

Universidad de las Ciencias Informáticas

Facultad 6



***Título: Aplicación web para el estudio y gestión de la información de árboles
genealógicos en el sistema alasARBOGEN 2.0***

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

Autor(es): Yoan Manuel Pérez Piñero

Rosnel Venero Acosta

Tutor(es): Ing. Reynaldo Alvarez Luna

Ing. Yunier Santana Aldana

Ciudad de la Habana, Cuba

“Año 52 de la Revolución”

Junio, 2010

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Yoan Manuel Pérez Piñero

Firma del Autor

Rosnel Venero Acosta

Firma del Autor

Ing. Reynaldo Alvarez Luna

Firma del Tutor

Ing. Yunier Santana Aldana

Firma del Tutor

AGRADECIMIENTOS

Agradecemos a la Revolución Cubana y a nuestro Comandante en Jefe, por darnos la oportunidad de estudiar en una Institución de esta magnitud. A todos los profesores que han aportado su grano de arena en nuestra formación profesional. A los compañeros de aula, de apartamento, de los momentos buenos y malos, a todos con los que hemos podido contar a lo largo de la carrera y la vida, muchas gracias por compartir espacios que no se olvidarán jamás. A nuestros tutores. Infinitos agradecimientos a nuestros familiares, que se han comportado como tal, brindando apoyo incondicional en todas las situaciones presentadas en nuestro trayecto estudiantil.

Palabras no pueden describir el sentimiento de gratitud que sentimos hacia nuestros padres, guías inspiradoras de nuestras acciones diarias, ejemplos de dignidad, humildad y honestidad, fuentes de amor y cariño; a ellos agradecemos, por confiar plenamente en nosotros, de todo corazón MUCHAS GRACIAS.

Rosnel y Yoan

DEDICATORIA

Casi 20 años de estudio y sacrificios se resumen en este trabajo, el cual quiero dedicar a todas las personas que han hecho posible que este sueño se convierta en realidad. A la Revolución Cubana y nuestro Comandante en Jefe por permitir mi superación en cada una de las instituciones por las que he transitado, así como a todos los profesores que han contribuido en mi formación revolucionaria y profesional.

A los “muchachos” del IPVCE, que han estado siempre presentes en momentos difíciles y de alegría, a Omar, Falero, Daniel, Rafa, Cristian, Talancón, Noel, William, Alejandro, y a nuestra “hemanita mayor” Any, fuente de cariño y apoyo incondicional. A los inigualables Robert, Javier, Frank, Fito, el Tavo, Ortiz, Daniel y Yohan, hermanos de la universidad que compartieron noches de estudio y de celebraciones. A todos los que de una forma u otra extendieron su mano para ayudar, a Carlos Mora, Pascal, David, Carlos, Edel, Asencio, Yadira, Cuan, Manzano, Reinier, en fin al grupo 6506. Al piquete de tenis de mesa, por ayudarme a despejar en momentos difíciles, además se mantuvieron siempre al tanto de todo. A los tutores, Tito y Rey. A mi compañero de tesis, por su responsabilidad ante el trabajo y los sacrificios realizados para cumplir con la tarea, por ser amigo.

A mi primo Raulín, a quien la vida le negó la oportunidad de estudiar hasta este nivel. A mi hermano Abel, exhortándolo a ser cada día mejor y superarse para ser un buen profesional. A toda mi familia por llenarme de felicidad y en especial a las personas más importantes en mi mundo, mis padres Rosabel y Nelson, por educarme y llevarme siempre por el buen camino, por apoyarme en todos los aspectos de la vida, brindarme su amor, y depositar una enorme confianza en mí, que me ha dado fuerzas para luchar y salir adelante.

Rosnel

Le dedico esta tesis, primero que todo, a todas las personas que me han apoyado desde el principio de mi vida, que me han dado amor y cariño, muy especialmente a mis padres Manuel y Margarita, que han confiado siempre en mí, me han dado todo su apoyo y amor, a mi hermana Yamaira y a mi querida sobrina Elianys, que gracias a ellos estoy aquí, a Leo y Olaya, por ser tan cariñosos, a toda mi familia.

A Odalys, especial para ella, que está día a día compartiendo conmigo los momentos buenos y malos. Por ser quien es.

A los amigos de la infancia Alexander y Reinier; a Yander y Darian, con los cuales he compartido momentos muy buenos.

A Adrián Quintero (Pascal) por ser amigo y ayudarme en los momentos que lo necesité, a Andrés Lázaro, que pasó por lo mismo que yo, a Rosnel mi compañero de tesis y amigo. A Carlos (Beruco), Cuan, Manzano, Edel, Reinier, Fito, a Joan Seijo, a Jorge Carlos, a Dresky, amigos todos. A los que me ayudaron de una forma u otra en la realización de la tesis: Asencio, David, Yadira, Yisel, Yosvany, Tito y Rey. A los que han estado en el mismo grupo que yo. En fin a los que han estado y estarán. A los que tarde a tarde estaban en las canchas compartiendo y divirtiéndose conmigo. A mis amigos y amigas, del pasado, del presente y del futuro.

A todo el que ha brindado su mano para demostrar que sí existe gente buena. Se la dedico, además, a los que pusieron una piedrecita en el camino, los que me dieron fuerzas para levantarme y superarme.

Yoan

RESUMEN

A raíz de la creación del Centro Nacional de Genética Médica (CNGM), institución rectora de esta especialidad en Cuba, se comenzaron a desarrollar nuevos proyectos, con el objetivo de colaborar en la reducción del impacto de los padecimientos genéticos a través de estrategias de prevención.

La aplicación informática alasARBOGEN 1.0 es fruto de las investigaciones realizadas en la Universidad de las Ciencias Informáticas, y fue creada para mejorar la forma de representación del árbol genealógico de un individuo.

La presente investigación arrojará como resultado una aplicación web que solucionará problemas existentes en la Red Nacional de Genética Médica, como la disponibilidad de la información y la generación de reportes a partir de distintos criterios, así como el estudio de la información gestionada por el sistema, surgidos a partir de la utilización del software alasARBOGEN 1.0.

PALABRAS CLAVE

alasARBOGEN, árbol genealógico, framework, genética, Java, sistema, software, Spring.

TABLA DE CONTENIDOS

AGRADECIMIENTOS II

DEDICATORIA..... III

RESUMEN..... IV

INTRODUCCIÓN..... 1

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS 5

 Introducción 5

 1.1 APLICACIONES WEB PARA LA GESTIÓN DE ÁRBOLES GENEALÓGICOS. 5

 1.2 METODOLOGÍA DE DESARROLLO DE SOFTWARE. 7

 El Proceso Unificado de Desarrollo de Software (RUP). 7

 Roles y Artefactos. 8

 1.3 LENGUAJE DE PROGRAMACIÓN. 10

 Java. 10

 1.4 TECNOLOGÍAS USADAS. 11

 Java Web Start (JWS). 11

 Framework Spring. 12

 Sistema de Seguridad Acegi para Spring 13

 Framework Hibernate. 13

 1.5 HERRAMIENTAS DE DESARROLLO DE SOFTWARE..... 14

 Entorno de desarrollo Eclipse 3.3. 14

 Visual Paradigm for UML Enterprise Edition 3.4..... 15

 Servidor web (Apache)..... 16

CONCLUSIONES..... 16

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....	17
Introducción	17
2.1 OBJETO DE ESTUDIO.....	17
2.1.1 Objetivos estratégicos de la organización.	17
2.1.2 Flujo actual de los procesos.	17
2.2 OBJETO DE AUTOMATIZACIÓN.	18
2.3 MODELO DE DOMINIO.....	18
2.4 REQUERIMIENTOS DEL SISTEMA.	19
2.4.1 Requerimientos funcionales.	19
2.4.2 Requerimientos no funcionales.	20
2.5 DEFINICIÓN DE CASOS DE USO (CU) DEL SISTEMA.	23
2.5.1 Patrón de Caso de Uso utilizado.	23
2.5.2 Actores del Sistema.	24
2.5.3 Diagrama de Casos de Uso del Sistema.	24
2.5.4 Descripción de los Casos de Uso del Sistema.	25
CONCLUSIONES.....	32
CAPÍTULO 3: DISEÑO DEL SISTEMA.....	33
Introducción	33
3.1 PATRONES.....	33
3.1.1 Patrón arquitectónico "Modelo-Vista-Controlador" (MVC).....	33
3.1.2 Patrones de diseño.	35
3.3 DIAGRAMAS DE CLASES DE DISEÑO.....	39
3.4 DIAGRAMAS DE SECUENCIA.	45
3.5 DIAGRAMA DE DESPLIEGUE.....	48

Tabla de Contenidos

CONCLUSIONES.....	49
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS	50
Introducción	50
4.1 IMPLEMENTACIÓN.....	50
4.1.1 Fragmentos de Código.	51
4.1.2 Diagramas de Componentes.....	52
4.2 PRUEBAS	55
4.2.1 Plan de Pruebas.	55
4.2.2 Diseño de Casos de Pruebas de Caja Negra.	57
4.2.3 No conformidades detectadas.....	62
CONCLUSIONES.....	64
CONCLUSIONES	65
RECOMENDACIONES	66
BIBLIOGRAFÍA.....	67
REFERENCIAS BIBLIOGRÁFICAS	69
ANEXOS.....	71

INTRODUCCIÓN

Con el significativo desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) se han abierto nuevos e interesantes canales, tanto para la entrega de los servicios a la sociedad, como para mejorar la calidad y oportunidad de la información a la que los ciudadanos pueden acceder, incidiendo en casi todos los aspectos de nuestra vida: el acceso al mercado de trabajo, la sanidad, la gestión económica, el diseño industrial y artístico, la comunicación, la información, la organización de las empresas e instituciones, sus métodos y actividades, la forma de comunicación interpersonal, la educación, entre otros. Su gran impacto en todos los ámbitos hace cada vez más difícil que el hombre pueda actuar eficientemente prescindiendo de ellas.

Sus principales aportes a las actividades humanas se concretan en funciones que facilitan la realización de los diferentes trabajos, al requerir cada uno de información, un determinado procesamiento de datos y comunicación con otras personas. (GRAELLS, 2000)

En Cuba, potenciar el uso masivo de las TIC a favor del desarrollo de la economía nacional, la sociedad y el servicio al ciudadano es una de las políticas de la informatización de la sociedad. Para apoyar esta tendencia se creó el Ministerio de la Informática y las Comunicaciones (MIC), organismo que, entre otras funciones, regula y controla la política y las estrategias para la evolución, producción, comercialización y utilización de los servicios y tecnologías de la Informática y las Comunicaciones.(COMUNICACIONES.)

Actualmente el Ministerio de Salud Pública (MINSAP) ha convocando a un grupo de instituciones propias del MIC y de otros organismos de la administración central del estado, para definir de conjunto la estrategia a desarrollar, con el objetivo de informatizar los sistemas de trabajo en este sector.

La Universidad de las Ciencias Informáticas (UCI) es uno de los centros rectores en esta integración; a través de ella comienzan a girar las estrategias de desarrollo de la Industria Cubana del Software.

Como resultado, la UCI trabaja en importantes proyectos asociados a la medicina, respaldando el esfuerzo del Sistema de Salud Cubano. Entre las organizaciones insertadas en la informatización se encuentra el Centro Nacional de Genética Médica (CNGM), creado el 5 de agosto de 2003 y de referencia

nacional para el Programa Cubano de diagnóstico, manejo y prevención de enfermedades genéticas y defectos congénitos, el mismo dirige la red de centros de salud de este tipo en el país y cumple tareas asistenciales, docentes e investigativas en dicho campo.

En el diagnóstico de cualquier enfermedad genética, resulta interesante el historial clínico de la familia que se va a estudiar, por lo que conocer el árbol genealógico es el primer paso para la identificación de algún riesgo.

El árbol genealógico es una representación gráfica que expone los datos de un individuo en una forma organizada y sistemática, sea en forma de árbol o tabla. Puede ser ascendiente o descendiente, exponiendo los ancestros o descendientes respectivamente. (Instituto Nacional del Cáncer)

Con el objetivo de facilitar el trabajo de los genetistas se creó en la UCI, alasARBOGEN 1.0, una aplicación informática, primera de su tipo en el país, para representar árboles genealógicos.

El despliegue de alasARBOGEN 1.0 por los diferentes centros de genética del país fue una tarea difícil; no se pudo utilizar la red de salud para realizar esta labor, por lo que se consumieron recursos como discos compactos de instalación por cada uno de ellos. Además, este software no brinda la posibilidad de generar reportes bajo ningún criterio, limitando así posibles estudios genéticos en la población, lo que puede dificultar la toma de decisiones futuras. La información gestionada por alasARBOGEN 1.0 no se encuentra disponible en la Red Nacional de Genética Médica, la aplicación brinda la posibilidad de guardar los datos en ficheros de texto, los cuales no podrán ser accedidos por genetistas de otros centros del país. Todos estos factores restringen la eficiencia de los servicios médicos brindados a la población cubana. Teniendo en cuenta lo anteriormente planteado surge como **problema científico** de la presente investigación:

La necesidad de disponibilidad y estudio de la información gestionada por alasARBOGEN 2.0, y se define como **objeto de estudio**: Los sistemas de gestión de información de árboles genealógicos.

El **campo de acción** que abarca el siguiente trabajo es: Aplicaciones web para el estudio y gestión de la información de árboles genealógicos; para dar solución al problema científico propuesto se define como

objetivo general: Desarrollar una aplicación web que permita la disponibilidad y el estudio de la información gestionada por las ARBOGEN 2.0.

Para dar cumplimiento al objetivo general se plantean como **objetivos específicos:**

- ✓ Identificar los requisitos de la aplicación web.
- ✓ Realizar el diseño de la aplicación web.
- ✓ Implementar la aplicación web.

Se definen como tareas científicas:

- ✓ Investigación del estado del arte para el estudio de árboles genealógicos.
- ✓ Investigación de las tendencias y tecnologías para el desarrollo de aplicaciones web.
- ✓ Identificación de requisitos funcionales y no funcionales.
- ✓ Diseño de la interfaz de usuario.
- ✓ Diseño de casos de uso.
- ✓ Diseño de clases y de subsistemas.
- ✓ Realización de los diagramas de componentes.
- ✓ Implementación de los componentes identificados.
- ✓ Realización de pruebas de caja negra.

El presente documento se ha estructurado de la siguiente manera:

Capítulo 1: Fundamentos Teóricos.

El capítulo comprende un breve estudio de las aplicaciones web destinadas a la gestión de árboles genealógicos, además se aborda acerca de la metodología, herramientas y tecnologías usadas para desarrollar la propuesta de la presente investigación.

Capítulo 2: Características del sistema.

En este capítulo se describen los problemas existentes en la Red Nacional de Genética Médica a partir de la utilización de la aplicación alasARBOGEN 1.0. Se especifican los requerimientos funcionales y no funcionales de la aplicación. Además, se definen y describen los casos de uso del sistema.

Capítulo 3: Diseño del sistema.

En el capítulo se exponen los patrones utilizados en el diseño del sistema. Además, se muestran diagramas de clases del diseño y diagramas de secuencia que servirán como base para la implementación del sistema.

Capítulo 4: Implementación y Pruebas.

Este capítulo contiene descripciones de la implementación del sistema en términos de componentes, para dar solución a los requisitos funcionales especificados anteriormente, además se aborda acerca de las pruebas realizadas al software.

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS

Introducción

El capítulo comprende un breve estudio de las aplicaciones web destinadas a la gestión de árboles genealógicos, además se aborda acerca de la metodología, herramientas y tecnologías usadas para desarrollar la aplicación web de la presente investigación.

1.1 APLICACIONES WEB PARA LA GESTIÓN DE ÁRBOLES GENEALÓGICOS.

El desarrollo alcanzado por Internet ha facilitado la creación del árbol genealógico, en la actualidad ya son más de tres las aplicaciones web que brindan al usuario la posibilidad de realizar esta acción, permitiendo de esta forma crear una gran comunidad en la que se puede invitar a amigos y familiares, para que hagan sus árboles y más tarde unirlos para ver sus coincidencias.

MiParentela.com¹. Esta aplicación ofrece un servicio en línea que facilita la creación de árboles genealógicos de manera colaborativa. La misma ofrece la opción de cargar fotografías para cada uno de los miembros de la familia, además de guardar el árbol a la computadora. Tiene un diseño vistoso, moderno y es ágil en su gestión.

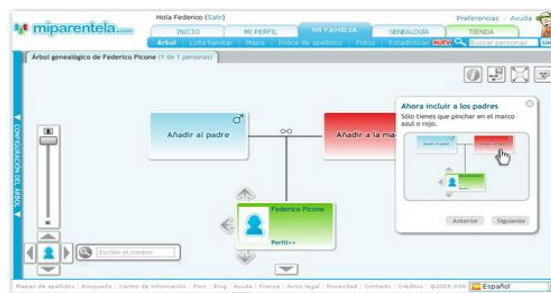


Figura 1: Aplicación Web MiParentela.com

¹ Disponible en <http://www.miparentela.com>

Capítulo 1 : Fundamentos Teóricos

Myheritage.com², permite compartir información fácilmente y en forma gratuita. Es muy sencillo y pone a disposición de los usuarios un motor de búsqueda, a través del cual pueden tratar de localizar antepasados con el objetivo de facilitar labores de investigación. Además, permite buscar árboles familiares de gente famosa.

Se utilizaron para implementar este sitio web los lenguajes PHP en su versión 5.2.9 y ASP, y se utilizó como servidor web Apache/2.2.3.



Figura 2: Aplicación Web Myheritage.com

Geni.com³, es una aplicación web “recién salida del horno” que permite al mejor estilo flash crear un árbol genealógico sin necesidad de mucho conocimiento técnico. La herramienta, basada en Flash, permite agregar información sobre la ascendencia o descendencia de una persona de manera fácil y rápida, con la fascinante particularidad de que las personas que se vayan agregando (que obviamente sigan vivas) podrán participar añadiendo información que tal vez se desconoce. El árbol genealógico de una familia es privado y solamente tendrán acceso sus miembros directos. Geni no sólo da cabida a los nombres, sino que se puede insertar videos, fotografías e incluso un calendario con los próximos eventos o cumpleaños que atañen a la familia.

² Disponible en <http://www.myheritage.es>

³ Disponible en <http://www.geni.com>

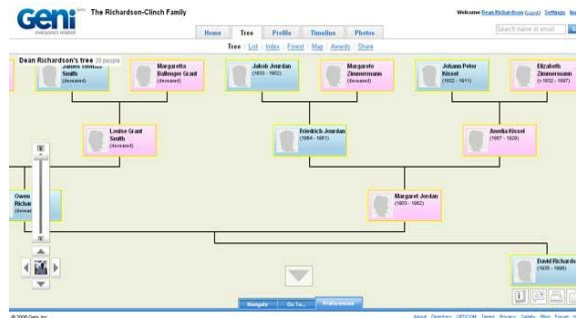


Figura 3: Aplicación Web Geni.com

Las aplicaciones web mencionados anteriormente y otras (kindo.com, genoom.com) encargadas de la creación de árboles genealógicos realizan esta labor entre otras acciones con fines netamente personales, satisfaciendo así demandas de los usuarios, muchos de los cuales acceden a ellas a manera de entretenimiento. Servicios tan importantes para la salud, específicamente en la rama de la genética médica no son ofrecidos por estas páginas web.

1.2 METODOLOGÍA DE DESARROLLO DE SOFTWARE.

Desarrollar un buen software depende de una serie de actividades y etapas, donde elegir la mejor metodología para un equipo de desarrollo en un determinado proyecto es trascendental para el éxito del producto.

Una metodología de desarrollo de software es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software. El papel predominante de las metodologías de desarrollo de software es sin duda esencial en un proyecto y en el paso inicial del mismo, éstas deben encajar en el equipo, guiar y organizar actividades que conlleven a las metas trazadas en el grupo. (ROBERTH G. FIGUEROA, 2008)

El Proceso Unificado de Desarrollo de Software (RUP).

RUP es un proceso para el desarrollo de un proyecto de software que define claramente quién, cómo, cuándo y qué debe hacerse en el proyecto, con tres características esenciales, está **dirigido por los**

Capítulo 1 : Fundamentos Teóricos

casos de uso: que orientan el proyecto a la importancia para el usuario y lo que éste quiere, está **centrado en la arquitectura:** que relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en qué orden, y es **iterativo e incremental:** dividiéndose el proyecto en mini proyectos donde los casos de uso y la arquitectura cumplen sus objetivos de manera más depurada. (IVAR JACOBSON, 2000)

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

En la investigación se decide utilizar RUP por constituir una metodología que provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga los requerimientos de los usuarios finales y además realiza evaluación en cada fase, que permite cambios de objetivos, así como un seguimiento detallado de cada una de ellas.

Roles y Artefactos.

Se denomina rol al desempeño de una persona en una situación dada. Los roles son realizados por uno o varios individuos trabajando en equipo. Un individuo puede jugar más de un rol.

Los artefactos son unidades de información creadas, producidas, cambiadas o utilizadas en el proceso de desarrollo de un software.

RUP define una serie de roles a lo largo de la realización de un proyecto, en el trabajo serán desarrollados los siguientes:

Analista de Sistema: Define los alcances del sistema e identifica a los actores y casos de uso que permiten modelar completa y consistentemente el mismo. Los siguientes artefactos serán producidos por este trabajador:

Capítulo 1 : Fundamentos Teóricos

- **Modelo de Casos de Uso:** Es un modelo del sistema que contiene actores, casos de uso y sus relaciones. Proporciona la entrada fundamental para el análisis y diseño.
- **Casos de Uso del Sistema:** Representan la forma en que los actores utilizan el sistema.

Diseñador: Identifica y define las responsabilidades, operaciones, atributos y relaciones de los elementos del diseño. Además, asegura que el diseño sea compatible con la arquitectura de software. El diseñador generará los siguientes artefactos:

- **Realización de Caso de Uso:** Es una colaboración que indica cómo se realiza/ejecuta un Caso de Uso, en términos de las clases de diseño y sus objetos.
- **Clase del diseño:** Una clase de diseño es una abstracción de una clase de implementación. Los métodos se especifican en lenguaje natural o en pseudocódigo.
- **Subsistema de Diseño:** Representa la forma de manejar los artefactos del modelo de diseño en piezas más manejables. Organiza los artefactos de diseño, dígame clases del diseño, realización de casos de uso y otros subsistemas.

Implementador: Es el responsable del desarrollo y pruebas de los componentes, de acuerdo con normas adoptadas en el proyecto. Se le puede asignar la responsabilidad de implementar una parte estructural del sistema (como una clase o subsistema de aplicación). Los artefactos que serán producidos por el implementador serán descritos a continuación:

- **Diagrama de Componentes:** Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable.
- **Casos de Prueba:** El propósito de un Caso de Prueba es especificar una forma de probar el sistema, incluyendo las entradas, los resultados esperados y las condiciones bajo las que se han de probar. Deben ser desarrollados para cada caso de uso.

1.3 LENGUAJE DE PROGRAMACIÓN.

Un lenguaje de programación es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. Por lo tanto, es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo. Son herramientas que nos permiten crear programas y software. (Lenguajes de Programación., 2009)

Java.

Es un lenguaje de programación sencillo, orientado a objetos, de propósito general e independiente de la plataforma de desarrollo, cuya sintaxis es muy parecida al lenguaje C o C++. Fue creado por James Gosling, cumpliendo así con el slogan de la compañía Sun Microsystems "compilar una vez y ejecutar en cualquier parte".

El mundo Java es muy amplio y variado. Este lenguaje posibilita fácilmente la "Integración externa", como el uso de herramientas, métodos y funcionalidades desarrolladas por otros programadores, esto supone una ventaja tanto en el ámbito del desarrollo como en la repercusión final de un proyecto. (Manual de Java.)

La indiferencia a la arquitectura representa sólo una parte de la portabilidad de Java. Este lenguaje especifica los tamaños de sus tipos de datos básicos y el comportamiento de sus operadores aritméticos, de manera que los programas son iguales en todas las plataformas. (Manual de Java.)

Las características principales que nos ofrece son:

- ✓ Es robusto: Realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos ayuda a detectar errores, lo antes posible, en el ciclo de desarrollo.
- ✓ Es de arquitectura neutral: Para establecer Java como parte integral de la red, el compilador compila su código a un fichero objeto de formato independiente de la arquitectura de la máquina en que se ejecutará.

- ✓ Es dinámico: Se beneficia todo lo posible de la tecnología orientada a objetos. No intenta conectar todos los módulos que comprenden una aplicación hasta el tiempo de ejecución.
- ✓ Es seguro: El sistema de Java tiene ciertas políticas que evitan que se puedan codificar virus con este lenguaje.
- ✓ Multithreaded: Un lenguaje que soporta múltiples threads es un lenguaje que puede ejecutar diferentes líneas de código al mismo tiempo.

Se ha elegido Java como lenguaje de programación para desarrollar los objetivos planteados en la investigación. Con este lenguaje se pueden programar páginas web dinámicas, con accesos a bases de datos, utilizando XML (Lenguaje de Marcas Extensibles, por sus siglas en Inglés) con cualquier tipo de conexión de red entre cualquier sistema. Además, en la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general.

1.4 TECNOLOGÍAS USADAS.

Java Web Start (JWS).

Es una solución de distribución de aplicaciones basada en tecnología Java. Es la canalización entre Internet y el sistema que permite al usuario ejecutar y gestionar aplicaciones desde la web. (MICROSYSTEMS., 2002)

Cuando se descarga por primera vez una aplicación que utiliza esta tecnología, el software de JWS se ejecuta automáticamente y guarda la aplicación localmente, en la memoria caché del equipo. De este modo, las subsiguientes ejecuciones son prácticamente instantáneas, ya que los recursos necesarios están disponibles de forma local. Cada vez que se inicia la aplicación, el componente de software comprueba si en la sede web de la misma hay una versión nueva disponible; si es así, la descarga y la ejecuta de forma automática. Las aplicaciones ejecutadas con JWS se ejecutarán de forma predeterminada en un entorno restringido ("zona protegida") con acceso limitado a los archivos y a la red. Por tanto, la ejecución de aplicaciones mediante esta tecnología mantiene la seguridad e integridad de los sistemas. (MICROSYSTEMS., 2002)

Capítulo 1 : Fundamentos Teóricos

Con el uso de JWS el usuario no necesita pasar por un proceso traumático de descarga e instalación de la aplicación para poder ejecutarla. Únicamente tiene que pinchar un enlace en su navegador y la aplicación se descarga, se instala y se ejecuta de manera automática. Además, se encarga de crear los accesos directos correspondientes en el escritorio y menú de inicio. (MARIÑÁN, 2002)

Framework Spring.

El Framework Spring (también conocido simplemente como Spring) es un framework de código abierto de desarrollo de aplicaciones para la plataforma Java. Por su diseño, el mismo ofrece mucha libertad a los desarrolladores en este lenguaje, con soluciones muy bien documentadas y fáciles de usar para las prácticas comunes en la industria. Sus características fundamentales pueden emplearse en cualquier aplicación hecha en Java.

Una de las principales ventajas que brinda Spring es que proporciona la posibilidad de integrarse con otras herramientas o incluso otros frameworks, con el fin de obtener los beneficios que el desarrollador desea de cada una de ellas. Para esto brinda diferentes módulos que se pueden utilizar o no al desarrollar una aplicación, dependiendo de las necesidades de ésta y del usuario. Otro de los aspectos más interesantes del framework es que la complejidad de la aplicación es proporcional a la complejidad del problema que se está resolviendo. (RICO., 2006)

Ventajas de Spring: (RICO., 2006)

- Mayor Velocidad de Desarrollo.
- La Tendencia en Java: Se ha popularizado en la comunidad de programadores en Java al considerársele el sustituto del modelo de Enterprise JavaBean. Es un framework maduro con varios años en el mercado.
- Spring MVC (Modelo-Vista-Controlador) ofrece una división limpia entre Controladores, Modelos y Vistas.
- Tiene una interfaz bien definida para la capa de negocio.

Spring Framework provee amplio soporte para Hibernate. En particular, brinda implementaciones de

Objeto de Acceso a Datos (DAO) que ofrecen diversas utilidades para acceder a la sesión de Hibernate.

Sistema de Seguridad Acegi para Spring

Acegi proporciona servicios de seguridad dentro de Spring Framework y aunque no forma parte directa de él está íntimamente ligado con éste. Es uno de los mejores frameworks de seguridad existentes en Java, potente, flexible y a la vez sencillo de configurar, sin necesidad de modificar código ya existente. (HELENA LORENZO GRANIZO, 2009)

Spring brinda un módulo especial (utilizado en la aplicación web desarrollada) para realizar la integración con Acegi y poder llevar a cabo una mejora en la cuestión de seguridad del sistema, utilizando las funcionalidades que brinda, como la autenticación de usuarios, seguridad de peticiones web y en la capa de servicios, entre otros.

Framework Hibernate.

Hibernate es un motor de persistencia de código abierto. Permite mapear un modelo de clases a un modelo relacional sin imponer ningún tipo de restricción en ambos diseños, además de desarrollar clases persistentes a partir de clases comunes, incluyendo asociación, herencia, polimorfismo, composición y colecciones de objetos. Cuenta con una amplia documentación, tanto a nivel de libros publicados como disponibles gratuitamente en la web. Actualmente es el rey indiscutible de la persistencia de datos. Desde su versión 1.0, el motor no ha parado de evolucionar, incorporando todas las nuevas ideas que se iban agregando en este campo. (JUAN MEDÍN PIÑEIRO, 2006)

Se utilizó Hibernate como una abstracción de la capa de persistencia, lo que permite que el código sea independiente de la Base de Datos utilizada. Las siguientes características del framework facilitaron el desarrollo de la aplicación:

- Modelo de programación natural: Hibernate soporta el paradigma de orientación a objetos de una manera natural.
- Persistencia transparente: Permite operar proporcionando persistencia de una manera transparente para el desarrollador.

- Facilidades en consultas: Esto se debe en parte a que se realizan en un potente lenguaje de consultas orientado a objetos.(ANDALUCIA.)

1.5 HERRAMIENTAS DE DESARROLLO DE SOFTWARE.

Las herramientas de desarrollo de software (HDS) han desempeñado un importante papel en el desarrollo de aplicaciones. Actualmente se consideran herramientas basadas en computadoras que asisten el ciclo de vida de un software.

El soporte que brindan las HDS al proceso de desarrollo proporciona importantes ventajas para el equipo de trabajo. Permiten automatizar acciones bien definidas, reduciendo la carga cognitiva del ingeniero de software, además establecen métodos efectivos para almacenar y utilizar los datos, y automatizan porciones del análisis y diseño. (LORNEL A. RIVAS)

Entorno de desarrollo Eclipse 3.3.

Eclipse es un entorno de desarrollo integrado (IDE), desarrollado por International Business Machines (IBM) y que se distribuye mediante una licencia de código abierto, la EPL o Eclipse Public License. En un primer momento Eclipse era solo un IDE para Java, pero mediante un sistema de plugins flexible se ha convertido en un IDE genérico soportado por la comunidad y en la actualidad se puede utilizar para desarrollar aplicaciones en lenguajes de programación como C/C++, Java, Python, Groovy, Perl y otros, soportando las plataformas de desarrollo Java 2 Platform Standard Edition (J2SE) y Java 2 Platform Enterprise Edition (J2EE).

Dispone de un editor de texto con resaltado de sintaxis. La compilación es en tiempo real. Tiene asistentes (wizards) para creación de proyectos, clases, test y refactorización. También, a través de "plugins" libremente disponibles es posible añadir control de versiones con Subversion e integración con Hibernate. (AVANZADAS, 2007)

Visual Paradigm for UML Enterprise Edition 3.4.

Visual Paradigm es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de un software. Contribuye a la rápida construcción de aplicaciones de calidad, y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

Principales características:

- ✓ Ingeniería inversa: Código a modelo, código a diagrama.
- ✓ Editor de Detalles de Casos de Uso: Entorno todo en uno, para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.

Esta herramienta utiliza **UML** como lenguaje de modelado, mediante el cual es posible establecer una serie de requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código.

El lenguaje se centra en la representación gráfica de un sistema de software. El mismo indica cómo crear y leer los modelos. (ORALLO.)

Las funciones de UML se pueden sintetizar en:

- Visualizar: Permite expresar de una forma gráfica un sistema, de manera que otra persona lo puedan entender.
 - Especificar: Permite especificar cuáles son las características de un sistema antes de su construcción.
 - Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.
 - Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

Servidor web (Apache).

Un servidor web es un programa que sirve para atender y responder a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP o el protocolo HTTPS.

Apache es el servidor web hecho por excelencia, su robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa. Es una tecnología gratuita, que posee su código fuente abierto y se ejecuta en varios Sistemas Operativos, como Windows y UNIX (GNU, BSD, entre otros), por lo que se considera prácticamente universal. (CIBERAULA, 2010)

Tomcat

Es el servidor web y de aplicaciones del proyecto Jakarta, se dice que es servidor web ya que gestiona solicitudes y respuestas Http (incluye el servidor Apache) gracias a sus conectores Http; además es servidor de aplicaciones o contenedor de Servlets/Java Server Pages (JSP). Es desarrollado en un entorno abierto y participativo, y publicado bajo la licencia del software Apache. Además es multiplataforma, es decir, se puede usar en varios sistemas operativos.

CONCLUSIONES.

Las aplicaciones web analizadas anteriormente no están diseñadas para fines médicos. Son consideradas las más importantes en la realización de árboles genealógicos, por el nivel de aceptación de los usuarios y la calidad de los servicios que brindan. Para implementar la aplicación se utilizó Java como lenguaje de programación, Eclipse como IDE y Spring e Hibernate como frameworks. Como herramienta Case el Visual Paradigm y el lenguaje UML para el modelado.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.

Introducción

En este capítulo se describen los problemas existentes en la Red Nacional de Genética Médica a partir de la utilización de la aplicación *ARBOGEN 1.0*. Se especifican los requerimientos funcionales y no funcionales de la aplicación. Además, se definen y describen los casos de uso del sistema.

2.1 OBJETO DE ESTUDIO.

2.1.1 Objetivos estratégicos de la organización.

“La Red Nacional de Genética está integrada por 184 centros ubicados en todos los municipios y provincias del país, coordinados por el Centro Nacional de Genética Médica, que conducen el Programa nacional para el diagnóstico, manejo y prevención de enfermedades genéticas y defectos congénitos. Esta red cumple tareas asistenciales, docentes e investigativas en el campo de la genética médica y la inmunología, enfocada a solucionar los problemas de salud de carácter genético en la población cubana, con el propósito de mejorar la calidad de vida y elevar el bienestar de nuestro pueblo.” (LIEN LE SANCHEZ, 2008)

En la actualidad se brindan consultas en el CNGM, así como en los centros de genética provinciales y municipales, siendo estos últimos las instituciones primarias de atención a la población. De acuerdo con el cuadro clínico del paciente, el genetista decide realizar o no el árbol genealógico del mismo, con el objetivo de profundizar su estudio.

2.1.2 Flujo actual de los procesos.

Si el genetista decide realizarle al paciente el árbol genealógico, procede a preguntarle sus vínculos familiares y todos los datos que puedan aportar a la investigación.

Capítulo 2 : Características del Sistema

A raíz del uso de la aplicación alasARBOGEN 1.0, surgen nuevas limitaciones en cuanto a los servicios que brinda la misma. En caso de que una persona ya tenga registrado el árbol genealógico, si ésta se presenta en otro centro, entonces habrá que realizarle nuevamente dicho árbol, debido a que la información no se encuentra centralizada y disponible en la red.

Actualmente, el proceso de comunicación entre los genetistas de los distintos centros genéticos del país es casi nula, limitando así posibles intercambios de conocimientos acerca de las enfermedades o situaciones que se presentan en el estudio de los pacientes. Esta aplicación no ofrece la posibilidad de generar reportes, impidiendo así un mayor conocimiento de la situación actual de la población en este sentido.

2.2 OBJETO DE AUTOMATIZACIÓN.

El sistema permitirá cargar árboles genealógicos desde ficheros y visualizar los mismos, además de guardarlos a la base de datos. Se emitirán reportes de acuerdo a diferentes criterios y se integrará a la aplicación de escritorio a través de la tecnología Java Web Start, para su despliegue y actualización en la red de salud.

2.3 MODELO DE DOMINIO.

Un Modelo de Dominio, es una representación visual estática del entorno real del proyecto. Puede utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis, como paso previo al diseño de un sistema, ya sea de software o de otro tipo. El objetivo del modelado del dominio es comprender y describir las clases más importantes dentro del contexto del sistema, y ayudar a los usuarios, clientes, desarrolladores y otros interesados a utilizar un vocabulario común. (IVAR JACOBSON, 2000)

En el siguiente modelo de dominio (ver Figura 4) se evidencia la falta de comunicación de alasARBOGEN 1.0 con la Red Nacional de Genética Médica. Con el comienzo de la utilización de esta aplicación en los centros genéticos del país y la instalación en cada uno de ellos de un modem para establecer comunicación con Infomed, nacen necesidades con vistas a mejorar los procesos realizados por

Capítulo 2 : Características del Sistema

los especialistas, para brindar un mejor servicio a la población y contribuir a mejorar la calidad de vida de los ciudadanos cubanos.

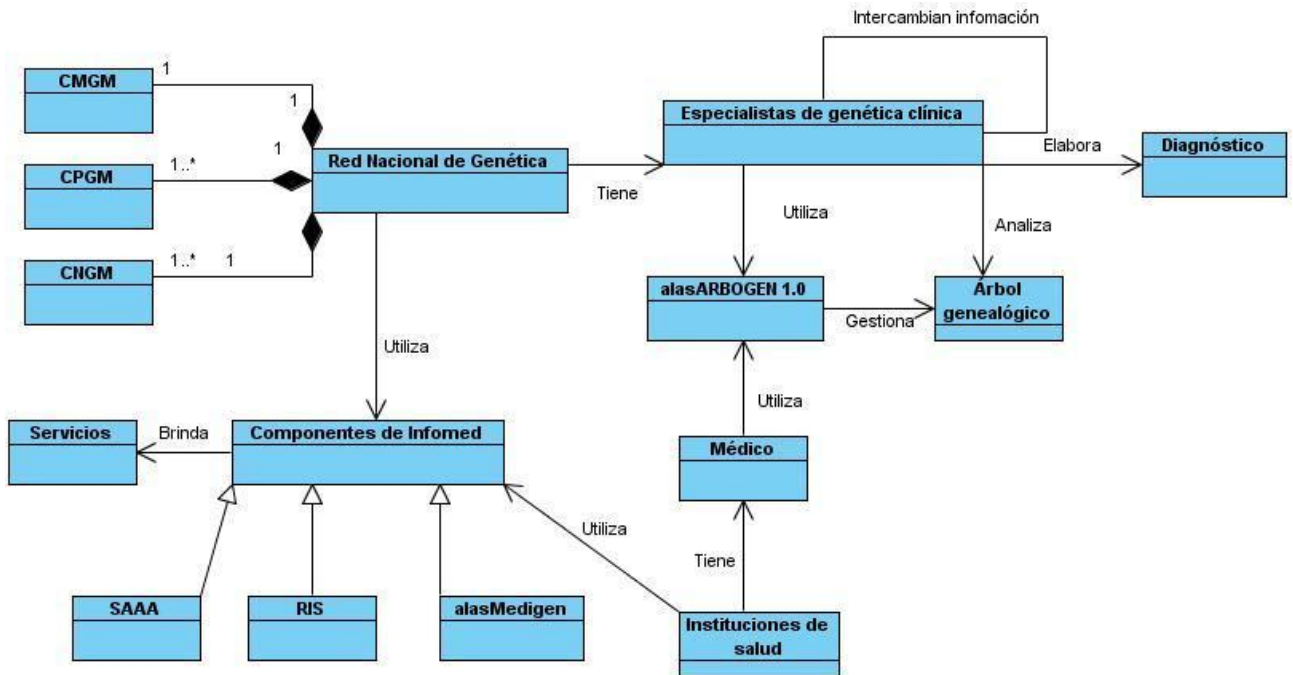


Figura 4: Modelo de Dominio.

2.4 REQUERIMIENTOS DEL SISTEMA.

Un requerimiento: es una condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema, para satisfacer un contrato, estándar, u otro documento impuesto formalmente. (IVAR JACOBSON, 2000)

2.4.1 Requerimientos funcionales.

Un requerimiento funcional define el comportamiento interno del software: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que muestran cómo los casos de uso serán llevados a la práctica. Los requerimientos funcionales se mantienen invariables sin importar con qué cualidades o propiedades se relacionen.

Capítulo 2 : Características del Sistema

A continuación se muestran los requerimientos funcionales del sistema:

- ✓ RF1- Autenticar Usuario.
- ✓ RF2- Gestionar Usuario.
 - ↳ RF2.1- Registrar usuario.
 - ↳ RF2.2- Modificar usuario.
 - ↳ RF2.3- Listar usuarios.
 - ↳ RF2.4- Eliminar usuario.
- ✓ RF-3 Visualizar árbol genealógico.
- ✓ RF-4 Cargar árbol genealógico desde fichero.
- ✓ RF-5 Generar Reporte para Médico.
- ✓ RF-6 Gestionar Nomencladores.
 - ↳ RF6.1- Registrar nomenclador.
 - ↳ RF6.2- Modificar nomenclador.
 - ↳ RF6.3- Listar nomencladores.
 - ↳ RF6.4- Eliminar nomenclador.
- ✓ RF-7 Compartir árbol genealógico.
- ✓ RF-8 Cambiar Contraseña.
- ✓ RF-9 Imprimir Reporte.
- ✓ RF-10 Generar Reporte para Administrador.

2.4.2 Requerimientos no funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. En muchos casos, estos requerimientos son fundamentales en el éxito del producto y generalmente están vinculados a requerimientos funcionales, además corresponden a aspectos tales como la disponibilidad, mantenibilidad, flexibilidad, seguridad, facilidad de uso, entre otros.

Capítulo 2 : Características del Sistema

Apariencia e interfaz externa

Las imágenes y colores deben estar identificados con el negocio del sistema, proporcionando un ambiente agradable y fácil para el usuario. La interfaz externa debe estar diseñada para observarse en cualquier resolución igual o superior a 1024x768.

Usabilidad

El acceso a la aplicación debe realizarse de forma fácil y rápida. La misma contará con un menú que satisfaga las necesidades de los usuarios. Puede ser usada por cualquier persona que posea conocimientos básicos en el manejo de una computadora y el ambiente de un navegador web.

Rendimiento

La velocidad de procesamiento y la capacidad de respuesta de la aplicación web deben ser rápidas, en dependencia del tipo de conexión que esté establecida con el servidor y las características de la red.

Soporte

Una vez concluida la aplicación, se realizarán varias pruebas para comprobar su funcionalidad. Con la puesta en práctica de la aplicación deben quedar satisfechas las necesidades de los usuarios, mediante actualizaciones y mejoras a la misma.

Seguridad

Existen niveles de acceso a la aplicación web, en dependencia del rol que desempeñe cada usuario.

Software

Se requiere para el funcionamiento de la aplicación, disponer de la Máquina Virtual de Java 5.0. Los usuarios de la aplicación deberán contar con un navegador Internet Explorer 5.5, Mozilla Firefox 2.0, o el Opera 8.5.

Capítulo 2 : Características del Sistema

Hardware

Para el desarrollo y ejecución de la aplicación se precisará:

Para el servidor de aplicaciones:

- Microprocesador Pentium III a 2.8 GHz o superior.
- 512 MB de RAM o superior.
- 100 MB de espacio en disco duro para alojar la aplicación web.

Para el cliente:

- Conexión al servidor a través de un módem o tarjeta de red.
- Procesador Pentium III o superior.
- 128 MB RAM como mínimo (Recomendado 256 MB RAM o superior).
- 200 MB para la instalación completa de Mozilla Firefox, Opera o Internet Explorer.
- 200 MB para el funcionamiento de la aplicación web.

Se hace necesario contar con una impresora para la impresión de los reportes.

Disponibilidad

La aplicación debe estar disponible a tiempo completo, y debe recuperarse rápidamente ante cualquier tipo de fallo.

Requisitos legales

Las tecnologías y herramientas que se utilicen para desarrollar la aplicación web deben estar bajo la licencia de software libre.

Persistencia

La información generada por la aplicación debe ser almacenada en base de datos permanentemente, con el objetivo de poder generar reportes y realizar estudios posteriores.

Portabilidad

La aplicación permite ejecutarse en dos de los sistemas operativos más usados en la actualidad, Windows y Linux.

2.5 DEFINICIÓN DE CASOS DE USO (CU) DEL SISTEMA.

Los casos de uso son artefactos narrativos, que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. Cada CU proporciona uno o más escenarios, que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico. (IVAR JACOBSON, 2000)

2.5.1 Patrón de Caso de Uso utilizado.

CRUD

El patrón se basa en la fusión de casos de uso simples para formar una unidad conceptual.

✓ **CRUD Completo:** Este patrón, que consta de un caso de uso llamado Información CRUD (Gestionar Información), modela todas las operaciones que pueden ser realizadas sobre una parte de la información de un tipo específico, tales como creación, lectura, actualización y eliminación.

A continuación se muestra un ejemplo de la utilización de este patrón en la aplicación, específicamente en el caso de uso Gestionar Usuario.

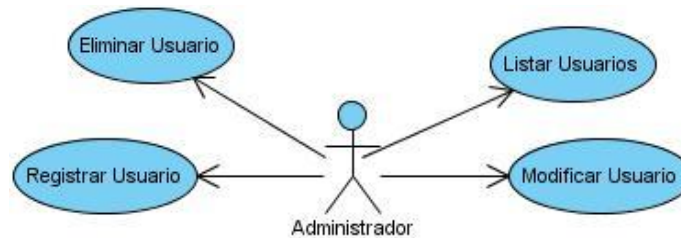


Figura 5: DCU Gestionar Usuario (antes de aplicar el patrón CRUD).

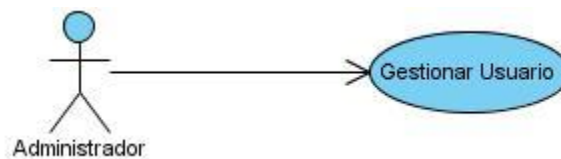


Figura 6: DCU Gestionar Usuario (después de aplicar el patrón CRUD).

2.5.2 Actores del Sistema.

Actor	Descripción
Administrador	Interactúa con el sistema, único encargado de gestionar usuarios y nomencladores.
Genetista	Son los genetistas o médicos de las instituciones de salud. Interactúan con el sistema, pueden generar reportes, compartir árboles genealógicos, entre otras acciones.

2.5.3 Diagrama de Casos de Uso del Sistema.

Un diagrama de CU: es una representación gráfica de parte o el total de los actores y casos de uso del sistema, incluyendo sus interacciones, es decir, describe lo que hace un sistema desde el punto de vista de un observador externo. Estos diagramas se utilizan para ilustrar los requerimientos del sistema, al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo. (IVAR JACOBSON, 2000)

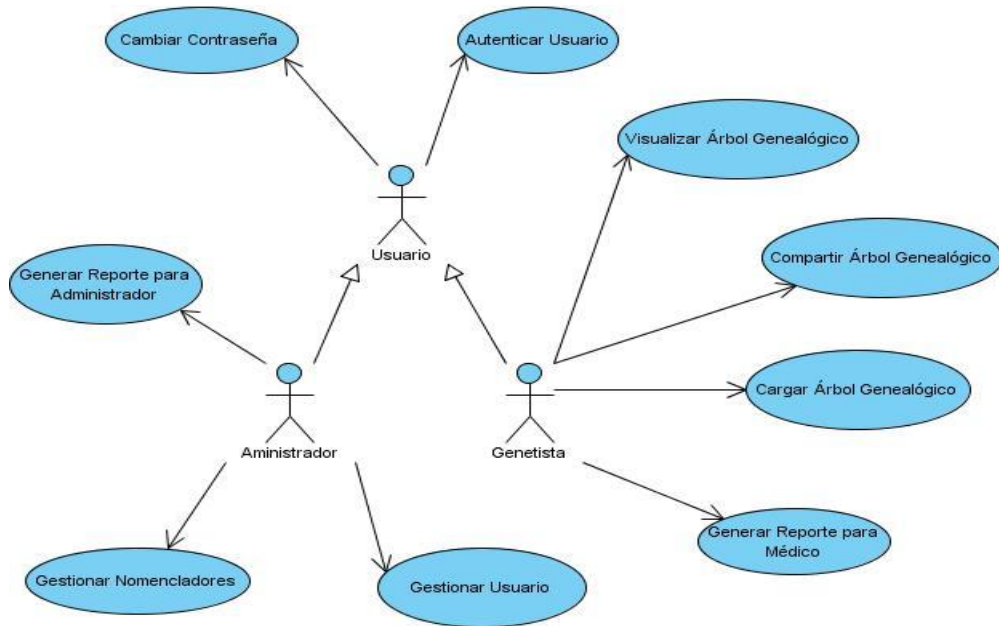


Figura 7: Diagrama de Casos de Uso del Sistema.

2.5.4 Descripción de los Casos de Uso del Sistema.

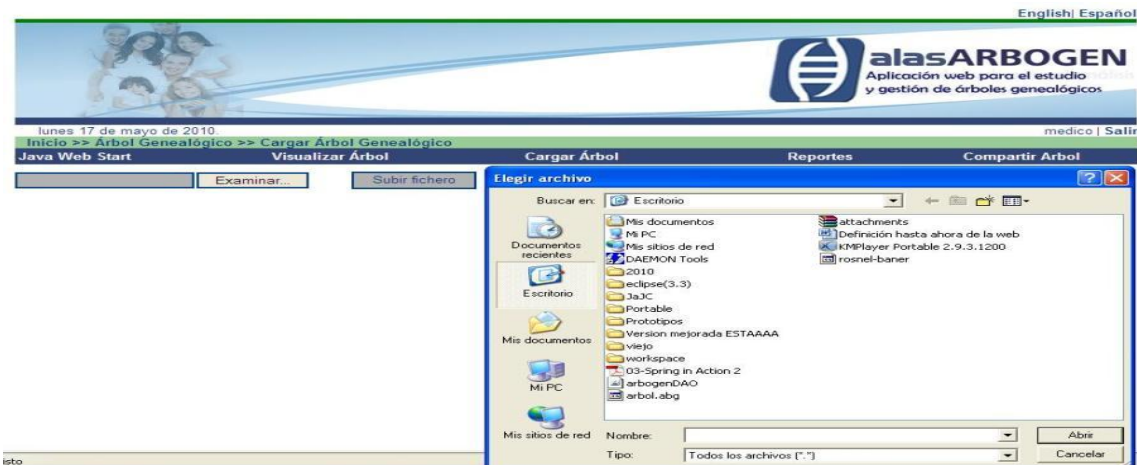
2.5.4.1 Caso de Uso – Cargar árbol genealógico.

Caso de Uso:	Cargar árbol genealógico.
Actor:	Genetista
Resumen:	El Caso de Uso se inicia cuando el genetista desea cargar a la aplicación un árbol genealógico de un paciente, el sistema muestra la interfaz correspondiente.
Precondiciones:	El genetista debe estar autenticado como usuario.
Referencias	RF 4
Prioridad	Crítico.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El genetista selecciona la opción	2. El sistema muestra la interfaz

Capítulo 2 : Características del Sistema

“Cargar árbol genealógico”	correspondiente, donde el genetista debe seleccionar la dirección donde se encuentra el fichero a cargar.
3. El genetista carga el archivo y selecciona la opción Subir Fichero.	4. El sistema guarda el archivo a la base de datos y muestra un mensaje de notificación. Finaliza el Caso de Uso.

Prototipo de Interfaz



Flujos Alternos

Acción del Actor	Respuesta del Sistema
	4.1 Si el sistema no puede cargar el fichero muestra un mensaje indicándolo.
Poscondiciones	El sistema guarda un árbol genealógico.

2.5.4.2 Caso de Uso – Generar Reporte para Médico.

Caso de Uso:	Generar Reporte para Médico.
Actor:	Genetista.
Resumen:	El Caso de Uso se inicia cuando el genetista decide realizar algún reporte.

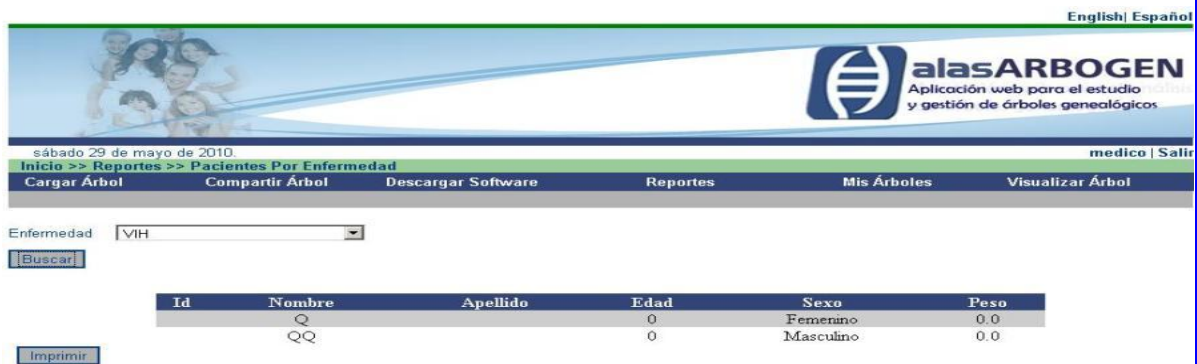
Capítulo 2 : Características del Sistema

Precondiciones:	El genetista deberá estar autenticado como usuario.	
Referencias	RF 5	
Prioridad	Crítico.	
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
1. El genetista selecciona la opción Generar Reporte.		2. El sistema muestra un submenú, donde el genetista debe escoger el criterio de reporte que desea generar.
3. El genetista selecciona el criterio de reporte que desea.		4. En dependencia del criterio de reporte seleccionado, el sistema muestra la interfaz correspondiente: <ul style="list-style-type: none"> • Si el genetista selecciona el criterio Pacientes por Enfermedad, ir a la sección “Reporte de pacientes por enfermedad”. • Si el genetista selecciona el criterio Pacientes con intento de suicidio, ir a la sección “Reporte de pacientes con intento de suicidio”. • Si el genetista selecciona el criterio Pacientes con más de una enfermedad, ir a la sección “Reporte de pacientes con más de una enfermedad”.
Sección “Reporte de pacientes por enfermedad”		
Acción del Actor		Respuesta del Sistema
1. El genetista selecciona el criterio Pacientes por enfermedad.		2. El sistema muestra la interfaz donde el usuario debe escoger la enfermedad, para buscar los pacientes que la padecen.

Capítulo 2 : Características del Sistema

3. Si genetista selecciona la enfermedad que desea y oprime el botón Buscar.	4. El sistema muestra una interfaz con los pacientes que padecen dicha enfermedad, brindando la opción de imprimir el reporte.
5. El genetista oprime el botón Imprimir.	6. El sistema imprime el reporte. Finaliza el Caso de Uso.

Prototipo de Interfaz



Flujos Alternos

Acción del Actor	Respuesta del Sistema
	3.1 Si el usuario no selecciona ninguna enfermedad y oprime el botón buscar, el sistema muestra un mensaje indicando que no existen resultados para esa búsqueda.
5.1 Si el usuario no desea imprimir el reporte Finaliza el Caso de Uso.	

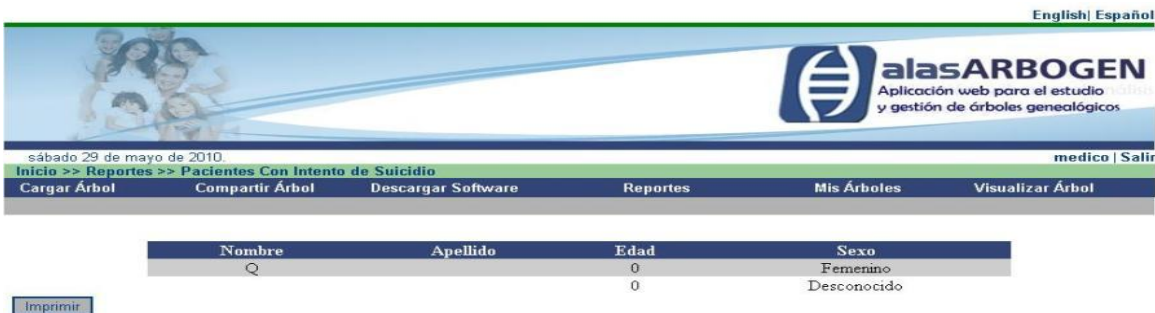
Sección "Reporte de pacientes con intento de suicidio"

Acción del Actor	Respuesta del Sistema
1. El usuario selecciona el criterio Pacientes con intento de suicidio.	2. El sistema muestra la lista de pacientes que han intentado suicidarse,

Capítulo 2 : Características del Sistema

	brindando la posibilidad de imprimir el reporte. Finaliza el Caso de Uso.
3. El genetista oprime el botón Imprimir.	4. El sistema imprime el reporte. Finaliza el Caso de Uso.

Prototipo de Interfaz



Flujos Alternos

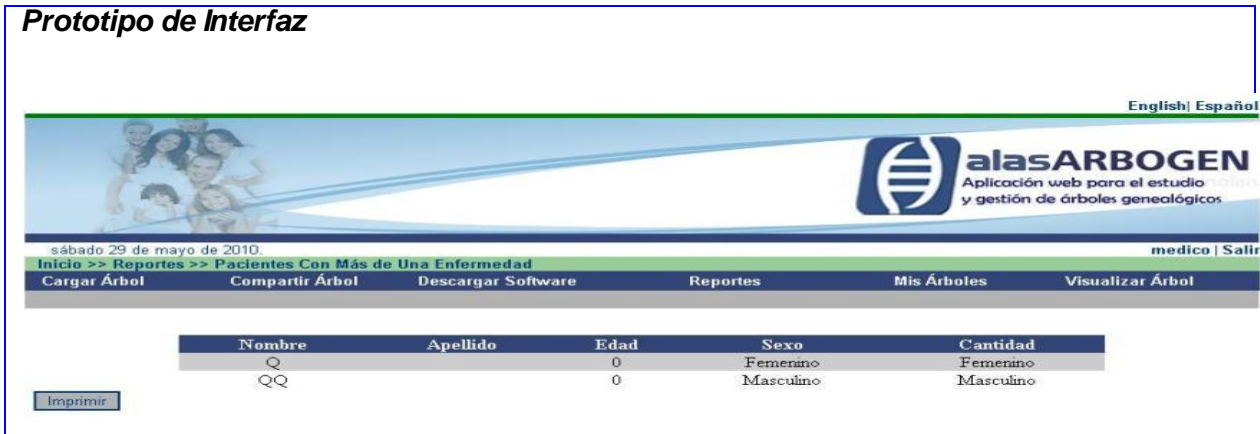
Acción del Actor	Respuesta del Sistema
	2.1 Si no existen pacientes que hayan intentado suicidarse, el Sistema muestra un mensaje indicándolo.
3.1 Si el genetista no desea imprimir el reporte Finaliza el Caso de Uso.	

Sección "Reporte de pacientes con más de una enfermedad"

Acción del Actor	Respuesta del Sistema
1. El genetista selecciona el criterio Pacientes con más de una enfermedad.	2. El sistema muestra una interfaz con todos los pacientes que padecen más de una enfermedad, brindando la posibilidad de imprimir el reporte. Finaliza el Caso de Uso.
3. El genetista oprime el botón Imprimir	

Capítulo 2 : Características del Sistema

Prototipo de Interfaz



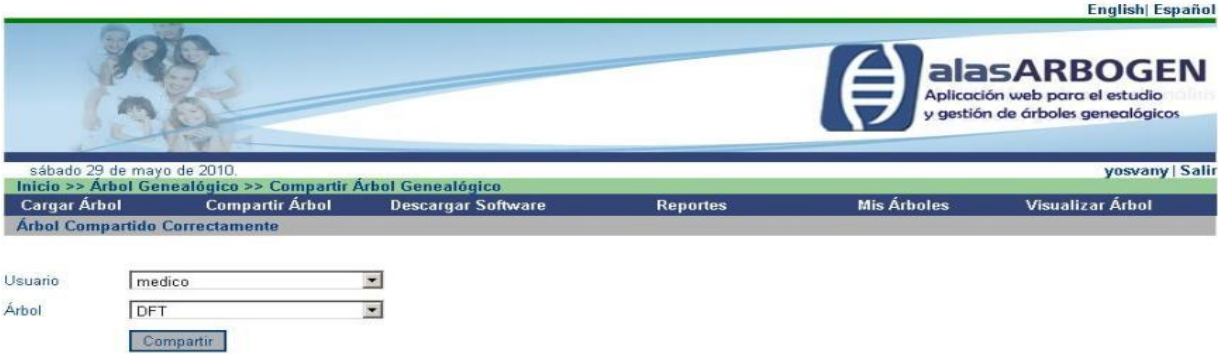
Flujos Alternos

Acción del Actor	Respuesta del Sistema
	2.1. Si no existen pacientes que padecen más de una enfermedad, el sistema muestra un mensaje indicándolo.
3.1 Si el genetista no desea imprimir el reporte Finaliza el Caso de Uso.	
Poscondiciones	Se genera un Reporte.

2.5.4.3 Caso de Uso – Compartir árbol genealógico.

Caso de Uso:	Compartir árbol genealógico
Actor:	Genetista
Resumen:	El Caso de Uso se inicia cuando el usuario desea compartir árbol genealógico, el sistema muestra la interfaz correspondiente.
Precondiciones:	El genetista deberá estar autenticado como usuario.
Referencias	RF 7
Prioridad	Crítico

Capítulo 2 : Características del Sistema

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El genetista selecciona la opción "Compartir árbol genealógico".	2. El sistema muestra una interfaz donde el usuario debe escoger el árbol que quiere compartir y el médico a quien va destinado.
3. El genetista selecciona el árbol, el médico y oprime el botón compartir.	4. EL sistema verifica que el árbol no haya sido compartido ya a ese usuario.
	5. El sistema comparte el árbol genealógico y muestra un mensaje de notificación. Finaliza el Caso de Uso.
Prototipo de Interfaz	
	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	3.1 Si el genetista no selecciona los datos correspondientes, el sistema muestra un mensaje indicándolo.
	4.1 Si el árbol estaba compartido anteriormente al usuario seleccionado, el sistema muestra un

Capítulo 2 : Características del Sistema

	mensaje indicándolo.
Poscondiciones	Se comparte un árbol genealógico.

Las restantes descripciones de CU se encuentran el documento Materiales Complementarios.

CONCLUSIONES.

En el capítulo se identificaron los principales problemas que llevaron a realizar esta investigación, de acuerdo con las necesidades actuales de la Red Nacional de Genética Médica. Además, se expusieron los requerimientos que debe cumplir el sistema, los casos de uso del mismo y las descripciones de estos.

CAPÍTULO 3: DISEÑO DEL SISTEMA

Introducción

En el capítulo se exponen los patrones utilizados en el diseño del sistema. Además, se muestran diagramas de clases del diseño y diagramas de secuencia que servirán como base para la implementación del sistema.

3.1 PATRONES.

Un patrón es una unidad de información nombrada, instructiva e intuitiva, que captura la esencia de una familia exitosa de soluciones probadas, a un problema recurrente dentro de un cierto contexto. El objetivo de los patrones es crear un lenguaje común a una comunidad de desarrolladores, para comunicar experiencias sobre los problemas y sus soluciones. Pueden referirse a distintos niveles de abstracción, desde un proceso de desarrollo hasta la utilización eficiente de un lenguaje de programación. (IVAR JACOBSON, 2000)

3.1.1 Patrón arquitectónico "Modelo-Vista-Controlador" (MVC).

Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web.

A continuación, se muestra (ver figura 8) en una estructura de paquetes el funcionamiento del patrón MVC en la aplicación web desarrollada.

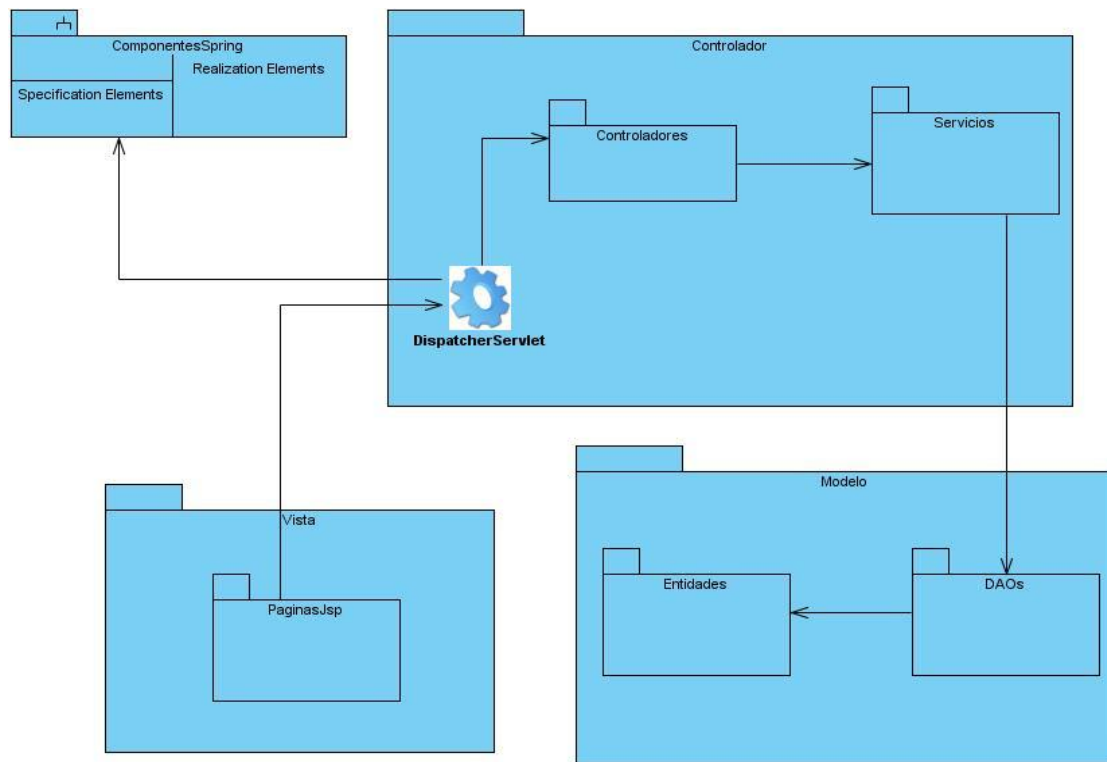


Figura 8: Patrón Modelo-Vista-Controlador.

Modelo: Es la representación específica del dominio de la información sobre la cual funciona la aplicación. El modelo es otra forma de llamar a la capa de dominio. Es el responsable de acceder a la capa de almacenamiento de datos y definir las reglas de negocio.

Vista: Presenta el modelo en un formato adecuado para interactuar, usualmente es un elemento de interfaz de usuario. Es responsable de recibir datos del modelo y mostrarlos al usuario.

Controlador: Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

Spring – MVC: Es uno de los módulos del framework Spring y como su propio nombre indica implementa una arquitectura Modelo-Vista-Controlador que se utilizó como base para desarrollar la aplicación web del presente trabajo. Este módulo es confiable, tiene capacidades y funcionalidades suficientes e ideales para el desarrollo de un buen software.

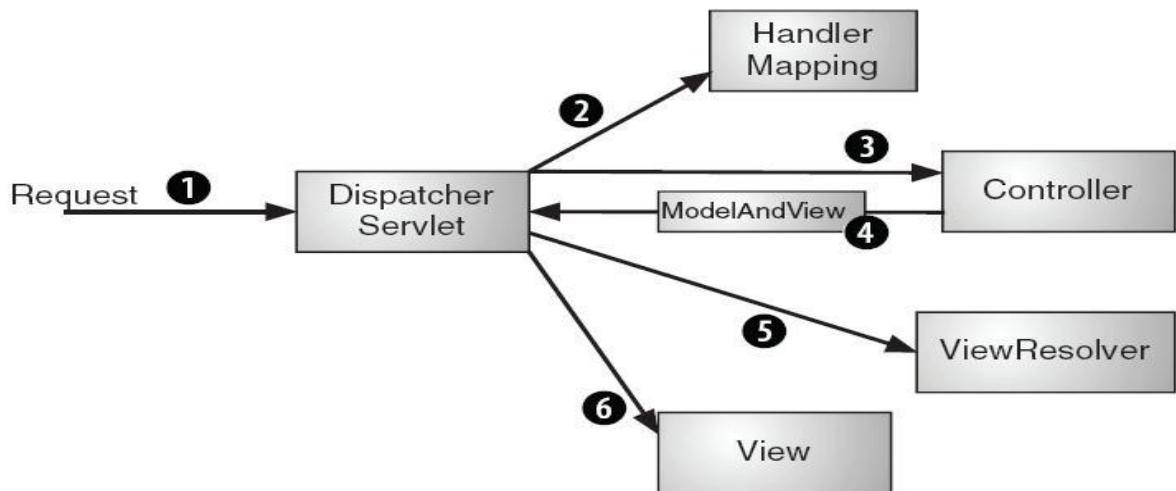


Figura 9: Módulo Spring MVC.

El patrón MVC fue diseñado para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples. Es usado principalmente en aplicaciones que manejan gran cantidad de datos y transacciones complejas, donde se requiere una mejor separación de conceptos para que el desarrollo esté estructurado de una mejor manera, facilitando la programación en diferentes capas, de una forma paralela e independiente.

3.1.2 Patrones de diseño.

Patrones Grasp aplicados.

GRASP es un acrónimo que significa General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades). El nombre se eligió para indicar la importancia de captar estos principios, si se quiere diseñar eficazmente un software orientado a objetos.

Bajo Acoplamiento

En la aplicación desarrollada no existe una herencia profunda entre clases, lo que significa que una clase no depende de muchas clases, evitando problemas como el difícil entendimiento de ellas cuando estén aisladas, y facilitando la reutilización de las mismas, al comunicarse entre ellas lo menos posible.

La siguiente figura evidencia la existencia de este patrón en la aplicación.

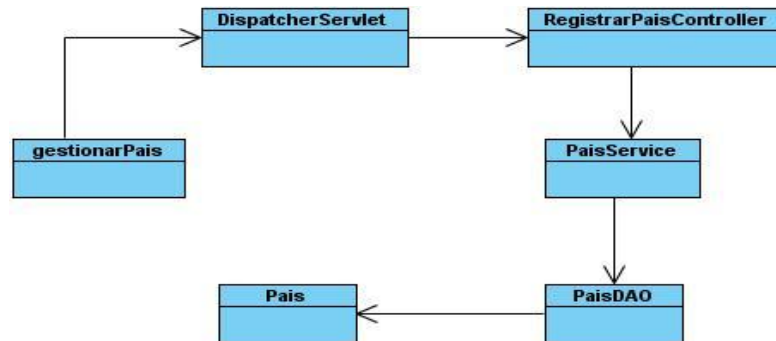


Figura 10: Ejemplo de la aplicación del patrón Bajo Acoplamiento.

Alta Cohesión

Para garantizar la existencia de este patrón es necesario mantener clases que no realicen un trabajo excesivo. Buenas prácticas como la creación de paquetes de servicios, o clases agrupadas por funcionalidades que son fácilmente reutilizables y el uso de interfaces guían al sistema a seguir el principio de alta cohesión.

En la figura 11 se muestra un ejemplo del patrón alta cohesión.

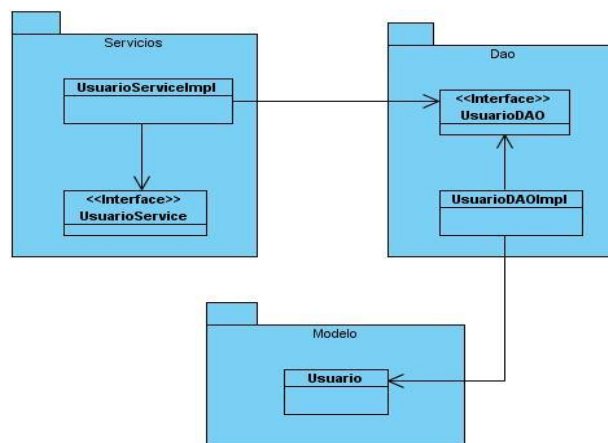


Figura 11: Ejemplo de la aplicación del patrón Alta Cohesión.

Experto

Una aplicación puede requerir el cumplimiento de cientos o miles de responsabilidades; asignando éstas, de forma adecuada, el sistema tiende a ser más fácil de entender y habrá más oportunidades de reutilizar los componentes en futuras aplicaciones. La responsabilidad de realizar una labor es de la clase que cuenta con la información necesaria para cumplir el compromiso.

A continuación se muestra un ejemplo del uso de este patrón.

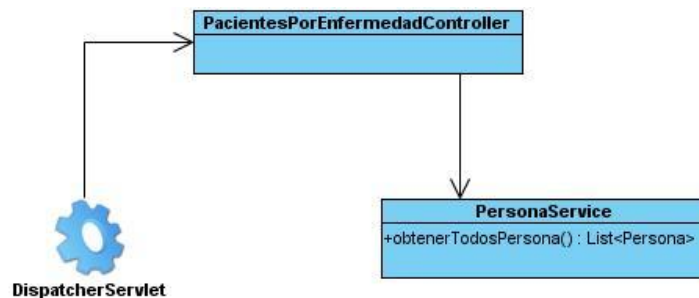


Figura 12: Ejemplo de aplicación del patrón Experto.

Indirección

Se asigna la responsabilidad a un objeto intermedio, para que medie entre otros componentes o servicios, y éstos no terminen directamente acoplados. El intermediario crea una indirección entre el resto de los componentes o servicios. Un ejemplo de cómo se evidencia el patrón en la aplicación se muestra en la siguiente imagen:

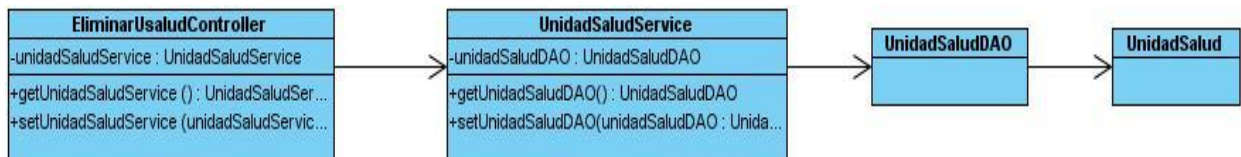


Figura 13: Ejemplo de la aplicación del patrón Indirección.

Creador

Crear objetos es una de las actividades más frecuentes en un sistema orientado a objetos. Este patrón guía la asignación de responsabilidades relacionadas con la creación de ellos. El diseño bien asignado puede soportar una mayor claridad, encapsulamiento y reusabilidad.

Patrones GOF aplicados.

Singleton (Instancia única): Es un patrón creacional (según las clasificaciones⁴ de los patrones GOF). Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella, teniendo un acceso controlado a dicha instancia y dando la posibilidad de llevar un control estricto sobre cómo y cuándo acceden los clientes a la misma.

El siguiente código (ver figura 14) pone de manifiesto el uso de este patrón:

⁴ Los patrones GOF (Banda de los cuatro, por sus siglas en Inglés) se clasifican en tres grupos de acuerdo al propósito para el cual han sido definidos:

- **Creacionales:** Tratan con las formas de crear instancias de objetos. El objetivo de estos patrones es de abstraer el proceso de instanciación y ocultar los detalles de cómo los objetos son creados o inicializados.
- **Estructurales:** Describen cómo las clases y los objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades.
- **De comportamiento:** Ayudan a definir la comunicación e interacción entre los objetos de un sistema. El propósito es reducir el acoplamiento entre los objetos.

```
<bean id="registrarPaisController"
class="configWeb.administracion.gestionarNomencladores.controladores.RegistrarPa
isController">
  <property name="commandClass">
    <value>model.Pais</value>
  </property>
  <property name="commandName">
    <value>pais</value>
  </property>
  <property name="formView">
    <value>gestionarPais</value>
  </property>
  <property name="paisService">
    <ref bean="paisService" />
  </property>
  <property name="validator" >
    <bean
      class="configWeb.administracion.gestionarNomencladores.controladores
        .validadores.PaisValidator"/>
  </property>
</bean>
```

Figura 14: Código de ejemplo del Patrón Singleton.

Interface (Interfaz): Es considerado un patrón de diseño fundamental (no entra en ninguna de las clasificaciones de los patrones GOF). Define un comportamiento independiente de donde vaya a ser utilizado. La siguiente figura evidencia el uso de este patrón:

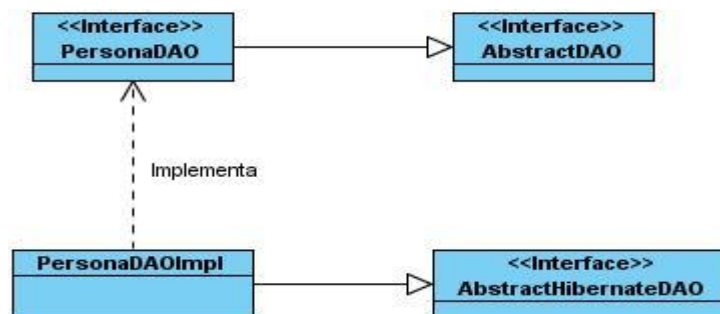


Figura 15: Ejemplo de la aplicación del patrón Interface.

3.3 DIAGRAMAS DE CLASES DE DISEÑO.

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema. También son importantes para la especificación, documentación y construcción de sistemas ejecutables.

Son utilizados durante el proceso de análisis y diseño del sistema, donde se crea el diseño conceptual de la información que se manejará en el mismo, y los componentes que se encargarán de la relación entre uno y otro.

A continuación se muestran los diagramas de clases de diseño de los Casos de Uso del sistema, en los mismos se hace uso del patrón Modelo-Vista-Controlador, mencionado anteriormente:

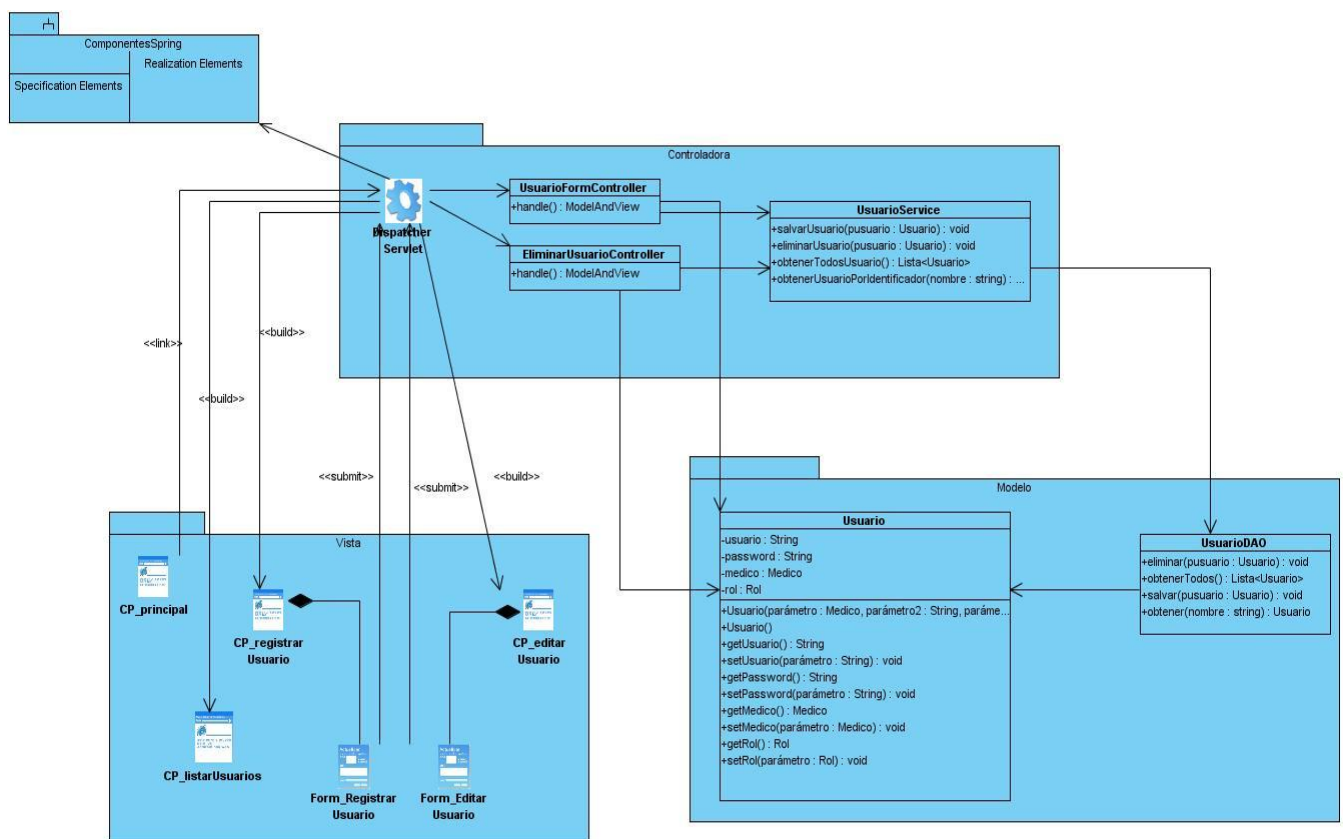


Figura 16: Diagrama de Clases de Diseño (CU Gestionar Usuario).

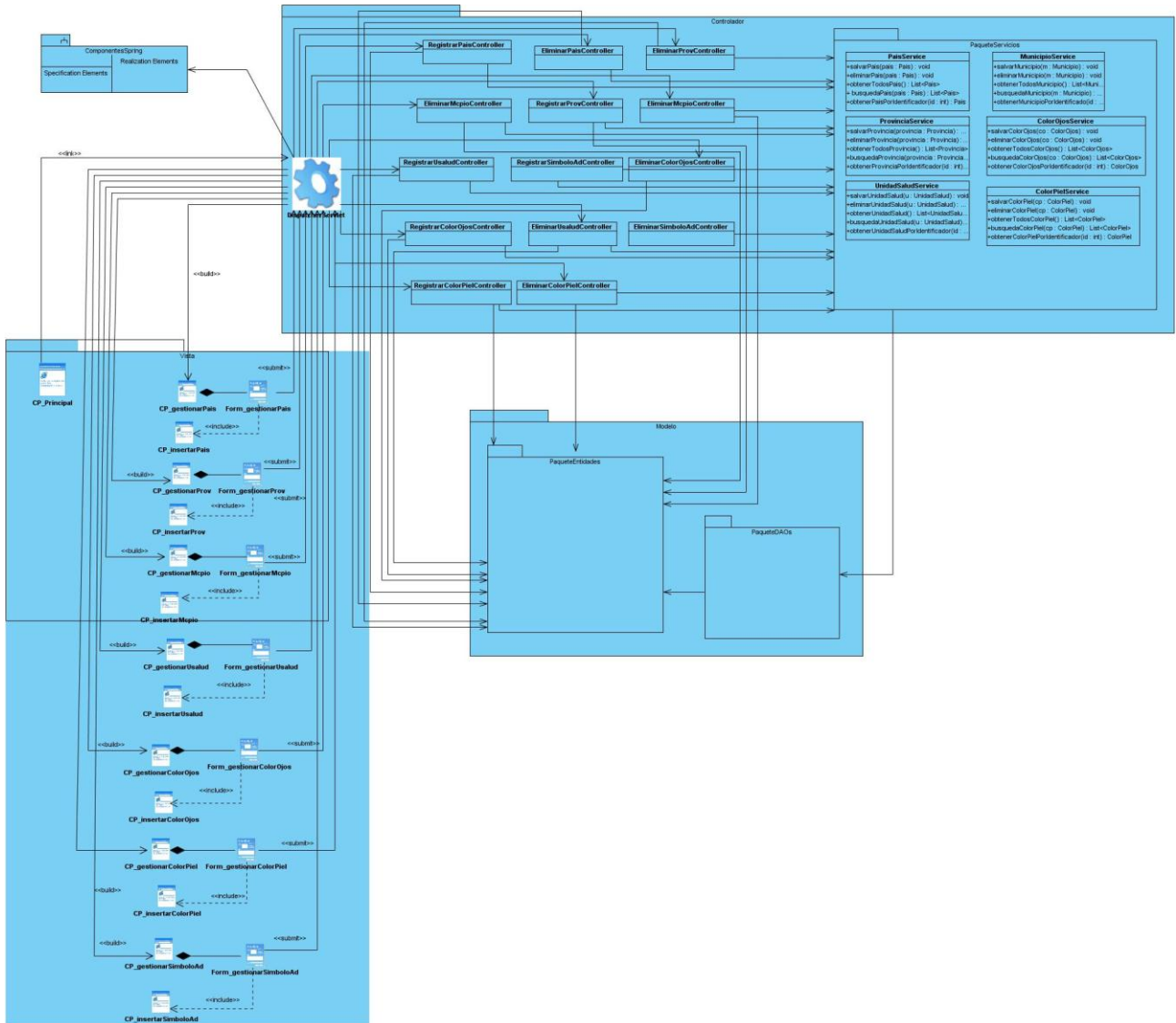


Figura 17: Diagrama de Clases de Diseño (CU Gestionar Nomencladores).

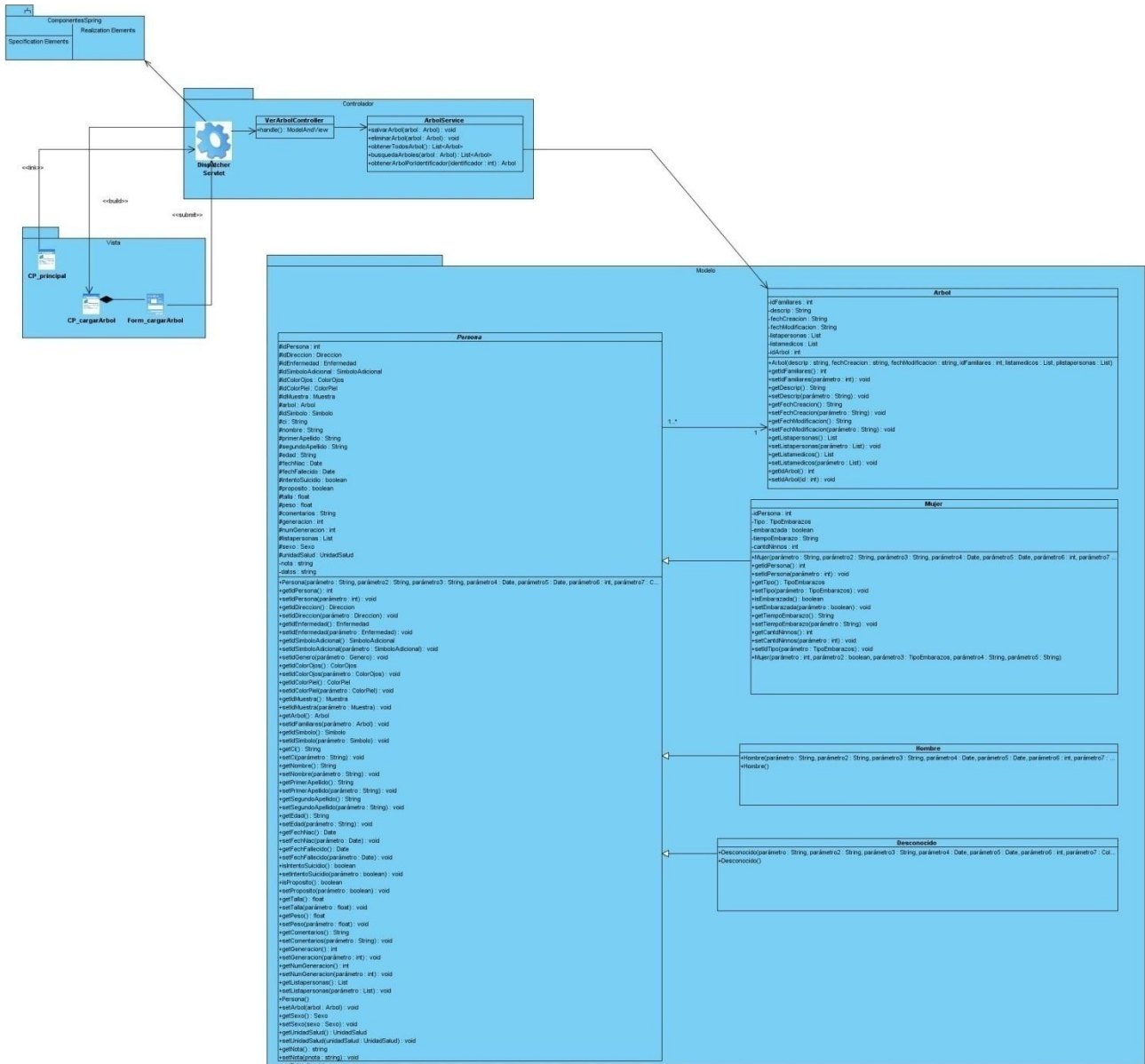


Figura 18: Diagrama de Clases de Diseño (CU Cargar Árbol Genealógico).

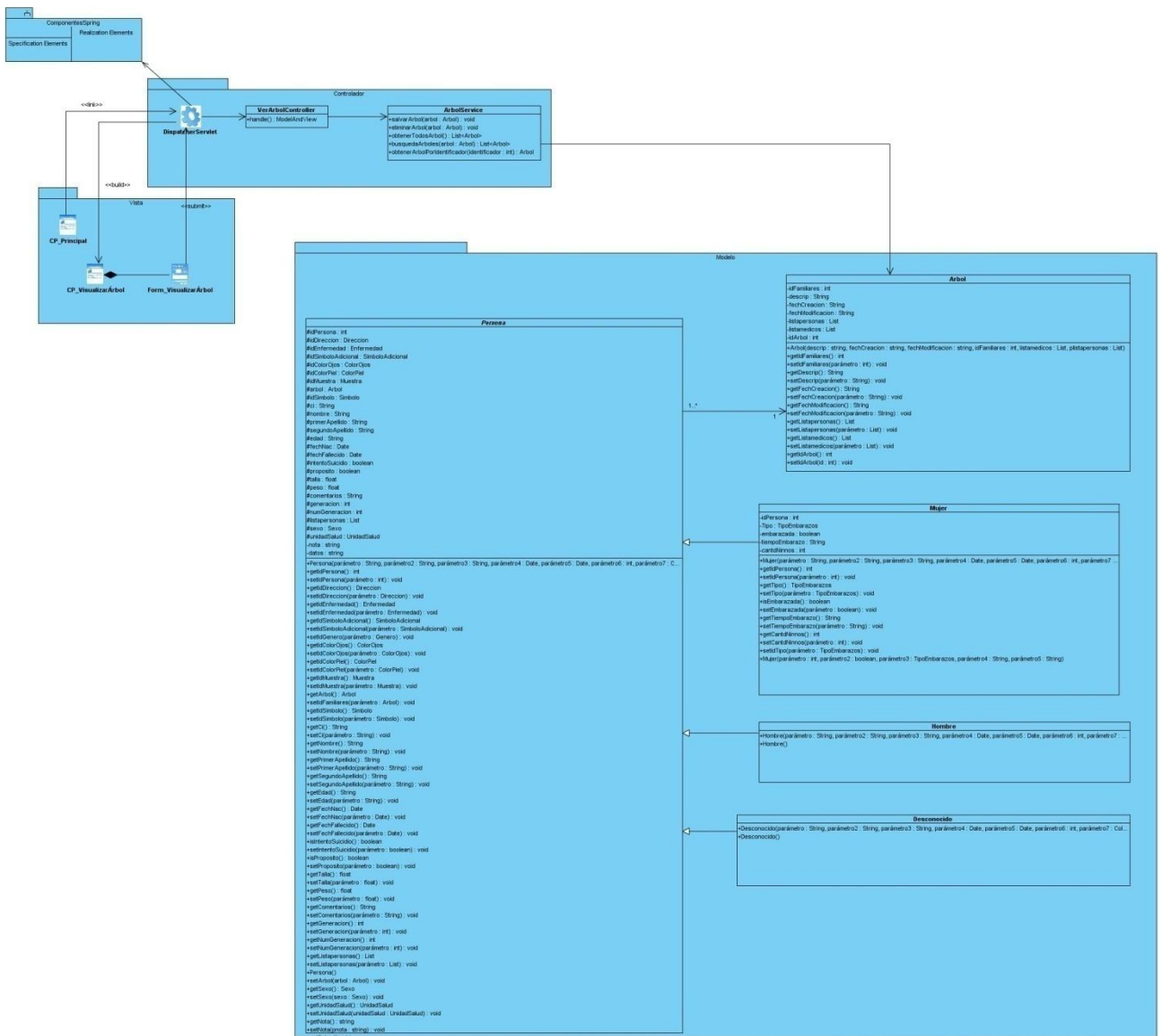


Figura 19: Diagrama de Clases de Diseño (CU Visualizar Árbol Genealógico).

Capítulo 3 : Diseño del Sistema

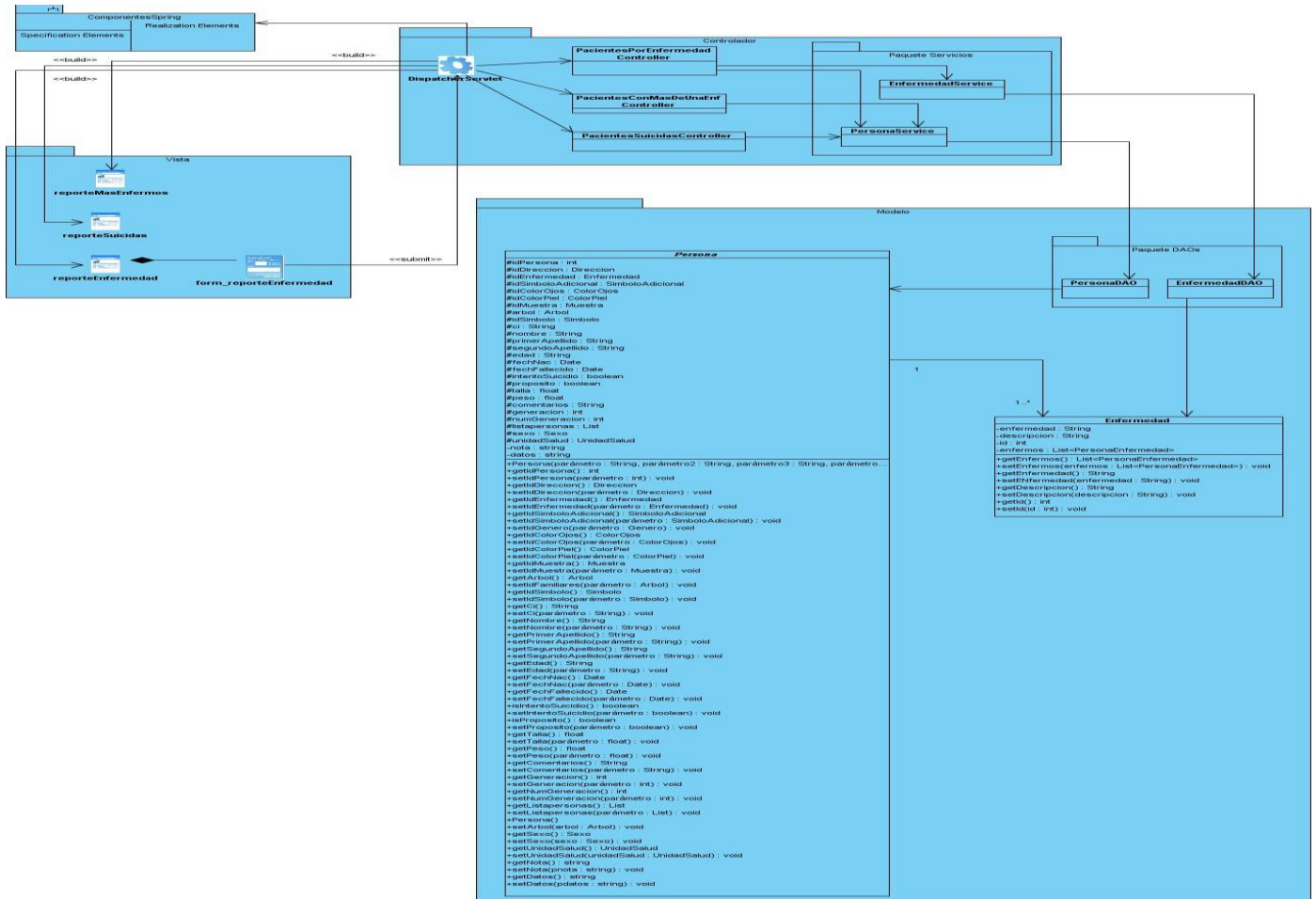


Figura 20: Diagrama de Clases de Diseño (CU Generar Reporte para Médico).

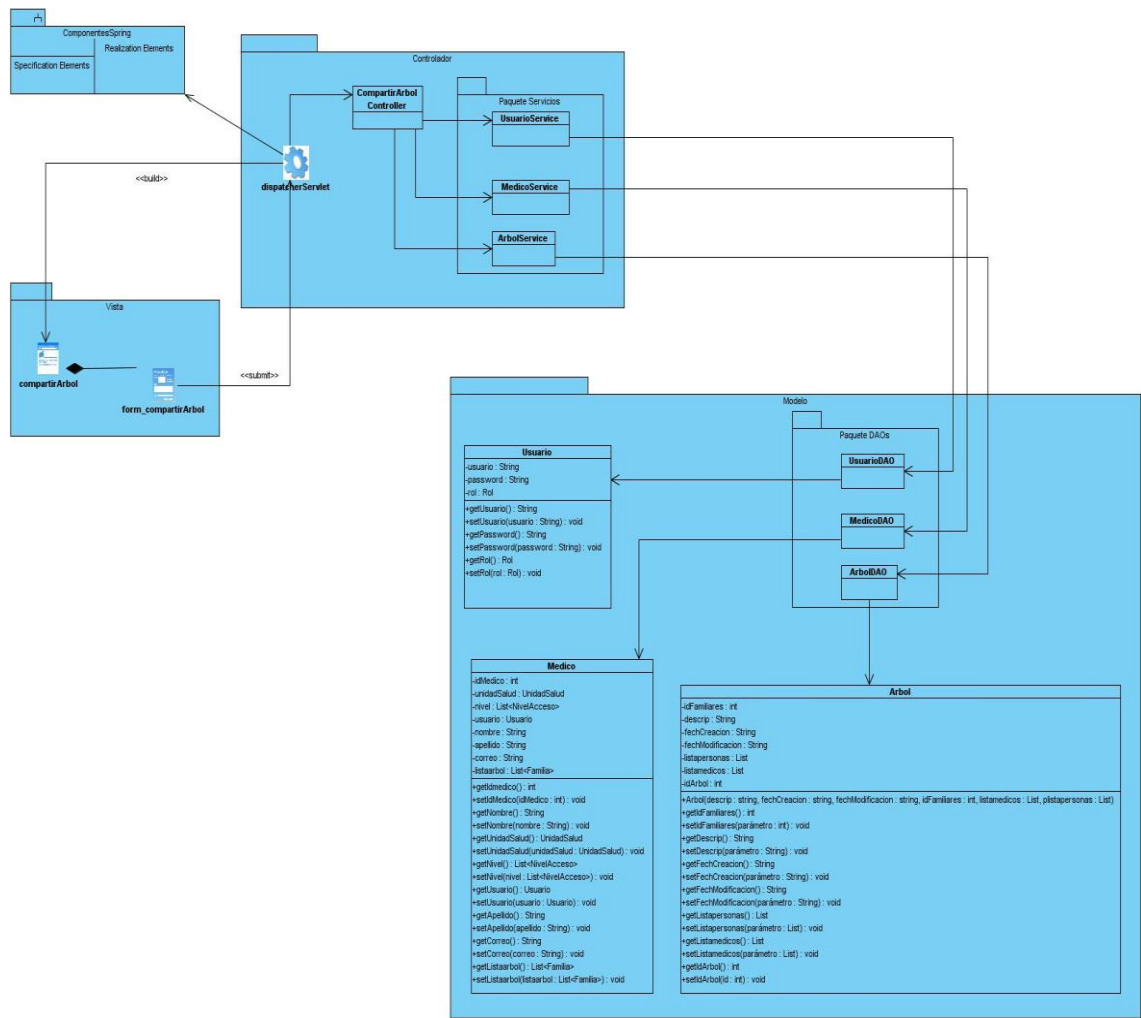


Figura 21: Diagrama de Clases de Diseño (CU Compartir Árbol Genealógico).

3.4 DIAGRAMAS DE SECUENCIA

El diagrama de secuencia forma parte del modelado dinámico del sistema. Es útil para identificar llamadas a realizar o posibles errores del modelado estático, que imposibiliten el flujo de información o de llamadas entre los componentes de éste. Además, es bueno para mostrar las comunicaciones entre objetos y los mensajes que disparan dichas comunicaciones.

A continuación se muestran varios diagramas de secuencia, en el documento de materiales complementarios serán expuestos los restantes.

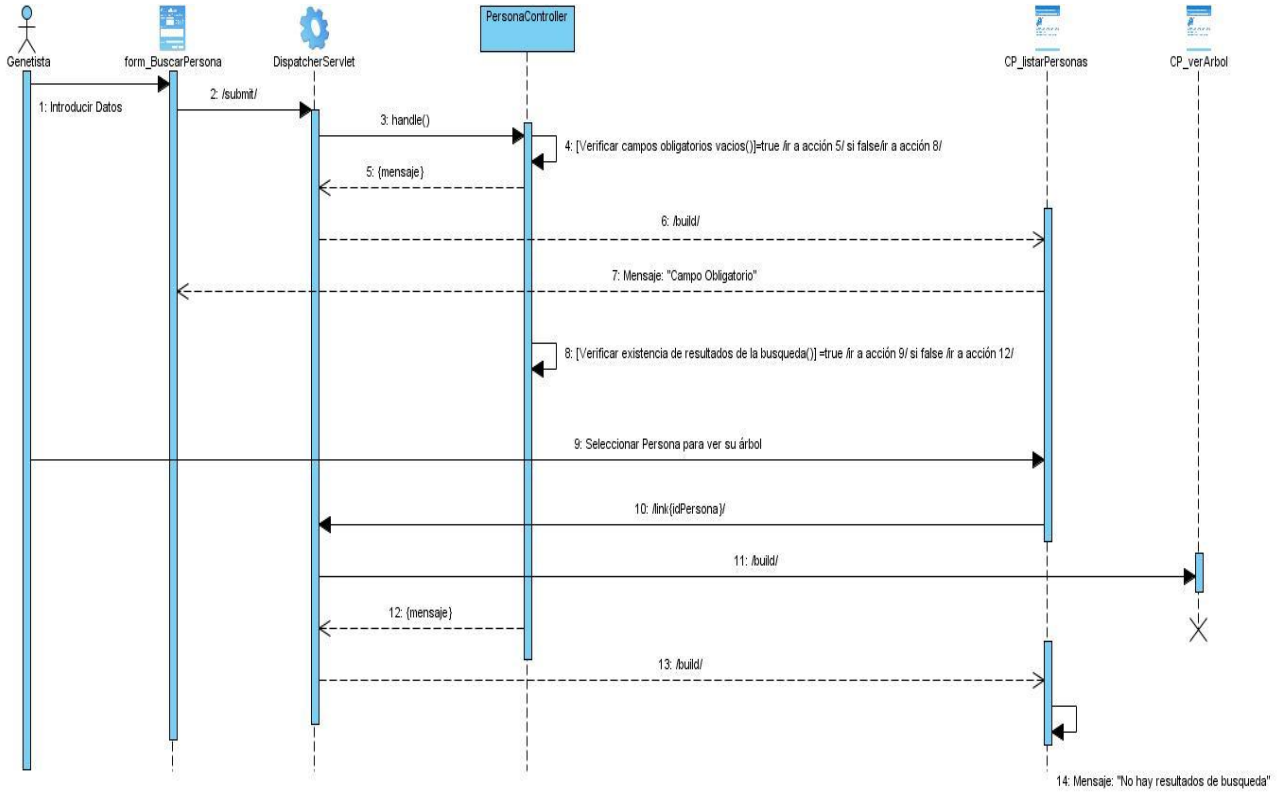


Figura 22: Diagrama de Secuencia (CU Visualizar árbol genealógico).

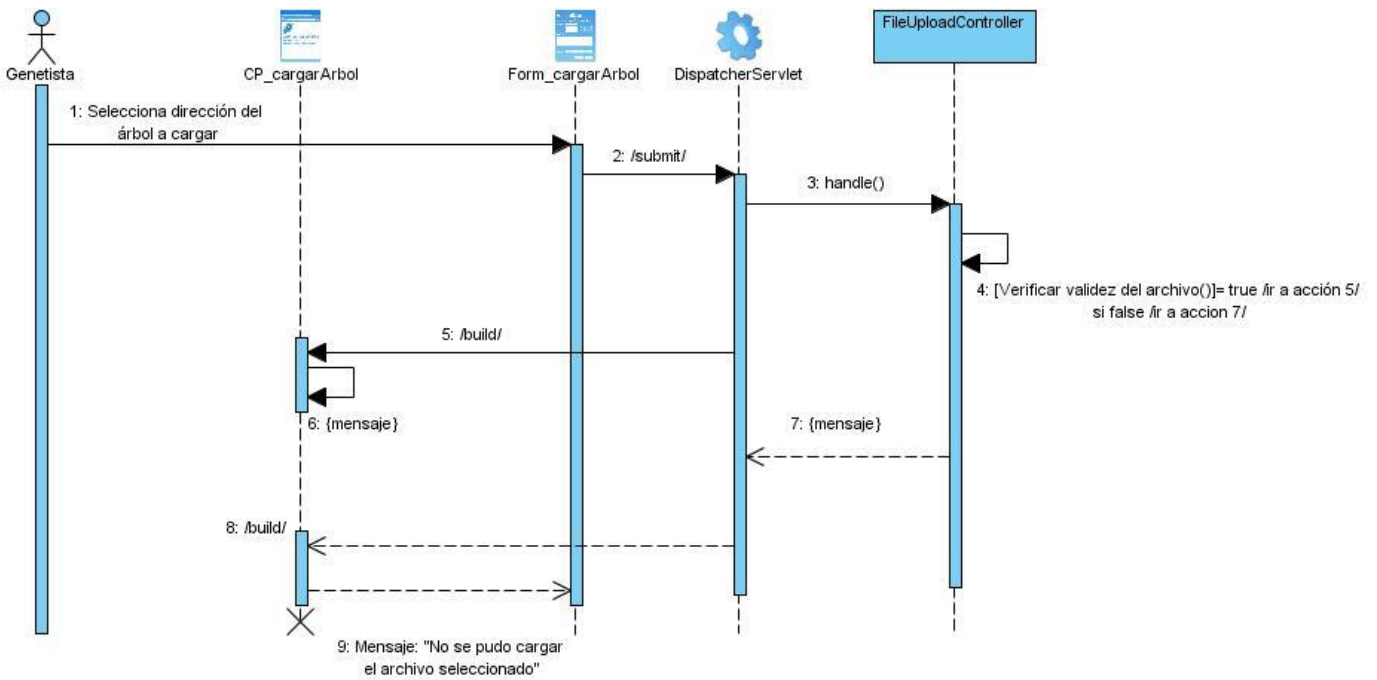


Figura 23: Diagrama de Secuencia (CU Cargar árbol genealógico).

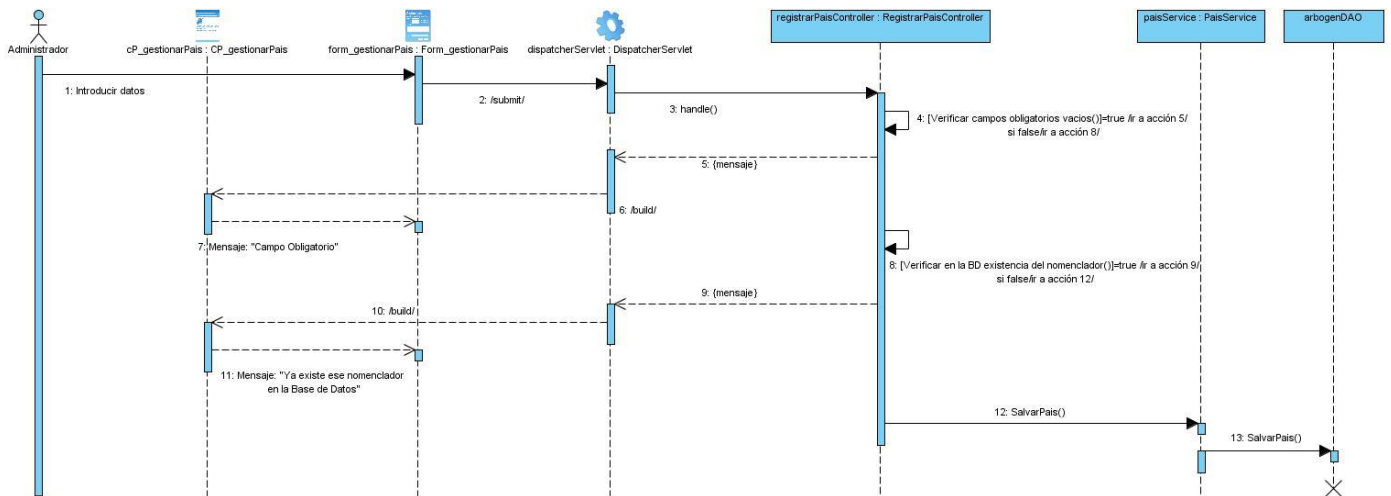


Figura 24: Diagrama de Secuencia (CU Gestionar Nomencladores, Escenario Registrar Nomenclador).

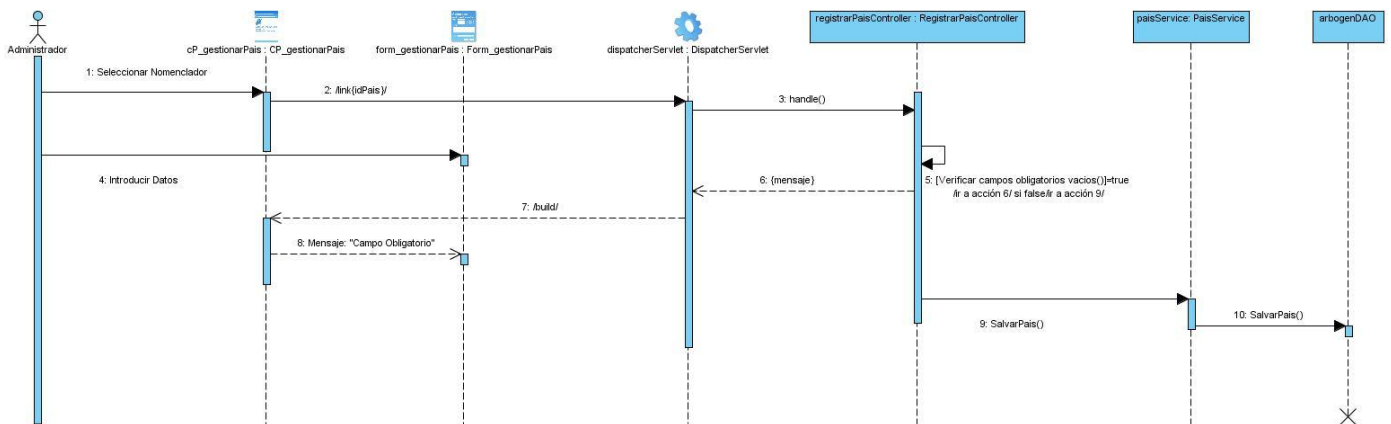


Figura 25: Diagrama de Secuencia (CU Gestionar Nomencladores, Escenario Editar Nomenclador).

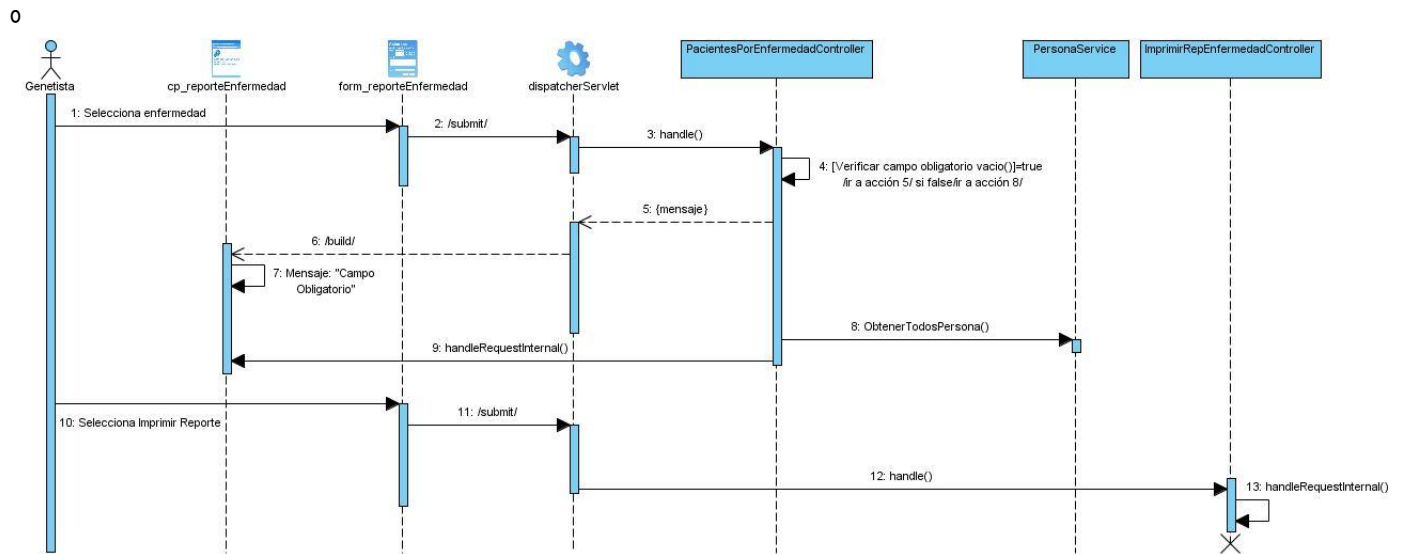


Figura 26: Diagrama de Secuencia (CU Generar Reporte para Médico, Escenario Generar Reporte por Enfermedad).

3.5 DIAGRAMA DE DESPLIEGUE.

Es un tipo de diagrama UML, que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. Este tipo de diagrama se preocupa principalmente de la distribución, entrega e instalación de las partes que constituyen el sistema físico.

A continuación se muestra el diagrama de despliegue (ver figura 20) correspondiente al sistema alasARBOGEN 2.0. En el mismo se evidencia una máquina cliente que se conecta a un servidor de aplicaciones (mediante el protocolo de hipertexto HTTP) y éste a través del protocolo TCP/IP establece conexión con el servidor de base de datos. La PC cliente además se conecta a una impresora para la impresión de los reportes mediante un puerto USB.

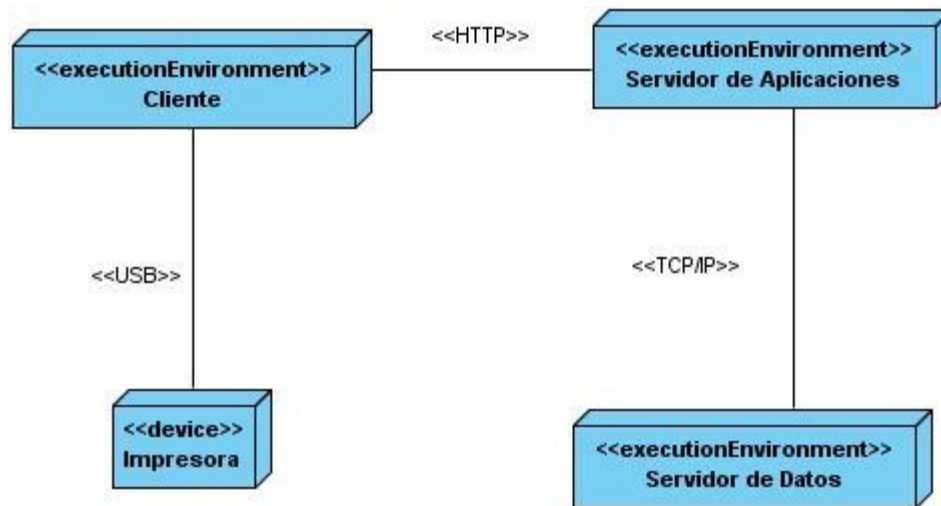


Figura 27: Diagrama de Despliegue del Sistema alasARBOGEN 2.0.

CONCLUSIONES.

En el capítulo se mostraron algunos diagramas de secuencia y de clases del diseño de los casos de uso del sistema. Se utilizó el patrón arquitectónico Modelo-Vista-Controlador para la implementación del software, así como varios patrones de diseño, por ejemplo: Creador, Bajo Acoplamiento y Alta Cohesión.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

Introducción

Este capítulo contiene descripciones de la implementación del sistema en términos de componentes, para dar solución a los requisitos funcionales especificados anteriormente, además se aborda acerca de las pruebas realizadas al software.

4.1 IMPLEMENTACIÓN.

En la etapa de Implementación se comienza con el resultado de la etapa de Diseño y se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ejecutables y similares.

El objetivo principal de esta etapa es desarrollar la arquitectura y el sistema como un todo. De forma más específica, los propósitos de la Implementación son: (IVAR JACOBSON, 2000)

- Definir la organización del código.
- Planificar las integraciones de sistema necesarias en cada iteración.
- Implementar las clases y subsistemas encontrados durante el Diseño.
- Probar los componentes individualmente, y a continuación integrarlos compilándolos y enlazándolos en uno o más ejecutables.

4.1.1 Fragmentos de Código.

```
if(request.getParameter("tipoReporte").equals("porEnfermedad")){
    String enfermedad=request.getParameter("enferme");

    Enfermedad enfer=enfermedadService.obtenerEnfermedadPorIdentificador(Integer.parseInt(enfermedad));

    mv.addObject("enfermedad",enfer);

    List<Persona> pers=personaService.obtenerTodosPersona();

    List<Persona> enf= new ArrayList<Persona>();
    for (int i = 0; i < pers.size(); i++) {

        for (int j = 0; j < pers.get(i).getEnfermos().size(); j++) {

            if(pers.get(i).getEnfermos().get(j).getId().getIdEnfermedad()==enfer.getId()){
                enf.add(pers.get(i));
            }
        }
    }
    mv.addObject("enf", enf);
}
```

Figura 28: Fragmentos de código de la clase ImprimirRepEnfermedadController.

```
protected ModelAndView onSubmit(HttpServletRequest request,
    HttpServletResponse response, Object command, BindException errors)
    throws Exception {
    ModelAndView r = new ModelAndView("compartirArbol");
    ...
    String user = request.getParameter("user");
    String arbol = request.getParameter("arbol");
    /**
     * DATOS DE COMPARTIR EL ARBOL, NIVEL DE ACCESO
     */

    List<NivelAcceso> listAccesos=nivelAccesoService.obtenerTodosNivelAcceso();
    int cont=0;
    for (int j = 0; j < listAccesos.size(); j++) {
        if( (listAccesos.get(j).getId().getIdMedico()==m.getIdMedico())
            && (listAccesos.get(j).getId().getIdFamilia()==Integer.parseInt

                (arbol) )){
            cont++;
            break;
        }
    }

    if(cont==0){
        NivelAccesoId nId=new NivelAccesoId();
        nId.setIdFamilia(Integer.parseInt(arbol));
        nId.setIdMedico(m.getIdMedico());
        NivelAcceso n=new NivelAcceso();
        n.setId(nId);
        nivelAccesoService.salvarNivelAcceso(n);
        request.getSession().setAttribute("myHelpMessage", "arbol.compartido");
    }
    else{
        request.getSession().setAttribute("myHelpMessage", "arbol.no.compartido");
    }
    ...
}
```

Figura 29: Fragmentos de código de la clase CompartirArbolController.

4.1.2 Diagramas de Componentes.

Representan cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes. Los diagramas de componentes prevalecen en el campo de la arquitectura de software, pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema. (DANIELE, 2007)

A continuación se muestran varios diagramas de componentes del sistema:

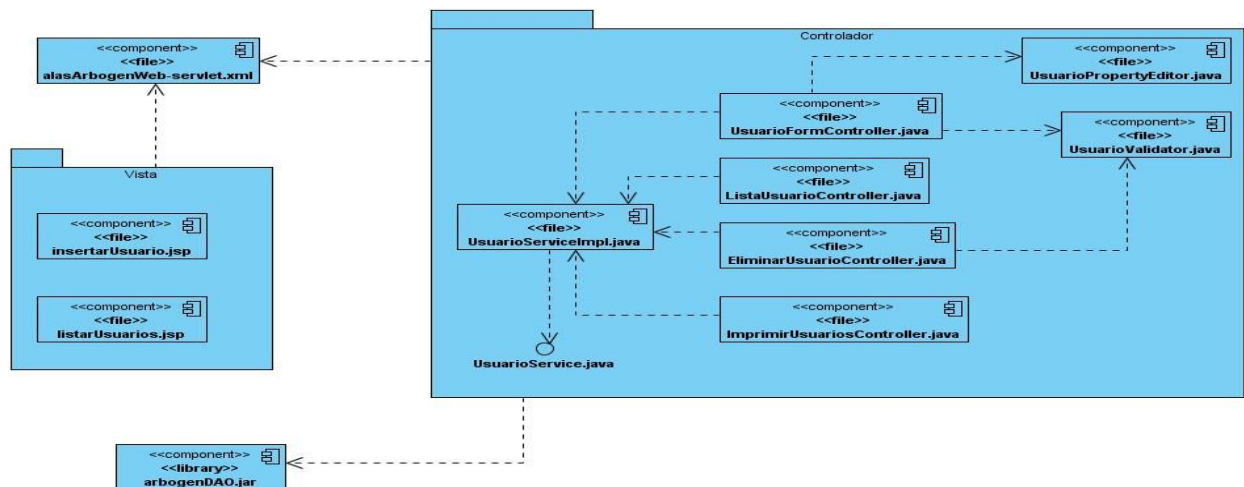


Figura 30: Diagrama de Componentes (CU Gestionar Usuario).

Capítulo 4 : Implementación y Pruebas

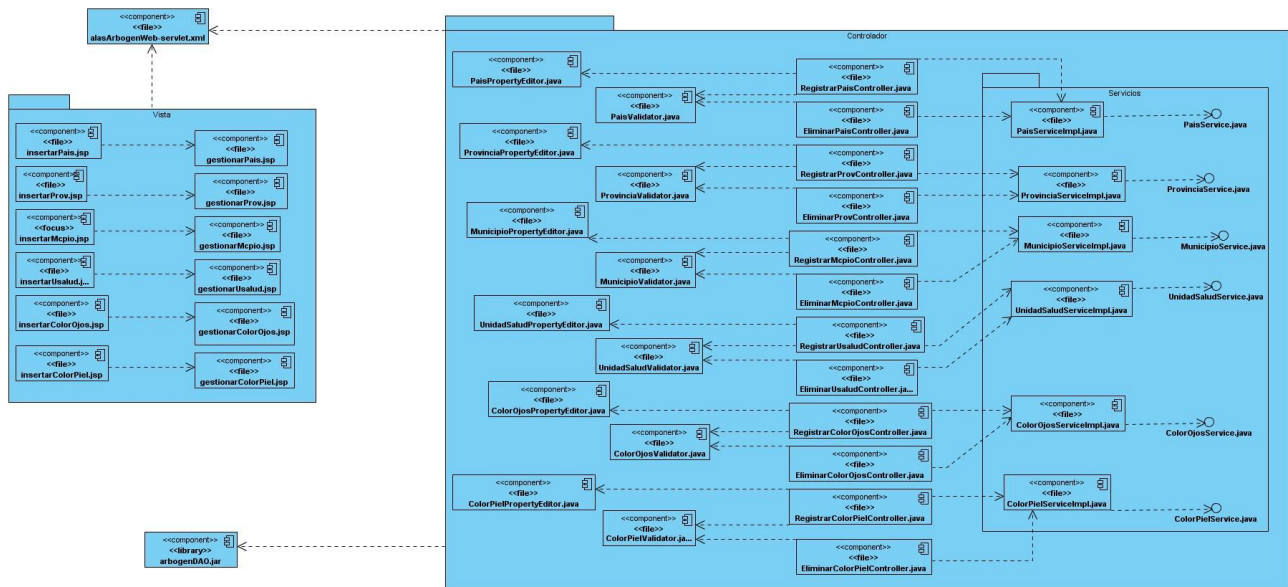


Figura 31: Diagrama de Componentes (CU Gestionar Nomencladores).

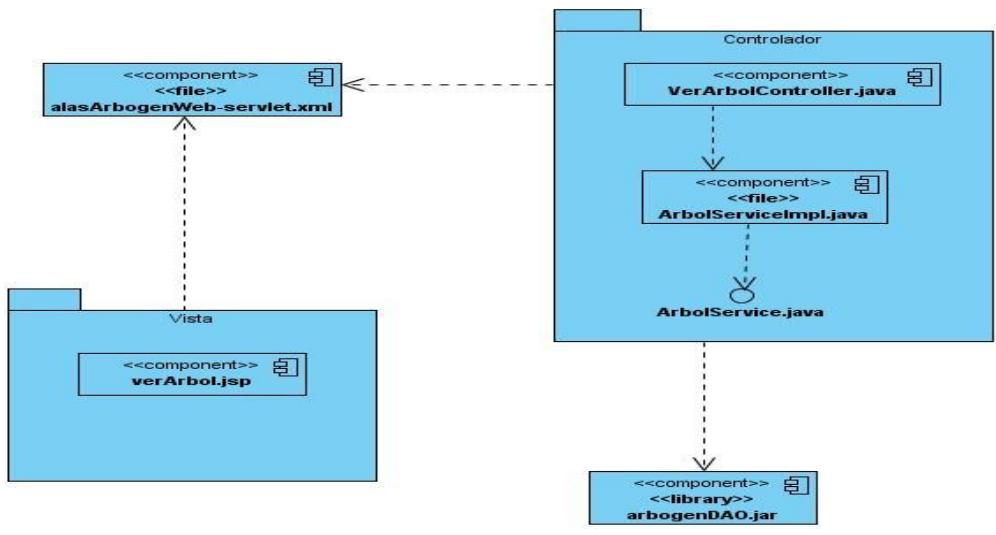


Figura 32: Diagrama de Componentes (CU Visualizar Árbol Genealógico).

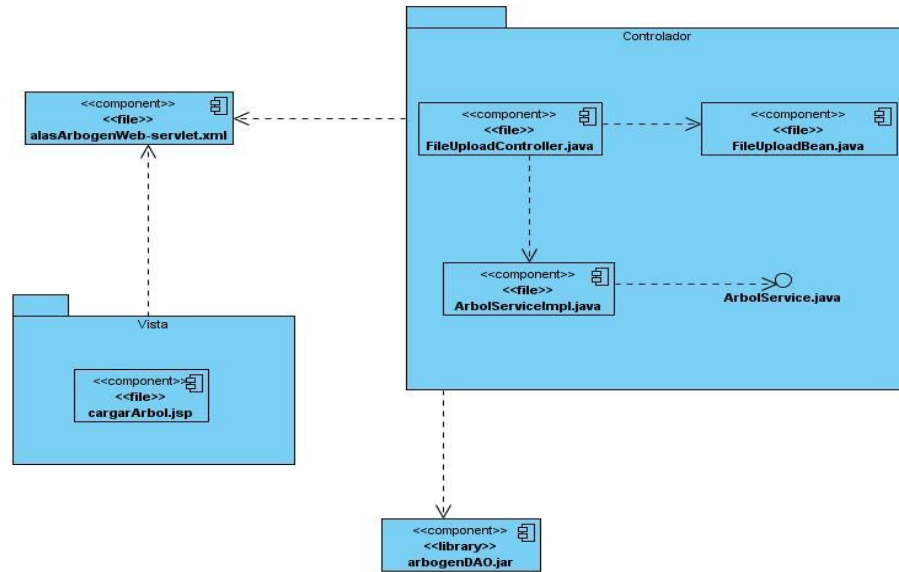


Figura 33: Diagrama de Componentes (CU Cargar Árbol Genealógico).

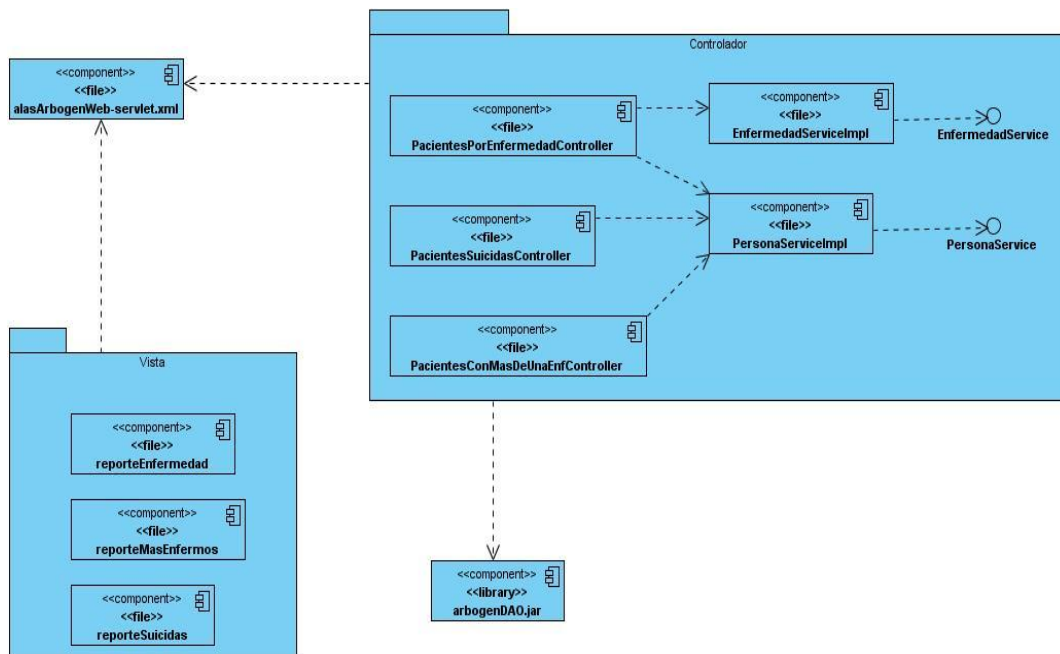


Figura 34: Diagrama de Componentes (CU Generar Reporte para Médico).

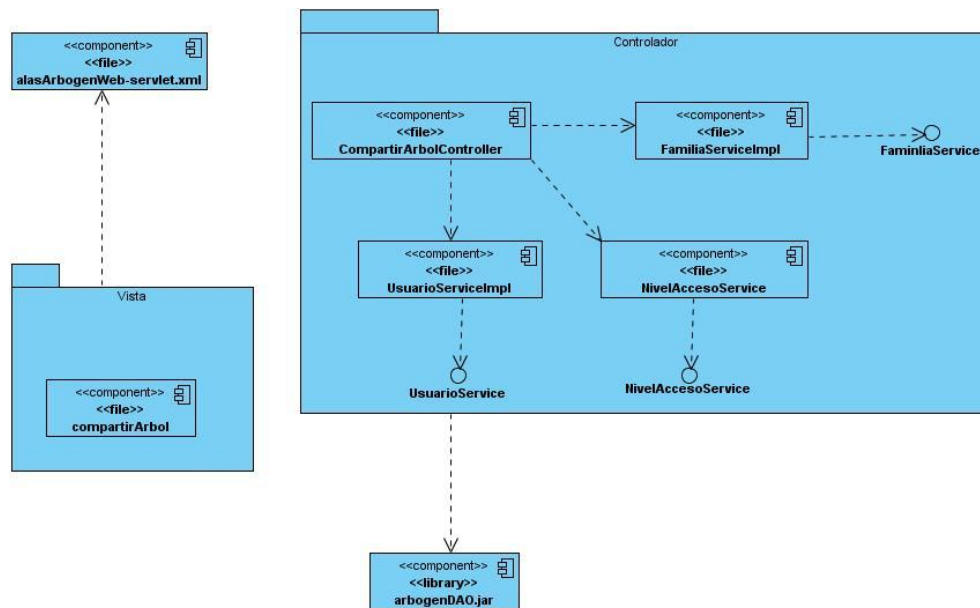


Figura 35: Diagrama de Componentes (CU Compartir Árbol Genealógico).

4.2 PRUEBAS

Las pruebas de software son un elemento crítico para la garantía de la calidad del mismo y representan una revisión final de las especificaciones, el diseño y la codificación. Son una actividad en la cual un sistema o componente de éste es ejecutado bajo condiciones o requerimientos especificados, los resultados se observan y registran, y se hace una evaluación de algún aspecto del sistema o componente. El objetivo de las pruebas de software es descubrir errores. (PRESSMAN, 2005)

4.2.1 Plan de Pruebas.

El plan de pruebas describe las estrategias, recursos y la planificación de las pruebas. La estrategia de prueba incluye la definición del tipo de prueba a realizar por cada iteración y sus objetivos. Para cada prueba se debe especificar en qué consiste la misma, además de los resultados esperados. (PRESSMAN, 2005)

A continuación se muestra el plan de pruebas y el diseño de las pruebas de caja negra para varios casos de uso del sistema.

Capítulo 4 : Implementación y Pruebas

Fecha de Inicio	Fecha de Fin	Actividades	Personal Implicado
26/4/2010	26/4/2010	Aceptación del Plan de Pruebas.	Rosnel Venero. Yoan Manuel Pérez.
28/4/2010	28/4/2010	Verificación de las condiciones previas para iniciar las pruebas.	Rosnel Venero. Yoan Manuel Pérez.
Primera iteración de la aplicación web alasARBOGEN.			
3/5/2010	3/5/2010	Ejecución de las pruebas de caja negra al CU Autenticar Usuario.	Rosnel Venero. Yoan Manuel Pérez.
4/5/2010	4/5/2010	Ejecución de las pruebas de caja negra al CU Gestionar Usuario.	Rosnel Venero. Yoan Manuel Pérez.
7/5/2010	7/5/2010	Ejecución de las pruebas de caja negra al CU Gestionar Nomencladores.	Rosnel Venero. Yoan Manuel Pérez.
22/5/2010	23/5/2010	Ejecución de las pruebas de Caja negra al CU Generar Reporte para Médico.	Rosnel Venero. Yoan Manuel Pérez.
22/5/2010	23/5/2010	Ejecución de las pruebas de Caja negra al CU Generar Reporte para Administrador.	Rosnel Venero. Yoan Manuel Pérez.
23/5/2010	24/5/2010	Ejecución de las pruebas de Caja negra al CU Compartir Árbol Genealógico.	Rosnel Venero. Yoan Manuel Pérez.

Capítulo 4 : Implementación y Pruebas

29/5/2010	30/5/2010	Ejecución de las pruebas de Caja negra al CU Cargar Árbol Genealógico.	Rosnel Venero. Yoan Manuel Pérez.
-----------	-----------	--	--

4.2.2 Diseño de Casos de Pruebas de Caja Negra.

Las pruebas de caja negra (también denominadas pruebas de comportamiento) se centran en los requisitos funcionales del software, se llevan a cabo sobre la interfaz del mismo y permiten obtener un conjunto de condiciones de entrada, que ejerciten completamente todos los requisitos funcionales de un programa. (PRESSMAN, 2005)

Caso de Uso: Autenticar Usuario.

Nombre de la Sección.	Escenarios de la Sección.	Descripción de la Funcionalidad	Flujo Central
SC1: Autenticar Usuario.	ESC1: Autenticar Usuario.	El usuario selecciona en el menú la opción Autenticar. Aparece una interfaz donde debe escribir el usuario y su contraseña, el sistema valida los datos y autentica al usuario.	El usuario oprime la opción Autenticar.

Caso de Uso: Gestionar Usuario.

Nombre de la Sección.	Escenarios de la Sección.	Descripción de la Funcionalidad	Flujo Central
SC1: Registrar Usuario.	ESC1: Registrar Usuario.	El administrador selecciona en el menú Gestionar Usuario la opción Registrar.	EL Administrador oprime la opción Registrar del menú

Capítulo 4 : Implementación y Pruebas

		El sistema muestra una interfaz donde el administrador debe llenar los datos del mismo, dígame Nombre, Contraseña, Rol y oprime el botón Insertar. El sistema valida los datos y registra el usuario en la base de datos.	Gestionar Usuario.
SC2: Modificar Usuario.	ESC2: Modificar Usuario.	El administrador selecciona en el menú Gestionar Usuario la opción Modificar. El sistema muestra una interfaz, listando todos los usuarios existentes en la base de datos. El administrador selecciona el usuario que desea modificar. El sistema muestra una interfaz con los datos del usuario, el administrador modifica los mismos y oprime el botón Editar. El sistema valida los datos y los actualiza en la base de datos.	EL Administrador oprime la opción Modificar del menú Gestionar Usuario.
SC3: Listar Usuarios.	ESC3: Listar Usuarios.	El administrador selecciona en el menú Gestionar Usuario la opción Listar. El sistema muestra un submenú donde el administrador la opción listar usuarios. El sistema muestra una interfaz con los usuarios de la base de datos que sean Administradores.	EL Administrador oprime la opción Listar del menú Gestionar Usuario.
	ESC4: Listar Médicos.	El administrador selecciona en el menú Gestionar Usuario la opción Listar. El sistema muestra un submenú donde el administrador la opción listar médicos. El sistema muestra una interfaz con los usuarios de la base de datos que sean	EL Administrador oprime la opción Listar del menú Gestionar Usuario.

Capítulo 4 : Implementación y Pruebas

		médicos.	
SC4: Eliminar Usuario.	ESC4: Eliminar Usuario.	El administrador selecciona en el menú Gestionar Usuario la opción Eliminar. El sistema muestra una interfaz, listando todos los usuarios existentes en la base de datos. El administrador elimina el usuario que desea.	EL Administrador oprime la opción Eliminar del menú Gestionar Usuario.

Caso de Uso Gestionar Nomencladores.

Nombre de la Sección.	Escenarios de la Sección.	Descripción de la Funcionalidad	Flujo Central
SC1: Registrar Nomenclador.	ESC1: Registrar Nomenclador.	El administrador selecciona en el menú Gestionar Nomencladores el tipo de nomenclador que desea gestionar. El sistema muestra una interfaz con los nomencladores de ese tipo existentes en la base de datos, brindando la opción de Adicionar, Modificar y Eliminar un nomenclador. El administrador oprime el signo de más (+, significa adicionar). El sistema muestra los campos para realizar la acción. El administrador llena los datos y oprime el botón insertar. El sistema valida los datos y finalmente registra el nomenclador en la base de datos.	EL Administrador oprime la opción Adicionar (signo de más '+') en la lista de nomencladores.
SC2: Modificar Nomenclador.	ESC2: Modificar Nomenclador.	El administrador selecciona en el menú Gestionar Nomencladores el tipo de nomenclador que desea gestionar. El	EL Administrador oprime la opción Modificar (lápiz) en la

Capítulo 4 : Implementación y Pruebas

		<p>sistema muestra una interfaz con los nomencladores de ese tipo existentes en la base de datos, brindando la opción de Adicionar, Modificar y Eliminar un nomenclador. El administrador selecciona el nomenclador que desea modificar. El sistema muestra los datos de dicho nomenclador. El administrador modifica los datos y oprime el botón Editar. El sistema valida los datos y actualiza los mismos en la base de datos.</p>	<p>lista de nomencladores.</p>
<p>SC3: Eliminar Nomenclador.</p>	<p>ESC3: Modificar Nomenclador.</p>	<p>El administrador selecciona en el menú Gestionar Nomencladores el tipo de nomenclador que desea gestionar. El sistema muestra una interfaz con los nomencladores de ese tipo existentes en la base de datos, brindando la opción de Adicionar, Modificar y Eliminar un nomenclador. El administrador selecciona el nomenclador que desea eliminar, el sistema muestra un mensaje de notificación y elimina el nomenclador.</p>	<p>EL Administrador oprime la opción eliminar en la lista de nomencladores.</p>

Capítulo 4 : Implementación y Pruebas

Caso de Uso Generar Reporte para Médico.

Nombre de la Sección.	Escenarios de la Sección.	Descripción de la Funcionalidad	Flujo Central
SC: Generar Reporte.	ESC1: Generar Reporte: Pacientes por Enfermedad.	El genetista, en el menú Reportes, selecciona la opción Pacientes Por Enfermedad. El sistema muestra una interfaz, donde el genetista debe escoger la enfermedad para ver el listado de personas que la padecen. Además brinda la opción de imprimir el reporte.	El genetista, en el menú Reporte, selecciona la opción Pacientes Por Enfermedad.
	ESC2: Generar Reporte: Pacientes con Intento de Suicidio.	El genetista, en el menú Reportes, selecciona la opción Pacientes con Intento de Suicidio. El sistema muestra una interfaz con la lista de pacientes que han intentado suicidarse. Además brinda la opción de imprimir el reporte.	El genetista, en el menú Reportes, selecciona la opción Pacientes con Intento de Suicidio.
	ESC3: Generar Reporte: Pacientes que padecen más de una enfermedad.	El genetista, en el menú Reportes, selecciona la opción Pacientes que padecen más de una enfermedad. El sistema muestra una interfaz con la lista de pacientes que han intentado suicidarse. Además brinda la opción de imprimir el reporte.	

Capítulo 4 : Implementación y Pruebas

Caso de Uso Compartir Árbol Genealógico.

Nombre de la Sección.	Escenarios de la Sección.	Descripción de la Funcionalidad	Flujo Central
SC1: Compartir Árbol Genealógico.	ESC1: Compartir Árbol Genealógico.	El genetista selecciona en el menú la opción Compartir Árbol Genealógico. El sistema muestra una interfaz, donde el genetista debe seleccionar el árbol a compartir, a quien va dirigido y oprime el botón compartir.	El genetista oprime la opción Compartir Árbol Genealógico.

Caso de Uso Cargar Árbol Genealógico.

Nombre de la Sección.	Escenarios de la Sección.	Descripción de la Funcionalidad	Flujo Central
SC1: Cargar Árbol Genealógico.	ESC1: Cargar Árbol Genealógico.	El genetista selecciona en el menú la opción Cargar Árbol Genealógico. El sistema muestra una interfaz, donde el genetista debe seleccionar el archivo a cargar.	El genetista oprime la opción Cargar Árbol Genealógico.

Capítulo 4 : Implementación y Pruebas

Caso de Uso Generar reporte para Administrador.

Nombre de la Sección.	Escenarios de la Sección.	Descripción de la Funcionalidad	Flujo Central
SC: Generar Reporte.	ESC1: Generar Reporte de Pacientes.	El Administrador, en el menú Reportes, selecciona la opción Lista de Pacientes. El sistema muestra una interfaz con el listado de pacientes existentes en la base de datos, brindando la posibilidad de filtrarlos por país y provincia.	El genetista, en el menú Reporte, selecciona la opción Listado de Pacientes.
	ESC2: Generar Reporte de Familias.	El Administrador, en el menú Reportes, selecciona la opción Lista de Familias. El sistema muestra una interfaz con el listado de familias existentes en la base de datos, brindando la posibilidad de filtrarlas por país y provincia.	El genetista, en el menú Reportes, selecciona la opción Listado de Familias.

4.2.3 No conformidades detectadas.

Elemento	No conformidad	Aspecto correspondiente	Etapa de detección	Signif	No signif
Página listarUsuarios.jsp	No se actualizaba la lista de usuarios.	Cuando se eliminaba un usuario.	Etapa de aplicación de pruebas al Caso de Uso Gestionar Usuario.	X	
Página	No actualizaba	Cuando se	Etapa de	X	

Capítulo 4 : Implementación y Pruebas

gestionarUsalud.jsp	el comboBox donde se listan los municipios los municipios.	seleccionaba la provincia.	aplicación de pruebas al Caso de Uso Gestionar Nomencladores.		
Página gestionarMcpio.jsp	No actualizaba el comboBox de las Provincias.	Cuando se seleccionaba el país.	Etapa de aplicación de pruebas al Caso de Uso Gestionar Nomencladores.	X	
reporteEnfermedad.jsp	No llenaba el comboBox de las enfermedades.	Cuando se oprimía la opción Pacientes por enfermedad.	Etapa de aplicación de pruebas al Caso de Uso Generar Reporte para Médico.	X	

CONCLUSIONES.

Se implementó la aplicación web capaz de gestionar información de árboles genealógicos. Además de realizaron pruebas exploratorias a la aplicación detectando varias no conformidades, las cuales fueron solucionadas para un mejor funcionamiento del sistema.

CONCLUSIONES

La aplicación que se elaboró como resultado de la presente investigación, a partir de la versión 1.0 de alasARBOGEN, constituye un importante aporte al Centro Nacional de Genética Médica y a toda la red nacional de dicha institución, permitiendo un avance en las investigaciones que ellos llevan a cabo.

- ✓ Se identificaron las funcionalidades de la herramienta informática.
- ✓ Se diseñó la solución utilizando patrones de diseño que mejoran la eficiencia del proceso de codificación.
- ✓ Se desarrolló la aplicación web para el Sistema alasARBOGEN 2.0, capaz de mantener la disponibilidad de los datos gestionados en el mismo.
- ✓ La generación de reportes a través de la aplicación web desarrollada ayuda a realizar estudios en la rama de la genética médica.

RECOMENDACIONES

- ✓ En próximas versiones de la aplicación, consumir servicios del Registro Informatizado de Salud (RIS).
- ✓ Implementar un área para la comunicación y el intercambio entre especialistas.

BIBLIOGRAFÍA

1. Ciberaula [En línea]. - 5 de febrero de 2010. - http://linux.ciberaula.com/articulo/linux_apache_intro/.
2. **Comunicaciones Ministerio de la Informática y las Comunicaciones** [En línea] // Ministerio de la Informática y las Comunicaciones. - 22 de noviembre de 2009. - <http://www.mic.gov.cu/hmicfunciones.aspx..>
3. **Craig Walls Ryan Breidenbach** Spring in Action (Second Edition) [Libro].
4. **Daniele Ing. Marcela** Teoría 11: EL ARTE DE MODELAR. - 2007.
5. El Modelo de Dominio es una representación visual estática del entorno real objeto del proyecto. // Modelo de Dominio . - 2007 .
6. **González Carlos Sánchez** Seguridad no intrusiva con Acegi Security System for Spring [Libro].
7. **González Héctor Suárez** Manual Hibernate [Libro]. - 2003.
8. **Graells Dr. Perez Marquès.** [En línea]. - 22 de febrero de 2010. - <http://www.pangea.orgperemarquestic.net>.
9. **Helena Lorenzo Granizo Javier Murillo Yagüe, Alberto Velázquez Alonso.** Sistema de Control y Comunicación Inalámbrica de Redes Sensoriales.. - Madrid : [s.n.], 2009.
10. <http://www.multimania.es/> [En línea] // <http://www.multimania.es/>. - 17 de abril de 2010. - <http://usuarios.multimania.es/oopere/uml.htm>.
11. Integra soluciones avanzadas [En línea]. - 4 de febrero de 2010. - <http://www.integraas.com/Lanzada-la-version-Europa-de-Eclipse.html>.
12. **Ivar Jacobson Grady Booch, James Rumbaugh** El Proceso Unificado de Desarrollo de Software [Libro]. - Madrid : [s.n.], 2000.
13. javaHispano, Tu lenguaje, tu comunidad. [En línea]. - 1 de septiembre de 2002. - 10 de marzo de 2010. - http://www.javahispano.org/contenidos/es/exprimiendo_javawebstart/.
14. **Juan Medín Piñeiro Antonio García Figueras** Hacia una arquitectura con JavaServer Faces, Spring, Hibernate y otros frameworks.. - Sevilla, España : [s.n.], 2006.
15. Junta de Andalucía [En línea]. - 12 de marzo de 2009. - <http://www.juntadeandalucia.es/xwiki/bin/view/MADEJA/Hibernate#caracteristicas>.

16. Lenguajes de Programación [En línea]. - 11 de enero de 2010. - <http://www.lenguajes-de-programacion.com/lenguajes-de-programacion.shtml>.
17. **Lien Le Sanchez Rynaldo Rosado Rosello** alasARBOGEN: aplicación informática para la representación de árboles genealógicos.. - Ciudad de la Habana, Cuba : [s.n.], 2008.
18. **Lornel A. Rivas María Pérez, Luis E. Mendoza y Anna Grimán.** Herramientas de Desarrollo de Software: Hacia la Construcción de una Ontología [Publicación periódica]. - Maracay, Aragua, Venezuela : [s.n.].
19. Manual de Java [En línea]. - 11 de enero de 2010. - <http://www.manual-java.com/manualjava/caracteristicas-java.html>.
20. **Marinilli Mauro** Java Deployment whit JNLP and WebStart [Libro]. - 2001.
21. **Morán José María Foces** Framework para la capa de presentación de aplicaciones web.. - 2009.
22. **Orallo Enrique Hernández** El Lenguaje Unificado de Modelado (UML).
23. **Pressman Roger S.** Ingeniería de Software: Un enfoque práctico [Libro]. - 2005.
24. **Rico Mario Alfredo Sánchez** Sistema de administración y control de renta de películas y libros vía web utilizando Spring. - Cholula, Puebla, México. : [s.n.], 2006.
25. **Roberth G. Figueroa Camilo J. Solís , Armando A. Cabrera** METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES [Publicación periódica].
26. **Roberth G. Figueroa Camilo J. Solís, Armando A. Cabrera** METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES [Publicación periódica].
27. **Ruiz Juan José González** APLICACIÓN WEB DE UNA EMPRESA DE PRODUCTOS HORTOFRUTÍCOLAS. - Málaga : [s.n.].
28. **Sancho Jesús Bobadilla** Aplicaciones web en cliente: Applets.
29. scribd.com [En línea]. - 21 de enero de 2010. - <http://www.scribd.com/doc/7844685/CONCEPTOS-DE-RUP>.
30. Sitio de descargas de software [En línea]. - 4 de febrero de 2010. - http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.
31. **Sun Microsystems, Inc.** JavaServer Pages™ Standard Tag Library [Libro]. - 2002.

REFERENCIAS BIBLIOGRÁFICAS

- ANDALUCIA., J. D. *Hibernate*. publicado el: 12 de marzo 2010 de última actualización: 12 de marzo 2010. Disponible en: <http://www.juntadeandalucia.es/xwiki/bin/view/MADEJA/Hibernate#caracteristicas>.
- AVANZADAS, I. S. *Lanzada la versión Europa de Eclipse*. publicado el: 4 de febrero de 2010 de 2007, última actualización: 4 de febrero de 2010. Integra Soluciones Avanzadas p. Disponible en: <http://www.integraas.com/Lanzada-la-version-Europa-de-Eclipse.html>.
- CIBERAULA. *Una Introducción a APACHE*. 2010, Disponible en: http://linux.ciberaula.com/articulo/linux_apache_intro/.
- COMUNICACIONES., M. D. L. I. Y. L. *Funciones* Ciudad de la Habana: Disponible en: <http://www.mic.gov.cu/hmicfunciones.aspx>.
- DANIELE, I. M. *Teoría 11: EL ARTE DE MODELAR*. 2007, Disponible en: http://www.google.com/cu/url?sa=t&source=web&ct=res&cd=12&ved=0CD4QFjAL&url=http%3A%2F%2Fante proyecto-protesis-mano-robotica.googlecode.com%2Ffiles%2FTEORIA_11_UML_componentes%2520e%2520interfaces%2520%28buen%25C3%25ADsimo%29.pdf&rct=j&q=diagrama+de+componentes&ei=2jP3S57DI4GC8gbDgpDTCg&usq=AFQjCNEFq9oMBc-EDaBK6FI0Wf_7Yc8Mtg.
- GRAELLS, D. P. M. *LAS TIC Y SUS APORTACIONES A LA SOCIEDAD*. . 2000, Disponible en: <http://www.pangea.org/peremarquestic.net>.
- HELENA LORENZO GRANIZO, J. M. Y., ALBERTO VELÁZQUEZ ALONSO. *Sistema de Control y Comunicación Inalámbrica de Redes Sensoriales*. Facultad de Informática. Universidad Complutense de Madrid., 2009.
- Instituto Nacional del Cáncer* Disponible en: <http://www.cancer.gov/diccionario/?CdrID=44868>.
- IVAR JACOBSON, G. B., JAMES RUMBAUGH. *El Proceso Unificado de Desarrollo de Software*. Madrid: 2000.
- JUAN MEDÍN PIÑEIRO, A. G. F. *Hacia una arquitectura con JavaServer Faces, Spring, Hibernate y otros frameworks*. 2006.
- Lenguajes de Programación*. 2009, [Consultado el: 11 enero de 2010]. Disponible en: <http://www.lenguajes-de-programacion.com/lenguajes-de-programacion.shtml>.
- LIEN LE SANCHEZ, R. R. R. *alasARBOGEN: aplicación informática para la representación de árboles genealógicos*. Universidad de las Ciencias Informáticas., 2008.

Referencias Bibliográficas

LORNEL A. RIVAS, M. P., LUIS E. MENDOZA Y ANNA GRIMÁN. *Herramientas de Desarrollo de Software: Hacia la Construcción de una Ontología*.

Manual de Java. Disponible en: <http://www.manual-java.com/manualjava/caracteristicas-java.html>.

MARIÑÁN, M. P. Exprimiendo Java Web Start. 2002, nº p. javaHispano. Tu lenguaje, tu comunidad. Disponible en: http://www.javahispano.org/contenidos/es/exprimiendo_javawebstart/.

MICROSYSTEMS., S. *Tecnología Java Web Start*. 2002, Disponible en: httpjava.sun.com/javasetechnologiesdesktopjavawebstart1.2esdocsReadme_es.html.

ORALLO., E. H. *El Lenguaje Unificado de Modelado (UML)*. Disponible en: <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>.

PRESSMAN, R. S. *Ingeniería de Software: Un enfoque práctico*. 2005.

RICO., M. A. S. *Sistema de administración y control de renta de películas y libros vía web utilizando Spring*. 2006.

ROBERTH G. FIGUEROA, C. J. S., ARMANDO A. CABRERA. *METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES*. 2008, Disponible en: http://www.google.com.cu/#hl=es&source=hp&q=METODOLOG%C3%8DAS+TRADICIONALES+V+S.+METODOLOG%C3%8DAS+%C3%81GILES.+%2B+Roberth+G.+Figueroa%2C+Camilo+J.+Sol%C3%ADs+%2C+Armando+A.+Cabrera.&btnG=Buscar+con+Google&aq=f&aqi=&aql=&oq=METODOLOG%C3%8DAS+TRADICIONALES+VS.+METODOLOG%C3%8DAS+%C3%81GILES.+%2B+Roberth+G.+Figueroa%2C+Camilo+J.+Sol%C3%ADs+%2C+Armando+A.+Cabrera.&gs_rfai=&fp=961d0ff772c8c24f.

ANEXOS

ANEXO 1: Diseño de pruebas por cada sección del Caso de Uso Gestionar Nomencladores.

A continuación se muestra un ejemplo de nomenclador seleccionado para realizar las pruebas al presente Caso de Uso.

Sección 1: Registrar Nomenclador.

Variable

1- Nombre

Id del escenario	Escenario	V1	Respuesta del sistema	Resultado de la prueba
ESC 1	Registrar nomenclador	V	El sistema valida el dato y registra el nomenclador.	El sistema registra el nuevo nomenclador en la Base de Datos.

Sección 2: Modificar Nomenclador.

Variable:

1- Nombre

Id del escenario	Escenario	V1	Respuesta del sistema	Resultado de la prueba
ESC 2	Modificar nomenclador	V	El sistema valida el dato y modifica el nomenclador.	El sistema modifica el dato del nomenclador en la Base de Datos.

Sección 3: Eliminar Nomenclador.

Variable:

1. País

Id del escenario	Escenario	V1	Respuesta del sistema	Resultado de la prueba
ESC 3	Eliminar nomenclador	V	El sistema elimina el nomenclador país.	El sistema elimina el nomenclador país.

Existe un documento nombrado Materiales Complementarios donde se encuentran los demás anexos del presente trabajo.