UNIVERSIDADE
DE VIGO

ESCOLA SUPERIOR DE ENXEÑERÍA INFORMÁTICA

PRÁCTICA 7: PLSQL

GRUPO: BDII4_1

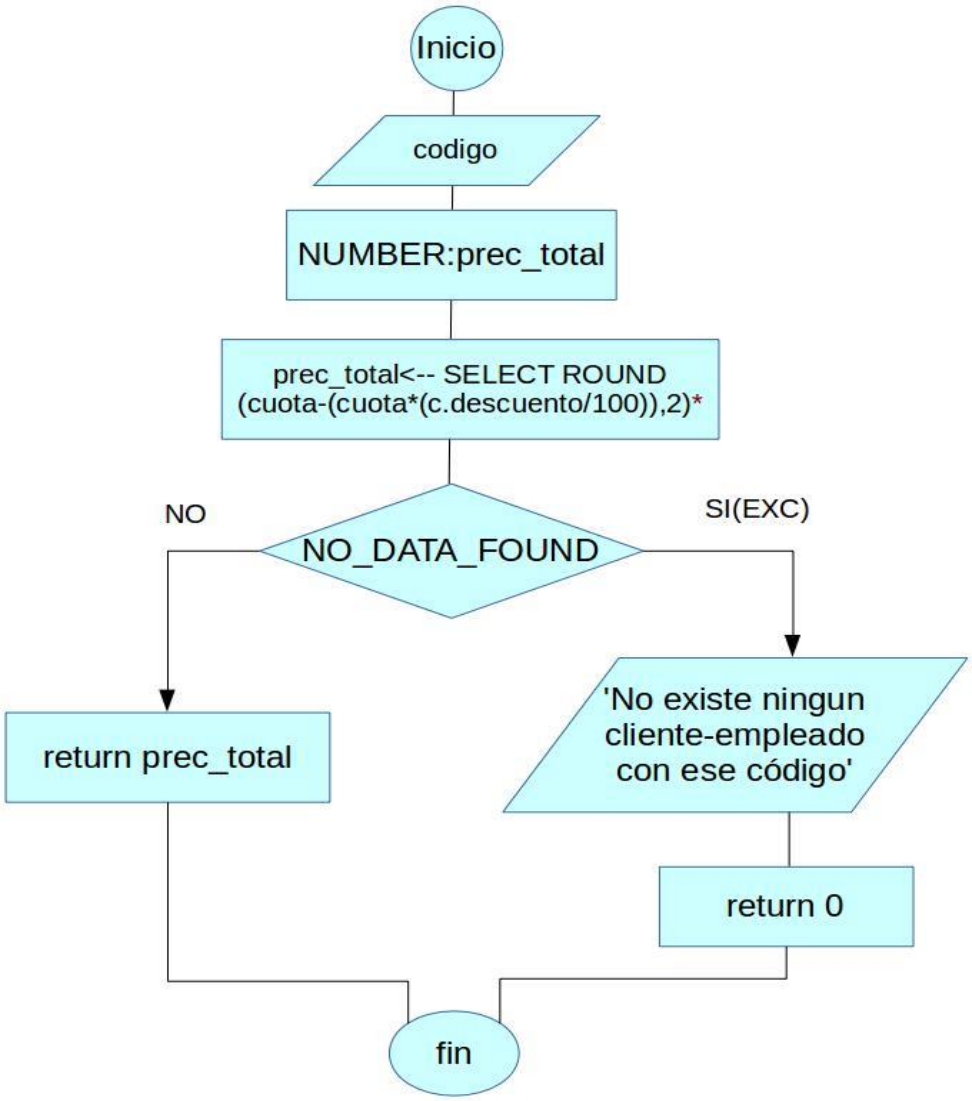
TEMÁTICA: EDIT

DNI: 45147328F NOMBRE: CARRERA GROBA, ABRAHAM
DNI: 44484761R NOMBRE: CRUZ GONZÁLEZ, BRUNO
DNI: 53861461S NOMBRE: FERNÁNDEZ OUTEDA, ATENEA
DNI: 76583535K NOMBRE: RODRÍGUEZ IGLESIAS, SILVIA

BASES DE DATOS II 2016-2017

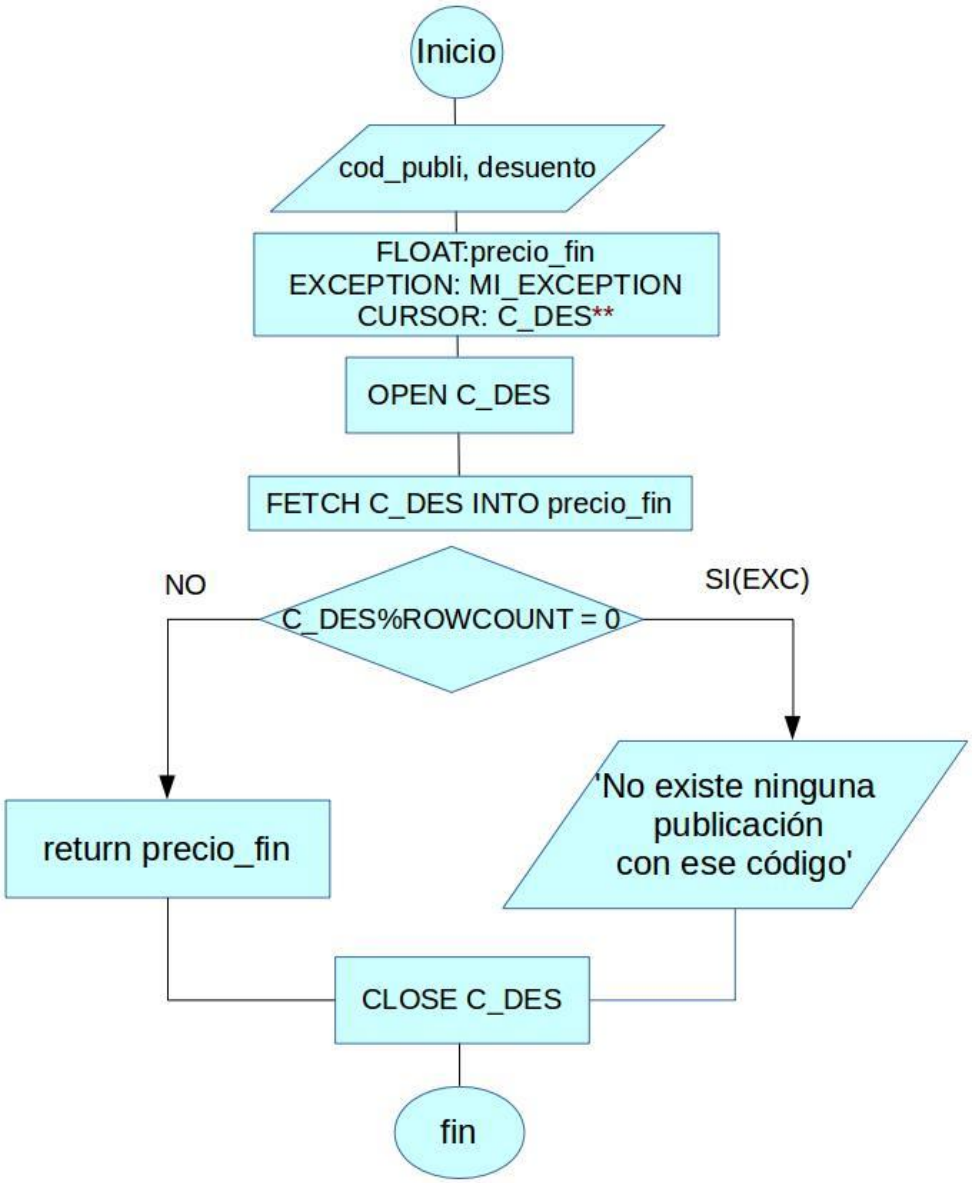
DIAGRAMAS DE FLUJO DE PROCEDIMIENTOS Y FUNCIONES

```
/* Función: descue(codigo IN INT); */
```



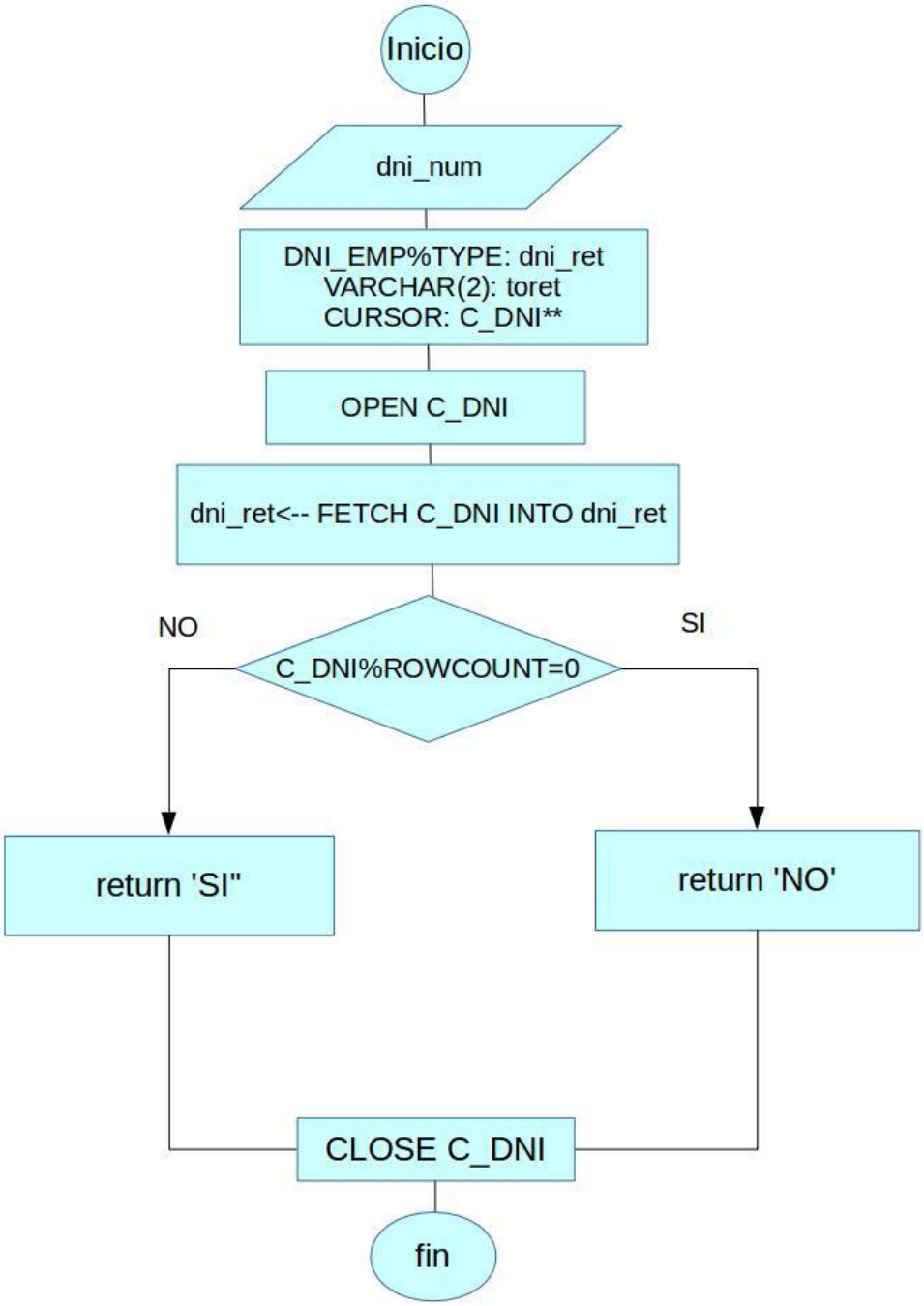

```
*INTO prec_total
   FROM PERSONA_FISICA a, CLIENTE b, CLIENTE_EMP c
   WHERE a.cod_cli=b.cod_cli AND a.dni_p_f=c.dni_cliente_emp AND a.cod_cli=codigo;
```

```
/* Función precio_desc(cod_publi IN INT, descuento IN FLOAT); */
```



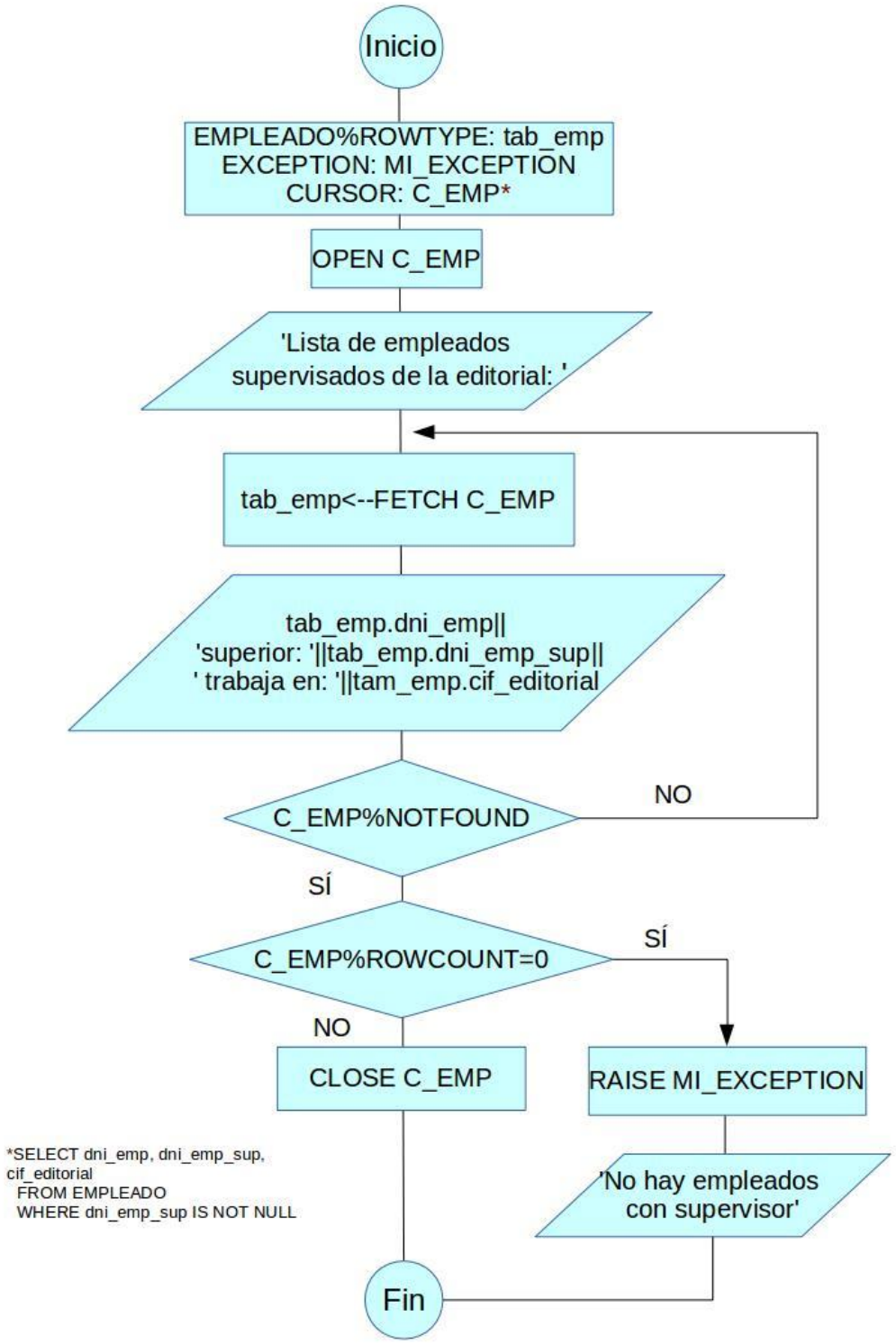

```
**SELECT ROUND(precio
* (1-(descuento/100)),2)
   FROM PUBLICACION
   WHERE
numero_publicacion =
cod_publi;


*INTO precio_fin
  FROM PUBLICACION WHERE
numero_publicacion = cod_publi
```

```
/* Función: dniEmp(dni_num IN VARCHAR2); */
```

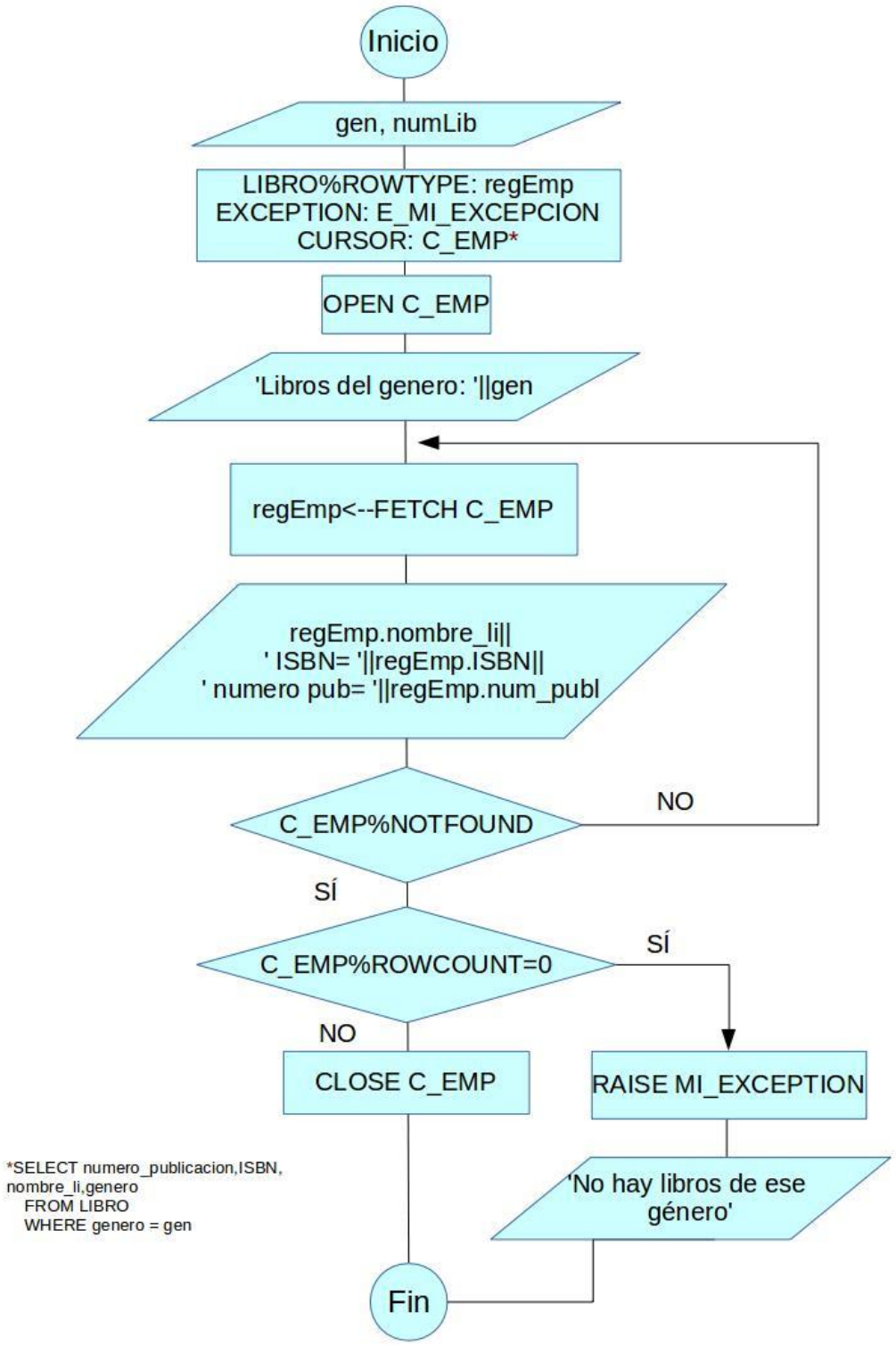


*WHERE dni_emp = dni_num AND dni_emp LIKE ('_____')

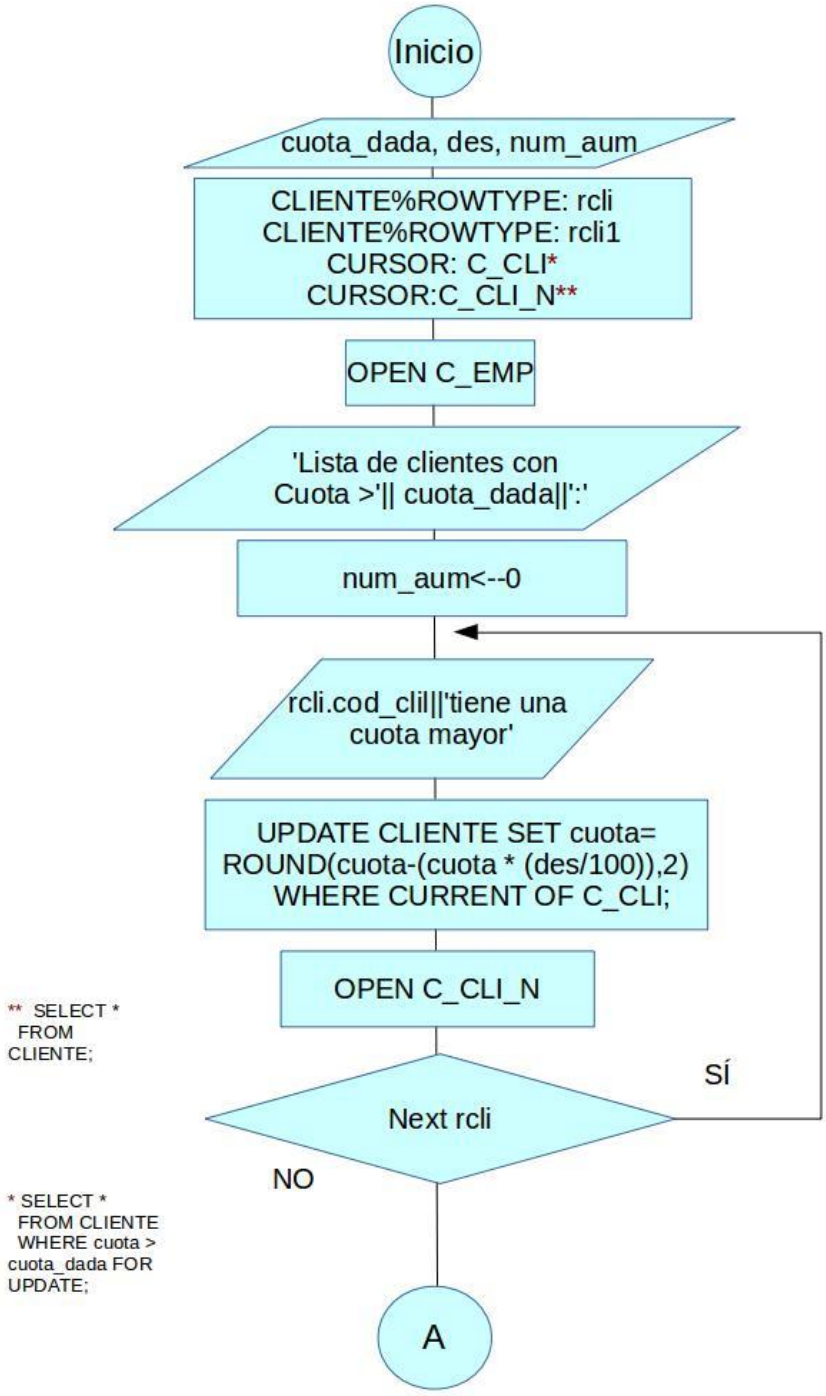
```
/* Procedimiento: ListaEmp(); */
```

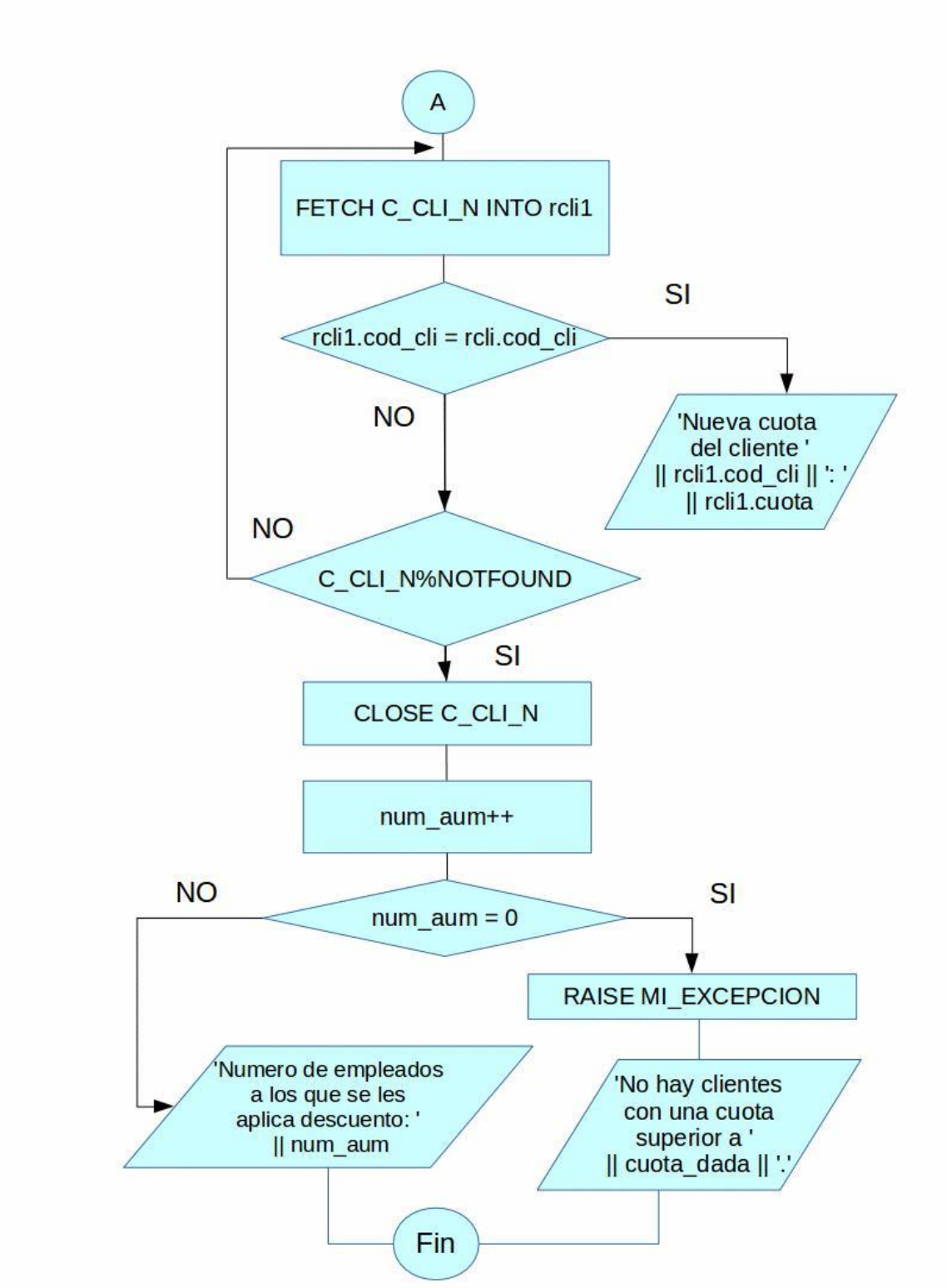


```
/* Procedimiento: ListaLib(gen IN VARCHAR2, numLib OUT NUMBER);
*/
```



```
/* Procedimiento: AumentDesc(cuota_dada IN INT, des IN FLOAT,
num_aum OUT INT); */
```




A
FETCH C_CLI_N INTO rcli1
rcli1.cod_cli = rcli.cod_cli
SI
'Nueva cuota del cliente ' || rcli1.cod_cli || ': ' || rcli1.cuota
NO
NO
C_CLI_N%NOTFOUND
SI
CLOSE C_CLI_N
num_aum++
NO
num_aum = 0
SI
RAISE MI_EXCEPCION
'Numero de empleados a los que se les aplica descuento: ' || num_aum
'No hay clientes con una cuota superior a ' || cuota_dada || '.'
Fin

SENTENCIAS DE DEFINICIÓN DE PROCEDIMIENTOS Y FUNCIONES

```
/*Calcula la cuota que debe pagar un cliente-empleado despues de
aplicarle el descuento correspondiente*/
CREATE OR REPLACE
FUNCTION descue (codigo IN INT)
RETURN NUMBER
IS
  prec_total NUMBER;

BEGIN

    SELECT ROUND(cuota-(cuota*(c.descuento/100)),2) INTO
prec_total
    FROM PERSONA_FISICA a, CLIENTE b, CLIENTE_EMP c
    WHERE a.cod_cli=b.cod_cli AND a.dni_p_f=c.dni_cliente_emp
AND a.cod_cli=codigo;
    RETURN prec_total;


  EXCEPTION WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT('No existe ningun cliente-empleado con ese
código ===> ');
    RETURN 0;

END descue;
/
show errors;


/*Calcula el precio de una publicación tras aplicar un descuento
dado*/
CREATE OR REPLACE
FUNCTION precio_desc (cod_publi IN INT, descuento IN FLOAT)
RETURN FLOAT
IS
  precio_fin FLOAT;
  MI_EXCEPTION EXCEPTION;
  CURSOR C_DES IS
    SELECT ROUND(precio * (1-(descuento/100)),2)
    FROM PUBLICACION
    WHERE numero_publicacion = cod_publi;
BEGIN
  OPEN C_DES;
    FETCH C_DES INTO precio_fin;
    IF (C_DES%ROWCOUNT = 0) THEN
      RAISE MI_EXCEPTION;
    ELSE
      RETURN precio_fin;
    END IF;
  CLOSE C_DES;
```

```
  EXCEPTION WHEN MI_EXCEPTION THEN
    DBMS_OUTPUT.PUT('No existe ninguna publicación con ese
código ===> ');
    RETURN 0;

END precio_desc;
/
show errors;



/*Comprueba si existe un empleado con un dni dado*/
CREATE OR REPLACE
FUNCTION dniEmp (dni_num IN VARCHAR2)
RETURN VARCHAR2
IS
  dni_ret EMPLEADO.DNI_EMP%TYPE;
  toRet VARCHAR2(2);
  CURSOR C_DNI IS
    SELECT dni_emp
    FROM EMPLEADO
    WHERE dni_emp = dni_num AND dni_emp LIKE ('_________');
BEGIN
  OPEN C_DNI;
    FETCH C_DNI INTO dni_ret;
    IF (C_DNI%ROWCOUNT = 0) THEN
      RETURN 'NO';
    ELSE
      RETURN 'SI';
    END IF;
  CLOSE C_DNI;

END dniEmp;
/
show errors;


/*Procedimiento que muestra una lista de empleados que tienen
supervisor.*/
CREATE OR REPLACE
PROCEDURE ListaEmp
IS
  tab_emp EMPLEADO%ROWTYPE;
  MI_EXCEPTION EXCEPTION;

CURSOR C_EMP IS
  SELECT dni_emp, dni_emp_sup, cif_editorial
  FROM EMPLEADO
  WHERE dni_emp_sup IS NOT NULL;

BEGIN
  OPEN C_EMP;
  DBMS_OUTPUT.PUT_LINE('Lista de empleados supervisados de la
editorial: ');
  LOOP
```

```
  FETCH C_EMP INTO tab_emp;
  EXIT WHEN C_EMP%NOTFOUND;
  DBMS_OUTPUT.PUT_LINE(tab_emp.dni_emp || ' superior: ' ||
tab_emp.dni_emp_sup || ' trabaja en: ' ||
tab_emp.cif_editorial);
  END LOOP;
  IF (C_EMP%ROWCOUNT = 0) THEN
    RAISE MI_EXCEPTION;
  END IF;
  CLOSE C_EMP;


EXCEPTION
  WHEN MI_EXCEPTION THEN
    DBMS_OUTPUT.PUT_LINE('No hay empleados con supervisor.');
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Codigo: ' || SQLCODE);
    DBMS_OUTPUT.PUT_LINE('[Mensaje]: ' || SUBSTR(SQLERRM, 11,
100));
END ListaEmp;
/
show errors;



/*Muestra el listado de libros pertenecientes a un género
dado.*/
CREATE OR REPLACE
PROCEDURE ListaLib(gen IN VARCHAR2, numLib OUT NUMBER)
IS
   regEmp LIBRO%ROWTYPE;
   E_MI_EXCEPCION EXCEPTION;

  CURSOR C_EMP IS
    SELECT numero_publicacion,ISBN, nombre_li,genero
    FROM LIBRO
    WHERE genero = gen;
BEGIN
   OPEN C_EMP;
   DBMS_OUTPUT.PUT_LINE('Libros del genero: "' || gen || '"');
   LOOP
      FETCH C_EMP INTO regEmp;
      EXIT WHEN C_EMP%NOTFOUND;
      DBMS_OUTPUT.PUT(regEmp.nombre_li || ', ISBN=' ||
regEmp.ISBN || ', Numero de publicacion=' ||
regEmp.numero_publicacion );
      DBMS_OUTPUT.PUT_LINE('');
   END LOOP;

   numLib := C_EMP%ROWCOUNT;
   IF (numLib = 0) THEN
      RAISE E_MI_EXCEPCION;
   END IF;

   CLOSE C_EMP;
```

```
EXCEPTION
   WHEN E_MI_EXCEPCION THEN
      DBMS_OUTPUT.PUT_LINE('No hay libros de ese género.');
   WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('Código: ' || SQLCODE);
      DBMS_OUTPUT.PUT_LINE('[Mensaje]: ' || SUBSTR(SQLERRM, 11,
100));
END ListaLib;
/
show errors


/*Procedimiento que muestra una lista de clientes cuya cuota es
mayor a una dada y actualiza automaticamente la cuota que deben
pagar despues de aplicarle un descuento.*/
CREATE OR REPLACE
PROCEDURE AumentDesc (cuota_dada IN INT, des IN FLOAT, num_aum
OUT INT)
IS
  rcli CLIENTE%ROWTYPE;
  rcli1 CLIENTE%ROWTYPE;
  MI_EXCEPTION EXCEPTION;

CURSOR C_CLI IS
  SELECT *
  FROM CLIENTE
  WHERE cuota > cuota_dada FOR UPDATE;

CURSOR C_CLI_N IS
  SELECT *
  FROM CLIENTE;

BEGIN
  DBMS_OUTPUT.PUT_LINE('Lista de código de clientes con cuota
mayor que ' || cuota_dada || ': ');
  num_aum := 0;
  FOR rcli IN C_CLI LOOP
    DBMS_OUTPUT.PUT_LINE(rcli.cod_cli || ' tiene una cuota de: '
|| rcli.cuota);
    UPDATE CLIENTE SET cuota=ROUND(cuota-(cuota * (des/100)),2)
    WHERE CURRENT OF C_CLI;
    OPEN C_CLI_N;
    LOOP
      FETCH C_CLI_N INTO rcli1;
      EXIT WHEN C_CLI_N%NOTFOUND;
        IF (rcli1.cod_cli = rcli.cod_cli) THEN
          DBMS_OUTPUT.PUT_LINE('Nueva cuota del cliente ' ||
rcli1.cod_cli || ': ' || rcli1.cuota);
        END IF;
      END LOOP;
    CLOSE C_CLI_N;

    num_aum := num_aum + 1;
  END LOOP;

  IF (num_aum = 0) THEN
```

```
    RAISE MI_EXCEPTION;
  END IF;

  DBMS_OUTPUT.PUT_LINE('Numero de empleados a los que se les
aplica descuento: ' || num_aum);

  EXCEPTION
   WHEN MI_EXCEPTION THEN
      DBMS_OUTPUT.PUT_LINE('No hay clientes con una cuota
superior a ' || cuota_dada || '.');
   WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('Código: ' || SQLCODE);
      DBMS_OUTPUT.PUT_LINE('[Mensaje]: ' || SUBSTR(SQLERRM, 11,
100));

END AumentDesc;
/
show errors;
```

SENTENCIAS DE DEFINICIÓN DE BLOQUES DE PRUEBAS

```
SET SERVEROUTPUT ON
DECLARE
  toRet FLOAT;
  toRet1 VARCHAR2(2);
  toRet2 FLOAT;
  codigo INT;
  descuento FLOAT;
  dni VARCHAR2(9);

BEGIN
  DBMS_OUTPUT.NEW_LINE;

--FUNCIONES
  DBMS_OUTPUT.PUT_LINE('======>INICIO FUNCION: desc');
  codigo := 24;
  toRet:= descue(codigo);
  DBMS_OUTPUT.PUT_LINE('Cuota con descuento del empleado ' ||
codigo ||  ': ' || toRet);
  codigo := 0;
  toRet:= descue(codigo);
  DBMS_OUTPUT.PUT_LINE('Cuota con descuento del empleado ' ||
codigo ||  ': ' || toRet);
  DBMS_OUTPUT.PUT_LINE('======>FIN FUNCION: desc');
  DBMS_OUTPUT.NEW_LINE;
  DBMS_OUTPUT.PUT_LINE('======>INICIO FUNCION: precio_desc');
  codigo := 102;
  descuento := 2.5;
  toRet:= precio_desc(codigo,descuento);
  DBMS_OUTPUT.PUT_LINE('Precio de la publicacion ' || codigo ||
' con el descuento del ' || descuento ||  ': ' || toRet);
  codigo := 1;
  descuento := 12;
  toRet:= precio_desc(codigo,descuento);
  DBMS_OUTPUT.PUT_LINE('Precio de la publicacion ' || codigo ||
' con el descuento del ' || descuento ||  ': ' || toRet);
  DBMS_OUTPUT.PUT_LINE('======>FIN FUNCION: precio_desc');
  DBMS_OUTPUT.NEW_LINE;
  DBMS_OUTPUT.PUT_LINE('======>INICIO FUNCION: dniEmp');
  dni := '76583535K';
  toRet1 := dniEmp(dni);
  DBMS_OUTPUT.PUT_LINE('Existe dni ' || dni || ': ' || toRet1);
  dni := '46475836A';
  toRet1 := dniEmp(dni);
  DBMS_OUTPUT.PUT_LINE('Existe dni ' || dni || ': ' || toRet1);
  DBMS_OUTPUT.PUT_LINE('======>FIN FUNCION: dniEmp');
  DBMS_OUTPUT.NEW_LINE;

-- PROCEDIMIENTOS
  DBMS_OUTPUT.PUT_LINE('======>INICIO PROCEDIMIENTO: ListaEmp');
  ListaEmp();
  DBMS_OUTPUT.PUT_LINE('======>FIN PROCEDIMIENTO: ListaEmp');
```

```
  DBMS_OUTPUT.NEW_LINE;
  DBMS_OUTPUT.PUT_LINE('======>INICIO PROCEDIMIENTO: ListaLib');
  ListaLib('Ficcion', toRet2);
  DBMS_OUTPUT.NEW_LINE;
  ListaLib('Accion', toRet2);
  DBMS_OUTPUT.PUT_LINE('======>FIN PROCEDIMIENTO: ListaLib');
  DBMS_OUTPUT.NEW_LINE;
  DBMS_OUTPUT.PUT_LINE('======>INICIO PROCEDIMIENTO:
AumentDesc');
  AumentDesc(6,15,toRet2);
  DBMS_OUTPUT.PUT_LINE('======>FIN PROCEDIMIENTO: AumentDesc');
  DBMS_OUTPUT.NEW_LINE;

EXCEPTION
   WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('[EXCEPCIÓN]');
      DBMS_OUTPUT.PUT_LINE('[Código]: ' || SQLCODE);
      DBMS_OUTPUT.PUT_LINE('[Mensaje]: ' || SUBSTR(SQLERRM, 11,
100));

END;
/
```

HOJA DE FIRMAS

DNI: 45147328F NOMBRE: CARRERA GROBA, ABRAHAM

DNI: 44484761R NOMBRE: CRUZ GONZÁLEZ, BRUNO

DNI: 53861461S NOMBRE: FERNÁNDEZ OUTEDA, ATENEA

DNI: 76583535K NOMBRE: RODRÍGUEZ IGLESIAS, SILVIA