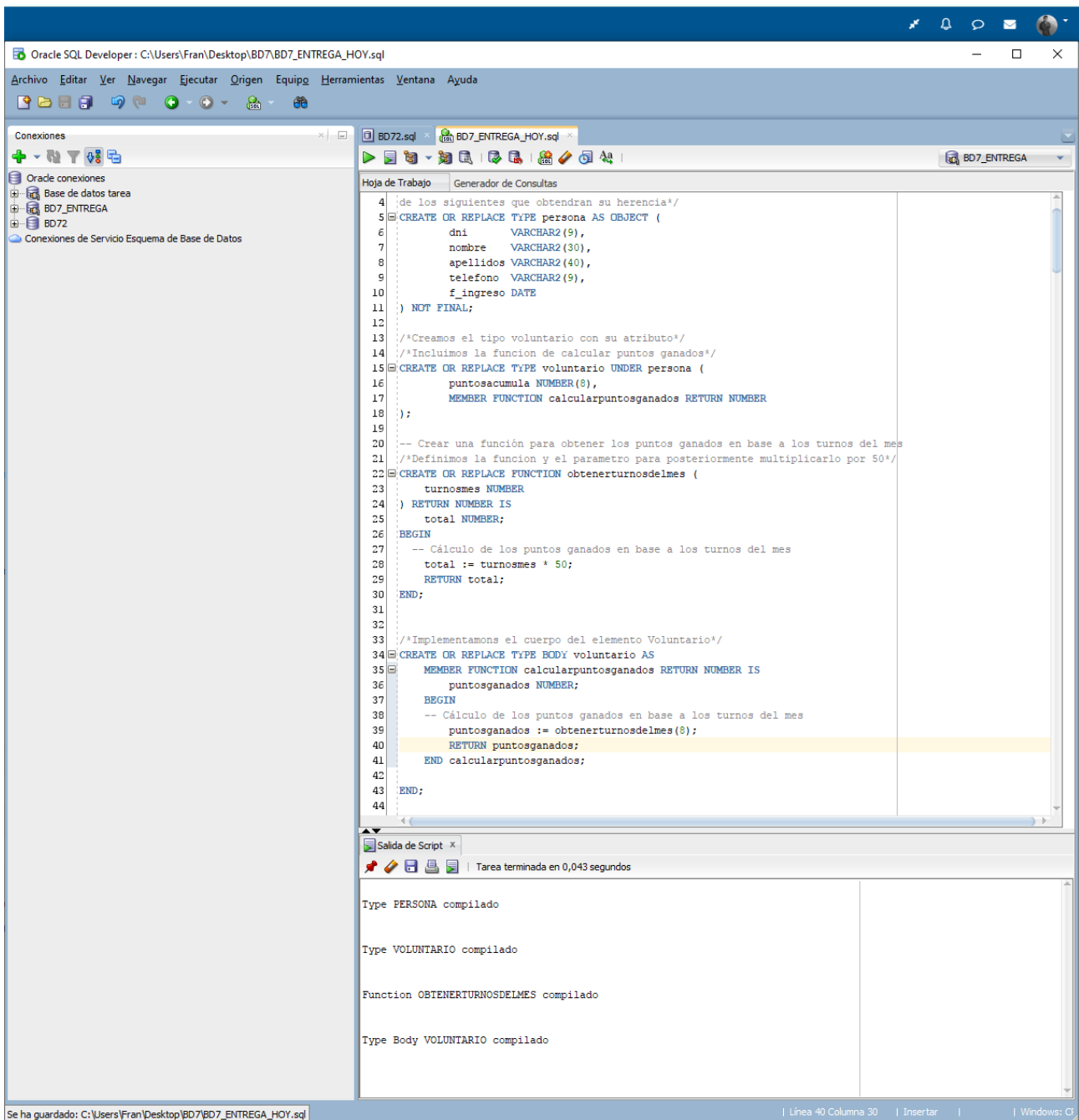


BD:ACTIVIDAD 7

Para documentar esta actividad voy a intentar optar por lo práctico, ya que está detallado ampliamente en el código, y en algunos subapartados solo hay que copiar la estructura de los atributos, voy a intentar mostrar la compilación de todos los subapartados por actividad, y haré hincapié en los que considere que merecen ser desarrollados.



The screenshot displays the Oracle SQL Developer interface. The main window shows a SQL script with the following content:

```
4  de los siguientes que obtendran su herencia*/
5  CREATE OR REPLACE TYPE persona AS OBJECT (
6      dni          VARCHAR2(9),
7      nombre       VARCHAR2(30),
8      apellidos    VARCHAR2(40),
9      telefono     VARCHAR2(9),
10     f_ingreso     DATE
11 ) NOT FINAL;
12
13 /*Creamos el tipo voluntario con su atributo*/
14 /*Incluimos la funcion de calcular puntos ganados*/
15 CREATE OR REPLACE TYPE voluntario UNDER persona (
16     puntosacumula NUMBER(8),
17     MEMBER FUNCTION calcularpuntosganados RETURN NUMBER
18 );
19
20 -- Crear una función para obtener los puntos ganados en base a los turnos del mes
21 /*Definimos la funcion y el parametro para posteriormente multiplicarlo por 50*/
22 CREATE OR REPLACE FUNCTION obtenerturnosdelmes (
23     turnosmes NUMBER
24 ) RETURN NUMBER IS
25     total NUMBER;
26 BEGIN
27     -- Cálculo de los puntos ganados en base a los turnos del mes
28     total := turnosmes * 50;
29     RETURN total;
30 END;
31
32
33 /*Implementamons el cuerpo del elemento Voluntario*/
34 CREATE OR REPLACE TYPE BODY voluntario AS
35     MEMBER FUNCTION calcularpuntosganados RETURN NUMBER IS
36     puntosganados NUMBER;
37 BEGIN
38     -- Cálculo de los puntos ganados en base a los turnos del mes
39     puntosganados := obtenerturnosdelmes(8);
40     RETURN puntosganados;
41 END calcularpuntosganados;
42
43 END;
44
```

Below the script, the 'Salida de Script' (Script Output) window shows the compilation results:

```
Tarea terminada en 0,043 segundos

Type PERSONA compilado

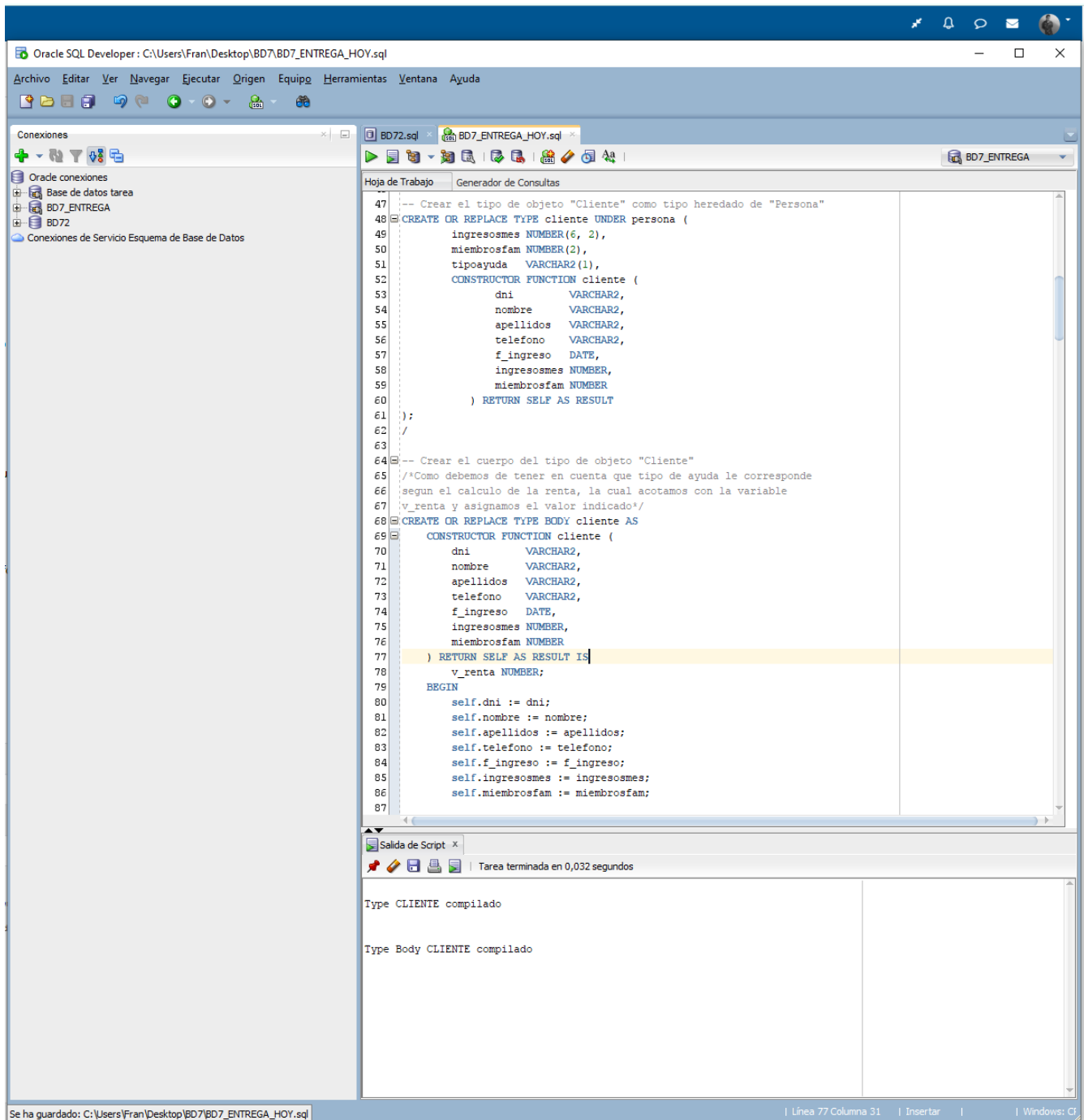
Type VOLUNTARIO compilado

Function OBTENERTURNOSDELMES compilado

Type Body VOLUNTARIO compilado
```

The status bar at the bottom indicates the file is saved at 'C:\Users\Fran\Desktop\BD7\BD7_ENTREGA_HOY.sql' and the cursor is at 'Linea 40 Columna 30'.

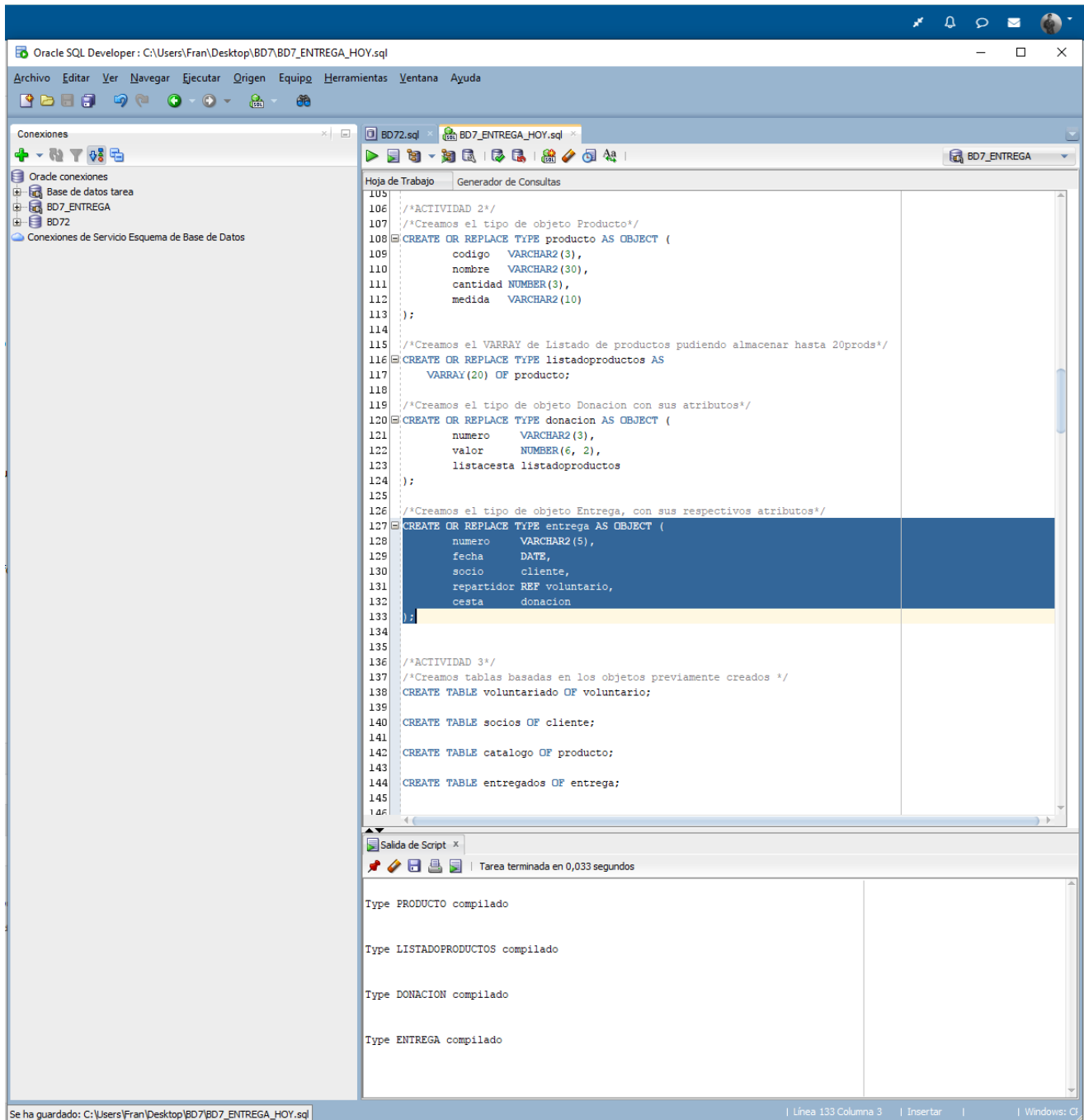
Como podemos observar, hemos compilado 3 tipos y una función. Lo más relevante de esta secuencia es incluir el parámetro “NOT FINAL” para permitir que el tipo voluntario obtenga la herencia de persona. Así mismo hemos definido la función que permite calcular los turnos del mes y los puntos ganados. También hemos incluido la función que hace que interactúen estos parámetros con el tipo y cuerpo de voluntario.



Aquí hemos creado el tipo cliente y hemos definido su cuerpo, en vista de que tenemos que calcular un parámetro variable que asigna un valor según un cálculo entre el ingreso de mes y los miembros de la familia. Como observamos se han compilado correctamente.

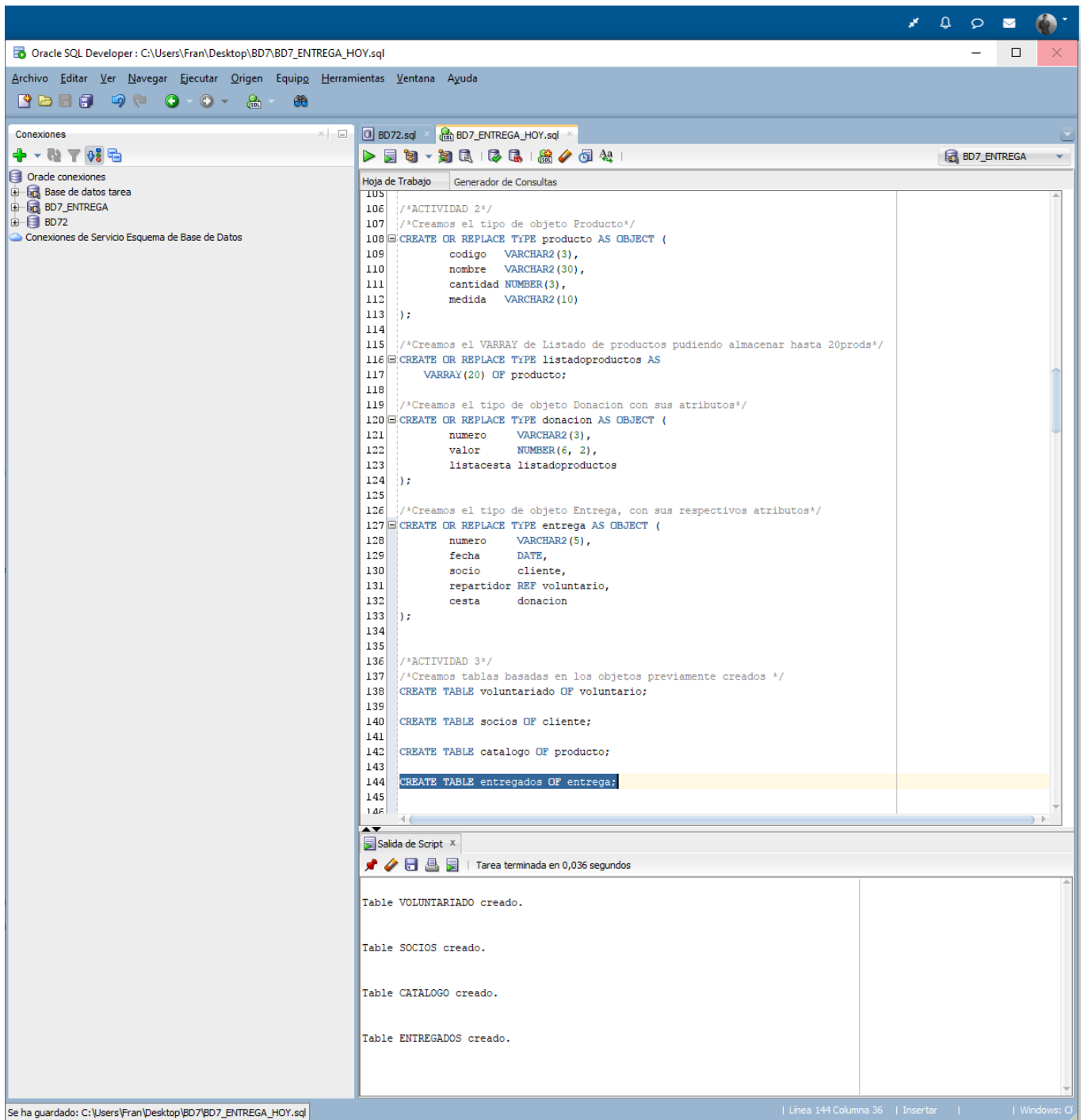
ACTIVIDADES 2 Y 3:

Lo que contienen estas actividades se trata de la definición de tipos y un VARRAY para posteriormente insertarlo en las tablas correspondientes:



Aquí podemos observar como hemos compilado los 4 tipos que se piden en la actividad 2 con sus correspondientes atributos. En el caso de "listadoproductos" se trata de un VARRAY que va a contener los productos que definiremos posteriormente.

En la definición de los atributos solo nos limitamos a trasladar lo que se nos indica en la actividad.



En la actividad 3 simplemente creamos la tabla con el nombre que nos han indicado he insertamos los objetos que hemos creado en el orden deseado. Se puede ver en el output que la compilación ha sido un éxito.

ACTIVIDAD 4:

Esta actividad es la que más elementos contempla, y por lo tanto la más larga. Los pasos a seguir son relativamente más complejos, pero los desgaremos poco a poco. Hemos usado las estructuras de bloques PL/SQL, vamos con la primera.

```

145
146
147 /*ACTIVIDAD 4*/
148 /*Bloques PL/SQL de voluntario, cliente y producto */
149 DECLARE
150     v_voluntario1 voluntario;
151     v_voluntario2 voluntario;
152     v_puntos      NUMBER;
153 BEGIN
154     -- Creamos primera instancia de Voluntario
155     v_voluntario1 := voluntario('100000000A', 'Jaime', 'Sánchez Terol', '111222333', TO_DATE('15/03/2018',
156         'DD/MM/YYYY'), 200);
157
158     -- Creamos segunda instancia de Voluntario
159     v_voluntario2 := voluntario('200000000B', 'Carmen', 'Mira González', '888999000', TO_DATE('8/02/2020',
160         'DD/MM/YYYY'), 100);
161
162     -- Insertamos las instancias en la tabla VOLUNTARIADO
163     INSERT INTO voluntariado VALUES v_voluntario1;
164
165     INSERT INTO voluntariado VALUES v_voluntario2;
166
167     -- Usamos la funcion creada para calcular los puntos ganados por el voluntario con DNI 100000000A
168     v_puntos := v_voluntario1.calcularpuntosganados();
169
170     -- Actualizamos los puntos acumulados del voluntario con DNI 100000000A en la tabla VOLUNTARIADO
171     UPDATE voluntariado
172     SET
173         puntosacumula = puntosacumula + v_puntos
174     WHERE
175         dni = '100000000A';
176
177     dbms_output.put_line('Puntos ganados por el voluntario con DNI 100000000A: ' || v_puntos);
178     COMMIT;
179 END;
180
181
182
183 /*BLOQUE CLIENTE*/
184
185 /*Declaramos 3 instancias de cliente*/

```

Salida de Script x

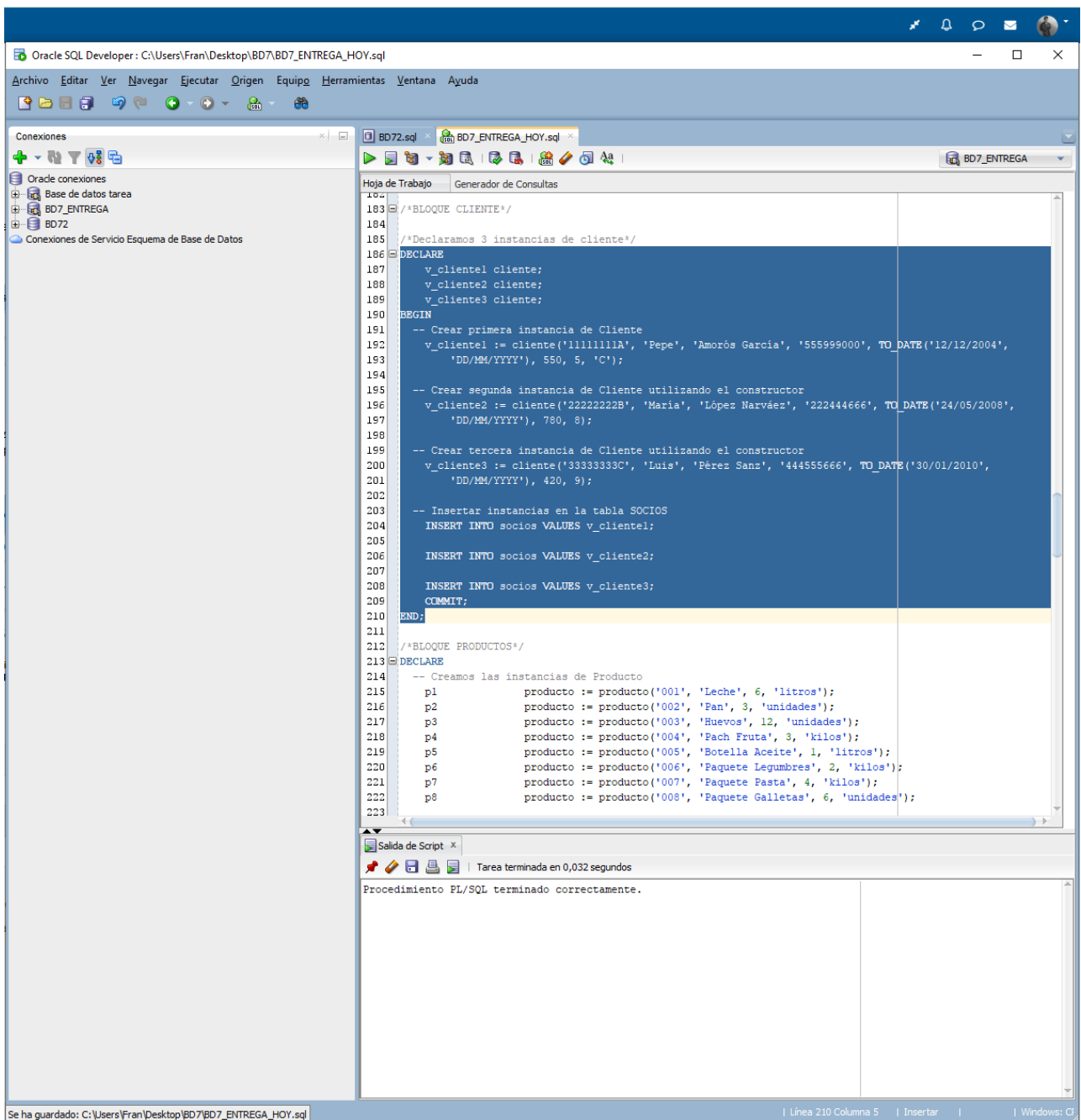
Tarea terminada en 0,043 segundos

Procedimiento PL/SQL terminado correctamente.

Se ha guardado: C:\Users\Fran\Desktop\BD7\BD7_ENTREGA_HOY.sql | Línea 180 Columna 1 | Insertar | Windows: O

Primero creamos el bloque en el que vamos a definir dos instancias de voluntario. Después de definir las instancias, las insertamos en la tabla de voluntariado.

Ejecutamos la función que habíamos definido previamente para calcular los puntos ganados por el voluntario 1 y actualizamos la información en su tabla correspondiente.



En este siguiente apartado, hemos compilado el bloque cliente.

Hemos creado 3 instancias de cliente, rellenando el constructor con los atributos que habíamos definido previamente, hemos hecho lo mismo que en el apartado anterior, al introducir las nuevas instancias en la tabla "socios"

Este siguiente bloque es el más largo, lo he enfocado de una manera práctica en la que todas las variables e instancias que están relacionadas estén reunidas. Por lo tanto, es el que más deberemos de desgranar. Primero la creación de las instancias de producto y la definición de las variables.

```

209 COMMIT;
210 END;
211
212 /*BLOQUE PRODUCTOS*/
213 DECLARE
214 -- Creamos las instancias de Producto
215 p1          producto := producto('001', 'Leche', 6, 'litros');
216 p2          producto := producto('002', 'Pan', 3, 'unidades');
217 p3          producto := producto('003', 'Huevos', 12, 'unidades');
218 p4          producto := producto('004', 'Pach Fruta', 3, 'kilos');
219 p5          producto := producto('005', 'Botella Aceite', 1, 'litros');
220 p6          producto := producto('006', 'Paquete Legumbres', 2, 'kilos');
221 p7          producto := producto('007', 'Paquete Pasta', 4, 'kilos');
222 p8          producto := producto('008', 'Paquete Galletas', 6, 'unidades');
223
224 -- Creamos las instancias de ListadoProductos utilizando los productos previamente creados
225 listaproductos1 listadoproductos := listadoproductos(p1, p3, p5, p7);
226 listaproductos2 listadoproductos := listadoproductos(p1, p2, p4, p8);
227 listaproductos3 listadoproductos := listadoproductos(p3, p5, p6);
228
229 -- Variables correspondientes a donaciones
230
231 donacion1     donacion;
232 donacion2     donacion;
233 donacion3     donacion;
234
235 -- Variables para guardar las referencias de voluntarios
236 refvoluntario1 REF voluntario;
237 refvoluntario2 REF voluntario;
238
239 -- Variables para guardar las instancias de clientes
240 cliente1      cliente;
241 cliente2      cliente;
242
243 -- Variables para guardar las instancias de Entrega
244 entrega1      entrega;
245 entrega2      entrega;
246 entrega3      entrega;
247 entrega4      entrega;
248 BEGIN
249

```

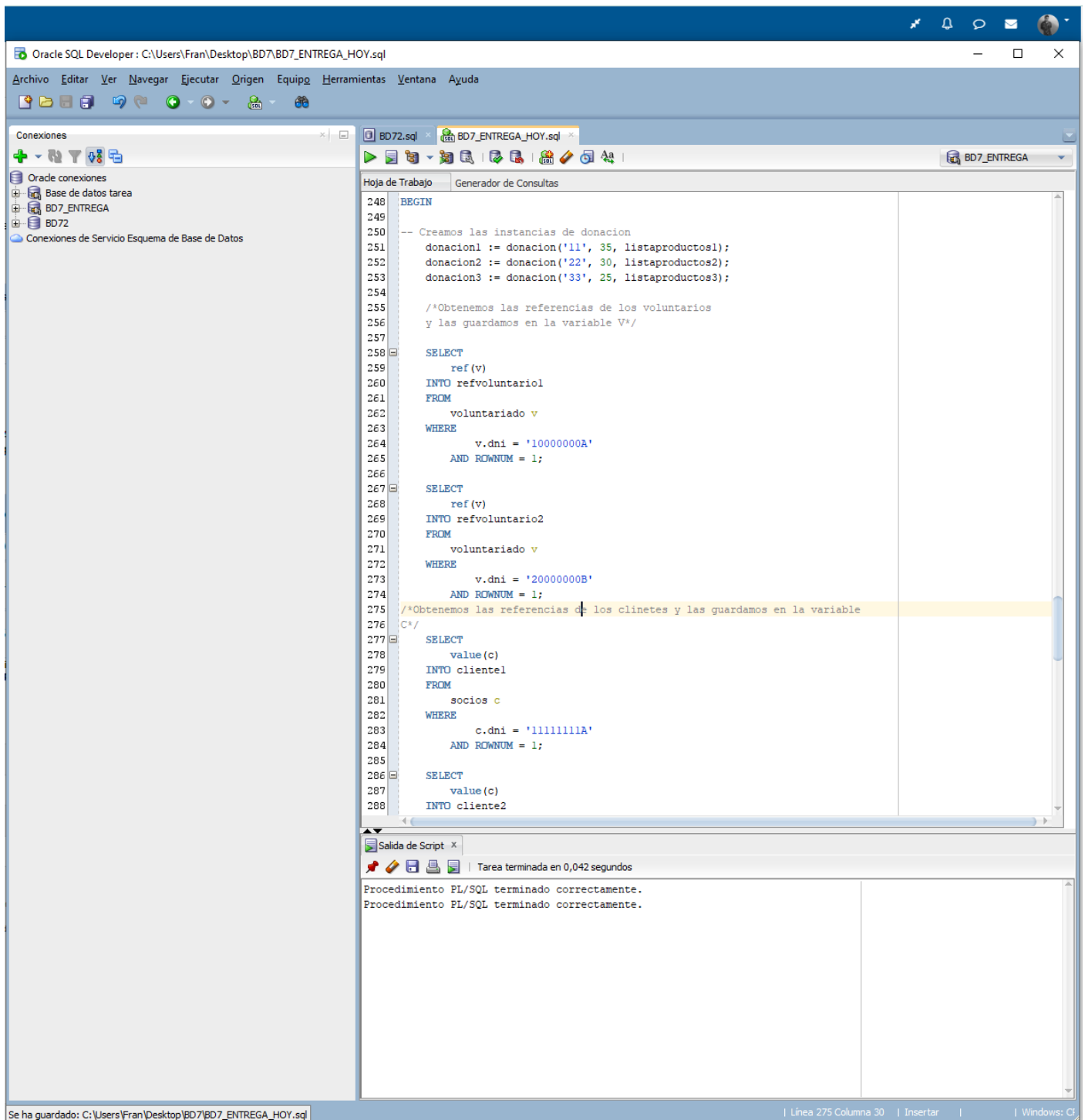
Salida de Script x

Tarea terminada en 0,042 segundos

Procedimiento PL/SQL terminado correctamente.
Procedimiento PL/SQL terminado correctamente.

Se ha guardado: C:\Users\Fran\Desktop\BD7\BD7_ENTREGA_HOY.sql | Línea 275 Columna 30 | Insertar | Windows: C

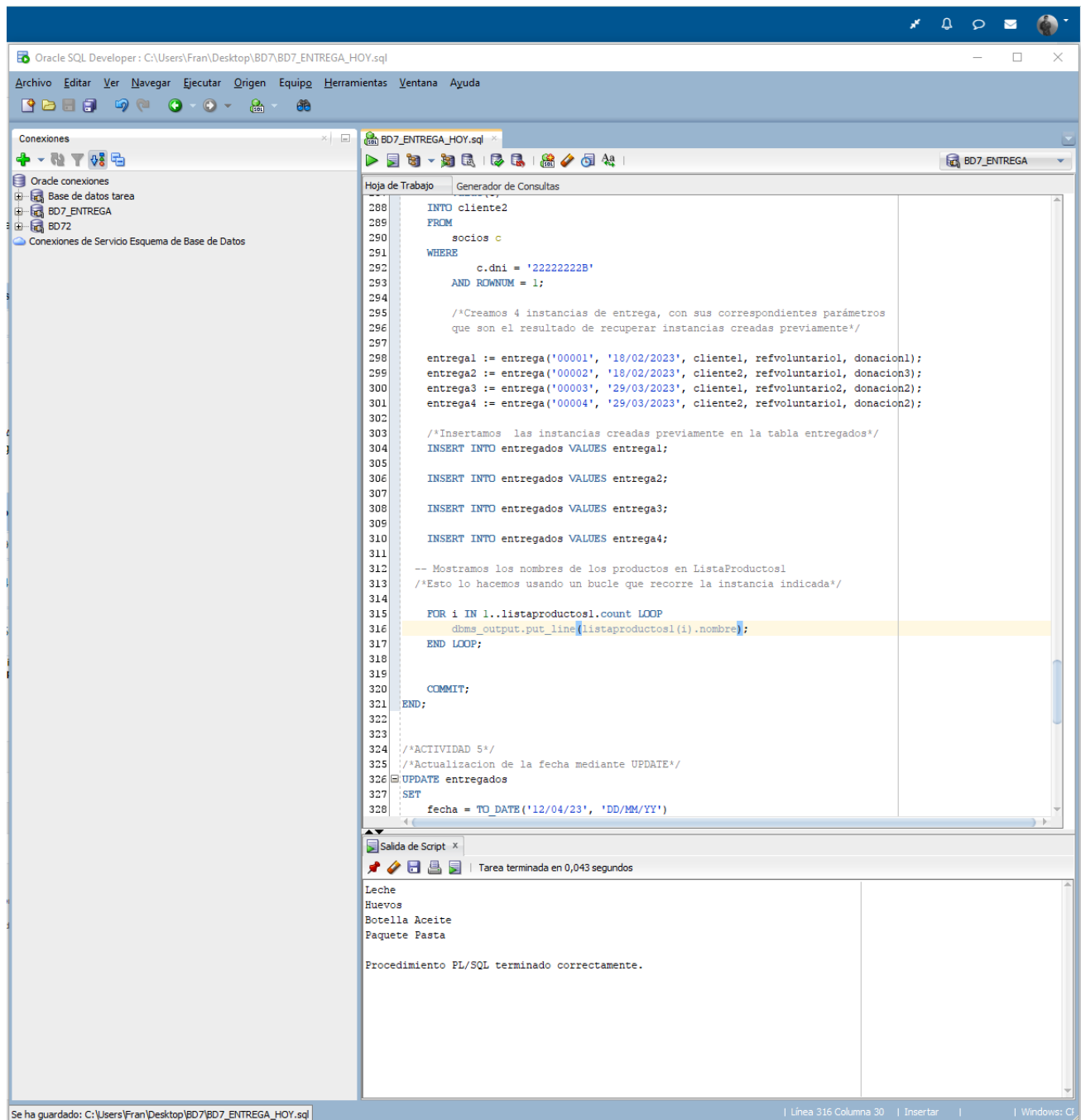
Creamos las instancias de producto, 8 en total. Le asignamos la variable pX para luego usarla en la instancia de listaproductosX. Posteriormente, como vamos a usarlas en este bloque, definimos variables para donación, voluntario, cliente y entrega. En el output se ve como el procedimiento se ha compilado correctamente, al ejecutarlo completo, por supuesto.



Creamos las instancias de donación, tal y como nos indican usando “listaproduktosX” para recuperar el contenido de la variable.

Creamos variables para guardar la información de los voluntarios y los clientes que nos indican.

Como se puede ver las obtenemos de sus tablas correspondientes y recuperamos el valor “dni” de ambos, el cual especificamos. También limitamos con ROWNUM el numero de columnas que queremos almacenar.



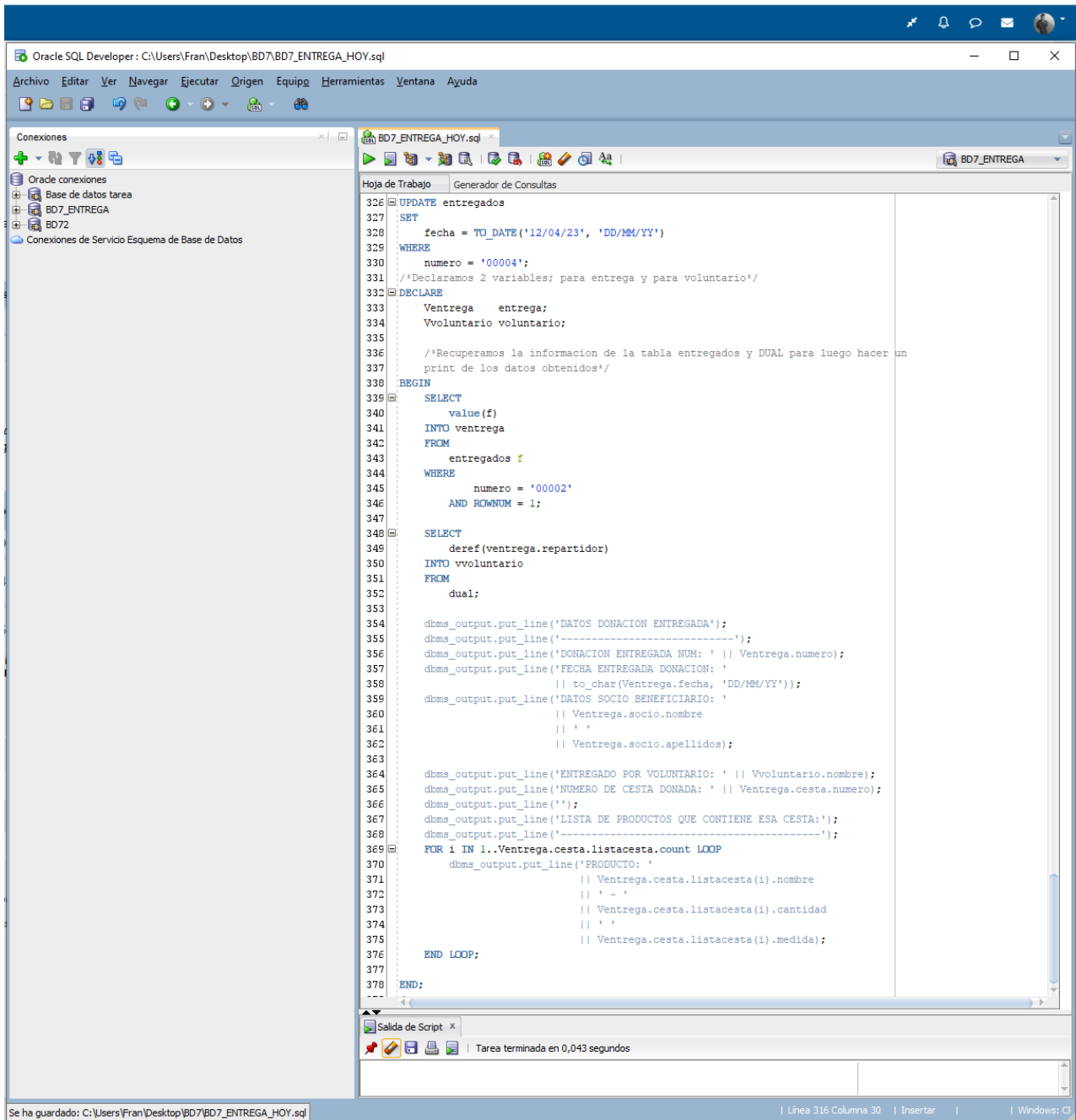
Para finalizar, creamos 4 instancias de entrega, pero en las que deberemos de recuperar los parámetros que hemos creado previamente para cliente, voluntario y donación con la distinta información almacenado en cada una de sus variables. Una vez definidas, las insertamos en la tabla entregados.

Para terminar, creamos un pequeño bucle que recorra la información contenida en "listaproductos1".

Como podemos comprobar en el la salida al compilar nos muestra los 4 productos que contiene.

ACTIVIDAD 5:

Trabajaremos con una actualización y con un print de variables que recoge todo lo definido previamente.



En el primer apartado, actualizamos la fecha de la entrega numero 00004 en la tabla entregados. Una operación sencilla que ya contemplábamos en las unidades 4 y 5.

En el apartado siguiente deberemos de declarar dos variables de las que vamos a recuperar información posteriormente. De la tabla entregados almacenamos la información de la entrega 00002, y del objeto voluntario, usando una tabla ficticia (DUAL) almacenaremos la información correspondiente a qué entrega realiza un repartidor en concreto.

Posteriormente imprimimos por pantalla los datos que nos solicitan, haciendo uso de la variable creada al principio y los atributos heredados.

Por último, recorreremos con un bucle el contenido de la cesta que corresponde al cliente.

A continuación, el resultado de compilar este bloque:

The screenshot shows the Oracle SQL Developer interface. The main window displays a PL/SQL script named 'BD7_ENTREGA_HOY.sql'. The script is a PL/SQL block that updates a record in the 'BD7_ENTREGA' table and then prints the details of the donation and the contents of the basket. The script is as follows:

```

350 BEGIN
351   FROM
352   dual;
353
354   dbms_output.put_line('DATOS DONACION ENTREGADA');
355   dbms_output.put_line('-----');
356   dbms_output.put_line('DONACION ENTREGADA NUM: ' || Ventrega.numero);
357   dbms_output.put_line('FECHA ENTREGADA DONACION: '
358   || to_char(Ventrega.fecha, 'DD/MM/YY'));
359   dbms_output.put_line('DATOS SOCIO BENEFICIARIO: '
360   || Ventrega.socio.nombre
361   || ' '
362   || Ventrega.socio.apellidos);
363
364   dbms_output.put_line('ENTREGADO POR VOLUNTARIO: ' || Vvoluntario.nombre);
365   dbms_output.put_line('NUMERO DE CESTA DONADA: ' || Ventrega.cesta.numero);
366   dbms_output.put_line('');
367   dbms_output.put_line('LISTA DE PRODUCTOS QUE CONTIENE ESA CESTA:');
368   dbms_output.put_line('-----');
369   FOR i IN 1..Ventrega.cesta.listacesta.count LOOP
370     dbms_output.put_line('PRODUCTO: '
371     || Ventrega.cesta.listacesta(i).nombre
372     || ' - '
373     || Ventrega.cesta.listacesta(i).cantidad
374     || ' '
375     || Ventrega.cesta.listacesta(i).medida);
376   END LOOP;
377
378 END;
379 /

```

The output window shows the results of the script execution:

```

2 filas actualizadas.

DATOS DONACION ENTREGADA
-----
DONACION ENTREGADA NUM: 00002
FECHA ENTREGADA DONACION: 18/02/23
DATOS SOCIO BENEFICIARIO: Maria López Narváez
ENTREGADO POR VOLUNTARIO: Jaime
NUMERO DE CESTA DONADA: 33

LISTA DE PRODUCTOS QUE CONTIENE ESA CESTA:
-----
PRODUCTO: Huevos - 12 unidades
PRODUCTO: Botella Aceite - 1 litros
PRODUCTO: Paquete Legumbres - 2 kilos

Procedimiento PL/SQL terminado correctamente.

```

Aquí se puede ver como se actualiza la fecha y posteriormente se hace el print que hemos creado. Hemos usado el mismo número de entrega que se proporciona en la actividad para comprobar que los valores son correctos.

En efecto es así, salvo por el valor de la caja de huevos, que por lo visto en el ejemplo facilitado al crear su instancia se creo como: pack huevos- 6 unidades.

Pero después de comprobar, la llamada a la variable es correcta porque la instancia que corresponde con el producto huevo contiene la información mostrada. –
(Código más adelante)

/*ACTIVIDAD 1*/

/*Creamos el objeto "persona" con los atributos indicados
tambien indicamos NOT FINAL para que pueda ser elemento padre
de los siguientes que obtendran su herencia*/

```
CREATE OR REPLACE TYPE persona AS OBJECT (  
    dni          VARCHAR2(9),  
    nombre       VARCHAR2(30),  
    apellidos    VARCHAR2(40),  
    telefono     VARCHAR2(9),  
    f_ingreso    DATE  
)  
NOT FINAL;
```

/*Creamos el tipo voluntario con su atributo*/
/*Incluimos la funcion de calcular puntos ganados*/

```
CREATE OR REPLACE TYPE voluntario UNDER persona (  
    puntosacumula NUMBER(8),  
    MEMBER FUNCTION calcularpuntosganados RETURN NUMBER  
);
```

-- Crear una función para obtener los puntos ganados en base a los
turnos del mes

/*Definimos la funcion y el parametro para posteriormente
multiplicarlo por 50*/

```
CREATE OR REPLACE FUNCTION obtenerturnosdelmes (  
    turnosmes NUMBER  
)  
RETURN NUMBER IS  
    total NUMBER;  
BEGIN  
    -- Cálculo de los puntos ganados en base a los turnos del mes  
    total := turnosmes * 50;  
    RETURN total;  
END;
```

/*Implementamons el cuerpo del elemento Voluntario*/

```
CREATE OR REPLACE TYPE BODY voluntario AS  
    MEMBER FUNCTION calcularpuntosganados RETURN NUMBER IS  
        puntosganados NUMBER;  
    BEGIN  
        -- Cálculo de los puntos ganados en base a los turnos del mes  
        puntosganados := obtenerturnosdelmes(8);  
        RETURN puntosganados;  
    END calcularpuntosganados;  
END;
```

```
-- Crear el tipo de objeto "Cliente" como tipo heredado de "Persona"
CREATE OR REPLACE TYPE cliente UNDER persona (
    ingresosmes NUMBER(6, 2),
    miembrosfam NUMBER(2),
    tipoayuda    VARCHAR2(1),
    CONSTRUCTOR FUNCTION cliente (
        dni          VARCHAR2,
        nombre       VARCHAR2,
        apellidos    VARCHAR2,
        telefono     VARCHAR2,
        f_ingreso    DATE,
        ingresosmes  NUMBER,
        miembrosfam  NUMBER
    ) RETURN SELF AS RESULT
);
/

-- Crear el cuerpo del tipo de objeto "Cliente"
/*Como debemos de tener en cuenta que tipo de ayuda le corresponde
segun el calculo de la renta, la cual acotamos con la variable
v_renta y asignamos el valor indicado*/
CREATE OR REPLACE TYPE BODY cliente AS
    CONSTRUCTOR FUNCTION cliente (
        dni          VARCHAR2,
        nombre       VARCHAR2,
        apellidos    VARCHAR2,
        telefono     VARCHAR2,
        f_ingreso    DATE,
        ingresosmes  NUMBER,
        miembrosfam  NUMBER
    ) RETURN SELF AS RESULT IS
        v_renta NUMBER;
    BEGIN
        self.dni := dni;
        self.nombre := nombre;
        self.apellidos := apellidos;
        self.telefono := telefono;
        self.f_ingreso := f_ingreso;
        self.ingresosmes := ingresosmes;
        self.miembrosfam := miembrosfam;

        -- Calcula la renta
        v_renta := ingresosmes / miembrosfam;
```

```

        -- Asignamos el valor del atributo tipoAyuda según la renta
calculada
        IF v_renta <= 50 THEN
            self.tipoayuda := 'A';
        ELSIF v_renta <= 100 THEN
            self.tipoayuda := 'B';
        ELSE
            self.tipoayuda := 'C';
        END IF;

        RETURN;
    END cliente;

END;

/*ACTIVIDAD 2*/
/*Creamos el tipo de objeto Producto*/
CREATE OR REPLACE TYPE producto AS OBJECT (
    codigo    VARCHAR2(3),
    nombre    VARCHAR2(30),
    cantidad  NUMBER(3),
    medida    VARCHAR2(10)
);

/*Creamos el VARRAY de Listado de productos pudiendo almacenar hasta
20prods*/
CREATE OR REPLACE TYPE listadoproductos AS
    VARRAY(20) OF producto;

/*Creamos el tipo de objeto Donacion con sus atributos*/
CREATE OR REPLACE TYPE donacion AS OBJECT (
    numero    VARCHAR2(3),
    valor     NUMBER(6, 2),
    listacesta listadoproductos
);

/*Creamos el tipo de objeto Entrega, con sus respectivos atributos*/
CREATE OR REPLACE TYPE entrega AS OBJECT (
    numero    VARCHAR2(5),
    fecha     DATE,
    socio     cliente,
    repartidor REF voluntario,
    cesta     donacion
);

/*ACTIVIDAD 3*/
/*Creamos tablas basadas en los objetos previamente creados */
CREATE TABLE voluntariado OF voluntario;

CREATE TABLE socios OF cliente;

CREATE TABLE catalogo OF producto;

CREATE TABLE entregados OF entrega;

```

```
/*ACTIVIDAD 4*/
/*Bloques PL/SQL de voluntario, cliente y producto */
DECLARE
    v_voluntario1 voluntario;
    v_voluntario2 voluntario;
    v_puntos      NUMBER;
BEGIN
    -- Creamos primera instancia de Voluntario
    v_voluntario1 := voluntario('10000000A', 'Jaime', 'Sánchez Terol',
    '111222333', TO_DATE('15/03/2018',
    'DD/MM/YYYY'), 200);

    -- Creamos segunda instancia de Voluntario
    v_voluntario2 := voluntario('20000000B', 'Carmen', 'Mira
    González', '888999000', TO_DATE('8/02/2020',
    'DD/MM/YYYY'), 100);

    -- Insertamos las instancias en la tabla VOLUNTARIADO
    INSERT INTO voluntariado VALUES v_voluntario1;

    INSERT INTO voluntariado VALUES v_voluntario2;

    -- Usamos la funcion creada para calcular los puntos ganados por el
    voluntario con DNI 10000000A
    v_puntos := v_voluntario1.calcularpuntosganados();

    -- Actualizamos los puntos acumulados del voluntario con DNI
    10000000A en la tabla VOLUNTARIADO
    UPDATE voluntariado
    SET
        puntosacumula = puntosacumula + v_puntos
    WHERE
        dni = '10000000A';

    dbms_output.put_line('Puntos ganados por el voluntario con DNI
    10000000A: ' || v_puntos);
    COMMIT;
END;
```



```
/*BLOQUE CLIENTE*/

/*Declaramos 3 instancias de cliente*/
DECLARE
    v_cliente1 cliente;
    v_cliente2 cliente;
    v_cliente3 cliente;
BEGIN
    -- Crear primera instancia de Cliente
    v_cliente1 := cliente('11111111A', 'Pepe', 'Amorós García',
        '555999000', TO_DATE('12/12/2004',
            'DD/MM/YYYY'), 550, 5, 'C');

    -- Crear segunda instancia de Cliente utilizando el constructor
    v_cliente2 := cliente('22222222B', 'María', 'López Narváez',
        '222444666', TO_DATE('24/05/2008',
            'DD/MM/YYYY'), 780, 8);

    -- Crear tercera instancia de Cliente utilizando el constructor
    v_cliente3 := cliente('33333333C', 'Luis', 'Pérez Sanz',
        '444555666', TO_DATE('30/01/2010',
            'DD/MM/YYYY'), 420, 9);

    -- Insertar instancias en la tabla SOCIOS
    INSERT INTO socios VALUES v_cliente1;

    INSERT INTO socios VALUES v_cliente2;

    INSERT INTO socios VALUES v_cliente3;
    COMMIT;
END;

/*BLOQUE PRODUCTOS*/
DECLARE
    -- Creamos las instancias de Producto
    p1          producto := producto('001', 'Leche', 6, 'litros');
    p2          producto := producto('002', 'Pan', 3, 'unidades');
    p3          producto := producto('003', 'Huevos', 12,
'unidades');
    p4          producto := producto('004', 'Pach Fruta', 3,
'kilos');
    p5          producto := producto('005', 'Botella Aceite', 1,
'litros');
    p6          producto := producto('006', 'Paquete Legumbres',
2, 'kilos');
    p7          producto := producto('007', 'Paquete Pasta', 4,
'kilos');
    p8          producto := producto('008', 'Paquete Galletas', 6,
'unidades');

    -- Creamos las instancias de ListadoProductos utilizando los
productos previamente creados
    listaproductos1 listadoproductos := listadoproductos(p1, p3, p5,
p7);
    listaproductos2 listadoproductos := listadoproductos(p1, p2, p4,
p8);
    listaproductos3 listadoproductos := listadoproductos(p3, p5, p6);
```

```
-- Variables correspondientes a donaciones
donacion1      donacion;
donacion2      donacion;
donacion3      donacion;

-- Variables para guardar las referencias de voluntarios
refvoluntario1 REF voluntario;
refvoluntario2 REF voluntario;

-- Variables para guardar las instancias de clientes
cliente1       cliente;
cliente2       cliente;

-- Variables para guardar las instancias de Entrega
entrega1       entrega;
entrega2       entrega;
entrega3       entrega;
entrega4       entrega;
BEGIN

-- Creamos las instancias de donacion
donacion1 := donacion('11', 35, listaproductos1);
donacion2 := donacion('22', 30, listaproductos2);
donacion3 := donacion('33', 25, listaproductos3);

/*Obtenemos las referencias de los voluntarios
y las guardamos en la variable V*/

SELECT
    ref(v)
INTO refvoluntario1
FROM
    voluntariado v
WHERE
    v.dni = '10000000A'
    AND ROWNUM = 1;

SELECT
    ref(v)
INTO refvoluntario2
FROM
    voluntariado v
WHERE
    v.dni = '20000000B'
    AND ROWNUM = 1;
```

```
/*Obtenemos las referencias de los clinetes y las guardamos en la
variable
C*/
SELECT
    value(c)
INTO cliente1
FROM
    socios c
WHERE
    c.dni = '11111111A'
    AND ROWNUM = 1;

SELECT
    value(c)
INTO cliente2
FROM
    socios c
WHERE
    c.dni = '22222222B'
    AND ROWNUM = 1;

/*Creamos 4 instancias de entrega, con sus correspondientes
parámetros
que son el resultado de recuperar instancias creadas
previamente*/

entrega1 := entrega('00001', '18/02/2023', cliente1,
refvoluntario1, donacion1);
entrega2 := entrega('00002', '18/02/2023', cliente2,
refvoluntario1, donacion3);
entrega3 := entrega('00003', '29/03/2023', cliente1,
refvoluntario2, donacion2);
entrega4 := entrega('00004', '29/03/2023', cliente2,
refvoluntario1, donacion2);

/*Insertamos las instancias creadas previamente en la tabla
entregados*/
INSERT INTO entregados VALUES entrega1;

INSERT INTO entregados VALUES entrega2;

INSERT INTO entregados VALUES entrega3;

INSERT INTO entregados VALUES entrega4;

-- Mostramos los nombres de los productos en ListaProductos1
/*Esto lo hacemos usando un bucle que recorre la instancia
indicada*/

FOR i IN 1..listap productos1.count LOOP
    dbms_output.put_line(listap productos1(i).nombre);
END LOOP;

COMMIT;

END;
```

```

/*ACTIVIDAD 5*/
/*Actualizacion de la fecha mediante UPDATE*/
UPDATE entregados
SET
    fecha = TO_DATE('12/04/23', 'DD/MM/YY')
WHERE
    numero = '00004';
/*Declaramos 2 variables; para entrega y para voluntario*/
DECLARE
    Ventrega    entrega;
    Vvoluntario voluntario;

    /*Recuperamos la informacion de la tabla entregados y DUAL para
    luego hacer un
    print de los datos obtenidos*/
BEGIN
    SELECT
        value(f)
    INTO ventrega
    FROM
        entregados f
    WHERE
        numero = '00002'
        AND ROWNUM = 1;

    SELECT
        deref(ventrega.repartidor)
    INTO vvoluntario
    FROM
        dual;

    dbms_output.put_line('DATOS DONACION ENTREGADA');
    dbms_output.put_line('-----');
    dbms_output.put_line('DONACION ENTREGADA NUM: ' ||
Ventrega.numero);
    dbms_output.put_line('FECHA ENTREGADA DONACION: '
        || to_char(Ventrega.fecha, 'DD/MM/YY'));
    dbms_output.put_line('DATOS SOCIO BENEFICIARIO: '
        || Ventrega.socio.nombre
        || ' '
        || Ventrega.socio.apellidos);

    dbms_output.put_line('ENTREGADO POR VOLUNTARIO: ' ||
Vvoluntario.nombre);
    dbms_output.put_line('NUMERO DE CESTA DONADA: ' ||
Ventrega.cesta.numero);
    dbms_output.put_line('');
    dbms_output.put_line('LISTA DE PRODUCTOS QUE CONTIENE ESA
CESTA:');
    dbms_output.put_line('-----');
    ');
    FOR i IN 1..Ventrega.cesta.listacesta.count LOOP
        dbms_output.put_line('PRODUCTO: '
            || Ventrega.cesta.listacesta(i).nombre
            || ' - '
            || Ventrega.cesta.listacesta(i).cantidad
            || ' ');
    END LOOP;
END;

```

```
|| Ventrega.cesta.listacesta(i).medida);  
END LOOP;
```