



Universidad de las Ciencias Informáticas

Facultad 7

Trabajo de Diploma para optar por el título de
Ingeniero en ciencias Informáticas

**Título: Módulo Epidemiología del Sistema de Información
Hospitalaria alas HIS.**

Autores:

Yunaisy Chacón García

Omar García Bonelly

Tutora:

Ing. Keila García Nogueira

Ciudad de La Habana, Junio de 2009

“Año del 50 aniversario del triunfo de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los 26 días del mes de Junio del año 2009.

Yunaisy Chacón García

Omar García Bonelly

Keila García Nogueira

Firma del Autor

Firma del Autor

Firma del Tutor



'Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber.'

Albert Einstein

DATOS DE CONTACTO

Síntesis del Tutor: Ing. Keila García Nogueira: Instructor recién graduado en el año 2007 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Profesor Facultad # 7. Ha impartido la asignatura Ingeniería de software. Forma parte del proyecto Gestión Hospitalaria.

Correo electrónico: kgarcia@uci.cu

AGRADECIMIENTOS

De Yunaisy:

A Fidel y a la Revolución Cubana, por darme la oportunidad de estudiar en una escuela de excelencia y forjarme como ingeniera y persona de bien.

A la UCI, por haberme dado todas las alegrías, amigos y experiencias inolvidables que llevaré siempre conmigo.

A mis padres por todo su amor, confianza y apoyo incondicional.

A mis hermanitas del alma Kiki y Yummy, que son mi razón de ser.

A mis madrinas queridas Isa y Rosi.

A Tata, por llegar a mi vida en el momento preciso, por estar a mi lado cuando lo necesite y darme su cariño y comprensión.

A Yoe, por ser sobre todas las cosas mi amigo, por estar presente en los momentos buenos y malos brindándome su apoyo incondicional, por su confianza, y sobre todas las cosas por su cariño inigualable y por creer siempre en mí.

A mi compañero de tesis Omarito, por ayudarme a cumplir el sueño de mi vida.

A mi tutora Keila, por todo su aliento, apoyo, constancia y dedicación.

A Maykell, mi oponente, por todo su ayuda y cooperación.

A mis grandes amigas de siempre Maylin, Yunet y Yureilys.

A Iliana, Yisel, Alain, Kenia, Daisy, José, Gilberto, Duniesqui.

En fin, a todas aquellas otras personas que de una forma u otra me brindaron su apoyo y me ayudaron en la realización de este trabajo.

A todos una vez más, MUCHAS GRACIAS.

AGRADECIMIENTOS

De Omar:

A Yeny y Leo, que me apoyaron muchísimo.

A mi tutora Keila, que más que tutora ha sido una compañera más de tesis.

A Yunaisy, que tanto se ha esforzado junto conmigo.

*Especialmente, a mi mamá, mi hermano y mi tío, que sin ellos no hubiese podido estudiar en
la UCI.*

A mis amigos de proyecto que me han ayudado cuando lo he necesitado.

*Y a todos los que de una forma u otra me han ayudado a lo largo de los cinco años de mi
carrera.*

DEDICATORIA

De Yunaisy:

A mi papá por ser mi fiel ejemplo a seguir y motivo de inspiración, por confiar y no perder la esperanza en mí, por darme fuerzas para seguir adelante.

A mi mamá por todo su amor y dedicación, por su paciencia y sacrificio.

A mis hermanitas queridas Yummy y Kiki.

De Omar:

A mi mamá, quien nunca deja de sacrificarse por mí.

RESUMEN

El área de epidemiología, constituye un sector importante en las instituciones hospitalarias, ya que es la encargada de realizar las actividades necesarias para la notificación inmediata de enfermedades. Sin embargo, en la mayoría de estos centros, no existe una aplicación estandarizada que gestione los procesos de esta área. Por lo que el objetivo de este trabajo, es desarrollar el módulo de epidemiología del Sistema de Información Hospitalaria alas HIS, que facilite la gestión de información en esta área de las instituciones hospitalarias.

Para la elaboración del mismo, se utilizó el Proceso unificado racional (RUP), especificando los procesos con la notación para el modelado de procesos de negocio (BPMN). Por otra parte el uso del patrón Modelo-Vista-Controlador, proporcionó una arquitectura bien definida, la cual fue aplicada al diseño e implementación de la solución.

La utilización del sistema en centros hospitalarios, brindará mayor rapidez y eficiencia en cuanto a la notificación de enfermedades de riesgo. Al mismo tiempo, permitirá mantener un control efectivo de pacientes con enfermedades transmisibles, así como la vigilancia de infecciones intrahospitalarias.

PALABRAS CLAVE

Epidemiología, morbilidad, mortalidad, enfermedades de notificación obligatoria, infecciones intrahospitalarias.

TABLA DE CONTENIDOS

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	5
1.1 Conceptos básicos relacionados con el problema planteado.....	5
1.2 Sistemas automatizados existentes	6
1.3 Herramientas y tecnologías de desarrollo.....	14
1.4 Lenguaje de programación.....	20
1.5 Metodologías de desarrollo	22
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	27
2.1 Descripción de los procesos del negocio	27
2.2 Objeto de automatización.....	29
2.3 Modelo del negocio.....	30
2.4 Especificación de los requisitos de software	34
2.5 Modelo de casos de uso del sistema.....	40
2.6 Descripción Textual de los Casos de Uso.....	43
CAPÍTULO 3: DISEÑO DEL SISTEMA.....	49
3.1 Descripción de la arquitectura	49
3.2 Modelo de diseño	50
3.3 Descripción de las clases del diseño	64
CAPÍTULO 4: IMPLEMENTACIÓN	68
4.1 Modelo de Datos	68
4.2 Implementación.....	75
4.3 Tratamiento de errores	78
4.4 Seguridad	78

4.5 Estrategias de codificación. Estándares y estilos a utilizar	79
CONCLUSIONES	87
RECOMENDACIONES	88
REFERENCIAS BIBLIOGRÁFICAS	89
BIBLIOGRAFÍA	93
ANEXOS	95
Anexo 1: Diagramas de procesos de negocio	95
Anexo 2: Diagramas de Casos de Usos por procesos.....	97
GLOSARIO	99

INTRODUCCIÓN

En la actualidad, uno de los principales factores que influyen sobre la calidad de los servicios que brinda cualquier empresa u organización, es, sin lugar a dudas, el control de la información, que surge producto de su interacción con la sociedad. El vertiginoso crecimiento de los volúmenes de datos que se originan, dificulta su manipulación enormemente, y por ello en algunos casos es necesario buscar nuevos métodos para administrarla.

A lo largo de la historia, el hombre ha creado estrategias y medios de almacenamiento para lograr que su información permanezca organizada, segura y disponible. Sin embargo, no todas las prácticas propuestas han tenido éxito y otras no han sido lo suficientemente eficientes, como para poder aplicarlas.

No fue hasta después de la segunda mitad del siglo XX con el surgimiento de las computadoras, que realmente se descubrió una forma más confiable de manipular la información: los programas computarizados. Estos han permitido que se eliminen gradualmente, los errores humanos que cometen los tradicionales secretarios de oficina con los estantes y gavetas.

Estos programas han ido evolucionando, paulatinamente, hasta convertirse en sistemas informáticos, los cuales han cobrado gran popularidad y aceptación en todos los sectores sociales. Han propiciado que se agilice el flujo de las actividades de las instituciones donde se aplica, que mejore su rendimiento, a su vez sirven de apoyo para la toma de decisiones y explotan al máximo las innovaciones tecnológicas con el objetivo de informatizar la sociedad.

Este adelanto tecnológico se ha venido adentrando en la rama de la salud para mejorar la atención a pacientes así como, para aliviar la pesada carga de los trabajadores de esta esfera. Estos operarios invierten diariamente, demasiado tiempo y esfuerzo gestionando la información que fluye de los procesos hospitalarios. De la correcta manipulación de los datos, depende la toma de decisiones de los doctores y el éxito de la atención que reciben los pacientes.

Con este fin, han sido muchos los sistemas que se han creado, desde la década del setenta. Estas herramientas, en la actualidad, reciben el nombre de: Sistemas de Información Hospitalaria (HIS, por sus siglas en inglés), los cuales han ido progresando según se perfecciona la estructura interna de las instituciones de salud o se pule la organización de los servicios sanitarios que brindan.

Los HIS responden directamente a las distintas áreas que posee la entidad médica que lo utilice, las cuales tienen un alto nivel de interrelación. Una de ellas es la epidemiológica, que se encarga, por lo general, de investigar aquellas enfermedades que pueden constituir epidemias o brotes dentro de la

población, además de mantener el control sobre las infecciones nosocomiales y accidentes laborales que ocurren dentro del propio hospital.

El control sanitario es un proceso que es responsabilidad de varios profesionales pertenecientes a diferentes disciplinas y a las distintas áreas de un hospital o instituciones, de ahí la importancia de que estos puedan compartir información sobre los pacientes, de una manera sencilla, segura y conservando el significado original de los datos.

En gran parte del mundo, los procesos de gestión hospitalarias no están completamente automatizados. La mayor parte de estos se hace de forma manual y en algunos casos se cuenta con pequeñas aplicaciones que no constituyen un software estándar para todo el país.

En el caso específico del área de epidemiología, la mayoría de los sistemas existentes no forman parte de un software de gestión hospitalario, sino que constituyen aplicaciones independientes las cuales no se nutren de datos generados por las otras secciones hospitalarias. En la totalidad de los casos la información es introducida en la aplicación por el personal que la opera. Todo esto trae consigo la demora, pérdida e inconsistencia de los datos, sin contar además que complica la labor de los empleados.

Toda esta situación provoca que el trabajo en el departamento de epidemiología se vea afectado, y se torne difícil la creación de las fichas de investigación epidemiológicas para las Enfermedades de Notificación Obligatoria (ENO). La principal causa que dificulta este proceso es la búsqueda de pacientes con ENO, pues esto implica revisar grandes registros o documentos en distintas áreas del hospital tales como emergencia, anatomía patológica y admisión.

En ocasiones, surgen accidentes laborales, los cuales también son tramitados por la unidad epidemiológica, sin embargo, el proceso manual dificulta la consulta de los antecedentes patológicos del trabajador para aplicarle vacunas o tratamientos para evitar infecciones. Al mismo tiempo se lleva un control estricto sobre los biológicos que se distribuyen para la atención hospitalaria, esta es otra de las actividades que pudiera agilizarse con un sistema informático, siendo una de las más importantes de esta área debido al costo de cada uno de estos productos.

Por otro lado, vale señalar que la investigación de infecciones intrahospitalarias y las estadísticas que se generan como resultado de la información recolectada, no son las más cercanas a la realidad, ya que no existe un sistema que estandarice y calcule los consolidados necesarios para la notificación.

Al mismo tiempo el problema del almacenamiento de los datos se agrava debido a la cantidad de documentos que deben ser generados de manera organizada, por lo que se necesitan locales más grandes para archivar los mismos.

Esta situación da lugar también a que las respuestas a solicitudes de información se atrasen, así como la planificación de las diferentes actividades. Por su parte, esto provoca la duplicidad de datos en varios de los departamentos del hospital, en muchos casos, por el desconocimiento de la existencia de los mismos en un área u otra. Otra de las desventajas que trae consigo el procesamiento de información de forma manual es que en muchos casos impide la actualización de los registros debido a la demora del llenado de datos y errores de otro tipo que se cometen durante este procedimiento.

Esto ha dado lugar a la necesidad de crear sistemas hospitalarios que agilicen estos procesos, lo cual de una forma u otra se ha convertido en un problema debido a factores tales como: los errores en el diseño de los sistemas de información; la falta de formación del personal; los cambios permanentes en la tecnología de la información que generan un horizonte inestable y cambiante. De ahí la importancia de definir un modelo que pueda brindar continuidad y viabilidad, adaptándose a las estructuras cambiantes y que sea válido frente a los distintos estilos de gestión, entre otros.

Luego de haber analizado lo dicho anteriormente, se define el siguiente **problema a resolver**: ¿Cómo facilitar la gestión de información relacionada con los procesos en el área de epidemiología de las instituciones hospitalarias?

Este problema se enmarca en el **objeto de estudio** concerniente a los procesos de gestión de información en las instituciones hospitalarias. De manera que el **campo de acción** comprende los procesos de gestión de información en el área de epidemiología de las instituciones hospitalarias.

Basado en esa idea se define el **objetivo general** del presente trabajo: Desarrollar el módulo de epidemiología del Sistema de Información Hospitalaria alas HIS, que facilite la gestión de información en esta área de las instituciones hospitalarias.

Para dar cumplimiento a este objetivo se trazaron las siguientes **tareas a desarrollar**:

1. Evaluar las tendencias actuales en el mundo de los sistemas de información hospitalaria.
2. Identificar los principales procesos llevados a cabo en el área epidemiológica de las instituciones hospitalarias.
3. Definir las funcionalidades del módulo de epidemiología del sistema de información hospitalaria.
4. Asimilar la arquitectura definida por el área temática Gestión hospitalaria.

5. Realizar el modelo de Diseño a partir del modelo de análisis existente.
6. Implementar los procesos de negocio que se relacionan:
 - Realizar vigilancia epidemiológica de la morbilidad hospitalaria.
 - Realizar vigilancia epidemiológica de la mortalidad hospitalaria.
 - Realizar vigilancia de infecciones intrahospitalarias.

Este documento se encuentra estructurado en cuatro capítulos, el primero de ellos, **FUNDAMENTACIÓN TEÓRICA**, donde se estudian los conceptos y sistemas relacionados con la gestión hospitalaria, específicamente enfocados a los procesos de epidemiología; así como las tendencias y tecnologías más utilizadas para la elaboración de los SIH.

Seguidamente el capítulo **CARACTERÍSTICAS DEL SISTEMA**, contiene un marco conceptual asociado a la información que será manipulada por el sistema, llegándose a un acuerdo sobre las funcionalidades, requerimientos deseados y el objeto de automatización.

El tercer capítulo **DISEÑO DEL SISTEMA** se centra en la modelación detallada y la construcción de la estructura de la aplicación. En el cuarto y último, **IMPLEMENTACIÓN**, se implementan las clases y subsistemas en términos de componentes. Se presenta la propuesta de solución para lograr una gestión más eficiente de los procesos hospitalarios asociados al área.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Este capítulo tiene como objetivo abordar diferentes elementos que brindan la base teórica y conceptual para el desarrollo de un HIS que contenga el módulo de epidemiología. En el mismo se realiza un estudio de estos sistemas en el mundo, así como también se describen las técnicas, tecnologías, metodologías y herramientas de software definidas por la dirección del Área Temática de Hospitales, con las que se realizará el proceso de desarrollo.

1.1 Conceptos básicos relacionados con el problema planteado

Para lograr una mejor comprensión con relación al problema planteado, es necesario abordar algunos conceptos relacionados con el tema a tratar.

La investigación en curso gira en torno a la Epidemiología, ciencia que estudia la forma en que se presentan, se transmiten y se previenen las enfermedades. En otras palabras, es el estudio de las epidemias, es decir, de las enfermedades que afectan transitoriamente a muchas personas en un sitio determinado [1].

Una de las principales actividades de esta rama es la *vigilancia epidemiológica*, la cual se define como la recolección sistemática, continua, oportuna y confiable de información relevante y necesaria sobre algunas condiciones de salud de la población [2]. El análisis e interpretación de los datos debe proporcionar bases para la toma de decisiones, y al mismo tiempo ser utilizada para su difusión. Se enfoca principalmente en eventos o casos ya ocurridos, pero cobra fuerza la necesidad de hacerlo también sobre los factores de riesgo que son causa o facilitan su ocurrencia.

Al realizar la vigilancia epidemiológica se identifican diferentes actividades enfocadas a la morbilidad, mortalidad, infecciones intrahospitalarias, accidentes laborales e inspecciones sanitarias.

La *morbilidad*, se describe como el número proporcional de personas que enferman en una población y tiempo determinados [3]. Pueden ser remediadas, curadas o controladas y algunas de ellas pueden ser prevenidas. Es por esto que constituye el estudio de los efectos de una enfermedad en una población en el sentido de la proporción de personas que enferman en un sitio y tiempo determinado.

Por otra parte, la *mortalidad*, se refiere a la tasa de defunciones o el número de defunciones, es decir, de muertes, en un grupo determinado de personas en un período determinado [4].

Las *infecciones intrahospitalarias*, también conocidas como infecciones nosocomiales, son aquellas que se producen dentro de un hospital o una institución de salud [5]. Los pacientes hospitalizados corren el riesgo de adquirir cualquier bacteria o germen, que pueden desencadenar una infección nosocomial.

La vigilancia epidemiológica involucra dentro de sus actividades de morbilidad y mortalidad, la identificación, procesamiento y denuncia de las conocidas *Enfermedades de Notificación Obligatoria (ENO)*, que son aquellas que se consideran de gran importancia para la salud pública [6]. Estas constituyen enfermedades infectocontagiosas, que en algunos casos pueden ser crónicas. Algunas de ellas son de denuncia o notificación obligatoria, internacional o nacional. Este proceso debe ser efectuado por médicos o autoridades de instituciones, a la autoridad sanitaria más próxima.

Como resultado de la identificación de ENO y otras enfermedades crónicas, son elaboradas las *fichas de investigación epidemiológica*. Estas guardan datos personales del paciente, así como la condición actual del mismo, relativa a la enfermedad sospechada. En esta ficha se plasman además, los resultados de los exámenes que se le practican a los pacientes. Algunas de estas son únicas, pero existen otras que en dependencia de la patología permiten ser elaboradas en más de una ocasión.

1.2 Sistemas automatizados existentes

Actualmente, en el campo de la medicina, existen sistemas de gestión hospitalaria, los cuales tramitan los procesos epidemiológicos, así como sistemas independientes, que sólo manejan información epidemiológica. Estos se encargan de gestionar la información de morbi-mortalidad en los hospitales, así como las infecciones, accidentes laborales entre otras actividades necesarias para la notificación inmediata de enfermedades.

El sistema NOTI de la Oficina General de Epidemiología del Ministerio de Salud de Perú, es una de las aplicaciones encontradas, que se encarga de la vigilancia epidemiológica. Su primera versión fue realizada en 1993, esta benefició el sistema de vigilancia epidemiológica del país, y en sus dos primeros años fue utilizado principalmente por la Oficina General de Epidemiología [7].

Para el año 1995, se le realizaron los cambios necesarios al software para que pudiera ser utilizado por las direcciones de salud con el fin de optimizar la vigilancia epidemiológica en estos ámbitos. NOTI está integrado a otros sistemas tales como HIS de la Oficina General de Estadística e Informática, sistemas de información VIH/SIDA, sistemas de información tuberculosis, sistemas de información malaria, entre otros.

Los sistemas de información VIH/SIDA cuentan con grandes debilidades, que hasta cierto punto son críticas, estas se resumen en los siguientes aspectos:

- ✓ Existen múltiples fuentes de información operando simultáneamente y desarticuladamente por diferentes instancias del sistema de salud, sobre una gran cantidad de indicadores.
- ✓ Los propósitos de los sistemas de información son, principalmente, estadísticos. Generan datos de reporte, pero no sirven ni se usan mayormente para gestionar, controlar o monitorear las estrategias o intervenciones de salud, ni proporcionan datos útiles por nivel de gestión.
- ✓ Los sistemas de información no tienen o no están adecuados a modelos de datos estándar, con lo que se limita la capacidad de uso para la gestión y cruces de información (datos claves son estándares de registro y recopilación: pacientes, establecimiento, diagnósticos, exámenes, medicamentos).
- ✓ La mayoría de los datos que se registran y procesan no tienen valor transaccional. Los sistemas no automatizan procesos críticos de la gestión de salud. Por tanto no existe mecanismo autoregulatorio de la oportunidad, calidad y confiabilidad de la información.
- ✓ Gran cantidad de iniciativas de desarrollo de nuevos sistemas y aplicativos informáticos puestos en marcha en los últimos años, han fracasado en su proceso de diseño, o su aplicación se ha descontinuado cuando los promotores (internos y externos) han desaparecido.
- ✓ Actualmente, se hacen encuestas por parte del **Instituto Nacional de Estadística e Informática (INEI)**, que incluyen, Encuesta Nacional de Hogares(ENAHO) y Encuesta Demográfica y de Salud Familiar(ENDES), que deberían o podrían incluir aspectos de salud y no son utilizadas para la toma de decisiones.
- ✓ No existe un código único de los pacientes con VIH, cada paciente tiene un código para la Terapia Antirretroviral (TARGA), otro para laboratorio y otro para epidemiología.
- ✓ La información que se produce no permite visualizar nuevos grupos vulnerables a la enfermedad (niños y niñas, adolescentes, mujeres, población indígena, población en extrema pobreza), ni los cambios que puedan haber ocurrido con las intervenciones de los últimos años.
- ✓ Los indicadores no se han definido consensuadamente y no se incluyen indicadores cualitativos.
- ✓ Falta organizar procesos de retroalimentación a los niveles locales y de análisis de la información, para la toma de decisiones.

- ✓ Los recursos humanos en los establecimientos de salud son insuficientes, más aún cuando tienen que ocupar tiempo en el llenado de formatos para los diferentes sistemas de información [8].

Otro programa dirigido a la gestión de procesos epidemiológicos es Epidat, el cual ha sido desarrollado por instituciones públicas de libre distribución. Este proyecto surge a principios de 1991 en la Dirección General de Salud de la Xunta de Galicia, a raíz de la necesidad de un programa para la gestión de consultas estadísticas y epidemiológicas, producto a la poca existencia y accesibilidad de programas de este tipo.

Este proyecto fue un convenio entre la Organización Panamericana de Salud (OPS) y la Consejería de Sanidad de la Xunta de Galicia. En 1994 salió la primera versión 1.0 que era una calculadora básica en torno DOS, la cual se distribuyó con una gran demanda, principalmente en Brasil. Ya para el año 1997, apareció la versión 2.0 en torno a Windows y en 1998 la versión multilingüe 2.1, que permite seleccionar entre Catalán, Español, gallego, Inglés y Portugués. Para el 2000 se desarrolló la versión 3.0.

Esta última versión opera en ambiente Windows 98 SE o superiores. En esta tercera versión pueden destacarse tres características importantes que lo diferencian de las versiones anteriores que se resumen en:

- Se adopta el ambiente habitual de Microsoft Office, incorporando al entorno de trabajo las barras de menú y herramientas.
- El programa dispone de ayuda en hipertexto con un enfoque más didáctico y crítico, incluyendo fundamentos estadísticos y epidemiológicos.
- Se incorpora una ventana general de resultados, que se puede editar y guardar en formato rtf.
- Se incorpora en casi todos los módulos la posibilidad de importar los datos de forma automática a partir de archivos con tablas en formato dbf, Access o Excel.
- Respecto a los aspectos técnicos, se ha incrementado sustancialmente el contenido del programa con la incorporación de cinco nuevos módulos y opciones adicionales dentro de aquellos módulos que ya se figuraban en la versión previa [9].

Epi Info TM es otro software, mediante el cual epidemiólogos y otros profesionales de la salud y la medicina pueden desarrollar rápidamente cuestionarios, personalizar el proceso de entrada de datos, e introducir y analizar datos.

Brinda la posibilidad de crear estadísticas epidemiológicas, tablas, gráficos y mapas, mediante el uso de comandos tales como *READ*, *FREQ*, *LIST*, *TABLES*, *GRAPH*, y *MAP*. Este también está habilitado para Windows. Entre las principales características con que cuenta este software encontramos:

- Máxima compatibilidad con los estándares de la industria, incluyendo:
 - Microsoft Access y SQL y otras bases de datos ODBC
 - Visual Basic, Versión 6
 - World Wide Web y los navegadores HTML
- Informe de Epi, es una herramienta que permite al usuario combinar análisis de salida, entrada de datos y los datos contenidos en Access o SQL Server, lo que presentará en un formato profesional. Los informes generados se pueden guardar como archivos HTML para facilitar su distribución o publicación web.
- Mapa de Epi, un ArcView ® compatible con el SIG

Adentrándonos en los requerimientos del sistema, de forma general, podemos decir que este opera en ambiente Windows 98, NT 4.0, 2000, XP o Vista. En cuanto a la memoria RAM es necesario una capacidad de 32 MG para acceso obligatorio, adicionando a demás 64MG para Windows 4.0 y 2000, y 128 MG para Windows XP. También necesita un procesador de 200 megahercios, aunque se recomienda que para el caso de Windows XP o vista sea de 300 megahercios.

Para su instalación debe existir en el disco duro (Unidad C) una capacidad libre como mínimo de 390 megabytes. Un requisito importante para el trabajo en Windows es que el paquete de Servicios 3 de Windows XP debe estar instalado; y para Vista debe estar instalado el control de edición para aplicaciones DHLM [10].

El Sistema Nacional de Vigilancia Epidemiológica (SINAVE) de México, para un eficaz funcionamiento, cuenta con un sistema que genera en todos sus niveles, información homogénea acerca de los riesgos y daños a la salud de la población, así como del impacto de los planes y programas de los servicios de salud: el Sistema único de Información para la Vigilancia Epidemiológica (SUIVE).

Unido a este, también están otras herramientas como el Sistema Único Automatizado para la Vigilancia Epidemiológica (SUAVE), la Red Hospitalaria para la Vigilancia Epidemiológica (RHOVE), los sistemas especiales de vigilancia epidemiológica y el Sistema Epidemiológico y Estadístico de las Defunciones (SEED). Estos son sistemas automatizados que concentran información epidemiológica de forma uniforme y oportuna, y facilitan su análisis, es decir, son un conjunto de herramientas que le permiten

agilizar los procesos de recopilación, envío, análisis e interpretación de la información, así como para brindar atención especial a problemas particulares de salud.

Cobertura nacional de los sistemas automatizados



Fig 1

El Sistema Único Automatizado para la Vigilancia Epidemiológica (**SUAVE**) es un paquete de cómputo que concentra la información del Sistema Nacional de Vigilancia Epidemiológica (SINAVE), generada por el Sistema Nacional de Salud (SNS). El mismo opera en el nivel jurisdiccional y permite capturar la información del nivel local, donde puede ser analizada, agiliza el envío por correo electrónico y hace más sencillo, confiable y oportuno el proceso. En este programa es integrada la información sobre los 82 padecimientos sujetos a notificación semanal y 16 inmediata; de ellos, 69 son reportados en el formato Epi-1-95.

Atributos del SUAVE

Características		<ul style="list-style-type: none">• Autoinstalable• Facilita la captura de información del formato Epi-1-95• Contiene bases de datos de programas específicos• Permite obtener reportes de las bases de datos capturadas• Ofrece reportes de manera gráfica y con mapas• Contiene información histórica de morbilidad
Ventajas		<ul style="list-style-type: none">• Permite al usuario su manejo, aun con pocos conocimientos de computación• Concentra información del SNS• Captura información de los nuevos casos de enfermedades• Permite el envío por correo electrónico de las bases de datos capturadas

Fig 2

La Red Hospitalaria para la Vigilancia Epidemiológica (**RHOVE**) es un componente del SINAVE que opera en los hospitales generales y de especialidad para cubrir las necesidades de información acerca de enfermedades de notificación obligatoria y de infecciones nosocomiales, así como de otros padecimientos, entre ellos el mielomeningocele y la fibrosis quística; la mortalidad hospitalaria, en especial la materna y la de menores de edad.

Actualmente, opera en más de ochenta unidades hospitalarias y en los institutos nacionales de salud y cuenta con la participación activa de especialistas en epidemiología, infectología, enfermería en salud pública, informática y otras especialidades clínicas. Está integrada por instrumentos, normas y procedimientos homogéneos para el manejo integral de la información epidemiológica; incrementa la eficiencia en el uso de los recursos y mejora la capacidad de respuesta de las autoridades.

Características de la RHOVE

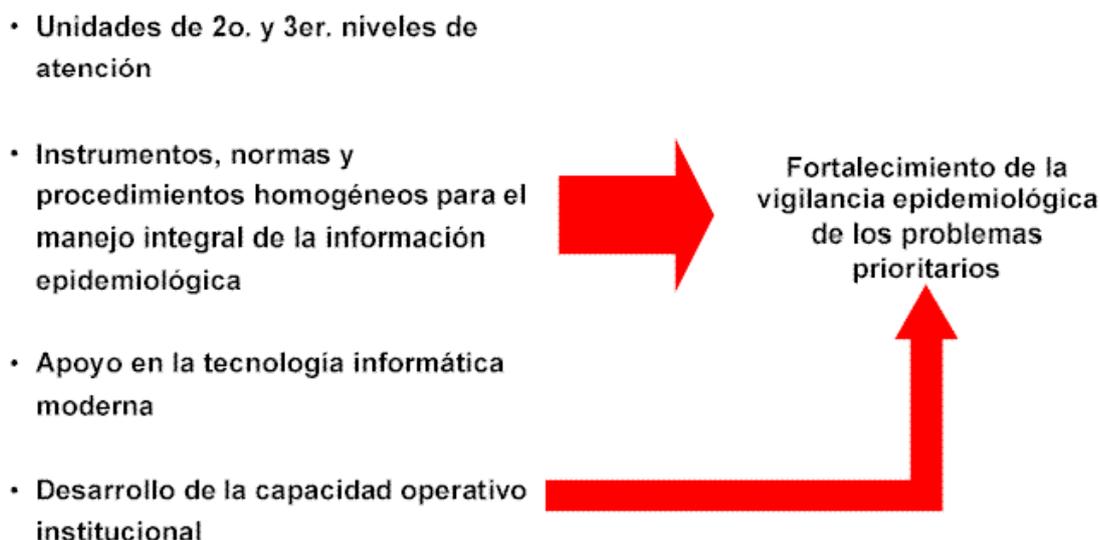


Fig 3

Existen padecimientos que por su magnitud, trascendencia, características o la gravedad de los daños que producen en la población, o bien por compromisos institucionales, son sujetos de la atención especial del Sistema único de Información para la Vigilancia Epidemiológica (SUIVE). Para la vigilancia epidemiológica de estas enfermedades se han puesto en práctica *sistemas especiales de vigilancia epidemiológica*, que frecuentemente requieren estrategias específicas de operación, además del SUAVE, como una red nacional de laboratorios para el diagnóstico, así como fuentes alternativas de información.

Para la recolección y concentración de la información, son utilizados formatos (Epi-2-95, para estudio de caso, Epi-3-95 para estudio de brote, formatos especiales de cada programa); para el flujo de información se utiliza diversos medios de comunicación (fax, teléfono, correo electrónico); asimismo para la concentración y análisis se ha desarrollado sistemas de cómputo especiales.

Estos sistemas son operados en coordinación con los programas sustantivos que integran los esfuerzos de los niveles e instituciones necesarios para su atención y se pueden clasificar de la siguiente manera: Enfermedades Prevenibles por Vacunación, Enfermedades Transmitidas por Vector, VIH-SIDA y Enfermedades de Transmisión Sexual, Tuberculosis y Lepra, Influenza, Intoxicación por Plaguicidas, IRA/EDA, Vigilancia Simplificada, Urgencias Epidemiológicas y Desastres, Programa de

Prevención y Control del Cólera, Cáncer Cérvico-uterino, el Registro Histopatológico de Neoplasias en México, el Sistema de Vigilancia Epidemiológica de los Defectos al Nacimiento, el de Lesiones por Causa Externa y el de las Adicciones.

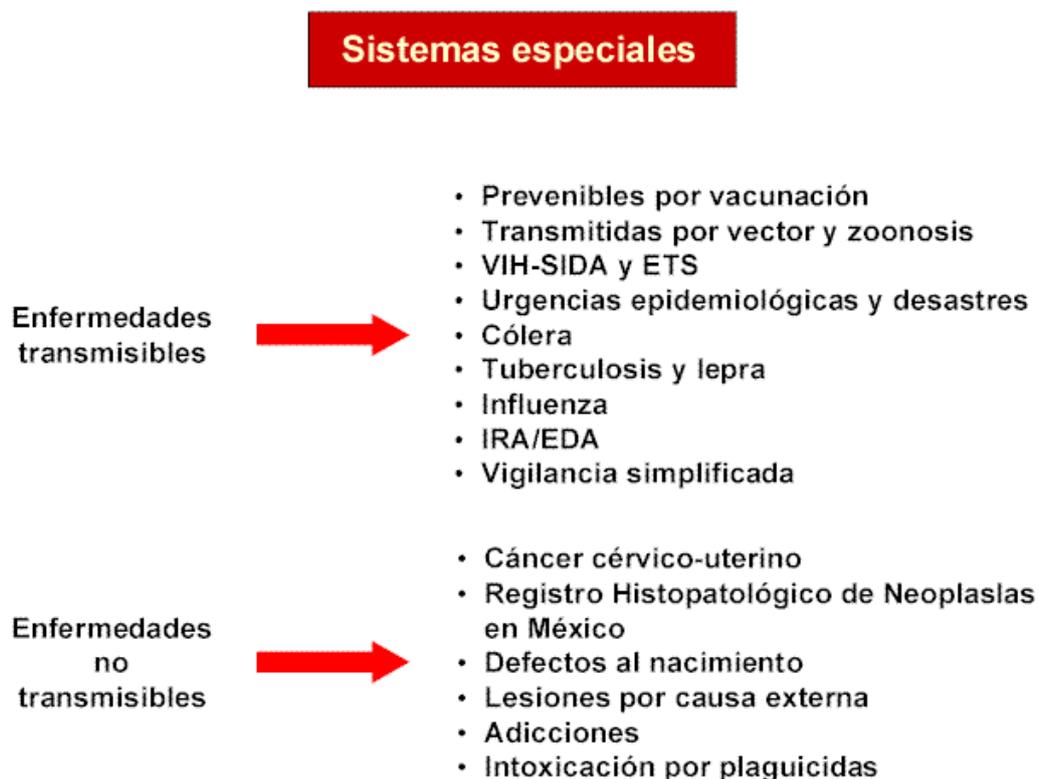


Fig 4

El Sistema Epidemiológico y Estadístico de las Defunciones (SEED) recopila información proveniente de los certificados de defunción y tiene por objeto llevar un registro de las causas de muerte en la población, para detectar riesgos y así estar en posibilidad de desarrollar acciones de salud encaminadas a evitar que la población muera por estas causas.

El SEED es una estrategia para contar con información oportuna sobre la mortalidad; se apoya en un programa de cómputo diseñado para operar en red a nivel nacional, lo que implica grandes ventajas en el manejo de los datos. Mediante este, es posible establecer un sistema de alerta e iniciar acciones inmediatas ante la presencia de problemas de notificación inmediata y urgencias epidemiológicas de importancia local, regional y nacional.

Con la información generada por el SUIVE, se actualiza el panorama epidemiológico nacional, estatal, jurisdiccional y municipal; de igual forma es posible obtener datos semanales, mensuales y anuales; y

en las enfermedades de notificación inmediata, diariamente. La mortalidad permite identificar las áreas de riesgo y las principales causas de muerte en la población y su vinculación con los sistemas especiales de vigilancia. Estos productos recogen información acerca de los grupos poblacionales, lugares, temporalidad y factores de riesgo para cada padecimiento; así, es posible diseñar nuevos programas de salud, reorientar los ya existentes y enfocarlos de manera que logren el objetivo de prevenir y controlar los problemas de salud más importantes [11].

1.3 Herramientas y tecnologías de desarrollo

Eclipse

Eclipse es un entorno de desarrollo integrado de código abierto, portable y multiplataforma. Este fue diseñado originalmente por la empresa IBM y actualmente, es desarrollado por la Fundación Eclipse, una organización independiente, sin ánimo de lucro que fomenta una comunidad de código abierto [12].

Se basa en el uso de módulos (*plugins*), lo cual hace posible el trabajo en múltiples lenguajes de programación como son Java, C++, PHP, Perl y que se le puedan añadir otras funcionalidades. Permite la integración con la herramienta Visual Paradigm a través del SDE Enterprise edition, lo que proporciona un mejor entendimiento entre analistas y desarrolladores.

Además, Eclipse posibilita el trabajo en equipo mediante CVS, que es un sistema de control de versiones y posee un entorno formado por perspectivas personalizables, los cuales son una combinación de vistas y editores que muestran los diversos aspectos de los recursos del proyecto organizados por el rol o la tarea del desarrollador.

JBoss Seam

Es un *framework* de código abierto que une diferentes tecnologías y estándares java adicionando algunas funcionalidades no contempladas. Esto garantiza la plena comunicación de los elementos de la capa de presentación, de negocio y de acceso a datos.

Este integra la capa de presentación (JSF) con la capa de negocios y persistencia (EJB). Con *Seam* basta agregar anotaciones propias de éste a los objetos Entidad y Session de EJB, lo que permite escribir menos código Java y XML.

Otra característica importante es que se puede hacer validaciones en los POJOs (Plain Object Java) y además manejar directamente la lógica de la aplicación y de negocios desde las *sessions beans* [13].

Seam también se integra perfectamente con otros *frameworks* como: RichFaces, ICEFaces (soportan AjaX) MyFaces, Hibernate y Spring.

Alguno de los *frameworks* con los que se integra son:

- Java Server Faces (JSF): Para el desarrollo de la interfaz de usuario, aunque en este sentido Seam contiene la librería *RichFaces/Ajax4JSf*, que reduce enormemente el esfuerzo de los programadores con los componentes web.
- Hibernate: Para el mapeo y el acceso con la base de datos.

Una de las principales ventajas del uso de Seam es que permite el control de sus componentes mediante anotaciones, lo cual reduce enormemente la cantidad de archivos XML de configuración.

JBoss Server

JBoss Server es un servidor de aplicaciones J2EE (plataforma de programación Java) de código abierto implementado en Java puro, lo que implica que puede ser utilizado en cualquier sistema operativo que lo soporte.

Es el primer servidor de aplicaciones de código abierto, preparado para la producción y certificado J2EE 1.4, y ofrece una plataforma de alto rendimiento para aplicaciones de *e-business* [14].

Posee la ventaja de ser compatible con la versión 2.0 del *framework* Seam.

Las características destacadas de JBoss incluyen:

- Producto de licencia de código abierto sin coste adicional.
- Soporta los estándares:
 - ✓ Portlet Specification and API 1.0 (JSR-168)
 - ✓ Content Repository for Java Technology API (JSR-170)
 - ✓ Java Server Faces 1.2 (JSR-252)
 - ✓ Java Management Extensión (JMX) 1.2
 - ✓ Compatibilidad 100% con J2EE 1.4 al utilizar JBoss AS

- Confiable a nivel de empresa
- Incrustable, orientado a arquitectura de servicios.
- Flexibilidad consistente
- Servicios del *middleware* para cualquier objeto de Java

JBoss tools

JBoss Tools es un conjunto de plugins para Eclipse diseñados para ayudar a los desarrolladores de J2EE JBoss a desarrollar sus aplicaciones de forma rápida y con menos esfuerzo [15].

Algunos de estos plugins son:

- RichFaces VE: Editor visual para componentes HTML, JSF y RichFaces
- Seam tools: Incluye soporte para la integración de los componentes del *framework* Seam.
- Hibernate tools: Sirve de apoyo para la utilización de los componentes del *framework* Hibernate y para el mapeo con la base de datos.

RichFaces

RichFaces es una rica biblioteca de componentes para JSF y un avanzado marco para integrar fácilmente capacidades Ajax en el desarrollo de aplicaciones de negocios. Los componentes de la interfaz de usuario de RichFaces vienen preparados para su uso fuera del paquete, así los desarrolladores ahorrarán tiempo para la creación de aplicaciones Web [16].

RichFaces permite definir (por medio de etiquetas de JSF) diferentes partes de una página JSF que se desee actualizar con una solicitud Ajax, proporcionando así varias opciones para enviar peticiones Ajax al servidor.

PostgreSQL

PostgreSQL es un sistema de base de datos relacional de código abierto, que se destaca por su robustez, escalabilidad y cumplimiento de los estándares SQL. Este cuenta con diversas versiones para sistemas operativos tales como Linux, Windows, Mac OS X, Solaris, BSD, Tru64 y otros.

Sus funciones incluyen replicación, respaldos pesados, optimizador avanzado, codificado de caracteres de multibyte, y un tamaño de base de datos ilimitado [17].

También soporta Atomicidad, Consistencia, Aislamiento y Durabilidad (ACID acrónimo de Atomicity, Consistency, Isolation and Durability), lo que proporciona la realización de transacciones seguras, además de vistas, uniones, claves extranjeras, procedimientos almacenados, triggers, entre otras cosas. Incluye la mayor parte de los tipos de datos especificados en los estándares SQL92 y SQL99, como: entero, numérico, booleano, char, varchar, fecha, interval o timestamp.

Otra característica interesante de PostgreSQL es la alta concurrencia, que evita tener que bloquear una tabla cuando se está escribiendo en ella. Asimismo permite realizar copias de seguridad en línea, replicación asíncrona, transacciones anidadas y optimizador de consultas.

En PostgreSQL el tamaño máximo de la base de datos es ilimitado; el de una tabla asciende a 32 TB, el de una fila a 1.6 TB y el de un campo de datos a 1 GB; el número de filas en una tabla es ilimitado, pero no el de columnas, que oscila entre 250 y 1600 columnas por tabla. El número de índices por tabla es también ilimitado.

Esta es la primera base de datos de código abierto en implementar Recorrido Sincronizado, que reduce el uso de E/S en aplicaciones de minería de datos.

El grupo de Windows ha implementado un sistema de compilación con Visual C++, lo cual mejora la estabilidad y rendimiento en Windows, así como la accesibilidad para otros contribuyentes Windows. Nuevas opciones de registro (logging) han sido agregadas y el sobrecosto del recolector de estadísticas ha sido disminuido para hacer más fácil el monitoreo de sus servidores.

La versión 8.3 de PostgreSQL trae consigo nuevas características en fan de mejorar el trabajo de los usuarios, tales como: soporte SQL/XML de acuerdo al estándar ANSI, incluyendo exportación en formato XML; herramienta avanzada de búsqueda en texto, TSearch2, incorporada en la distribución central, con mejor manejo y nuevos diccionarios e idiomas; soporte de autenticación GSSAPI y SSPI; y nuevos tipos de datos: UUIDs, ENUMs y arreglos de tipos compuestos.

PostgreSQL Maestro

PostgreSQL Maestro es una herramienta de administración GUI de alta calidad para el sistema operativo de Windows cuyo objetivo es la administración, control y desarrollo en servidores PostgreSQL [18].

Este permite crear, editar, copiar, extraer y bajar todo objeto de las bases de datos tales como esquemas, tablas, vistas, funciones, dominios, reglas, secuencias, idiomas, operadores, etc. Además brinda la posibilidad de construir consultas visualmente, ejecutar consultas y Scripts SQL, así como visualizar y editar datos incluyendo BLOBs.

Por otra parte, esta herramienta brinda la posibilidad de representar datos como diagramas, exportar e importar datos desde y hacia los formatos de archivos de uso más popular, administrar roles PostgreSQL, usuarios, grupos y sus privilegios; sin contar con que usa una serie de herramientas adicionales diseñadas para la más fácil y eficiente operación con el Servidor PostgreSQL.

Esta herramienta brinda soporte a todas las versiones de PostgreSQL de la 7.3 a la 8.2, permite una fácil administración de base de datos y gestión de objetos particulares de la base de datos.

PostgreSQL Maestro ofrece gran facilidad para la creación de tablas, obteniendo una descarga de su metadata y dato. Permite obtener reportes de su base de datos, los cuales se pueden imprimir en formatos de alta calidad, ejecutando cualquier Script de SQL. También posibilita realizar operaciones de arrastrar y soltar, o presionando el juego de teclas Ctrl+C/Ctrl+V para copiar una tabla de una base de datos a otra y una serie adicional de características.

Framework Hibernate

Hibernate es una capa de persistencia objeto/relacional y un generador de sentencias SQL. Permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos [19]. De una manera muy rápida y optimizada podremos generar BBDD en cualquiera de los entornos soportados: Oracle, DB2, MySQL, entre otras. Es de código abierto.

Uno de los posibles procesos de desarrollo consiste en, una vez tengamos el diseño de datos realizado, mapear este a ficheros XML siguiendo la DTD de mapeo de Hibernate. Desde estos podremos generar el código de nuestros objetos persistentes en clases Java y también crear BBDD independientemente del entorno escogido. Se integra en cualquier tipo de aplicación justo por encima del contenedor de datos.

Visual Paradigm

Visual Paradigm es una herramienta CASE que da soporte al modelado visual con UML y con la notación para la gestión de procesos de negocio (BPMN por sus siglas en inglés). Esta permite recorrer el ciclo de vida del desarrollo de software, desde el modelado de negocio hasta el despliegue del producto.

Dentro de las ventajas de esta herramienta se encuentra la rápida construcción de aplicaciones con gran calidad y menor costo. Permite generar diagramas de clases, código inverso, código desde diagramas y documentación. Proporciona además abundantes tutoriales de UML, así como demostraciones interactivas de dicho lenguaje.

Entre sus características podemos mencionar [20]:

- Soporte de UML versión 2.1
- Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento
- Modelado colaborativo con CVS y Subversion (nueva característica)
- Interoperabilidad con modelos UML 2 (metamodelos UML 2.x para plataforma Eclipse) a través de XMI (nueva característica).
- Ingeniería de ida y vuelta
- Ingeniería inversa - Código a modelo, código a diagrama
- Ingeniería inversa Java, C++, Esquemas XML, XML, .NET exe/dll, CORBA IDL
- Generación de código - Modelo a código, diagrama a código
- Editor de Detalles de Casos de Uso - Entorno todo-en-uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
- Diagramas EJB - Visualización de sistemas EJB.
- Generación de código y despliegue de EJB's - Generación de *beans* para el desarrollo y despliegue de aplicaciones.
- Diagramas de flujo de datos
- Soporte ORM - Generación de objetos Java desde la base de datos

- Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos
- Ingeniería inversa de bases de datos - Desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación
- Generador de informes para la documentación
- Distribución automática de diagramas - Reorganización de las figuras y conectores de los diagramas UML
- Importación y exportación de ficheros XMI
- Integración con Visio - Dibujo de diagramas UML con plantillas (*stencils*) de MS Visio
- Editor de figuras
- Otras herramientas y *plugins* de modelado UML

Esta herramienta case se integra a la Plataforma Java, la cual funciona en sistemas operativos Windows, Linux y Mac OS X. La integración es vista con ambientes inteligentes de desarrollo (SDE) para Eclipse, NetBeans, Oracle JDeveloper, JBuilder, entre otros.

1.4 Lenguaje de programación

Un lenguaje de programación es una secuencia de símbolos creados por la mente humana con el fin de elaborar programas computarizados. Este llevándolo a una definición más científica, se puede decir que constituye un conjunto de reglas sintácticas y semánticas, que definen su estructura y el significado de sus elementos, respectivamente.

Java

Dentro de los lenguajes de programación se encuentra Java el cual es utilizado para crear aplicaciones informáticas. Es un lenguaje orientado a objeto, de una plataforma independiente. La programación en Java, permite el desarrollo de aplicaciones bajo el esquema de Cliente Servidor, como de aplicaciones distribuidas, lo que lo hace capaz de conectar dos o más computadoras u ordenadores, ejecutando tareas simultáneamente, y de esta forma logra distribuir el trabajo a realizar [21].

El lenguaje hereda mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel. Debido a esto ocurren menos errores relacionados con la manipulación directa de punteros o memoria, por lo que se conoce como un lenguaje potente.

Entre sus principales características se destacan:

- Una misma aplicación puede funcionar en diversos tipos de ordenadores y sistemas operativos: Windows, Linux, Solaris, MacOS-X, así como en otros dispositivos inteligentes, de ahí que es multiplataforma.
- Los programas Java pueden ser aplicaciones independientes (que corren en una ventana propia) o "*applets*": pequeños programas interactivos que se encuentran incrustados en una página web y pueden funcionar con cualquier tipo de navegador: Explorer, Netscape, Ópera.
- Es un lenguaje "orientado a objetos", por lo que los programas se construyen a partir de módulos independientes, y estos se pueden transformar o ampliar fácilmente.
- Su desarrollo está impulsado por un amplio colectivo de empresas y organizaciones, y conecta con la filosofía de software abierto y entorno colaborativo.
- Elimina muchas de las características de otros lenguajes como C++, para mantener reducidas las especificaciones del lenguaje y añadir características muy útiles como el *garbage collector* (en español, recolector de basura). No es necesario preocuparse de liberar memoria, el recolector se encarga de ello y como es un *hilo* de baja prioridad, cuando entra en acción, permite liberar bloques de memoria muy grandes, lo que reduce la fragmentación de la memoria.
- Reduce en un 50% los errores más comunes de programación con lenguajes como C y C++ al eliminar muchas de las características de éstos, entre las que destacan:
 - ✓ aritmética de punteros
 - ✓ no existen referencias
 - ✓ registros (struct)
 - ✓ definición de tipos (typedef)
 - ✓ macros (#define)
 - ✓ necesidad de liberar memoria (free)

1.5 Metodologías de desarrollo

Una metodología de desarrollo de software es un conjunto de procedimientos y técnicas que ayudan y guían al desarrollo de productos software [22]. Al igual que un proceso, esta define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo.

Actualmente, no existe una metodología de software universal, ya que las características de cada proyecto son diferentes, lo que trae consigo que el proceso sea configurable.

Existen varias metodologías Orientada a Objeto (OO), basadas en UML dentro de las cuales se puede mencionar el Proceso Unificado racional (RUP).

RUP

El Proceso Unificado de Desarrollo, es una de las metodologías de desarrollo de software más utilizada en el mundo para el análisis, implementación y documentación de sistemas orientados a objetos [23].

Cabe señalar que este no es simplemente un proceso, sino que es un marco de trabajo extensible que puede ser adaptado a organizaciones o proyectos específicos. Generalmente es aplicado a grandes proyectos de desarrollo de software. La forma disciplinada de asignar tareas y responsabilidades, es una de las características fundamentales de RUP, que implementa las buenas prácticas en Ingeniería de Software. Por otra parte el desarrollo iterativo permite que se planifiquen correctamente las funcionalidades.

Durante todo el ciclo de desarrollo, se administran los requisitos, con el objetivo de mantener la trazabilidad de los requerimientos funcionales, siendo así el proceso guiado por casos de uso. Otra de las características fundamentales de RUP es el uso de arquitectura basada en componentes, lo que permite la reutilización de los mismos para otros sistemas o funcionalidades del propio software. RUP además, es centrado en la arquitectura, por lo que es fundamental identificar los casos de uso de mayor prioridad, e importancia basándose en este criterio.

Dentro de sus disciplinas gestiona el control de cambios, que permite mantener al equipo trabajando en los mismos artefactos, en cualquier momento del desarrollo del producto. En su modelación RUP define como sus principales elementos a los trabajadores, las actividades, los artefactos y los flujos de actividades. Los trabajadores son los propietarios de elementos o artefactos y se encargan de realizar

las actividades, las cuales describen cómo una tarea es realizada por un trabajador y a su vez, manipulan elementos.

Los artefactos constituyen los productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables. El flujo de actividades describe cuando estas son realizadas por trabajadores y produce un resultado de valor observable. El ciclo de vida RUP es una implementación del desarrollo en espiral, creado ensamblando los elementos en secuencias semiordenadas, organizando las tareas en fases e iteraciones.

RUP divide el proceso en cuatro fases:

- **Conceptualización (Concepción o Inicio):** Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.
- **Elaboración:** Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido.
- **Construcción:** Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene 1 o varios *release* o versiones del producto que han pasado las pruebas.
- **Transición:** El producto ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores.

Dentro de estas se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades. Estas últimas se han agrupado en grupos lógicos definiéndose nueve flujos de trabajo principales. Los seis primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo.

Dentro de las disciplinas de de ingeniería se encuentran las siguientes:

El **modelado del negocio**, el cual describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.

La **captura de requerimientos** define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.

El **análisis y diseño** describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.

La disciplina de **implementación** define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.

El flujo de trabajo de **Pruebas** se encarga de buscar los defectos a lo largo del ciclo de vida.

El **despliegue** produce el *release* del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.

Las disciplinas de apoyo que define RUP son:

Administración del proyecto, la cual involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.

Administración de configuración y cambios que describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.

Ambiente el cual contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

BPMN

En el flujo de trabajo de Modelado del negocio, se describen fundamentalmente los procesos de negocio. Por un acuerdo de la dirección del Área Temática de Hospitales, para la descripción de tales procesos se definió la utilización de BPMN.

BPMN, traducido al español como Notación para el Modelado de Procesos de Negocio, es una notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo.

El principal objetivo de BPMN es proveer una notación estándar que sea fácil de entender por parte de todos los involucrados e interesados del negocio, díganse los analistas de negocio que son los encargados de definir y redefinir los procesos, los desarrolladores técnicos quienes implementan los procesos y los gerentes y administradores del negocio que son lo que monitorean y gestionan los procesos [24].

En síntesis, esta notación tiene la finalidad de servir como lenguaje común para cerrar la brecha de comunicación que frecuentemente se presenta entre el diseño de los procesos de negocio y su implementación.

UML

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, *Unified Modeling Language*) es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software [25]. A pesar de no ser un estándar oficial, es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.

Este se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software como es el caso de RUP, pero no especifica en sí mismo qué metodología o proceso usar. UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Estos tienen como objetivo fundamental presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. Es importante destacar que un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema.

En este capítulo, después de haber realizado un estudio de los sistemas de gestión epidemiológicos, se puede concluir que los procesos relacionados con esta área de la medicina no se realizan de igual forma en todas las instituciones hospitalarias. Actualmente, la información sanitaria está repartida en múltiples sistemas de información heterogéneos y autónomos, lo que hace que el acceso uniforme a los registros clínicos sea una tarea problemática. Lo cual se debe en su mayoría, a que la elección de la arquitectura de software para la construcción de sistemas que informaticen los distintos servicios de un hospital, difieran de la seleccionada por los otros o simplemente posee dificultades de integración.

Para el desarrollo de la aplicación debido al manejo eficiente y racional de los recursos de memoria, por estar bajo licencias de código abierto y facilitar el trabajo de los desarrolladores, se utiliza *RichFaces* como biblioteca de componentes web, *Hibernate* como mapeador objeto-relacional, *Jboss 4.2 GA* como servidor de aplicaciones web, Eclipse como entorno de desarrollo integrado y Seam como *framework* de integración.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

En este capítulo, se realiza una breve descripción de los procesos del negocio relacionados con el área de epidemiología. Se especifican los actores y trabajadores del negocio, así como los casos de uso y los diagramas de procesos del negocio. Además, se definen los requerimientos funcionales y no funcionales, a partir de los cuales se representan los casos de uso del sistema, la descripción de los mismos y se hace una descripción textual de los casos de uso.

2.1 Descripción de los procesos del negocio

Vigilancia epidemiológica de la morbilidad hospitalaria

La vigilancia epidemiológica de la morbilidad hospitalaria tiene como principal objetivo la creación de fichas de investigación epidemiológicas (FIE) para enfermedades de notificación obligatoria (ENO) y otras enfermedades crónicas [26].

El personal de epidemiología recorre las diferentes áreas del hospital con el objetivo de crear las diferentes FIE. En la de Emergencia, se revisa el libro de emergencia, documento en el cual se listan los pacientes atendidos con sus respectivas patologías. De igual forma, en la terapia intensiva se chequea la historia clínica del paciente hospitalizado verificando el diagnóstico presuntivo del mismo, pues puede suceder que por la gravedad al llegar al hospital, sea enviado directamente a la medicina crítica.

Por otra parte, se revisan las órdenes de admisión que llegan a esta área, pero en este caso realiza una clasificación de las mismas a partir del diagnóstico presuntivo del médico, identificando los pacientes con ENO, enfermedades crónicas, las embarazadas y recién nacidos.

Al identificar enfermedades crónicas, se actualiza la ficha de morbilidad registrada por enfermedades, aparatos y sistemas. Algunas de estas patologías propician la creación de la FIE correspondiente. Luego de identificar las enfermedades en las diferentes áreas, el personal de epidemiología comienza a elaborar las fichas de investigación epidemiológicas, para aquellas que tengan un modelo definido.

La ficha de investigación de casos es elaborada a los pacientes que presentan una patología que no tiene un formato especializado. En todos los casos esta ficha se crea una sola vez para una ENO determinada.

A medida que se identifica una enfermedad de notificación obligatoria, se va actualizando el registro de enfermedades de notificación obligatoria para la morbilidad. Este reporte es enviado semanalmente a la dirección epidemiológica del municipio o distrito correspondiente.

Las FIE de forma general contienen los datos básicos del paciente, datos referentes a la enfermedad y a exámenes de laboratorio. Esta es llenada a partir del interrogatorio que realiza el personal de epidemiología en el servicio de hospitalización, o a partir de la HC. En ese momento se garantiza la toma de una muestra con el objetivo de realizar los estudios para descartar el diagnóstico del paciente.

En ocasiones, el hospital se auxilia de un Instituto o un centro de altos estudios biológicos para procesar los exámenes de laboratorio. En estos casos se crean dos tipos de fichas particulares las cuales se envían a estos recintos, que al obtener los resultados los envían al hospital y al distrito sanitario o municipio correspondiente. Si los exámenes se pueden realizar en el hospital, se esperan los resultados para añadirlos a la FIE y enviarla a la instancia superior. En ambos casos siempre queda una copia en la unidad de vigilancia epidemiológica.

Vigilancia epidemiológica de la mortalidad hospitalaria

El proceso para la vigilancia epidemiológica de la mortalidad hospitalaria también permite la elaboración de FIE, pero a fallecidos que no se pudieron captar por ninguna de las áreas mencionadas en el proceso de morbilidad.

Además de las FIE, se crean otras fichas particulares si los fallecidos son madres, mujeres en edad fértil o niños y niñas menores de 5 años.

En este caso, el personal de epidemiología revisa el Libro de defunciones ubicado en el área de enfermería y el Libro de autopsias de Anatomía patológica. En el caso correspondiente crea una de las tres fichas mencionadas y si es un fallecido que además tiene una ENO, se verifica que ya tenga su FIE y en caso de no tenerla, se hace necesario crearla.

Cuando es identificado un fallecido perteneciente al Proyecto madre se llena el Consolidado correspondiente. Si se creó una FIE o una Ficha de investigación de caso se actualiza el Registro de enfermedades de notificación obligatoria para la mortalidad.

En el caso, de que un fallecimiento haya sido a causa de dengue se crea un Formato de autopsia verbal el cual es enviado al distrito sanitario o municipio de igual forma que las fichas que se crean.

Vigilancia de infecciones intrahospitalarias

La vigilancia de infecciones intrahospitalarias es realizada por el personal médico y de enfermería, el cual se encarga de realizar la vigilancia microbiológica revisando los libros de bacteriología en busca de los resultados de los pacientes, y se encarga de llenar el Registro de vigilancia microbiológica. Además se elabora el formulario para registro de actividades diarias de vigilancia activa de infecciones hospitalarias, verificando en los diferentes servicios los factores de riesgo tanto endógenos y exógenos. Esta actividad se efectúa diariamente, a través del monitoreo.

En el caso de que se sospeche la existencia de la IIH, se crea una Ficha de IIH para cada paciente que presente la misma.

Semanal y mensualmente se realiza el consolidado de IIH que contiene todos los pacientes con IIH identificados en el hospital en un rango de tiempo.

2.2 Objeto de automatización

Como resultado de estos procesos, existen una serie de actividades y registros que serán automatizados. Dentro de estos últimos, se encuentran las Fichas de Investigación Epidemiológicas (FIE), Fichas de investigación de casos (FIC), Fichas de Infección intrahospitalaria (FIIH), entre otras.

Con el desarrollo de un sistema de gestión hospitalaria, estos registros serán automatizados, proporcionando mayor eficacia en el trabajo realizado en esta área del hospital.

Para realizar la vigilancia epidemiológica de la morbilidad hospitalaria, se elaboran una serie de formatos. En el proceso manual, el personal de epidemiología visita las distintas áreas del hospital en busca de las posibles ENO y enfermedades crónicas. Con la informatización, las distintas áreas se encargarán de notificar las enfermedades correspondientes a la unidad de epidemiología. Estas notificaciones permitirán la creación de cada uno de los formatos de las diferentes FIE.

Para realizar la vigilancia epidemiológica de la mortalidad hospitalaria también se crean las FIE, en caso de que el fallecido no se haya identificado por las diferentes áreas. Al mismo tiempo son elaborados los formatos para mortalidad materna, infantil y de mujeres en edad fértil. Con la construcción de un HIS se podrán crear las diferentes fichas de mortalidad materno-infantil a partir del certificado de defunción, al mismo tiempo el sistema permitirá la creación de FIE a partir de las notificaciones.

Dentro de los formatos que se obtienen como resultado de la vigilancia de infecciones intrahospitalarias, se identifican los registros de vigilancia microbiológica y de actividades diarias de vigilancia activa de infecciones hospitalarias.

Por otra parte, se crea la ficha de investigación de infecciones intrahospitalaria, a los pacientes que hayan adquirido un germen o bacteria que desencadene en una enfermedad.

2.3 Modelo del negocio

BPMN como notación de modelado de procesos de negocio define a sus elementos gráficos en cuatro categorías básicas: objetos de flujo, objetos de conexión, *Swimlanes* (calles o carriles de piscina) y artefactos [27]. Cada uno de estos elementos enmarca entre sí a otros.

Dentro de los objetos de flujo, se encuentran los eventos, los cuales se definen como una acción que ocurre en el trascurso de un proceso. Existen tres tipos de eventos: inicial, intermedio y final. Estos afectan el flujo del proceso y suelen tener causa o impacto.

Las actividades están incluidas en este grupo, y son trabajos que se realizan dentro de un proceso. Las mismas pueden ser atómicas o compuestas; y se clasifican en procesos, subprocesos y tareas. Un proceso es una actividad que se realiza dentro de una compañía. Este a su vez puede tener involucrado subprocesos, que son actividades compuestas que dentro tienen detalladamente el flujo de otras actividades. Las tareas son actividades que se incluyen dentro de un proceso y se utilizan cuando no se descompone el modelo en más detalle. Estas son ejecutadas por un usuario y/o aplicación.

Por último, en este grupo se encuentran las bifurcaciones o alternativas (*Gateways*). Estos se usan para controlar la divergencia o convergencia de la secuencia de flujo, es decir, determinan las tradicionales decisiones, así como la creación de nuevos caminos, la fusión de estos o la unión.

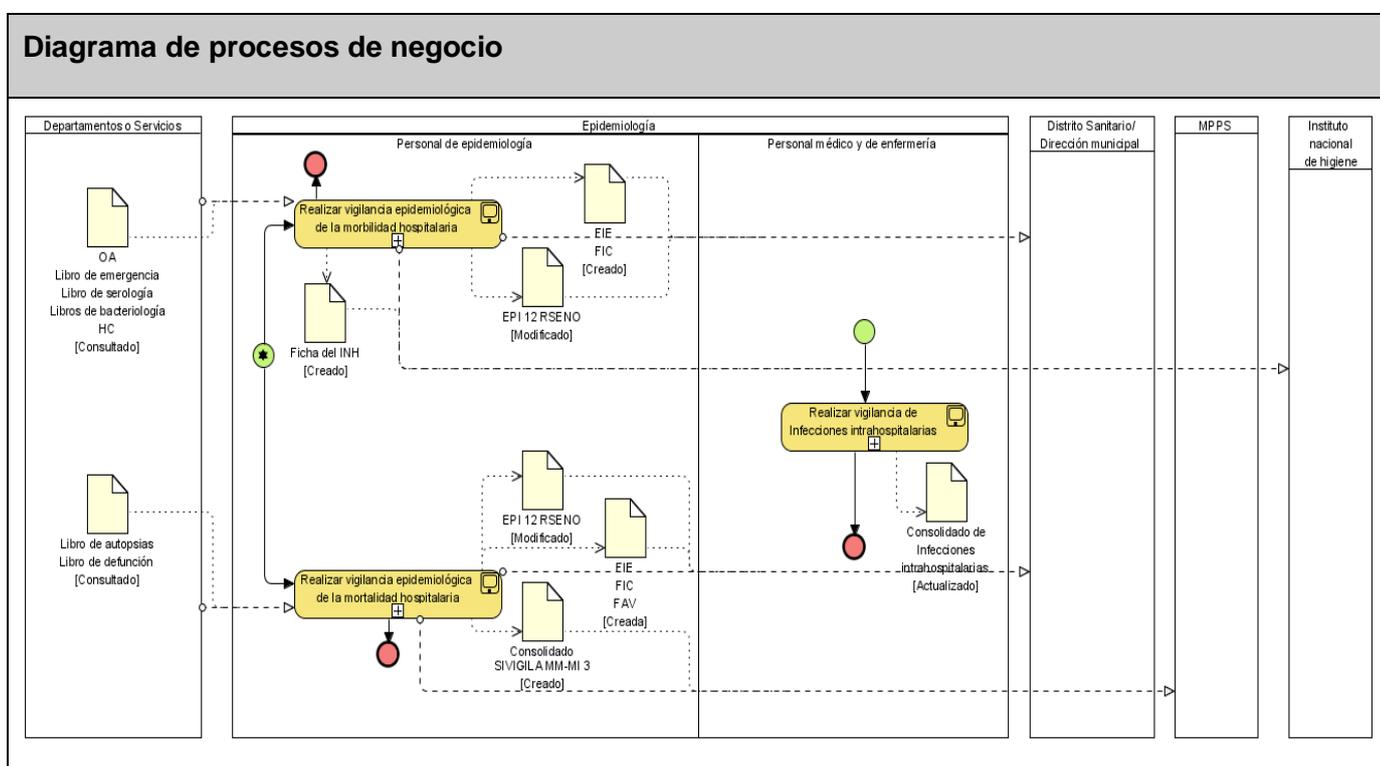
Los objetos de conexión definen el progreso del proceso, es decir, forman el esqueleto básico de la estructura de un proceso uniendo a dos objetos. Existen dos formas de entrelazar, mediante un flujo o una asociación.

Un flujo puede ser de dos categorías, de secuencia o mensajes. El flujo de secuencia es utilizado para mostrar el orden en que las actividades se ejecutan y tienen una sola fuente y un solo destino (eventos, actividades y gateways). El flujo de mensajes es usado para mostrar la comunicación lógica entre dos entidades que están preparadas para mandarlos o recibirlos. La asociación es utilizada para unir información y artefactos con objetos de flujo. Esta muestra las entradas y salidas de las actividades.

Para cada proceso se define un diagrama donde se describen las actividades o tareas del mismo, así como el flujo de entrada y salida de objetos para estas.

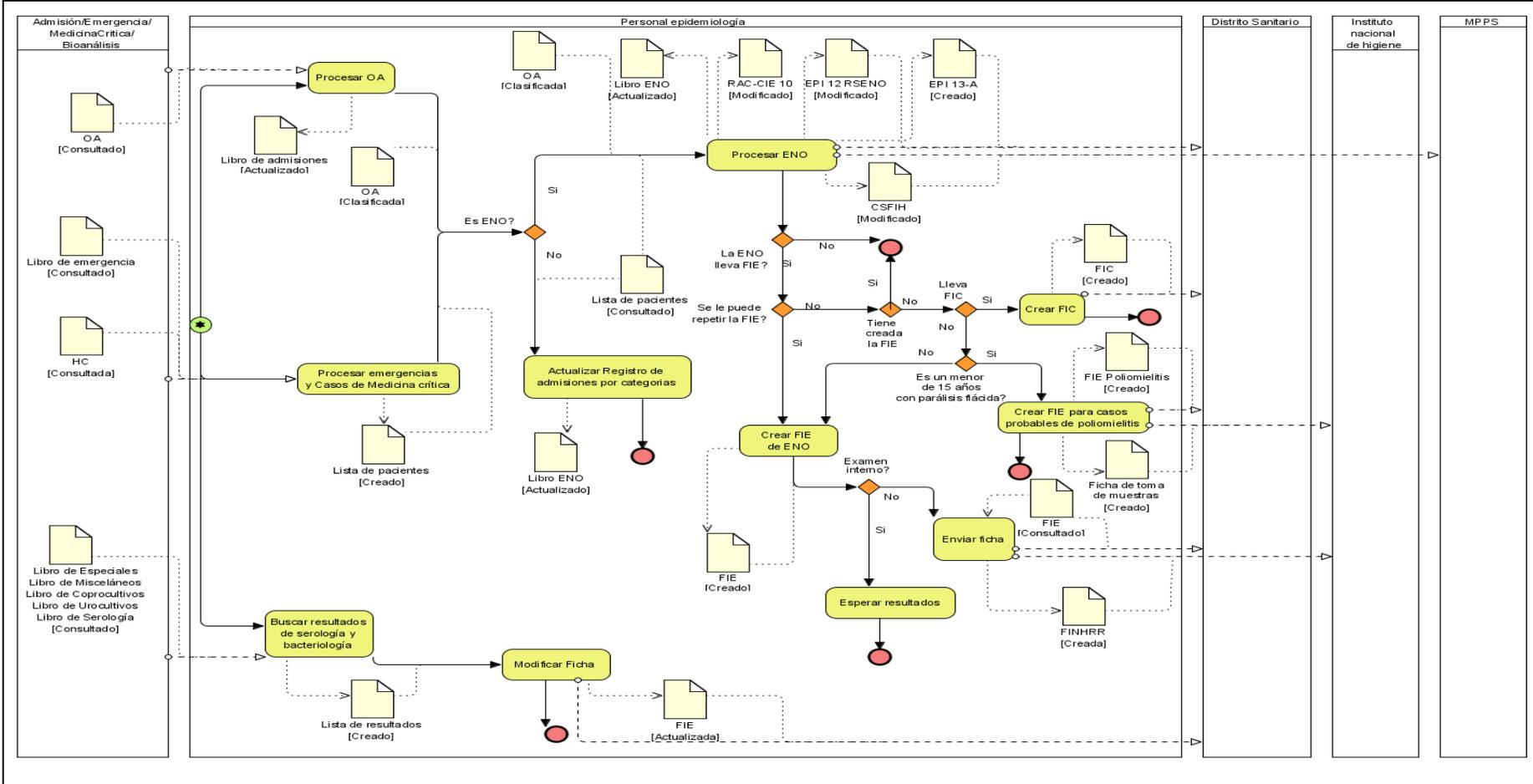
Los diagramas están compuestos por calles, que representan a los actores y trabajadores del negocio; donde el mayor cúmulo de actividades está en las calles de los trabajadores. En estas se muestran las principales acciones que se realizan en un orden lógico y que responden al proceso de negocio descrito.

Por otra parte, se incluyen los objetos que se generan como parte de las actividades, los cuales a su vez sirven de entrada para la realización de las mismas. A continuación se representa el diagrama de procesos de negocio.



En este diagrama se visualizan los tres procesos fundamentales que constituyen el objeto de estudio del trabajo de investigación. Estos son especificados con diagramas de procesos particulares (Ver [Anexo 1](#)), que describen los flujos de actividades, tal es el caso del correspondiente al proceso Realizar vigilancia epidemiológica de la morbilidad hospitalaria que se muestra a continuación:

Realizar vigilancia epidemiológica de la morbilidad hospitalaria



Capítulo 2. Características del sistema.

Para la descripción de estos diagramas, se utiliza una ficha de procesos, que especifica cada una de las actividades relacionadas gráficamente. Esta ficha identifica la misión del proceso a describir, definiendo un responsable para la validación de la información, que en este caso es un especialista funcional.

Al mismo tiempo, la ficha de procesos define los actores involucrados en el mismo, los cuales responden a los principales roles dentro del cúmulo de actividades. Se han identificado los siguientes roles que intervienen en los procesos que se desarrollan en la investigación en curso:

Rol	Justificación
Personal de epidemiología	Interviene en los procesos de Realizar vigilancia epidemiológica de la morbilidad y mortalidad hospitalaria respectivamente. Este es el encargado de realizar las actividades inherentes a la unidad correspondiente a estos procesos.
Personal médico y de enfermería	Interviene en el proceso de Realizar vigilancia de infecciones intrahospitalarias. Es el encargado de realizar la vigilancia en cuanto a accidentes laborales por pinchazos, salpicaduras o cortaduras. También realiza las actividades correspondientes a este proceso en la unidad. Está acreditado para autorizar los certificados de defunción en situaciones extraordinarias.

La ficha contiene además, una especificación del flujo de información que se establece en cada actividad graficada en el diagrama. Estas describen detalladamente el comportamiento del proceso, incluyendo los artefactos necesarios para su realización, así como los resultantes y en ambos casos se especifica la frecuencia y el formato de los mismos.

Por otra parte, se adicionan a la ficha, las principales reglas identificadas dentro del proceso de negocio, siendo este uno de los aspectos fundamentales para el futuro desarrollo del sistema, ya que las reglas constituyen restricciones que el mismo debe tener. Un ejemplo claro se evidencia en la regla que se refiere a la creación única de FIEs, donde se relacionan un grupo de patologías que sólo

permiten crear una vez este formato, a pesar de que el paciente se presente con frecuencia con el mismo diagnóstico.

Finalmente, existe en la ficha un espacio para referenciar los procedimientos con los que cuente la organización y que puedan aportar información sustancial a los procesos identificados, tal es el caso del Manual de Síndrome febril ictero hemorrágico.

La descripción de los procesos mediante diagramas y fichas permite un mejor entendimiento del negocio. A través de las actividades identificadas, se pueden definir entonces las principales funcionalidades o características que el sistema debe tener.

2.4 Especificación de los requisitos de software

Requisitos Funcionales

Los requisitos funcionales de un software son capacidades o condiciones que el sistema debe cumplir, es decir, describen las acciones que este debe ser capaz de realizar. Para los procesos de negocio estudiados y las actividades a automatizar identificadas durante la investigación se definen los siguientes requisitos funcionales:

Funcionalidades comunes dentro del módulo.

- Buscar notificación de enfermedades con formato para FIE.
- Crear FIE de SIREVA.
- Crear FIE de dengue.
- Crear FIE de Síndrome febriles ictero hemorrágico.
- Crear FIE de Enfermedad respiratoria aguda.
- Crear FIE de sarampión- rubéola
- Crear FIE de VIH/SIDA.
- Crear FIE de Enfermedad transmitida por alimentos.
- Crear FIC sospechoso de rabia
- Crear FIE de Poliomiелitis.
- Crear Ficha de muestra de heces de casos de parálisis flácida.
- Crear Ficha de investigación de caso.
- Crear FIC supuestos atribuibles a la vacunación e inmunización.
- Crear FIE de cólera.

- Crear FIE de Hepatitis viral aguda.
- Crear FIE de Leptospirosis.
- Crear FIE de parotiditis.
- Crear FIC de Difteria.
- Crear FIE de Intoxicación por plaguicidas.
- Crear FIE de tuberculosis.
- Crear Ficha del Instituto Nacional de Higiene (INH).
- Buscar Ficha de investigación epidemiológica
- Eliminar Ficha de investigación epidemiológica.
- Ver Ficha de investigación epidemiológica.
- Modificar Ficha de investigación epidemiológica.
- Enviar por correo.

Realizar vigilancia epidemiológica de la morbilidad hospitalaria.

- Crear Plantilla para identificación de alimentos sospechosos.
- Crear Ficha de investigación de manipuladores.
- Crear Ficha de investigación de casos en un brote.
- Crear Resumen de Ficha de Investigación de brotes.

Realizar vigilancia epidemiológica de la mortalidad hospitalaria

- Crear FIE para la vigilancia de la Mortalidad Materna.
- Crear FIE para la vigilancia de la mortalidad en menores de 5 años.
- Crear formato de autopsia para pacientes fallecidos por dengue.
- Buscar formato de autopsia para pacientes fallecidos por dengue.
- Ver formato de autopsia para pacientes fallecidos por dengue.
- Modificar formato de autopsia para pacientes fallecidos por dengue.
- Eliminar formato de autopsia para pacientes fallecidos por dengue.
- Buscar Fichas del Proyecto madre.
- Ver ficha del Proyecto madre.
- Modificar ficha del Proyecto madre.
- Eliminar ficha del Proyecto madre.

Realizar vigilancia de infecciones intrahospitalarias.

- Crear Formulario para registro de actividades diarias de vigilancia activa de infecciones hospitalarias.
- Realizar monitoreo
- Seleccionar Formulario para registro de actividades diarias de vigilancia activa de infecciones hospitalarias.
- Crear Ficha de Infección intrahospitalaria.
- Seleccionar paciente con IIH (Infección intrahospitalaria).
- Buscar Ficha de infección intrahospitalaria.
- Ver Ficha de infección intrahospitalaria.
- Modificar Ficha de Infección intrahospitalaria.
- Eliminar Ficha de Infección intrahospitalaria.

Requisitos no funcionales

Los requisitos no funcionales de un software son propiedades o cualidades que el producto debe tener. Estos muestran las características que hacen al producto atractivo, usable, rápido o confiable. No alteran la funcionalidad del producto, se mantienen invariables sin importar con que propiedades o cualidades se relacionen. Atendiendo a determinadas categorías se definieron los siguientes requisitos no funcionales, para el sistema a desarrollar:

Usabilidad

El sistema estará diseñado de manera que los usuarios adquieran las habilidades necesarias para explotarlo en un tiempo reducido:

- Usuarios normales: 20 días
- Usuarios avanzados: 30 días

Seguridad

- Se mantendrá seguridad y control a nivel de usuario, garantizando el acceso de los mismos sólo a los niveles establecidos de acuerdo a la función que realizan. Las contraseñas podrán cambiarse sólo por el propio usuario o por el administrador del sistema.
- Se registrarán todas las acciones que se realizan, llevando el control de las actividades de cada usuario en todo momento.

- El sistema proporcionará un registro de actividades (log) de cada usuario en el sistema.
- Ninguna información que se haya ingresado en el sistema será eliminada físicamente de la BD.
- El sistema permitirá la recuperación de la información de la base de datos a partir de los respaldos o salvadas realizadas.

Rendimiento

- El sistema minimizará el volumen de datos en las peticiones y además optimizará el uso de recursos críticos como la memoria.
- El sistema respetará buenas prácticas de programación para incrementar el rendimiento en operaciones costosas para la máquina virtual como la creación de objetos.

Soporte

- Se permitirá la creación de usuarios, otorgamiento de privilegios y roles, asignación de perfiles y activación de permisos.
- Se permitirá administración remota, monitoreo del funcionamiento del sistema en los centros hospitalarios y detección de fallas de comunicación.
- Se permitirá realizar copias de seguridad de la base de datos hacia otro dispositivo de almacenamiento externo, además de recuperar la base de datos a partir de los respaldos realizados.
- Se permitirá el chequeo de las operaciones y acceso de los usuarios al sistema. Se permitirá establecer parámetros de configuración del sistema y actualización de nomencladores.

Hardware

➤ **Estaciones de trabajo**

- En la solución se incluyen estaciones de trabajo para las consultas del Sistema de Información Hospitalaria alas HIS, las que necesitan capacidad de hardware que soporte un sistema operativo que cuente con un navegador actualizado y que siga los estándares web (se recomienda IE 7 o superior o Firefox 2.x).
- Las estaciones de trabajo deben tener de 256Mb de memoria RAM y un microprocesador de 2.0Hz con Sistema operativo Linux.

➤ Servidores

La solución estará conformada, fundamentalmente, por servidores de alta capacidad de procesamiento y redundancia, que permitan garantizar movilidad y residencia de la información y las aplicaciones bajo esquemas seguros y confiables.

- Servidores de Base de datos: 1 DL380 G5, Procesador Intel® Xeon® 5140 Dual-Core 4GB de memoria y 2x72GB de disco y sistema operativo Linux.
- Servidores de Aplicaciones: 2 DL380 G5, Procesador Intel® Xeon® 5140 Dual-Core 4GB de memoria y 2x72GB de disco y sistema operativo Linux.
- Servidores de Intercambio: 1 DL380 G5, Procesador Intel® Xeon® 5140 Dual-Core 2 GB de memoria y 2x72GB de disco y sistema operativo Linux.

Software

- El servidor debe correr en sistemas operativos Windows, Unix y Linux, utilizando la plataforma JAVA (Java virtual machine, JBoss AS y PostgreSQL).
- Los clientes deberán disponer de un navegador web, estos pueden ser IE 7 o superior, Opera 9 superior, Google chrome 1 o superior y Firefox 2 o superior.

Restricciones de diseño

- La capa de presentación contendrá todas las vistas y la lógica de la presentación. El flujo web se manejará de forma declarativa y basándose en definiciones de procesos del negocio.
- La capa del negocio mantendrá el estado de las conversaciones y procesos del negocio que concurrentemente pueden estar siendo ejecutados por cada usuario.
- La capa de acceso a datos contendrá las entidades y los objetos de acceso a datos correspondientes a las mismas. El acceso a datos está basado en el estándar JPA y particularmente en la implementación de éste.

Requisitos para la documentación de usuarios en línea y ayuda del sistema.

- Se posibilitará el uso de ayudas dinámicas y tutoriales en línea sobre el funcionamiento del sistema.

Interfaz

➤ **Interfaces de usuario**

- Las ventanas del sistema contendrán claro y bien estructurados los datos, además de permitir la interpretación correcta de la información.
- La interfaz contará con accesos directos y menús desplegados que faciliten y aceleren su utilización.
- La entrada de datos incorrecta será detectada claramente e informada al usuario.
- Todos los textos y mensajes en pantalla aparecerán en idioma español.

➤ **Interfaces de comunicación**

- Para el intercambio electrónico de datos entre aplicaciones se usará el estándar HL7 (Health Level Seven).
- El sistema usará el formato estándar WSDL para la descripción de los servicios web.
- El sistema implementará mecanismos de encriptación de datos para el intercambio de información con sistemas externos.
- El sistema utilizará mecanismos de compactación de los datos que se intercambiarán con sistemas externos con el objetivo de minimizar el tráfico en la red y economizar el ancho de banda.

Portabilidad

- El producto podrá ser utilizado bajo los sistemas operativos Linux o Windows.

Luego de la definición de las principales funcionalidades que el sistema debe tener, así como las cualidades del producto a desarrollar, se procede a la definición del modelo de casos de uso del sistema. Este se compone por actores, casos de uso y las relaciones entre ellos, y responde a cada una de las funcionalidades definidas en este epígrafe.

2.5 Modelo de casos de uso del sistema

Actores del sistema

Actor	Descripción
Usuario	Usuario global que permite la autenticación en el sistema y que valida al mismo dándole un rol.
Personal de epidemiología	Es el encargado de realizar las actividades inherentes a la unidad así como los procesos de morbi-mortalidad.
Personal médico y de enfermería	Es el encargado de realizar la vigilancia en cuanto a accidentes laborales por pinchazos, salpicaduras o cortaduras. También realiza la vigilancia de IIH. Está acreditado para autorizar los certificados de defunción en situaciones extraordinarias.

Vista global de actores

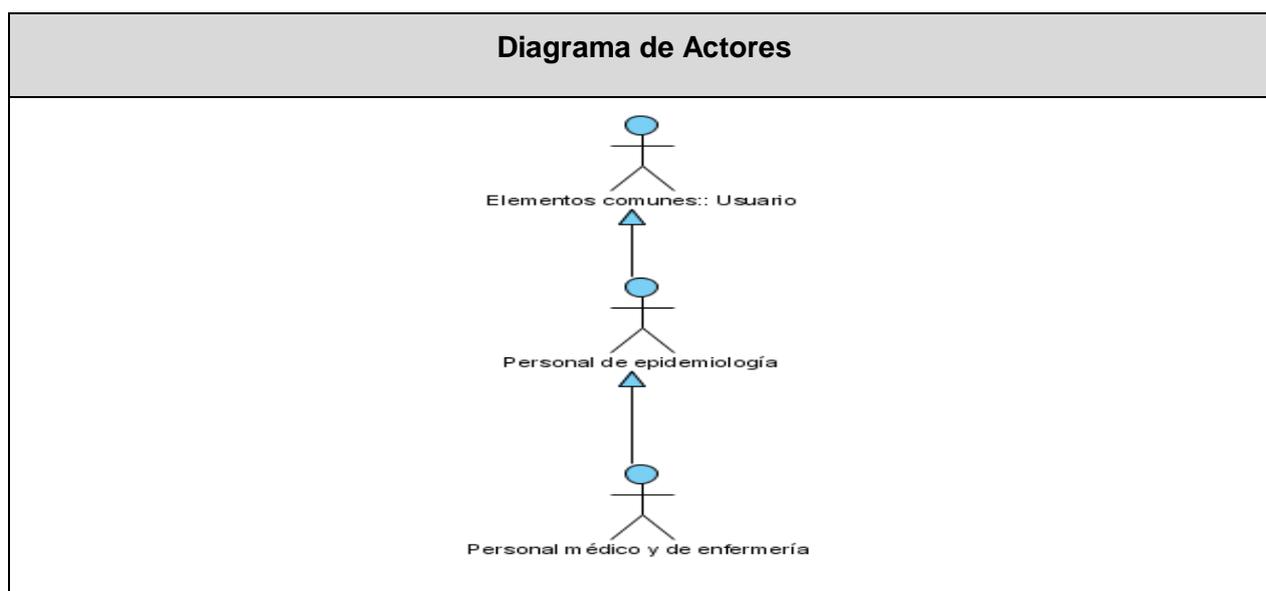
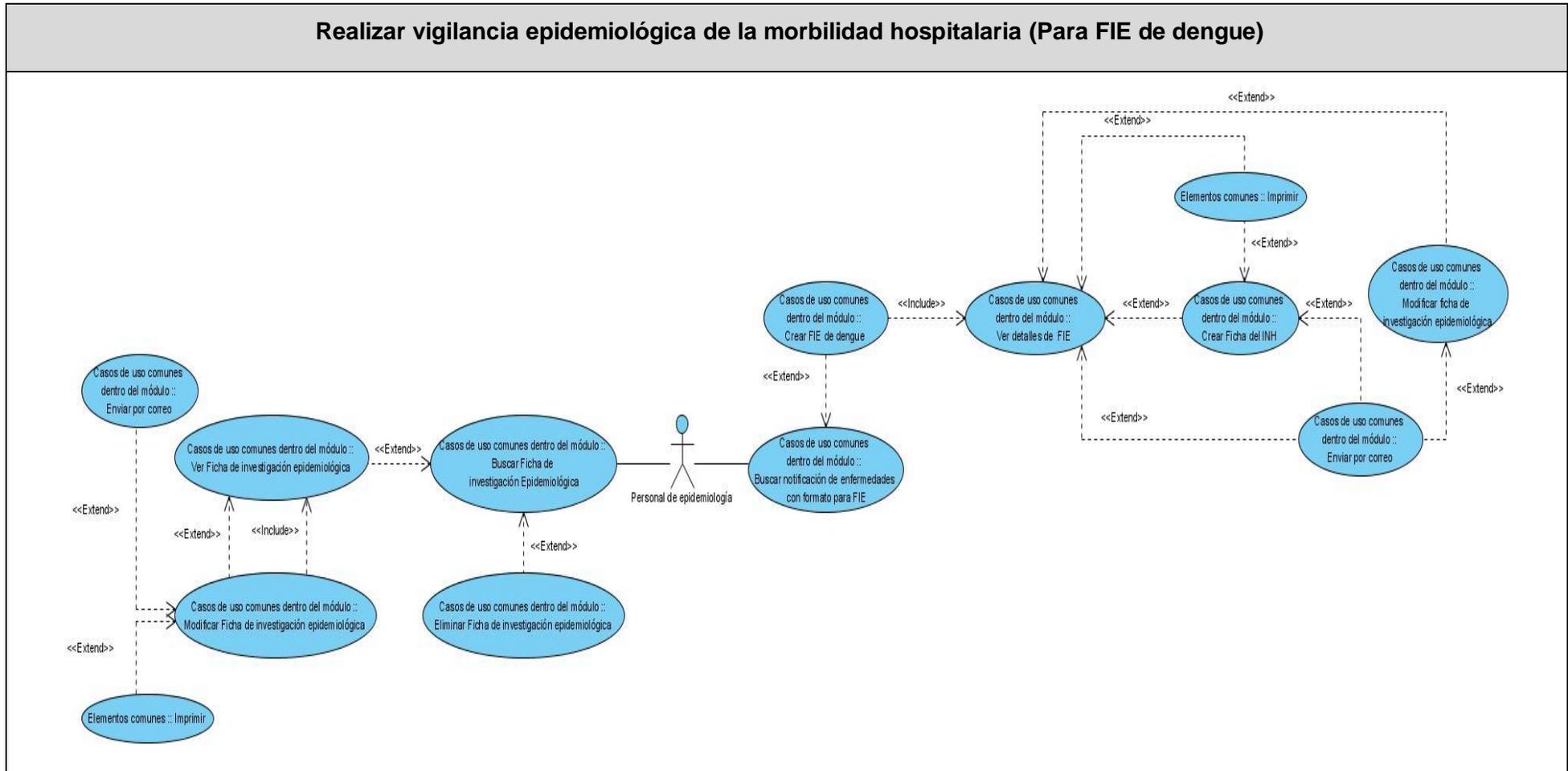
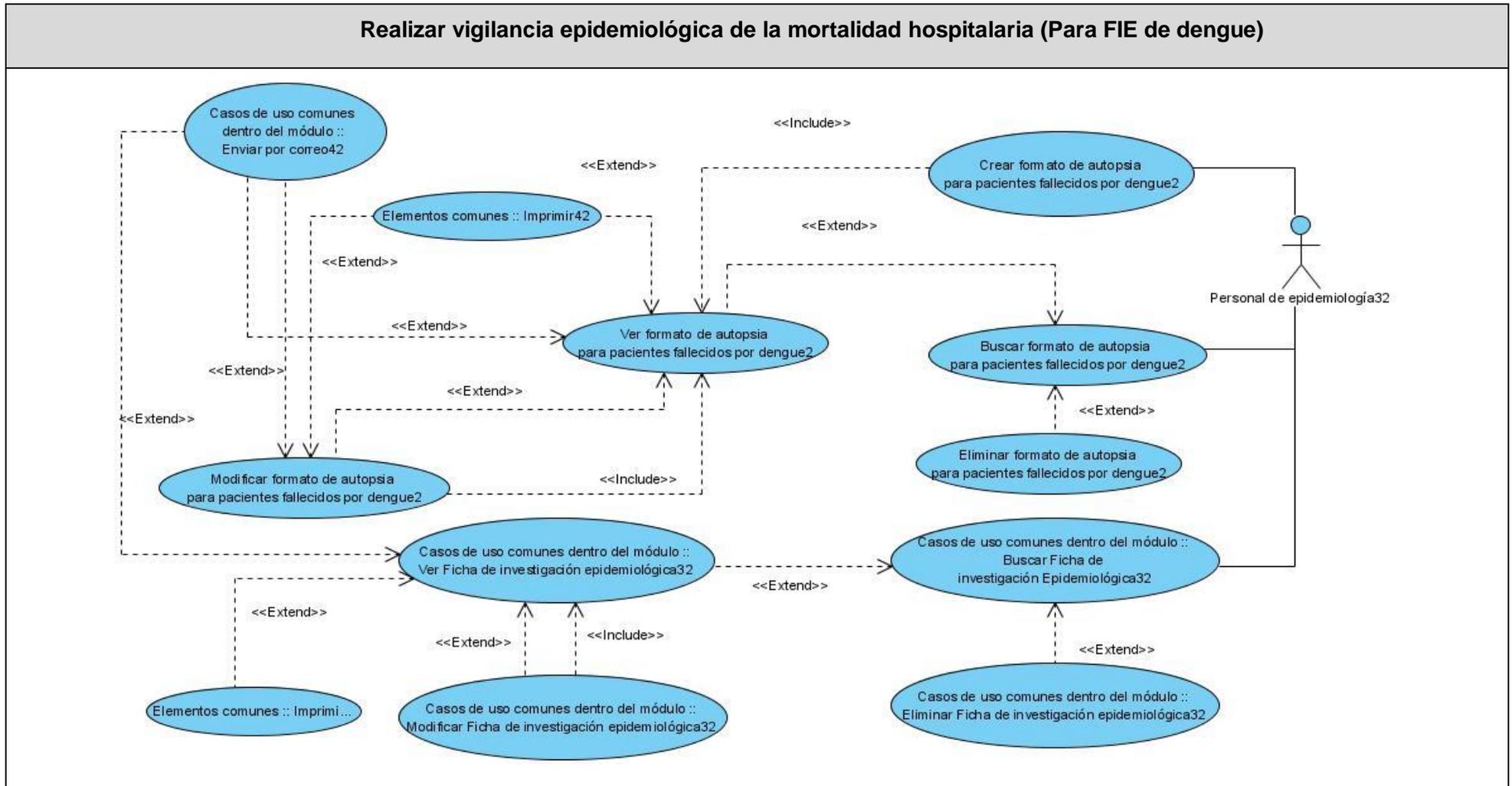
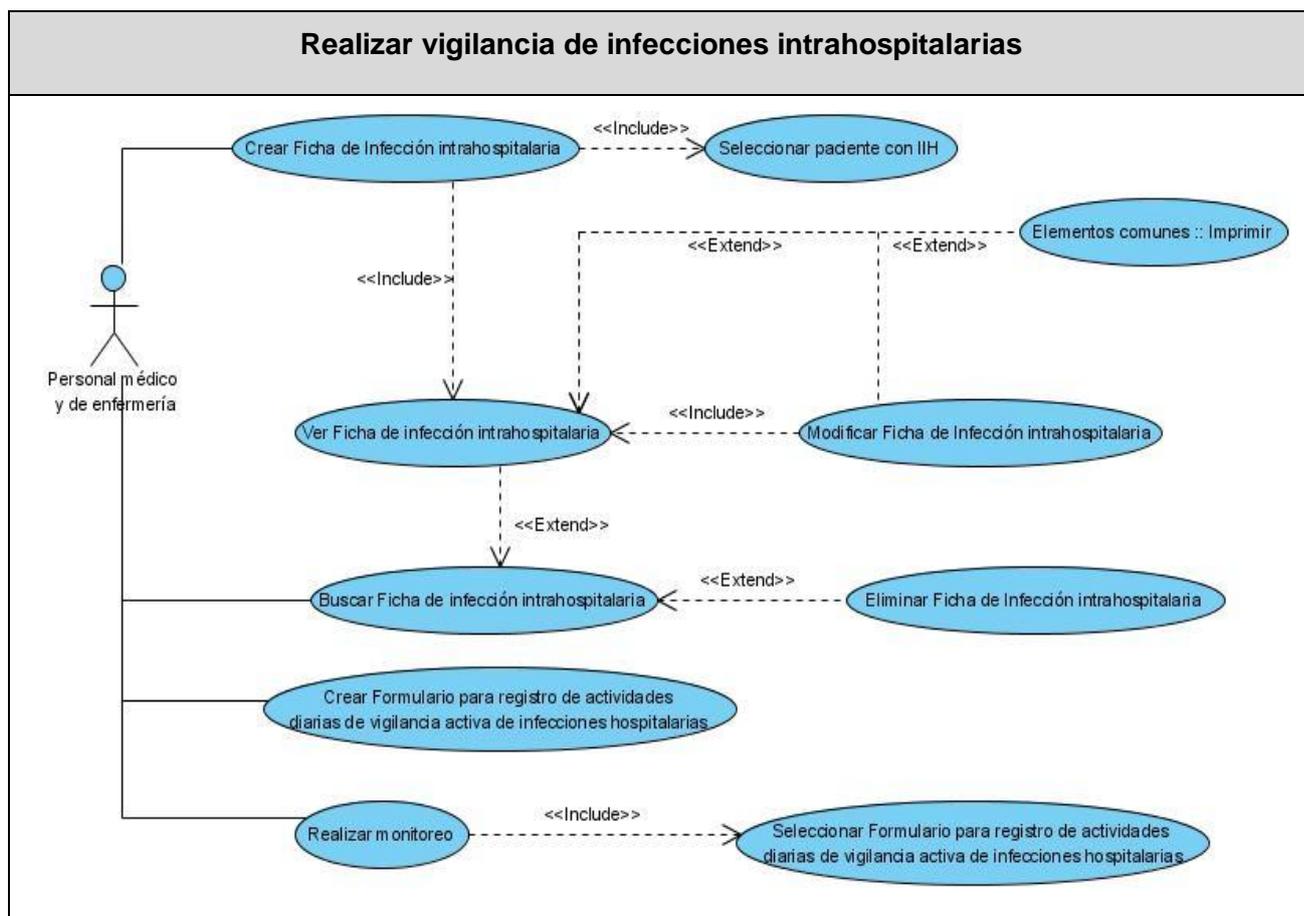


Diagrama de casos de uso







2.6 Descripción Textual de los Casos de Uso

Buscar notificación de enfermedades con formato para FIE.

2.2.1	Buscar notificación de enfermedades con formato para FIE
Actor	Personal de epidemiología
Descripción	El caso de uso inicia cuando el personal de epidemiología accede a la opción Buscar notificación de enfermedades con formato para FIE. El sistema brinda la posibilidad de introducir criterios de búsqueda para localizar la notificación, el personal de epidemiología introduce los datos que considera como criterios para realizar una búsqueda. El sistema busca y muestra las notificaciones que cumplen con los criterios de búsqueda, el caso

	de uso termina.
Precondición	-
Poscondición	Se buscaron las notificaciones de enfermedades con formato para FIE, dado criterios.
Referencia	<ul style="list-style-type: none"> • Crear FIE de SIREVA • Crear FIE de dengue • Crear FIE de Síndrome febriles ictero hemorrágico • Crear FIE de Enfermedad respiratoria aguda • Crear FIE de sarampión- rubéola • Crear FIE de VIH/SIDA • Crear FIE de enfermedad transmitida por alimentos • Crear FIC sospechoso de rabia • Crear FIE de Poliomielitis • Crear Ficha de investigación de caso • Crear FIC supuestos atribuibles a la vacunación e inmunización • Crear FIC de difteria • Crear FIE de hepatitis viral aguda • Crear FIE de cólera • Crear FIE de leptospirosis • Crear FIE de intoxicación por plaguicidas • Crear FIE de tuberculosis • Crear FIE de Síndrome rubéola congénita • Crear FIE de brucelosis • Crear FIE de síndrome respiratorio agudo severo • Crear FIE de emponzoñamiento ofídico

Crear FIE de Dengue

2.2.3	Crear FIE de dengue
Actor	Personal de epidemiología
Descripción	El caso de uso inicia cuando el Personal de epidemiología accede a la opción Crear FIE de dengue. El sistema brinda la posibilidad de introducir los datos para crear la FIE de dengue, el Personal de epidemiología introduce los datos de la FIE de dengue. El sistema crea la FIE de dengue y el caso de uso termina.
Precondición	Debe haberse seleccionado el paciente notificado con dengue.
Poscondición	Se creó una FIE de dengue.
Referencia	<ul style="list-style-type: none"> • Ver detalles de Ficha de investigación epidemiológica

Buscar Ficha de investigación Epidemiológica

2.2.22	Buscar Ficha de investigación Epidemiológica
Actor	Personal de epidemiología
Descripción	El caso de uso inicia cuando el actor accede a la opción Buscar Ficha de investigación Epidemiológica, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar la Ficha de investigación Epidemiológica, el actor introduce los datos que considera como criterios para realizar una búsqueda, el sistema busca y muestra las Fichas de investigación Epidemiológica que cumplen con los criterios de búsqueda, el caso de uso termina.
Precondición	Debe existir al menos una Ficha de investigación Epidemiológica.

Poscondición	Se buscó una Ficha de investigación epidemiológica dado criterios.
Referencia	<ul style="list-style-type: none"> • Ver Ficha de investigación epidemiológica • Eliminar Ficha de investigación epidemiológica

Eliminar Ficha de investigación epidemiológica

2.2.23	Eliminar Ficha de investigación epidemiológica
Actor	Personal de epidemiología
Descripción	El caso de uso inicia cuando el actor selecciona una Ficha de investigación epidemiológica y accede a la opción Eliminar ficha de investigación epidemiológica, el sistema elimina la Ficha de investigación epidemiológica, el caso de uso termina.
Precondición	Para eliminar una ficha de investigación epidemiológica, esta debe haber sido seleccionada.
Poscondición	Se eliminó una Ficha de investigación epidemiológica por el actor.
Referencia	-

Ver Ficha de investigación epidemiológica

2.2.24	Ver Ficha de investigación epidemiológica
Actor	Personal de epidemiología
Descripción	El caso de uso inicia cuando el actor selecciona una Ficha de investigación epidemiológica y accede a la opción de Ver Ficha de investigación epidemiológica, el sistema muestra los datos de la Ficha de investigación epidemiológica y el caso de uso termina.
Precondición	Para ver los datos de una Ficha de investigación epidemiológica,

	esta debe estar seleccionada.
Poscondición	Se vieron los datos de una Ficha de investigación epidemiológica.
Referencia	<ul style="list-style-type: none"> • Modificar Ficha de investigación epidemiológica

Ver detalles de Ficha de investigación epidemiológica

2.2.21	Ver detalles de Ficha de investigación epidemiológica
Actor	Personal de epidemiología
Descripción	El caso de uso inicia cuando se ha creado o modificado una Ficha de investigación epidemiológica. El sistema muestra los detalles de la Ficha de investigación epidemiológica creada o modificada y el caso de uso termina.
Precondición	Para ver los detalles de una Ficha de investigación epidemiológica, esta debe haberse creado o modificado.
Poscondición	Se vieron los detalles de una Ficha de investigación epidemiológica.
Referencia	<ul style="list-style-type: none"> • Modificar Ficha de investigación epidemiológica • Crear FINH • Enviar por correo • Elementos comunes :: Imprimir

Modificar Ficha de investigación epidemiológica

2.2.25	Modificar Ficha de investigación epidemiológica
Actor	Personal de epidemiología
Descripción	El caso de uso inicia cuando el actor selecciona una Ficha de

	investigación epidemiológica y accede a la opción Modificar Ficha de investigación epidemiológica, el sistema muestra los datos de la Ficha de investigación epidemiológica y brinda la posibilidad de cambiar sus valores ya sea introduciendo nuevos o seleccionando diferentes, el actor modifica los datos que necesita, el sistema actualiza los datos de la Ficha de investigación epidemiológica, el caso de uso termina.
Precondición	Para modificar los datos de una Ficha de investigación epidemiológica, esta debe haber sido seleccionada y visualizada.
Poscondición	Se modificó una Ficha de investigación epidemiológica por el actor.
Referencia	<ul style="list-style-type: none">• Enviar por correo• Elementos comunes :: Imprimir

En este capítulo, se describieron los procesos del negocio -los cuales constituyen la base para el desarrollo del sistema- a partir de las fichas de procesos y los diagramas de procesos de negocio en los cuales se identificaron los actores involucrados, así como las actividades de conjunto con los artefactos. A partir de estas actividades, se definieron los requerimientos funcionales y no funcionales del sistema, los cuales obtuvieron su traducción en el Modelo de casos de uso. Como resultado de este análisis se deriva el diseño de una herramienta automatizada que mejore el funcionamiento de los procesos descritos anteriormente.

CAPÍTULO 3: DISEÑO DEL SISTEMA

En el presente capítulo se realiza una descripción detallada del sistema propuesto donde se describe la arquitectura definida por el Área temática de Hospitales, además se aborda el diseño de la solución, modelándose los artefactos necesarios que contribuyen a la implementación del sistema.

3.1 Descripción de la arquitectura

La arquitectura de software representa básicamente la estructura de un sistema y comprende los principales componentes del mismo, sus propiedades externamente visibles y las relaciones entre ellos. Constituye además un modelo comprensible de cómo está estructurado el sistema y de cómo trabajan juntos sus componentes.

Para el desarrollo del sistema, se opta por la puesta en práctica de una arquitectura basada en el patrón Modelo Vista Controlador, muy usado en aplicaciones web, el cual separa los datos de una aplicación, la interfaz de usuario y la lógica de control, en tres componentes distintos [28]: el modelo, donde se ubican los datos y las reglas del negocio; la vista, que muestra la información del modelo al usuario; y el controlador, que gestiona las entradas del mismo. Esto se hace con el objetivo de realizar un diseño que desacople la vista del modelo y que permita la reusabilidad de los componentes. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos.

Siguiendo este patrón, se agrupan los componentes del módulo en tres capas fundamentales: presentación, negocio y acceso a datos, con el fin de darles una mejor organización según la función que realizan, permitiendo que en un momento determinado, un elemento de una capa pueda ser sustituido causando el mínimo de alteraciones en otro elemento que lo utilice.

La capa de presentación está formada por páginas XHTML, las cuales contienen formularios que mediante controles JSF y RichFaces recogen, validan y muestran los datos que el usuario provee y solicita en cada una de las operaciones que realiza.

La capa de negocio está integrada por clases controladoras, que encierran la lógica del negocio del módulo, a las cuales, mediante anotaciones que provee el *framework* Seam, se les especifica el contexto en que se encuentran (conversacional, evento, página, etc.), los cuales definen el estado de los datos y las entidades que manejan.

La capa de acceso a datos está constituida por componentes Hibernate, mediante los cuales todo el proceso de inserción, selección, modificación y eliminación de los datos se hace orientado a objetos, a través de un manejador de entidades; además de que incorporan validación de campos.

La comunicación entre los elementos de estas capas está regida por el *framework* Seam, que permite, mediante anotaciones, que las páginas de la interfaz de usuario referencien las funcionalidades definidas en las clases controladoras, y que estas puedan usar los componentes de acceso a datos y otros de la capa del negocio.

A partir del patrón arquitectónico definido para el desarrollo de la aplicación, se puede proceder a diseñar los casos de uso del sistema.

3.2 Modelo de diseño

El diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, o sea, cómo cumple el sistema sus objetivos. Además debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades. Por otra parte tiene como propósito traducir los requisitos a una especificación que describe cómo implementar el sistema, es decir, pretende crear un plano del modelo de implementación.

Para la elaboración del diseño, generalmente se utilizan un grupo de patrones, o modelos para lograr los objetivos específicos. Estos forman parte de las mejores prácticas de investigadores y diseñadores de software orientado a objetos. De manera general, los patrones constituyen soluciones simples y elegantes a problemas específicos y comunes del diseño *orientado a objetos*, basados en la experiencia y que se ha demostrado que funcionan [29].

Los patrones de diseño constituyen un vocabulario común de buenas soluciones, perfectamente identificadas y aplicables a distintos problemas típicos de diseño, que pueden encontrarse en diferentes contextos. Estos facilitan la reutilización del conocimiento experto como componentes de diseño y mejoran así la documentación, comprensión y comunicación del diseño final. Dentro de los más conocidos están los GRASP o patrones de asignación de responsabilidades.

Los patrones GRASP se utilizan con el objetivo de asignar responsabilidades a las diferentes clases que se definen en el diseño. Dentro de este grupo se identifican cinco patrones muy utilizados: experto, creador, alta cohesión, bajo acoplamiento y el controlador [30].

En el diseño se modela el sistema para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen. Al mismo tiempo se define la arquitectura del sistema. En este modelo, los casos de uso son realizados por las clases del diseño y sus objetos, a partir de los cuales se forma el diagrama de clases del diseño.

Una *clase de diseño* es aquella suficientemente detallada que sirve como base para generar código fuente o lo que es lo mismo, es aquella cuya especificación es completa hasta un nivel que se pueda implementar [31].

Los **diagramas de clases de diseño** exponen un conjunto de interfaces, colaboraciones y sus relaciones. Se utilizan para modelar la vista de diseño estática de un sistema. Estos son de gran importancia, ya que permiten visualizar, especificar y documentar modelos estructurales. Los diagramas de clases de diseño forman parte de las realizaciones de casos de usos.

Otro de los artefactos indispensables en la realización de los casos de uso son los **diagramas de interacción**, que muestran gráficamente como los objetos se comunican entre ellos a fin de cumplir con los requerimientos. Los diagramas de interacción juegan un papel importante en esta disciplina, ya que sirven para modelar los aspectos dinámicos de un sistema [32].

Dentro de los diagramas de interacción, se definen dos tipos fundamentales, los de secuencia y colaboración, generalmente se recomienda utilizar las colaboraciones en el análisis y la secuencia en el diseño. Un diagrama de secuencia es un diagrama de interacción que destaca la ordenación temporal de los mensajes.

A partir de los principales aspectos a tener en cuenta para la elaboración del modelo de diseño, se define una estructura de paquetes que permita dividir el sistema en trozos manejables para su futura implementación. Se utilizan dos criterios de empaquetamiento: por procesos y por clases. A partir de los procesos definidos en el sistema, se grafican los paquetes correspondientes, estos utilizan el paquete repositorio de clases para su funcionamiento.

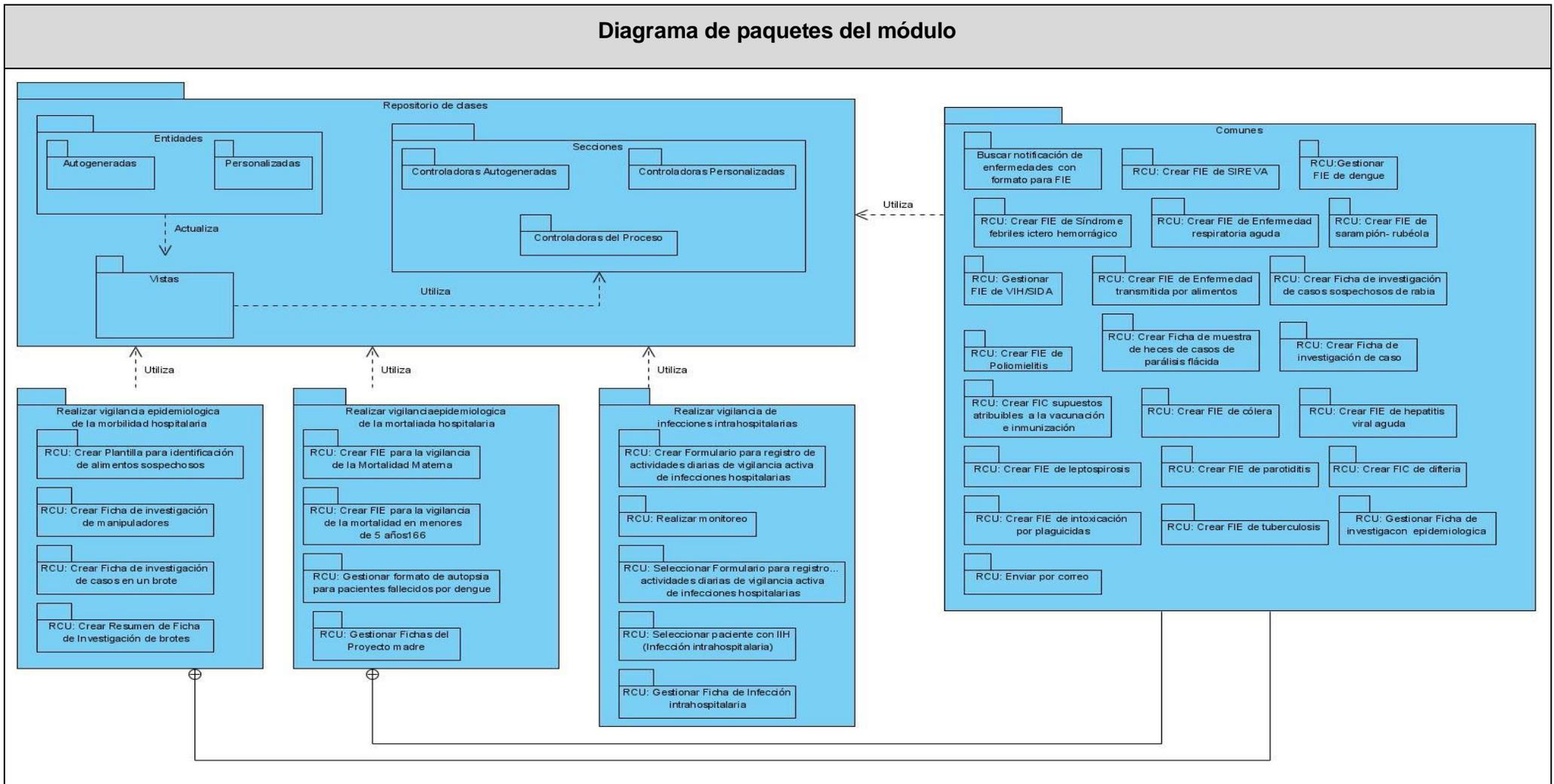
Un paquete referente a procesos, está conformado por subpaquetes que responden a las realizaciones de casos de uso, donde cada una de ellas contiene un diagrama de clases del diseño y los respectivos diagramas de secuencia.

El paquete repositorio contiene tres subpaquetes, uno para las entidades, uno para las sesiones y otro para las vistas. El subpaquete de entidades contiene las clases autogeneradas y personalizadas. El subpaquete de sesiones, a su vez, está conformado por las clases controladoras autogeneradas,

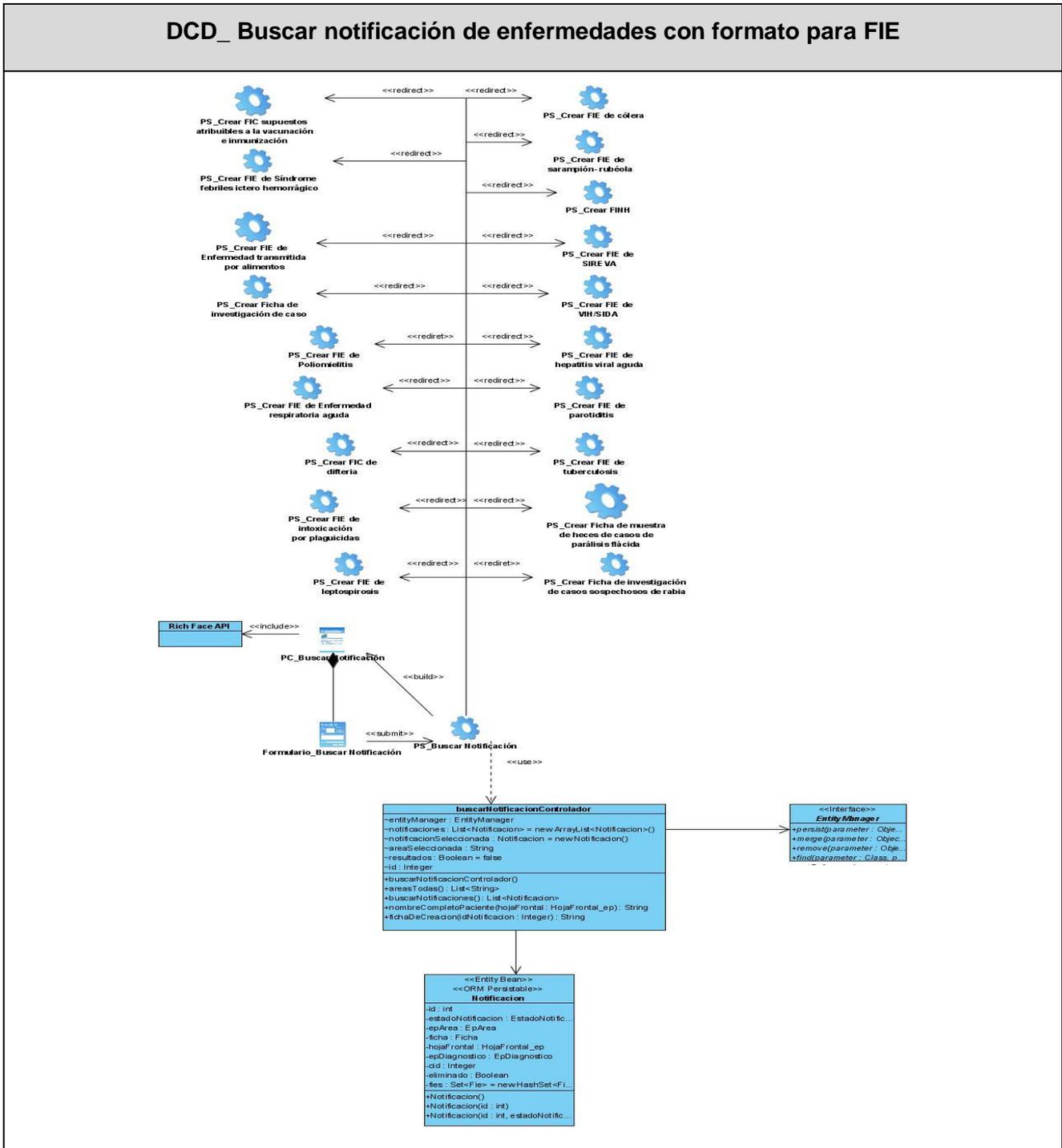
personalizas y del proceso. Finalmente, el subpaquete de las vistas, contiene los contenidos web referentes a las páginas clientes y los formularios que las componen.

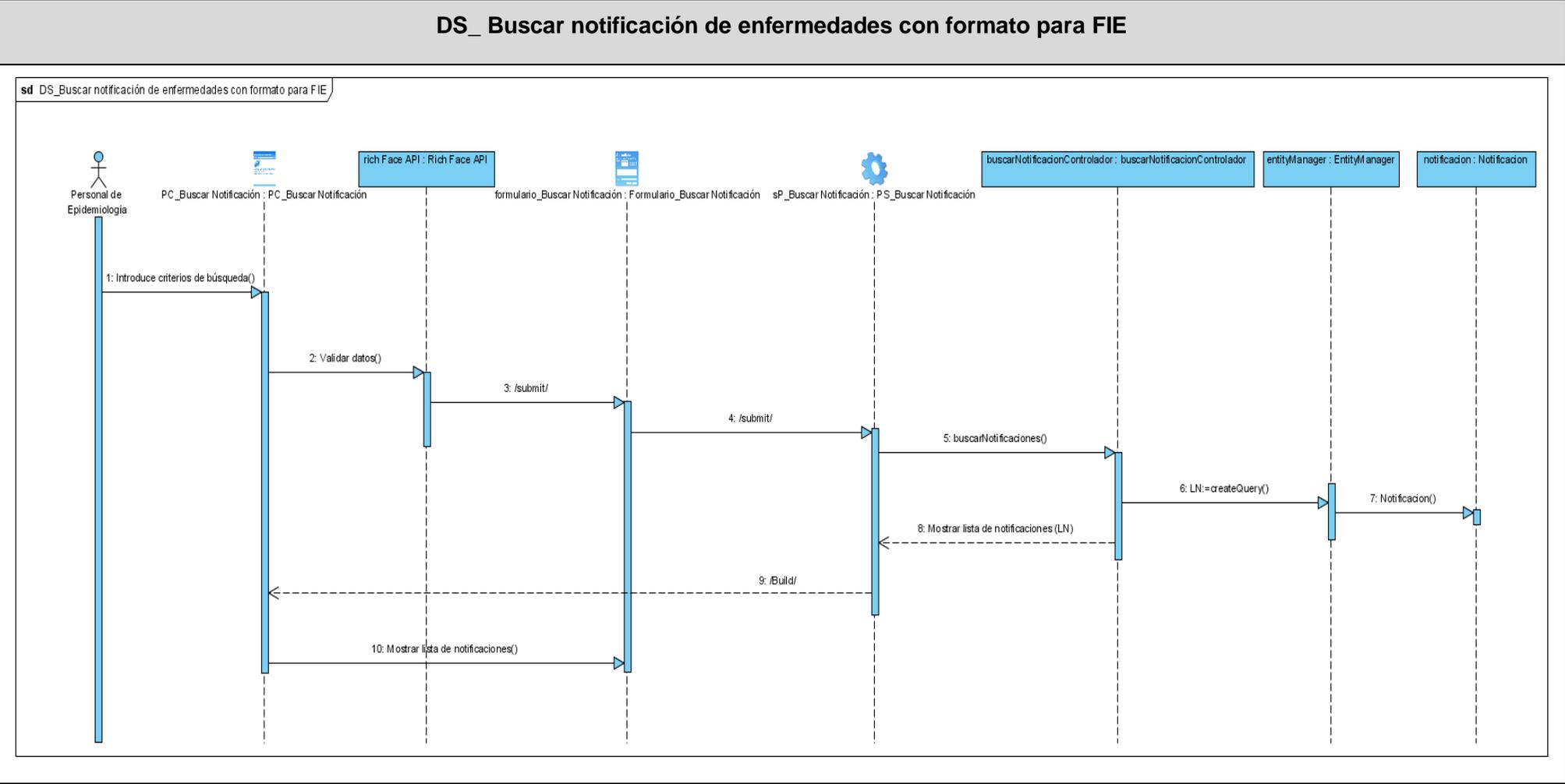
A continuación se presenta el diagrama de paquetes del sistema propuesto:

Diagrama de paquetes del módulo

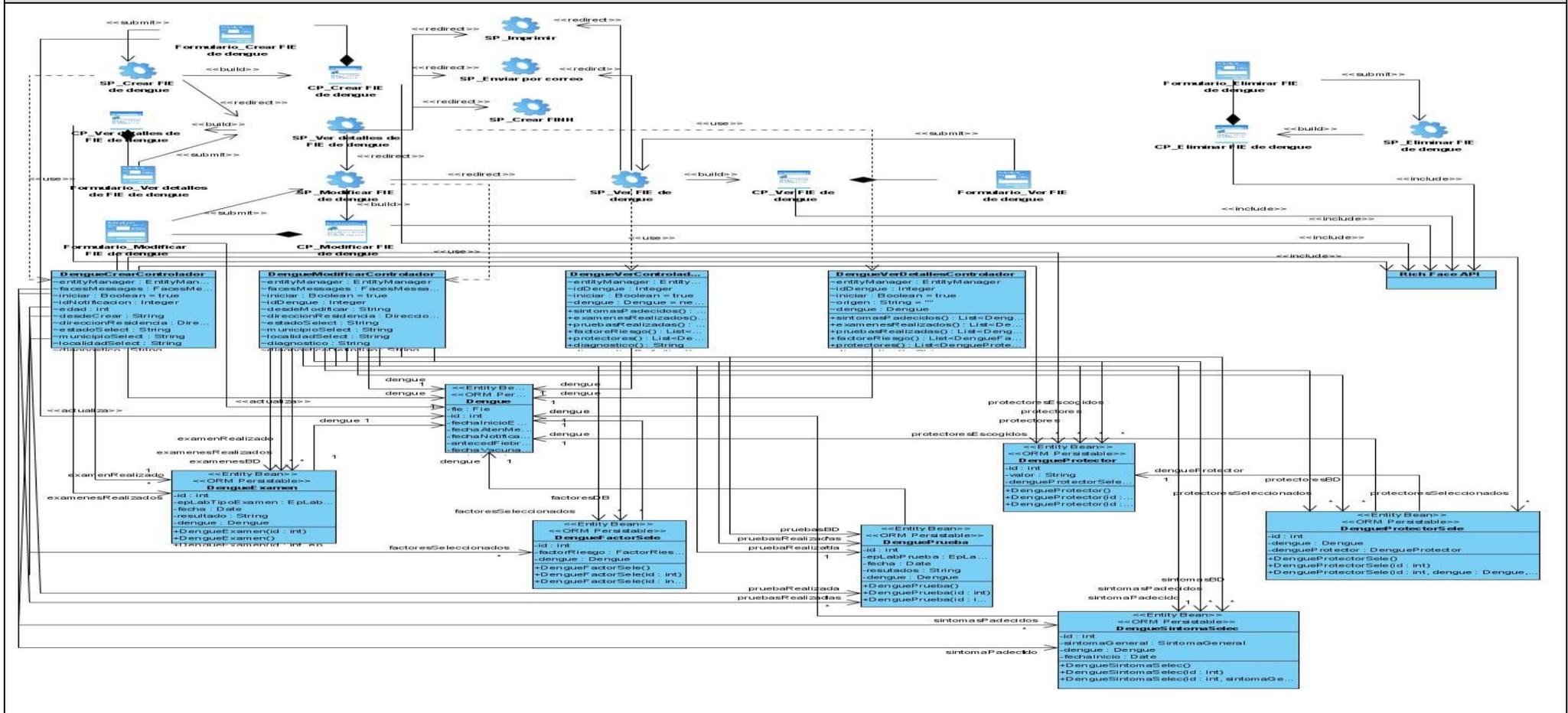


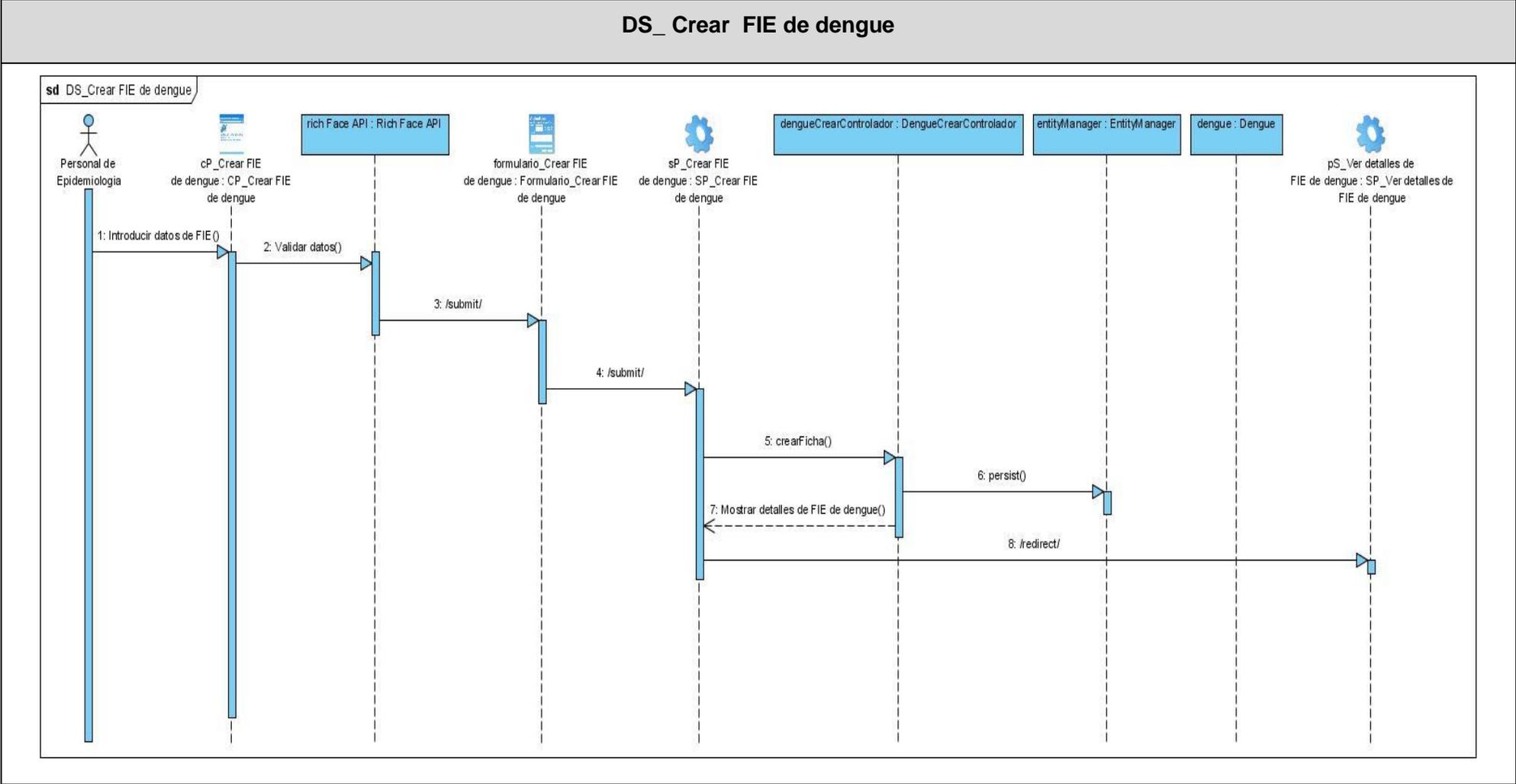
A continuación se muestra la realización de algunos casos de uso arquitectónicamente significativos del proceso Realizar vigilancia epidemiológica de la morbilidad hospitalaria.

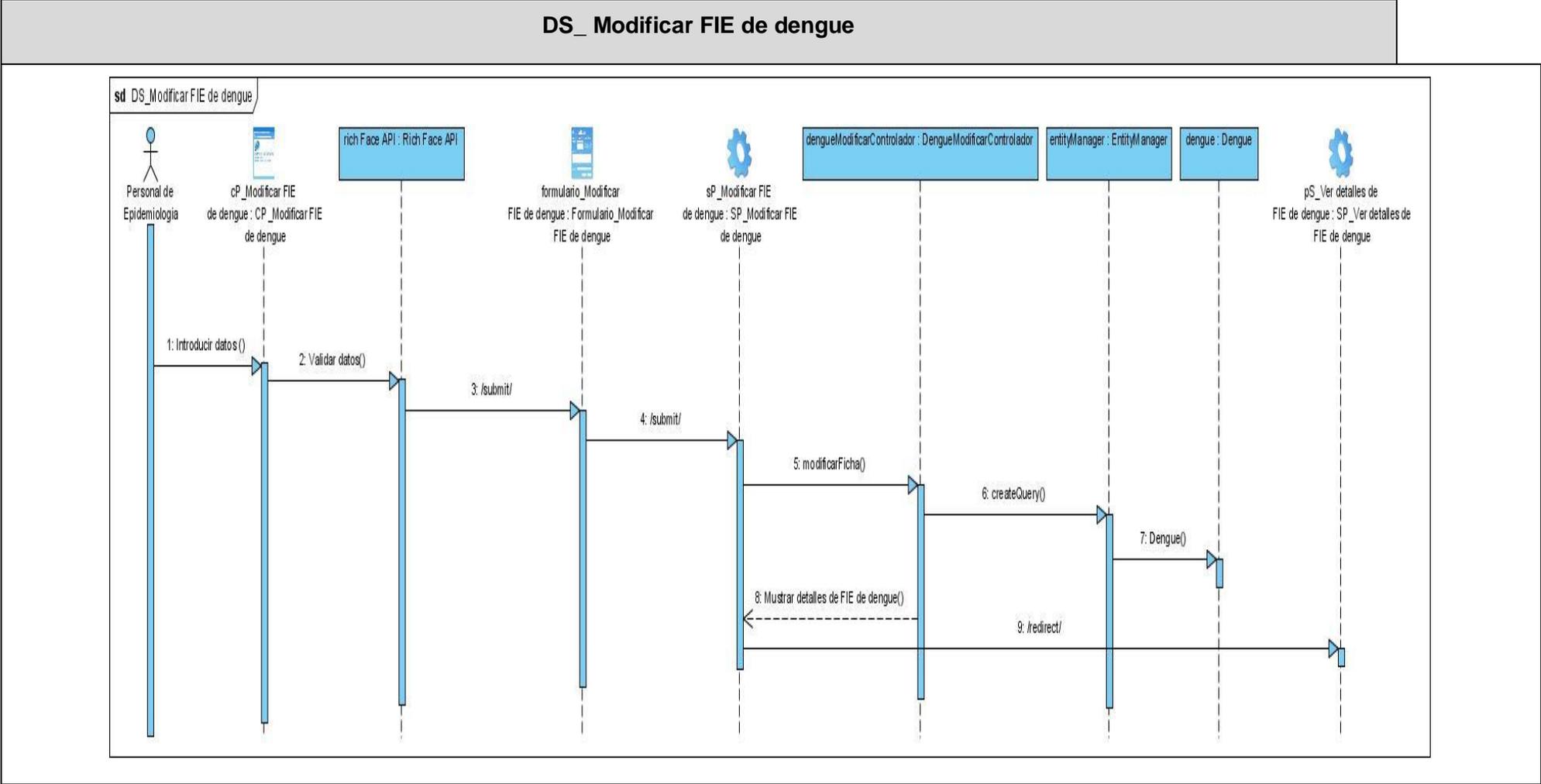


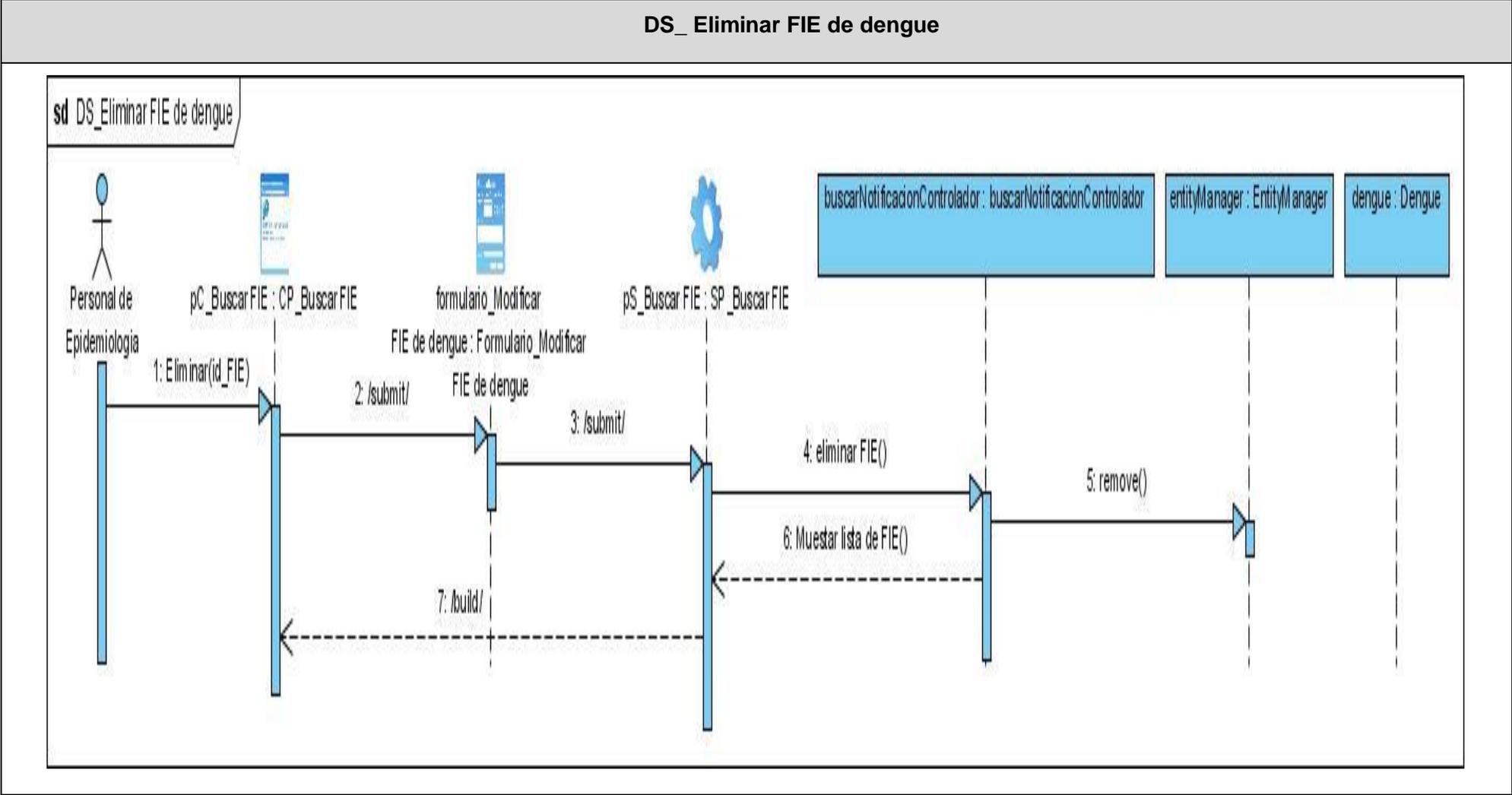


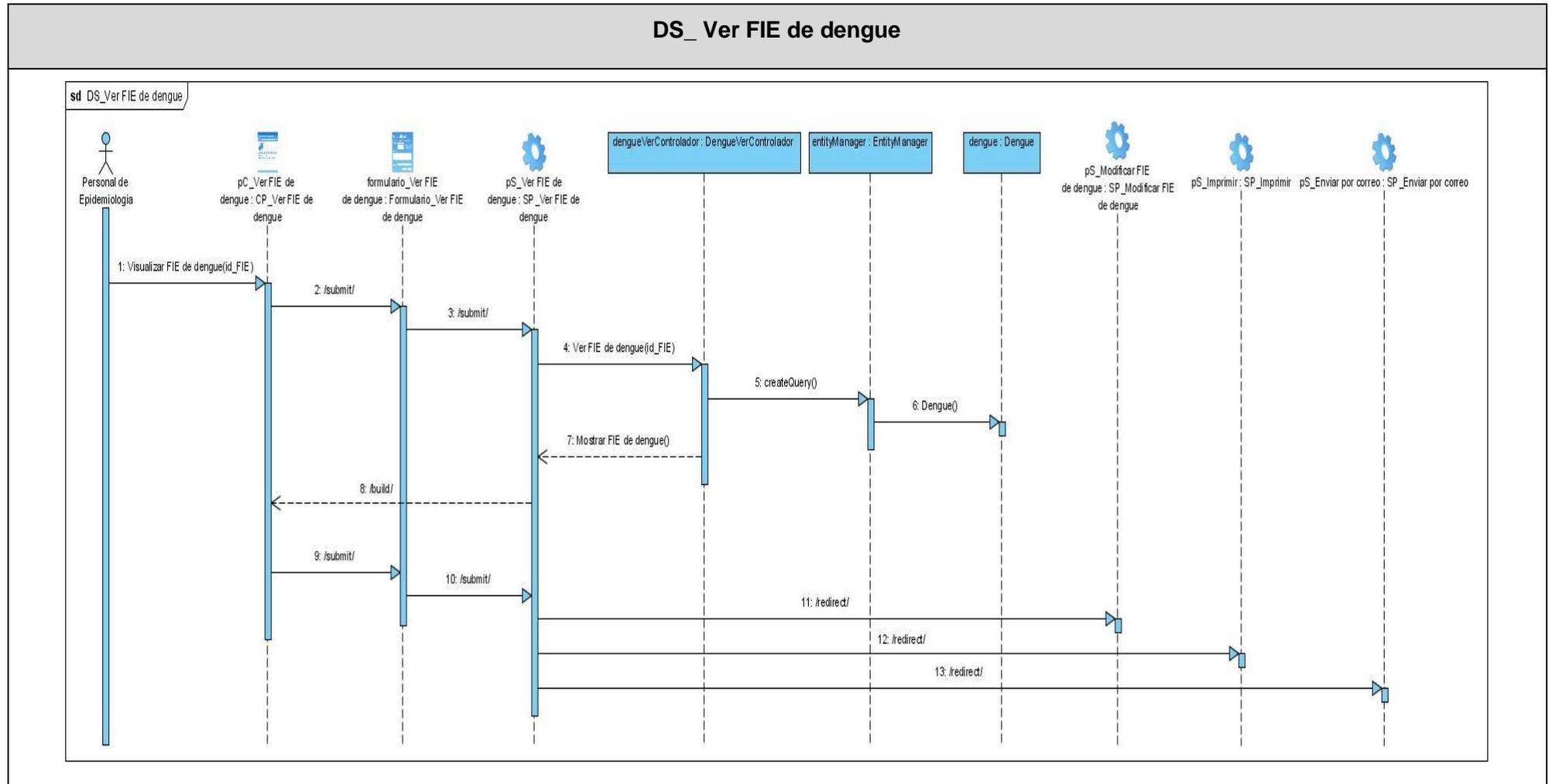
DCD_Gestionar FIE de dengue

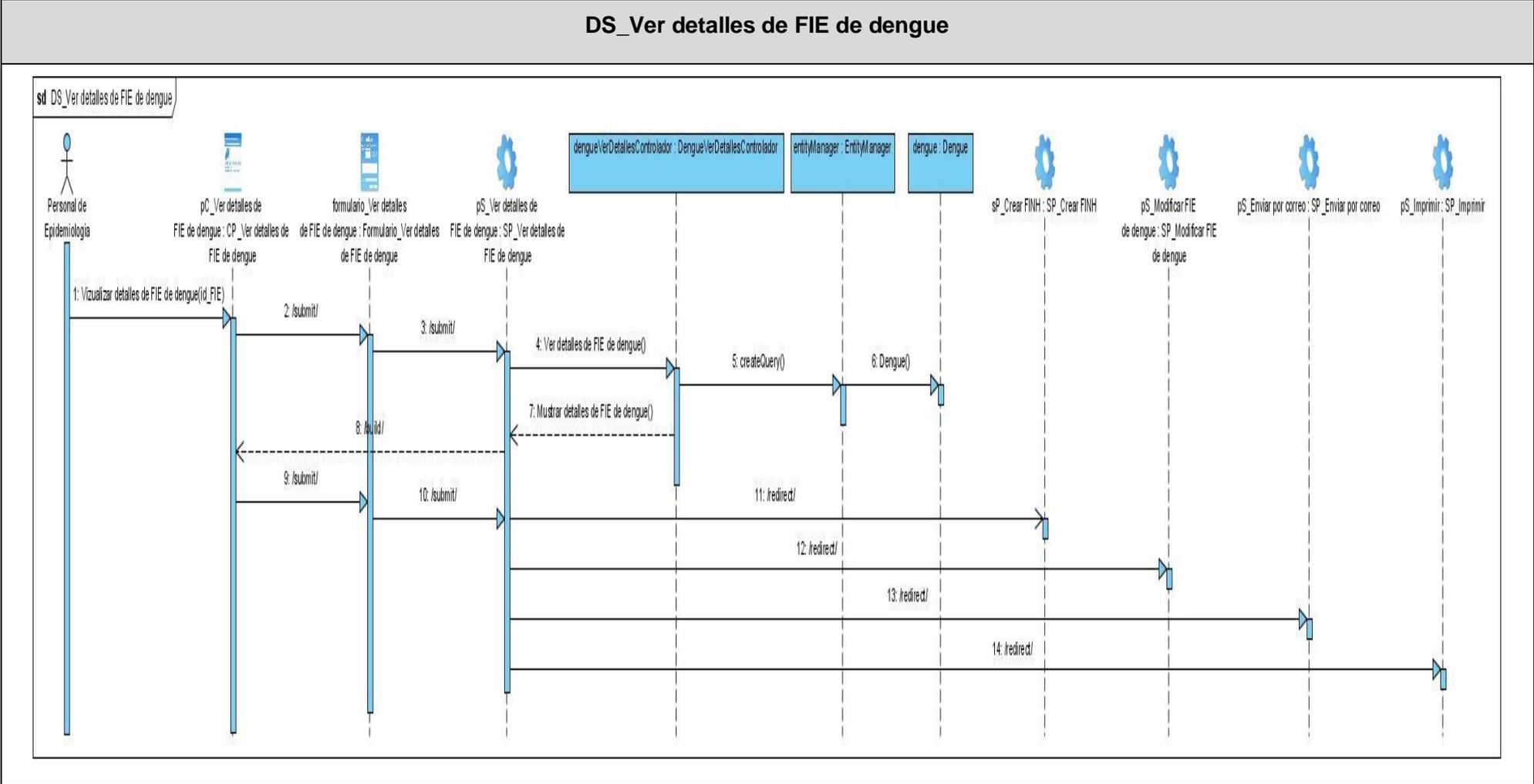


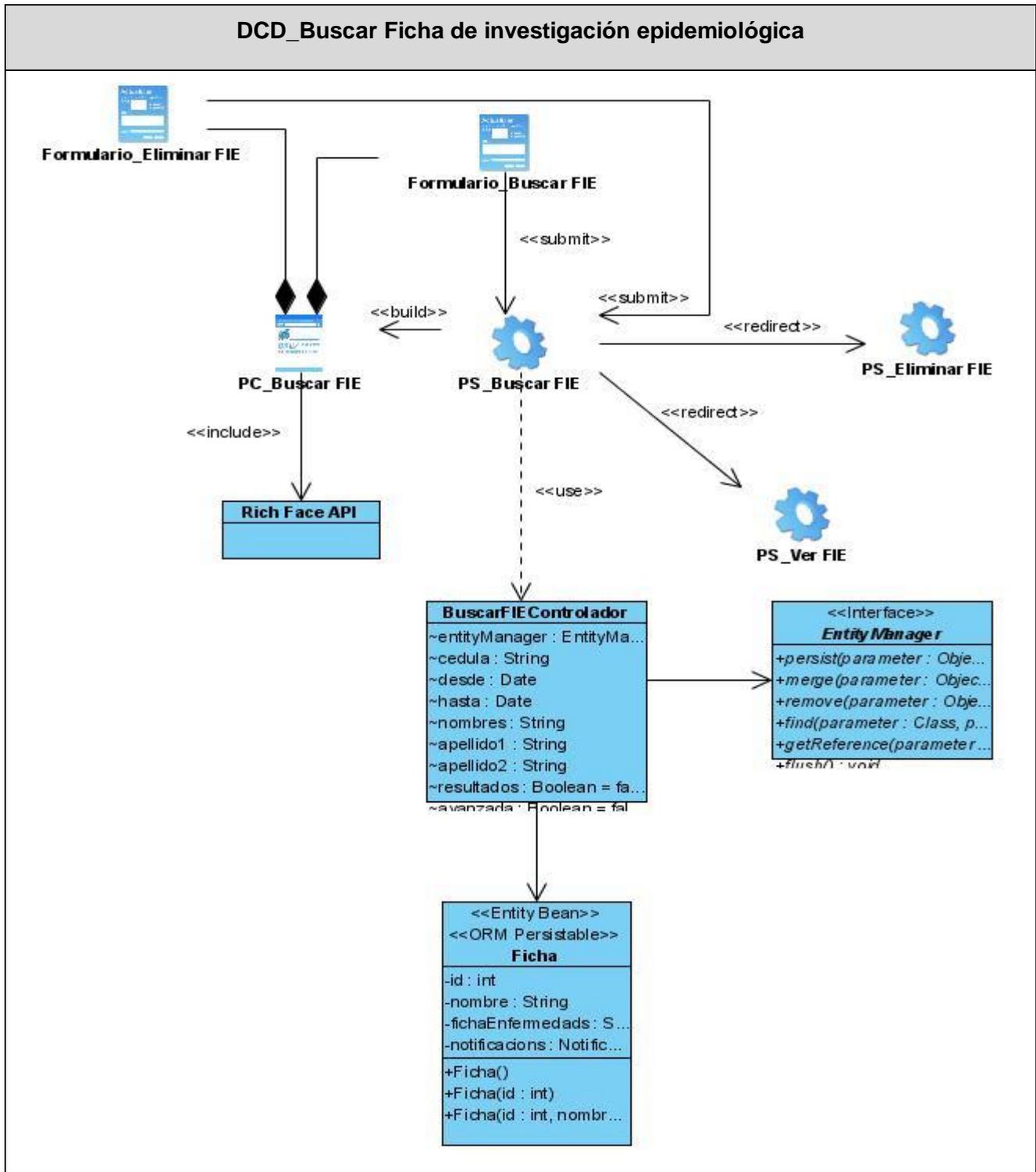




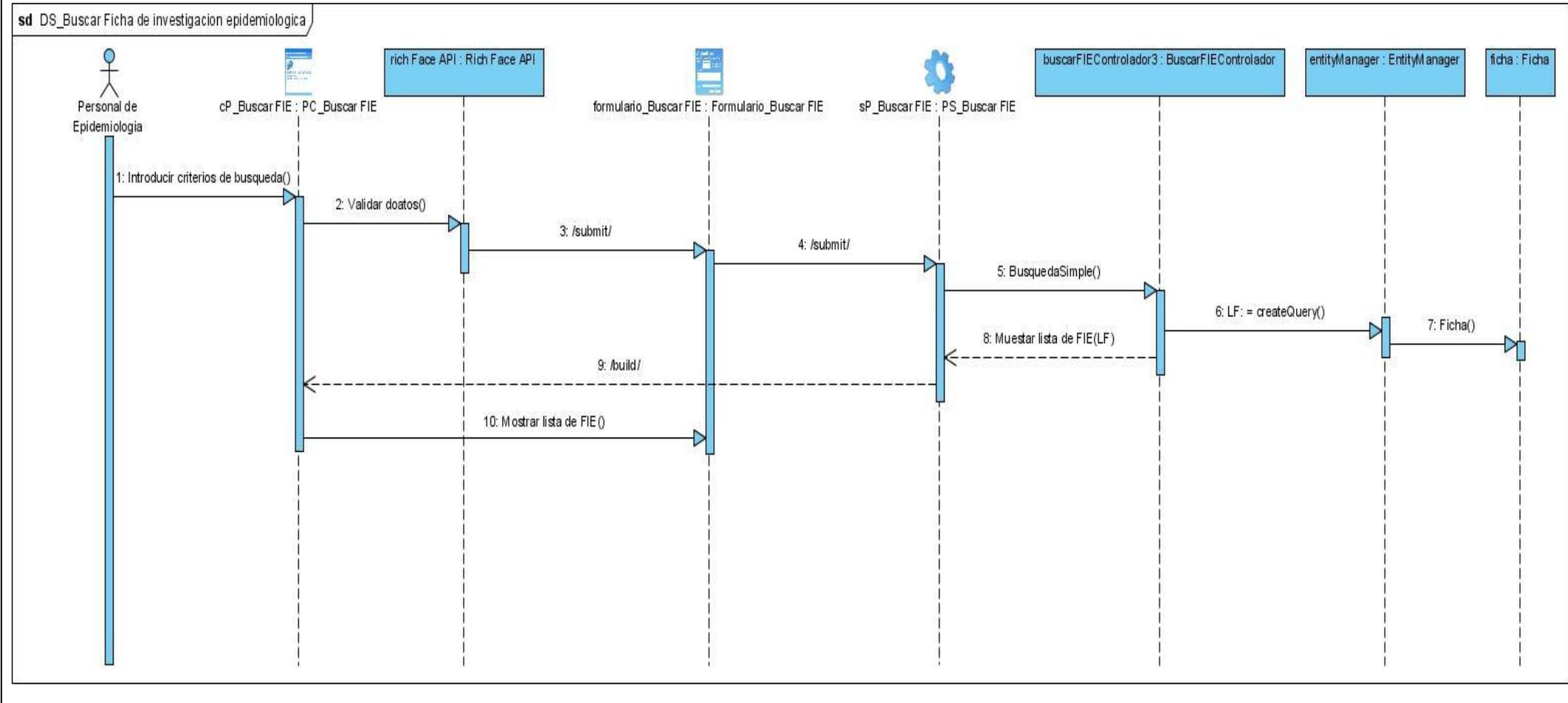








DS_Buscar Ficha de investigación epidemiológica



3.3 Descripción de las clases del diseño

Nombre: SP_Buscar Notificación	
Tipo de clase: Página servidora	
Descripción:	Es una página que se ejecuta del lado del servidor en la capa de Presentación. Recibe y valida los datos que se envían desde la página cliente. Codifica la información y construye las estructuras que serán enviadas a la capa de negocio. Se invoca al método del negocio para el registro de los nuevos datos, una vez concluida la ejecución de sus responsabilidades.

Nombre: Formulario_Buscar Notificación	
Tipo de clase: Formulario	
Descripción:	Esta contenido dentro de la página cliente. En él es donde se introducen los datos de búsqueda, los cuales serán enviados a la pagina servidora para ser procesados, y además muestra los resultados de la búsqueda.

Nombre: PC_Buscar Notificación	
Tipo de clase: Página cliente	
Descripción:	Es una página que se ejecuta del lado del cliente. Le permite al Personal de epidemiología introducir los criterios de búsqueda para buscar las notificaciones realizadas.

Nombre: DengueCrearControlador	
Tipo de clase: Controladora	

Para cada responsabilidad:	
Nombre:	dengueCrearControlador()
Descripción:	Constructor de la clase
Nombre:	diagnosticos() :List<String>
Descripción:	Devuelve los posibles diagnosticos parciales
Nombre:	diagnosticosDefinitivos() : List<String>
Descripción:	Devuelve los posibles diagnosticos definitivos
Nombre:	adicionarPruebaRealizada()
Descripción:	adiciona una prueba de laboratorio realizada
Nombre:	adicionarSintomaPadecido()
Descripción:	Adiciona un síntoma padecido con su fecha y descripción
Nombre:	adicionarExamenRealizado()
Descripción:	Adiciona un examen de laboratorio realizado
Nombre:	crearFicha()
Descripción:	Crea una ficha de Dengue y persiste todos sus datos en la base de datos

Nombre: DengueModificarControlador	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	cargarDatos()

Descripción:	Carga los datos de la ficha que se desea modificar
Nombre:	modificarFicha()
Descripción:	Modifica los datos de la ficha y persiste sus datos en la base de datos

Nombre: DengueVerControlador	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	cargarDatos()
Descripción:	Carga todos los datos de la ficha que se desea ver

Nombre: DengueVerDetallesControlador	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	cargarDatos()
Descripción:	Carga todos los datos de la ficha que se ha creado o se ha modificado

Nombre: PS_Imprimir	
Tipo de clase: Página servidora	
Descripción:	Es una página que se ejecuta del lado del servidor. Le permite al Personal de epidemiología imprimir la FIE.

Nombre: PS_Enviar por correo	
Tipo de clase: Página servidora	
Descripción:	Es una página que se ejecuta del lado del servidor. Le permite al Personal de epidemiología enviar por correo la FIE.

Como resultado del estudio realizado en este capítulo, correspondiente al flujo de diseño, se identificaron las clases fundamentales que deben ser definidas para que el sistema funcione satisfactoriamente. Asimismo fueron especificados los atributos y métodos que deben tener las clases para brindarle al desarrollador una idea clara de lo que se debe implementar. Además se obtuvo la realización de casos de uso por procesos, donde se elaboraron los diagramas de clases y los diagramas de secuencia correspondientes a cada escenario.

CAPÍTULO 4: IMPLEMENTACIÓN

En este capítulo, se describe el flujo de trabajo correspondiente a la implementación. Partiendo de los resultados logrados en el diseño, se obtiene el modelo de datos. Por otra parte se define el modelo de implementación, el cual está conformado por los diagramas de componentes y el diagrama de despliegue.

4.1 Modelo de datos

Un modelo de datos es "un conjunto de conceptos, reglas y convenciones que nos permiten describir y en ocasiones manipular los datos de un cierto mundo real, que deseamos almacenar en la base de datos"[33].

Una de las técnicas más utilizadas para el modelado de datos de un sistema de información es el modelado entidad relación mediante diagramas o modelos entidad relación (denominado por sus siglas, E-R "Entity relationship", o, "DER" Diagrama de Entidad Relación). Estos constituyen una representación conceptual de la información, mediante la cual se pretende "visualizar" los objetos que pertenecen a la Base de Datos como entidades (se corresponde al concepto de objeto de la Programación Orientada a Objetos) las cuales tienen atributos y se vinculan mediante relaciones.

A continuación se muestran algunos ejemplos de DER pertenecientes al proceso de morbilidad, principalmente asociados a las Fichas de Investigación Epidemiológica.

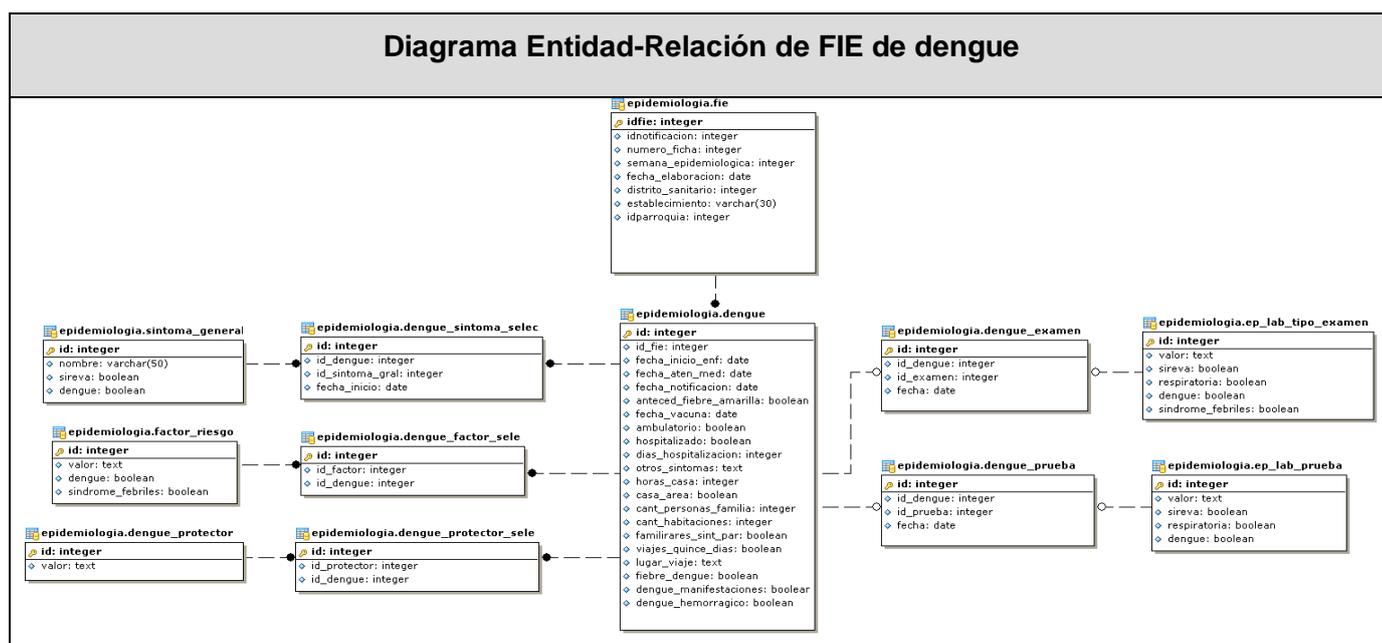
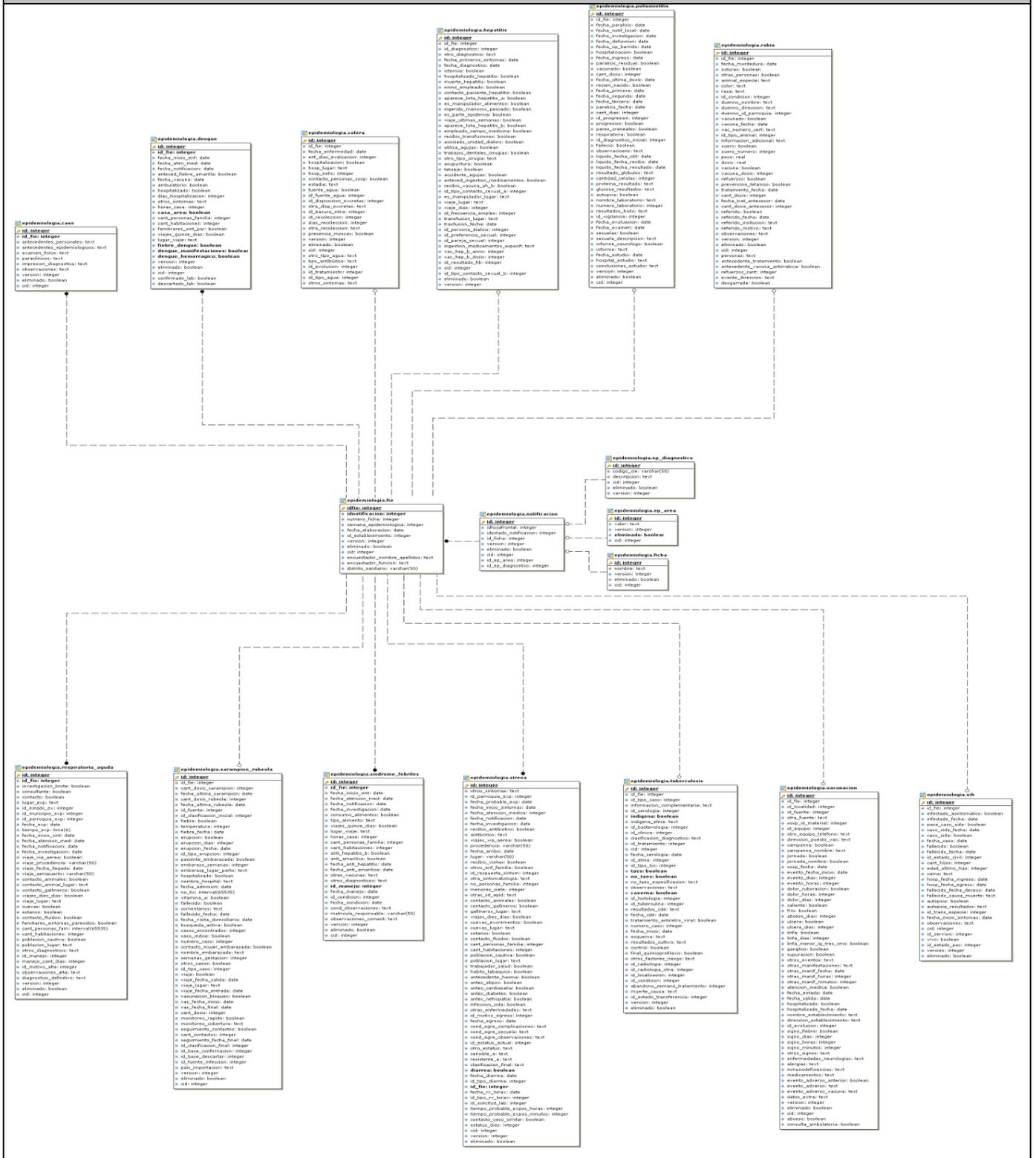
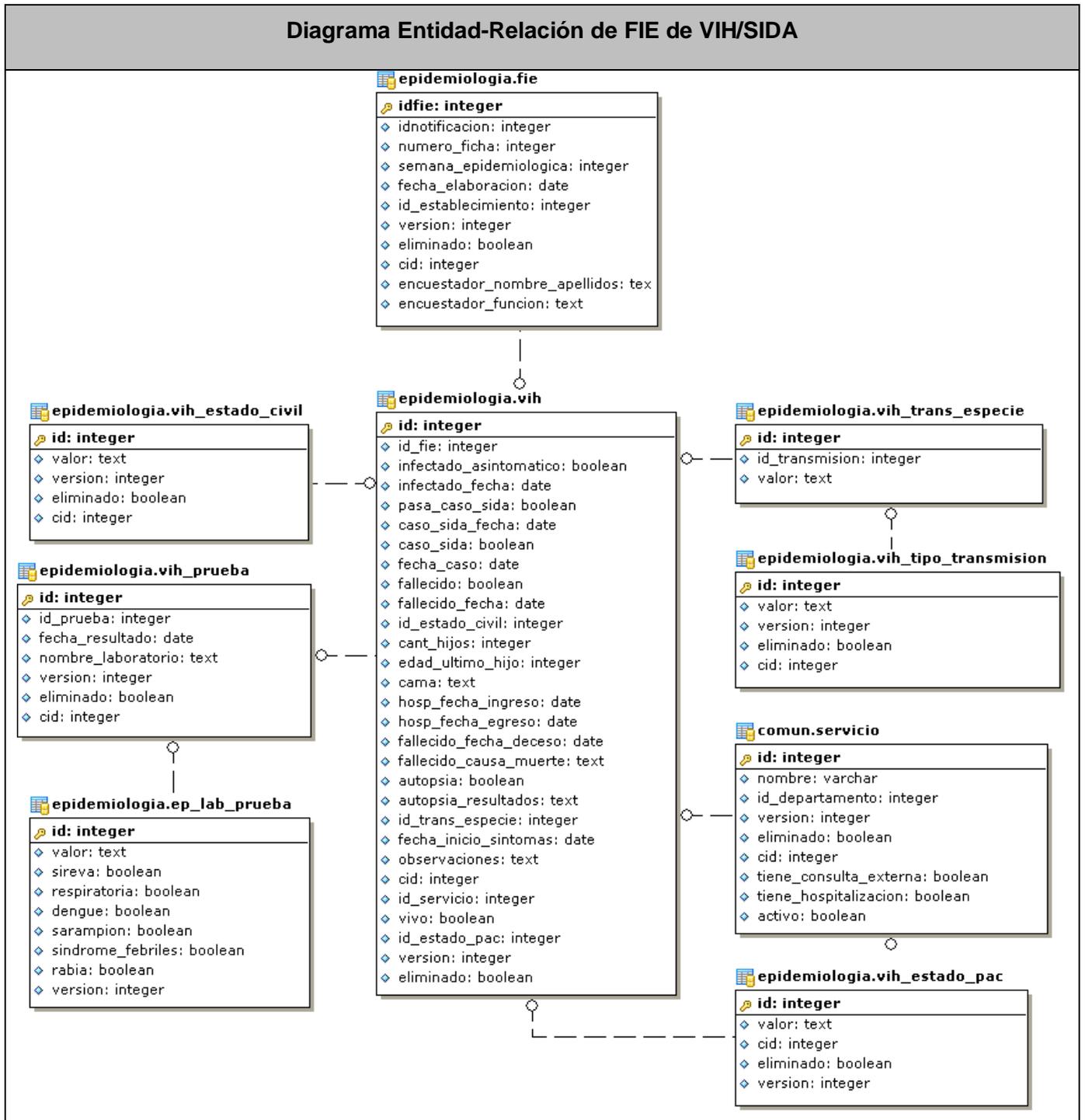


Diagrama Entidad-Relación de FIE





Descripción de las tablas.

Nombre: notificación		
Descripción: Almacena los datos generales de la notificación		
Atributo	Tipo	Descripción
id	integer	Es el # que identifica cada notificación
idestado_notificacion	integer	Es el # del estado que tiene la notificación
Id_ficha	integer	Es el # del tipo de ficha de investigación
id_ep_area	integer	Es el # del área de donde proviene la notificación
Id_ep_diagnostico	integer	Es el # del diagnóstico de la notificación

Nombre: fie		
Descripción: Almacena los datos generales de las Ficha de Investigación Epidemiológica		
Atributo	Tipo	Descripción
idfie	integer	Es el # que identifica a cada fie
idnotificacion	integer	Es el # de la notificación por el cual se generó la fie
numero_ficha	integer	Es el # que identifica a la fie
semana_epidemiologica	integer	Es el # de la semana en que fue creada la fie.
fecha_elaboracion	date	Fecha en que elaboró la fie
distrito_sanitario	integer	El # del distrito sanitario en el cual se creó la fie
establecimiento	Varchar(30)	Nombre del establecimiento en el cual se creó la fie

idparroquia	integer	El # de la parroquia en la cual se creó la fie
-------------	---------	--

Nombre: caso		
Descripción: Almacena los datos generales de las Ficha de Investigación de Caso		
Atributo	Tipo	Descripción
id	integer	Es el # que identifica a la FIC
id_fie	integer	Es el # que la identifica como fie.
antecedentes_personales	text	Los antecedentes personales del paciente
antecedentes_epidemiologicos	text	Los antecedentes epidemiológicos del paciente
examen_fisico	text	Descripción de los exámenes físicos
paraclinicos	text	Descripción de los exámenes paraclínicos

Nombre: vih		
Descripción: Almacena todos los datos generales referente a las FIE de VIH creada a los pacientes.		
Atributo	Tipo	Descripción
id	integer	Es el # que identifica a la fie de VIH
id_fie	integer	Es el # que la identifica como fie.
infectado_asintomatico	boolean	Si el paciente presenta o no síntomas
infectado_fecha	date	Fecha en que fue infectado
pasa_caso_sida	boolean	Si es un caso de sida o no

caso_sida_fecha	date	Fecha en que fue diagnosticado como un caso de sida
caso_sida	boolean	Si es un caso de sida o no
fecha_caso	date	Fecha en que se identifico el caso
fallecido	boolean	Si el paciente es fallecido o no
fallecido_fecha	date	Fecha en que falleció el paciente
#id_estado_civil	integer	Es el # que identifica si en paciente es casado o no
cant_hijos	integer	La cantidad de hijos que tiene el paciente
edad_ultimo_hijo	integer	La edad del menor de sus hijos
cama	text	La cama en que está hospitalizado
hosp_fecha_ingreso	date	Fecha en que fue ingresado
hosp_fecha_egreso	date	Fecha en que fue dado de alta el paciente
fallecido_fecha_deceso	date	Fecha en que falleció el paciente
fallecido_causa_muerte	text	Causa de muerte
autopsia	boolean	Si se le hizo o no la autopsia
autopsia_resultados	text	Resultados de la autopsia
fecha_inicio_sintomas	date	Fecha en que aparecieron los primeros síntomas de la enfermedad
observaciones	text	Observaciones realizadas
id_servicio	integer	Es el # que identifica al servicio que se le dio al paciente

vivo	boolean	Si el paciente está vivo o no
id_estado_pac	integer	Es el # que identifica el estado del paciente

Nombre: dengue		
Descripción: Almacena todos los datos generales referente a las FIE de dengue creada a los pacientes.		
Atributo	Tipo	Descripción
id	integer	Es el # que identifica a la fie de dengue
id_fie	integer	Es el # que la identifica como fie.
fecha_inicio_enf	date	Fecha de inicio de la enfermedad
fecha_aten_med	date	Fecha en que fue atendido por la enfermedad
fecha_notificacion	date	Fecha en que fue notificado con la enfermedad
anteced_fiebre_amarilla	boolean	Si tiene o no antecedentes de fiebre amarilla
fecha_vacuna	date	Fecha en que fue vacunado contra la enfermedad
ambulatorio	boolean	Si está de ingreso o no en el hogar
hospitalizado	boolean	Si está hospitalizado o no
días_hospitalizacion	integer	Cantidad de días que estuvo hospitalizado
otros_sintomas	text	Otros síntomas que se presentaron
horas_casa	integer	Cantidad de horas que estuvo en la casa
cant_personas_familia	integer	Cantidad de personas del núcleo familiar

cant_habitaciones	integer	Cantidad de habitaciones del hogar
familiares_sint_par	boolean	Si algún familiar presenta síntomas o no
viajes_quince_dias	boolean	Si ha viajado en los últimos 15 días en que diagnosticado con la enfermedad
lugar_viaje	text	Lugares que visitó antes de ser diagnosticado con la enfermedad
fiebre_dengue	boolean	Si ha tenido fiebre producto a la enfermedad o no
dengue_manifestaciones	boolean	Si ha tenido alguna manifestación producto a la enfermedad
dengue_hemorragico	boolean	Si es dengue hemorrágico o no

4.2 Implementación

El flujo de trabajo de implementación detalla cómo los elementos del modelo del diseño se implementan en términos de componentes y representa cómo se organizan en el modelo de despliegue. Permite obtener los diagramas de despliegue y componentes los cuales son artefactos que conforman lo que se conoce como un modelo de implementación.

Diagrama de despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes *hardware* y *software* en el sistema final [34]. Es un grafo de nodos unidos por conexiones de comunicación, donde un nodo es un objeto físico en tiempo de ejecución que representa un recurso computacional, generalmente con memoria y capacidad de procesamiento.

Los nodos se utilizan para modelar la topología del hardware sobre el que se ejecuta el sistema. Representan típicamente un procesador o un dispositivo sobre el que se pueden desplegar y ejecutar los componentes, los cuales son los elementos que participan en la ejecución de un sistema [35].

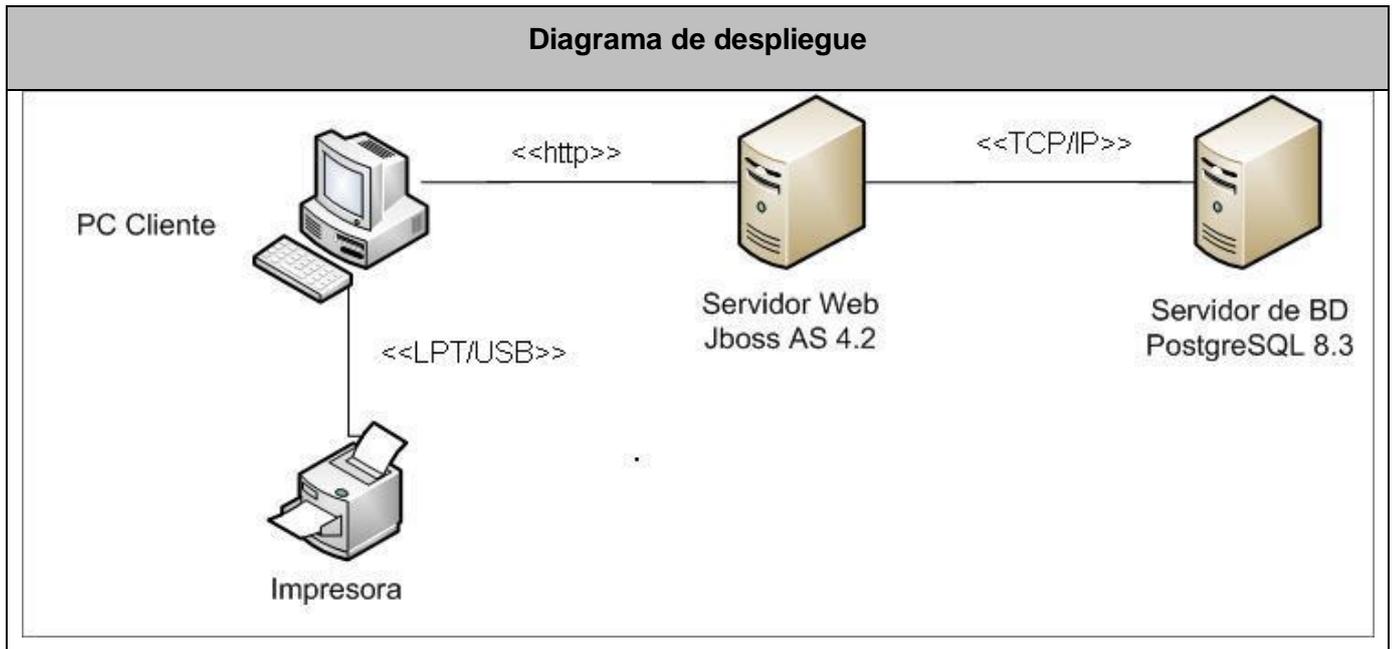
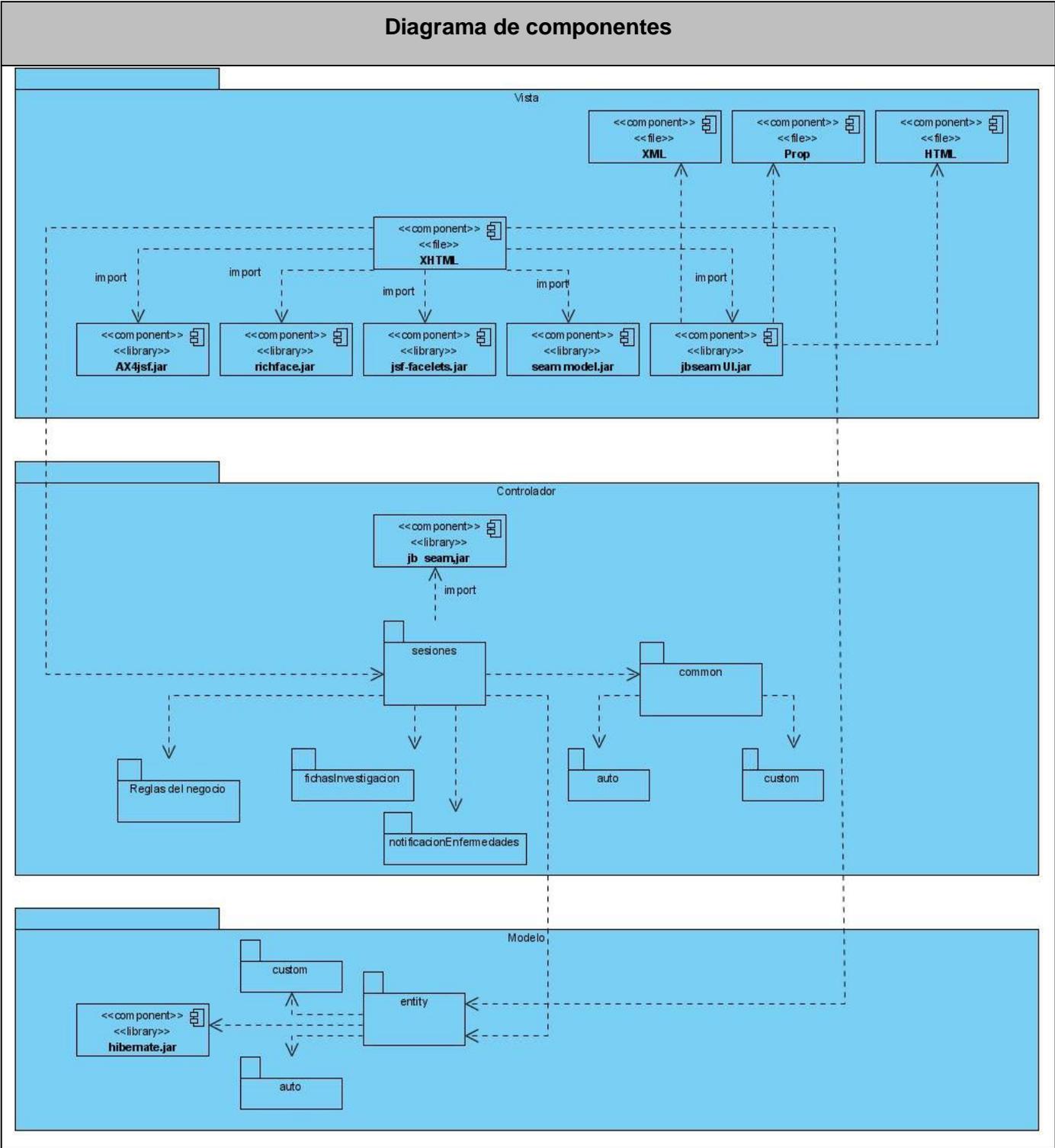


Diagrama de componentes.

Los diagramas de componentes se utilizan para describir la vista de implementación estática de un sistema [36]. Estos muestran las organizaciones y dependencias lógicas entre componentes software. Un componente es una parte modular de un sistema, desplegable y reemplazable que encapsula implementación y un conjunto de interfaces y proporciona la realización de los mismos, es decir, representan todos los tipos de elementos de software que entran en la fabricación de aplicaciones informáticas. Estos pueden ser de código fuente, binarios o ejecutables.

Desde el punto de vista del diagrama de componentes, se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del *software*, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo.

Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes. Un paquete en un diagrama de componentes representa una división física del sistema. Los paquetes se organizan en una jerarquía de capas donde cada capa tiene una interfaz bien definida [37].



4.3 Tratamiento de errores

Una excepción es un suceso o evento en tiempo de ejecución que puede causar que una funcionalidad fracase, las mismas se alcanzan cuando un error u otro evento interrumpen la ejecución normal de un programa.

El tratamiento de excepciones es un mecanismo que transfiere el control de un programa en el punto donde ocurre la excepción al manipulador adecuado, posibilitando que continúe la ejecución del mismo aún en presencia de un error.

En el sistema que se propone, el control de las excepciones se lleva a cabo en todas las porciones de código donde pueda surgir alguna situación inesperada, especialmente donde se ejecutan sentencias que manipulan los datos que viajan desde y hacia la base de datos. También se controlan los errores que pueden surgir en la validación de algunos datos provenientes de la interfaz de usuario, puesto que encierran una lógica compleja en cierta medida.

Existe un archivo XML, denominado `page.xml`, que engloba la configuración de todos los mensajes que se deben mostrar por cada tipo de excepción, así como la página a la que el sistema redirecciona en caso de la aparición de un error sorpresivo.

Por otra parte, se hace uso de un componente del *framework Seam*, para capturar los mensajes que surgen en el código de las clases controladoras, se trata del *FacesMessages*, el cual permite mostrar dichos mensajes directamente en la interfaz de usuario, los cuales no sólo provienen de errores, sino también de notificaciones y otras situaciones de interés para quien interactúe con la aplicación.

4.4 Seguridad

Como parte de la seguridad del sistema que se propone, se concibe un control de acceso a nivel de usuarios y contraseñas, incluyendo la diferenciación de tipos de usuario, logrando así la visibilidad sólo a las áreas establecidas de acuerdo a la función que realizan. Las contraseñas sólo podrán ser cambiadas por el usuario o por el administrador del sistema. Por otra parte, el sistema permitirá llevar una traza de todas las operaciones llevadas cabo por cada usuario mediante un registro de actividades en todo momento, a través de una bitácora de sucesos.

Otro de los aspectos que se tiene en cuenta, es la fidelidad de los datos. Esta se mantiene mediante el cifrado de la información proveniente de la comunicación entre los componentes internos del sistema y

otros sistemas que soliciten información desde otro hospital. Esto asegura la información pues impide la lectura o modificación de los datos confidenciales que se manejan.

4.5 Estrategias de codificación. Estándares y estilos a utilizar

El objetivo fundamental de los estándares de codificación, es proporcionar un estilo de programación homogéneo, con el fin de que todos los desarrolladores de un proyecto trabajen de forma coordinada, y que puedan entender el código en menos tiempo, como si un único programador hubiera escrito todo el código de una sola vez.

Un aspecto importante que repercute directamente en el hecho de que un programador comprenda un sistema de software, es sin duda la legibilidad del código fuente. Cuando esto ocurre, se pueda lograr el mantenimiento del código, que es la facilidad con que el sistema de software puede modificarse para añadirle nuevas características, modificar las ya existentes, depurar errores, o mejorar el rendimiento.

Si se aplica de forma continuada un estándar de codificación bien definido, se utilizan técnicas de programación apropiadas y posteriormente, se efectúan revisiones del código de rutinas, caben muchas posibilidades de que un proyecto de software se convierta en un sistema de software fácil de comprender y de mantener.

Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código de alta calidad es de gran importancia para la calidad del software y para obtener un buen rendimiento.

Para el desarrollo del sistema propuesto se utilizan varios estándares de codificación, tales como:

Notación Camell: Se utiliza para denotar variables y parámetros. En esta notación, si el identificador es una palabra simple se escribe todo con minúscula, pero si es compuesta, la primera letra de todas las palabras que viene a continuación de la primera comienza con mayúscula.

A continuación se muestran algunas restricciones de la nomenclatura del código haciendo uso de los estándares de codificación mencionados anteriormente:

Identación.	
Inicio y fin de bloque	Se recomienda dejar dos espacios en blanco desde la instrucción anterior para el inicio y fin de bloque {}. Lo mismo sucede para el caso de las instrucciones if,

	else, for, while, do while, switch, foreach.
Aspectos generales	<p>El indentado debe ser de dos espacios por bloque de código. No se debe usar el tabulador; ya que este puede variar según la computadora o la configuración de dicha tecla.</p> <p>Los inicios ({) y cierre (}) de ámbito deben estar alineados debajo de la declaración a la que pertenecen y deben evitarse si hay sólo una instrucción.</p> <p>Nunca colocar { en la línea de un código cualquiera, esto requiere una línea propia.</p>

Comentarios, separadores, líneas, espacios en blanco y márgenes.		
Ubicación de comentarios	Al inicio de cada clase o función y al final de cada bloque de código.	Se recomienda comentar al inicio de la clase o función especificando el objetivo de la misma así como los parámetros que usa (especificar tipos de dato, y objetivo del parámetro) entre otras cosas.
Líneas en blanco	Se emplean antes y después de métodos, clases y estructuras.	Se recomienda dejar una línea en blanco antes y después de la declaración de una clase o de una estructura y de la implementación de una función
Espacios en blanco	Entre operadores lógicos y aritméticos.	Se recomienda usar espacios en blanco entre estos operadores para lograr una mayor legibilidad en el código. Ejemplo: producto = nomproducto
Aspectos generales	Sobre el comentario	Se debe evitar comentar cada línea de código. Cuando el comentario se aplica a un grupo de instrucciones debe estar seguido de una línea en blanco. En caso de que se necesite comentar una sola instrucción se suprime la línea en blanco o se escribe a continuación de la instrucción

	Sobre los espacios en blanco	<p>No se debe usar espacio en blanco:</p> <p>Después del corchete abierto y antes del cerrado de un array.</p> <p>Después del paréntesis abierto y antes del cerrado.</p> <p>Antes de un punto y coma.</p>
--	------------------------------	--

Variables y constantes.		
Apariencia de constantes	Todas sus letras en mayúscula	Se deben declarar las constantes con todas sus letras en mayúscula.
Aspectos generales	Nombres de las variables y constantes	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito de la misma.

Clases y Objetos.		
Apariencia de clases y objetos	Primera letra en mayúscula	Los nombres de las clases deben comenzar con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación Pascal Ejemplo: MiClase(). Para el caso de las instancias se comenzara con un prefijo que identificara el tipo de dato, este se escribirá en minúscula.
Apariencia de atributos	Primera letra en minúscula	El nombre que se le da a los atributos de las clases debe comenzar con la primera letra en minúscula, la cual estará en correspondencia al tipo de dato al que se refiere, en caso de que sea un nombre compuesto se empleará notación Camello.
Declaración de	Agrupados por tipos	Los parámetros que se le pasan a las funciones

parámetro en funciones	Poner los string 1 numéricos 2, además, agrupar según valores por defecto.	se recomienda sean declarados de forma tal que estén agrupados por el tipo de dato que contienen, especificando el tipo de datos (tabla 51.2).
Aspectos generales.	Sobre las clases, los objetos, los atributos y las funciones.	El nombre empleado para las clases, objetos, atributos y funciones debe permitir que con sólo leerlo se conozca el propósito de los mismos.

Bases de datos, tablas, esquemas y campos		
Apariencia de la base de datos.	Las 2 primeras letras representan el tipo.	Los nombres de las Bases de Datos deben comenzar con el prefijo bd a continuación underscore y luego el nombre completamente en minúscula, en caso de que sea un nombre compuesto se empleará notación. Los nombres serán cortos y descriptivos. Ejemplo: bd_balancematerial.
Apariencia de las vistas	Las 2 primeras letras representan el tipo. Todas las letras en minúscula	El nombre a emplear para las vistas deben comenzar con el prefijo vt seguido de underscore y el nombre debe escribirse con todas las letras en minúscula para evitar problemas con el Case Sensitive del gestor. Ejemplo: create view 'vt_finanzas';
Apariencia de las tablas	Las 2 primeras letras representan el tipo. Todas las letras en minúscula.	El nombre a emplear para las tablas debe comenzar con el prefijo tb seguido de underscore y luego debe escribirse todas las letras en minúscula, en caso de que sea un nombre compuesto se utilizara underscore para

		<p>separarlo.</p> <p>Ejemplo: 'tb_producto';</p>
<p>Tablas que representen Relaciones</p>	<p>Las 2 primeras letras representan el tipo. Todas las letras en minúscula.</p>	<p>El nombre a emplear para estas tablas de relación debe comenzar con el prefijo tr seguido de underscore y el nombre de será la concatenación del nombre de las dos tablas que la generaron separados por underscore todo en minúscula.</p> <p>Ejemplo: 'tr_paciente_enfermedad'</p>
<p>Tablas que representen nomencladores.</p>	<p>Las 2 primeras letras representan el tipo. Todas las letras en minúscula.</p>	<p>El nombre a emplear para estas tablas de relación debe comenzar con el prefijo tn seguido de underscore. El nombre de será corto y descriptivo todo en minúscula.</p> <p>Ejemplo: 'tn_color_piel'</p>
<p>Apariencia de los procedimientos almacenados.</p>	<p>Las 2 primeras letras representan el tipo. Todas las letras en minúscula.</p>	<p>El nombre a emplear para los procedimientos debe comenzar con el prefijo pa seguido de underscore y luego debe escribirse todas las letras en minúscula en caso de que sea un nombre compuesto se utilizara underscore para separarlo.</p> <p>Ejemplo:</p> <p>pa _ paciente_especialidad.</p>
<p>Apariencia de los campos</p>	<p>Todas las letras en minúscula.</p>	<p>El nombre a emplear para los campos debe escribirse con todas las letras en minúscula para evitar problemas con el Case Sensitive del gestor.</p>

		Ejemplo: 'id_producto';
Nombre de los campos	En caso de identificadores.	Todos los campos identificadores van a comenzar con el identificador id seguido de underscore y posteriormente el nombre del campo Ejemplo: id_municipio.
Sentencias SQL	Todas las letras en mayúscula.	Las palabras correspondientes a las sentencias SQL y sus parámetros deben ir en mayúsculas.
Aspectos generales	Sobre las BD, vistas, tablas atributos y procedimientos.	El nombre empleado para las Bases de Datos, las vistas, las tablas, los campos y los procedimientos almacenados, deben permitir que con sólo leerlos se conozca el propósito de los mismos.

Controles		
Apariencia de los controles.	Los controles tendrán un prefijo para el tipo de datos en minúscula.	El nombre que se le da a los controles deben comenzar con las primeras letras en minúscula, las cuales identificarán el tipo de datos al que se refiere (ver tabla 1.2), en caso de que sea un nombre compuesto se empleará notación CamellCasing**. Ejemplo: btnAceptar

Tipo Datos	Prefijo	Ejemplo
Int	i	iCantPacientes
flota	f	fPesoPaciente
double	d	dPesoCarro
bool	b	bPacienteActivo
string	s	sNombrePaciente
char	c	cLetra
De tipo enum	e	eSexo
byte	b	bCantDiasPaciente
sbyte	sb	sbEdadPaciente
short	sh	shVariableShort
ushort	us	usVariableUshort
uint	ui	uiVariableUint
long	l	lVariableLong
ulong	ul	ulVariableUlong
decimal	dc	dcVariableDecimal
Objetos	o	oPacienteHistorico
Objetos de tipo Struct	st	stUnaStruct

Control	Prefijo	Ejemplo
Botón	btn	btnAceptar
Etiqueta	lbl	lblNombre
Lista/Menú	mn	mnPrincipal
Campo de Texto	txt	txtFecha
Botón de Opción	bpt	optSexo
Casilla de Verificación	chx	chxBorrar
Casilla de Selección	cbx	cbxSexo

En este capítulo, se obtuvo el modelo de datos correspondiente a los procesos que se desarrollan como parte de la investigación. Este constituyó uno de los puntos de entrada para la realización de las actividades concernientes al flujo de trabajo de implementación. De esta forma, se elaboraron los artefactos correspondientes, como son el modelo de implementación el cual engloba el diagrama de despliegue y el diagrama de componentes. Por otra parte, se especificaron los estándares de codificación utilizados durante el desarrollo del sistema propuesto.

CONCLUSIONES

Con el desarrollo del presente trabajo se arriba a las siguientes conclusiones:

- El estudio realizado a cerca de los principales sistemas que a nivel mundial gestionan información proveniente del área de epidemiología evidenció que estos, en su mayoría, son autónomos, heterogéneos y propietarios.
- Se identificaron los principales procesos que se llevan a cabo en el área de epidemiología, los cuales están asociados a la vigilancia de la mortalidad y morbilidad hospitalarias, así como las infecciones intrahospitalarias.
- Se definió como principal funcionalidad del módulo de epidemiología del Sistema de Información Hospitalaria la gestión de las fichas de investigación epidemiológicas.
- Se asimiló la arquitectura definida por el área temática Gestión hospitalaria la cual contribuyó al desarrollo de un sistema robusto y flexible.
- El diseño del sistema propuesto facilitó un mejor entendimiento a los diseñadores de base de datos e implementadores.

De esta forma, se cumplió con los objetivos trazados para la elaboración del trabajo de diploma, se desarrolló un sistema informático, que permite mejorar los procesos de gestión de la información, en el área de epidemiología en las instituciones hospitalarias.

RECOMENDACIONES

Se recomienda al área temática:

- Continuar con el desarrollo de las funcionalidades correspondientes a la gestión de accidentes laborales y las inspecciones sanitarias.
- Implementar el proceso relacionado con la distribución de certificados de defunción y de nacimiento, así como las funcionalidades que gestionan la interconsulta.
- Incluir nuevos formatos de fichas de investigación epidemiológicas, que especifiquen datos relativos a la patología de la misma.

REFERENCIAS BIBLIOGRÁFICAS

1. RADA GABRIEL. *Definiciones: Epidemiología*. 2007
Disponible en: <http://escuela.med.puc.cl/recursos/recepidem/introductorios4.htm>
2. CONCHA, ALBERTO; VILLAVECES, ANDRÉS. *Guías para el diseño, implementación y evaluación de sistemas de vigilancia epidemiológica de lesiones*. 2001.
Disponible en: <http://www.paho.org/Spanish/AD/DPC/NC/guidelines-5-sp.PDF>
3. *Morbilidad*. 25 mayo 2009.
Disponible en: <http://es.mimi.hu/medicina/morbilidad.html>
http://www.cancer.gov/Templates/db_alpha.aspx?Cdrid=44514&lang=spanish
4. INSTITUTO NACIONAL DE CÁNCER. *Diccionario de cáncer*.
Disponible en: http://www.cancer.gov/Templates/db_alpha.aspx?Cdrid=496502&lang=spanish
5. *Nosocomios*. 2009
Disponible en: <http://www.latinsalud.com/articulos/00807.asp?ap=2>
6. U.S. NATIONAL LIBRARY OF MEDICINE. *Enfermedades de declaración obligatoria*. Mayo 2009.
Disponible en: <http://www.nlm.nih.gov/medlineplus/spanish/ency/article/001929.htm>
7. MINISTERIO DE SALUD PERÚ. *Fortalecimiento de las unidades notificantes en el sistema de vigilancia epidemiológica*. Mayo 2009.
Disponible en:
http://www.tacna.minsa.gob.pe/uploads/transferecia/depi/Capacitacion_Vigilancia_Epidemiologica07.pdf
8. *Partners in Impact Forum*. Glion, Switzerland. Marzo 2007.
Disponible en:
http://www.theglobalfund.org/documents/terg/impactforum/PER_01_PresentacionSI-FG_PERU.pdf

9. XUNTA DE GALICIA. *Epidat*. Marzo 2009

Disponible en: <http://www.paho.org/spanish/sha/epidat.htm>

10. CENTERS FOR DISEASE CONTROL AND PREVENTION (CDC). *¿What Is Epi Info™?*. Agosto 2008.

Disponible en: <http://www.cdc.gov/epiinfo/>

11. KURI, PABLO; ET AL. *Sistema Nacional de Vigilancia Epidemiológica*. México 2000.

Disponible en: <http://www.dgepi.salud.gob.mx/infoepi/manuales/SNVE/SINAVE.htm>

12. *Eclipse (Software)*. Mayo 2009.

Disponible en: <http://plataformaclipse.com/>

13. *JBoss Seam Framework*. Febrero 2008

Disponible en: <http://wilmanchamba.wordpress.com/2008/02/20/jboss-seam-framework/>

14. INTERBS SRL. *JBoss*. Mayo 2009.

Disponible en: <http://www.baehost.com.ar/jboss.php>

15. *JBoss tools*. Febrero 2009

Disponible en: http://es.wikipedia.org/wiki/JBoss_Seam

16. JUNTA DE ANDALUCÍA. *RichFaces*. Febrero 2009

Disponible en:

<http://www.juntadeandalucia.es/xwiki/bin/view/MADEJA/RichFaces#caracteristicas%20Seam>

17. INGEMAN, ANDERS. *PostgreSQL*.2009

Disponible en: <http://www.osalt.com/es/postgresql>

18. *PostgreSQL Maestro 8.3*. Marzo 2008.

Disponible en:

http://www.freedownloadcenter.com/es/Programacion/Base_de_Datos_y_Netes/PostgreSQL_Maestro.html .

19. RED HAT MIDDLEWARE. *Relational Persistence for Java and .NET*.2009

Disponible en: <https://hibernate.bluemars.net/>

20. *Visual Paradigm for UML*. 5 de marzo del 2007.

Disponible en:

[http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/)

21. ARNOLD, KEN; GOSLING, JAMES; HOLMES, DAVID. *El lenguaje de Programación Java*. Addison Wesley. 2009.

Disponible en: <http://www.lenguajes-de-programacion.com/programacion-java.shtml>

22. *Metodologías de desarrollo de software*. Diciembre 2006

Disponible en: <http://www.um.es/docencia/barzana/IAGP/lagp2.html#BM1>

23. GOMEZ, JUAN P. *Fundamentos de la metodología RUP*. Septiembre 2007.

Disponible en: <http://www.scribd.com/doc/297224/RUP>

24. *Teoría de Sistemas*. 2005

Disponible en: <http://www.kyberne.com/Documentos%20de%20Partners/13%20-%20TGS%20-%20ModProcesos-BPMN.ppt>.

25. SALINAS, PATRICIO; HISTCHFELD, NANCY. *Tutorial de UML*. Octubre 2006.

Disponible en: <http://www.dcc.uchile.cl/~psalinas/uml/introduccion.html>

26. GARCÍA, KEILA. *Procesos básicos epidemiológicos para un sistema de gestión hospitalario*. 2009.

Disponible en: http://informatica2009.sld.cu/Members/kgarcia_nogueira/procesos-basicos-epidemiologicos-para-un-sistema-de-gestion-hospitalario/at_download/trabajo.

27. UGARTE, JORGE. *BPMN estándar para modelamiento de procesos*.

Disponible en: <http://www.slideshare.net/gugarte/bpmn-estandar-para-modelamiento-de-procesos-presentation>

28. SANCHEZ, EDGAR. *MVC - Modelo Vista Controlador*. Enero 2009

Disponible en: <http://ecuador.latindevelopers.net/blogs/edgarsanchez/archive/2009/01/28/el-patr-243-n-mvc-para-asp-net-ya-est-225-en-release-candidate.aspx>

29. GARCÍA, JOAQUÍN. *Patrones de diseño*. 27 de Mayo de 2005

Disponible en: <http://www.ingenierossoftware.com/analisisydiseno/patrones-diseno.php>

30. SAAVEDRA, JORGE. *PATRONES GRASP (Patrones de Software para la asignación General de Responsabilidad)*. Parte 2. Mayo 2007.

Disponible en: <http://jorgesaavedra.wordpress.com/category/patrones-grasp/>

31. *Clases de diseño*. Mayo 2009

Disponible en: <http://www.taringa.net/posts/info/1492028/Workflow-De-Dise%C3%B1o,-Clases-de-Dise%C3%B1o,-Interfaces,-Diagrama.html>

32. FERNANDEZ, ANA. *Diagramas de Interacción*. Marzo 2001.

Disponible en: <http://tvdi.det.uvigo.es/~avilas/UML/node41.html>

33. GONZÁLEZ, RAÚL. *Modelo de datos*. Mayo 2009

Disponible en: <http://mundogeek.net/archivos/2004/08/26/modelo-de-datos/>

34. FERNANDEZ, ANA. *Diagrama de despliegue*. Marzo 2001.

Disponible en: <http://tvdi.det.uvigo.es/~avilas/UML/node50.html>

35. *Modelo de Implementación: Diagramas de Componentes y Despliegue*. Mayo 2009

Disponible en: <http://www.dsi.uclm.es/asignaturas/42530/pdf/M2tema12.pdf>

36. FERNANDEZ, ANA. *Diagramas UML*. Marzo 2001.

Disponible en: <http://tvdi.det.uvigo.es/~avilas/UML/node22.html>

37. FERNANDEZ, ANA. *Diagramas de componentes*. Marzo 2001.

Disponible en: <http://tvdi.det.uvigo.es/~avilas/UML/node49.html>

BIBLIOGRAFÍA

- ARNOLD, KEN; GOSLING, JAMES; HOLMES, DAVID. *El lenguaje de Programación Java*. Addison Wesley. 2009.
- Ayuda de Rational Suite 2003. Consultada en enero 2009
- CENTERS FOR DISEASE CONTROL AND PREVENTION (CDC). *¿What Is Epi Info™?*. Agosto 2008.
- Clases de diseño*. Mayo 2009
- CONCHA, ALBERTO; VILLAVECES, ANDRÉS. *Guías para el diseño, implementación y evaluación de sistemas de vigilancia epidemiológica de lesiones*. 2001.
- Eclipse (Software)*. Mayo 2009.
- FERNÁNDEZ, ANA. *Diagramas de componentes*. Marzo 2001.
- FERNÁNDEZ, ANA. *Diagrama de despliegue*. Marzo 2001.
- FERNÁNDEZ, ANA. *Diagramas de Interacción*. Marzo 2001.
- FERNÁNDEZ, ANA. *Diagramas UML*. Marzo 2001.
- GARCÍA, JOAQUÍN. *Patrones de diseño*. 27 de Mayo de 2005
- GARCÍA, KEILA. *Procesos básicos epidemiológicos para un sistema de gestión hospitalario*. 2009.
- GÓMEZ, JUAN P. *Fundamentos de la metodología RUP*. Septiembre 2007.
- GONZÁLEZ, RAÚL. *Modelo de datos*. Mayo 2009
- INGEMAN, ANDERS. *PostgreSQL*. 2009
- INSTITUTO NACIONAL DE CÁNCER. *Diccionario de cáncer*
- INTERBS SRL. *JBoss*. Mayo 2009.
- JBoss Seam Framework*. Febrero 2008
- JBoss tools*. Febrero 2009
- JUNTA DE ANDALUCÍA. *RichFaces*. Febrero 2009
- KURI, PABLO; ET AL. *Sistema Nacional de Vigilancia Epidemiológica*. México 2000.
- LARMAN, CRAIG. *Uml y patrones*. Capítulo 18. Prentice Hall Iberoamericana. 1999
- Metodologías de desarrollo de software*. Diciembre 2006
- MINISTERIO DE SALUD PERÚ. *Fortalecimiento de las unidades notificantes en el sistema de vigilancia epidemiológica*. Mayo 2009.
- Modelo de Implementación: Diagramas de Componentes y Despliegue*. Mayo 2009
- Molpeceres, A. *Procesos de desarrollo: RUP, XP Y FDD*. 2002
- Morbilidad*. 25 mayo 2009.
- Nosocomios*. 2009
- Partners in Impact Forum*. Glion, Switzerland. Marzo 2007.

PostgreSQL Maestro 8.3. Marzo 2008.

PRESSMAN, ROGER. *“Ingeniería del Software. Un enfoque práctico”*. McGraw-Hill/Interamericana de España. 2002.

RADA GABRIEL. *Definiciones: Epidemiología*. 2007.

RED HAT MIDDLEWARE. *Relational Persistence for Java and .NET*. 2009

RUMBAUGH, JAMES, JACOBSON, IVAR; BOOCH, GRADY, *El lenguaje unificado de modelado. Manual de referencia*. Addison Wesley. 2000.

RUMBAUGH, JAMES, JACOBSON, IVAR; BOOCH, GRADY, *“El proceso unificado de desarrollo”*. Addison Wesley. 2000.

SAAVEDRA, JORGE. *PATRONES GRASP (Patrones de Software para la asignación General de Responsabilidad)*. Parte 2. Mayo 2007.

SALINAS, PATRICIO; HISTCHFELD, NANCY. *Tutorial de UML*. Octubre 2006.

SANCHEZ, EDGAR. *MVC - Modelo Vista Controlador*. Enero 2009

SCHMULLER, JOSEPH. *Aprendiendo UML en 24 horas*. 1999.

Teoría de Sistemas. 2005

UGARTE, JORGE. *BPMN estándar para modelamiento de procesos*.

UML y la Modelación de datos.pdf. Whitepaper de Rational Rose.

U.S. NATIONAL LIBRARY OF MEDICINE. *Enfermedades de declaración obligatoria*. Mayo 2009.

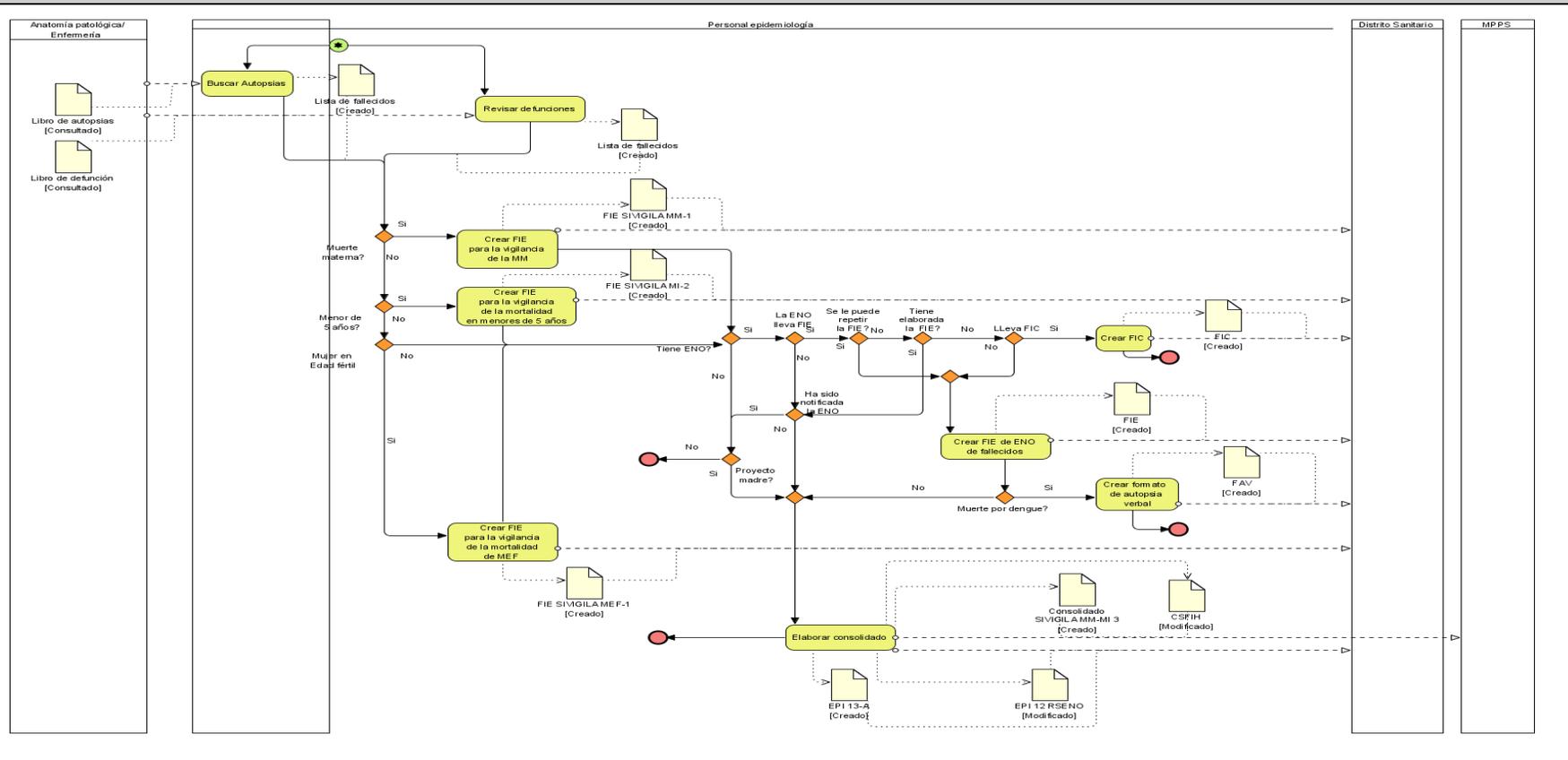
Visual Paradigm for UML..5 de marzo del 2007.

XUNTA DE GALICIA. *Epidat*. Marzo 2009

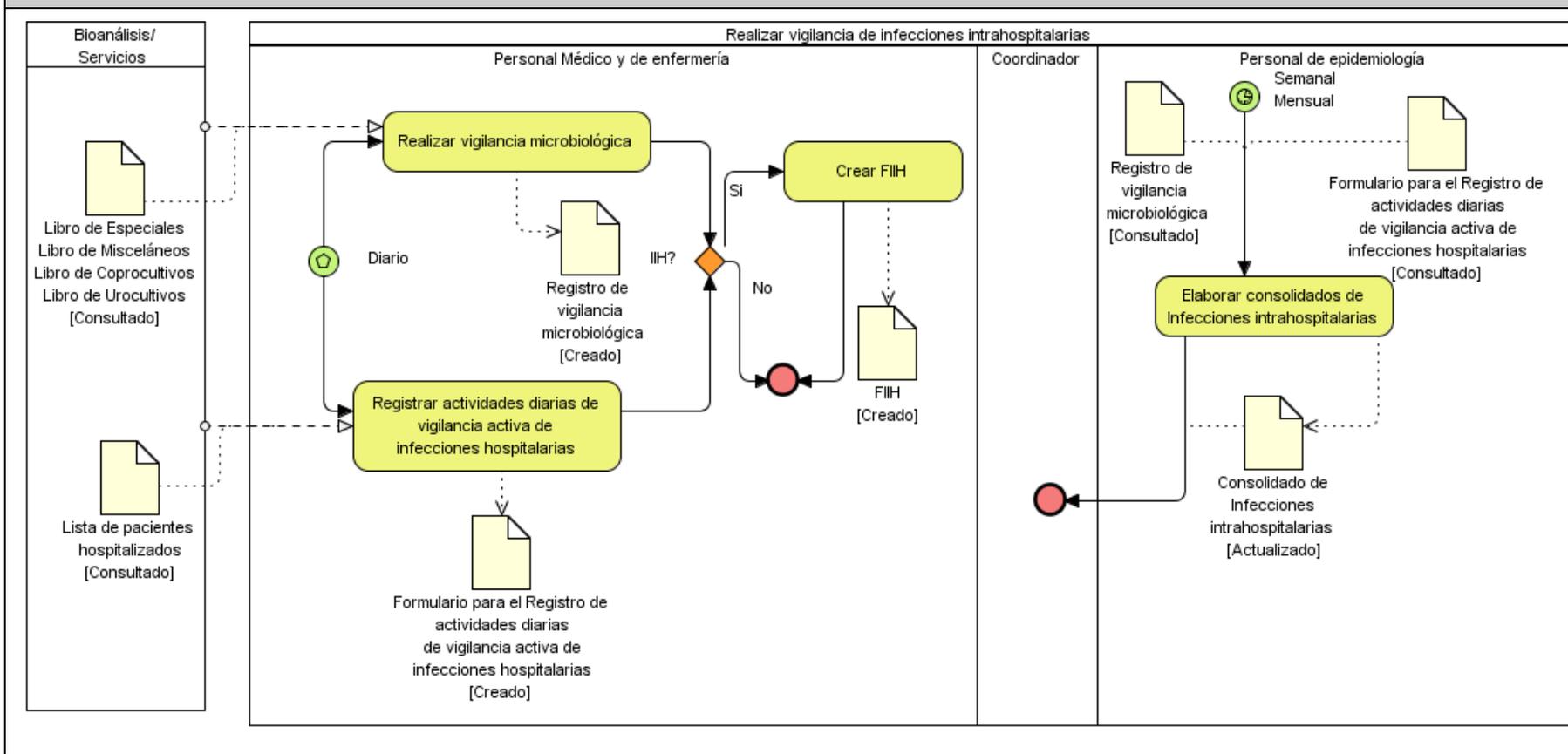
ANEXOS

Anexo 1: Diagramas de procesos de negocio

Realizar vigilancia epidemiológica de la mortalidad hospitalaria

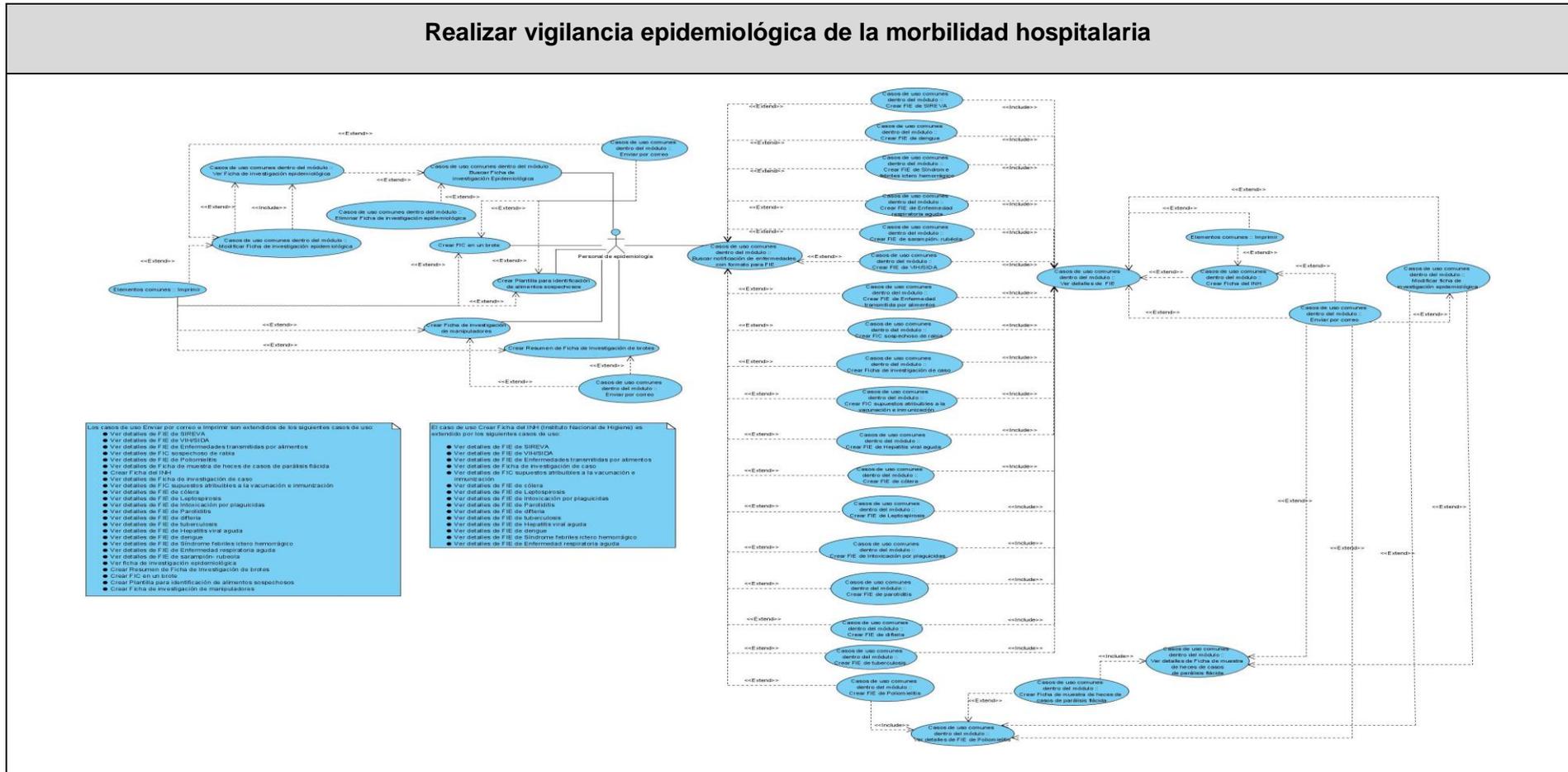


Realizar vigilancia de infecciones intrahospitalarias

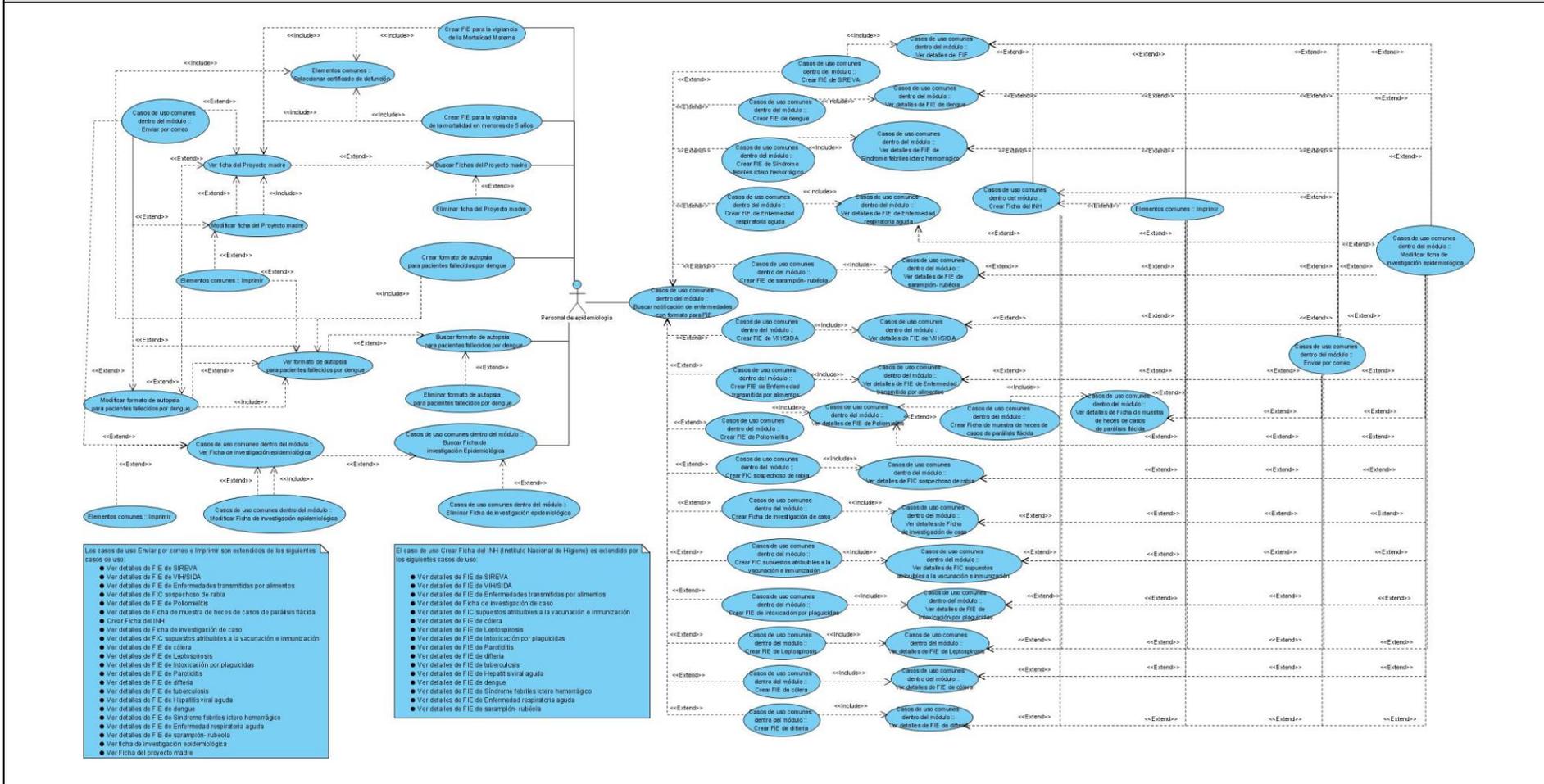


Anexo 2: Diagramas de Casos de Usos por procesos.

Realizar vigilancia epidemiológica de la morbilidad hospitalaria



Realizar vigilancia epidemiológica de la mortalidad hospitalaria



GLOSARIO

API: Una **interfaz de programación de aplicaciones** o **API** (del inglés **Application Programming Interface**) es el conjunto de funciones y procedimientos (o métodos, si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Base de datos: serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular.

CASE: es una sigla, que corresponde a las iniciales de: Computer Aided Software Engineering; y en su traducción al Español significa Ingeniería de Software Asistida por Computación. Las herramientas CASE son un conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de sistemas de información. Estas abarcan todos los pasos del proceso de software, y también aquellas actividades generales que se aplican a lo largo de todo el proceso. Son un complemento de la caja de herramientas del ingeniero del software, además ayudan a asegurar que la calidad sea algo diseñado antes de llegar a construir el producto.

CVS (Concurrent Versions System): es una aplicación informática que implementa un sistema de control de versiones: mantiene el registro de todo el trabajo y los cambios en los ficheros (código fuente principalmente) que forman un proyecto (de programa) y permite que distintos desarrolladores (potencialmente situados a gran distancia) colaboren.

Complemento (o **plug-in** en inglés): es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API.

Consolidado: Resumen que contiene los principales parámetros medidos para un programa en específico. Muestra resultados numéricos o estadísticos en dependencia de la periodicidad con que se elabore.

Encriptación: es el proceso para volver ilegible información considera importante.

Enfermedad de notificación obligatoria: Enfermedad infectocontagiosa, que en algunos casos puede ser crónica. Algunas enfermedades infectocontagiosas son de denuncia o notificación obligatoria, internacional o nacional. La notificación debe ser efectuada por médicos o autoridades de instituciones, a la autoridad sanitaria más próxima.

Entorno de desarrollo integrado (en inglés Integrated Development Environment ('IDE')): es un programa compuesto por un conjunto de herramientas para un programador.

Epidemiología: Ciencia que estudia la forma en que se presentan, se transmiten y se previenen las enfermedades.

Ficha de investigación epidemiológica (FIE): La Ficha de investigación epidemiológica se crea a los pacientes que se les detecta una Enfermedad de Notificación Obligatoria (ENO). No hay modelos de fichas definidos para todas las ENO, debido a que estos son creados en dependencia de la necesidad epidemiológica en un área determinada. Guarda datos personales del paciente, así como la condición actual del mismo. En esta ficha se plasman además, los resultados de los exámenes que se le practican a los pacientes. Algunas de estas son únicas, pero existen otras fichas que en dependencia de la patología permiten ser elaboradas en más de una ocasión.

Oficio: Carta o documento que se elabora con un objetivo definido, ya sea, solicitando, aprobando, recibiendo alguna información o recurso y que se utiliza como constancia de un hecho determinado.

Framework: En el desarrollo de software, un *framework* es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un *framework* puede incluir soporte de programas, bibliotecas y un lenguaje de scripting entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Garbage collector (en español, recolector de basura): es un mecanismo implícito de gestión de memoria implementado en algunos lenguajes de programación de tipo interpretado o semi-interpretado.

Gestor de Base de Datos: Es un conjunto de programas que permiten crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad.

Hardware: Conjunto de elementos materiales que conforman una computadora, sin embargo, es usual que sea utilizado en una forma más amplia, generalmente para describir componentes físicos de una tecnología, así el hardware puede ser de un equipo militar importante, un equipo electrónico, un equipo informático o un robot. En informática también se aplica a los periféricos de una computadora tales como el disco duro, CD-ROM, disquetera (floppy), etc. En dicho conjunto se incluyen los dispositivos electrónicos y electromecánicos, circuitos, cables, armarios o cajas, periféricos de todo tipo y cualquier otro elemento físico involucrado.

HL7 (Health Level Seven): es un conjunto de estándares para el intercambio electrónico de información médica.

Librerías: una librería o biblioteca es un conjunto de procedimientos y *funciones* (subprogramas) agrupadas en un archivo con el fin de ser aprovechadas por otros programas.

Log: es un registro oficial de eventos durante un periodo de tiempo en particular. Para los profesionales en seguridad informática es usado para registrar datos o información sobre quién, qué, cuándo, dónde y por un evento ocurre para un dispositivo en particular o aplicación.

Morbilidad: Morbilidad es el estudio de los efectos de una enfermedad en una población en el sentido de la proporción de personas que enferman en un sitio y tiempo determinado.

Mortalidad: La mortalidad es un término demográfico que designa un número proporcional de muertes en una población y tiempo determinado.

Navegador web (en inglés, browser): es un programa o aplicación informática que nos va a permitir movernos por internet y acceder al contenido de las webs, blogs, foros, galerías fotográficas, etc., de Internet.

Orientado a Objeto: Significa que el software se organiza como una colección de objetos discretos que contiene tanto estructura de datos como también un comportamiento.

Programa: Orientado a patologías, áreas, entre otros aspectos, que contiene un grupo de directrices que describen los procedimientos y normas establecidas para cada uno de estos. Brinda además un grupo de instrumentos para la recolección de la información.

Servidor: es una computadora central, de gran capacidad, compartida por las otras computadoras de la red, llamadas Clientes o estaciones de trabajo (workstations), ya que reciben el servicio de almacenar, controlar y compartir la información contenida en el servidor.

Sistemas operativos: Conjunto de programas que se integran con el hardware para facilitar al usuario, el aprovechamiento de los recursos disponibles.

Software: Al conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema (hardware). Esto incluye aplicaciones informáticas tales como un procesador de textos, que permite al usuario realizar una tarea, y software de sistema como un sistema operativo, que permite al resto de programas funcionar adecuadamente, facilitando la interacción con los componentes físicos y el resto de aplicaciones.

WSDL: son las siglas en inglés de "Lenguaje de Descripción de Servicios Web" (o "Web Services Description Language"). Es un lenguaje que está basado en XML y que permite la descripción de los servicios web desplegados. WSDL se utiliza también para la localización y ubicación de estos servicios en Internet. Un documento WSDL no es más que un documento XML que describe ciertas

características propias de un servicio web, así como su localización y aquellos parámetros y métodos que soporta.

XML: Sigla en inglés de eXtensible Markup Language («lenguaje de marcas extensible»), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML). Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Algunos de estos lenguajes que usan XML para su definición son XHTML, SVG, MathML.