

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS  
FACULTAD 1

*Módulos Historia de Salud Individual e Historia de  
Salud Familiar para la Dirección de Registros  
Médicos del Ministerio de Salud Pública.*

Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas

**Autor: Ariel Humberto Corrales Sardina**

**Tutores:**

**Ing. Ana María Leyva Agra**

**Ing. Karina Barbara Martinez Casas**

**Ing. Yulio Aleman Jimenez**

**Habana, Cuba, junio 2018**

**“Año 60 de la Revolución”**

*Módulos Historia de Salud Individual e Historia de Salud Familiar para la Dirección de Registros Médicos del Ministerio de Salud Pública.*



**Declaración de Autoría:**

Declaro ser el autor del presente trabajo de diploma y reconozco a la Xetid, Empresa de Tecnología de la Información para la Defensa, los derechos patrimoniales del mismo, con carácter exclusivo.

Para que así conste firman la presente a los \_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

**Firma del Autor**

**Ariel Humberto Corrales Sardina**

---

**Firma del Tutor**

**Ing. Ana María Leyva Agra**

---

**Firma del Tutor**

**Ing. Karina Barbara Martínez Casas**

---

**Firma del Tutor**

**Ing. Yulio Aleman Jimenez**

*Frase*

*...La tarea que los graduados de la U.C.I. tienen por delante es grandiosa*

*Espero que la cumplan y la cumplirán...*

*Fidel Castro Ruz*





**Quiero agradecer:**

**A mis padres por darme la vida y apoyo en todo momento.**

**A mi hermana por estar conmigo en las buenas y las malas.**

**A mi novia la cual quiero y me ha ayudado a continuar mis estudios.**

**A mis tutoras que me dieron su apoyo en todo momento.**

**A mis compañeros que de una forma u otra ayudaron en la culminación no solo de mi tesis sino de mi carrera en general.**



## **Resumen**

El presente Trabajo de Diploma tiene como objetivo desarrollar módulos para la gestión de información de las Historias de Salud tanto Individual como Familiar, vinculadas con cada paciente de las instalaciones médicas del país. En el desarrollo de los módulos se utilizó la metodología propuesta en el manual PRODESOFIT en su versión 1.5 y el lenguaje de modelado UML. Además, se utilizó como lenguaje de programación PHP, JavaScript y HTML. Se hizo uso de los framework Zend Framework, Doctrine y el marco de trabajo Zeolides. También se emplea el sistema gestor de base de datos PostgreSQL y el servidor de aplicaciones web Apache. Este software informático consta de dos módulos los cuales representan la información almacenada por los médicos en cada interacción con los pacientes. Los mismos muestran datos no solo de cada paciente y su salud, sino también de sus condiciones de vida y su relación con la sociedad en la que reside. Esto permite a los profesionales de la salud contar con un mayor conocimiento del entorno en el cual se desenvuelve el paciente para realizar un mejor diagnóstico.

**Palabras clave:** Historia de Salud Familiar, Historia de Salud Individual, paciente, software informático.



Índice

INTRODUCCIÓN .....	12
Capítulo 1: Fundamentación teórica sobre la gestión de información de historias de salud individual e historias de salud familiar.....	17
1.1 Introducción .....	17
1.2 Conceptos Fundamentales .....	17
1.3 Aplicaciones web que gestionan información de salud.....	18
1.4 Proceso de desarrollo de software .....	20
1.5 Lenguajes de modelado UML v8.0 y BPMN .....	21
1.6 Herramienta de modelado.....	22
1.7 Lenguajes de programación.....	22
1.8 Sistemas gestores de base de datos .....	23
1.9 Administrador de base de datos.....	24
1.10 Entorno Integrado de Desarrollo(I.D.E.) .....	24
1.11 Marco de trabajo .....	24
1.12 Servidor para aplicaciones web .....	26
1.13 Conclusiones del capítulo .....	27
Capítulo 2: Análisis y diseño de los módulos de historias de salud individual e historias de salud familiar.	28
2.1 Introducción .....	28
2.2 Propuesta de solución.....	28
2.3 Modelado del negocio .....	28
2.4 Requisitos .....	31
2.4.1 Fuentes y técnicas para la obtención de requisitos .....	32
2.4.2 Especificación de requisitos .....	32
2.4.2.1 Requisitos funcionales: .....	32
2.4.2.2 Requisitos no funcionales del sistema.....	33
2.4.3 Descripción de requisitos .....	37
2.5 Análisis y diseño .....	39
2.5.1 Diseño arquitectónico.....	39



2.5.2 Modelado de datos.....	40
2.5.2.1 Estándar de base de datos .....	43
2.5.2.2 Plan de respaldo de la base de datos .....	43
2.5.3 Modelado del diseño .....	44
2.5.3.1 Patrones de diseño .....	44
2.5.3.2 Diagramas de clases del diseño.....	47
2.6 Prototipos de interfaz de usuario.....	48
2.7 Conclusiones del capítulo .....	49
Capítulo 3: Implementación, pruebas y validación de los módulos de historias de salud individual e historias de salud familiar.....	50
3.1 Introducción .....	50
3.2 Implementación.....	50
3.2.1 Modelo de implementación .....	50
3.2.1.1 Diagrama de componentes .....	50
3.2.1.2 Diagrama de despliegue .....	51
3.2.2 Estándares de implementación .....	52
3.2.3 Interfaz gráfica de usuario.....	53
3.3 Pruebas de software .....	55
3.3.1 Métodos de pruebas .....	56
3.3.1.1 Tipos de pruebas de caja negra.....	56
3.3.1.2 Pruebas de Integración .....	60
3.3.1.3 Pruebas de Carga y Estrés .....	62
3.3.1.4 Tipos de pruebas de caja blanca.....	64
3.3.2 Evaluación del objetivo de la investigación .....	65
3.4 Conclusiones del capítulo .....	66
Conclusiones generales.....	67
Recomendaciones .....	68
Referencias Bibliográficas:.....	69
Bibliografía consultada.....	72
Glosario de términos:.....	75



Anexos.....	76
Anexo 1. Entrevista realizada al médico especialista .....	76
Anexo 2. Entrevista realizada al médico especialista .....	77
Anexo 3. Descripción de requisitos .....	78
Anexo 4 Diagrama entidad relación .....	82
Anexo 5 Diagramas de clases del diseño con estereotipos web .....	84
Anexo 6 Prototipos de interfaz de usuarios .....	86





## Índice de ilustraciones

<b>Figura 1. Modelo Conceptual del negocio de la propuesta de solución. (Elaboración propia)</b> .....	29
<b>Figura 2. Diagrama de actividades del proceso de negocio gestionar HSI. (Elaboración propia)</b> .....	30
<b>Figura 3. Diagrama de actividades del proceso de negocio gestionar HSF. (Elaboración propia)</b> ....	31
<b>Figura 4. Diagrama de Requisitos caso genérico gestionar concepto. (Elaboración propia)</b> .....	37
<b>Figura 5. Diagrama de actividades caso adicionar integrante a la familia. (Elaboración propia)</b> .....	38
<b>Figura 6. Diagrama entidad relación caso HSF. (Elaboración propia)</b> .....	41
<b>Figura 7. Diagrama entidad relación caso HSI. (Elaboración propia)</b> .....	42
<b>Figura 8. Método de la clase HsfminsapController. (Elaboración propia)</b> .....	45
<b>Figura 9. Instancia de HsfminsapModel en HsfminsapController. (Elaboración propia)</b> .....	45
<b>Figura 10. Ejemplo del patrón bajo acoplamiento en la clase HsfminsapModel. (Elaboración propia)</b> .....	46
<b>Figura 11. Uso de consultas a la Base de Datos con Doctrine. (Elaboración propia)</b> .....	46
<b>Figura 12. Diagrama de clases del diseño con estereotipos web caso listar HSF. (Elaboración propia)</b> .....	47
<b>Figura 13. PIU 1 Adicionar Familia. (Elaboración propia)</b> .....	48
<b>Figura 14. Diagrama de componentes. (Elaboración propia)</b> .....	51
<b>Figura 15. Diagrama de despliegue. (Elaboración propia)</b> .....	52
<b>Figura 16. Crear HSI. (Elaboración propia)</b> .....	53
<b>Figura 17. Buscar HSI. (Elaboración propia)</b> .....	54
<b>Figura 18. Crear y Editar HSF. (Elaboración propia)</b> .....	54
<b>Figura 19. Listar HSF. (Elaboración propia)</b> .....	55
<b>Figura 20. Gráfico piramidal (Módulos HSF e HSI). (Elaboración propia)</b> .....	64
<b>Figura 21. Gráfico de Abstracción/Inestabilidad (Módulos HSF e HSI). (Elaboración propia)</b> .....	65
<b>Figura 22. Entrevista realizada al médico especialista de la universidad. (Elaboración propia)</b> .....	76
<b>Figura 23. Modelo de Historia clínica de un ciudadano. (Tomada de google)</b> .....	77
<b>Figura 24. Diagrama de requisitos. Buscar ciudadano. (Elaboración propia)</b> .....	78
<b>Figura 25. Diagrama de actividades. Buscar ciudadano. (Elaboración propia)</b> .....	79



<b>Figura 26. Diagrama de actividades. Adicionar Integrante de la familia. (Elaboración propia)</b> .....	80
<b>Figura 27. Diagrama de actividades. Editar integrante de la familia. (Elaboración propia)</b> .....	81
<b>Figura 28. Diagrama entidad relación caso HSF. (Elaboración propia)</b> .....	82
<b>Figura 29. Diagrama entidad relación caso HSI. (Elaboración propia)</b> .....	83
<b>Figura 30. Diagrama de clases con estereotipos web del caso de adicionar un nuevo integrante a la familia. (Elaboración propia)</b> .....	84
<b>Figura 31. Diagrama de clases con estereotipos web para adicionar una HSF. (Elaboración propia)</b> .....	85
<b>Figura 32. PIU Buscar ciudadano. (Elaboración propia)</b> .....	86
<b>Figura 33. PIU Listar HSF. (Elaboración propia)</b> .....	86
<b>Figura 34. PIU Modificar y Adicionar Integrante de la familia. (Elaboración propia)</b> .....	87



## Índice de Tablas

<b>Tabla 1. Requisitos de Hardware. (Elaboración propia)</b> .....	33
<b>Tabla 2. Requisitos de Software. (Elaboración propia)</b> .....	33
<b>Tabla 3. Estrategia de pruebas. (Elaboración propia)</b> .....	55
<b>Tabla 4. Prueba realizada al requisito buscar ciudadano e HSI. (Elaboración propia)</b> .....	56
<b>Tabla 5. Prueba realizada al requisito añadir HSI. (Elaboración propia)</b> .....	57
<b>Tabla 6. Prueba realizada al requisito modificar HSI. (Elaboración propia)</b> .....	57
<b>Tabla 7. Prueba realizada al requisito adicionar HSF. (Elaboración propia)</b> .....	58
<b>Tabla 8. Prueba realizada al requisito buscar HSF. (Elaboración propia)</b> .....	59
<b>Tabla 9. Resumen sobre Pruebas unitarias. (Elaboración propia)</b> .....	60
<b>Tabla 10. Prueba de Integración módulos Estructura y Seguridad. (Elaboración propia)</b> .....	61
<b>Tabla 11. Prueba de Integración módulo Persona. (Elaboración propia)</b> .....	61
<b>Tabla 12. Prueba de Integración módulo Datos Maestros. (Elaboración propia)</b> .....	62
<b>Tabla 13. Resultados de carga y estrés para 100 usuarios. (Elaboración propia)</b> .....	63

## INTRODUCCIÓN

El uso de la informática en la medicina es una de sus aplicaciones más comunes e importantes desde hace varias décadas; la misma ha permitido al sector de la salud, no solo contar con métodos novedosos, sencillos y eficaces de gestión administrativa en consultas, hospitales y centros de investigación biomédica, sino también disponer de complejas aplicaciones informáticas que reducen la posibilidad de error en el diagnóstico de las enfermedades.

Los profesionales de la salud tienen como misión la atención y educación de la comunidad en cuestiones de la salud, superación profesional e investigación, pero además tienen el deber de brindar una información fiable, al llenar correctamente los modelos de trabajo. En el proceso se utilizan diferentes formularios tales como: Hoja de Actividades Diarias, Historia de Salud Familiar e Individual, Notificaciones de Enfermedades de Declaración Obligatoria, Certificados Médicos de Defunción, entre otros; los mismos satisfacen las necesidades de los trabajadores de la salud que están en función del procesamiento informativo para la producción estadística sanitaria que comienzan en el Policlínico hasta la Dirección Nacional de Estadísticas del Ministerio de Salud Pública (MINSAP).

Los formularios de trabajos médicos desempeñan una labor muy importante, gracias a la cual se obtienen las bases de conocimiento para estudios epidemiológicos, que permiten apoyar las etapas de la administración, así como las actividades docentes e investigativas y extra sectoriales. Además, permiten establecer líneas de conducta para la reducción de la mortalidad, facilitan los trabajos de los grupos de riesgos en prevención primaria, posibilita las comparaciones con períodos anteriores y con situaciones de otras regiones dentro o fuera del país.

La informatización de la sociedad es una de las actividades en las cuales se centra la Empresa de Tecnologías de la Información para la Defensa (Xetid), colocando sus esfuerzos en el desarrollo de sistemas basados en tecnologías libres como principio de desarrollo. Uno de los centros con los que cuenta la Xetid es el centro de Soporte a la Toma de Decisiones (STD), el cual está actualmente involucrado en el desarrollo de un sistema que gestione y formalice la información de las historias de salud familiar e historias de salud individual.



El llenado de estos modelos puede ser tedioso, esta actividad consta de grandes volúmenes de información la cual se debe procesar de forma manual para su posterior análisis, estos modelos corren riesgo de pérdida y deterioro, además de que la consulta de estos documentos conlleva una considerable suma de tiempo, esto es una de las causas que provoca que los grupos dispensariales no sean definidos correctamente. Estos elementos afectan negativamente la toma de decisiones por parte de los directivos, no existe control sobre la dispensarización de la población y además los documentos pueden presentarse de forma duplicada.

La situación planteada anteriormente permitió identificar el siguiente **problema de investigación**: ¿Cómo mejorar el proceso de registro de los modelos de Historia de Salud Individual (HSI) e Historia de Salud Familiar (HSF) para reducir la pérdida y deterioro de los mismos y contribuir al control de la dispensarización de la población?

Este problema se enmarca en el **objeto de estudio**: Sistemas informáticos web de recolección de datos médicos.

El **campo de acción** abarcado es: Sistemas informáticos web para la recolección de datos sobre la Historia de Salud Individual e Historia de Salud Familiar.

**Objetivo general**: Desarrollar los módulos para la gestión de los modelos de Historia de Salud Individual (HSI) e Historia de Salud Familiar (HSF) que contribuya reducir la pérdida y deterioro de los mismos y contribuir al control de la dispensarización de la población.

**Objetivos específicos:**

- Elaborar el marco teórico de la investigación acerca de soluciones informáticas existentes para la gestión y formalización de las historias de salud individual e historias de salud familiar.
- Diseñar los módulos de Historia de Salud Individual e Historia de Salud Familiar.
- Implementar los módulos de Historia de Salud Individual e Historia de Salud Familiar.
- Evaluar los módulos Historia de Salud Individual e Historia de Salud Familiar mediante pruebas de software.



Para resolver la problemática planteada, se formularon las siguientes **preguntas científicas**:

- ¿Qué referentes teóricos sirven de base para la investigación referente a soluciones informáticas existentes para la gestión y formalización de las Historias de Salud Individual e Historias de Salud Familiar?
- ¿Cómo estructurar el proceso de diseño de los módulos de Historia de Salud Individual e Historia de Salud Familiar para la Dirección de Registros Médicos del Ministerio de Salud Pública?
- ¿Cómo estructurar el proceso de implementación de los módulos Historia de Salud Individual e Historia de Salud Familiar para la Dirección de Registros Médicos del Ministerio de Salud Pública?
- ¿Cómo validar la funcionalidad de los módulos de Historia de Salud Individual e Historia de Salud Familiar para la Dirección de Registros Médicos del Ministerio de Salud Pública?

**Tareas de la Investigación:**

1. Realización de una revisión bibliográfica de los sistemas relacionados a la Historia de Salud Individual e Historia de Salud Familiar.
2. Realización de la captura de requerimientos de los módulos con el especialista.
3. Diseño de los módulos Historia de Salud Individual e Historia de Salud Familiar.
4. Realización de la propuesta de modelo de datos.
5. Implementación de los módulos Historia de Salud Individual e Historia de Salud Familiar.
6. Validación de los módulos Historia de Salud Individual e Historia de Salud Familiar.



En el desarrollo del trabajo de diploma se definieron los siguientes métodos de investigación científica:

**Métodos teóricos:**

**Modelación:** Este método se empleó para modelar los diferentes diagramas de ingeniería de software para una mayor comprensión del trabajo que se realiza y los objetivos que se deben cumplir, dando respuesta a los requerimientos planteados con anterioridad.

**Métodos Empíricos:**

**La entrevista:** Se utilizó para la obtención de información a través del diálogo, por lo que requiere de ciertas habilidades por parte del entrevistador. La entrevista puede o no, estar estructurada a partir de un cuestionario.

Este método no fue estructurado mediante cuestionario, se realizó una entrevista al cliente en la cual se acordaron los requisitos funcionales y no funcionales de la aplicación (véase en el anexo 1 los datos de recogidos en la entrevista).

**Análisis documental:** Se utilizó para consultar trabajos y buscar información relacionada con los sistemas de salud y los sistemas de gestión de información para la elaboración del marco teórico de la investigación.

El análisis de la documentación realizado a modelos de trabajo otorgó de forma conveniente una salida para el visual de la aplicación.



El contenido del presente trabajo, se estructuró en los siguientes tres capítulos:

**Capítulo 1. Fundamentación teórica sobre la gestión de información de historias de salud individual e historias de salud familiar:** expone los elementos teóricos de la investigación. Se realiza el estudio e investigación de soluciones informáticas existentes para la gestión y formalización de las historias de salud familiar e historias de salud individual. Se determinan la metodología de desarrollo, herramientas y tecnologías que se utilizan para desarrollar la solución propuesta.

**Capítulo 2. Análisis y diseño de los módulos de historias de salud individual e historias de salud familiar:** contendrá toda la información referente al análisis y diseño de la aplicación, contando en sí con el modelo de dominio, los diagramas de clases, requisitos funcionales y no funcionales que debe cumplir, así como propuesta y diseño de la arquitectura del sistema para lograr una mayor aceptación por parte de los usuarios.

**Capítulo 3. Implementación, pruebas y validación de los módulos de historias de salud individual e historias de salud familiar:** describirá el proceso de implementación y prueba de la solución informática para los módulos de Historia de Salud Familiar e Historia de Salud Individual. Contará con el diagrama de despliegue, así como la descripción de algunos casos de prueba.





## **Capítulo 1: Fundamentación teórica sobre la gestión de información de historias de salud individual e historias de salud familiar.**

### **1.1 Introducción**

El objetivo que se persigue en el presente capítulo es abordar distintos aspectos que se utilizan como soporte teórico del sistema diseñado. Se expone a través de una descripción de los conceptos asociados al problema. Se define además el objeto de estudio, la situación problemática y los objetivos generales y específicos de la presente investigación.

### **1.2 Conceptos Fundamentales**

#### **Dispensarización**

Observación permanente y dinámica a individuos, familia y comunidad con el objetivo de controlar riesgos y daños a la salud individual y colectiva, desarrollando las prestaciones para devolver la salud a cada uno de ellos según un enfoque de riesgo y el grupo en el que se encontrarán, y abordando a las familias y comunidades como un todo.

“Para la Dispensarización, el proceso se inicia con la clasificación de la población según grupos dispensariales, estos son:

Grupo I: Supuestamente sano.

Grupo II: Con riesgo.

Grupo III: Enfermos.

Grupo IV: Con secuelas” (Medicina General Integral, 2018)

#### **Historia de Salud Familiar**

El historial de salud familiar es una colección de datos sobre las condiciones médicas que afectan a la familia, así como de las costumbres alimenticias, actividades y entornos (NCBI Bookshelf, 2018).



## Historia de Salud Individual

La historia de salud individual constituye un documento básico de la atención médica primaria y puede ser considerada como la guía metodológica para identificar integralmente los problemas de salud de cada persona y también para plasmar los aspectos normales del funcionamiento del organismo (Historia de Salud Individual, 2018).

En el proceso de investigación se asumirán los conceptos ya definidos anteriormente.

### 1.3 Aplicaciones web que gestionan información de salud

A nivel internacional se han desarrollado aplicaciones para dispositivos web con el fin de mejorar la gestión hospitalaria, la investigación solo se enmarca en aquellas que ubican entre sus funcionalidades el almacenamiento de los datos de los pacientes y/o asistencia de los profesionales de la salud. A continuación, se describen dichos sistemas:

- **Sistema Médico en Línea:** Es una aplicación web pensada y creada para médicos de todas las especialidades que requieran llevar el control de consulta de sus pacientes, citas e informes médicos en una forma rápida, sencilla y eficiente. El Sistema Médico en Línea (SML) cuenta con las siguientes ventajas:
  - **Movilidad y Accesibilidad:** Podrá acceder a la información sin restricciones de horario y desde cualquier sitio.
  - **Fácil Implementación:** Con el SML sólo se necesita registrar y solicitar un plan que se adapte al consultorio.
  - **Seguro y Confiable:** Toda la data será respaldada diariamente en los servidores dedicados (SML, 2018).
- **MediCloud:** Es una plataforma gratuita, diseñada para el fácil almacenamiento en la nube de información médica de pacientes de diferentes profesionales de la salud. Dado que la información se almacena en la nube, la misma puede ser consultada desde cualquier dispositivo con internet, en cualquier momento y en cualquier parte del mundo. Al mismo tiempo, la plataforma ofrece a los



usuarios la facilidad de administrar sus clínicas de forma digital, conectar con diferentes proveedores para automatizar procesos de servicio y pertenecer a una comunidad científica latinoamericana (MediCloud, 2018).

- **DriCloud:** Es un software de soporte médico el cual contiene:
  - **Historia Clínica:** Para todas las especialidades. Fácil e intuitivo de usar. Diseño espectacular.
  - **Seguridad:** Datos en *Microsoft Cloud*: Máxima seguridad y accesibilidad (DriCloud, 2018).

A un nivel nacional se ha desarrollado una aplicación informática para la gestión hospitalaria y la recolección de datos a través de historias clínicas. A continuación, se especifica el sistema encontrado.

**Galén:** Es un sistema desarrollado específicamente para cubrir las necesidades asistenciales y de información de un Servicio de Oncología, que no solo incorpora un gestor de información, sino también módulos para la práctica asistencial diaria, Hospital de día e investigación clínica. Además, dispone de los componentes necesarios para realizar una adecuada explotación estadística de información relativa a 13.645 pacientes diagnosticados y tratados en un único Servicio de Oncología. Las funcionalidades asistenciales de Galén constituyen una Historia Clínica Electrónica completa, y refleja fielmente la metodología de trabajo utilizada sobre el formato papel (Galén, 2018).

**XAVIA:** Solución integral en un único producto para la gestión médica de hospitales y centros de salud. La misma permite la recolección, almacenamiento, procesamiento y comunicación de información relacionada con la atención al paciente, así como información administrativa del hospital (XAVIA - ucisis, 2018).

Después de analizar y estudiar los sistemas mencionados, se pudo apreciar que las funcionalidades implementadas para la realización de estos varían en dependencia de la necesidad establecida por los médicos del centro. Estos procedimientos no se tomaron en cuenta para el desarrollo de los módulos debido a que las licencias de las aplicaciones encontradas son privativas, las cuales representan un impedimento en la obtención del código de dicha aplicación por parte de la empresa Xetid.



El desarrollo de los módulos HSF e HSI está siendo una colaboración entre los centros Softel y Xetid para una mejor comprensión y abstracción de las necesidades de la población en general. En otro aspecto las tecnologías de las aplicaciones encontradas a nivel nacional e internacional realizan un seguimiento de las atenciones secundarias (atención de hospitales), mientras que los módulos de HSF e HSI tendrán un seguimiento de la atención primaria (atención de consultorios), no obstante realizaron aportes significativos en el conocimiento de las funcionalidades que deberían seguir los módulos a implementar.

#### 1.4 Proceso de desarrollo de software

Un proceso de desarrollo de software tiene como objetivo la producción eficiente de un producto de software que satisfaga los requisitos de un especialista funcional con una planificación y una estimación de recursos predecibles.

La Empresa de Tecnologías de la Información para la Defensa (Xetid) utiliza la metodología de desarrollo propuesta en el manual Prodesoft en su versión 1.5, la cual expresa los puntos necesarios para el desarrollo del ciclo evolutivo de un software.

El ciclo de vida compuesto por 5 fases: **inicio**, **modelación**, **construcción**, **explotación experimental** y **despliegue**, que comprenden todas las actividades, desde el momento en que surge la idea de crear un nuevo producto de software, hasta aquel en que el producto deja definitivamente de ser utilizado por el último de sus usuarios (Prodesoft, 2017).

Durante la **fase de Inicio** se logra una visión preliminar de la problemática a resolver y se definen los recursos relevantes para la ejecución del proyecto. Es decir, se describen los objetivos y el alcance del proyecto, se identifican los involucrados y ejecutores (entidades involucradas), se estima de manera general las actividades a realizar durante todo el ciclo de desarrollo del proyecto (Cronograma General), se establece la estrategia a seguir para realizar la modelación del negocio, la captura de requisitos y de ser necesario se estiman los recursos materiales que deberán ser adquiridos (Prodesoft, 2017).

En la **fase de Modelación** se capturan las partes esenciales del sistema, donde se identifican los procesos de negocio fundamentales y se aceptan los requisitos funcionales, obteniéndose la línea base de



la arquitectura y una estrategia de construcción de la aplicación aprobada por los implicados en el proyecto. El hito fundamental de esta fase es la liberación de la arquitectura de sistema, datos y despliegue (Prodesoft, 2017).

En la **fase de Construcción** se aclaran los requisitos restantes y se completa el desarrollo del sistema sobre una base estable de la arquitectura. Las fases anteriores sólo dieron una arquitectura básica que es aquí refinada de manera incremental conforme se construye el producto. En esta fase todas las características, componentes, y requisitos deben ser integrados, implementados, y probados en su totalidad, obteniendo una versión liberada del producto (Prodesoft, 2017).

Durante la **fase de Explotación Experimental** se convierte la versión liberada del producto en una solución estable, donde se eliminan los errores que surgen durante las pruebas y se obtiene una certificación funcional y de seguridad del producto (Prodesoft, 2017).

En la **fase de Despliegue** se instala y configura el sistema para un ambiente de producción real, se capacita al personal que usará la aplicación y se continúa dando soporte durante la explotación del sistema, culminando de ser preciso con transferencias tecnológicas (Prodesoft, 2017).

### **1.5 Lenguajes de modelado UML v8.0 y BPMN**

El Lenguaje Unificado de Modelado (UML) 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 (UML, 2017).

BPMN es un lenguaje de notación gráfica, ampliamente aceptado para el modelado de procesos de negocio. El objetivo principal de BPMN es "proporcionar una notación que sea fácilmente comprensible por todos los usuarios de negocios, desde los analistas de negocio que crean los borradores iniciales de los procesos, hasta los desarrolladores técnicos responsables de la aplicación (BPMN vs UML, 2017).

El UML será utilizado para representar la relación entre entidades, mientras que el BPMN es explícitamente para la representación del proceso de negocio de la problemática a resolver.



## 1.6 Herramienta de modelado

Las herramientas de modelado de sistemas informáticos permiten crear un "simulacro" del sistema, a bajo costo y riesgo mínimo. A bajo costo porque es un conjunto de gráficos y textos que representan el sistema, pero no son el sistema físico real. Además, minimizan los riesgos, porque los cambios que se deban realizar, se pueden realizar más fácil y rápidamente sobre el modelo que sobre el sistema ya implementado (Herramienta de Modelado, 2017).

### Visual Paradigm v5.0

*Visual Paradigm* es una herramienta CASE: Ingeniería de Software Asistida por Computación. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación (Visual Paradigm, 2017).

Esta herramienta de modelado fue utilizada para la confección de los diagramas de actividades, los diagramas de proceso de negocio, los diagramas conceptuales, los diagramas de modelado de base de datos, los diagramas de requerimientos y los prototipos de interfaz de usuarios.

## 1.7 Lenguajes de programación

### PHP v5.4

"*Hypertext Preprocessor*" es un lenguaje interpretado de alto nivel embebido en páginas *HTML* y ejecutado en el servidor. El cliente solamente recibe el resultado de su ejecución en el servidor, sin ninguna posibilidad de determinar que código ha producido el resultado recibido. El servidor web puede ser incluso configurado para que procese todos los ficheros con *PHP*. (...) *PHP* también soporta el uso de otros servicios que usen protocolos como *IMAP*, *SNMP*, *NNTP*, *POP3*, *HTTP* y derivados (AULBACH, y otros).



## HTML v5

*HTML5* es la última versión de *HTML*. El término representa dos conceptos diferentes:

- Se trata de una nueva versión de *HTML*, con nuevos elementos, atributos y comportamientos.
- Contiene un conjunto más amplio de tecnologías que permite a los sitios Web y a las aplicaciones ser más diversas y de gran alcance (HTML 5, 2017).

## JavaScript v1.8

*JavaScript* es el lenguaje interpretado orientado a objetos desarrollado por *Netscape* que se utiliza en millones de páginas *web* y aplicaciones de servidor en todo el mundo. *JavaScript* como lenguaje orientado a objetos (JavaScript, 2017).

Estos lenguajes de programación son utilizados para funcionalidades específicas, el lenguaje *HTML 5* es utilizado para la confección visual de la aplicación, se escoge este para que el sistema tenga un mejor rendimiento en el procesamiento, *JavaScript* es utilizado para la verificación de las validaciones de los requisitos funcionales y *PHP 5* es aplicado para realizar el desarrollo de la aplicación.

## 1.8 Sistemas gestores de base de datos

### PostgreSQL v9.4

*PostgreSQL* es un sistema de gestión de bases de datos de código abierto que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre apoyada por organizaciones comerciales. Dicha comunidad es denominada el *PGDG (PostgreSQL Global Development Group)* (Gestores de BD, 2018) (PostgreSQL, 2018).



## 1.9 Administrador de base de datos

### PgAdmin III v1.20.0

Herramienta de código abierto para la administración de bases de datos *PostgreSQL*. Diseñada para responder a las necesidades de los usuarios, desde escribir consultas *SQL* simples hasta desarrollar bases de datos complejas. La aplicación incluye un editor *SQL* con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar comandos programados y soporte para la replicación. La conexión al servidor puede hacerse mediante la familia de protocolos de internet (*TCP/IP* por sus siglas en inglés) y encriptarse mediante el protocolo de Capa de Conexión Segura (*SSL* por sus siglas en inglés) para mayor seguridad (PGADMIN, 2017).

## 1.10 Entorno Integrado de Desarrollo(I.D.E.)

### NetBeans v8.2

*NetBeans IDE* es un entorno de desarrollo: una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en *Java* - pero puede servir para otro lenguaje de programación. Existe además un número importante de módulos para extender el *NetBeans IDE*. *NetBeans IDE* es un producto libre y gratuito sin restricciones de uso (Netbeans, 2018).

## 1.11 Marco de trabajo

Un marco de trabajo es un conjunto de conceptos, prácticas, herramientas, librerías y criterios para enfocar una de problemática particular, para enfrentar y resolver nuevos problemas de índole similar.

Ejemplos de marcos de trabajo utilizados actualmente u ofrecidos por organismos o empresas de normalización son:

- *Resource Description Framework*, un conjunto de reglas del *World Wide Web Consortium* para describir cualquier recurso de internet, como un sitio *Web* y su contenido.





- *Internet Business Framework*, un grupo de programas que forman la base tecnológica para el producto *mySAP* de *SAP*, la compañía alemana que comercializa una línea de productos de gestión de recursos empresariales.
- *Sender Policy Framework*, un enfoque definido y una programación para hacer más seguro el correo electrónico.
- *Zachman framework*, una estructura lógica destinada a proporcionar una representación integral de una empresa de tecnología de la información que es independiente de las herramientas y métodos utilizados en cualquier negocio de tecnologías de la Información en particular.

Algunas de las características esenciales de los marcos de trabajo son:

- Propone un modelo de colaboración.
- Reutiliza el diseño y el código.
- Incorpora el conocimiento para el dominio que el Marco de trabajo fue diseñado (Framework, 2018).

### **Zeolides v2.2.0**

El marco de trabajo para el desarrollo de aplicaciones *web* es el responsable de proveer un entorno de ejecución para los proyectos con funcionalidades y características óptimas de desarrollo. Brinda prestaciones que aseguran un alto grado de calidad e integración de las aplicaciones realizadas en dicha plataforma tecnológica. Aspectos como la seguridad, el monitoreo de trazas, la gestión del personal, los reportes, entre otros, son contenidos entre las funcionalidades de la solución. Por lo que permite la obtención de soluciones con un alto grado de robustez, dadas por el número de funcionalidades brindadas durante el desarrollo de soluciones *web* (Zeolides, 2017).



### **Zend Framework v1.9.7**

*Zend Framework* (ZF) es un *framework open source* para *PHP* desarrollado por *Zend*. ZF implementa el patrón Modelo Vista Controlador (MVC), es 100% orientado a objetos y sus componentes tienen un bajo acoplamiento por lo que los puedes usar en forma independiente (ZendFramework, 2017).

### **Doctrine v1.2.1**

El *Doctrine Project* es el hogar de varias bibliotecas *PHP* principalmente enfocadas en el almacenamiento de bases de datos y el mapeo de objetos. Los proyectos centrales son un *Object Relational Mapper* (mapeo de objetos relacionales) y la *Database Abstraction Layer* (Capa de abstracción de la base de datos) sobre la que se basa. Doctrine se ha beneficiado enormemente de los conceptos de *Hibernate ORM* y los ha adaptado para adaptarse al lenguaje *PHP* (Doctrine, 2017).

### **1.12 Servidor para aplicaciones web**

Un servidor de aplicaciones es un programa de servidor en un equipo en una red distribuida que proporciona la lógica de negocio para un programa de aplicación. El servidor de aplicaciones se ve frecuentemente como parte de una aplicación de tres niveles, que consta de un servidor gráfico de interfaz de usuario (*GUI*), un servidor de aplicaciones (lógica empresarial) y un servidor de bases de datos y transacciones (Servidor de aplicaciones, 2017).

### **Apache HTTP Server v2.0**

*Apache* es un proyecto de código abierto y uso gratuito, multiplataforma, muy robusto y se destaca por su seguridad y rendimiento. Por otro lado, viene con una licencia sin restricciones y se caracteriza por implementar características de uso frecuente (ApacheServer, 2018).

Las tecnologías utilizadas para la confección de los módulos HSF e HSI son necesariamente las versiones propuestas, por una necesidad de compatibilidad con el sistema Dirección de Vigilancia de la Salud (D.V.S.) al cual va estar integrados los módulos HSF e HSI.



### **1.13 Conclusiones del capítulo**

Después del desarrollo de este capítulo se puede arribar a las siguientes conclusiones:

- El estudio de los principales sistemas de gestión hospitalaria existentes en la actualidad permitió entender los principales procesos, módulos y funcionalidades que estos deben cumplir.
- La selección de tecnologías para el desarrollo de la propuesta de solución, permitió obtener una base tecnológica sólida para la implementación de la misma, basado en tecnologías libres que garantiza la soberanía tecnológica desde el punto de vista a la gestión hospitalaria.
- El estudio de los elementos teórico relacionados con la gestión de las Historias de Salud Familiar e Historias de Salud Individual permitió una mejor comprensión de los procesos a informatizar.



## **Capítulo 2: Análisis y diseño de los módulos de historias de salud individual e historias de salud familiar.**

### **2.1 Introducción**

En este capítulo se abordan las principales características de la propuesta de solución. Se define el modelo de negocio que ilustrará las relaciones entre los principales conceptos del sistema. También, se desarrollan las fases iniciales de la metodología de desarrollo propuesta en el manual Prodesoft: modelación y construcción; se presentan los diferentes artefactos que genera, tales como los diagramas de negocio y de base de datos, el levantamiento de los requisitos funcionales y no funcionales, una descripción de los mismos y su administración.

### **2.2 Propuesta de solución**

Los módulos HSF e HSI tienen como principal característica, la administración y gestión de las historias de salud de los ciudadanos. Estos podrán almacenar, modificar y eliminar la información de los pacientes atendidos por los especialistas de salud, en cada una de las entidades del país. Los módulos propuestos se integran con los módulos de Estructura, Seguridad, Persona y División Política Administrativa (DPA) que permite a los usuarios conocer la situación de los ciudadanos en cualquier entidad de salud pública del país, realizar consultas y modificar las historias de cada individuo.

### **2.3 Modelado del negocio**

Este subcapítulo es una muestra de cómo está conformado el negocio en cuestión, especificando como están relacionados los conceptos y cómo funcionan a través de procesos que manejan los especialistas de salud para realizar las modificaciones en las HSF e HSI.

#### **Modelo conceptual**

A partir de un análisis a los procesos que requieren ser informatizados, se realizó un modelo conceptual donde se aprecian los diferentes conceptos asociados a la gestión de las HSF e HSI por parte de los profesionales de la salud del territorio nacional y la relación estructural de los mismos. A continuación, se describen los conceptos que componen el modelo conceptual:



**Actores:** Profesionales de la salud con la capacidad de modificar las historias de salud de los ciudadanos.

**Pacientes:** Ciudadano al cual se le debe realizar una intervención médica la cual incurrirá en una modificación de la historia clínica de salud.

**Historia de Salud Familiar:** Colección de datos sobre las condiciones médicas que afectan a la familia, así como de las costumbres alimenticias, actividades y entornos.

**Historia de Salud Individual:** Documento básico de la atención médica primaria y puede ser considerada como la guía metodológica para identificar integralmente los problemas de salud de cada persona.

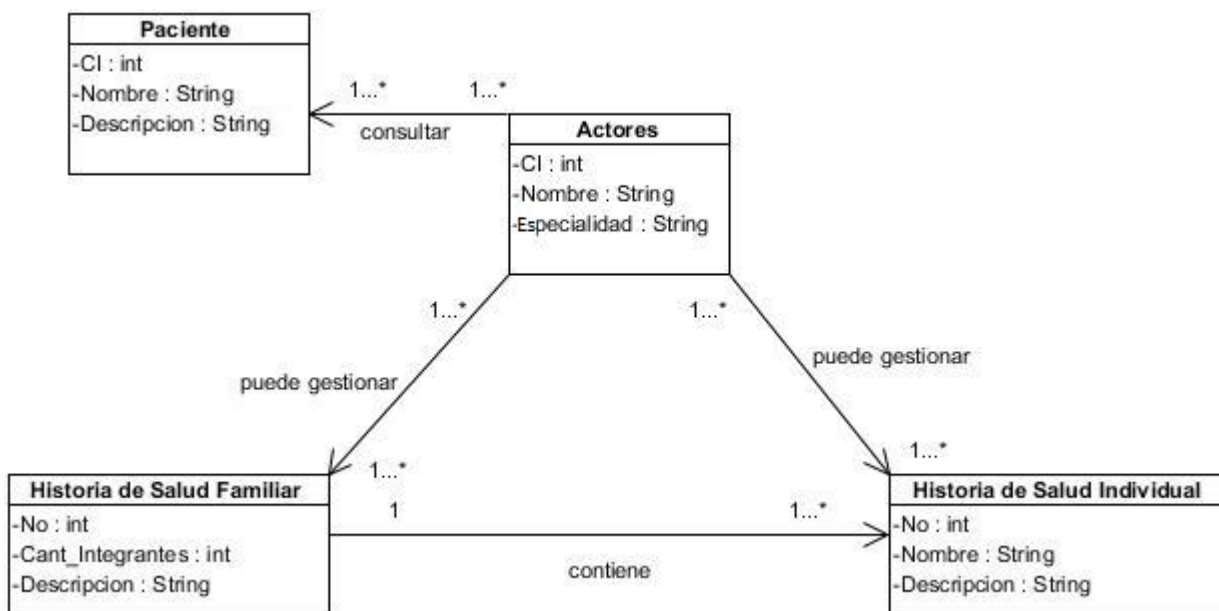


Figura 1. Modelo Conceptual del negocio de la propuesta de solución. (Elaboración propia)

### Descripción de los procesos de negocios

Este mapa de proceso describe el negocio y el flujo de trabajo que sigue la Dirección Nacional del Ministerio de Salud Pública en cuanto a la confección de las HSF e HSI.



Los profesionales de la salud específicamente los médicos interactúan directamente con las HSF e HSI. Primeramente, se realiza la confección de una HSI para una persona específica, en la cual se registrarán todos los problemas de salud de la persona, esta a su vez estará incluida dentro de una HSF la cual recogerá todas las HSI de una familia específica, para su mejor estudio y comprensión.

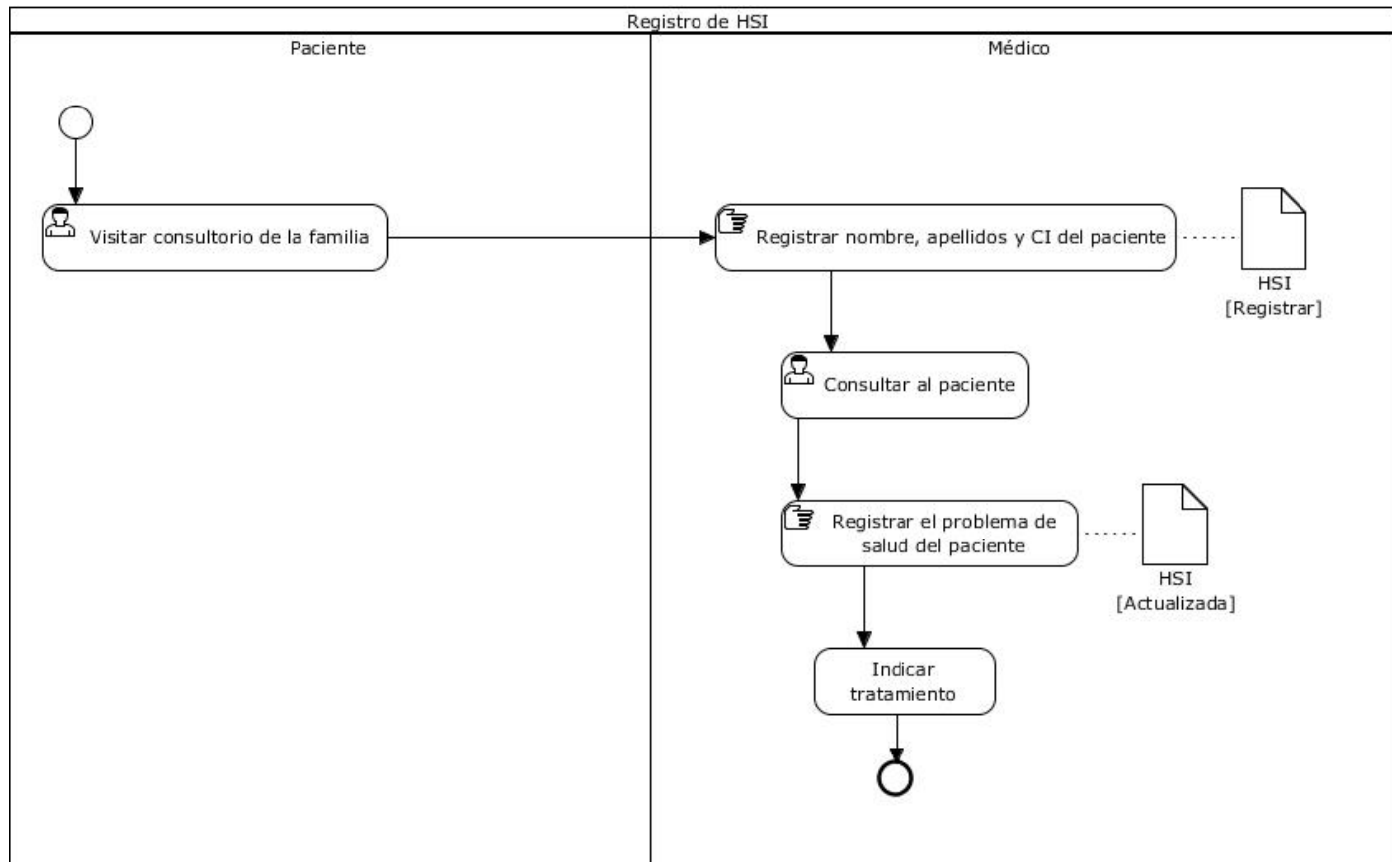


Figura 2. Diagrama de actividades del proceso de negocio gestionar HSI. (Elaboración propia)

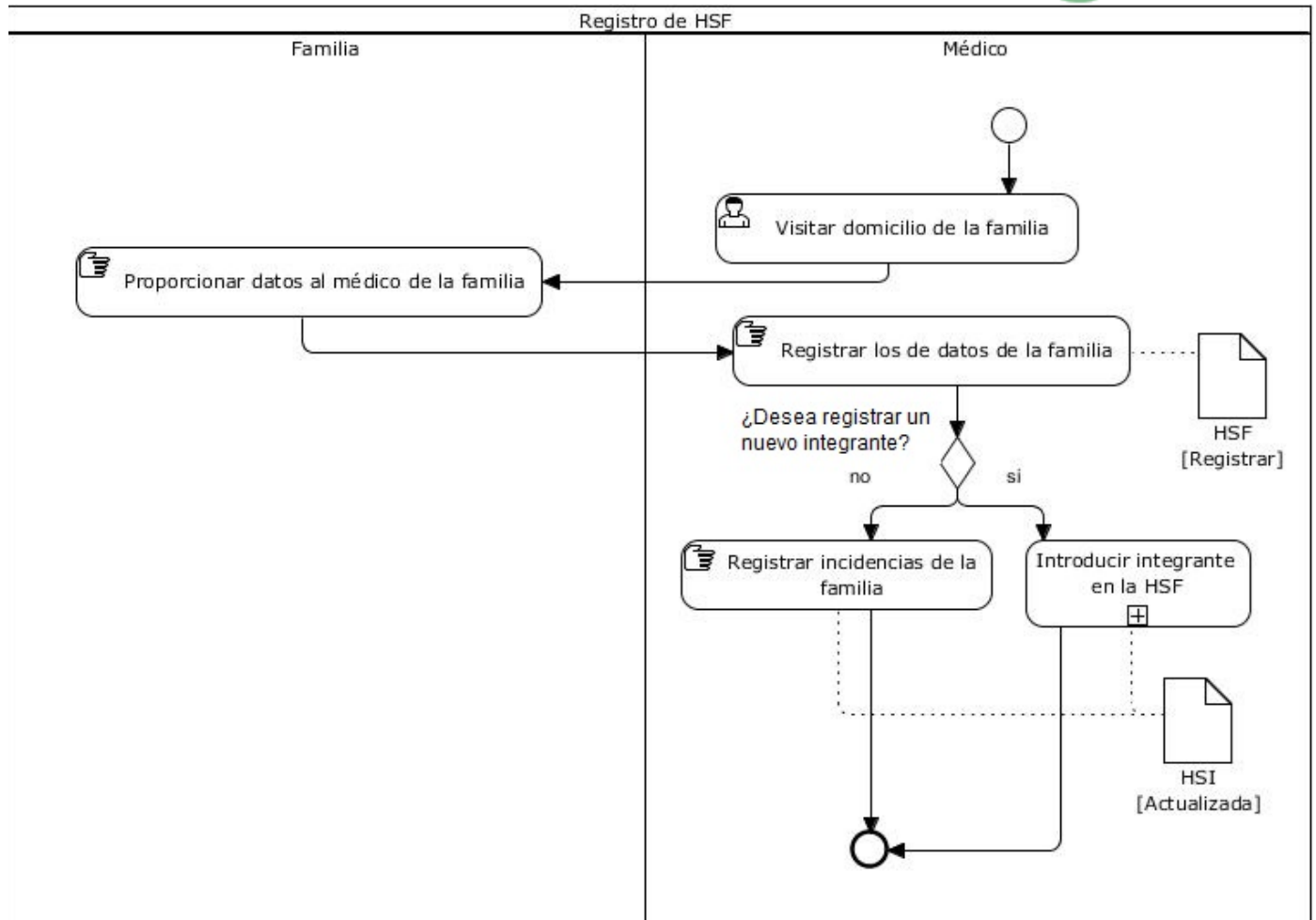


Figura 3. Diagrama de actividades del proceso de negocio gestionar HSF. (Elaboración propia)

## 2.4 Requisitos

En este epígrafe se evidencia el trabajo con los requisitos del sistema, la fuente para su obtención y los métodos utilizados para el levantamiento de los mismos. También contiene la especificación de los requisitos donde se clasifican en funcionales y no funcionales, y la descripción de los principales requisitos funcionales del sistema a través de diagramas de actividades.



### **2.4.1 Fuentes y técnicas para la obtención de requisitos**

La fuente principal para la obtención de los requisitos de sistema fue la documentación que brindo la Dirección Nacional del Ministerio de Salud Pública, las técnicas utilizadas fueron el análisis de la documentación, las entrevistas a profesionales, lo cual permitió una mejor comprensión de las necesidades del cliente y las especificaciones del sistema dando un mejor panorama de los requisitos funcionales y no funcionales.

La técnica de obtención de requisitos **análisis de documentación** consiste en obtener la información sobre los requerimientos funcionales y no funcionales de software a partir de documentos que ya están elaborados. Dicha técnica es útil cuando los expertos en la materia no están disponibles para ser entrevistados o ya no forman parte de la organización (PMOINFORMATICA, 2018).

#### **Documentación consultada**

-Modelos de HSF e HSI (véase en el anexo 2).

### **2.4.2 Especificación de requisitos**

A continuación, se muestran los requisitos funcionales con sus especificaciones para el sistema informático de gestión de HSF e HSI.

#### **2.4.2.1 Requisitos funcionales:**

**RF 1** Adicionar Historia de Salud Individual

**RF 2** Modificar Historia de Salud Individual

**RF 3** Buscar Historia de Salud Individual

**RF 4** Adicionar Historia de Salud Familiar

**RF 5** Modificar Historia de Salud Familiar





**RF 6** Buscar Historia de Salud Familiar

**RF 7** Adicionar integrante de la familia

**RF 8** Eliminar integrante de la familia

**RF 9** Modificar integrante de la familia

**RF 10** Buscar ciudadano

### 2.4.2.2 Requisitos no funcionales del sistema

**RNF1:**

**Tabla 1. Requisitos de Hardware. (Elaboración propia)**

Características	PC Cliente	Servidor de aplicaciones	de Servidor de base de datos
Memoria RAM	1GB	8GB	8GB
Capacidad de almacenamiento	100GB	10GB	500GB
Capacidad de procesamiento	3.0 GHz	16 GHz	16 GHz
Capacidad de transmisión de datos	1MB	1MB	1MB

**RNF2:**

**Tabla 2. Requisitos de Software. (Elaboración propia)**

PC Cliente	Servidor de aplicaciones	Servidor de bases de datos
Mozilla Firefox 57 o superior, recomendado 57.	Sistema operativo Debian server 7.0.	PostgreSQL 9.4 como Sistema Gestor de Base de Datos
Ofimática.	Servidor Web Apache 2.4 o superior, con módulo PHP 5.	
Soporte para formato PDF		



**RNF3: Rendimiento/Eficiencia:**

RNF3.1: Los tiempos de respuesta y velocidad de procesamiento de la información son rápidos, no mayores de 5 segundos para las actualizaciones y 10 para las recuperaciones.

RNF3.2: El tiempo máximo de espera está en dependencia de las operaciones realizadas, el mayor tiempo no excede los 30 segundos.

**RNF4: Aplicación:**

La cantidad máxima de usuarios concurrentes en el sistema está regulada principalmente por las prestaciones de los servidores tanto el servidor *Web* como el de Bases de datos, a mayores prestaciones aumenta la cantidad de usuarios soportados por el sistema. Para los requerimientos mínimos descritos en este documento de *hardware* pueden interactuar con el sistema un total de 100 usuarios sin afrontar dificultades de rendimiento y hasta 150 usuarios obteniendo tiempos de respuesta fuera de los rangos establecidos anteriormente.

**RNF5: Confiabilidad:**

La información del sistema está disponible todo el tiempo que el usuario solicite, debe prever contingencias que pueden afectar la prestación estable y permanente del servicio al desarrollo de aplicaciones. Por ejemplo, prever casos de excepciones y errores por parte de los programadores como usuarios finales del producto final.

**RNF6: Seguridad:**

RNF6.1: La información está protegida contra accesos no autorizados utilizando mecanismos de validación que puedan garantizar el cumplimiento de esto: cuenta, contraseña y nivel de acceso, de manera que cada uno puede tener disponible solamente las opciones relacionadas con su actividad y tenga datos de acceso propios, garantizando así la confidencialidad.

RNF6.2: El módulo de seguridad es el encargado de la generación y control de los usuarios y roles del sistema.



**RNF7: Integridad:**

La información manejada por el *software* es objeto de cuidadosa protección contra la corrupción y estados inconsistentes, mediante el mantenimiento de las bases de datos, así como la salva de la información se realiza a través de cada componente, de igual manera el origen y autoridad de los datos.

**RNF8: Disponibilidad:**

RNF8.1: A los programadores de las aplicaciones como usuarios finales del Marco de Trabajo se les garantiza el acceso a la información y los dispositivos o mecanismos utilizados para lograr la seguridad no oculta o retrasa a los usuarios para obtener los datos deseados en un momento dado de manera que se les permite a los usuarios tener atribución respecto a sus funciones de trabajo.

RNF8.2: Sólo tienen acceso al componente, las personas calificadas para interactuar con el marco de trabajo el cual contiene el subsistema externo de seguridad que garantiza la gestión de roles y usuarios para el sistema.

**RNF9: Confidencialidad:**

La información manejada por el sistema está protegida de acceso no autorizado y divulgación.

**RNF10: Usabilidad:**

RNF10.1: El sistema emplea barras de progreso para indicar el estado de los procesos que por su complejidad requieran de un tiempo de procesamiento apreciable por los usuarios.

RNF10.2: Se utiliza el idioma español para todo el sistema.

RNF10.3: El *software* tiene siempre visible la opción de Ayuda, lo que posibilita un mejor aprovechamiento de sus funcionalidades.



**RNF11: Restricciones de diseño:**

El producto de software está diseñado sobre una arquitectura N-capas y emplea los estándares establecidos en la entidad.

**RNF12: Interfaz:**

Para el desarrollo del sistema se utilizaron los estándares empleados en la empresa. La distribución mínima de la pantalla es de 800 x 600. Se utilizan los protocolos de comunicación *http*, *https* y *tcp/ip*. Soporta los periféricos mouse y teclado.

**RNF13: Portabilidad:**

El sistema es multiplataforma (*Linux* y *Windows* fundamentalmente). Al tratarse de una aplicación *web* puede accederse desde cualquier dispositivo que cumplan con las especificaciones de hardware y *software* establecidas.

**RNF14: Políticos-Culturales:**

El sistema solo contiene palabras en el idioma español y debe respetar los términos empleados por la organización.

**RNF15: Licencias y patentes:**

Utiliza la licencia *General Public License* (GPL).

**RNF16: Aplicación de estándares:**

El sistema está regido por diferentes estándares establecidos:

- 1 Estándar para la modelación de procesos de negocio y gestión de requisitos.
- 2 Estándar de codificación.
- 3 Estándar de documentación.



### 2.4.3 Descripción de requisitos

El sistema está compuesto por 10 requisitos funcionales, a continuación, se expone un ejemplo de diagrama de actividades para el RF 7 Adicionar integrante de la familia que servirá de ejemplo para la descripción de este tipo de requisitos generalizado en el sistema (véase en los anexos #3 el resto de las descripciones).

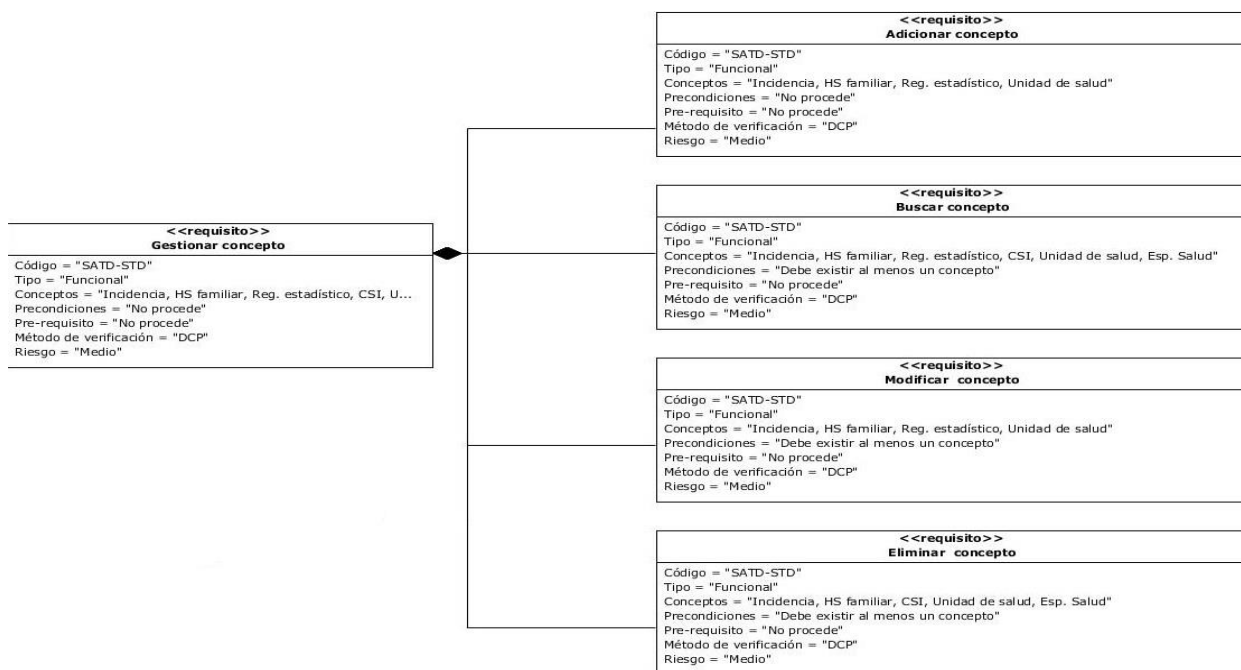


Figura 4. Diagrama de Requisitos caso genérico gestionar concepto. (Elaboración propia)

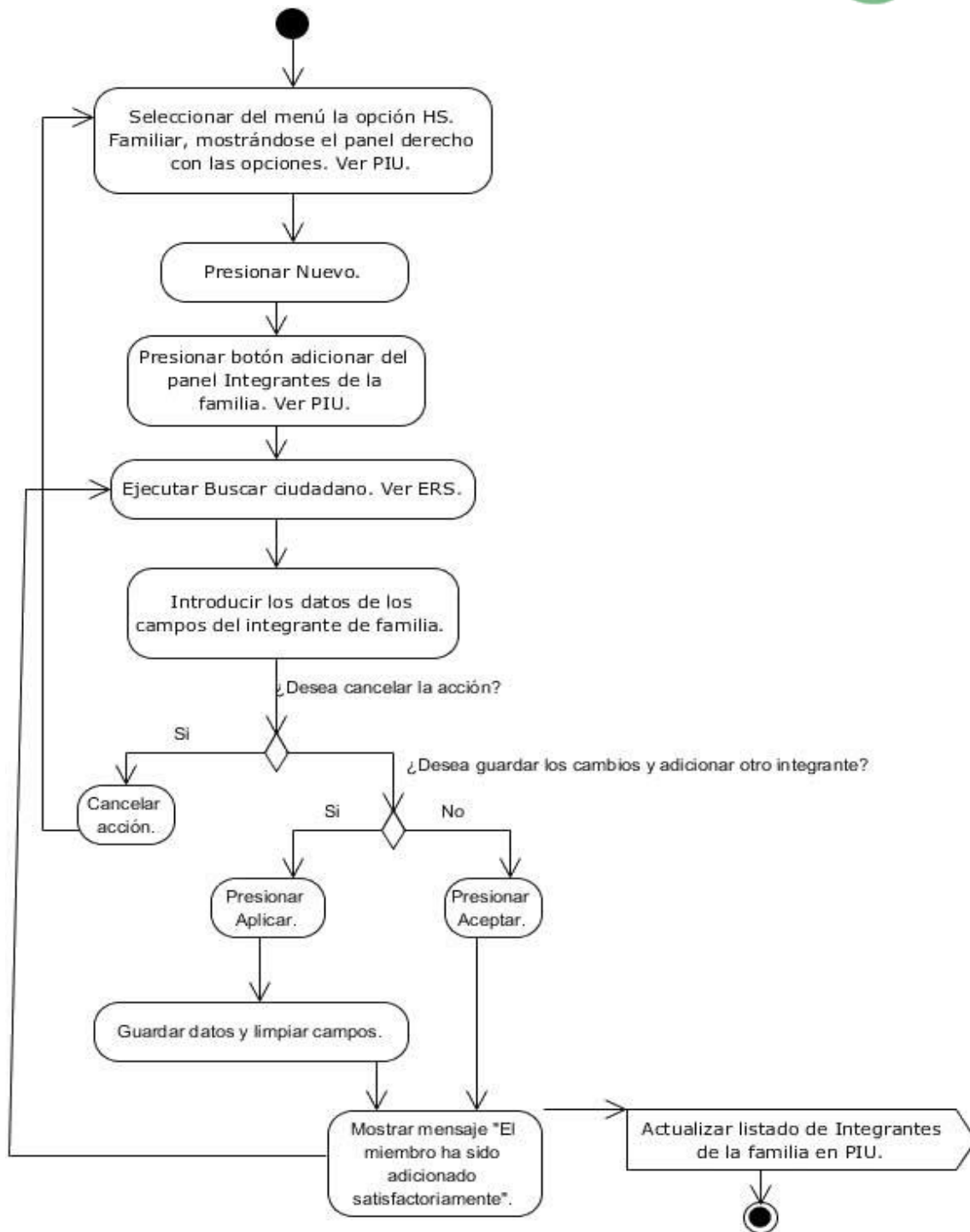


Figura 5. Diagrama de actividades caso adicionar integrante a la familia. (Elaboración propia)



## **2.5 Análisis y diseño**

Este subcapítulo permitirá traducir los requisitos funcionales del sistema a través del diseño arquitectónico, el modelado de datos y modelado del diseño, los cuales mediante refinamientos tienen como propósito ser el esquema para la implementación del sistema.

### **2.5.1 Diseño arquitectónico**

El diseño arquitectónico nos permite establecer un marco para identificar los principales componentes de un sistema y las comunicaciones entre estos componentes.

Para la implementación de la solución se propone utilizar arquitectura en capas y el patrón MVC el cual se utiliza para integrar la capa de presentación con las capas de negocio y/o servicio y está formado por tres niveles:

- El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio. Integra las clases que contienen las consultas a la Base de Datos, así como la definición de las tablas, sus relaciones y atributos.
- La vista transforma el modelo en una página web que permite al usuario interactuar con ella. Determina la interfaz que se muestra finalmente al cliente para su intercambio con la aplicación.
- El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista. Gestiona todas las peticiones del usuario y se encarga de darles respuesta.



### 2.5.2 Modelado de datos

Para el modelado de la base de datos el centro STD utiliza un híbrido entre bases de datos relacionales y no relacionales, se evidencia la relación entre tablas con el uso de las llaves foráneas como método de unificación entre las mismas y con el objetivo de ganar en rendimiento se hace uso de json donde se almacenan los valores de los *datos nomencados* (listado de datos predefinidos) para minimizar el uso de sentencias INNER JOIN debido a que el json almacena los datos que se necesitarían.

A continuación, se muestran los diagramas entidad relación definidos para los módulos de HSF e HSI de La Dirección Nacional del Ministerio de Salud Pública.

El diagrama entidad relación caso HSF consta de 14 entidades de las cuales 2 entidades son externas al módulo. Las entidades y atributos de este caso específico son descritos mediante el diagrama en los anexos #4. A continuación será especificado mediante el mismo método el diagrama entidad relación del caso HSI el cual consta con 5 entidades de las cuales 1 es externa al sistema.



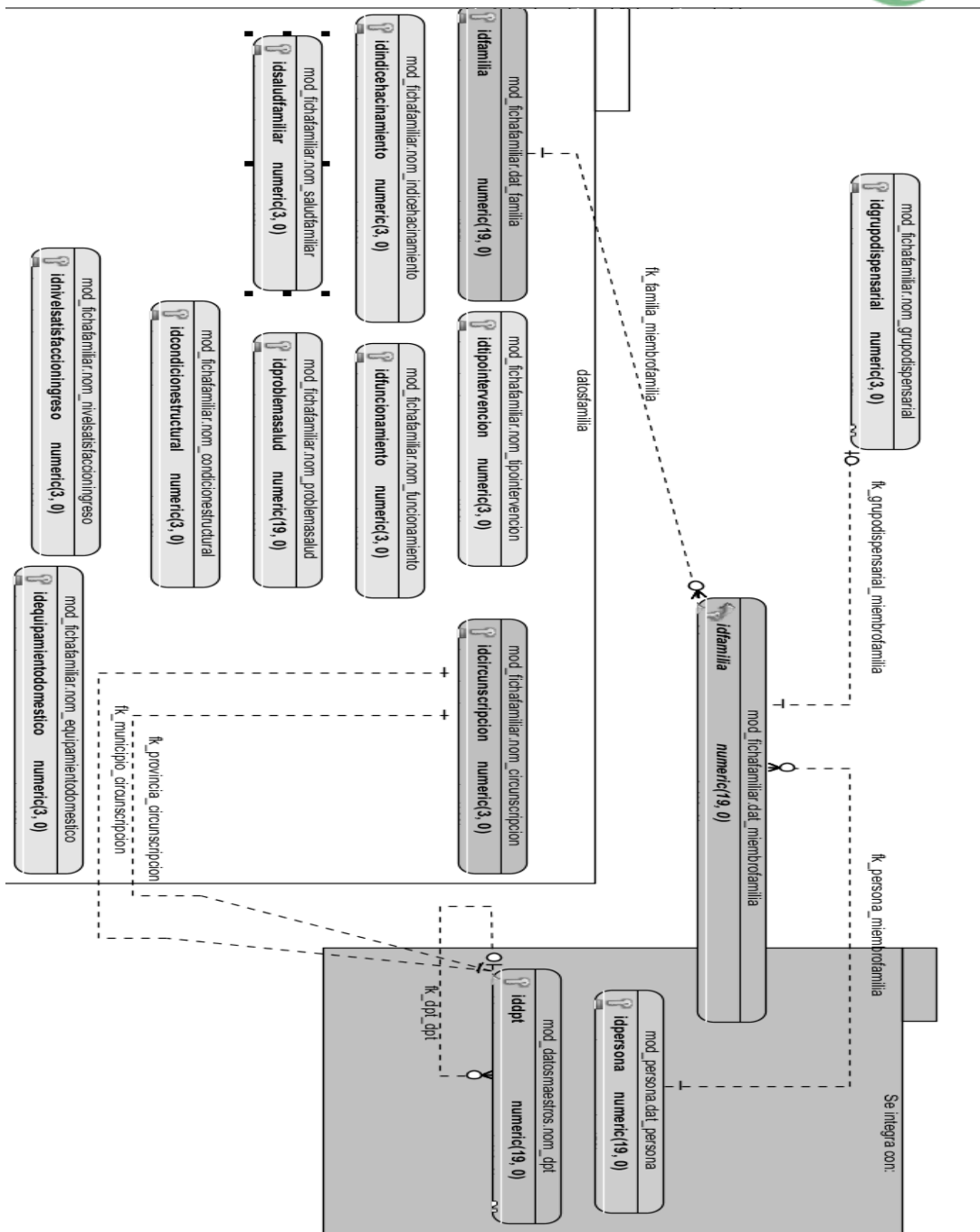


Figura 6. Diagrama entidad relación caso HSF. (Elaboración propia)





### 2.5.2.1 Estándar de base de datos

El desarrollo de los sistemas informáticos se enfoca mucho en las buenas prácticas, patrones de diseño y vocabulario exclusivo como las arquitecturas de los sistemas, entre otros. Y aunque, en la confección de una base de datos no hay que ser tan rigurosos ya que la misma no exige una arquitectura, también es necesario establecer ciertos parámetros a la hora de crear la misma. Por tales motivos es utilizado como estándar de base de datos el estándar de **llave subrogada**.

**Llaves subrogadas:** este patrón es muy utilizado para facilitar la interacción con la base de dato en un futuro. El mismo plantea que se genere una llave primara única para cada entidad, en vez de usar un atributo identificador en el contexto dado. El uso del patrón permite que las tablas sean más fáciles de consultar a partir del identificador, pues los tipos de datos son iguales en cada una de las tablas. Dicho patrón se manifiesta en la definición de la llave primaria de todas las tablas del modelo físico porque se generan llaves primarias independientes a los atributos de las entidades.

### 2.5.2.2 Plan de respaldo de la base de datos

El plan de respaldo de la información de la base de datos será llevado de forma automática por los servidores diariamente en un horario no laboral. Este plan de respaldo es llevado a cabo mediante un minucioso trabajo el cual está reflejado en las siguientes políticas de procedimiento.

#### **Políticas de Procedimiento:**

1. Todas las carpetas de los servidores deberán respaldarse periódicamente.
2. Todos los respaldos deberán conservarse conforme lo acordado con las áreas correspondientes.
3. Los respaldos se harán de acuerdo al documento de políticas de manejo y control de respaldo.
4. Todos los equipos centrales, bases de datos y sistemas informáticos deberán contar con planes de contingencia.
5. El usuario es responsable de los respaldos de los equipos personales.



### 2.5.3 Modelado del diseño

Este epígrafe muestra cómo están relacionados los componentes de cada requisito del sistema, para un mejor entendimiento de las conexiones entre estos.

#### 2.5.3.1 Patrones de diseño

Los patrones de diseño no son más que soluciones a problemas recurrentes en el entorno de desarrollo. Estos se pueden considerar como guías para solucionar varias veces un problema del mismo tipo (Ingeniería del software, 2018).

Permiten solucionar problemas de optimización de código y reutilización de componentes.

En el desarrollo del presente trabajo se utilizaron patrones de diseño para la realización de los diferentes diagramas, tales como los patrones Modelo Vista Controlador, *GRASP* (*General Responsibility Assignment software Patterns*, traducido como: Los patrones generales de asignación de responsabilidad de *software*) y *GOF* (*Gang of Four*, traducido al español Grupo de Cuatro). A continuación, se realiza una descripción de los mismos.

**Patrones *GRASP* utilizados:**

#### **Patrón Controlador:**

El patrón controlador es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado.



```
public function apiRestHsfAction()
{
    try {
        $result = array();
        switch ($this->_request->getParam('query')) {
            case 'getAll':{...7 lines }
            case 'get':{...4 lines }
            case 'insert':{...4 lines }
            case 'delete':
                $ids = json_decode($this->_request->getParam('params'));
                $result = $this->model->delHsf($ids);
                break;
            case 'getnivel':
                $result = $this->model->getNiveleducacional();
                break;
            case 'getproblemasalud':{...6 lines }
            case 'getConsultorios':
```

Figura 8. Método de la clase HsfminsapController. (Elaboración propia)

**Patrón Creador:**

Este patrón se utiliza para la asignación de responsabilidades a las clases relacionadas con la creación de objetos, de forma tal que una instancia de un objeto sólo pueda ser creada por el objeto que contiene la información necesaria para ello.

```
*/
class HsfminsapController extends ZendExt_Controller_Secure
{
    private $model;

    public function init()
    {
        parent::init();
        $this->model = new HsfminsapModel();
    }

    function hsfminsapAction()
    {
        $dirconfig = explode('std_sginflite', str_replace(DIRECTORY_SEPARATOR, '/', __DIR__));
```

Figura 9. Instancia de HsfminsapModel en HsfminsapController. (Elaboración propia)



### Patrón Bajo Acoplamiento:

El acoplamiento mide la fuerza con que una clase está conectada a otra, de esta forma una clase con bajo acoplamiento debe tener un número mínimo de dependencia con otras clases. El uso del patrón Creador contribuye al bajo acoplamiento entre las clases del sistema.

```
$m->save();
Seventos['icono'] = 'folder_shared';
Seventos['color'] = 'blue';
Seventos ['titulo'] = "Historia de salud Individual";
Seventos ['modulo'] = "hsi";
Seventos ['mensaje'] = "Nombre: {$miembro->persona->PrimerNombre} {$miembro->persona->SegundoNombre} {$miembro->persona->PrimerApellido} {$m
Actualmente la información registrada es la siguiente: <br>
Nivel educacional: {$nivel[0]['denominacion']}<br>
Labor que realiza: {$miembro->labor}<br>
Problemas: {$pro}<br>
Observaciones: {$miembro->observaciones}<br>
La información del miembro fue modificada por el usuario: $usuario";
$seg = ZendExt_IoC_Inter::getInstance()->sqlfittle_seguimiento->adicionarSeguimiento('persona', $miembro->idpersona, array($seventos));
}
```

Figura 10. Ejemplo del patrón bajo acoplamiento en la clase HsfminsapModel. (Elaboración propia)

### Patrón GOF utilizado:

#### Singleton (Instancia Única):

En ocasiones es necesario mantener una visibilidad global a una única instancia de una clase de tal forma que sea accedida en cualquier parte de la aplicación, este patrón garantiza el comportamiento descrito anteriormente. Se usa al hacer las consultas a la base de datos (BD) con *Doctrine*, un ejemplo de la aplicación de este patrón.

```
$localidad = $params->localidad->iddpt;
$result1 = Doctrine_Manager::getInstance()
->getConnection()
->execute(" Select mod_datosmaestros.f_obtenerprovinciamcpio('$localidad');");
->fetchAll(PDO::FETCH_ASSOC);
$provMunic = split("/", $result1[0]['f_obtenerprovinciamcpio']);
$historia->municipio = $provMunic[1];
$historia->localidad = $params->localidad->iddpt;
$historia->provincia = $provMunic[0];
$params->miembros = [];
$historia->metadatos = json_encode($params);
$historia->save();
Doctrine_Manager::getInstance()
->getConnection()
->commit();
```

Figura 11. Uso de consultas a la Base de Datos con Doctrine. (Elaboración propia)



### 2.5.3.2 Diagramas de clases del diseño

Los diagramas de clase del diseño son utilizados en el proceso de desarrollo de software para un mejor entendimiento de la información que se maneja en el sistema y los componentes que se encargaran del funcionamiento y la relación entre uno y otro.

A continuación, se muestra un ejemplo de diagrama de clases del diseño con estereotipos *web*, el resto se podrán apreciar en los anexos #5.

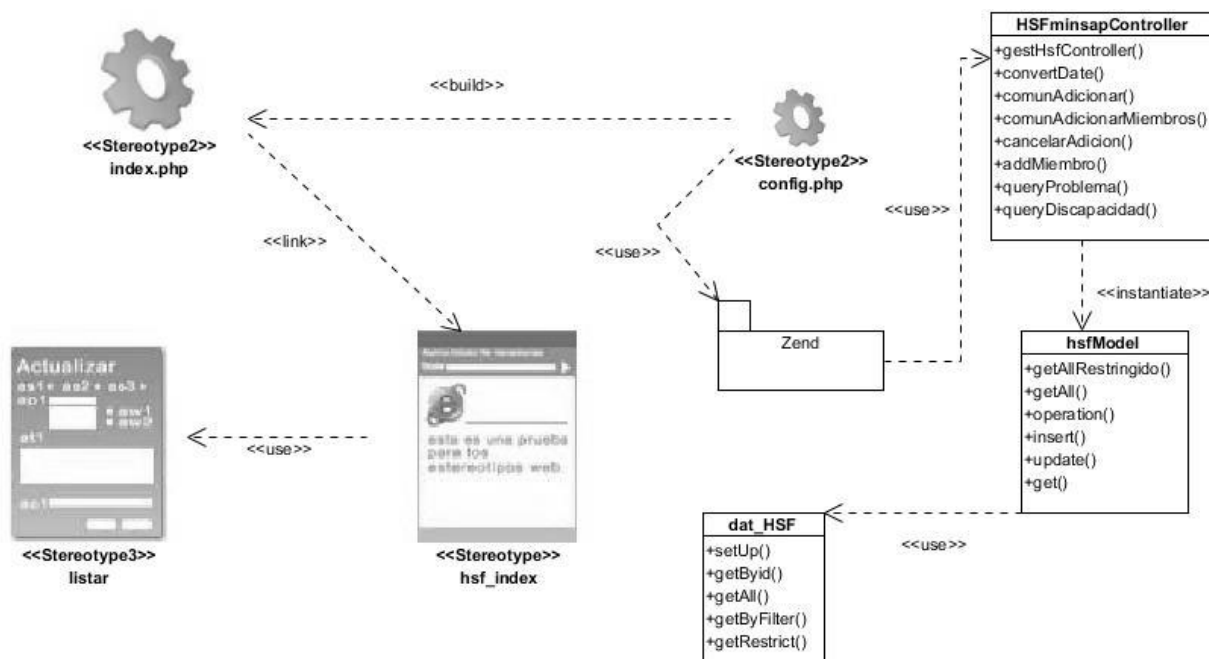


Figura 12. Diagrama de clases del diseño con estereotipos web caso listar HSF. (Elaboración propia)



## 2.6 Prototipos de interfaz de usuario

La representación de los prototipos de interfaz de usuario (PIU) se realizó siguiendo la metodología utilizada para la elaboración de este trabajo investigativo.

A continuación, podemos ver el PIU para el requisito adicionar familia para los demás ver Anexo 6.

The form is divided into several sections:

- Location:** Circunscripción (dropdown), Edificio (text input), Apartamento (text input).
- Integrantes de la familia:** A table with columns: Nombre y apellidos, Edad, Sexo, CI. Buttons: Adicionar, Editar, Eliminar.
- Evaluación familiar:** Funcionamiento familiar (dropdown), Evaluación de la salud familiar (dropdown), Intervención familiar (dropdown). Anotaciones especiales: (text area).
- Otro datos de la localización y dirección de la vivienda:** Localización (text input), Fecha de evaluación (text input), Dirección (text input).
- Condiciones de la vida material de la familia:** Condiciones estructurales (dropdown), Índice de hacinamiento (dropdown), Equipamiento del hogar (dropdown), Satisfacción de la familia de los ingresos (dropdown). Observaciones: (text area).

Buttons at the bottom: Cancelar, Aplicar, Aceptar.

Figura 13. PIU 1 Adicionar Familia. (Elaboración propia)





## **2.7 Conclusiones del capítulo**

La confección del capítulo anterior propone las siguientes conclusiones:

- La definición de los requisitos funcionales y no funcionales de la solución propuesta permitió una mejor comprensión de los resultados que se pretenden obtener de una manera más coherente y sirvieron de guía para la implementación de los módulos.
- La aplicación de patrones GOF y GRASP permitió al autor definir la estructura interna del software obtenido, garantizando en todo momento el cumplimiento de buenas prácticas de desarrollo de software, la reutilización de código y la simplificación de futuras tareas de mantenimiento.



## **Capítulo 3: Implementación, pruebas y validación de los módulos de historias de salud individual e historias de salud familiar**

### **3.1 Introducción**

En este capítulo se describe la implementación de la propuesta de solución, fase donde se cumple con los requisitos obtenidos al inicio de la investigación. Se crean los diagramas de componentes y de despliegue, además de presentarse las principales interfaces de usuario de la solución. A partir del código resultante y su funcionamiento se ejecutan las pruebas unitarias y de aceptación a la solución desarrollada.

### **3.2 Implementación**

Una vez concluido el diseño de la aplicación, corresponde la fase de implementación de la propuesta de solución. Donde todos los objetivos de esta fase van destinados al desarrollo de la aplicación.

#### **3.2.1 Modelo de implementación**

##### **3.2.1.1 Diagrama de componentes**

Los diagramas de componentes permiten visualizar la estructura del sistema y el comportamiento del servicio que estos componentes proporcionan al sistema desarrollado. En la siguiente figura se muestra el diagrama de componentes de la solución.

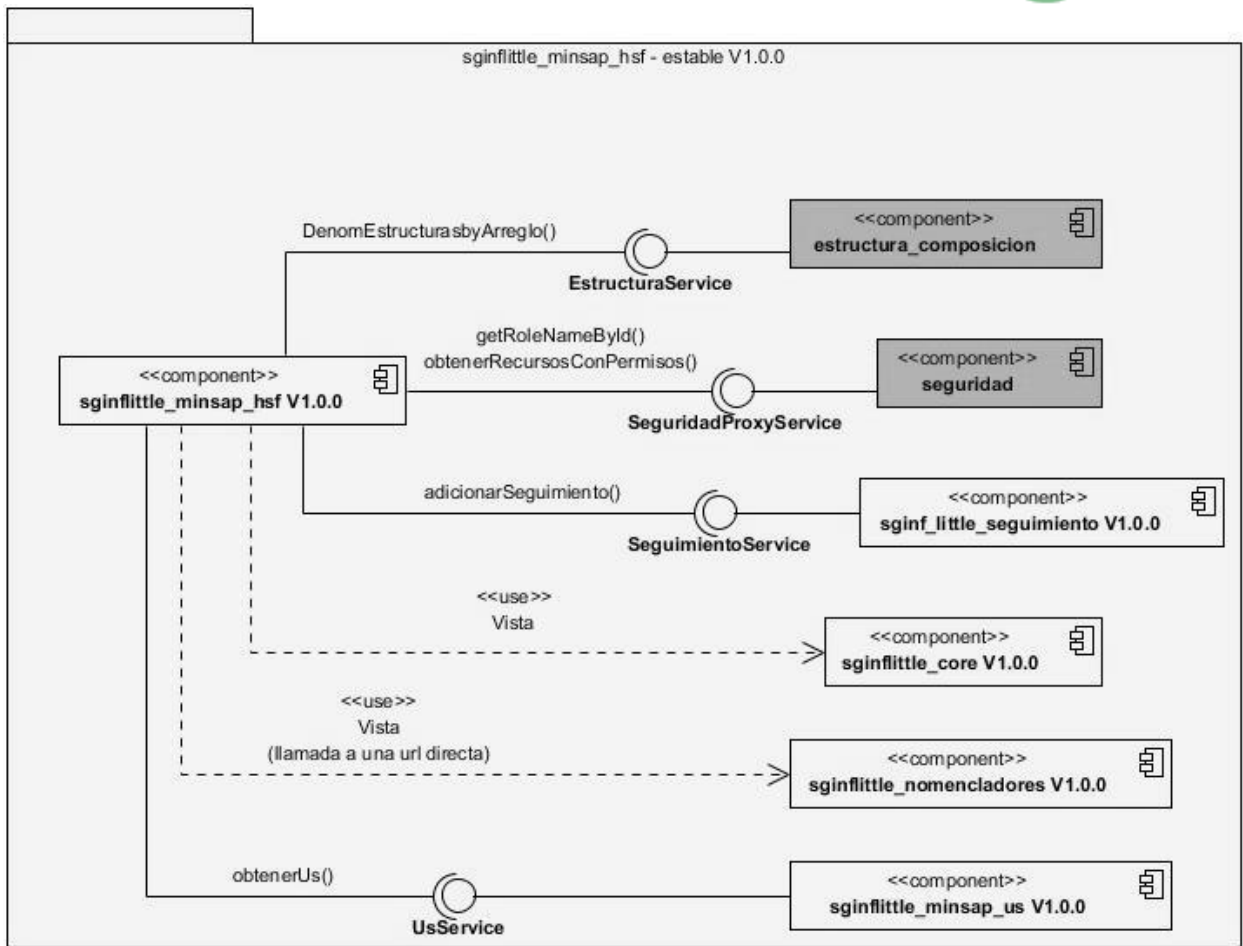


Figura 14. Diagrama de componentes. (Elaboración propia)

### 3.2.1.2 Diagrama de despliegue

En este diagrama se define el nodo PC-Cliente desde donde se originan las peticiones mediante el protocolo *HTTPS* al sistema DVS alojado en el servidor de aplicaciones *web*. El sistema mantendrá una comunicación con el servidor de Base de Datos *PostgreSQL* a través del protocolo *TCP/IP*.

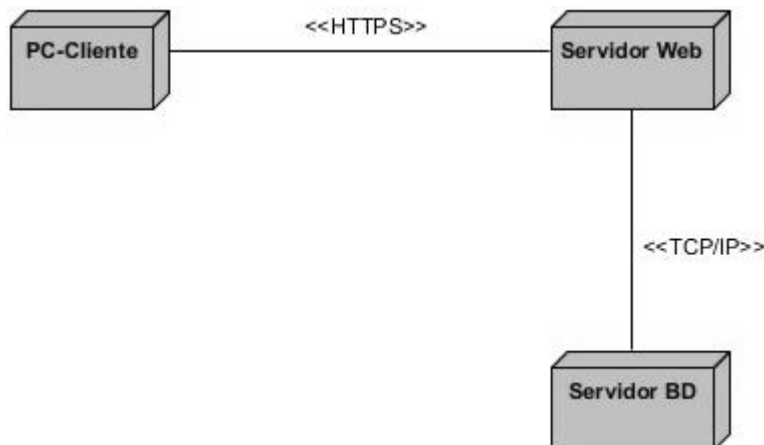


Figura 15. Diagrama de despliegue. (Elaboración propia)

### 3.2.2 Estándares de implementación

Los estándares de implementación se basan en la unificación del trabajo para un mayor entendimiento del código desarrollado. Los mismos se proponen para contribuir a una sola idea de desarrollo para esto se utilizaron los siguientes estándares de codificación propuestos en los documentos “Estándar de Codificación para PHP” v 2.0, y “Estándar de Diseño de Interfaces para las aplicaciones de Gestión” v 1.0.

Algunos de los estándares reflejados en estos documentos son:

- ✓ El nombre de una clase comienza con la primera letra en mayúscula.

Ejemplo: `class HsfminsapModel extends ZendExt_Model`

- ✓ El nombre de los atributos de una clase comienza con la primera letra en minúscula.

Ejemplo: `($historia = new datHSF());`

- ✓ Los nombres de las clases pertenecientes al dominio deben empezar con la palabra “Dat” y el de las clases del modelo deben tener la palabra “Model” al final.

Ejemplo: `(hsfModel.js)`

En las clases controladoras:

- ✓ Los nombres de los métodos serán definidos con la palabra “Action” al final.

Ejemplo: `(vm.gestActions = function(action) {...})`

- ✓ Cada método será público.



- ✓ El nombre de cada clase controladora debe terminar con la palabra “Controller”.  
Ejemplo: (gestHsfControlller.js)
- ✓ Se utilizará el método Post para las llamadas.

### 3.2.3 Interfaz gráfica de usuario

La interfaz de usuario es el medio mediante el cual el usuario puede comunicarse directamente con la aplicación. Las mismas deben ser de fácil comprensión y agradables para el usuario final. A continuación, se muestran las interfaces de usuario de la solución desarrollada.

Ciudadano

CI	NOMBRE Y APELLIDOS	EDAD	SEXO	DIRECCIÓN	G.D
94031328162	Ariel Humberto Corrales Sardina	24	M	Calle 26 e/ Calle 8va y Calle Maximo Gomez, Edificio 7 Apto 13, Reparto Roble, Guanabacoa, La Habana.	I

Problema(s) de salud

Nivel educacional

Labor que realiza

Observaciones

Figura 16. Crear HSI. (Elaboración propia)

Módulos Historia de Salud Individual e Historia de Salud Familiar para la Dirección de Registros Médicos del Ministerio de Salud Pública.



**▼ Buscar Ciudadano**

CI: 94031328162

Nombre:

Segundo nombre:

Primer apellido:

Segundo apellido:

Provincia:

Municipio:

Sexo:

Fecha nacimiento:

**BUSCAR**

**Listado de ciudadanos**

CI	NOMBRE Y APELLIDOS	EDAD	SEXO	DIRECCIÓN	
94031328162	Ariel Humberto Corrales Sardina	24	M	Calle 26 e/ Calle 8va y Calle Maximo Gomez, Edificio 7 Apto 13, Reparto Roble, Guanabacoa, La Habana.	<input checked="" type="checkbox"/>

**Figura 17. Buscar HSI. (Elaboración propia)**

**▼ Datos primarios**

Circunscripción \*: 32

Edificio: 48

Vivienda/Apto \*: 104

Consultorio \*: CMF UCI 1 / LA LISA

Localización \*: CUBA, LA HABANA

Dirección \*: UCI

**▼ Integrantes de la Familia**

**+ NUEVO**

NO	NOMBRE(S) Y APELLIDOS	EDAD	SEXO	PROBLEMAS DE SALUD	GRUPO D.	
1	Cesar Manuel Cruzata de la Cruz	27	M		I	<input checked="" type="checkbox"/> <input type="checkbox"/>

**▼ Evaluación familiar**

Funcionamiento familiar: Riesgo de disfuncionalidad

Evaluación de salud: Problema de salud por condiciones materiales d

Intervención: Educativa

**Figura 18. Crear y Editar HSF. (Elaboración propia)**



▼ Filtro de búsqueda

Circunscripción <input style="width: 95%;" type="text"/>	Consultorio <input style="width: 95%;" type="text"/>	Edificio <input style="width: 95%;" type="text"/>	Vivienda/Apto <input style="width: 95%;" type="text"/>	HSF <input style="width: 95%;" type="text"/>	CI <input style="width: 95%;" type="text"/>
---	---	--	---	---	--

+ NUEVO

HSF	INT	EVALUACIÓN	FUNCIONAMIENTO	FECHA ACTUALIZACIÓN	
32.48.104	1	Problema de salud por condiciones materiales de la vida	Riesgo de disfuncionalidad	24/05/2018	✓ x
32.148.104	1	Problema de salud por condiciones materiales de la vida	Riesgo de disfuncionalidad	08/05/2018	✓ x
32.6.307	2	Problema de salud por condiciones materiales de la vida	Riesgo de disfuncionalidad	08/05/2018	✓ x
32.119.105	2	Problema de salud por condiciones materiales de la vida	Riesgo de disfuncionalidad	04/05/2018	✓ x
32.22.206	1	Problema de salud por condiciones materiales de la vida	Riesgo de disfuncionalidad	25/04/2018	✓ x
32.22.206	1	Problema de salud por condiciones materiales de la vida	Riesgo de disfuncionalidad	25/04/2018	✓ x
32.75.204	1	Problema de salud por condiciones materiales de la vida	Riesgo de disfuncionalidad	12/04/2018	✓ x

**Figura 19. Listar HSF. (Elaboración propia)**

### 3.3 Pruebas de software

Las pruebas de *software* son acciones que se le realizan a la aplicación para verificar la calidad del producto de *software* desarrollado. Las mismas tienen como objetivo la detección de errores y defectos que pasaron desapercibidos durante la implementación. En la siguiente tabla se muestra la estrategia de pruebas propuesta para aplicar a los módulos HSF e HSI.

**Tabla 3. Estrategia de pruebas. (Elaboración propia)**

Nombre de la prueba	Método	Técnica
Funcionales	Caja Negra	Partición equivalente
Integración	Caja Negra	Casos de prueba
Sistema (carga y estrés)	Caja Negra	Automática mediante la herramienta JMETER
Unidad	Caja Blanca	Diagrama Piramidal y Gráfico de Abstracción/Inestabilidad por la



herramienta Jenkins.

### 3.3.1 Métodos de pruebas

**Las pruebas de caja blanca** realizan un seguimiento del código fuente según va ejecutando los casos de prueba, de forma que se determinan concretamente las instrucciones, bloques, etc. en los que existen errores.

**Las pruebas de caja negra** permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos.

#### 3.3.1.1 Tipos de pruebas de caja negra

##### Pruebas funcionales:

Prueba realizada al requisito buscar ciudadano.

**Tabla 4. Prueba realizada al requisito buscar ciudadano e HSI. (Elaboración propia)**

Id del escenario	Escenario	Variables	
		C.I	Respuesta del sistema
Ep.1.1	Buscar ciudadano e HSI de forma correcta	V [C.I (Con formato incorrecto y tamaño válido)]	Muestra una alerta de si encontró o no el ciudadano y lo muestra.
Ep.1.2	Buscar ciudadano e HSI de forma incorrecta	I [C.I (Con formato incorrecto y tamaño válido)]	El campo se muestra rojo y no realiza ninguna acción.
		I [C.I (Con formato correcto y tamaño incorrecto)]	El campo se muestra rojo y no realiza ninguna acción.
		I [C.I (Con formato y tamaño incorrecto)]	El campo se muestra rojo y no realiza ninguna acción.





Valores correctos:

C.I: Números y tamaño de 9 caracteres.

**Tabla 5. Prueba realizada al requisito añadir HSI. (Elaboración propia)**

Variables			
<b>Id del escenario</b>	<b>Escenario</b>	<b>Problemas de salud</b>	<b>Respuesta del sistema</b>
Ep.1.1	Adicionar HSI de forma correcta	V [Problema de salud (seleccionando las opciones válidas)]	Muestra una alerta especificando que se modificó la HSI.
Ep.1.2	Adicionar HSI sin completar los campos	I [Problema de salud (sin seleccionar opciones)]	El campo no realiza ninguna acción.

**Tabla 6. Prueba realizada al requisito modificar HSI. (Elaboración propia)**

Variables			
<b>Id del escenario</b>	<b>Escenario</b>	<b>Problemas de salud</b>	<b>Respuesta del sistema</b>
Ep.1.1	Modificar HSI de forma correcta	V [Problema de salud (seleccionando las opciones válidas)]	Muestra una alerta especificando que se modificó la HSI.
Ep.1.2	Modificar HSI de forma incorrecta	I [Problema de salud (sin seleccionar opciones)]	El campo no realiza ninguna acción.



**Tabla 7. Prueba realizada al requisito adicionar HSF. (Elaboración propia)**

Variables								
<b>Id del escenario</b>	<b>Escenario</b>	<b>Circunscripción</b>	<b>Localización</b>	<b>Vivienda/Apto</b>	<b>Consultorio</b>	<b>Dirección</b>	<b>Respuesta del sistema</b>	
Ep.1.1	Adicionar HSF de forma correcta	V [ seleccionando valores correctos]	V [ seleccionando valores correctos]	V [valores correctos]	V [ seleccionando valores correctos]	V [ seleccionando valores]	Muestra una alerta especificando que se modificó la HSF	
Ep.1.2	Adicionar HSF de forma incorrecta	I [sin seleccionar valores correctos]	V [ seleccionando valores correctos]	V [valores correctos]	V [ seleccionando valores correctos]	V [ seleccionando valores]	El campo se muestra rojo y no realiza ninguna acción.	
		V [ seleccionando valores correctos]	I [sin seleccionar valores correctos]	V [valores correctos]	V [ seleccionando valores correctos]	V [ seleccionando valores]	El campo se muestra rojo y no realiza ninguna acción.	
		V [ seleccionando valores correctos]	V [ seleccionando valores correctos]	I [valores incorrectos]	V [ seleccionando valores correctos]	V [ seleccionando valores]	El campo se muestra rojo y no realiza ninguna acción.	
		V [ seleccionando valores correctos]	V [ seleccionando valores correctos]	V [valores correctos]	I [sin seleccionar valores correctos]	V [ seleccionando valores]	El campo se muestra rojo y no realiza ninguna acción.	
		V [ seleccionando valores correctos]	V [ seleccionando valores correctos]	V [ seleccionando valores correctos]	V [valores correctos]	V [ seleccionando valores correctos]	I [sin seleccionar valores]	El campo se muestra rojo y no realiza ninguna acción.
		I [sin seleccionar valores correctos]	I [sin seleccionar valores correctos]	I [valores incorrectos]	I [sin seleccionar valores correctos]	I [sin seleccionar valores]	Los campos se muestra rojo y no realiza ninguna acción.	

Valores correctos:

Circunscripción: Valores numéricos.

Localización: Selección de un valor de las opciones.

Vivienda/Apto: valores numéricos.

Consultorio: Selección de un valor de las opciones.

Dirección: Campo de llenado obligatorio.



**Tabla 8. Prueba realizada al requisito buscar HSF. (Elaboración propia)**

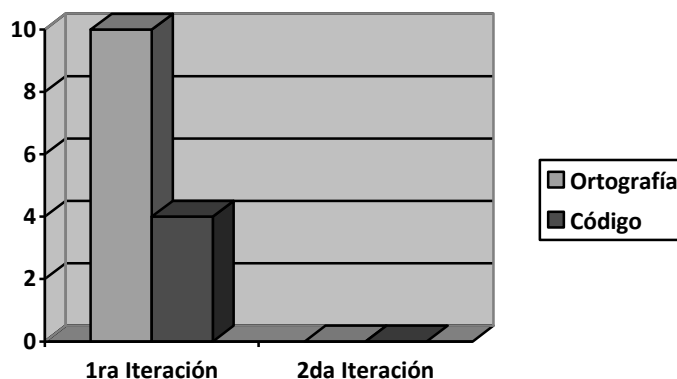
Variables				
<b>Id del escenario</b>	<b>Escenario</b>	<b>Circunscripción</b>	<b>Consultorio</b>	<b>Respuesta del sistema</b>
Ep.1.1	Buscar HSF de forma correcta	V [ seleccionando valores correctos]	V [ seleccionando valores correctos]	Muestra una alerta especificando que se encontró la HSF
Ep.1.2	Buscar HSF de forma incorrecta	I [sin seleccionar valores correctos]	V [ seleccionando valores correctos]	El campo se muestra rojo y no realiza ninguna acción.
		V [ seleccionando valores correctos]	I [sin seleccionar valores correctos]	El campo se muestra rojo y no realiza ninguna acción.
		I [sin seleccionar valores correctos]	I [sin seleccionar valores correctos]	Los campos se muestran rojos y no realizan ninguna acción.

Como resultado final de las pruebas se obtuvo que:

En una primera iteración de las pruebas se detectaron un total de 14 no conformidades de las cuales 10 eran sobre faltas de ortografía y las otras 4 en relación al código, quedando pendiente para una nueva iteración. Para la segunda iteración de las pruebas se corrigieron los errores encontrados en la primera iteración y no se detectaron nuevas no conformidades concluyendo las mismas con 2 iteraciones.



Tabla 9. Resumen sobre Pruebas unitarias. (Elaboración propia)



### 3.3.1.2 Pruebas de Integración

La prueba de integración es una técnica sistemática para construir la estructura del programa mientras que, al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados con la interacción.

En la validación de la solución se probó la integración de los módulos HSF e HSI con los módulos Estructura, Seguridad, División Política Administrativa(D.P.A.) Persona y Datos Maestros. Las siguientes tablas muestran los casos de prueba de integración realizado entre los módulos. Al finalizar las pruebas de integración no se detectaron errores asociados a la interacción entre los módulos HSF e HSI con los módulos ya mencionados.



**Tabla 10. Prueba de Integración módulos Estructura y Seguridad. (Elaboración propia)**

<b>Módulo al cual se integra</b>	<b>Estructura y Seguridad</b>
<b>Condiciones de Ejecución</b>	Los módulos de Estructura y Seguridad almacenen los datos en la base de datos central y exista conexión con la misma.
<b>Descripción de la prueba</b>	Comprobar que los módulos HSF e HSI son capaces de realizar el acceso a funcionalidades y seguridad de negocio a partir de la información gestionada por los módulos de Estructura y Seguridad.
<b>Entradas/Pasos de ejecución</b>	Los módulos de Estructura y Seguridad introducen en la base de datos central los datos y los módulos HSF e HSI consultan estos datos y definen la seguridad del sistema.
<b>Resultado esperado</b>	Los usuarios tienen acceso a las funcionalidades de acuerdo al rol y sus responsabilidades.
<b>Evaluación</b>	Prueba satisfactoria.

**Tabla 11 Prueba de Integración módulo Persona. (Elaboración propia)**

<b>Módulo al cual se integra</b>	<b>Persona</b>
<b>Condiciones de Ejecución</b>	El módulo Persona almacene una nueva persona en la base de datos central y exista conexión con la misma.
<b>Descripción de la prueba</b>	Comprobar que los módulos HSF e HSI son capaces de visualizar las nuevas personas gestionadas por el módulo de Persona.
<b>Entradas/Pasos de ejecución</b>	El módulo Persona introduce en la base de datos central los datos de las nuevas personas y los módulos HSF e HSI consultan estos datos en el sistema.
<b>Resultado esperado</b>	Los usuarios tienen acceso a los datos introducidos por el módulo Persona.



<b>Evaluación</b>	Prueba satisfactoria.
-------------------	-----------------------

Tabla 12. Prueba de Integración módulo Datos Maestros. (Elaboración propia)

Módulo al cual se integra	Datos Maestros
<b>Condiciones de Ejecución</b>	El módulo Datos Maestros introduce los datos en la base de datos central y exista conexión con la misma.
<b>Descripción de la prueba</b>	Comprobar que los módulos HSF e HSI son capaces de realizar el acceso a funcionalidades y seguridad de negocio a partir de la información gestionada por el módulo Datos Maestros.
<b>Entradas/Pasos de ejecución</b>	El módulo Datos Maestros introduce en la base de datos central los datos y los módulos HSF e HSI consultan estos datos en el sistema.
<b>Resultado esperado</b>	Los usuarios tienen acceso a los datos introducidos de acuerdo al rol y sus responsabilidades.
<b>Evaluación</b>	Prueba satisfactoria.

### 3.3.1.3 Pruebas de Carga y Estrés

Las pruebas de rendimiento son realizadas luego de que el sistema está completamente integrado, con el objetivo de observar el comportamiento del sistema bajo una cantidad de peticiones esperadas y comprobar que cumpla determinadas tareas en condiciones particulares de trabajo.

Esta prueba fue realizada para 100 usuarios conectados en una 1ra iteración. Para una mejor comprensión de la tabla se detallan a continuación cada uno de los parámetros usados:

- **Muestras #:** indica la cantidad de usuarios haciendo peticiones de manera concurrente.
- **Media:** indica el máximo de tiempo de ejecución invertido para una petición.
- **Mediana:** significa que el 50% de las peticiones realizadas tardaron menos del valor reflejado.
- **Min:** indica el mínimo de tiempo de ejecución invertido para una petición.
- **Max:** indica el máximo de tiempo de ejecución invertido para una petición.

Módulos Historia de Salud Individual e Historia de Salud Familiar para la Dirección de Registros Médicos del Ministerio de Salud Pública.



- **% Error:** indica la relación entre el total de peticiones y el número de peticiones que originaron errores.
- **Rendimiento:** hace referencia al número de peticiones que el servidor puede procesar en un segundo.
- **Kb/sec:** rendimiento medido en Kilobytes por segundo.

**Tabla 13. Resultados de carga y estrés para 100 usuarios. (Elaboración propia)**

Etiqueta	# Muestras	Media	Mediana	Línea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
3184 /std/st...	100	1757	1872	2263	849	2363	0,00%	40,9/sec	70,9
3186 /std/st...	100	378	372	697	24	860	0,00%	61,7/sec	35,7
3187 /std/st...	100	168	43	486	26	653	0,00%	91,8/sec	245,4
3185 /std/st...	100	59	38	101	26	320	0,00%	195,7/sec	1358,2
3188 /std/st...	100	34	33	39	25	184	0,00%	267,4/sec	852,5
3189 /std/st...	100	37	29	37	24	397	0,00%	137,4/sec	473,3
3190 /favico...	100	51	29	36	23	737	0,00%	69,6/sec	19,6
3191 /std/st...	100	88	36	187	26	745	0,00%	46,5/sec	419,5
3192 /std/st...	100	212	195	402	28	779	0,00%	35,7/sec	321,9
3195 /std/st...	100	569	550	885	154	1160	0,00%	25,4/sec	44,1
3193 /std/st...	100	864	885	1010	397	1249	0,00%	19,9/sec	34,5
3197 /std/st...	100	981	974	1097	815	1215	0,00%	18,2/sec	31,5
3194 /std/st...	100	1044	1059	1156	800	1229	0,00%	18,4/sec	31,9
3198 /std/st...	100	1063	1061	1208	760	1431	0,00%	18,3/sec	31,7
3196 /std/st...	100	1034	1028	1211	710	1427	0,00%	18,3/sec	31,7
3202 /std/st...	100	224	208	375	3	588	0,00%	21,9/sec	34,0
3199 /std/st...	100	929	896	1055	706	1608	0,00%	21,2/sec	36,7
3201 /std/st...	100	968	979	1173	212	1510	0,00%	25,2/sec	43,2
3200 /std/st...	100	922	965	1240	76	1679	0,00%	31,3/sec	51,4
3203 /std/st...	100	522	490	878	52	1259	0,00%	47,0/sec	73,1
3206 /succe...	100	0	0	0	0	1	100,00%	72,3/sec	120,0
3207 /succe...	100	0	0	0	0	0	100,00%	72,5/sec	120,4
Total	2200	541	411	1124	0	2363	9,09%	176,5/sec	471,7

En cada una de las iteraciones con la muestra de usuarios establecida la ocurrencia de errores es de 0.00% lo que demuestra que las peticiones son realizadas de forma satisfactoria en todo momento.



### 3.3.1.4 Tipos de pruebas de caja blanca

#### Herramienta utilizada Jenkins:

La pirámide mostrada a continuación representa en su parte derecha el nivel de acoplamiento existente entre las clases de los módulos. Además, en su parte izquierda se hace referencia al tamaño de los módulos teniendo en cuenta las cantidades de paquetes (NOP), clases (NOC), métodos (NOM) y líneas de código (LOC), así como también la suma de complejidades ciclomáticas de todos los métodos de los módulos.

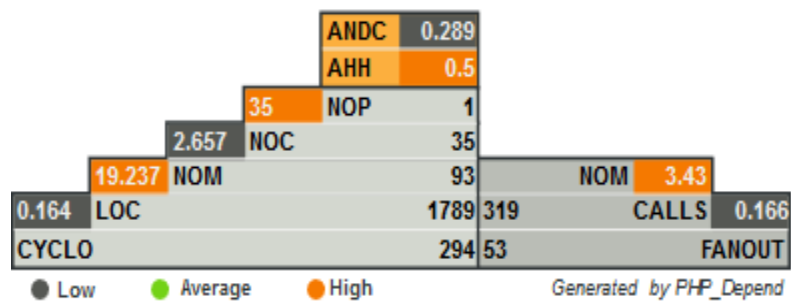
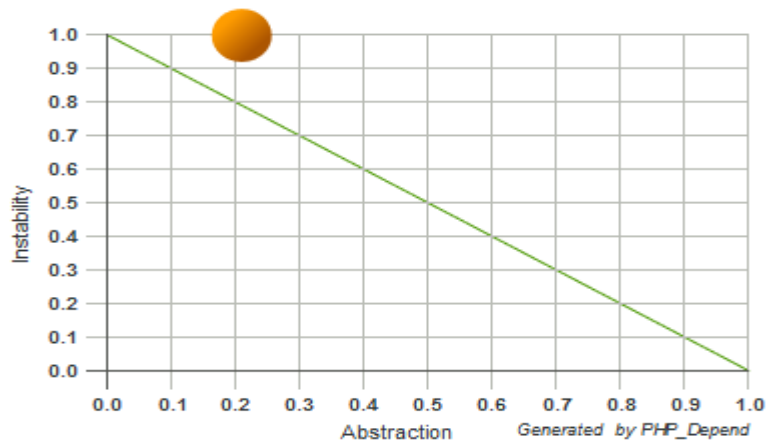


Figura 20. Gráfico piramidal (Módulos HSF e HSI). (Elaboración propia)

El siguiente gráfico de Abstracción/Inestabilidad, hace referencia en su eje vertical al acoplamiento entre paquetes o clases, en el cual mientras más cercano a uno sea el valor obtenido, mayor inestabilidad tendrá el paquete, indicando una mayor dependencia a otros. En su eje horizontal muestra la relación entre clases abstractas y clases concretas donde el valor uno muestra una mayor cantidad de clases e interfaces abstractas.





**Figura 21. Gráfico de Abstracción/Inestabilidad (Módulos HSF e HSI). (Elaboración propia)**

Después de un análisis realizado a los gráficos obtenidos se puede concluir que:

Los módulos HSF e HSI poseen un alto nivel de herencia (AHH), las llamadas realizadas por cantidad de métodos (CALLS/NOM) junto al promedio de líneas de código por método (LOC/NOM) sobrepasan el máximo permisible.

### 3.3.2 Evaluación del objetivo de la investigación

Los objetivos propuestos que dieron inicio al desarrollo de esta aplicación e investigación se fueron cumpliendo mediante el transcurso de cada una de las fases de desarrollo del proyecto, se logró completar el desarrollo de un software que permite recolectar la información de los pacientes que asisten a las consultas médicas no solo de hospitales sino de toda entidad a lo largo del país, este software cumple con cada una de las expectativas y requisitos propuestos e idealizados por parte del cliente y el equipo de desarrollo.



### **3.4 Conclusiones del capítulo**

Con el desarrollo del capítulo anterior se arribó a las siguientes conclusiones:

- Durante la etapa de implementación el uso de los estándares de codificación definidos permitió desarrollar un código reutilizable, comprendido por el equipo de desarrollo haciendo más simple la puesta a punto de la solución propuesta.
- La descripción de las pruebas utilizadas para asegurar la calidad del software, permitió obtener resultados satisfactorios, asegurando que el sistema implementado no contiene errores y tiene la aceptación requerida.



### **Conclusiones generales**

- El estudio de los principales sistemas de gestión hospitalaria existentes en la actualidad determinó que su implantación es poco factible por estar dirigido a fines específicos y otros no pueden ser adquiridos, sin embargo, permitió entender los principales procesos, módulos y funcionalidades que estos deben cumplir.
- La obtención de los requerimientos funcionales y no funcionales obtenidos a partir del proceso de identificación de los requisitos y los artefactos generados, constituyeron elementos claves en la construcción de la propuesta de solución.
- La realización de las pruebas de software, permitió el correcto funcionamiento de los módulos HSF e HSI.
- Los módulos desarrollados permiten facilitar la gestión de la información de las historias clínicas de salud de los ciudadanos a partir de:
  - La realización de un registro de las historias de los pacientes que visitan una entidad médica.
  - Genera una vista de las últimas familias atendidas por un especialista.



### **Recomendaciones**

Se recomienda al MINSAP el despliegue del Sistema de Vigilancia de Salud con la integración de los módulos HSF e HSI para la Dirección de Registros Médicos del MINSAP en todas las entidades médicas de primer orden de la nación, para lograr una mayor retroalimentación de la información de los pacientes en todo momento.

Para el desarrollo de futuras versiones de los módulos HSF e HSI para la Dirección de Registros Médicos del MINSAP se recomienda al centro STD de la empresa Xetid:

- La inclusión de un cuadro resumen de almacenamiento de las historias clínicas generadas por los hospitales.



### Referencias Bibliográficas:

1. *¿Qué es un framework?* (s.f.). Recuperado el 3 de enero de 2018, de <https://www.orix.es/que-es-un-framework-y-para-que-se-utiliza>
2. *ApacheServer*. (6 de febrero de 2018). Obtenido de Versión 2.4 de la documentación del Servidor de HTTP Apache - Servidor Apache HTTP Versión 2.4.: <http://httpd.apache.org/docs/current/>.
3. AULBACH, E., WINSTEAD, J., TORBEN WILSON, L., LERDORF, R., ZMIEVSKI, A., & AHTO, J. (s.f.). *Manual de PHP*. S.l.: s.n.
4. *BPMN vs UML*. (21 de noviembre de 2017). Obtenido de <https://articulosit.files.wordpress.com/2014/01/bpm-vs-uml.pdf>
5. *Doctrine*. (23 de noviembre de 2017). Obtenido de <http://www.doctrine-project.org/>.
6. *DriCloud*. (6 de febrero de 2018). Obtenido de <https://dricloud.com/>
7. *Framework*. (3 de enero de 2018). Obtenido de <https://www.orix.es/que-es-un-framework-y-para-que-se-utiliza>
8. *Galén*. (12 de febrero de 2018). Obtenido de Sistema de Información para la gestión y coordinación de procesos en un servicio de Oncología.: <http://www.lcc.uma.es/~lfranco/B12-Ribelles+Jerez++10.pdf>.
9. *Gestores de BD*. (3 de enero de 2018). Obtenido de <https://es.slideshare.net/nipas/10-sgbd>.
10. *Herramienta de Modelado*. (21 de noviembre de 2017). Obtenido de [http://www.alegsa.com.ar/Dic/herramienta\\_de\\_modelado.php](http://www.alegsa.com.ar/Dic/herramienta_de_modelado.php).
11. *Hsitoria de Salud Individual*. (6 de febrero de 2018). Obtenido de <https://es.scribd.com/document/88004267/Historia-de-Salud-Individual>



12. *HTML 5*. (22 de noviembre de 2017). Obtenido de HTML5 - HTML | MDN:  
<https://developer.mozilla.org/es/docs/HTML/HTML5>.
13. *Ingeniería del software*. (8 de febrero de 2018). Obtenido de  
<http://cotana.informatica.edu.bo/downloads/Id-Ingenieria.de.software.enfoque.practico.7ed.Pressman.PDF>
14. *JavaScript*. (22 de noviembre de 2017). Obtenido de Documentación web de MDN:  
[https://developer.mozilla.org/es/docs/Web/JavaScript/Acerca\\_de\\_JavaScript](https://developer.mozilla.org/es/docs/Web/JavaScript/Acerca_de_JavaScript).
15. *Medicina General Integral*. (6 de febrero de 2018). Obtenido de  
<http://sparraparramecubana.blogspot.com/2011/04/medicina-general-integral.html>
16. *MediCloud*. (6 de febrero de 2018). Obtenido de <https://medicloud.me/>
17. *NCBI Bookshelf*. (6 de febrero de 2018). Obtenido de ¿Qué es el historial de salud familiar? Y ¿Cómo me afecta? - ¿Cómo hablo con mi familia sobre Gaucher?:  
<https://www.ncbi.nlm.nih.gov/books/NBK115576/>
18. *Netbeans*. (3 de enero de 2018). Obtenido de <https://netbeans.org/>.
19. *PGADMIN*. (24 de noviembre de 2017). Obtenido de PgAdmin PostgreSQL Tools:  
<http://www.pgadmin.org/translation/>
20. *PMOINFORMATICA*. (6 de febrero de 2018). Obtenido de PMOINFORMATICA. 7 Técnicas de levantamiento de requerimientos software.: <http://www.pmoinformatica.com/2016/08/tecnicas-levantamiento-requerimientos.html>.
21. *PostgreSQL*. (7 de febrero de 2018). Obtenido de <https://www.postgresql.org>
22. *Prodesoft*. (20 de noviembre de 2017). Obtenido de Prodesoft | Intranet Corporativa XETID.:  
<https://intranet.xetid.cu/content/prodesoft>.



23. *Servidor de aplicaciones*. (25 de noviembre de 2017). Obtenido de <http://searchdatacenter.techtarget.com/es/definicion/Servidor-de-aplicaciones>.
24. *SML*. (6 de febrero de 2018). Obtenido de Sistema Médico en Línea: <http://smlmedico.com/>
25. *UML*. (21 de noviembre de 2017). Obtenido de El Lenguaje Unificado de Modelado: <https://campusvirtual.ull.es/ocw/mod/resource/view.php?id=8373>.
26. *Visual Paradigm*. (21 de noviembre de 2017). Obtenido de <https://www.visual-paradigm.com/>.
27. *XAVIA - ucisis*. (1 de marzo de 2018). Obtenido de <https://www.felti.org/sites/default/files/archivos/postulacion/ucisis.pdf>.
28. *ZendFramework*. (23 de noviembre de 2017). Obtenido de <http://www.maestrosdelweb.com/guia-zend/>.
29. *Zeolides*. (20 de noviembre de 2017). Obtenido de Zeolides - Xetid.: <https://www.xetid.cu/producto/33>.



### Bibliografía consultada

1. APACHE FOUNDATION, 2017. Welcome! - The Apache HTTP Server Project. [en línea]. [Consulta: 21 noviembre 2017]. Disponible en: <http://httpd.apache.org/>.
2. APLICACIONES WEB, 2008. Arquitectura cliente-servidor. [en línea]. 2018 2008. Disponible en: <https://sites.google.com/site/4appweb/tarea/2-4-arquitectura-cliente-servidor>.
3. ARMANDO SUÁREZ CUETO, BALLESTER, E.G., PATRICIO MARTÍNEZ-BARCO y PALOMA MOREDA, 2006. *Apuntes de Bases de Datos* [en línea]. S.I.: Universidad de Alicante. Departamento de Lenguajes y Sistemas Informáticos. Disponible en: <http://hdl.handle.net/10045/2990>.
4. BENCHMARK CCM y CREATIVE COMMONS, 2016. Introducción a las Bases de Datos. [en línea]. marzo 2016. Disponible en: <http://es.ccm.net/contents/66-introduccion-bases-de-datos>.
5. CAMILO E. SUÁREZ REBAZA, [sin fecha]. *Desarrollo de Software Orientado a Objetos* [en línea]. S.I.: s.n. Disponible en: [http://biblioteca.uns.edu.pe/saladocentes/archivoz/publicacionez/Cuaderno\\_Electronico\\_\\_\\_Desarrollo\\_de\\_Software\\_Orientado\\_a\\_Objeto.pdf](http://biblioteca.uns.edu.pe/saladocentes/archivoz/publicacionez/Cuaderno_Electronico___Desarrollo_de_Software_Orientado_a_Objeto.pdf).
6. CARLOS A. GUERRERO, JOHANNA M SUÁREZ y LUZ E GU, 2013. Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web. . 2013.
7. CÉSAR ARTURO GUERRA y BUZZ SG, 2007. Obtención de Requerimientos. Técnicas y Estrategia. ,
8. CLUBENSAYOS, 2014. *Introducción a la Computación. Historias de los lenguajes de computación* [en línea]. S.I.: ClubEnsayos.com. Disponible en: <https://www.clubensayos.com/Tecnolog%C3%ADa/Introducci%C3%B3n-A-La-Computaci%C3%B3n/1715803.html>.
9. DANIEL ENRIQUE PÉREZ ROJAS, 2014. *Manual de Usuario de JMeter para la realización de pruebas de rendimiento a las Aplicaciones Web*. S.I.: Centro de Calidad, Estánderes y Seguridad. XETID.





10. DELVIS ECHEVERRÍA PEREZ y ARIANNIS ABELLA PAUMIER, 2014. *Testing como Práctica para Evaluar la Eficiencia en Aplicaciones Web*. S.l.: Revista Latinoamericana de Ingeniería de Software.
11. Diccionario de la lengua española. [en línea], 2015. Disponible en: <http://www.wordreference.com/definicion/gesti%C3%B3n>.
12. FRANK BUSCHMANN, REGINE MEUNIER, HANS ROHNERT, PETER SOMMERLAD y MICHAEL STAL, [sin fecha]. *Pattern-oriented software architecture (POSA)*. S.l.: s.n.
13. IDAILIS GUTIÉRREZ REYES, 2015. *Manual de instalación, configuración y administración de Jenkins*. S.l.: Centro de Calidad, Estánderes y Seguridad. XETID.
14. LARMAN CRAIG, 1999. *UML y Patrones*. New York : PRENTICE HALL: s.n. ISBN 970-17-0261-1.
15. LIBROSWEB, 2006. *Introducción al JavaScript* [en línea]. LibrosWeb.es: s.n. Disponible en: <http://librosweb.es/libro/javascript>.
16. LISSA CURBELO OLIVA, LAURA S. ORTEGA RETURETA, YOANNA COLUMBIÉ CISNERO y YUDELKY GONZÁLEZ MILÁN, 2017. Prodesoft v1.5. [en línea]. [Consulta: 20 noviembre 2017]. Disponible en: <https://intranet.xetid.cu/content/prodesoft>.
17. MARÍA JESUS LAMARCA LAPUENTE, 2013. *Hipertexto: El nuevo concepto de documento en la cultura de la imagen*. Universidad Complutense de Madrid: s.n.
18. Mastermagazine. [en línea], 2012. Disponible en: <http://mastermagazine.info/termino/5234.php>.
19. MICHAEL BLAHA, 2010. *Patterns of data modeling*. Segunda Edición. S.l.: CRC Press Taylor & Francis Group. ISBN 1-4398-1989-0.
20. MICROSOFT, 2018a. *La Guía de Arquitectura Versión 2.0a de Microsoft* [en línea]. S.l.: s.n. Disponible en: url <http://carlosreynoso.com.ar/archivos/arquitectura/Estilos.PDF>.
21. MICROSOFT, 2018b. Revisiones de código y estándares de codificación. [en línea]. 2018. Disponible en: <https://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>.



22. MICROSOFT, [sin fecha]. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. S.l.: s.n.
23. ORACLE CORPORATION, 2016. The NetBeans Platform. [en línea]. 2016. Disponible en: <https://netbeans.org/dwnloads>.
24. PMOINFORMATICA, 2016. *7 Técnicas de levantamiento de requerimientos de software* [en línea]. S.l.: s.n. Disponible en: <http://www.pmoinformatica.com/2016/08/tecnicas-levantamiento-requerimientos.html>.
25. PMOINFORMATICA, 2017. Pruebas de software. [en línea]. [Consulta: 16 noviembre 2017]. Disponible en: <http://www.pmoinformatica.com/p/pruebas-de-software.html>.
26. RAFAEL MARTÍNEZ, 2015. *Portal en español sobre PostgreSQL* [en línea]. PostgreSQL.es: PostgreSQL. Disponible en: <http://www.postgresql.org.es>.
27. ROBERT PRESSMAN, 2010. *Software Engineering. A practitioner's approach*. McGraw-Hill, 7ma edición,. S.l.: s.n.
28. ROBERTO H. SAMPIERI, CARLOS FERNÁNDEZ COLLADO y PILAR BAPTISTA LUCIO, 2014. *Metodología de la Investigación*. 6ta Edición. México: Mc Craw Hill Education. ISBN 978-1-4562-2396-0.
29. RS PRESSMAN y JM TROYA, 2013. *Ingeniería de Software un enfoque práctico*. Séptima edición. Madrid: s.n. ISBN 978-607-15-0314-5.
30. STIG SAETHER BAKKEN, ALEXANDER AULBACH, EGON SCHMID, JIM WINSTEAD y LARS TORBEN WILSON, 2001. *PHP Manual* [en línea]. My PHP.net: s.n. Disponible en: <https://secure.php.net/manual/es/preface.php>.
31. VISUAL PARADIGM, 2010. *Visual Paradigm. ULM tool, business process modeler and database designer for software development team*. [en línea]. S.l.: s.n. Disponible en: <http://www.visual-paradigm.com>.
32. XETID, 2015. Manual de usuario del framework ZEOLIDES v2.2.0. . 2015.



### Glosario de términos:

**Hardware:** conjuntos de componentes que integran la parte física de una computadora.

**Software:** equipo lógico o soporte lógico de un sistema informático, que comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos que son llamados hardware.

**Base de Datos (BD):** colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite. Una base de datos es un sistema de archivos electrónico.

**SGBD:** un Sistema Gestor de Base de Datos tiene la función de controlar la entrada y salida de los datos de una base de datos.

**Web:** esta palabra es utilizada para denominar uno de los servicios más importantes de la red Internet.

**HTTP:** Hypertext Transfer Protocol que en español significa Protocolo de Transferencia de Hipertexto, es utilizado para establecer una conexión web a través de un navegador.

**CYCLO:** Promedio de la complejidad ciclomática del código.

**NOM:** Se reflejan las clases cuyo número de métodos (nom) sea de 10 o mayor.

**NOC:** Reflejan los paquetes que posean los números de clases de 26 o mayor.

**CALLS:** Número de llamadas a métodos y funciones.

**ANDC:** Número de clases derivadas.

**AHH:** Altura promedio de jerarquía desde la clase raíz hasta las subclases que dependen de ellas.



## Anexos

### Anexo 1. Entrevista realizada al médico especialista

**Nombre:** Osmundo Almaguel Speck

**Especialidad:** Medicina General Integral (MGI).

**Área de trabajo:** Policlínico-Hospital de la universidad de las ciencias informáticas (UCI).

**Necesidad desde su punto de vista:** virtualización de las historias de salud individual e historias de salud familiar.

**Requisitos:**

- Gestionar las Historias de Salud Individual.
- Gestionar las Historias de Salud Familiar.
- ...|

Figura 22. Entrevista realizada al médico especialista de la universidad. (Elaboración propia)





### Anexo 3. Descripción de requisitos

<b>&lt;&lt;requisito&gt;&gt; Buscar ciudadano</b>
Código = "" Tipo = "Funcional" Conceptos = "Ciudadano" Precondiciones = "No procede" Pre-requisito = "No procede" Método de verificación = "No procede" Riesgo = "Medio"

Figura 24. Diagrama de requisitos. Buscar ciudadano. (Elaboración propia)

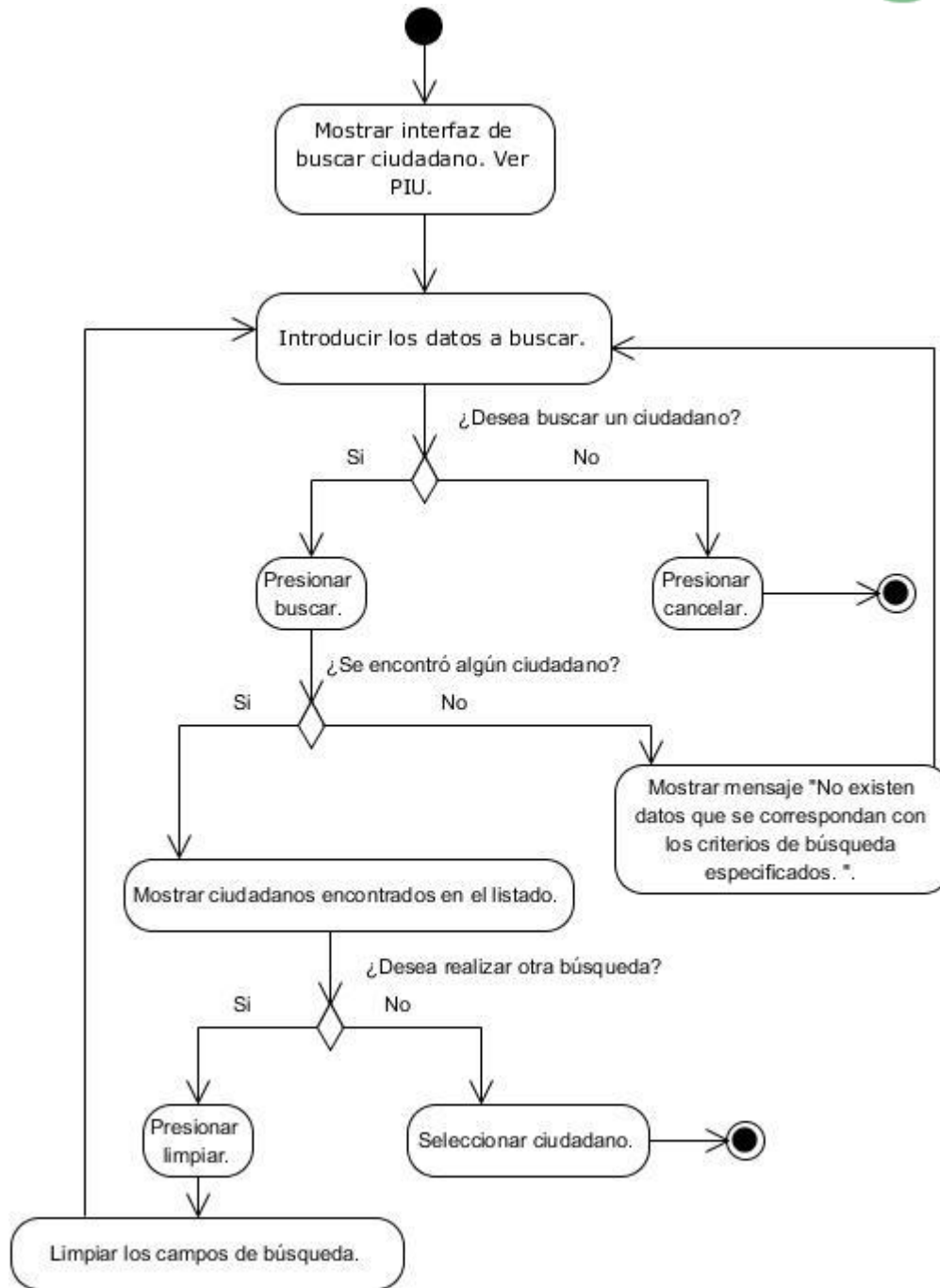


Figura 25. Diagrama de actividades. Buscar ciudadano. (Elaboración propia)

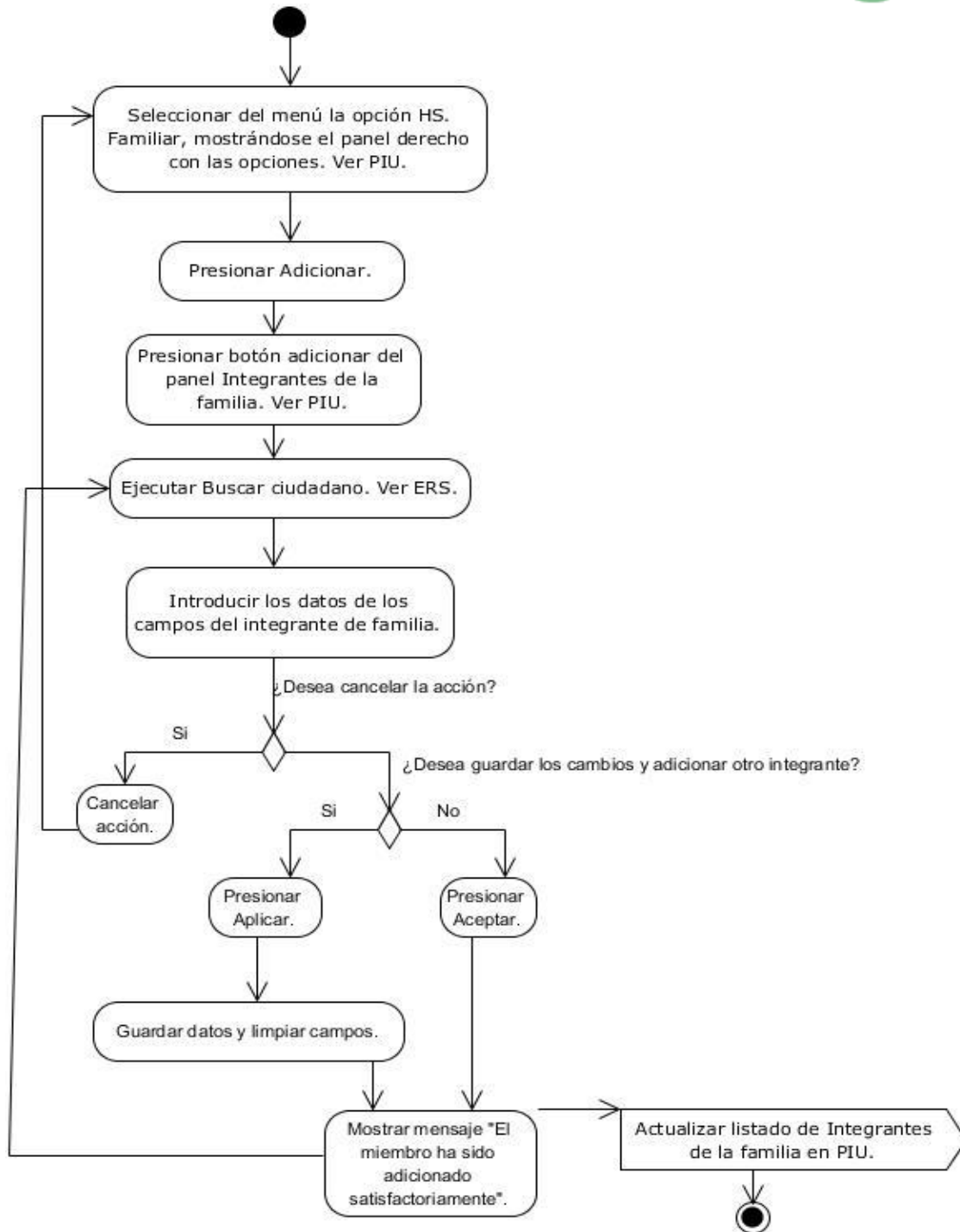


Figura 26. Diagrama de actividades. Adicionar Integrante de la familia. (Elaboración propia)



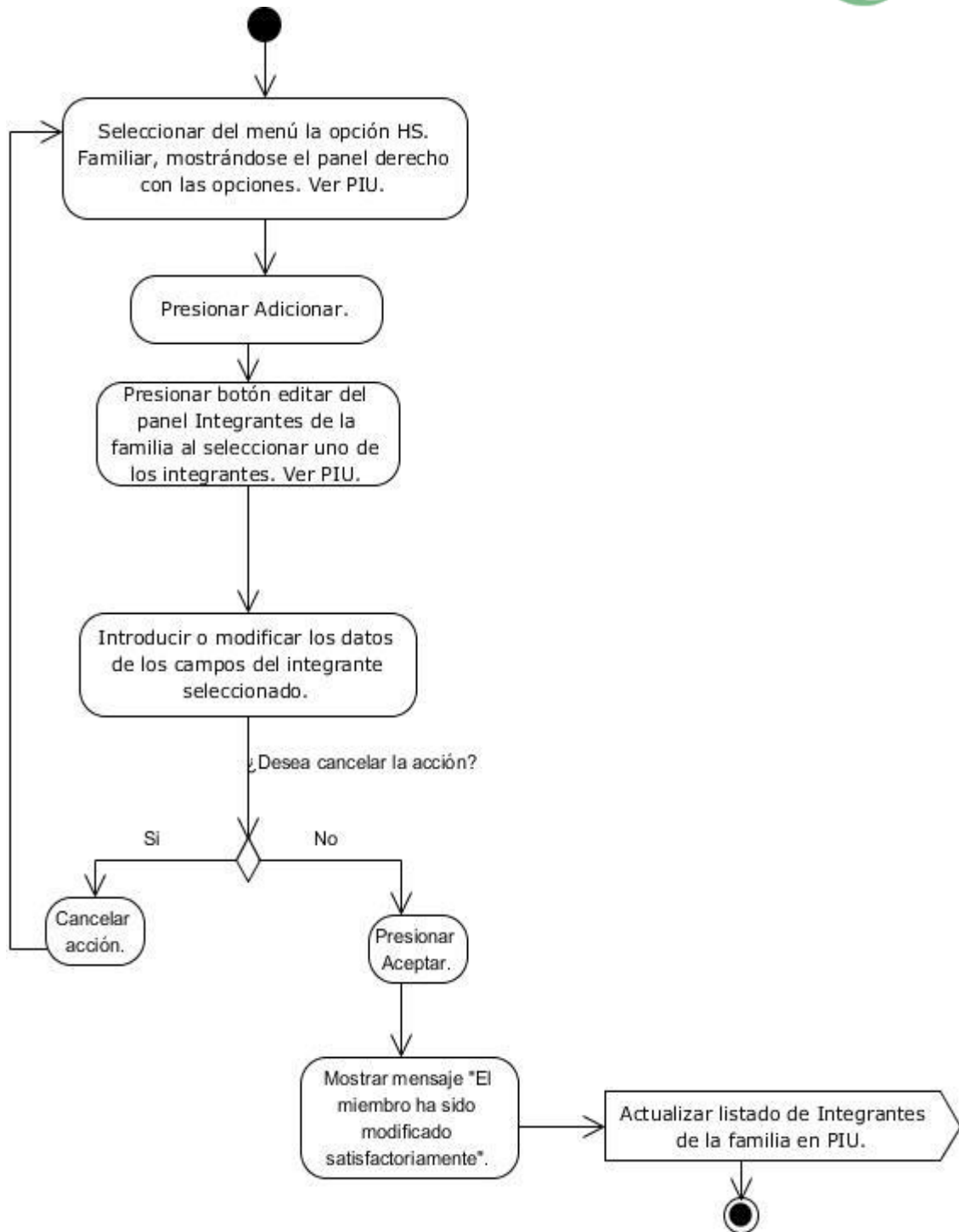
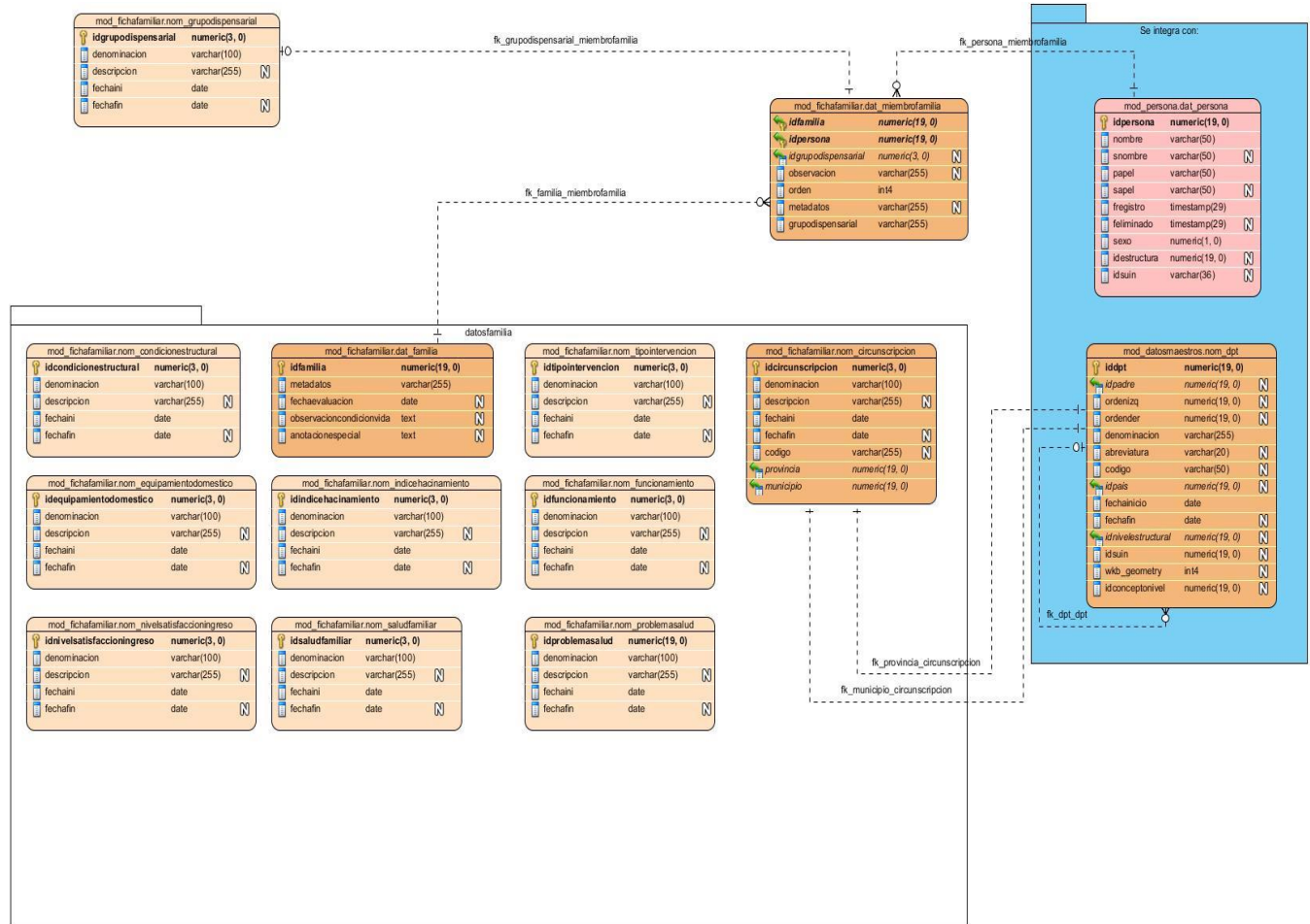


Figura 27. Diagrama de actividades. Editar integrante de la familia. (Elaboración propia)



**Anexo 4 Diagrama entidad relación**



**Figura 28. Diagrama entidad relación caso HSF. (Elaboración propia)**

Módulos Historia de Salud Individual e Historia de Salud Familiar para la Dirección de Registros Médicos del Ministerio de Salud Pública.

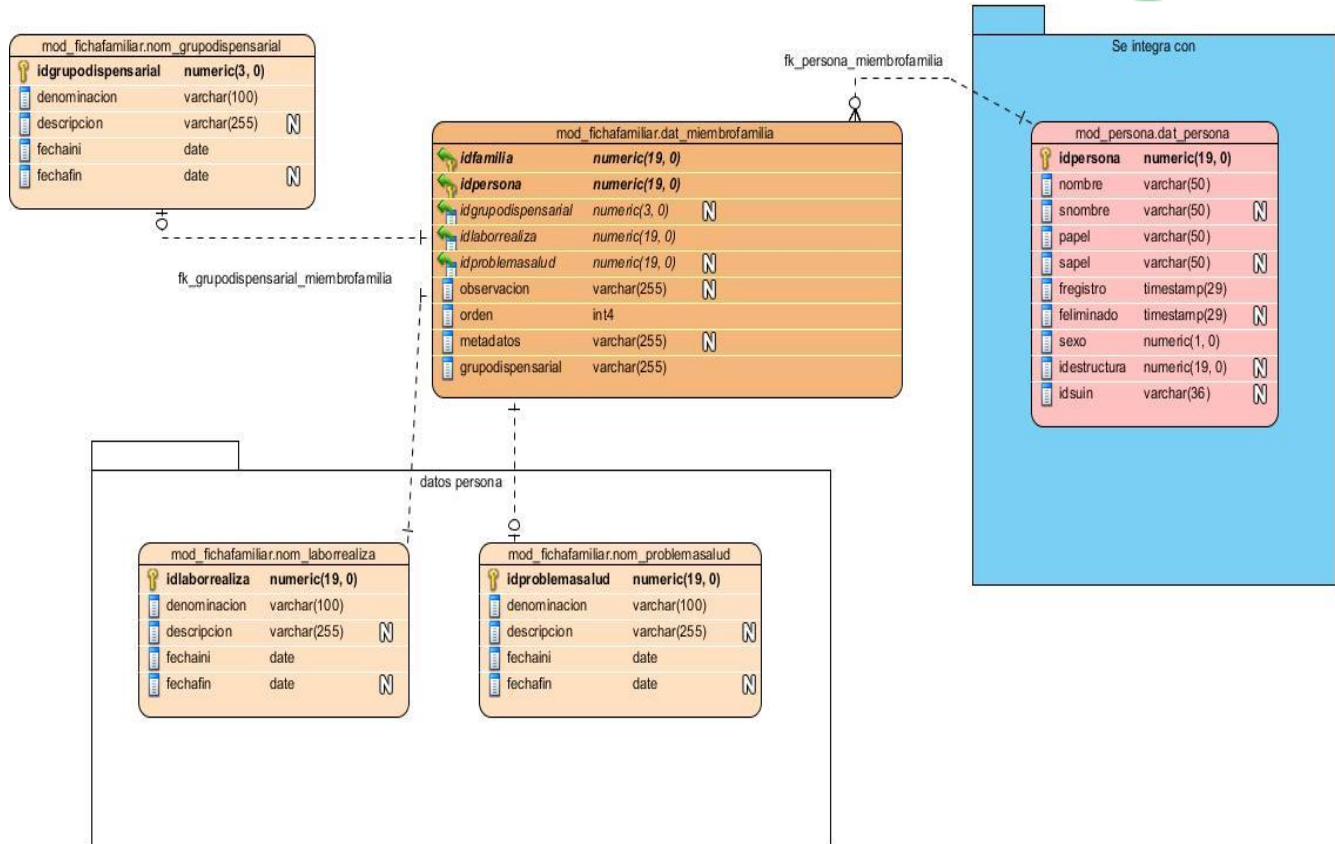


Figura 29. Diagrama entidad relación caso HSI. (Elaboración propia)



Anexo 5 Diagramas de clases del diseño con estereotipos web

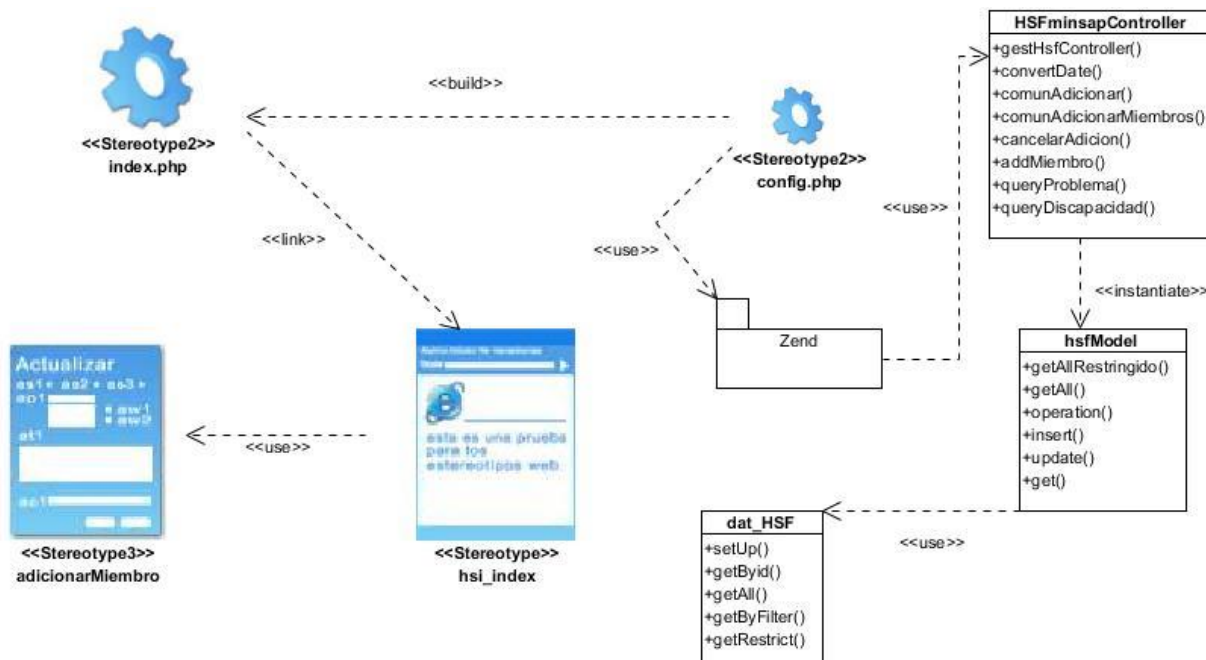


Figura 30. Diagrama de clases con estereotipos web del caso de adicionar un nuevo integrante a la familia. (Elaboración propia)

Módulos Historia de Salud Individual e Historia de Salud Familiar para la Dirección de Registros Médicos del Ministerio de Salud Pública.

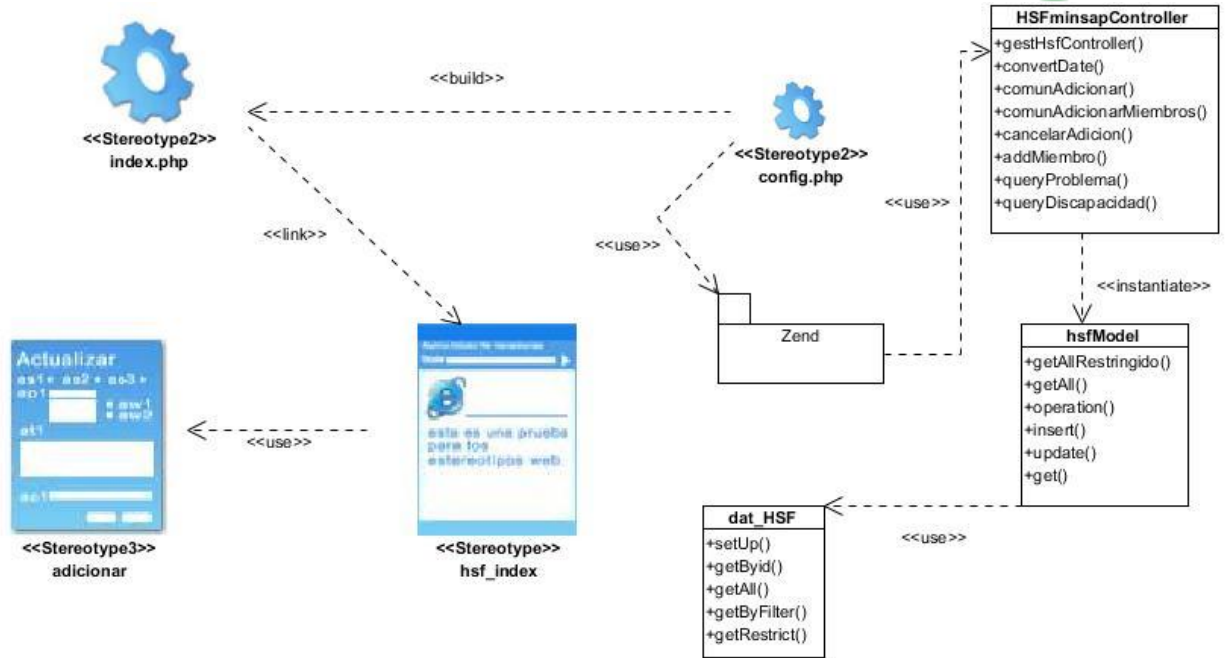


Figura 31. Diagrama de clases con estereotipos web para adicionar una HSF. (Elaboración propia)



### Anexo 6 Prototipos de interfaz de usuarios

Buscar ciudadano

CI	Nombre	Segundo apellido	Primer apellido	Segundo apellido	Buscar	Limpiar
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>		

	CI	Nombre(s) y Apellidos	Dirección
<input type="checkbox"/>			

Página 1 Cantidad por página 5 1-5 de 8

Figura 32. PIU Buscar ciudadano. (Elaboración propia)

carloscav  
DPS CIEGO DE ÁVILA  
DPS\_JDV

[Inicio](#)  
[HS. Familiar](#)

Historiasde salud familiar

HSF	Integrantes	Evaluación	Funcionamiento	Intervención	Fecha actualizada
<input type="checkbox"/>					

Detalles Editar Eliminar

Página 1 Cantidad por página 5 1-5 de 8

Figura 33. PIU Listar HSF. (Elaboración propia)

Módulos Historia de Salud Individual e Historia de Salud Familiar para la Dirección de Registros Médicos del Ministerio de Salud Pública.



Se muestra la interfaz para filtrar una persona:

Datos del ciudadano				
<input type="text" value="Nombre(s) y apellidos"/>				
Nombre y apellidos	Carné de identidad	Edad	Sexo	Color de piel
Nacionalidad	Pasaporte	Teléfono	Dirección	

Luego de filtrada la persona:

Problema(s) de salud	Factor(es) de riesgo
<input type="text" value="Problema de salud"/>	<input type="text" value="Factor(es) de riesgo"/>
Discapacidad(es)	Alergia(s)
<input type="text" value="Discapacidad(es)"/>	<input type="text" value="Alergia(s)"/>
Nivel educacional	Labor que realiza
<input type="text" value="Nivel educaci..."/>	<input type="text" value="Labor que realiza"/>
Observaciones:	
<input type="text"/>	
<input type="button" value="Cancelar"/> <input type="button" value="Aceptar"/>	

**Nivel educacional:**

- Primaria - sin terminar.
- Primaria terminada.
- Secundaria terminada
- Técnico medio terminado (especificar la especialidad)
- Pre- universitario terminado.
- Universitario terminado (especificar la especialidad)
- Círculo o jardín infantil.
- Vías no formales.
- Pre- escolar no institucionalizado
- No escolarizado apto para el estudio.
- No escolarizado no apto para el estudio.

**Factor(es) de riesgo:**

- Tabaquismo
- Alcoholismo

Figura 34. PIU Modificar y Adicionar Integrante de la familia. (Elaboración propia)