

Práctica de Diseño de Software

Grado en Ingeniería Informática
Mención Ingeniería del Software
Universidad de Valladolid

curso 2017-2018

Capítulo 1

Supuesto práctico

Para consultar las normas de entrega vaya al final del documento (página 13).

1.1. Adaptado del enunciado de la asignatura Modelado de Software en el curso 2017-2018

En este proyecto se persigue la automatización del vivero y floristería Holanda, dedicado al cultivo y venta de plantas y flores cortadas. Las plantas y flores son producidas en el propio vivero. En caso de que el invernadero no disponga de alguna planta, se cuenta con una lista de proveedores habituales a los que solicita los pedidos oportunos. Los proveedores también facilitan productos auxiliares (fertilizantes, plaguicidas, etc.). La ubicación del vivero y de la tienda, destinada a la atención al cliente, es contigua, de forma que todas las ventas pueden ser satisfechas en la propia tienda, desde la cual se accede al vivero.

1.1.1. Restricciones adicionales

El sistema deberá utilizar una base de datos, cuyo diseño se aporta, implementada con Derby.

Deberá asegurarse que todos los usuarios están previamente identificados en el sistema para acceder a cualquier función, y que las funciones serán las que correspondan al usuario según su rol en la empresa.

...

Se ha decidido separar el sistema en dos subsistemas. El subsistema dedicado a los clientes para realizar encargos y compras será una aplicación web y no será abordado en este asignatura. El otro subsistema es el que se aborda (parcialmente) en esta asignatura suponiendo la creación de una aplicación de escritorio con acceso a BD. En teoría la BD sería centralizada en modo

cliente-servidor, pero para facilitar las prácticas en este caso supondremos la conexión a una BD local.

Se han realizado las fases de Elicitación y Especificación de Requisitos, se han descrito los requisitos funcionales, no funcionales. Se ha realizado la fase de análisis incluyendo el Modelado del Dominio y la especificación de casos de uso en análisis. A continuación se aporta una documentación parcial de requisitos y análisis que incluye: el Modelo del Dominio, un fragmento del diagrama de casos de uso y la especificación de cuatro de los casos de uso. Se aporta también el esquema de la base de datos (modelo lógico y físico) realizado en la primera fase del Diseño a partir del modelado conceptual expresado en el Modelo de Dominio.

Capítulo 2

Documentación parcial de Requisitos y Análisis

2.1. Vista parcial del diagrama de Casos de Uso

Se aporta una vista parcial del diagrama de Casos de Uso. Para esta práctica se han escogido aquellos que se encuentran resaltados y para los que se adjunta una especificación.

Se han definido distintos actores. Los que vamos a considerar en este trabajo son:

- el actor generalización Empleado
- el actor Dependiente
- el actor Operario
- el actor Administrativo

En la Figura 2.1 se muestra el mencionado fragmento del diagrama de casos de uso. Tal como se aprecia en dicha figura, para este trabajo vamos a tener en cuenta los siguientes casos de uso:

Empleado: Identificarse

Dependiente Registrar venta directa

Operario Modificar estado de lote

Administrativo Consultar facturas pendientes de pago

Es importante considerar el caso de uso “Identificarse en el sistema”. El empleado se identifica con su dni y contraseña, si todo va bien, se carga su perfil con las opciones que puede realizar este tipo de empleado. Para esta

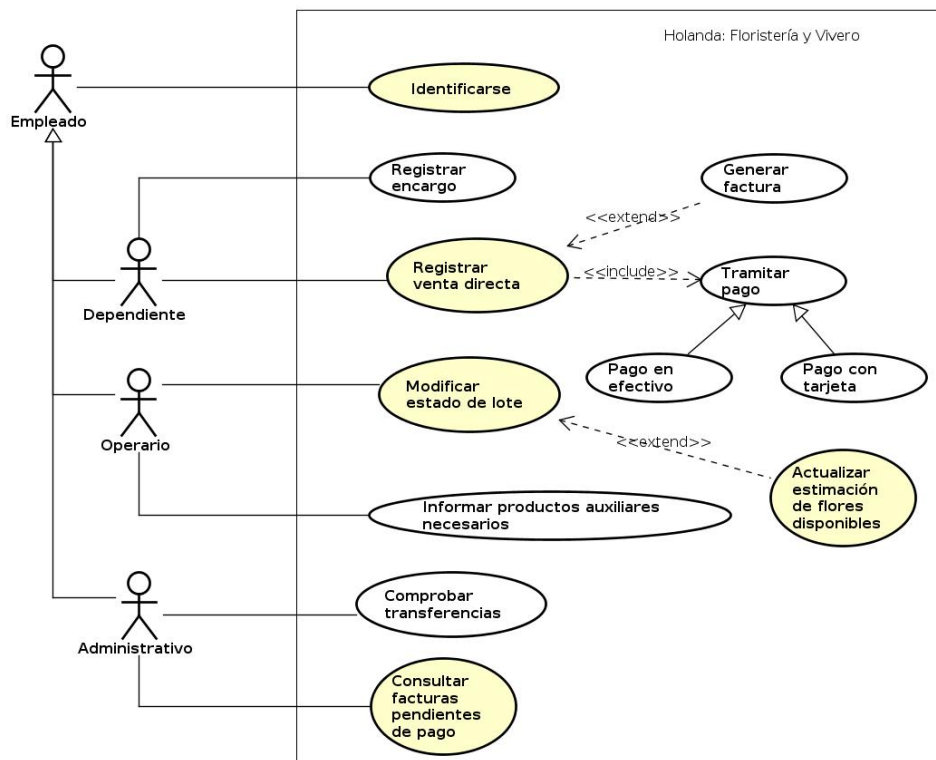


Figura 2.1: Fragmento del Diagrama de casos de uso. Se muestran sólo algunos ejemplos por actor

práctica, a cada tipo de actor se le mostrará una lista de opciones. Esta lista debe contener al menos el CU que se pide realizar para ese actor en este supuesto práctico. El resto de opciones estarán identificadas con el nombre del actor y un número. Su funcionalidad asociada será mostrar un mensaje que informe de “Opción aún no implementada”.

Los casos de uso no sombreados no se realizarán aunque tenga una relación de include/extend con los casos de usos principales.

2.2. Modelo del dominio

En la fase de análisis se ha obtenido el modelo del dominio que se muestra en la Figura 2.2. En dicho diagrama se ha omitido a propósito la especificación de restricciones OCL que sí era objetivo de la asignatura de Modelado de Sistemas Software. Debe tenerse en cuenta que éste es una simplificación a partir del supuesto práctico de la asignatura de Modelado. Nótese que no se han incorporado en este modelo de dominio algunos atributos de información y clases. El objetivo es simplificar. Por otra parte es necesario aclarar que el modelo que aquí se presenta no puede ser considerado como solución a un supuesto evaluable en la asignatura de Modelado.

En los siguientes apartados se describen los escenarios de los casos de uso mencionados.

2.3. Especificación de casos de uso

De la misma forma que se aclaró en el apartado relativo al Modelo del Dominio, la especificación de casos de uso que aquí se describe no se corresponde exactamente con lo esperado en el supuesto planteado en la asignatura de Modelado. Se han realizado muchas simplificaciones para que no sea necesario trabajar con demasiadas clases de dominio ni tablas en la base de datos.

2.3.1. Empleado: Identificarse

Actor: Empleado

Caso de Uso: Identificarse

Precondición: El actor no se encuentra identificado en el sistema

Secuencia normal:

1. El actor Empleado introduce dni y contraseña.
2. El sistema comprueba que dicho par login-password corresponden con un empleado en activo y muestra las opciones correspondientes a su rol en el sistema.

Alternativas y excepciones:

- (2 a) Si no existe empleado con dni indicado, el sistema muestra un mensaje de error, a continuación el caso de uso queda sin efecto.
- (2 b) Si la contraseña es incorrecta para el empleado con el dni indicado, el sistema muestra un mensaje de error, a continuación el caso de uso queda sin efecto.
- (2 c) Si el empleado no está en activo (contratado y trabajando), el sistema muestra un mensaje de error y a continuación el caso de uso queda sin efecto.

Postcondición: El empleado está identificado en el sistema y sus opciones de trabajo mostradas.

2.3.2. Dependiente: Registrar venta directa

2.3.3. Operario: Modificar estado de lote

2.3.4. Administrativo: Consultar facturas pendientes de pago

2.4. Restricciones a tener en cuenta en la implementación

El sistema deberá utilizar una base de datos cuyo esquema (modelo lógico) se aporta. Deberá utilizarse el script de creación de la base de datos (modelo físico) se aporta. El sistema será una aplicación de escritorio independiente del sistema operativo que se desarrollará en JAVA, jdk 8 utilizando swing y JDBC. El sistema será desarrollado en NETBEANS en la misma versión que se encuentra instalada actualmente en los laboratorios de la Escuela.

Capítulo 3

Normas de entrega y criterios de evaluación

3.1. Indicaciones para la entrega

El seguimiento del proyecto se realizará a través de la aplicación PIVOTAL TRACKER, www.pivotaltracker.com. Todos los alumnos se habrán creado una cuenta en PIVOTAL TRACKER y habrán sido añadidos al proyecto correspondiente según su equipo de trabajo. El alumno deberá estar identificado con su correo en alumnos.uva.es y su identificación como login en los laboratorios de la Escuela.

Todos los equipos disponen de un canal de comunicación en la instancia rocket de la Escuela (<https://rocket.inf.uva.es>).

Para realizar la entrega se preparará una *release* en PIVOTAL TRACKER. En dicha *release* se adjuntará un archivo rar, zip o tgz (tar.gz) con el contenido que se especifica en el apartado 3.2.

Se deja a decisión de los alumnos miembros de un equipo, que la entrega correspondiente a la *release* se realice mediante un repositorio en el GITLAB de la Escuela (cumpliendo estrictamente la estructura y contenidos que se especifican en el apartado 3.2). Esto último es opcional ya que no forma parte de los contenidos de esta asignatura pero puede ser preferencia del alumnado que ya domina estas tecnologías trabajar de esta forma. Para esto último deberán seguirse las siguientes normas:

- La entrega se realizará añadiendo a la profesora (usuario **yania**) con permisos de tipo **Reporter** al repositorio que contiene el proyecto en el GITLAB de la Escuela cuando ya no se vaya a realizar ningún *commit+push*. Cualquier *push* al repositorio una vez vencido el plazo de entrega será penalizado con 0 en la Práctica en la convocatoria correspondiente.
- El enlace (url) al repositorio y cualquier documentación necesaria al

respecto serán anotadas en la *release* en PIVOTAL TRACKER.

3.2. Estructura y contenidos de la entrega

La entrega tendrá la siguiente estructura de carpetas:

```
modelos/  
app/
```

En la carpeta **modelos** se encontrará un archivo ASTAH PROFESSIONAL profesional (del que nuestra Escuela tiene licencia académica) cuya versión debe ser la misma que la instalada actualmente en los laboratorios de la Escuela. Los modelos y diagramas contenidos en dicho archivo ASTAH PROFESSIONAL que son evaluables corresponden a lo realizado en la fase de diseño. Se tendrá cuidado de alojar todo lo realizado en un modelo llamado Diseño organizado con sus respectivos submodelos y diagramas.

Se contará con los diagramas necesarios para representar la arquitectura de referencia, los estilos universales decomposition style, uses style, inheritance style y data model así como los diagramas de clases de diseño detallado y los diagramas de secuencia con la realización en diseño de los casos de uso. Se aportará un diagrama de estados para modelar la interfaz de usuario del sistema (en el que se modelará únicamente lo necesario para los casos de uso que se especifican en este documento). Los diagramas tendrán que ser legibles y comprensibles. Si los diagramas se hacen excesivamente grandes deberán utilizarse los elementos que ofrece UML para reducir el tamaño y la complejidad de los modelos.

En la carpeta **app** se espera una estructura como la siguiente:

```
app/netbeansProject/  
app/db/
```

La carpeta **app/netbeansProject** contendrá como su nombre indica el proyecto NETBEANS que implementa el caso de uso. La implementación no debe incumplir el diseño propuesto.

La carpeta **app/db** contendrá un archivo database.config que indicará el nombre de la base de datos, el usuario y el password así como unos scripts que permitirán la regeneración de la base de datos (suministrado con este enunciado) así como tantos scripts SQL como sean necesarios para poblar automáticamente la base de datos de cara a probar la aplicación (puede ser uno o varios separados) siempre que se documente apropiadamente el propósito de cada uno.

Ni en el control de versiones, ni en los archivos de entrega debe residir la base de datos Derby.

Fecha límite para la entrega

A continuación se especifican las fechas límite de entrega para cada convocatoria. Esta será la fecha de la *release* en PIVOTAL TRACKER. Si no se cumple la fecha límite, el equipo será penalizado con 0 en la Práctica en la convocatoria correspondiente.

convocatoria ordinaria 3 de junio de 2018

convocatoria extraordinaria 24 de junio de 2018

3.3. Criterios de Evaluación

Se valorará:

- (a) la aplicación y consistencia en el diseño de la arquitectura de 3 capas (capas estrictas) combinada con MVC, se puede considerar además añadir una capa transversal de servicios (capa relajada);
- (b) la aplicación de los patrones GRASP y algunos patrones de diseño conocidos, en particular la aplicación de patrones de acceso a datos;
- (c) la corrección y completitud de los modelos UML.
- (d) la calidad de la solución.

Los criterios anteriores tendrán el mayor peso en la evaluación de la práctica (70 %) y se desglosan de la siguiente forma:

- diagramas de la arquitectura de referencia y descomposición modular- 0,5/10
- diagramas de dependencias entre capas - 0,4/10
 - ¿Se diseñan adecuadamente las dependencias necesarias para el CU Identificarse? 0,1/10
 - ¿Se diseñan adecuadamente las dependencias necesarias para el CU Registrar venta directa? - 0,1/10
 - ¿Se diseñan adecuadamente las dependencias necesarias para el CU Modificar estado de lote? - 0,1/10
 - ¿Se diseñan adecuadamente las dependencias necesarias para el CU Consultar facturas pendientes de pago? - 0,1/10
- diagramas de clases de diseño detallado - 0,5/10
- diagrama de estados que modela la interfaz - 0,1/10
- Diagramas de Secuencia Realización en Diseño-

- ¿Se diseña adecuadamente la realización del CU Identificarse (incluye mecanismo de persistencia)? - 1/10
- ¿Se diseña adecuadamente la realización del CU Registrar venta directa (incluye mecanismo de persistencia)? - 1,5/10
- ¿Se diseña adecuadamente la realización del CU Modificar estado de lote (incluye mecanismo de persistencia)? - 1,5/10
- ¿Se diseña adecuadamente la realización del CU Consultar facturas pendientes de pago)? - 1,5/10

Respecto del código de la aplicación (30 %) se valorará:

- (e) que sea consistente con el diseño arquitectónico (1/10);
- (f) que sea consistente con el modelo dinámico diseñado (es decir, con los diagramas de secuencia que describen la realización en diseño de los casos de uso) (1/10);
- (g) que se comporte según lo esperado en las situaciones válidas y que no acepte situaciones inválidas comunicando al usuario los errores que se controlan (pruebas de aceptación superadas) (1/10).

La calidad del código de la aplicación será analizada mediante PMD, SonarQube y JDepend. Los problemas de calidad detectados por estas herramientas conducirán a una penalización en la nota en este 30 %.

La forma de uso de PIVOTAL TRACKER no será evaluable pero sí obligatorio. En caso de no utilizarse la práctica no podrá ser considerada entregada.

El uso de rocket no será evaluable ni obligatorio, aunque es recomendable para introducir debates en los que pueda participar el profesor y dar sugerencias o donde preguntar dudas que pueda resolver el profesor.