

Actividades Entregables de la Unidad 2: Programación estructurada

Resultados de Aprendizaje (RAs) Trabajados:

Este conjunto de actividades está diseñado para afianzar los siguientes RAs del módulo de Programación:

- **RA2, RA3, RA5 y RA6**
 - **Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.** (Aplicado en el uso de if-else, switch, for, while y do-while para controlar el flujo de los programas).
 - **Desarrolla programas aplicando la programación estructurada e introduciendo el tratamiento de datos.** (Aplicado en el uso de arrays y matrices para almacenar y gestionar colecciones de datos como notas, precios o butacas).

Objetivo General: Aplicar los conocimientos sobre variables, condicionales, bucles y arrays (vectores y matrices) para resolver problemas prácticos y estructurados.

Actividad 1: La Cooperativa de Naranjas

La cooperativa de naranjas "Casablanca" necesita un programa para calcular el precio final por kilo de naranja en el momento de la venta. Este precio final depende de un **precio inicial** y se ajusta según el **tipo** (A o B) y el **tamaño** (1 o 2) de la naranja.

Tu misión es escribir un programa que determine cuánto recibirá un productor por cada kilo de naranja, aplicando las siguientes reglas:

- Si es de **tipo A**:
 - Se **suman 10 céntimos** al precio si es de **tamaño 1**.
 - Se **suman 25 céntimos** al precio si es de **tamaño 2**.
- Si es de **tipo B**:
 - Se **restan 5 céntimos** al precio si es de **tamaño 1**.
 - Se **restan 10 céntimos** al precio si es de **tamaño 2**.

El programa debe solicitar el precio inicial, el tipo y el tamaño para realizar el cálculo.

#Pista: Este problema es ideal para resolverlo con una estructura `if-else if` o con un `switch` anidado.

Ejemplo de Salida

```
--- COOPERATIVA CASABLANCA: CALCULADORA DE PRECIOS ---
```

```
Introduce los siguientes datos:
```

```
- Precio inicial del kilo de naranja:
```

```
0.30
```

```
- Tipo de naranja (A/B):
```

```
A
```

```
- Tamaño de naranja (1/2):
```

```
2
```

```
--- RESULTADO ---
```

```
El precio final de venta es 0.55 €/kg
```

Entrega Actividad 1

- Genera un fichero **Java** con el nombre `ud2_actividad1_[tu_nombre].java`. Por ejemplo, en mi caso, `ud2_actividad1_jose_abad.java`.

<!--

Actividad 2: Gestor de Calificaciones de Clase

Un profesor necesita una herramienta digital para gestionar las notas de su clase. Tu tarea es crear un programa que funcione como un completo gestor de calificaciones.

El programa debe seguir estos pasos:

1. Al iniciarse, preguntará **cuántos alumnos hay en la clase**.
2. Pedirá la nota de cada uno de los alumnos y las guardará en un **array de tipo double**.
3. Una vez introducidas las notas, mostrará un **menú repetitivo** que ofrezca las siguientes opciones:
 1. Mostrar todas las notas.
 2. Calcular y mostrar la **nota promedio** de la clase.
 3. Indicar cuántos alumnos están **aprobados** ($nota \geq 5$) y cuántos **suspensos**.
 4. Mostrar la **nota más alta y la más baja** de la clase.
 5. Salir del programa.
4. El menú debe seguir apareciendo hasta que el usuario elija la opción de salir.

#Pista: Un bucle `do-while` es perfecto para el menú, y un `switch` para gestionar las opciones. ¡No olvides validar las entradas del usuario!

Entrega Actividad 2

- Genera un fichero `Java` con el nombre `ud2_actividad2_[tu_nombre].java`. Por ejemplo, en mi caso, `ud2_actividad2_jose_abad.java`.

--- -->

Actividad 2: Análisis de Facturación

Una empresa que vende desinfectantes necesita un programa para analizar rápidamente sus ventas. En cada factura figura el **código del artículo**, la **cantidad vendida en litros** y el **precio por litro**.

El programa debe procesar la información de **5 facturas** (que se pedirán por teclado) y, al final, mostrar un resumen con los siguientes datos:

1. **Facturación total:** La suma de los importes de todas las facturas.
2. **Litros del Artículo 1:** La cantidad total de litros vendidos específicamente para el artículo con código 1.
3. **Facturas Superiores a 200€:** El número de facturas cuyo importe total superó los 200 €.

#Pista: Necesitarás un bucle que se repita 5 veces. Dentro del bucle, pide los datos de cada factura y actualiza tus variables acumuladoras y contadoras. Piensa ¿cuántos arrays necesitas para almacenar la información?

Ejemplo de Entrada y Salida

```
// El usuario introducirá estos datos
--- EJEMPLO DE ENTRADA ---
- Código: 1, Litros: 30, Precio/L: 4.5
- Código: 2, Litros: 20, Precio/L: 5.0
- Código: 1, Litros: 50, Precio/L: 4.5
- Código: 3, Litros: 10, Precio/L: 5.8
- Código: 3, Litros: 20, Precio/L: 5.8

// El programa mostrará el Resumen de Ventas
--- RESUMEN DE VENTAS ---
* Facturación total: 634.00 €
* Cantidad de litros vendidos del artículo #1: 80
* Número de facturas de más de 200 €: 1
```

Entrega Actividad 2

- Genera un fichero Java con el nombre `ud2_actividad2_[tu_nombre].java`. Por

ejemplo, en mi caso, `ud2_actividad2_jose_abad.java`.

Actividad 3: Simulador de Cajero Automático

¡Vamos a programar el software de un cajero automático! El cajero debe permitir a los usuarios consultar su saldo, ingresar y retirar dinero.

El programa debe cumplir los siguientes requisitos:

1. El cajero dispone de billetes de 500, 200, 100, 50, 20, 10 y 5 euros. Guarda estas denominaciones en un **array de enteros**.
2. El saldo inicial del usuario será de 0.
3. El programa mostrará un **menú repetitivo** con las siguientes opciones:
 - Consultar saldo actual.
 - Ingresar dinero (pide una cantidad y la suma al saldo).
 - Retirar dinero.
 - Salir.
4. La opción "**Retirar dinero**" es la más compleja:
 - Debe comprobar si hay saldo suficiente. Si no lo hay, mostrará un mensaje de error.
 - Si hay saldo, debe calcular el **menor número posible de billetes** para entregar la cantidad solicitada.
 - Finalmente, debe mostrar cuántos billetes de cada tipo entrega.

Ejemplo de Ejecución

```
----- MENÚ CAJERO AUTOMÁTICO -----
```

1. Consultar saldo
2. Ingresar dinero
3. Retirar dinero
4. Salir

```
Elige una opción: 2  
Cantidad a ingresar: 385  
Saldo actual: 385.0 €
```

```
----- MENÚ CAJERO AUTOMÁTICO -----
```

1. Consultar saldo
2. Ingresar dinero
3. Retirar dinero
4. Salir

```
Elige una opción: 3
```

```
Cantidad a retirar: 280
```

```
Desglose de billetes entregados:
```

- 1 billete de 200€
- 1 billete de 50€
- 1 billete de 20€
- 1 billete de 10€

```
Operación realizada con éxito.
```

```
Saldo actual: 105.0 €
```

Entrega Actividad 3

- Genera un fichero **Java** con el nombre `ud2_actividad3_[tu_nombre].java`. Por ejemplo, en mi caso, `ud2_actividad3_jose_abad.java`.

Actividad 4: Gestor de Butacas de Cine

¡Has sido contratado para programar el sistema de venta de entradas de un pequeño cine! La sala de cine se representará como una **matriz de caracteres**, donde cada posición es una butaca.

El programa debe realizar lo siguiente:

1. Al iniciarse, crea una matriz de 5 filas por 5 columnas para representar la sala.
2. Inicializa todas las butacas con el carácter 'L' (Libre).

3. Muestra un **menú repetitivo** con las siguientes opciones:

1. **Mostrar butacas:** Dibuja en la consola el estado actual de la sala, mostrando las filas y columnas y el carácter de cada butaca (L o 0 de Ocupada).
2. **Comprar entrada:** Pide al usuario la fila y la columna de la butaca que desea.
 - Si la butaca está libre (L), la cambia a ocupada (0) y muestra un mensaje de "Compra realizada".
 - Si la butaca ya está ocupada (0), debe mostrar un mensaje de "Butaca no disponible".
 - Debe validar que la fila y columna introducidas sean correctas.
3. **Mostrar estadísticas:** Calcula y muestra el número de butacas libres, el número de butacas ocupadas y el total recaudado (suponiendo que cada entrada cuesta 5€).
4. **Salir.**

Ejemplo de Salida

```
----- CINE DAM -----
```

1. Mostrar butacas
2. Comprar entrada
3. Ver estadísticas
4. Salir

Elige una opción: 1

Asientos:

0 1 2 3 4

0 L L L L L
1 L L L L L
2 L L L L L
3 L L L L L
4 L L L L L

```
----- CINE DAM -----
```

1. Mostrar butacas
2. Comprar entrada
3. Ver estadísticas
4. Salir

Elige una opción: 2

Introduce la fila (0-4): 2

Introduce la columna (0-4): 3

Compra realizada con éxito.

----- CINE DAM -----

1. Mostrar butacas
2. Comprar entrada
3. Ver estadísticas
4. Salir

Elige una opción: 1

Asientos:

0 1 2 3 4

0 L L L L L

1 L L L L L

2 L L L 0 L

3 L L L L L

4 L L L L L

Elige una opción: 3

--- ESTADÍSTICAS ---

Butacas Ocupadas: 1

Butacas Libres: 24

Total Recaudado: 5.0 €

Entrega Actividad 4

- Genera un fichero Java con el nombre `ud2_actividad4_[tu_nombre].java`. Por ejemplo, en mi caso, `ud2_actividad4_jose_abad.java`.