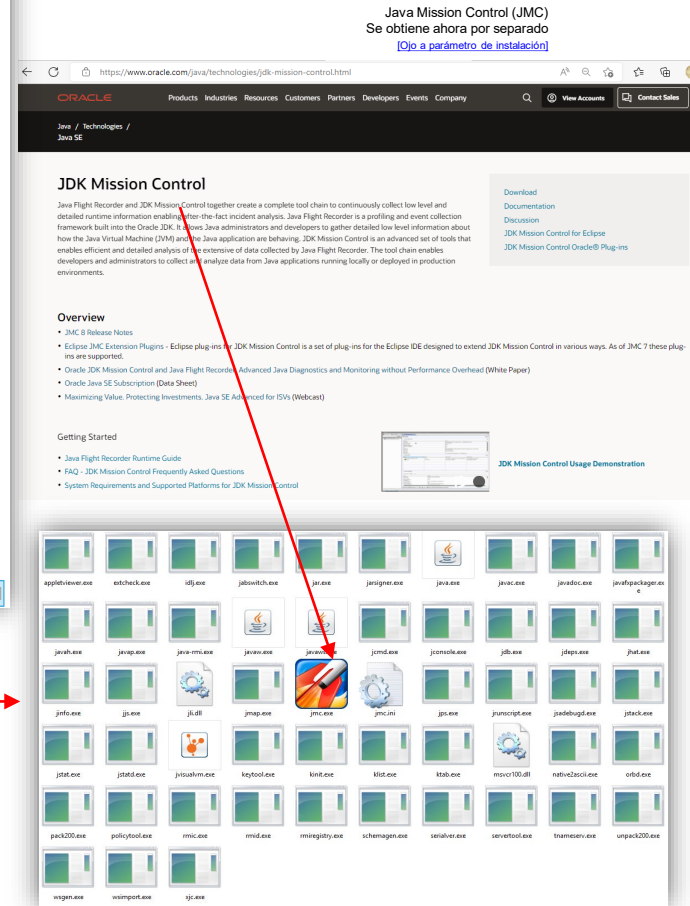
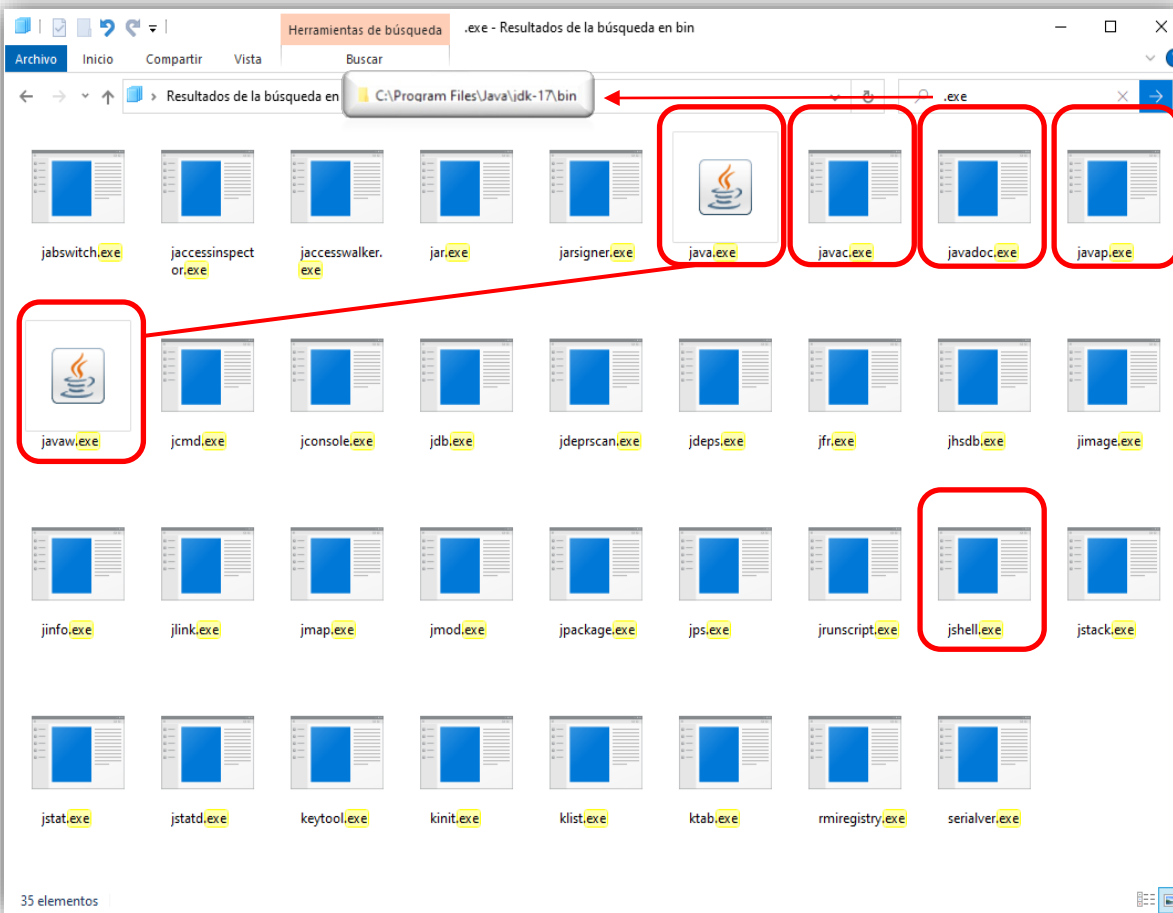


**¿QUÉ TENDRÉ A MI DISPOSICIÓN
PARA DESARROLLAR EN JAVA?**

(COMPILACIÓN, EJECUCIÓN, ETC.)

Las herramientas de ejecución y desarrollo de Java

El directorio "/bin"



En una versión anterior...

En distintas versiones van cambiando herramientas, pero en todo caso pueden obtenerse las que no se encuentran por defecto. Las imprescindibles las encontramos siempre.

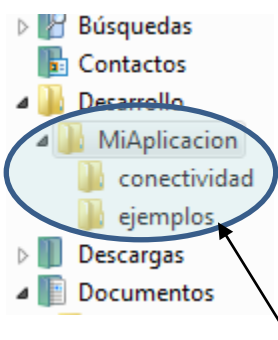
Compilación y ejecución

```
D:\>java
Usage: java [options] <mainclass> [args...]
       (to execute a class)
or java [options] -jar <jarfile> [args...]
       (to execute a jar file)
or java [options] -m <module>[/<mainclass>] [args...]
       java [options] --module <module>[/<mainclass>] [args...]
       (to execute the main class in a module)
or java [options] <sourcefile> [args]
       (to execute a single source-file program)

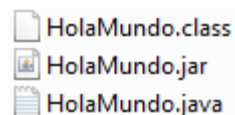
Arguments following the main class, source file, -jar <jarfile>,
-m or --module <module>/<mainclass> are passed as the arguments to
main class.

where options include:

-cp <class search path of directories and zip/jar files>
-classpath <class search path of directories and zip/jar files>
--class-path <class search path of directories and zip/jar files>
      A ; separated list of directories, JAR archives,
      and ZIP archives to search for class files.
-p <module path>
--module-path <module path>...
      A ; separated list of directories, each directory
      is a directory of modules.
--upgrade-module-path <module path>
```



Un programa Java comienza por la ejecución de un fichero (una clase) que puede ir invocando la ejecución de otras dentro de uno o varios subárboles de directorios o ficheros JAR (cada subárbol o cada JAR es una “biblioteca de clases”)



```
D:\>java -version
java version "17" 2021-09-14 LTS
Java(TM) SE Runtime Environment (build 17+35-LTS-2724)
Java HotSpot(TM) 64-Bit Server VM (build 17+35-LTS-2724, mixed mode, sharing)
```

```
D:\>javac -version
javac 17
```

```
D:\>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
--enable-preview     Enable preview language features. To be used in conjunction with either -source or --release.
-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
```

Compilar y ejecutar estando en “MiAplicacion”

```
>javac ejemplos/HolaMundo.java
>java ejemplos.HolaMundo
```

Compilar y ejecutar estando en otro directorio

```
>javac -cp c:/Desarrollo/MiAplicacion ejemplos/HolaMundo.java
>java -cp c:/Desarrollo/MiAplicacion ejemplos.HolaMundo
```

Ejecutar mediante un JAR

```
>java -cp c:/Desarrollo/MiAplicacion/ejemplos/HolaMundo.jar HolaMundo
>java -jar c:/Desarrollo/MiAplicacion/ejemplos/HolaMundo.jar
```

Comprobaremos todo esto... escribiendo el HolaMundo.java, compilando, ejecutando, desensamblando, decompilando (<http://www.javadecompilers.com/>)
Haremos lo mismo con Netbeans. (<http://java-decompiler.github.io/>)

**Importante: el uso de packages.

___ Lo contrario: ofuscación ___

HolaMundo.java

```
package ejemplos;

public class HolaMundo {
    public static void main(String[] args) {
        System.out.println("Hola, mundo");
    }
}
```

javac ejemplos/HolaMundo.java

HolaMundo.class

```
00000000 ca fe ba be 00 00 00 33 00 22 0a 00 02 00 03 07 .....3.....
00000010 00 04 0c 00 05 00 06 01 00 10 6a 61 76 61 2f 6c ..... java/l
00000020 61 6e 67 2f 4f 62 6a 65 63 74 01 00 06 3c 69 6e [ang/Object.<init>
00000030 69 74 3e 81 00 03 28 29 56 09 00 08 00 09 07 00 it>...(JV.....
00000040 0a 0c 00 0b 00 0c 01 00 10 6a 61 76 61 2f 6c 61 ..... java/la
00000050 6e 67 2f 53 79 73 74 65 6d 01 00 03 6f 75 74 01 ng/System...out.
00000060 00 15 4c 6a 61 76 61 2f 69 6f 2f 50 72 69 6e 74 ..Ljava/io/Print
00000070 53 74 72 65 61 6d 3b 08 00 0e 01 00 0b 48 6f 6c Stream;...Hol
00000080 61 2c 20 6d 75 6e 64 6f 0a 00 10 00 11 07 00 12 a, mundo.....
00000090 0c 00 13 00 14 01 00 13 6a 61 76 61 2f 69 6f 2f ..... java/io/
000000a0 50 72 69 6e 74 53 74 72 65 61 6d 01 00 07 70 72 PrintStream...pr
000000b0 69 6e 74 6c 6e 01 00 15 28 4c 6a 61 76 61 2f 6c intln...(Ljava/l
000000c0 61 6e 67 2f 53 74 72 69 6e 67 3b 29 56 07 00 16 ang/String;).V...
000000d0 01 00 12 65 6a 65 6d 70 6c 6f 73 2f 48 6f 6c 61 ...ejemplos/Hola
000000e0 4d 75 6e 64 6f 01 00 04 43 6f 64 65 01 00 0f 4c Mundo...Code...L
000000f0 69 6e 65 4e 75 6d 62 65 72 54 61 62 6c 65 01 00 lineNumberTable...
00000100 12 4c 6f 63 61 6c 56 61 72 69 61 62 6c 65 64 61 .LocalVariableTa
00000110 62 6c 65 01 00 04 74 68 69 73 01 00 14 4c 65 6a ble...this...Lej
00000120 65 6d 70 6c 6f 73 2f 48 6f 6c 61 4d 75 6e 64 6f emplos/HolaMundo
00000130 3b 01 00 04 6d 61 69 6e 01 00 16 28 5b 4c 6a 61 ;...main...([Lja
00000140 76 61 2f 6c 61 6e 67 2f 53 74 72 69 6e 67 3b 29 va/Lang/String;)
00000150 56 01 00 04 61 72 6f 73 01 00 13 5b 4c 6a 61 76 V...args...[Ljav
00000160 61 2f 6c 61 6e 67 2f 53 74 72 69 6e 67 3b 01 00 a/Lang/String;...
00000170 0a 53 6f 75 72 63 65 46 69 6c 65 01 00 0e 48 6f .SourceFile...Ho
00000180 6c 61 4d 75 6e 64 6f 2e 6a 61 76 61 00 21 00 15 laMundo.java!...
00000190 00 02 00 00 00 00 02 00 01 00 05 00 06 00 01 ...../.....*
000001a0 00 17 00 00 00 2f 00 01 00 01 00 00 00 05 2a b7 ...../.....*
000001b0 00 01 b1 00 00 00 02 00 18 00 00 00 06 00 01 00 ...../.....*
000001c0 00 00 04 00 19 00 00 00 0c 00 01 00 00 00 05 00 ...../.....*
000001d0 1a 00 1b 00 00 00 09 00 1c 00 1d 00 01 00 17 00 ...../.....*
000001e0 00 00 37 00 02 00 01 00 00 00 09 b2 00 07 12 0d ...../.....*
000001f0 b6 00 0f b1 00 00 00 02 00 18 00 00 00 0a 00 02 ...../.....*
00000200 00 00 00 06 00 08 07 00 09 19 00 00 00 0c 00 01 ...../.....*
00000210 00 00 00 09 00 1e 00 1f 00 00 00 01 00 20 00 00 ...../.....*
00000220 00 02 00 21 ...../.....*
```

javap -c -v HolaMundo.class

```
Java SE 23 = 67 (0x43)
Java SE 22 = 66 (0x42)
Java SE 21 = 65 (0x41)
Java SE 20 = 64 (0x40)
Java SE 19 = 63 (0x3F)
Java SE 18 = 62 (0x3E)
Java SE 17 = 61 (0x3D)
Java SE 16 = 60 (0x3C)
Java SE 15 = 59 (0x3B)
Java SE 14 = 58 (0x3A)
Java SE 13 = 57 (0x39)
Java SE 12 = 56 (0x38)
Java SE 11 = 55 (0x37)
Java SE 10 = 54 (0x36)
Java SE 9 = 53 (0x35)
Java SE 8 = 52 (0x34)
Java SE 7 = 51 (0x33)
Java SE 6 = 50 (0x32)
J2SE 5.0 (1.5) = 49 (0x31)
J2SE 1.4 = 48 (0x30)
J2SE 1.3 = 47 (0x2F)
J2SE 1.2 = 46 (0x2E)
JDK 1.1 = 45 (0x2D)
JDK 1.0 = 44 (0x2C)
```

Sobre el "Magic Number" We used to go to lunch at a place called St Michael's Alley. According to local legend, in the deep dark past, the Grateful Dead used to perform there before they made it big. It was a pretty funky place that was definitely a Grateful Dead Kinda Place. When Jerry died, they even put up a little Buddhist-esque shrine. When we used to go there, we referred to the place as Cafe Dead. Somewhere along the line it was noticed that this was a HEX number. I was re-vamping some file format code and needed a couple of magic numbers: one for the persistent object file, and one for classes. I used CAFEDEAD for the object file format, and in grepping for 4 character hex words that fit after "CAFE" (it seemed to be a good theme) I hit on BABE and decided to use it. At that time, it didn't seem terribly important or destined to go anywhere but the trash-can of history. So CAFEDEAD became the class file format, and CAFEDEAD was the persistent object format. But the persistent object facility went away, and along with it went the use of CAFEDEAD - it was eventually replaced by RMI. James Gosling

```
Last modified 27-sep-2024; size 548 bytes
MD5 checksum 864aa3bef57d7acdda9fd02601e427b4
Compiled from "HolaMundo.java"
public class ejemplos.HolaMundo
  minor version: 0
  major version: 51
  flags: ACC_PUBLIC, ACC_SUPER
Constant pool:
 #1 = Methodref          #2.#3      // java/lang/Object.<init>:(JV)
 #2 = Class               #4        // java/lang/Object
 #3 = NameAndType        #5:#6      // <init>:(JV)
 #4 = Utf8                java/lang/Object
 #5 = Utf8                <init>
 #6 = Utf8                (JV)
 #7 = Fieldref           #8.#9      // java/lang/System.out:Ljava/io/PrintStream;
 #8 = Class               #10       // java/lang/System
 #9 = NameAndType        #11:#12     // out:Ljava/io/PrintStream;
 #10 = Utf8               java/lang/System
 #11 = Utf8               out
 #12 = Utf8               Ljava/io/PrintStream;
 #13 = String              #14      // Hola, mundo
 #14 = Utf8               Hola, mundo
 #15 = Methodref         #16.#17     // java/io/PrintStream.println:(Ljava/lang/String;)V
 #16 = Class              #18       // java/io/PrintStream
 #17 = NameAndType       #19:#20     // println:(Ljava/lang/String;)V
 #18 = Utf8               java/io/PrintStream
 #19 = Utf8               println
 #20 = Utf8               (Ljava/lang/String;)V
 #21 = Class              #22     // ejemplos/HolaMundo
 #22 = Utf8               ejemplos/HolaMundo
 #23 = Utf8               Code
 #24 = Utf8               LineNumberTable
 #25 = Utf8               LocalVariableTable
 #26 = Utf8               this
 #27 = Utf8               Lejemplos/HolaMundo;
 #28 = Utf8               main
 #29 = Utf8               ([Ljava/lang/String;)V
 #30 = Utf8               args
 #31 = Utf8               [Ljava/lang/String;
 #32 = Utf8               SourceFile
 #33 = Utf8               HolaMundo.java
{
  public ejemplos.HolaMundo();
    descriptor: (JV)
    flags: ACC_PUBLIC
    Code:
      stack=1, locals=1, args_size=1
      0: aload_0
      1: invokespecial #1 // Method java/lang/Object.<init>:(JV)
      4: return
  LineNumberTable:
    line 4: 0
  LocalVariableTable:
    Start Length Slot Name Signature
     0      5      0  this Lejemplos/HolaMundo;

  public static void main([Ljava/lang/String;);
    descriptor: ([Ljava/lang/String;)V
    flags: ACC_PUBLIC, ACC_STATIC
    Code:
      stack=2, locals=1, args_size=1
      0: getstatic #7 // Field java/lang/System.out:Ljava/io/PrintStream;
      3: ldc #13 // String Hola, mundo
      5: invokevirtual #15 // Method java/io/PrintStream.println:(Ljava/lang/String;)V
      8: return
  LineNumberTable:
    line 6: 0
    line 7: 8
  LocalVariableTable:
    Start Length Slot Name Signature
     0      9      0  args [Ljava/lang/String;

  SourceFile: "HolaMundo.java"
}
```

