**Ejercicios funciones y Bucles**

Ejercicio 1: Cálculo de ingresos mensuales

Una pequeña tienda registra sus ventas diarias en un arreglo. Se necesita escribir una función que use un bucle para calcular los ingresos totales mensuales.

\*\*Instrucciones\*\*:

- Crea un arreglo `ventas` que almacene las ventas diarias de una tienda durante 30 días.

- Escribe una función `calcularIngresosMensuales` que recorra el arreglo y sume las ventas.

- Imprime el total de ingresos mensuales.

Ejercicio 2: Generar reportes de inventario

Una tienda de electrónica necesita un informe del inventario, mostrando los productos que tienen menos de 5 unidades en stock.

\*\*Instrucciones\*\*:

- Crea un arreglo `productos` que almacene los nombres de los productos y otro arreglo `stocks` que almacene la cantidad de stock de cada producto.

- Escribe una función `generarReporteBajoStock` que imprima los productos con menos de 5 unidades.

public class Inventario {

public static void main(String[] args) {

String[] productos = {"Televisor", "Celular", "Laptop", "Tablet", "Cámara"};

int[] stocks = {3, 10, 2, 6, 4};

generarReporteBajoStock(productos, stocks);

}

public static void generarReporteBajoStock(String[] productos, int[] stocks) {

System.out.println("Productos con bajo stock:");

for (int i = 0; i < productos.length; i++) {

if (stocks[i] < 5) {

System.out.println(productos[i] + " - Unidades: " + stocks[i]);

}

}

}

}

Ejercicio 3: Gestión de clientes

Una pequeña empresa de servicios tiene una lista de clientes a los que debe enviar facturas, pero quiere procesar solo aquellos que tengan facturas pendientes por más de $500.

\*\*Instrucciones\*\*:

- Crea un arreglo `clientes` que almacene los nombres de los clientes y un arreglo `facturasPendientes` con el monto de sus facturas pendientes.

- Escribe una función `enviarFacturas` que imprima los nombres de los clientes con facturas mayores a $500.

Ejercicio 4: Cálculo de descuentos para fidelización

Una empresa ofrece descuentos a sus clientes dependiendo de la cantidad de compras que han realizado. Si han comprado más de 10 veces, reciben un 10% de descuento.

\*\*Instrucciones\*\*:

- Crea un arreglo `clientes` y otro `compras` que almacene la cantidad de compras de cada cliente.

- Escribe una función `calcularDescuentos` que recorra el arreglo y aplique el descuento a los clientes que califiquen.

¡Claro! Aquí tienes cinco ejercicios adicionales en Java, orientados a pequeñas empresas y enfocados en bucles y funciones.

Ejercicio 5: Promedio de satisfacción del cliente

Una empresa desea conocer el promedio de satisfacción de sus clientes basado en encuestas de calificación de 1 a 5.

\*\*Instrucciones\*\*:

- Crea un arreglo `calificaciones` con las puntuaciones de satisfacción de los clientes.

- Escribe una función `calcularPromedioSatisfaccion` que recorra el arreglo y calcule el promedio de satisfacción.

Ejercicio 6: Cálculo de impuestos

Una pequeña empresa debe calcular el impuesto de venta para cada una de las facturas emitidas.

\*\*Instrucciones\*\*:

- Crea un arreglo `facturas` con los montos de las facturas de los clientes.

- Escribe una función `calcularImpuesto` que aplique un 21% de impuesto a cada factura.

Ejercicio 7: Descuento por compras grandes

Una tienda de ropa ofrece un 15% de descuento a las compras superiores a $500.

\*\*Instrucciones\*\*:

- Crea un arreglo `compras` que almacene los montos de las compras de los clientes.

- Escribe una función `aplicarDescuento` que aplique el descuento a las compras que superen los $500.

Ejercicio 8: Control de horas trabajadas

Una empresa de servicios necesita llevar el control de las horas trabajadas por sus empleados durante una semana.

\*\*Instrucciones\*\*:

- Crea un arreglo `empleados` con los nombres de los empleados y otro arreglo `horasTrabajadas` con las horas trabajadas en la semana.

- Escribe una función `calcularPagoSemanal` que calcule el pago semanal (a razón de $15 por hora trabajada).

Ejercicio 9: Calcular el precio final con descuentos

Una empresa quiere calcular el precio final de sus productos después de aplicar un descuento general del 10%.

\*\*Instrucciones\*\*:

- Crea un arreglo `preciosOriginales` que almacene el precio original de varios productos.

- Escribe una función `calcularPrecioFinal` que aplique el descuento a cada producto y devuelva el precio final.