

Universidad de las Ciencias Informáticas

Facultad 7



Tema: *Control Sanitario Internacional*

Título: *Sistema Integral de Vigilancia Epidemiológica*

**Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas**

Autor: *Yarel Leyva Leyva*

Tutores: *Ing. Yunaysy Ortiz Batista*

Ing. Ricardo Collada de la Rosa

“Año del 50 aniversario del triunfo de la Revolución”

Ciudad de la Habana, Junio 2009

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y le reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo lo presentado a los 10 días del mes de Junio del año 2009.

Yarel Leyva Leyva

Firma del Autor

Yunaysy Ortiz Batista

Firma de la tutor

Ricardo Collada de la Rosa

Firma del tutor

DATOS DEL CONTACTO

Yunaysy Ortiz Batista: Ing. en Informática, graduada en la Universidad de Holguín en 2006. Profesora de PP5 en la Facultad 7.

Ricardo Collada de la Rosa: Profesor recién graduado de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI). Curso 2007-2008. Profesor de NADSS en la Facultad 7.

AGRADECIMIENTOS

A la revolución y Fidel, por haberme dado la oportunidad de vivir cinco años en una universidad como esta y darme la posibilidad de preparar un trabajo como este.

A mi mamá: “Viejuca eres la mejor... Mi guía, mi fuerza para seguir adelante ante los caminos oscuros y no detenerme ante nada”.

A mi papá que no dejó de llamarme nunca para saber como iba todo, por eso hoy le digo: “Viejo, ya somos ingenieros... Después te diré si sacamos *Sobresaliente*. Un beso grande”.

A mi padrasto que es mi otro padre. Eres lo mejor y lo sabes.

A mi hermanote Maikel: “Pipo, ya pronto podremos tener nuestra compañía MaikoSoft. Sigue adelante, que juntos tenemos que echar para adelante a los viejucos.”

A mis tías Moraima, Magalis, Mirna, que siempre me apoyaron desde casa.

A toda mi familia por estar siempre que los necesité.

A Velázquez, que más que un amigo, es el padre que tuve lejos.

A la gente que hizo de estos cinco años, los mejores de mis veintitrés: mis compañeros desde primer año, los que vienen conmigo desde el pre y han pasado por las mismas cosas que yo. A mi novia Solainy, porque sin ella y su matraquilla, jamás hubiera terminado mi tesis. Un beso para ti mi brujita linda. A todos Uds.: “Muchas Gracias”.

A mis tutores por su apoyo todo el tiempo. Gracias por ayudarme cuando los necesité.

DEDICATORIA

A mi abuelita querida. Todo esto es para ti y por ti. Lamento no tenerte aquí conmigo para decirte que te quiero y darte las gracias por haber hecho de mí, lo que soy hoy. Un hombre de grandes esperanzas.

Cuídate allá, donde quiera que estés.

A mi madre, esa mujer que ha hecho hasta lo imposible para que esté hoy aquí y haya podido terminar todo este trabajo.

RESUMEN

El Control Sanitario Internacional en Cuba es una estrategia trazada por el Estado y el Ministerio de Salud Pública (MINSAP) para detectar la introducción de enfermedades transmisibles. Con el objetivo de evitar la propagación de estas e impedir afectaciones a la población en caso de existir alguna situación de riesgo epidemiológico. Este peligro potencial se ha incrementado en los últimos años por la ampliación de las relaciones internacionales con países donde estas enfermedades son potencialmente significativas.

El presente trabajo tiene como objetivo realizar el diseño de una nueva versión del sistema Higiene y Epidemiología que cumpla con todas las funcionalidades identificadas por el cliente y esté en correspondencia con la arquitectura definida por la Facultad 7 para el Área Temática Sistemas Especializados.

Para desarrollar el sistema se ha seleccionado como metodología el Proceso Unificado de Desarrollo (RUP), que hace uso del Lenguaje Unificado de Modelado (UML). Se obtuvieron artefactos de los flujos de trabajo: Negocio, Requerimientos y Diseño. También se utilizó la herramienta Visual Paradigm.

Como resultado de este trabajo de diploma se obtuvo el prototipo no funcional del Sistema Integral de Higiene y Epidemiología que posibilitará su posterior desarrollo, debido a que corrige los errores de la versión anterior, 1.0.

TABLA DE CONTENIDOS

RESUMEN..... 6

INTRODUCCIÓN..... 9

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA..... 12

1.1. Sistemas Informáticos Desarrollados..... 12

1.1.1. Informatización del Control Sanitario Internacional en la Escuela Latinoamericana de Medicina en Cuba (ELAM)..... 12

1.1.2. Software para el Sistema Integrado de Vigilancia de Dengue..... 13

1.1.3. Control Sanitario Internacional de Higiene y Epidemiología [Versión 1.0] 13

1.2. Justificación del Sistema Propuesto..... 14

1.3. Lenguaje de Modelado a Utilizar 15

1.3.1. Lenguaje Unificado de Modelado (UML)..... 15

1.4. Metodologías de Desarrollo..... 16

1.4.1. Proceso Unificado de Desarrollo (RUP) 16

1.4.2. Librería y Framework. 17

1.5. Herramientas CASE. 18

1.5.1. Visual Paradigm. 19

1.6. Herramientas de Desarrollo..... 19

1.6.1. Servidor Web Apache. 20

1.6.2. Navegadores Web..... 21

1.6.3. Adobe Dreamweaver..... 21

1.7. Patrones a utilizar 22

1.7.1. Patrones de Arquitectura..... 22

1.7.2. Patrones de Diseño..... 25

CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA..... 28

2.1. Objeto de estudio..... 28

2.1.1.	Flujo actual de los procesos de negocio.....	28
2.1.2.	Análisis crítico de los procesos.....	29
2.1.3.	Objeto de Automatización.....	30
2.2.	Modelo de Negocio.....	30
2.2.1.	Descripción de los actores y trabajadores del negocio.....	30
2.2.2.	Reglas del Negocio.....	32
2.2.3.	Diagrama de Casos de Uso del Negocio.....	33
2.2.4.	Descripción de los Casos de Uso del Negocio.....	33
<i>CAPÍTULO 3 DISEÑO DEL SISTEMA</i>		78
3.1	Modelo de Diseño.....	78
3.1.1	Diagrama de Clases del Diseño.....	81
3.1.2	Descripción de Clases del Diseño.....	82
3.1.3	Diagramas de Interacción.....	88
3.1.4	Diagrama de despliegue.....	90
<i>CONCLUSIONES</i>		92
<i>RECOMENDACIONES</i>		93
<i>REFERENCIAS BIBLIOGRÁFICAS</i>		94
<i>BIBLIOGRAFÍA</i>		96
<i>ANEXO 1</i>		102
<i>ANEXO 2</i>		108

INTRODUCCIÓN

En Cuba, desde los inicios de la Revolución, se han establecido y fortalecido las relaciones diplomáticas, culturales y comerciales con otros países. Lo que ha traído como consecuencia, el aumento considerable del tráfico internacional de personal cubano y extranjero; hacia y desde regiones endémicas de enfermedades transmisibles; por ello se incrementa el riesgo de propagación de las mismas. Sobre todo, si se tiene en cuenta que en la isla existen vectores capaces de propagar estas enfermedades de forma vertiginosa pues el ecosistema brinda condiciones óptimas para su desarrollo y reproducción.

Por las razones anteriores, se considera que el Control Sanitario Internacional tiene gran importancia para la salud pública cubana, que tiene creados y estructurados los sistemas de vigilancia epidemiológica a viajeros, enfermedades transmisibles y a la comunidad en general. Esto ha permitido que durante los años de Revolución, se hayan logrado avances solo comparables con países desarrollados; donde las enfermedades crónicas tienen un papel relevante y las transmisibles son muy escasas.

El programa nacional de Control Sanitario Internacional (CSI), incluye los programas de Salud Ambiental, Control de Vectores, así como la Vigilancia Epidemiológica. Los que han sido y son aún una prioridad fundamental del estado cubano y el MINSAP en especial. Este trabajo forma parte de los sistemas desarrollados para fortalecer la Vigilancia Epidemiológica en el país.

En la actualidad, los procesos de control y vigilancia epidemiológica se realizan a través de redes de información establecidas por el MINSAP que van desde el Área de Salud hasta el propio Ministerio. Estas redes permiten tener conocimiento de datos significativos concernientes al estado epidemiológico del país a todos los niveles. Sin embargo, la información se mueve a través de conversaciones telefónicas o correspondencia por correo convencional.

A pesar de estar muy bien organizado, el sistema de vigilancia epidemiológica puede ser susceptible a errores humanos, que pueden traer como consecuencia que no siempre se manejan datos totalmente actualizados. Además de incoherencia en la información que se recibe de los distintos niveles y en cierta medida, la pérdida de datos valiosos. También hay que destacar que para almacenar la información de los datos históricos de pacientes de enfermedades tropicales y de mayor significación se utilizan libros de Microsoft Office Excel, en los cuales es muy complejo realizar un análisis estadístico o cuantitativo a posteriori, pues no todas las fuentes de información envían los datos en el mismo formato. (1)

Durante el curso 2007-2008 los esfuerzos productivos de la UCI y la Facultad 7 se concentraron en desarrollar la primera versión de un sistema que permitiera viabilizar la gestión de la información relacionada con la vigilancia epidemiológica al viajero y a las enfermedades tropicales en Cuba (Sistema Higiene y Epidemiología). Este sistema aunque responde a un conjunto de necesidades, no cumple totalmente con los requerimientos identificados por los clientes. Entre sus principales limitaciones, se destacan la falta de conexión con el sistema de Vectores para obtener la información referente a la situación entomológica de las manzanas donde reside un determinado paciente, y de la misma forma, valorar el riesgo epidemiológico de la región.

Esta versión, tampoco contempló la gestión de la información relacionada con las Inspecciones Sanitarias Estatales que se le deben realizar a las aeronaves al arribar al país. No permitió una serie de salidas (reportes) que posibilitaran un análisis estadístico de la información por parte de los usuarios finales. Además, el sistema se desarrolló sobre el framework CodeIgniter, que a pesar de ser un marco de trabajo sencillo de utilizar, carece de un conjunto de funcionalidades que pudieran limitar el desarrollo óptimo de nuevas versiones del sistema actual.

Teniendo en cuenta la situación existente, se identificó como **problema a resolver** que la versión actual del sistema Higiene y Epidemiología no cumple con todas las funcionalidades identificadas por el cliente y no está en correspondencia con la arquitectura definida por la Facultad 7 para el Área Temática Sistemas Especializados.

Por tanto el **objeto de estudio** se enmarcó en el proceso de gestión de la información referente al Control Sanitario Internacional, delimitando como **campo de acción** el proceso de gestión de la información referente a la vigilancia epidemiológica al viajero y al control de las enfermedades transmisibles en Cuba. Para dar solución al problema planteado, se determinó como **objetivo general**, realizar el diseño de una nueva versión del sistema Higiene y Epidemiología que cumpla con todas las funcionalidades identificadas por el cliente y que esté en correspondencia con la arquitectura definida por la Facultad 7 para el Área Temática Sistemas Especializados.

Para poder llevar a la práctica estos posibles resultados y dar cumplimiento a los objetivos trazados, se desarrollarán las siguientes **tareas de la investigación**:

1. Valorar las técnicas de programación, lenguaje y framework propuestos por la Facultad para el Área Temática Sistemas Especializados.

2. Analizar el actual sistema de Higiene y Epidemiología, centrándose en la gestión de la información de dicha versión.
3. Refinar los Modelos de Negocio y Sistema.
4. Analizar la integración con otros componentes ya existentes en el Sistema de Información para la Salud (SISalud) y con el sistema de Detección y Control de Vectores.
5. Obtener el prototipo no funcional de la nueva versión y el diseño de todas las funcionalidades de la versión anterior, usando el framework Symfony.
6. Obtener todos los artefactos como resultado del Diseño de la nueva versión.

Para dar cumplimiento a estas tareas de manera satisfactoria el presente trabajo de diploma está estructurado en **3** capítulos:

Capítulo 1: Fundamentación Teórica: Se hace un análisis de los sistemas existentes para la gestión de la vigilancia epidemiológica que se han desarrollado en Cuba. Se realiza una valoración sobre las tecnologías y metodologías actuales, incluyendo los patrones que se utilizan para el diseño de la aplicación.

Capítulo 2: Características del Sistema: Describe el flujo actual de los procesos de negocio y se realiza un análisis crítico de la ejecución de estos. Se plantea cuál es el objeto de automatización, dando paso al modelado del negocio donde se definen los actores y trabajadores del negocio con una descripción de los mismos y las reglas a tener en cuenta durante todo el proceso. Posteriormente, se hace alusión a los requisitos funcionales y no funcionales con que cuenta el sistema. Finalmente se muestran cuáles son los Casos de Usos del mismo, con una descripción detallada de ellos.

Capítulo 3: Diseño del Sistema: Describe algunas características del framework a utilizar; Symfony; así como el modelo de diseño y muestra los diagramas de diseño correspondientes a cada uno de los Casos de Uso del sistema propuesto, así como sus respectivos diagramas de secuencia.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

En este capítulo se hace un análisis de los sistemas existentes para la gestión de la vigilancia epidemiológica que se han desarrollado en Cuba. Se realiza una valoración sobre las tecnologías y metodologías actuales, incluyendo los patrones que se utilizan para el diseño de la aplicación.

1.1. Sistemas Informáticos Desarrollados.

1.1.1. Informatización del Control Sanitario Internacional en la Escuela Latinoamericana de Medicina en Cuba (ELAM).

La Escuela Latinoamericana de Medicina en Cuba (ELAM) concibió un software que se programó en lenguaje Delphi. Fue utilizado con el objetivo de agilizar el arribo de estudiantes extranjeros en los aeropuertos, provenientes de regiones con una alta prevalencia de enfermedades transmisibles para cursar estudios en sus instituciones (2).

Este software permite la manipulación de diferentes tablas de datos en Excel donde se registran los datos generales de los estudiantes: por año de estudio, sexo, país, grupo docente, facultad a la cual pertenecen, fecha de arribo al país, sintomatología o los signos relevantes encontrados al examen médico realizado, tratamientos antipalúdicos aplicados y un registro de exámenes realizados con sus resultados por año (3).

Permite al usuario moverse sobre toda la base de datos a través de un buscador, dándole varias opciones de búsqueda, las cuales pueden ser utilizadas en diferentes procedimientos. Realizar un reporte rápido de totales de arribos por año, por países, por facultades y provincias, inexistentes al Control Sanitario Internacional y al tratamiento radical antipalúdico. Ofrece una seguridad para el acceso de personal no autorizado a los datos y que no sea susceptible de ser modificada fácilmente (4).

Este software fue hecho para una institución con características específicas teniendo en cuenta determinados datos y estructura; por lo que no posibilita su expansión a otras instituciones u organizaciones; no es una solución inmediata para el Control Sanitario Internacional en temas como la vigilancia epidemiológica.

Capítulo 1. Fundamentación Teórica

1.1.2. Software para el Sistema Integrado de Vigilancia de Dengue.

El Sistema Integrado de Vigilancia de Dengue (SIVD) es una investigación que requería de la recogida de información clínico-epidemiológica de los pacientes que eran diagnosticados como sospechosos de dengue, además de información ambiental recogida por parte de los compañeros de la Campaña de Lucha Anti vectorial y de datos entomológicos provenientes del laboratorio a tal efecto.

El SIVD requería un análisis integrado de toda la información con vistas a lograr un abordaje integral a la vigilancia del Dengue, además de salidas para un sistema de información geográfico, que por un procesamiento y análisis manual sería muy engorroso por la periodicidad de recogida de la información (ciclos de 11 días) y el volumen de la misma, esto justificó la concepción de este software, el cual se introduciría en un inicio en el Municipio Cotorro y posteriormente en el municipio Centro Habana, ambos en Ciudad de La Habana (5). Se diseñó una base de datos, teniendo en cuenta los modelos que se aplican en el campo: Informe de cierre de ciclo de la Campaña de Lucha Antivectorial (hoja de vaciamiento); Control de focos positivos de *Aedes Aegypti*; y Registro epidemiológico; datos que provienen de los locales inspeccionados, de las muestras recogidas por los operarios y de los pacientes con síntoma febril que entran al sistema.

La base de datos se diseñó en Access 2000 con protección por clave y que incluía la estructura político-administrativa del país pues se realizó todo el diseño y programación de la aplicación con vistas a que la misma pudiera ser explotada en cualquier parte del país; se incluye la información relativa a manzanas pues es la unidad de análisis del sistema; además de tablas particulares para información de ambiente, entomología y clínico-epidemiológica (6).

La aplicación se programó en Borland Delphi 7 Professional sobre sistema operativo Windows 98SE, con tecnología DAO(**D**ocument **A**cces **O**bject) para el acceso a datos, tipo de aplicación MDI (“Multiple Document Interface”) y uso de bibliotecas de componentes libres. Se conciben varios módulos que permiten dar salida a los objetivos para los cuales se creó.

1.1.3. Control Sanitario Internacional de Higiene y Epidemiología [Versión 1.0]

El grupo de Sistemas Especializados en salud de la Universidad de las Ciencias Informáticas, durante el 2008, implementó el sistema Control Sanitario Internacional (CSI). Entre sus módulos contiene el subsistema de Higiene y Epidemiología, que tiene dos partes fundamentales: el control de enfermedades

Capítulo 1. Fundamentación Teórica

tropicales -que esta primera versión solo se limitó al control de Dengue- mediante la vigilancia a pacientes con síntomas febriles inespecíficos, y el seguimiento a viajeros con el objetivo de evitar la introducción y propagación de estas enfermedades en la región. Además, recoge una serie de datos necesarios y permite al usuario estar actualizado en cada momento de la situación epidemiológica en el país.

El software gestiona la información en la base: aeropuertos, hospitales y unidades de salud, los que muestran la información más actualizada a las autoridades sanitarias en materia de control epidemiológico en los niveles municipal, provincial y nacional, según los permisos dentro del sistema. Al estar centralizado evita la duplicidad y pérdida total o parcial de información sensible sobre el estado epidemiológico del país y permite a los directivos de la especialidad, tomar decisiones precisas y asignar los recursos necesarios para dar respuesta a una situación determinada, así como para prevenirla. Este módulo del sistema CSI, permite también la gestión de pruebas de laboratorio y seguimiento a pacientes con dengue o sospechosos a padecer esta enfermedad.

El sistema CSI está implementado en el lenguaje de programación PHP, con su base de datos en MySQL. La combinación de ambos se ha popularizado en la actualidad dado el buen rendimiento de las aplicaciones web desarrolladas sobre ellas, así como la versatilidad para poder hacer diferentes tipos de software, el dinamismo y la velocidad de respuesta. Hay que tener en cuenta además que es software libre y totalmente gratuito, lo cual supone un ahorro por concepto de compra de licencias, soporte y mantenimiento (7).

1.2. Justificación del Sistema Propuesto.

A pesar de que existen sistemas informáticos para dar solución al control de las enfermedades transmisibles, solo se ha implementado un software en la Facultad 7 de la UCI, que resuelve parcialmente el manejo de la información, pero no está en completa correspondencia con la arquitectura definida para el Área Temática Sistemas Especializados al estar implementado sobre el Framework CodeIgniter y presenta una interfaz gráfica poco amigable y segura, tampoco responde a todos los requisitos expuestos por el cliente, faltándole funcionalidades necesarias que integren varios módulos que conforman al Control Sanitario Internacional, para mantener una vigilancia epidemiológica del país a gran escala.

CodeIgniter es un framework orientado a la programación orientada a objetos pero con faltas de organización de código. Cuando es un proceso de software que requiere nuevas funcionalidades y esto trae consigo el incremento del mismo o migración de base datos, así como creación de nuevas versiones

Capítulo 1. Fundamentación Teórica

o algunos parches para brindar soporte, la aplicación tiende a crecer en gran medida lo que hace más complejo el manejo u organización de sus archivos.

Mientras que Symfony se organiza más en este sentido, brindando una buena programación orientada a objetos, de fácil continuidad o migración, capaz de generar códigos mediante pequeños comandos a través de una consola, que ahorra tiempo y espacio del desarrollador. Además, está en constante desarrollo y cada vez resulta más eficiente en la producción de grandes aplicaciones Web. A gran diferencia de CodeIgniter, Symfony está pensado para llevar a cabo un proceso de desarrollo de software mucho más seguro, grande y robusto.

Es por ello que se propuso el diseño de una aplicación Web, fundamentada en el uso del Framework Symfony, con una arquitectura más amigable y vistosa al usuario, con la ayuda de **YUI**. Que refine los problemas existentes en el Negocio y Sistema respectivamente, para así obtener un prototipo no funcional de esta segunda versión del módulo Higiene y Epidemiología correspondiente al Control Sanitario Internacional.

Para diseñar el sistema se utilizó el Lenguaje Unificado de Modelado (**UML**) debido a que es el más conocido y utilizado en la actualidad. Como metodología de desarrollo de software, **RUP** debido a que proporciona un acercamiento disciplinado a la asignación de tareas y responsabilidades en una organización de desarrollo, quedando bien documentado el software de forma perdurable y extensible. La herramienta **CASE** empleada fue el Visual Paradigm, porque soporta aplicaciones Web y exporta código de alrededor de 10 lenguajes de programación que incluyen **PHP**, que es el lenguaje que debe utilizarse en la implementación del sistema.

Es válido señalar que tanto las herramientas como metodología y lenguaje fueron analizadas por el grupo de arquitectura de la Facultad 7 para el Área Temática Sistemas Especializados y han sido plasmadas en los lineamientos del documento de arquitectura. A continuación son mostradas algunas de sus características.

1.3. Lenguaje de Modelado a Utilizar.

1.3.1. Lenguaje Unificado de Modelado (UML).

Lenguaje Unificado de Modelado (UML por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje

Capítulo 1. Fundamentación Teórica

que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables (8).

Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado de Desarrollo) pero no especifica en sí mismo qué metodología o proceso usar (9).

1.4. Metodologías de Desarrollo.

En un proyecto de desarrollo de software la metodología define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo. Las Metodologías son un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software. A continuación se muestra la metodología que se utilizará para el desarrollo del sistema.

1.4.1. Proceso Unificado de Desarrollo (RUP).

El Proceso Unificado de Desarrollo es un proceso de ingeniería del software. Proporciona un acercamiento disciplinado a la asignación de tareas y responsabilidades en una organización de desarrollo. Su propósito es asegurar la producción de software de alta calidad que se ajuste a las necesidades de sus usuarios finales con unos costos y calendario predecibles (10). Es una metodología basada en un pequeño grupo de principios claves: el equipo de un proyecto de software debe planificar el desarrollo; debe conocer hacia donde se dirige; debe documentar el proyecto de una manera perdurable y extensible.

Además incorpora el concepto de "mejores prácticas" para la ingeniería de software, definido por tres características fundamentales (11):

- 1. Dirigido por Casos de Uso:** el desarrollo está dirigido a satisfacer las necesidades de los usuarios del sistema expresadas en Casos de Uso.

Capítulo 1. Fundamentación Teórica

- 2. Centrado en la arquitectura:** el desarrollo se centra en una arquitectura bien definida, con relaciones claras entre sus distintos componentes.
- 3. Iterativo e Incremental:** el problema y la solución se organizan en pequeñas piezas, de manera que cada iteración se dirige específicamente al desarrollo de un conjunto de ellas. Cada iteración se construye sobre la base creada por las iteraciones anteriores, agregándole capacidades al sistema.

La metodología RUP divide en 4 fases el desarrollo del software:

- **Inicio:** el objetivo en esta etapa es determinar la visión del proyecto.
- **Elaboración:** en esta etapa el objetivo es determinar la arquitectura óptima.
- **Construcción:** en esta etapa el objetivo es llegar a obtener la capacidad operacional inicial.
- **Transición:** el objetivo es llegar a obtener una versión del proyecto lista para instalarse.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

1.4.2. Librería y Framework.

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Separa la lógica de negocio, de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.)(12) como en plataformas Windows.

Para conocer un poco más respecto a este framework es necesario saber cuales son sus características principales.

Capítulo 1. Fundamentación Teórica

Características generales de Symfony:

- ✓ Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares)
- ✓ Independiente del sistema gestor de bases de datos.
- ✓ Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- ✓ Basado en la premisa de "*convenir en vez de configurar*", en la que el desarrollador solo debe configurar aquello que no es convencional.
- ✓ Sigue la mayoría de *mejores prácticas* y patrones de diseño para la web
- ✓ Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- ✓ Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- ✓ Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

Yahoo User Interface (YUI) son librerías JavaScript y CCS de código abierto liberadas bajo licencia BSD. Provee una gran cantidad de utilidades en JavaScript y controles de interfaz de usuario lo que permite la homogenización en las interfaces visuales de los productos de software. Es un extenso catálogo con cerca de 300 códigos JavaScript y CSS de uso libremente para proyectos web, ya sean personales o comerciales. Tiene ejemplos, código fuente y documentación relacionada para poder utilizar y personalizar todos los *scripts*. Los *scripts* están divididos en categorías según su utilidad: animación, autocompletar, cuentagotas, treeview, DOM, CSS, entre otros (13). En total, se encuentran hasta 40 categorías distintas con varios *scripts*, cada una permite utilizarlas de una forma básica o más avanzada.

1.5. Herramientas CASE.

Las herramientas **CASE** (**C**omputer **A**ided **S**oftware **E**ngineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas

Capítulo 1. Fundamentación Teórica

pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores, entre otras (14).

1.5.1. Visual Paradigm.

El Visual Paradigm es una herramienta **CASE** que utiliza “UML” como lenguaje de modelado, con el uso del acercamiento orientado al objeto. Esta herramienta apoya los estándares más altos de las notaciones de Java y de UML. Genera productos de calidad, soporta aplicaciones web y es fácil de instalar y actualizar. El análisis de las transiciones de diseño y, a continuación, a la aplicación se encuentran perfectamente integradas en la herramienta CASE, reduciendo así significativamente los esfuerzos en todas las etapas del ciclo de vida de desarrollo de software (15). Este software de modelado ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, generar código inverso, código desde diagramas y documentación.

Características del Visual Paradigm:

- Entorno de creación de diagramas para UML 2.0.
- Diseño centrado en Casos de Uso y enfocado al negocio que genera un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de integrarse en los principales IDEs. (Integrated Development Environment)
- Disponibilidad en múltiples plataformas.
- Exporta código de alrededor de 10 lenguajes de programación incluyendo el PHP.

1.6. Herramientas de Desarrollo.

Otras herramientas a utilizar para el desarrollo de esta investigación son las siguientes.

Capítulo 1. Fundamentación Teórica

1.6.1. Servidor Web Apache.

Apache es un software libre, servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras. Presenta, entre otras características, mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido. Ha sido desde Abril de 1996 el servidor HTTP más usado. Se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Características de Apache (16)

- ✓ Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- ✓ Es una tecnología gratuita de código fuente abierto.
- ✓ Es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache. Actualmente existen muchos módulos para Apache que son adaptables a éste, y están ahí para instalarlos cuando se necesite. Otra cosa importante es que cualquiera que posea una experiencia decente en la programación de C o Perl puede escribir un módulo para realizar una función determinada.
- ✓ Trabaja con gran cantidad de Perl, PHP y otros lenguajes de script. Perl destaca en el mundo del script y Apache utiliza su parte del pastel de Perl tanto con soporte CGI como con soporte mod perl. También trabaja con Java y páginas jsp. Teniendo todo el soporte que se necesita para tener páginas dinámicas.
- ✓ Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.
- ✓ Tiene una alta configurabilidad en la creación y gestión de logs. Apache permite la creación de ficheros de log a medida del administrador, de este modo puedes tener un mayor control sobre lo que sucede en el servidor.

Capítulo 1. Fundamentación Teórica

1.6.2. Navegadores Web.

Los **navegadores** pueden considerarse como una interfaz de usuario universal. Dentro de sus funciones están la petición de las páginas Web, la representación adecuada de sus contenidos y la gestión de los posibles errores que se puedan producir.

En su mayoría están dotados de posibilidades de ejecución de programas de tipo script, con modelos de objetos que permiten manipular los contenidos de los documentos. Estos lenguajes de programación son VBScript, JScript (ambas de Microsoft) y JavaScript (de Netscape), y proporcionan las soluciones llamadas del lado del cliente, client side y permiten realizar validaciones de datos recogidos en las páginas antes de enviarlos al servidor y proporcionan un alto grado de interacción con el usuario dentro del documento.

Otras de las posibilidades de los navegadores es la gestión del llamado HTML dinámico (Dinamic HTML, DHTML). Éste está compuesto de HTML, hojas de estilo en cascada, (Cascade Style Sheets, CSS), modelo de objetos y scripts de programación que permiten formatear y posicionar correctamente los distintos elementos HTML.

En esta línea, los navegadores han ido un poco más allá y permiten las visualizaciones de documentos XML (eXtensible Markup Language) después de haber sido transformado adecuadamente a HTML por las hojas de estilo extensibles (eXtensible Style Sheets, XSL). De esta manera se puede elegir visualizar ciertos elementos y otros no, dependiendo de las circunstancias.

Ejemplos de navegadores son el Internet Explorer y Mozilla Firefox, ambos, muestran cada vez mayores mejoras y un buen funcionamiento, de fácil usabilidad, portabilidad, eficiencia.

1.6.3. Adobe Dreamweaver.

Dreamweaver es un editor visual profesional para la creación de sitios y páginas web. Con Dreamweaver resulta fácil crear y editar páginas compatibles con cualquier explorador y plataforma.

Presenta validaciones dinámicas en distintos navegadores, es decir que las etiquetas HTML y las reglas CSS, siempre estarán legibles, siendo éstas compatibles con cualquier navegador. El soporte para CSS es más amplio y la edición gráfica, como por ejemplo rotar, recortar o redimensionar, integrada a la interfaz de Dreamweaver, sin tener que salir del programa.

Capítulo 1. Fundamentación Teórica

Tiene implementado el uso de SFTP, es decir el FTP seguro, así como el soporte para tecnologías más sofisticadas como ColdFusion, J2EE, PHP, .NET, espacios de nombre XML, objetos de control de formulario ASP.NET, nuevo contenido de referencia y comportamientos de servidor PHP.

1.7. Patrones a utilizar.

Un patrón describe un problema que ocurre una y otra vez en el entorno y describe también el núcleo de la solución al problema, de forma que puede reutilizarse continuamente(17). Existen diversas clases de patrones pero se utilizarán los patrones de arquitectura y los patrones de diseño para diseñar la aplicación web que concierne a esta investigación.

1.7.1. Patrones de Arquitectura.

La arquitectura a utilizar fue definida por el grupo de arquitectura MINSAP-MIC para todos los programas implementados que se despliegan en INFOMED utilizando los patrones arquitectónicos. Un patrón de arquitectura de software describe un problema particular y recurrente del diseño, que surge en un contexto específico, y presenta un esquema genérico y probado de su solución.

A continuación se muestran los patrones seleccionados con algunas de las características particulares.

Modelo-Vista-Controlador.

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos: el Modelo, la Vista y el Controlador.

El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, la modelo se encarga de gestionar el acceso a los datos y la Lógica de negocio y el controlador es el responsable de recibir los eventos de entrada desde la vista e invocar cambios en la modelo, es la intermediaria entre la vista y la modelo (18).

Arquitectura en tres capas.

Una buena arquitectura de software debe facilitar los requerimientos de mantenimiento, reusabilidad, escalabilidad y robustez del mismo. Al concertar la solución de un problema como una serie de capas, cada capa debe ocuparse de un subconjunto de responsabilidades altamente cohesionadas y tener poco acoplamiento con las demás. Los cambios internos en cualquier capa deben ocasionar la menor cantidad

Capítulo 1. Fundamentación Teórica

posible de cambios en las restantes (19). Capa de presentación: es la que presenta el sistema al usuario, le comunica la información y captura la información del usuario dando un mínimo de proceso. Esta capa se comunica únicamente con la capa de negocio.

Capa de negocio: es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio o lógica del negocio pues es donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él.

Capa de datos: es donde residen los datos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Una ventaja evidente de este modelo es que la capa de presentación puede desarrollarse de variadas maneras simultáneas; dígase cliente Web, aplicación Windows, aplicación para otro SO, etc. mientras menos responsabilidades recaigan en esta capa tanto mayor será la facilidad de desarrollar múltiples versiones de la misma. Otra ventaja sería la posibilidad de migrar de servidor de bases de datos con un mínimo de cambios en el sistema, en tal caso, los cambios se concentrarían en la capa de datos, quizás hubiera que hacer pequeños ajustes en la capa de negocio, pero nunca en la capa de presentación.

Orientado a Servicios (SOA).

La Arquitectura Orientada a Servicios (en inglés Service-Oriented Architecture o SOA), es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requerimientos de software del usuario. SOA proporciona una metodología y un marco de trabajo para documentar las capacidades de negocio y puede dar soporte a las actividades de integración y consolidación (20).

Cuando se menciona esta arquitectura se está hablando de un juego de servicios residentes en Internet o en una intranet. Existen un grupo de estándares que se encuentran ligados a los servicios web. Donde se incluyen los siguientes (21):

- XML (Extensible Markup Language), lenguaje de marcas extensible. Permite definir la gramática de lenguajes específicos para diferentes necesidades.

Capítulo 1. Fundamentación Teórica

- HTTP (Hypertext Transfer Protocol) es el protocolo que define la sintaxis y la semántica que utilizan los elementos software de la arquitectura web (clientes, servidores, proxies) para comunicarse. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor.
- SOAP (siglas de Simple Object Access Protocol) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. SOAP es uno de los protocolos utilizados en los servicios Web.
- WSDL (siglas de Web Services Description Language), se utiliza para describir la interfaz pública a los servicios Web. Está basado en XML y describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo.
- UDDI (siglas de Universal Description, Discovery and Integration), es uno de los estándares básicos de los servicios Web cuyo objetivo es ser accedido por los mensajes SOAP y dar paso a documentos WSDL, en los que se describen los requisitos del protocolo y los formatos del mensaje solicitado para interactuar con los servicios Web del catálogo de registros.

Hay que considerar, sin embargo, que un sistema SOA no necesariamente necesita utilizar estos estándares para ser "orientado a servicios" pero es altamente recomendable su uso.

En un ambiente SOA, los nodos de la red hacen disponibles sus recursos a otros participantes en la red como servicios independientes a los que tienen acceso de un modo estandarizado. La mayoría de las definiciones de SOA identifican la utilización de Servicios Web (empleando SOAP y WSDL) en su implementación, no obstante se puede implementar dicha arquitectura utilizando cualquier tecnología basada en servicios.

Arquitectura basada en componentes.

La arquitectura software de una aplicación basada en componentes consiste en uno o un número pequeño de componentes específicos de la aplicación (que se diseñan específicamente para ella), que hacen uso de otros muchos componentes prefabricados que se ensamblan entre sí para proporcionar los servicios que se necesitan en la aplicación (22). En la tecnología de componentes la interfaz constituye el elemento básico de interconectividad. Cada componente debe describir de forma completa las interfaces que ofrece,

Capítulo 1. Fundamentación Teórica

así como las interfaces que requiere para su operación. Y debe operar correctamente con independencia de los mecanismos internos que utilice para soportar la funcionalidad de la interfaz.

Características muy relevantes de la tecnología de programación basada en componentes son la modularidad, la reusabilidad y compatibilidad y en todos ellos coincide con la tecnología orientada a objetos de la que se puede considerar una evolución. Sin embargo, en la tecnología basada en componentes también se requiere robustez ya que los componentes han de operar en entornos mucho más heterogéneos y diversos. El desarrollo de software basado en componentes es la evolución natural de la ingeniería software para mejorar la calidad, disminuir los tiempos de desarrollo y gestionar la creciente complejidad de los sistemas.

1.7.2. Patrones de Diseño.

Los **patrones de diseño** (*design patterns*) son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Un patrón de diseño es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

Experto (Expert).

Lo que plantea este patrón es asignar una responsabilidad al experto en información, es decir, la clase que tiene la información necesaria para cumplir con la responsabilidad. El problema que resuelve el patrón experto está referido al principio más básico mediante el cual las responsabilidades son asignadas en el diseño orientado a objetos.

El experto es usado más que cualquier otro patrón en la asignación de responsabilidades, y en un principio en el diseño orientado a objetos. Hay que notar que el cumplimiento de una responsabilidad requiere información que está esparcida entre clases diferentes de objetos. Esto implica que hay muchos "expertos " parciales que colaboran en la tarea (23). La encapsulación es mantenida, desde que los objetos usan sus propias informaciones para realizar tareas. Esto permite poco acoplamiento, lo cual conduce a sistemas más robustos y de mantenimiento mucho más fácil.

Capítulo 1. Fundamentación Teórica

Alta Cohesión.

La alta cohesión plantea que la información que almacena una clase debe de ser coherente y está en la mayor medida de lo posible relacionada con la clase (24). Tiene entre sus beneficios los mostrados a continuación.

Beneficios

1. Mejoran la claridad del diseño.
2. Simplificación del cambio.
3. Genera bajo acoplamiento.
4. Facilita la reutilización.

Bajo Acoplamiento.

Es asignar una responsabilidad de manera que el acoplamiento permanezca bajo. Una clase tiene bajo acoplamiento si esta no depende demasiado, de otros elementos.

Beneficios

1. No se afectan por cambio en otros componentes.
2. Fáciles de entender por separado.
3. Fáciles de reutilización.

Es la idea de tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clase.

Creador.

El patrón creador ayuda a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases. La nueva instancia deberá ser creada por la clase que: Tiene la información necesaria para realizar la creación del objeto, usa directamente las instancias creadas del objeto y almacena o maneja varias instancias de la clase (25).

Capítulo 1. Fundamentación Teórica

Controlador.

Una clase es controladora cuando se le asigna la responsabilidad de recibir o manejar un mensaje de evento del sistema a una clase que representa una de las siguientes opciones:

1. Representa el sistema global, dispositivo o subsistema.
2. Representa un escenario de Casos de Uso en el que tiene lugar un evento de sistema.

Un controlador es el responsable de gestionar un evento de entrada. Esto facilita la centralización de actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Un error muy común, es asignarle demasiada responsabilidad y alto nivel de acoplamiento con el resto de los componentes del sistema.

Conclusiones

En este capítulo se realizó un análisis de los sistemas que se han desarrollado con el objetivo de lograr un mayor control de la situación epidemiológica en el país, observándose que ninguno de estos sistemas daba solución a la situación problemática planteada.

Con el uso de la metodología y tecnologías propuestas se realizará un mejor diseño del sistema Higiene y Epidemiología dado el avance y desarrollo internacional que adquieren diariamente, además de estar en correspondencia con la arquitectura definida.

Se demostró que el uso de Symfony y YUI, permitirán un manejo de datos más eficiente, rápido, elegante, seguro, configurable, dadas las facilidades de este framework y librerías respectivamente.

Se logró ajustar el desarrollo del sistema planteado a las técnicas propuestas en la arquitectura definida por la Facultad 7 para el Área Temática Sistemas Especializados.

CAPÍTULO 2

CARACTERÍSTICAS DEL SISTEMA

En este capítulo se describe el flujo actual de los procesos de negocio y se realiza un análisis crítico de la ejecución de estos. Se plantea cuál es el objeto de automatización, dando paso al modelado del negocio donde se definen los actores y trabajadores del negocio con una descripción de los mismos y las reglas a tener en cuenta durante todo el proceso. Posteriormente, se hace alusión a los requisitos funcionales y no funcionales con que cuenta el sistema. Finalmente se muestran cuáles son los Casos de Usos del mismo, con una descripción detallada de ellos.

2.1. Objeto de estudio.

2.1.1. Flujo actual de los procesos de negocio.

Todo el proceso de vigilancia epidemiológica inicia una vez que al paciente se le detecta en el Área de Salud síntomas febriles inespecíficos (que no se haya detectado la causa que provoca la fiebre) y se le orienta realizarse los exámenes correspondientes para detectar si tiene alguna de las enfermedades que se encuentran sujetas al Control Sanitario Internacional.

Cuando un paciente acude a consulta con síntomas febriles inespecíficos se le orienta que se realice un examen de sangre al sexto día de presentar los síntomas en su Área de Salud, de la cual se envía una muestra al Laboratorio Provincial de Higiene y Epidemiología (LPHE) y se utiliza dicha muestra para realizar la prueba de IgmSuma, si el resultado de esta es positivo o borderline (en el límite) el paciente pasa a ser un caso sospechoso de Dengue. Posteriormente una muestra de esta misma sangre es enviada al laboratorio del Instituto de Medicina Tropical Pedro Kourí (IPK) para realizar la prueba de Elisa.

Una vez que están los resultados de la misma, si es positivo o borderline, el paciente pasa a ser un caso probable de Dengue y se le orienta la prueba de IGG. A los 21 días se le toma al paciente una segunda muestra de sangre que es enviada al LPHE donde se prepara y se envía al IPK para analizarla, si el resultado de esta prueba es positivo o indeterminado, el paciente pasa a ser un caso confirmado de Dengue y se toman las medidas pertinentes para su tratamiento.

Capítulo 2. Características del Sistema

Otra de las pruebas que se realiza es el PCR/Aislamiento que se les puede hacer a los pacientes sin necesidad de que pasen por el proceso anterior o en el transcurso del mismo, si el resultado de la misma es positivo, el paciente pasa directamente a ser un caso confirmado de Dengue.

Esta prueba evita todo el largo proceso que se menciona con anterioridad, pero tiene el inconveniente de que es muy costosa y el país no tiene el presupuesto para costear que esta se realice a un gran número de personas, es por ello que solo se realiza en casos extraordinarios.

Un aspecto fundamental a tener en cuenta es el arribo de viajeros cubanos, estudiantes extranjeros residentes en Cuba o personal diplomático residente en el país, que llegan a Cuba procedentes de zonas endémicas de dengue, los mismos son controlados como parte de la vigilancia epidemiológica del Programa de Control Sanitario Internacional, una vez que estos arriban al aeropuerto son atendidos por el personal de salud y si algunos son detectados con síntomas febriles inespecíficos se trasladan al IPK, donde pasan por el mismo proceso que los pacientes detectados en el Área de Salud.

En caso de que en el aeropuerto no se le detecten síntomas febriles inespecíficos deben ser visitados por el médico de la familia en las primeras 72 horas después de haber llegado al Área de Salud. El médico de la familia debe mantener una vigilancia estricta de los mismos por la posibilidad de haber estado en contacto con la enfermedad y ser portadores de la misma, por lo que se convierte su presencia en un factor de riesgo local que incrementa la probabilidad de existencia de un brote de Dengue en la zona.

En todo momento se necesita que los niveles del Sistema Nacional de Salud estén al tanto de la situación higiénica epidemiológica del país. El Área de Salud debe saber con frecuencia la cantidad de viajeros que han arribado a su zona, cuál es el estado de sus pacientes y si éstos se encuentran sujetos al Control Sanitario Internacional. (1)

2.1.2. Análisis crítico de los procesos.

En la actualidad el control y vigilancia epidemiológica de los pacientes con posibles enfermedades transmisibles se realiza a través del intercambio de información desde las Áreas de Salud hasta el Vice ministerio de Higiene y Epidemiología, permitiendo a todos los niveles estar informados de la situación epidemiológica del país con determinada periodicidad. Debido a que toda esta información es gestionada por los especialistas en papel, de forma manual, por vía telefónica y en el mejor de los casos es enviada por correo electrónico en tablas de Microsoft Office Excel, impide que se realice el trabajo con la rapidez y

Capítulo 2. Características del Sistema

calidad requerida, dificultándole a los diferentes niveles del Programa de Higiene y Epidemiología la toma oportuna de decisiones.

2.1.3. Objeto de Automatización.

En esta investigación se desea automatizar todo el proceso de gestión de la información referente a la vigilancia epidemiológica al viajero y al control de las enfermedades transmisibles en Cuba, donde se registran los datos pertinentes de los pacientes que sean detectados con síntomas febriles inespecíficos, tanto en el Área de Salud como en el aeropuerto. Una vez que sean registrados los pacientes, a través de la aplicación se les podrán orientar las pruebas pertinentes registrando los resultados de cada una de ellas.

Con este sistema se pretende que todos los niveles del Sistema Nacional de Salud puedan visualizar toda la información referente a aquellas personas que ya estén registradas en el sistema. Por ejemplo: el Centro Municipal de Higiene y Epidemiología visualizará todo lo referente a los pacientes que existen en su municipio, el Centro Provincial de Higiene y Epidemiología visualizará todos los pacientes de su provincia y así sucesivamente hasta el nivel nacional.

2.2. Modelo de Negocio.

El flujo de trabajo Modelado del Negocio se desarrolla principalmente en la fase de Inicio, donde se crea una primera versión que describe el contexto del sistema a construir.

2.2.1. Descripción de los actores y trabajadores del negocio.

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos, con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados.

En esta sección se muestran los diferentes actores y trabajadores con que contará el sistema.

Actores	Descripción
	Paciente que presenta síntomas febriles

Capítulo 2. Características del Sistema

Paciente	inespecíficos, acude a su área de salud donde se le orientan y realizan los exámenes pertinentes en la detección de Dengue.
Viajero	Arriba al país procedente de zonas endémicas de enfermedades transmisibles y es sometido a vigilancia epidemiológica por el personal médico. Se remitirá directamente al IPK si se le detectan síntomas febriles inespecíficos para realizarles las pruebas pertinentes.

Tabla 2.1 Descripción de los actores del negocio.

Los trabajadores del negocio son aquellas personas o sistemas que están involucrados en uno o más procesos del negocio, que participan en ellos.

Trabajadores	Descripción
Médico	Encargado de detectar que el paciente tiene fiebre inespecífica y de orientarles los exámenes pertinentes.
Laboratorista del Área de Salud	Es el que toma las muestras de sangre a los pacientes que sean detectados por el médico con síntomas febriles inespecíficos.
Técnico de laboratorio del LPHE	Es el encargado de realizar la prueba de IgmSuma a partir de las muestras de sangre enviadas desde el área de salud.
Técnico de laboratorio del IPK	Es quien realiza la prueba Elisa, IGG y PCR/Aislamiento a partir de las muestras de sangre enviadas desde el LPHE.

Capítulo 2. Características del Sistema

Personal Frontera	Es el encargado de detectar en el aeropuerto un viajero con síntomas febriles inespecíficos y de enviarlo directamente al IPK.
-------------------	--

Tabla 2.2 Descripción de los Trabajadores del Negocio.

2.2.2. Reglas del Negocio.

El Control Sanitario Internacional describe como políticas que deben cumplirse o condiciones que deben satisfacerse para mantener el control y vigilancia epidemiológica del viajero y las enfermedades transmisibles en Cuba las siguientes:

- El paciente pasa a ser un febril inespecífico cuando el médico no detecta el motivo de la fiebre, por lo que se le orientan las pruebas necesarias en busca de Dengue.
- IgmSuma es la primera prueba que se le orienta al paciente y si el resultado es positivo o borderline entonces el paciente pasa a ser un caso sospechoso de Dengue.
- Si el paciente es un caso sospechoso de dengue entonces se le orienta la prueba de Elisa y si resulta positiva o borderline pasa a ser un caso probable de dengue.
- Si el paciente es un caso probable de dengue se le orienta la prueba de IGG y si el resultado es positivo o indeterminado el paciente pasa a ser un caso confirmado de dengue y se toman las medidas pertinentes para su tratamiento.
- La prueba PCR/Aislamiento solo es aplicada en algunos casos excepcionales sin importar el estado del paciente. En caso de resultar positiva, el paciente inmediatamente se convierte en un caso confirmado de dengue.
- Los viajeros que serán sometidos a vigilancia epidemiológica en el aeropuerto serán aquellos que arriben en un vuelo procedente de zonas endémicas y que además sean estudiantes extranjeros que radican en Cuba, cubanos que cumplen misión en el extranjero o personal diplomático extranjero residente en Cuba. Todos aquellos que se encuentren registrados en el Registro de Ciudadano (RC).

2.2.3. Diagrama de Casos de Uso del Negocio.

Un diagrama de Casos de Uso del negocio representa gráficamente a los procesos del negocio y su interacción con los actores del negocio. A continuación se muestra el diagrama de Casos de Uso del negocio.

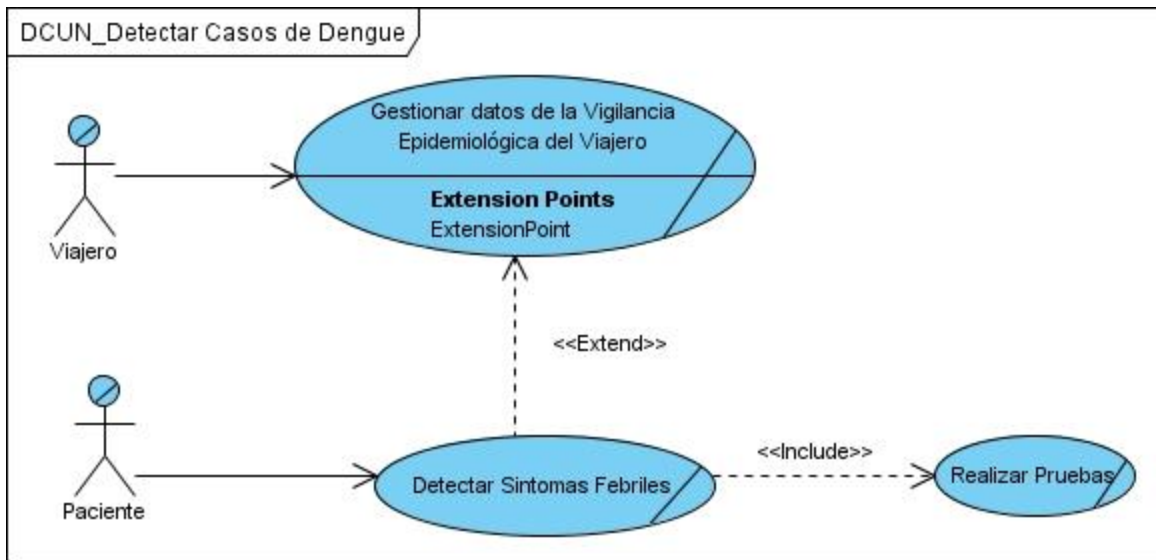


Fig.2.1 Diagrama de Casos de Uso del negocio

2.2.4. Descripción de los Casos de Uso del Negocio.

A continuación se muestra una descripción detallada de los Casos de Uso del negocio que fueron identificados para el control de enfermedades transmisibles en Cuba y la vigilancia epidemiológica al viajero.

Caso de Uso del Negocio	Detectar Síntomas Febriles
Actores	Paciente
Resumen	El caso de uso del negocio se inicia cuando el paciente llega al área de salud, el médico detecta que tiene síntomas febriles inespecíficos y termina cuando el médico

Capítulo 2. Características del Sistema

	orienta realizarse la prueba IgmSuma.
Acción del Actor	Respuesta del proceso del negocio
1. El paciente llega al área de salud.	2. El médico le pide los datos al paciente.
3. El paciente da sus datos.	4. El médico anota los datos y pregunta sobre padecimientos.
5. El paciente informa los síntomas.	6. El médico detecta que tiene fiebre inespecífica y le orienta la prueba de IgmSuma, terminando así el caso de uso del negocio.

Tabla 2.3 Descripción del caso uso del negocio Detectar Síntomas Febriles.

Caso de Uso del Negocio	Realizar Pruebas
Actores	Paciente
Resumen	El caso de uso del negocio se inicia cuando el paciente se dirige al laboratorio del área de salud donde le toman una muestra de sangre, esta muestra de sangre es enviada al LPHE y al IPK donde le realizan las pruebas correspondientes para saber si es un caso confirmado de dengue.
Acción del Actor	Respuesta del proceso del negocio
1. El paciente se dirige al laboratorio del Área de Salud a tomarse la muestra de sangre.	2. El laboratorista del AS le toma la muestra de sangre al paciente y la envía al LPHE.

Capítulo 2. Características del Sistema

	3. El Laboratorista del LPHE realiza la prueba IgmSuma.
4. Si el resultado es positivo o borderline el paciente pasa a ser un caso sospechoso de dengue.	5. El médico lo incluye en una lista como sospechoso de dengue, le orienta la prueba Elisa. Se envía muestra de sangre al IPK para realizar la prueba. Si el resultado es negativo, ir a 10.
6. Si el resultado es positivo o borderline el paciente pasa a ser un caso probable de dengue.	7. El médico lo incluye en una lista como probable de dengue y le orienta la prueba IGG. Se envía muestra de sangre al IPK para realizar la prueba. Si el resultado es negativo, ir a 10.
8. El paciente pasa a ser un caso confirmado de dengue.	9. El médico lo incluye en una lista de casos confirmados de dengue, toma las medidas pertinentes para su tratamiento y termina el CUN.
	10. El médico notifica al paciente y termina el CUN

Tabla 2.4 Descripción del caso de uso del negocio Realizar Pruebas.

Caso de uso del negocio	Gestionar datos de vigilancia epidemiológica del viajero
Actores	Viajero

Capítulo 2. Características del Sistema

Resumen	El caso de uso del negocio se inicia cuando llega un viajero al aeropuerto procedente de zonas endémicas y se le recogen todos los datos al viajero. Luego es sometido a vigilancia epidemiológica por el personal médico y si le detectan algún síntoma es remitido directamente al IPK. El caso de uso del negocio finaliza cuando se le recogen algunos datos al viajero.
Acción del actor	Respuesta del proceso de negocio
1. El viajero arriba al aeropuerto en un vuelo procedente de zonas endémicas.	2. El responsable del aeropuerto le pide los datos al viajero y los registra.
	3. El viajero es sometido a vigilancia epidemiológica por el personal médico del aeropuerto.
	4. Si se detecta algún viajero con síntoma febril, ir a 5. De lo contrario, ir a 6.
	5. El responsable del aeropuerto remite al viajero al IPK y termina el caso de uso del negocio.
6. El viajero se retira y finaliza el caso de uso del negocio.	

Tabla 2.5 Descripción del caso de uso del negocio Gestionar datos de vigilancia epidemiológica del viajero.

2.2.5 Diagramas de Actividades de los Casos de Uso del negocio.

Un diagrama de actividad describe un proceso que muestra el orden de las tareas o actividades que logran los objetivos del negocio. A continuación se muestran los diagramas de actividades de los Casos de Uso del negocio para un mejor entendimiento de los procesos de negocio en cuanto al control de enfermedades transmisibles en Cuba y la vigilancia epidemiológica al viajero.

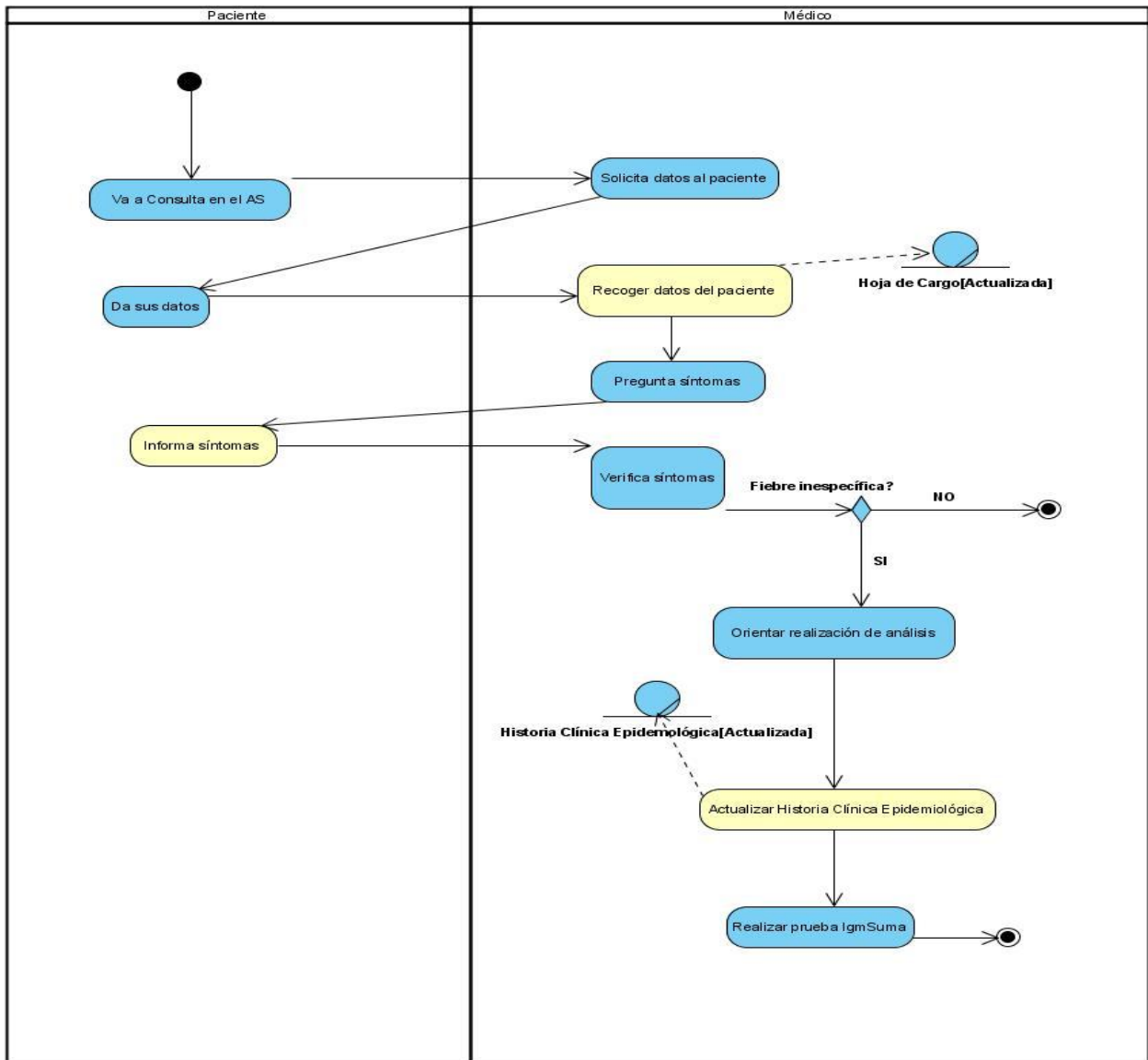


Fig.2.2 Diagrama de actividades CUN Detectar Síntomas Febriles.

Capítulo 2. Características del Sistema

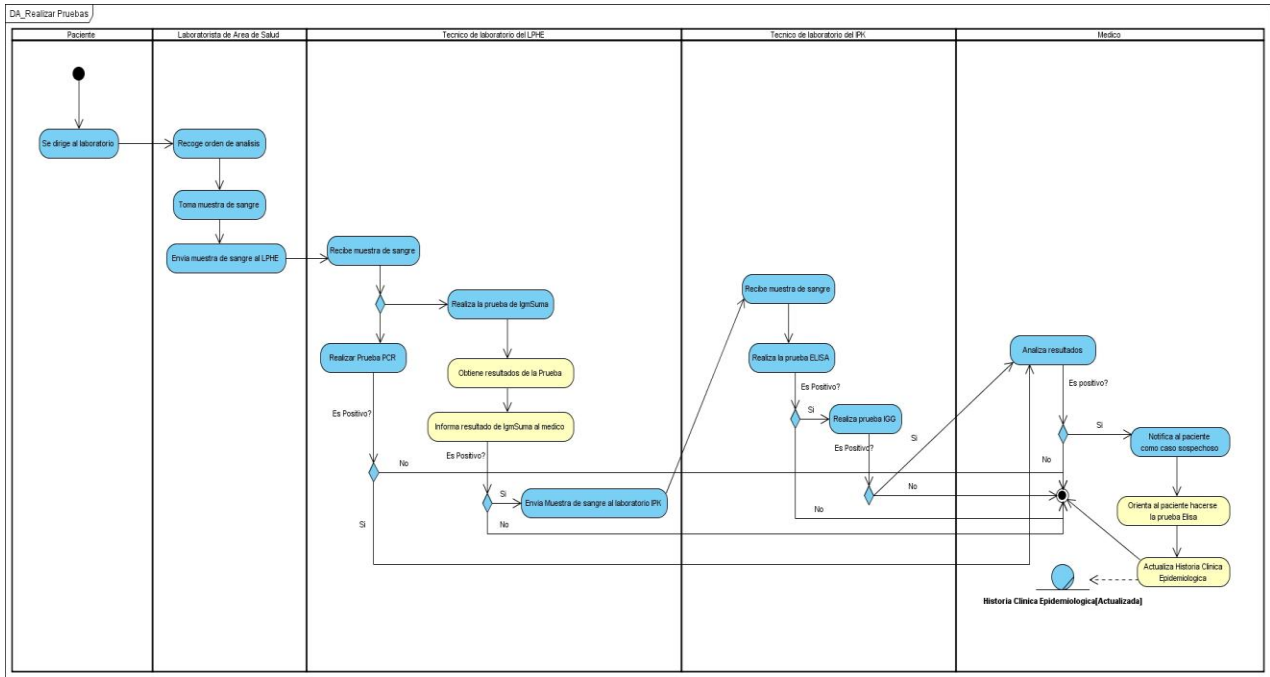


Fig.2.3 Diagrama de actividades CUN Realizar Pruebas.

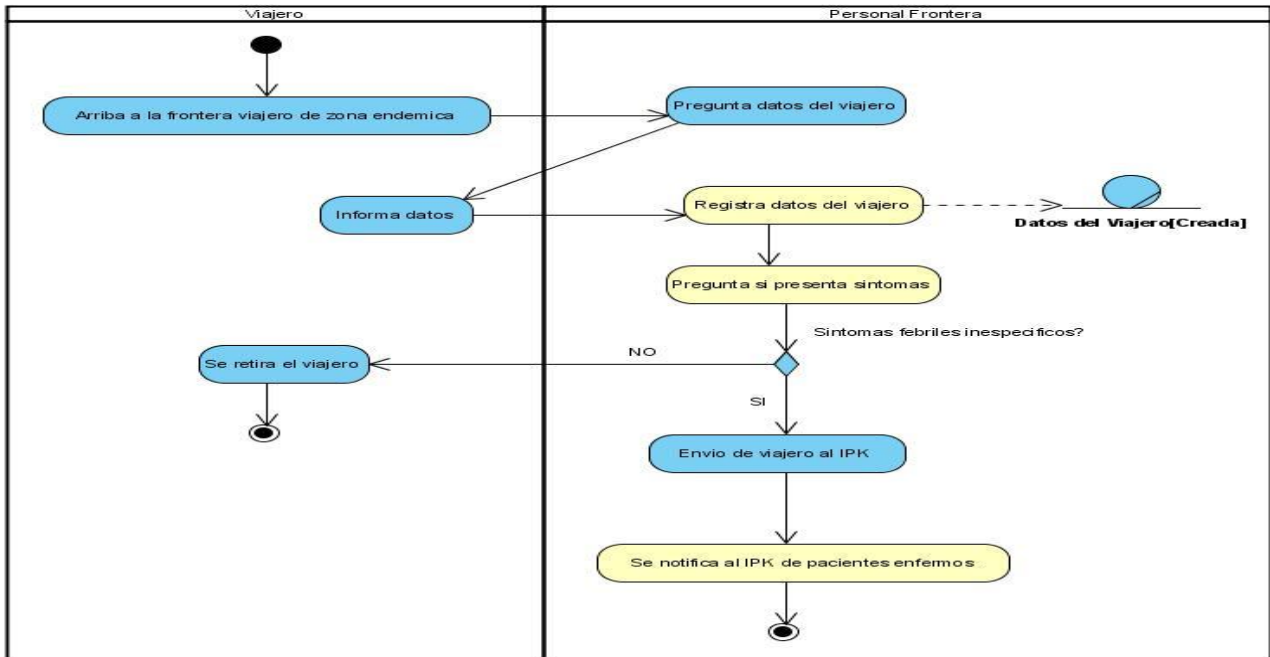


Fig.2.4 Diagrama de actividades Gestionar Datos de Vigilancia Epidemiológica al Viajero.

2.2.6 Modelo de objetos del negocio.

El siguiente modelo de objetos del negocio muestra la relación que existe entre los trabajadores y las entidades del negocio en el control de enfermedades transmisibles en Cuba y la vigilancia epidemiológica al viajero.

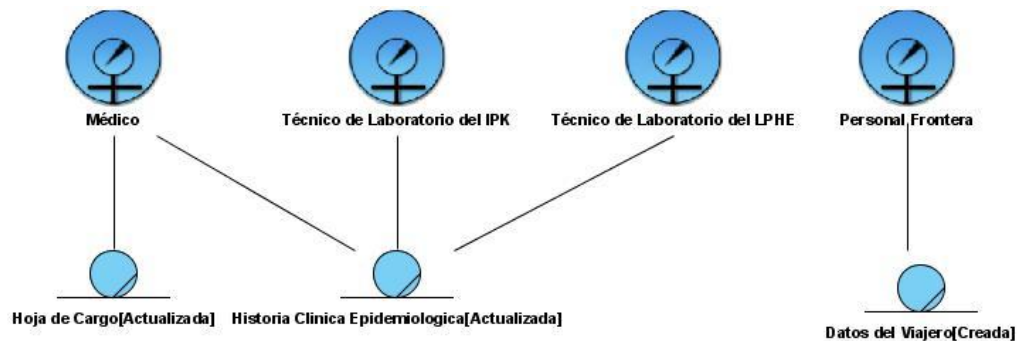


Fig.2.8 Modelo de objetos del negocio.

2.3 Especificación de los requisitos del software.

En el ciclo de desarrollo de un software la determinación de los requisitos constituye una tarea fundamental, pues es donde se define lo que el sistema debe hacer, se identifican las funcionalidades requeridas y las restricciones que se imponen. Los requerimientos pueden dividirse en funcionales y no funcionales.

2.3.1 Requerimientos Funcionales.

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir los cuales van a satisfacer las necesidades de los clientes. Luego de conocer los procesos de negocio y las actividades a automatizar se definieron los siguientes requisitos funcionales:

1. Gestionar datos de pacientes con síntomas febriles inespecíficos.

- 1.1. Adicionar datos de pacientes con síntomas febriles inespecíficos.
- 1.2. Modificar datos de pacientes con síntomas febriles inespecíficos.
- 1.3. Visualizar datos de pacientes con síntomas febriles inespecíficos.
- 1.4. Eliminar datos de pacientes con síntomas febriles inespecíficos.

1.5. Cambiar pacientes con síntomas febriles inespecíficos.

1.6. Registrar pruebas orientadas al febril.

1.7. Buscar listado de pacientes con síntomas febriles inespecíficos.

2. Modificar Estado de Pacientes.

3. Visualizar listado de Pacientes por estados.

4. Visualizar resultado de la prueba IgmSuma.

5. Visualizar resultados de las pruebas realizadas en el IPK.

6. Gestionar resultado de la prueba de IgmSuma.

6.1 Insertar Resultado de la prueba de IgmSuma.

6.2 Modificar Resultado de la prueba de IgmSuma.

6.3 Eliminar resultado de la prueba de IgmSuma.

6.4 Buscar listado de resultado de la prueba de IgmSuma.

7. Visualizar pruebas pendientes a realizar de IgmSuma.

8. Gestionar resultados de las pruebas realizadas en el IPK.

8.1 Insertar Resultados de las pruebas realizadas en el IPK.

8.2 Modificar resultados de las pruebas realizadas en el IPK.

8.3 Eliminar resultados de las pruebas realizadas en el IPK.

8.4 Buscar listado de resultados de las pruebas realizadas en el IPK.

9. Visualizar pruebas pendientes.

10. Gestionar Vuelos.

10.1 Registrar datos de los vuelos.

10.2 Visualizar datos de los vuelos.

10.3 Modificar datos de los vuelos.

10.4 Buscar datos de los vuelos.

10.5 Eliminar datos de los vuelos.

11. Gestionar Viajeros.

11.1 Registrar datos de los viajeros.

11.2 Modificar datos de los viajeros.

11.3 Visualizar datos de los viajeros.

11.4 Eliminar datos de los viajeros.

11.5 Cambiar viajeros.

11.6 Buscar datos de los viajeros.

12. Visualizar listado de la entrada de viajeros al país.

13. Visualizar listado de viajeros enviados al IPK.

14. Visualizar cantidad de casos por municipios.

15. Visualizar listado de entrada de viajeros por municipios.

16. Visualizar cantidad de casos por provincias.

17. Visualizar listado de entrada de viajeros por provincias.

18. Visualizar cantidad de casos del país.

19. Configurar Sistema.

19.1 Mostrar datos de los nomencladores.

19.2 Modificar datos de los nomencladores.

19.3 Adicionar datos de los nomencladores.

19.4 Eliminar datos de los nomencladores.

Nomencladores: *Organismos, País, Aerolínea, Estado Febril, Sexo, Aeropuerto, Motivos de Viaje, Tipos de Laboratorio, Laboratorios, Tipos de Prueba, Resultados Cualitativos.*

2.3.1 Requerimientos no Funcionales.

Los requerimientos no funcionales especifican las propiedades con que contará el sistema, como restricciones del entorno, rendimiento, dependencias de la plataforma, facilidad de mantenimiento, extensibilidad y fiabilidad.

A continuación se muestran los requerimientos mínimos que fueron identificados a través de la investigación realizada.

Apariencia o Interfaz Externa

- Se podrán distinguir colores atractivos y acordes con los recomendados para los softwares de salud.
- Debe poseer un ambiente amigable, intuitivo, sencillo y de fácil navegación, tratando así de impedir el rechazo por parte del usuario al tener que interactuar con un sistema no conocido.
- Paginación de reportes de búsqueda, y listados.
- Diseño perfectamente encuadrado para resoluciones de 1024 x 768, pero preparado para verse en otras resoluciones.

Usabilidad

- La aplicación Web debe ser flexible y de fácil aprendizaje, pues se trata en todo lo posible de mantener un estándar de operabilidad que logre que las interacciones del usuario con el sistema sean predecibles y familiares.
- El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente Web en sentido general.

Soporte

- El sistema contará con una ayuda para facilitar al usuario el manejo de la aplicación.
- Estará bien documentado para garantizar futuros mantenimientos.

Portabilidad

- El sistema podrá ejecutarse sobre plataforma Linux, Windows XP.

Capítulo 2. Características del Sistema

Seguridad.

- La autenticación de los usuarios en el sistema será garantizada por el Sistema de Autenticación, Autorización y Auditoría (SAAA) al cual estará conectada la aplicación.
- Las funcionalidades deberán estar restringidas de acuerdo a los diferentes roles con el objetivo de garantizar que el acceso a la información sea según el nivel de autorización.
- El sistema deberá estar protegido ante el acceso no autorizado a la información.

Software

Cliente:

- El cliente podrá tener acceso al sistema a través de los navegadores Web, Mozilla Firefox 3.X o Internet Explorer 7.0.
- Sistema operativo Linux o Windows XP

Servidor:

- Sistema operativo Linux.
- Servidor Web Apache 2.0 y PHP 5.
- Servidor de Base de Datos PostgreSQL 5.1.
- Framework Symfony.

Hardware

Cliente:

- Procesador Pentium III.
- 256 de memoria RAM.
- Monitor VGA.
- Tarjeta de red.

Servidor:

- Procesador Pentium IV.
- 2GB de memoria RAM.
- Disco Duro de 80 GB.
- Monitor tipo VGA.
- Tarjeta de red.

Diseño

- Se utilizarán los patrones de diseño GRASP.
- Se utilizará como metodología de desarrollo RUP (Rational Unified Process), como lenguaje de modelado UML (Unified Modeling Language) y como herramienta case el Visual Paradigm.
- Para el desarrollo del prototipo no funcional se utilizará como lenguaje de programación PHP y como entorno de desarrollo integrado (IDE) Dreamweaver.

Ayuda y Documentación en línea

- Contará con un manual de usuario para que se pueda explotar al máximo.
- Contará con una ayuda digital, a la cual se podrá acceder desde cualquier parte de la aplicación.

2.4 Modelado del Sistema.

El modelado del sistema es una actividad de gran importancia que se desarrolla dentro el Flujo de trabajo de Requerimientos. El modelo del sistema propuesto representa los requerimientos ya identificados, así como la interacción de los usuarios del sistema con el mismo.

2.4.1 Definición de los actores.

Un actor del sistema es una persona, grupo de personas o sistema que interactúa con la aplicación. Cada trabajador del negocio (inclusive si fuera un sistema ya existente) que tiene actividades a automatizar es un candidato a actor del sistema. Si algún actor del negocio va a interactuar con el sistema, entonces también será un actor del sistema.

Los actores del sistema:

Capítulo 2. Características del Sistema

- No son parte de él.
- Pueden intercambiar información con él.
- Pueden ser un recipiente pasivo de información.
- Pueden representar el rol que juega una o varias personas, un equipo o un mismo sistema.

Actor	Descripción
Responsable del AS/Hospital.	Es el responsable en el área de salud de gestionar todos los datos de un paciente que se detecte como febril inespecífico. Este actor podrá modificar toda la información que desee referente a los pacientes de su AS.
Visualizador en el AS/Hospital.	Es el encargado en el AS de visualizar un listado con todos los datos de los pacientes por el estado deseado, ya sea febril inespecífico, sospechoso, probable o confirmado de dengue. Este actor no podrá realizar ningún cambio en la aplicación.
Responsable LPHE.	Es el responsable en el Laboratorio Provincial de Higiene y Epidemiología de gestionar el resultado referente a las pruebas de IgmSuma.
Responsable en el IPK.	Es el encargado en el Instituto Pedro Kourí de gestionar el resultado referente a las pruebas Elisa, IGG y PCR/Aislamiento.
Responsable en el Municipio.	Es el encargado de visualizar los casos de Dengue y los viajeros de un municipio determinado que arriban al país, procedentes de zonas endémicas de enfermedades transmisibles, así como de visualizar la

Capítulo 2. Características del Sistema

	cantidad de casos de dengue en el Área de Salud.
Responsable de la Provincia.	Es el encargado de visualizar la cantidad de casos de dengue que existe en la provincia a la que pertenece, así como visualizar el listado de los viajeros que arriban al municipio, procedentes de países endémicos de enfermedades transmisibles.
Responsable en el Vice Ministerio de Higiene y Epidemiología.	Es el encargado de visualizar los casos de dengue y el listado de entrada de viajeros, de todas las provincias, que arriban al país, procedentes de zonas endémicas de enfermedades transmisibles, así como visualizar la cantidad de casos de dengue del país.
Responsable del Aeropuerto.	Es el responsable de gestionar toda la información de los vuelos y los viajeros que llegan al país procedente de zonas endémicas.
Visualizador en el Aeropuerto.	Se encarga en el aeropuerto de visualizar toda la información referente a los viajeros que arribaron al país y los que fueron enviados al IPK.
Administrador.	Es el responsable de la configuración del sistema, como por ejemplo: gestionar la información referente a los nomencladores con que cuenta la aplicación.
Registro de Ciudadano (RC)	Registro que se encarga de gestionar los datos de todos los pacientes.
Registro de Unidades de Salud (RUS).	Registro que se encarga de gestionar el nombre de la US en la que se encuentra el usuario y cuál es el nivel al que se subordina.

Registro de Ubicación (RU)	Registro que permite listar los municipios y las provincias.
Sistema de Detención y Control de Vectores.	Módulo del Control Sanitario Internacional que permite obtener dado un paciente la situación entomológica de la manzana en que reside el mismo.

Tabla 2.9 Descripción de los actores del sistema.

2.4.2 Diagrama de Casos de Uso del sistema.

Organización de los Elementos: Casos de Uso en Paquetes.

Debido a la complejidad del Modelo de Casos de Uso del Sistema, se agruparon los casos de uso en subconjuntos más pequeños o paquetes, ofreciendo una vista más detallada para lograr una mejor manipulación y organización de los mismos.

Para agrupar los casos de uso en paquetes se tuvo en cuenta que los elementos estuvieran estrechamente relacionados entre sí.

La relación que se establece entre estos paquetes es de Dependencia, teniendo en cuenta que los elementos de un paquete conocen de alguna forma los del otro y estos a su vez se encuentran acoplados.

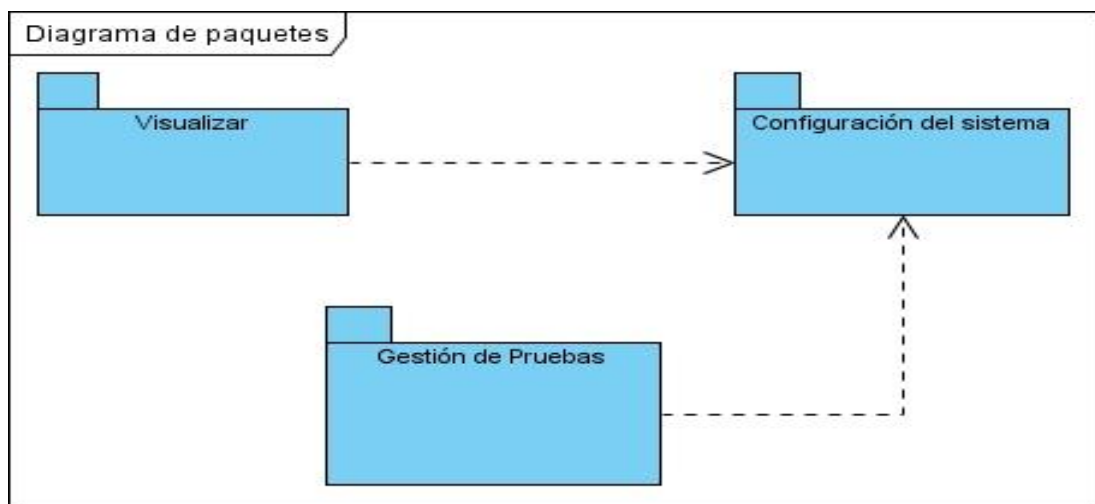


Fig. 2.9 Diagrama de Paquetes

Capítulo 2. Características del Sistema

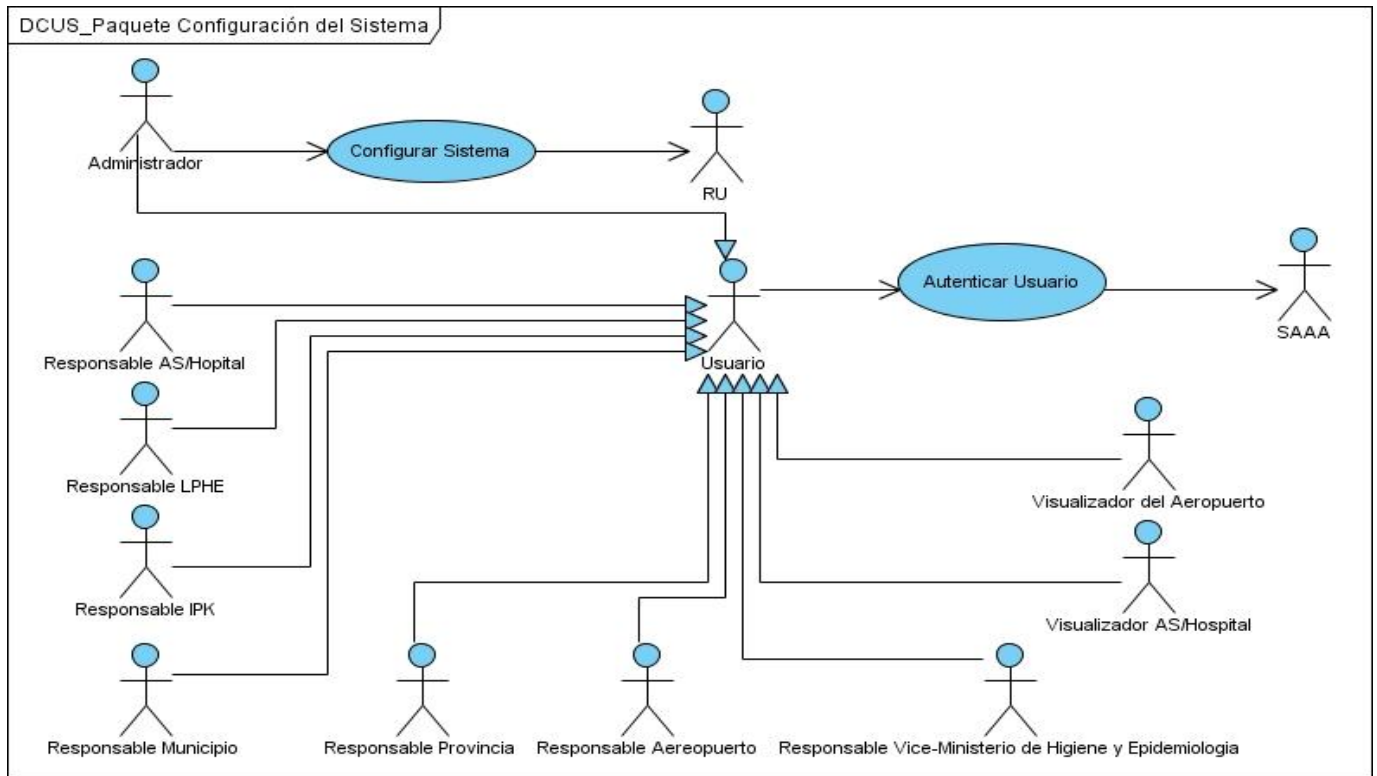


Fig. 2.10 Diagrama de Paquete Configuración del Sistema

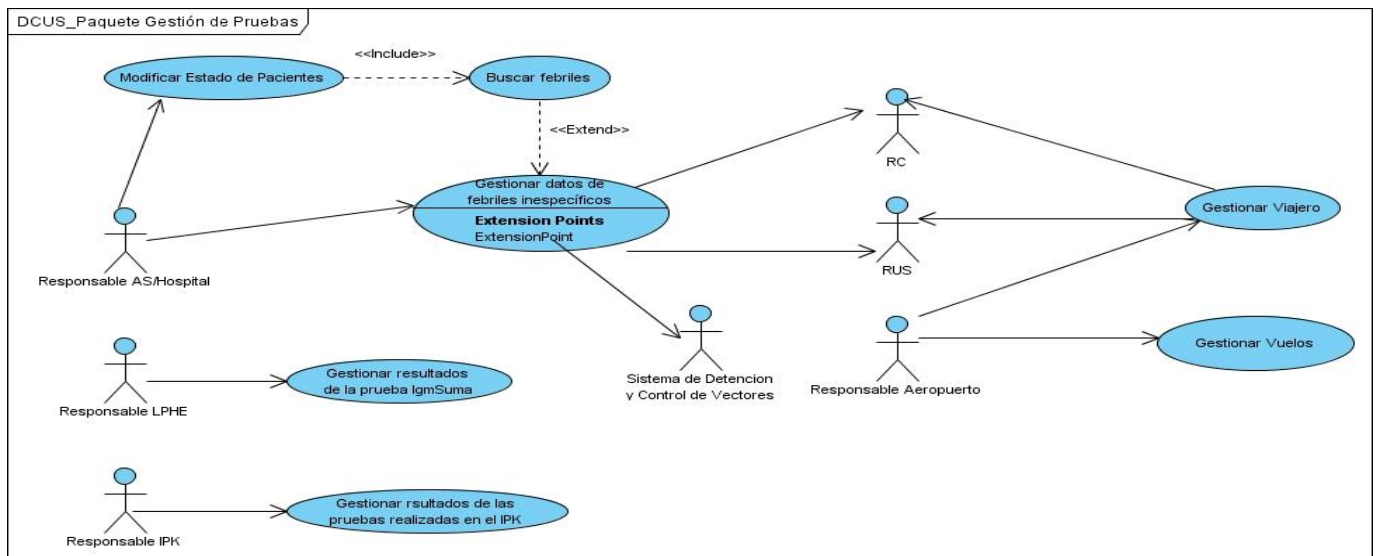


Fig. 2.11 Diagrama de Paquete Gestión de Prueba

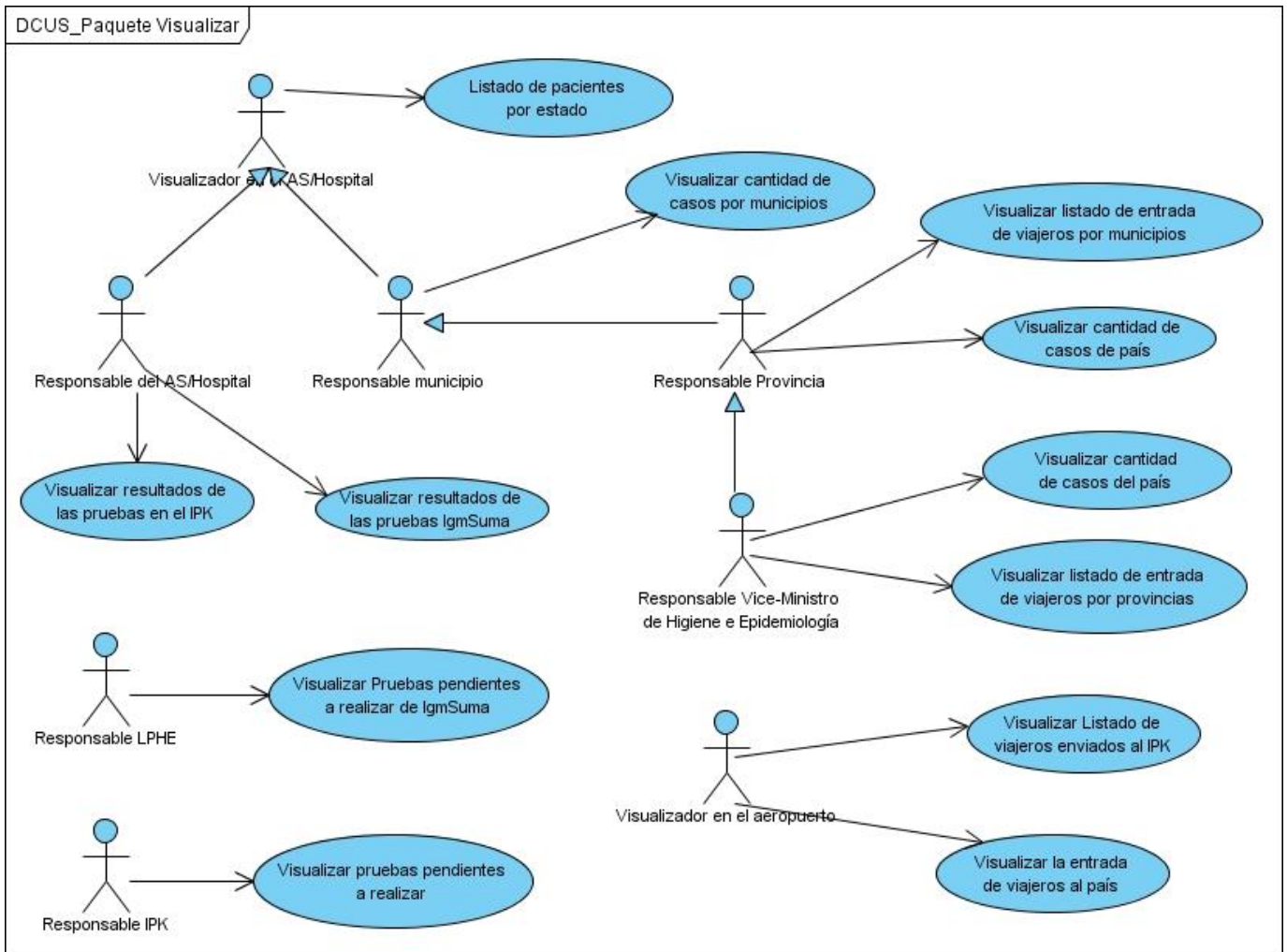


Fig. 2.12 Diagrama de Paquete Visualizar

2.4.3 Listado de Casos de Uso.

Los Casos de Uso son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. Por lo tanto, establece un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema.

A continuación se muestra el listado de los Casos de Usos con una descripción textual para su mejor entendimiento (extendida para los Casos de Uso arquitectónicamente significativos y resumida para el resto).

Capítulo 2. Características del Sistema

CU-1	Modificar Estado de Pacientes
Actor	Responsable del AS/Hospital
Descripción	El caso de uso se inicia cuando el Responsable del AS/Hospital cambie el estado del paciente en caso de haber cometido algún error o después de haber recibido el resultado positivo de alguna prueba orientada al paciente. El caso de uso finaliza una vez actualizados los datos.
Referencia	RF-2

Tabla 2.10 Caso de Uso Modificar Estado de Pacientes.

CU-2	Visualizar listado de Pacientes por estados.
Actor	Visualizador en el AS/Hospital
Descripción	El caso de uso se inicia cuando el Visualizador del Área de Salud desee visualizar el listado de pacientes pertenecientes a su área de salud por estados, ya sea febril inespecífico, sospechoso, probable o confirmado. El caso de uso finaliza cuando se muestre un listado con los resultados en dependencias de los filtros seleccionados.
Referencia	RF-3

Tabla 2.11 Caso de uso Visualizar listado de Pacientes por estados.

CU-3	Visualizar resultados de la prueba IgmSuma.
Actor	Responsable del AS/Hospital

Capítulo 2. Características del Sistema

Descripción	El caso de uso se inicia cuando el responsable del AS busque los resultados de la prueba de IgmSuma perteneciente a un paciente determinado para saber si el paciente es un caso sospechoso de dengue. El caso de uso finaliza cuando se muestre un listado con todos los resultados encontrados.
Referencia	RF-4, RF-6

Tabla 2.12 Caso de Uso Visualizar resultados de las prueba IgmSuma.

CU-4	Visualizar resultados de las pruebas realizadas en el IPK.
Actor	Responsable del AS/Hospital
Descripción	El caso de uso se inicia cuando el responsable del AS busque los resultados de las pruebas realizadas en el IPK ya sea la prueba de Elisa, IGG o PCR/Aislamiento perteneciente a un paciente determinado para saber si es un caso probable o confirmado de dengue. El caso de uso finaliza cuando se muestre un listado con todos los resultados encontrados.
Referencia	RF-5, RF-8

Tabla 2.13 Caso de Uso Visualizar resultados de las pruebas realizadas en el IPK.

CU-5	Visualizar pruebas pendientes a realizar de IgmSuma.
Actor	Responsable del Laboratorio Provincial de Higiene y Epidemiología.
Descripción	El caso de uso se inicia cuando el responsable del LPHE busque las pruebas de IgmSuma que están pendientes por realizar

Capítulo 2. Características del Sistema

	pertencientes a los pacientes de su provincia. El caso de uso finaliza cuando se muestre un listado con todos los resultados encontrados.
Referencia	RF-7

Tabla 2.14 Caso de Uso Visualizar pruebas pendientes a realizar de IgmSuma.

CU-6	Visualizar pruebas pendientes a realizar en el IPK.
Actor	Responsable del Instituto Pedro Kourí.
Descripción	El caso de uso se inicia cuando el responsable del IPK busque las pruebas de IGG, Elisa y PCR/Aislamiento que están pendientes por realizar pertenecientes a los pacientes de todo el país. El caso de uso finaliza cuando se muestre un listado con todos los resultados encontrados..
Referencia	RF-9

Tabla 2.15 Caso de Uso Visualizar pruebas pendientes en el IPK.

CU-7	Visualizar listado de la entrada de viajeros al país.
Actor	Visualizador del Aeropuerto.
Descripción	El caso de uso se inicia cuando el visualizador del aeropuerto busque el listado de viajeros que entraron al país procedente de zonas endémicas según criterios de búsqueda. El caso de uso finaliza cuando se muestre un listado con todos los resultados encontrados.
Referencia	RF-12

Tabla 2.16 Caso de Uso Visualizar listado de la entrada de viajeros al país.

Capítulo 2. Características del Sistema

CU-8	Visualizar listado de viajeros enviados al IPK.
Actor	Visualizador del Aeropuerto.
Descripción	El caso de uso se inicia cuando el visualizador del aeropuerto busque por varios criterios el listado de los viajeros que fueron enviados al IPK a su llegada al país. El caso de uso finaliza cuando se muestre un listado con todos los resultados encontrados.
Referencia	RF-13

Tabla 2.17 Caso de Uso Visualizar listado de viajeros enviados al IPK.

CU-9	Visualizar cantidad de casos del municipio.
Actor	Responsable del Municipio.
Descripción	El caso de uso se inicia cuando el responsable del municipio busque la cantidad de casos que existe en el municipio al cual pertenece. El caso de uso finaliza cuando se muestre un listado con todos los resultados encontrados.
Referencia	RF-14

Tabla 2.18 Caso de Uso Visualizar cantidad de casos del municipio

CU-10	Visualizar listado de entrada de viajeros por municipios.
Actor	Responsable de la Provincia.
Descripción	El caso de uso se inicia cuando el visualizador del aeropuerto busque el listado de viajeros que entraron al país procedente de zonas endémicas según criterios de búsqueda. El caso de uso

Capítulo 2. Características del Sistema

	finaliza cuando se muestre un listado con todos los resultados encontrados.
Referencia	RF-15

Tabla 2.19 Caso de Uso Visualizar listado de entrada de viajeros por municipios

CU-11	Visualizar cantidad de casos de la provincia
Actor	Responsable de la Provincia.
Descripción	El caso de uso se inicia cuando el responsable de la provincia busque la cantidad de casos que existe en la provincia a la cual pertenece. El caso de uso finaliza cuando acceda a otra opción.
Referencia	RF-16

Tabla 2.20 Caso de Uso Visualizar cantidad de casos de la provincia

CU-12	Visualizar listado de entrada de viajeros por provincias.
Actor	Responsable del Vice-Ministerio de Higiene y Epidemiología.
Descripción	El caso de uso se inicia cuando el responsable del Vice-Ministerio de Higiene y Epidemiología busque el listado de los viajeros que entraron al país procedente de zonas endémicas varias provincias. El caso de uso finaliza cuando se muestre un listado con todos los resultados encontrados.
Referencia	RF-17

Tabla 2.21 Visualizar listado de entrada de viajeros por provincias.

CU-13	Visualizar cantidad de casos del país.
-------	--

Capítulo 2. Características del Sistema

Actor	Responsable del Viceministro de Higiene y Epidemiología.
Descripción	El caso de uso se inicia cuando el responsable del Viceministro de Higiene y Epidemiología busca la cantidad de casos que existe en todo el país. El caso de uso finaliza cuando se muestre un listado con todos los resultados encontrados.
Referencia	RF-18

Tabla 2.22 Caso de Uso Visualizar cantidad de casos del país.

2.4.4 Casos de Uso Expandidos del Sistema.

A continuación se muestran las descripciones textuales de los Casos de Usos expandidos del sistema.

Caso de Uso:	Gestionar datos de febriles inespecíficos.
Actores:	Responsable del Área de Salud (inicia), Sistema de Detención y Control de Vectores (sistema externo).
Resumen:	El CU se inicia cuando responsable del AS accede a la aplicación para registrar, modificar, visualizar o eliminar los datos de un paciente febril inespecífico. Podrá también orientarles pruebas al febril. El CU termina cuando el responsable del AS realiza cualquiera de estas acciones.
Precondiciones:	El usuario debe registrarse en la aplicación como responsable del Área de Salud.
Referencias:	RF-1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7
Sección “Registrar datos de pacientes con síntomas febriles inespecíficos.”	
Flujo Normal de Eventos	

Capítulo 2. Características del Sistema

Acción del Actor	Respuesta del Sistema
1. El Responsable del Área de Salud selecciona la opción del menú “Adicionar Febril”.	2. Muestra una pantalla que permite adicionar un febril mostrando algún(os) criterio(s) de búsqueda como nombre, apellidos y carnet de identidad.
3. El Responsable del Área de Salud introduce algún(os) dato(s) de la persona que desea insertar como febril y presiona el botón “Buscar”.	4. Muestra una pantalla con el nombre, el número del carnet y los apellidos de la(s) persona(s) que coinciden con el (los) dato(s) introducidos por el responsable del AS. Toda esta información mostrada es tomada del Registro de Ciudadano (RC).
5. El Responsable del Área de Salud presiona la opción “Seleccionar” de la persona que desea insertar como febril.	6. Muestra una pantalla con el nombre, apellidos y carnet de la persona seleccionada más otros datos que se deben llenar del paciente, como: fecha de la consulta, si es un febril inespecífico, los datos del centro de estudio o trabajo y los datos del control de foco.
7. El Responsable del Área de Salud introduce los datos y presiona el botón “Aceptar”.	8. El sistema valida los datos.
	9. Muestra una pantalla con todos los datos del paciente sin poder ser modificados y termina el caso de uso.
Flujos Alternos	
Acción del Actor	Acción del Sistema
	8.1. Si no se introducen todos los datos o estos están

Capítulo 2. Características del Sistema

	incorrectos, el sistema debe lanzar un mensaje de error y termina el caso de uso.
Sección “Modificar datos de pacientes con síntomas febriles inespecíficos”	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Responsable del Área de Salud presiona la opción “Editar” del paciente seleccionado.	2. Realiza una búsqueda interna de los datos del paciente y los muestra en una pantalla con la posibilidad de modificarlos.
3. El Responsable del Área de Salud modifica los datos deseados y presiona el botón “Aceptar.”	4. El sistema valida los datos.
	5. Muestra una pantalla con todos los datos del febril incluyendo los que fueron modificados y termina el caso de uso.
Flujos Alternos	
Acción del Actor	Acción del Sistema
	4.1. Si no se introducen todos los datos o estos están incorrectos, el sistema debe lanzar un mensaje de error y termina el caso de uso.
Sección “Visualizar datos de pacientes con síntomas febriles inespecíficos.”	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema

Capítulo 2. Características del Sistema

1. El Responsable del Área de Salud presiona la opción “Ver Detalles” del febril deseado.	2. Realiza una búsqueda interna y muestra una pantalla para visualizar todos los datos del febril seleccionado y termina el caso de uso.
Sección “Eliminar datos de pacientes con síntomas febriles inespecíficos.”	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Responsable del Área de Salud solicita la opción “Eliminar” del febril seleccionado.	2. Muestra un mensaje de confirmación, donde se pregunta si realmente desea eliminar el febril seleccionado.
	3. Realiza una búsqueda del febril y lo elimina, terminando el caso de uso.
Sección “Cambiar pacientes con síntomas febriles inespecíficos.”	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Responsable del Área de Salud presiona la opción “Cambiar” del febril seleccionado.	2. Muestra una pantalla para introducir algún(os) criterio(s) de búsqueda de la persona que desea cambiar.
3. El Responsable del Área de Salud introduce algún(os) criterio(s) de búsqueda de la persona que desea cambiar y presiona el botón “Buscar”.	4. Muestra una pantalla con un listado de la(s) persona(as) que coincidan con los criterios introducidos por el usuario.
5. El Responsable del Área de Salud	6. Muestra una pantalla con la persona deseada con

Capítulo 2. Características del Sistema

presiona la opción “Seleccionar “de la persona deseada.	los datos del febril seleccionado y termina el caso de uso.
Sección “Registrar pruebas orientadas al febril.”	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Responsable del Área de Salud presiona el botón “Orientar Pruebas” del febril seleccionado.	2. Muestra una pantalla para introducir tipo de prueba, tipo de laboratorio, número vial y otros datos de la prueba a realizar.
3. El Responsable del Área de Salud introduce los datos de la prueba a realizar y presiona el botón “Aceptar”	4. El sistema valida los datos.
	5. Muestra un mensaje de que la prueba fue enviada al laboratorio seleccionado y termina el caso de uso.
Flujos Alternos	
Acción del Actor	Acción del Sistema
	4.1. Si no se introducen todos los datos o estos están incorrectos, el sistema debe lanzar un mensaje de error y termina el caso de uso.
Sección “Buscar datos de pacientes con síntomas febriles inespecíficos.”	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema

Capítulo 2. Características del Sistema

1. El Responsable del Área de Salud solicita la opción del menú "Buscar Febriles"	2. Aparece una pantalla para buscar los febriles mostrando algún(os) criterios de búsqueda como: nombre, apellidos, carnet de identidad entre otros.
3. El Responsable del Área de Salud introduce uno o varios criterios de búsquedas pertenecientes al febril que desea buscar y presiona el botón "Buscar".	4. Muestra una pantalla con todos los datos del (los) febril(es) que coincida(n) con los criterios de búsqueda introducidos.
Poscondiciones	La aplicación queda actualizada después de realizar cualquiera de los escenarios descritos anteriormente. Especificar resultados por escenarios.

Tabla 2.23 Descripción Expandida del CU Gestionar datos de febriles inespecíficos.

En este Caso de Uso aunque suelen ser parecidas las funcionalidades de *Cambiar* y *Modificar*, no es así. Cuando se escoge la opción *Cambiar* solamente se permite al usuario seleccionar otra persona del listado de las mismas y la acción *Modificar* solamente cambia los datos febriles de una persona determinada.

Caso de Uso:	Gestionar resultado de la prueba de IgmSuma.
Actores:	Responsable del LPHE.
Resumen:	El CU se inicia cuando el responsable del laboratorio provincial de higiene y epidemiología accede a insertar, eliminar, modificar o visualizar el resultado de la prueba de IgmSuma. El caso de uso termina cuando el responsable del LPHE realiza una de las acciones anteriormente planteadas.
Precondiciones:	El usuario debe registrarse en la aplicación como responsable del LPHE.
Referencias:	RF-6.1, 6.2, 6.3, 6.4, 6.5.
Sección "Insertar Resultado de la prueba de IgmSuma."	

Capítulo 2. Características del Sistema

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1 1. El responsable del LPHE selecciona la opción del menú “Ver Pruebas Pendientes”.	2. Muestra una pantalla para buscar las pruebas pendientes por algún(os) criterio(s) de búsqueda como: tipo de prueba, laboratorio y fecha de recibimiento.
3. El responsable del LPHE introduce algún(os) dato(s) de la prueba a la cual desea insertarle el resultado y presiona el botón “Buscar”.	4. El sistema muestra una pantalla con el número del vial, la fecha de envío, el nombre y los apellidos del paciente correspondiente a la(s) prueba(s) que coincidan con los criterios de búsquedas introducidos por el responsable del LPHE.
5. El responsable del LPHE accede a la opción “Insertar Resultados” de la prueba deseada.	6. Muestra una pantalla con los campos que debe llenar para registrar el resultado de la prueba como fecha del resultado, cualitativo, cuantitativo, campo contra valor y valor de corte.
7. El responsable del LPHE introduce todos los datos de la prueba y presiona el botón “Aceptar”.	8. El sistema valida los datos.
	9. Registra el resultado de la prueba, muestra un mensaje que se han registrado dichos resultados y termina el caso de uso.
Flujos Alternos	
Acción del Actor	Acción del Sistema
	8.1 Si no se introducen todos los datos o estos están incorrectos, el sistema debe lanzar un mensaje de

Capítulo 2. Características del Sistema

	error y termina el caso de uso.
Sección “Buscar Resultados de la prueba de IgmSuma.”	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El responsable del LPHE selecciona la opción del menú “Ver Pruebas Realizadas”.	2. Muestra una pantalla para buscar las pruebas realizadas por algún(os) criterio(s) de búsqueda como: tipo de prueba, tipo de resultado y la fecha en que fue realizada.
3. El responsable del LPHE introduce algún(os) dato(s) de la prueba que desea buscar y presiona el botón “Buscar”.	4. El sistema muestra una pantalla con el número vial, fecha en que fue enviada la muestra, tipo de prueba, fecha resultado, cuantitativo, cualitativo, valor de corte y campo contra valor de las pruebas que coinciden con el(los) criterio(s) de búsqueda introducido(s) y termina el caso de uso.
Sección “Eliminar Resultados de la prueba de IgmSuma”	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El responsable del LPHE selecciona la opción “Eliminar” de la prueba deseada.	2. Muestra un mensaje preguntando si realmente se desea eliminar la prueba.
	3. Se elimina la prueba y se muestra un mensaje de que el resultado de la prueba ha sido eliminado y termina el caso de uso.

Capítulo 2. Características del Sistema

Sección “Modificar Resultado de la prueba de IgmSuma.”	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El responsable del LPHE selecciona la opción “Editar” de la prueba deseada.	2. El sistema realiza una búsqueda de la prueba seleccionada y muestra una pantalla con la fecha del resultado, cualitativo, cuantitativo, campo contra valor y valor de corte con la posibilidad de modificarlos.
3. El responsable del LPHE modifica el(los) dato(s) deseado(s) y presiona el botón “Aceptar”.	4. El sistema valida los datos.
	5. Modifica los resultados de la prueba, muestra una pantalla con el resultado de la prueba modificado y termina el caso de uso.
Flujo Alternos	
Acción del Actor	Respuesta del Sistema
	4.1. Si no se introducen todos los datos o estos están incorrectos, el sistema debe lanzar un mensaje de error y termina el caso de uso.
Flujos Alternos	
Acción del Actor	Acción del Sistema
	4.1. Si no se introducen todos los datos o estos están incorrectos, el sistema debe lanzar un mensaje de

Capítulo 2. Características del Sistema

	error y termina el caso de uso.
Poscondiciones	La aplicación queda actualizada después de realizar cualquiera de los escenarios descritos anteriormente.

Tabla 2.24 Descripción expandida del CU Gestionar resultado de la prueba de IgmSuma.

Caso de Uso:	Gestionar resultado de las pruebas realizadas en el IPK.
Actores:	Responsable del IPK
Resumen:	El CU se inicia cuando el responsable del IPK accede a la aplicación para insertar, eliminar, modificar o visualizar el resultado de las pruebas de Elisa, IGG o PCR/Aislamiento. El caso de uso termina cuando el responsable del IPK realiza una de las acciones anteriormente planteadas.
Precondiciones:	El usuario debe registrarse en la aplicación como Responsable del IPK.
Referencias:	RF-8.1, 8.2, 8.3, 8.4, 8.5.
Sección “Insertar Resultado de las pruebas realizadas en el IPK.”	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El responsable del IPK selecciona la opción del menú “Ver Pruebas Pendientes”.	2. El sistema muestra una pantalla para buscar las pruebas pendientes por algún criterio de búsqueda.
3. El responsable del laboratorio introduce algún(os) criterio(s) de búsqueda de la prueba que desea insertar los resultados y presiona el botón	4. El sistema muestra una pantalla con el número del vial, fecha en que fue enviada la muestra, tipo de prueba y otros datos de la(s) prueba(s) que coincida(n) con el(los) criterio(s) de búsqueda

Capítulo 2. Características del Sistema

“Buscar”.	insertado(s) por el técnico del IPK.
5. El responsable del IPK accede a la opción “Insertar Resultados” de la prueba deseada.	6. Muestra una pantalla con fecha de resultado, cualitativo, cuantitativo, campo contra valor y valor de corte de la prueba seleccionada.
7. El responsable del IPK llena todos los datos de la prueba seleccionada y presiona el botón “Aceptar”.	8. El sistema valida los datos.
	9. El sistema registra los resultados, muestra un mensaje de que se han registrado dichos resultados y termina el caso de uso.
Flujos Alternos	
Acción del Actor	Acción del Sistema
	8.1 Si no se introducen todos los datos o estos están incorrectos, el sistema debe lanzar un mensaje de error y termina el caso de uso.
Sección “Modificar Resultados de las pruebas realizadas en el IPK.”	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El responsable del IPK selecciona la opción “Editar”, de la prueba deseada.	2. Muestra una pantalla con la fecha del resultado, si es cualitativo, cuantitativo, campo contra valor y valor de corte de la prueba que desea modificar.
3. El responsable del IPK modifica el(los) resultado(s) que desea y presiona el	4. El sistema valida los datos.

Capítulo 2. Características del Sistema

botón "Aceptar".	
	5. El sistema modifica los datos y termina el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4.1. Si no se introducen todos los datos o estos están incorrectos, muestra un mensaje de error y termina el caso de uso.
Sección "Buscar Resultados de las pruebas realizadas en el IPK."	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El responsable del IPK selecciona la opción del menú "Ver Pruebas Realizadas".	2. El sistema muestra una pantalla donde se puede(n) introducir algún(os) criterio(s) de búsqueda de la(s) prueba(s) que se quiere(n) buscar.
3. El responsable del IPK introduce algún(os) criterio(os) de búsqueda de la prueba que desea buscar y presiona el botón "Buscar".	4. El sistema muestra una pantalla con el número de vial, tipo de prueba, fecha del resultado, si es cualitativo, cuantitativo y otros datos de la(s) prueba(s) que coincida(n) con el(los) criterio(s) de búsqueda introducidos por el usuario y termina el caso de uso.
Sección "Eliminar Resultados de las pruebas realizadas en el IPK"	
Flujo Normal de Eventos	

Capítulo 2. Características del Sistema

Acción del Actor	Respuesta del Sistema
1. El responsable del IPK selecciona la opción "Eliminar" de la prueba deseada.	2. El sistema muestra un mensaje preguntando si realmente desea eliminar el resultado.
	3. El sistema elimina el resultado de la prueba, muestra un mensaje que ha sido eliminado dicho resultado y termina el caso de uso.

Tabla 2.25 Descripción expandida de CU Gestionar resultado de las pruebas realizadas en el IPK.

Caso de Uso:	Gestionar Vuelos.
Actores:	Responsable del Aeropuerto.
Resumen:	El CU se inicia cuando el responsable del aeropuerto accede a registrar, modificar, eliminar, visualizar o buscar los datos de un vuelo procedente de algún país endémico de enfermedades transmisibles. El CU termina cuando se el responsable del aeropuerto realiza alguna de las acciones anteriores.
Precondiciones:	El usuario debe registrarse en la aplicación como Responsable en el Aeropuerto, sea provincial o municipal.
Referencias:	RF-10.1, 10.2, 10.3, 10.4, 10.5.
Sección "Registrar datos de los vuelos"	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El responsable del aeropuerto solicita la opción del menú "Insertar Vuelo".	2. Muestra la pantalla para insertar el número del vuelo, matrícula, fecha de arribo, aeropuerto, país de

Capítulo 2. Características del Sistema

	procedencia y otros datos del vuelo.
3. El responsable del aeropuerto introduce los datos del vuelo y presiona el botón "Aceptar".	4. El sistema valida los datos.
	5. Registra los datos del vuelo, muestra una pantalla con los datos del vuelo recién insertado sin poder modificarlos y termina el caso de uso.
Flujos Alternos	
Acción del Actor	Acción del Sistema
	4.1. Si no se introducen todos los datos o estos están incorrectos, el sistema de mostrar un mensaje de error y termina el caso uso.
Sección "Modificar datos de los Vuelos"	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El responsable del aeropuerto presiona la opción "Editar" del vuelo deseado.	2. Muestra los datos del vuelo seleccionado con la posibilidad de modificarlos.
3. Cambia los datos que desea modificar y presiona el botón "Modificar".	4. El sistema valida los datos.
	5. Modifica los datos del vuelo, muestra una pantalla con los datos del vuelo incluyendo los que fueron modificados y termina el caso de uso.
Flujos Alternos	

Capítulo 2. Características del Sistema

Acción del Actor	Respuesta del Sistema
	4.1. Si no se introducen todos los datos o estos están incorrectos, muestra un mensaje de error y termina el caso de uso.
Sección “Buscar datos de los vuelos”	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El responsable de aeropuerto solicita la opción del menú “Buscar Vuelos”	2. Muestra la pantalla para que el usuario pueda buscar un vuelo por uno o varios criterios de búsqueda.
3. Introduce los criterios de búsqueda y presiona el botón “Buscar.”	4. Muestra una pantalla con el número de vuelo, matrícula, fecha de arribo, aeropuerto, país de procedencia y la aerolínea de el(los) vuelo(s) que coincide(n) con el(los) criterio(s) de búsqueda y termina el caso de uso.
Sección “Eliminar datos de los vuelos”	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El responsable del aeropuerto presiona la opción “Eliminar” del vuelo deseado”.	2. El sistema muestra un mensaje preguntando si realmente desea eliminar el vuelo.
	3. Elimina los datos del vuelo, muestra un mensaje informando que el vuelo ha sido eliminado y termina

Capítulo 2. Características del Sistema

		el caso de uso.
Sección “Visualizar datos de los Vuelos”		
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
1. El responsable del aeropuerto presiona la opción “Seleccionar” del vuelo deseado.		2. Muestra una pantalla con el número de vuelo, matrícula, aeropuerto, aerolínea, país de procedencia entre otros datos del vuelo seleccionado sin la posibilidad de modificarlos y termina el caso de uso.
Poscondiciones	El sistema queda actualizado luego de realizar cualquier acción descrita con anterioridad.	

Tabla 2.26 Descripción Expandida del CU Gestionar Vuelo

Caso de Uso:	Gestionar Viajeros.
Actores:	Responsable del Aeropuerto.
Resumen:	El CU se inicia cuando el responsable del aeropuerto accede a registrar, modificar, visualizar, eliminar o buscar los datos de un viajero que ha arribado en algún vuelo procedente de zonas endémicas de enfermedades transmisibles. Estos viajeros solo pueden ser: extranjeros residentes en Cuba, estudiantes extranjeros o cubanos que van a cumplir misiones a otros países. El CU termina cuando el responsable del aeropuerto realiza alguna de las acciones anteriores.
Precondiciones:	El usuario debe registrarse en la aplicación como responsable del aeropuerto, sea provincial o municipal.
Referencias:	RF-11.1, 11.2, 11.3, 11.4, 11.5, 11.6.

Capítulo 2. Características del Sistema

Sección “Registrar datos de los vuelos”	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Al visualizar en la pantalla los datos de un vuelo, el responsable del aeropuerto presiona el botón “Adicionar Viajeros”.	2. Realiza una búsqueda de las personas en el Registro de Ciudadano (RC) y los muestra en una pantalla que permite adicionarlo como viajero con tan solo seleccionarlo.
3. El responsable del aeropuerto introduce algún(os) criterio(s) de búsqueda y presiona el botón “Buscar”.	4. Muestra una pantalla con el carnet, el nombre y los apellidos de la(s) persona(s) que coincidan con el (los) criterio(s) de búsqueda.
5. El responsable del aeropuerto presiona la opción “Seleccionar” de esa persona.	6. Aparece una pantalla para llenar otros datos como: si es remitido al IPK, si es sometido a tarjeta roja, sometido a tarjeta de advertencia, entre otros.
7. El responsable del aeropuerto llena todos los datos y presiona el botón “Aceptar”	8. El sistema valida los datos.
	9. Se adiciona el viajero en ese vuelo con los datos introducidos y termina el caso de uso.
Flujos Alternos	
Acción del Actor	Acción del Sistema
	8.1. Si no se introducen todos los datos necesarios o estos están incorrectos, el sistema debe mostrar un mensaje de error y termina el caso de uso.

Capítulo 2. Características del Sistema

Sección “Modificar datos de los viajeros”	
Precondición	Debe haberse realizado una búsqueda del viajero
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El responsable del aeropuerto presiona la opción “Editar” del viajero seleccionado.	2. Muestra una pantalla con todos los datos del viajero con la posibilidad de modificarlos.
3. El responsable del aeropuerto modifica los datos deseados y presiona el botón “Aceptar”.	4. Valida los datos.
	5. Modifica los datos, muestra una pantalla con todos los datos del viajero seleccionado incluyendo los que fueron modificados y termina el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4.1. Si no se introducen todos los datos necesarios o estos están incorrectos, el sistema debe mostrar un mensaje de error y termina el caso de uso.
Sección “Buscar listado de los viajeros”	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El responsable del aeropuerto solicita la	2. Muestra la pantalla para que el usuario pueda

Capítulo 2. Características del Sistema

opción del menú “Buscar Viajeros”.	buscar un viajero introduciendo uno o varios criterios de búsqueda.
3. El responsable del aeropuerto introduce los criterios de búsqueda y presiona el botón “Buscar.”	4. Muestra nombre, apellidos, provincia, municipio, procedencia y dirección de el (los) viajero(s) que coincida(n) con los datos introducidos y termina el caso de uso.
Sección “Eliminar datos de los viajeros”	
Precondición	Debe haberse realizado una búsqueda del viajero
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El responsable del aeropuerto presiona la opción “Eliminar” del vuelo deseado”.	2. El sistema muestra un mensaje preguntando si realmente desea eliminar los datos del viajero.
	3. Elimina los datos del viajero, muestra un mensaje informando que el viajero ha sido eliminado y termina el caso de uso.
Sección “Visualizar datos de los viajeros”	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El responsable del aeropuerto presiona la opción “Ver Detalles” del viajero deseado.	2. Muestra una pantalla con el carnet, nombre, apellidos, edad, sexo, país de procedencia, motivo del viaje y otros datos del viajero seleccionado y termina el caso de uso.
Sección “Cambiar viajeros”	

Capítulo 2. Características del Sistema

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El responsable del aeropuerto presiona la opción “Cambiar” del viajero deseado.	2. Muestra una pantalla para introducir algún criterio de búsqueda de una persona.
3. El responsable del aeropuerto introduce los criterios de búsqueda y presiona el botón “Buscar”.	4. Muestra una lista con las personas que coincidan con los criterios de búsqueda.
5. El responsable del aeropuerto presiona la opción “Seleccionar” de la persona deseada.	6. Cambia el nombre, apellidos y el carnet del viajero seleccionado por estos mismos datos de la persona seleccionada y termina el caso de uso.
Poscondiciones	El sistema queda actualizado luego de realizar cualquier acción descrita con anterioridad.

Tabla 2.27 Descripción expandida de CU Gestionar Viajeros.

Caso de Uso:	Configurar Sistema.
Actores:	Administrador del Sistema.
Resumen:	El CU se inicia cuando el administrador accede a insertar, modificar, visualizar o eliminar cualquier nomenclador (<i>Organismos, País, Aerolínea, Estado Febril, Sexo, Aeropuerto, Motivos de Viaje, Tipos de Laboratorio, Laboratorios, Tipos de Prueba, Resultados Cualitativos</i>). El CU termina cuando el administrador realiza cualquiera de estas opciones.
Precondiciones:	Debe registrarse como administrador del sistema.

Capítulo 2. Características del Sistema

Referencias:	RF-21.1, 21.2, 21.3, 21.4.
Sección “Mostrar datos de los Nomencladores”	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Selecciona en el menú la opción “Configurar Sistema”.	2. Muestra una pantalla para seleccionar el nomenclador con el que desea trabajar.
3. El administrador del sistema selecciona el nomenclador con el cual desea trabajar y presiona el botón “Aceptar”.	4. Muestra los datos que existen del nomenclador seleccionado con la opción de editarlos, eliminarlos y de adicionar uno nuevo.
Flujos Alternos	
Acción del Actor	Acción del Sistema
3.1. El administrador del sistema presiona el botón “Aceptar” sin haber seleccionado ningún nomenclador.	3.2. Muestra un mensaje que debe seleccionar un nomenclador y termina el caso de uso.
Sección “Modificar datos de los Nomencladores”	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El administrador del sistema presiona la opción “Editar” del nomenclador deseado.	2. Muestra una pantalla que permite modificar el nomenclador seleccionado.
3. El administrador del sistema modifica los	4. Modifica los datos del nomenclador

Capítulo 2. Características del Sistema

datos del nomenclador seleccionado y presiona el botón “Aceptar”.	seleccionado y termina el caso de uso.
Sección “Adicionar datos de los nomencladores”	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El administrador del sistema introduce el nuevo dato que quiere agregarle al nomenclador y presiona el botón “Adicionar”.	2. Valida los datos.
	3. Registra los nuevos datos, muestra una pantalla con todos los datos del nomenclador incluyendo el que fue adicionado y termina el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	2.1. Si no se introducen todos los datos o estos están incorrectos, muestra un mensaje de error y termina el caso de uso.
Sección “Eliminar datos de los nomencladores”	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El administrador del sistema presiona la opción “Eliminar” de algún dato del	2. Muestra un mensaje preguntando si realmente desea eliminar la información.

Capítulo 2. Características del Sistema

nomenclador deseado.	
	3. Elimina la información y termina el caso de uso.
Poscondiciones	El sistema queda actualizado luego de realizar cualquier acción descrita con anterioridad.

Tabla 2.28 Descripción expandida del CU Configurar Sistema.

Conclusiones

En este capítulo se profundizó en las características del sistema propuesto que deben resolver los problemas existentes relacionados con los procesos de vigilancia epidemiológica y el control de enfermedades transmisibles. Para el desarrollo de este se modeló el negocio, para ello se elaboró el diagrama de Casos de Uso y una descripción detallada de los actores y trabajadores.

Se desglosaron los requisitos funcionales y no funcionales del sistema propuesto, así como los Casos de Uso del Sistema, los que permiten tener un mayor y mejor conocimiento de las funcionalidades con que contará el sistema a desarrollar.

CAPÍTULO 3 DISEÑO DEL SISTEMA

El presente capítulo describe algunas características del framework a utilizar: Symfony; así como el modelo de diseño de la aplicación, el cual estará en correspondencia con dicho framework. Se muestran además, los diagramas de diseño correspondientes a cada uno de los Casos de Uso del sistema propuesto, así como sus respectivos diagramas de secuencia.

3.1 Modelo de Diseño.

El modelo de diseño es un modelo de objetos que describe la realización de los Casos de Uso, y sirve como una abstracción del modelo de implementación y el código fuente. Es usado como una entrada inicial en las actividades de implementación y prueba.

El Área Temática Sistemas Especializados para la Salud identificó en su arquitectura el uso del framework Symfony lo que hace necesario el conocimiento del mismo para el desarrollo de un buen diseño del sistema.

Symfony

Symfony está basado en un patrón clásico del diseño web conocido como arquitectura MVC, que está formado por tres niveles:

- El Modelo, que representa la lógica de negocio.
- La Vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- El Controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

Cada nivel o capa está compuesto en varias partes:

Modelo

- Abstracción de la base de datos.
- Acceso a los datos.

Vista

- Vista (*Success*).
- Plantilla (*Templates*).
- Layout.

Controlador

- Controlador frontal.
- Acción (*Actions*).

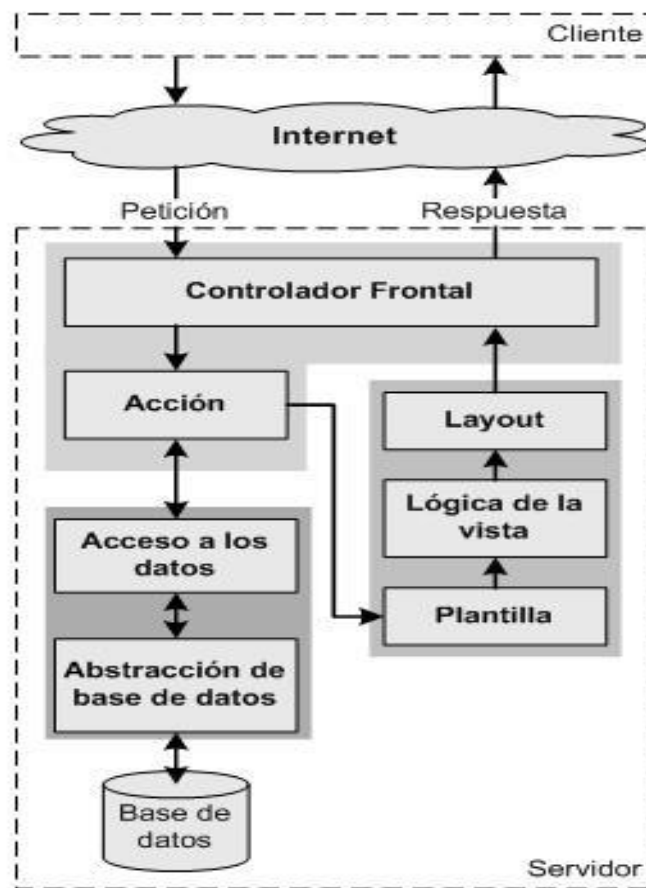


Fig. 3.1: Esquema del MVC en Symfony.

➤ Características del MVC.

El Controlador que contiene el código que une la lógica de negocio con la presentación, está compuesto por el controlador frontal que es el único punto de entrada a la aplicación y las acciones que se encargan de verificar la validez de las peticiones y preparar los datos para la vista.

La Vista está formada por el layout que posee el código común a todas las acciones, las plantillas que contienen el código asociado a cada acción en particular y la lógica que se puede manipular a través de un fichero de configuración sencillo sin necesidad de programarla.

Por último, el Modelo está formado por la capa de acceso a datos cuyas clases son generadas automáticamente por Doctrine (ORM) en función de las estructuras de datos de la aplicación y por la capa de abstracción de la base de datos PDO, que es completamente transparente al programador; así si se desea migrar de bases de datos en cualquier momento, no se debe reescribir el código, siendo necesario solamente modificar un parámetro en un archivo de configuración.

➤ **Patrones de diseño.**

Symfony utiliza las mejores prácticas de diseño para aplicaciones Web. Implementa una serie de patrones GOF (*Gang of Four*).

En la categoría Creacionales:

Singleton (*Instancia única*): Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia.

Abstract Factory (*Fábrica abstracta*): Permite trabajar con objetos de distintas familias de manera que las familias no se mezclen entre sí y haciendo transparente el tipo de familia concreta que se esté usando. Cuando el framework necesita por ejemplo crear un nuevo objeto para una petición, busca en la definición de la factoría el nombre de la clase que se debe utilizar para esta tarea.

En la categoría Estructurales:

Decorator (*Envoltorio*): Añade funcionalidad a una clase dinámicamente. El archivo layout.php, que también se denomina plantilla global, almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el layout, o si se mira desde el otro punto de vista, el layout decora la plantilla.

3.1.1 Diagrama de Clases del Diseño.

Los diagramas de clases son los más utilizados en el modelado de sistemas orientados a objetos. Un diagrama de clases muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Los diagramas de clases son importantes no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables, aplicando ingeniería directa e inversa.

A continuación se muestran algunos de los diagramas de clase del diseño de los Casos de Usos críticos del sistema de Vigilancia Epidemiológica; Gestionar datos de pacientes con síntomas febriles y Configurar Sistema, el resto de los diagramas se podrán encontrar en el **Anexo 1** del documento.

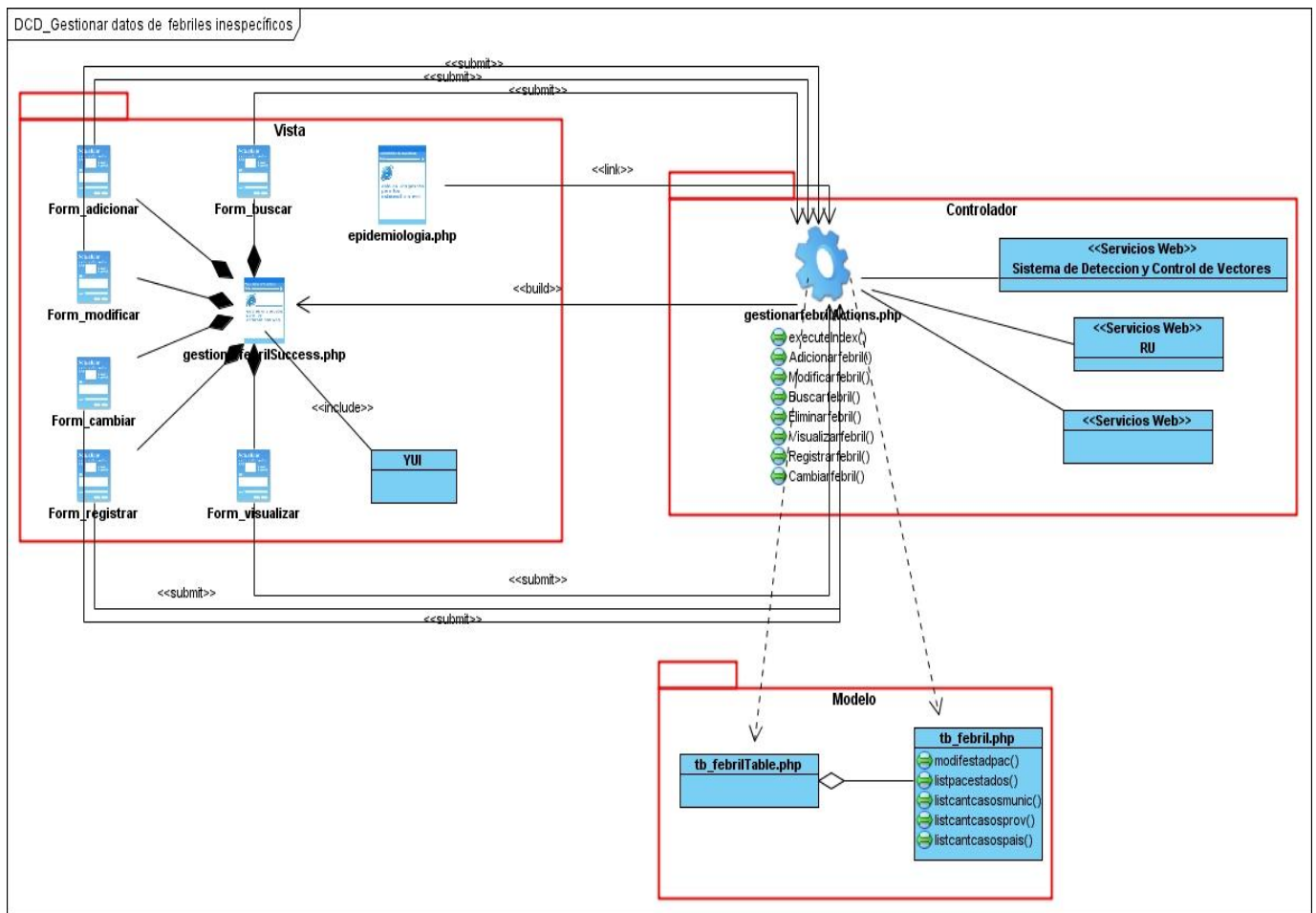


Fig. 3.2 Diagrama de Clases del Diseño Gestionar datos de febriles inespecíficos.

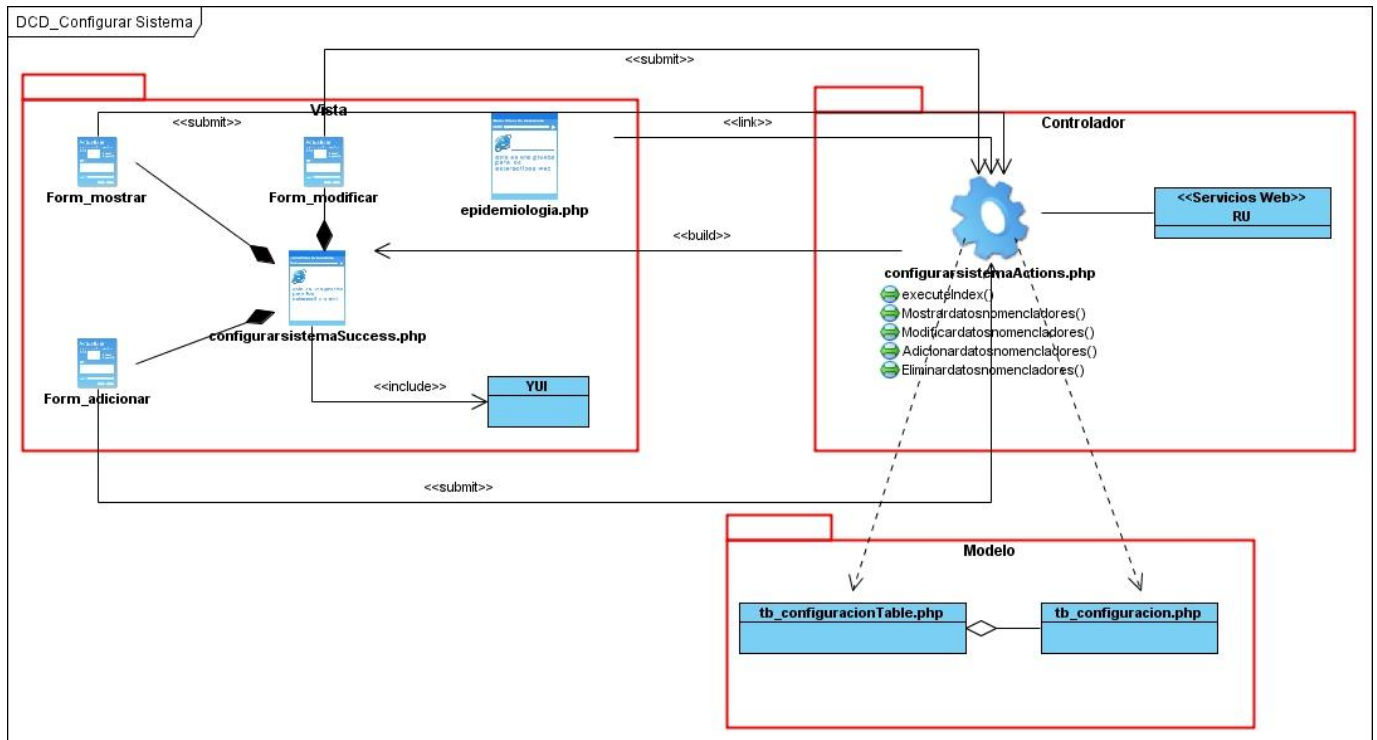


Fig. 3.3 Diagrama de Clases del Diseño Configurar Sistema.

3.1.2 Descripción de Clases del Diseño.

Gestionar datos de febriles inespecíficos: <u>gestionarsebrilActions.php</u>	
Tipo de Clase: <u>Servidora</u>	
Nombre	Descripción
executeIndex()	Ejecuta la vista indeSuccess.php, que contiene todos los formularios para gestionar febriles inespecíficos (Adicionar, Modificar, Buscar, Eliminar, Visualizar, Registrar y Cambiar) y serán cargados usando librerías YUI.
Adicionarfebril()	Solicita a la vista indexSuccess.php mostrar el formulario para adicionar una persona determinada como febril, en la base de datos

	del sistema.
Modificarfebril()	Solicita a la vista indexSuccess.php mostrar el formulario para modificar donde aparecen los datos del febril, que se cambian y envían actualizados a la base de datos del sistema.
Buscarfebril()	Solicita a la vista indexSuccess.php mostrar el formulario para buscar donde se brindan algunos criterios para la realización de la búsqueda, que se efectúa por dicha vía o de no seleccionar ningún criterio se realiza una búsqueda de todos los campos.
Eliminarfebril()	Selecciona al febril y elimina, mostrando como resultado un mensaje de información al usuario.
Visualizarfebril()	Solicita a la vista indexSuccess.php mostrar el formulario para visualizar febriles.
Registrarfebril()	Solicita a la vista indexSuccess.php mostrar el formulario para registrar nuevos febriles al Sistema.
Cambiarfebril()	Solicita a la vista indexSuccess.php mostrar el formulario para cambiar el estado de un febril.

Tabla 3.1 Descripción de la clase Gestionar datos de febriles inespecíficos.

Gestionar resultado de la prueba de IgmSuma: <u>gestionarresultpruebigmsumaActions.php</u>	
Tipo de Clase: <u>Servidora</u>	
Nombre	Descripción
executelIndex()	Ejecuta la vista indeSuccess.php , que contiene todos los formularios para gestionar los resultados de la prueba IgmSuma (Insertar, Modificar, Visualizar, Eliminar, Buscar) y serán cargados

	usando librerías YUI.
Insertarresulpruebigmsuma()	Solicita a la vista indexSuccess.php mostrar el formulario para insertar los resultados.
Modificarresulpruebigmsuma()	Solicita a la vista indexSuccess.php mostrar el formulario para modificar donde aparecen los resultados de las pruebas realizadas, que se cambian y envían actualizados a la base de datos del sistema.
Visualizarresulpruebigmsuma()	Solicita a la vista indexSuccess.php mostrar el formulario para visualizar los resultados de las pruebas IgmSuma y los muestra en una lista.
Eliminarresulpruebigmsuma()	Selecciona un resultado y lo elimina, mostrando como resultado un mensaje de información al usuario.
Buscarresulpruebigmsuma()	Solicita a la vista indexSuccess.php mostrar el formulario para buscar donde se brindan algunos criterios para la realización de la búsqueda, que se efectúa por dicha vía o de no seleccionar ningún criterio se realiza una búsqueda de todos los campos.

Tabla 3.2 Descripción de la clase Gestionar resultado de la prueba de IgmSuma.

Gestionar resultado de la prueba de IgmSuma: <u>gestionarrspruebrealipkActions.php</u>	
Tipo de Clase: <u>Servidora</u>	
Nombre	Descripción
executeIndex()	Ejecuta la vista indeSuccess.php , que contiene todos los formularios para gestionar los resultados de las pruebas realizadas en el IPK (Insertar, Modificar, Visualizar, Eliminar, Buscar) y serán

	cargados usando librerías YUI.
Insertarrespruebrealipk ()	Solicita a la vista indexSuccess.php mostrar el formulario para insertar los resultados.
Modificarrespruebrealipk ()	Solicita a la vista indexSuccess.php mostrar el formulario para modificar donde aparecen los resultados de las pruebas realizadas, que se cambian y envían actualizados a la base de datos del sistema.
Visualizarrespruebrealipk ()	Solicita a la vista indexSuccess.php mostrar el formulario para visualizar resultados de las pruebas del IPK y los muestra en una lista.
Eliminarrespruebrealipk ()	Selecciona un resultado y lo elimina, mostrando como resultado un mensaje de información al usuario.
Buscarrespruebrealipk ()	Solicita a la vista indexSuccess.php mostrar el formulario para buscar donde se brindan algunos criterios para la realización de la búsqueda, que se efectúa por dicha vía o de no seleccionar ningún criterio se realiza una búsqueda de todos los campos.

Tabla 3.3 Descripción de la clase Gestionar resultados de las pruebas realizadas en el IPK.

Gestionar Viajeros: <u>gestionarviajerosActions.php</u>	
Tipo de Clase: <u>Servidora</u>	
Nombre	Descripción
executeIndex()	Ejecuta la vista indeSuccess.php , que contiene todos los formularios para gestionar los viajeros (Registrar, Modificar, Visualizar, Eliminar, Cambiar, Buscar) y serán cargados usando

	librerías YUI.
Registrardatosviajero()	Solicita a la vista indexSuccess.php mostrar el formulario para registrar los datos de un viajero.
Modificardatosviajero()	Solicita a la vista indexSuccess.php mostrar el formulario para modificar donde aparecen los datos del un viajero determinado, permitiéndole al usuario cambiar la información y actualizarlos.
Visualizardatosviajero()	Solicita a la vista indexSuccess.php mostrar el formulario para visualizar los viajeros y los muestra en una lista.
Eliminardatosviajero()	Selecciona un viajero de una lista y lo elimina, mostrando como resultado un mensaje de información al usuario.
Cambiardatosviajero()	Solicita a la vista indexSuccess.php mostrar el formulario para cambiar un determinado viajero.
Buscardatosviajero()	Solicita a la vista indexSuccess.php mostrar el formulario para buscar donde se brindan algunos criterios para la realización de la búsqueda, que se efectúa por dicha vía o de no seleccionar ningún criterio se realiza una búsqueda de todos los campos.

Tabla 3.4 Descripción de la clase Gestionar Viajeros.

Gestionar Viajeros: <u>gestionarvuelosActions.php</u>	
Tipo de Clase: <u>Servidora</u>	
Nombre	Descripción
executeIndex()	Ejecuta la vista indeSuccess.php , que contiene todos los formularios para gestionar los vuelos (Registrar, Modificar,

	Visualizar, Eliminar, Buscar) y serán cargados usando librerías YUI.
Registrardatosvuelo()	Solicita a la vista indexSuccess.php mostrar el formulario para registrar los datos de un vuelo.
Modificardatosvuelo()	Solicita a la vista indexSuccess.php mostrar el formulario para modificar donde aparecen los datos del un vuelo determinado, permitiéndole al usuario cambiar la información y actualizarlos.
Visualizardatosvuelo()	Solicita a la vista indexSuccess.php mostrar el formulario para visualizar los vuelos y los muestra en una lista.
Eliminardatosvuelo()	Selecciona un vuelo de una lista y lo elimina, mostrando como resultado un mensaje de información al usuario.
Buscardatosvuelo()	Solicita a la vista indexSuccess.php mostrar el formulario para buscar donde se brindan algunos criterios para la realización de la búsqueda, que se efectúa por dicha vía o de no seleccionar ningún criterio se realiza una búsqueda de todos los campos.

Tabla 3.5 Descripción de la clase Gestionar Vuelos.

Gestionar Viajeros: <u>configurarsistemaActions.php</u>	
Tipo de Clase: <u>Servidora</u>	
Nombre	Descripción
executeIndex()	Ejecuta la vista indeSuccess.php , que contiene todos los formularios para gestionar los resultados de la prueba IgmSuma (Insertar, Modificar, Visualizar, Eliminar, Buscar) y serán cargados usando librerías YUI.
Adicionardatosnomencladores()	Solicita a la vista indexSuccess.php mostrar el formulario para

	registrar los datos de un vuelo.
Modificardatosnomencladores()	Solicita a la vista indexSuccess.php mostrar el formulario para modificar donde aparecen los datos del un vuelo determinado, permitiéndole al usuario cambiar la información y actualizarlos.
Mostrardatosnomencladores()	Solicita a la vista indexSuccess.php mostrar el formulario para visualizar los vuelos y los muestra en una lista.
Eliminardatosvuelo()	Selecciona un vuelo de una lista y lo elimina, mostrando como resultado un mensaje de información al usuario.

Tabla 3.6 Descripción de la clase Configurar Sistema.

Gestionar datos de febriles inespecíficos	
Tipo de Clase: <u>Vista (Página Cliente)</u>	
Nombre	<u>indexSuccess.php</u>
Descripción	Crea todos los formularios necesarios para gestionar los datos requeridos (Adicionar, Modificar, Buscar, Visualizar, Registrar, Eliminar, Cambiar) con el uso de las librerías YUI, que permiten mostrar solamente el que es solicitado en un momento determinado.

Tabla 3.7 Descripción de la clase indexSuccess.php

Todas las vistas para los diferentes Casos de Uso realizan la misma funcionalidad que la anterior expuesta, por tal razón no cumple objetivo hacer reiterativas las otras descripciones.

3.1.3 Diagramas de Interacción.

Los Diagramas de Interacción muestran una interacción concreta: un conjunto de objetos y sus relaciones, junto con los mensajes que se envían entre ellos.

Características:

- Modelan el comportamiento dinámico del sistema; el flujo de control en una operación.
- Describen la interacción entre objetos; los objetos interactúan a través de mensajes para cumplir ciertas tareas.
- Las interacciones proveen un comportamiento y típicamente implementan un Caso de Uso.
- Existen dos tipos de diagramas de interacción en UML:
 - ✓ **Diagramas de Secuencia (dimensión temporal).**
 - ✓ Diagramas de Colaboración (dimensión estructural).

Un **Diagrama de Secuencia** muestra el comportamiento de un escenario específico de un caso de uso, los eventos que generan los actores externos, el orden y entre los sistemas.

A continuación se muestran una serie de diagramas de secuencia modelados para la realización del problema propuesto, entre ellos los correspondientes al caso de uso Gestionar datos de febriles inespecíficos. El resto se encuentra en el **Anexo 2** del documento.

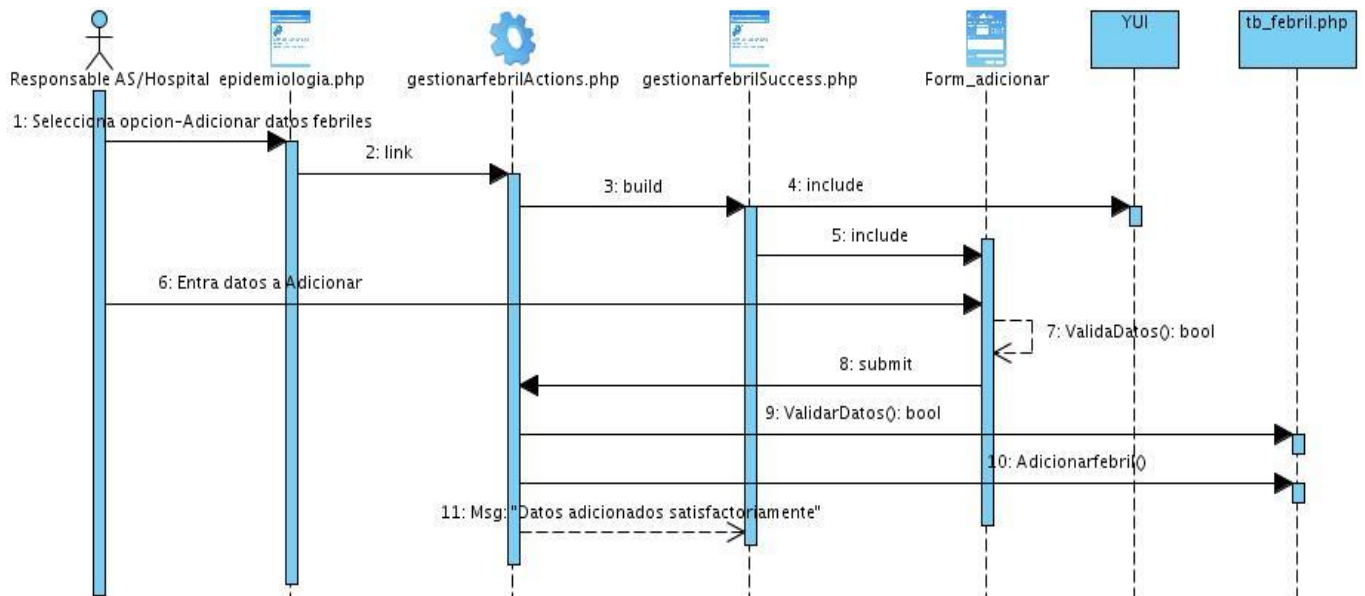


Fig. 3.8 Diagrama de Secuencia Adicionar datos de febriles inespecíficos.

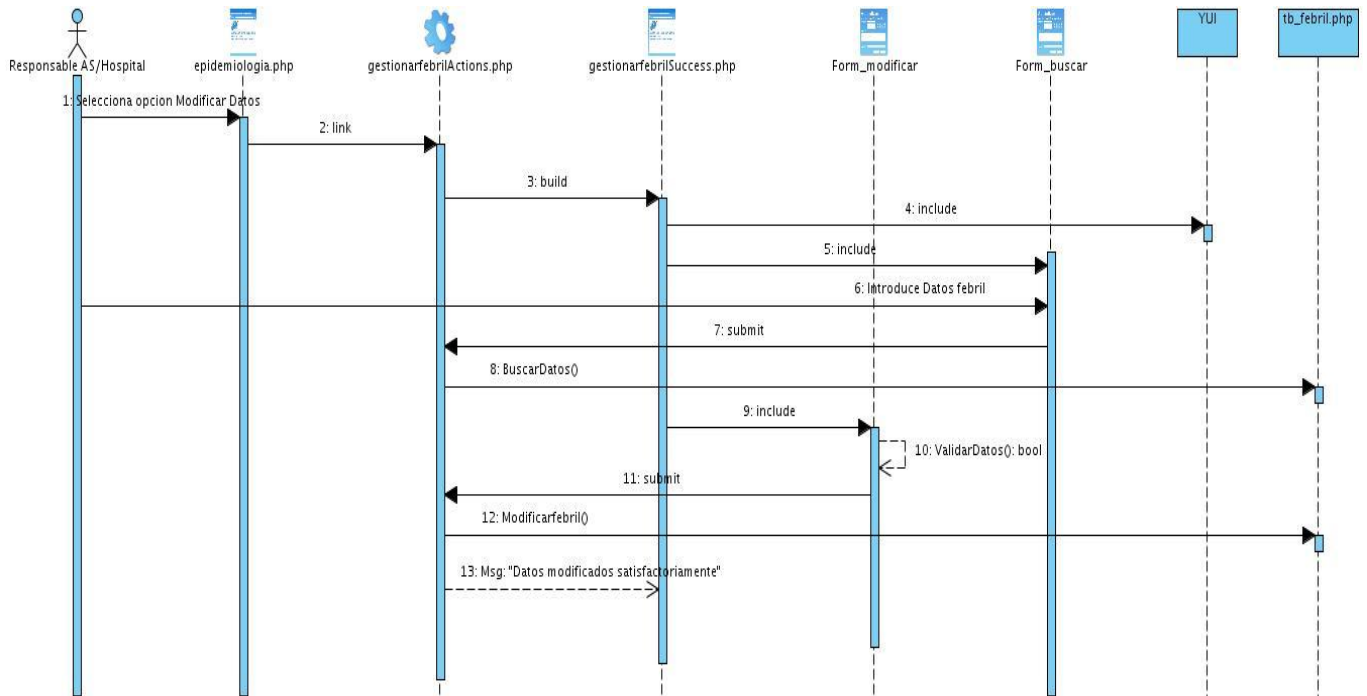


Fig. 3.9 Diagrama de Secuencia Modificar datos de febriles inespecíficos.

3.1.4 Diagrama de despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos). Estarán formados por instancias de los componentes software que representan manifestaciones del código en tiempo de ejecución.

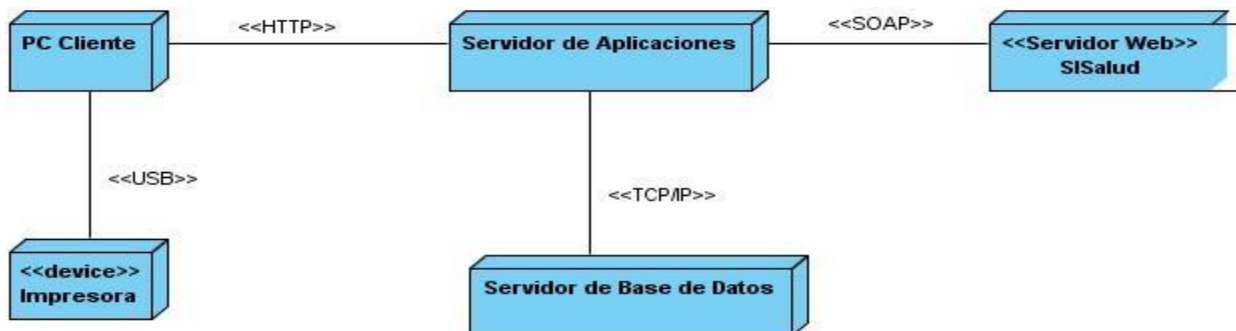


Fig. 3.13 Diagrama de Despliegue.

Conclusiones

En este capítulo se explicaron algunas de las características esenciales del MVC, así como estándares y patrones de diseño que utiliza Symfony, lo que facilita y permite a la vez comprender el mecanismo utilizado para el modelado de diagramas de las clases del diseño y secuencia respectivamente, los que de una manera gráfica argumentan las interacciones que se tendrán con el sistema, así como sus funcionalidades más necesarias, lo que brinda al programador una vista más precisa de cómo debe desarrollarse dicho sistema.

Se realizó el diagrama de despliegue para que se tenga conocimiento de las relaciones entre los componentes hardware y software del sistema.

CONCLUSIONES

Una vez concluido el presente Trabajo de Diploma se lograron cumplir el objetivo y las tareas propuestas:

Se realizó un estudio de las diferentes aplicaciones elaboradas y diseñadas con el objetivo de mantener un control en la vigilancia epidemiológica, donde se demostró que las mismas no se ajustan a las necesidades del Control Sanitario Internacional porque solo se aplican a pequeños y específicos sectores, donde los datos recogidos no cubren las expectativas generales.

Se definieron como herramientas, metodología, lenguaje, framework y librería a utilizar: Visual Paradigm, RUP (Proceso Unificado de Desarrollo), UML (Lenguaje de Modelado Unificado), Servidor Web Apache, el framework Symfony y la librería de javascript YUI (Interfaz de Usuario de Yahoo), que fueron propuestas en la arquitectura de la facultad 7 para el Área Temática Sistemas Especializados. Estas a su vez permitieron modelar los flujos de trabajo propuestos por el Proceso Unificado de Desarrollo: Modelado del Negocio, Gestión de Requerimientos, Análisis y Diseño obteniéndose los artefactos que se generan en cada uno de estos flujos.

Se realizó el prototipo no funcional de la versión 2.0 del Sistema Higiene y Epidemiología con sus respectivas funcionalidades mejorando su arquitectura, utilizándose como herramientas y tecnologías las definidas por el Área Temática Sistemas Especializados para el desarrollo de sus aplicaciones como son: el lenguaje de programación PHP, el servidor de aplicaciones Apache, el conjunto de librerías javascript YUI y el framework Symfony.

RECOMENDACIONES

El trabajo realizado cumplió con los objetivos propuestos, aunque algunas funcionalidades requieren ser estudiadas y añadidas al sistema. Por lo que las recomendaciones van dirigidas fundamentalmente a sugerir acciones para complementar el producto obtenido. Por lo antes planteado se recomienda:

1. Agregar funcionalidades que no fueron incluidas en la investigación, al ampliar el diseño de un grupo enfermedades transmisibles que se encuentran contenidas en el Control Sanitario Internacional, como malaria, paludismo.
2. Extender las funcionalidades del sistema en los puertos para mantener la vigilancia epidemiológica de la misma manera que se realiza en las áreas de salud y aeropuertos.

REFERENCIAS BIBLIOGRÁFICAS.

- (1)-Mena Ramírez, Libeth y Montero Eireos, Lisbeth. Tesis Diseño del Módulo de Higiene y Epidemiología.
- (2)-Sánchez Rojas, Alexis. <http://www.sld.cu/galerias/pdf/sitios/vigilancia/ramirezmilvia.pdf>. Habana: s.n.
- (3, 4)- Ídem a referencia 2.
- (5)-Sistema integrado de vigilancia para la prevención de dengue. Álvarez Valdés., Dr. Ángel M. 3, Ciudad Habana: Ciencias Médicas, 2007, Vol. 59. ISSN 0375-0760.
- (6 y 7)-Suárez Medina, Ramón. Informática 2007. [En línea] [Citado el: 8 de 10 de 2008.] <http://www.informatica2007.sld.cu/Members/ramonsm/software-para-el-sistema-integrado-de-vigilancia-de-dengue/>
- (8)-Collada de la Rosa, Ing. Ricardo. Tutor de tesis Sistema Integral de Vigilancia Epidemiológica.
- (9 y 10)-Rumbaugh, James, Jacobson, Ivar y Booch, Grady. El Lenguaje Unificado de Modelado. Manual de Referencia. s.l.: Addison Wesley, 1999.
- (11)-Krucchten, Philippe. The Rational Unified Process An Introduction. s.l.: Addison Wesley, 2001.
- (12)-Symfony. [En línea] <http://www.symfony-project.org/>
- (13)-YUI: Yahoo User Interface. [En línea] <http://www.n4gash.com/2008/yahoo-user-interface-library-yui-ejemplos-codigos-javascript/>
- (14)-CABALLERO, ISMAEL y VISCAINO, AURORA. Una Herramienta CASE para ADOO: Visual Paradigm. . Ciudad Real: s.n., 2003.
- (15)-Paradigm, Visual. Visual Paradigm UML for Enterprise Edition. [En línea] [Citado el: 05 de 12 de 2008.] <http://www.visual-paradigm.com/product/vpum/>.
- (16)-Introducción a Apache. [En línea] [Citado el: 20 de 12 de 2008] http://linux.ciberaula.com/articulo/linux_apache_intro/
- (17)-Patrones de diseño. [En línea] [Citado el: 8 de 12 de 2008.] <http://www.ingenierossoftware.com/analysisydiseno/patrones-diseno.php>.
- (18)-Modelo-Vista-Controlador. [En línea] [Citado el: 25 de 11 de 2008.] <http://www.proactiva-calidad.com/java/patrones/mvc.html>.

Referencias Bibliográficas

- (19)-Arquitectura en 3 capas. [En línea] [Citado el: 12 de 1 de 2009]
<http://www.slideshare.net/Decimo/arquitectura-3-capas>
- (20)-Arquitectura Orientada a Servicio. [En línea] [Citado el: 05 de 12 de 2008.]
[http://www.logiclibrary.com/.](http://www.logiclibrary.com/)
- (21)-Bellido, Alexis. Maestros del Web. Estándares web. [En línea] [Citado el: 3 de 27 de 2008.]
[http://www.maestrosdelweb.com/editorial/estandaresweb/.](http://www.maestrosdelweb.com/editorial/estandaresweb/)
- (22)-Arquitectura Basada en Componente. [En línea] [Citado el: 05 de 12 de 2008.]
[http://msguayaquil.com/blogs/julioc/archive/2006/05/08/Desarrollo-de-Software-Basado-en-Componentes.aspx.](http://msguayaquil.com/blogs/julioc/archive/2006/05/08/Desarrollo-de-Software-Basado-en-Componentes.aspx)
- (23)-Isla, J y Gutiérrez, F. "Modelado Estructural de Patrones de Diseño: Diagramas REP".s.l: SugarLoafPLoP, 2003.
- (24)-Larman, Craig. UML y Patrones. Introducción al análisis y diseño orientado a objetos. México: Prentice Hall, 1999. ISBN: 970-17-0261-1.
- (25)-Saavedra Gutiérrez, Jorge. El Mundo Informático. Patrones de Software para la asignación General de Responsabilidad. [En línea] <http://jorgesaavedra.wordpress.com/category/patrones-grasp/>

BIBLIOGRAFÍA

- ❖ **Alea Caridad Salazar** La Informática y su impacto social. [Conference]. - Cuba : [s.n.], 2000.
- ❖ **Álvarez Valdés. Dr. Ángel M** Sistema integrado de vigilancia para la prevención de dengue [Journal]. - Ciudad Habana : Ciencias Médicas, 2007. - 3 : Vol. 59. - ISSN 0375-0760.
- ❖ **Arquitectura Basada en Componentes** [En línea]. - 05 de 02 de 2008. - <http://msguayaquil.com/blogs/julioc/archive/2006/05/08/Desarrollo-de-Software-Basado-en-Componentes.aspx>.
- ❖ **Arquitectura en 3 capas** [En línea] // II JORNADA DE CIENCIA Y TECNOLOGIA CESCA 2007. - 22 de 1 de 2008. - <http://www.cescaarequipa.edu.pe/jornada%20II.html>.
- ❖ **Arquitectura Orientada a Servicio** [En línea]. - 05 de 02 de 2008. - <http://www.logiclibrary.com/>.
- ❖ **Batista R y Feal P** La Epidemiología en la Atención Primaria de Salud. [Publicación periódica]. - La Habana : [s.n.], 2001.
- ❖ **Bellido Alexis** Maestros del Web [En línea] // Estándares web. - 3 de 27 de 2008. - <http://www.maestrosdelweb.com/editorial/estandaresweb/>.
- ❖ **Benenson Abram** El Control de las Enfermedades Transmisibles en el Hombre. Informe Oficial de la Asociación Estadounidense de Salud Pública. [Informe]. - Washington DC: : Organización Panamericana de la Salud; , 2004.
- ❖ **Boggs Wendy y Boggs Michael** Mastering UML with Rational Rose 2002 [Libro]. - [s.l.] : SYBEX Inc, 1999. - ISBN: 0-7821-4017-3.
- ❖ **CABALLERO ISMAEL y VISCAINO AURORA** Una Herramienta CASE para ADOO: Visual Paradigm. [Informe]. - Ciudad Real : [s.n.], 2003.
- ❖ **Dra. Rebeca M [y otros]** Revista Cubana Higiene y Epidemiología [Publicación periódica]. - Ciudad de la Habana : Ciencias Médicas, 2000. - 3 : Vol. 38. - ISSN 1561-3003.
- ❖ **Enterprise Architect** [En línea]. - 14 de 12 de 2007. - <http://www.sparxsystems.com.ar/products/ea.html>.

- ❖ **Epidemiología Dirección General de** Manual para la Vigilancia Epidemiológica internacional [Informe]. - México : [s.n.], 2001.
- ❖ **ERIKA CAMACHO FABIO CARDESO, GABRIEL NUÑEZ** ARQUITECTURAS DE SOFTWARE [En línea]. - 1 de 2 de 2008. - <http://prof.usb.ve/lmendoza/Documentos/PS-6116/>.
- ❖ **Extreme Programming** [En línea]. - 11 de 12 de 2007. - <http://www.extremeprogramming.org/>.
- ❖ **Extreme Programming** [En línea]. - 05 de 02 de 2008. - <http://elezeta.com.ar/2004/08/27/extreme-programming-xp/>.
- ❖ **Fariñas AT Reinoso** Técnicas rápidas cualitativas para la vigilancia. - Escuela Nacional de Salud Pública: La Habana : VIGIWEB, 2005.. - ISBN 959-7158-24-8.
- ❖ **Fernández Adán O** Ciencias de la salud pública. - La Habana : Ecimed, 2005.
- ❖ **Garzón Darwin Jiménez** RUP [En línea]. - 05 de 02 de 2008. - <http://codeticainge.googlepages.com/guiaing.pdf..>
- ❖ **Gil Sandra Victoria Hurtado** Representación de la arquitectura de software usando UML [Libro]. - 2000.
- ❖ **Gutierrez Jorge A. Saavedra** Patrones GRASP [En línea]. - 05 de 02 de 2008. - <http://jorgesaavedra.wordpress.com/2006/08/17/patrones-grasp-craig-larman/>.
- ❖ **Humphrey Watts** Introducción al Proceso Software Personal [Libro]. - Madrid : Pearson Education, S.A, 2001. - ISBN: 84-7829-052-4, 328.
- ❖ **Isla J y Gutierrez F** "Modelado Estructural de Patrones de Diseño: Diagramas REP" [Informe]. - [s.l.] : SugarLoafPLoP, 2003.
- ❖ **Jacobson Ivar, Booch Grady y Runbaugh James** El proceso unificado de desarrollo de software [Libro]. - Madrid : Pearson Education, S.A, 2000. - ISBN: 84-7829-036-2.
- ❖ **Joseph Schmuller** APRENDIENDO UML EN 24 HORAS [Libro]. - México : PEARSON EDUCACION, 2000. - ISBN: 968-444-463-X.
- ❖ **Kruchten Philippe** The Rational Unified Process An Introduction [Libro]. - [s.l.] : Addison Wesley, 2001.

- ❖ **Larman Craig** UML y Patrones. Introducción al análisis y diseño orientado a objetos [Libro]. - México : Prentice Hall, 1999. - ISBN: 970-17-0261-1.
- ❖ **López Moreno S, Garrido Latorre F y Hernández Avila M** Desarrollo histórico de la Epidemiología: su formación como disciplina científica. [Publicación periódica]. - México : [s.n.], 2000. - 2 : Vol. 42.
- ❖ **Lovelle Juan Manuel Cuevas** Introducción a UML [Libro]. - 1999.
- ❖ **Manuel Antonio Rodríguez Fernández José María González Pinto, Augusto Renato Pérez Mayo, Maricela Guzmán Cáceres,** Episteme [En línea]. - 15 de 12 de 07. - <http://www.uvmnet.edu/investigacion/episteme>.
- ❖ **Martínez S** Epidemiología y Sociedad [Publicación periódica]. - Ciudad Habana : Revista Cubana de Higiene y Epidemiología, 2003. - 2 : Vol. 41. - Disponible en URL: <http://bvs.sld.cu/revistas/>.
- ❖ **Microsoft Corporation** Arquitectura orientada a servicios [En línea] // Principios de diseño de servicios: patrones y antipatrones de servicios. - 29 de 1 de 2008. - <http://www.microsoft.com/spanish/msdn/articulos/archivo/121205/voices/SOADesign.mspix>.
- ❖ **Microsoft Corporation** La Arquitectura Orientada a Servicios (SOA) de Microsoft [En línea]. - 1 de 2 de 2008. - <http://www.zonatorrida.biz/>.
- ❖ **Microsoft Solution Framework** [En línea]. - 14 de 12 de 2007. - http://www.mentores.net/articulos/intro_microsoft_sol_frame.htm.
- ❖ **Ministerio de la Informática y las Comunicaciones de Cuba (MIC)** [En línea]. - 28 de 11 de 2007. - <http://www.mic.gov.cu/hinfosoc.aspx>.
- ❖ **Ministerio de Relaciones Exteriores de la República de Cuba** [En línea]. - 10 de 11 de 2007. - http://www.cubaminrex.cu/Sociedad_Informacion/Cuba_SI/Informatizacion.htm.
- ❖ **Modelo-Vista-Controlador** [En línea]. - 25 de 1 de 2008. - <http://www.proactiva-calidad.com/java/patrones/mvc.html>.
- ❖ **Molpeceres Alberto** Procesos de desarrollo, RUP,XP y FDD [En línea]. - 15 de 12 de 2002. - 05 de 02 de 2008. -

<http://www.willydev.net/InsiteCreation/v1.0/Descargas/articulos/general/cualxpfddrup.pdf>

- ❖ **Organización Mundial de la Salud** Reglamento Sanitario internacional [Informe]. - Ginebra: Suiza : [s.n.], 1995.
- ❖ **Paradigm Visual** Visual Paradigm UML for Enterprise Edition [En línea]. - 05 de 06 de 2008. - <http://www.visual-paradigm.com/product/vpuml/>.
- ❖ **Patrones de diseño** [En línea]. - 1 de 25 de 2008. - <http://www.ingenierossoftware.com/analisisydiseno/patrones-diseno.php>.
- ❖ **Portal Biblioteca UCI** [En línea]. - 10 de 12 de 2007. - <http://biblioteca.uci.cu/bives/titdigitales.htm#igs>.
- ❖ **PRESSMAN R. S.** Ingeniería de Software. Un enfoque Práctico. Quinta [Libro]. - La Habana : Editorial Félix Varela, 2005.
- ❖ **Pressman Roger** Ingeniería de Software: Un enfoque práctico. [Libro].
- ❖ **Programa de Control Sanitario Internacional** [En línea]. - 15 de 12 de 2007. - <http://aps.sld.cu/e/proxy.html>.
- ❖ **Rational Rose** [En línea]. - 14 de 12 de 2007. - <http://www.rational.com.ar/herramientas/roseenterprise.html> .
- ❖ **Rational Rose** Enterprise Edition [En línea]. - 05 de 02 de 2008. - http://www.ciao.es/Rational_Rose_Enterprise_Edition__Opinion_612900.
- ❖ **Rational Unified Process(RUP).** [En línea]. - 2006. - <https://pid.dsic.upv.es/C1/Material/Documentos%20Disponibles/Introducción%20a%20RUP.doc>
- ❖ **Revista Cubana de Higiene y Epidemiología** [Publicación periódica]. - Ciudad de la Habana : Ciencias Médicas, 2008. - 1 : Vol. 46. - ISSN 1561-3003 .
- ❖ **Reynoso Carlos Billy** Introducción a la Arquitectura de Software [Libro]. - Buenos Aires : [s.n.], 2004.

- ❖ **Rumbaugh James, Jacobson Ivar y Booch Grady** El Lenguaje Unificado de Modelado. Manual de Referencia [Libro]. - [s.l.] : Addison Wesley, 1999.
- ❖ **Saavedra Gutiérrez Jorge** El Mundo Informático [En línea] // Patrones de Software para la asignación General de Responsabilidad. -
<http://jorgesaavedra.wordpress.com/category/patrones-grasp/>.
- ❖ **Salazar Alea Caridad Cortina Rodríguez Antonio** Diseñando Aplicaciones Distribuidas [Informe]. - Cuba : [s.n.], 2007.
- ❖ **Sánchez Rios Sergio** Metodologías de Análisis y Diseño [En línea] // Diseño O.O – **Partones de diseño**. - 4 de 26 de 2008. -
http://www.uvmsf.cl/~ssanchez/images/Metodologias/Unidad8_MAD.pdf.
- ❖ **Sánchez Rojas Alexis** <http://www.sld.cu/galerias/pdf/sitios/vigilancia/ramirezmilvia.pdf> [Informe]. - Habana : [s.n.].
- ❖ **Sánchez Tarragó Lic. Nancy** Unidad de Análisis y Tendencias en Salud. [En línea] // Área de Higiene y Epidemiología. Ministerio de Salud Pública.. - 28 de 5 de 2008.
- ❖ **Schmuller Joseph** APRENDIENDO UML EN 24 HORAS [Libro]. - México : PEARSON EDUCACION, 2000. - ISBN: 968-444-463-X.
- ❖ **Suárez Medina Ramón** Informática 2007 [En línea]. - 5 de 12 de 2007. -
[http://www.informatica2007.sld.cu/Members/ramonsm/software-para-el-sistema-integrado-de-vigilancia-de-dengue/..](http://www.informatica2007.sld.cu/Members/ramonsm/software-para-el-sistema-integrado-de-vigilancia-de-dengue/)
- ❖ **Umbrello** [En línea] // Manual de Umbrello UML Modeller. - 14 de 12 de 2007. -
[http://docs.kde.org/stable/es/kdesdk/umbrello/..](http://docs.kde.org/stable/es/kdesdk/umbrello/)
- ❖ **Umbrello UML Modeller** [En línea]. - 2006. - <http://uml.sourceforge.net/index.php>.
- ❖ **Vázquez S [y otros]** Detección de IgM contra el virus del dengue en sangre entera absorbida en papel de filtro. [Publicación periódica]. - Ciudad Habana : Ciencias Médicas, 1998. - 174–178 : Vol. 3.

- ❖ **Visual Paradigm** [En línea]. - 27 de 1 de 2008. - [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/) .
- ❖ **Visual Paradigm** [En línea]. - 05 de 04 de 2008. - <http://www.visual-paradigm.com/news/vpsuite33/vpuml63.jsp>.
- ❖ **Web oficial de RUP** [En línea]. - 29 de 11 de 2007. - <http://www.rational.com/products/rup/>.
- ❖ **Web oficial de UML** [En línea]. - 10 de 1 de 2008. - <http://www.omg.org/uml/>.
- ❖ **Xavier Ferré Grau María Isabel Sánchez Segura** Desarrollo Orientado a Objetos con UML [Libro].

ANEXO 1

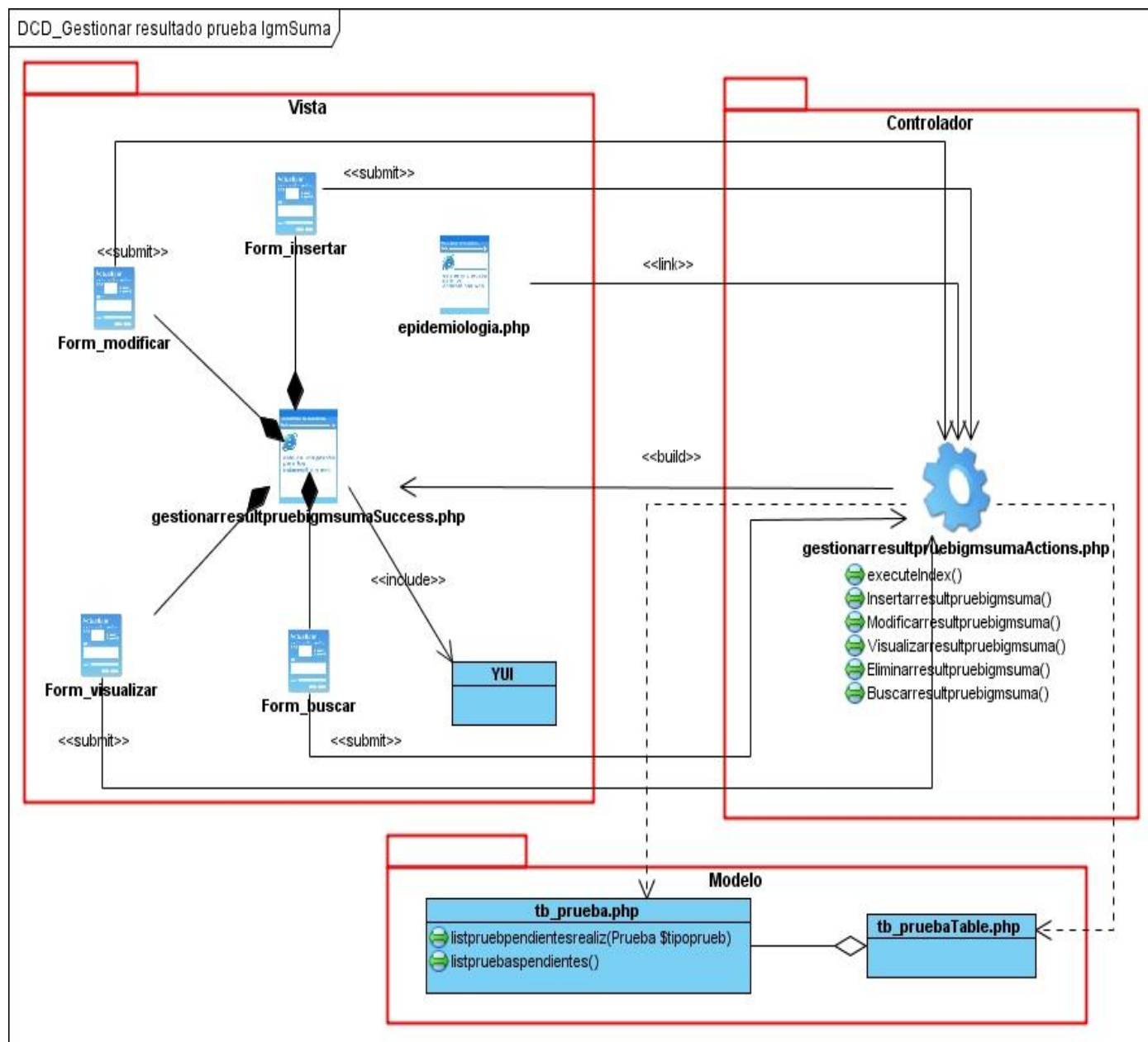


Fig. 3.4 Diagrama de Clases del Diseño Gestionar resultado de la prueba de IgmSuma.

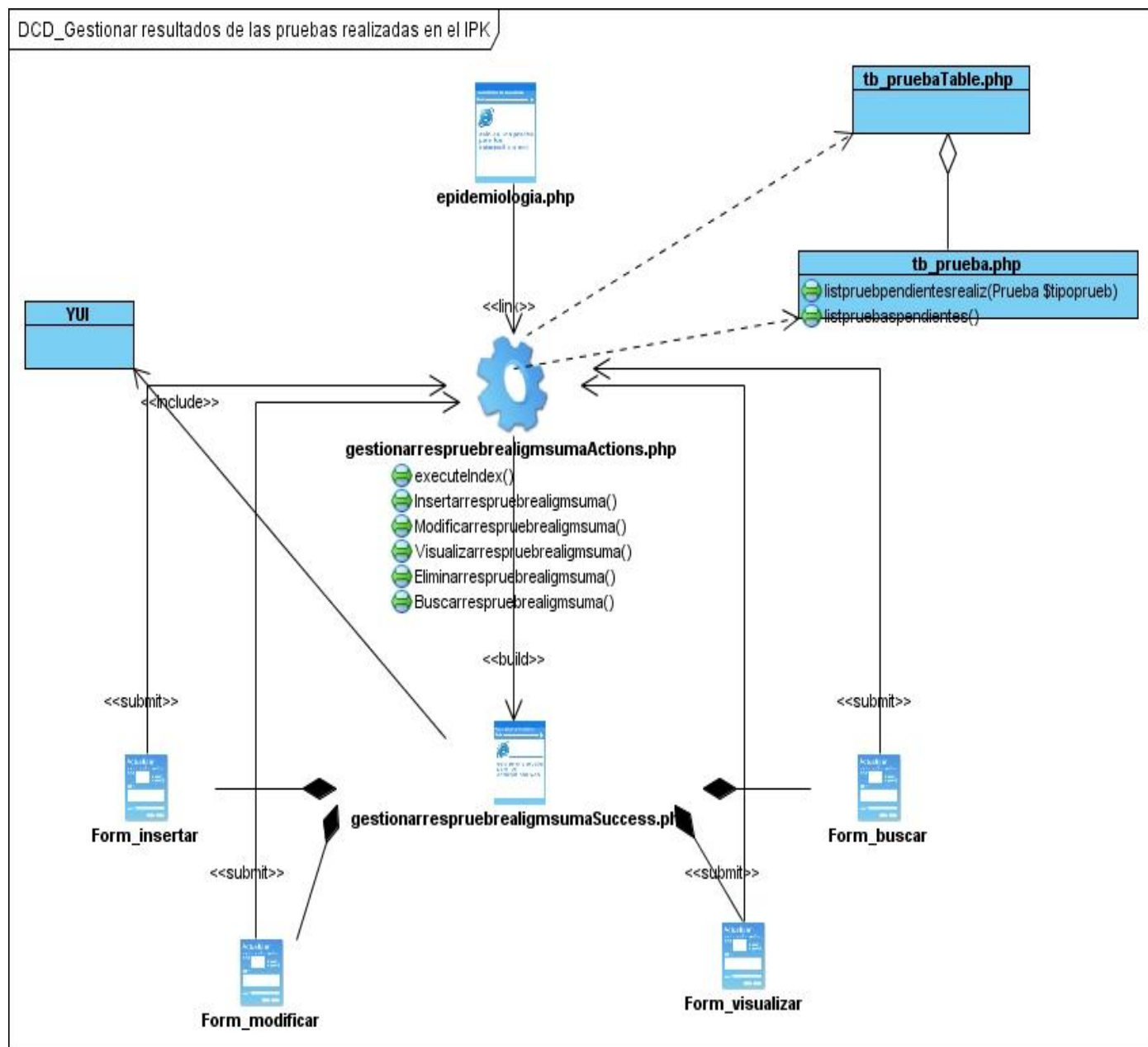


Fig. 3.5 Diagrama de Clases del Diseño Gestionar resultados de las pruebas realizadas en el IPK.

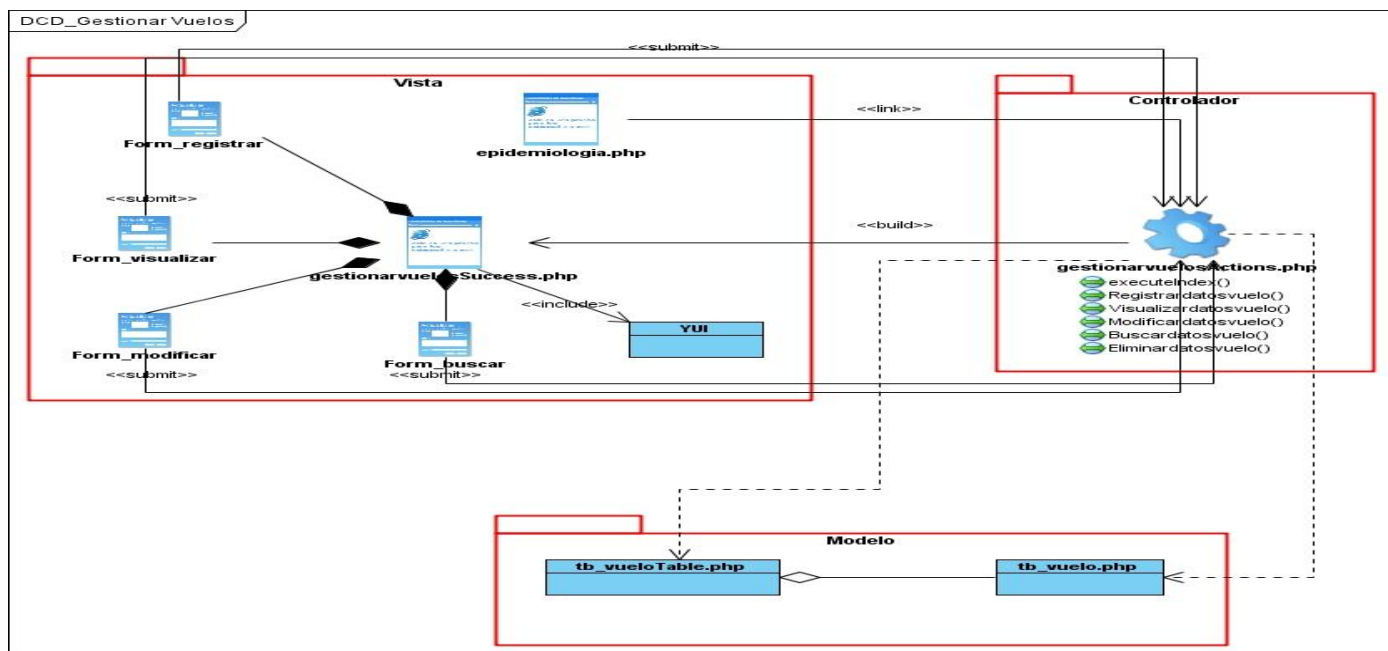


Fig. 3.6 Diagrama de Clases del Diseño Gestionar Vuelos.

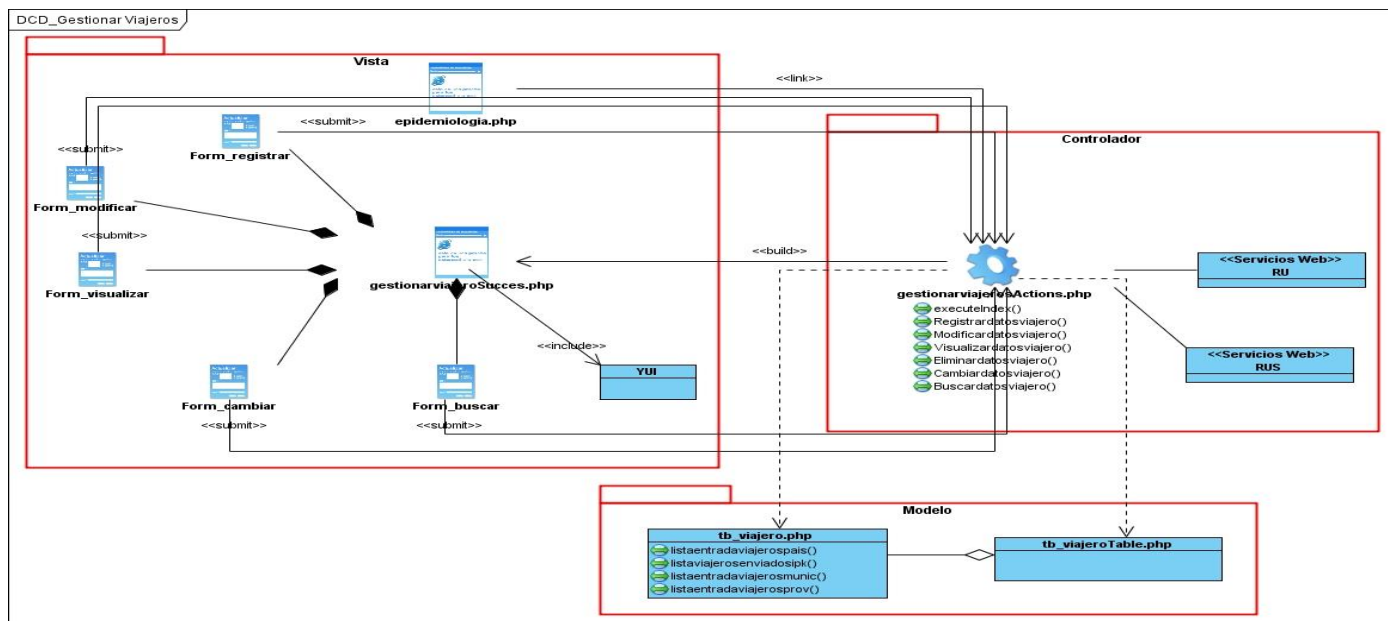


Fig. 3.7 Diagrama de Clases del Diseño Gestionar Viajeros.

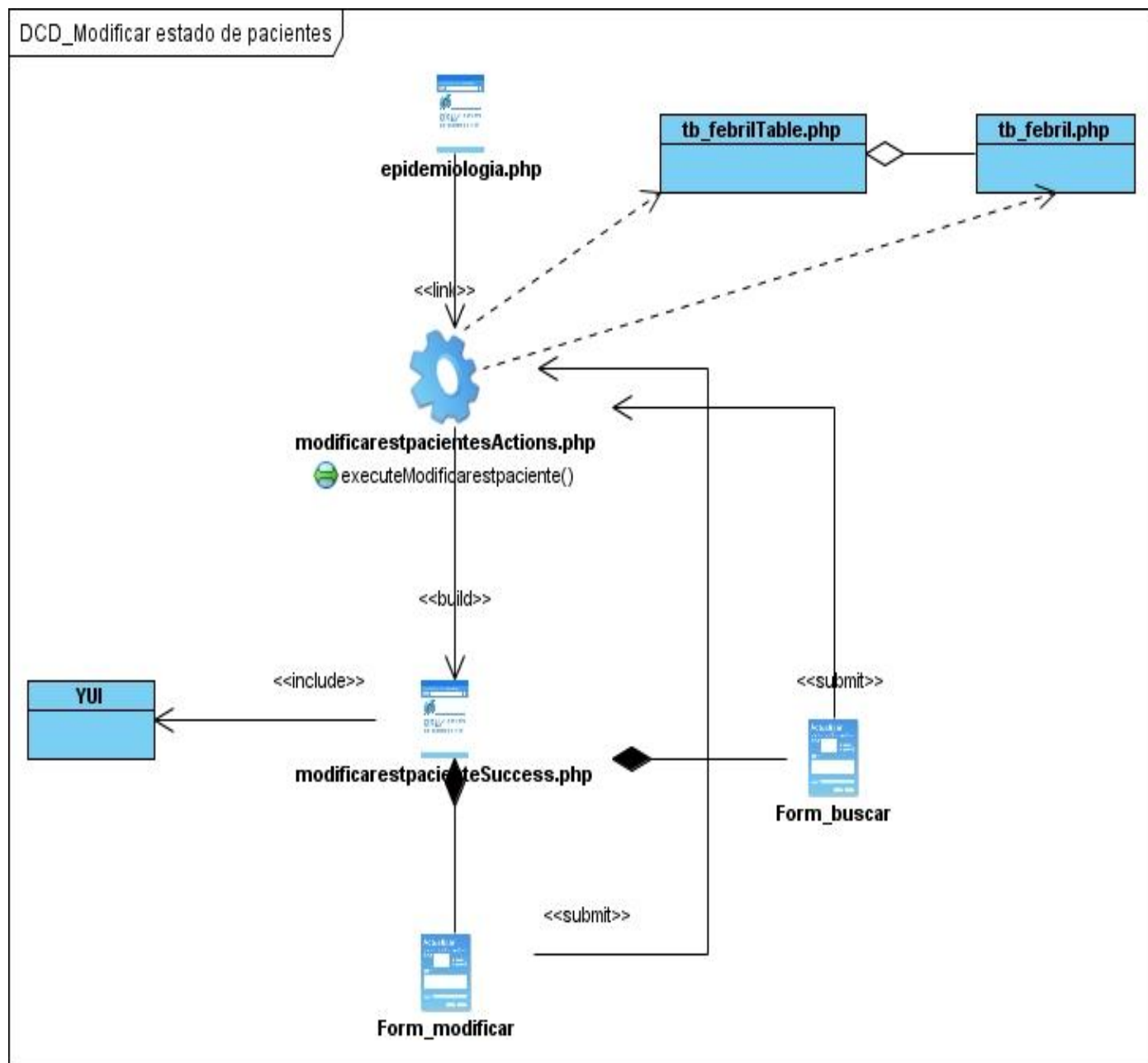


Fig. 3.8 Diagrama de Clases del Diseño Modificar estado de Pacientes.

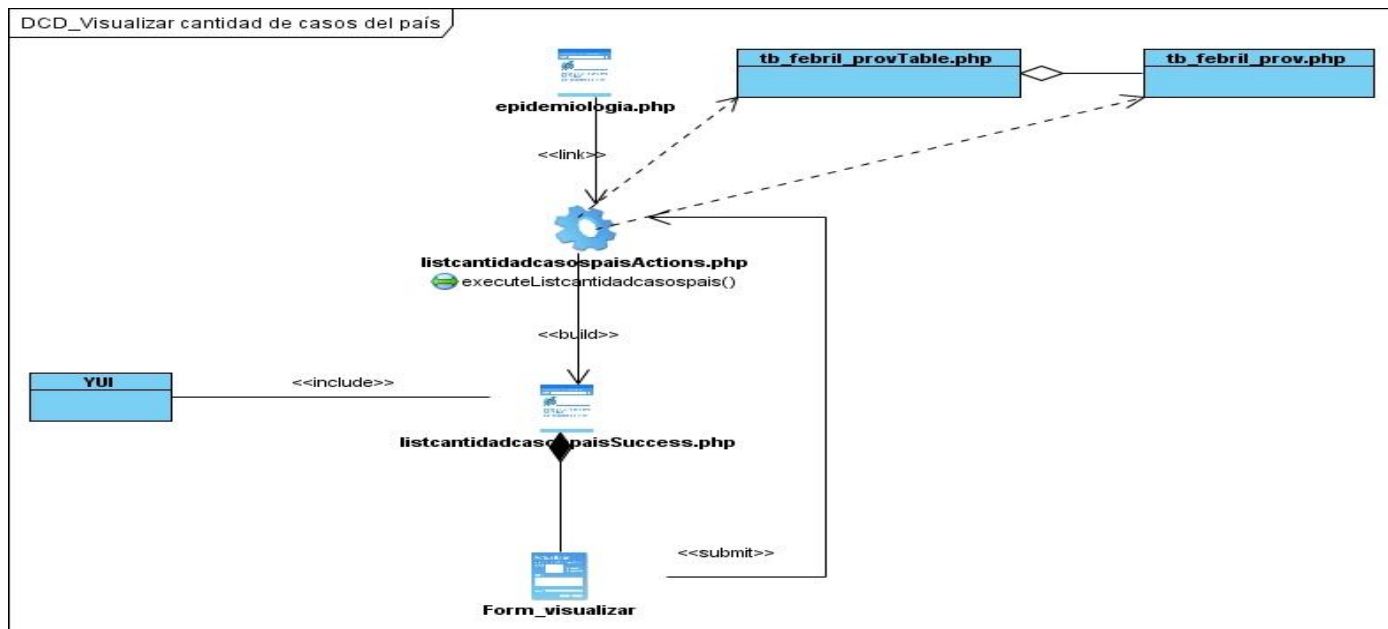


Fig. 3.9 Diagrama de Clases del Diseño Visualizar cantidad de casos del país.

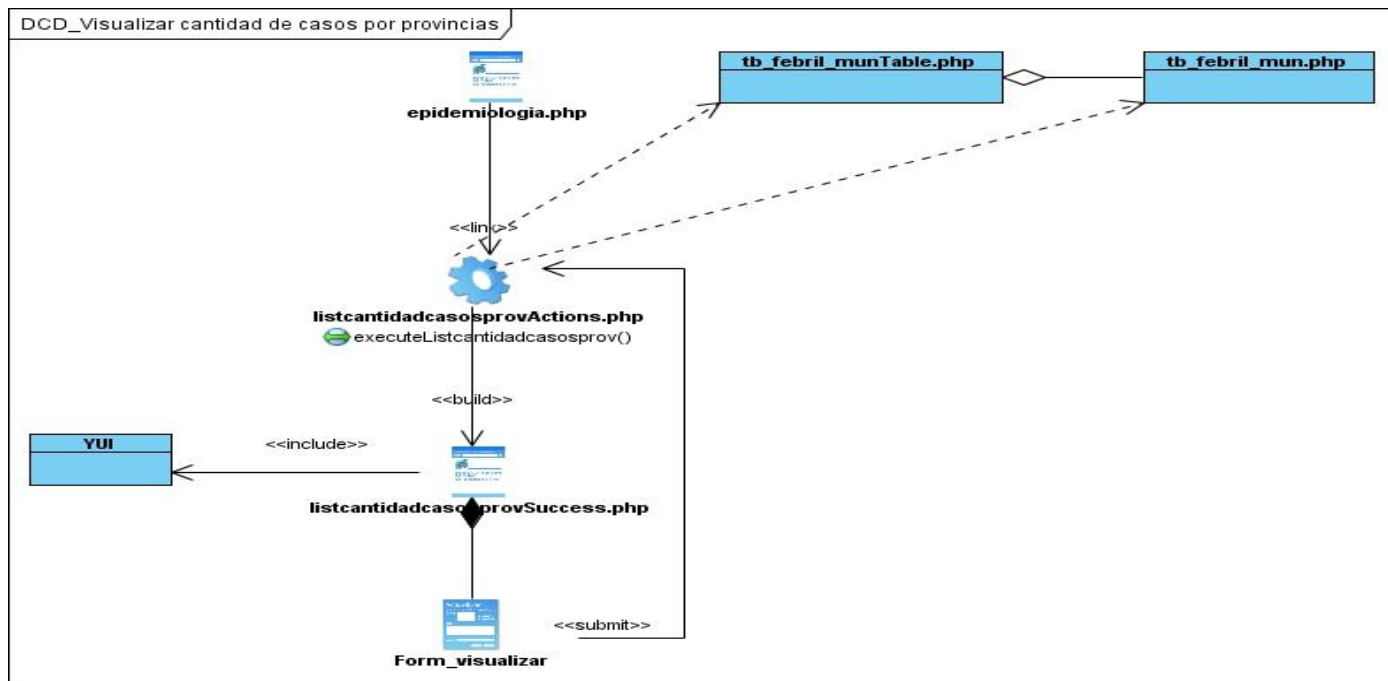


Fig. 3.10 Diagrama de Clases del Diseño Visualizar cantidad de casos por provincia.

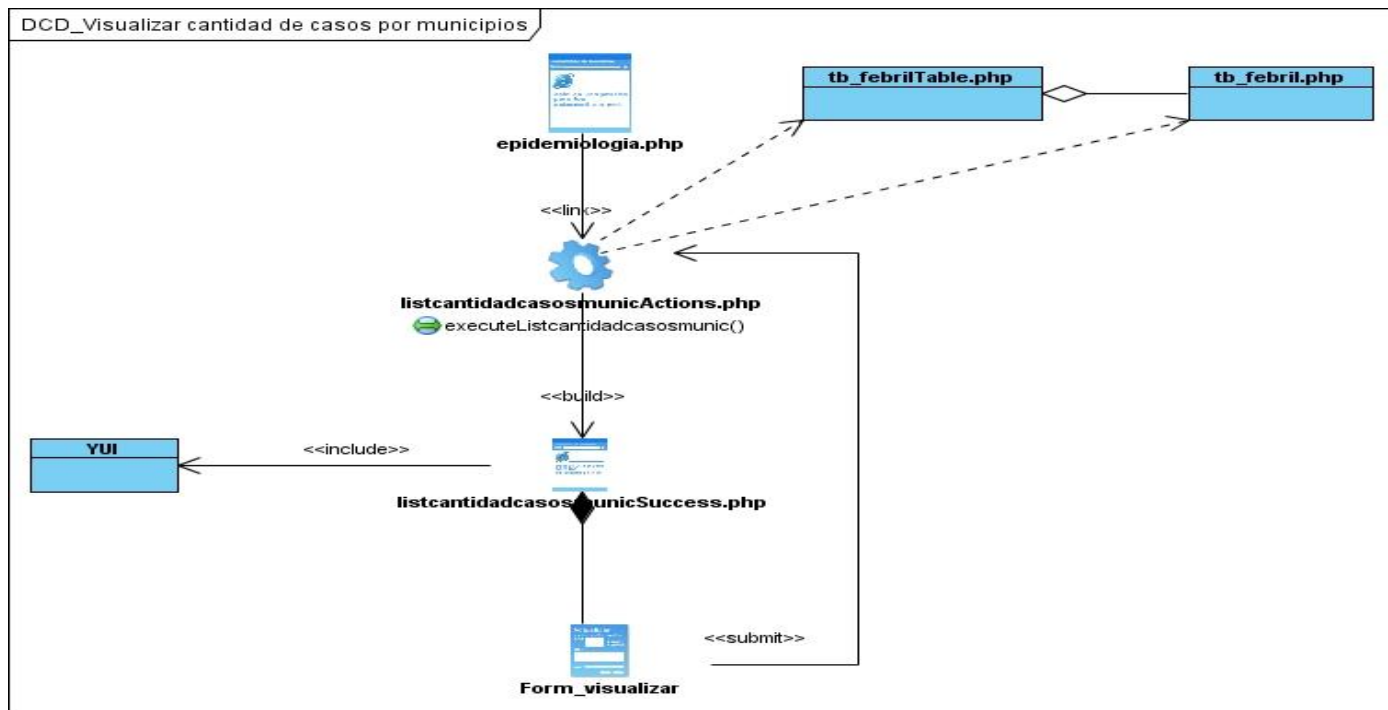


Fig. 3.11 Diagrama de Clases del Diseño Visualizar cantidad de casos por municipios.

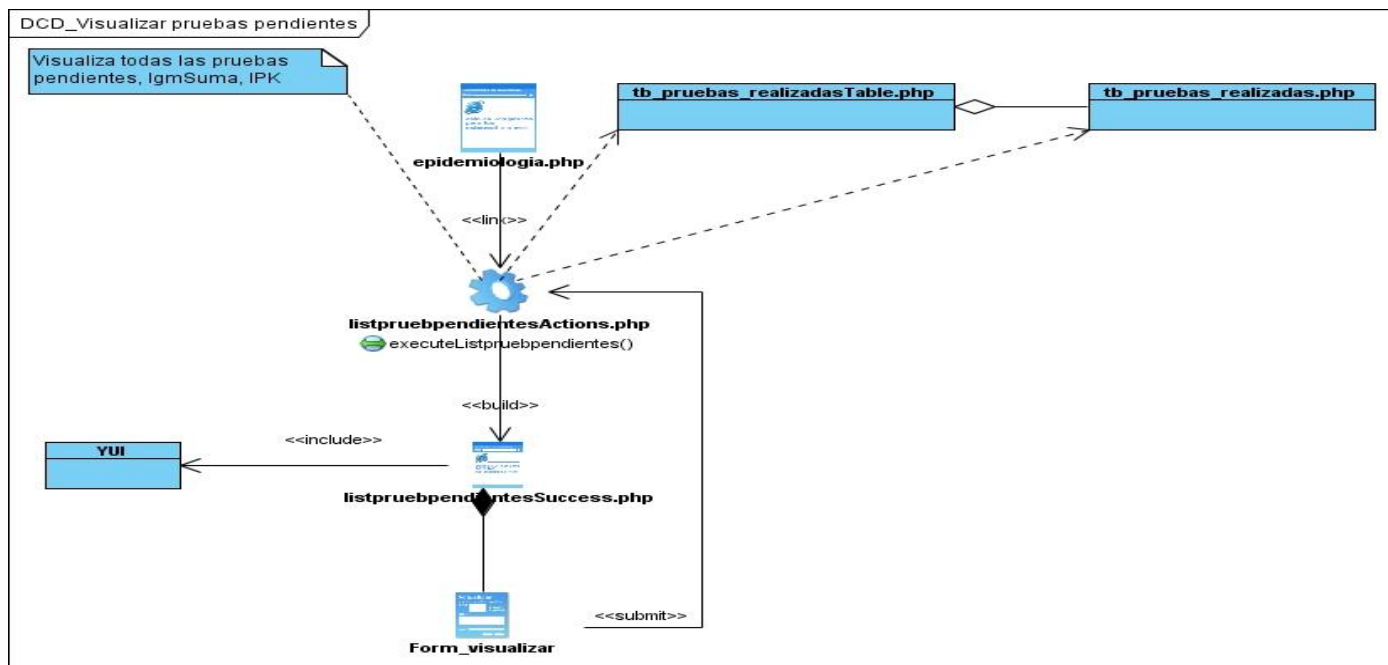


Fig. 3.8 Diagrama de Clases del Diseño Visualizar pruebas pendientes.

ANEXO 2

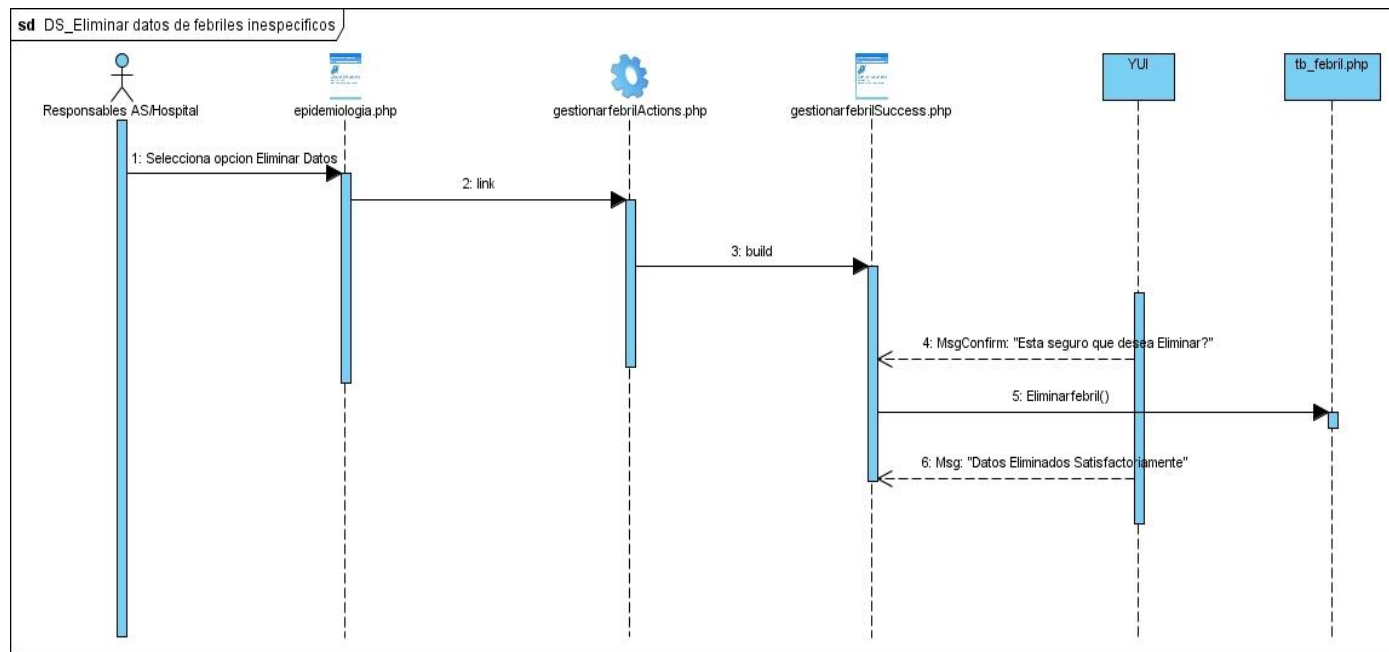


Fig. 3.10 Diagrama de Secuencia Eliminar datos de febriles inespecíficos.

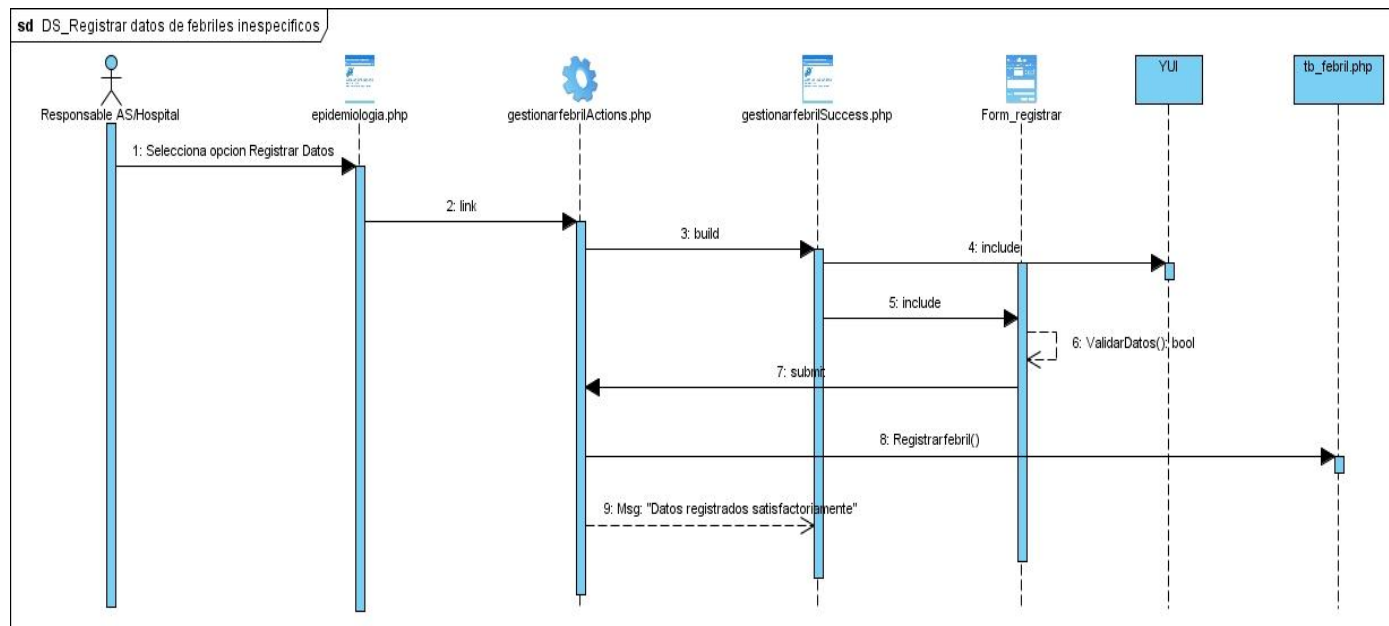


Fig. 3.11 Diagrama de Secuencia Registrar datos de febriles inespecíficos.

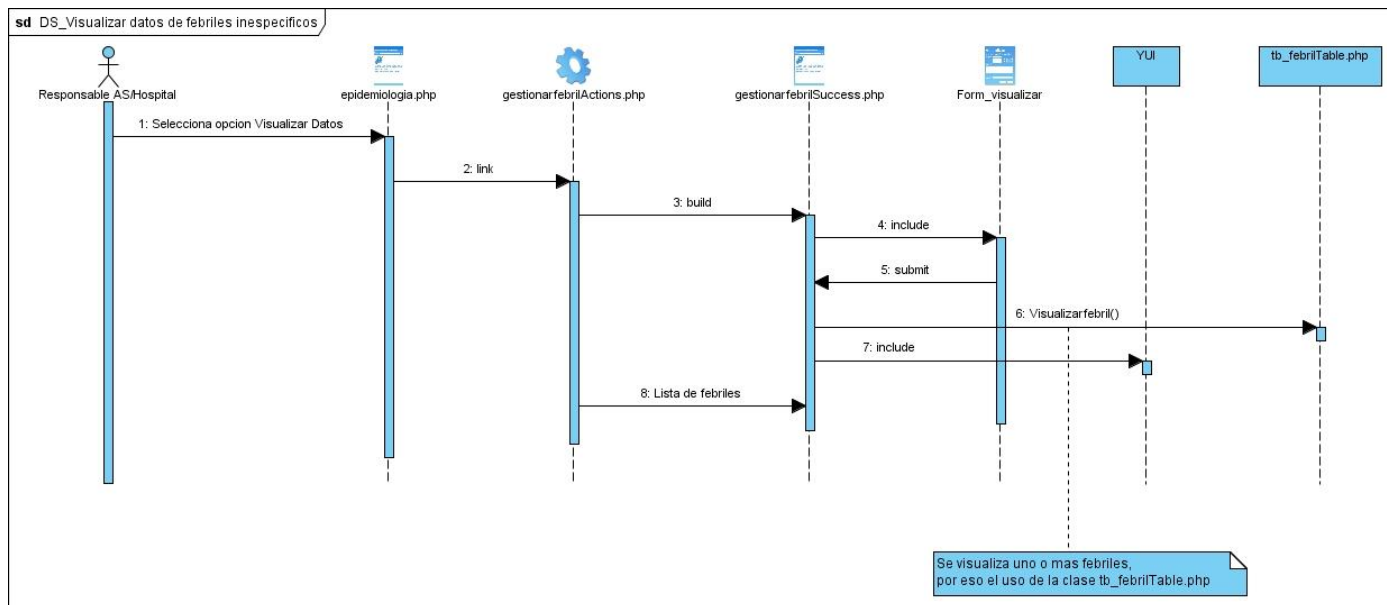


Fig. 3.12 Diagrama de Secuencia Visualizar datos de febriles inespecificos.

GLOSARIO DE TÉRMINOS.

ADO: un Data Access Object (DAO, Objeto de Acceso a Datos) es un componente de software que suministra una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos, tales como una Base de datos o un archivo. El término se aplica frecuentemente al Patrón de diseño Object.

CASE: Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

Caso de Uso: Especificación de las secuencias de acciones, incluyendo variaciones que un sistema lleva a cabo y que conduce a un resultado observable de interés para un actor determinado.

Dengue: Enfermedad febril aguda, transmisibles en los trópicos y en África. También conocida como *fiebre rompe-huesos*, caracterizada por fiebre y dolor intenso en las articulaciones y músculos, inflamación de los ganglios linfáticos y erupción ocasional de la piel. El dengue se transmite a los humanos por el mosquito hembra *Stegomyia aegypti* (antes denominado *Aedes aegypti*).

ELISA: (Ensayo inmunoabsorbente ligado a enzimas): es una técnica basada en la detección de un antígeno inmovilizado sobre una fase sólida mediante anticuerpos que directa o indirectamente producen una reacción. Este principio tiene muchas de las propiedades de un inmunoensayo ideal: es versátil, robusto, simple en su realización, empleas reactivas económicas y consigues, mediante el uso de la fase sólida, de una separación fácil entre la fracción retenida y la fracción libre.

Epidemiología: Es la disciplina científica que estudia la distribución, frecuencia, determinantes, relaciones, predicciones y control de los factores relacionados con la salud y enfermedad. La epidemiología en sentido estricto, que podría denominarse humana, ocupa un lugar especial en la intersección entre las ciencias biomédicas y las ciencias sociales y aplica los métodos y principios de estas ciencias al estudio de la salud y la enfermedad en poblaciones humanas determinadas.

Herramienta Case: Ingeniería de sistemas asistida por ordenador (Computer-Aided System Engineering - CASE) es la aplicación de tecnología informática a las actividades, las técnicas y las metodologías propias de desarrollo de sistemas. Su objetivo es automatizar o apoyar una o más fases del ciclo de vida del desarrollo de sistemas.

IDE: (Integrated Development Environment) Un entorno de desarrollo integrado es un programa compuesto por un conjunto de herramientas para un programador.

INFOMED: Infomed es la Red Telemática de Salud en Cuba. Dedicada a proporcionar información médica actualizada tanto nacional como internacional utilizando las nuevas tecnologías de la información y las comunicaciones (NTIC).

IGG: (Inmunoglobulina G): es una de las cinco clases de anticuerpos humorales producidos por el organismo. Se trata de la inmunoglobulina predominante en los fluidos internos del cuerpo, como son la sangre, el líquido cefalorraquídeo y el líquido peritoneal (líquido presente en la cavidad abdominal). Esta proteína especializada es sintetizada por el organismo en respuesta a la invasión de bacterias, hongos y virus.

Interfaz de usuario: Interfaz a través de la cual un usuario interactúa con un sistema.

Metodologías: Se encargan de elaborar estrategias de desarrollo de software que promuevan prácticas adaptativas en vez de predictivas; centradas en las personas o los equipos, orientadas hacia la funcionalidad y la entrega, de comunicación intensiva y que requieren implicación directa del cliente.

MIC: Ministerio de la Informática y las Comunicaciones, es el organismo rector de estas disciplinas en Cuba.

MINSAP: El Ministerio de Salud Pública es el Organismo rector del Sistema Nacional de Salud, encargado de dirigir, ejecutar y controlar la aplicación de la política del Estado y del Gobierno en cuanto a la Salud Pública, el desarrollo de las Ciencias Médicas y la Industria Médico Farmacéutica.

Multiplataforma: Poder funcionar o mantener una interoperabilidad de forma similar en diferentes sistemas operativos o plataformas.

Negocio: Cualquier ambiente o entorno en cual esta enmarcado el problema.

Patrones: Describe un problema que ocurre una y otra vez en su entorno y describe también el núcleo de la solución al problema, de forma que puede reutilizarse continuamente.

PCR: (La reacción en cadena de la polimerasa) conocida por sus siglas en inglés (Polymerase Chain Reaction): es una técnica de biología molecular cuyo objetivo es obtener un gran número de copias de un fragmento de ADN particular. Esta técnica sirve para amplificar un fragmento de ADN; su utilidad es que,

tras la amplificación, resulta mucho más fácil identificar con una muy alta probabilidad virus o bacterias causantes de una enfermedad, identificar personas (cadáveres) o hacer investigación científica sobre el ADN amplificado.

Prototipo de interfaz de usuario: Prototipo ejecutable de una interfaz de usuario pero que puede, en los elementos iniciales del desarrollo, consistir únicamente en dibujos de papel, diseño de pantallas etc.

Reglas del Negocio: Describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio.

Requisito: Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.

Requisito funcional: Requisito que especifica una acción que debe ser capaz de realizar el sistema, sin considerar restricciones físicas, requisito que especifica comportamiento de entrada/salida de un sistema.

Rol: Papel, cometido o función que tiene o desempeña que interpreta un actor.

RUP: Proceso Unificado de Software (Rational Unified Process).

SIVD: Sistema Integrado de Vigilancia de Dengue.

Software: Conjunto de instrucciones escritas en un determinado lenguaje, que dirigen a un ordenador para la ejecución de una serie de operaciones, con el objetivo de resolver un problema que se ha definido previamente.

TIC: Las Tecnologías de la Información y la Comunicación son un conjunto de servicios, redes, software, aparatos que tienen como fin el mejoramiento de la calidad de vida de las personas dentro de un entorno, y que se integran a un sistema de información interconectado y complementario.

UCI: Universidad de Ciencias Informáticas.

UML: El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación.

Vigilancia Epidemiológica: Término inicialmente aplicado a un conjunto de medidas inherentes a la observación de la evolución de enfermos infecciosos o sospechosos y sus contactos, es decir, que no era

más que la aplicación del método epidemiológico para el control individual de casos. En su desarrollo, la vigilancia se extendió y su marco de trabajo fue incluyendo todos los aspectos y mecanismos que intervienen en la aparición de enfermedades transmisibles o no, agudas o crónicas, así como desviaciones de la salud, tanto en el orden individual como de toda la población.

XHTML: (acrónimo inglés de eXtensible Hypertext Markup Language (lenguaje extensible de marcado de hipertexto), es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web. XHTML es la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones más estrictas de XML.)

XP: EXtreme Programming. Es una metodología de desarrollo de software.