



Diseño y Desarrollo de Sistemas de Información

Grado en Ingeniería Informática

Proyecto de Prácticas



Objetivo

- Construir un Sistema de Información realizando las fases del modelado conceptual de la base de datos, transformación al modelo relacional e implementación con un lenguaje de programación.

Tecnologías

- Entorno de desarrollo de integrado – IDE (se recomienda Netbeans)
- Lenguaje de programación Java
- Sistema Gestor de Bases de Datos (se recomienda Oracle o MySQL)
- OPCIONAL: Herramienta de mapeo objeto relacional (ORM) Hibernate

Enunciado

El proyecto consistirá en el diseño y desarrollo (implementación) de un Sistema de Información cuyos requisitos se describen en el siguiente texto:

La Organización no Gubernamental "Nadie sin comer" ha organizado una campaña de recogida de alimentos para distribuirlos posteriormente entre familias sin recursos. Para ello, desea mantener un Sistema de Información con el que se pueda llevar una gestión del stock y un control de trazabilidad de los alimentos recogidos durante la campaña actual.

Es necesario mantener información sobre los voluntarios que participarán en la campaña. Los voluntarios pueden ser personas individuales (a título personal) o instituciones. De las personas se quiere almacenar su DNI, nombre y apellidos, teléfono de contacto, correo electrónico, edad y localidad de residencia. De las instituciones es necesario mantener el CIF, nombre, razón social y teléfono de contacto.

Cada alimento que se recoja durante la campaña debe estar identificado por un código único, una descripción y su fecha de caducidad. De los establecimientos en los que se encuentran los alimentos se almacenará su nombre, dirección y localidad, y tendrán un código único que los identifiquen.

Durante la campaña, un voluntario puede recoger alimentos en diferentes establecimientos. Habrá que mantener, por tanto, un registro de los alimentos que recogen los voluntarios en cada establecimiento y la fecha en la que lo recogió. En el sistema sólo se almacenan los alimentos que recogen los voluntarios.

Por ejemplo, el voluntario con código V1 puede recoger, el día 12/11/2016 en el establecimiento E1, tres paquetes de arroz, con códigos A123, A343 y A837, y un pack de 6 yogures de fresa con código Y838. Nótese que los packs son indivisibles, por tanto, no se guardarán 6 yogures con un código cada uno, sino que constará como un pack con un código único. El voluntario llevará los alimentos recogidos a la ONG donde se registrará en la base de datos lo que el voluntario ha entregado. Los alimentos recogidos se almacenan en las instalaciones de la ONG desde donde, posteriormente, se repartirán a las distintas familias que lo soliciten (no hay que almacenar a qué familia se ha entregado cada alimento). Además de mantener un registro de todos los alimentos recogidos es necesario saber qué alimentos ya han sido entregados y, por tanto, no se encuentran en los almacenes de la ONG.



¿Qué hay que hacer?

1. Proponer, utilizando el modelo Entidad-Relación Extendido, un esquema conceptual que recoja la semántica descrita en el enunciado.
2. Realizar la transformación del esquema conceptual al modelo lógico (modelo relacional) contemplando las restricciones planteadas en el enunciado.
3. Escribir un script para construir el esquema de la base de datos en el SGBD Oracle.
4. Escribir un script para insertar un número de registros suficientes para comprobar, posteriormente, la funcionalidad del programa.
5. Implementar, utilizando el lenguaje de programación Java, una aplicación permita, al menos, realizar las siguientes operaciones:
 - Insertar un voluntario (persona)
 - Eliminar un voluntario (institución)
 - Obtener un listado de los alimentos que se han recogido en un determinado establecimiento entre dos fechas
 - Insertar la información de la recogida de un alimento por parte de un voluntario en un determinado establecimiento
 - Actualizar el CIF de una institución
 - Obtener toda la información de los alimentos caducados que todavía están en el almacén (esto debe realizarse mediante un **procedimiento almacenado** que devuelva, en un cursor, la información solicitada). Además, la operación deberá eliminar todos los registros de los alimentos caducados que todavía no se han entregado.
 - Obtener el número total de productos recogidos por un voluntario (introduciendo su identificador por teclado). Si ese número es mayor que 5, se mostrarán los alimentos recogidos por dicho voluntario y los establecimientos en los que han sido recogidos. Si es menor o igual que 5, sólo se mostrarán los datos personales del voluntario. Recordad que esta operación **puede realizarse en un entorno concurrente**, en el que se pueden estar insertando o eliminado, a la misma vez, otros registros de recogida de alimentos.

NOTAS IMPORTANTES:

- Para que las operaciones se ejecuten correctamente deben cumplir los requisitos del enunciado y controlar todas las posibles excepciones. Aunque el control de las excepciones puede hacerse a nivel de datos (en el esquema de la base de datos) o a nivel de implementación (en el propio programa), se recomienda que se haga a nivel de datos.
- El sistema tiene que seguir, obligatoriamente, la arquitectura en tres capas: aplicación, persistencia y datos
- El mapeado puede llevarse a cabo mediante la herramienta Hibernate o bien, puede ser realizado de forma manual.

¿Qué se va a evaluar?

1. La calidad del esquema conceptual (hasta 2 puntos)
2. La transformación y el esquema lógico (hasta 1 punto)
3. La funcionalidad del programa (hasta 7 puntos)