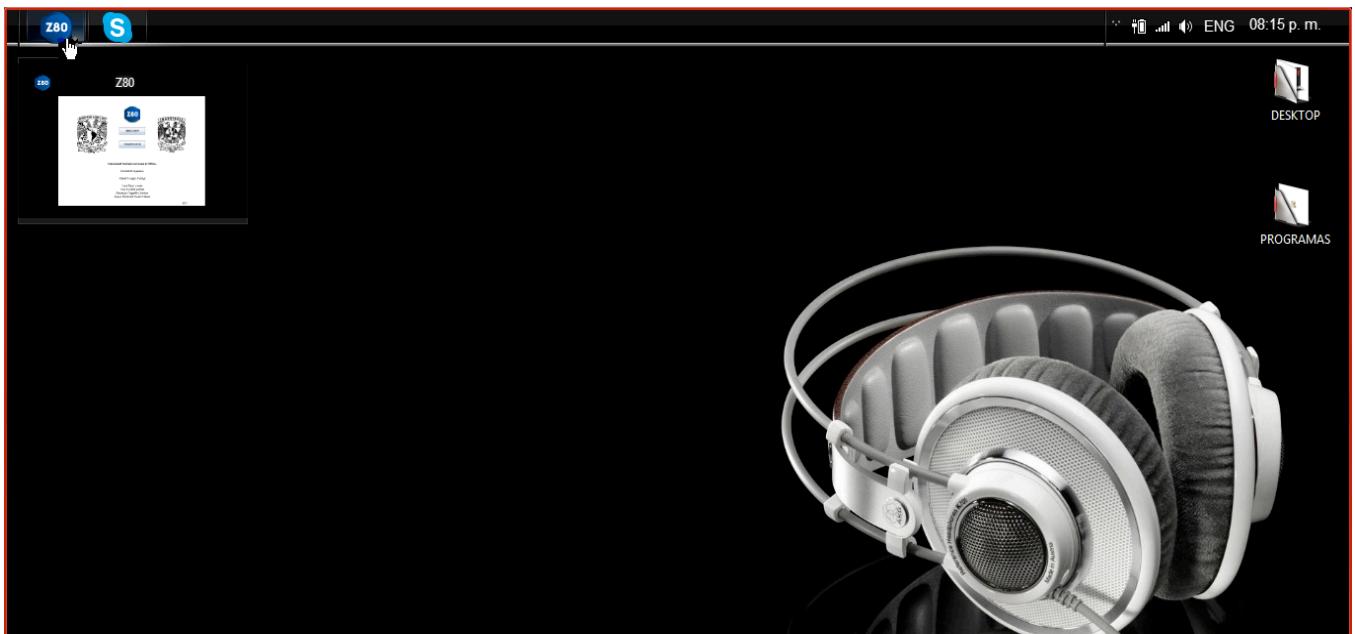
Java SE Development Kit 7u21 Demos y muestras Descargas
Java SE Development Kit 7u21 Demos y muestras Descargas se publican bajo la licencia BSD Oracle

Producto / Descripción del archivo	Tamaño del archivo	Descargar
Linux ARM v6/v7 Soft Float ABI	8.09 MB	jdk-7u21-linux-arm-sfp-demos.tar.gz
Linux x86	14.13 MB	jdk-7u21-linux-i586-demos.rpm
Linux x86	14.06 MB	jdk-7u21-linux-i586-demos.tar.gz
Linux x64	14.18 MB	jdk-7u21-linux-x64-demos.rpm
Linux x64	14.08 MB	jdk-7u21-linux-x64-demos.tar.gz
Mac OS X	12.8 MB	jdk-7u21-macosx-x86_64-demos.tar.gz

Se descarga y ejecuta siguiendo los pasos que se requieren. Lo siguiente es validar que esté instalado y acceder a la ruta de la carpeta de java. Se tienen dos carpetas, una es el jre7 o máquina virtual y la otra es el kit de desarrollo JDK que es la que nos interesa, al cual accedemos seleccionando la carpeta bin y copiaremos su ruta que en acto seguido pegaremos en path (variable de entorno) ubicada en la configuración avanzada del sistema. Por ultimo le daremos aceptar para guardar los cambios realizados.

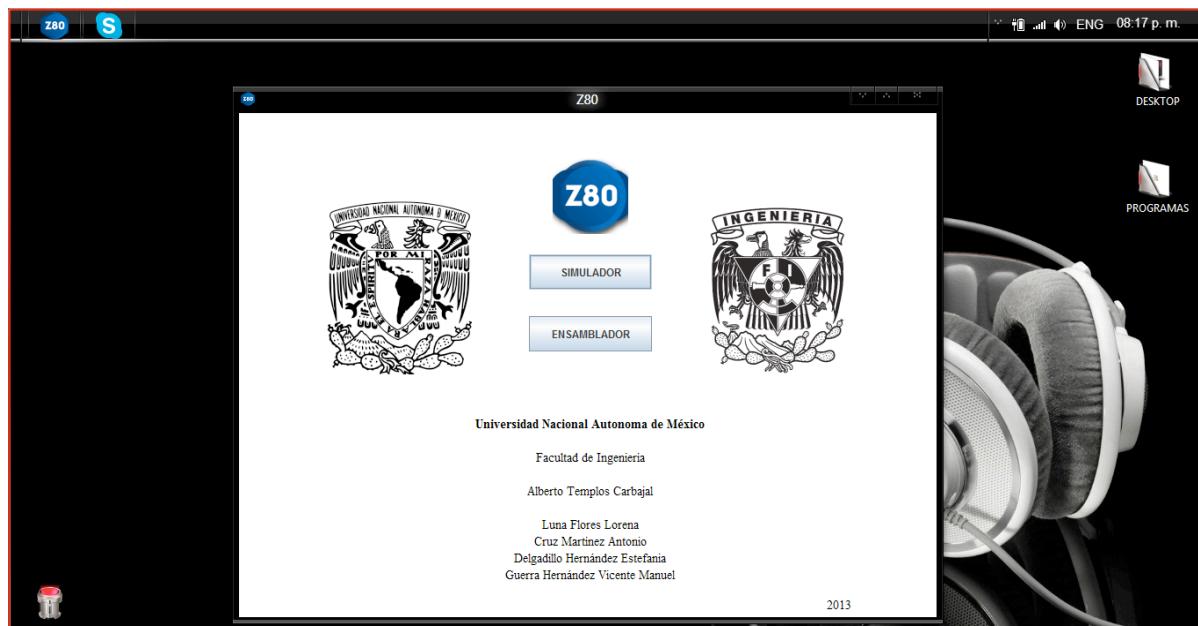


Comenzaremos por la pantalla de presentación del simulador que aparece al presionar el icono Z80, la cual se muestra a continuación:

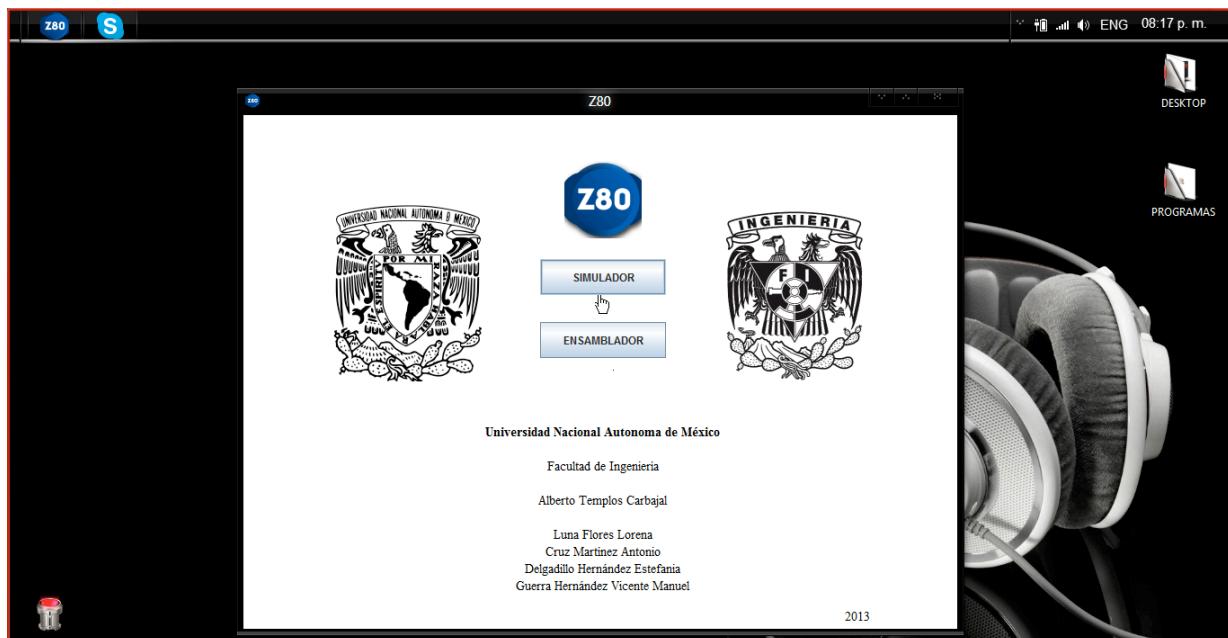


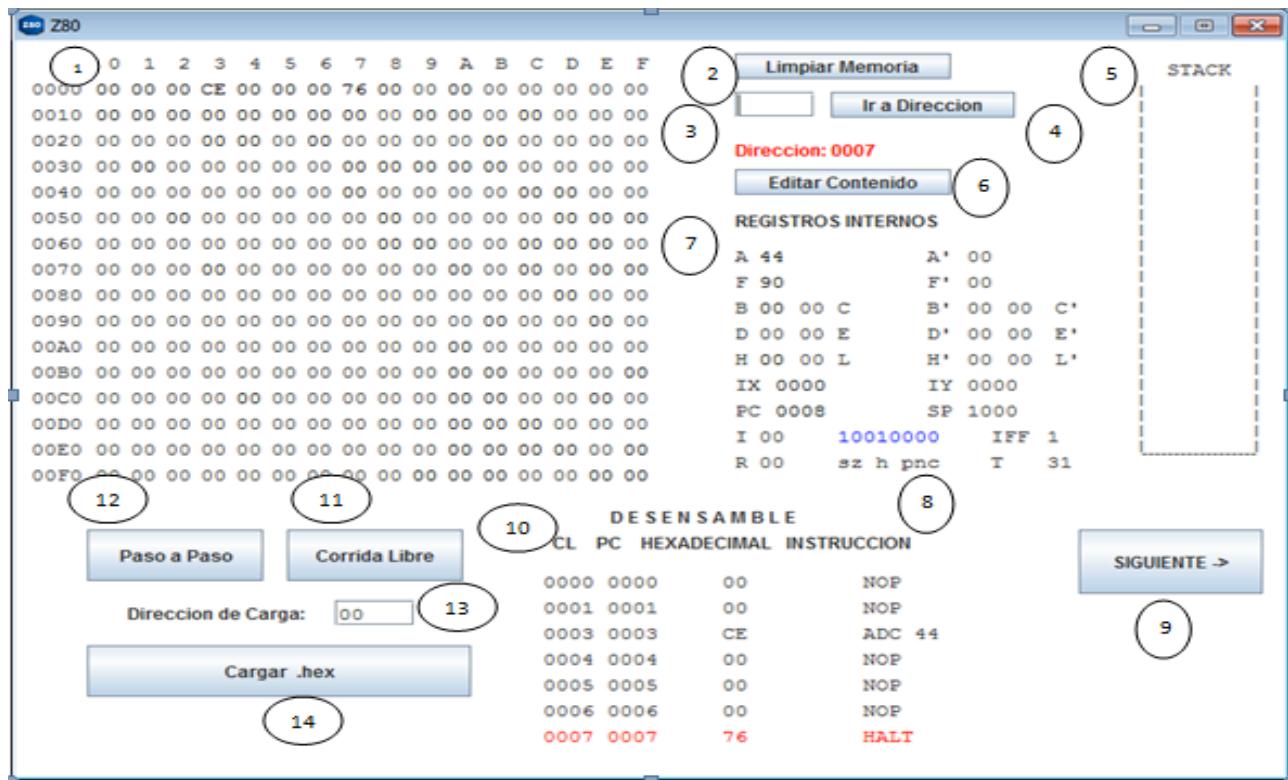
NOTA: Es necesario mencionar que el simulador solo funciona con archivos en hexadecimal, esto quiere decir que se debe de tener un archivo en este formato para que el simulador pueda funcionar , (para el ensamblador es necesario consultar el manual correspondiente).

En la siguiente pantalla se muestran las dos opciones a elegir.



Al presionar la opción simulador aparecerá una ventana como la que se muestra a continuación:



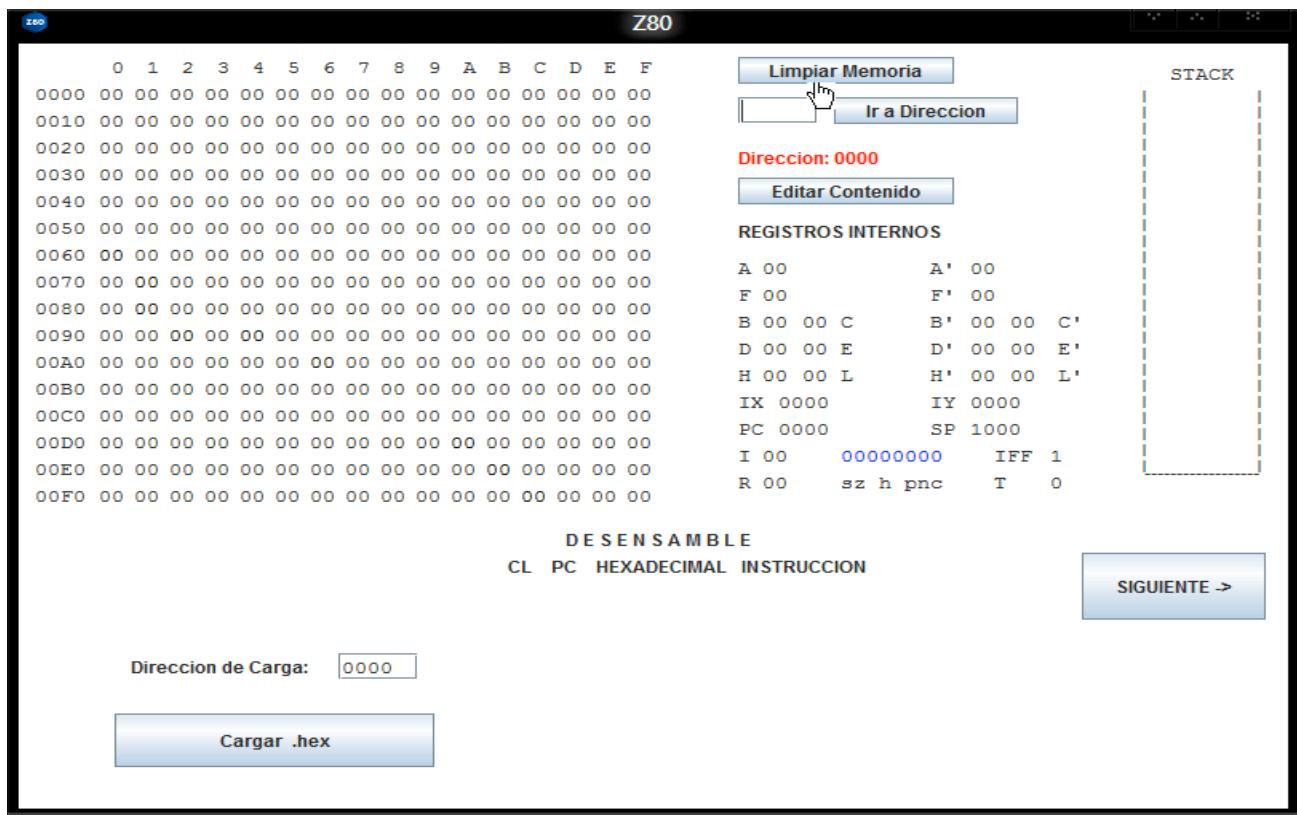


A continuación se explicarán cada elemento de esta interfaz.

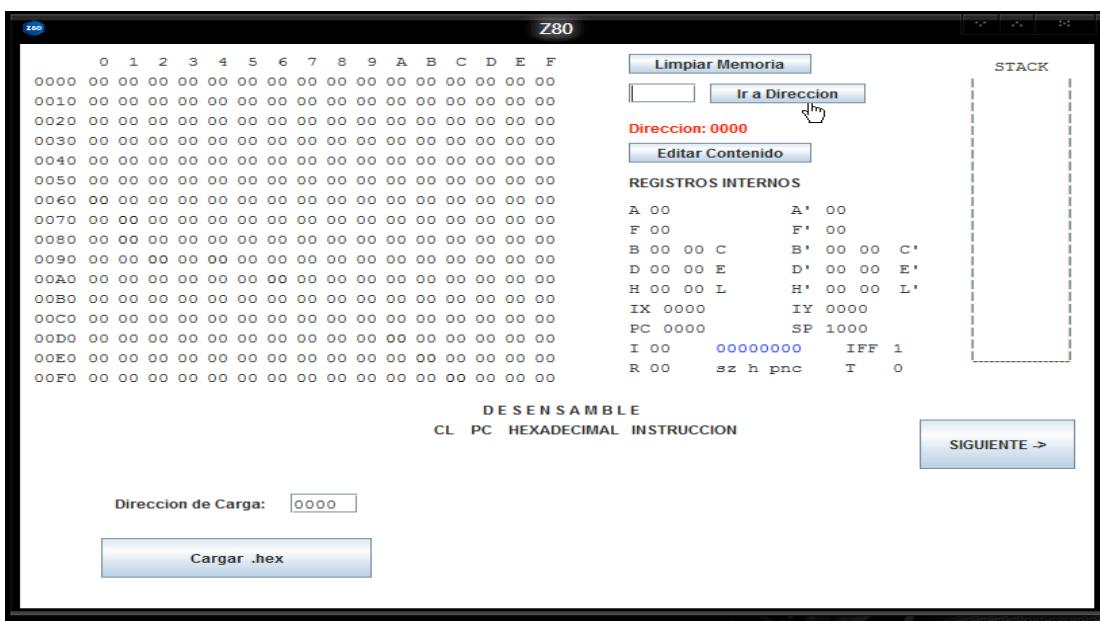
1.- Nos localizamos en la memoria la cual tiene varios usos:

- La primera es que cuando cargamos un archivo hexadecimal nos coloca en cada espacio las instrucciones (de 8 bits cada una) y cuando estamos en el desensamblaje se muestran las instrucciones.
- Puede editar el contenido directamente sin tener que estar en el código para cambiarlo.
- Al momento de pedir otra dirección de memoria es necesario moverse con los botones correspondientes.

2.-Botón “Limpiar Memoria” nos ayuda para poder cargar otro archivo o simplemente limpiar la memoria.

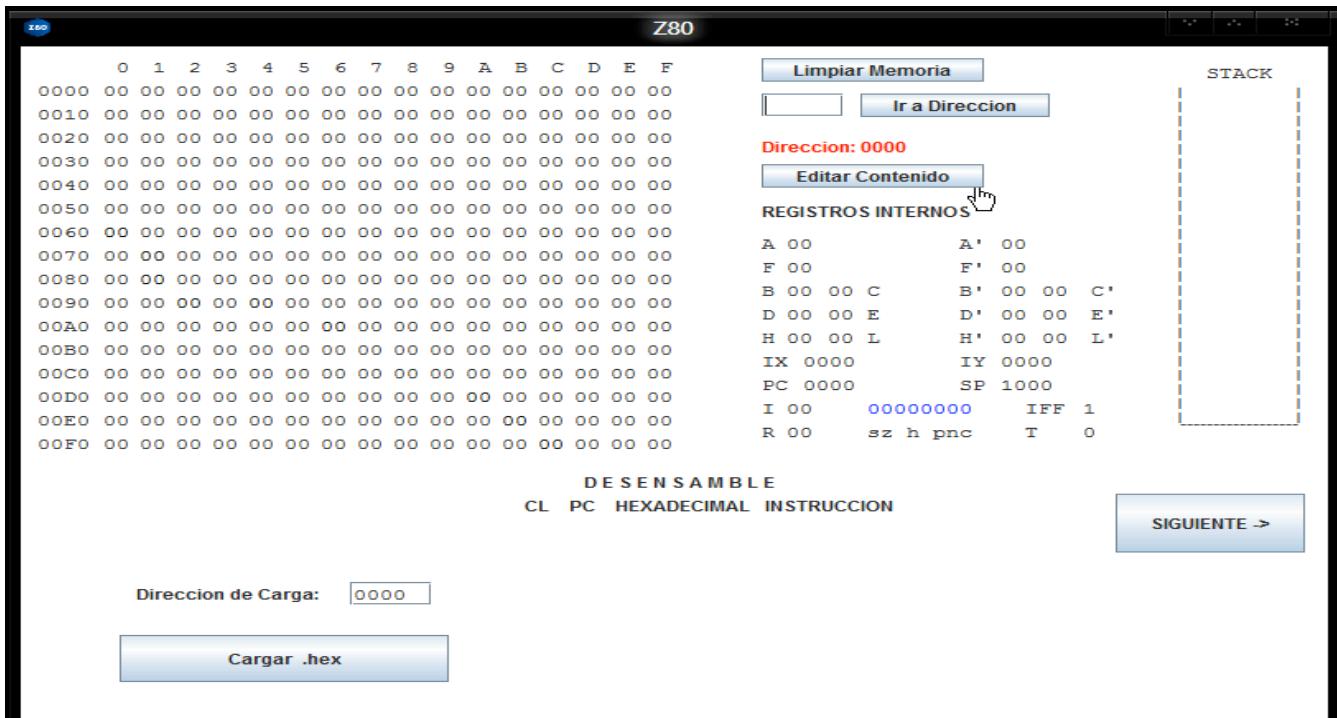


3.- Los incisos 3 y 4 están relacionados evidentemente puesto que el 3 es el espacio de cuatro caracteres (ya que es una dirección y no puede tener más dígitos), si se desea cambiar de dirección para ver el contenido de cierta localidad de memoria o cambiar su contenido hacemos uso de este espacio. El 4 es el botón “Ir a Dirección” este obedece la instrucción que menciona, contando como dato el contenido del inciso 3 para poder ir a esa localidad de memoria y no a cualquiera.



5.- Stack o pila en español, esta nos muestra cómo se usa la pila y su contenido. Recordando que la pila tiene un llenado del tipo LIFO esto quiere decir “primeras entradas, últimas salidas” donde los últimos datos introducidos son los primeros en salir. En este espacio de la interfaz se sigue la metodología anterior y se llena según el programa que estemos corriendo en ese momento.

6.- La manera de poder cambiar el contenido y evidentemente el trayecto de un programa es editarlo desde su archivo hexadecimal antes de cargarlo, cambiarlo directamente en la memoria o con ayuda de este botón “Editar Contenido”.



Como podemos observar al hacer click en “Editar Contenido” nos aparece ahora un espacio en blanco y el botón de “Actualizar”, esto quiere decir que podemos editar el contenido de la localidad que deseemos. En el espacio en blanco se escribe el contenido y es solo para dos caracteres puesto que solo se permite un byte por localidad.

7.- Registros Internos es la lista de todos los registros que se usan en el programa que se esté ejecutando, a continuación se describen:

- a) El registro A es el más importante y usado en los programas, se le asigna el nombre “Acumulador” ya que es sobre quien se realizan operaciones básicas, lógicas y se puede cambiar su contenido desde el comienzo en la interfaz, en la corrida libre nos muestra hasta el final su último valor asignado y en “paso a paso” tenemos el valor del acumulador de cada instrucción viendo así cómo cambia.
- b) Esta el registro F que es el encargado de llevar el estado del procesador o la cuenta y las modificaciones de nuestras banderas, en este registro es donde caen todos los valores de estas asignándole un valor según las banderas que estén prendidas o apagadas, de la misma manera que el acumulador se puede ver su contenido en la “Corrida Libre” y en el “Paso a paso”.
- c) Los registros B, C, D, E, H, L son registros de un byte que nos permiten guardar u operar en nuestro programa puesto que son variables, que de la misma manera que el acumulador podemos editar al inicio y en las dos diferentes modalidades del programa.
- d) Los registros con apostrofe son los registros secundarios que sirven para respaldar a los principales, debido a que son una copia de los principales no pueden ser editados como los anteriores.
- e) Los registros IX e IY tienen cuatro espacios porque son registros de índice que apuntan o señalan a una dirección de una localidad de memoria es por eso que son de 2 byte, se pueden modificar en el inicio y ver su contenido en “Corrida Libre” y en el “Paso a paso”.
- f) El PC contiene la dirección de la siguiente instrucción próxima a ejecutarse.
- g) El SP es un apuntador a una dirección de una localidad de la pila, este registro nos ayuda a movernos sobre la pila teniendo así un control de los datos y no se pueden editar como los anteriores.
- h) Registro I es un registro de un byte, es el registro de Interrupciones y no se pueden editar.
- i) Registro R es un registro de dos bytes que se usa cuando se quiere refrescar la memoria, contiene la dirección de localidades de memoria a refrescar y no se pueden editar.
- j) En este espacio también tenemos al tiempo de ejecución declarado con la letra T que es la suma de todos los tiempos de reloj de cada instrucción que contiene el programa, no se puede editar como los anteriores.
- h) IFF: Indicador que permite o prohíbe las interrupciones.

8.- Las banderas, estas banderas nos ayudan a saber el estado de nuestro procesador no se puede editar como los anteriores y van cambiando su valor conforme avanza el programa, tenemos 6 banderas que son las siguientes:

I) S: La bandera de Signo, esta bandera se enciende o se apaga según el signo del registro que se esté usando.

II) Z: La bandera de cero, esta bandera se enciende o se apaga si el registro que se esté usando es nulo o tiene algún valor.

III) H: La bandera de medio acarreo, esta bandera se enciende o se apaga si en la operación de los primeros 4 bits se tiene un acarreo.

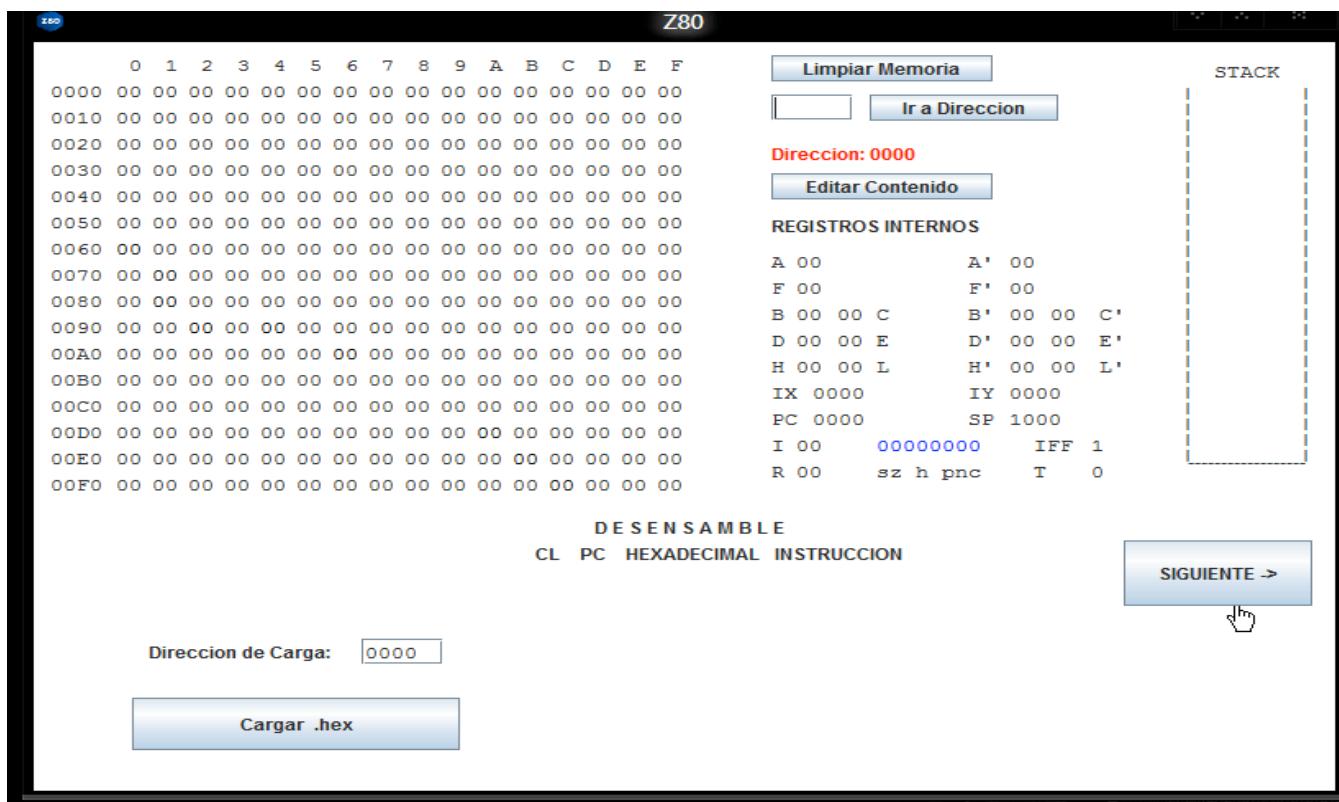
IV) P/V: La bandera de Paridad u over flow, esta bandera se enciende o se apaga según el resultado de la operación que se esté realizando, si es una operación lógica se va a alterar P y si es una operación aritmética se alterará a V.

V) N: La bandera de negativo, esta bandera se enciende o se apaga según el resultado del registro que se esté usando, si el resultado es menor que 0 esta bandera se enciende.

VI) A: La bandera de Acarreo, esta bandera se enciende o se apaga si en la operación de registros se tiene un acarreo.

Como se había mencionado anteriormente se tiene un registro que guarda el valor de las banderas en un número entero que es el registro F, para completar las banderas y para que se puedan guardar como un arreglo de 8 bits es necesario otros dos espacios y estos son el tercero y el quinto, de esta manera el registro F queda de la siguiente manera F[S,Z,_H,_P/V,N,A].

9.- Botón “Siguiente” sirve para cuando se hace uso del desensamblaje y necesitamos pasar a la siguiente instrucción.



Z80

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0000	3E	0F	C6	00	27	47	3E	0F	FE	00	27	80	27	76	00	00
0010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

DESEN SAMBLE
CL PC HEXADECIMAL INSTRUCCION

0001 0001 3E LD A,n

Limpiar Memoria Ir a Direccion STACK

Direccion: 0000 Editar Contenido

REGISTROS INTERNOS

A 0F	A' 00
F 00	F' 00
B 00 00 C	B' 00 00 C'
D 00 00 E	D' 00 00 E'
H 00 00 L	H' 00 00 L'
IX 0000	IY 0000
PC 0002	SP 1000
I 00 00000000	IFF 1
R 00 sz h pnc	T 7

SIGUIENTE ➔

10.- Desensamble cuenta con varias columnas que se explicaran a continuación:

- CL: Es el contador de localidades como su nombre lo dice cuenta las localidades que se usan según el tamaño de las instrucciones.
- PC: El Pc se mencionó anteriormente cuenta también lo que hace es moverse por las localidades de memoria.
- Hexadecimal: Nos muestra el valor de la instrucción que se ejecuta en hexadecimal, recordando que para poder llevar a cabo el simulador se necesita un archivo en hexadecimal.
- Instrucción: Muestra las instrucciones que se están ejecutando y es le sirve al usuario para ver que se está realizando.

Z80

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	3E	0F	C6	00	27	47	3E	0F	FE	00	27	80	27	76	00 00
0010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00 00
0020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00 00
0030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00 00
0040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00 00
0050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00 00
0060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00 00
0070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00 00
0080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00 00
0090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00 00
00A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00 00
00B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00 00
00C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00 00
00D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00 00
00E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00 00
00F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00 00

REGISTROS INTERNOS

A 1E	A' 00
F 90	F' 00
B 0F 00 C	B' 00 00 C'
D 00 00 E	D' 00 00 E'
H 00 00 L	H' 00 00 L'
IX 0000	IY 0000
PC 000E	SP 1000
I 00 10010000	IFF 1
R 00 sz h pnc	T 52

DESEN SAMBLE

CL	PC	HEXADECIMAL	INSTRUCCION
0005	0005	47	LD B,A
0007	0007	3E	LD A,n
0009	0009	FE	CP n
000A	000A	27	CP n
000B	000B	80	ADD B
000C	000C	27	ADD B
000D	000D	76	HALT

STACK

Direccion: 0000

Limpiar Memoria

Ir a Direccion

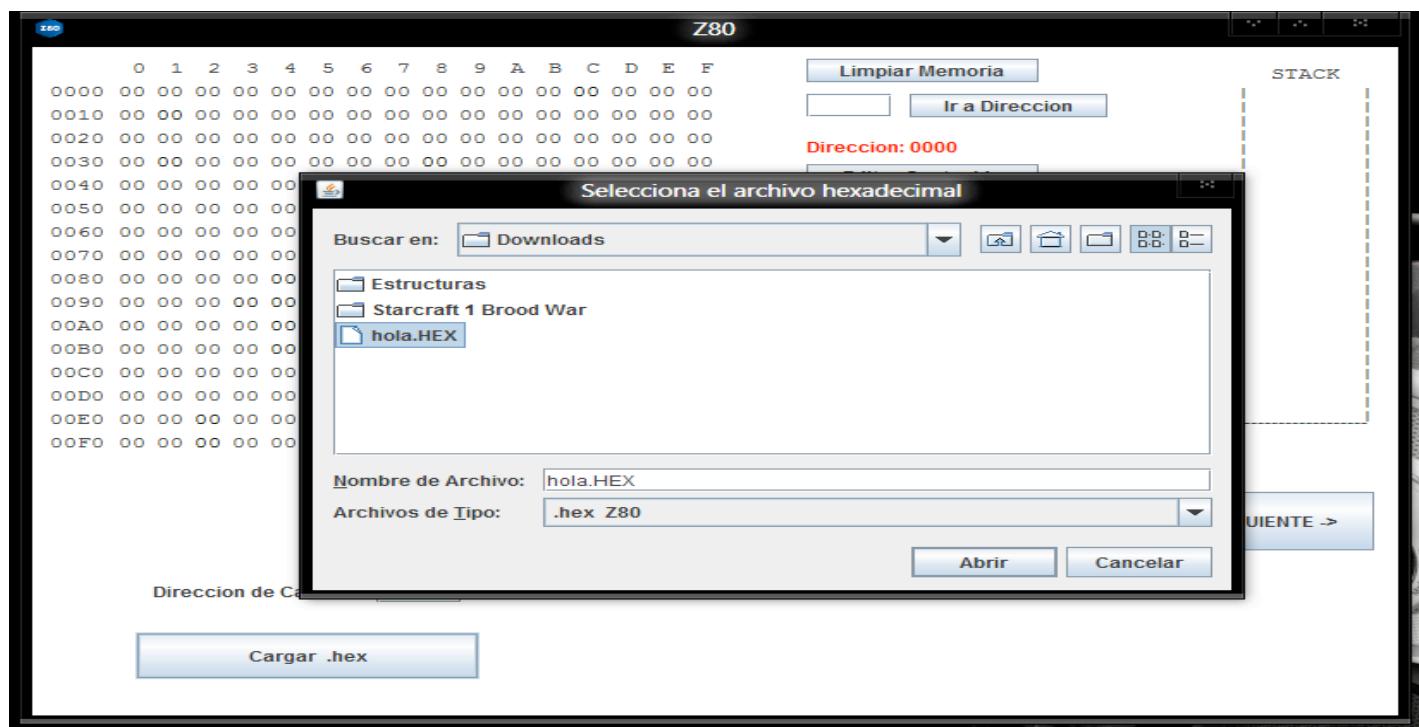
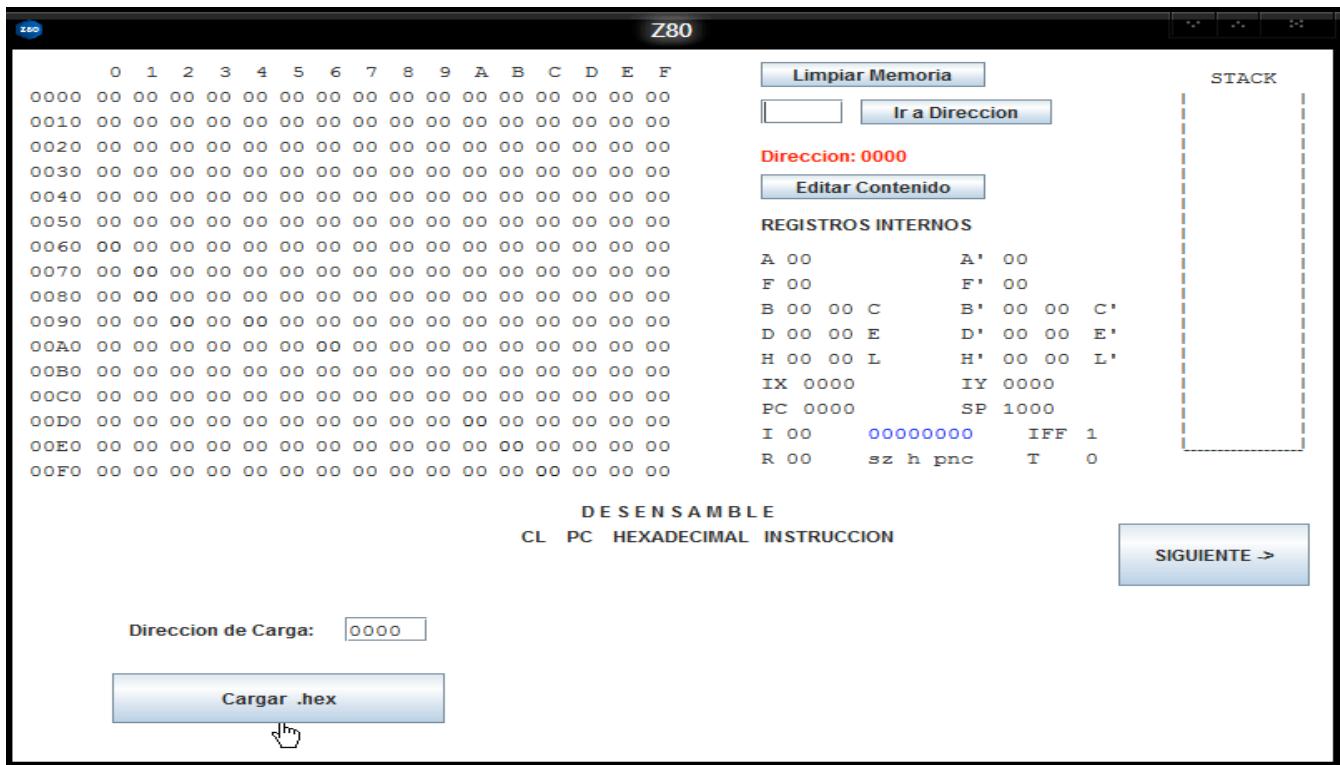
Editar Contenido

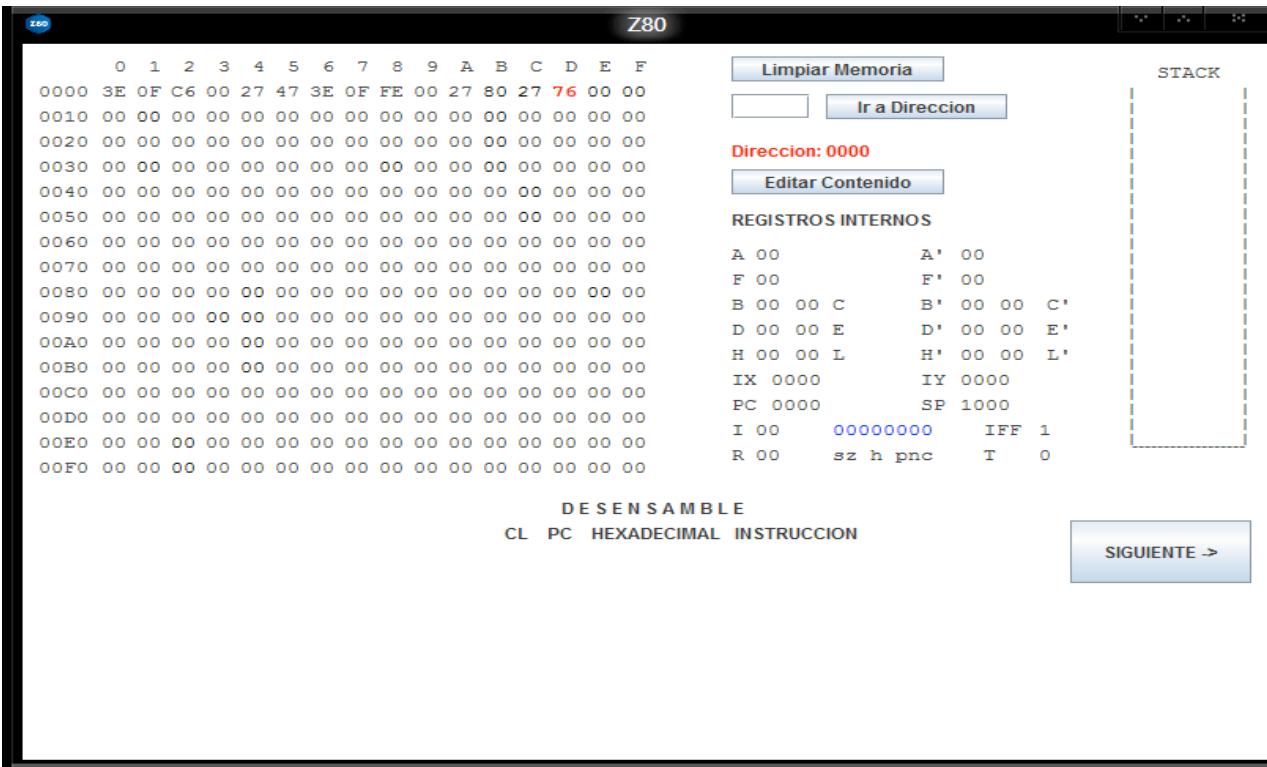
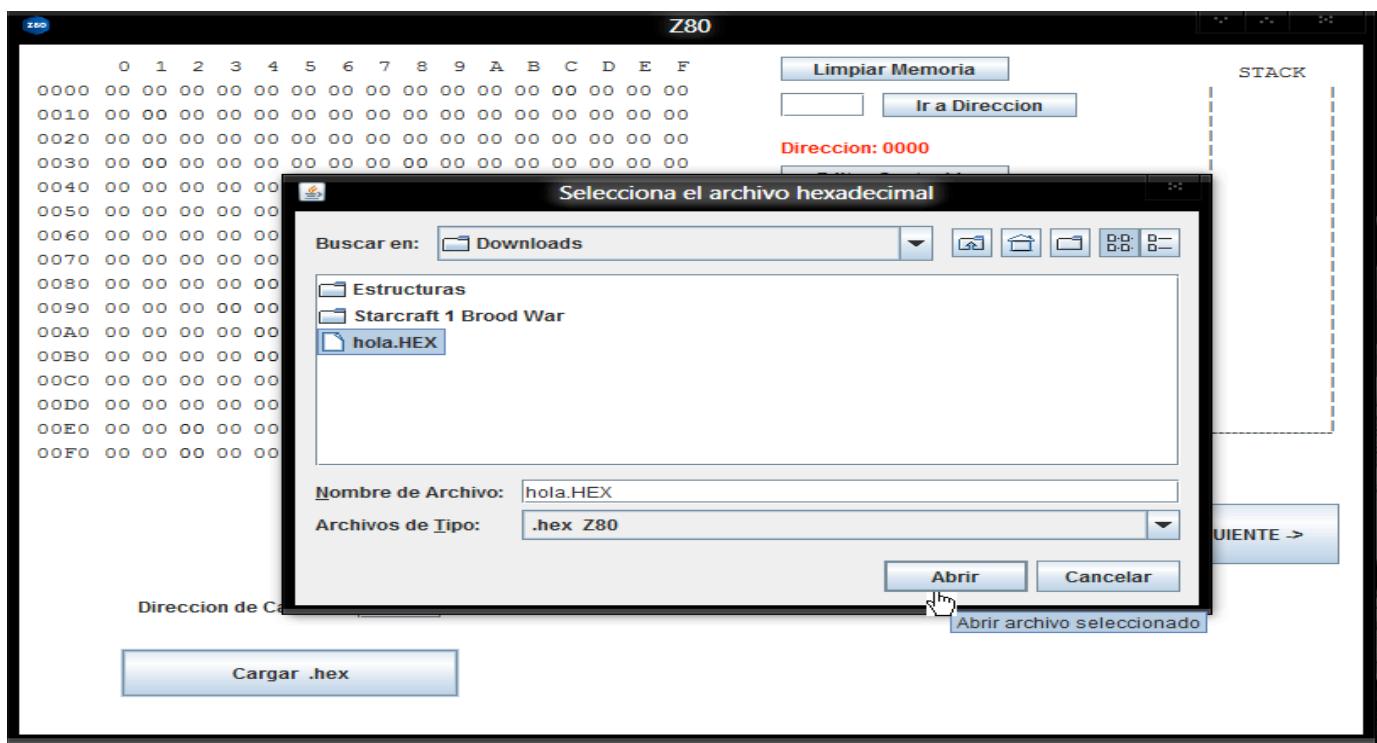
SIGUIENTE ➔

11.- Corrida Libre, en este botón se realiza el programa sin ninguna interrupción ni a lapsos, en realidad solo se ejecuta y se muestra lo que se tiene al final tanto en nuestros registros como en el PC, banderas, tiempo de ejecución y en el desensamblaje.

12.- Paso a paso, con este botón se realiza el programa como su nombre lo dice paso a paso, recordando que en este caso se genera el botón de siguiente que ayuda a darle continuidad al programa o con un enter, este sirve para que el usuario identifique un error y pueda modificarlo.

13.- El 13 y 14 están relacionados puesto que el 13 es un espacio de cuatro dígitos que es la dirección desde donde se empezará a cargar el archivo hexadecimal siendo el botón del inciso 14, en el espacio de carga es necesario saber la dirección donde se inicializará el programa.





NOTAS:

- LAS INSTRUCCIONES RELACIONADAS CON LOS PERIFÉRICOS NO ESTÁN CONTEMPLADAS POR LO QUE SI SE HACE USO DE ESTAS PODRÍA CAUSAR UN ERROR EN EL SIMULADOR.
- EL SIMULADOR VERIFICA QUE EL BYTE DE CHECKSUM PARA VER QUE EL ARCHIVO HEXADECIMAL NO HAYA SIDO MODIFICADO