



## Ejercicios I – Aplicaciones básicas Java

Utiliza el JDK desde la Consola para realizar los siguientes ejercicios.

Para cada uno debes:

- 1º) Crea la estructura necesaria para cualquier proyecto: carpetas **src** y **bin**.
- 2º) Edita un nuevo fichero fuente con tu editor preferido (VS Code, Notepad++, Komodo Edit, etc.)
- 3º) Utiliza el compilador (javac) con la opción **-d**
- 4º) Ejecuta la aplicación (java) y verifica que funciona.

### 1) HolaMundo: HolaApp

Escribe la aplicación "Hola Mundo" en Java.

### 2) EsParApp

Escribe una nueva aplicación que nos pida un número entero y escriba en consola si es PAR o es IMPAR.

- Utiliza para escribir el método **print** o **println**.  
Consulta → **RESUMEN apartado 6.1.**
- Utiliza para leer el método **showInputDialog** de la clase `javax.swing.JOptionPane`  
Consulta → **RESUMEN apartado 6.4.**
- Consulta cuál es el operador **MOD** en Java.

### 3) Raices Ecuación 2º grado: RaicesApp

Escribe una aplicación Java que calcule las raíces de una ecuación de 2º grado.

$$ax^2 + bx + c = 0$$

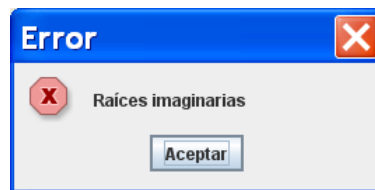
**Solución:**

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

- Los coeficientes a,b y c serán leídos por teclado usando el método **showInputDialog** de la clase `javax.swing.JOptionPane`

Consulta → **RESUMEN apartado 6.4.**

- Utiliza el método **sqrt** de la clase **java.lang.Math** para calcular la raíz cuadrada. Consulta su uso en → **Documentación del API de Java.**
- Si la expresión que hay dentro de la raíz cuadrada (el discriminante) da un resultado negativo, se dice que las raíces son imaginarias. En este caso, se deberá mostrar una ventana de Error que diga "Raíces imaginarias".
- Utiliza el método **showMessageDialog** de la clase **javax.swing.JOptionPane**  
Consulta → **RESUMEN apartado 6.2.**



- Ejemplo: para 3, -9 y 4, la solución es  $X_1: 0.542573$  y  $X_2: 2.457427$
- Ejemplo: para 2, 4 y 5, las raíces son imaginarias. (el valor de dentro de la raíz es -24)
- Comprueba los resultados en: <https://es.onlimeschool.com/math/assistance/equation/quadratic/>

### Utiliza un IDE de tu elección para realizar los siguientes ejercicios

**IDE** (Apache NetBeans o Eclipse)

- Crea un **nuevo proyecto** Java
- Añade una **nueva clase** Java con el método **main** (en NetBeans selecciona Java Main Class).
- Escribe el código Java con el **editor integrado**.
- Compila el código y fíjate qué **errores de compilación** existen.
- Ejecuta la aplicación: visualiza la salida de la ejecución en el panel **Output** o **Console**.

#### 4) Riego parcela: RiegoApp

Tenemos una parcela circular que necesitamos regar y vallar.

Escribe una aplicación Java que pida por teclado la superficie de la parcela y calcule dos valores:



- a) la longitud en metros del brazo necesario para regarla (pivote).
- b) los metros de valla que necesitamos para rodear la parcela.

El programa debe escribir en pantalla la siguiente frase, usando una sola llamada a **printf**.

Ejemplo: para 2000 metros cuadrados, escribirá

La longitud del brazo debe ser de 25,23 m. y la valla tendrá 158,53 m.

- Consulta el código de formato para usar **printf** con valores numéricos reales.
- Existe la constante **PI** y el método **pow** dentro de la clase `java.lang.Math`. El método `pow` permite calcular la potencia.  
Consulta su uso en → **Documentación del API de Java**.

#### 5) Mayor de tres números: MayorApp

Escribe una aplicación Java que lea tres números enteros y escriba por consola cuál de ellos es el mayor. En el caso de que dos o los tres sean iguales, se escribirá en consola que no son los tres distintos.

- Utiliza la instrucción **condicional doble** en Java: `if-else`
- Consulta los **operadores lógicos** en Java (AND / OR / NOT).

#### 6) BisiestoApp

Escribe una aplicación Java que nos pida un año y nos escriba si es bisiesto o no.

- Utiliza la sentencia **if-else** con operadores lógicos.

#### 7) Lista de números: ListaNumerosApp

Escribe una aplicación Java que escriba en pantalla los números del 1 a un número leído por teclado de 3 en 3. Al final mostrará la cantidad de números escritos.

- Utiliza la sentencia **while**.
- Ejemplo: si tecleo 10:  
1  
4

7  
10  
---

**Total: 4**

## 8) Divisibles\_2\_3App

Escribe una aplicación Java que muestre los números divisibles por 2 O por 3 desde 1 hasta N, siendo N un valor pedido por teclado.  
Al final mostrará la cuenta total.

## 9) Tabla de trigonometría: TablaTrigonoApp

Escribe una aplicación Java que escriba en la consola una tabla con el seno, coseno y tangente para todos los grados entre 0° y 360°.

- **AVISO:** los métodos de la **clase Math** trabajan en **radianes**.

Consulta su uso en [→ Documentación del API de Java.](#)

- **AVISO:** Para el valor de 90° y 270° no hay solución para la Tangente.

- Comprueba que el valor del seno de 90° es 1 y la tangente de 45° es 1.

GRADOS	SENO	COSENO	TANGENTE
0	0,00	1,00	0,00
1	0,02	1,00	0,02
2	0,03	1,00	0,03
3	0,05	1,00	0,05
4	0,07	1,00	0,07
5	0,09	1,00	0,09
...			
90	1,00	0,00	INF

- Utiliza la sentencia **while** primero.
- Después, sustitúyela por la sentencia **for**.

## 10) Serie Fibonacci: SerieFibonacciApp

Escribe una aplicación Java que escriba en pantalla la serie de Fibonacci sin superar un número que será pedido por teclado.

- Utiliza la instrucción repetitiva **while** o **for**.

- **Ejemplo:** si tecleo 10  
1 1 2 3 5 8

- **Ejemplo:** si tecleo 21  
1 1 2 3 5 8 13 21

## 11) FactorialApp (versión iterativa)

Escribe una aplicación Java que escriba el factorial de un número dado por teclado.

- Valida que el número tecleado sea mayor que cero. Utiliza la sentencia **do-while** para la validación.
- Utiliza la instrucción repetitiva **for** para la parte del cálculo.
- **Ejemplo:** para 4 debe escribir en consola

*El factorial de 4 es 24*

## 12) Adivinar un número: AdivinaClaveApp

Escribe una aplicación Java que compruebe si el número tecleado coincide con el elegido.

- El número elegido es 9. Se definirá como **CONSTANTE**. Por ejemplo:  
**final** int CLAVE = 9;
- El nº de intentos máximo es de 3 que se definirá como **CONSTANTE**.
- Utiliza la instrucción repetitiva **while**.
- Muestra una **caja de diálogo** que diga si acertó o no, con el tipo de icono correspondiente.

## 13) Cuadrado de asteriscos: CuadradoApp

Escribe una aplicación Java que lea un número entero N por teclado y nos dibuje un cuadrado formado por asteriscos de lado N.

- Utiliza la instrucción repetitiva **while**.

## 14) Día del mes: DiaMesApp

Escribe una aplicación Java que lea un número de mes y escriba su nombre. Por ejemplo, si tecleo 3, escribirá: "Mes 3 es Marzo"

- En caso de que el mes tecleado no sea válido, escribirá "Mes XX NO existe"
- Utiliza la instrucción **switch** con **case ->** .

### 15) Número de días de cada mes: DiasPorMesApp

Escribe una aplicación Java que lea un número de mes y escriba el número de días que tiene dicho mes.

- Validación: en caso de que el mes no sea válido, escribirá "Mes inválido" y volverá a leer. Utiliza el **while** o **do-while**.
- Utiliza una **expresión switch** con la sentencia **yield**.
- MEJORA: podrías mejorar la aplicación leyendo el año en caso de que tecleen 2 (febrero) para comprobar si ese año es bisiesto o no lo es.

### 16) Pirámide: PiramideApp

Escribe una aplicación Java que escriba en consola una pirámide de altura H, formada por un carácter dado.

- El valor de la altura y el carácter a usar se leerán por teclado
- Utiliza la instrucción repetitiva **for**.

### 17) Pirámide invertida: PiramideInvApp

Se pide lo mismo que el ejercicio anterior, pero la pirámide que debe escribir debe estar invertida.

### 18) Rombo de asteriscos: RomboApp

Escribe una aplicación Java que lea un número entero N por teclado y nos dibuje un rombo de altura N formado por asteriscos.

### 19) Tabla Unicode: TablaUnicodeApp

Escribe una aplicación Java que escriba en la consola la tabla de caracteres UNICODE desde el código 33 al 126.

- La presentación debe hacerse en una tabla con cinco columnas.
- Utiliza el **"casting"** de tipos.
- Utiliza constantes de tipo **short** para definir el primer carácter y el último, y también para definir el nº de columnas de la tabla.

Por ejemplo:

```
final short PRIMER_CAR = 33;
```