Diseño y Desarrollo de Sistemas de Información. Proyecto Final.

Alejandro Trujillo Caballero

20 de enero de 2015

Índice

1. Introducción

En esta práctica se pretende realizar el proceso completo de diseño e implementación de un sistema de información. Partiendo de un enunciado del cual se extraerán los requisitos, se realizará en diagrama entidad-relación del cuál se extraerá el modelo lógico a implementar en el servidor Oracle del laboratorio de práticas mediante SQL.

Para el manejo de esta base de datos, se implementará una aplicación en Java que se conectará remotamente al servidor.

2. Diseño de la base de datos.

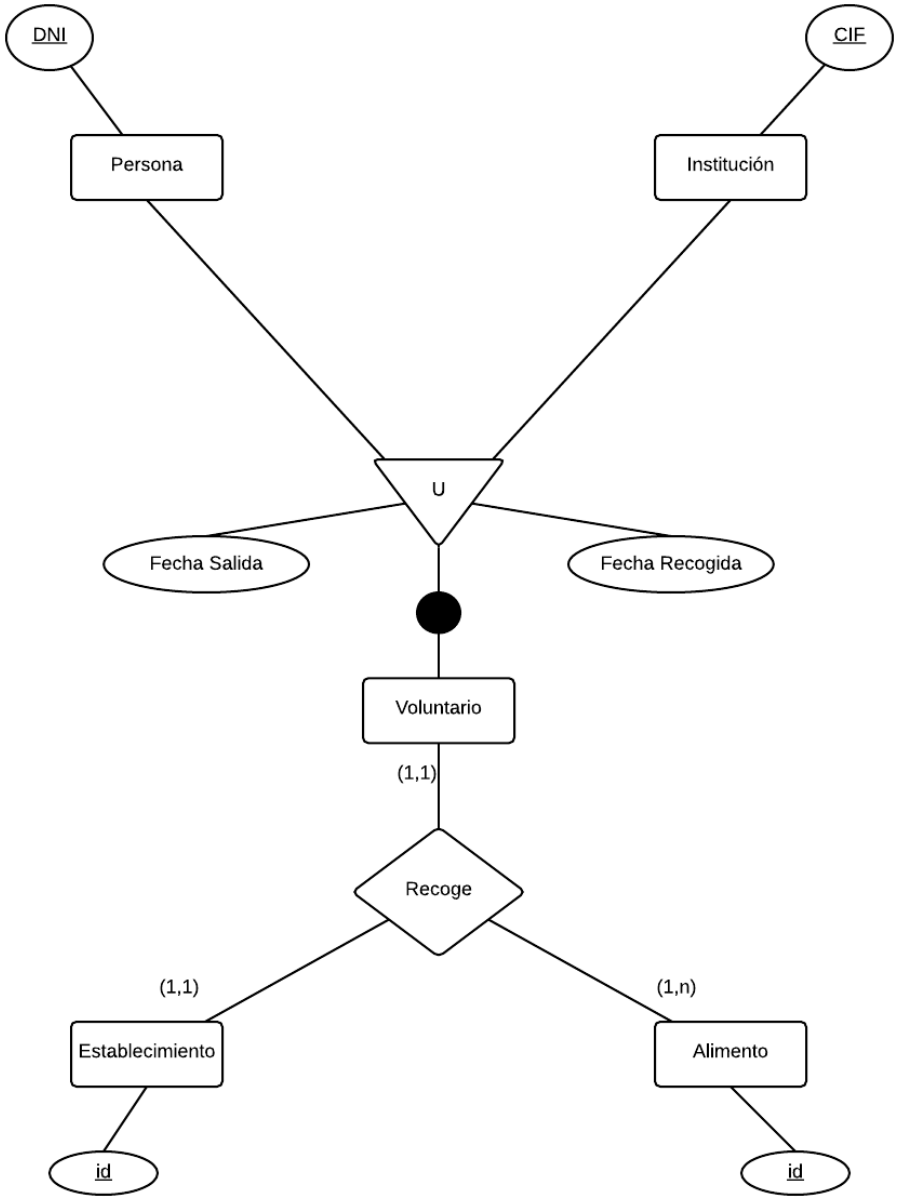

Figura 1: Diagrama Entidad-Relación de la base de datos

En la figura 1 se expone el diagrama entidad-relación diseñado para cumplir los requisitos del enunciado. Solo se han incluido los atributos que forman las claves primarias de las entidades de cara a facilitar la comprensión de la estructura general.

Los atributos de cada una de las entidades son:

Persona: DNI, nombre, apellidos, edad, correo electrónico, localidad, teléfono.

Institución: CIF, nombre, razón social, teléfono.

Establecimiento: Id, nombre, dirección, localidad.

Alimento: Id, descripción, fecha de caducidad.

3. Modelo lógico de la base de datos e implementación

Explicaremos la transformación al modelo lógico de la base de datos directamente sobre el código SQL utilizado para crearla:

```
--------- TABLA ALIMENTO ---------------------------------------
drop table ALIMENTO CASCADE CONSTRAINT;

create table ALIMENTO (
        id number,
        descripcion varchar(50) not null,
        fecha_caducidad date not null,

        CONSTRAINT alimentoClave
                PRIMARY KEY (id)
);
```

La entidad Alimento se transforma de forma directa en una tabla que contiene sus atributos y su clave primaria.

```

-------- TABLA ESTABLECIMIENTO ----------------------------------
drop table ESTABLECIMIENTO CASCADE CONSTRAINT;

create table ESTABLECIMIENTO (
        id number,
        nombre varchar(50) not null,
        direccion varchar(50) not null,
        localidad varchar(50) not null,


        CONSTRAINT establecimientoClave
                PRIMARY KEY (id)
);
```

La entidad Establecimiento, al igual que alimento se transforma en una tabla de manera directa.

```
------- TABLA VOLUNTARIO ----------------------------------------
drop table VOLUNTARIO CASCADE CONSTRAINT;

create table VOLUNTARIO (
        id number,

        CONSTRAINT voluntarioCalve
                PRIMARY KEY (id)
);
```

La categoría Voluntario se traduce como una tabla que unicamente contiene la clave mediante la cual se unificarán Persona e Institución.

```
-------- TABLA INSTITUCION -------------------------------------
drop table INSTITUCION CASCADE CONSTRAINT;

create table INSTITUCION (
        cif char(9),
        nombre varchar(50) not null,
        razon_social varchar(50),
        telefono number not null,
        idVoluntario number not null UNIQUE,

        CONSTRAINT institucionClave
                PRIMARY KEY (cif),
        CONSTRAINT voluntarioAjena
                FOREIGN KEY (idVoluntario) REFERENCES VOLUNTARIO (id) on
                    delete cascade
);
```

La entidad Institución se traduce a una tabla, con el atributo añadido idVoluntario como clave ajena a la tabla Voluntario.

```
------- TABLA PERSONA -----------------------------------------
drop table PERSONA CASCADE CONSTRAINT;

create table PERSONA (
        dni char(9),
        nombre varchar(50) not null,
        apellido1 varchar(50) not null,
        apellido2 varchar(50),
        edad number(2) not null,
        e_mail varchar(50),
        localidad varchar(50) not null,
        telefono number not null,
        idVoluntario number not null UNIQUE,

        CONSTRAINT personaClave
                PRIMARY KEY (dni),
        CONSTRAINT volAjena
                FOREIGN KEY (idVoluntario) REFERENCES VOLUNTARIO (id) on
                    delete cascade
);
```

La entidad Persona se traduce de forma análoga a Institución.

```
------ TABLA RECOGE -------------------------------------------
drop table RECOGE CASCADE CONSTRAINT;

create table RECOGE (
        alimento number,
        voluntario number not null,
        establecimiento number not null,
        fecha_recogida date not null,
        fecha_salida date,
        entregado number(1),
```

```
11
12          CONSTRAINT recogeClave
13                  PRIMARY KEY (alimento),
14          CONSTRAINT alimentoAjena
15                  FOREIGN KEY (alimento) REFERENCES ALIMENTO (id),
16          CONSTRAINT voluntarioAje
17                  FOREIGN KEY (voluntario) REFERENCES VOLUNTARIO (id) on
                        delete set null,
18          CONSTRAINT establecimientoAjena
19                  FOREIGN KEY (establecimiento) REFERENCES ESTABLECIMIENTO (
                        id) on delete set null
20
21  );
```

La relación Recoge se traduce como una tabla independiente, con claves ajenas a alimento, voluntario y establecimiento.

El control de la consistencia de las claves de idVoluntario para personas e instituciones se realizará utilizando dos disparadores:

```
1   create or replace trigger institucion_o_persona
2           before insert on INSTITUCION
3           for each row
4           DECLARE
5                   idVoluntario_erroneo EXCEPTION;
6                   cantidad number;
7           BEGIN
8                   SELECT count(*) INTO cantidad
9                   FROM PERSONA p
10                  WHERE p.idVoluntario = :new.idVoluntario;
11                  IF cantidad <> 0 THEN
12                          RAISE idVoluntario_erroneo;
13                  END IF;
14          END
15  ;
```

```
1   create or replace trigger persona_o_institucion
2           before insert on PERSONA
3           for each row
4           DECLARE
5                   idVoluntario_erroneo EXCEPTION;
6                   cantidad number;
7           BEGIN
8                   SELECT count(*) INTO cantidad
9                   FROM INSTITUCION I
10                  WHERE I.idVoluntario = :new.idVoluntario;
11                  IF cantidad <> 0 THEN
12                          RAISE idVoluntario_erroneo;
13                  END IF;
14          END
15  ;
```

4. Aplicación Java

El programa Java será un aplicación de consola que contemplará en un menú las opciones de inserción/consulta/modificación/borrado que se solicitan en el enunciado.

4.1. Estructura de clases

El código de la aplicación se divide en los cuatro paquetes que se citan a continuación, se explicará brevemente el contenido de cada uno de ellos pero no se incluirá código en esta memoria debido a su extensión.

Aplicación: Contiene la estructura básica de la aplicación, el main y la conexión al servidor.

Consola: Contiene la definición del menú de consola y el control de la entrada por teclado.

Datos: Contiene las clases que representan a cada una de las tablas y entidades de la base de datos, de cara a trabajar con ellas con facilidad.

Persistencia: Contiene las clases que manejan la lógica y la comunicación con la base de datos (inserciones, consultas...).