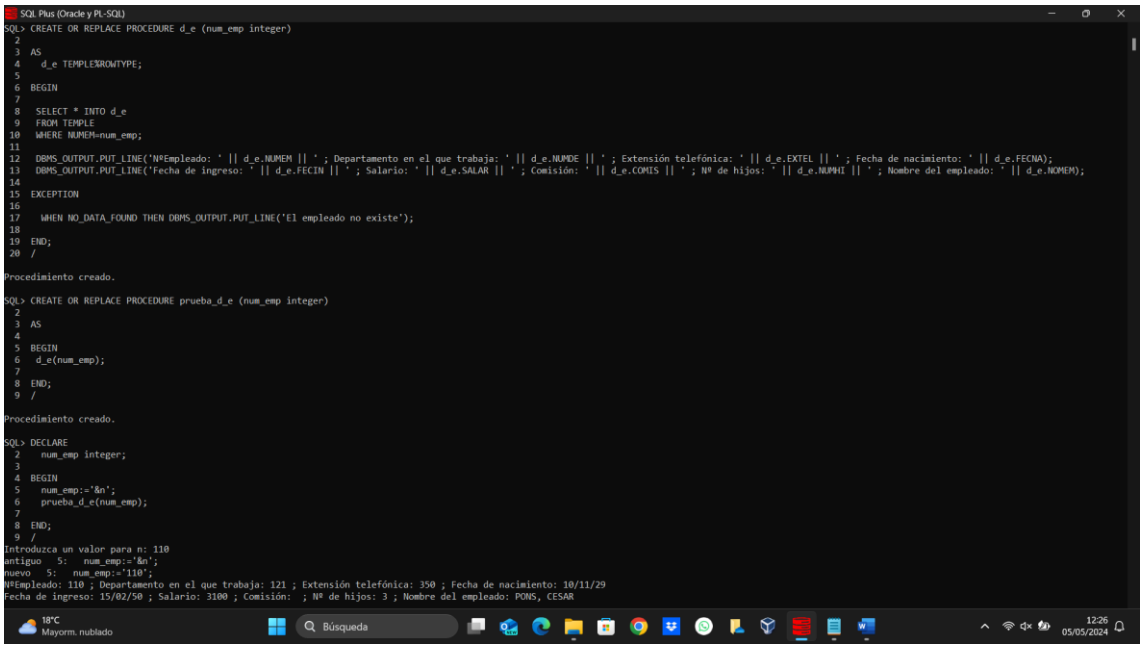
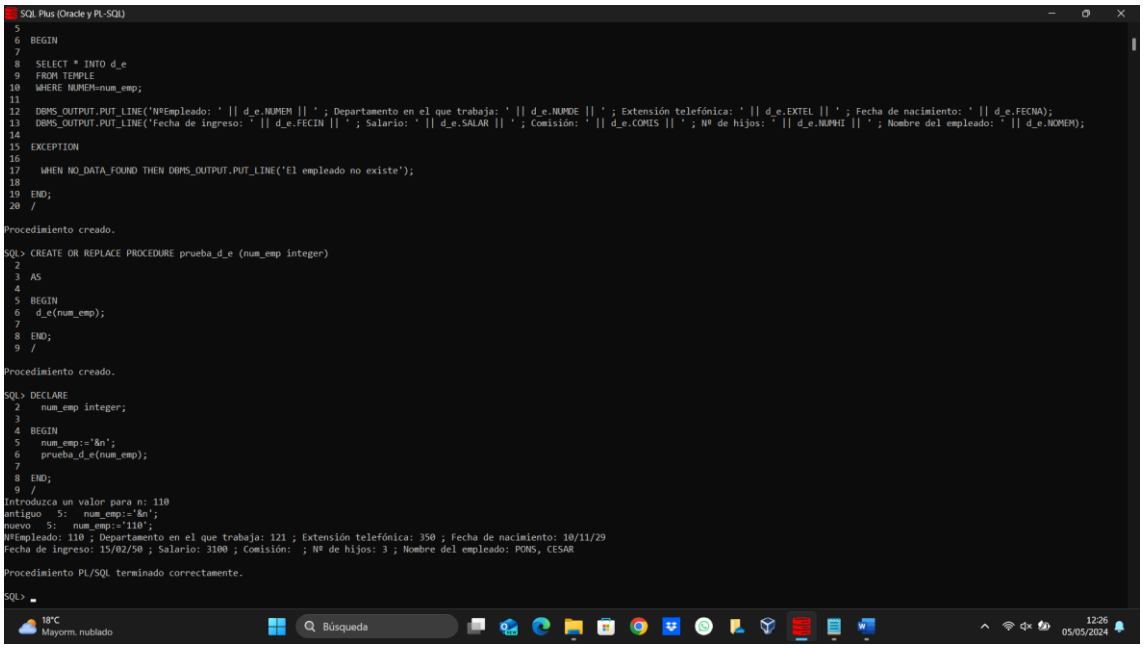
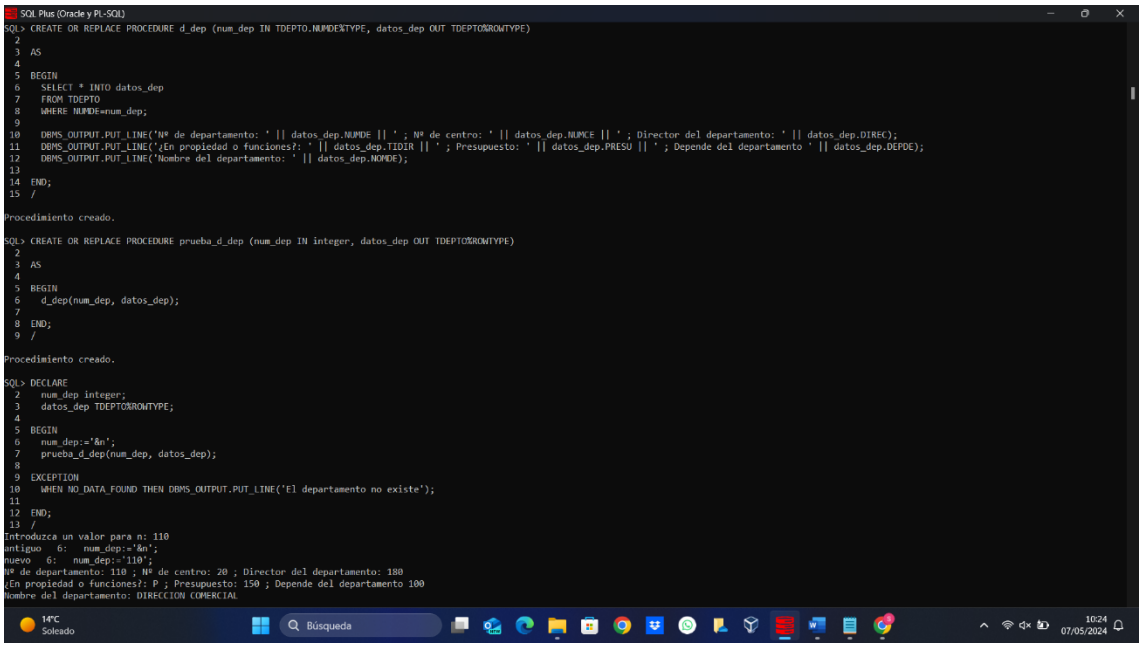
Queremos crear los siguientes subprogramas:

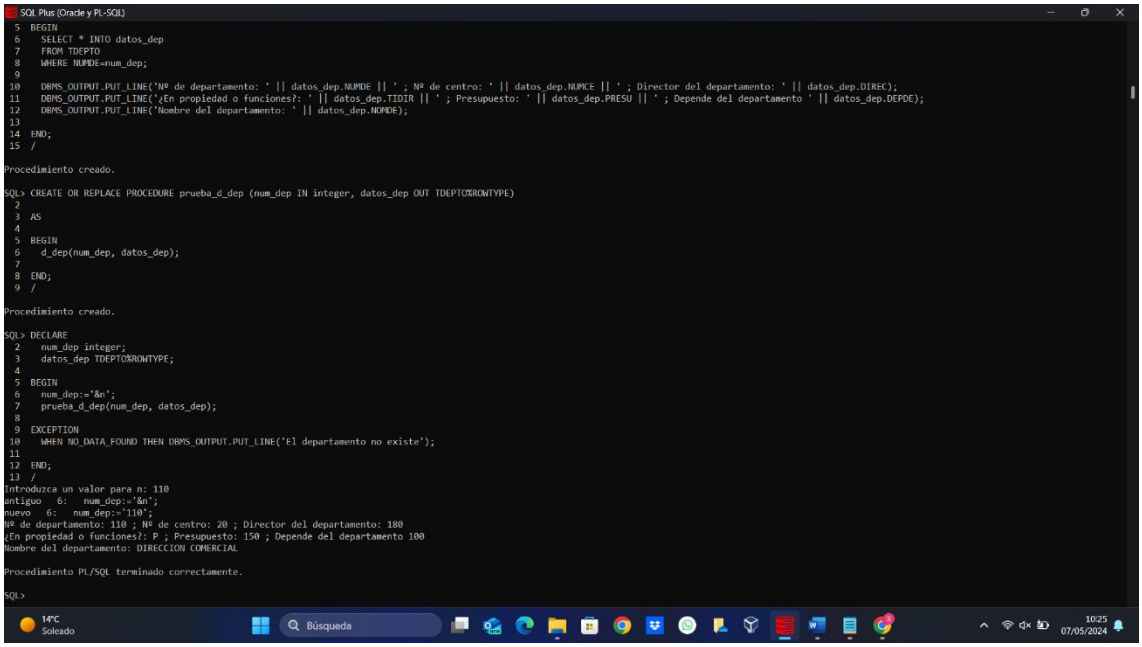
1.- **Procedimiento** que escriba los datos de un empleado a partir del número de empleado, el número de empleado es un parámetro de entrada del procedimiento. Hay que probarlo desde un bloque anónimo y desde otro procedimiento. Si no existe se controla mediante una EXCEPCIÓN



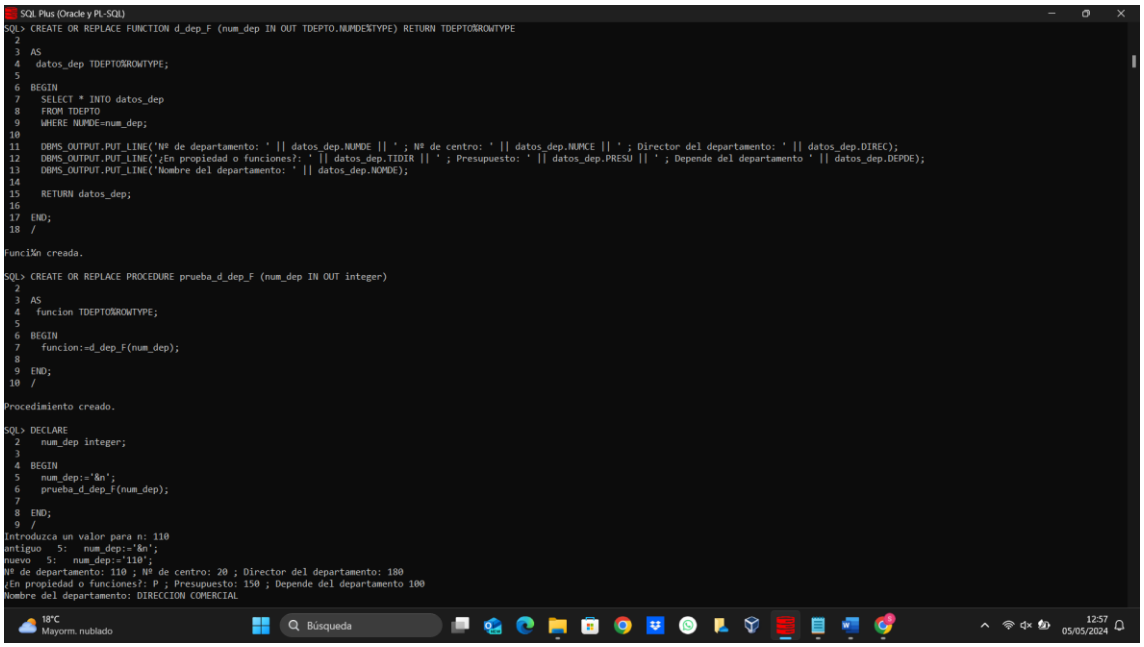



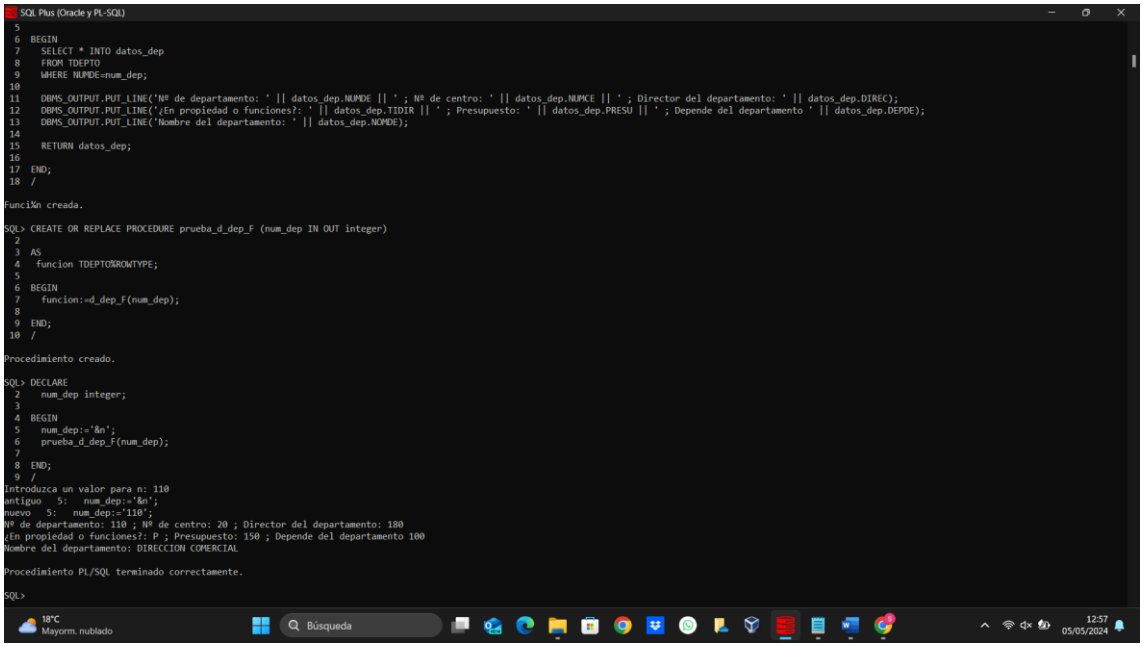

2.- **Procedimiento** que devuelva, mediante un parámetro de salida, los datos de un departamento a partir del número de departamento, el número de departamento es un parámetro de entrada del procedimiento. Hay que probarlo desde un bloque anónimo y desde otro procedimiento. (Aquí no utilicéis EXCEPCIÓN)




3.- **Función** que devuelve todos los datos de un departamento a partir del número de departamento, el número de departamento es un parámetro de entrada a la función. Hay que probarlo desde un bloque anónimo y desde otro procedimiento.

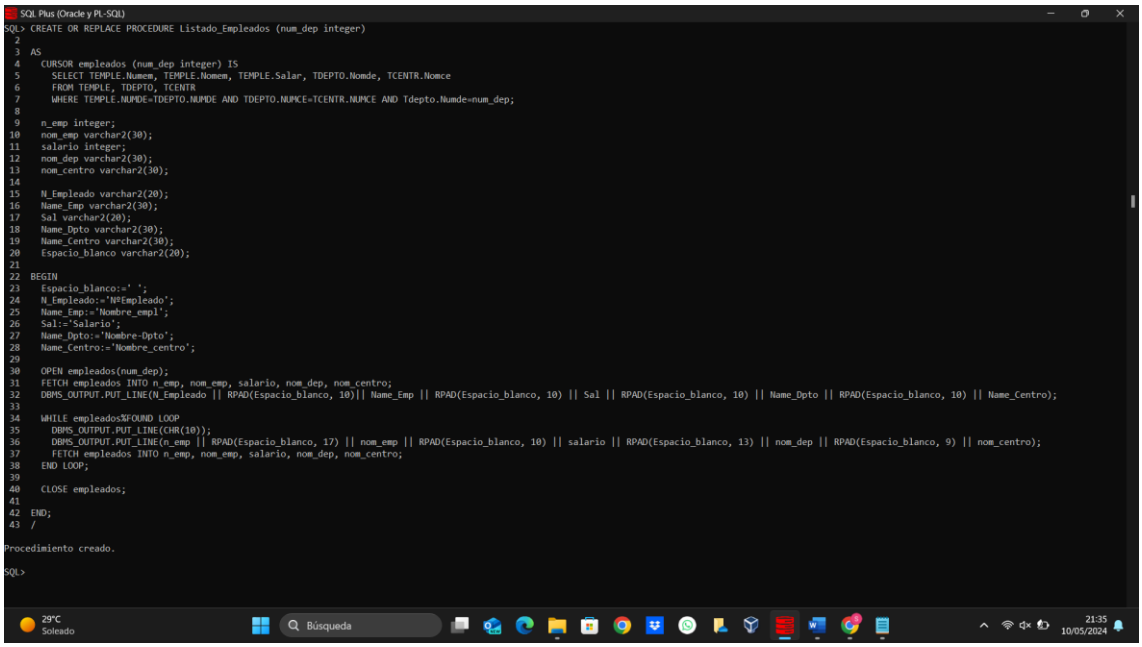


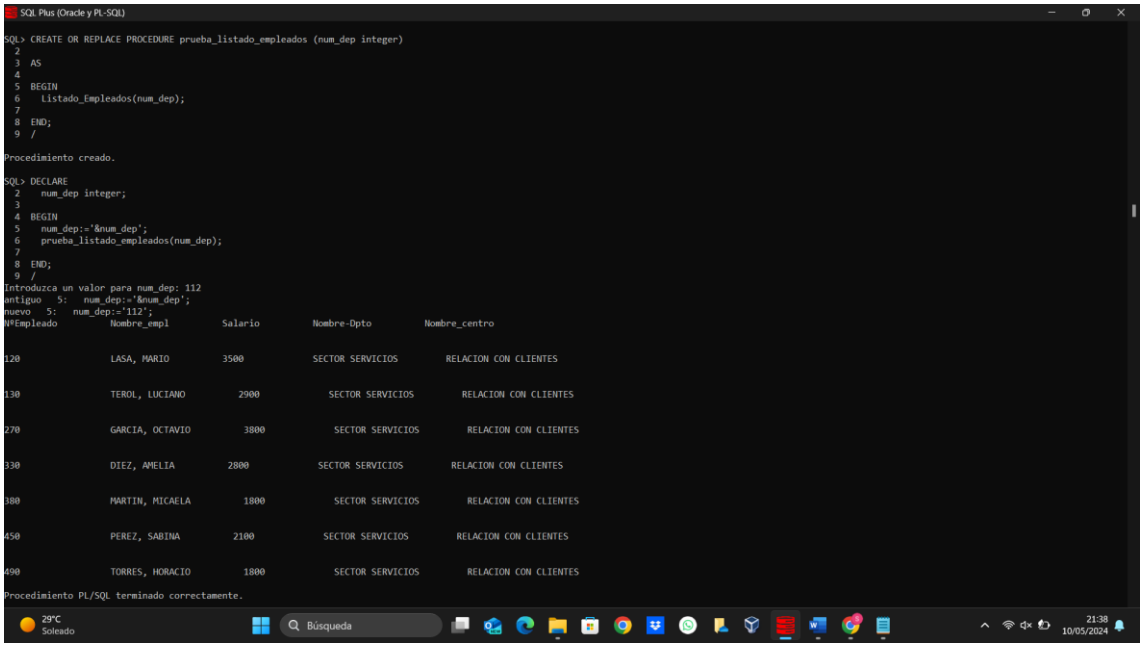

4.- **Procedimiento** que muestra un listado de empleados (a partir de su número de departamento), de la siguiente manera:

| **Nº Empleado** | **Nombre_empl** | **Salario** | **Nombre-Dpto** | **Nombre_centro** |
|---|---|---|---|---|
| ::::::::: | :::::::::::::::::::: | ::::: | ::::::: | ::::::: |
| ..... | ..... | ..... | ..... | ..... |

Hay que probarlo desde un bloque anónimo y desde otro procedimiento.

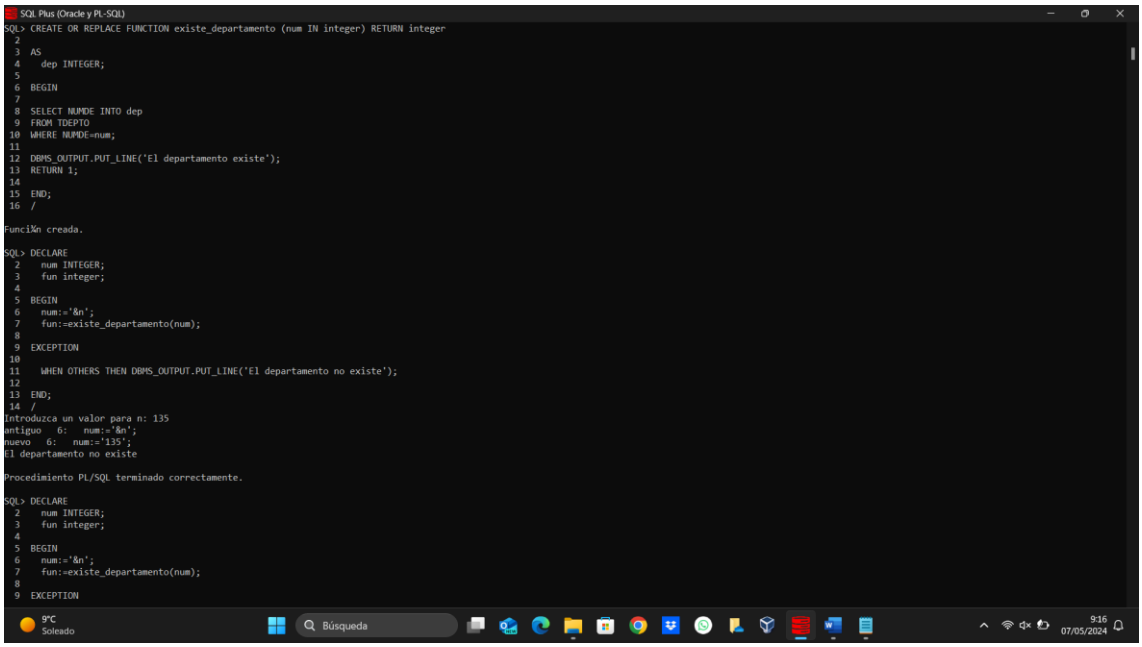


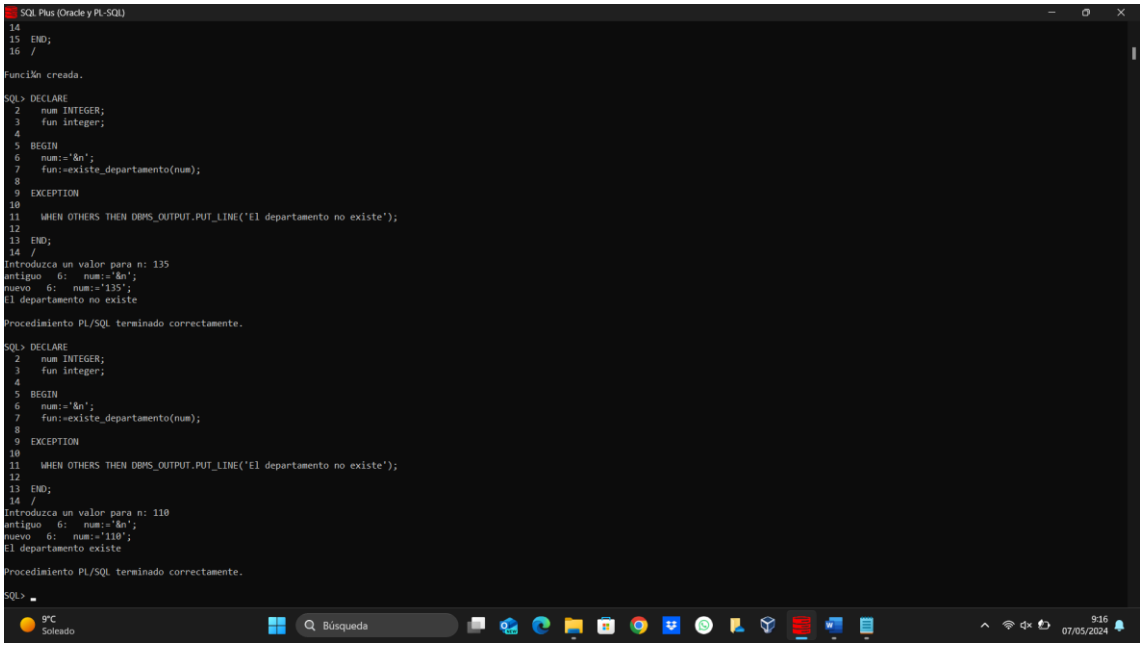
```
SQL Plus (Oracle y PL-SQL)
SQL> CREATE OR REPLACE PROCEDURE Listado_Empleados (num_dep integer)
  2
  3  AS
  4    CURSOR empleados (num_dep integer) IS
  5      SELECT TEMPLE.Numem, TEMPLE.Nomem, TEMPLE.Salar, TDEPTO.Nomde, TCENTR.Nomce
  6      FROM TEMPLE, TDEPTO, TCENTR
  7      WHERE TEMPLE.NUMDE=TDEPTO.NUMDE AND TDEPTO.NUMCE=TCENTR.NUMCE AND Tdepto.Numde=num_dep;
  8
  9    n_emp integer;
 10    nom_emp varchar2(30);
 11    salario integer;
 12    nom_dep varchar2(30);
 13    nom_centro varchar2(30);
 14
 15    N_Empleado varchar2(20);
 16    Name_Emp varchar2(30);
 17    Sal varchar2(20);
 18    Name_Dpto varchar2(30);
 19    Name_Centro varchar2(30);
 20    Espacio_blanco varchar2(20);
 21
 22  BEGIN
 23    Espacio_blanco:=' ';
 24    N_Empleado:='NºEmpleado';
 25    Name_Emp:='Nombre_empl';
 26    Sal:='Salario';
 27    Name_Dpto:='Nombre-Dpto';
 28    Name_Centro:='Nombre_centro';
 29
 30    OPEN empleados(num_dep);
 31    FETCH empleados INTO n_emp, nom_emp, salario, nom_dep, nom_centro;
 32    DBMS_OUTPUT.PUT_LINE(N_Empleado || RPAD(Espacio_blanco, 10)|| Name_Emp || RPAD(Espacio_blanco, 10) || Sal || RPAD(Espacio_blanco, 10) || Name_Dpto || RPAD(Espacio_blanco, 10) || Name_Centro);
 33
 34    WHILE empleados%FOUND LOOP
 35      DBMS_OUTPUT.PUT_LINE(CHR(10));
 36      DBMS_OUTPUT.PUT_LINE(n_emp || RPAD(Espacio_blanco, 17) || nom_emp || RPAD(Espacio_blanco, 10) || salario || RPAD(Espacio_blanco, 13) || nom_dep || RPAD(Espacio_blanco, 9) || nom_centro);
 37      FETCH empleados INTO n_emp, nom_emp, salario, nom_dep, nom_centro;
 38    END LOOP;
 39
 40    CLOSE empleados;
 41
 42  END;
 43  /

Procedimiento creado.

SQL>
```



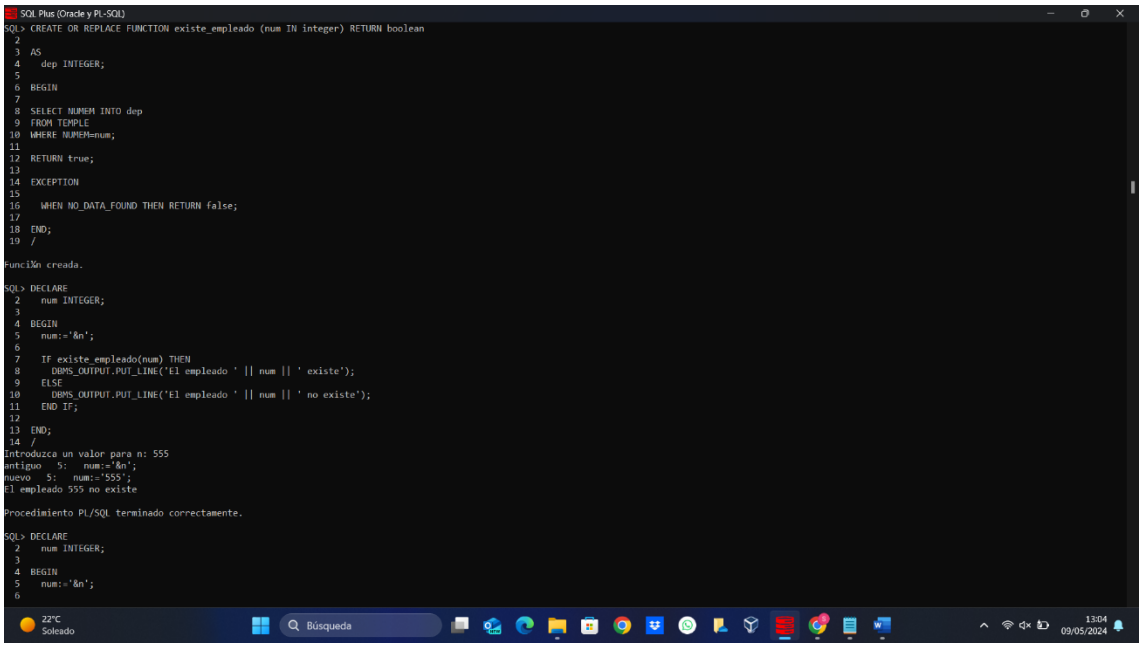
```
SQL Plus (Oracle y PL-SQL)
SQL> CREATE OR REPLACE PROCEDURE prueba_listado_empleados (num_dep integer)
  2
  3  AS
  4
  5  BEGIN
  6    Listado_Empleados(num_dep);
  7
  8  END;
  9  /

Procedimiento creado.

SQL> DECLARE
  2    num_dep integer;
  3
  4  BEGIN
  5    num_dep:='&num_dep';
  6    prueba_listado_empleados(num_dep);
  7
  8  END;
  9  /
Introduzca un valor para num_dep: 112
antiguo   5:   num_dep:='&num_dep';
nuevo   5:   num_dep:='112';
NºEmpleado          Nombre_empl          Salario          Nombre-Dpto          Nombre_centro

120                 LASA, MARIO          3500             SECTOR SERVICIOS         RELACION CON CLIENTES

130                 TEROL, LUCIANO          2900             SECTOR SERVICIOS         RELACION CON CLIENTES

270                 GARCIA, OCTAVIO          3800             SECTOR SERVICIOS         RELACION CON CLIENTES

330                 DIEZ, AMELIA         2800             SECTOR SERVICIOS         RELACION CON CLIENTES

380                 MARTIN, MICAELA          1800             SECTOR SERVICIOS         RELACION CON CLIENTES

450                 PEREZ, SABINA         2100             SECTOR SERVICIOS         RELACION CON CLIENTES

490                 TORRES, HORACIO          1800             SECTOR SERVICIOS         RELACION CON CLIENTES

Procedimiento PL/SQL terminado correctamente.
```

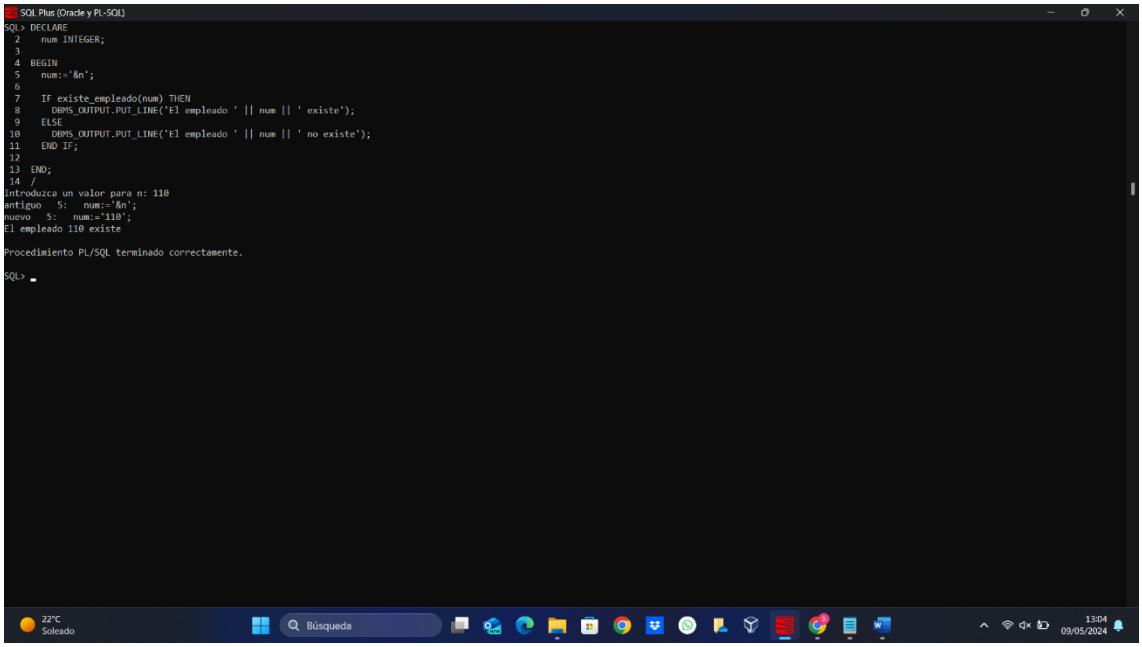
5. - **Función** que compruebe si existe un departamento con el número que se le pase (parámetro de entrada). Devolverá verdadero o falso, 1 ó 0 lo que creáis conveniente.



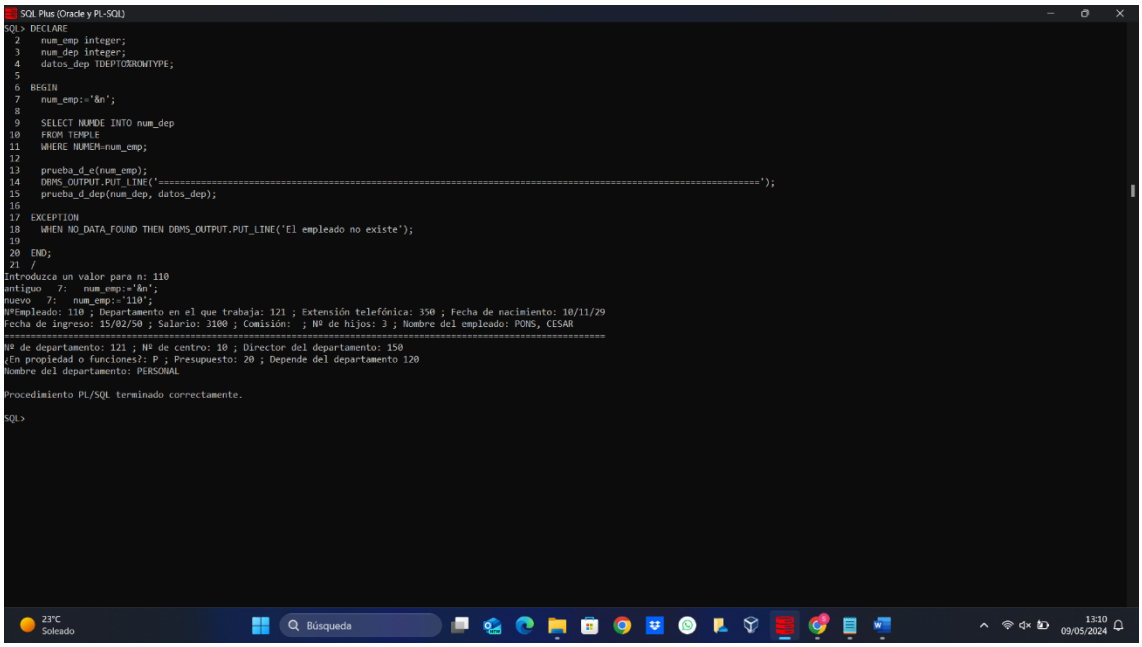



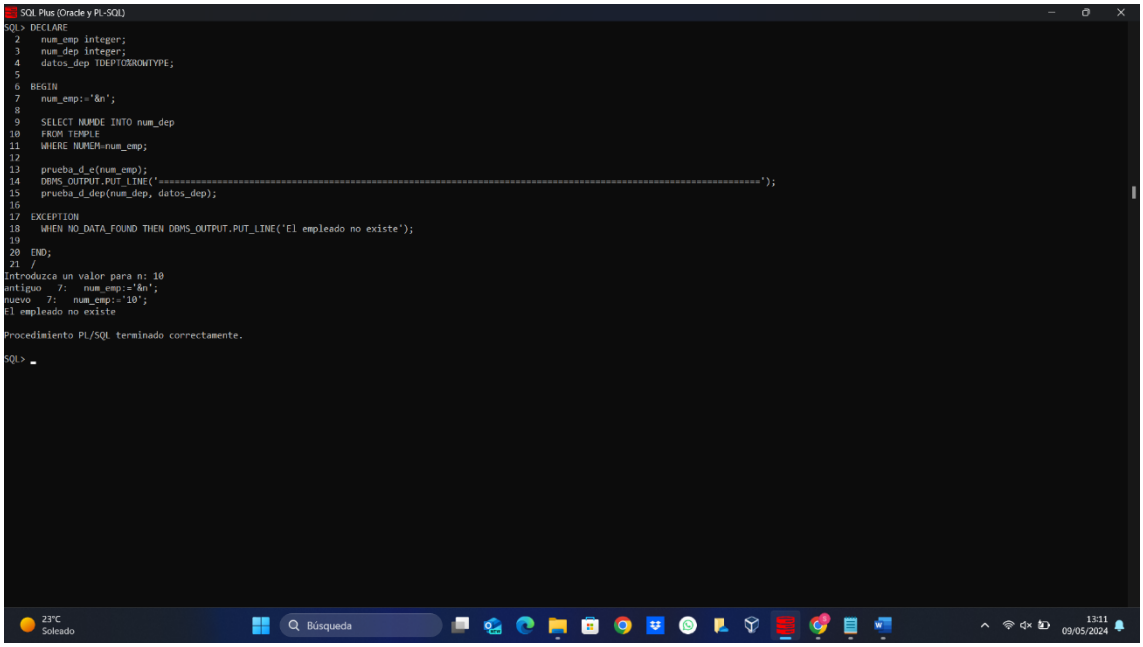

6.- **Función** que compruebe si existe un empleado con el número que se le pase (parámetro de entrada). Devolverá verdadero o falso, 1 ó 0 lo que creáis conveniente.



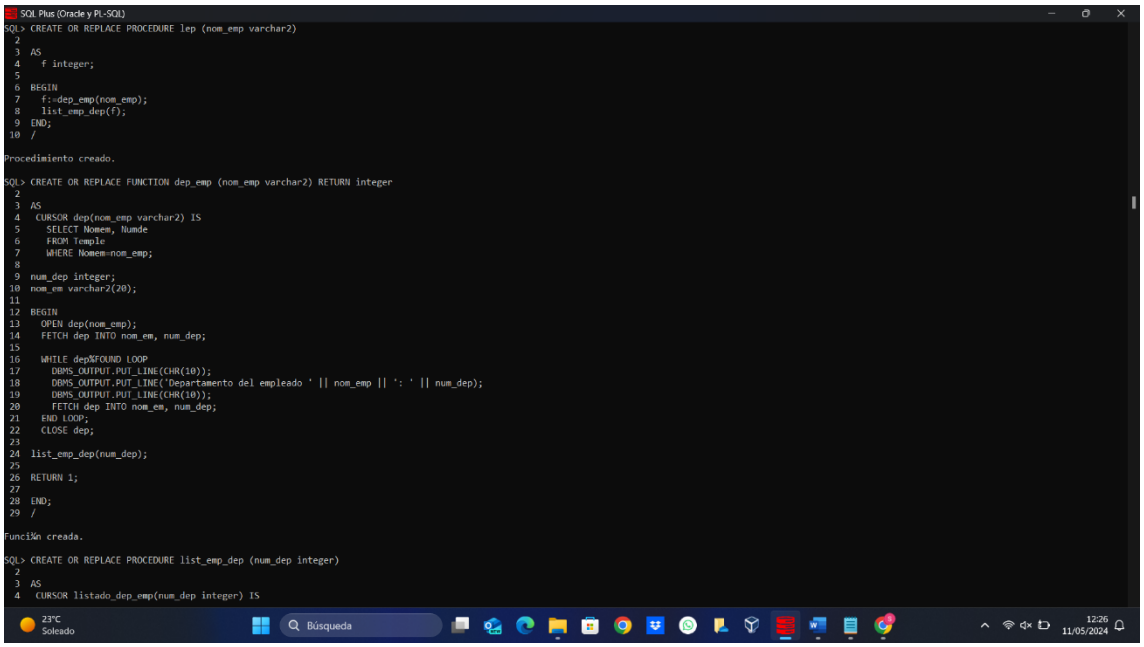




7. **Procedimiento** o **bloque anónimo** que a partir de un número de empleado, si existe, nos muestre todos los datos del empleado y de su departamento utilizando los subprogramas anteriores.



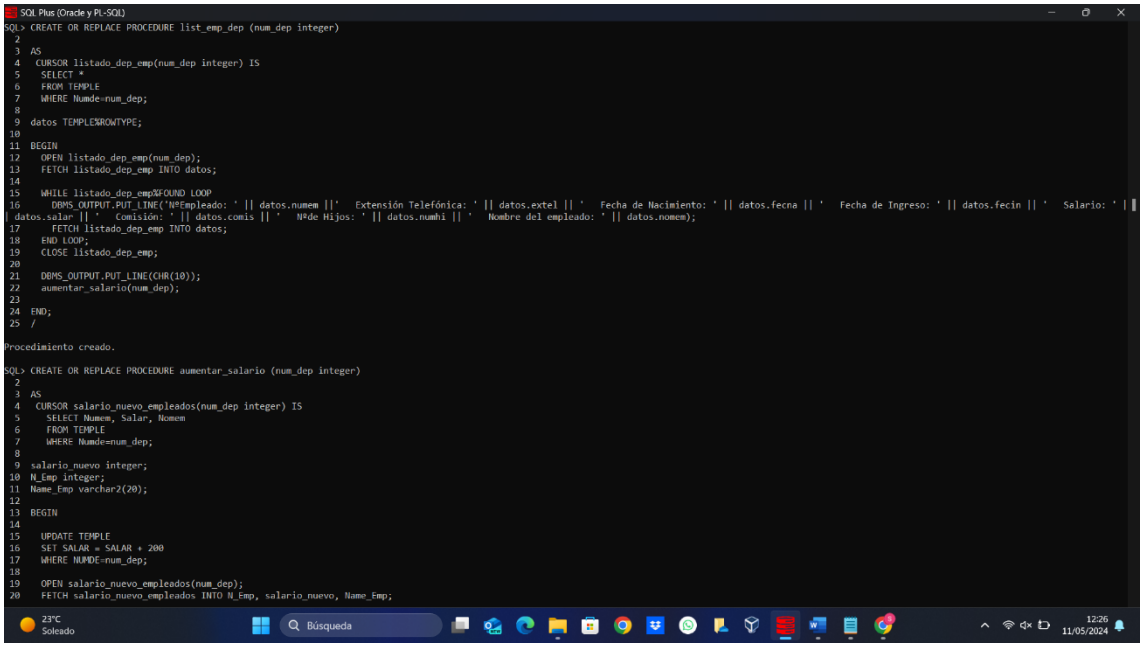




8.- **Procedimiento** al que le pasamos un nombre de empleado como parámetro de entrada y debe obtener, mediante una función que también hay que crear, el departamento de ese empleado. Una vez tengamos el departamento el procedimiento hace un listado de los empleados de ese departamento utilizando otro procedimiento que también hay que crear (se le pasa el número de departamento), aumenta el salario de los empleados de ese departamento mediante otro procedimiento (también se le pasa el número de departamento) y se vuelven a escribir los empleados para comprobar que se ha cambiado el salario. Hay que probarlo desde un bloque anónimo y desde otro procedimiento.

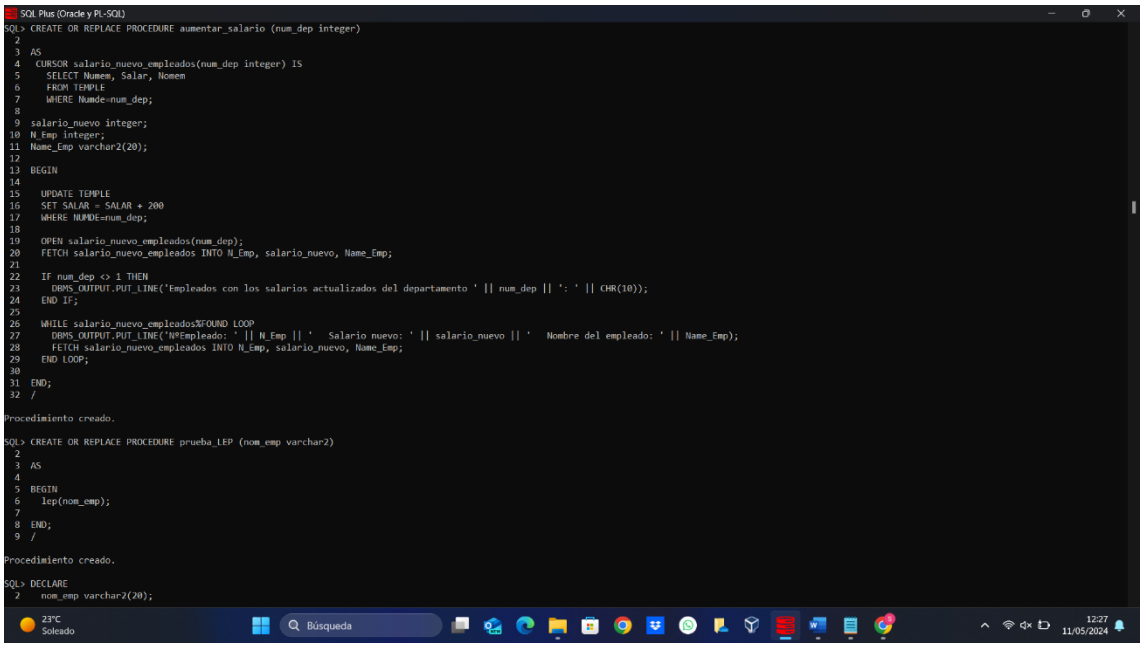


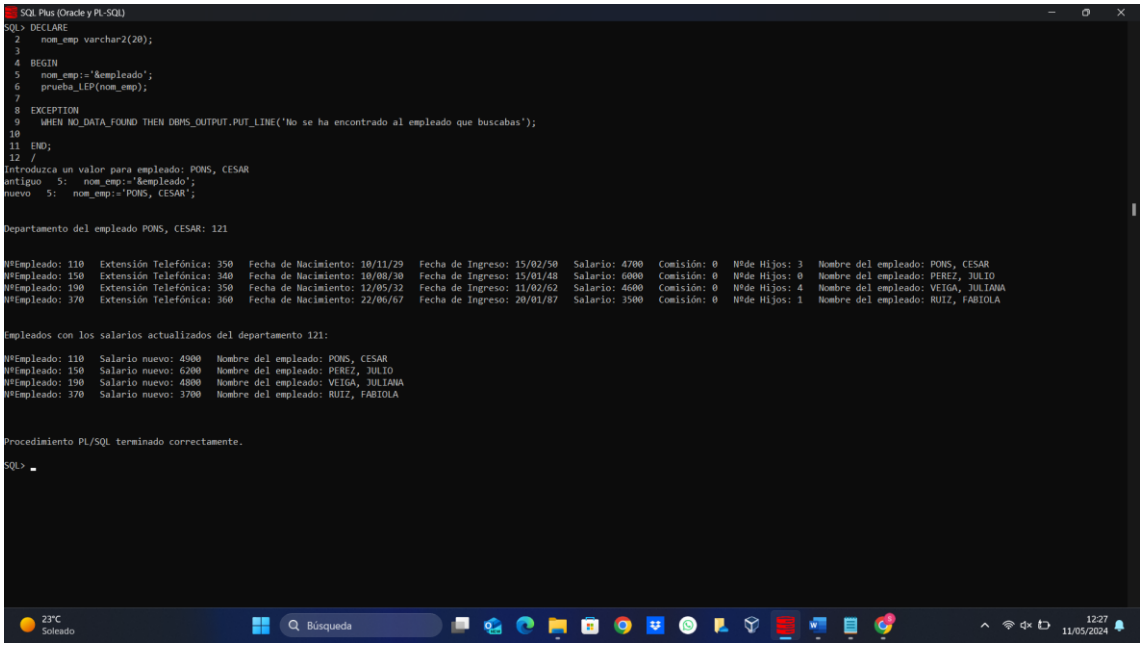
```
SQL Plus (Oracle y PL-SQL)
SQL> CREATE OR REPLACE PROCEDURE lep (nom_emp varchar2)
  2
  3  AS
  4    f integer;
  5
  6  BEGIN
  7    f:=dep_emp(nom_emp);
  8    list_emp_dep(f);
  9  END;
 10  /

Procedimiento creado.

SQL> CREATE OR REPLACE FUNCTION dep_emp (nom_emp varchar2) RETURN integer
  2
  3  AS
  4   CURSOR dep(nom_emp varchar2) IS
  5     SELECT Nomem, Numde
  6     FROM Temple
  7     WHERE Nomem=nom_emp;
  8
  9  num_dep integer;
 10  nom_em varchar2(20);
 11
 12  BEGIN
 13    OPEN dep(nom_emp);
 14    FETCH dep INTO nom_em, num_dep;
 15
 16    WHILE dep%FOUND LOOP
 17      DBMS_OUTPUT.PUT_LINE(CHR(10));
 18      DBMS_OUTPUT.PUT_LINE('Departamento del empleado ' || nom_emp || ': ' || num_dep);
 19      DBMS_OUTPUT.PUT_LINE(CHR(10));
 20      FETCH dep INTO nom_em, num_dep;
 21    END LOOP;
 22    CLOSE dep;
 23
 24  list_emp_dep(num_dep);
 25
 26  RETURN 1;
 27
 28  END;
 29  /

Funci¾n creada.

SQL> CREATE OR REPLACE PROCEDURE list_emp_dep (num_dep integer)
  2
  3  AS
  4   CURSOR listado_dep_emp(num_dep integer) IS
```



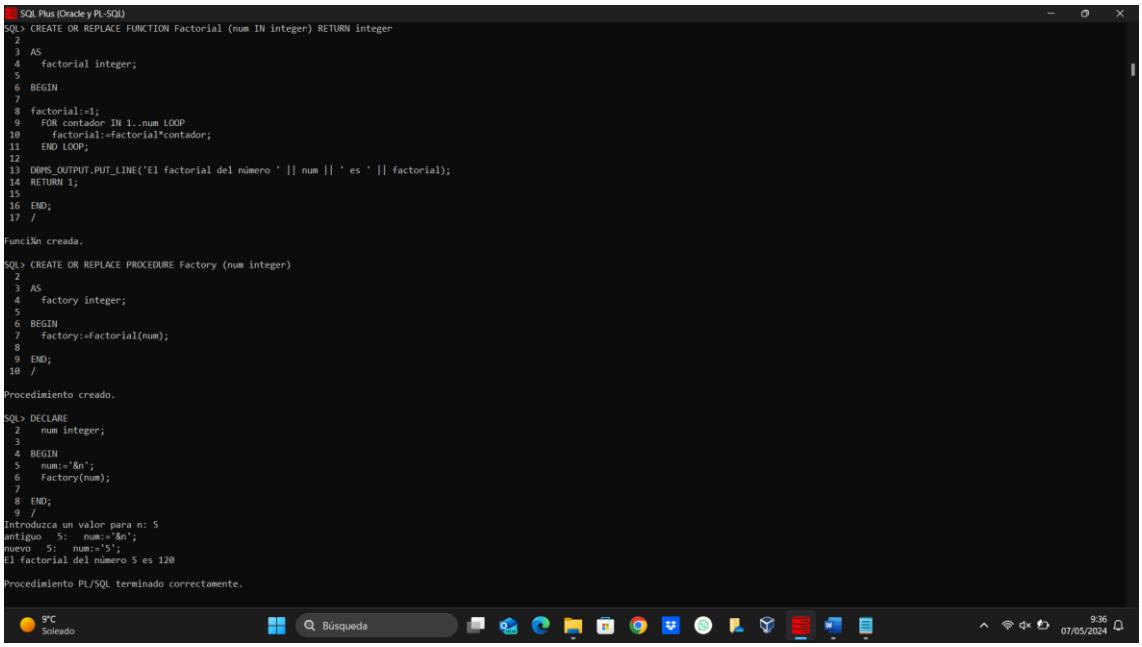


```
SQL Plus (Oracle y PL-SQL)
SQL> CREATE OR REPLACE PROCEDURE list_emp_dep (num_dep integer)
  2
  3  AS
  4   CURSOR listado_dep_emp(num_dep integer) IS
  5    SELECT *
  6    FROM TEMPLE
  7    WHERE Numde=num_dep;
  8
  9  datos TEMPLE%ROWTYPE;
 10
 11  BEGIN
 12    OPEN listado_dep_emp(num_dep);
 13    FETCH listado_dep_emp INTO datos;
 14
 15    WHILE listado_dep_emp%FOUND LOOP
 16      DBMS_OUTPUT.PUT_LINE('Nºempleado: ' || datos.numem ||'  Extensión Telefónica: ' || datos.extel || '   Fecha de Nacimiento: ' || datos.fecna || '   Fecha de Ingreso: ' || datos.fecin || '   Salario: ' ||
| datos.salar || '   Comisión: ' || datos.comis || '   Nºde Hijos: ' || datos.numhi || '   Nombre del empleado: ' || datos.nomem);
 17      FETCH listado_dep_emp INTO datos;
 18    END LOOP;
 19    CLOSE listado_dep_emp;
 20
 21    DBMS_OUTPUT.PUT_LINE(CHR(10));
 22    aumentar_salario(num_dep);
 23
 24  END;
 25  /

Procedimiento creado.

SQL> CREATE OR REPLACE PROCEDURE aumentar_salario (num_dep integer)
  2
  3  AS
  4   CURSOR salario_nuevo_empleados(num_dep integer) IS
  5     SELECT Numem, Salar, Nomem
  6     FROM TEMPLE
  7     WHERE Numde=num_dep;
  8
  9  salario_nuevo integer;
 10  N_Emp integer;
 11  Name_Emp varchar2(20);
 12
 13  BEGIN
 14
 15    UPDATE TEMPLE
 16    SET SALAR = SALAR + 200
 17    WHERE NUMDE=num_dep;
 18
 19    OPEN salario_nuevo_empleados(num_dep);
 20    FETCH salario_nuevo_empleados INTO N_Emp, salario_nuevo, Name_Emp;
```




9.- **Función** que calcule el factorial de un número que se le pasa como parámetro. Se prueba desde un procedimiento y desde un bloque anónimo que utilizarán una variable de sustitución para pedir el número.


SQL Plus (Oracle y PL-SQL)
SQL> CREATE OR REPLACE FUNCTION Factorial (num IN integer) RETURN integer
2
3 AS
4 factorial integer;
5
6 BEGIN
7
8 factorial:=1;
9 FOR contador IN 1..num LOOP
10 factorial:=factorial*contador;
11 END LOOP;
12
13 DBMS_OUTPUT.PUT_LINE('El factorial del número ' || num || ' es ' || factorial);
14 RETURN 1;
15
16 END;
17 /
Funci&n creada.
SQL> CREATE OR REPLACE PROCEDURE Factory (num integer)
2
3 AS
4 factory integer;
5
6 BEGIN
7 factory:=Factorial(num);
8
9 END;
10 /
Procedimiento creado.
SQL> DECLARE
2 num integer;
3
4 BEGIN
5 num:='&n';
6 Factory(num);
7
8 END;
9 /
Introduzca un valor para n: 5
antiguo 5: num:='&n';
nuevo 5: num:='5';
El factorial del número 5 es 120
Procedimiento PL/SQL terminado correctamente.
9°C
Soleado
Búsqueda
9:36
07/05/2024