3. Construye un bloque PL/SQL que pida el precio de un producto y el valor del IVA a

aplicarle, e imprima el valor del total del producto.

Para que salga en pantalla→ **set serveroutput on;**

declare

nom\_art articulos.articulo%type;

precio articulos.precio\_venta%type;

iva number(10):= 1.21;

total number(10);

begin

select precio\_venta into precio from articulos where cod\_fabricante=25;

total:= precio\*iva;

dbms\_output.put\_line(nom\_art || ' cuesta ' || total || '€');

end;

4. Construye un bloque PL/SQL que seleccione el artículo de mayor PVP en la tabla

ARTICULOS y almacene su valor en una variable de SQL para imprimirlo a

continuación.

Para que salga en pantalla→ **set serveroutput on;**

declare

nom\_art articulos.articulo%type;

precio articulos.precio\_venta%type;

begin

select max(precio\_costo) into precio from articulos;

dbms\_output.put\_line(nom\_art || ' cuesta ' || precio || '€');

end;

5. Crea y almacena en un fichero un bloque PL/SQL que inserte un nuevo artículo en la

tabla ARTICULOS. Los datos del código y nombre deben pedirse previamente por

teclado para darlos como argumentos.

CREATE OR REPLACE PROCEDURE insertar\_art (

nom\_art VARCHAR2,

fabr NUMBER,

peso NUMBER,

cate VARCHAR2,

preven NUMBER,

precos NUMBER,

exi NUMBER)

as

BEGIN

INSERT INTO ARTICULOS (ARTICULO, COD\_FABRICANTE, PESO, CATEGORIA, PRECIO\_VENTA, PRECIO\_COSTO, EXISTENCIAS) VALUES (nom\_art, fabr, peso, cate, preven, precos, exi);

DBMS\_OUTPUT.PUT\_LINE('Se ha introducido nuevo producto ' || nom\_art);

END insertar\_art;

**ejecutar el procedimiento**

Para que salga en pantalla→ **set serveroutput on;**

execute insertar\_art('leche', 20, 2, 'Primera', 50, 40, 33);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Otra forma de hacer el 5**

**Ejercicio 5**

Declare

codigo VARCHAR2(5) := &codigo;

precio NUMBER(38) := &precio;

descri VARCHAR2(30) := &descri;

exist NUMBER(38) := &exist;

Begin

INSERT INTO ARTICULO (CODARTI, PRECIOARTI, DESCARTI, EXISTARTI)

VALUES (codigo, precio, descri, exist);

DBMS\_OUTPUT.PUT\_LINE('Articulo insertado: ');

END;

6. Un bloque PL/SQL debe actualizar la dirección de un cliente en la tabla CLIENTES.

Para ello, el bloque pedirá el código del cliente y su nueva dirección.

Declare

nif number(38) := &nif;

nueva\_dir VARCHAR2(30) := &nueva\_dir;

Begin

UPDATE Cliente

set DIRCLI = nueva\_dir

where NIFCLI = nif;

DBMS\_OUTPUT.PUT\_LINE('La nueva dirección es: ' || nueva\_dir);

END;

7. Crea un bloque que borre todos los proveedores de un país que se pedirá por teclado.

Debe imprimirse el número de proveedores que se han borrado.

**Practica 2**

1. Crea una tabla MENSAJES con un solo campo VALOR, de tipo VARCHAR2(5). Crea

un bloque que inserte 8 elementos en la tabla con valores del 1 al 10, excepto el 4 y 5.

CREATE TABLE MENSAJES

(

VALOR VARCHAR2(5)NOT NULL

);

BEGIN

FOR numero IN 1..10 LOOP

IF numero <> 4 AND numero <> 5 THEN

INSERT INTO MENSAJES VALUES (numero);

END IF;

END LOOP;

END;

2.

Un bloque PL/SQL debe aceptar el código de un artículo y actualizar su precio según las

siguientes indicaciones. Si el artículo tiene un PVP menor a 100 ptas., su precio debe ser

aumentado en 5 ptas. Si está entre 100 y 1.000 ptas. su precio subirá 15 y si excede las

1.000 ptas., el precio del producto debe subir 50 ptas. Si el PVP fuese NULL, el

aumento sería 0.

ALTER TABLE ARTICULOS

ADD ID\_COD VARCHAR2(10) ;

INSERT INTO ARTICULOS (ARTICULO, COD\_FABRICANTE, PESO,CATEGORIA, PRECIO\_VENTA, PRECIO\_COSTO, EXISTENCIAS, ID\_COD)

VALUES ('MIEL', 2, 3, 'Primera', 20, 5, 34, 263);

DECLARE

codart [articulos.id](http://articulos.id/)%TYPE :=&codart;

precio articulos.precio\_costo%TYPE;

precio\_nuevo articulos.precio\_costo%TYPE;

pvp articulos.precio\_venta%TYPE;

BEGIN

SELECT precio\_venta INTO pvp FROM ARTICULOS where id=codart;

SELECT precio\_costo INTO precio FROM ARTICULOS where id=codart;

IF (pvp < 100) THEN

precio\_nuevo := precio + 5;

ELSIF (pvp BETWEEN 100 AND 1000) THEN

precio\_nuevo := precio + 15;

ELSIF (pvp > 1000) THEN

precio\_nuevo := precio + 50;

ELSE

precio\_nuevo := precio;

END IF;

UPDATE ARTICULOS SET precio\_costo=precio\_nuevo where id=codart;

DBMS\_OUTPUT.PUT\_LINE('Se ha actualizado el precio de art nº' || codart || '. Precio anterior: ' || precio || '. Precio nuevo: ' || precio\_nuevo);

END;