

MANUAL BÁSICO DE JAVA.



1. [QUE ES JAVA.](#)
2. [PARA QUE SIRVE JAVA.](#)
3. [QUE ES EL JDK.](#)
4. [QUE ES EL JRE.](#)
5. [QUE ES UNA JVM.](#)
6. [DE DONDE OBTENER EL COMPILADOR DE JAVA.](#)
7. [COMO INSTALAR EL COMPILADOR DE JAVA.](#)
8. [COMO SE CONFIGURA EL COMPILADOR DE JAVA.](#)
9. [COMO UTILIZAR EL COMPILADOR DE JAVA.](#)
10. [COMO HACER MI PRIMER PROGRAMA EN JAVA.](#)
11. [DIFERENTES FORMAS DE PROGRAMAR EN JAVA.](#)
12. [COMO HACER UN .JAR](#)

QUE ES JAVA.

Quizá aun no has tenido la necesidad de programar en Java, o tal vez has escuchado hablar de java pero no sepas para que es, o en el peor de los casos nunca has escuchado ni siquiera por error la palabra java, pero no te preocupes a continuación tratare de explicarte paso a paso, desde como instalarlo hasta como hacer tu primer programa en Java.

Pero antes de esto, te daré una breve explicación acerca de Java y su historia:

Java nació en 1991, cuando un grupo de ingenieros de Sun Microsystems trataron de diseñar un nuevo lenguaje de programación destinado a electrodomésticos. La reducida potencia de cálculo y memoria de los electrodomésticos llevó a desarrollar un lenguaje sencillo capaz de generar código de tamaño muy reducido.

Dicho proyecto fue denominado Green , el cual desemboco en el desarrollo de un lenguaje basado en C++ al que su creador, James Gosling, llamo Oak debido a un roble que tenía a la vista desde su ventada en las oficinas de Sun. Posteriormente se descubrió que ya existía un lenguaje de programación con el mismo nombre. Cuando un grupo de gente de Sun visitó una cafetería local, sugirieron de nombre Java (una variedad de café) y así se quedó.

Java es simplemente un lenguaje de programación revolucionario.

Java ha revolucionado el desarrollo de software mediante el código orientado a objetos independiente de la plataforma, con uso intensivo de multimedia, para aplicaciones basadas en Internet, intranets o extranets.

PARA QUE SIRVE JAVA.

Bueno pues Java como ya vimos, es un lenguaje de programación, uno de los más utilizados en el mercado ya que cuanta con muchas características que lo hacen potente, en la actualidad, Java se utiliza para desarrollar aplicaciones empresariales a gran escala, para mejorar la funcionalidad de los servidores de World Wide Web para proporcionar aplicaciones para los dispositivos domésticos como los son teléfonos celulares, radiolocalizadores y asistentes digitales personales y para otros tantos propósitos.

Java se utiliza ahora para crear páginas Web con contenido dinámico e interactivo, para desarrollar aplicaciones empresariales que gran escala, para mejorar la funcionalidad de los servidores Web, para proporcionar aplicaciones a los dispositivos domésticos, etc.

QUE ES EL JDK.

El JDK (Java Development Kit) o Equipo de Desarrollo de Java, en español, contiene las herramientas que permiten a los usuarios crear aplicaciones en Java. Simplemente se trata de un conjunto de programas y librerías que permiten desarrollar, compilar y ejecutar programas en

Java. Está desarrollado por la división JavaSoft de Sun Microsystem. Las versiones más recientes incluyen la arquitectura de componentes JavaBeans y soporte para JDBC.

Existen versiones de JDK para prácticamente todos los sistemas Operativos y existen también distintos programas comerciales.

Sun distribuye gratuitamente el JDK "oficial" para los siguientes sistemas operativos: Windows, Solaris y Linux.

Los JDK incorporan una herramienta de Debugger lo que significa que se puede detener la ejecución de un programa en la línea que se desee para poder conocer el valor de las variables en ese momento.

QUE ES EL JRE.

El JRE (Java Runtime Environment) Entorno en Tiempo de Ejecución de Java, en español, consta de la Máquina Virtual de Java, Java bibliotecas, y todos los demás componentes necesarios para ejecutar aplicaciones Java y applets.

Se puede decir que es prácticamente la versión reducida del JDK que permite simplemente ejecutar código Java, no permite compilar ni utilizar el Debugger.


Viene incluido en el JDK pero también puede instalarse por separado y también es de libre distribución. En este caso simplemente conviene instalarlo por separado cuando no se desea programar en Java si no que simplemente eres una persona común y corriente que navega por la Web.

QUE ES UNA JVM.

Una JVM (Java Virtual Machine) o máquina virtual Java, en español, es simplemente una pieza de software que permite la tecnología Java a ser reconocido y ejecutado con éxito en una amplia gama de plataformas de hardware. Las máquinas virtuales de Java son tan nombrado porque proporcionan un entorno necesario para el byte code de Java a ser ejecutado. La flexibilidad de una JVM permite que un applet Java para ser escrito sólo una vez, pero capaz de ser ejecutado en prácticamente cualquier sistema operativo.

Las máquinas virtuales Java aceptar normalizados formato de código binario. Compiladores Java traducir este código en el formato necesario antes de que se ejecuta en una JVM. Este enfoque asegura organizado de la portabilidad y la estabilidad de Java que se apliquen de una amplia gama de dispositivos de hardware.

DE DONDE OBTENER EL COMPILADOR.

En primer lugar lo que tienes que hacer es simplemente dar clic en el siguiente vinculo el cual te llevara directamente a la zona de descargas en la página de , en donde podrás obtener de forma gratuita el compilador de Java junto con la maquina virtual.

Una vez que ya abriste el vínculo hacia la página principal de descargas de Sun Microsystems, se abrirá la siguiente ventana:



Sun Java Solaris Communities My SDN Account Join SDN

Sun Sun Developer Network (SDN) search tips Search

APIs Downloads Products Support Training Participate

SDN Home > Java Technology > Java SE >

Java SE Downloads

It's time
Download the complete environment and runtime environment
» [Get the JDK download](#)

Overview Technologies Reference Community Support **Downloads**

Latest Release | Next Release (Early Access) | Embedded Use | Real-Time | Previous Releases

Confused or having trouble downloading or installing? See the [download help page](#).
» Supported System Configurations

Java SE Runtime Environment (JRE) 6 Update 10 » Download
This release provides a new Java Plug-in that combines features of applet and Java Web Start technologies, a new Direct3D pipeline for Microsoft platforms, an updated documentation bundle, and more.
[Installation Instructions](#) | [ReadMe](#) | [ReleaseNotes](#) | [Sun License](#) | [Third Party Licenses](#)

Java SE Development Kit (JDK) 6 Update 10 » Download
The Java SE Development Kit (JDK) includes the Java SE Runtime Environment (JRE) and command-line development tools that are useful for developing applets and applications.
» [More info about Java SE 6 Update 10 ...](#)
[FAQ](#) | [Installation Instructions](#) | [ReadMe](#) | [ReleaseNotes](#) | [Sun License](#) | [Third Party Licenses](#)

JDK 6 Update 10 with Java EE » Download
This distribution of the Java SE Development Kit (JDK) is included in the Java EE 5 SDK, which contains the GlassFish application server and provides web services, component-model, management, and communications APIs that make it the industry standard for implementing enterprise-class service-oriented architecture (SOA) and Web 2.0 applications.

» [Java SE Site Map](#)

Regional Downloads
[Japanese](#)
[日本語版](#)

Related Resources
» [Java SE for Business](#)
» [Compatibility](#)
» [Performance](#)
» [Security](#)
» [Mobility](#)
» [Timezone Updates](#)

Related Downloads
» [XML and Web Services](#)
» [Java Media Framework](#)

Popular Topics
» [JDK 6 Adoption Guide](#)
» [Java Platform Migration Guide](#)
» [Garbage Collection Tuning](#)
» [Troubleshooting Java SE](#)

Sun Resources
» [BigAdmin \(sysadmin resources\)](#)
» [Sun Web Learning Center](#)

Ahora que si quieres entrar a la página de Sun Microsystems tendrás que dar clic en la parte de descargas:

Sun Java Solaris Communities My SDN Account Join SDN

Sun Developer Network (SDN) search tips Search

APIs Downloads Products Support Training Participate

SDN Home > Java Technology > Java SE >

Java SE Downloads

It's time
Download the complete environment and runtime environment
» Get the JDK download

Overview Technologies Reference Community Support **Downloads**

» Java SE Site Map

Regional Downloads
Japanese
日本語版

Related Resources
» Java SE for Business
» Compatibility
» Performance
» Security
» Mobility
» Timezone Updates

Related Downloads
» XML and Web Services
» Java Media Framework

Popular Topics
» JDK 6 Adoption Guide
» Java Platform Migration Guide
» Garbage Collection Tuning
» Troubleshooting Java SE

Sun Resources
» BigAdmin (sysadmin resources)
» Sun Web Learning Center

Latest Release | Next Release (Early Access) | Embedded Use | Real-Time | Previous Releases

Confused or having trouble downloading or installing? See the download help page.
» Supported System Configurations

Java SE Runtime Environment (JRE) 6 Update 10 » Download
This release provides a new Java Plug-in that combines features of applet and Java Web Start technologies, a new Direct3D pipeline for Microsoft platforms, an updated documentation bundle, and more.
Installation Instructions | ReadMe | ReleaseNotes | Sun License | Third Party Licenses

Java SE Development Kit (JDK) 6 Update 10 » Download
The Java SE Development Kit (JDK) includes the Java SE Runtime Environment (JRE) and command-line development tools that are useful for developing applets and applications.
» More info about Java SE 6 Update 10 ...
FAQ | Installation Instructions | ReadMe | ReleaseNotes | Sun License | Third Party Licenses

JDK 6 Update 10 with Java EE » Download
This distribution of the Java SE Development Kit (JDK) is included in the Java EE 5 SDK, which contains the GlassFish application server and provides web services, component-model, management, and communications APIs that make it the industry standard for implementing enterprise-class service-oriented architecture (SOA) and Web 2.0 applications.

Una ves aquí dentro tendrás que elegir el archivo “Java SE Development Kit (JDK) 6 Update 10” ya que aquí se encuentra el compilador JDK y la maquina virtual que se encuentran en el archivo JRE que como ya explique previamente aquí es donde se encuentra la maquina virtual de java o JVM que es la que hace correr aplicaciones de java en nuestra PC.

Sun Java Solaris Communities My SDN Account Join SDN

Sun Developer Network (SDN) search tips Search

SDN Home > Java Technology > Java SE >

Java SE Downloads

It's time
Download the complete environment and runtime environment
» [Get the JDK download](#)

Overview Technologies Reference Community Support **Downloads**

Latest Release | Next Release (Early Access) | Embedded Use | Real-Time | Previous Releases

Confused or having trouble downloading or installing? See the download help page.
» [Supported System Configurations](#)

Java SE Runtime Environment (JRE) 6 Update 10 [Download](#)
This release provides a new Java Plug-in that combines features of applet and Java Web Start technologies, a new Direct3D pipeline for Microsoft platforms, an updated documentation bundle, and more.
[Installation Instructions](#) | [ReadMe](#) | [ReleaseNotes](#) | [Sun License](#) | [Third Party Licenses](#)

Java SE Development Kit (JDK) 6 Update 10 [Download](#)
The Java SE Development Kit (JDK) includes the Java SE Runtime Environment (JRE) and command-line development tools that are useful for developing applets and applications.
» [More info about Java SE 6 Update 10 ...](#)
[FAQ](#) | [Installation Instructions](#) | [ReadMe](#) | [ReleaseNotes](#) | [Sun License](#) | [Third Party Licenses](#)

JDK 6 Update 10 with Java EE [Download](#)
This distribution of the Java SE Development Kit (JDK) is included in the Java EE 5 SDK, which contains the GlassFish application server and provides web services, component-model, management, and communications APIs that make it the industry standard for implementing enterprise-class service-oriented architecture (SOA) and Web 2.0 applications.

» [Java SE Site Map](#)

Regional Downloads
[Japanese](#)
[日本語版](#)

Related Resources
» [Java SE for Business](#)
» [Compatibility](#)
» [Performance](#)
» [Security](#)
» [Mobility](#)
» [Timezone Updates](#)

Related Downloads
» [XML and Web Services](#)
» [Java Media Framework](#)

Popular Topics
» [JDK 6 Adoption Guide](#)
» [Java Platform Migration Guide](#)
» [Garbage Collection Tuning](#)
» [Troubleshooting Java SE](#)

Sun Resources
» [BigAdmin \(sysadmin resources\)](#)
» [Sun Web Learning Center](#)

Al hacer clic en el vínculo hacia la parte de descarga del archivo "Download", te mandaran hacia la siguiente página en donde tendrás que seleccionar la plataforma en donde deseas instalar java, suponemos que estas utilizando Windows ya que es la plataforma más común entre los usuarios de PC's y es por eso que toda la demás información va orientada a la plataforma de Windows.



Así que por lo tanto tendrás que dar clic en la selección de plataformas y elegir Windows como sigue:



Una vez elegida la plataforma de Windows, tendrás que seleccionar o aceptar el contrato de licencia de Sun Microsystems, ya que sin esto no podrás descargar el archivo que contiene el compilador y la maquina virtual.



Java Solaris Communities My SDN Account Join SDN

Sun Downloads
microsystems

SDN Home > Download Center >

Java SE Development Kit 6u10 First Customer Ship

Provide Information, then Continue to Download

There are no 64-bit versions of the Java Plugin, Java Web Start or Java Control Panel; however the 32-bit versions of the JRE can be installed on 64-bit systems in order to obtain this functionality. Note that only 32-bit browsers are supported at this time.

Select Platform and Language for your download:

Platform:

Language:

I agree to the Java SE Development Kit 6 License Agreement

[Continue »](#)

NetBeans Simple, Intuitive IDE

Getting Started?

- » New to Java Center
- » New to Solaris Center
- » Sun Studio


Download Resources

- » FAQs
- » Download History
- » Sun Download Manager
- » Download Center Customer Service

Ya que haz aceptado el contrato de licencia tendrás que dar clic en el botón para continuar hacia la descarga “Continue”, para poder pasar hacia el siguiente vínculo en donde ahora si se ejecutara la descarga del archivo “Java SE Development Kit (JDK) 6 Update 10”.


Cabe aclarar que los números “6” y “10” que se encuentran en el nombre del archivo anteriormente mencionado, son las versiones del compilador “JDK” y JVM “JRE” respectivamente, y estos irán cambiando respectivamente según haya actualización del software.

Java ▾ Solaris ▾ Communities ▾ My SDN Account ▾ Join SDN ▾


 Downloads

SDN Home > Download Center >

Download Java SE Development Kit 6u10 First Customer Ship for Windows, Multi-language



Download Information and Files


 **Get the latest Java Runtime Environment to use Sun Download Manager**

Internet Explorer Users: Check the top of this page for a "Java(TM) Web Start ActiveX Control" message in the information bar. If it appears, click it to finish detecting your Java version.

We were unable to detect a recent version of Java Runtime Environment (JRE) on your system. With the latest JRE, you can automatically download, install, and run Sun Download Manager (SDM) directly from this page. We highly recommend SDM to easily manage your downloads (pause, resume, restart, verify, and more). Visit java.com for the latest JRE.

Required Files

File Description and Name	Size
Windows Offline Installation jdk-6u10-windows-i586-p.exe	72.88 MB



Notes:

- For download problems or questions, please see the [Download Center FAQ](#).
- If you logged in first, you can complete this download any time in the next 30 days. Just visit your [Download History](#).
- For Customer Service, contact [Download Center Customer Service](#).

NetBeans Simple, Intuitive IDE

Getting Started?

- » [New to Java Center](#)
- » [New to Solaris Center](#)
- » [Sun Studio](#)

Download Resources

- » [FAQs](#)
- » [Download History](#)
- » [Sun Download Manager](#)
- » [Download Center Customer Service](#)

Related Resources

- » [Java.sun.com](#)
- » [Solaris Developer Center](#)
- » [JavaFX](#)
- » [Web Developer Resource Center](#)
- » [Developer Services](#)
- » [JavaOne Online](#)
- » [Sun Student Developer Program](#)

Ahora si, una vez entendido y echo todos los pasos anteriores tendrás que dar clic sobre el nombre del archivo "Java SE Development Kit (JDK) 6 Update 10", para poder descargarlo.

Y pues como todo, llevara su tiempo la descarga según tu tipo de conexión a internet.

Después de la descarga tendrás que instalar el software y para eso tendrás que leer la siguiente información.

COMO INSTALAR EL COMPILADOR DE JAVA.

Ahora si, una vez que ya hiciste todo el proceso de descarga, tendrás que instalarlo no, pues para eso bajaste el compilador.

Lo primero que tienes que hacer es dar clic en el archivo previamente descargado desde la página de Sun Microsystems.



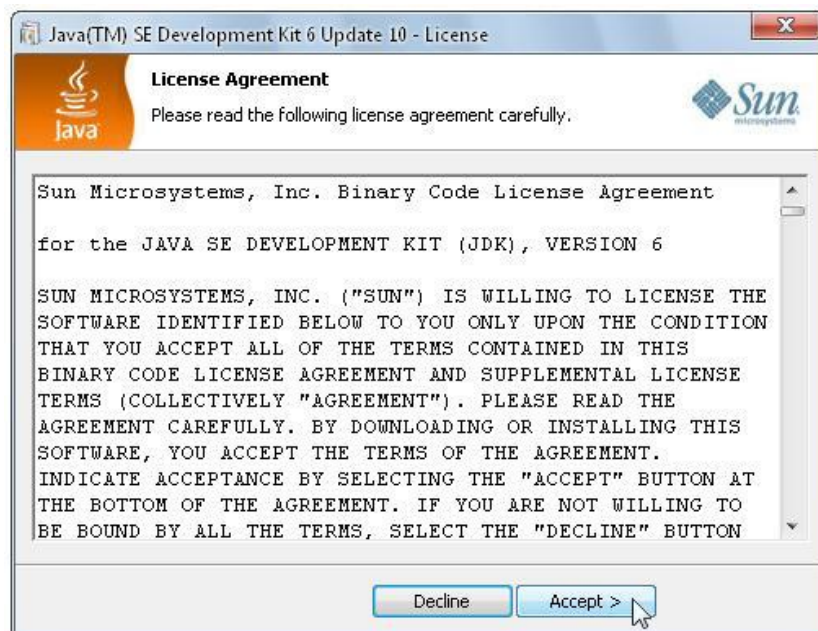
jdk-6u10-windows-i586-p

Una vez que ya abriste el archivo del compilador tendrás que seguir las siguientes instrucciones para poder instalarlo con éxito.

Se abrirá la siguiente ventana en donde dan la bienvenida a la instalación del compilador "JDK" y "JRE". Y tendrás que esperar algunos segundos mientras se realiza la carga de los archivos a instalar.

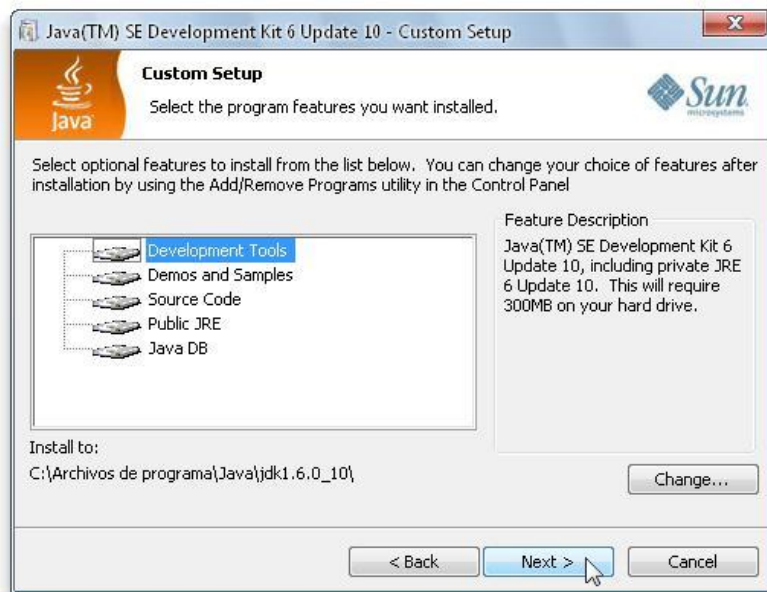


Una vez que ya se cargaron todos los archivos necesarios para la instalación se abrirá la siguiente ventana.



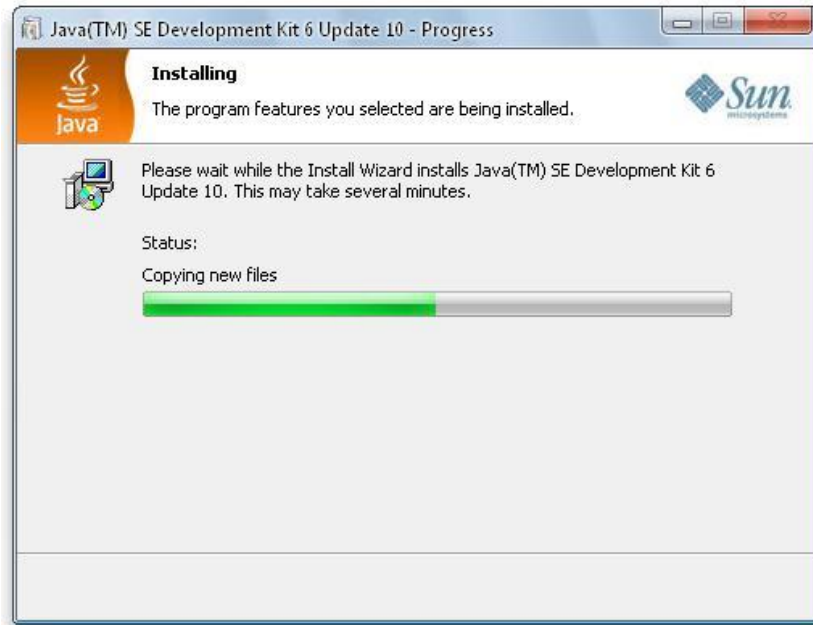
En esta ventana, tendrás que dar clic en el botón de aceptar “Accept” ya que con esto estarás aceptando todos los términos de licencia del software.

Después de aceptar el contrato de licencia, el programa de instalación te mandara hacia la siguiente ventana.



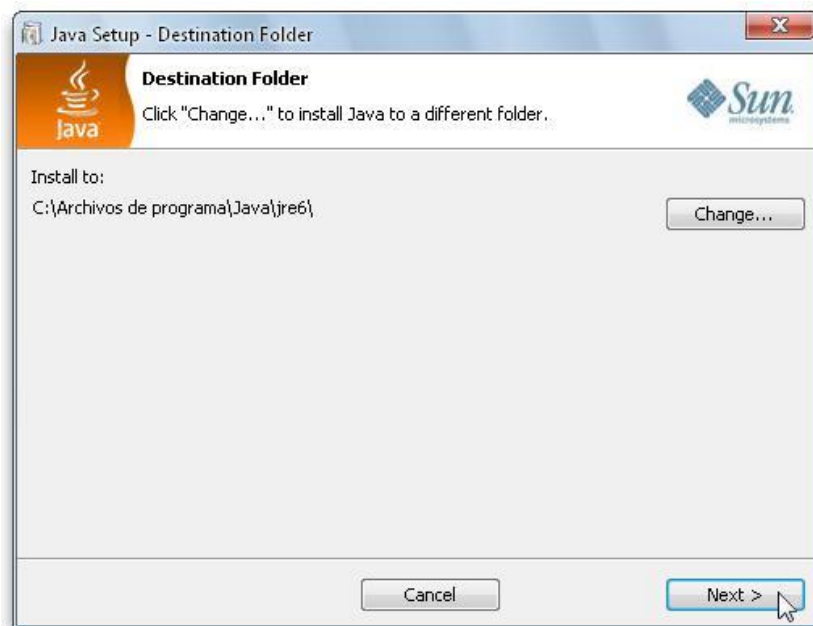
Aquí es donde se indican todos los elementos a instalar y la ruta que da por default el programa de instalación en donde se copiara el compilador, y como vemos, el nombre de la carpeta que esta al final de la ruta de instalación o sea “jdk1.6.0_10”, indica que se va a copiar la el JDK con el numero de la versión. Y es aquí en donde se encuentra el compilador de java.

En caso de no querer instalar el compilador en la ruta que se da pro default pues tendrás que dar clic en el botón “Change...” para poder cambiar la ruta de instalación a donde se te pegue la gana. Una vez elegida la ruta en donde se desea hacer la instalación, ya sea que la hayas cambiado o dejado como esta, pues tendrás que dar clic en el botón de siguiente “Next>” par a continuar con la instalación. Una vez que ya diste clic en el botón “Next>” se abrirá la siguiente ventana.



En esta ventana lo único que se indica es el progreso que hay con la instalación y el nombre de los archivos instalados, esto tal vez podrá llevarse algunos minutos.

Más o menos a la mitad de la instalación de java, se aparecerá otra ventana.



En esta ventana, indica otra nueva instalación, pero esta es ahora de la carpeta "jre6" esto indica que se instalará el "JRE" y el número de la versión, es aquí en donde se instalará la máquina virtual o JVM.

Y nuevamente dará la ruta de instalación que pone por default el programa de instalación y como anterior mente lo mencione, si no te gusta esa ruta, tendrás que dar clic en el botón “Change...” para poder cambiar la ruta de instalación a donde mejor te parezca. Una ves hecho esto tendrás que dar clic en el botón de siguiente “Next>” para poder continuar con la instalación.

Ahora si ya que has dado clic en el botón “Next>” se abrirá la siguiente ventana:



En esta ventana se indica el progreso de la instalación de la JRE.

Después de algunos minutos, mientras se instala el software de Java se abrirá la siguiente ventana.



Aquí solo te indican el fin del proceso de instalación, y también te invitaran a que te registres en a página de Sun Microsystems, esto no es necesario, prácticamente solo se pide para saber cuantas personas ocupas su software y mantenerte al tanto de las nuevas actualizaciones del producto, es posible que automáticamente se abra una ventana de internet en donde te pedirán registrarte pero solo bastara con cerrarla, claro esto lo dejo a tu consideración.

Posteriormente tendrás que dar clic en el botón de finalizar “*Finish*”, para terminar con la instalación.

NOTA: Algo muy importante es que si tienes instalado en tu PC algún Firewall (software utilizado en redes de computadoras para controlar las comunicaciones, permitiéndolas o prohibiéndolas) pues puede ser que mientras estas instalando el software de Java, se aparezcan ventanas como estas:



Aquí lo único que tendrás que hacer es seleccionar “Permitir”, dar clic en “utilizar siempre esta acción”, esto es para que no se aparezcan estas ventanas repetidas veces y por ultimo dar clic en el botón “*Aceptar*”. Dichas ventanas son para que algunos archivos de las carpetas JDK y JRE puedan tener acceso a internet, esto se hace con la finalidad e mantenerte al tanto sobre las actualizaciones de software y es posible que solo se abran dos ventanas así, una para el JDK y otra para JRE. En caso de que no tengas instalado algún Firewall pues no te preocupes.

COMO SE CONFIGURA EL COMPILADOR DE JAVA.

Bueno pues en las versiones recientes del software de Java no es necesario configurar el producto. Esto se hacia en versiones un tanto antiguas como por ejemplo la “jdk-1_5_0_02-windows-i586-p” la cual se tenía que configurar de la siguiente forma:

Para Windows XP o 2000

1. Una vez instalado el archivo. Dar clic derecho sobre MiPC->Propiedades->Opciones avanzadas->Variables de entorno.
2. En las variables del sistema, localizar la variable PATH y seleccionar editar.
3. Agregar al PATH la ubicación donde se ha instalado el JSDK de la siguiente manera: La variable PATH ya contiene cierta información, para agregar un directorio nuevo hay que separarlo por punto y coma (;). escribir ; (punto y coma) al final del contenido actual del path, y agregar el directorio donde se encuentra el compilador de java, ejemplo:

...ared\DLLShared\;C:\Archivos de programa\Java\jdk1.5.0_02\bin

que es el directorio donde se instalaba por lo regular el jskd.

4. Dar clic en aceptar.
5. Necesitábamos agregar una nueva variable llamada CLASSPATH en las variables de entorno: Dar clic en nueva, como nombre tendrá CLASSPATH y como valor llevará el lugar donde se instalaron las bibliotecas de Java, que regularmente están en:

C:\Archivos de programa\Java\jdk1.5.0_02\jre\lib

Entonces el valor quedará como sigue:

C:\Archivos de programa\Java\jdk1.5.0_02\jre\lib;

Notar el . (punto) al final de la línea que servirá para hacer referencia al directorio actual. Con esto debía estar configurado nuestro jskd, y para probarlo ir a Inicio-> ejecutar-> escribir cmd y en la ventana de comandos escribir:

C:\> java – versión

Para invocar a la máquina virtual (JVM) o

C:\> javac

Para invocar al compilador.

Windows98 y Milenio.

1. Editar el archivo autoexce.bat ubicado en el directorio raíz, es decir en C:\, para ello bastaba con abrir el archivo con el notepad o bloc de notas.
2. Agregar la siguiente línea:

PATH=%PATH%;C:\Archivos de programa\Java\jdk1.5.0_02\bin

3. De igual manera se agregaria el CLASSPATH en otra linea:

`CLASSPATH=C:\Archivos de programa\Java\jdk1.5.0_02\jre\lib;`

4. Después de realizar esto se tendrá que reiniciar la máquina para que los cambios surgieran efecto.
- Cabe aclarar que todo esto se hacia en versiones anteriores del software de Java, para esta versión ya no se tiene que hacer esta configuración.

COMO UTILIZAR EL COMPILADOR DE JAVA.

Bueno ahora si, una vez descargado e instalado el compilador de java junto con la maquina virtual, podremos comenzar a programar en este lenguaje.

Antes que nada daré una breve explicación sobre el uso general de Java.

Java es un lenguaje de programación orientado a objetos para referirme a esto más adelante solo utilizare las siglas (POO). Y entenderemos por objeto que: son, esencialmente, componentes de software reutilizables que modelan piezas de programas de software en términos de propiedades (como el nombre, color y tamaño) y comportamientos (como calcular, desplazarse y comunicarse).

Algunos de los elementos que se utilizan en la POO se muestran en la siguiente tabla.

ELEMENTOS	DEFINICIÓN
Objetos	Se trata de una encapsulación tanto de datos (variables) como de métodos (funciones o procedimientos para la manipulación de los datos). Posee métodos que determinan su forma de operar y variables manejadas por dichos métodos las cuales determinan su estado actual.
Mensajes	Se encargan del intercambio de información entre los objetos por lo que surgen cambios en el estado de los mismos. Al conjunto de mensajes por las que un objeto puede ser afectado se le denomina protocolo. El envío de una señal es el llamado a una función miembro o método.
Métodos	Son funciones o procedimientos que se definen dentro de las clases y establecen el comportamiento del objeto al recibir un mensaje. Otra de sus funciones es la de enviar una señal a un segundo objeto con el fin de solicitar alguna acción o información. Los datos o variables relacionadas con el método almacenan información para el mismo.
Clases	Es un molde para crear objetos del tipo de dicha clase. Por ejemplo un molde para gelatina con forma de estrella siempre creará gelatinas con forma de estrella aunque los valores de sus atributos como color, sabor, textura puedan ser modificados. La clase define tanto los

	<p>métodos como los atributos que establecen las características a compartir por todo el conjunto de objetos generados por la misma y los encapsula.</p> <p>Por lo anterior se describe al objeto como instancia de clase o a una clase como n tipo definido por el usuario.</p>
Subclases	Es una clase descendiente de otra desde un punto de vista jerárquico y adopta o hereda todos los datos y métodos de la clase madre.
Abstracción	<p>Implica una operación intelectual en la que se aísla un determinado elemento excluyendo a otros que puedan estar relacionados con él. A través de esta se describe en forma general los aspectos necesarios para una panorámica general del problema. En la POO se debe lograr la abstracción de métodos y datos que han de ser almacenados en una clase par la construcción de nuevos objetos que como ya se ha mencionado puedan ser reutilizables reduciendo posteriormente el trabajo de desarrollo.</p> <p>Un ejemplo de abstracción puede ser el pensar en la función general de un radio considerando el selector de la estación, el control de volumen, el formato que maneja (disco compacto, casete, cinta magnética) sin tener que pensar en os detalles electrónicos de dichos dispositivos.</p> <p>Cuando se abstraen objetos que definen un estructura de datos determinada junto con sus métodos se denomina tipo abstracto de datos.</p>
Encapsulamiento	<p>Consiste en la protección tanto de los datos como de los métodos de un objeto de manera que nadie pueda descubrir su estructura interna. Al encapsular datos y métodos de un objeto se deben proporcionar formas de poder interactuar con ellos sin necesidad de abrirlos.</p> <p>Por ejemplo, pensemos en un automóvil de tipo estándar; nosotros al manejar tenemos acceso a la transmisión del auto a través de la palanca de velocidad, la cual permite incrementar o decrementar la potencia del vehículo sin que tengamos que conocer o acceder al sistema de engranes que esta dentro de la caja de velocidad, de este modo la palanca de velocidad es el método que el fabricante ofrece para poder interactuar con la transmisión</p>
Herencia	<p>Se trata de un mecanismo por el cual se transfieren los datos y métodos de una o más clases a una subclase. Por ejemplo, considera la clase Mamíferos (poseen sistema respiratorio, sistema nervioso, sistema reproductor) y supongamos que se genera una subclase denominada Humano (razona, es bípedo). Al ser humano una clase de mamíferos hereda todos los elementos de esta última por lo que la subclase humano posee sistema respiratorio, sistema nervioso, sistema reproductor, razona y es bípedo.</p> <p>La herencia facilita la reutilización de código ya que si necesito usar métodos de una clase ya existente en la clase que estoy construyendo</p>

	solo necesitaría heredar los elementos de la primera clase a la mía con lo cual evito reescribir todo el código de dichos métodos de nuevo, disminuyendo así el tiempo de desarrollo y el costo de la aplicación.
Polimorfismo	Establece diversas formas de comportamiento para un método y cada forma depende de la clase en la que se implemente. Con esto pueden existir una multitud de métodos distintos con un solo nombre que usan el mismo acceso.

Después de estos conceptos bastante importantes y que espero entiendas bien ya que te serán de mucha utilidad cuando quieras desarrollar aplicaciones en Java, daré un brevario cultural de cómo utilizar algunos comandos del sistema de la PC por si nunca los has utilizado o por si ya se te olvidaron.

En primer lugar lo que tienes que hacer es abrir el símbolo del sistema desplazándote por la siguiente ruta:

Inicio->Todos los programas->Accesorios->Símbolo del sistema.

Una vez abierto la ventana de Símbolo del sistema podrás practicar insertando los siguientes comandos, de hecho entre los más importantes para poder utilizar de forma adecuada el compilador de Java.

COMANDOS	FUNCION
cd..	Regresa al directorio principal carpeta por carpeta.
cd\	Regresa a la raíz del directorio directamente.
cls	Limpia la ventana de símbolo del sistema.
cd "ruta..."	Va hacia la dirección indicada carpeta por carpeta.
cd "ruta...\ruta..."	Va directamente hacia la dirección indicada.
exit	Sale del símbolo del sistema.
help	Muestra todos los comandos del símbolo del sistema.
"unidad":	Una vez en el directorio raíz (Unidad de almacenamiento), con este comando podrás moverte hacia otra unidad.
javac	Únicamente funciona, yendo a la ruta en donde fue instalado el compilador y muestra algunos componentes instalados.
java	Sirve para correr programas en java previamente compilados.

COMO HACER MI PRIMER PROGRAMA EN JAVA.

Bueno, pues ahora si, los programas en Java pasan a través de 5 fases para poder ejecutarlos y estas fases son: edición, compilación, carga, verificación y ejecución. A continuación iré explicando cada fase paso a paso.

1. **Edición:** Antes que nada, tendrás que aprender que los programas en Java se escriben en cualquier editor de texto, esto puede ser en un Bloc de Notas y para ejemplificarlo primero abriremos uno, en donde escribiremos el siguiente código.

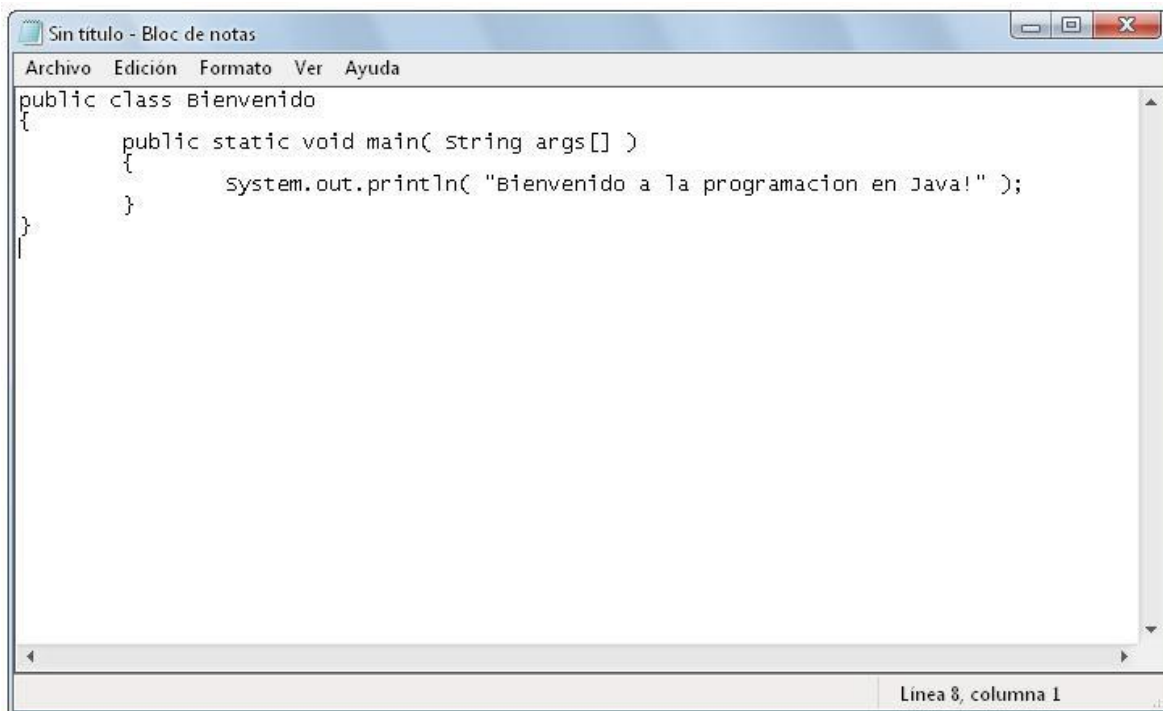
```
public class Bienvenido
{
    public static void main( String args[] )
    {
        System.out.println( "Bienvenido a la programacion en Java!" );
    }
}
```

Algo muy importante que debes tener en cuenta es que los nombres de los archivos con código creados en un editor de texto se guardan con la extensión “.java” así que no olvides guardar tu archivo con esta extensión y en la carpeta bin en donde instalaste el compilador, si no cambiaste la ruta, debe ser esta:

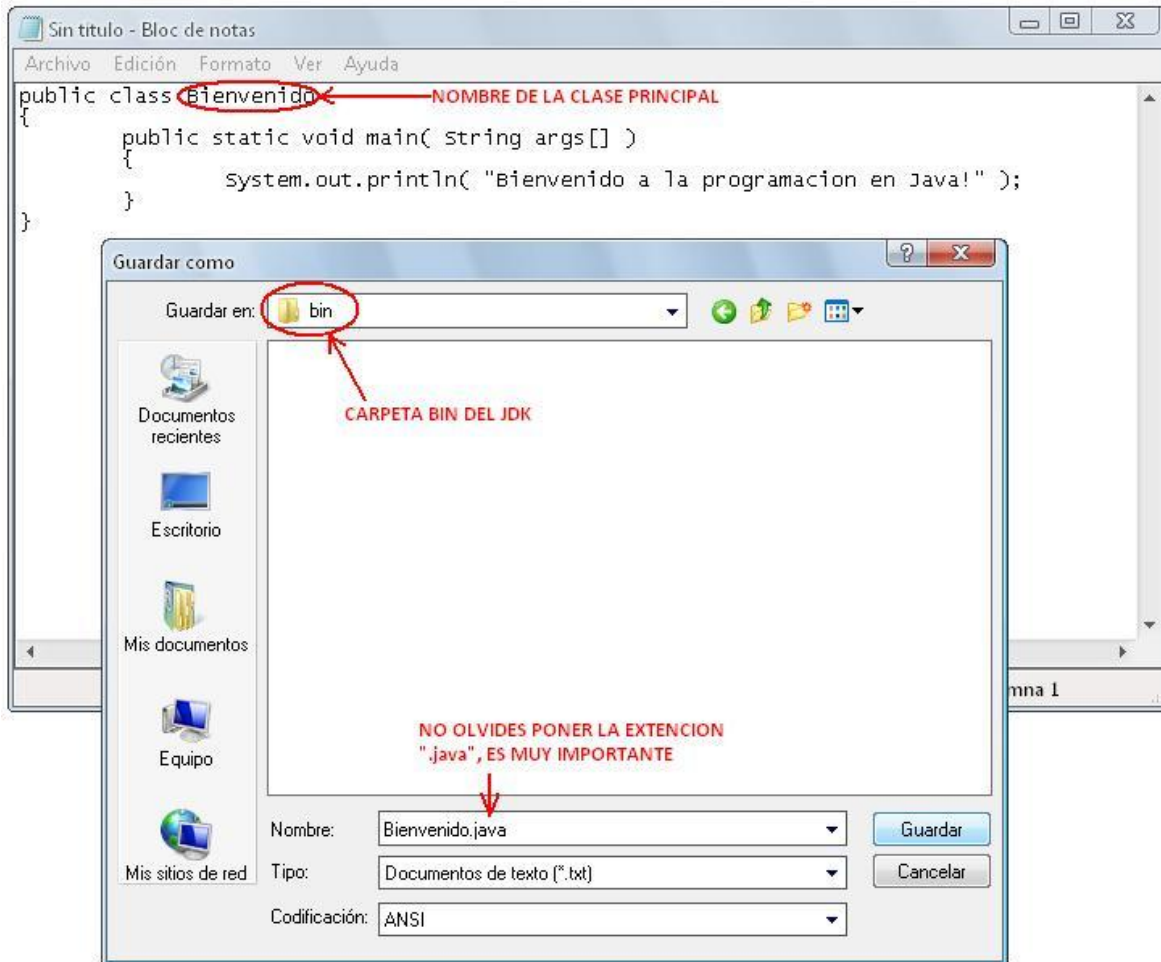
C:\Archivos de programa\Java\jdk1.6.0_10\bin

Aquí muestro las ventanas que deberán aparecerte.

Primero abres el bloc de notas y escribes el código:

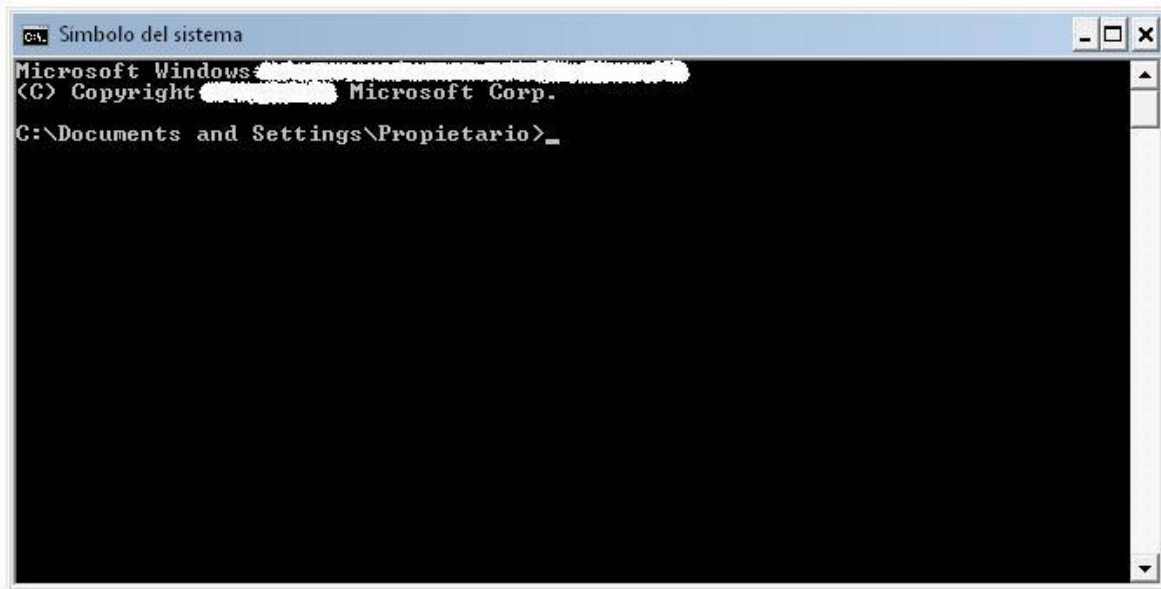


Posteriormente guardas el archivo con el mismo nombre de la clase principal y sin olvidar poner la extensión “.java” que es muy importante.



Luego de haber dado clic en el botón de “Guardar” en la dirección en donde se instaló el JKD que por lo regular es esta; “C:\Archivos de programa\Java\jdk1.6.0_10\bin” tendrás que abrir el símbolo del sistema en:

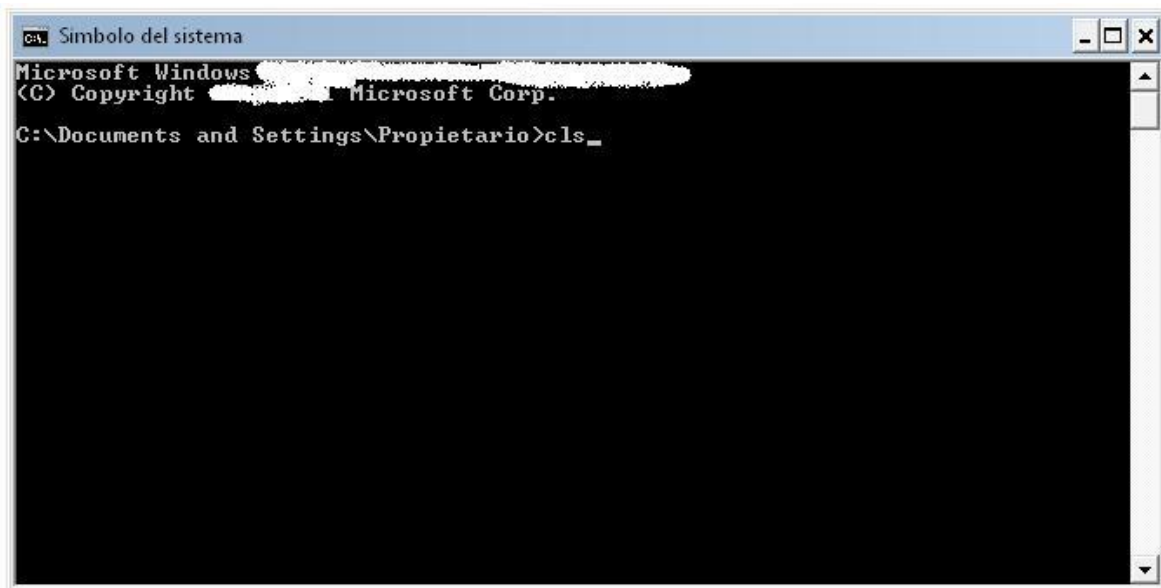
Inicio->Todos los programas->Accesorios->Símbolo del sistema



```
CA: Simbolo del sistema
Microsoft Windows [Versión 5.0.2600.5512]
(C) Copyright 2006 Microsoft Corp.
C:\Documents and Settings\Propietario>_
```

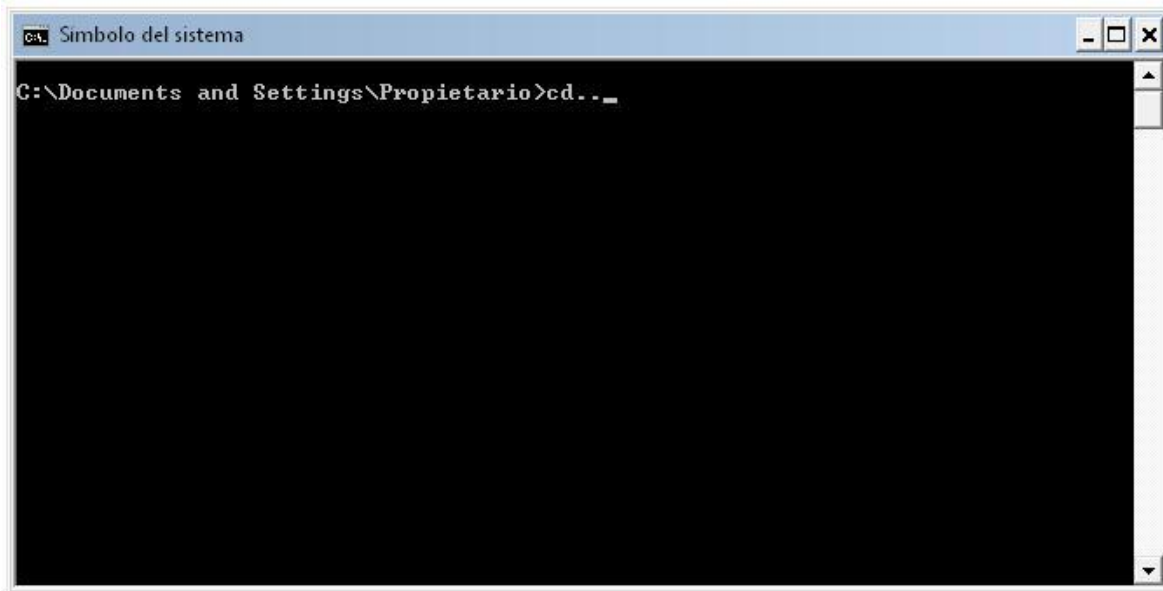
Aquí tendrás que empezar a utilizar los comandos antes explicados del símbolo del sistema para poder desplazarte hacia la carpeta bin del JDK.

En primer lugar ejecutar “cls” para limpiar la pantalla esto lo hago únicamente para tener más espacio para escribir.



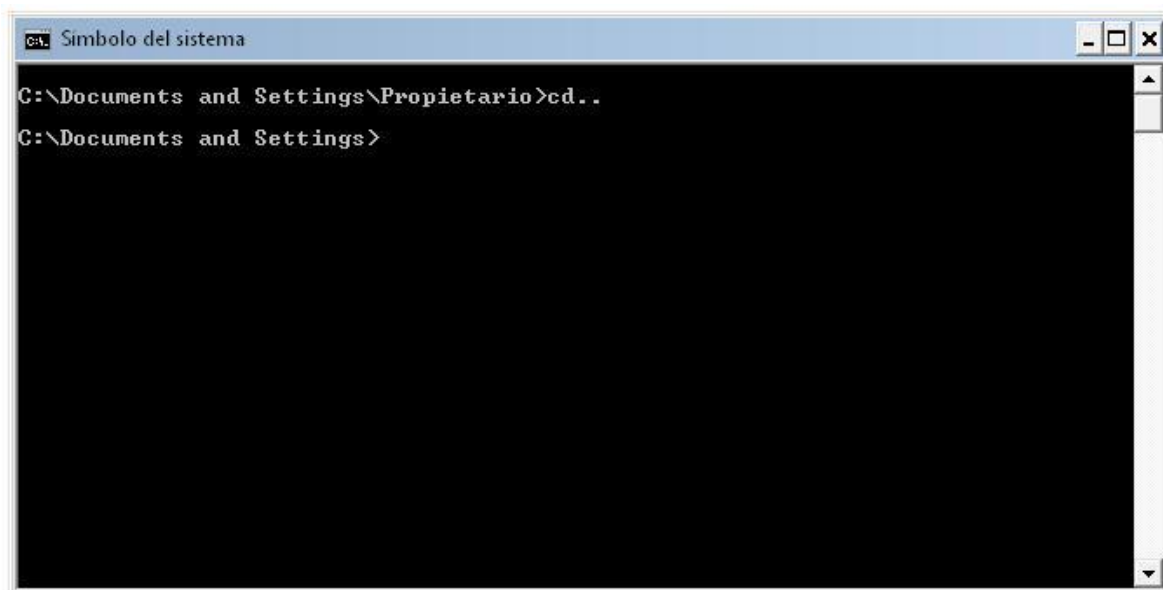
```
CA: Simbolo del sistema
Microsoft Windows [Versión 5.0.2600.5512]
(C) Copyright 2006 Microsoft Corp.
C:\Documents and Settings\Propietario>cls_
```

Luego se ejecutarán, ya sea la instrucción “*cd..*” o “*cd*” para desplazarte hacia el directorio raíz o simplemente hacia el disco duro (unidad C).



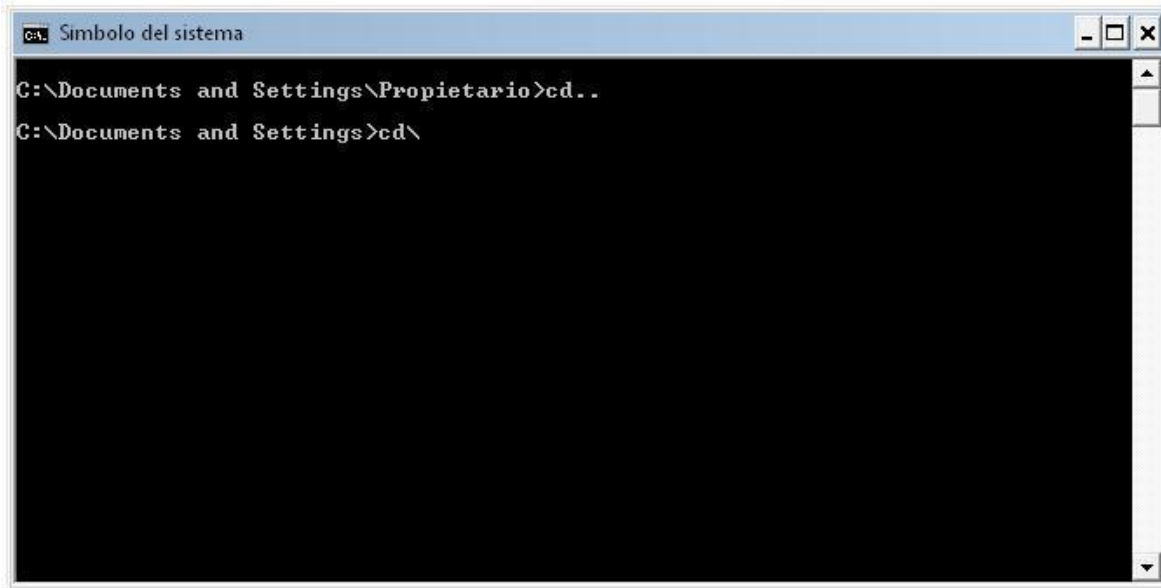
```
C:\Documents and Settings\Propietario>cd.._
```

No olvides dar ENTER para que se ejecuten los comandos.



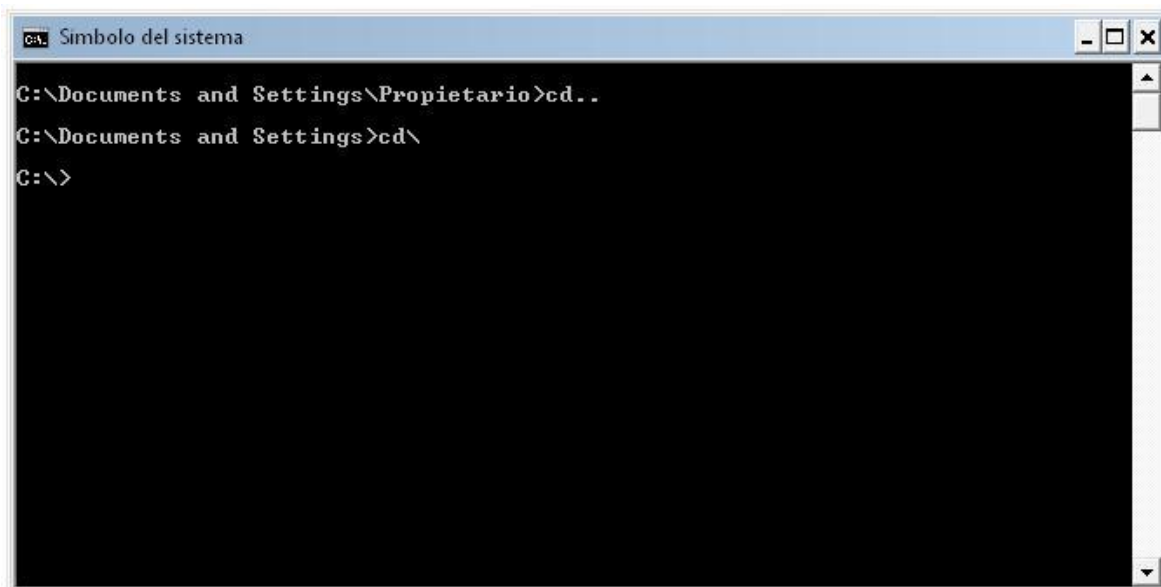
```
C:\Documents and Settings\Propietario>cd..  
C:\Documents and Settings>
```

No olvides dar ENTER para que se ejecuten los comandos.



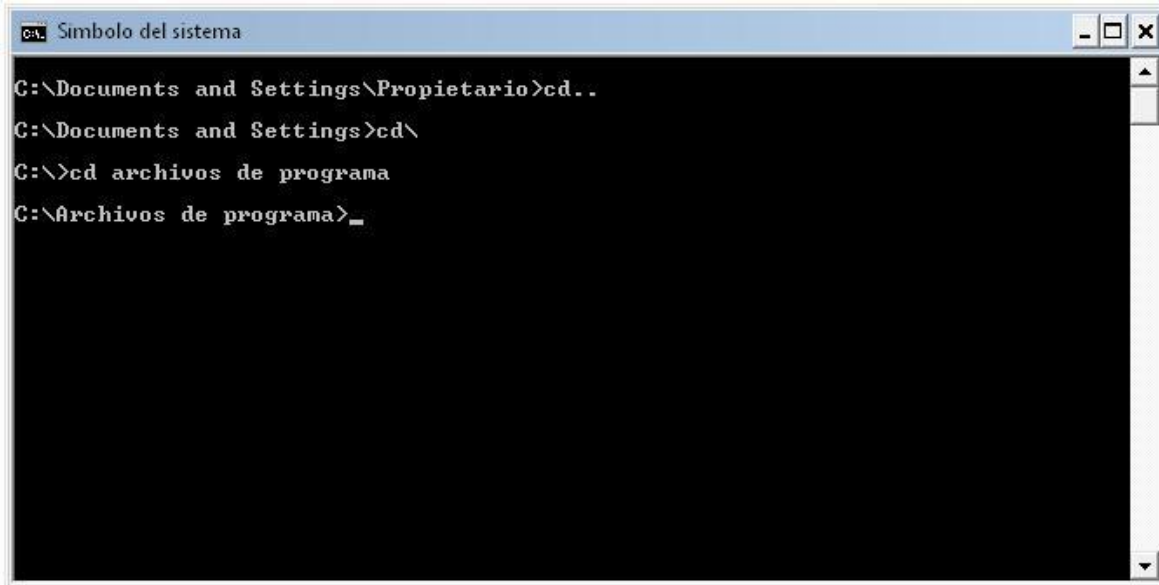
```
CA: Simbolo del sistema
C:\Documents and Settings\Propietario>cd..
C:\Documents and Settings>cd\
```

No olvides dar ENTER para que se ejecuten los comandos.



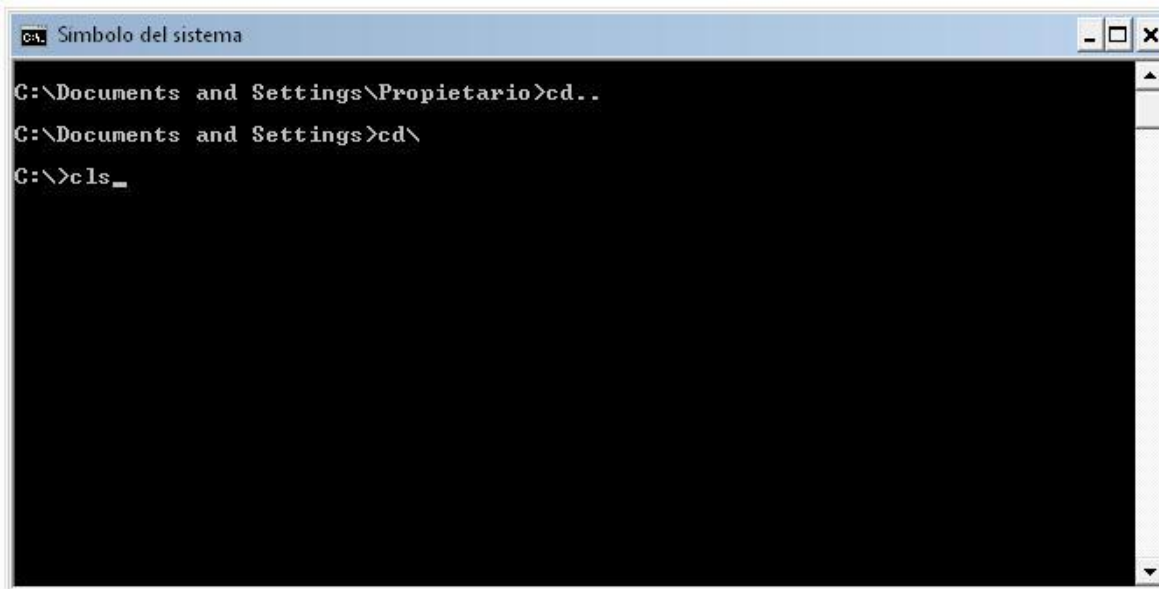
```
CA: Simbolo del sistema
C:\Documents and Settings\Propietario>cd..
C:\Documents and Settings>cd\
C:\>
```

Una vez que ya estas en la raíz del directorio, tendrás que ejecutar, ya sea los comandos "cd" junto con la dirección para que vayas carpeta por carpeta o el comando "cd dir\dir" para que vayas a la ruta específica directamente, enseguida muestro la serie de ventanas que se tendrán que aparecer:



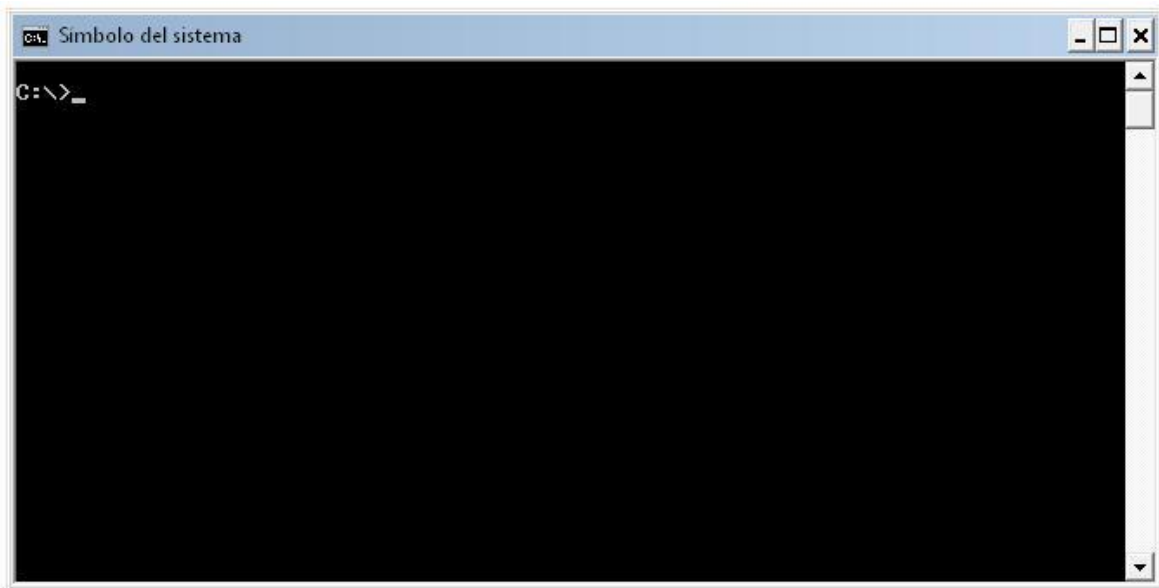
```
CA: Simbolo del sistema
C:\Documents and Settings\Propietario>cd..
C:\Documents and Settings>cd\
C:\>cd archivos de programa
C:\Archivos de programa>_
```

No olvides dar ENTER para que se ejecuten los comandos.



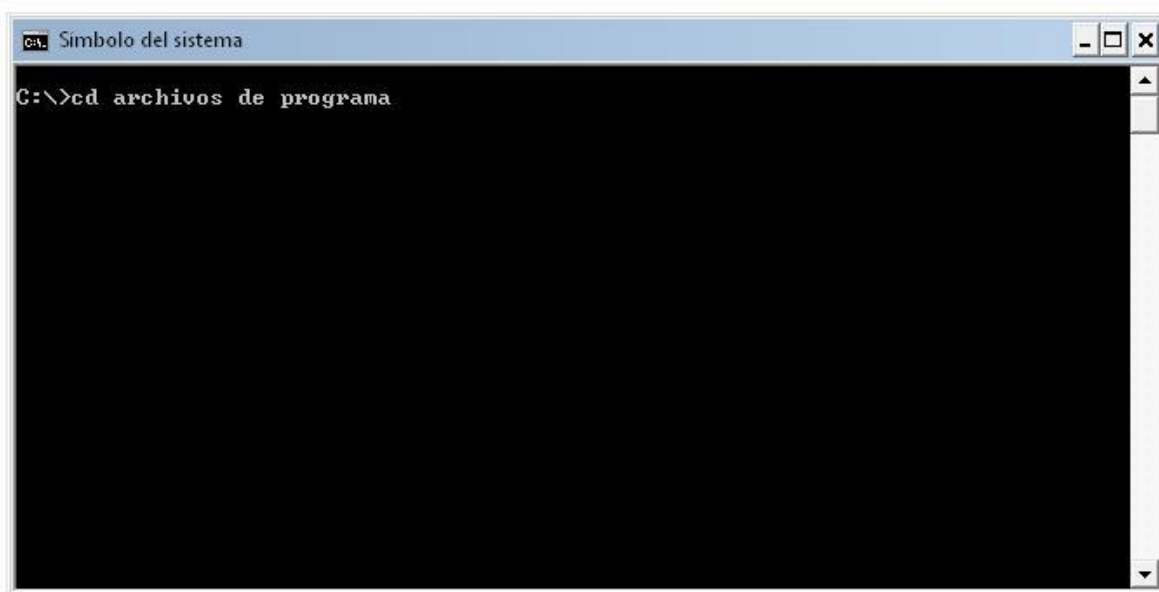
```
CA: Simbolo del sistema
C:\Documents and Settings\Propietario>cd..
C:\Documents and Settings>cd\
C:\>cls_
```

No olvides dar ENTER para que se ejecuten los comandos.



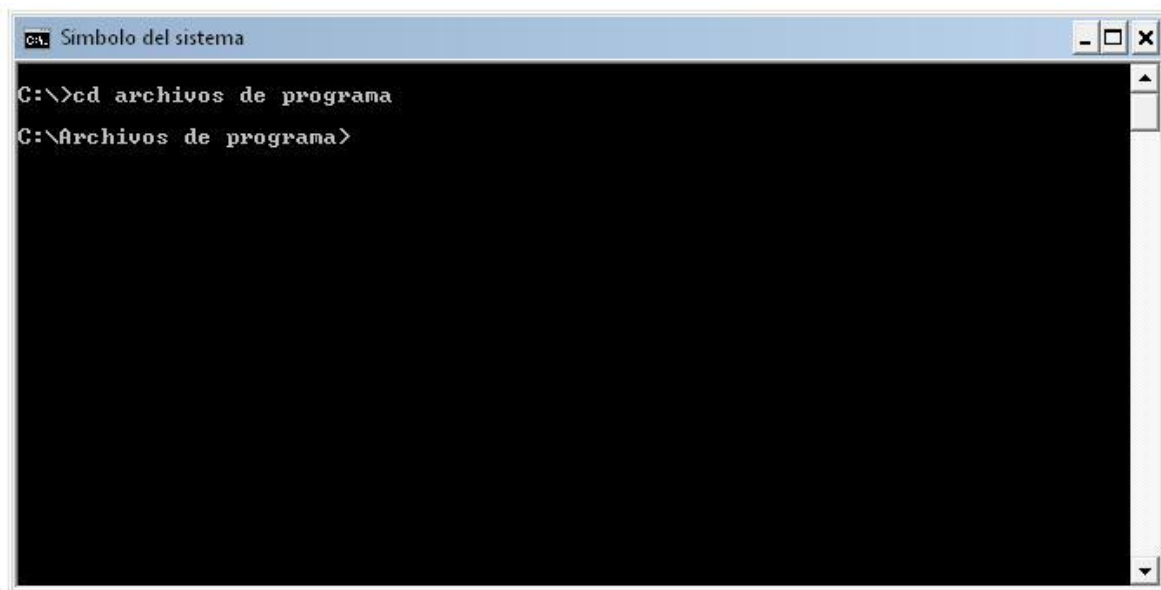
```
C:\>_
```

No olvides dar ENTER para que se ejecuten los comandos.



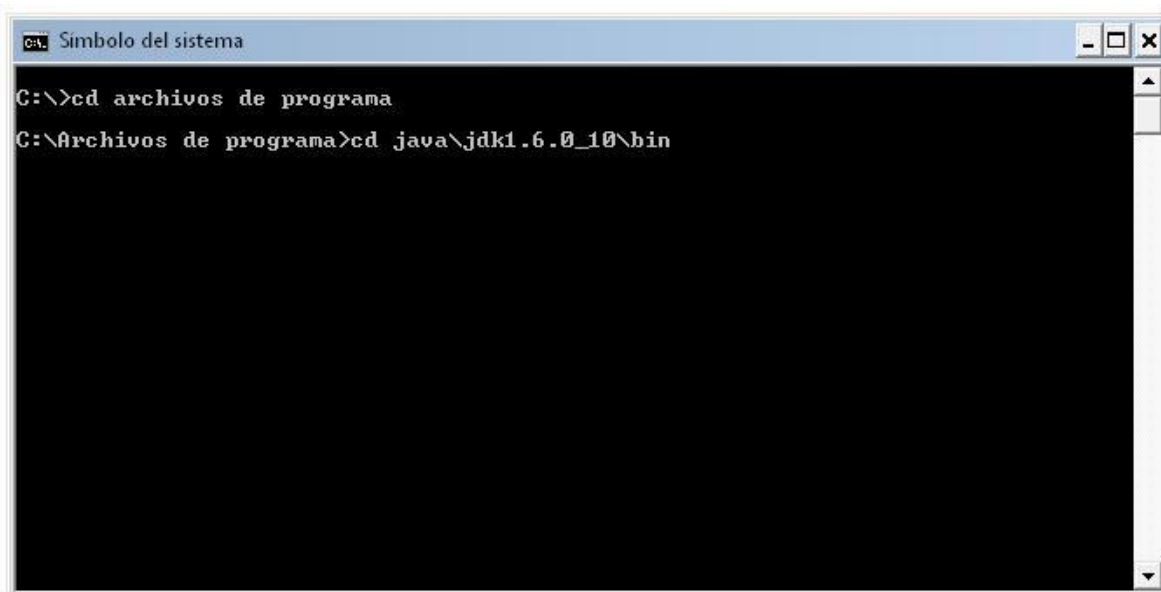
```
C:\>cd archivos de programa
```

No olvides dar ENTER para que se ejecuten los comandos.



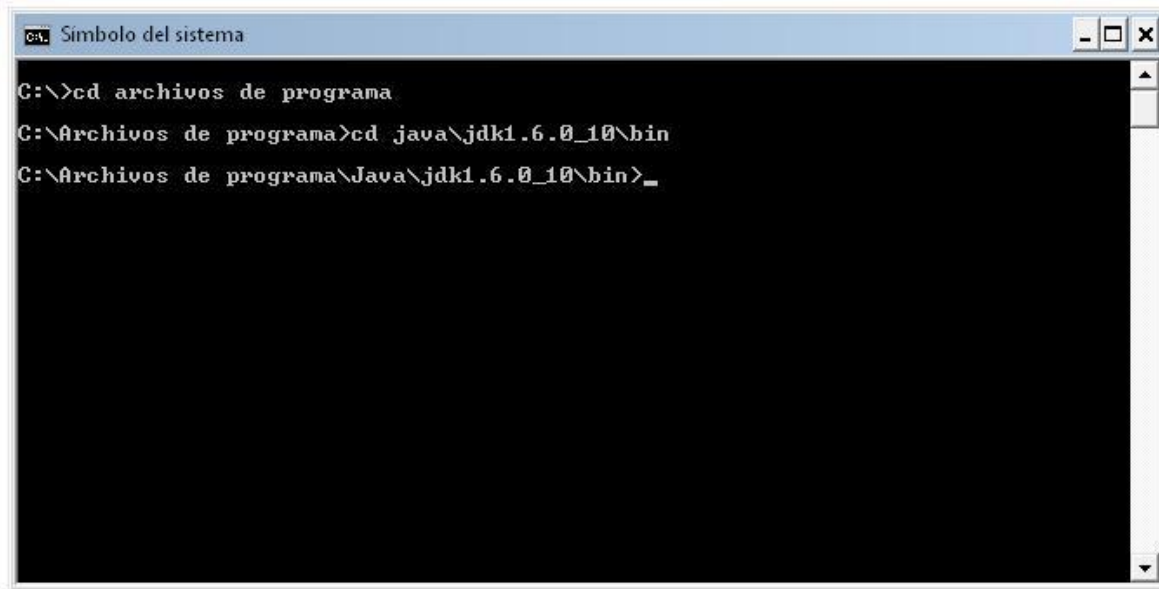
```
C:\>cd archivos de programa
C:\Archivos de programa>
```

No olvides dar ENTER para que se ejecuten los comandos.



```
C:\>cd archivos de programa
C:\Archivos de programa>cd java\jdk1.6.0_10\bin
```

No olvides dar ENTER para que se ejecuten los comandos.

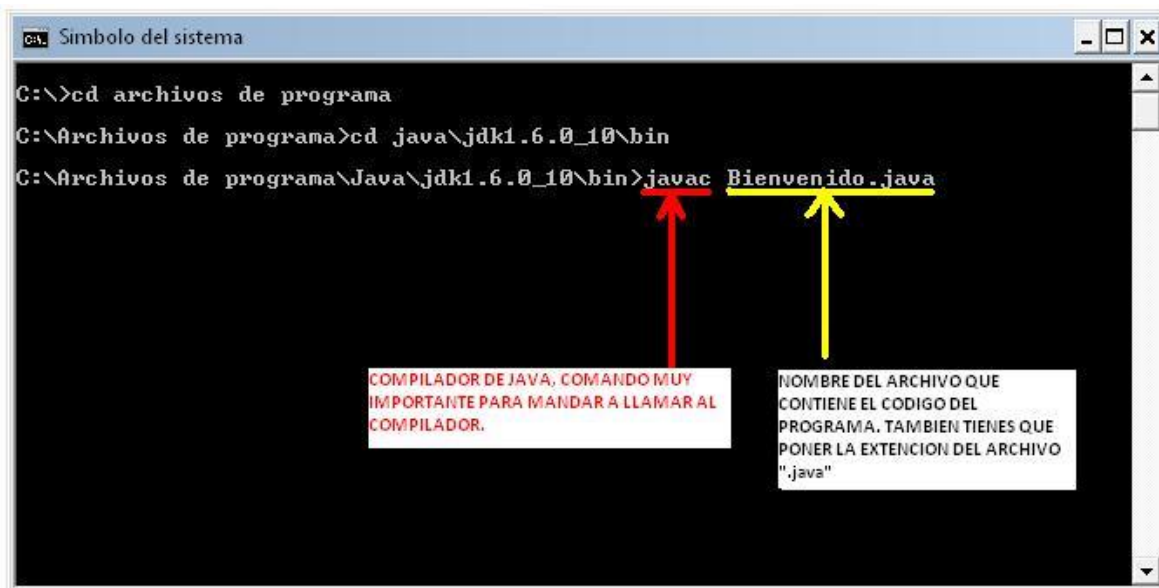


```
C:\>cd archivos de programa
C:\Archivos de programa>cd java\jdk1.6.0_10\bin
C:\Archivos de programa\Java\jdk1.6.0_10\bin>
```

Una vez hecho todo esto, se tendrá que pasar a la siguiente fase.

- 2. Compilado:** En esta fase, vendrá el proceso de compilación, esto quiere decir que crea los códigos de Bytes y los almacena en disco, en un archivo con extensión “.class”

Para poder llevar a cavo este proceso, se tendrán que ejecutar los siguientes comandos como se ilustran a continuación:



```
C:\>cd archivos de programa
C:\Archivos de programa>cd java\jdk1.6.0_10\bin
C:\Archivos de programa\Java\jdk1.6.0_10\bin>javac Bienvenido.java
```

COMPILADOR DE JAVA. COMANDO MUY IMPORTANTE PARA MANDAR A LLAMAR AL COMPILADOR.

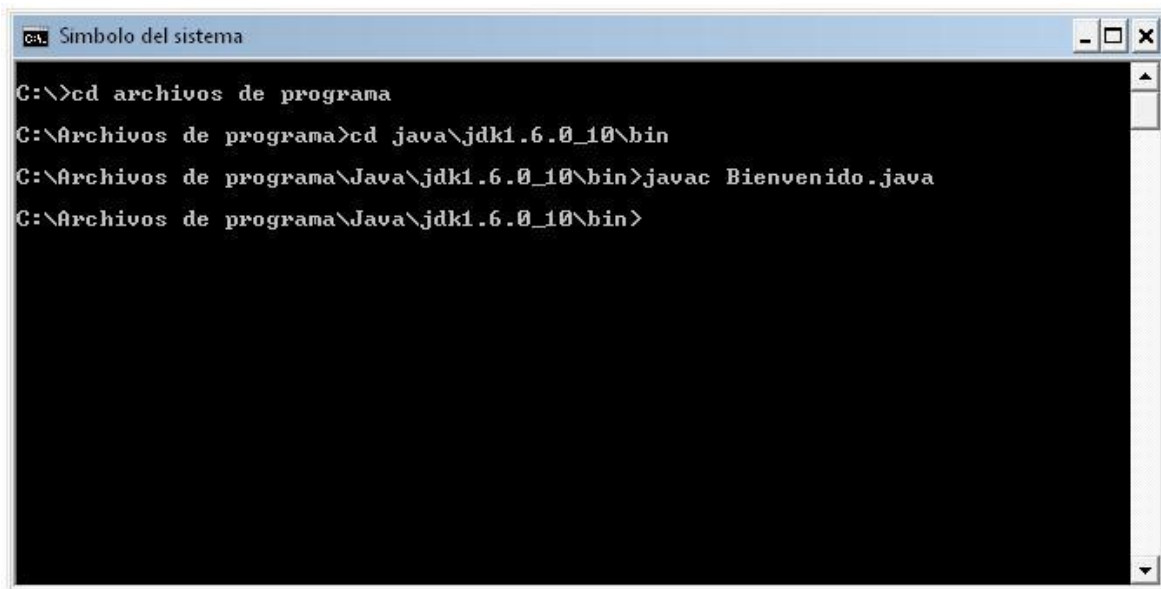
NOMBRE DEL ARCHIVO QUE CONTIENE EL CODIGO DEL PROGRAMA. TAMBIEN TIENES QUE PONER LA EXTENCION DEL ARCHIVO ".java"

En esta parte de la creación de tu primer programa es en donde viene el proceso de compilación, como te abras dado cuenta, en la ventana de comandos tendrás que escribir.

```
“javac Bienvenido.java”
```

Esto es: **javac** significa que se tendrá que utilizar en compilador de “javac”, esta instrucción manda a llamar simplemente al compilador y enseguida va el nombre del archivo que contiene el código del programa. “Bienvenido.java” Aquí se pone el nombre del archivo tal y como lo hayas guardado, es importante tomar en cuenta la Mayúsculas, recuerda que el nombre del archivo debe ser igual que el nombre de la clase principal, si no es así, es muy probable que a la hora de compilar se te anuncie ese error, en ese caso tendrás que abrir el código y modificar todos los errores que tengas y posteriormente guardar los cambios al archivo para que estos surjan efecto a la hora de volver a compilarlo.

En caso de que no tengas errores, simplemente en la ventana de comandos, se mostrara una nueva línea indicando la ruta seleccionada para que puedas meter más instrucciones como se muestra en la siguiente ventana.



```
ca. Simbolo del sistema
C:\>cd archivos de programa
C:\Archivos de programa>cd java\jdk1.6.0_10\bin
C:\Archivos de programa\Java\jdk1.6.0_10\bin>javac Bienvenido.java
C:\Archivos de programa\Java\jdk1.6.0_10\bin>
```

3. **Cargador de clases:** En esta tercer fase, ahora si, una vez que termino el proceso de compilación y si no tuviste ningún error se creara un archivo con la extensión “.class”, y es aquí, en estos archivos en donde se encuentra el código que puede leer la computadora (contiene códigos de bytes).



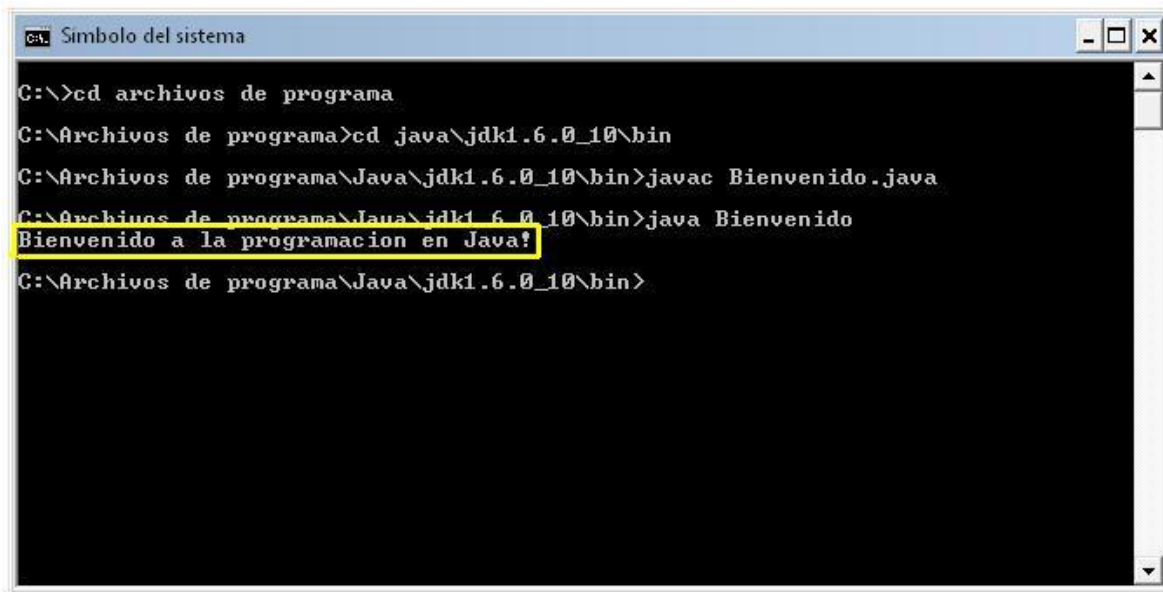
Es posible que entre más líneas tenga tu código, o sea cuando ya hagas aplicaciones muy grandes, este tipo de archivos se irán incrementando.

4. **Verificador de código de bytes:** En esta curta fase, el verificador de código de bytes únicamente confirma que todos los códigos de bytes sean válidos y no violen las restricciones de seguridad de Java.
5. **Ejecución:** Una vez hecho todo lo anterior ahora si, llego la hora de ejecutar tu programa. Esto se hará ejecutando el comando “*java*” junto con el nombre del archivo que contiene el código en bytes o sea el archivo “.class”

A screenshot of a Windows command prompt window titled 'Simbolo del sistema'. The window has a black background with white text. The text shows the following commands and their outputs:

```
C:\>cd archivos de programa
C:\Archivos de programa>cd java\jdk1.6.0_10\bin
C:\Archivos de programa\Java\jdk1.6.0_10\bin>javac Bienvenido.java
C:\Archivos de programa\Java\jdk1.6.0_10\bin>java Bienvenido
```

Una vez ejecutado el comando “*java Bienvenido*” y dar ENTER, se mostrara la siguiente ventana:



```
CA: Simbolo del sistema
C:\>cd archivos de programa
C:\Archivos de programa>cd java\jdk1.6.0_10\bin
C:\Archivos de programa\Java\jdk1.6.0_10\bin>javac Bienvenido.java
C:\Archivos de programa\Java\jdk1.6.0_10\bin>java Bienvenido
Bienvenido a la programacion en Java!
C:\Archivos de programa\Java\jdk1.6.0_10\bin>
```

Esta forma de compilar el programa se llama modo consola, es muy parecido a la forma en que se ejecutan los programas de C o C++, quizá no tenga una apariencia muy llamativa pero es muy importante aprenderla, en el siguiente capítulo mostrare otras 2 formas mucho más llamativas de programar en Java.

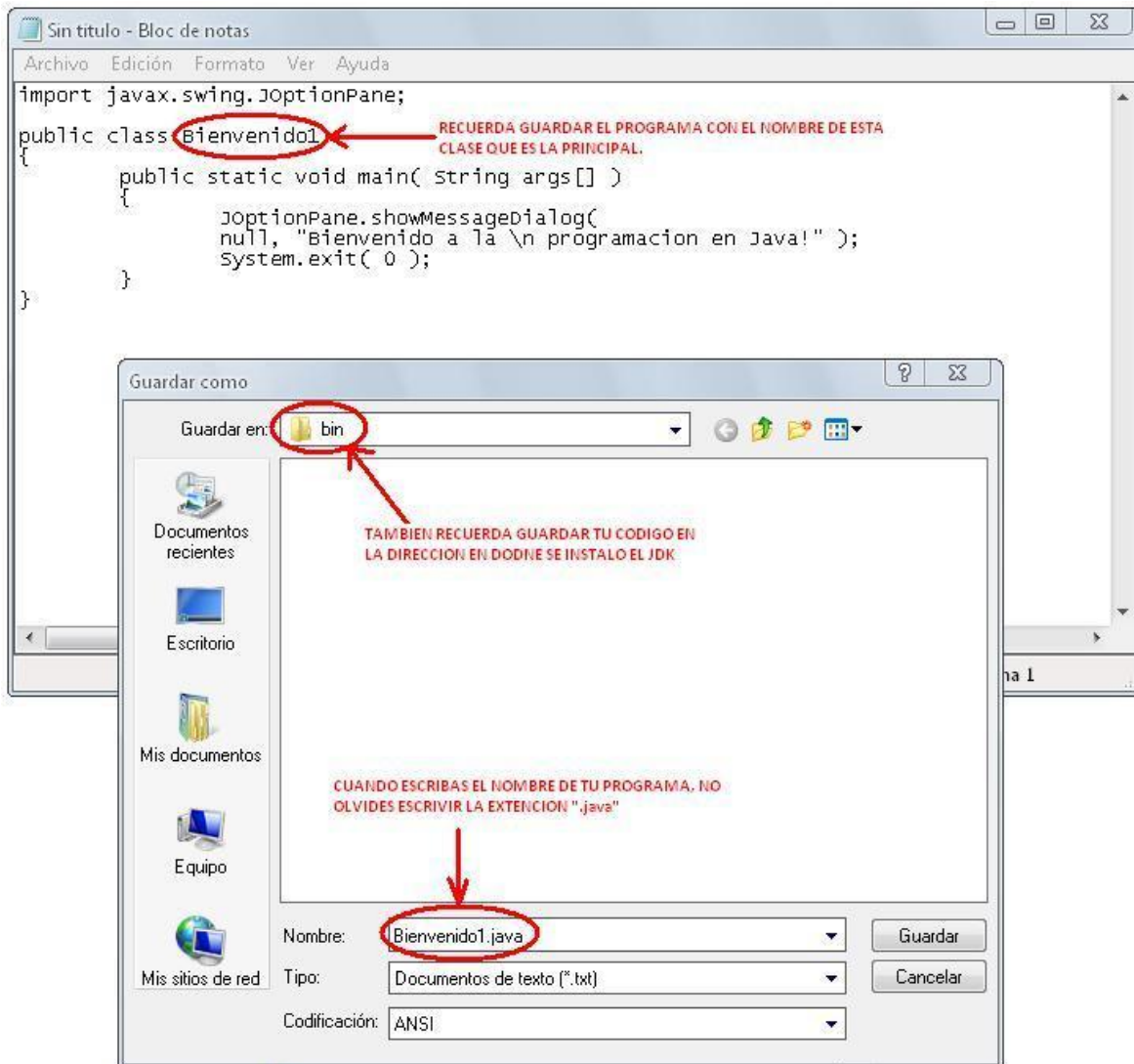
DIFERENTES FORMAS DE PROGRAMAR EN JAVA.

Bueno, en general hay tres formas para programar en java, y estas con, la forma que acabamos de ver, "**Modo Consola**", "**Aplicaciones**", "**Applet**".

Ahora veremos y explicare, el modo de "Aplicación", esta forma es mucho más visible que la de modo consola, para poder crear una "Aplicación", En realidad lo único que cambia es la forma de escribir el código, ya que la forma de compilar es igual que la del Modo Consola.

Para poder programar, al igual que en el programa anterior, tendras que arir un editor de texto como el Bloc de Notas y escribir el siguiente código:

```
import javax.swing.JOptionPane;
public class Bienvenido1
{
    public static void main( String args[] )
    {
        JOptionPane.showMessageDialog( null, "Bienvenido a la programación en Java!" );
        System.exit( 0 );
    }
}
```



Una vez guardado el programa en la carpeta en donde está el compilador de Java, nuevamente tendrás que abrir una ventana de comandos "Símbolo del Sistema", para que puedas compilar tu código.

Ahora solo mostrare en una sola ventana todo lo que tienes que hacer, ya que en el programa anterior puse todas las ventanas, paso a paso.

```

C:\Documents and Settings\Propietario>cd\
C:\>cd archivos de programa\java\jdk1.6.0_10\bin
C:\Archivos de programa\Java\jdk1.6.0_10\bin>javac Bienvenido1.java
C:\Archivos de programa\Java\jdk1.6.0_10\bin>java Bienvenido1_
  
```

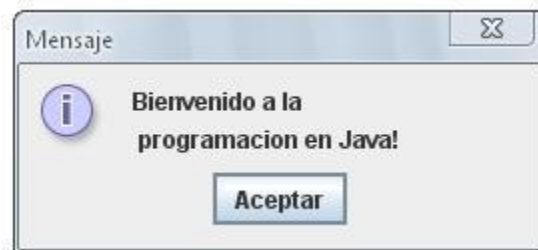
COMANDO cd\ PARA REGESAR AL DIRECTORIO RAIZ

COMANCO cd PARA IR AL DIR JDK EN LA BIN

UNA VEZ COMPILADO EL CODIGO, SE PROCEDE A EJECUTARLO CON EL CODIGO java, SEGUIDO DEL NOMBRE DEL ARCHIVO CREADO ".class" EL CUAL CONTIENE EL CÓDIGO EN BYTES PARA PODER EJECUTAR EL PROGRAMA.

COMANDO javac PARA MANDAR A LLAMAR AL COMPILADOR DE JAVA, SEGUIDO DEL NOMBRE DEL ARCHIVO QUE CONTIENE EL CÓDIGO DEL PROGRAMA, SEGUIDO DE LA EXTENSION ".java" PARA QUE SE PUEDA COMPILAR.

Una vez compilado el código, como se explico en la ventana anterior, a la hora de ejecutarlo con el comando “java”, aparecerá la ventana de nuestro programa:



Y como te lo prometí, este tipo de programación, es mucho más visual que la del modo consola, ero si no entendiste ni que escribiste, a continuación tratare de explicarte el código.

En la primer línea en donde escribiste:

```
import javax.swing.JOptionPane;
```

Bueno pues aquí el programa usa una declaración **import**. Los programas utilizan declaraciones **import** para identificar las clases predefinidas que se utilizan en un programa en Java. El compilador trata de cerciorarse de que estas utilizando correctamente las clases del API¹ de Java, debemos indicar el paquete en el que se encuentra esa clase.

¹ API por sus siglas (Interfaz de Programación de Aplicaciones de Java), es en donde se encuentran los paquetes que contienen clases predefinidas que tú como programador podrás utilizar, en vez de reescribir el código, por ejemplo como para el cuadro de texto del programa anterior. Es por eso que un paquete es una colección de clases. Los paquetes del API de Java se dividen en básicos y opcionales. La mayoría de los paquetes del API de Java comienzan, ya sea con “java” (Paquetes Básicos), “javax” (Paquetes Opcionales), o en algunos casos hay paquetes en el API de java que comienzan con “org”.

Es por eso que en esta línea, se le indica al compilador que nuestro programa esta utilizando la clase ***JOptionPane*** del paquete ***javax.swing***. Este paquete contiene muchas clases que ayudan a los programadores de Java a crear Interfaces Gráficas de Usuario (GUI's) para las aplicaciones.

La siguiente línea del programa:

```
public class Bienvenido1
```

Aquí lo que se hace es declarar el nombre de la clase pública, o sea el nombre de la clase principal para poder correr un programa en Java.

A continuación en la siguiente línea:

```
public static void main( String args[] )
```

Aquí, se hace referencia a que el método empezara la ejecución de la aplicación de Java.

En la siguiente línea vemos una llave “{” abierta y en la línea 10 vemos una llave cerrada “}”, esto, como en lenguaje c, es para establecer un determinado espacio para el método, esto indica que las instrucciones escritas posteriormente solo pertenecen al método “***main***”

En las siguientes líneas:

```
JOptionPane.showMessageDialog(null, “Bienvenido a la \n programacion en Java!”);
```

Aquí se hace referencia al se manda a llamara al método ***showMessageDialog*** de la clase ***JOptionPane***, para mostrar el cuadro de texto en donde se mostrara escrito el mensaje que esta entre comillas. A su vez, este método requiere dos argumentos y estos se separan por comas “,” y estos argumentos son: ***null***, determina que el cuadro de dialogo se colocara en el centro de la pantalla, el segundo argumento es simplemente el mensaje a mostrar.

En la siguiente línea:

```
System.exit(0);
```

Aquí simplemente termina la aplicación con la ventana utilizando en método ***exit*** de la clase ***System***². El elemento “0” indica al método ***exit*** que l programa terminó correctamente, un valor diferente indica generalmente que ocurrió un error.

² Esto se requiere para terminar cualquier aplicación que muestre una interfaz gráfica de usuario. La clase ***System*** se encuentra en el paquete ***java.lang***. Este paquete es el único de la API de Java que no requiere de una declaración ***import***.

Otra forma de programar en Java es la generación de **applets**.

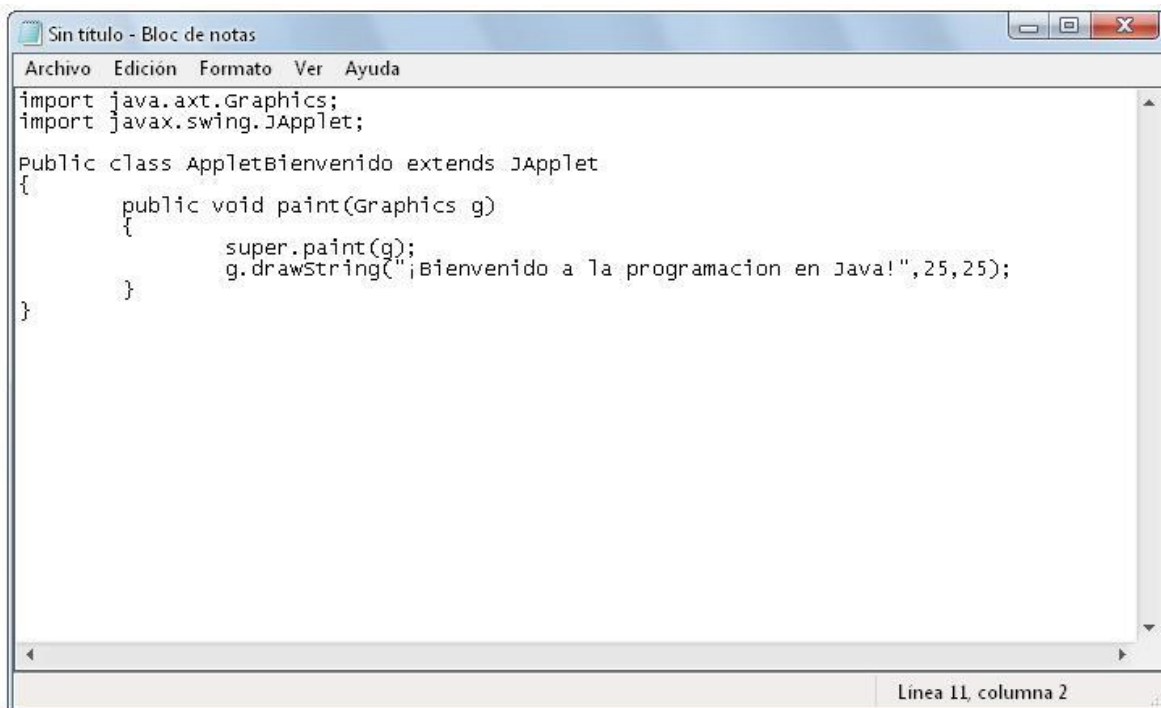
Esta es una manera de incluir código a ejecutar en los clientes que visualizan una página web. Se trata de pequeños programas hechos en Java, que se transfieren con las páginas web y que el navegador ejecuta en el espacio de la página.

Los *applets* son más difíciles de programar que los scripts en Javascript y requerirán unos conocimientos básicos o medios del lenguaje Java.

La principal ventaja de utilizar *applets* consiste en que son mucho menos dependientes del navegador que los scripts en Javascript, incluso independientes del sistema operativo del ordenador donde se ejecutan. Además, Java es más potente que Javascript, por lo que el número de aplicaciones de los *applets* podrá ser mayor.

A continuación mostrare un ejemplo de creación de un *applet*, paso a paso.

En primer lugar tendrás que escribir el siguiente código en un editor de textos, para esta utilizare un bloc de notas:

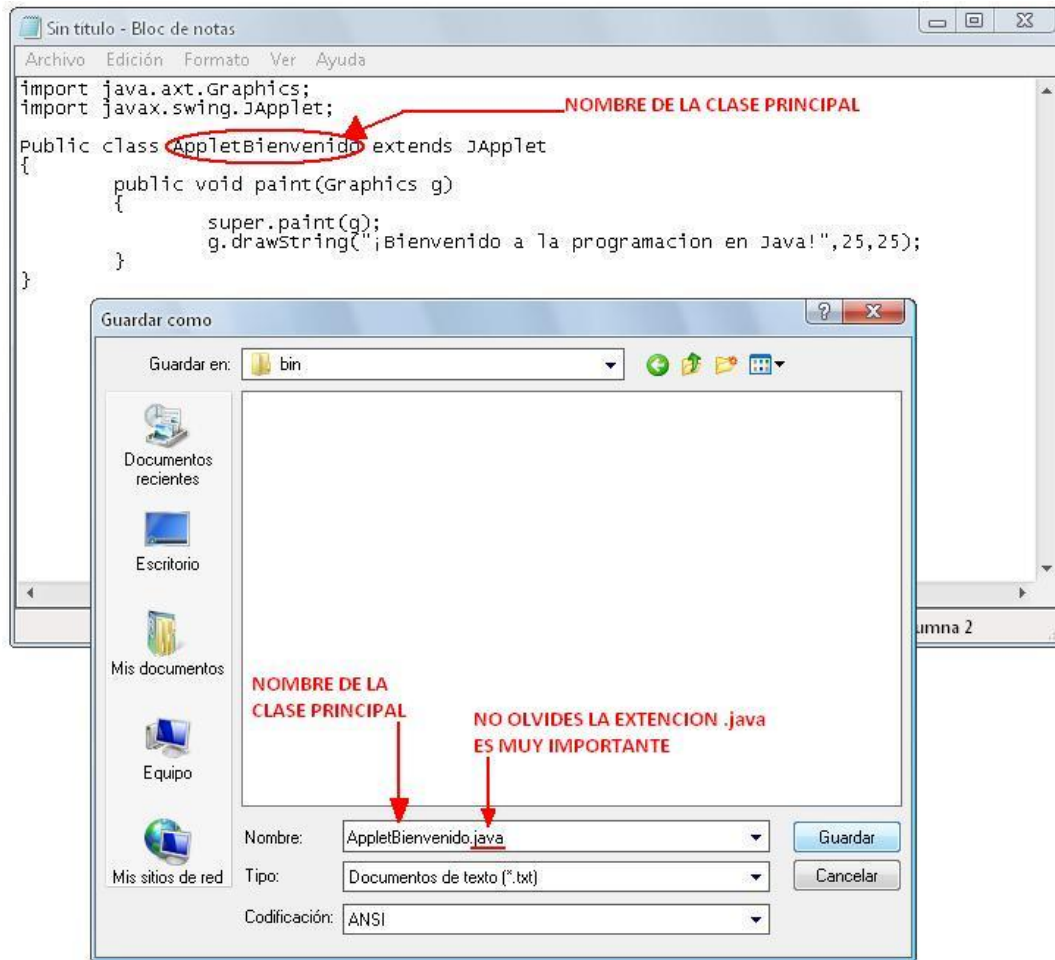
A screenshot of a Windows Notepad window titled "Sin titulo - Bloc de notas". The window contains the following Java code:

```
import java.awt.Graphics;
import javax.swing.JApplet;

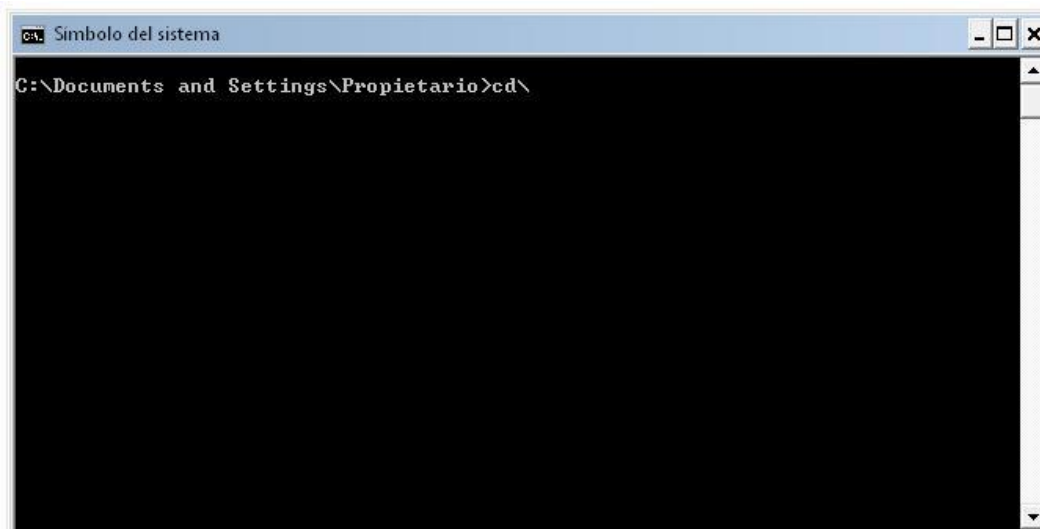
public class AppletBienvenido extends JApplet
{
    public void paint(Graphics g)
    {
        super.paint(g);
        g.drawString("¡Bienvenido a la programación en Java!", 25, 25);
    }
}
```

The status bar at the bottom right of the window indicates "Línea 11, columna 2".

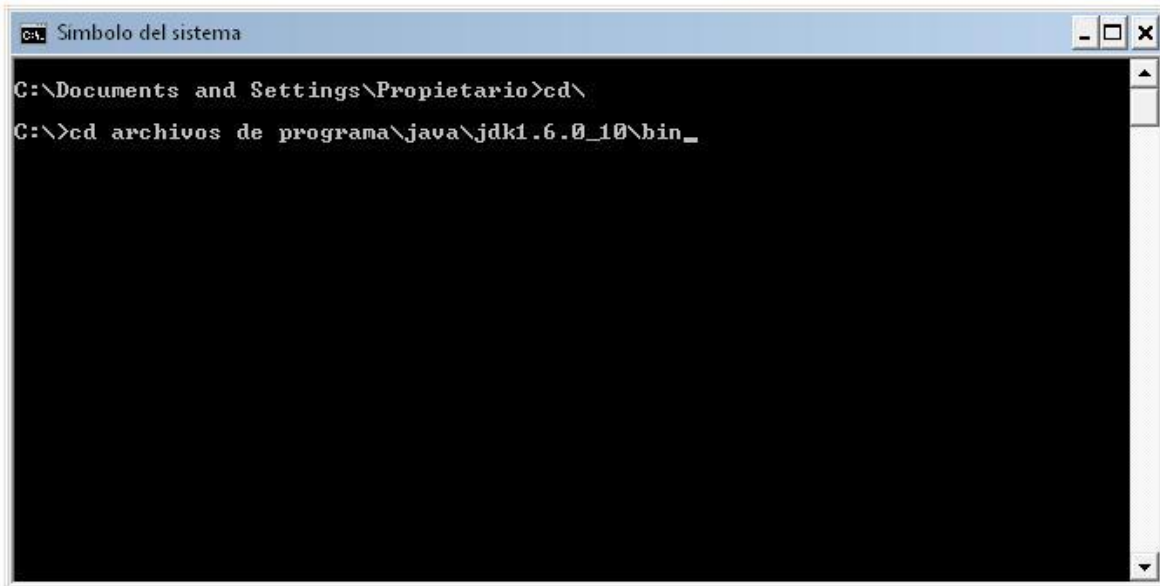
Una vez escrito el código en un Bloc de Notas, tendrás que guardar el archivo con el nombre de la clase principal tal y como lo hayas escrito, en este caso "AppletBienvenido".



No olvides escribir la extensión ".java" ya que sin esta extensión, no podrás compilar el código. Ahora si, una vez guardado el código con la extensión, tendrás que ejecutar el Símbolo del Sistema y escribir los comandos descritos anteriormente para desplazarte por las direcciones de la PC.

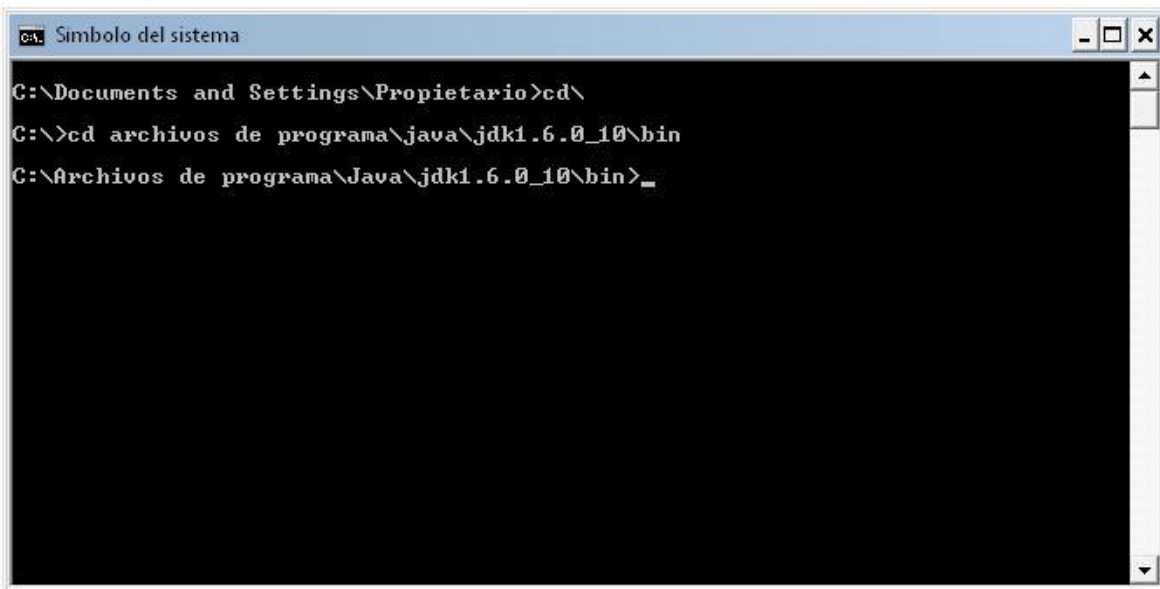


Es esta parte se empezó escribiendo el comando “cd\” para despasarme directamente hasta la raíz del directorio, posteriormente se utilizo el comando “cd” para ir directamente hasta el directorio en donde se encuentra instalado el compilador de Java.



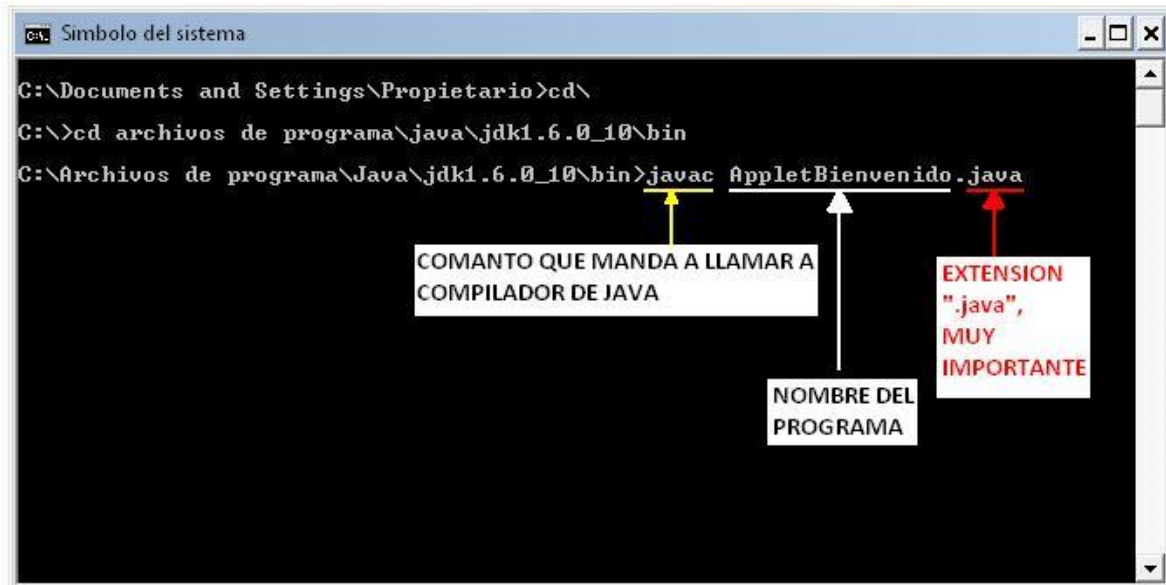
```
CA Simbolo del sistema
C:\Documents and Settings\Propietario>cd\
C:\>cd archivos de programa\java\jdk1.6.0_10\bin_
```

No olvides dar ENTER para que se ejecuten los comandos.



```
CA Simbolo del sistema
C:\Documents and Settings\Propietario>cd\
C:\>cd archivos de programa\java\jdk1.6.0_10\bin
C:\Archivos de programa\Java\jdk1.6.0_10\bin>_
```

Ahora, como ya vimos, para poder compilar un programa de java se tiene que escribir el comando "javac" seguido del nombre del código junto con la extensión del programa, o sea ".java".



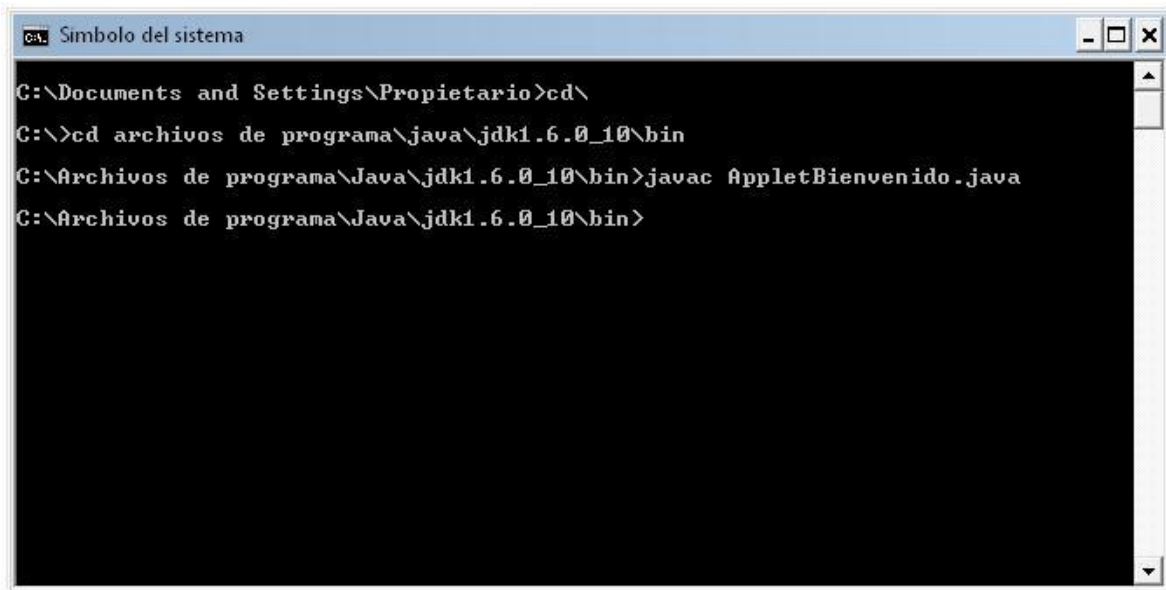
```
C:\Documents and Settings\Propietario>cd\  
C:\>cd archivos de programa\java\jdk1.6.0_10\bin  
C:\Archivos de programa\Java\jdk1.6.0_10\bin>javac AppletBienvenido.java
```

COMANTO QUE MANDA A LLAMAR A COMPILADOR DE JAVA

NOMBRE DEL PROGRAMA

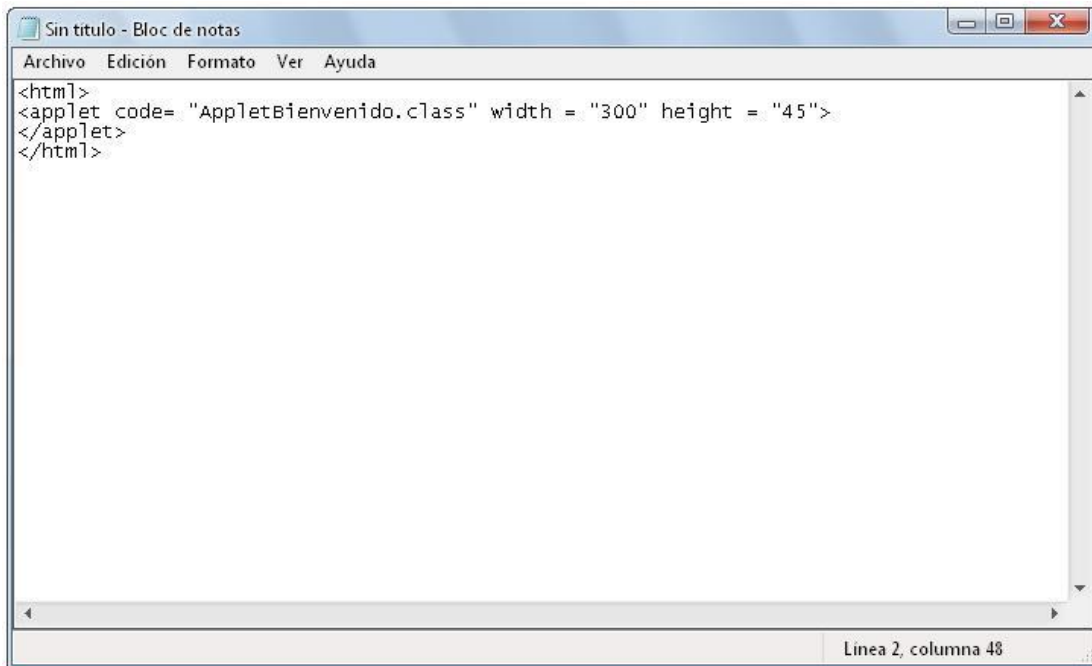
EXTENSION ".java", MUY IMPORTANTE

Una vez compilado el código y que se te aparezca la ventana siguiente en donde indica que el código esta bien y que no existe ningún error tendrás que seguir al otro paso.



```
C:\Documents and Settings\Propietario>cd\  
C:\>cd archivos de programa\java\jdk1.6.0_10\bin  
C:\Archivos de programa\Java\jdk1.6.0_10\bin>javac AppletBienvenido.java  
C:\Archivos de programa\Java\jdk1.6.0_10\bin>
```

Ya que no tuviste ningún error al compilar tu programa tendrás que abrir otro editor de texto o Bloc de Notas en donde escribirás el siguiente código.

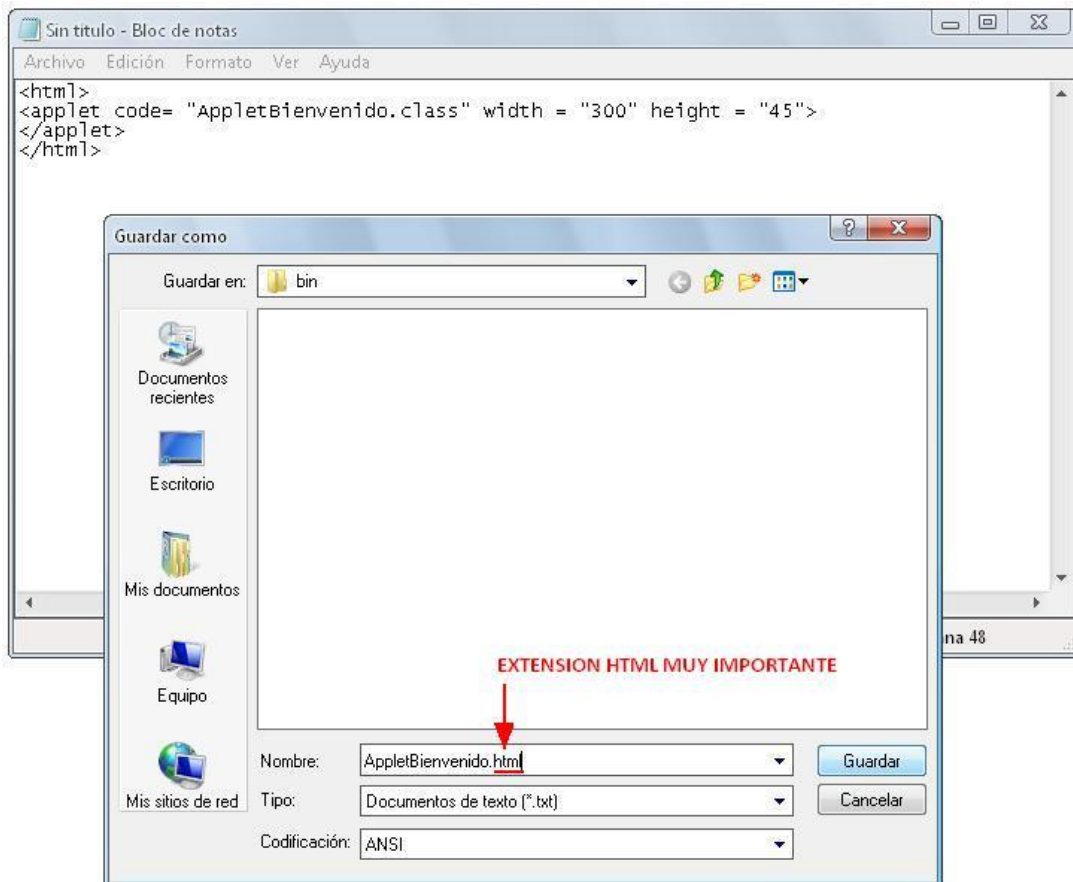


Sin título - Bloc de notas

```
Archivo Edición Formato Ver Ayuda
<html>
<applet code= "AppletBienvenido.class" width = "300" height = "45">
</applet>
</html>
```

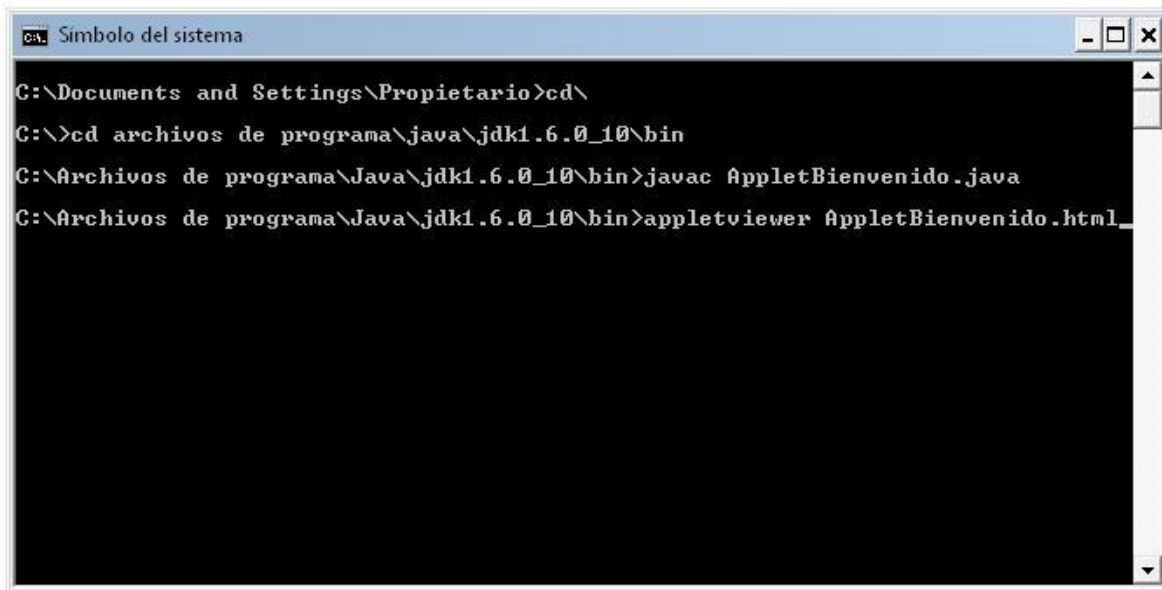
Linea 2, columna 48

Este código es para el applet pueda ser ejecutado ya que se tiene que abrir con un archivo "html" que como ya vimos es el encargado de abrir aplicaciones de la web.



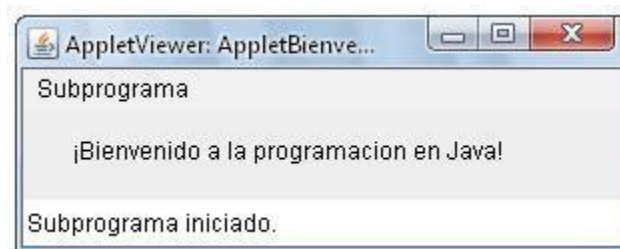
Una vez escrito el código, tienes que guardar el archivo pero ahora con la extensión “.html” como se muestra arriba y también con el nombre que guardaste el programa principal que viene siendo el archivo con extensión “.java”. Este archivo se tiene que guardar también en la carpeta en donde esta el compilador de java, o sea en la carpeta “bin” del “jdk”.

Ya que hayas terminado de guardar el archivo que contiene el código del “html”, hay dos formas de ejecutar tu programa, la primera es: regresar a Símbolo del Sistema en donde escribirás el comando *appletviewer*³ para que se pueda ejecutar el archivo con la extensión “.html” seguido del nombre del archivo junto con su extensión correspondiente que viene siendo la “html”.



```
Simbolo del sistema
C:\Documents and Settings\Propietario>cd\
C:\>cd archivos de programa\java\jdk1.6.0_10\bin
C:\Archivos de programa\Java\jdk1.6.0_10\bin>javac AppletBienvenido.java
C:\Archivos de programa\Java\jdk1.6.0_10\bin>appletviewer AppletBienvenido.html
```

Al dar ENTER después de escribir lo anterior se abrirá una ventana que muestra tu código de JAVA pero en una ventana distinta y que va orientada a la web, es decir podrás hacer programas en Java para que puedan ser ejecutados en cualquier navegador de internet.

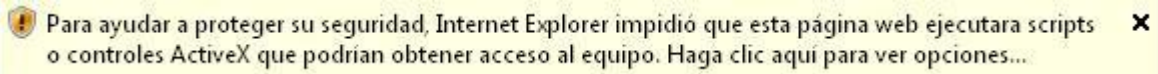


³ El comando *appletviewer*, es una aplicación que permite ver en funcionamiento applets, sin la necesidad de utilizar un navegador de World-Wide-Web.

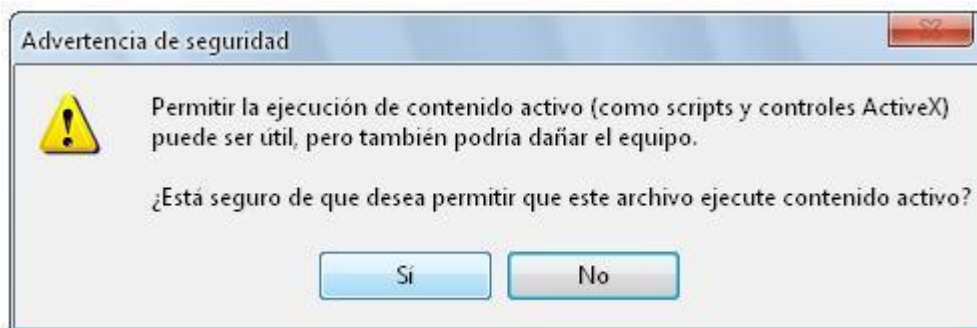
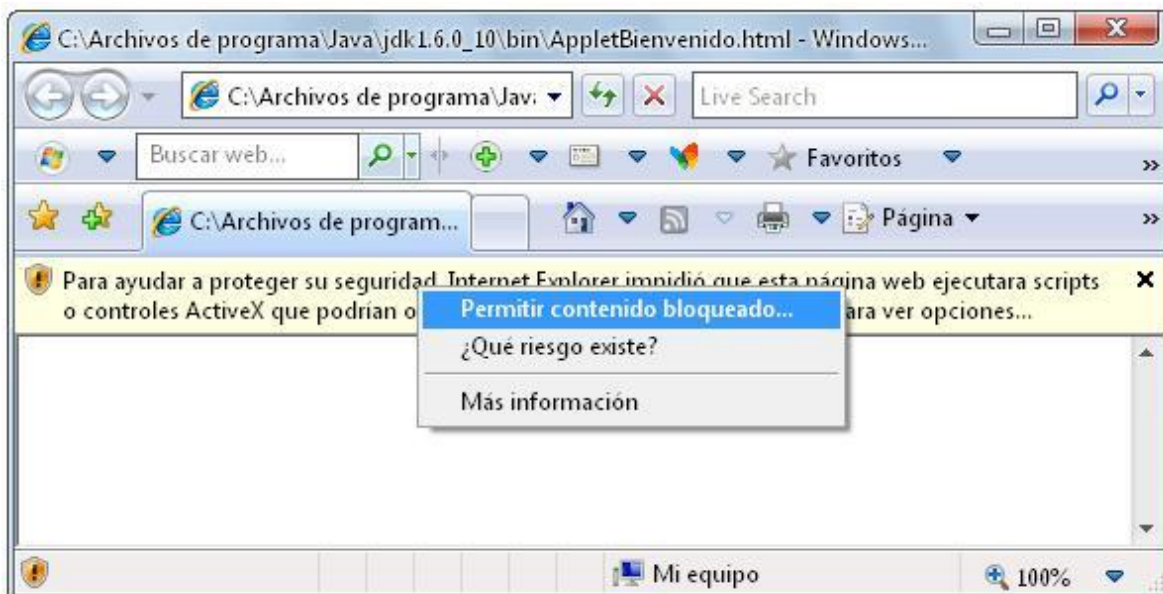
La segunda forma más directa de ejecutar un *applet*, es simplemente dar doble clic en el archivo que contiene la extensión "*html*"



Al abrir este archivo, simplemente se ejecutara el navegador que tengas, es posible que se aparezca un mensaje similar al siguiente:

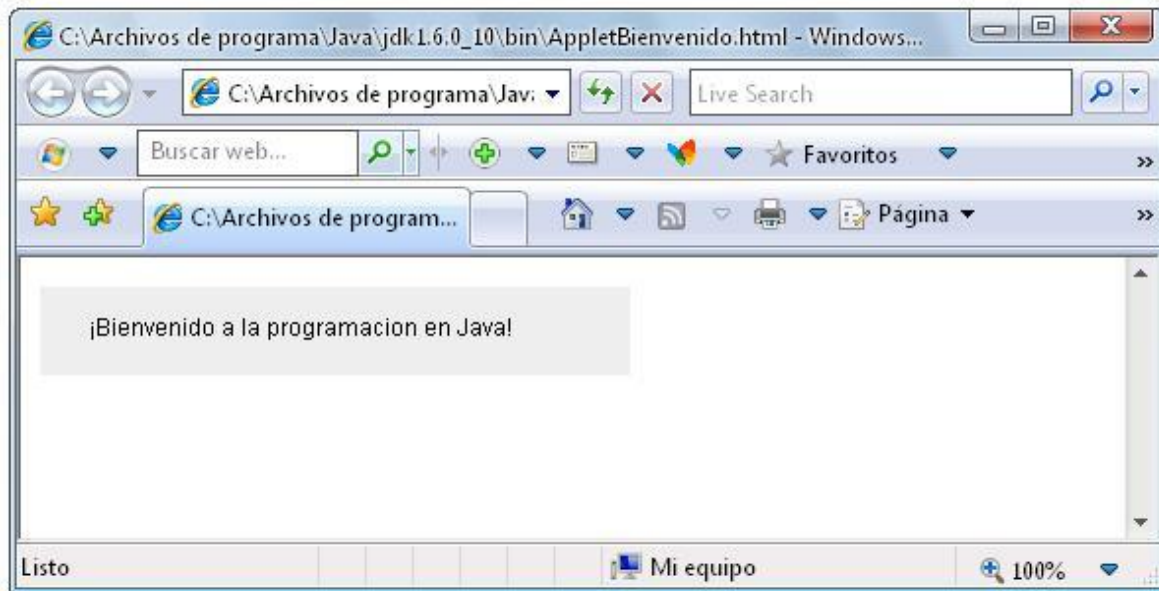


Aquí lo único que tendrás que hacer es dar clic derecho sobre el mensaje y dar clic en "*Permitir contenido bloqueado...*"



Este tipo de mensajes simplemente te informan acerca de que hay aplicaciones que pudieran introducirse de forma maliciosa a tu ordenador en el caso de entrar a internet, cave mencionar que un applet que tú hagas no será necesario mantener una conexión a internet para

que esta pueda ser ejecutada y por consecuencia no tendrás que preocuparte de los Virus de la red así que una vez hecho lo anterior se abrirá la siguiente ventana en la cual se ejecutara el código que escribiste desde un principio.



Una vez hecho todo esto, te explicare todo lo que escribiste en cuanto a código.

En las primeras líneas de tu primer código:

```
import java.awt.Graphics;
```

```
import javax.swing.JApplet;
```

Lo único que hiciste fue importar las clases *Graphics* del paquete *java.awt* y *JApplet* del paquete *javax.swing* eso servirá para que se pueda hacer un applet y para poder pintar el texto.

En la siguiente línea:

```
public class AppletBienvenido extends JApplet
```

Solo mencionas el nombre de la clase principal que viene siendo *AppletBienvenido*, junto con la extensión *JApplet*, esta extensión se hace únicamente para poder manejar *Applets*.

En la otra línea:

```
public void paint(Graphics g)
```

Aquí solo estas creando un método nuevo que contendrá lo que pintaras dentro del applet. La parte "*public void*" es en donde se hace referencia a la creación de un método de manera pública, la parte "*paint*" indica que se va a pintar algo, es decir en este caso se pintara un texto. "*(Graphics g)*" indica que se declarara una variable "*g*" de la clase *Graphics*.

Posteriormente el la siguiente línea:

```
super.paint(g);
```

Aquí se manda a llamar la versión del método *paint* de la superclase *JApplet*.

En la última línea del código.

```
g.drawString("¡Bienvenido a la programacion en Java!",25,25);
```

Indica ala PC que realice una acción dibujar los caracteres de la cadena ("*¡Bienvenido a la programacion en Java!*") en el applet. Los números "25, 25" indican a la PC en donde se pintara el texto, esta pate de los números la puedes modificar, esto te ayudara a guiarte y tener una noción de en donde se esta pintando. Las coordenadas son x, y respectivamente.

Como podrás darte cuenta, el proceso de programar en Java quizá sea un poco tedioso, ya que eso de estar abriendo Blocs de Notas y Símbolo del Sistema e introducir los comandos sea un poco aburrido y a la ves desesperante, es por eso que existen diversos programas que facilitan todo el procedimiento de edición y compilación, son muy buenos y bastantemente fáciles de utilizar, entre los más conocidos se encuentran JCreator, NetBeans y Eclipse. A continuación mencionare una breve explicación de cada uno de ellos.

JCreator

JCreator es un entorno de desarrollo para Java, esta basado en la programación de C. Existen dos versiones de este programa, una es comercial (JCreator PRO) y por la misma razón contiene algunas utilidades adicionales en comparación a la versión gratuita (JCreator LE). El programa JCreator actúa como una máscara sobre el JDK y permite usarlo en ambiente de Windows. Por ello el Java Development Kit debe estar instalado previamente en la máquina para poder instalar JCreator.

Para conseguir el programa hay que seguir el vínculo dando clic en el ícono del programa que se encuentra abajo, este vinculo te llevara a la pagina principal de JCreator, tendrás que ir a la zona de descargas.



NetBeans IDE

Es un ambiente Integrado de desarrollo "Integrated development environment" (es por eso lo de IDE) desarrollado en Java y creado para crear aplicaciones Java; nos sirve para escribir, compilar, testear y debugear dichas aplicaciones e incluye un editor de textos, herramientas de diseño visuales, entre otras características como que es **Open Source**.....

Para obtener el programa, solo bastara con seguir el vínculo del icono NetBeans de abajo, este vinculo te llevara a la página principal de NetBeans así que tendrás que ir a la zona de descargas.



Eclipse

Eclipse es una plataforma de desarrollo basada en Java. Es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, ha sido usada para desarrollar entornos de desarrollo integrados (IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse).

Para poder obtener este programa, solo basta con hacer clic en el vínculo que se encuentra en el icono del programa aquí abajo, Este vinculo te llevara a la pagina principal de Eclipse, tendrás que ir a la zona de descargas.



COMO HACER UN .JAR

En primer lugar, antes que nada, explicare que es un ".jar", para empezar un ".jar" viene siendo el sustituto de un ".exe" ya que el lenguaje de Java no se puede crear archivos ejecutables debido a que es un lenguaje multiplataforma, esto es que se puede correr en cualquier Sistema operativo, y los archivos ejecutables son dependientes de cada plataforma.

Esto provoca que tengamos que usar la consola para ejecutar nuestros programas; lo que supone un gran inconveniente para usuarios sobre todo acostumbrados a los entornos gráficos.

Es por eso que un archivo ".jar" es el sustituto de un ".exe" en Windows y como mencione anteriormente este manual va dirigido a la plataforma de Windows.

Existen varias formas de crear archivos “.jar” pero aquí solo mencionare una y es la siguiente.

En primer lugar tendrás que haber creado una aplicación de java, y para ejemplificar este proceso utilizaremos el programa “*Bienvenido1*” que creamos como aplicación.

Para esto tendrás que pasar previamente por todo el proceso de compilación y ejecución para la verificación del código.

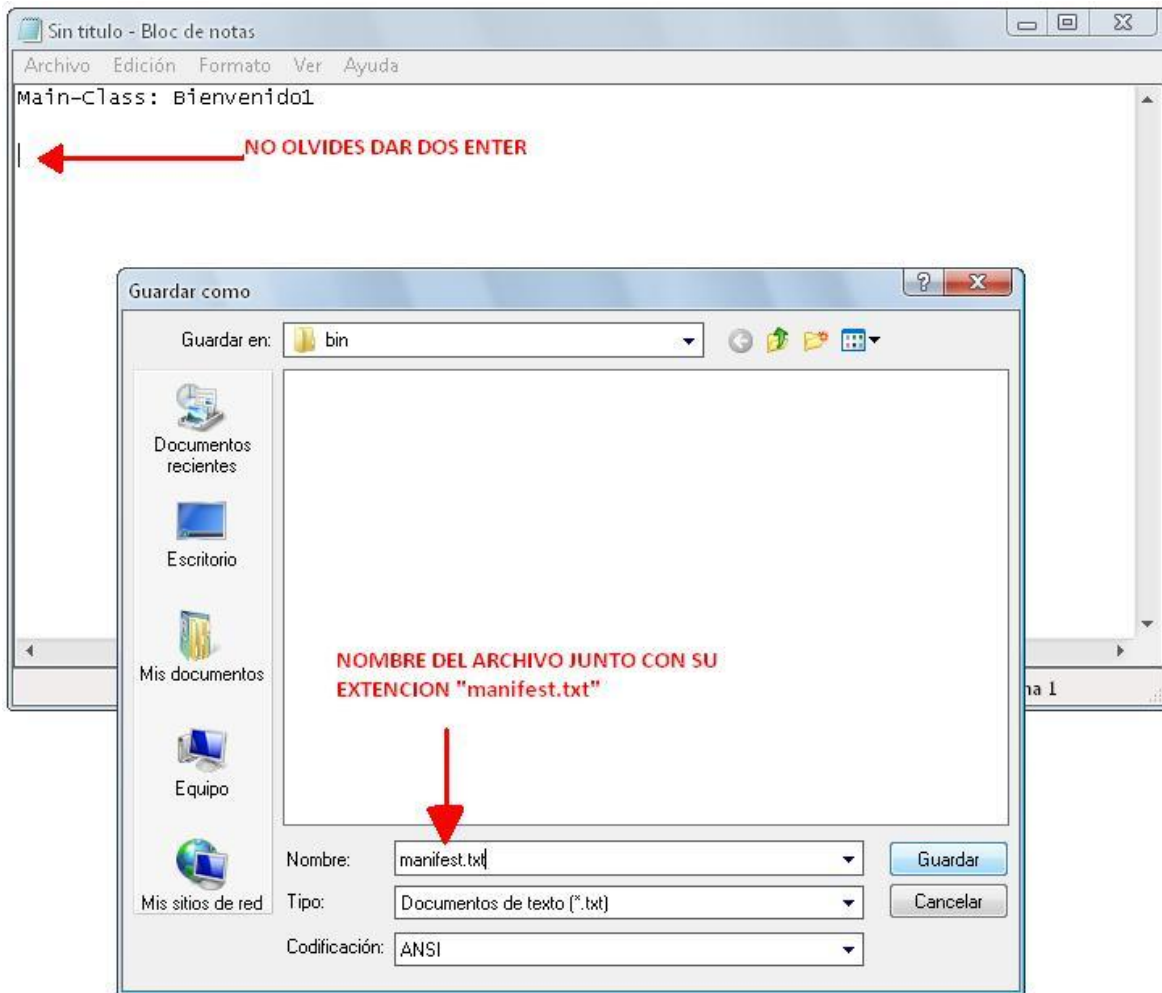
Ahora, teniendo en cuenta que el código ya esta compilado puesto que lo utilizamos en ejemplos anteriores, deberemos tener los siguientes archivos:



Recordando que el primer archivo contiene el código de java y el segundo archivo contiene los códigos que puede entender tu ordenador.

Ahora si vendrá el procedimiento para crear un “.jar”. En primer lugar tienes que abrir un Bloc de Notas o cualquier editor de texto y escribir lo siguiente:

Main-Class: Nombre del programa (dar dos enter)



Es importante eso de dar dos ENTER puesto que si no lo haces simplemente el archivo se creara pero no podrá ejecutarse.

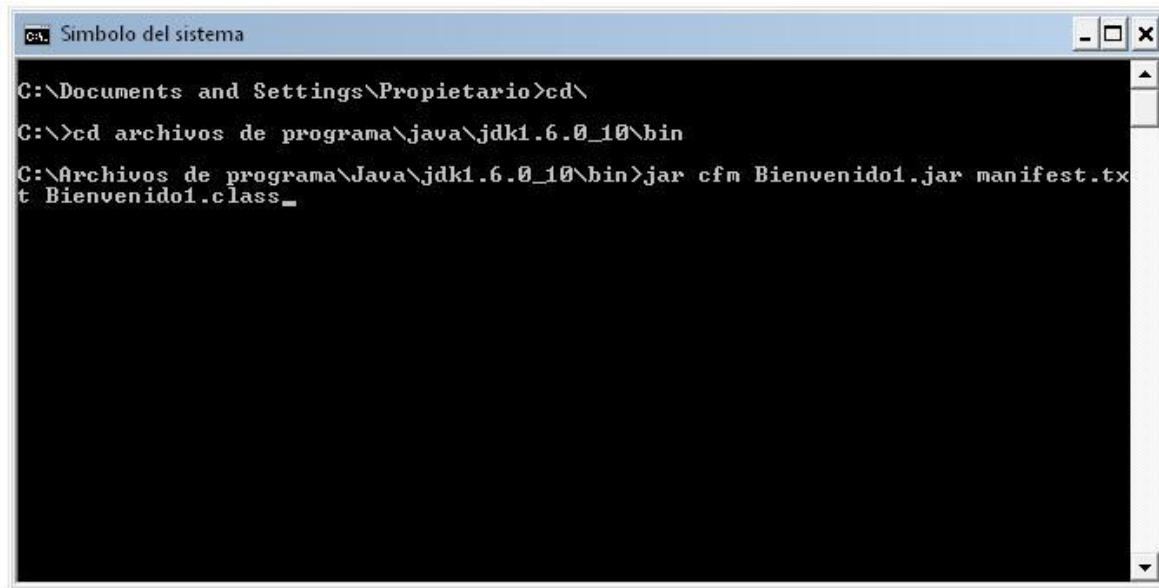
Después de esto deberás obtener el siguiente archivo



También es importante guardarlo en la carpeta “bin” del “jdk” en donde se encuentra instalado Java, ya que es en esa dirección en donde podrás la siguiente instrucción.

Ahora tendrás que ir nuevamente a Símbolo del Sistema a ejecutar la siguiente instrucción.

```
jar cfm nombrequequierasdar.jar manifest.txt Nombredelaclase.class
```



```
C:\Documents and Settings\Propietario>cd\  
C:\>cd archivos de programa\java\jdk1.6.0_10\bin  
C:\Archivos de programa\Java\jdk1.6.0_10\bin>jar cfm Bienvenido1.jar manifest.txt Bienvenido1.class_
```

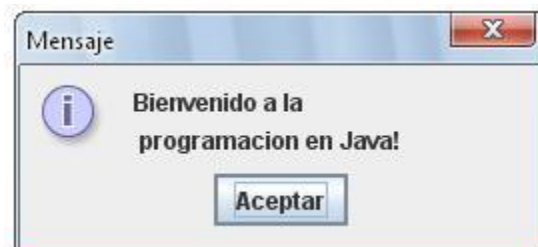
Ya no explicare los comandos que se utilizan para desplazarse por esta ventana puesto que ya los mencione muchas veces.

En cuanto a la nueva línea, el comando *"jar cfm"* se utiliza para crear el *".jar"*, posteriormente *"Bienvenido1.jar"* es el nombre que se le dará al archivo con dicha extensión, este nombre si puede cambiar, la siguiente instrucción *"manifest.txt"* solo manda a llamar el archivo creado previamente como ya vimos, en la ultima instrucción tendrás que poner el nombre de todos los archivos *".class"* que se hayan creado a la hora de compilar tu código, estos pueden variar de acuerdo al tamaño en líneas que contenga tu programa, es decir que entre más grande sea tu código, más archivos *".class"* tendrás. Pero como aquí únicamente es un mensaje pues solo se creo un archivo.

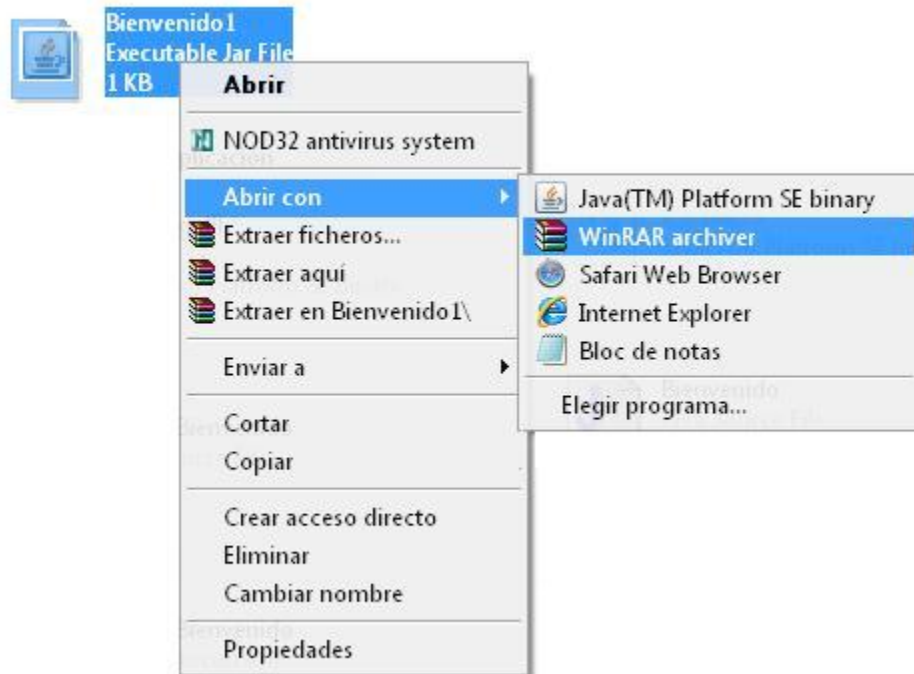
Una vez escrito en las instrucciones anteriores y haber dado ENTER, obtendrás un archivo *".jar"*.



Solo bastara con dar doble clic sobre el y listo, ya podrás obtener un archivo ejecutable para la plataforma de Windows



Una recomendación que te puedo dar es que en lugar de escribir dichos archivos “.class”, en caso de tener un código muy grande, únicamente tienes que abrir tu archivo “.jar” con WinRAR y añadir únicamente seleccionando y arrastrando estos archivos “.class”



Esto lo menciono, ya que te puede resultar demasiado tedioso estar escribiendo el nombre de cada archivo en el Símbolo del Sistema. Esto se puede hacer puesto que el archivo “.jar” solo jala una clase y posteriormente se le pueden añadir mas elementos solo arrastrándolos.