Boletín 1 PL/SQL: Ejercicios básicos

EJERCICIO 1. Crea una función que sume dos números y devuelva el resultado. Realiza una llamada a dicha función desde un bloque anónimo.

```
CREATE OR REPLACE FUNCTION sumar_numeros (
        num1 IN NUMBER,
        num2 IN NUMBER
)
RETURN NUMBER
IS
        total NUMBER;
BEGIN
        total := num1 + num2;
        RETURN total;
END;
/


DECLARE
        resultado NUMBER;
BEGIN
        resultado := sumar_numeros(10, 20);
        DBMS_OUTPUT.PUT_LINE('La suma es: ' || resultado);
END;
/
```

EJERCICIO 2. Crear una función que calcule el área de un triángulo dado su base y altura. Realiza una llamada a dicha función desde un bloque anónimo.

```
CREATE OR REPLACE FUNCTION area_triangulo (
        base IN NUMBER,
        altura IN NUMBER
)
RETURN NUMBER
IS
        area NUMBER;
BEGIN
        area := (base * altura) / 2;
        RETURN area;
END;
/

DECLARE
        resultado NUMBER;
BEGIN
        resultado := area_triangulo(5, 8);
        DBMS_OUTPUT.PUT_LINE('El área del triángulo es: ' || resultado);
END;
/
```

EJERCICIO 3. Crea una función que determine si un número dado es par o impar. Realiza una llamada a dicha función desde un bloque anónimo.

```
CREATE OR REPLACE FUNCTION es_par (
        num IN NUMBER
)
RETURN VARCHAR2
IS
BEGIN
        IF MOD(num, 2) = 0 THEN
                RETURN 'Par';
        ELSE
                RETURN 'Impar';
        END IF;
END;
/

DECLARE
        numero NUMBER := 7;
        resultado VARCHAR2(10);
BEGIN
        resultado := es_par(numero);
        DBMS_OUTPUT.PUT_LINE(numero || ' es: ' || resultado);
END;
/
```

EJERCICIO 4. Crea una función que calcule el factorial de un número dado. Realiza una llamada a dicha función desde un bloque anónimo.

```
CREATE OR REPLACE FUNCTION factorial (
        num IN NUMBER
)
RETURN NUMBER
IS
        resultado NUMBER := 1;
BEGIN
        FOR i IN 1..num LOOP
                resultado := resultado * i;
        END LOOP;

        RETURN resultado;
END;
/

DECLARE
        numero NUMBER := 5;
        resultado NUMBER;
BEGIN
        resultado := factorial(numero);
        DBMS_OUTPUT.PUT_LINE('El factorial de ' || numero || ' es: ' || resultado);
END;
/
```

EJERCICIO 5. Crea una función que determine si un número dado es primo o no. Realiza una llamada a dicha función desde un bloque anónimo.

```
CREATE OR REPLACE FUNCTION es_primo (
        num IN NUMBER
)
RETURN VARCHAR2
IS
        contador NUMBER := 0;
BEGIN
        FOR i IN 2..(num/2) LOOP
                IF MOD(num, i) = 0 THEN
                        contador := contador + 1;
                END IF;
        END LOOP;

        IF contador = 0 THEN
                RETURN 'Primo';
        ELSE
                RETURN 'No primo';
        END IF;
END;
/

DECLARE
        numero NUMBER := 17;
        resultado VARCHAR2(10);
BEGIN
        resultado := es_primo(numero);
        DBMS_OUTPUT.PUT_LINE(numero || ' es: ' || resultado);
END;
/
```

EJERCICIO 6. Crea una función que determine si una cadena dada es un palíndromo o no. Realiza una llamada a dicha función desde un bloque anónimo.

```
CREATE OR REPLACE FUNCTION es_palindromo (
   cadena IN VARCHAR2
)
RETURN VARCHAR2
IS
        cadena_invertida VARCHAR2(100);
BEGIN
        FOR i IN REVERSE 1..LENGTH(cadena) LOOP
                cadena_invertida := cadena_invertida || SUBSTR(cadena, i, 1);
        END LOOP;

        IF cadena = cadena_invertida THEN
                RETURN 'Palíndromo';
        ELSE
                RETURN 'No palíndromo';
        END IF;
END;
/

DECLARE
        texto VARCHAR2(100) := 'abccba';
        resultado VARCHAR2(15);
BEGIN
        resultado := es_palindromo(texto);
        DBMS_OUTPUT.PUT_LINE('La cadena "' || texto || '" es: ' || resultado);
END;
/
```