

**CREACIÓN DE OBJETOS**

- Crearemos los siguientes objetos:

**Objeto: DATOSCOMUNES**

```
CREATE OR REPLACE TYPE DATOSCOMUNES AS OBJECT
(
  DNI          VARCHAR2(9),
  NOMBRE       VARCHAR2(50),
  APELLIDOS    VARCHAR2(100),
  DIRECCION    VARCHAR2(100),
  POBLACION    VARCHAR2(150),
  EMAIL        VARCHAR2(150),
  TELEFONO     NUMBER(9)
) NOT FINAL;
```

**Objeto: ARTICULO**

```
CREATE OR REPLACE TYPE ARTICULO IS OBJECT
(
  IDARTICULO      NUMBER(5),
  NOMBREARTICULO  VARCHAR2(50),
  PRECIOUNITARIO  NUMBER(6,2),
  UNIDADESEXISTENTES NUMBER(4),

  CONSTRUCTOR FUNCTION ARTICULO(IDARTICULO NUMBER,NOMBREARTICULO VARCHAR2,
    DESCRIPCION VARCHAR2,PRECIOUNITARIO NUMBER,UNIDADESEXITENTES NUMBER)
    RETURN SELF AS RESULT,
  CONSTRUCTOR FUNCTION ARTICULO(IDARTICULO NUMBER, NOMBREARTICULO VARCHAR2,
    PRECIOUNITARIO NUMBER)
    RETURN SELF AS RESULT,
  CONSTRUCTOR FUNCTION ARTICULO RETURN SELF AS RESULT,

  MEMBER PROCEDURE SETIDARTICULO(IDARTICULO NUMBER),
  MEMBER FUNCTION GETIDARTICULO RETURN NUMBER,
  MEMBER FUNCTION GETNOMBREARTICULO RETURN VARCHAR2,
  MEMBER PROCEDURE SETNOMBREARTICULO(NOMBREARTICULO VARCHAR2),
  MEMBER PROCEDURE SETPRECIOUNITARIO(PRECIOUNITARIO NUMBER),
  MEMBER FUNCTION GETPRECIOUNITARIO RETURN NUMBER,
  MEMBER FUNCTION GETUNIDADESEXISTENTE RETURN NUMBER,
  MEMBER PROCEDURE SETUNIDADESEXISTENTES(UNIDADESEXISTENTES NUMBER)
);
```

```
CREATE OR REPLACE TYPE BODY ARTICULO AS
```

```
CONSTRUCTOR FUNCTION ARTICULO(IDARTICULO NUMBER,NOMBREARTICULO VARCHAR2,
  DESCRIPCION VARCHAR2,PRECIOUNITARIO NUMBER,UNIDADESEXITENTES NUMBER)
  RETURN SELF AS RESULT AS
BEGIN
  SELF.SETIDARTICULO(IDARTICULO);
  SELF.SETNOMBREARTICULO(UPPER(NOMBREARTICULO)||+ '-' ||UPPER(DESCRIPCION));
  SELF.SETPRECIOUNITARIO(PRECIOUNITARIO);
  SELF.SETUNIDADESEXISTENTES(UNIDADESEXISTENTES);
  RETURN;
END ARTICULO;
```

```

CONSTRUCTOR    FUNCTION ARTICULO(IDARTICULO  NUMBER, NOMBREARTICULO VARCHAR2,
                                PRECIOUNITARIO NUMBER)
    RETURN SELF AS RESULT AS
BEGIN
    SELF.SETIDARTICULO(IDARTICULO);
    SELF.SETNOMBREARTICULO(UPPER(NOMBREARTICULO));
    SELF.SETPRECIOUNITARIO(PRECIOUNITARIO);
    SELF.SETUNIDADESEXISTENTES(0);
    RETURN;
END ARTICULO;

```

```

CONSTRUCTOR    FUNCTION ARTICULO          RETURN SELF AS RESULT AS
BEGIN
    SELF.SETIDARTICULO(NULL);
    SELF.SETNOMBREARTICULO(NULL);
    SELF.SETPRECIOUNITARIO(NULL);
    SELF.SETUNIDADESEXISTENTES(NULL);
    RETURN;
END;
MEMBER FUNCTION GETIDARTICULO RETURN NUMBER AS
BEGIN
    RETURN SELF.IDARTICULO;
END GETIDARTICULO;

```

```

MEMBER PROCEDURE SETIDARTICULO(IDARTICULO NUMBER) AS
BEGIN
    SELF.IDARTICULO:=IDARTICULO;
END SETIDARTICULO;

```

```

MEMBER FUNCTION GETNOMBREARTICULO RETURN VARCHAR2 AS
BEGIN
    RETURN SELF.NOMBREARTICULO;
END GETNOMBREARTICULO;

```

```

MEMBER PROCEDURE SETNOMBREARTICULO(NOMBREARTICULO VARCHAR2) AS
BEGIN
    SELF.NOMBREARTICULO:=UPPER(NOMBREARTICULO);
END SETNOMBREARTICULO;

```

```

MEMBER FUNCTION GETPRECIOUNITARIO RETURN NUMBER AS
BEGIN

    RETURN SELF.PRECIOUNITARIO;
END GETPRECIOUNITARIO;

```

```

MEMBER PROCEDURE SETPRECIOUNITARIO(PRECIOUNITARIO NUMBER) AS
BEGIN
    SELF.PRECIOUNITARIO:=PRECIOUNITARIO;
END SETPRECIOUNITARIO;

```

```

MEMBER FUNCTION GETUNIDADESEXISTENTE RETURN NUMBER AS
BEGIN

    RETURN UNIDADESEXISTENTES;
END GETUNIDADESEXISTENTE;

```

```

MEMBER PROCEDURE SETUNIDADESEXISTENTES(UNIDADESEXISTENTES NUMBER) AS
BEGIN
    SELF.UNIDADESEXISTENTES:=UNIDADESEXISTENTES;
END SETUNIDADESEXISTENTES;

```

```
END;
```

### **Objeto: CLIENTE**

```

CREATE OR REPLACE TYPE CLIENTE UNDER DATOSCOMUNES
(
    IDCLIENTE    NUMBER(6),
    FECHAALTA    DATE,
    TIPO         NUMBER(1),

    CONSTRUCTOR FUNCTION CLIENTE          RETURN SELF AS    RESULT,

    CONSTRUCTOR FUNCTION CLIENTE(DNI VARCHAR2,NOMBRE VARCHAR2, APELLIDO1 VARCHAR2,
        APELLIDO2 VARCHAR2, DIRECCION VARCHAR2, POBLACION VARCHAR2,
        EMAIL VARCHAR2,TELEFONO NUMBER,
        IDCLIENTE NUMBER,FECHAALTA DATE,TIPO NUMBER)
        RETURN SELF AS    RESULT,

    CONSTRUCTOR FUNCTION CLIENTE(DNI VARCHAR2,NOMBRE VARCHAR2, APELLIDOS VARCHAR2,
        DIRECCION VARCHAR2, POBLACION VARCHAR2,
        EMAIL VARCHAR2,TELEFONO NUMBER,
        IDCLIENTE NUMBER,FECHAALTA DATE,TIPO NUMBER)
        RETURN SELF AS    RESULT,

    CONSTRUCTOR FUNCTION CLIENTE(COMUNES DATOSCOMUNES ,IDCLIENTE NUMBER,FECHAALTA
DATE,TIPO NUMBER)
        RETURN SELF AS    RESULT,

    MEMBER FUNCTION GETDNI RETURN VARCHAR2,
    MEMBER PROCEDURE SETDNI(DNI VARCHAR2),
    MEMBER FUNCTION GETNOMBRE RETURN VARCHAR2,
    MEMBER PROCEDURE SETNOMBRE(NOMBRE VARCHAR2),
    MEMBER FUNCTION GETAPELLIDOS RETURN VARCHAR2,
    MEMBER PROCEDURE SETAPELLIDOS(APELLIDOS VARCHAR2),
    MEMBER FUNCTION GETDIRECCION RETURN VARCHAR2,
    MEMBER PROCEDURE SETDIRECCION(DIRECCION VARCHAR2),
    MEMBER FUNCTION GETPOBLACION RETURN VARCHAR2,
    MEMBER PROCEDURE SETPOBLACION(POBLACION VARCHAR2),
    MEMBER FUNCTION GETEMAIL RETURN VARCHAR2,
    MEMBER PROCEDURE SETEMAIL(EMAIL VARCHAR2),
    MEMBER FUNCTION GETTELEFONO RETURN NUMBER,
    MEMBER PROCEDURE SETTELEFONO(TELEFONO NUMBER),
    MEMBER FUNCTION GETIDCLIENTE RETURN NUMBER,
    MEMBER PROCEDURE SETIDCLIENTE(IDCLIENTE NUMBER),
    MEMBER FUNCTION GETFECHAALTA RETURN DATE,
    MEMBER PROCEDURE SETFECHAALTA (FECHAALTA DATE),
    MEMBER FUNCTION GETTIPO RETURN NUMBER,
    MEMBER PROCEDURE SETTIPO(TIPO NUMBER),
    MEMBER FUNCTION NOMBRECOMPLETO RETURN VARCHAR2
);

```

CREATE OR REPLACE TYPE **BODY CLIENTE** AS

```
CONSTRUCTOR FUNCTION CLIENTE      RETURN SELF  AS      RESULT AS
BEGIN
    SELF.SETDNI(NULL);
    SELF.SETNOMBRE(NULL);
    SELF.SETAPELLIDOS(NULL);
    SELF.SETDIRECCION(NULL);
    SELF.SETPOBLACION(NULL);
    SELF.SETEMAIL(NULL);
    SELF.SETTELEFONO(NULL);
    SELF.SETIDCLIENTE(NULL);
    SELF.SETFECHAALTA(NULL);
    SELF.SETTIPO(NULL);
    RETURN;
END CLIENTE;
```

```
CONSTRUCTOR FUNCTION CLIENTE(DNI VARCHAR2,NOMBRE VARCHAR2, APELLIDO1 VARCHAR2,
APELLIDO2 VARCHAR2,DIRECCION VARCHAR2, POBLACION VARCHAR2,
EMAIL VARCHAR2,TELEFONO NUMBER,
IDCLIENTE NUMBER,FECHAALTA DATE,TIPO NUMBER)
    RETURN SELF  AS      RESULT AS
BEGIN
    SELF.SETDNI(DNI);
    SELF.SETNOMBRE(NOMBRE);
    SELF.SETAPELLIDOS(APELLIDO1||' '||APELLIDO2);
    SELF.SETDIRECCION(DIRECCION);
    SELF.SETPOBLACION(POBLACION);
    SELF.SETEMAIL(EMAIL);
    SELF.SETTELEFONO(TELEFONO);
    SELF.SETIDCLIENTE(IDCLIENTE);
    SELF.SETFECHAALTA(FECHAALTA);
    SELF.SETTIPO(TIPO);
    RETURN;
END CLIENTE;
```

```
CONSTRUCTOR FUNCTION CLIENTE(DNI VARCHAR2,NOMBRE VARCHAR2, APELLIDOS VARCHAR2,
DIRECCION VARCHAR2, POBLACION VARCHAR2,
EMAIL VARCHAR2,TELEFONO NUMBER,
IDCLIENTE NUMBER,FECHAALTA DATE,TIPO NUMBER)
    RETURN SELF  AS      RESULT AS
BEGIN
    SELF.SETDNI(DNI);
    SELF.SETNOMBRE(NOMBRE);
    SELF.SETAPELLIDOS(APELLIDOS);
    SELF.SETDIRECCION(DIRECCION);
    SELF.SETPOBLACION(POBLACION);
    SELF.SETEMAIL(EMAIL);
    SELF.SETTELEFONO(TELEFONO);
    SELF.SETIDCLIENTE(IDCLIENTE);
    SELF.SETFECHAALTA(FECHAALTA);
    SELF.SETTIPO(TIPO);
    RETURN;
END CLIENTE;
```

```
CONSTRUCTOR FUNCTION CLIENTE(COMUNES DATOSCOMUNES,
IDCLIENTE NUMBER,FECHAALTA DATE,TIPO NUMBER)
    RETURN SELF  AS      RESULT AS
```

```

BEGIN
    SELF.SETDNI(COMUNES.DNI);
    SELF.SETNOMBRE(COMUNES.NOMBRE);
    SELF.SETAPELLIDOS(COMUNES.APELLIDOS);
    SELF.SETDIRECCION(COMUNES.DIRECCION);
    SELF.SETPOBLACION(COMUNES.POBLACION);
    SELF.SETEMAIL(COMUNES.EMAIL);
    SELF.SETTELEFONO(COMUNES.TELEFONO);
    SELF.SETIDCLIENTE(IDCLIENTE);
    SELF.SETFECHAALTA(FECHAALTA);
    SELF.SETTIPO(TIPO);
    RETURN;
END CLIENTE;

MEMBER FUNCTION GETDNI RETURN VARCHAR2 AS
BEGIN
    RETURN SELF.DNI;
END GETDNI;

MEMBER PROCEDURE SETDNI(DNI VARCHAR2) AS
BEGIN
    SELF.DNI:=UPPER(DNI);
END SETDNI;

MEMBER FUNCTION GETNOMBRE RETURN VARCHAR2 AS
BEGIN
    RETURN SELF.NOMBRE;
END GETNOMBRE;

MEMBER PROCEDURE SETNOMBRE(NOMBRE VARCHAR2) AS
BEGIN
    SELF.NOMBRE:=UPPER(NOMBRE);
END SETNOMBRE;

MEMBER FUNCTION GETAPELLIDOS RETURN VARCHAR2 AS
BEGIN
    RETURN SELF.APELLIDOS;
END GETAPELLIDOS;

MEMBER PROCEDURE SETAPELLIDOS(APELLIDOS VARCHAR2) AS
BEGIN
    SELF.APELLIDOS:=UPPER(APELLIDOS);
END SETAPELLIDOS;

MEMBER FUNCTION GETDIRECCION RETURN VARCHAR2 AS
BEGIN
    RETURN SELF.DIRECCION;
END GETDIRECCION;

MEMBER PROCEDURE SETDIRECCION(DIRECCION VARCHAR2) AS
BEGIN
    SELF.DIRECCION:=UPPER(DIRECCION);
END SETDIRECCION;

MEMBER FUNCTION GETPOBLACION RETURN VARCHAR2 AS
BEGIN
    RETURN SELF.POBLACION;
END GETPOBLACION;

```

```

MEMBER PROCEDURE SETPOBLACION(POBLACION VARCHAR2) AS
BEGIN
    SELF.POBLACION:=UPPER(POBLACION);
END SETPOBLACION;

MEMBER FUNCTION GETEMAIL RETURN VARCHAR2 AS
BEGIN
    RETURN SELF.EMAIL;
END GETEMAIL;

MEMBER PROCEDURE SETEMAIL(EMAIL VARCHAR2) AS
BEGIN
    SELF.EMAIL:=UPPER(EMAIL);
END SETEMAIL;

MEMBER FUNCTION GETTELEFONO RETURN NUMBER AS
BEGIN
    RETURN SELF.TELEFONO;
END GETTELEFONO;

MEMBER PROCEDURE SETTELEFONO(TELEFONO NUMBER) AS
BEGIN
    SELF.TELEFONO:=TELEFONO;
END SETTELEFONO;

MEMBER FUNCTION GETIDCLIENTE RETURN NUMBER AS
BEGIN
    RETURN SELF.IDCLIENTE;
END GETIDCLIENTE;

MEMBER PROCEDURE SETIDCLIENTE(IDCLIENTE NUMBER) AS
BEGIN
    SELF.IDCLIENTE:=IDCLIENTE;
END SETIDCLIENTE;

MEMBER FUNCTION GETFECHAALTA RETURN DATE AS
BEGIN
    RETURN SELF.FECHAALTA;
END GETFECHAALTA;

MEMBER PROCEDURE SETFECHAALTA (FECHAALTA DATE) AS
BEGIN
    SELF.FECHAALTA:=FECHAALTA;
END SETFECHAALTA;

MEMBER FUNCTION GETTIPO RETURN NUMBER AS
BEGIN
    RETURN SELF.TIPO;
END GETTIPO;

MEMBER PROCEDURE SETTIPO(TIPO NUMBER) AS
BEGIN
    SELF.TIPO:=TIPO;
END SETTIPO;

MEMBER FUNCTION NOMBRECOMPLETO RETURN VARCHAR2 AS
BEGIN
    RETURN SELF.APELLIDOS||' ' , '||SELF.NOMBRE;
END NOMBRECOMPLETO;
END;

```

- Crearemos un Objeto: **LINEAVENTA** para almacenar cada venta de un artículo:

```
CREATE OR REPLACE TYPE LINEAVENTA IS OBJECT (
    UNIDADES NUMBER,
    LOSARTICULOS ARTICULO
);
```

- Crearemos un **VARRAY** de hasta 20 **LINEAVENTA** (una por cada artículo que forme una venta):

```
CREATE OR REPLACE TYPE LISTAARTICULOS IS VARRAY(20) OF LINEAVENTA;
```

### **Objeto: VENTA**

```
CREATE OR REPLACE TYPE VENTA AS OBJECT
(
    IDVENTA          NUMBER(6),
    VCLIENTE         REF CLIENTE,
    VFECHA           DATE,
    VARTICULOS       LISTAARTICULOS,
    -- Constructores
    CONSTRUCTOR FUNCTION VENTA(IDVENTA NUMBER,VCLIENTE REF CLIENTE, VFECHA DATE,
        VARTICULOS LISTAARTICULOS)
        RETURN SELF AS RESULT,
    CONSTRUCTOR FUNCTION VENTA RETURN SELF AS RESULT,

    -- Métodos Geters y Seters
    MEMBER FUNCTION GETIDVENTA RETURN NUMBER,
    MEMBER PROCEDURE SETIDVENTA(IDVENTA NUMBER),

    MEMBER FUNCTION GETVCLIENTE RETURN REF CLIENTE,
    MEMBER PROCEDURE SETVCLIENTE(VCLIENTE REF CLIENTE),

    MEMBER FUNCTION GETVFECHA RETURN DATE,
    MEMBER PROCEDURE SETVFECHA(VFECHA DATE),

    MEMBER FUNCTION GETVARTICULOS RETURN LISTAARTICULOS,
    MEMBER PROCEDURE SETVARTICULOS(VARTICULOS LISTAARTICULOS)
);
```

```
CREATE OR REPLACE TYPE BODY VENTA AS
```

```
    CONSTRUCTOR FUNCTION VENTA(IDVENTA NUMBER,VCLIENTE REF CLIENTE, VFECHA DATE,
        VARTICULOS LISTAARTICULOS)
        RETURN SELF AS RESULT AS
```

```
BEGIN
```

```
    SELF.SETIDVENTA(IDVENTA);
    SELF.SETVCLIENTE(VCLIENTE);
    SELF.SETVFECHA(VFECHA);
    SELF.SETVARTICULOS(VARTICULOS);
    RETURN;
```

```
END VENTA;
```

```
CONSTRUCTOR FUNCTION VENTA RETURN SELF AS RESULT AS
```

```
BEGIN
```

```
    SELF.SETIDVENTA(NULL);
    SELF.SETVCLIENTE(NULL);
    SELF.SETVFECHA(NULL);
```

```
    SELF.SETVARTICULOS(NULL);  
    RETURN;  
END VENTA;
```

```
MEMBER FUNCTION GETIDVENTA RETURN NUMBER AS  
BEGIN  
    RETURN SELF.IDVENTA;  
END GETIDVENTA;
```

```
MEMBER PROCEDURE SETIDVENTA(IDVENTA NUMBER) AS  
BEGIN  
    SELF.IDVENTA:=IDVENTA;  
END SETIDVENTA;
```

```
MEMBER FUNCTION GETVCLIENTE RETURN REF CLIENTE AS  
BEGIN  
    RETURN SELF.VCLIENTE;  
END GETVCLIENTE;
```

```
MEMBER PROCEDURE SETVCLIENTE(VCLIENTE REF CLIENTE) AS  
BEGIN  
    SELF.VCLIENTE:=VCLIENTE;  
END SETVCLIENTE;
```

```
MEMBER FUNCTION GETVFECHA RETURN DATE AS  
BEGIN  
    RETURN SELF.VFECHA;  
END GETVFECHA;
```

```
MEMBER PROCEDURE SETVFECHA(VFECHA DATE) AS  
BEGIN  
    SELF.VFECHA:=VFECHA;  
END SETVFECHA;
```

```
MEMBER FUNCTION GETVARTICULOS RETURN LISTAARTICULOS AS  
BEGIN  
    RETURN SELF.VARTICULOS;  
END GETVARTICULOS;
```

```
MEMBER PROCEDURE SETVARTICULOS(VARTICULOS LISTAARTICULOS) AS  
BEGIN  
    SELF.VARTICULOS:=VARTICULOS;  
END SETVARTICULOS;
```

```
END;
```

## **CREACIÓN DE LAS TABLAS**

```
CREATE TABLE ARTICULOS OF ARTICULO (PRIMARY KEY (IDARTICULO));
```

```
CREATE TABLE CLIENTES OF CLIENTE( PRIMARY KEY(DNI));
```

```
CREATE TABLE VENTAS OF VENTA ( PRIMARY KEY(IDVENTA));
```



## EJEMPLO DE INSERTAR ARTICULOS:

```
INSERT INTO ARTICULOS VALUES(1,'ARTICULO 1',100,10);
INSERT INTO ARTICULOS VALUES
    (ARTICULO(2,'ARTICULO 2','DESCRIPCION 2',200,20));

INSERT INTO ARTICULOS VALUES(ARTICULO(3,'ARTICULO 3',300));
COMMIT;
```

- En un bloque anónimo a través de los métodos:

```
DECLARE
    AR1 ARTICULO:=NEW ARTICULO(NULL,NULL,NULL,NULL);
BEGIN
    AR1.SETIDARTICULO(4);
    AR1.SETNOMBREARTICULO('ARTICULO4');
    AR1.SETPRECIOUNITARIO(400);
    AR1.SETUNIDADESEXISTENTES(40);
    INSERT INTO ARTICULOS VALUES (AR1);
    COMMIT;
END;
```

- Para visualizar los registros de la tabla ARTICULOS:

```
SELECT * FROM ARTICULOS;
```

## EJEMPLO DE INSERTAR CLIENTES:

- Directamente:

```
INSERT INTO CLIENTES VALUES('1A','NOMBRE 1','APELLIDOS 1','C/DIRECCION,1',
    'POBLACION1','EMAIL1@AAA.AA',926111111,1,'01/01/2000',1);
COMMIT;
```

- Mediante un bloque anónimo:

```
DECLARE
    COMUNES DATOSCOMUNES;
    CL1      CLIENTE;

BEGIN
    -- Es igual poner NEW q no ponerlo

    CL1:= NEW CLIENTE();
    CL1.SETDNI('3a');
    CL1.SETNOMBRE('nombre 3');
    CL1.SETAPELLIDOS('apellidos 3');
    CL1.SETDIRECCION('C/direccion,3');
    CL1.SETPOBLACION('poblacion3');
    CL1.SETEMAIL('email3AAA.AA');
    CL1.SETTELEFONO(926333333);
    CL1.SETIDCLIENTE(3);
    CL1.SETFECHAALTA('03/03/2003');
    CL1.SETTIPO(3);
    INSERT INTO CLIENTES VALUES(CL1);

    CL1:=CLIENTE('2a','nombre 2','apellidos 12','APELLIDO 22', 'C/direccion,2',
```

```

        'poblacion3', 'email3AAA.AA',926333333,2,'03/03/2003',1);
INSERT INTO CLIENTES VALUES(CL1);

CL1:=CLIENTE(DATOSCOMUNES('4a','nombre 4','apellidos 4', 'C/direccion,4',
        'poblacion4', 'email4@AAA.AA',926333333),4,'03/03/2003',1);
INSERT INTO CLIENTES VALUES(CL1);
INSERT INTO CLIENTES VALUES(CLIENTE(DATOSCOMUNES('5a','nombre 5','apellidos 5',
        'C/direccion,2', 'poblacion1', 'email5@AAA.AA',926333333),5,'03/03/2005',2));

COMUNES:=DATOSCOMUNES('6A','NOMBRE 6','APELLIDOS 6', 'C/DIRECCION,2',
        'POBLACION1', 'EMAIL6@AAA.AA',926222222);

CL1:=CLIENTE(COMUNES,4,'03/03/2003',1);

INSERT INTO CLIENTES VALUES (CL1);
COMMIT;
END;

```

### **EJEMPLO DE INSERTAR VENTAS:**

```

DECLARE
    AR1      ARTICULO;
    RC       REF CLIENTE;
    L        LISTAARTICULOS:=NEW LISTAARTICULOS(NULL,NULL,NULL);
    VT       VENTA:=NEW VENTA();
BEGIN
    -- Cogiendo el valor de un artículo almacenado en la tabla ARTICULOS
    SELECT VALUE(A) INTO AR1 FROM ARTICULOS A WHERE IDARTICULO=1;
    L(1):=LINEAVENTA(10,AR1);

    -- Pasándoselo a través del objeto ARTICULO
    L(2):=LINEAVENTA(20,ARTICULO(2,'Articulo2',250,20));

    L(3):=LINEAVENTA(30,ARTICULO(5,'Articulo5','descrip',300,30));

    -- Cogemos la referencia del cliente cuyo DNI 3 de la tabla CLIENTES

    SELECT REF(C) INTO RC FROM CLIENTES C WHERE C.DNI='2A';

    VT.SETIDVENTA(1);
    VT.SETVCLIENTE(RC);
    VT.SETVFECHA('01/01/2015');
    VT.SETVARTICULOS(L);
    INSERT INTO VENTAS VALUES(VT);
    COMMIT;
END;

```

### **EJEMPLO DE LEER VENTAS:**

```

DECLARE
    VT       VENTA;
    CL1      CLIENTE;

BEGIN
    SELECT VALUE(V) INTO VT  FROM VENTAS V WHERE V.IDVENTA=1;
    SELECT Deref(VT.VCLIENTE) INTO CL1 FROM DUAL;
    DBMS_OUTPUT.PUT_LINE('CLIENTE: '||CL1.GETNOMBRE()||'  FECHA VENTA: '
        || VT.GETVFECHA());

```

```

        FOR I IN 1 .. VT.VARTICULOS.COUNT LOOP
            DBMS_OUTPUT.PUT_LINE(
                'ARTICULO :'||VT.VARTICULOS(I).LOSARTICULOS.GETNOMBREARTICULO()
                || ' UNIDADES:' ||VT.VARTICULOS(I).UNIDADES
                || ' PRECIO UNITARIO : '
                ||VT.VARTICULOS(I).LOSARTICULOS.GETPRECIOUNITARIO() );
        END LOOP;
    END;

```

### **EJEMPLO DE LEER VENTAS CON UN CURSOR:**

```

DECLARE
    CL1    CLIENTE;
    V      VENTA;
BEGIN
    FOR VT IN (SELECT * FROM VENTAS V WHERE V.VCLIENTE.DNI='2A') LOOP
        SELECT Deref(VT.VCLIENTE) INTO CL1 FROM DUAL;
        DBMS_OUTPUT.PUT_LINE('CLIENTE: ' ||CL1.GETNOMBRE() || ' FECHA VENTA: '
            || VT.VFECHA);
        FOR I IN 1 .. VT.VARTICULOS.COUNT LOOP
            DBMS_OUTPUT.PUT_LINE('ARTICULO : '
                ||VT.VARTICULOS(I).LOSARTICULOS.GETNOMBREARTICULO()
                || ' UNIDADES:' ||VT.VARTICULOS(I).UNIDADES
                || ' PRECIO UNITARIO : '
                ||VT.VARTICULOS(I).LOSARTICULOS.GETPRECIOUNITARIO() );
        END LOOP;
    END LOOP;
END;

```