

Tarea 7: Uso de Bases de Datos Objeto-Relacionales

Francisco Javier Sueza Rodríguez

24 de mayo de 2023

Centro:	IES Aguadulce
Ciclo Formativo:	Desarrollo Aplicaciones Web (Distancia)
Asignatura:	Bases de Datos
Tema:	Tema 7 - Uso de Bases de Datos Objeto-Relacionales

Índice

1	Caso Práctico	4
2	Actividades	4
2.1	Actividad 1	4
2.1.1	Enunciado	4
2.1.2	Solución	5
2.2	Actividad 2	7
2.2.1	Enunciado	7
2.2.2	Solución	8
2.3	Actividad 3	8
2.3.1	Enunciado	8
2.3.2	Solución	8
2.4	Actividad 4	8
2.4.1	Solución	10
2.5	Actividad 5	13
2.5.1	Enunciado	13
2.5.2	Solución	13

Índice de figuras

2.1	Creación del Objeto Persona	5
2.2	Creación del Objeto Voluntario	6
2.3	Creación e inserción en la tabla Voluntariado	11
2.4	Tabla Voluntariado actualizad	11
2.5	Creación e inserción de socios	12
2.6	Tabla Socios actualizad	12

1. Caso Práctico

Una organización sin ánimo de lucro llamada LA CESTA SOLIDARIA ofrece la posibilidad de repartir alimentos a las personas que más lo necesiten. Para ello, los alimentos que los ceden gratuitamente empresas y particulares se organizan en cestas con diferentes tipos de productos y los voluntarios se encargan de repartirlas a los clientes o socios que acuden al almacén registrándose para ello con los datos personales, renta y miembros en la unidad familiar.

Hasta ahora, todo lo que hemos visto ha sido sobre bases de datos relacionales, sin embargo la aparición de las bases de datos orientadas a objetos nos ofrece nuevas perspectivas para aprovechar de manera más óptima las bases de datos.

De esta manera se trata es de ver las ventajas que nos ofrecen este tipo de bases de datos principalmente en aspectos como la reutilización de código, el trabajo colaborativo, y la interconexión de las bases de datos con otros lenguajes de programación orientados a objetos.

Como Juan ha realizado recientemente un curso de bases de datos orientadas a objetos, Ada le ha propuesto a él que utilizando los conocimientos adquiridos en el curso realice la base de datos orientada a objetos que tienen que desarrollar para la gestión de esta organización.

2. Actividades

El propósito de esta tarea es crear los objetos, tablas y bloques de código necesarios para crear la BD objeto-relacional que se propone a continuación y probarla mediante la creación de instancias creadas a partir de esos objetos.

En la organización **LA CESTA SOLIDARIA** se desea informatizar la gestión de los voluntarios que reparten diferentes donaciones en forma de cestas con productos a los clientes o socios más vulnerables dependiendo del tipo de ayuda que necesiten. Tú también puedes colaborar con esta organización informatizando una BD objeto-relacional que maneje la funcionalidad que nos solicitan. Para ello debes crear los objetos y tablas siguientes.

2.1. Actividad 1

2.1.1. Enunciado

Realiza los siguientes subapartados:

- a) Crea el tipo de objeto “**Persona**” con los siguientes atributos teniendo en cuenta que otros objetos heredarán de éste:

```
DNI          VARCHAR2(9),
nombre       VARCHAR2(30),
apellidos    VARCHAR2(40),
telefono     VARCHAR2(9),
f_ingreso    DATE
```

- b) Crea, como tipo heredado de “**Persona**”, el tipo de objeto “**Voluntario**” con los siguientes atributos:

```
puntosAcumula NUMBER(8)
```

Y el siguiente método:

- **calcularPuntosGanados**: que devolverá la cantidad resultante de multiplicar los turnos del mes que ha colaborado en la organización por 50.

c) Crea, como tipo heredado de “**Persona**”, el tipo de objeto “**Cliente**” con los siguientes atributos:

```
ingresosMes NUMBER(6,2),  
miembrosFam NUMBER(2),  
tipoAyuda VARCHAR2(1)
```

Y el siguiente método constructor:

- **Cliente(parámetros_necesarios)** : el valor del atributo tipoAyuda se inicializará de forma automática cuando se cree una instancia mediante este constructor en el que se pasan como parámetros necesarios todos los atributos excepto el de tipoAyuda que deberá inicializarse automáticamente de la siguiente forma:

- Si la renta es menor o igual que 50 el atributo tipoAyuda es A
- Si la renta es mayor que 50 y menor o igual que 100 el atributo tipoAyuda es B.
- Si la renta es mayor que 100 entonces el atributo tipoAyuda es C.

Para calcular la renta se necesitarán los datos ingreso mensual y miembros de la unidad familiar del cliente. La fórmula sería la siguiente:

```
renta = ingresosMes/miembrosFam
```

2.1.2. Solución

En este apartado vamos a empezar a crear los **tipos de objetos** que vamos a usar en nuestra base de datos:

a) En primer lugar hemos creado el objeto **Persona**, como podemos ver en la siguiente imagen.

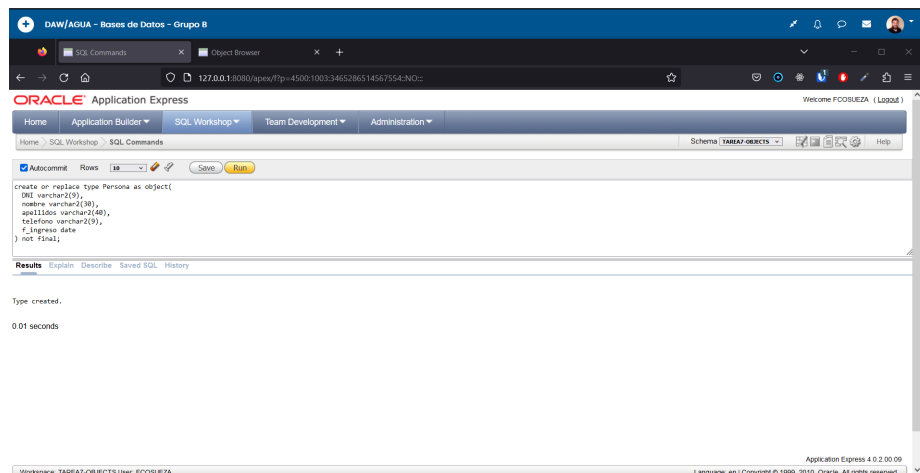


Figura 2.1: Creación del Objeto Persona

Hemos creado este objeto con los datos proporcionados, añadiendo **NOT FINAL** al final, para indicar que va a ser un objeto del que van a heredar otros. El código empleado para su definición ha sido el siguiente:

```
CREATE OR REPLACE TYPE PERSONA as object (
  DNI varchar2(9),
  nombre varchar2(30),
  apellidos varchar2(40),
  telefono varchar2(9),
  f_ingreso date
) not final
```

- b) El siguiente objeto creado ha sido el tipo **Voluntario**, el cual es un tipo **hijo** de Persona, lo que hemos indicado con la palabra **UNDER** en su definición, como podemos ver en la siguiente captura.

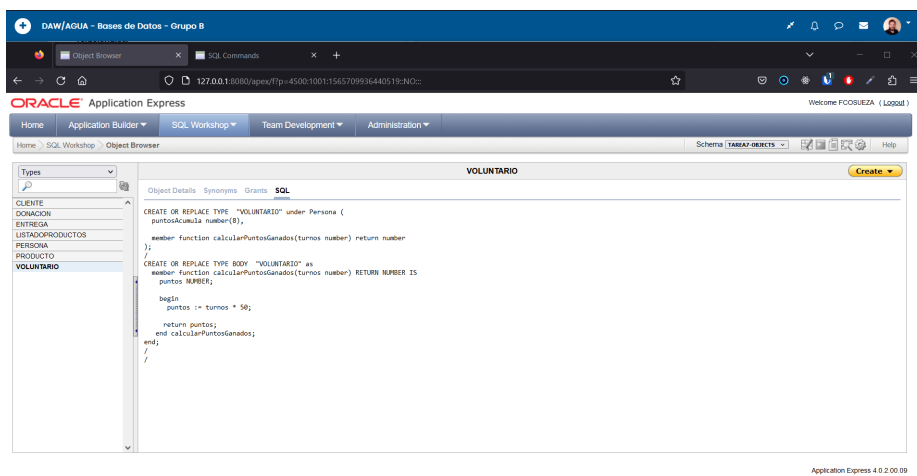


Figura 2.2: Creación del Objeto Voluntario

Además, este objeto incorpora un **método** para calcular los puntos que ha ganado un voluntario según la cantidad de turnos que ha realizado en el mes, lo cual hemos definido dentro de su correspondiente **TYPE BODY**. El código empleado ha sido el siguiente.

```
CREATE OR REPLACE TYPE VOLUNTARIO under Persona (
  puntosAcumula number(8),
  member function calcularPuntosGanados(turnos number) return number
);
/

CREATE OR REPLACE TYPE BODY VOLUNTARIO as
  member function calcularPuntosGanados(turnos number) RETURN NUMBER IS
    puntos NUMBER;

  begin
    puntos := turnos * 50;

    return puntos;
  end calcularPuntosGanados;
end;
```

- c) EL siguiente objeto creado ha sido el objeto **Cliente**, que además de heredar de persona, tiene un constructor propio, como podemos ver en el código empleado a continuación.

```

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
    );
/

-- Definición del constructor Cliente que calcula automáticamente el valor de tipoAyuda

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

        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;

        renta := ingresosMes / miembrosFam;

        IF (renta <= 50) THEN
            SELF.tipoAyuda := 'A';
        ELSIF (renta <= 100) THEN
            SELF.tipoAyuda := 'B';
        ELSE
            SELF.tipoAyuda := 'C';
        END IF;

        RETURN;
    END;

END;

```

2.2. Actividad 2

2.2.1. Enunciado

Realiza los siguientes subapartados:

- Crea el tipo de objeto “**Producto**” con los siguientes atributos:

```

codigo    VARCHAR2(3),
nombre    VARCHAR2(30),
cantidad  NUMBER(3)
medida    VARCHAR2(10)

```

- Crea una colección VARRAY llamada **ListadoProductos** en la que se puedan almacenar hasta 20 objetos de tipo Productos.
- Crea el tipo de objeto “**Donacion**” con los siguientes atributos:

```
numero    VARCHAR2(3),
valor      NUMBER(6,2),
listaCesta ListadoProductos
```

d) Crea el tipo de objeto “**Entrega**” con los siguientes atributos:

```
numero    VARCHAR2(5),
fecha      DATE,
socio      Cliente,
repartidor REF Voluntario,
cesta      Donacion
```

2.2.2. Solución

Puede ver el código de la solución en el archivo adjunto, no me ha dado tiempo a realizar la explicación. :(

2.3. Actividad 3

2.3.1. Enunciado

Crea las siguientes tablas de objetos con la sentencia apropiada para cada uno de los subapartados:

- a) Una tabla llamada VOLUNTARIADO de objetos tipo Voluntario.
- b) Una tabla llamada SOCIOS de objetos tipo Cliente.
- c) Una tabla llamada CATALOGO de objetos tipo Producto.
- d) Una tabla llamada ENTREGADOS de objetos tipo Entrega.

2.3.2. Solución

En este apartado hemos creado las diferentes tablas, usando para ello el siguiente código:

```
CREATE TABLE VOLUNTARIADO OF VOLUNTARIO;
CREATE TABLE SOCIOS OF CLIENTE;
CREATE TABLE CATALOGO OF PRODUCTO;
CREATE TABLE ENTREGADOS OF ENTREGA;
```

2.4. Actividad 4

Para trabajar con los objetos y tablas creados debes usar instancias y bloques de PL/SQL. A continuación debes implementar **diferentes bloques PL/SQL** con lo que se pide **en cada uno de los siguientes subapartados**:

- a) Crea **dos** instancias de Voluntario con los siguientes datos:

DNI	Nombre	Apellidos	Teléfono	Fecha Ingreso	Puntos Acumula
10000000A	Jaime	Sánchez Terol	111222333	15/03/2018	200
20000000B	Carmen	Mira González	888999000	8/02/2020	100

A continuación y dentro del mismo bloque realiza las siguientes acciones:

- Inserta las instancias en la tabla VOLUNTARIADO.
- Utiliza el método que has creado en este objeto para calcular los puntos ganados por el voluntario con DNI 10000000A sabiendo que ha colaborado 8 turnos en un mes.
- Por último, actualiza en la tabla VOLUNTARIADO el valor correspondiente a este voluntario incrementando los puntos ganados a su acumulado.

b) Crea **tres** instancias de Cliente con los siguientes datos y después debes insertarlos en la tabla SOCIOS.

DNI	Nombre	Apellidos	Teléfono	Fecha Ingreso	Puntos Acumula
10000000A	Jaime	Sánchez Terol	111222333	15/03/2018	200
20000000B	Carmen	Mira González	888999000	8/02/2020	100

c) Crea **ocho** instancias de Producto con los siguientes datos e insértalos en la tabla CATALOGO.

Código	Nombre	Cantidad	Medida
001	Leche	6	litros
002	Pan	3	unidades
003	Huevos	12	unidades
004	Pach Fruta	3	kilos
005	Botella Aceite	1	litros
006	Paquete Legumbres	2	kilos
007	Paquete Pasta	4	kilos
008	Paquete Galletas	6	unidades

A continuación en ese mismo bloque PL/SQL realiza las siguientes acciones:

- Crea **tres** instancias de ListadoProductos con los siguientes datos:
 - ListaProductos1 contiene los objetos de tipo Producto : 001,003,005 y 007.
 - ListaProductos2 contiene los objetos de tipo Producto: 001,002,004 y 008.
 - ListaProductos3 contiene los objetos de tipo Producto: 003,005 y 006.
- Muestra en un mensaje por pantalla los nombres de todos los productos que forman la ListaProductos1 accediendo a dicha lista y recuperando esa información. Nota: NO hay que obtenerlos de las instancias propias de Producto sino de la instancia ListaProductos1 que almacena dichos elementos en el orden insertado. Visita el siguiente enlace para ver qué métodos se pueden emplear en las colecciones.

- Crea tres instancias de Donacion con los siguientes datos:

Número	Valor	Lista
11	35	ListaProductos1 (instancia creada anteriormente)
22	30	ListaProductos2 (instancia creada anteriormente)
33	25	ListaProductos3 (instancia creada anteriormente)

- Obtén la referencia de cada voluntario almacenado en la tabla correspondiente y guarda dichas referencias en variables del tipo apropiado.
- Obtén las instancias de los clientes 11111111A y 22222222B almacenados en la tabla correspondiente y guárdalos en variables del tipo apropiado.
- A continuación crea **cuatro** instancias de Entrega con la siguiente información teniendo en cuenta que:
 - en el atributo Socio debes insertar la instancia recuperada de cliente que se indica a continuación.
 - en el atributo Repartidor debes insertar la referencia del objeto voluntario recuperada correspondiente.
 - en el atributo Cesta debes insertar el objeto creado como instancias de Donacion que se indica a continuación.

Número	Fecha	Socio	Repartidor	Cesta
00001	18/02/2023	Instancia recuperada del cliente 11111111A	Referencia recuperada del voluntario 10000000A	Instancia creada con número 11
00002	18/02/2023	Instancia recuperada del cliente 22222222B	Referencia recuperada del voluntario 10000000A	Instancia creada con número 33
00003	29/03/2023	Instancia recuperada del cliente 11111111A	Referencia recuperada del voluntario 20000000B	Instancia creada con número 22
00004	29/03/2023	Instancia recuperada del cliente 22222222B	Referencia recuperada del voluntario 10000000A	Instancia creada con número 22

- Para finalizar el bloque, una vez creadas las cuatro instancias debes insertarlas en la tabla ENTREGADOS.

2.4.1. Solución

En este ejercicio hemos empezado a trabajar ya con las tablas creadas y nuestros tipos de objetos.

- En primer lugar hemos trabajado con la tabla voluntariado, donde hemos creado dos instancias de objetos **voluntario** y las hemos insertado como se ve a continuación.

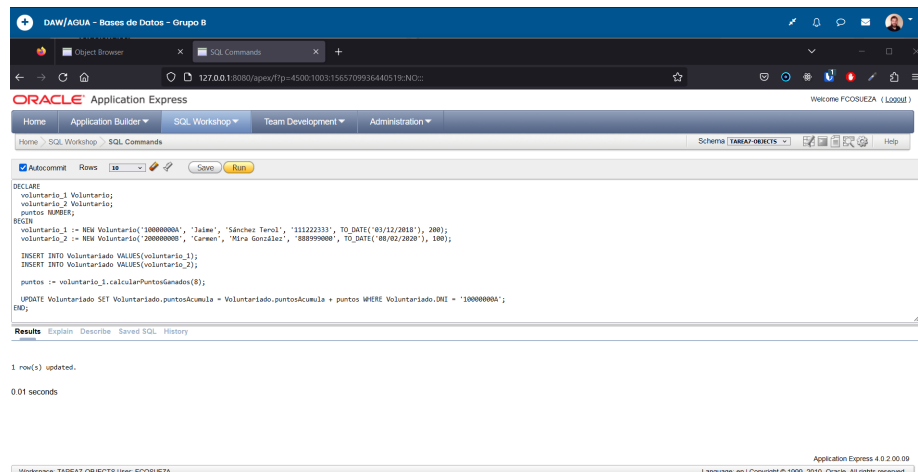


Figura 2.3: Creación e inserción en la tabla Voluntariado

Hemos realizado un SELECT para comprobar que se habían insertado correctamente los objetos, como podemos ver en la siguiente captura.

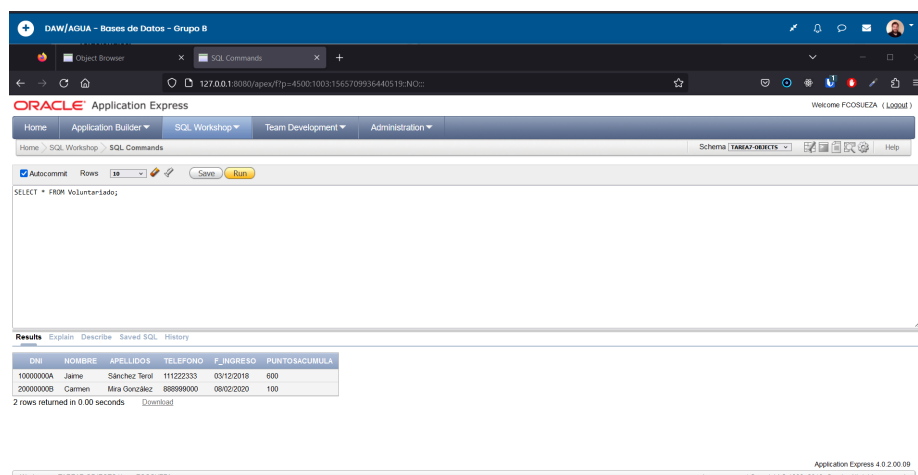


Figura 2.4: Tabla Voluntariado actualizada

El código empleado para este apartado ha sido el siguiente:

```

DECLARE
  voluntario_1 Voluntario;
  voluntario_2 Voluntario;
  puntos NUMBER;
BEGIN
  voluntario_1 := NEW Voluntario('10000000A', 'Jaime', 'Sánchez Terol', '111222333', TO_DATE('03/12/2018'), 200);
  voluntario_2 := NEW Voluntario('20000000B', 'Carmen', 'Mira González', '888999000', TO_DATE('08/02/2020'), 100);

  INSERT INTO Voluntariado VALUES(voluntario_1);
  INSERT INTO Voluntariado VALUES(voluntario_2);

  puntos := voluntario_1.calcularPuntosGanados(8);

  UPDATE Voluntariado SET Voluntariado.puntosAcumula = Voluntariado.puntosAcumula + puntos
  WHERE Voluntariado.DNI = '10000000A';
END;

```

- b) En este punto hemos trabajado con la tabla **Socios** creando varios y comprobando que el constructor que hemos creado funciona correctamente. Como vemos en la siguiente captura.

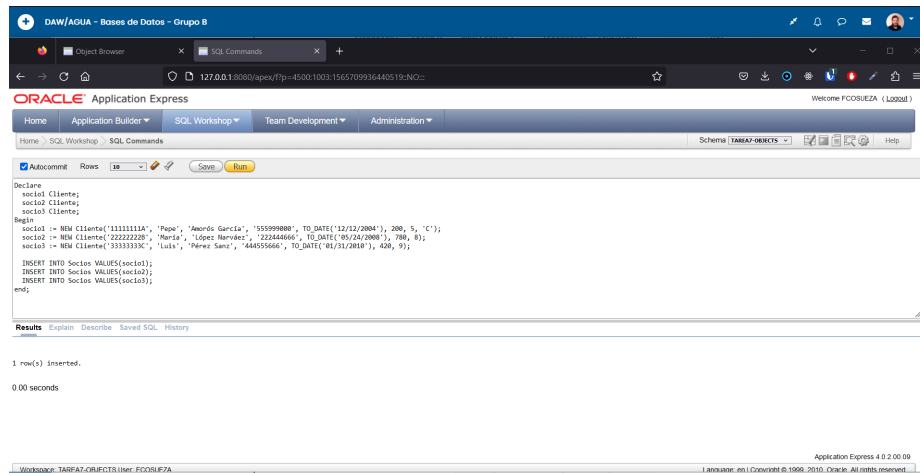


Figura 2.5: Creación e inserción de socios

Hemos comprobado que los datos se han creado correctamente, incluyendo el tipoAyuda generado automáticamente.

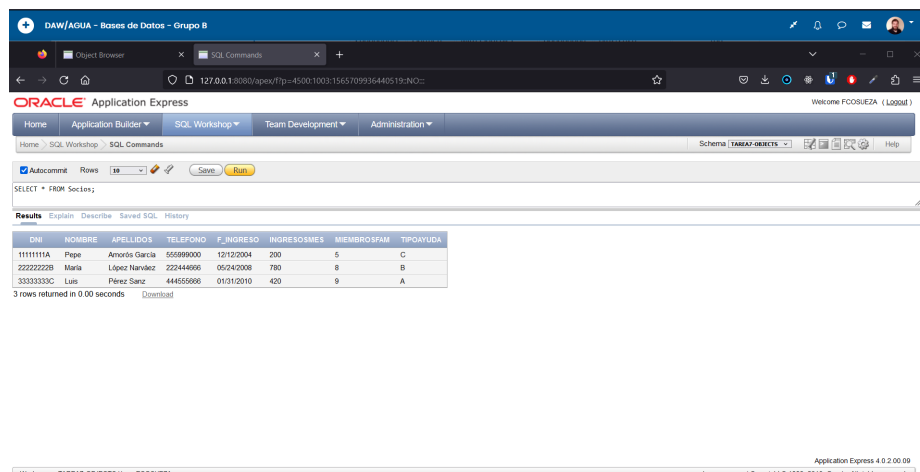


Figura 2.6: Tabla Socios actualizada

El código empleado a sido el siguiente:

```

Declare
socio1 Cliente;
socio2 Cliente;
socio3 Cliente;
Begin
socio1 := NEW Cliente('11111111A', 'Pepe', 'Amorós García', '555999000', TO_DATE('12/12/2004'), 200, 5, 'C');
socio2 := NEW Cliente('22222222B', 'Maria', 'López Narváez', '222444666', TO_DATE('05/24/2008'), 780, 8);
socio3 := NEW Cliente('33333333C', 'Luis', 'Pérez Sanz', '444555666', TO_DATE('01/31/2010'), 420, 9);

INSERT INTO Socios VALUES(socio1);
INSERT INTO Socios VALUES(socio2);
INSERT INTO Socios VALUES(socio3);
end;

```

2.5. Actividad 5

2.5.1. Enunciado

Realiza las siguientes operaciones sobre la tabla ENTREGADOS de forma independiente:

- En primer lugar debes actualizar la fecha de la entrega con número 00004 con el nuevo valor 12/04/23. (Realiza una única sentencia SQL)
- Después y mediante un bloque PL/SQL recupera información de la tabla ENTREGADOS e imprime por pantalla la siguiente información sobre una entrega determinada cualquiera: (Aquí se muestra un ejemplo de lo que tendría que salir por pantalla si buscamos información sobre la entrega número 00002.) Puedes hacerlo con cualquier otro número de entrega.

```
DATOS DONACIÓN ENTREGADA
-----
Donación entregada núm. : 00002
Fecha Entrega donación : 18/02/23
Datos socio beneficiario: María López Narváez
Entregado por voluntario: Jaime
Número de cesta donada  : 33
LISTA DE PRODUCTOS QUE CONTIENE ESA CESTA:
-----
Producto : Pack Huevos - 6 unidades
Producto : Botella Aceite - 2 litros
Producto : Paquete Legumbres - 2 kilos
```

Captura pantalla. Elaboración propia (Dominio público)

2.5.2. Solución