* Crear secuencia:

CREATE SEQUENCE mi\_secuencia;

* Declarar una variable:

mi\_variable NUMBER := 10;

* Declarar una constante:

mi\_constante CONSTANT NUMBER := 5;

* Estructura IF

IF mi\_variable > 5 THEN

dbms\_output.put\_line('La variable es mayor que 5');

END IF;

* Estructura FOR LOOP

FOR i IN 1..10 LOOP

dbms\_output.put\_line(i);

END LOOP;

* Estructura WHILE LOOP

WHILE mi\_variable > 0 LOOP

dbms\_output.put\_line(mi\_variable);

mi\_variable := mi\_variable - 1;

END LOOP;

* Crear una función:

CREATE FUNCTION mi\_funcion(p\_valor1 NUMBER, p\_valor2 NUMBER) RETURN NUMBER IS

mi\_resultado NUMBER;

BEGIN

mi\_resultado := p\_valor1 + p\_valor2;

RETURN mi\_resultado;

END;

* Crear un procedimiento

CREATE PROCEDURE mi\_procedimiento(p\_nombre IN VARCHAR2) IS

BEGIN

dbms\_output.put\_line('Hola ' || p\_nombre);

END;

* Manejo de excepciones

BEGIN

SELECT 1/0 INTO mi\_variable FROM dual;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line('Error: ' || SQLERRM);

END;

* Crear o reemplazar un procedimiento

CREATE OR REPLACE PROCEDURE calcular\_edad(p\_fecha\_nacimiento IN DATE, p\_edad OUT NUMBER) AS

BEGIN

p\_edad := TRUNC(MONTHS\_BETWEEN(SYSDATE, p\_fecha\_nacimiento)/12);

END calcular\_edad;

/