Práctica de Diseño de Software

Grado en Ingeniería Informática
Mención Ingeniería del Software
Universidad de Valladolid

curso 2020-2021

Capítulo 1

Supuesto práctico

Para consultar las normas de entrega vaya al final del documento (página 20).

1.1. Punto de partida para requisitos y análisis

En este proyecto se persigue realizar el diseño e implementación (parcial) correspondiente a la informatización de la gestión de ModelPC. ModelPC es una empresa de venta y montaje de ordenadores (PCs). Construyen PCs que posteriormente serán vendidos en grandes superficies comerciales distribuidas por todo el país. La empresa posee un almacén en el que guarda una gran cantidad de componentes (tarjetas, HDDs, cajas, etc.) que proceden de compras a los fabricantes de hardware. Se realizan compras de componentes a proveedores que cuando se reciben se dan de entrada en el almacén. Por otro lado, los componentes almacenados se utilizarán en la construcción de PCs. Estos PCs son montados por los técnicos de la empresa, que trabajan en los talleres. Los PCs, a su vez, serán almacenados para su posterior distribución y venta.

El almacén se divide en dos secciones: una para almacenar componentes y otra para almacenar los PCs ya construidos. La sección de componentes se divide en zonas, y cada zona almacenará exclusivamente un tipo de componentes (tarjetas gráficas, placas base, etc.). La sección de PCs se dividirá en zonas correspondientes a distintos equipos (una zona para equipos basados en Intel Core, otra para los basados en AMD, etc.). Tanto para componentes como para PCs, cada zona se dividirá en estanterías. La combinación de una sección, una zona, y una estantería es el "espacio de almacenamiento" básico y puede estar vacío o contener uno o varios objetos (componentes o PCs, según el caso). En las recepciones de material, los componentes recibidos serán adecuadamente etiquetados y almacenados en su espacio correspondiente. Se darán de alta en el sistema, indicando tipo de componente, fecha de llegada y ubicación en el almacén, entre otros datos. El sistema proporcionará a los empleados del almacén la información acerca de qué espacios se encuentran libres según el tipo de componente que se vaya a almacenar. Se espera que el sistema proporcione información actualizada acerca de la ocupación real del almacén y de las compras que se espera recibir. Esto permitiría a los empleados del almacén planificar por adelantado las recepciones, de forma que los componentes recibidos queden adecuadamente clasificados.

Los clientes de la empresa son, en general, grandes superficies. Los pedidos que hacen los clientes (grandes superficies) no son de uno o dos PCs sino de lotes de entre 20 y 100 ordenadores, según la temporada. Las configuraciones básicas de los PCs son impuestas por la dirección de la empresa ModelPC, y los clientes solicitan PCs de acuerdo con estas configuraciones prefijadas (no se hacen PCs "a medida" del cliente). Para cada configuración, se establecen una serie de parámetros básicos (tipo CPU, velocidad CPU, capacidad RAM, capacidad disco, etc.). A la hora de responder a los pedidos, debe contarse con existencias suficientes de los componentes requeridos.

El almacenaje de los PCs debería realizarse según configuraciones similares, para evitar búsquedas prolongadas por el almacén hasta dar con los PCs requeridos.

Cuando se procede a la preparación de pedidos, se desea que el sistema proporcione información actualizada acerca de los PCs que se encuentran en el almacén, diciendo qué PCs se encuentran reservados para su venta y cuáles no.

Se desea también poder predecir con cierta exactitud la disponibilidad de los pedidos realizados por los clientes. Por ejemplo, si un determinado cliente solicita 70 PCs y éstos no se encuentran en existencias, mediante la información que proporcione el sistema debería ser posible determinar si disponemos de material suficiente o no para construir los 70 PCs. Si se dispone de este material, entonces dicho pedido podrá ser servido al cabo de un tiempo y los componentes afectados podrían ser marcados como reservados"(para que no sean utilizados en la construcción de otro PC). Si no se dispone del material, se puede realizar un pedido de los componentes necesarios para construirlos.

(. . .)

En el proceso de montaje de los ordenadores, se seleccionan las piezas adecuadas del almacén. Dichas piezas, por tanto, dejan de pertenecer al stock de piezas. Al realizar el montaje del PC siempre se añade una etiqueta a cada PC finalizado y se indica su configuración. Posteriormente se almacena el PC. Dicho PC, por tanto, pasa a formar parte del stock.

(. . .)

Para el montaje de PCs se debe poder elegir los componentes por medio del sistema. Consultando por una descripción de componentes (marca y modelo), se podrán obtener listas de componentes que se encuentran en el almacén. Igualmente, para una descripción de componentes (marca y modelo) se podrá consultar cuántos se encuentran en almacén y cuántos se esperan recibir antes de una fecha dada.

(. . .)

El sistema deberá responder a consultas acerca de qué PCs en stock corresponden a un determinado pedido de un cliente. Igualmente, el sistema proporcionará información acerca de la localización en el almacén de los ordenadores correspondientes a un pedido de un cliente.

1.1.1. Restricciones adicionales

Se ha decidido separar el sistema en dos subsistemas, el que será desarrollado para el uso de los empleados de la empresa y el que será desarrollado para los clientes. El subsistema dedicado a los abonados será una aplicación web y no será abordada en esta asignatura sino probablemente en otra (¿?). Se está estudiando aún la forma en la que se identificarán los clientes ya que podría permitirse identificación a través cualquier proveedor de identidades OpenID.

Para el subsistema dedicado a los empleados de la empresa se ha decidido realizar una aplicación de escritorio con acceso a BD. El sistema deberá utilizar una base de datos (BD), 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.

En teoría la BD sería centralizada en modo cliente-servidor, pero para facilitar las prácticas en este caso supondremos la conexión a una BD en local. El acceso a los datos deberá ser diseñado e implementado de forma que sea inmediato realizar los cambios necesarios para que la BD pase a ser remota y centralizada, o se decida utilizar otro sistema gestor de bases de datos relacionales.

(. . .)

Se han realizado las fases de Elicitación y Especificación de Requisitos. Se han descrito los requisitos funcionales, no funcionales, de interacción y de información.

A partir de los requisitos, en la fase de análisis se ha realizado el Modelado del Dominio y la especificación de casos de uso en análisis. A continuación se aporta una documentación parcial de 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 diseño de la base de datos (diseño 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 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 Gerente de Ventas
- el actor Técnico del Taller
- el actor Personal de almacén

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

Empleado CU Identificarse

Personal de almacén CU Buscar espacios disponibles para PCs

Gerente de Ventas CU Procesar pedido PCs

Técnico del Taller CU Registrar montaje PC

Aclaraciones sobre el caso de uso "Identificarse": 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 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. Para listar el resto de opciones se utilizarán los otros ejemplos de CU que se muestran en la Figura 2.1. La funcionalidad asociada a estas opciones será mostrar un mensaje que informe de "Opción aún no implementada".

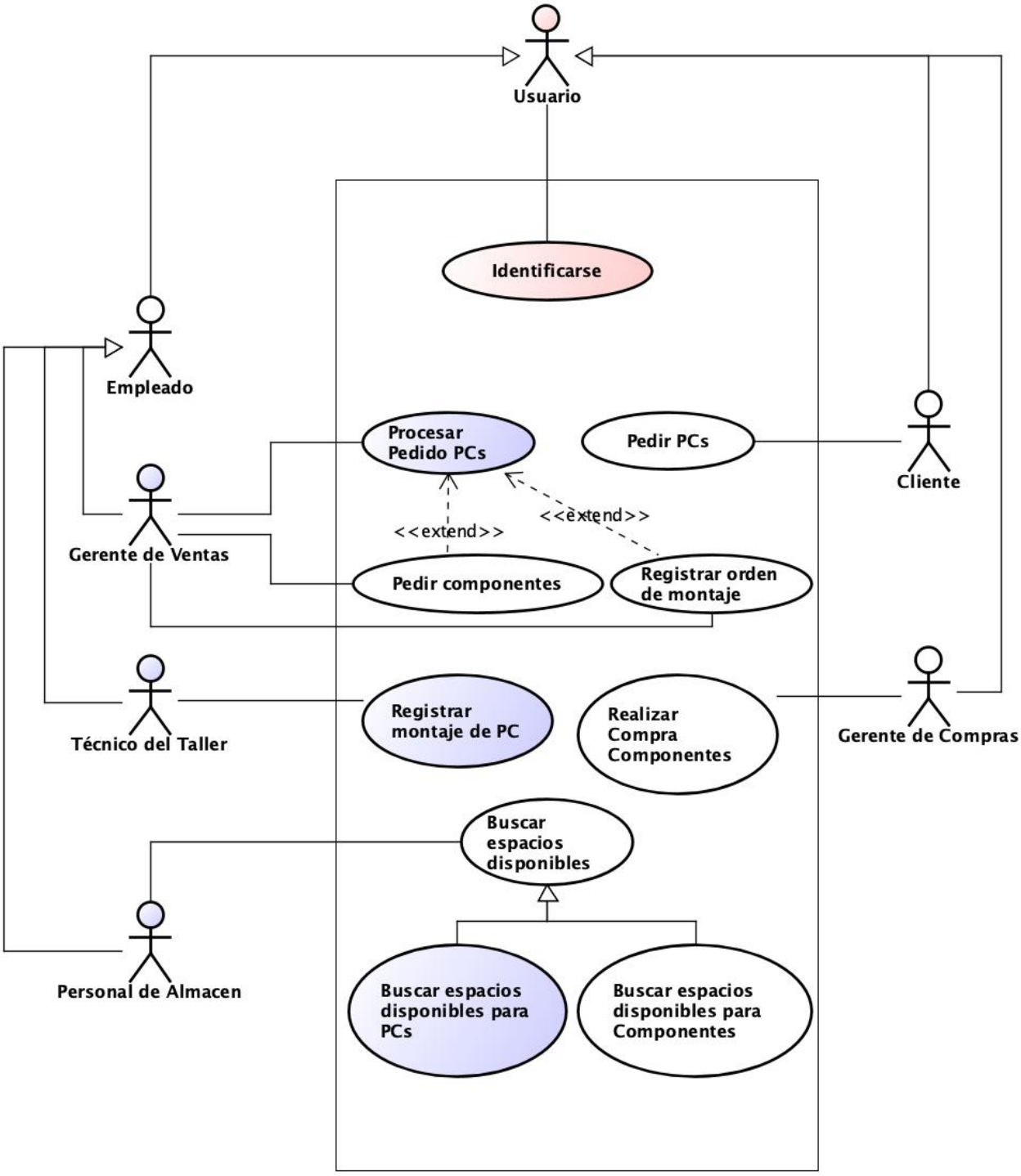

Figura 2.1: Fragmento del Diagrama de casos de uso. Se muestran sólo algunos ejemplos por actor.

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.

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

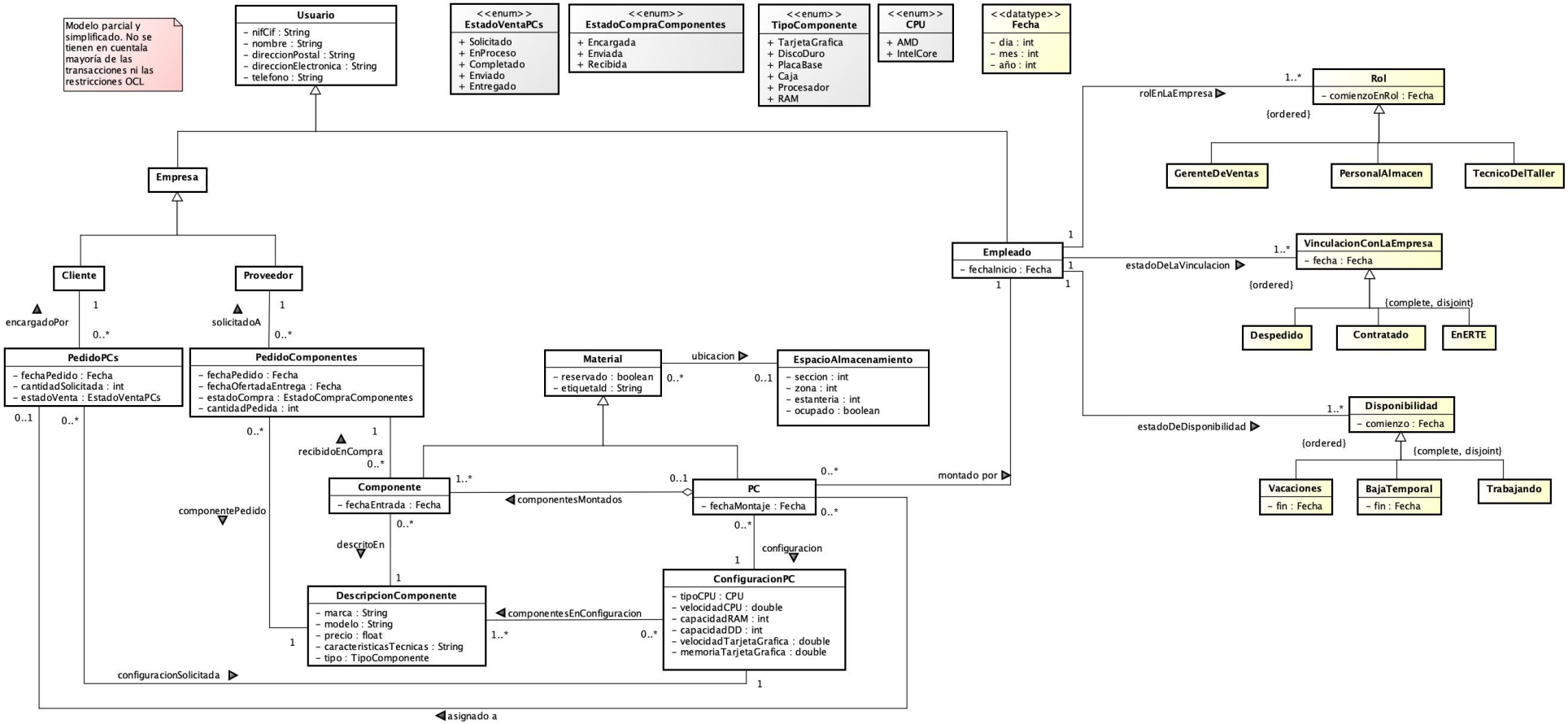

Figura 2.2: Modelo del Dominio. Punto de partida para el diagrama de clases inicial.

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 con un caso real. 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. 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 dni-contraseña 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 (activo es: 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. Buscar espacios disponibles para PCs

Actor: Personal de Almacén

Caso de Uso: Buscar espacios disponibles para PCs

Precondición: El actor se encuentra identificado en el sistema

Secuencia normal:

1. El actor solicita buscar espacios disponibles para almacenar PCs.
2. El sistema solicita el tipo de CPU para determinar la zona, y la cantidad de espacios a localizar.
3. El actor indica el tipo de CPU y la cantidad de espacios a localizar.
4. El sistema comprueba que se trata de un tipo y número correctos. Busca la cantidad de espacios disponibles solicitados en la zona de almacenamiento determinada por la CPU, muestra la lista con dichos espacios y el caso de uso finaliza.

Alternativas y excepciones:

(3a) El actor cancela y el caso de uso queda sin efecto.

(4a) Si el sistema comprueba que no se ha introducido un tipo de CPU correcto, muestra un mensaje de error y el caso de uso vuelve al paso (3).

(4b) Si el sistema comprueba que no se ha introducido una cantidad correcta, muestra un mensaje de error y el caso de uso vuelve al paso (3).

(4c) Si el sistema comprueba que no hay cantidad suficiente de espacios disponibles para la solicitud, muestra la información de aquellos espacios que sí están disponibles y un mensaje de cuántos faltan para completar la solicitud, y el caso de uso finaliza.

(4d) Si el sistema comprueba que no hay ningún espacio disponible en la zona de almacenamiento muestra un mensaje de aviso y el caso de uso finaliza.

Postcondición: En el caso de éxito se ha mostrado la lista con los espacios disponibles. En el caso de no espacios suficientes, se ha mostrado además cuántos espacios faltan para completar la solicitud.

2.3.3. Procesar pedido PCs

Actor: Gerente de Ventas

Caso de Uso: Procesar pedido PCs

Precondición: El actor se encuentra identificado en el sistema

Secuencia normal:

1. El actor selecciona la opción de procesar pedido.
2. El sistema muestra la lista de pedidos de PCs en estado "solicitado".
3. El actor elige un pedido de la lista.
4. El sistema muestra los datos del pedido, la configuración solicitada y la cantidad de PCs.
5. El actor solicita reservar PCs en stock con la configuración solicitada.
6. El sistema busca y muestra la cantidad de PCs en stock con la configuración solicitada que no están reservados. El sistema comprueba que no hay suficientes PCs en stock de la configuración solicitada, reserva los PCs disponibles y los asocia con el pedido; muestra un mensaje indicando cuántos PCs se reservan y cuántos deben montarse.
7. El actor solicita ver la disponibilidad de componentes para la configuración solicitada.
8. El sistema muestra por cada tipo de componente en la configuración solicitada la cantidad disponible (no reservados) en almacén. El sistema comprueba que hay suficientes para montar los PCs necesarios para completar el pedido. Pasa el pedido al estado "en proceso". **Se ejecuta el caso de uso Registrar orden de montaje**. El sistema solicita confirmación para procesar el pedido.
9. El actor confirma los cambios en el pedido.
10. El sistema registra los cambios y el Caso de Uso finaliza.

Alternativas y excepciones:

(2a) Si no hay PCs en estado "solicitado", el sistema muestra un mensaje y el caso de uso finaliza.

(3a, 5a, 7a, 9a) Si el actor cancela, el Caso de Uso queda sin efecto.

(6a) Si hay suficientes PCs en stock para completar el pedido, el sistema los marca como "reservado" y los asocia al pedido, pasa el pedido al estado "completado", solicita confirmación para actualizar el pedido y el CU continúa por el paso 9.

(8a) Si no hay componentes suficientes en stock, el sistema muestra un mensaje, si el actor lo indica **se ejecuta el Caso de Uso Pedir Componentes**, y el caso de uso queda sin efecto.

Postcondición: Escenario de éxito: El pedido de PCs del cliente queda en estado "en proceso" o "completado".

2.3.4. Registrar montaje PC

Actor: Técnico de Taller

Caso de Uso: Registrar montaje PC

Precondición: El actor se encuentra identificado en el sistema

Secuencia normal:

1. El actor selecciona la opción de registrar montaje de PC.
2. El sistema muestra las configuraciones de PCs
3. El actor selecciona la configuración correspondiente al PC montado.
4. El sistema muestra los pedidos, asociados a la configuración elegida, que están en estado "en proceso", ordenados por fecha de realización del pedido.
5. El actor selecciona un pedido.
6. El sistema solicita introducir etiqueta identificativa del nuevo PC.
7. El actor introduce etiqueta identificativa.
8. El sistema comprueba que no existe otro material almacenado con la misma etiqueta identificativa, registra el nuevo PC montado (marcándolo como "reservado"), y solicita introducir la etiqueta identificativa de los componentes utilizados en el montaje.
9. El actor introduce la etiqueta identificativa de cada componente montado en el PC.
10. El sistema comprueba la etiqueta identificativa introducida y agrega el componente al PC.
11. El actor confirma el registro del montaje del PC.
12. El sistema comprueba que ya se han introducido todos los componentes, asocia el PC montado y reservado con el pedido del cliente, seleccionado previamente por el actor.
13. El sistema registra los cambios y el caso de uso finaliza.

Alternativas y excepciones:

(3a, 5a, 7a, 9a, 11a) Si el actor cancela, el Caso de Uso queda sin efecto.

(8) Si existe otro material con la etiqueta identificativa introducida, el sistema muestra un mensaje de error y el caso de uso vuelve al paso 6.

(10a) Si la etiqueta identificativa no pertenece a un componente descrito en la configuración, el sistema muestra un mensaje de error y el caso de uso vuelve al paso 9.

(10b) Si la etiqueta identificativa pertenece a un componente ya añadido, el sistema muestra un mensaje y el caso de uso vuelve al paso 9.

(10c) Si no existe material con la etiqueta introducida, el sistema muestra un mensaje de error y el caso de uso vuelve al paso 9.

(11) Si aún quedan por agregar más componentes descritos en la configuración del PC, el actor continúa añadiendo por el paso 9.

(12a) Si el sistema comprueba que faltan componentes por añadir para completar la configuración, muestra un aviso y vuelve al paso 9.

(12b) Si es el último PC necesario para completar el pedido, el sistema asocia el PC al pedido, pasa el pedido del cliente al estado "completado" y el caso de uso continúa por el paso 13.

Postcondición: Se ha creado un nuevo PC, asociado con la configuración que lo describe, el PC es un agregado de un conjunto de componentes descritos en dicha configuración. El PC queda marcado como "reservado" para el pedido del cliente. Si era el último PC necesario para completar el pedido del cliente, el pedido pasa al estado "completado".

2.4. Restricciones a tener en cuenta en la implementación

El sistema deberá utilizar una base de datos cuyo esquema (diseño lógico) se aporta en la página 14, implementada con Derby. Deberá utilizarse el script de creación de la base de datos (diseño físico) se aporta en la página 15. El sistema será una aplicación de escritorio independiente del sistema operativo que se desarrollará en JAVA, jdk 11 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 (actualmente Netbeans 12.0).

2.5. Análisis arquitectónico

En el subsistema para el trabajo interno de la empresa se ha decidido aplicar una arquitectura de capas con tres capas estricta y una relajada de servicios útiles a todas las capas, MVC y DAO+DTO como patrón de acceso a datos. El DTO será una cadena JSON.

Capítulo 3

Diseño de Datos

3.1. Diseño lógico (datamodel style)

En la Figura 3.1 se muestra el diseño de los datos como esquema relacional (diseño lógico), utilizando una representación basada en UML con estereotipos. Los estereotipos «PK» y «FK» representan claves primarias y claves foráneas, respectivamente. Los clasificadores con el estereotipo «table» representan una tabla en el esquema relacional. Las tablas que representan un elemento del dominio se identifican con el estereotipo «entity». Las tablas que representa asociaciones se identifican con el estereotipo «association». Las tablas que representan valores enumerados se identifican con el estereotipo «enum» y aparecen sombreadas. La transformación al modelo relacional de la relación de herencia (Persona, Empleado, Abonado) presente en el Modelo del Dominio se ha realizado mediante referencias entre tablas. El modelado de roles temporales presentes en el Modelo del Dominio para Empleado se ha realizado mediante tablas asociación y tablas enum.

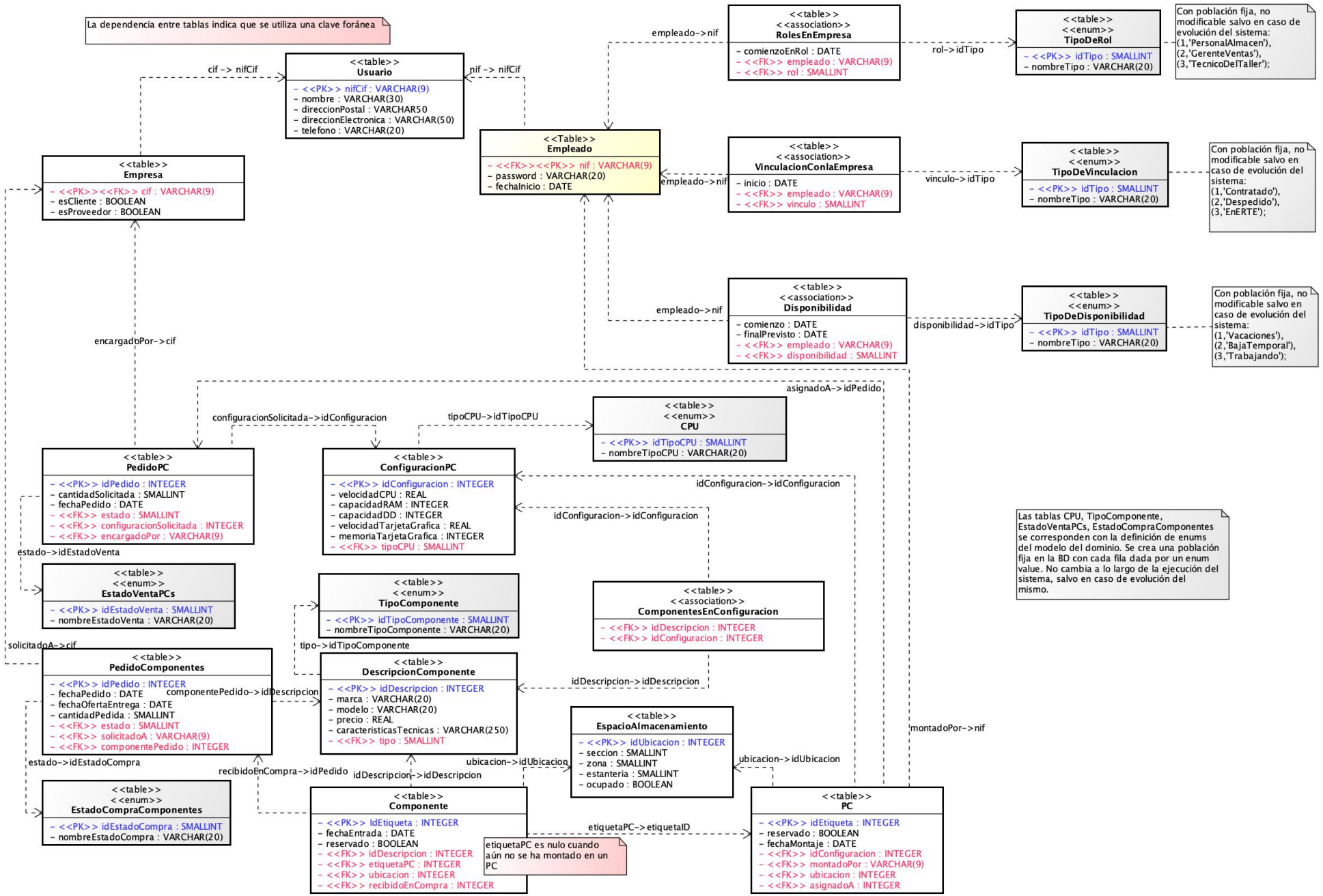

Figura 3.1: Diseño lógico de la base de datos relacional.

3.2. Scripts de creación de la base de datos

Este script puede obtenerse en un archivo almacenado como recurso en el aula virtual siguiendo el enlace: `https://aulas.inf.uva.es/pluginfile.php/61332/mod_resource/content/1/createTables.sql`.

```
--Derby does not support DROP TABLE IF EXISTS
DROP TABLE DISPONIBILIDAD;
DROP TABLE VINCULACIONCONLAEMPRESA;
DROP TABLE ROLESENEMPRESA;
DROP TABLE TIPODEDISPONIBILIDAD;
DROP TABLE TIPODEVINCULACION;
DROP TABLE TIPODEROL;
DROP TABLE COMPONENTE;
DROP TABLE PC;
DROP TABLE PEDIDOPC;
DROP TABLE COMPONENTESENCONFIGURACION;
DROP TABLE CONFIGURACIONPC;
DROP TABLE CPU;
DROP TABLE PEDIDOCOMPONENTES;
DROP TABLE ESTADOCOMPRACOMPONENTES;
DROP TABLE DESCRIPCIONCOMPONENTE;
DROP TABLE TIPOCOMPONENTE;
DROP TABLE ESTADOVENTAPCS;
DROP TABLE ESPACIOALMACENAMIENTO;
DROP TABLE EMPRESA;
DROP TABLE EMPLEADO;
DROP TABLE USUARIO;

-- Enum
create table TIPODEROL
(
        IdTipo SMALLINT not null,
        NombreTipo VARCHAR(20) not null unique,
                PRIMARY KEY(IdTipo)
);

INSERT INTO TIPODEROL
VALUES  (1,'PersonalAlmacen'),
        (2,'GerenteVentas'),
        (3,'TecnicoDelTaller');

-- Enum
create table TIPODEVINCULACION
(
        IdTipo SMALLINT not null,
        NombreTipo VARCHAR(20) not null unique,
                PRIMARY KEY(IdTipo)

);

INSERT INTO TIPODEVINCULACION
VALUES  (1,'Contratado'),
        (2,'Despedido'),
        (3,'EnERTE');

-- Enum
create table TIPODEDISPONIBILIDAD
(
        IdTipo SMALLINT not null,
        NombreTipo VARCHAR(20) not null unique,
                PRIMARY KEY(IdTipo)
);

INSERT INTO TIPODEDISPONIBILIDAD
```

```
VALUES  (1,'Vacaciones'),
        (2,'BajaTemporal'),
                (3, 'Trabajando');


create table USUARIO
(
        NifCif VARCHAR(9) not null primary key,
        Nombre VARCHAR(30) not null,
        DireccionPostal VARCHAR(50),
        DireccionElectronica VARCHAR(50),
        Telefono VARCHAR(20)
);

create table EMPRESA
(
        Cif VARCHAR(9) not null primary key,
        EsCliente BOOLEAN,
        EsProveedor BOOLEAN,
            FOREIGN KEY(Cif) REFERENCES USUARIO(NifCif)
);

create table EMPLEADO
(
        Nif VARCHAR(9) not null primary key,
        Password VARCHAR(20) not null,
        FechaInicio DATE not null,
            FOREIGN KEY(Nif) REFERENCES USUARIO(NifCif)
);

-- Association
create table ROLESENEMPRESA
(
        ComienzoEnRol DATE not null,
        Empleado VARCHAR(9) not null,
        Rol SMALLINT not null,
            FOREIGN KEY(Empleado) REFERENCES EMPLEADO(Nif),
            FOREIGN KEY(Rol) REFERENCES TIPODEROL(IdTipo)
);

-- Association
create table VINCULACIONCONLAEMPRESA
(
        inicio DATE not null,
        Empleado VARCHAR(9) not null,
        Vinculo SMALLINT not null,
                FOREIGN KEY(Empleado) REFERENCES EMPLEADO(Nif),
                FOREIGN KEY(Vinculo) REFERENCES TIPODEVINCULACION(IdTipo)
);

-- Association
create table DISPONIBILIDADEMPLEADO
(
        Comienzo DATE not null,
        FinalPrevisto DATE,
        Empleado VARCHAR(9) not null,
        Disponibilidad SMALLINT not null,
                FOREIGN KEY(Empleado) REFERENCES EMPLEADO(Nif),
                FOREIGN KEY(Disponibilidad) REFERENCES TIPODEDISPONIBILIDAD(IdTipo)
);

create table ESPACIOALMACENAMIENTO
(
        IdUbicacion INTEGER not null primary key,
        Seccion SMALLINT,
        Zona SMALLINT not null,
```

```
        Estanteria SMALLINT,
        Ocupado BOOLEAN
);

-- Enum
create table CPU
(
        IdTipoCPU SMALLINT not null primary key,
        NombreTipoCPU VARCHAR(20) not null
);

INSERT INTO CPU
VALUES  (1,'AMD'),
        (2,'IntelCore');


create table CONFIGURACIONPC
(
        IdConfiguracion INTEGER not null primary key,
        TipoCPU SMALLINT not null,
        VelocidadCPU REAL not null,
        CapacidadRAM INTEGER not null,
        CapacidadDD INTEGER not null,
        VelocidadTarjetaGrafica REAL,
        MemoriaTarjetaGrafica INTEGER,
            FOREIGN KEY(TipoCPU) REFERENCES CPU(IdTipoCPU)
);

--Enum
create table TIPOCOMPONENTE
(
        IdTipoComponente SMALLINT not null primary key,
        NombreTipoComponente VARCHAR(20) not null
);

INSERT INTO TIPOCOMPONENTE
VALUES  (1,'TarjetaGrafica'),
        (2,'DiscoDuro'),
        (3,'PlacaBase'),
        (4,'Caja'),
        (5,'Procesador'),
        (6,'RAM');

create table DESCRIPCIONCOMPONENTE
(
        IdDescripcion INTEGER not null primary key,
        Tipo SMALLINT not null,
        Marca VARCHAR(20) not null,
        Modelo VARCHAR(20),
        Precio REAL,
        CaracteristicasTecnicas VARCHAR(250),
            FOREIGN KEY(Tipo) REFERENCES TIPOCOMPONENTE(IdTipoComponente)
);

-- Association
create table COMPONENTESENCONFIGURACION
(
        IdDescripcion INTEGER not null,
        IdConfiguracion INTEGER not null,
            FOREIGN KEY(IdConfiguracion) REFERENCES CONFIGURACIONPC(IdConfiguracion)
                ,
            FOREIGN KEY(IdDescripcion) REFERENCES DESCRIPCIONCOMPONENTE(
                IdDescripcion)
);

-- Enum
```

```sql
create table ESTADOVENTAPCS
(
        IdEstadoVenta SMALLINT not null primary key,
        NombreEstadoVenta VARCHAR(20) not null
);

INSERT INTO ESTADOVENTAPCs
VALUES  (1,'Solicitado'),
        (2,'EnProceso'),
        (3,'Completado'),
        (4,'Enviado'),
        (5,'Entregado');

create table PEDIDOPC
(
        IdPedido INTEGER not null primary key,
        CantidadSolicitada SMALLINT not null,
        FechaPedido DATE not null,
    Estado SMALLINT not null,
        ConfiguracionSolicitada INTEGER not null,
        EncargadoPor VARCHAR(9) not null,
            FOREIGN KEY(ConfiguracionSolicitada) REFERENCES CONFIGURACIONPC(
                IdConfiguracion),
            FOREIGN KEY(EncargadoPor) REFERENCES EMPRESA(Cif),
            FOREIGN KEY(Estado) REFERENCES ESTADOVENTAPCS(IdEstadoVenta)
);

-- Enum
create table ESTADOCOMPRACOMPONENTES
(
        IdEstadoCompra SMALLINT not null primary key,
        NombreEstadoCompra VARCHAR(20) not null
);

INSERT INTO ESTADOCOMPRACOMPONENTES
VALUES  (1,'Encargada'),
        (2,'Enviada'),
        (3,'Recibida');

create table PEDIDOCOMPONENTES
(
        IdPedido INTEGER not null primary key,
        ComponentePedido INTEGER not null,
        CantidadPedida SMALLINT,
        FechaPedido DATE not null,
        FechaOfertaEntrega DATE,
        Estado SMALLINT not null,
        SolicitadoA VARCHAR(9) not null,
        FOREIGN KEY(SolicitadoA) REFERENCES EMPRESA(Cif),
            FOREIGN KEY(ComponentePedido) REFERENCES DESCRIPCIONCOMPONENTE(
                IdDescripcion),
            FOREIGN KEY(Estado) REFERENCES ESTADOCOMPRACOMPONENTES(IdEstadoCompra)
);

create table PC
(
        IdEtiqueta INTEGER not null primary key,
        Reservado BOOLEAN,
        FechaMontaje DATE not null,
        IdConfiguracion INTEGER not null,
        MontadoPor VARCHAR(9) not null,
        IdPedido INTEGER,
        Ubicacion INTEGER,
            FOREIGN KEY(IdConfiguracion) REFERENCES CONFIGURACIONPC(IdConfiguracion)
                ,
            FOREIGN KEY(MontadoPor) REFERENCES EMPLEADO(Nif),
```

```
            FOREIGN KEY(Ubicacion) REFERENCES ESPACIOALMACENAMIENTO(IdUbicacion),
                FOREIGN KEY(IdPedido) REFERENCES PEDIDOPC(IdPedido)
);

create table COMPONENTE
(
        IdEtiqueta INTEGER not null primary key,
        FechaEntrada DATE not null,
        Reservado BOOLEAN,
        IdDescripcion INTEGER not null,
        EtiquetaPC INTEGER,
        RecibidoEnCompra INTEGER not null,
        Ubicacion INTEGER,
            FOREIGN KEY(Ubicacion) REFERENCES ESPACIOALMACENAMIENTO(IdUbicacion),
            FOREIGN KEY(EtiquetaPC) REFERENCES PC(IdEtiqueta),
            FOREIGN KEY(RecibidoEnCompra) REFERENCES PEDIDOCOMPONENTES(IdPedido),
            FOREIGN KEY(IdDescripcion) REFERENCES DESCRIPCIONCOMPONENTE(
                IdDescripcion)
);
```

Capítulo 4

Normas de entrega y criterios de evaluación

4.1. Indicaciones para la entrega

El seguimiento del proyecto se realizará a través de la aplicación Pivotal Tracker: `https://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.

La comunicación con los equipos puede hacer mediante Pivotal Tracker o mediante el 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 indicará la url de un repositorio en el GitLab de la Escuela. El contenido del repositorio debe cumplir estrictamente la estructura y contenidos que se especifican en el apartado 4.2).

Para esto último deberán seguirse las siguientes normas:

- La entrega se realizará añadiendo a la profesora (usuario `yania`/`marga`, según corresponda) 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.
- La versión final correspondiente a la release deberá estar integrada en la rama master.

4.2. Estructura y contenidos de la entrega

La entrega tendrá la siguiente estructura de carpetas:

```
models/
app/
```

En la carpeta `models` se encontrará un archivo Astah Professional o Visual Paradigm (a elección del equipo). Los modelos y diagramas contenidos en dicho archivo Astah Professional o Visual Paradigm 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 styl*e, *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 que se piden en este enunciado. Adicionalmente, 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 config.db de propiedades con lo necesario para conectarse con la BD. Este archivo indicará la url con el puerto, el nombre de la base de datos, el usuario y el password exactamente como se indica a continuación:

```
url=jdbc:derby://localhost:1527/modelpc
user=adminmodelpc
password=modelpc2021
```

En dicha carpeta (`app/db`) también se encontrarán unos scripts que permitirán la regeneración de la base de datos (el suministrado con este enunciado para crear las tablas) 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 por casos de uso o escenarios) 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 como tal.

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: 7 de junio de 2021

convocatoria extraordinaria: 28 de junio de 2021

Será obligatorio realizar una presentación y defensa de la práctica por parte de todo el equipo

4.3. Criterios de Evaluación

La evaluación de la práctica tiene un peso total del 50 % de la asignatura.

Respecto del proyecto entregado 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;

(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 montaje PC? - 0,1/10
  - ¿Se diseñan adecuadamente las dependencias necesarias para el CU Procesar pedido PCs? - 0,1/10
  - ¿Se diseñan adecuadamente las dependencias necesarias para el CU Buscar espacios disponibles para PCs? - 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 con la Realización en Diseño de los CU-
  - ¿Se diseña adecuadamente la realización del CU Identificarse (incluye mecanismo de persistencia)? - 1/10
  - ¿Se diseña adecuadamente la realización del CU Registrar montaje PC (incluye mecanismo de persistencia)?- 1,5/10
  - ¿Se diseña adecuadamente la realización del CU Procesar pedido PCs (incluye mecanismo de persistencia)? - 1,5/10
  - ¿Se diseña adecuadamente la realización del CU Buscar espacios disponibles para PCs)? - 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 forma de uso de Pivotal Tracker no será evaluable, pero el uso será obligatorio. En caso de no utilizarse, la práctica no podrá ser considerada entregada ya que las *releases* parciales y la *release* final se realizan por esa vía.

Las releases parciales no son evaluables. Solamente marcan el ritmo del proyecto.

El uso de rocket no será obligatorio, aunque es recomendable. Se bonificará la participación de los estudiantes en el canal de la asignatura `https://rocket.inf.uva.es/group/ds20-21`, tanto en relación con la parte de teoría como con la parte práctica. Los estudiantes realizarán preguntas, podrán aportar contribuciones a las preguntas de los compañeros. Tanto los compañeros como las profesoras podrán "premiar" la participación con una reacción. La cantidad y calidad de la participación en el canal podrá ser bonificada con un extra de hasta el 0,5 (el 5 %) de la nota de la asignatura. Si bien la nota de la asignatura no podrá resultar mayor de 10.

Adicionalmente es importante el uso de este canal para notificar de algún error u omisión detectado en el enunciado, los modelos, el script sql para la creación de la BD, etc..

La calidad y la seguridad del código de la aplicación será analizada mediante SonarQube como se explica en el aula virtual (`https://aulas.inf.uva.es/mod/page/view.php?id=27169`).

Los proyectos con buenos indicadores de calidad y seguridad, detectados por esta herramienta, podrán optar a un premio.

El premio a repartir son puntos en la asignatura. Se otorgará 0,9, 0,6 y 0,3 como premio al primero, segundo, y tercer lugar en el concurso, respectivamente. No habrá empates en ninguna de las tres posiciones del concurso. Al sumar los premios obtenidos en la nota de los ganadores, no podrá sobrepasarse los 10 puntos como nota global de la asignatura.

4.3.1. Reglas del Concurso

Se elaborará un ranking entre los equipos participantes.
Para entrar en el ranking se requiere:

- pasar el QualityGate DS definido en el servidor sonarqube (marca verde Passed por el contrario de marca roja Failed). Este QualityGate se puede consultar en: `https://sonarqube.inf.uva.es/quality_gates/show/2`.
- haber implementado toda la funcionalidad especificada en los casos de uso de este enunciado.

Una vez dentro del ranking se establecerá un orden inverso (de mayor a menor puntuación, es decir, menos puntos mejor posición en el ranking), asignando una puntuación a cada equipo: $RDT + DD + VRE + BRE$
Donde RTD es la ratio de deuda técnica acumulada, DD es la densidad de duplicados, VRE es la tasa de esfuerzo para remediar las vulnerabilidades y BRE es la tasa de esfuerzo para remediar bugs potenciales.
Los empates se resolverán aplicando los siguientes criterios en este orden:

1. el que tenga menor ratio de deuda técnica,
2. el que tenga menor severidad de los code smells, es decir, el que tenga menor la ratio blocking+major)/violations
3. el que tenga menor porcentaje de código duplicado,
4. el que tenga menos bugs potenciales
5. el que tenga menos vulnerabilidades
6. el que tenga menor complejidad ciclomática (Cyclomatic complexity)
7. el que tenga menor complejidad cognitiva (Cognitive complexity)
8. finalmente el que tenga mayor densidad de comentarios. Los comentarios deberán ser todos de tipo javadoc. Si bien los comentarios javadoc no son evaluables en esta práctica, todos sabemos lo importante que son para la mantenibilidad (principio de autodocumentación). Al no permitir otro tipo de comentarios en el código se aplica el principio: Si tienes necesidad de escribir un comentario escribe una función.

Las clases que representan las Vistas (en el patrón MVC) no participan en el concurso. Estas clases, además de tener una gran parte de código automáticamente generado, tienen una herencia de nivel 5 que no podemos quitarnos porque forma parte Swing. En el aula virtual se explica cómo excluirlas del análisis automatizado.
Debido a las reglas definidas, el resultado del concurso deberá ser refrendado por una revisión manual de las profesoras. Se ejecutarán de nuevo los análisis de todos los proyectos para garantizar que se han analizado en las mismas condiciones. Se revisará la implementación para asegurar que se implementa correctamente toda la funcionalidad especificada (incluidas alternativas y excepciones).

4.3.2. Hitos en la realización del proyecto

A continuación se indican los hitos que se han definido para el seguimiento y realización de este proyecto.

- 18/4 - Release 1. Comienza el concurso con el código implementado para el Caso de Uso Identificarse.
- 02/5 - Release 2
- 16/5 - Release 3

- 30/5 - Release 4. Se cierra el concurso. Se entiende que la implementación del código ha finalizado.
- 07/06 - Entrega final de la práctica (la formal y evaluable): día del examen de la asignatura.
- en la semana del 14 al 18 de junio: presentación y defensa de la práctica con las profesoras.

En convocatoria extraordinaria se podrá entregar la práctica el día del examen que es el 28 de junio. Los días 5 y 6 de julio se podrá escoger un día y hora para una defensa de la práctica con las profesoras.