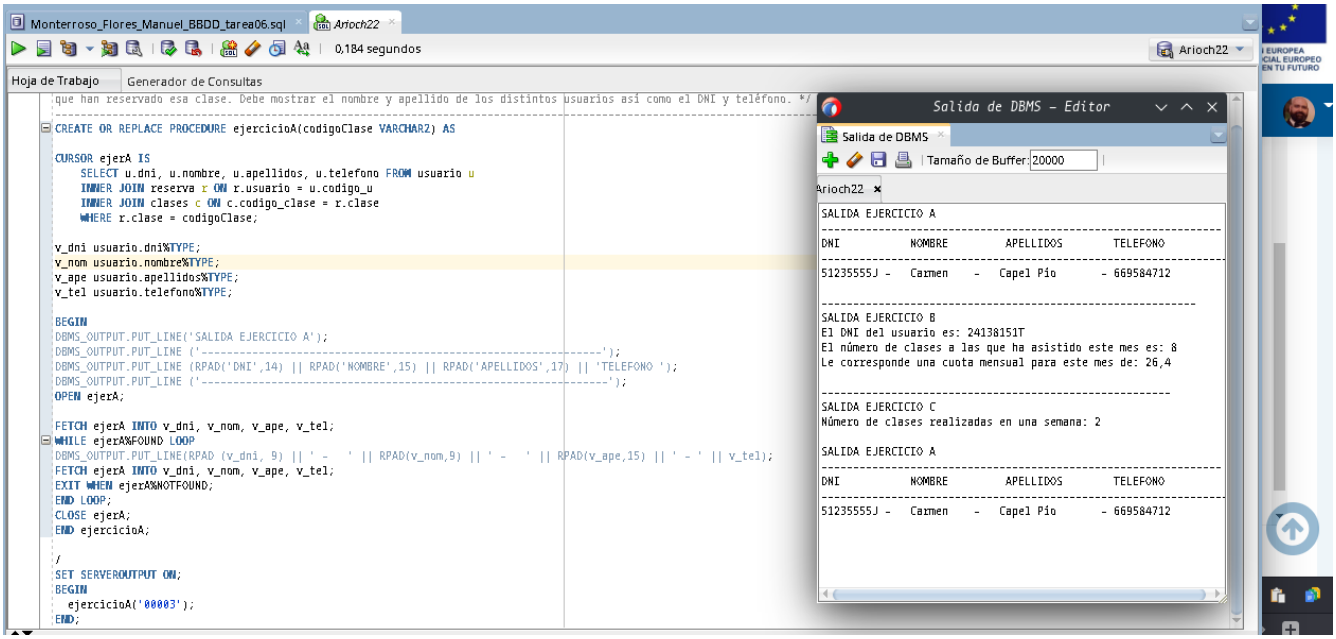
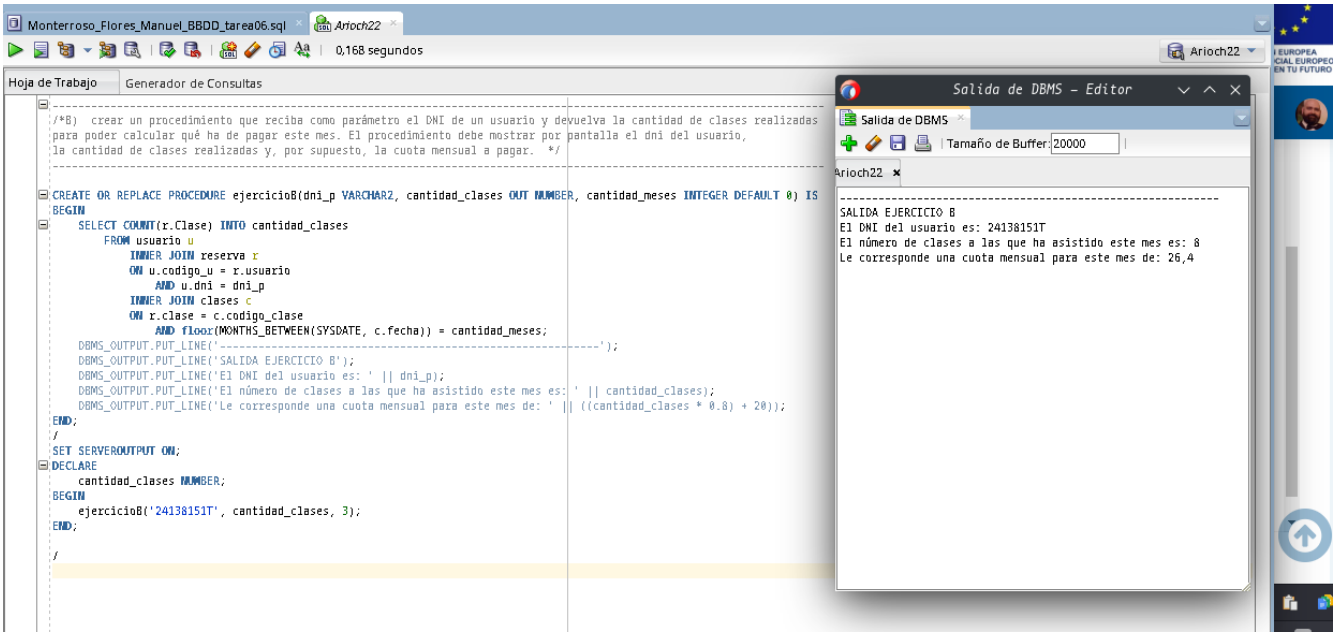
```
--SET SERVEROUTPUT ON
-------------------------------------------------------------------------------------------------------------------------------
/*A) Crear un procedimiento que reciba como parámetro el código de la clase y muestre por pantalla un listado con todos los usuarios
que han reservado esa clase. Debe mostrar el nombre y apellido de los distintos usuarios así como el DNI y teléfono. */
-------------------------------------------------------------------------------------------------------------------------------
CREATE OR REPLACE PROCEDURE ejercicioA(codigoClase VARCHAR2) AS

CURSOR ejerA IS
    SELECT u.dni, u.nombre, u.apellidos, u.telefono FROM usuario u
    INNER JOIN reserva r ON r.usuario = u.codigo_u
    INNER JOIN clases c ON c.codigo_clase = r.clase
    WHERE r.clase = codigoClase;

v_dni usuario.dni%TYPE;
v_nom usuario.nombre%TYPE;
v_ape usuario.apellidos%TYPE;
v_tel usuario.telefono%TYPE;

BEGIN
DBMS_OUTPUT.PUT_LINE('SALIDA EJERCICIO A');
DBMS_OUTPUT.PUT_LINE ('-----------------------------------------------------------------');
DBMS_OUTPUT.PUT_LINE (RPAD('DNI',14) || RPAD('NOMBRE',15) || RPAD('APELLIDOS',17) || 'TELEFONO ');
DBMS_OUTPUT.PUT_LINE ('-----------------------------------------------------------------');
OPEN ejerA;

FETCH ejerA INTO v_dni, v_nom, v_ape, v_tel;
WHILE ejerA%FOUND LOOP
DBMS_OUTPUT.PUT_LINE(RPAD (v_dni, 9) || ' -   ' || RPAD(v_nom,9) || ' -   ' || RPAD(v_ape,15) || ' - ' || v_tel);
FETCH ejerA INTO v_dni, v_nom, v_ape, v_tel;
EXIT WHEN ejerA%NOTFOUND;
END LOOP;
CLOSE ejerA;
END ejercicioA;

/
SET SERVEROUTPUT ON;
BEGIN
  ejercicioA('00003');
END;

/
```

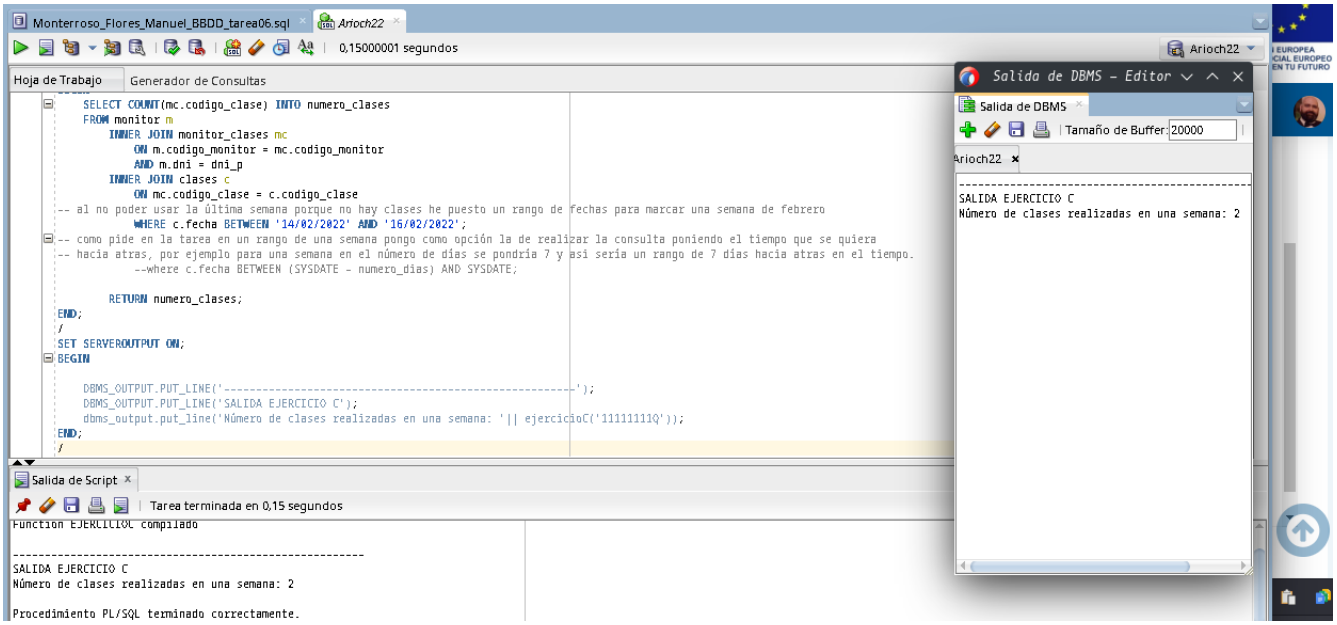
```
-----------------------------------------------------------------------------------------------------------------------
/*B)  crear un procedimiento que reciba como parámetro el DNI de un usuario y devuelva la cantidad de clases realizadas
para poder calcular qué ha de pagar este mes. El procedimiento debe mostrar por pantalla el dni del usuario,
la cantidad de clases realizadas y, por supuesto, la cuota mensual a pagar.  */
-----------------------------------------------------------------------------------------------------------------------

CREATE OR REPLACE PROCEDURE ejercicioB(dni_p VARCHAR2, cantidad_clases OUT NUMBER, cantidad_meses INTEGER DEFAULT 0) IS
BEGIN
   SELECT COUNT(r.Clase) INTO cantidad_clases
      FROM usuario u
         INNER JOIN reserva r
         ON u.codigo_u = r.usuario
            AND u.dni = dni_p
         INNER JOIN clases c
         ON r.clase = c.codigo_clase
            AND floor(MONTHS_BETWEEN(SYSDATE, c.fecha)) = cantidad_meses;
   DBMS_OUTPUT.PUT_LINE('------------------------------------------------------------');
   DBMS_OUTPUT.PUT_LINE('SALIDA EJERCICIO B');
   DBMS_OUTPUT.PUT_LINE('El DNI del usuario es: ' || dni_p);
   DBMS_OUTPUT.PUT_LINE('El número de clases a las que ha asistido este mes es: ' || cantidad_clases);
   DBMS_OUTPUT.PUT_LINE('Le corresponde una cuota mensual para este mes de: ' || ((cantidad_clases * 0.8) + 20));
END;
/
SET SERVEROUTPUT ON;
DECLARE
   cantidad_clases NUMBER;
BEGIN
   ejercicioB('24138151T', cantidad_clases, 3);
END;

/

-----------------------------------------------------------------------------------------------------------------------
--------
/*C) Crea una función que reciba el nombre de un monitor y nos indique en cuántas clases ha impartido durante la semana en curso. */
-----------------------------------------------------------------------------------------------------------------------
--------
CREATE OR REPLACE FUNCTION ejercicioC(dni_p VARCHAR2 /*, numero_dias INTEGER DEFAULT 1 variable para segundo ejemplo*/) RETURN INTEGER IS
   numero_clases INTEGER;
BEGIN
```

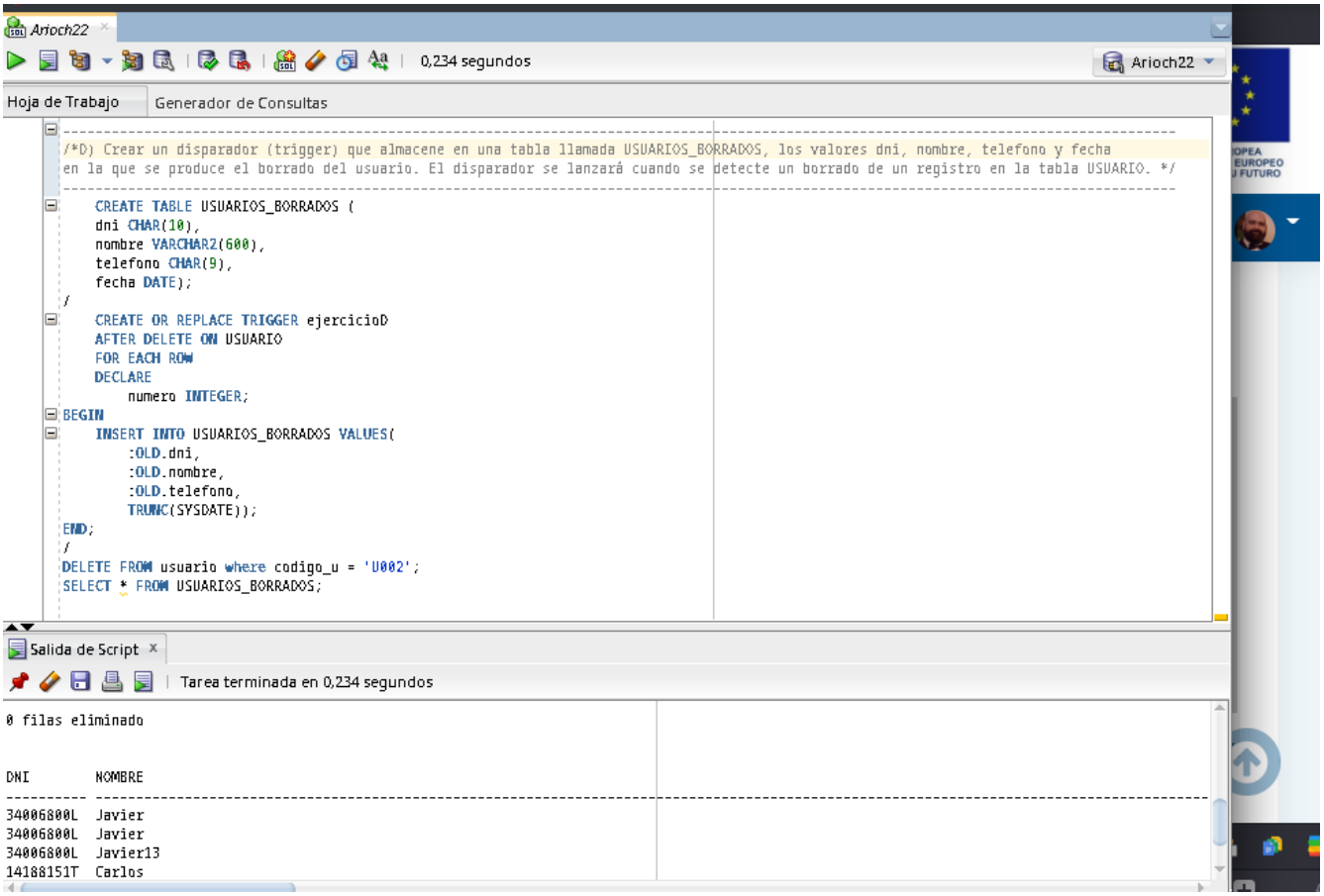
```
    SELECT COUNT(mc.codigo_clase) INTO numero_clases
    FROM monitor m
        INNER JOIN monitor_clases mc
            ON m.codigo_monitor = mc.codigo_monitor
            AND m.dni = dni_p
        INNER JOIN clases c
            ON mc.codigo_clase = c.codigo_clase
-- al no poder usar la última semana porque no hay clases he puesto un rango de fechas para marcar una semana de febrero
            WHERE c.fecha BETWEEN '14/02/2022' AND '16/02/2022';
-- como pide en la tarea en un rango de una semana pongo como opción la de realizar la consulta poniendo el tiempo que se quiera
-- hacia atras, por ejemplo para una semana en el número de días se pondría 7 y asi sería un rango de 7 días hacia atras en el tiempo.
            --where c.fecha BETWEEN (SYSDATE - numero_dias) AND SYSDATE;

        RETURN numero_clases;
END;
/
SET SERVEROUTPUT ON;
BEGIN

    DBMS_OUTPUT.PUT_LINE('-----------------------------------------------------');
    DBMS_OUTPUT.PUT_LINE('SALIDA EJERCICIO C');
    dbms_output.put_line('Número de clases realizadas en una semana: '|| ejercicioC('11111111Q'));
END;
/

-------------------------------------------------------------------------------------------------------------------------
/*D) Crear un disparador (trigger) que almacene en una tabla llamada USUARIOS_BORRADOS, los valores dni, nombre, telefono y fecha
en la que se produce el borrado del usuario. El disparador se lanzará cuando se detecte un borrado de un registro en la tabla USUARIO. */
-------------------------------------------------------------------------------------------------------------------------
    CREATE TABLE USUARIOS_BORRADOS (
    dni CHAR(10),
    nombre VARCHAR2(600),
    telefono CHAR(9),
    fecha DATE);
/
    CREATE OR REPLACE TRIGGER ejercicioD
    AFTER DELETE ON USUARIO
    FOR each ROW
    DECLARE
        numero INTEGER;
```

```
BEGIN
   INSERT INTO USUARIOS_BORRADOS VALUES(
      :old.dni,
      :old.nombre,
      :old.telefono,
      TRUNC(SYSDATE));
END;
/
DELETE FROM usuario where codigo_u = 'U002';
SELECT * FROM USUARIOS_BORRADOS;
```




Monterroso_Flores_Manuel_BBDD_tarea06.sql
Arioch22
0,168 segundos
Arioch22
Hoja de Trabajo
Generador de Consultas
/*B) crear un procedimiento que reciba como parámetro el DNI de un usuario y devuelva la cantidad de clases realizadas
para poder calcular qué ha de pagar este mes. El procedimiento debe mostrar por pantalla el dni del usuario,
la cantidad de clases realizadas y, por supuesto, la cuota mensual a pagar. */
CREATE OR REPLACE PROCEDURE ejercicioB(dni_p VARCHAR2, cantidad_clases OUT NUMBER, cantidad_meses INTEGER DEFAULT 0) IS
BEGIN
SELECT COUNT(r.Clase) INTO cantidad_clases
FROM usuario u
INNER JOIN reserva r
ON u.codigo_u = r.usuario
AND u.dni = dni_p
INNER JOIN clases c
ON r.clase = c.codigo_clase
AND floor(MONTHS_BETWEEN(SYSDATE, c.fecha)) = cantidad_meses;
DBMS_OUTPUT.PUT_LINE('-------------------------------------------------------');
DBMS_OUTPUT.PUT_LINE('SALIDA EJERCICIO B');
DBMS_OUTPUT.PUT_LINE('El DNI del usuario es: ' || dni_p);
DBMS_OUTPUT.PUT_LINE('El número de clases a las que ha asistido este mes es: ' || cantidad_clases);
DBMS_OUTPUT.PUT_LINE('Le corresponde una cuota mensual para este mes de: ' || ((cantidad_clases * 0.8) + 20));
END;
/
SET SERVEROUTPUT ON;
DECLARE
cantidad_clases NUMBER;
BEGIN
ejercicioB('24138151T', cantidad_clases, 3);
END;
/
Salida de DBMS - Editor
Salida de DBMS
Tamaño de Buffer: 20000
Arioch22
SALIDA EJERCICIO B
El DNI del usuario es: 24138151T
El número de clases a las que ha asistido este mes es: 8
Le corresponde una cuota mensual para este mes de: 26,4


Monterroso_Flores_Manuel_BBDD_tarea06.sql
Arioch22
0,15000001 segundos
Arioch22
Hoja de Trabajo
Generador de Consultas
SELECT COUNT(mc.codigo_clase) INTO numero_clases
FROM monitor m
INNER JOIN monitor_clases mc
ON m.codigo_monitor = mc.codigo_monitor
AND m.dni = dni_p
INNER JOIN clases c
ON mc.codigo_clase = c.codigo_clase
-- al no poder usar la última semana porque no hay clases he puesto un rango de fechas para marcar una semana de febrero
WHERE c.fecha BETWEEN '14/02/2022' AND '16/02/2022';
-- como pide en la tarea en un rango de una semana pongo como opción la de realizar la consulta poniendo el tiempo que se quiera
-- hacia atras, por ejemplo para una semana en el número de días se pondría 7 y así sería un rango de 7 días hacia atras en el tiempo.
--where c.fecha BETWEEN (SYSDATE - numero_dias) AND SYSDATE;
RETURN numero_clases;
END;
/
SET SERVEROUTPUT ON;
BEGIN
DBMS_OUTPUT.PUT_LINE('-------------------------------------------------------');
DBMS_OUTPUT.PUT_LINE('SALIDA EJERCICIO C');
dbms_output.put_line('Número de clases realizadas en una semana: '|| ejercicioC('11111111Q'));
END;
/
Salida de DBMS - Editor
Salida de DBMS
Tamaño de Buffer: 20000
Arioch22
SALIDA EJERCICIO C
Número de clases realizadas en una semana: 2
Salida de Script
Tarea terminada en 0,15 segundos
Function EJERCICIOC compilado
SALIDA EJERCICIO C
Número de clases realizadas en una semana: 2
Procedimiento PL/SQL terminado correctamente.


Arioch22
0,234 segundos
Arioch22
Hoja de Trabajo
Generador de Consultas
/*D) Crear un disparador (trigger) que almacene en una tabla llamada USUARIOS_BORRADOS, los valores dni, nombre, telefono y fecha
en la que se produce el borrado del usuario. El disparador se lanzará cuando se detecte un borrado de un registro en la tabla USUARIO. */
CREATE TABLE USUARIOS_BORRADOS (
dni CHAR(10),
nombre VARCHAR2(600),
telefono CHAR(9),
fecha DATE);
/
CREATE OR REPLACE TRIGGER ejercicioD
AFTER DELETE ON USUARIO
FOR EACH ROW
DECLARE
numero INTEGER;
BEGIN
INSERT INTO USUARIOS_BORRADOS VALUES(
:OLD.dni,
:OLD.nombre,
:OLD.telefono,
TRUNC(SYSDATE));
END;
/
DELETE FROM usuario where codigo_u = 'U002';
SELECT * FROM USUARIOS_BORRADOS;
Salida de Script
Tarea terminada en 0,234 segundos
0 filas eliminado
DNI NOMBRE
34006800L Javier
34006800L Javier
34006800L Javier13
14188151T Carlos