



UNIVERSIDAD TECMILENIO

Programacion Orientada a Objetos

Hector Aguilar Mogollan

Emiliano Rodriguez Montes

ACTIVIDAD 2

OBJETIVO:

El objetivo de esta actividad es desarrollar habilidades en Java relacionadas con la interacción con el usuario, el manejo de estructuras de control y el uso de arreglos. En la primera parte, se busca que el estudiante implemente estructuras de selección y manipulación de cadenas para personalizar la respuesta según el nombre ingresado. En la segunda parte, se pretende reforzar el uso de estructuras de repetición y arreglos al generar, almacenar y mostrar una secuencia de números pares, fomentando así el pensamiento lógico y el dominio de conceptos fundamentales de programación.

CODIGO 1:

```
public class Act_2 {  
    public static void main(String[] args) {  
        try {  
            Scanner scanner = new Scanner(System.in);  
            System.out.print("Ingresa tu nombre: ");  
            String nombre = scanner.nextLine().toLowerCase();  
            switch (nombre) {  
                case "Emiliano" -> System.out.println(" Hola, Emiliano! Bienvenido ¿Que puedo hacer por ti?.");  
                case "Angel" -> System.out.println(" Hola, Angel! Me alegra verte por aqui.");  
                case "Pedro" -> System.out.println(" Hola, Pedro! ¿Que haremos el dia de hoy?.");  
                case "Naomi" -> System.out.println(" Hola, Naomi! Qué gusto verte por aquí.");  
                default -> System.out.println("Hola, " + nombre + ". No te tengo registrado, pero bienvenido.");  
            }  
            int[] numerosPares = new int[50];  
            int numero = 2, indice = 0;  
            while (numero <= 100) {  
                numerosPares[indice] = numero;  
                numero += 2;  
                indice++;  
            }  
            System.out.println("Lista de los numeros pares del 2 al 100:");  
            indice = 0;  
            while (indice < numerosPares.length) {  
                System.out.print(numerosPares[indice] + " ");  
                indice++;  
            }  
        }  
    }  
}
```

Este programa empieza pidiendo al usuario que introduzca su nombre, utilizando la clase Scanner para capturar la entrada. Para evitar discrepancias entre mayúsculas y minúsculas, el nombre ingresado se convierte a letras minúsculas. Luego, se

emplea una estructura de control switch para verificar si el nombre coincide con algunas opciones predefinidas. Según la coincidencia, se muestra un mensaje específico. En caso de que el nombre no esté en la lista, el programa no identificará al usuario, pero aun así le permitirá continuar.

Los elementos más relevantes fueron:

- Empleo de Scanner para recibir la entrada del usuario.
- Conversión del texto a minúsculas con el método toLowerCase().
- Uso de switch para la comparación de cadenas.
- Generación de mensajes personalizados en función del nombre ingresado.

```
public class Act_2 {  
  
    public static void main(String[] args) {  
        try (  
            Scanner scanner = new Scanner(System.in)) {  
                System.out.print("Ingresa tu nombre: ");  
                String nombre = scanner.nextLine().toLowerCase();
```

También empleé la opción "case" dentro de la estructura de control para definir y gestionar múltiples usuarios, permitiéndoles acceder al programa. Además, incluí un caso "default" para manejar situaciones en las que el usuario no esté registrado, asegurando que, aunque no sea reconocido, aún pueda ingresar con una advertencia correspondiente.

```
switch (nombre) {  
    case "Emiliano" -> System.out.println(" Hola, Emiliano! Bienvenido ¿Que puedo hacer por ti?.");  
    case "Angel" -> System.out.println(" Hola, Angel! Me alegra verte por aqui.");  
    case "Pedro" -> System.out.println(" Hola, Pedro! ¿Que haremos el dia de hoy?.");  
    case "Naomi" -> System.out.println(" Hola, Naomi! Qué gusto verte por aquí.");  
    default -> System.out.println("Hola, " + nombre + ". No te tengo registrado, pero bienvenido.");  
}
```

En la segunda parte, este programa crea una lista de números enteros pares desde el 2 hasta el 100 y los guarda en un arreglo. Para ello, se emplea una estructura de repetición while. Inicialmente se establece el número en 2 y un índice en 0. Luego, mediante un bucle while, se van almacenando los números pares en el arreglo hasta llegar a 100. Posteriormente, otro bucle while recorre el arreglo y muestra en pantalla los valores almacenados.

Los puntos clave fueron:

- Uso de un arreglo (int[]) para almacenar los números pares.
- Empleo de while en lugar de for para generar la secuencia.
- Utilización de un segundo while para recorrer y mostrar los valores del arreglo.

```

int[] numerosPares = new int[50];
int numero = 2, indice = 0;
while (numero <= 100) {
    numerosPares[indice] = numero;
    numero += 2;
    indice++;
}
System.out.println("Lista de los numeros pares del 2 al 100:");
indice = 0;
while (indice < numerosPares.length) {
    System.out.print(numerosPares[indice] + " ");
    indice++;
}

```

Este código crea un arreglo de enteros con 50 posiciones para almacenar los números pares del 2 al 100. Se inicializan las variables “numero” en 2 e “indice” en 0. Luego, un bucle “while” almacena en el arreglo los números pares incrementando “numero” en 2 y “indice” en 1 hasta llegar a 100. Posteriormente, se imprime un mensaje indicando que se mostrarán los números pares, se reinicia “indice” a 0 y otro bucle “while” recorre el arreglo para imprimir en pantalla los valores almacenados, asegurando que se muestren los números pares correctamente.

CONCLUSION:

En conclusión este código en Java primero solicita al usuario que ingrese su nombre y, mediante una estructura “switch”, verifica si coincide con nombres predefinidos para mostrar un mensaje personalizado; si no coincide, muestra un mensaje de bienvenida genérico. Luego, genera un arreglo con los números pares del 2 al 100 utilizando un bucle “while”, y finalmente, otro “while” recorre el arreglo para imprimir en pantalla los números almacenados. En general, el programa combina interacción con el usuario, estructuras de control y manipulación de arreglos.