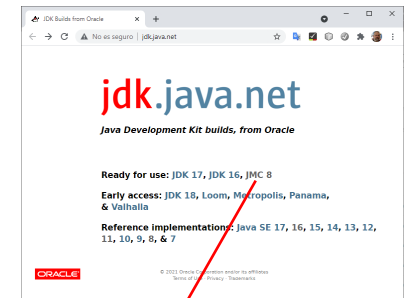
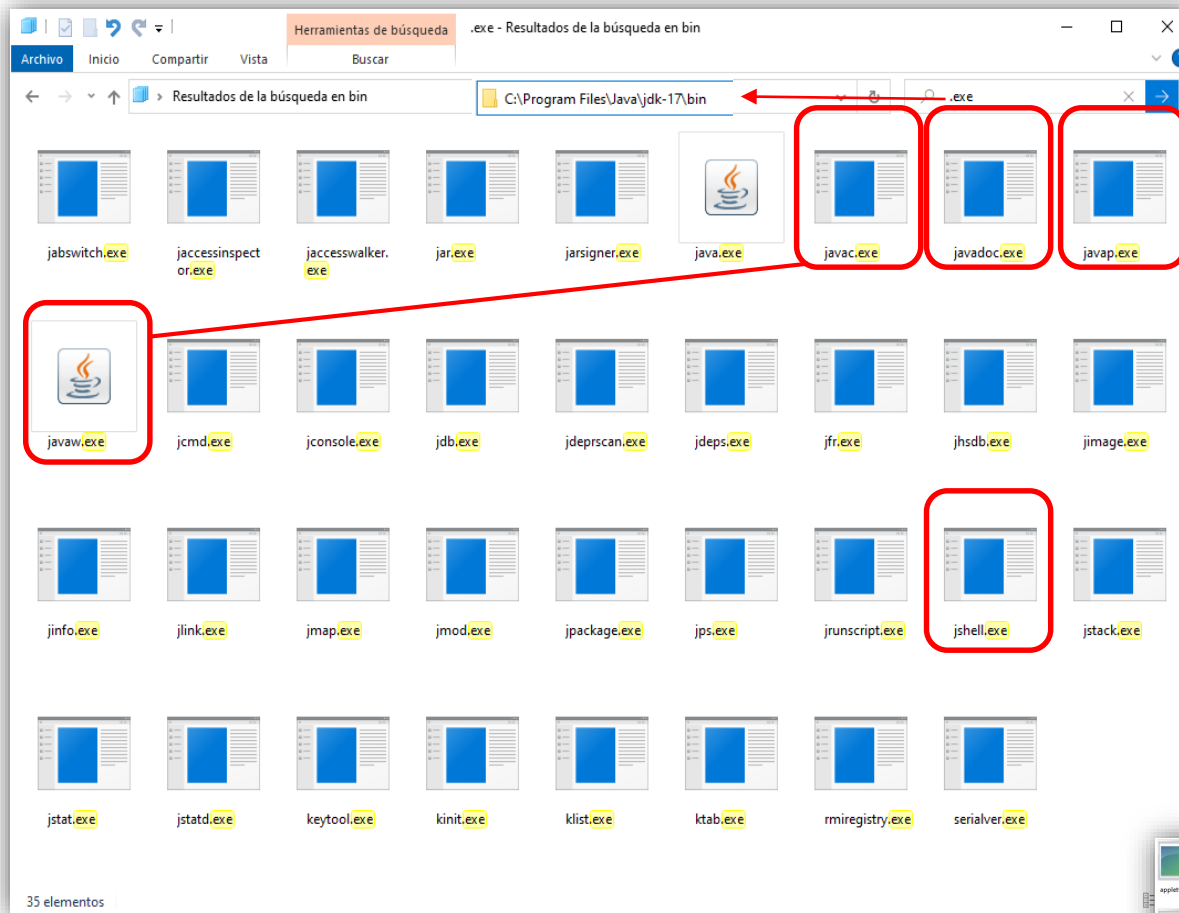


**¿QUÉ TENDRÉ A MI DISPOSICIÓN  
PARA DESARROLLAR EN JAVA?**

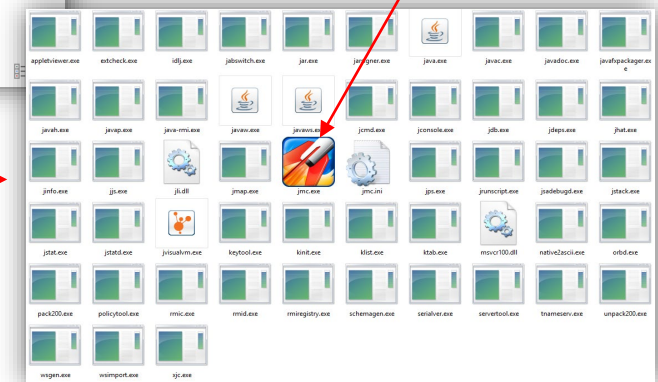
**(COMPILACIÓN, EJECUCIÓN, ETC. )**

# Las herramientas de ejecución y desarrollo de Java El directorio “/bin”



Java Mission Control (JMC)  
Se obtiene ahora por separado  
[\[Ojo a parámetro de instalación\]](#)

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.

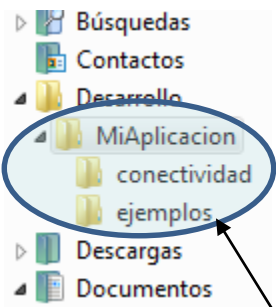
where options include:

-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>
```

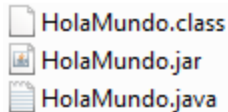
```
D:\>java -version
java version "17" 2021-09-14 LTS
Java(TM) SE Runtime Environment (build 17+35-LTS-2724)
Java HotSpot(TM) 64-Bit Server VM (build 17+35-LTS-2724, mixed mode, sharing)
```

```
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
```



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”)



OJO: para hacer esta prueba hay que incluir la línea **package ejemplos;** en el código fuente de HolaMundo.java

```
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
```