

Descargando Java 9

1.- Para descargar java en la versión 9 se ingresa a la página de Oracle, donde se seleccionará la opción de descargar java, dependiendo del sistema operativo que se tenga en la computadora

www.oracle.com/technetwork/java/javase/downloads/jdk9-downloads-3848520.html

ORACLE

Menu

Sign In

Cot

Oracle Technology Network / Java / Java SE / Downloads

Java SE

Java EE

Java ME

Java SE Advanced & Suite

Java Embedded

Java DB

Web Tier

Java Card

Java TV

New to Java

Community

Java Magazine

Overview

Downloads

Documentation

Community

Technologies

Training

Java SE Development Kit 9 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:

- Java Developer Newsletter: From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to **Java**.
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK 9.0.4 checksum

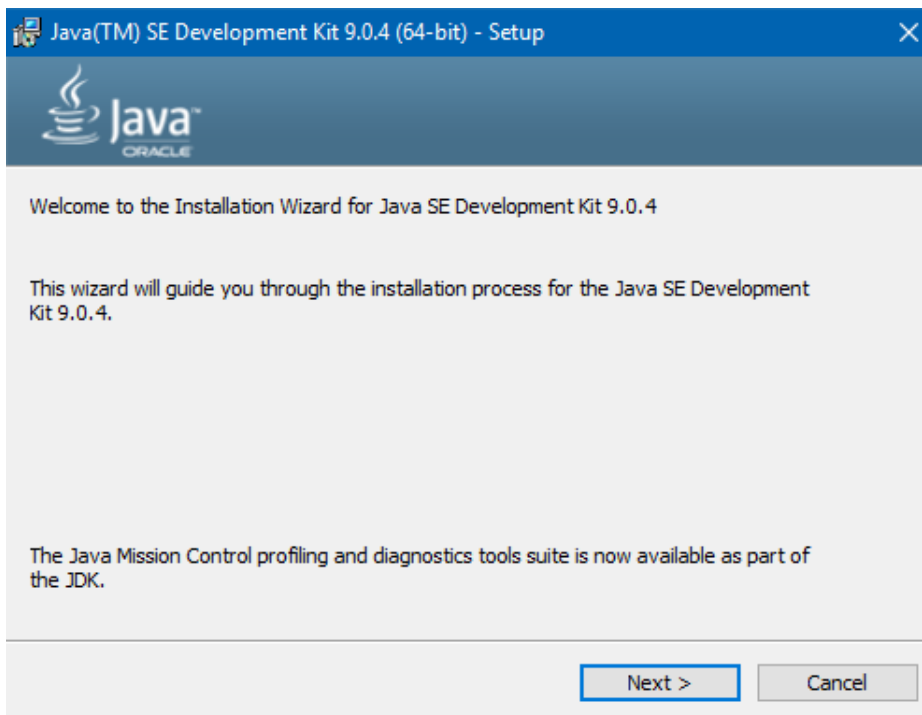
Java SE Development Kit 9.0.4

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

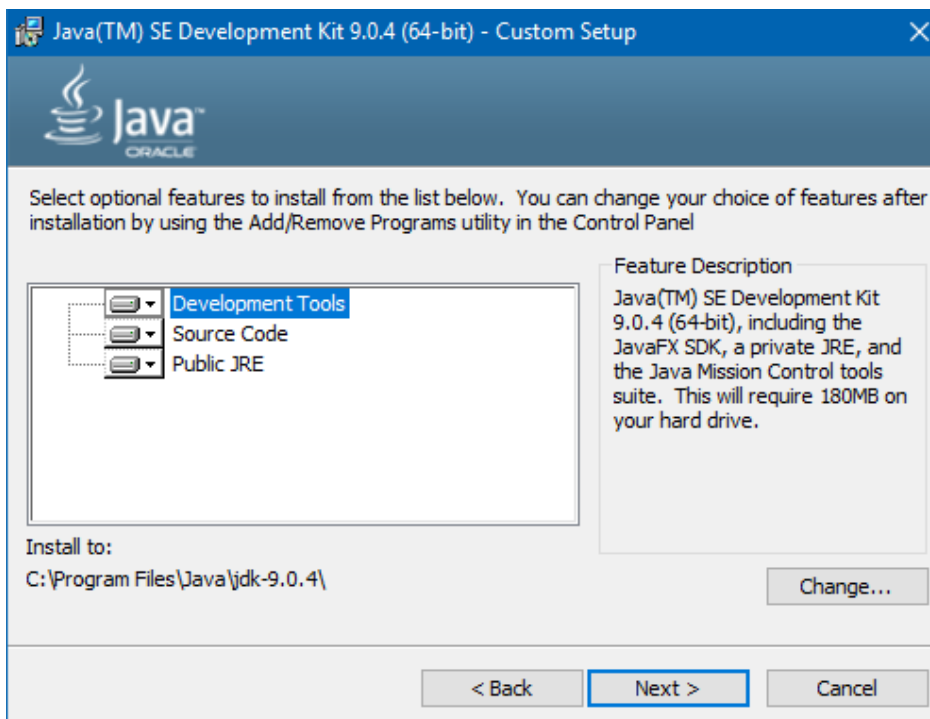
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux	305.07 MB	jdk-9.0.4_linux-x64_bin.rpm
Linux	338.21 MB	jdk-9.0.4_linux-x64_bin.tar.gz
macOS	382.11 MB	jdk-9.0.4_osx-x64_bin.dmg
Windows	375.56 MB	jdk-9.0.4_windows-x64_bin.exe
Solaris SPARC	206.97 MB	jdk-9.0.4_solaris-sparcv9_bin.tar.gz

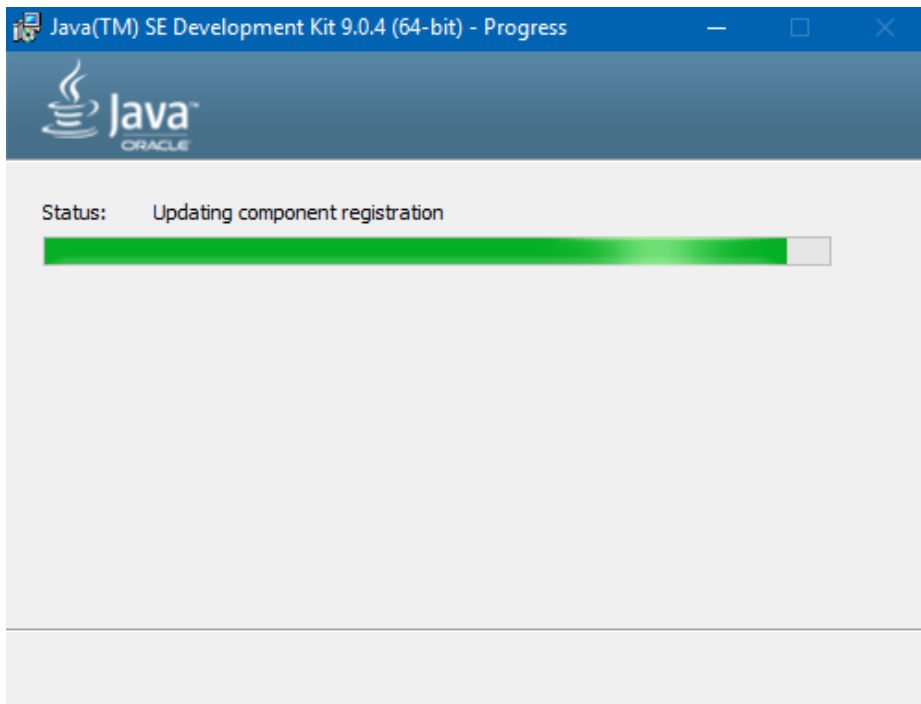
2.-Una vez descargado se deberá abrir y aparecerán ventanas las cuales preguntarán si se desea ejecutar, seleccionando “aceptar” aparecerá la siguiente ventana, la cual indica la instalación de java. Se selecciona “Next >”.



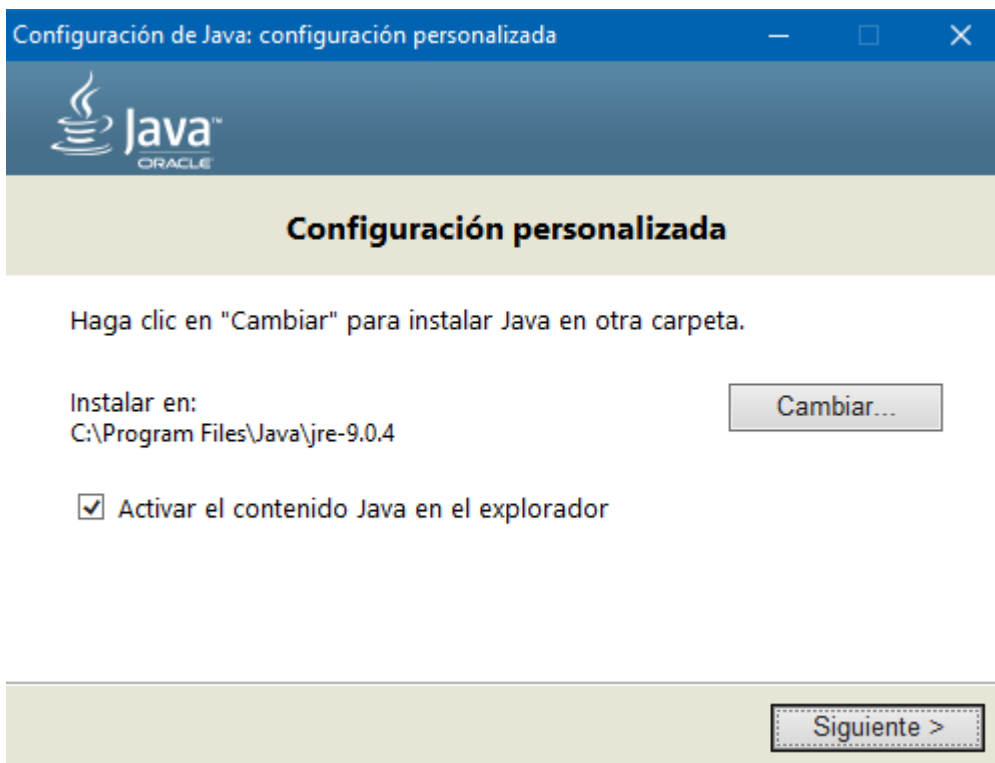
3.- Preguntará cuál paquete se desea instalar, se selecciona “Development Tools” y se da click en “Next >”.



4.- Aparecerá una nueva ventana la cual indica el progreso de instalación de los componentes.



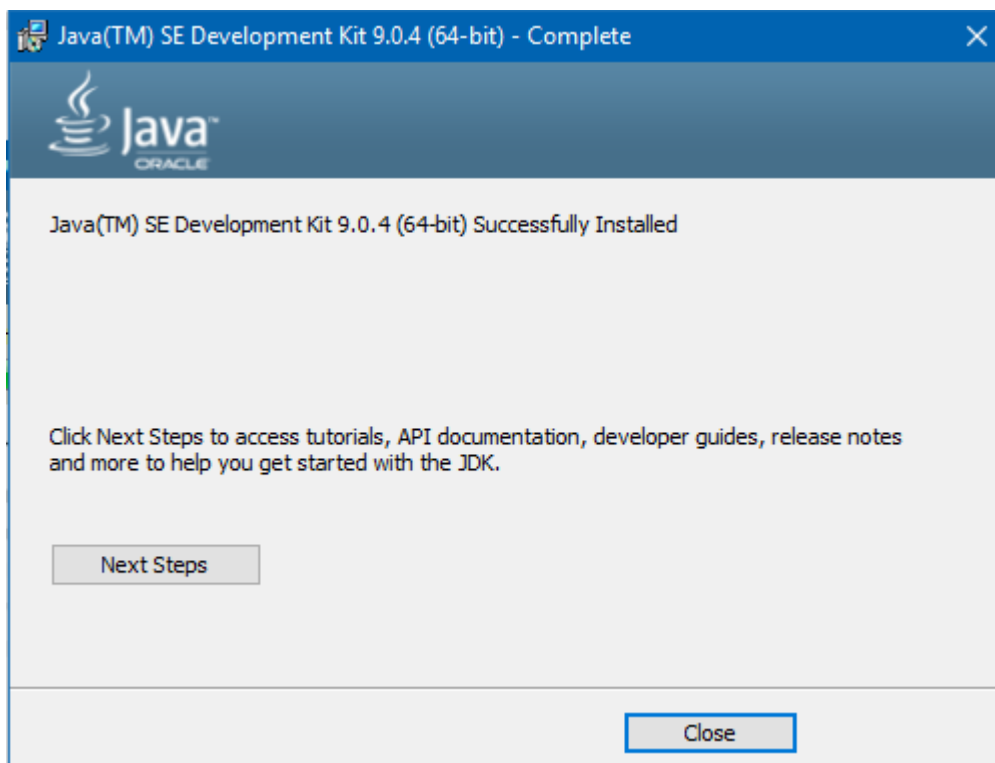
5.- Una vez completada esa descarga, aparecerá una ventana la cual preguntará en qué lugar se quiere guardar los archivos de java, en este caso se guarda en el disco C en la carpeta "Program Files".



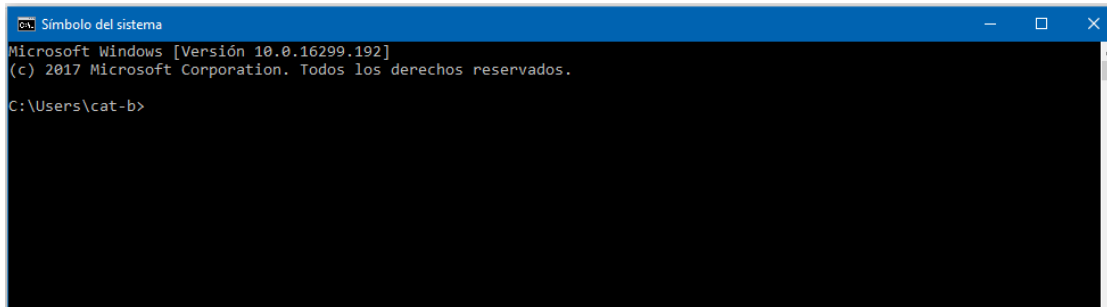
6.- Una vez seleccionado la carpeta iniciará la descarga completa de java.



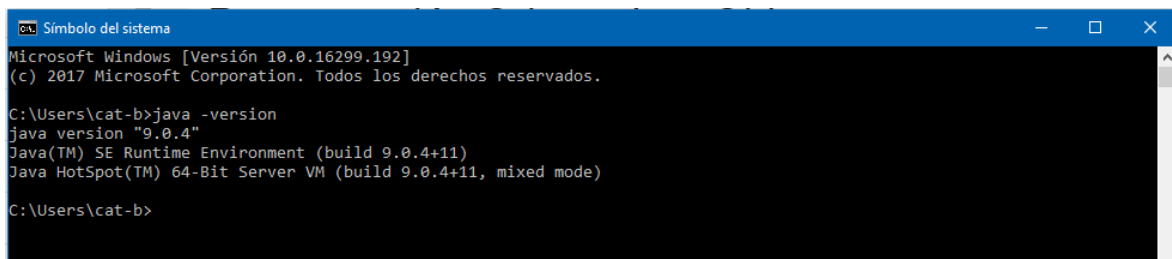
7.-Al terminar la descarga, nos muestra una ventana la cual da por concluida la instalación de Java.



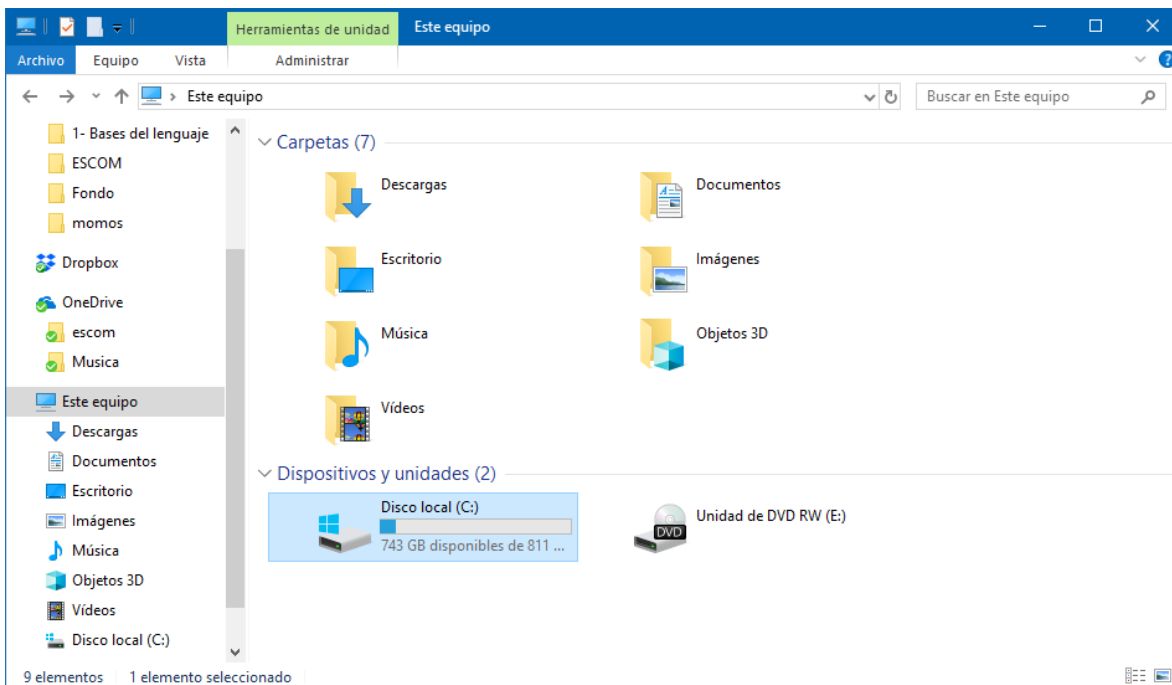
8.- Para comprobar que se descargo correctamente, se busca desde Windows “cmd” y aparecerá la consola de “Símbolo de sistema”.



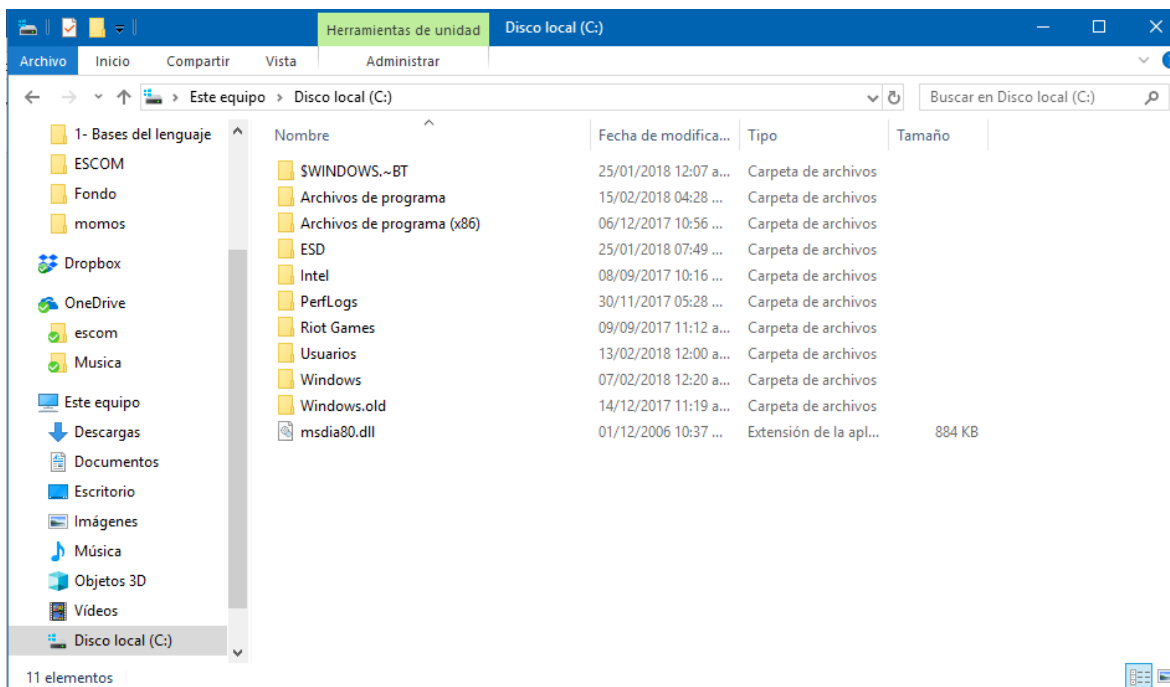
9.- Se escribirá “java -version” para comprobar la versión de java



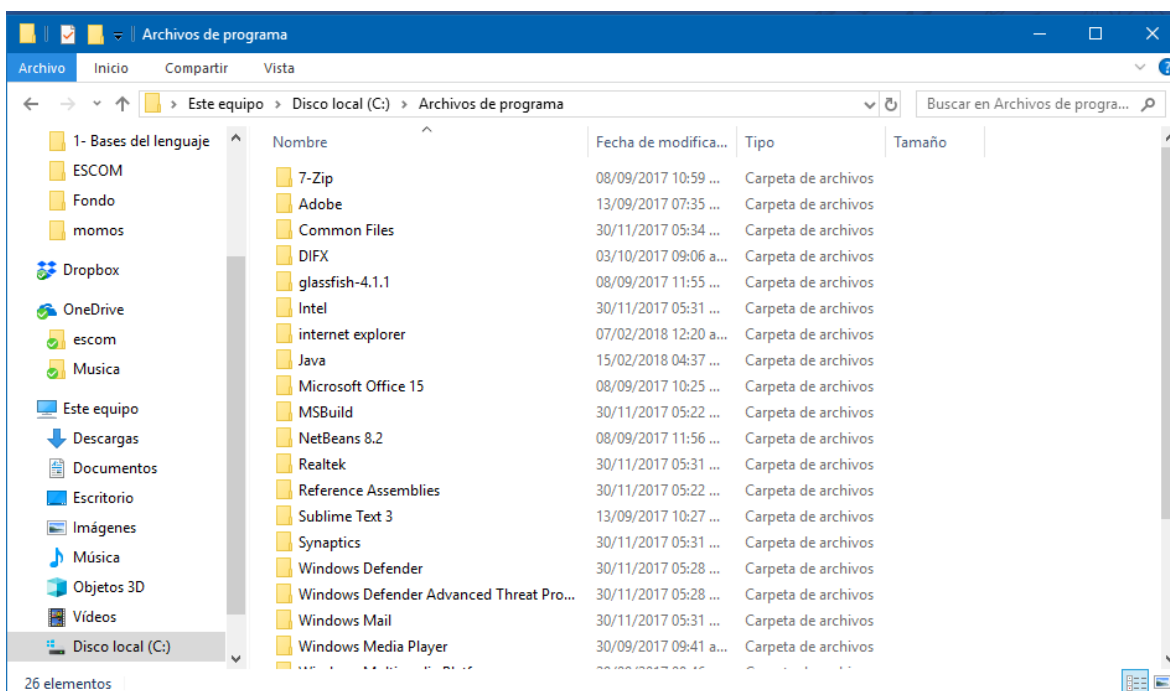
10.-Ahora, para instalar correctamente el JDK desde el explorador de archivos se buscará el lugar donde se guardo la carpeta de java, en este caso fue en el disco local “C”.



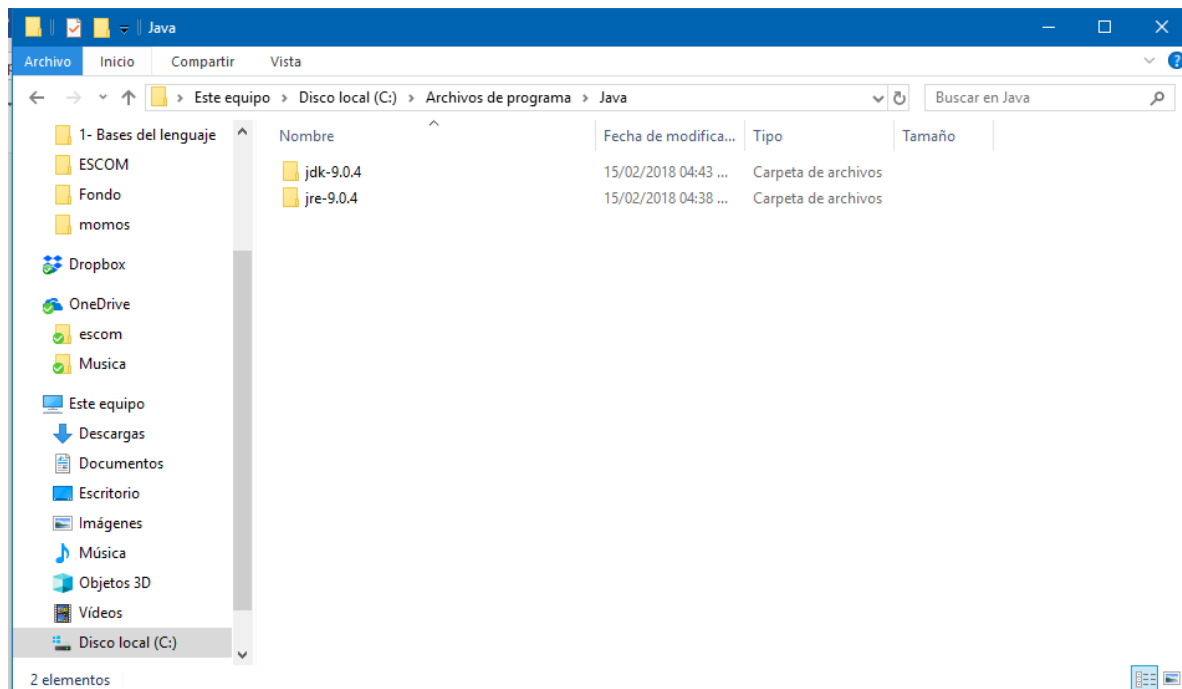
11.- Una vez dentro de la carpeta del disco local, se abre la carpeta de “Archivos de programa”



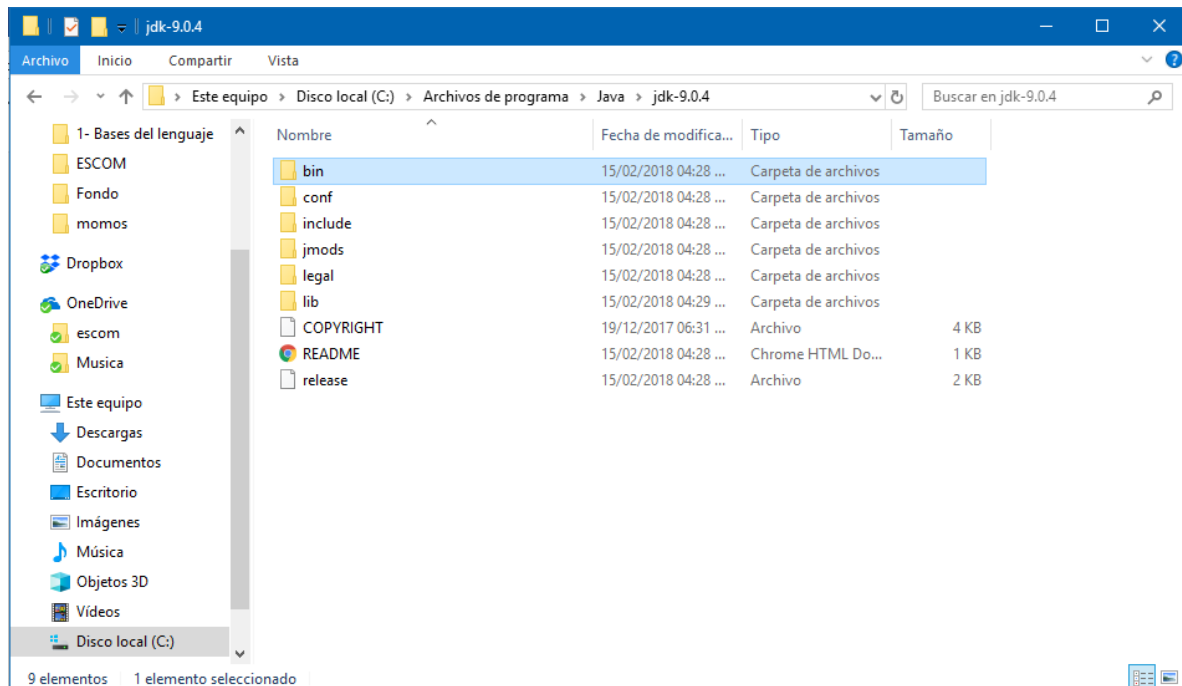
12.-Una vez abierta la carpeta se busca la carpeta de “Java” y se abre.



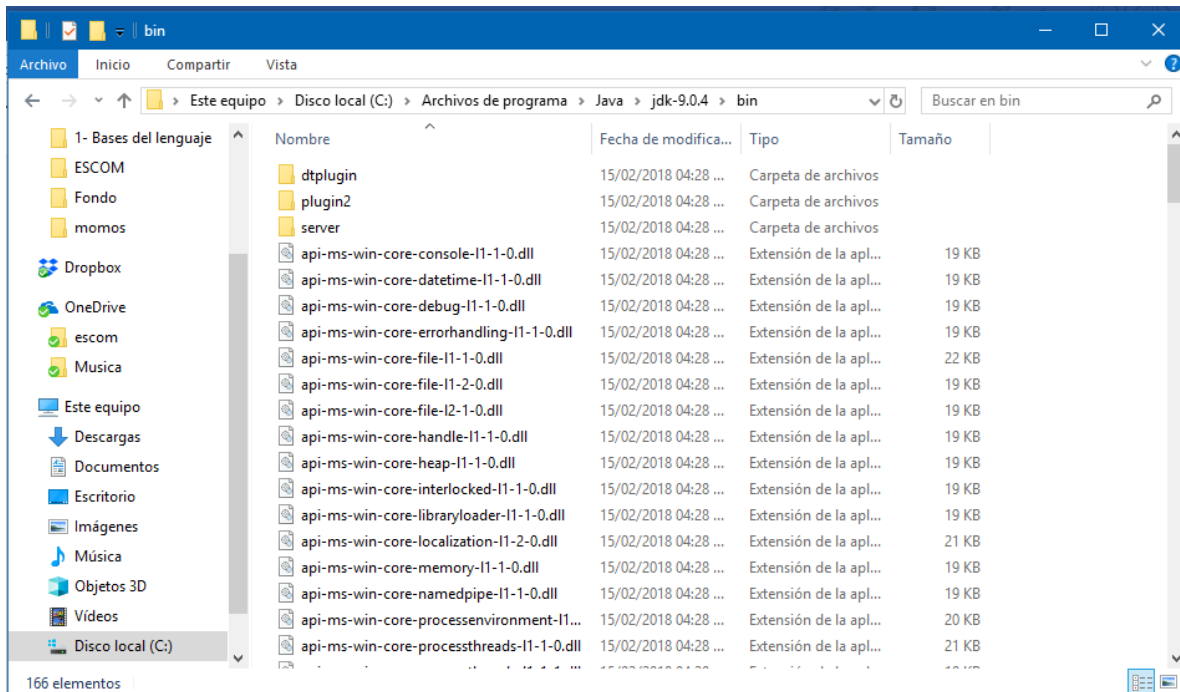
13.- Una vez abierta se selecciona la carpeta del “jdk-9.0.4”



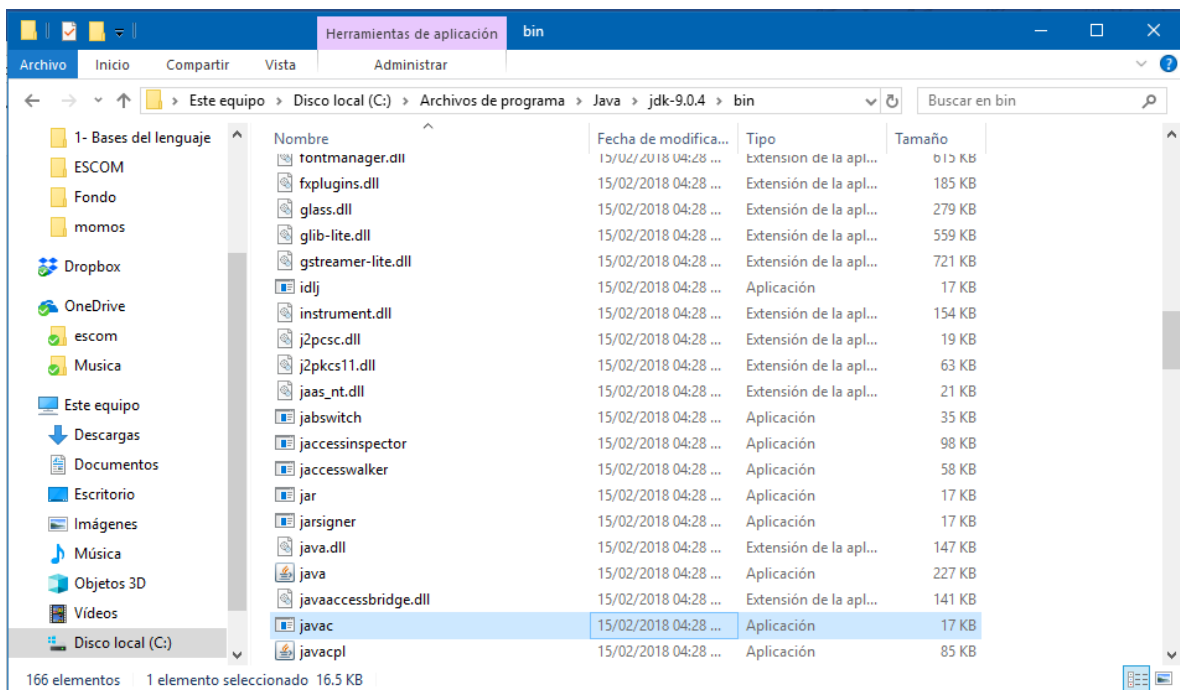
14.- Y se abre la carpeta de “bin”



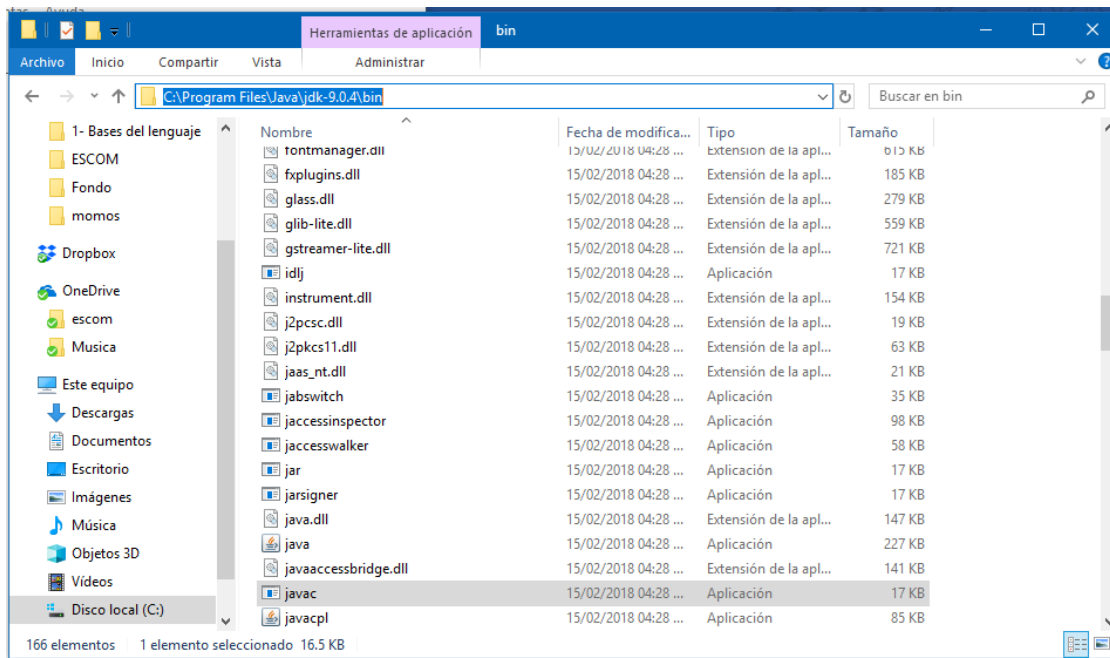
15.- Una vez ahí, habrá muchos archivos, se deberá buscar el que diga "javac"



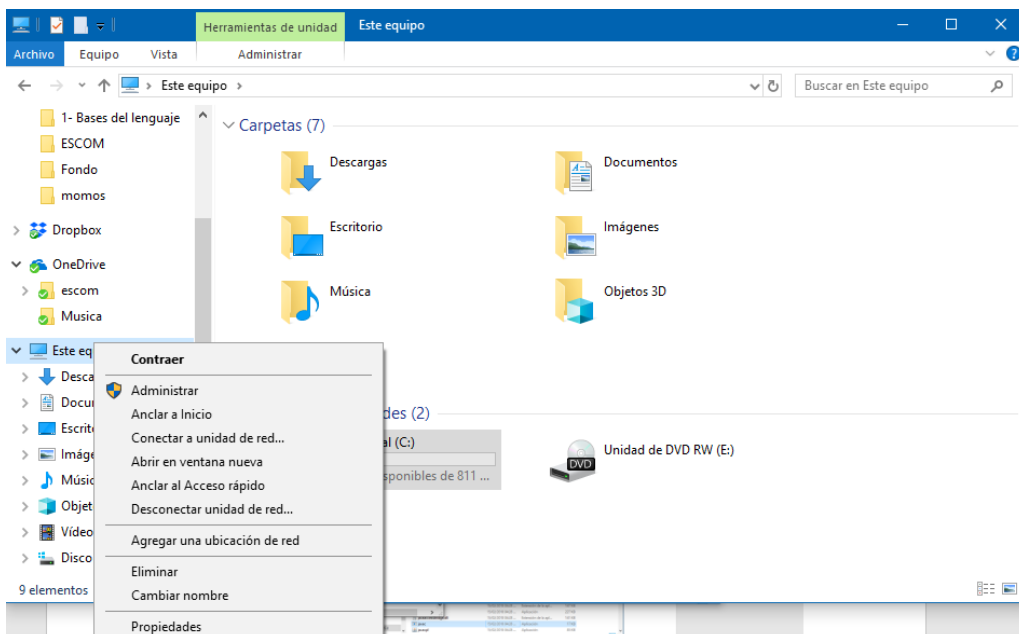
16.- Una vez encontrada se selecciona.



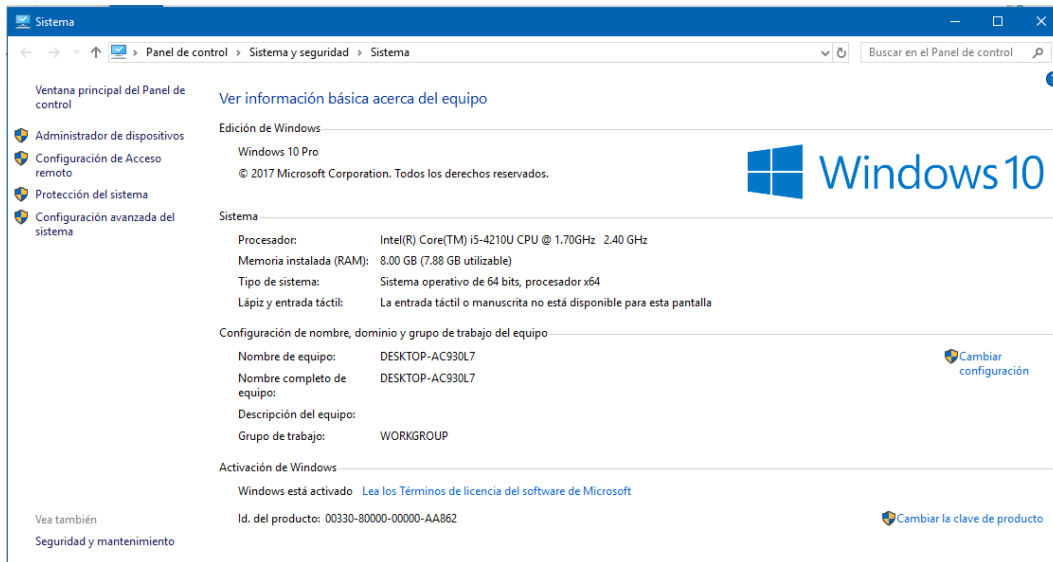
17.- En la barra donde aparece las carpetas que se han ido seleccionando se le da click izquierdo y se copia esa dirección.



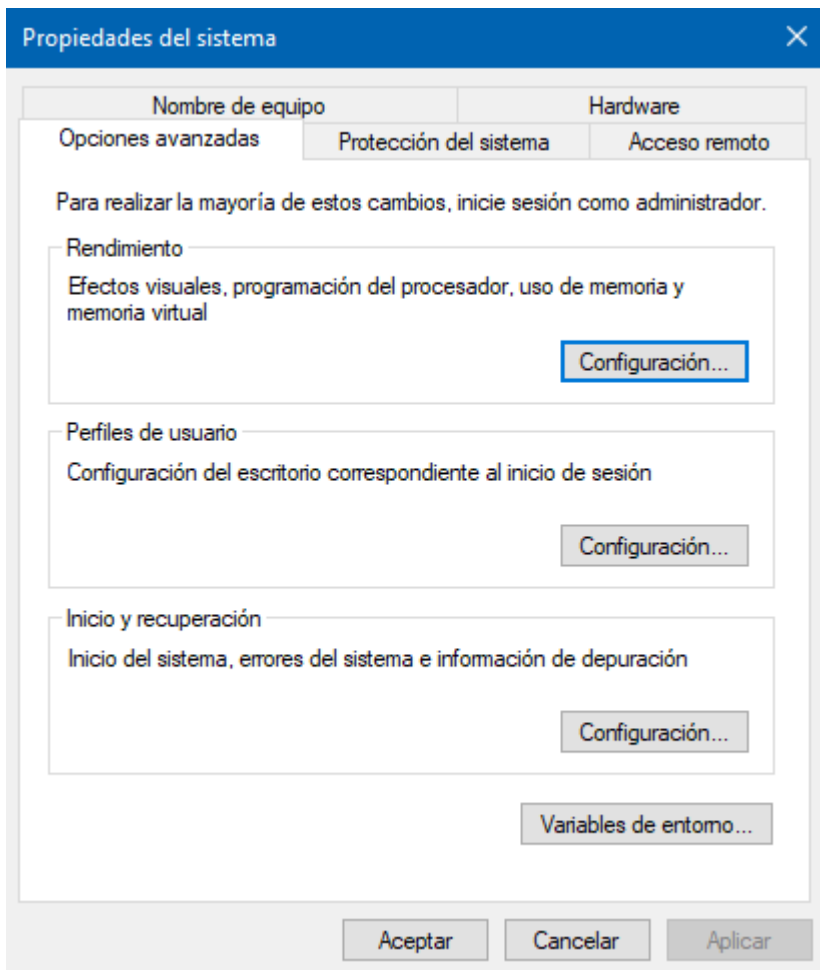
18.- En la parte derecha del explorador de archivos está el icono de “Este equipo” el cual se le deberá dar click izquierdo y aparecerá una pequeña ventana, se selecciona “propiedades”.



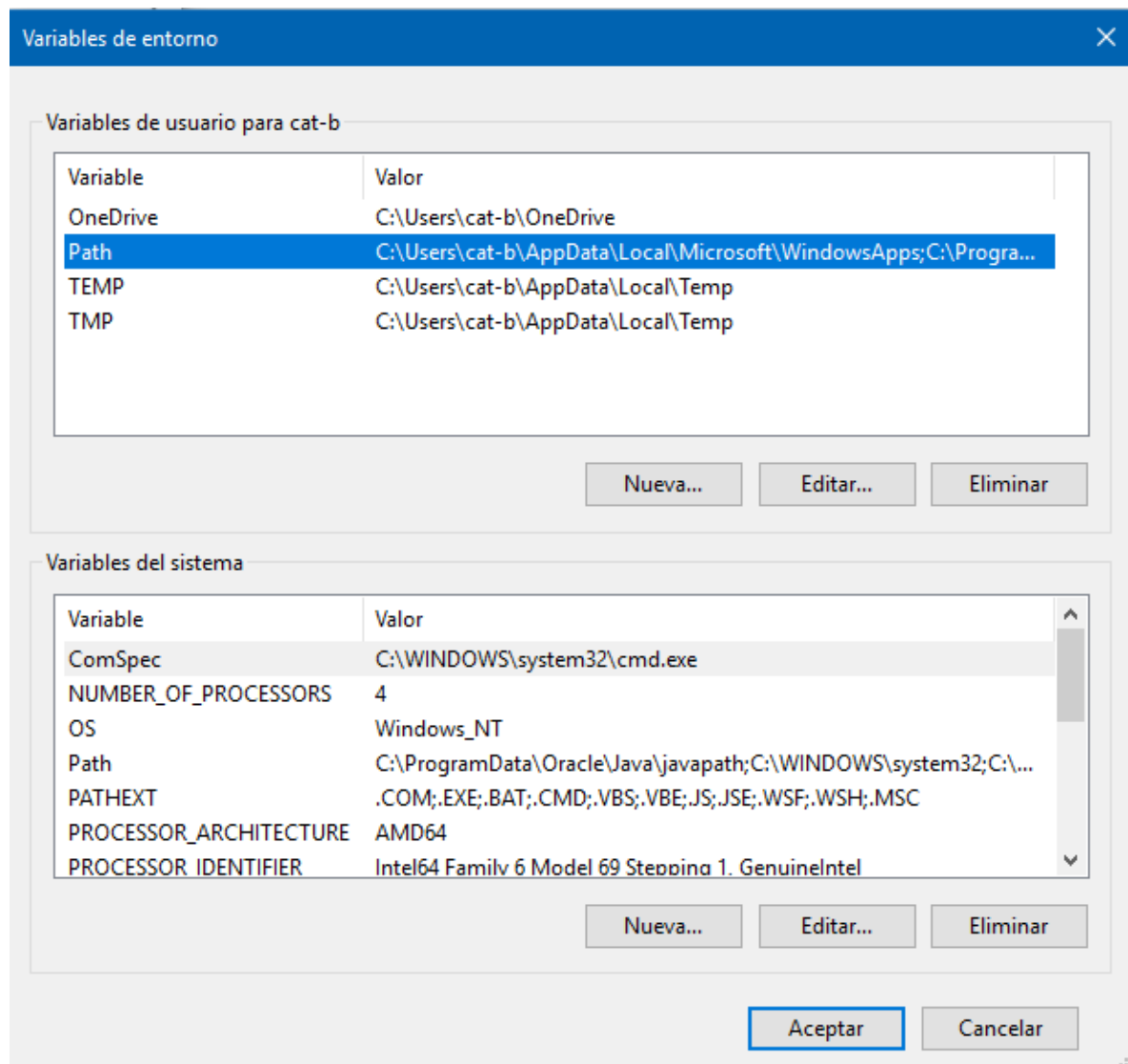
19.- Aparecerá una ventana la cual contiene información de la computadora. Se selecciona “Configuración avanzada del sistema”



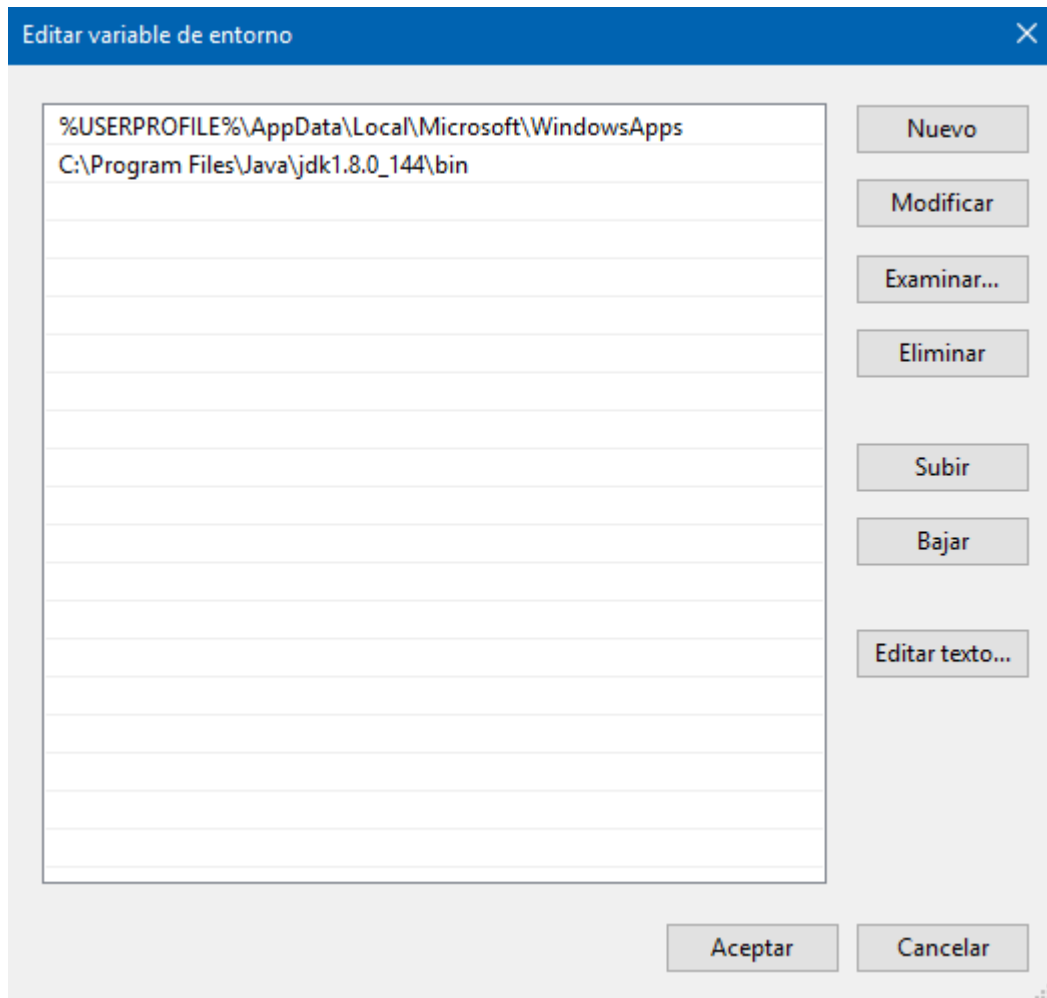
20.- Aparecerá una ventana como la siguiente y se selecciona “Variables de entorno”.



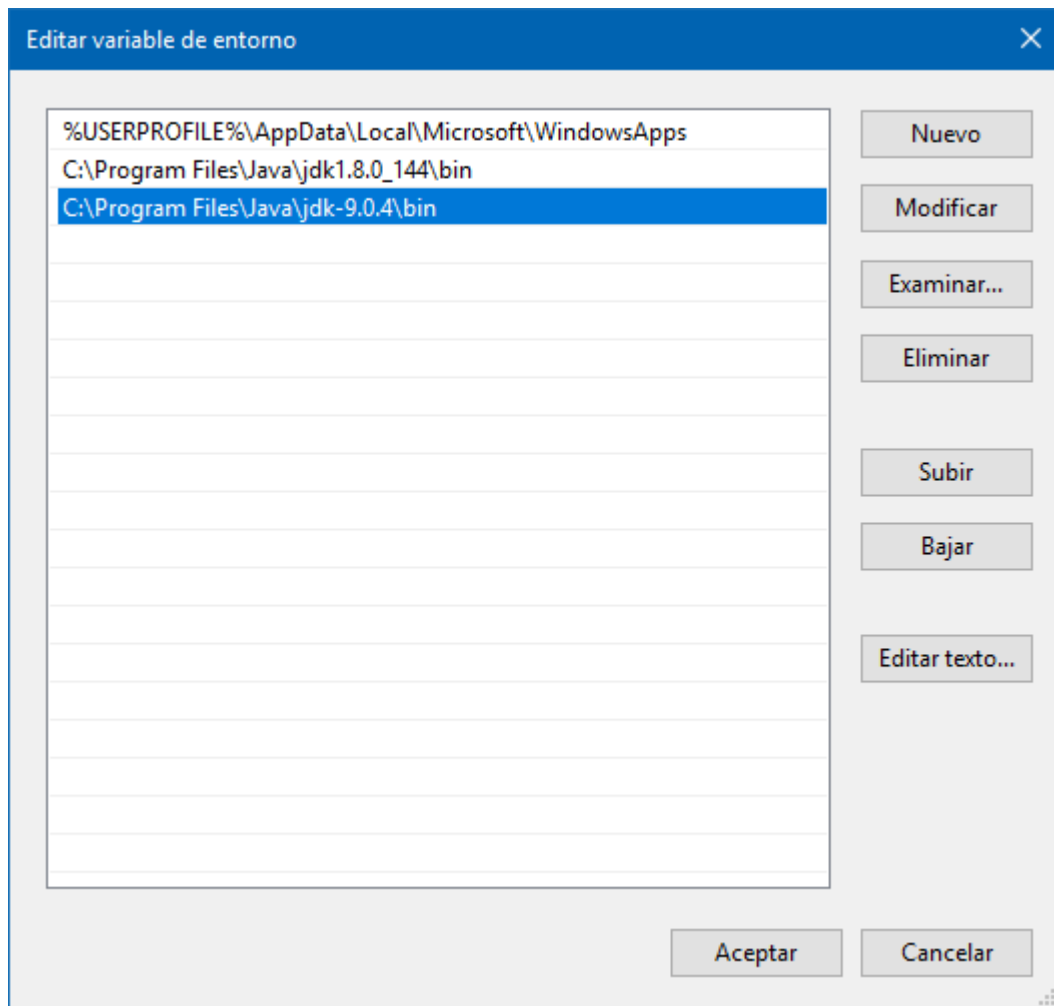
21.- Abrirá una ventana y se deberá elegir "Path". Una vez elegida se da click izquierdo en "Editar ..." PERO de la primera parte de esta ventana, es decir el que está dentro de "variables de usuario para cat-b".



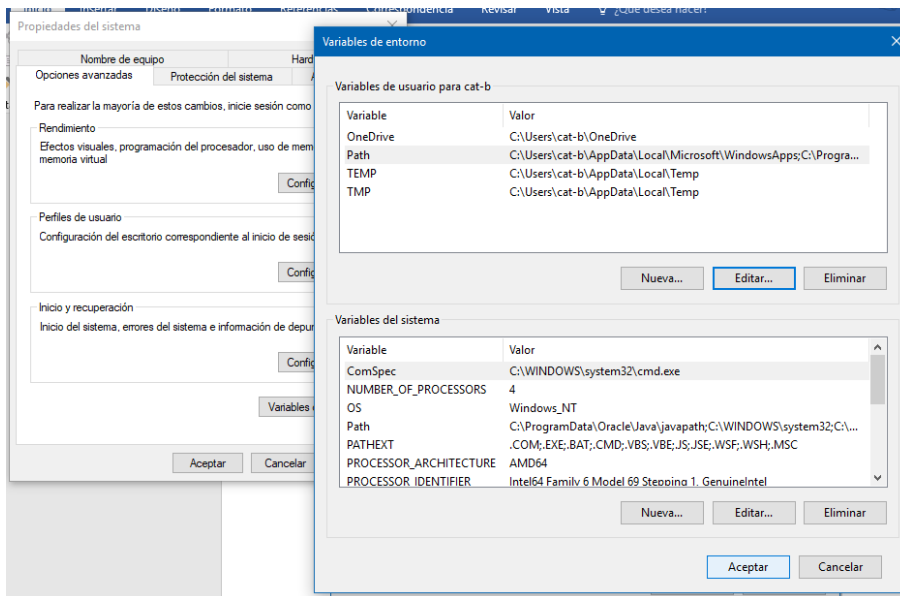
22.- Entonces se mostrará una ventana en la cual se debe pegar en alguna de las líneas que están en la variable de entorno lo que ya teníamos copiado.



23.- Una vez copiado, se selecciona "Aceptar".



24.- Desaparecerán las ventanas conforme se seleccione “Aceptar”.



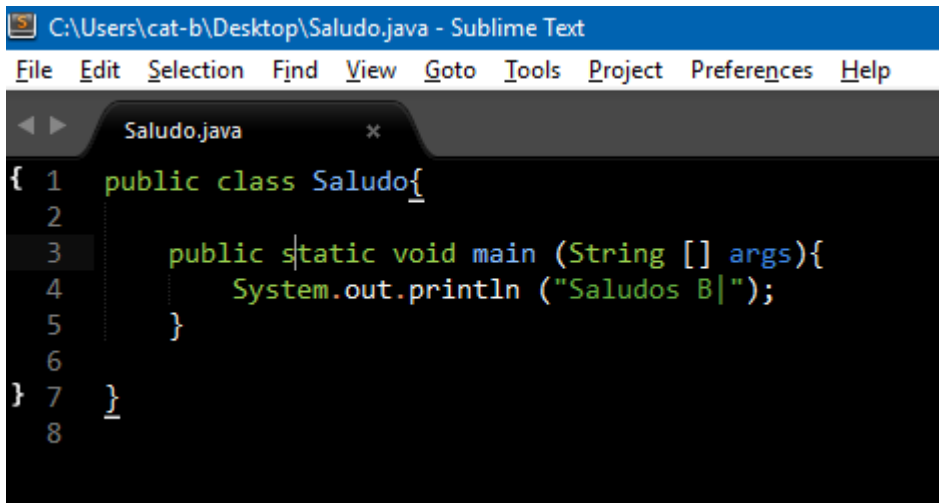
25.- Una vez hecho lo anterior se abrirá la consola del Símbolo del sistema, para saber si funciona el jdk, se escribe “javac” y deberá aparecer lo siguiente:

```

C:\Users\cat-b>javac
Microsoft Windows [Versión 10.0.16299.192]
(c) 2017 Microsoft Corporation. Todos los derechos reservados.

C:\Users\cat-b>javac
Usage: javac <options> <source files>
where possible options include:
  @<filename>           Read options and filenames from file
  -Akey[=value]         Options to pass to annotation processors
  --add-modules <module>(<module>)*
                        Root modules to resolve in addition to the initial modules, or all modules
                        on the module path if <module> is ALL-MODULE-PATH.
  --boot-class-path <path>, -bootclasspath <path>
                        Override location of bootstrap class files
  --class-path <path>, -classpath <path>, -cp <path>
                        Specify where to find user class files and annotation processors
  -d <directory>        Specify where to place generated class files
  -deprecation
                        Output source locations where deprecated APIs are used
  -encoding <encoding>  Specify character encoding used by source files
  -endorseddirs <dirs>  Override location of endorsed standards path
  -extdirs <dirs>       Override location of installed extensions
  -g                    Generate all debugging info
  -g:{lines,vars,source}
                        Generate only some debugging info
  -g:none               Generate no debugging info
  -h <directory>        Specify where to place generated native header files
  --help, -help         Print this help message
  --help-extra, -X       Print help on extra options
  -implicit:{none,class}
                        Specify whether or not to generate class files for implicitly referenced files
  -J<flag>               Pass <flag> directly to the runtime system
  --limit-modules <module>(<module>)*
                        Limit the universe of observable modules
  --module <module-name>, -m <module-name>
                        Compile only the specified module, check timestamps
  --module-path <path>, -p <path>
                        Specify where to find application modules
  --module-source-path <module-source-path>
                        Specify where to find input source files for multiple modules
  --module-version <version>
                        Specify version of modules that are being compiled
  -nowarn                Generate no warnings
  -parameters            Generate metadata for reflection on method parameters
  
```

26.- Con un pequeño programa de saludo se comprobará que se instaló correctamente. Se guardará el archivo con “.java” en un lugar que se reconozca fácilmente, en este caso en el escritorio con el nombre de “Saludo.java”, la clase se debe escribir como se guardo el archivo, de lo contrario aparecerán errores.

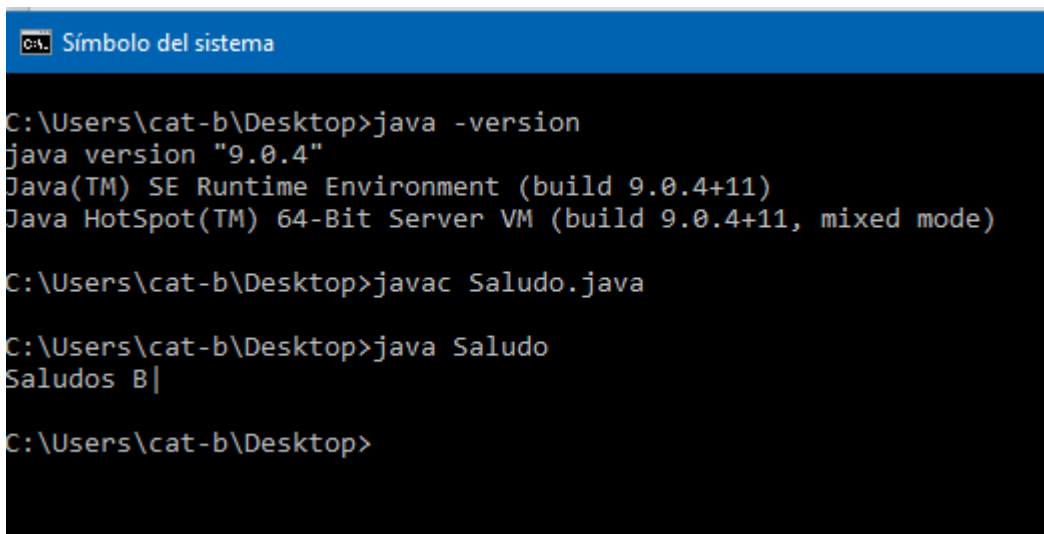


```
C:\Users\cat-b\Desktop\Saludo.java - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help

Saludo.java x
{ 1 public class Saludo{
  2
  3     public static void main (String [] args){
  4         System.out.println ("Saludos B|");
  5     }
  6
  7 }
  8 }
```

27.- En el símbolo del sistema para acceder al escritorio donde se guardo el programa, se escribe “cd Desktop”, después “javac Saludo.java” Esto es para compilar el programa y comprobar que esté bien. Si no hay errores se mostrará “C:\Users\cat-b\Desktop>” y seguido de esto se escribe “java Saludo” esto es para ejecutar y comprobar que funciona correctamente.

Se mostrará el saludo que se escribió y así se sabe que funciona bien.



```
C:\> Símbolo del sistema

C:\Users\cat-b\Desktop>java -version
java version "9.0.4"
Java(TM) SE Runtime Environment (build 9.0.4+11)
Java HotSpot(TM) 64-Bit Server VM (build 9.0.4+11, mixed mode)

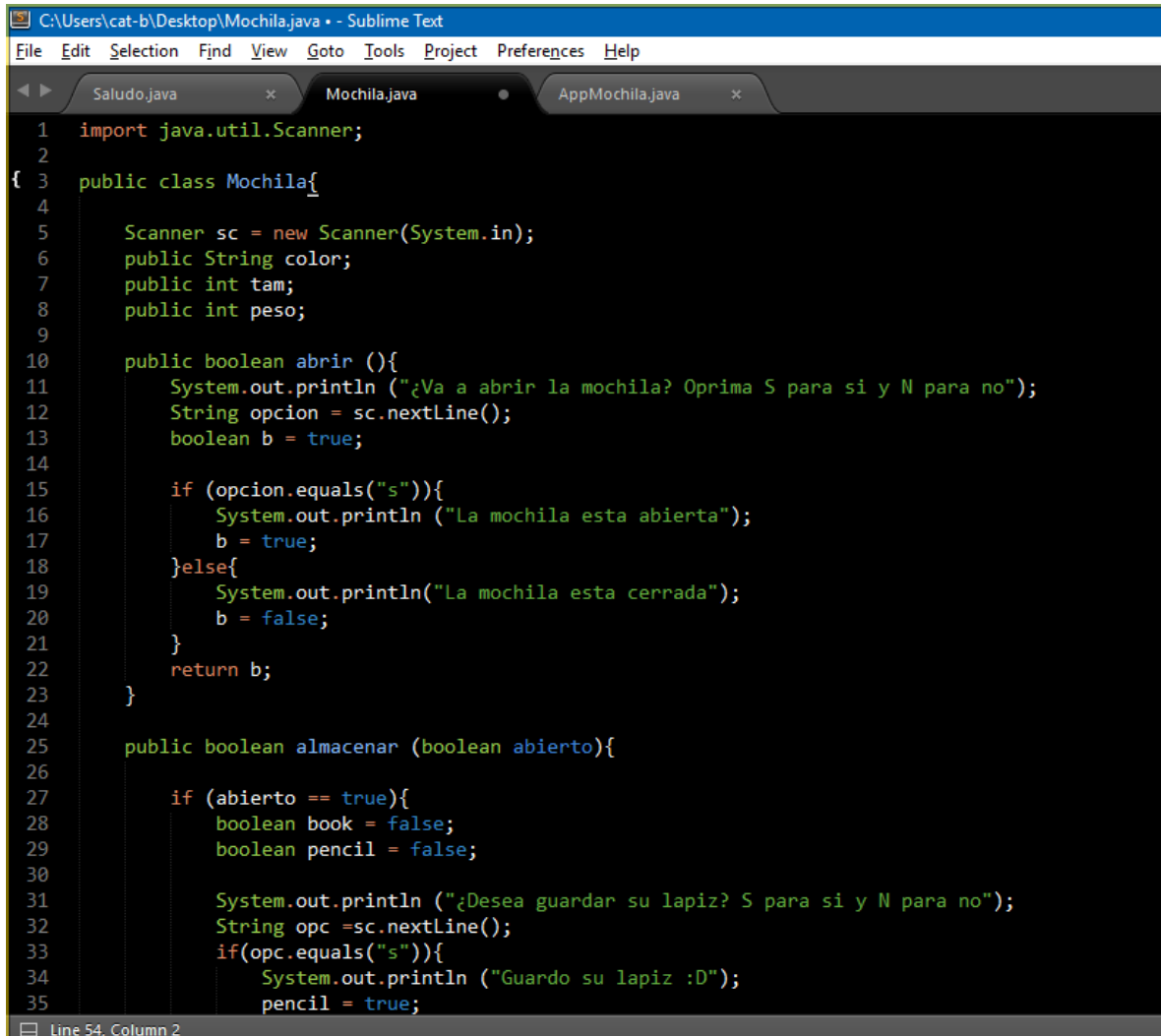
C:\Users\cat-b\Desktop>javac Saludo.java

C:\Users\cat-b\Desktop>java Saludo
Saludos B|

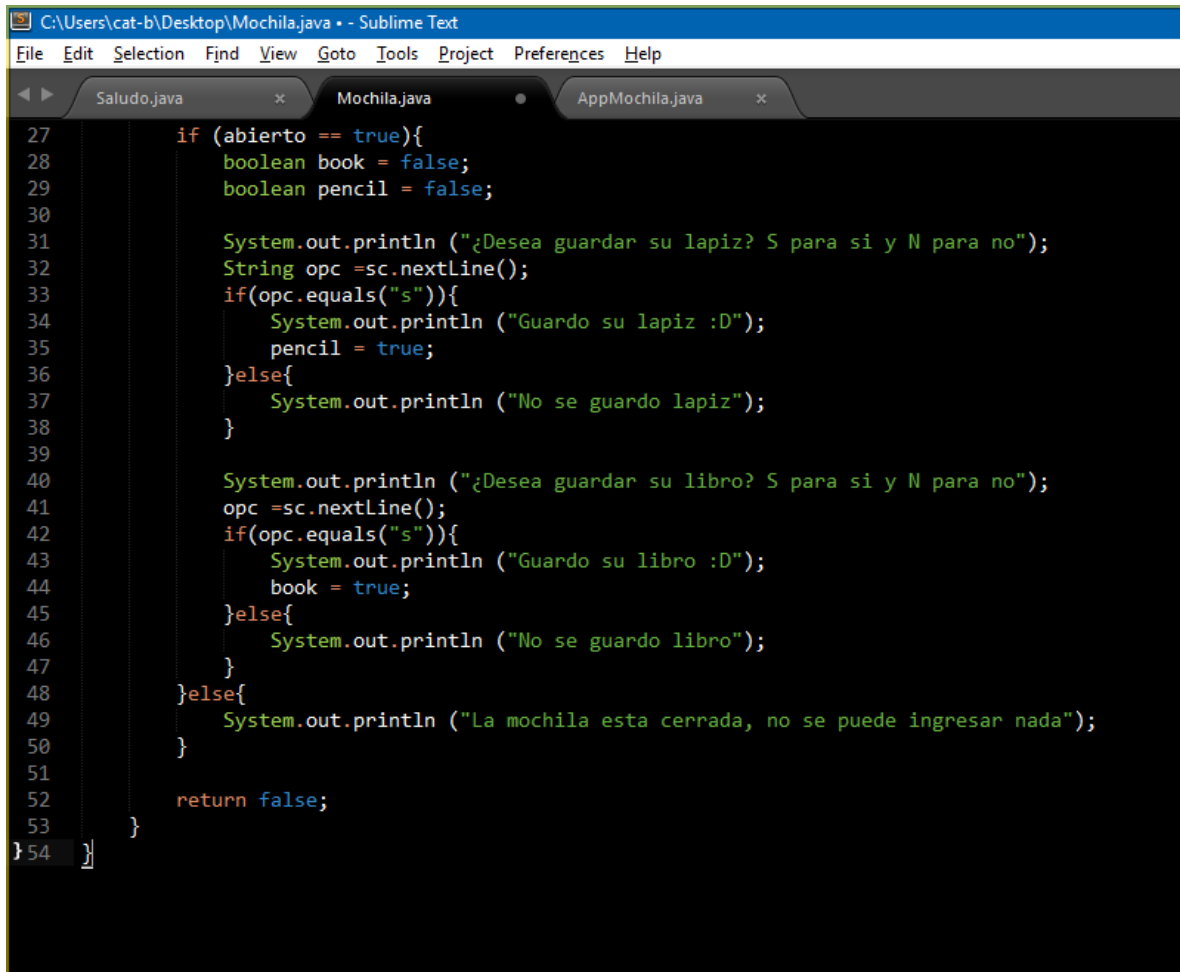
C:\Users\cat-b\Desktop>
```

28.- Ahora con el programa de “Mochila” Se hará una prueba de la interacción entre una clase principal y la clase que contiene los atributos y métodos del objeto.

Las siguientes dos imágenes representan la clase que contiene los atributos y métodos del objeto “Mochila”.



```
C:\Users\cat-b\Desktop\Mochila.java - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
Saludo.java x Mochila.java AppMochila.java x
1 import java.util.Scanner;
2
3 public class Mochila{
4
5     Scanner sc = new Scanner(System.in);
6     public String color;
7     public int tam;
8     public int peso;
9
10    public boolean abrir (){
11        System.out.println ("¿Va a abrir la mochila? Oprima S para si y N para no");
12        String opcion = sc.nextLine();
13        boolean b = true;
14
15        if (opcion.equals("s")){
16            System.out.println ("La mochila esta abierta");
17            b = true;
18        }else{
19            System.out.println("La mochila esta cerrada");
20            b = false;
21        }
22        return b;
23    }
24
25    public boolean almacenar (boolean abierto){
26
27        if (abierto == true){
28            boolean book = false;
29            boolean pencil = false;
30
31            System.out.println ("¿Desea guardar su lapiz? S para si y N para no");
32            String opc =sc.nextLine();
33            if(opc.equals("s")){
34                System.out.println ("Guardo su lapiz :D");
35                pencil = true;
```

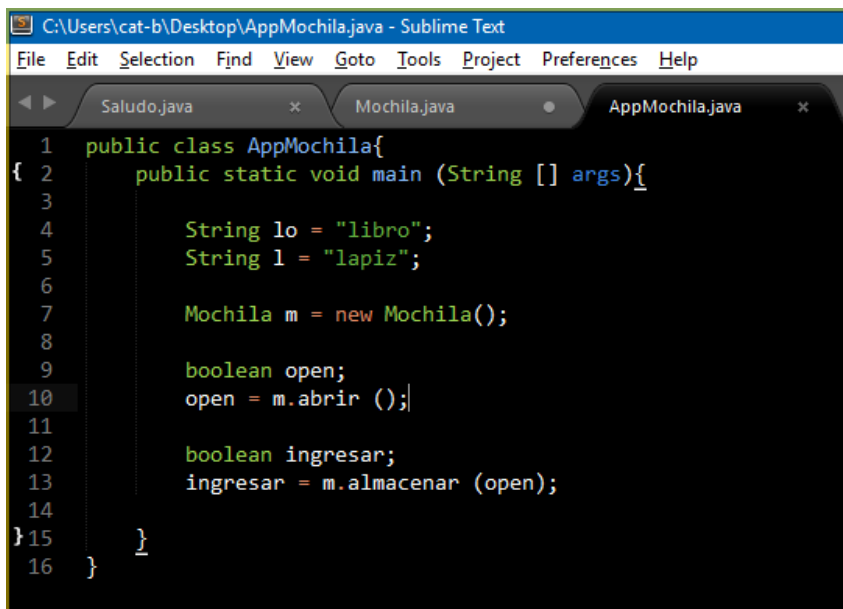



```
C:\Users\cat-b\Desktop\Mochila.java - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help

Saludo.java x Mochila.java AppMochila.java x

27     if (abierto == true){
28         boolean book = false;
29         boolean pencil = false;
30
31         System.out.println ("¿Desea guardar su lapiz? S para si y N para no");
32         String opc =sc.nextLine();
33         if(opc.equals("s")){
34             System.out.println ("Guardo su lapiz :D");
35             pencil = true;
36         }else{
37             System.out.println ("No se guardo lapiz");
38         }
39
40         System.out.println ("¿Desea guardar su libro? S para si y N para no");
41         opc =sc.nextLine();
42         if(opc.equals("s")){
43             System.out.println ("Guardo su libro :D");
44             book = true;
45         }else{
46             System.out.println ("No se guardo libro");
47         }
48     }else{
49         System.out.println ("La mochila esta cerrada, no se puede ingresar nada");
50     }
51
52     return false;
53 }
54 }
```

29.- La siguiente imagen muestra la clase principal y cómo se instancia un objeto de la clase "Mochila" en otra clase.

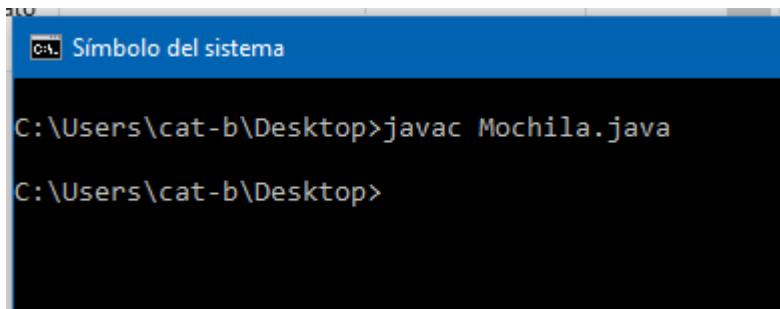


```
C:\Users\cat-b\Desktop\AppMochila.java - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help

Saludo.java x Mochila.java AppMochila.java x

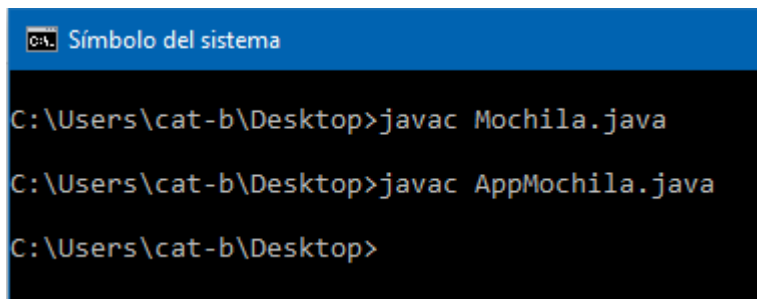
1 public class AppMochila{
2     public static void main (String [] args){
3
4         String lo = "libro";
5         String l = "lapiz";
6
7         Mochila m = new Mochila();
8
9         boolean open;
10        open = m.abrir ();
11
12        boolean ingresar;
13        ingresar = m.almacenar (open);
14
15    }
16 }
```

30.- Dese el símbolo del sistema se compila primero "Mochila.java".



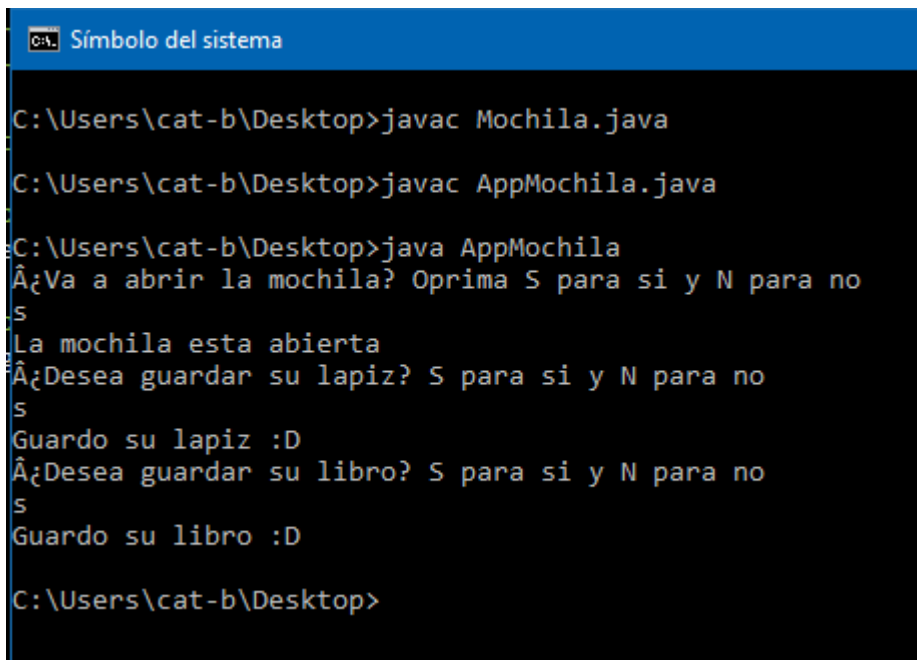
```
C:\Users\cat-b\Desktop>javac Mochila.java  
C:\Users\cat-b\Desktop>
```

31.- Si no hay errores se compila la clase principal, en este caso "AppMochila.java".



```
C:\Users\cat-b\Desktop>javac Mochila.java  
C:\Users\cat-b\Desktop>javac AppMochila.java  
C:\Users\cat-b\Desktop>
```

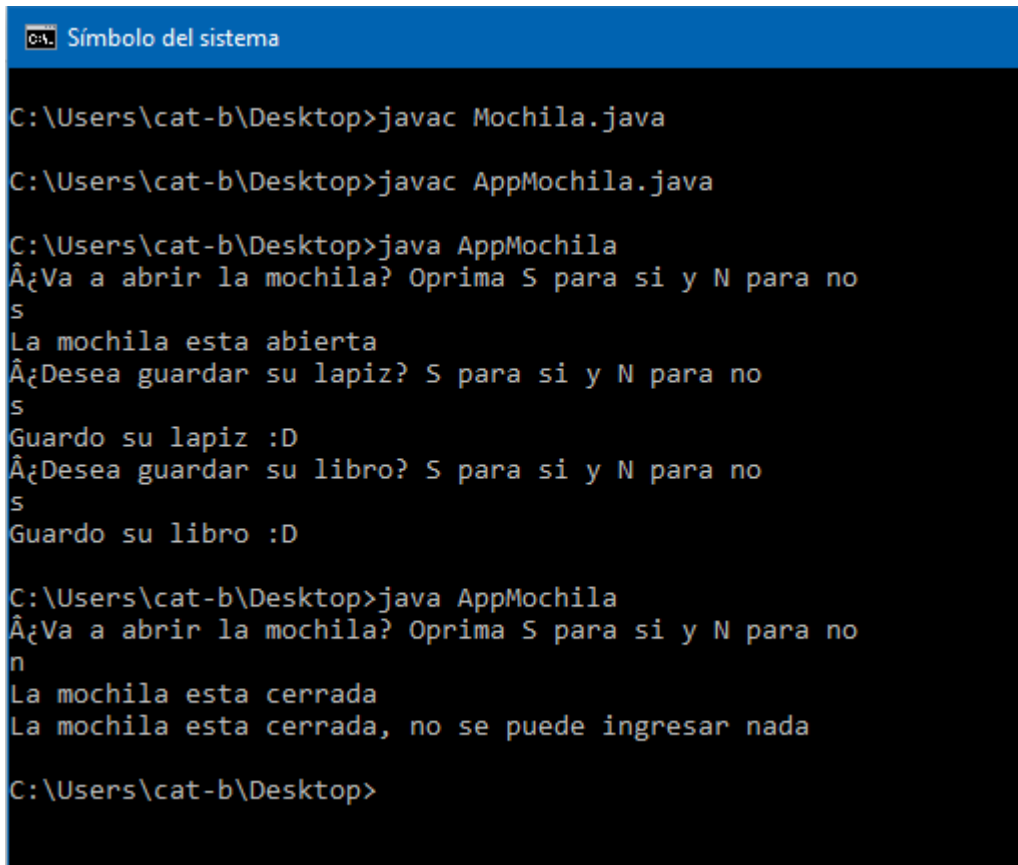
32.- Ahora solo se ejecuta la clase principal, ya que contiene el método principal "main".



```
C:\Users\cat-b\Desktop>javac Mochila.java  
C:\Users\cat-b\Desktop>javac AppMochila.java  
C:\Users\cat-b\Desktop>java AppMochila  
¿Va a abrir la mochila? Oprima S para si y N para no  
s  
La mochila esta abierta  
¿Desea guardar su lapiz? S para si y N para no  
s  
Guardo su lapiz :D  
¿Desea guardar su libro? S para si y N para no  
s  
Guardo su libro :D  
C:\Users\cat-b\Desktop>
```

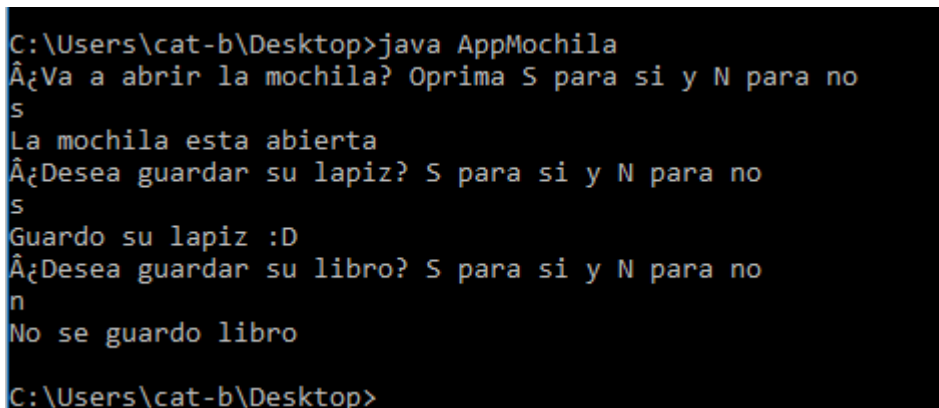
Se muestra los resultados diciéndole "si" a todo y funciona correctamente.

33.- En esta captura de pantalla se muestra el resultado diciéndole “no” a la pregunta y con esto ya no sigue la segunda pregunta que se había hecho, ya que no puedes ingresar nada a una mochila estando cerrada.



```
C:\Users\cat-b\Desktop>javac Mochila.java
C:\Users\cat-b\Desktop>javac AppMochila.java
C:\Users\cat-b\Desktop>java AppMochila
¿Va a abrir la mochila? Oprima S para si y N para no
S
La mochila esta abierta
¿Desea guardar su lapiz? S para si y N para no
S
Guardo su lapiz :D
¿Desea guardar su libro? S para si y N para no
S
Guardo su libro :D
C:\Users\cat-b\Desktop>java AppMochila
¿Va a abrir la mochila? Oprima S para si y N para no
N
La mochila esta cerrada
La mochila esta cerrada, no se puede ingresar nada
C:\Users\cat-b\Desktop>
```

34.- En esta prueba se muestra las diferentes respuestas después de elegir la opción que ingresa el usuario.



```
C:\Users\cat-b\Desktop>java AppMochila
¿Va a abrir la mochila? Oprima S para si y N para no
S
La mochila esta abierta
¿Desea guardar su lapiz? S para si y N para no
S
Guardo su lapiz :D
¿Desea guardar su libro? S para si y N para no
N
No se guardo libro
C:\Users\cat-b\Desktop>
```

Código Fuente

```
import java.util.Scanner;
```

```
public class Mochila{
```

```
    Scanner sc = new Scanner(System.in);
```

```
    public String color;
```

```
    public int tam;
```

```
    public int peso;
```

```
    public boolean abrir (){
```

```
        System.out.println ("¿Va a abrir la mochila? Oprima S para si y N para no");
```

```
        String opcion = sc.nextLine();
```

```
        boolean b = true;
```

```
        if (opcion.equals("s")){
```

```
            System.out.println ("La mochila esta abierta");
```

```
            b = true;
```

```
        }else{
```

```
            System.out.println("La mochila esta cerrada");
```

```
            b = false;
```

```
        }
```

```
        return b;
```

```
    }
```

```
    public boolean almacenar (boolean abierto){
```

```
        if (abierto == true){
```

```
            boolean book = false;
```

```
            boolean pencil = false;
```

```
            System.out.println ("¿Desea guardar su lapiz? S para si y N para no");
```

```
            String opc =sc.nextLine();
```

```
        if(opc.equals("s")){
            System.out.println ("Guardo su lapiz :D");
            pencil = true;
        }else{
            System.out.println ("No se guardo lapiz");
        }

        System.out.println ("¿Desea guardar su libro? S para si y N para no");
        opc =sc.nextLine();
        if(opc.equals("s")){
            System.out.println ("Guardo su libro :D");
            book = true;
        }else{
            System.out.println ("No se guardo libro");
        }
    }else{
        System.out.println ("La mochila esta cerrada, no se puede ingresar nada");
    }

    return false;
}
}
```

Código fuente de clase principal

```
public class AppMochila{
    public static void main (String [] args){
        String lo = "libro";
        String l = "lapiz";
        Mochila m = new Mochila();
        boolean open;
        open = m.abrir ();
        boolean ingresar;
        ingresar = m.almacenar (open);
    }
}
```