

Práctica de Diseño de Software

Grado en Ingeniería Informática
Mención Ingeniería del Software
Universidad de Valladolid

curso 2023-2024.
02/04/2024.
versión 1.0

Índice general

1. Supuesto práctico. Requisitos	2
1.1. Fragmento del documento de especificación de requisitos	4
1.1.1. Requisitos funcionales	5
1.1.2. Reglas de negocio	6
1.1.3. Requisitos no funcionales y restricciones	7
1.1.4. Requisitos de información	7
1.1.5. Restricciones adicionales	7
1.2. Casos de uso: especificación (parcial) de escenarios	8
1.2.1. Identificarse	8
1.2.2. Consultar emergencias atendidas	10
1.2.3. Modificar operador en turno	11
1.2.4. Atender llamada	12
2. Documentación parcial de Análisis	14
2.1. Vista parcial del diagrama de Casos de Uso	14
2.2. Modelo del dominio	16
2.3. Restricciones a tener en cuenta en la implementación	20
2.4. Análisis arquitectónico	20
3. Diseño de Datos	21
3.1. Diseño lógico (datamodel style)	21
3.2. Scripts de creación de la base de datos	24
4. Normas de entrega y criterios de evaluación	25
4.1. Indicaciones para la entrega	25
4.2. Estructura y contenidos de la entrega	25
4.3. Criterios de Evaluación	26
4.3.1. Reglas del Concurso	28
4.3.2. Hitos en la realización del proyecto	29

Capítulo 1

Supuesto práctico. Requisitos

Este supuesto está basado en una modificación del supuesto preparado para la práctica de la asignatura Modelado de Sistemas Software en el primer cuatrimestre de este curso 2023-2024.

El objetivo del supuesto es la elaboración de un sistema informático para la empresa ACMEur que se ocupa de la atención de emergencias sufridas por los asegurados de ACME.

La empresa dispone de una central desde la que trabajan el gerente y varios operadores con formación médica básica que gestionan las llamadas que se reciben de los asegurados, proporcionando consejos básicos y decidiendo si es preciso activar un operativo de emergencia o incluso, en casos de gravedad extrema, redirigir la llamada a los servicios públicos del 112.

La empresa utiliza varios consultorios privados como base para sus operativos, dotados de un vehículo de emergencias y formados por un conductor con formación básica como técnico de emergencias y un médico. Ambos disponen de un dispositivo ANDROID propiedad de la empresa.

Durante la jornada, los operadores atienden las llamadas de emergencia de los asegurados. Al recibir una llamada primero deben pedir al comunicante su nombre, para dirigirse a él, y el número de póliza que cubre al paciente para confirmar sus datos básicos (Apellidos, Nombre, sexo y fecha de nacimiento) con el sistema de la empresa. Si no consta un seguro de ACME para el paciente, si el seguro no está en vigor o si los datos del paciente no son coherentes con los reflejados en el sistema, se informa del problema al comunicante y se redirige la llamada al sistema del 112.

Las llamadas del tipo anterior se consideran «infundadas» y de ellas se anota el número de teléfono (que el operador ve en el terminal telefónico), el nombre del comunicante, la hora de inicio y la hora de finalización.

Si el paciente al que se refiere una llamada es asegurado de ACME, el operador pide una descripción de la emergencia al comunicante y toma nota de la misma en el sistema. Si se trata de una situación crítica, el operador transfiere la llamada al sistema del 112, que dispone de recursos más apropiados para estas situaciones.

Las llamadas del tipo anterior se consideran «críticas» y de ellas el sistema debe almacenar el número de teléfono desde el que se realizó la llamada, el nombre del comunicante, los datos del paciente, la hora de inicio y fin de la llamada y la descripción de la emergencia.

En el caso de que no se cumplan las condiciones anteriores, el operador proporciona una serie de consejos al comunicante, anotándolos en el sistema junto con el resultado de cada uno de ellos. Si con estos consejos la emergencia se resuelve, la llamada se considera «leve».

Sobre las llamadas del tipo anterior el sistema deberá conservar los datos del asegurado, el nombre del comunicante, el número de teléfono desde el que se realizó la llamada y las recomendaciones realizadas por el operador junto con el resultado de las mismas. También se almacena la hora de inicio y fin de la llamada.

Si la descripción proporcionada por el comunicante o el resultado de alguna de las recomendaciones lo requiere, el operador puede activar alguno de los operativos disponibles en ese momento. Para ello el operador solicita la dirección al comunicante. El operador se mantiene en línea hasta que el médico del operativo movilizado indica al sistema que ha llegado al punto, y se hace cargo del paciente.

Un operativo puede estar «disponible», «activado» o «bloqueado». El conductor es el responsable de indicar al principio de su turno que el operativo está disponible. Utiliza para ello la aplicación de su móvil. También utiliza la aplicación para solicitar su bloqueo al final de su turno, o en cualquier otra situación que le impida aceptar una activación. Sin embargo no se puede bloquear un operativo si no existe otro operativo disponible para esa base o si no existen 5 operativos disponibles en el sistema.

El conductor es responsable de aceptar la activación del equipo. La señal de activación se recibe como una notificación en el móvil del conductor que indica el identificador único proporcionado por el sistema a cada llamada, junto con la dirección de la misma. El conductor utiliza la aplicación para aceptar la activación y comunica el identificador de la llamada al médico, para que éste pueda consultar la información disponible sobre la misma en el sistema durante el trayecto.

Es el operador el que elige entre los equipos aquél que será activado en caso de necesidad. No obstante, si el sistema no recibe la aceptación en menos de un minuto, informa de ello al operador y le obliga a activar otra unidad disponible.

Cuando el operador solicita activar una unidad, el sistema debe proporcionarle la lista de las unidades disponibles junto con su ubicación, obtenida del GPS del que están dotadas todas ellas.

Cuando el médico se hace responsable del paciente lo indica desde su móvil. A partir de este momento atiende al paciente, aplicando una serie de «procedimientos» de los que debe tomar nota en el sistema. En algunos casos el médico puede ir tomando nota de los procedimientos en tiempo real en la aplicación, pero no es extraño que por motivos de urgencia deba confiar en su memoria y anotarlos más adelante. A través de su móvil podría grabar su propia voz expresando en voz alta los procedimientos que realiza, como ayuda para que se anote posteriormente en el sistema.

Los procedimientos médicos están todos descritos en la literatura médica, de modo que el médico sólo anota de cuál se trata, la hora de inicio, y el resultado que observa. Cada procedimiento médico tiene un nombre identificativo, una descripción larga así como descripción de posibles variantes. Es posible también indicar relaciones de unos procedimientos con otros tipo “en general cuando se aplica este procedimiento suele aplicarse este otro”, o estos otros, “este procedimiento es incompatible con este otro”, o estos otros.

El médico puede optar por trasladar al paciente a un hospital cercano. En ese caso debe anotar la hora en que se inicia el traslado y el hospital al que se realiza. Durante el trayecto es probable que el médico tenga que seguir aplicando procedimientos al paciente.

El médico debe anotar en el sistema la hora en que deja de ser responsable del paciente, ya sea porque ha llegado al hospital o porque su situación ha mejorado lo suficiente como para no requerir atención urgente. Esa hora es la hora de cierre de la llamada.

Cuando el operativo regresa a la base, el conductor es responsable de realizar las labores de limpieza necesarias y reponer el material utilizado antes de indicar en la aplicación que el operativo vuelve a estar disponible.

El gerente administrará los datos de los empleados. Puede añadir empleados o modificar sus datos. De los empleados en general necesita conocer nombre, DNI, teléfono, dirección, año en el que entró en la empresa, puesto de trabajo, ...)

De los médicos necesita conocer su número de colegiado. De los conductores el límite de vigencia de su permiso de conducción, que varía cuando lo renuevan.

De todos ellos, el gerente mantiene actualizada su disponibilidad como «disponible» o «de baja» (cuando no pueden trabajar por motivos, ya sea médicos o de otra índole). También debe tener constancia sobre si los empleados están de «vacaciones». En cuanto a la vinculación la empresa, el empleado puede ser vinculado con normalidad pero si concurren las condiciones, se debe registrar si se encuentra en «ERTE». En todos los casos es necesario conocer la fecha de inicio en la situación y, cuando se conozca, la fecha en la que finaliza dicha situación. Cuando un empleado abandona la empresa el gerente lo da de “baja” en el sistema, aunque eso no debe borrar su información, porque es importante saber qué operador, conductor o médico ha estado implicado en cada llamada que ha sido atendida por la empresa.

El gerente es responsable de organizar los turnos de trabajo de los empleados.

Los operadores trabajan en turnos de ocho horas, que empiezan a las 7, las 15 y las 23 horas de cada día. El turno de noche, empieza un día y termina el día siguiente, pero la fecha considerada para el turno es la de su final.

Los médicos y conductores hacen turnos de 12 horas, que empiezan a las 7 y a las 19 horas. Al igual que en el caso de los conductores, se considera como fecha del turno la de su final.

El gerente debe organizar los turnos con, al menos, 15 días de antelación, y lo hace por semanas. El gerente suele organizar los turnos de médicos y conductores, en el mismo momento, pero los turnos de los operadores suele organizarlos en un momento diferente. En el caso de médicos y conductores, tras un turno de 12 horas es obligatorio un descanso de, al menos, 24 horas.

Un operativo está formado entonces por un médico y un conductor asignados a una base y en un periodo de tiempo concreto. En la semana anterior el gerente tiene que asignar una unidad móvil en servicio a ese operativo.

Las unidades móviles están identificadas por su matrícula. Al darlas de alta, el gerente almacena la marca y modelo de las mismas. También el gerente indica si la unidad móvil está en el taller y cuando vuelve a estar en servicio normal. En el momento del alta se considera que está en servicio normal. Cuando son dadas de baja por el gerente es importante no perder la información de en qué llamadas han sido utilizadas, por motivos legales.

No se espera que las bases utilizadas por la empresa cambien, de modo que se considera razonable que en esas circunstancias se deba recurrir al equipo de desarrollo para que adapte el software de la empresa.

De cada póliza se dispone de los datos del tomador (dni, número de cuenta, apellidos y nombre) y de todos los asegurados (dni, apellidos, nombre y fecha de nacimiento) además de su número de póliza y su periodo de vigencia (fecha de inicio y fin de la misma). Pero el gerente no tiene entre sus funciones actualizar esa información.

En las oficinas centrales de la compañía se ocupan de la gestión de la flota de vehículos, de la gestión de los puntos de entrega-recogida, de la gestión de los empleados, de las redes de talleres asociados en cada localidad donde tiene la empresa puntos de entrega-recogida y de las aseguradoras con las que trabaja la compañía, así como de la facturación asociada a los talleres y aseguradoras. Esta parte queda excluida de este desarrollo.

Se comenzó con el desarrollo de un proyecto siguiendo una adaptación del Proceso Unificado de Desarrollo. En sus primeras fases (Inicio y Elaboración) ya se ha realizado la elicitación y especificación de requisitos, el análisis de estos requisitos, en particular el modelo de casos de uso y el modelado del dominio. A partir de los artefactos obtenidos de las tareas ya realizadas, vuestro equipo ha sido subcontratado para diseñar la arquitectura general del sistema, así como el diseño e implementación de una pequeña parte de éste, teniendo en cuenta la arquitectura general y la futura integración del resto de funcionalidad del sistema completo.

Podría decirse que sois un equipo en pruebas. Probablemente, fruto de la satisfacción de la parte contratante con la ejecución de vuestro subproyecto está en juego una futura contratación del diseño e implementación de una parte mucho mayor del resto del sistema.

La parte contratante os ha pasado el documento de especificación de requisitos del que se expone a continuación un fragmento.

1.1. Fragmento del documento de especificación de requisitos

Identificación de algunos roles de usuarios del sistema:

- Empleado de la central de emergencias
- Gerente de la central de emergencias
- Operador, uno de los empleados de la central de emergencias, que atiende las llamadas de emergencia

- Médico, uno de los empleados de la central de emergencias, que forma parte de los operativos
- Conductor, uno de los empleados de la central de emergencias, que forma parte de los operativos
- ... Se omiten otros roles por brevedad

1.1.1. Requisitos funcionales

RF-1 El sistema deberá permitir a los empleados identificarse.

RF-2 El sistema deberá permitir a los empleados consultar sus datos en la empresa.

RF-3 El sistema deberá permitir a los empleados solicitar vacaciones

RF-4 El sistema deberá permitir al gerente de la central de emergencias organizar los turnos, en particular:

RF-4a deberá permitir organizar los turnos de operadores.

RF-4b deberá permitir organizar los turnos de los operativos.

RF-5 El sistema deberá permitir al gerente de la central de emergencias consultar los turnos ya organizados.

RF-6 El sistema deberá permitir a un operador de la central de emergencias cuando atiende una llamada registrar la información sobre la misma, en particular el número de teléfono de origen, la fecha y hora de inicio, así como la fecha y hora de fin.

RF-7 El sistema deberá permitir a un operador de la central de emergencias registrar si la llamada que atiende es infundada cuando no es de un asegurado de la empresa, y en ese caso registrar el nombre del comunicante.

RF-8 El sistema deberá permitir a un operador de la central de emergencias registrar si la llamada que atiende es de un asegurado de la empresa y, en ese caso, registrar a qué asegurado corresponde.

RF-9 El sistema deberá permitir a un operador de la central de emergencias registrar si una llamada de un asegurado es crítica, y en ese caso registrar el nombre del comunicante.

RF-10 El sistema deberá permitir a un operador de la central de emergencias registrar si una llamada de un asegurado no es crítica y, por la descripción de la emergencia, registrar que no es leve y, en ese caso, activar un operativo que acuda al lugar de la llamada.

RF-11 El sistema deberá permitir a un operador de la central de emergencias registrar si una llamada de un asegurado no es crítica y, por la descripción de la emergencia, considerar si es leve y, en ese caso, permitirá registrar los consejos que da al comunicante, la descripción, resultado y si soluciona la razón de la llamada.

RF-12 El sistema deberá permitir a un operador de la central de emergencias registrar después de haber registrado varios consejos que no dan resultado, decidir y registrar que la llamada no es leve y, en ese caso, activar un operativo que acuda al lugar de la llamada.

RF-13 El sistema deberá permitir a un médico que acude en un operativo a la atención de una llamada de emergencia, registrar el momento en el que se hace cargo del paciente.

RF-14 El sistema deberá permitir a un médico que acude en un operativo a la atención de una llamada de emergencia, registrar los procedimientos que aplica al paciente, una vez se ha hecho cargo. En particular, permitirá registrar por cada procedimiento el momento en el que lo aplica, la descripción del procedimiento y el resultado observado.

RF-15 El sistema deberá permitir al médico que ha acudido con un operativo a resolver una emergencia, después de haber finalizado la misma, completar/modificar la información sobre los procedimientos que ha aplicado, si no ha podido anotar bien todo durante la solución de la emergencia.

RF-16 El sistema deberá permitir a los médicos y conductores las emergencias de las que se han hecho cargo en un operativo.

...

Etc., etc., etc.

1.1.2. Reglas de negocio

RN-1 Los empleados que no se encuentran en activo [RN-2] no podrán identificarse en el sistema.

RN-2 Un empleado está en activo cuando se encuentra “VinculadoConNormalidad” en cuanto a Vinculación actual (VinculadoConNormalidad, En ERTE, FueraDeLaEmpresa) y cuando se encuentra “Disponibile” en cuanto a Disponibilidad actual (Disponibile, DeVacaciones, DeBaja temporal).

RN-3 Las llamadas críticas se derivan al 112.

RN-4 Cada operativo está formado por un médico y un conductor que se encuentran asignados al mismo turno.

RN-5 Ni el médico y ni el conductor que forman un operativo pueden formar parte de otro operativo en el mismo turno.

RN-6 Los turnos de operadores, médicos y conductores se organizan con al menos 15 días de antelación.

RN-7 Los turnos de los operadores son de 8h y comienzan a las 7, a las 15 y a las 23 horas. Se considera como fecha de turno la de su final.

RN-8 Los turnos de los médicos y conductores son de 12 y comienzan a las 7 y a las 19 horas. Se considera como fecha de turno la de su final.

RN-9 Para los médicos y conductores, deben pasar al menos 24h entre un turno y el siguiente.

RN-10 Para poder ser asignado a un turno operadores, médicos y conductores no estarán de 'De baja' en el momento de la asignación a turno, ni 'De vacaciones' en la fecha del turno.

RN-11 Todos los turnos para la misma semana se organizan el mismo día.

RN-12 Las semanas comienzan en lunes.

RN-13 Los operativos se crean una semana antes del turno asignando el médico y el conductor concretos que estarán juntos en el operativo, y la unidad móvil (el vehículo). El operativo se crea 'bloqueado'.

RN-14 Para poder ser asignado a un operativo, el vehículo tiene que estar 'En Servicio' en el momento de la asignación

RN-15 Para poder ser asignados a un operativo el médico y el conductor no pueden estar 'De baja' ni 'de vacaciones' en el momento de la asignación.

RN-16 El médico que se hace cargo de un paciente, cuando se requiere la intervención de un operativo, no aplica procedimientos incompatibles.

RN-17 Las fechas y horas relativas a las llamadas no críticas, que requieren activar un operativo, se encuentran en una relación de inclusión correcta, es decir, todo ocurre después de la fecha y hora de inicio de la llamada. La fecha y hora de fin de la llamada coincide con la fecha y hora en la que se hace cargo el médico (establecer las condiciones implicando también la fecha y hora de activación, la fecha y hora de decisión de traslado al hospital (si se necesita), las fechas y horas de los procedimientos aplicados, si alguno, y la fecha y hora de cierre de la activación del dispositivo.

...

Etc., etc., etc.

1.1.3. Requisitos no funcionales y restricciones

Requisitos no funcionales

RNF-1 El sistema deberá trabajar con una base de datos relacional centralizada en un servidor dedicado.

Se omiten otros requisitos no funcionales en este trabajo para no condicionar al equipo (dado el contexto académico).

Restricciones

R-1 El sistema deberá tener en cuenta el Reglamento General de Protección de Datos (Reglamento UE 2016/679 del Parlamento Europeo y del Consejo: <https://www.aepd.es/es/informes-y-resoluciones/normativa-y-circulares>)

R-2 El sistema deberá tener en cuenta la Guía laboral del Ministerio de Trabajo y Economía Social, en particular “Entre el final de una jornada y el comienzo de la siguiente mediarán, como mínimo, doce horas.” https://www.mites.gob.es/es/Guia/texto/guia_6/contenidos/guia_6_14_1.htm#:~:text=Entre%20el%20final%20de%20una,doce%20horas.

Se omiten otras restricciones para no condicionar al equipo (dado el contexto académico).

1.1.4. Requisitos de información

Los requisitos de información se omiten por brevedad debido a que ya quedan recogidos en el modelado del dominio que se muestra en el Capítulo 2.

1.1.5. Restricciones adicionales

Se ha decidido separar el sistema en al menos tres subsistemas, el que será desarrollado para el uso de los empleados de la central de emergencias, el que será desarrollado para el uso de los empleados de las oficinas centrales de la empresa y el que será desarrollado para ser utilizado por los médicos y conductores cuando forman parte de un operativo en servicio. Estos subsistemas podrían dividirse en más subsistemas si procede y así se decidiera.

El subsistema desarrollado para ser utilizado por los médicos y conductores cuando forman parte de un operativo en servicio será una aplicación móvil para ANDROID y no será abordada en esta asignatura.

El subsistema desarrollado para el uso de los empleados de las oficinas centrales de la empresa ya ha sido desarrollado previamente y se encuentra en funcionamiento.

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

En teoría la BD debe ser 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 para facilitar la transición si se decide 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. Los requisitos de interacción (especificación textual de los casos de uso y sus escenarios) correspondientes a cuatro casos de uso se incluyen en este documento en la siguiente sección (Sec. 1.2).

A partir de los requisitos, en la fase de análisis se ha realizado el Modelado del Dominio y la realización de casos de uso en análisis (se omite en este enunciado). En el Capítulo 2 se aporta una documentación parcial de análisis que incluye: un fragmento del diagrama de casos de uso y el modelo del dominio.

En el Capítulo 3 se aporta 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 los diagramas del modelo de dominio.

1.2. Casos de uso: especificación (parcial) de escenarios

Se especifican parcialmente (solamente 4 casos de uso) los escenarios de interacción con el sistema.

La especificación de casos de uso que aquí se describe no se corresponde exactamente con lo esperado en 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. Otras simplificaciones realizadas tienen que ver con no considerar casos que puedan conducir a tener que cobrar al cliente porque no vamos implementar nada relacionado con facturación, cobros y pagos.

1.2.1. Identificarse

Actor: Empleado

Caso de Uso: Identificarse

Precondición: El actor no se encuentra identificado en el sistema

Secuencia principal:

1. El actor Empleado introduce nif y contraseña.
2. El sistema comprueba que dicho par nif-contraseña corresponden con un empleado en activo, muestra las opciones correspondientes a su rol en el sistema y el caso de uso finaliza.

Alternativas y excepciones:

- (2 a) Si no existe empleado con nif 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 nif 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 (según [RN-2], activo es: vinculado con normalidad y disponible), el sistema muestra un mensaje de error y a continuación el caso de uso queda sin efecto.

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

1.2.2. Consultar emergencias atendidas

Actor: Personal de operativo

Caso de Uso: Consultar emergencias atendidas

Precondición: El actor (generalización) personal de operativo se encuentra identificado en el sistema

Secuencia principal:

1. El actor solicita ver las emergencias que ha atendido su operativo en un turno, indicando la fecha del turno que desea consultar.
2. El sistema comprueba que el actor tuvo turno de operativo en la fecha dada, muestra las emergencias que activaron el operativo del que formaba parte en dicho turno, en particular fecha y hora de cada activación del operativo.
3. El actor selecciona una activación para ver detalles.
4. El sistema le muestra detalles de la emergencia correspondiente a dicha activación, teniendo en cuenta el RGDP (R-1), en particular muestra la fecha y hora de la activación, descripción de la emergencia, nombre de la vía a donde se acudió, fecha y hora en la que se hizo cargo del paciente y si se decidió su traslado al hospital, y el caso de uso vuelve al paso 3.

Alternativas y excepciones:

- (1a) Si el actor cancela, el caso de uso queda sin efecto.
- (2a) Si el sistema encuentra que el actor no tuvo turno de operativo en la fecha solicitada muestra un mensaje informativo y el caso de uso finaliza.
- (3a) Si el actor cancela, el caso de uso finaliza.

Postcondición: Escenario de éxito: El sistema muestra los turnos organizados para la fecha indicada.

Indicaciones de usabilidad: el actor podrá elegir filtrar y ver solo los turnos de los operadores o médicos o conductores.

1.2.3. Modificar operador en turno

Actor: Gerente de central de emergencias

Caso de Uso: Modificar operador en turno

Precondición: El actor Gerente de la central de emergencias se encuentra identificado en el sistema

Secuencia principal:

1. El actor solicita modificar un turno de operador, indicando la fecha del turno que desea modificar.
2. El sistema comprueba que hay turno de operador organizado para esa fecha, muestra la lista de operadores en el turno, en particular nombre, apellidos, NIF y teléfono de cada operador.
3. El actor selecciona un operador y solicita modificarlo por otro operador.
4. El sistema le muestra la lista de operadores que se conoce que están en activo ese día (no están de baja a fecha de hoy, ni de vacaciones a fecha del turno), no tienen turno organizado para ese día y hayan pasado al menos 12h desde su turno anterior (RN-7, RN-10, R-1), en particular muestra nombre, apellidos, NIF y teléfono de cada operador.
5. El actor selecciona un operador de la lista.
6. El sistema solicita confirmación del cambio propuesto.
7. El actor confirma.
8. El sistema actualiza el cambio y el caso de uso finaliza.

Alternativas y excepciones:

- (1a), (3a), (5a), (7a) Si el actor cancela, el caso de uso queda sin efecto.
- (2a) Si el sistema encuentra que no hay turnos de operador organizados para la fecha solicitada, muestra un mensaje informativo y el caso de uso queda sin efecto.
- (4a) Si el sistema comprueba que no hay operadores que cumplan las condiciones, dadas por RN-7, RN-10, R-1, muestra un mensaje informativo y el caso de uso queda sin efecto.
- (7a) Si el actor no confirma, el caso de uso vuelve al paso 2.

Postcondición: Escenario de éxito: El primer operador indicado en el caso de uso ya no está asignado al turno en la fecha dada. El segundo operador indicado en el caso de uso está asignado al turno de la fecha dada. El turno mantiene la misma cantidad de operadores asignados que tenía al inicio del caso de uso.

1.2.4. Atender llamada

Actor: Operador

Caso de Uso: Atender llamada

Precondición: El actor Operador se encuentra identificado en el sistema

Secuencia principal:

1. El actor solicita registrar una llamada entrante.
2. El sistema comprueba que el actor se encuentra de turno, registra la fecha y hora de inicio de la llamada, solicita número de teléfono de origen, el nombre del comunicante, el número de póliza que cubre al paciente, nombre, apellidos, sexo y fecha de nacimiento del paciente.
3. El actor introduce los datos solicitados.
4. El sistema comprueba que hay una póliza en vigor correspondiente al número de póliza y uno de los asegurados coincide con los datos del paciente comunicados, solicita descripción de la emergencia y si se considera crítica.
5. El actor introduce los datos solicitados.
6. El sistema comprueba que el actor ha indicado que la emergencia no se considera crítica, crea una nueva llamada no crítica con la fecha y hora de inicio, número de teléfono de origen, la descripción de la emergencia, los datos del asegurado que es el paciente de la emergencia, y el nombre del comunicante (también puede registrarse “el asegurado” como nombre del comunicante si es el caso) y solicita información sobre consejos dados al paciente, su resultado y si se requiere intervención de operativo.
7. El actor indica que no se requiere intervención de operativo, introduce la descripción del consejo que ha dado, el resultado observado por el comunicante y si el consejo resuelve la llamada.
8. El sistema comprueba que el consejo no resuelve la llamada, solicita la información de un nuevo consejo aportado y el caso de uso vuelve al paso 7.
9. El sistema registra la nueva llamada con todos los datos asociados y el caso de uso finaliza.

Alternativas y excepciones:

- (2a) Si el sistema comprueba que el actor no se encuentra de turno muestra un mensaje informativo y el caso de uso queda sin efecto.
- (4a) Si el sistema comprueba que no hay una póliza en vigor correspondiente al número de póliza o que los datos del paciente no son coherentes con uno de los asegurados de dicha póliza, informa al actor para que redirija la llamada al 112, registra la fecha y hora de fin de la llamada, crea una nueva llamada infundada con la fecha y hora de inicio, número de teléfono de origen, el nombre del comunicante, fecha y hora de fin de la llamada y el caso de uso continúa por el paso 9.
- (6a) Si el sistema comprueba que la llamada se considera crítica, informa al actor para que redirija la llamada al 112, registra la fecha y hora de fin de la llamada, crea una nueva llamada crítica con la fecha y hora de inicio, número de teléfono de origen, la descripción de la emergencia, el nombre del comunicante, fecha y hora de fin de la llamada y los datos del asegurado que es el paciente de la emergencia, y el caso de uso continúa por el paso 9.
- (7a) Si el actor indica que se requiere intervención de operativo, anota la llamada como no leve, ejecuta el caso de uso “Activar operativo de emergencia” y el caso de uso continúa por el paso 9.

(8a) Si el sistema comprueba que el consejo resuelve la llamada, anota la llamada como leve y el caso de uso continúa por el paso 9.

Postcondición: Escenario de éxito (en todos los casos excepto si el actor no se encuentra en turno): se ha creado una nueva llamada en el sistema del tipo necesario según haya sido el caso: infundada, crítica, o no crítica con los datos asociados.

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 generalización Personal de operativo
- el actor Gerente
- el actor Operador

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 operativo CU Consultar emergencias atendidas

Gerente CU Modificar operador en turno

Operador CU Atender llamada

Aclaraciones sobre el caso de uso “Identificarse”: El empleado se identifica con su nif 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”.

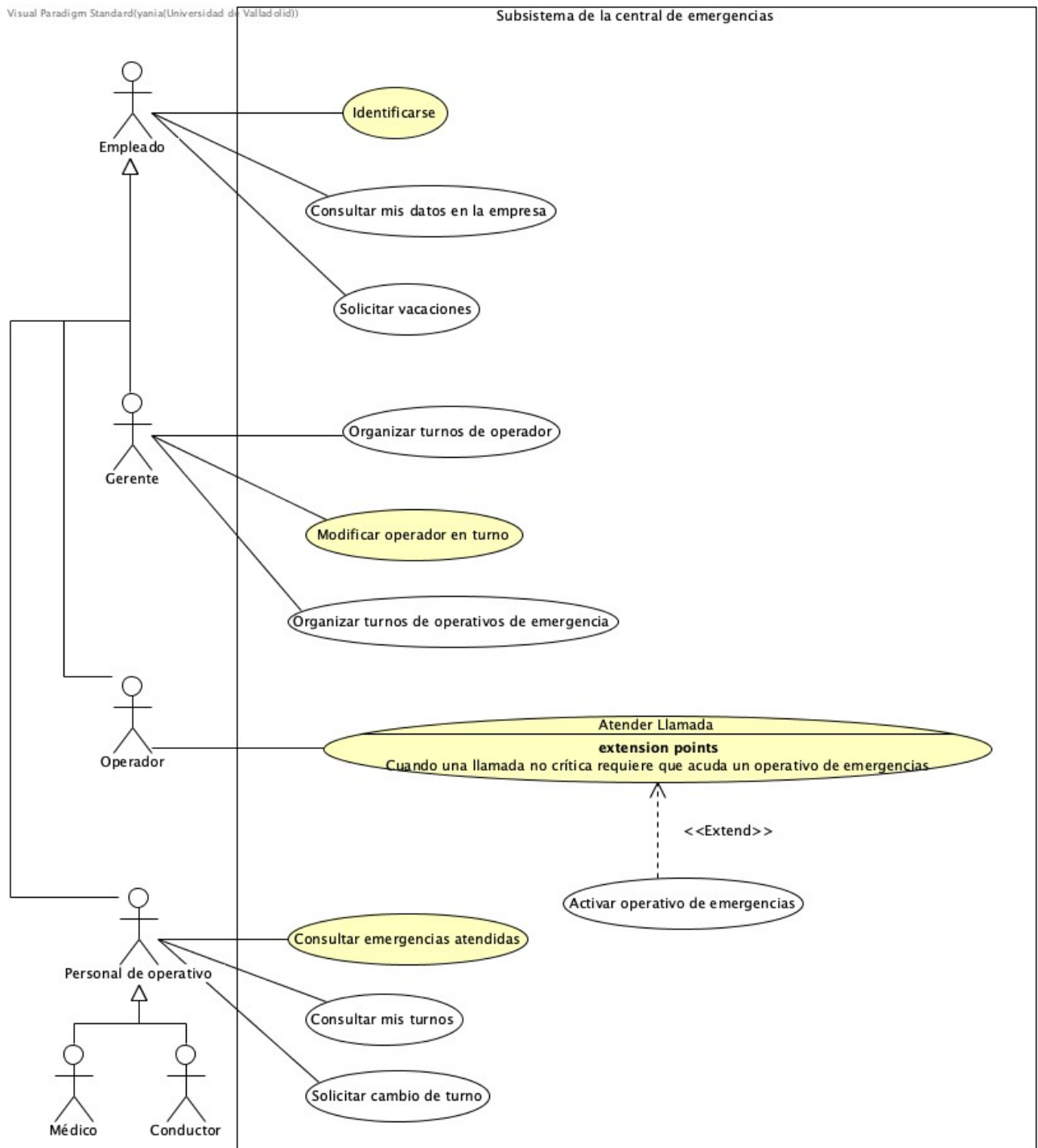
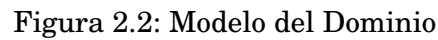


Figura 2.1: Fragmento del diagrama de casos de uso. Se muestran sólo algunos ejemplos por actor.

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 y la especificación de los detalles de algunas clases que no participan en ninguno de los casos de uso a diseñar e implementar.



A continuación se describen algunas reglas que debe cumplir la información del dominio mediante una descripción textual y especificación OCL. Se aplica la recomendación de colocar la restricción OCL en el contexto en el que más fácil resulte expresarla. Los datatype del dominio Fecha y Hora se consideran que tienen las operaciones de LocalDate y LocalTime en java.time. Una de estas operaciones en LocalDate, por tanto, en Fecha, es atTime que obtiene un LocalDateTime, combinando fecha y hora. LocalDateTime sería equivalente a un nuevo datatype Momento en el que estarían definidas las operaciones isAfter, isBefore, de la misma forma que en LocalDate (Fecha) y LocalTime (Hora). Estas restricciones sirven de ejemplo para otras que pueden analizarse al estudiar el dominio y sus requisitos.

En cada TurnoDeOperativo organizado hay asignados la misma cantidad de médicos que de conductores, que deberá garantizarse que tienen el rol de médicos y conductores en la fecha de creación del turno y en la fecha del turno.

```
context TurnoDeOperativo
  inv: medicos->size() = conductores->size()
  inv: medicos->forall(m | m.rolEnFecha(fechaCreacion).oclTypeOf(Medico)
    and m.rolEnFecha(fechaTurno).oclType(Medico))
  inv: conductores->forall(c | c.rolEnFecha(fechaCreacion).oclTypeOf(Conductor)
    and c.rolEnFecha(fechaTurno).oclType(Conductor))
```

Se añade a la clase Empleado la operación rolEnFecha(f: Fecha): Rol).

Cada operativo está formado por un médico y un conductor de entre los que están asignados al turno organizado, asociado al operativo. El médico y el conductor que forman un operativo no pueden estar en otro operativo en el mismo turno. Los turnos de operadores, médicos y conductores se organizan con al menos 15 días de antelación. Todos los turnos para la misma semana se organizan el mismo día.

```
context Operativo
  inv: turno.medicos->includes(medico)
    and turno.conductores->includes(conductor)

context TurnoDeOperativo
  inv: operativos->forall(o1, o2: Operativo |
    o1 <> o2 implies (o1.medico<>o2.medico
    and o1.conductor<>o2.conductor) )
  inv: fechaCreacion.until(fechaTurno, DAYS) >= 15
  inv: TurnoDeOperativo.allInstances()->forall(t1, t2: TurnoDeOperativo |
    ( t1 <> t2 and (
      t1.fechaTurno.get(ChronoField.ALIGNED_WEEK_OF_YEAR)=
      t2.fechaTurno.get(ChronoField.ALIGNED_WEEK_OF_YEAR) ) implies
      t1.fechaCreacion=t2.fechaCreacion)

context TurnoDeOperador
  inv: fechaCreacion.until(fechaTurno, DAYS) >= 15
  inv: TurnoDeOperador.allInstances()->forall(t1, t2: TurnoDeOperador |
    ( t1 <> t2 and (
      t1.fechaTurno.get(ChronoField.ALIGNED_WEEK_OF_YEAR)=
      t2.fechaTurno.get(ChronoField.ALIGNED_WEEK_OF_YEAR) ) implies
      t1.fechaCreacion=t2.fechaCreacion)
```

Los operativos se crean una semana antes del turno asignando el médico y el conductor concretos que estarán juntos en el operativo, y la unidad móvil (el vehículo). Para poder ser asignado a un operativo, el vehículo tiene que estar 'En Servicio' y el médico y el conductor no pueden estar 'De baja' ni 'de vacaciones'. Cada operativo se crea 'bloqueado'.

```
context Operativo::crea(m: Empleado, c: Empleado, v: Vehículo,
  t: TurnoDeOperativo, b: Consultorio): void
pre: m <> null and c <> null and v <> null and t <> null and b <> null
pre: LocalDate.now().until(t.fechaTurno, DAYS) = 7
...
pre: v.estado = EstadoDeVehiculo::enServicio
```

```

pre: m.situaciones->last().oclType() <> DeBaja and
    m.situaciones->last().oclType() <> DeVacaciones
...
post: fechaCreacion = LocalDate.now() and medico = m and conductor = c and
    vehiculo = v and turno = t and base = b
post: estado = EstadoDeOperativo::bloqueado

```

Se añade la operación crea, de creación de objetos (estereotipo create), en la clase Operativo. Una llamada no crítica es leve si con alguno de los consejos proporcionados se soluciona. Se requiere activar operativo cuando una llamada no crítica no se soluciona con los consejos proporcionados (y por lo tanto no es leve) o se marca como no leve desde un inicio por la descripción misma de la emergencia. Para activar un operativo cuando se requiere, este tiene que estar en su turno, estar 'disponible', tiene que haber otro operativo disponible en la misma base y al menos 5 operativos disponibles en el sistema.

```

context LlamadaNoCritica
    inv: esLeve = consejos->exists(c | c.soluciona)
    inv: requiereOperativo->notEmpty() implies not esLeve

context RequiereOperativo::crea(d: Dirección, nc: LlamadaNoCritica, o: Operativo): void
    pre: d <> null and nc <> null and o <> null
    pre: not nc.esLeve
    ...
    pre: o.enTurno(LocalDate.now(), LocalTime.now()) and
        o.estado = EstadoDeOperativo::disponible and
        o.base.asignados->exists(o1 | o <> o1 and
            o1.estado= EstadoDeOperativo::disponible) and
        Operativo.allInstances()->select(o2: Operativo | o <> o2 and
            o2.turno = o.turno and
            o2.estado = EstadoDeOperativo::disponible)->size() >= 5

    post: dirección = d and llamadaNoCritica = nc and operativo = o
    post: operativo.estado = EstadoDeOperativo::activado
    post: activación.fecha = LocalDate.now() and activación.hora = LocalTime.now()

```

Se añade la operación crea, de creación de objetos (estereotipo create), a la clase RequiereOperativo.

El médico que se hace cargo cuando se requiere operativo no aplica procedimientos incompatibles.

```

context SeHaceCargoMedico
    inv: procedimientos->forall(p1, p2: Procedimiento | p1 <> p2 implies
        not p1.descrito.esIncompatibleCon(p2.descrito))

context SeHaceCargoMedico::añadeProcedimientoAplicado(nuevo: Procedimiento): void
    pre: nuevo <> null and
        procedimientos->forall(p: Procedimiento | p <> nuevo and
            not p.descrito.esIncompatible(nuevo.descrito))

```

Se añade la operación añadeProcedimientoAplicado(nuevo: Procedimiento): void a la clase SeHaceCargoMedico.

Ningún empleado puede hacer dos turnos en el mismo día.

```

context Empleado
    inv: turnosMedicos.fechaTurno.intersection(turnosConductores.fechaTurno)->empty()
    inv: turnosDeOperador.fechaTurno.intersection(turnosMedicos.fechaTurno)->empty()
    inv: turnosDeOperador.fechaTurno.intersection(turnosConductores.fechaTurno)->empty()

```

Las fechas y horas relativas a las llamadas no críticas que requieren activar un operativo se encuentran en una relación de inclusión correcta, es decir, todo ocurre después de la fecha y hora de inicio de la llamada. La fecha y hora de fin de la llamada coincide con la fecha y hora en la que se hace cargo el médico (establecer las condiciones implicando también la fecha y hora de activación, la fecha y hora de decisión de traslado al hospital (si se necesita), las fechas y horas de los procedimientos aplicados, si alguno, y la fecha y hora de cierre de la activación del dispositivo.

```

context RequiereOperativo
    inv: activación.fecha.atTime(activacion.hora).isAfter(
        llamadaNoCritica.fechaInicio.atTime(llamadaNoCritica.horaInicio))
    inv: cierre.fecha.atTime(cierre.hora).isAfter(
        activación.fecha.atTime(activacion.hora))

context SeHaceCargoMedico
    inv: activación.fecha.atTime(activacion.hora).isBefore(
        fecha.atTime(hora))
    inv: activacion.requiereOperativo.llamadaNoCritica.fechaFin.atTime(
        activacion.requiereOperativo.llamadaNoCritica.horaFin).equals(
        fecha.atTime(hora))
    inv: fecha.atTime(hora).isBefore(
        activacion.requiereOperativo.cierre.fecha.atTime(
        activacion.requiereOperativo.cierre.hora))

context DecideTrasladoAHospital
    inv: fecha.atTime(hora).isAfter(seHaceCargoMedico.fecha.atTime(
        seHaceCargoMedico.hora))
    inv: fecha.atTime(hora).isBefore(
        seHaceCargoMedico.activacion.requiereOperativo.cierre.fecha.atTime(
        seHaceCargoMedico.activacion.requiereOperativo.cierre.hora))

context Procedimiento
    inv: fechaInicio.atTime(horaInicio).isAfter(
        seHaceCargoMedico.fecha.atTime(
        seHaceCargoMedico.hora))
    inv: fechaInicio.atTime(horaInicio).isBefore(
        seHaceCargoMedico.activacion.requiereOperativo.cierre.fecha.atTime(
        seHaceCargoMedico.activacion.requiereOperativo.cierre.hora))

```

2.3. Restricciones a tener en cuenta en la implementación

El sistema deberá utilizar una base de datos cuyo esquema (diseño lógico) se aporta en las páginas 22 y 23, implementada con Derby. Deberá utilizarse el script de creación de la base de datos (diseño físico) se aporta en la página 24. El subsistema, de la central de emergencias, a desarrollar será una aplicación de escritorio independiente del sistema operativo que se desarrollará en JAVA, jdk 11 utilizando Swing y JDBC para db-derby-10.14.2.0. El sistema será desarrollado en NETBEANS en la misma versión que se encuentra instalada actualmente en los laboratorios de la Escuela (actualmente Apache Netbeans 18).

Cuando se ejecuta el caso de uso “Activar operativo de emergencia” durante el caso de uso “Atender llamada” se indicará “Not implemented yet”.

2.4. Análisis arquitectónico

En el subsistema de la central de emergencias se ha decidido aplicar una arquitectura de capas con tres capas estricta y, en caso necesario una relajada de servicios útiles a todas las capas, se combina la arquitectura de tres capas con MVC. Se aplica DAO+DTO como patrón de acceso a datos. Se ha decidido que el DTO será una cadena JSON.

Capítulo 3

Diseño de Datos

3.1. Diseño lógico (datamodel style)

En las Figuras 3.1 y 3.2 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 representan asociaciones se identifican con el estereotipo «relation». Las tablas que representan valores enumerados se identifican con el estereotipo «enum» y aparecen sombreadas. La transformación al modelo relacional de las relaciones de herencia presentes en el Modelo del Dominio se ha realizado mediante referencias entre tablas. El modelado de roles temporales presentes en el modelo del dominio para empleados se ha realizado mediante tablas asociación y tablas enum.

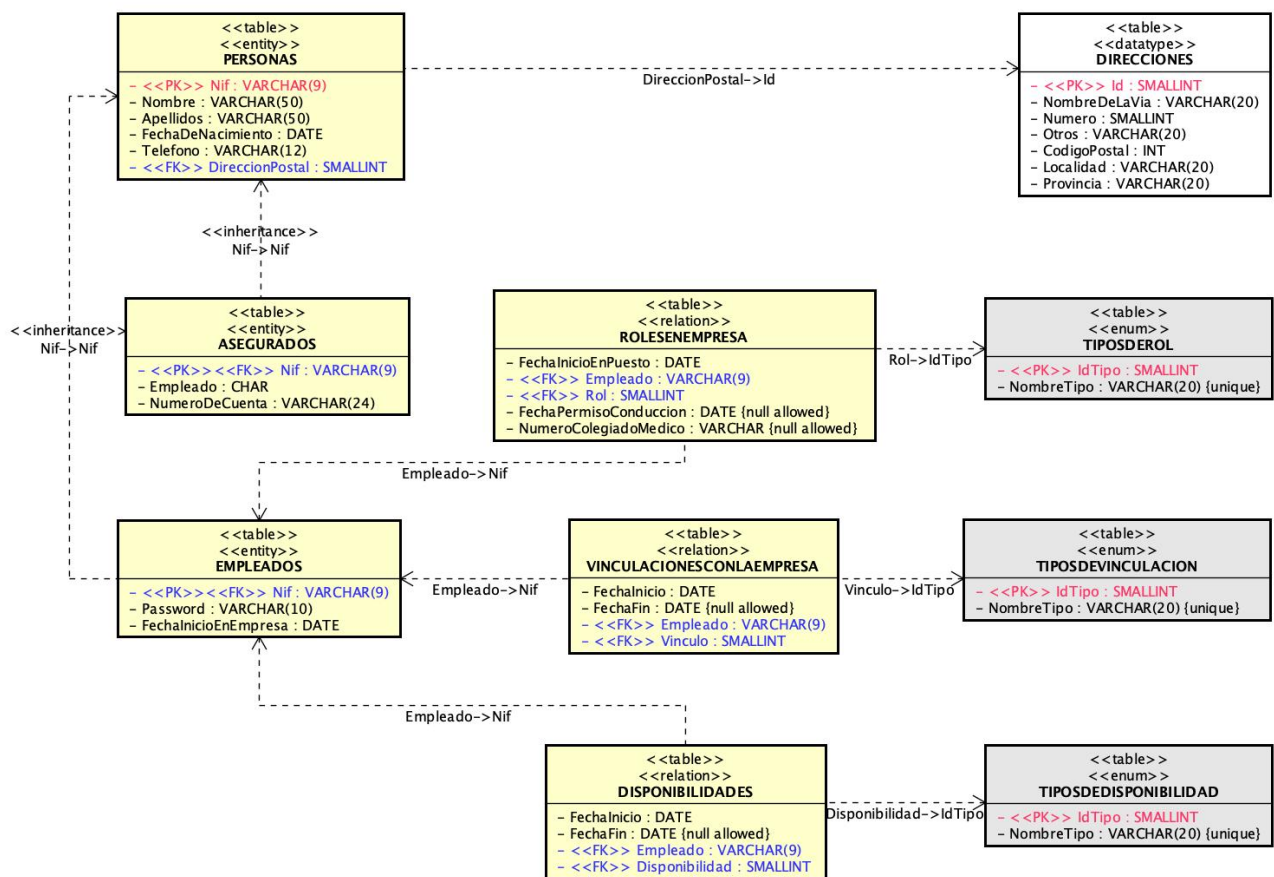


Figura 3.1: Diseño lógico de la base de datos relacional (parte relativa a empleados)

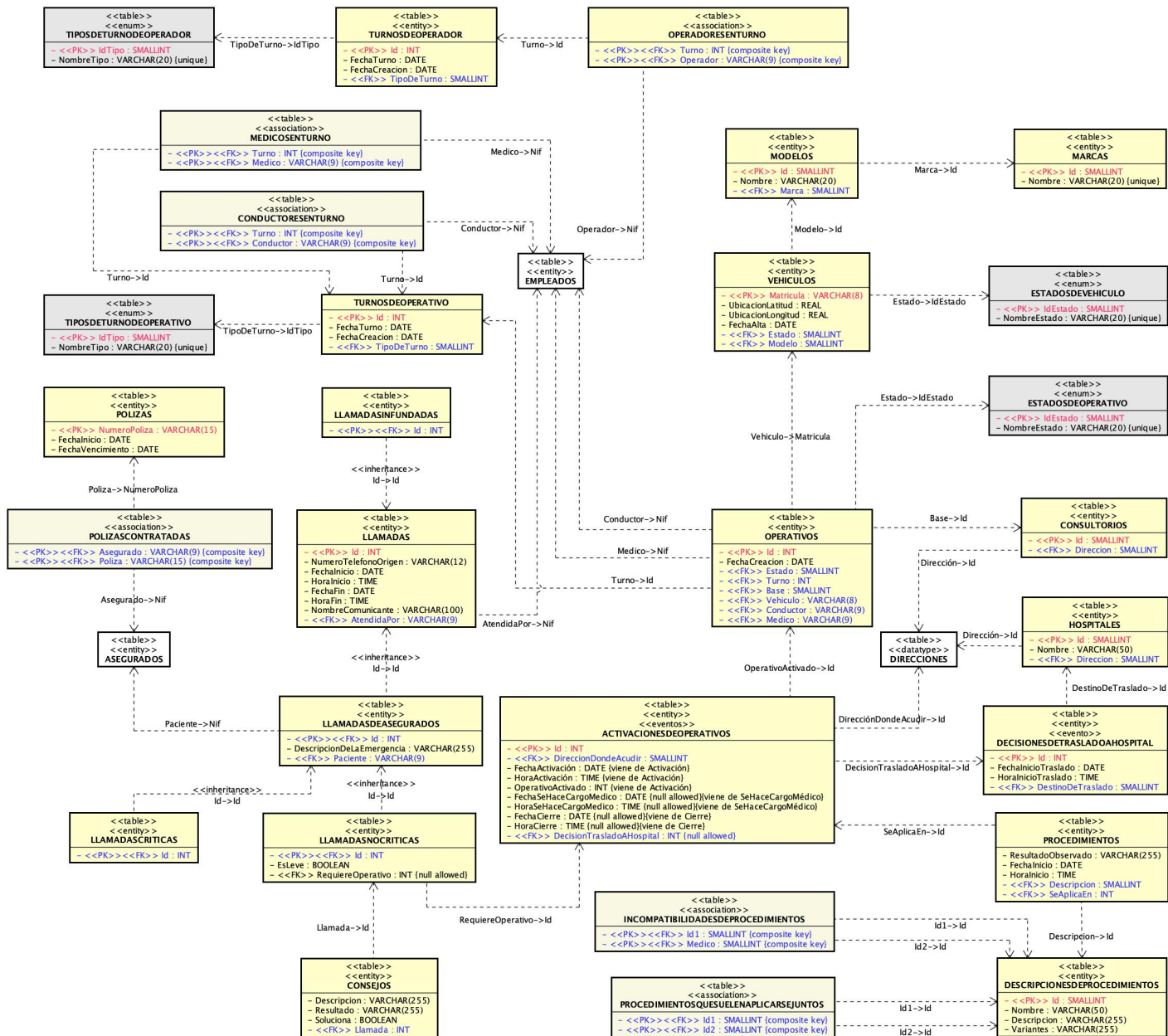


Figura 3.2: Diseño lógico de la base de datos relacional (resto)

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://campusvirtual.uva.es/pluginfile.php/5505678/mod_resource/content/12/createTables.sql.

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 estudiantes se habrán creado una cuenta en PIVOTAL TRACKER y habrán sido añadidos al proyecto correspondiente según su equipo de trabajo. El estudiante deberá estar identificado con su correo institucional (estudiantes.uva.es) y su identificación como login en los laboratorios de la Escuela.

La comunicación entre el equipo, y el equipo con los profesores, puede hacerse mediante PIVOTAL TRACKER.

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, con permisos de tipo *Reporter*, a los profesores (usuario *yania*, en todo caso, y usuarios *marga/marcorr*, según corresponda) 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 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 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. Los diagramas no pueden ser excesivamente grandes por lo que se deberán utilizar los elementos que ofrece UML para reducir el tamaño y la complejidad de los diagramas.

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. El proyecto de la implementación deberá estar gestionado con maven.

La carpeta `app/db` contendrá un archivo de propiedades llamado `config.db` 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/acmeur  
user=admin_acmeur  
password=acmeur2024
```

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.

En el control de versiones, y por tanto en los archivos de entrega, no debe residir la base de datos Derby como tal. Tampoco deberá encontrarse ninguna librería puesto que la gestión de dependencias será realizada por maven. El `.gitignore` deberá indicar target para que ningún binario que se pueda volver a generar resida en el repositorio.

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: 29 de mayo de 2024

convocatoria extraordinaria: 30 de junio de 2024

Será obligatorio realizar una presentación y defensa de la práctica por parte de todo el equipo (ver Sección 4.3.2 para hitos en el desarrollo y la defensa del proyecto).

4.3. Criterios de Evaluación

La evaluación de la práctica tiene un peso total del 45 % 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 Atender llamada? - 0,1/10
 - ¿Se diseñan adecuadamente las dependencias necesarias para el CU Modificar operador en turno? - 0,1/10
 - ¿Se diseñan adecuadamente las dependencias necesarias para el CU Consultar emergencias atendidas? - 0,1/10
- diagramas de clases de diseño detallado - 0,5/10
- diagramas de secuencia con la Realización en Diseño de los CU-
 - ¿Se diseña adecuadamente la realización del CU Identificarse? - 1/10
 - ¿Se diseña adecuadamente la realización del CU Atender llamada?- 1,5/10
 - ¿Se diseña adecuadamente la realización del CU Modificar operador en turno? - 1,5/10
 - ¿Se diseña adecuadamente la realización del CU Consultar emergencias atendidas? - 1,5/10
- diagrama de estados que modela la interfaz - 0,1/10

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/ds23-24>, 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 los profesores podrán “premiar” la participación con una reacción. La cantidad y calidad de la participación en el canal será valorada con hasta 0,5 puntos (consultar guía docente de la asignatura).

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 campus virtual (<https://campusvirtual.uva.es/course/view.php?id=44324§ion=17#tabs-tree-start>).

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 nota final de la práctica. 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.

En ningún caso la nota de la asignatura podrá resultar mayor de 10.

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.
- en las fases iniciales del concurso no se tendrá en cuenta la restricción anterior.

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 RDT 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. Todos estos indicadores se pueden consultar en Sonarqube.

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 de `javax.swing`. En el campus virtual se explica cómo excluirlas del análisis automatizado.

Regla fundamental:

- haber implementado toda la funcionalidad especificada en los casos de uso de este enunciado.

El incumplimiento de la regla fundamental será motivo de exclusión del ranking.

Debido a las reglas definidas, el resultado del concurso deberá ser refrendado por una revisión manual de los profesores. 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.

- 17/4 - Release 1. Comienza el concurso con el código implementado para el Caso de Uso Identificarse.
- 1/5 - Release 2.
- 15/5 - Release 3.
- 29/5 - Release 4. Se cierra el concurso. Entrega de la práctica en convocatoria ordinaria.
- Desde la entrega final de la práctica hasta el 14 de junio, incluido, los equipos podrán realizar las defensas, previa concertación de una fecha y hora con los profesores.

Las fechas se han definido de forma holgada. Desde el 3 de abril, considerado fecha de inicio del proyecto, hasta el 17 de abril, hay tiempo más que suficiente para haber conseguido la Release 1, dado que ya se ha trabajado en la semana 6 en prácticas en la realización en diseño del CU Identificarse, y se podrían adelantar tareas necesarias para conseguir la Release 2. Los hitos son puntos de control para el equipo.

En la defensa, el equipo presentará la demostración de la implementación desarrollada utilizando sus pruebas preparadas para ello, los diagramas de diseño realizados y la correspondencia entre el diseño presentado y el código implementado. El equipo es libre de decidir la organización, quién presenta qué parte. Asimismo, el equipo contestará a las preguntas que se les realicen. Para la defensa no se prepara una presentación con diapositivas. Se utilizará en todo momento Astah/Visual Paradigm, según sea el caso, y Netbeans. Es necesario que todos los miembros del equipo participen en la defensa.

En convocatoria extraordinaria se podrá entregar la práctica hasta el 30 de junio. Los días 1 y 2 de julio los equipos podrán realizar las defensas, previa concertación de una fecha y hora concretas con los profesores.

Los equipos son de 3 o 4 personas por lo que se deben repartir tareas en la realización de cada CU entre todos los miembros del equipo. Se recomienda fuertemente no cometer el error como equipo ni como miembro de un equipo de realizar siempre el mismo tipo de tarea a lo largo del desarrollo de la práctica.