

Funciones y procedimientos (I)

- 1. Crea un procedimiento que reciba dos números y visualice su suma. Ejecuta el procedimiento usando las funcionalidades de SQL Developer.**

```
CREATE OR REPLACE PROCEDURE sumarDosNumeros (V_NUM1 IN NUMBER, V_NUM2 IN NUMBER,
V_RESULTADO OUT NUMBER) IS
```

```
BEGIN
```

```
    V_RESULTADO := V_NUM1 + V_NUM2;
```

```
END sumarDosNumeros;
```

```
DECLARE
```

```
    V_NUMERO1 NUMBER;
```

```
    V_NUMERO2 NUMBER;
```

```
    V_SOLUCION NUMBER;
```

```
-- bloque anónimo
```

```
BEGIN
```

```
    V_NUMERO1 := 1;
```

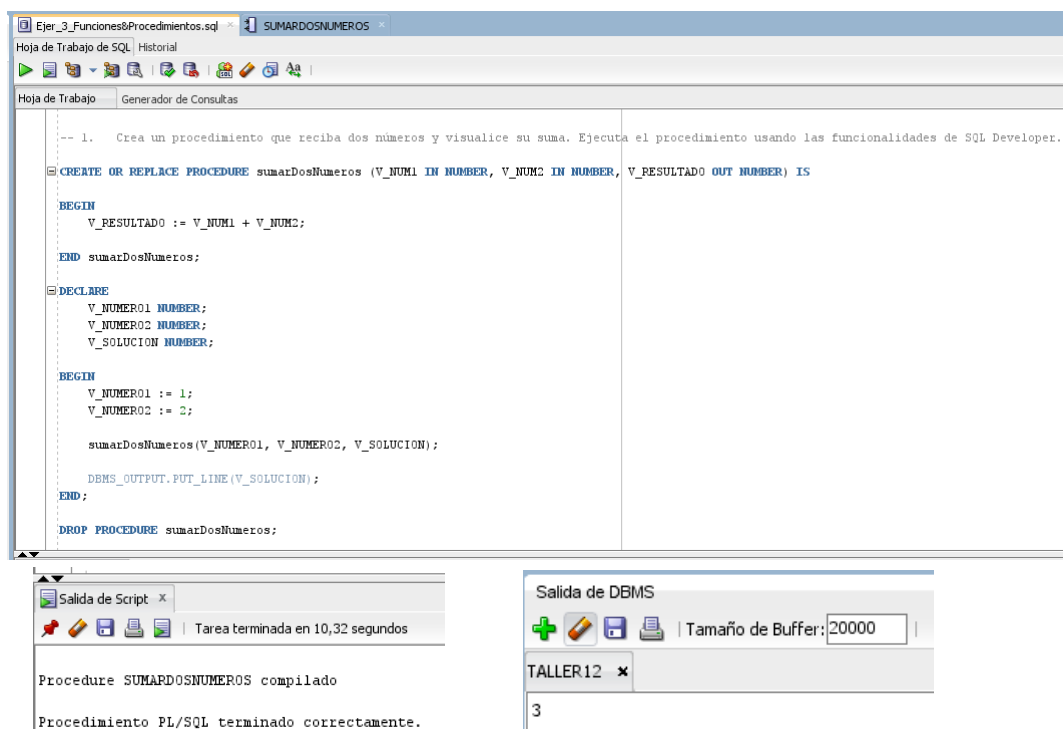
```
    V_NUMERO2 := 2;
```

```
    sumarDosNumeros(V_NUMERO1, V_NUMERO2, V_SOLUCION);
```

```
    DBMS_OUTPUT.PUT_LINE(V_SOLUCION);
```

```
END;
```

```
DROP PROCEDURE sumarDosNumeros;
```



2. Crea un procedimiento que reciba dos números y devuelva su suma. Crea un bloque anónimo para probar el procedimiento.

```
CREATE OR REPLACE PROCEDURE sumarDosNumeros (V_NUM1 IN NUMBER, V_NUM2 IN NUMBER,
V_RESULTADO OUT NUMBER)
```

```
IS
```

```
BEGIN
```

```
    V_RESULTADO := V_NUM1 + V_NUM2;
```

```
END;
```

```
-- bloque anónimo
```

```
DECLARE
```

```
    V_NUMERO1 NUMBER;
```

```
    V_NUMERO2 NUMBER;
```

```
    V_SOLUCION NUMBER;
```

```
BEGIN
```

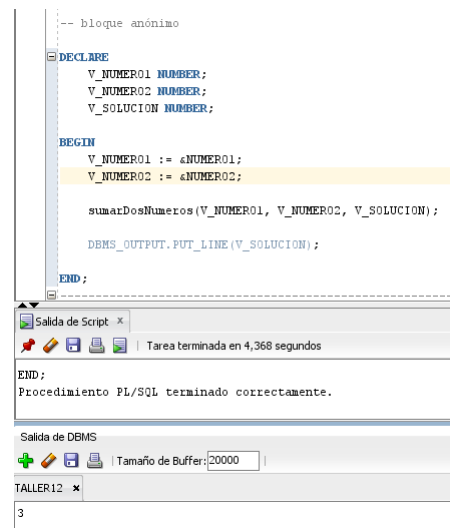
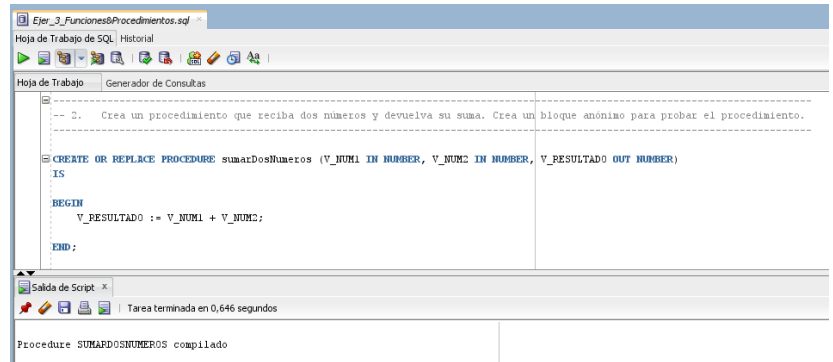
```
    V_NUMERO1 := &NUMERO1;
```

```
    V_NUMERO2 := &NUMERO2;
```

```
    sumarDosNumeros(V_NUMERO1, V_NUMERO2, V_SOLUCION);
```

```
    DBMS_OUTPUT.PUT_LINE(V_SOLUCION);
```

```
END;
```



3. Crea una función que reciba dos números y devuelva su suma. Crea un bloque anónimo para probar el procedimiento.

```
CREATE OR REPLACE FUNCTION sumarDosNumeros_FUNCION (V_NUM1 IN NUMBER, V_NUM2 IN
NUMBER)
```

```
RETURN NUMBER
```

```
IS
```

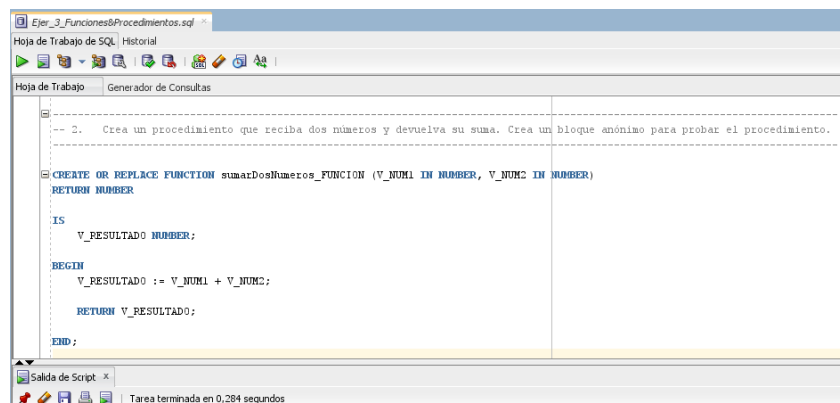
```
    V_RESULTADO NUMBER;
```

```
BEGIN
```

```
    V_RESULTADO := V_NUM1 + V_NUM2;
```

```
    RETURN V_RESULTADO;
```

```
END;
```



En realidad, se trata del apartado 3

-- bloque anónimo

```

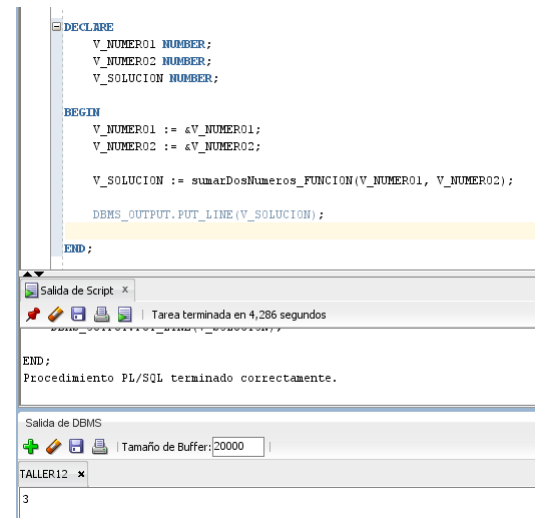
DECLARE
  V_NUMERO1 NUMBER;
  V_NUMERO2 NUMBER;
  V_SOLUCION NUMBER;

BEGIN
  V_NUMERO1 := &V_NUMERO1;
  V_NUMERO2 := &V_NUMERO2;

  V_SOLUCION := sumarDosNumeros_FUNCION(V_NUMERO1,
  V_NUMERO2);

  DBMS_OUTPUT.PUT_LINE(V_SOLUCION);

END;
```



Inserté un 1 para V_NUMERO1, y un 2 para V_NUMERO2

4. Crea un procedimiento que devuelva la suma del salario de todos los mecánicos. Crea un bloque anónimo para probar el procedimiento.

```

CREATE OR REPLACE PROCEDURE P_sumaSalarioMecanicos (V_SUMA OUT MECANICOS.SALARIO%TYPE)
-- aquí no hace falta que declare "V_SALARIO IN MECANICOS.SALARIO%TYPE"
IS
```

```

  CURSOR C_SALARIOS IS
  SELECT SUM(SALARIO)
  FROM MECANICOS;
```

```

BEGIN
  OPEN C_SALARIOS;
  FETCH C_SALARIOS INTO V_SUMA;
  CLOSE C_SALARIOS;
```

```

END P_sumaSalarioMecanicos;
```

-- bloque anónimo

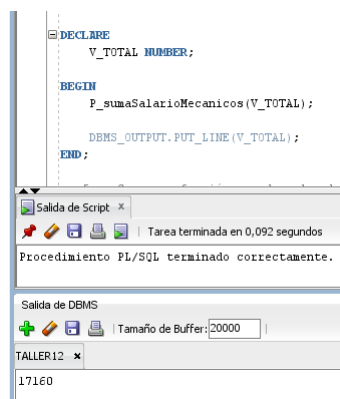
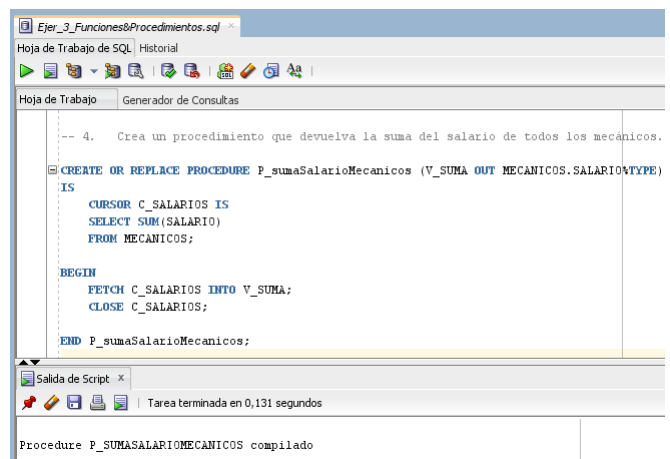
```

DECLARE
  V_TOTAL NUMBER;

BEGIN
  P_sumaSalarioMecanicos(V_TOTAL);

  DBMS_OUTPUT.PUT_LINE(V_TOTAL);

END;
```



5. Crea una función que devuelva la suma del salario de todos los mecánicos de un determinado puesto. Crea un bloque anónimo para probar la función.

```
CREATE OR REPLACE FUNCTION F_sumaSalarioMecanicos(V_PUESTO IN MECANICOS.PUESTO%TYPE)
RETURN NUMBER
IS
```

```
    CURSOR C_SALARIOS IS
    SELECT SUM(SALARIO)
    FROM MECANICOS
    WHERE PUESTO = V_PUESTO;
```

```
    V_SUMA NUMBER;
```

```
BEGIN
```

```
    OPEN C_SALARIOS;
    FETCH C_SALARIOS INTO V_SUMA;
    CLOSE C_SALARIOS;
```

```
    RETURN V_SUMA;
```

```
END;
```

```
-- bloque anónimo
```

```
DECLARE
```

```
    V_RESULTADO NUMBER;
    V_PUESTO1 MECANICOS.PUESTO%TYPE;
```

```
BEGIN
```

```
    V_PUESTO1 := '&PUESTO';
    V_RESULTADO := F_sumaSalarioMecanicos(V_PUESTO1);
```

```
    DBMS_OUTPUT.PUT_LINE('La suma de todos los salarios
es: ' || V_RESULTADO);
```

```
END;
```

The screenshot shows the SQL Developer interface with a script titled 'Ejer_3_Funciones y Procedimientos.sql'. The script contains the following SQL code:

```
-- 5. Crea una función que devuelva la suma del salario de todos los mecánicos de un determinado puesto.

CREATE OR REPLACE FUNCTION F_sumaSalarioMecanicos(V_PUESTO IN MECANICOS.PUESTO%TYPE) RETURN NUMBER
IS
    CURSOR C_SALARIOS IS
    SELECT SUM(SALARIO)
    FROM MECANICOS
    WHERE PUESTO = V_PUESTO;

    V_SUMA NUMBER;

BEGIN
    OPEN C_SALARIOS;
    FETCH C_SALARIOS INTO V_SUMA;
    CLOSE C_SALARIOS;

    RETURN V_SUMA;

END;
```

The status bar at the bottom indicates 'Function F_SUMASALARIOMECHANICOS compilado'.

The screenshot shows the execution of the anonymous block. The script contains the following SQL code:

```
DECLARE
    V_RESULTADO NUMBER;
    V_PUESTO1 MECANICOS.PUESTO%TYPE;

BEGIN
    V_PUESTO1 := '&PUESTO';
    V_RESULTADO := F_sumaSalarioMecanicos(V_PUESTO1);

    DBMS_OUTPUT.PUT_LINE('La suma de todos los salarios es: ' || V_RESULTADO);

END;
```

The status bar at the bottom indicates 'Tarea terminada en 8,785 segundos'. The output window shows the result: 'La suma de todos los salarios es: 2710'.

6. Crea una función que reciba el DNI de un mecánico y retorne su nombre. Prueba el procedimiento usando las funcionalidades de SQL Developer.

```
CREATE OR REPLACE FUNCTION F_retornarNombrePorDNI (V_DNI IN MECANICOS.DNI%TYPE) RETURN
MECANICOS.NOMBRE%TYPE
```

```
IS
```

```
CURSOR C_DNI IS
SELECT NOMBRE
FROM MECANICOS
WHERE DNI = V_DNI;

V_NOMBRE MECANICOS.NOMBRE%TYPE;
```

```
BEGIN
```

```
OPEN C_DNI;
FETCH C_DNI INTO V_NOMBRE;
CLOSE C_DNI;
```

```
RETURN V_NOMBRE;
```

```
END;
```

```
-- bloque anónimo
```

```
DECLARE
```

```
V_NOMBRE1 MECANICOS.NOMBRE%TYPE;
V_DNI1 MECANICOS.DNI%TYPE;
```

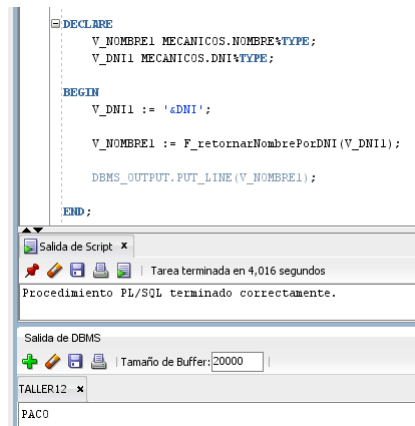
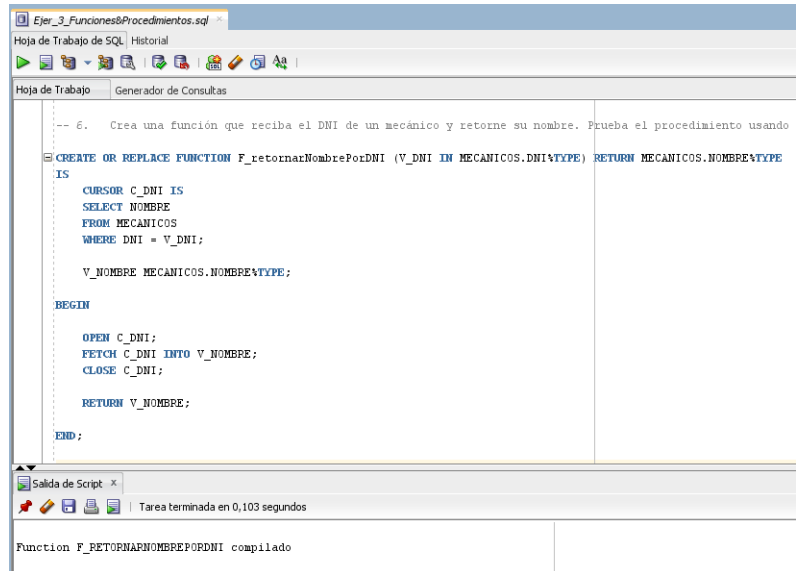
```
BEGIN
```

```
V_DNI1 := '&DNI';

V_NOMBRE1 := F_retornarNombrePorDNI(V_DNI1);

DBMS_OUTPUT.PUT_LINE(V_NOMBRE1);
```

```
END;
```



7. Crea una función que reciba el DNI de un mecánico y retorne su salario. Prueba el procedimiento usando las funcionalidades de SQL Developer.

```
CREATE OR REPLACE FUNCTION F_retornarSalarioPorDNI (V_DNI IN MECANICOS.DNI%TYPE) RETURN
MECANICOS.SALARIO%TYPE
IS
```

```
    CURSOR C_DNI IS
    SELECT SALARIO
    FROM MECANICOS
    WHERE DNI = V_DNI;

    V_SALARIO MECANICOS.SALARIO%TYPE;
```

```
BEGIN
```

```
    OPEN C_DNI;
    FETCH C_DNI INTO V_SALARIO;
    CLOSE C_DNI;
```

```
    RETURN V_SALARIO;
```

```
END;
```

```
-- bloque anónimo
```

```
DECLARE
```

```
    V_SALARIO1 MECANICOS.SALARIO%TYPE;
    V_DNI1 MECANICOS.DNI%TYPE;
```

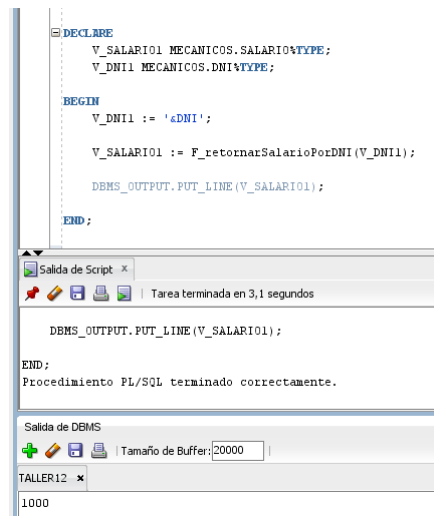
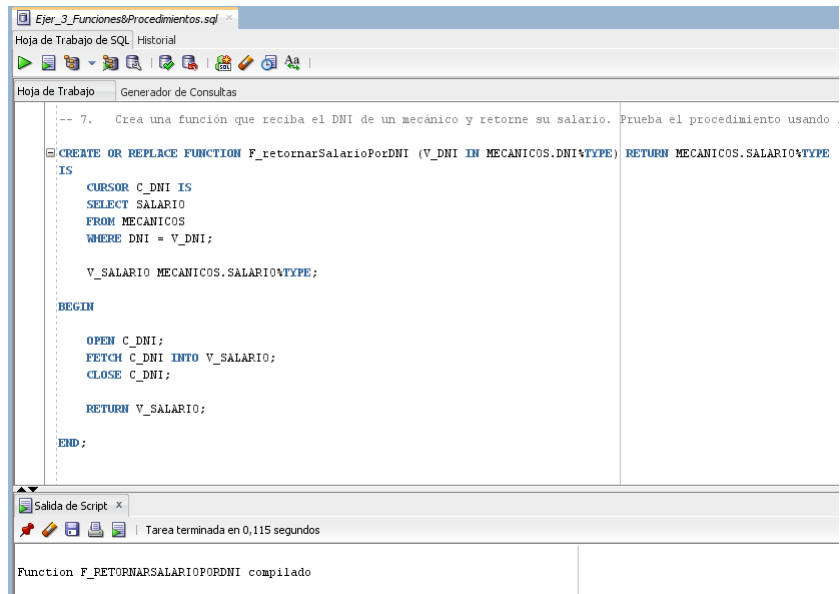
```
BEGIN
```

```
    V_DNI1 := '&DNI';
```

```
    V_SALARIO1 := F_retornarSalarioPorDNI(V_DNI1);
```

```
    DBMS_OUTPUT.PUT_LINE(V_SALARIO1);
```

```
END;
```



8. Crea un procedimiento (usando las funciones creadas anteriormente) que reciba el DNI de un mecánico y retorne su nombre y su salario. Prueba el procedimiento usando las funcionalidades de SQL Developer

```
CREATE OR REPLACE PROCEDURE P_retornarNombreyApellido(V_DNI2 IN MECANICOS.DNI%TYPE,
V_NOMBRE2 OUT MECANICOS.NOMBRE%TYPE, V_SALARIO2 OUT MECANICOS.SALARIO%TYPE)
IS
```

```
BEGIN
```

```
    V_NOMBRE2 := F_retornarNombrePorDNI(V_DNI2);
```

```
    V_SALARIO2 := F_retornarSalarioPorDNI(V_DNI2);
```

```
END P_retornarNombreyApellido;
```

```
-- bloque anónimo
```

```
DECLARE
```

```
    V_DNI MECANICOS.DNI%TYPE;
```

```
    V_NOMBRE
```

```
MECANICOS.NOMBRE%TYPE;
```

```
    V_SALARIO
```

```
MECANICOS.SALARIO%TYPE;
```

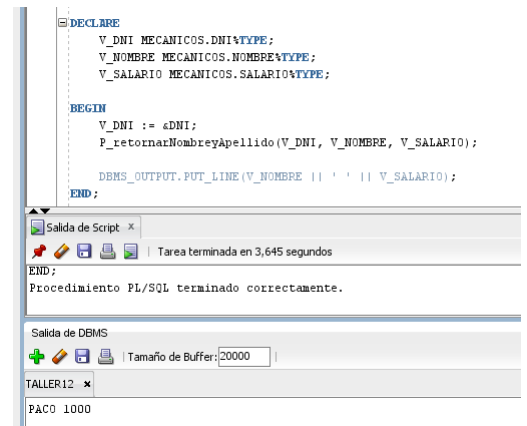
```
BEGIN
```

```
    V_DNI := &DNI;
```

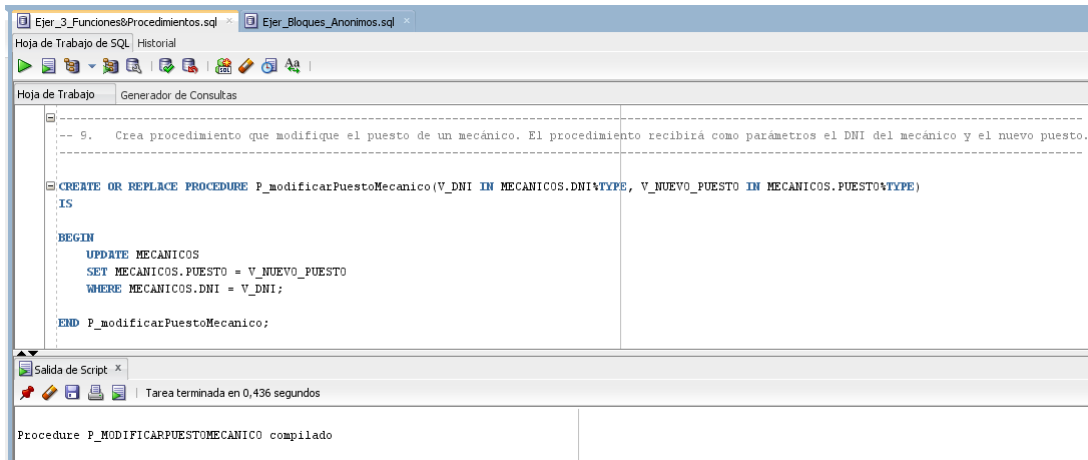
```
    P_retornarNombreyApellido(V_DNI, V_NOMBRE, V_SALARIO);
```

```
    DBMS_OUTPUT.PUT_LINE(V_NOMBRE || ' ' || V_SALARIO);
```

```
END;
```



9. Crea procedimiento que modifique el puesto de un mecánico. El procedimiento recibirá como parámetros el DNI del mecánico y el nuevo puesto.



```
CREATE OR REPLACE PROCEDURE P_modificarPuestoMecanico(V_DNI IN MECANICOS.DNI%TYPE,
V_NUEVO_PUESTO IN MECANICOS.PUESTO%TYPE)
IS
```

```
BEGIN
    UPDATE MECANICOS
    SET MECANICOS.PUESTO = V_NUEVO_PUESTO
    WHERE MECANICOS.DNI = V_DNI;

END P_modificarPuestoMecanico;
```

-- bloque anónimo

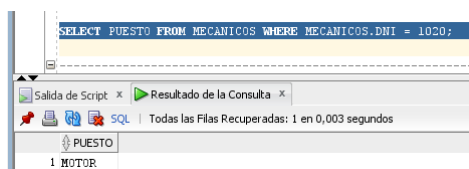
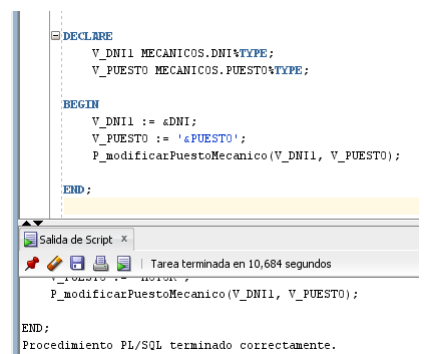
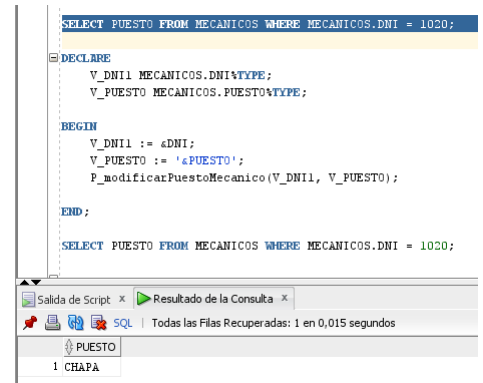
```
SELECT PUESTO FROM MECANICOS WHERE MECANICOS.DNI = 1020;
```

```
DECLARE
    V_DNI1 MECANICOS.DNI%TYPE;
    V_PUESTO MECANICOS.PUESTO%TYPE;

BEGIN
    V_DNI1 := &DNI;
    V_PUESTO := '&PUESTO';
    P_modificarPuestoMecanico(V_DNI1, V_PUESTO);

END;
```

```
SELECT PUESTO FROM MECANICOS WHERE MECANICOS.DNI = 1020;
```



10. Consulta todos los procedimientos y funciones del usuario almacenados en la base de datos y su situación (valid o invalid).

```
SELECT OBJECT_NAME, OBJECT_TYPE, STATUS
FROM USER_OBJECTS

-- FROM ALL_OBJECTS
WHERE OBJECT_TYPE IN ('PROCEDURE','FUNCTION');
```

-- 10. Consulta todos los procedimientos y funciones

```
SELECT OBJECT_NAME, OBJECT_TYPE, STATUS
FROM USER_OBJECTS
-- FROM ALL_OBJECTS
WHERE OBJECT_TYPE IN ('PROCEDURE','FUNCTION');
```

Salida de Script x Resultado de la Consulta x

Todas las Filas Recuperadas: 13 en 0,044 segundos

OBJECT_NAME	OBJECT_TYPE	STATUS
1 SUMARPOSNUMEROS	PROCEDURE	VALID
2 SUMASALARIOMECAVICOS	PROCEDURE	INVALID
3 COCHESFORMARCA	PROCEDURE	VALID
4 COCHESFORMARCA_FUNCION	FUNCTION	VALID
5 SUMARPOSNUMEROS_FUNCION	FUNCTION	VALID
6 RETORNARNOMBREPORDNI	FUNCTION	INVALID
7 P_SUMASALARIOMECAVICOS	PROCEDURE	VALID
8 F_SUMASALARIOMECAVICOS	FUNCTION	VALID
9 F_RETORNARNOMBREPORDNI	FUNCTION	VALID
10 F_RETORNARNARSALARIOORDNI	FUNCTION	VALID
11 P_RETORNARNOMBREYAPELLIDO	PROCEDURE	VALID
12 P_MODIFICARPUESTOMECAVICOS	PROCEDURE	VALID
13 AÑOS_FAB	PROCEDURE	INVALID