

Práctica de PAQUETES

1. Crear un paquete denominado REGIONES que tenga los siguientes componentes:

• PROCEDIMIENTOS:

- - ALTA_REGION, con parámetro de código y nombre Región. Debe devolver un error si la región ya existe. Inserta una nueva región en la tabla. Debe llamar a la función EXISTE_REGION para controlarlo.
- - BAJA_REGION, con parámetro de código de región y que debe borrar una región. Debe generar un error si la región no existe, Debe llamar a la función EXISTE_REGION para controlarlo
- - MOD_REGION: se le pasa un código y el nuevo nombre de la región Debe modificar el nombre de una región ya existente. Debe generar un error si la región no existe, Debe llamar a la función EXISTE_REGION para controlarlo

• FUNCIONES

- CON_REGION. Se le pasa un código de región y devuelve el nombre
- EXISTE_REGION. Devuelve verdadero si la región existe. Se usa en los procedimientos y por tanto es PRIVADA, no debe aparecer en la especificación del paquete

2. Crear un paquete denominado NOMINA que tenga sobrecargado la función CALCULAR_NOMINA de la siguiente forma:

- CALCULAR_NOMINA(NUMBER): se calcula el salario del empleado restando un 15% de IRPF.
- CALCULAR_NOMINA(NUMBER,NUMBER): el segundo parámetro es el porcentaje a aplicar. Se calcula el salario del empleado restando ese porcentaje al salario
- CALCULAR_NOMINA(NUMBER,NUMBER,CHAR): el segundo parámetro es el porcentaje a aplicar, el tercero vale 'V' . Se calcula el salario del empleado aumentando la comisión que le pertenece y restando ese porcentaje al salario siempre y cuando el empleado tenga comisión.

- Se cambia el IRPF POR ISR

Ejemplo 2:

- Ingresos totales durante el año fiscal: Q100,000
- Deducciones autorizadas: Q10,000
- Base imponible: $Q100,000 - Q10,000 = Q90,000$
- Tasa del ISR: 15%
- Impuesto a pagar: $Q90,000 \times 15\% = Q13,500$

```

-- PRIMER PAQUETE
CREATE OR REPLACE PACKAGE REGIONES
IS
    PROCEDURE ALTA_REGION(P_ID_REGION IN NUMBER, P_NOMBRE_REGION IN VARCHAR2);
    PROCEDURE BAJA_REGION (P_ID_REGION IN NUMBER);
    PROCEDURE MOD_REGION (P_ID_REGION IN NUMBER, P_NOMBRE_REGION IN VARCHAR2);
END;

-- *****

CREATE OR REPLACE PACKAGE BODY REGIONES
IS
    --primera funcion
    FUNCTION CON_REGION (P_ID_REGION NUMBER) RETURN VARCHAR2
    IS

```

Hoja de Trabajo Generador de Consultas

```

--primera funcion
    FUNCTION CON_REGION (P_ID_REGION NUMBER) RETURN VARCHAR2
    IS
        NOMBRE_REGION VARCHAR(25);
        BEGIN
            SELECT REGION_NAME INTO NOMBRE_REGION FROM REGIONS WHERE REGION_ID = P_ID_REGION;
            IF NOMBRE_REGION IS NULL THEN
                RETURN 'Nombre de region es null';
            END IF;

            RETURN NOMBRE_REGION;
        EXCEPTION
            WHEN NO_DATA_FOUND THEN
                RETURN 'error Región no encontrada';
        END CON_REGION;

    --
        segunda funcion
    FUNCTION EXISTE_REGION(P_ID_REGION NUMBER) RETURN BOOLEAN
    IS
        EXISTE NUMBER;
        BEGIN
            SELECT COUNT(*) INTO EXISTE FROM REGIONS WHERE REGION_ID = P_ID_REGION;
            DBMS_OUTPUT.PUT_LINE('EXISTE: ' || EXISTE);

```

```
SELECT COUNT(*) INTO EXISTE FROM REGIONS WHERE REGION_ID = P_ID_REGIIN;
-- DBMS_OUTPUT.PUT_LINE('EXISTE ' || EXISTE );
IF EXISTE > 0 THEN
    RETURN TRUE;
ELSE
    RETURN FALSE;
END IF;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN FALSE;
END EXISTE_REGION;

-- PRIMER PROCEDIMIENTO
PROCEDURE ALTA_REGION (P_ID_REGIIN IN NUMBER, P_NOMBRE_REGION IN VARCHAR2)
IS
BEGIN
    IF EXISTE_REGION(P_ID_REGIIN) THEN
        -- DBMS_OUTPUT.PUT_LINE('EXISTE');
        RAISE_APPLICATION_ERROR(-20001,'ERROR: LA REGION YA EXISTE!');
    ELSE
        INSERT INTO REGIONS VALUES (P_ID_REGIIN, P_NOMBRE_REGION);
        COMMIT;
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        RAISE;
END;
```

```
RAISE;

END;

-- SEGUNDO PROCEDIMIENTO
PROCEDURE BAJA_REGION (P_ID_REGIIN IN NUMBER)
IS
BEGIN
    IF EXISTE_REGION(P_ID_REGIIN) = FALSE THEN
        -- DBMS_OUTPUT.PUT_LINE('EXISTE');
        RAISE_APPLICATION_ERROR(-20001,'ERROR: LA REGION NO EXISTE!');
    ELSE
        DELETE FROM REGIONS WHERE REGION_ID = P_ID_REGIIN;
        COMMIT;
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        RAISE;
END;

--TERCER PROC
PROCEDURE MOD_REGION (P_ID_REGIIN IN NUMBER, P_NOMBRE_REGION IN VARCHAR2)
IS
BEGIN
    IF EXISTE_REGION(P_ID_REGIIN) = FALSE THEN
        RAISE_APPLICATION_ERROR(-20001,'ERROR: LA REGION NO EXISTE!');
    ELSE
```

```
Hoja de Trabajo  Generador de Consultas

ELSE
    UPDATE REGIONS SET REGION_NAME = P_NOMBRE_REGION WHERE REGION_ID = P_ID_REGION ;
    COMMIT;
END IF;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        RAISE;

END;

END REGIONES;

-- SEGUNDO PAQUETE
-- *****
CREATE OR REPLACE PACKAGE NOMINA
IS
    FUNCTION CALCULAR_NOMINA (P_EMPLOYEE_ID NUMBER) RETURN NUMBER;
    FUNCTION CALCULAR_NOMINA (P_EMPLOYEE_ID NUMBER, P_PORCENTAJE NUMBER) RETURN NUMBER;
    FUNCTION CALCULAR_NOMINA (P_EMPLOYEE_ID NUMBER, P_PORCENTAJE NUMBER, P_V_CHAR) RETURN NUMBER;
END;
```

```
Hoja de Trabajo  Generador de Consultas

END;

-- *****

CREATE OR REPLACE PACKAGE BODY NOMINA
IS
    -- PRIMER FUNCION
    FUNCTION CALCULAR_NOMINA (P_EMPLOYEE_ID NUMBER) RETURN NUMBER
    IS
        SALARIO  NUMBER(8,2);
        BEGIN
            SELECT SALARY INTO SALARIO FROM EMPLOYEES WHERE EMPLOYEE_ID = P_EMPLOYEE_ID;

            IF SALARIO IS NOT NULL THEN
                SALARIO := SALARIO /1.15 ;
                RETURN SALARIO;
            END IF;
        EXCEPTION
            WHEN NO_DATA_FOUND THEN
                RETURN -9999;

        END CALCULAR_NOMINA;

    -- PRIMERA SOBRECARGA
    FUNCTION CALCULAR_NOMINA (P_EMPLOYEE_ID NUMBER, P_PORCENTAJE NUMBER) RETURN NUMBER
    IS
```

```

FUNCTION CALCULAR_NOMINA (P_EMPLOYEE_ID NUMBER, P_PORCENTAJE NUMBER) RETURN NUMBER
IS
SALARIO NUMBER(8,2);
PORCENTAJE NUMBER;
BEGIN
SELECT SALARY INTO SALARIO FROM EMPLOYEES WHERE EMPLOYEE_ID = P_EMPLOYEE_ID;

IF SALARIO IS NOT NULL THEN
    PORCENTAJE := (P_PORCENTAJE/100)+1;
    SALARIO := SALARIO / PORCENTAJE ;
    RETURN SALARIO;
END IF;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN -9999;

END CALCULAR_NOMINA;

-- SEGUNDA SOBRECARGA
FUNCTION CALCULAR_NOMINA (P_EMPLOYEE_ID NUMBER, P_PORCENTAJE NUMBER, P_V CHAR) RETURN NUMBER
IS
SALARIO NUMBER(8,2);
PORCENTAJE NUMBER;
PORCENTAJE_COMISION NUMBER ;
BEGIN
SELECT SALARY, COMMISSION_PCT INTO SALARIO,PORCENTAJE_COMISION FROM EMPLOYEES WHERE EMPLOYEE_ID = P_EMPLOYEE_ID;
IF PORCENTAJE_COMISION IS NOT NULL AND UPPER(P_V) = 'V' THEN
    IF SALARIO IS NOT NULL THEN

```

Hoja de Trabajo Generador de Consultas

```

        IF SALARIO IS NOT NULL THEN
            SALARIO := SALARIO * (PORCENTAJE_COMISION +1);
            PORCENTAJE := (P_PORCENTAJE/100)+1;
            SALARIO := SALARIO / PORCENTAJE ;
            RETURN SALARIO;
        END IF;
    ELSIF PORCENTAJE_COMISION IS NULL AND UPPER(P_V) = 'V' THEN
        IF SALARIO IS NOT NULL THEN
            PORCENTAJE := (P_PORCENTAJE/100)+1;
            SALARIO := SALARIO / PORCENTAJE ;
            RETURN SALARIO;
        END IF;
    END IF;
    RETURN -9999;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN -9999;
END CALCULAR_NOMINA;

END NOMINA;

```

USO DE PROC 1

```
SET SERVEROUTPUT ON
DECLARE
V_SALARIO NUMBER ;
BEGIN
    REGIONES.ALTA_REGION(5,'REGION 5');
    -- REGIONES.BAJA_REGION(5);
    -- REGIONES.MOD_REGION(5, 'REGION 5 MOD');
    -- V_SALARIO := NOMINA.CALCULAR_NOMINA(121);
    -- V_SALARIO := NOMINA.CALCULAR_NOMINA(121,15);
    -- V_SALARIO := NOMINA.CALCULAR_NOMINA(121,15,'V');
    -- V_SALARIO := NOMINA.CALCULAR_NOMINA(150,15,'V');

    /* IF  V_SALARIO = -9999 THEN
        DBMS_OUTPUT.PUT_LINE('OCURRIO UN ERROR AL CALCULAR EL SALARIO! ');
    ELSE
        DBMS_OUTPUT.PUT_LINE('EL SALARIO CALCULADO ES: ' ||  V_SALARIO );
    END IF;*/
END;
```

Salida de Script x

Tarea terminada en 0.083 segundos

Procedimiento PL/SQL terminado correctamente.

USO DE PROC 2

```
SET SERVEROUTPUT ON
DECLARE
V_SALARIO NUMBER ;
BEGIN
    -- REGIONES.ALTA_REGION(5,'REGION 5');
    REGIONES.BAJA_REGION(5);
    -- REGIONES.MOD_REGION(5, 'REGION 5 MOD');
    -- V_SALARIO := NOMINA.CALCULAR_NOMINA(121);
    -- V_SALARIO := NOMINA.CALCULAR_NOMINA(121,15);
    -- V_SALARIO := NOMINA.CALCULAR_NOMINA(121,15,'V');
    -- V_SALARIO := NOMINA.CALCULAR_NOMINA(150,15,'V');

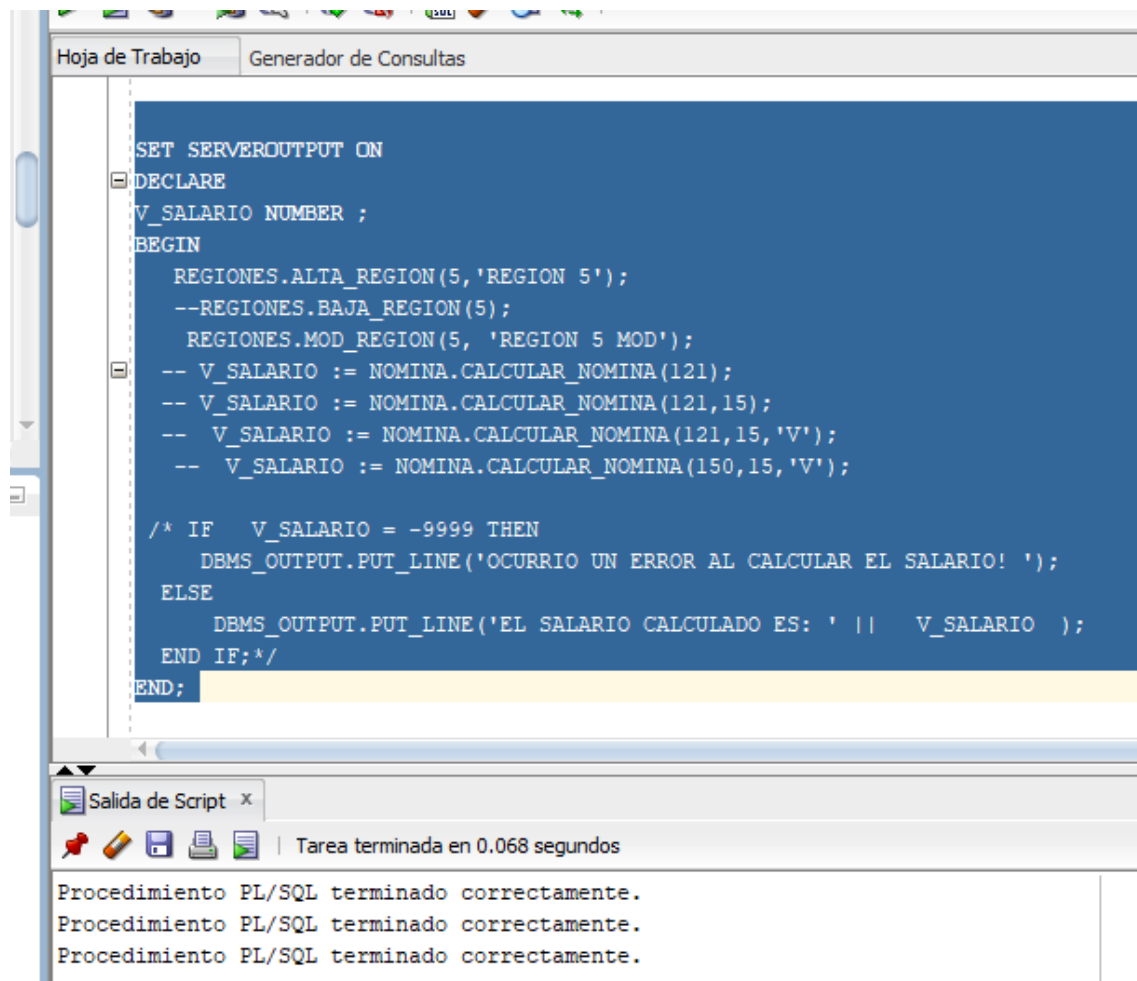
    /* IF  V_SALARIO = -9999 THEN
        DBMS_OUTPUT.PUT_LINE('OCURRIO UN ERROR AL CALCULAR EL SALARIO! ');
    ELSE
        DBMS_OUTPUT.PUT_LINE('EL SALARIO CALCULADO ES: ' ||  V_SALARIO );
    END IF;*/
END;
```

Salida de Script x

Tarea terminada en 0.077 segundos

Procedimiento PL/SQL terminado correctamente.
Procedimiento PL/SQL terminado correctamente.

USO PROC 3



The screenshot shows the SQL Developer interface with the 'Hoja de Trabajo' (Worksheet) tab active. The script editor contains the following PL/SQL code:

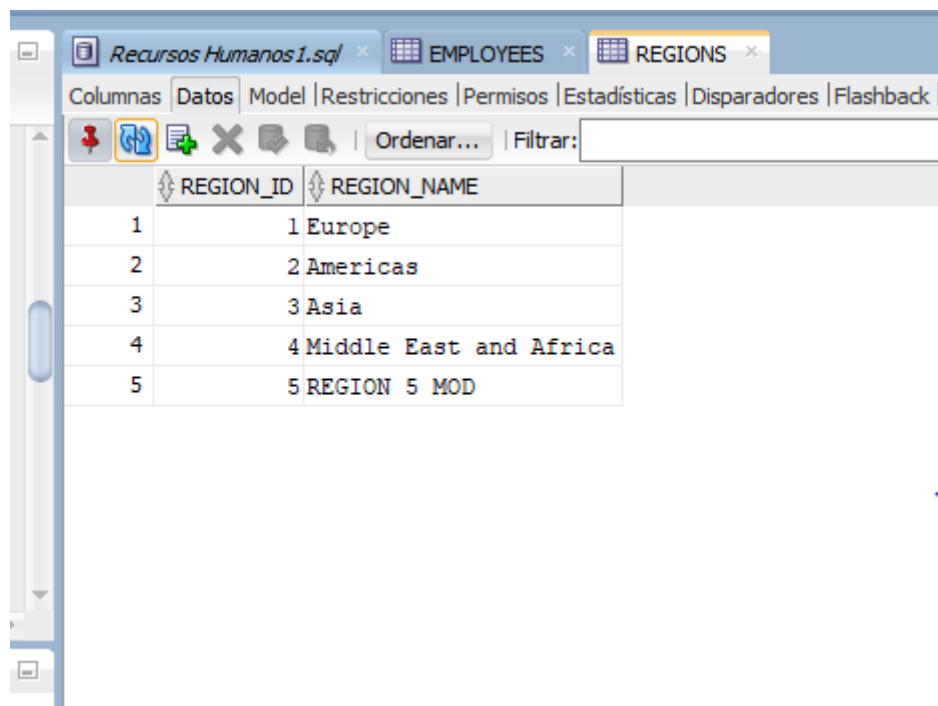
```
SET SERVEROUTPUT ON
DECLARE
V_SALARIO NUMBER ;
BEGIN
    REGIONES.ALTA_REGION(5, 'REGION 5');
    --REGIONES.BAJA_REGION(5);
    REGIONES.MOD_REGION(5, 'REGION 5 MOD');
    -- V_SALARIO := NOMINA.CALCULAR_NOMINA(121);
    -- V_SALARIO := NOMINA.CALCULAR_NOMINA(121,15);
    -- V_SALARIO := NOMINA.CALCULAR_NOMINA(121,15,'V');
    -- V_SALARIO := NOMINA.CALCULAR_NOMINA(150,15,'V');

    /* IF V_SALARIO = -9999 THEN
        DBMS_OUTPUT.PUT_LINE('OCURRIO UN ERROR AL CALCULAR EL SALARIO! ');
    ELSE
        DBMS_OUTPUT.PUT_LINE('EL SALARIO CALCULADO ES: ' || V_SALARIO );
    END IF;*/
END;
```

Below the script editor, the 'Salida de Script' (Script Output) window shows the execution results:

Tarea terminada en 0.068 segundos

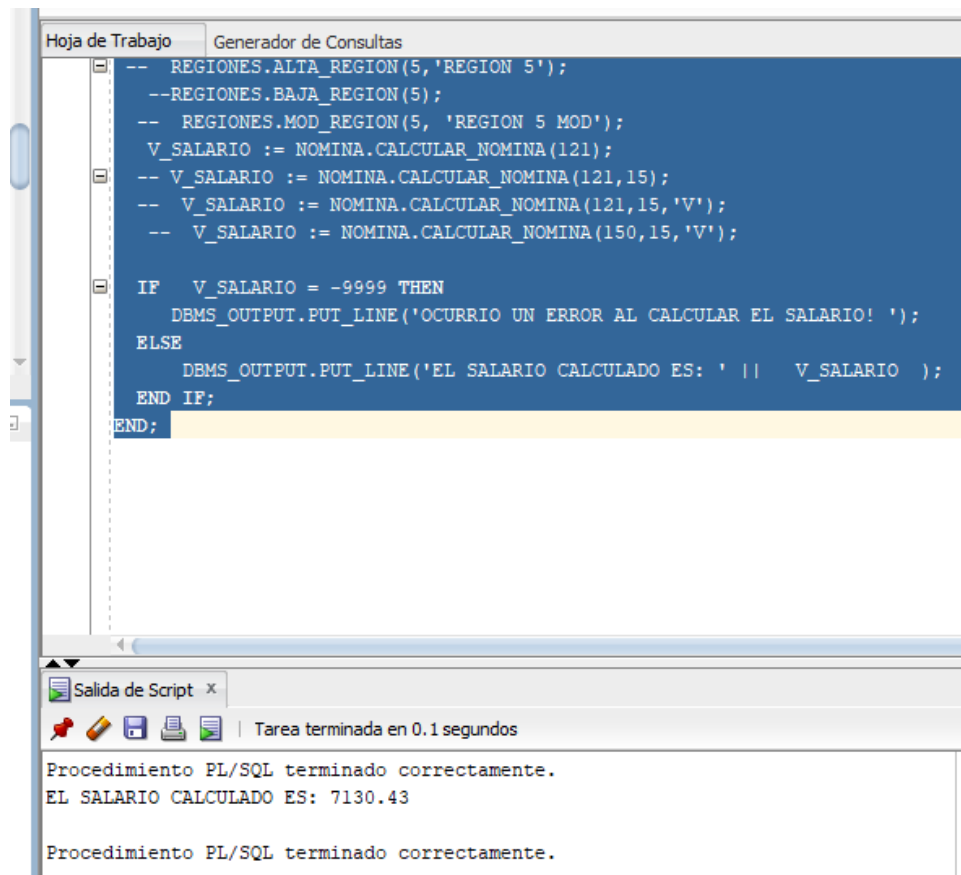
Procedimiento PL/SQL terminado correctamente.
Procedimiento PL/SQL terminado correctamente.
Procedimiento PL/SQL terminado correctamente.



The screenshot shows the SQL Developer interface with the 'Recursos Humanos1.sql' script open. The 'EMPLOYEES' and 'REGIONS' tables are visible in the top right. The 'REGIONS' table is selected, and the 'Columnas' (Columns) tab is active. The table data is displayed as follows:

REGION_ID	REGION_NAME
1	Europe
2	Americas
3	Asia
4	Middle East and Africa
5	REGION 5 MOD

USO FUNCION CALCULAR_NOMINA



The screenshot displays the Oracle SQL Developer environment. The top pane, titled 'Hoja de Trabajo' and 'Generador de Consultas', contains a PL/SQL script. The script defines a function `CALCULAR_NOMINA` and uses it to calculate a salary. The bottom pane, titled 'Salida de Script', shows the execution results, indicating that the procedure completed successfully and displaying the calculated salary value.

```
-- REGIONES.ALTA_REGION(5, 'REGION 5');
--REGIONES.BAJA_REGION(5);
-- REGIONES.MOD_REGION(5, 'REGION 5 MOD');
V_SALARIO := NOMINA.CALCULAR_NOMINA(121);
-- V_SALARIO := NOMINA.CALCULAR_NOMINA(121,15);
-- V_SALARIO := NOMINA.CALCULAR_NOMINA(121,15,'V');
-- V_SALARIO := NOMINA.CALCULAR_NOMINA(150,15,'V');

IF V_SALARIO = -9999 THEN
  DBMS_OUTPUT.PUT_LINE('OCURRIO UN ERROR AL CALCULAR EL SALARIO! ');
ELSE
  DBMS_OUTPUT.PUT_LINE('EL SALARIO CALCULADO ES: ' || V_SALARIO );
END IF;
END;
```

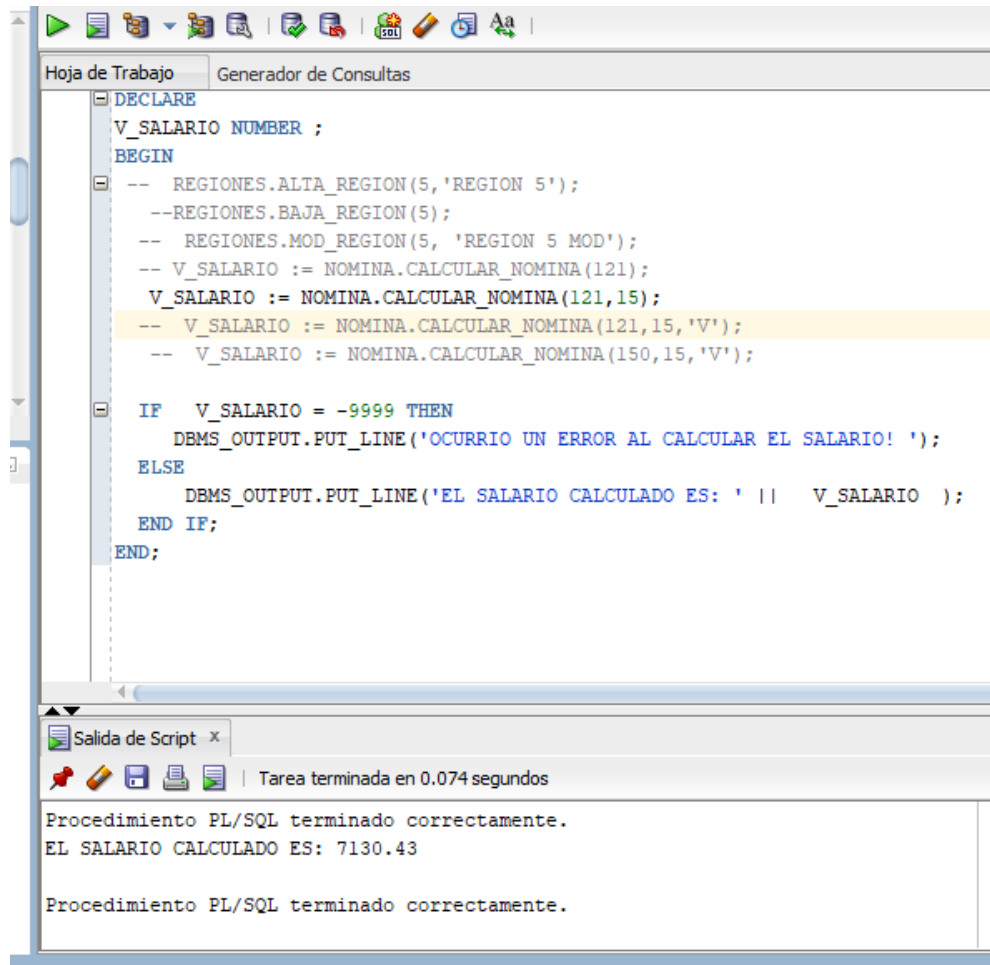
Salida de Script x

Tarea terminada en 0.1 segundos

Procedimiento PL/SQL terminado correctamente.
EL SALARIO CALCULADO ES: 7130.43

Procedimiento PL/SQL terminado correctamente.

USO SOBRECARGA 1



The screenshot displays the Oracle SQL Developer environment. The main window, titled 'Hoja de Trabajo' and 'Generador de Consultas', contains a PL/SQL script. The script declares a variable `V_SALARIO` of type `NUMBER` and begins a block. It includes several commented-out lines for region management and one active line for calculating the salary: `V_SALARIO := NOMINA.CALCULAR_NOMINA(121,15);`. Below this, there are more commented-out lines for different salary calculations. An `IF` statement checks if `V_SALARIO` equals -9999; if true, it outputs an error message, otherwise, it outputs the calculated salary. The script ends with `END;`. At the bottom, a 'Salida de Script' window shows the execution results: 'Procedimiento PL/SQL terminado correctamente.' followed by 'EL SALARIO CALCULADO ES: 7130.43' and another success message.

```
DECLARE
V_SALARIO NUMBER ;
BEGIN
-- REGIONES.ALTA_REGION(5,'REGION 5');
--REGIONES.BAJA_REGION(5);
-- REGIONES.MOD_REGION(5, 'REGION 5 MOD');
-- V_SALARIO := NOMINA.CALCULAR_NOMINA(121);
V_SALARIO := NOMINA.CALCULAR_NOMINA(121,15);
-- V_SALARIO := NOMINA.CALCULAR_NOMINA(121,15,'V');
-- V_SALARIO := NOMINA.CALCULAR_NOMINA(150,15,'V');

IF V_SALARIO = -9999 THEN
    DBMS_OUTPUT.PUT_LINE('OCURRIO UN ERROR AL CALCULAR EL SALARIO! ');
ELSE
    DBMS_OUTPUT.PUT_LINE('EL SALARIO CALCULADO ES: ' || V_SALARIO );
END IF;
END;
```

Salida de Script x

Tarea terminada en 0.074 segundos

Procedimiento PL/SQL terminado correctamente.
EL SALARIO CALCULADO ES: 7130.43

Procedimiento PL/SQL terminado correctamente.

USO SOBRECARGA 3, CON COMISION Y SIN COMISION





Hoja de Trabajo

Generador de Consultas

```
BEGIN
-- REGIONES.ALTA_REGION(5, 'REGION 5');
--REGIONES.BAJA_REGION(5);
-- REGIONES.MOD_REGION(5, 'REGION 5 MOD');
-- V_SALARIO := NOMINA.CALCULAR_NOMINA(121);
-- V_SALARIO := NOMINA.CALCULAR_NOMINA(121,15);
-- V_SALARIO := NOMINA.CALCULAR_NOMINA(121,15,'V');
  V_SALARIO := NOMINA.CALCULAR_NOMINA(150,15,'V');

  IF V_SALARIO = -9999 THEN
    DBMS_OUTPUT.PUT_LINE('OCURRIO UN ERROR AL CALCULAR EL SALARIO! ');
  ELSE
    DBMS_OUTPUT.PUT_LINE('EL SALARIO CALCULADO ES: ' || V_SALARIO );
  END IF;
END;
```

Salida de Script x

 | Tarea terminada en 0.068 segundos

```
EL SALARIO CALCULADO ES: 7130.43

Procedimiento PL/SQL terminado correctamente.
EL SALARIO CALCULADO ES: 11304.35

Procedimiento PL/SQL terminado correctamente.
```