# **ESTRUCTURAS DE CONTROL**

Una estructura de control es un código que nos permite controlar el valor de las variables y que el programa sea dinámico.

En realidad, son preguntas que realizamos sobre el programa. Dependiendo de la respuesta, realizará un código A o un código B.

O se realiza las instrucciones A o se realizan las instrucciones B. Una de las dos.

**Sintaxis Condicional:**

if (pregunta) then

--SOLO REALIZA ESTE CODIGO SI LA RESPUESTA ES TRUE

end if;

Tenemos la posibilidad que exista una pregunta y que deseemos hacer una acción cuando devuelva TRUE (SI) o cuando devuelva FALSE (NO).

En ese caso, se utiliza el ELSE

if (pregunta = true) then

--CODIGO TRUE (A)

else

--CODIGO FALSE (B)

end if;

Tenemos una tercera posibilidad. Que deseemos seguir preguntando y realicemos acciones diferentes dependiendo del valor de la variable.

En este caso, ya no es TRUE o FALSE, sino que puede haber más posibilidades.

Se utiliza la palabra **ELSIF**

if (valor = 5) then

--CODIGO PARA CUANDO SEA 5

elsif (valor = 6) then

--CODIGO PARA CUANDO SEA 6

elsif (valor = 9) then

--CODIGO PARA CUANDO SEA 9

else

--SI NO ENTRA EN NINGUNA DE LAS PREGUNTAS ANTERIORES

end if;

**Ejemplo para comprobar si un número es positivo o negativo**

*--Pedir un número al usuario.*

*--Debemos indicar si el número es positivo, negativo*

*--Que valor voy a guardar?? NUMERO*

declare

*--las variables en oracle suelen comenzar con v\_*

v\_numero int;

begin

*--pedir al usuario un numero*

v\_numero := &numero;

*--si el numero es positivo, mostramos un mensaje y si*

*--es negativo, mostramos otro mensaje distinto.*

if (v\_numero > 0) then

*--positivo*

dbms\_output.put\_line('Numero positivo');

else

dbms\_output.put\_line('Numero negativo');

end if;

end;

**Condiciones Nulas**

En PL-SQL decíamos que era preferible inicializar las variables, siempre y cuando se pueda determinar un valor, ya que si no PL/SQL las inicializaría a NULL. Pues bien, imaginemos que tenemos que hacer un procedimiento para que diga que valor, de los dos que pasamos es mayor o menor:

DECLARE

Num1 NUMBER;

Num2 NUMBER;

RSLT VARCHAR2(7);

BEGIN

…

IF **Num1 < Num2** THEN

RSLT := ‘Yes’;

ELSE

RSLT := ‘No’;

END IF;

END;

Funcionaria si asignásemos valores como por ejemplo Num1 := 3 y Num2 := 7. Pero supongamos que Num1 := 3 y que Num2 := NULL. La condición ( 3 < NULL) sera nula, ejecutará la cláusula ELSE, asignando a RSLT el valor de ‘No’. Para solucionar esto incluimos una comprobación de nulidad:

DECLARE

Num1 NUMBER;

Num2 NUMBER;

RSLT VARCHAR2(7);

BEGIN

…

IF **Num1 IS NULL** OR

**Num2 IS NULL**  THEN

RSLT := ‘Desconocido’;

ELSIF **Num1 < Num2**  THEN

RSLT := ‘Yes’;

ELSE

RSLT := ‘No’;

END IF;

END;

**Ejemplo para comprobar si un número es positivo, negativo o cero**

**--DEBEMOS COMPROBAR SI UN NUMERO ES POSITIVO, NEGATIVO o CERO**

**declare**

**v\_numero *int*;**

**begin**

**v\_numero := &numero;**

**if (v\_numero > 0) then**

**dbms\_output.put\_line('Es positivo...' || v\_numero);**

**elsif (v\_numero = 0) then**

**dbms\_output.put\_line('Es cero!!! ' || v\_numero);**

**elsif (v\_numero < 0) then**

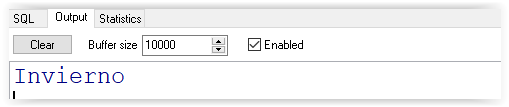
**dbms\_output.put\_line('Negativo ' || v\_numero);**

**end if;**

**dbms\_output.put\_line('Fin de programa 2');**

**end;**

**Ejemplo evaluar un número del 1 al 4 y comprobar si es una de las cuatro estaciones.**



Pedir un número al usuario del 1 al 4.

1.- Primavera

2.- Verano

3.- Otoño

4.- Invierno

Si nos da otro número, le indicamos que está mal.

**--EJEMPLO DE ESTACIONES ANUALES**

**declare**

**v\_estacion *int*;**

**begin**

**v\_estacion := &estacion;**

**if (v\_estacion = 1) then**

**dbms\_output.put\_line('Primavera');**

**elsif (v\_estacion = 2) then**

**dbms\_output.PUT\_LINE('Verano');**

**elsif (v\_estacion = 3) then**

**DBMS\_OUTPUT.PUT\_LINE('Otoño');**

**elsif (v\_estacion = 4) then**

**dbms\_output.put\_line('Invierno');**

**else**

**dbms\_output.put\_line('Valor incorrecto: ' || v\_estacion);**

**end if;**

**dbms\_output.put\_line('Fin de programa');**

**end;**

**Pediremos dos números al usuario y debemos devolver qué número es mayor.**

**declare**

**v\_num1 *int*;**

**v\_num2 *int*;**

**begin**

**v\_num1 := &num1;**

**v\_num2 := &num2;**

**if (v\_num1 > v\_num2) then**

**--MAYOR ES EL NUMERO 1**

**dbms\_output.put\_line('Número mayor ' || v\_num1);**

**else**

**dbms\_output.put\_line('Número mayor ' || v\_num2);**

**end if;**

**dbms\_output.put\_line('Fin de programa');**

**end;**