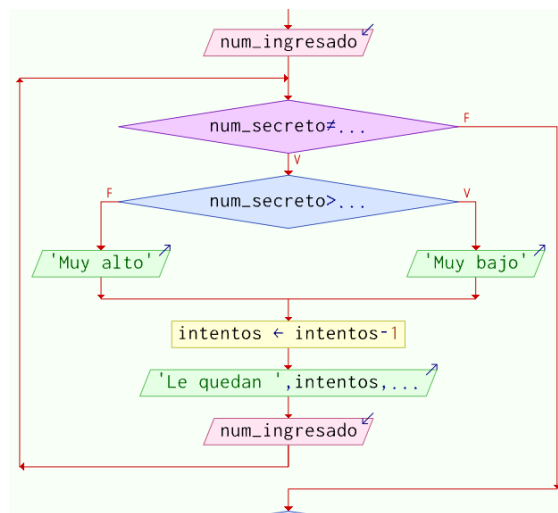


ARQUITECTURA DE PLATAFORMAS Y SERVICIOS DE TI III SEMESTRE



Puntos a tratar:

- Mientras (while)
- Hacer Mientras (do-while)
- Anidación de Estructuras repetitivas
- Desarrollo de ejercicios de aplicación.

Objetivo de la sesión:

Al concluir la sesión el estudiante estará en la capacidad de implementar las estructuras repetitivas while y do..while.

ESTRUCTURAS REPETITIVAS – while y do..while



TEMA 06

UD FUNDAMENTOS DE PROGRAMACIÓN

ING. BARRIOS QUISPE, RICHARD JHONSON

Ingeniero de sistemas

2025

Tema 06:

ESTRUCTURAS REPETITIVAS – while y do..while

1. ESTRUCTURA while

El while loop ofrece a los usuarios de Java la posibilidad de ejecutar cualquier número de indicaciones siempre que se cumpla una condición previamente definida. No se requiere una variable de recuento, por lo que la condición de terminación también puede definirse por separado.

La condición de terminación, que se anota entre paréntesis, funciona después del booleano. Esto significa que el while loop en Java ejecuta las indicaciones almacenadas mientras la condición especificada en el header sea verdadera. Si la condición de terminación cambia a "false" por primera vez, el bucle se termina. Como la condición de terminación ya está establecida en la cabecera del while loop en Java, se denomina bucle controlado por la cabecera.

Sintaxis de la estructura repetitiva Mientras ("while")		
Diagrama de Flujo	Pseudocódigo en castellano	Sintaxis en Java (while)
<pre> graph TD Condicion{condición} -- Si --> Acciones[Acciones] Acciones --> Condicion Condicion -- No --> Exit(()) style Exit fill:none,stroke:none </pre>	mientras condición hacer acción s1 . . acción sn fin_mientras	while (Condición) { <instrucción_1>; <instrucción_2>; ... <instrucción_n>; }

El flujo siempre sigue este orden:

- El programa comprueba la condición.
- Si se cumple con la condición, la aplicación Java ejecuta la indicación.
- Entonces el programa salta al principio.
- De nuevo, se comprueba la condición.
- Si ahora es "false", el programa finaliza el while loop en Java y continúa con el código.

2. ESTRUCTURA do..while

El bucle do while empieza con la palabra reservada do. Con ello, empieza un bloque de código. En dicho bloque, colocaremos todo el código del bucle, incremento o decremento incluido.

Finalmente, cuando acabemos de escribir todo el código, debemos utilizar la palabra while junto con la expresión. Esto se finaliza con un punto y coma.

El flujo siempre sigue este orden:

- El programa ingresa al bucle do..while.
- Se ejecutan todas las acciones del bloque do.



- Se evalúa la condición colocada en el while; de cumplirse regresa al inicio del bloque do, y vuelve a ejecutarse todo; en caso la condición no se cumpla; se sale del bucle y regresa al flujo del programa.

Sintaxis de la estructura repetitiva Repetir ("Repeat")		
<p><u>Diagrama de Flujo</u></p> <pre> graph TD Start(()) --> Acciones[Acciones] Acciones --> Condicion{condición} Condicion -- Verdadero --> Acciones Condicion -- Falso --> Exit(()) </pre>	<p><u>Pseudocódigo en castellano</u></p> <pre> repetir acción s1 acción s2 . . acción sn hasta_que <condición> </pre>	<p><u>Sintaxis en Java (do/while)</u></p> <pre> do{ <instrucción_1>; <instrucción_2>; ... <instrucción_n>; } while (Condición); </pre>

3. DESARROLLO DE EJERCICIOS DE APLICACIÓN DE LA ESTRUCTURA CONDICIONAL MÚLTIPLE.

EJER001: Dado un rango de números enteros, obtener la cantidad de números pares que contiene.

CODIFICACIÓN:

```

package while_dowhile;
import java.util.Scanner;

public class Busca_Pares {
    public static void main(String[] args) {
        //Variables
        int i,ni,nf,cp = 0;
        //Entrada
        Scanner teclado = new Scanner(System.in);
        System.out.print("Num. Inicial: ");
        ni = teclado.nextInt();
        System.out.print("Num. Final: ");
        nf = teclado.nextInt();
        //Proceso
        i = ni + 1;
        while(i < nf){
            if(i % 2 == 0){
                cp += 1;
            }
            i++;
        }
        //Salida
        System.out.println("");
        System.out.println("Cant. Pares: " + cp);
    }
}
  
```



EJER002: Obtener la suma los primeros “n” números naturales (del 1 hasta un límite ingresado).

CODIFICACIÓN:

```
package while_dowhile;
import java.util.Scanner;

public class Ejemplo01 {
    public static void main(String[] args) {
        //Variables
        int i=1;
        int n,s = 0;

        //Entrada
        Scanner teclado = new Scanner(System.in);
        System.out.print("Numero: ");
        n = teclado.nextInt();

        //Proceso
        while(i <= n){
            s = s + i;
            i = i + 1;
        }

        //Salida
        System.out.println("");
        System.out.println("Suma: " + s);
    }
}
```

EJER003: Hallar cuantos múltiplos de 5 hay en un rango desde el 1 hasta un límite ingresado.

CODIFICACIÓN:

```
package while_dowhile;
import java.util.Scanner;

public class Ejemplo02 {
    public static void main(String[] args) {
        //Variables
        int i,n,c = 0;

        //Entrada
        Scanner teclado = new Scanner(System.in);
        System.out.print("Numero : ");
        n = teclado.nextInt();

        //Proceso
        i = 1;
        while(i <= n){
```



```
        if(i % 5 == 0){
            c += 1;
        }
        i++;
    }

    //Salida
    System.out.println("");
    System.out.println("Cantidad: " + c);
}
}
```

EJERCICIOS DE AUTOESTUDIO PARA EL SIGUIENTE TEMA:

1. Ingrese su nombre y apellido y obtenga su nombre y apellido en mayúscula separado por una coma **Ejemplo:** RICARDO, GONZALES.
2. Al ingresar una letra determine si es una vocal.
3. Determine cuantas veces se repite una letra en una frase dada.
4. Dado una frase devolver la frase sin espacios en blanco.
5. Según las siguientes especificaciones, genere un código basado en el nombre ingresado.
 - i. **Carácter del código:** Primer carácter del nombre.
 - ii. **Carácter del código:** Tercer carácter del nombre.
 - iii. **Carácter del código:** Último carácter del nombre.
 - iv. **Carácter del código:** Cantidad de caracteres del nombre.
6. Dado una frase, devuelva la frase en forma encriptada, usando el método de convertir al siguiente carácter del ASCII, ejemplo si el carácter es A=65, devolverá B=66.

Material redactado con información proveniente de diversas fuentes.

