

## UT02. Bloque 2. Ejercicios: Trabajo con texto

---

### Ejercicio 1: Ficha de Presentación (Dificultad: Fácil)

**Resultado Esperado (ejemplo):**

```
--- Ficha Personal ---  
Mi nombre es: Carlos Santana  
Nací en: Valencia  
Mi comida favorita es: Paella
```

**Solución:**

```
public class FichaPresentacion {  
  
    public static void main(String[] args) {  
        // 1. Declaramos las tres variables de tipo String.  
        String nombreCompleto = "Carlos Santana";  
        String ciudadNacimiento = "Valencia";  
        String comidaFavorita = "Paella";  
  
        // 2. Mostramos la información por pantalla usando println.  
        // El operador '+' se usa para concatenar (unir) el texto fijo con  
        // el valor de la variable.  
        System.out.println("--- Ficha Personal ---");  
        System.out.println("Mi nombre es: " + nombreCompleto);  
        System.out.println("Nací en: " + ciudadNacimiento);  
        System.out.println("Mi comida favorita es: " + comidaFavorita);  
    }  
}
```

---

### Ejercicio 2: Creando un Camino (Dificultad: Fácil)

**Resultado Esperado:**

```
Paso 1... Paso 2... Paso 3... ¡He llegado!
```

**Solución:**

```
public class CreandoCamino {  
  
    public static void main(String[] args) {  
        // Usamos System.out.print() para que el cursor se quede en la
```

```
misma línea.  
    // Es crucial añadir los espacios manualmente al final de cada  
texto.  
    System.out.print("Paso 1... ");  
    System.out.print("Paso 2... ");  
    System.out.print("Paso 3... ");  
  
    // Usamos System.out.println() al final para imprimir el último  
mensaje  
    // y ahora sí, hacer un salto de línea.  
    System.out.println("¡He llegado!");  
}  
}
```

---

### Ejercicio 3: Ticket de Supermercado (Dificultad: Intermedia)

**Objetivo:** Practicar el uso de `System.out.printf()` con diferentes tipos de datos (texto, enteros y decimales).

**Enunciado:** Crea un programa que simule un ticket de compra simple. Declara variables para tres productos:

- **Producto 1:** Nombre: "Manzanas", Cantidad: 3, Precio/Ud: 1.50
- **Producto 2:** Nombre: "Leche Entera", Cantidad: 1, Precio/Ud: 0.90
- **Producto 3:** Nombre: "Galletas", Cantidad: 2, Precio/Ud: 2.10

**Resultado Esperado:**

```
--- TICKET DE COMPRA ---  
Producto: Manzanas, Cantidad: 3, Precio: 1.50€  
Producto: Leche Entera, Cantidad: 1, Precio: 0.90€  
Producto: Galletas, Cantidad: 2, Precio: 2.10€```
```

**Solución:**

```
public class TicketSupermercado {  
  
    public static void main(String[] args) {  
        // --- Datos del Producto 1 ---  
        String nombreProducto1 = "Manzanas";  
        int cantidad1 = 3;  
        double precio1 = 1.50;  
  
        // --- Datos del Producto 2 ---  
        String nombreProducto2 = "Leche Entera";  
        int cantidad2 = 1;  
        double precio2 = 0.90;  
    }  
}
```

```
// --- Datos del Producto 3 ---
String nombreProducto3 = "Galletas";
int cantidad3 = 2;
double precio3 = 2.10;

// Imprimimos la cabecera del ticket.
System.out.println("--- TICKET DE COMPRA ---");

// Usamos printf para formatear la salida.
// %s se reemplaza por un String.
// %d se reemplaza por un entero (decimal).
// %.2f se reemplaza por un double/float, mostrando exactamente 2
decimales.
// \n al final es el carácter de nueva línea.
System.out.printf("Producto: %s, Cantidad: %d, Precio: %.2f€\n",
nombreProducto1, cantidad1, precio1);
System.out.printf("Producto: %s, Cantidad: %d, Precio: %.2f€\n",
nombreProducto2, cantidad2, precio2);
System.out.printf("Producto: %s, Cantidad: %d, Precio: %.2f€\n",
nombreProducto3, cantidad3, precio3);
}
```

#### Ejercicio 4: Dibujando una Pirámide (Dificultad: Intermedia)

**Objetivo:** Usar la impresión por consola de forma creativa para generar figuras. Demuestra el control sobre los saltos de línea.

**Enunciado:** Escribe un programa que dibuje una pirámide simple usando el carácter asterisco (\*). La pirámide debe tener 5 niveles de altura.

**Resultado Esperado:**

```
  *
 ***
*****
*****
*****
```

**Solución:**

```
public class Piramide {

    public static void main(String[] args) {
        // --- VARIABLE DE CONTROL ---
        // ¡Cambia este valor para ver pirámides de diferentes tamaños!
        int altura = 7;
    }
}
```

```

        System.out.println("--- Pirámide Normal (Altura " + altura + ") --
        -");

        // --- BUCLE PARA LA PIRÁMIDE NORMAL ---

        // 1. Bucle Exterior: Controla las filas.
        // Se ejecutará 'altura' veces, una por cada nivel de la pirámide.
        // La variable 'i' representa el número de la fila actual (de 1 a
        la altura total).
        for (int i = 1; i <= altura; i++) {

            // 2. Primer Bucle Interior: Imprime los espacios en blanco.
            // En la fila 1, necesitamos 'altura - 1' espacios.
            // En la fila 2, necesitamos 'altura - 2' espacios.
            // ...y así sucesivamente. La fórmula es 'altura - i'.
            for (int j = 1; j <= altura - i; j++) {
                System.out.print(" "); // Ojo: print, no println
            }

            // 3. Segundo Bucle Interior: Imprime los asteriscos.
            // En la fila 1, necesitamos 1 asterisco.
            // En la fila 2, necesitamos 3 asteriscos.
            // En la fila 3, necesitamos 5 asteriscos.
            // La fórmula para esta secuencia de números impares es '(2 *
            i) - 1'.
            for (int k = 1; k <= (2 * i) - 1; k++) {
                System.out.print("*"); // Ojo: print, no println
            }

            // 4. Salto de línea.
            // Una vez que los espacios y los asteriscos de una fila se
            han impreso,
            // hacemos un salto de línea para empezar la siguiente fila.
            System.out.println();
        }
    }
}

```

### Ejercicio 5: Dibujando una Pirámide (Dificultad: Intermedia)

**Enunciado:** Repite el ejercicio anterior, pero ahora estando la pirámide invertida.

**Pistas:**

- Piensa que es el proceso opuesto al anterior. Empezarás con la línea más larga y terminarás con la más corta.
- La primera línea es la que no necesita espacios en blanco al principio para estar centrada.
- A partir de la segunda línea, tendrás que añadir espacios antes de los asteriscos para que se mantenga centrada. La segunda línea necesitará un espacio, la tercera dos, y así sucesivamente.
- El número de asteriscos deberá disminuir en cada paso. Si empezaste con 9, el siguiente nivel tendrá 7, luego 5...

**Resultado Esperado:**

```
*****
*****
*****
***
*
```

**Solución:**

```
public class PiramideInvertida {

    public static void main(String[] args) {

        // --- BUCLE PARA LA PIRÁMIDE INVERTIDA ---

        // 1. Bucle Exterior: Controla las filas, pero esta vez en orden
        // descendente.
        // Empezamos en la fila más ancha (i = altura) y vamos hacia la
        // más estrecha (i = 1).
        for (int i = altura; i >= 1; i--) {

            // 2. Primer Bucle Interior: Imprime los espacios en blanco.
            // La lógica es la misma que antes: 'altura - i'.
            // Cuando i = altura, (altura - altura) = 0 espacios.
            // Cuando i = altura - 1, (altura - (altura - 1)) = 1 espacio.
            for (int j = 1; j <= altura - i; j++) {
                System.out.print(" ");
            }

            // 3. Segundo Bucle Interior: Imprime los asteriscos.
            // La fórmula también es la misma: '(2 * i) - 1'.
            // Como 'i' va disminuyendo, el número de asteriscos también
            lo hará.
            for (int k = 1; k <= (2 * i) - 1; k++) {
                System.out.print("*");
            }

            // 4. Salto de línea para pasar a la siguiente fila.
            System.out.println();

        }
    }
}
```

---

**Ejercicio 6: Tabla de Clasificación (Dificultad: Avanzada)**

**Objetivo:** Dominar la alineación de columnas con `System.out.printf()` para crear tablas limpias y profesionales.

**Enunciado:** Crea un programa que muestre una tabla de clasificación de un torneo. La tabla debe tener columnas para "Posición", "Equipo" y "Puntos". Los datos son:

- 1º, "Dragones Rojos", 45 puntos
- 2º, "Tiburones de la Costa", 42 puntos
- 3º, "Águilas del Norte", 38 puntos
- 10º, "Gatos Salvajes", 21 puntos

La columna "Equipo" debe estar alineada a la izquierda y las columnas "Posición" y "Puntos" a la derecha. Asegúrate de que todo quede perfectamente alineado, incluso con el 10º puesto.

**Resultado Esperado:**

| POSICIÓN | EQUIPO                | PUNTOS |
|----------|-----------------------|--------|
| 1º       | Dragones Rojos        | 45     |
| 2º       | Tiburones de la Costa | 42     |
| 3º       | Águilas del Norte     | 38     |
| 10º      | Gatos Salvajes        | 21     |

**Solución:**

```
public class TablaClasificacion {  
  
    public static void main(String[] args) {  
        // --- Datos del Torneo ---  
        String pos1 = "1º";  
        String equipo1 = "Dragones Rojos";  
        int puntos1 = 45;  
  
        String pos2 = "2º";  
        String equipo2 = "Tiburones de la Costa";  
        int puntos2 = 42;  
  
        String pos3 = "3º";  
        String equipo3 = "Águilas del Norte";  
        int puntos3 = 38;  
  
        String pos10 = "10º";  
        String equipo10 = "Gatos Salvajes";  
        int puntos10 = 21;  
  
        // --- Plantilla de formato ---  
        // | %-8s | %-24s | %8s |  
        // %-8s: String, 8 caracteres de ancho, alineado a la IZQUIERDA  
(-).  
        // %-24s: String, 24 caracteres, alineado a la IZQUIERDA.  
        // %8s: String, 8 caracteres, alineado a la DERECHA (por defecto).
```

```
        // El mismo formato se usa para la cabecera y los datos para que
        todo encaje.

        System.out.println("-----");
    };
    System.out.printf("| %-8s | %-24s | %8s |\n", "POSICIÓN",
"EQUIPO", "PUNTOS");
    System.out.println("-----");
    };
    System.out.printf("| %8s | %-24s | %8d |\n", pos1, equipo1,
puntos1);
    System.out.printf("| %8s | %-24s | %8d |\n", pos2, equipo2,
puntos2);
    System.out.printf("| %8s | %-24s | %8d |\n", pos3, equipo3,
puntos3);
    System.out.printf("| %8s | %-24s | %8d |\n", pos10, equipo10,
puntos10);
    System.out.println("-----");
    };
    }
}
```

Ejercicio 6: Factura Detallada (Dificultad: Desafío)

**Objetivo:** Combinar el uso de variables, operaciones aritméticas y formato de salida avanzado para crear un programa más completo y práctico.

**Enunciado:** Escribe un programa que genere una factura. Declara variables para dos productos, incluyendo su nombre, cantidad y precio unitario. El programa debe:

- 1. Calcular el subtotal de cada producto (cantidad \* precio unitario).
- 2. Calcular el total general de la factura (suma de los subtotales).
- 3. Mostrar una tabla perfectamente alineada con las columnas: "Producto", "Cant.", "Precio Ud.", "Subtotal".
- 4. Mostrar el "TOTAL A PAGAR" al final, claramente destacado.

Resultado Esperado:

|                  |       |            |          |
|------------------|-------|------------|----------|
| =====            |       |            |          |
| FACTURA          |       |            |          |
| =====            |       |            |          |
| PRODUCTO         | CANT. | PRECIO UD. | SUBTOTAL |
| -----            |       |            |          |
| Teclado Mecánico | 2     | 89.99      | 179.98   |
| Ratón Gaming RGB | 1     | 54.50      | 54.50    |
| -----            |       |            |          |
| TOTAL A PAGAR:   |       |            | 234.48€  |
| =====            |       |            |          |

**Solución:**

```
public class FacturaDetallada {

    public static void main(String[] args) {
        // --- Datos de los productos ---
        String nombreProd1 = "Teclado Mecánico";
        int cantProd1 = 2;
        double precioProd1 = 89.99;

        String nombreProd2 = "Ratón Gaming RGB";
        int cantProd2 = 1;
        double precioProd2 = 54.50;

        // 1. Realizamos todos los cálculos ANTES de imprimir.
        double subtotal1 = cantProd1 * precioProd1;
        double subtotal2 = cantProd2 * precioProd2;
        double totalFactura = subtotal1 + subtotal2;

        // --- Impresión de la Factura ---

        System.out.println("=====
        =====");
        System.out.println("                                FACTURA");

        System.out.println("=====
        =====");

        // Cabecera de la tabla
        System.out.printf("%-20s | %5s | %12s | %12s\n", "PRODUCTO",
"CANT.", "PRECIO UD.", "SUBTOTAL");
        System.out.println("-----
        -----");

        // Filas de productos
        System.out.printf("%-20s | %5d | %12.2f | %12.2f\n", nombreProd1,
cantProd1, precioProd1, subtotal1);
        System.out.printf("%-20s | %5d | %12.2f | %12.2f\n", nombreProd2,
cantProd2, precioProd2, subtotal2);
        System.out.println("-----
        -----");

        // Total final
        System.out.printf("%42s %12.2f€\n", "TOTAL A PAGAR:",
totalFactura);

        System.out.println("=====
        =====");
    }
}
```