

# Resolución Trabajo Práctico:

## Plataforma Java

### Teoría

1. ¿Qué es la plataforma Java y cómo se diferencia del lenguaje del mismo nombre?

*La plataforma Java es un conjunto de programas y bibliotecas, desarrolladas por Sun, que permiten ejecutar aplicaciones multiplataforma. El lenguaje Java es solo una de las formas de desarrollar aplicaciones para dicha plataforma.*

2. ¿Qué es el bytecode de Java? ¿Cuál es su objetivo?

*Es el lenguaje que entiende la máquina virtual de Java. Su objetivo es ser un lenguaje binario (no es legible por el humano) que pueda ser ejecutado en cualquier plataforma que tenga instalado una máquina virtual de Java. Es decir, sirve como representación intermedia entre el código de la aplicación y la plataforma donde se ejecutará la misma. El lenguaje Java generalmente es compilado (transformado) a esta representación intermedia.*

3. ¿Cuál es la diferencia entre las distribuciones JRE y JDK?

*La JRE es una versión de la plataforma Java que contiene todo lo necesario para ejecutar aplicaciones Java (es decir, en bytecode). Un JRE es lo que requiere un usuario para ejecutar nuestra aplicación Java. El JDK, en cambio, contiene un JRE, pero agrega programas y bibliotecas requeridas por el desarrollador. Esta distribución es la que debería tener instalada el desarrollador.*

4. ¿Qué es la máquina virtual de Java?

*Es la parte central del JRE: es la encargada de correr las aplicaciones en bytecode. Existen diferentes versiones disponibles en diferentes plataformas (windows, linux, android).*

5. ¿Qué es el garbage collector?

*Es el programa encargado de liberar memoria ocupada por objetos que no se utilizan más. De esta manera, el desarrollador se desliga de la responsabilidad de indicar cuándo no utilizará más una sección de memoria.*

6. Nombre tres de los paquetes que pertenecen a la edición standard (SE) de la plataforma Java.

*java.util (donde están las colecciones: ArrayList, HashMap, TreeMap)*

*java.math (contiene utilidades matemáticas: PI, Random)*

*java.lang (contiene clases que permiten soportar el lenguaje: Object, String etc.)*

7. ¿Cuál es el punto de entrada de una aplicación desarrollada en Java?  
*El método main dentro de una clase. La clase que se utilizará como punto de entrada se debe especificar al correr la aplicación.*
8. ¿Cómo se pasan argumentos a una aplicación Java? ¿Cómo accedo a dichos argumentos por posición?  
*Utilizando el único parámetro que recibe el método **main**. Cada elemento se accede mediante corchetes, por ejemplo, args[0], args[1]. Empezando desde 0.*
9. ¿Cómo muestro un mensaje por “consola”? (En este caso, por el panel Output de NetBeans)  
*Utilizando el método println de System.out. Es decir,*  
**System.out.println(“Mensaje”);**

#### **Práctico**

1. Descargar e instalar todas las herramientas necesarias para desarrollar en Java: JDK y NetBeans.  
*Simplemente se verifica que cada alumno haya podido instalar todo el entorno. Las herramientas se buscan en Google.*
2. Crear un proyecto Java en NetBeans y mostrar por pantalla un mensaje.  
*Utilizar el asistente de NetBeans para crear un proyecto Java. El mensaje se muestra usando System.out.println (buscar en las filminas).*
3. Recibir el mensaje por argumento y agregar un mensaje utilizando las preferencias de ejecución de NetBeans.  
*Idem, en las filminas se explica el mecanismo.*
4. Modificar el ejercicio anterior, agregando un mensaje que indique cómo se utiliza nuestra aplicación en caso de que el usuario se haya olvidado de enviar un argumento.  
*Posible solución: Chequear en el main utilizando un if ... else.*

```
public class MiAplicacion {  
    public static void main(String[] args) {  
        if (args.length==0) // Si no hay argumentos.  
            System.out.println("Se debe ingresar un argumento.");  
        else  
            System.out.println(args[0]);  
    }  
}
```