

# Módulo: Bases de Datos

## Unidad 6: Programación de Bases de Datos

### Sesión 2: Bucles en PL/SQL

---

#### Descripción:

Durante esta unidad se ha estudiado como implementar bloques o unidades de programa básicos en PL/SQL: declarando variables, haciendo uso de sentencias SQL embebidas en el código PL/SQL, llevando a cabo instrucciones condicionales, bucles, etc.

Para practicar todos estos conceptos se propone la realización de los siguientes casos prácticos:

- Implemente un bloque anónimo que permita encontrar el mayor de 3 números dados.
- Implemente un bloque anónimo que permita realizar la operación de potencia haciendo uso de un bucle LOOP.

#### Criterios de Evaluación:

- RA05\_d: Se han definido y utilizado guiones para automatizar tareas.
- RA05\_e: Se ha hecho uso de las funciones proporcionadas por el sistema gestor.

#### Objetivos:

- Identificar las diversas formas de automatizar tareas.
- Reconocer los métodos de ejecución de guiones.
- Identificar las herramientas disponibles para editar guiones.
- Definir y utilizar guiones para automatizar tareas.
- Utilizar estructuras de control de flujo.

**Recursos:**

- Acceso a Internet.
- Oracle Express.
- SQL Developer

**Conceptos a revisar previamente:**

- Realizar el estudio de los apartados de la unidad:
  - Estructuras de control de flujo.
  - Herramientas para creación de guiones. Procedimientos de ejecución.
- Realizar Ejercicios Resueltos:
  - "Utilización de los distintos tipos de bucles"

## Resolución de la práctica:

- Implemente un bloque anónimo que permita encontrar el mayor de 3 números dados.

```
SET SERVEROUTPUT ON;
DECLARE

primero NUMBER:= &introduce_el_primer_numero;
segundo NUMBER:= &introduce_el_segundo_numero;
tercero NUMBER:= &introduce_el_tercer_numero;

mayor NUMBER;

BEGIN

IF primero > segundo then
    mayor := primero;
else
    mayor := segundo;
end if;

if tercero > mayor THEN
    mayor := tercero;
END IF;

DBMS_OUTPUT.PUT_LINE('El mayor es: ' || mayor );

END;
```

- Implemente un bloque anónimo que permita realizar la operación de potencia haciendo uso de un bucle LOOP

```
/* Versión LOOP el fallo que tiene que como loop se hace al menos una vez si exponente = 0 va a fallar*/

DECLARE
base NUMBER(3):= &introduzca_base;
exponente NUMBER(3):= &introduzca_exponente;
potencia NUMBER:=1;
contador NUMBER(3):=0;

BEGIN
LOOP
    potencia := potencia * base;
    contador:= contador+1;
```

```
EXIT WHEN contador>=exponente;
END LOOP;

DBMS_OUTPUT.put_line('El resultado de elevar ' || base || ' a ' || exponente || ' es de ' || potencia );
END;
```

/\* Versión WHILE\*/

```
DECLARE
  base NUMBER(3) := &introduzca_la_base;
  exponente NUMBER(3) := &introduzca_el_exponente;
  contador NUMBER(3) := 0;
  potencia NUMBER(3) := 1;
BEGIN

  WHILE contador<exponente LOOP
    potencia := potencia*base;
    contador := contador+1;
  END LOOP;

  DBMS_OUTPUT.PUT_LINE('El resultado de elevar ' || base || ' a ' || exponente || ' es ' || potencia);
END;
```

/\*Versión FOR\*/

```
SET SERVEROUTPUT ON;
DECLARE
  base NUMBER(3):= &introduzca_base;
  exponente NUMBER(3):= &introduzca_exponente;
  potencia NUMBER:=1;

BEGIN

  FOR i IN 1..exponente LOOP
    potencia := potencia * base;
  END LOOP;

  DBMS_OUTPUT.put_line('El resultado de elevar ' || base || ' a ' || exponente || ' es de ' || potencia );

END;
```