**SOLUCIÓN INTEGRAL DE SOFTWARE PARA**

**LA CORPORACIÓN JARDÍN DE LOS ABUELOS**

**ANDRÉS FELIPE SANTOS LOZANO**

**CÓDIGO: 2220071019**

**UNIVERSIDAD DE IBAGUÉ**

**FACULTAD DE INGENIERÍA**

**PROGRAMA DE INGENIERÍA DE SISTEMAS**

**IBAGUÉ - TOLIMA**

**2012**

**SOLUCIÓN INTEGRAL DE SOFTWARE PARA**

**LA CORPORACIÓN JARDÍN DE LOS ABUELOS**

**REALIZADO POR:**

**ANDRÉS FELIPE SANTOS LOZANO**

**CÓDIGO: 2220071019**

**DIRECTOR DE PROYECTO DE GRADO:**

**INGENIERO IVÁN A. DAVID-VENEGAS**

**DOCENTE DEL PROGRAMA INGENIERÍA DE SISTEMAS**

**UNIVERSIDAD DE IBAGUÉ**

**FACULTAD DE INGENIERÍA**

**PROGRAMA DE INGENIERÍA DE SISTEMAS**

**IBAGUÉ - TOLIMA**

**2012**

**NOTA DE ACEPTACIÓN**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Firma del Presidente del Jurado**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Firma del jurado**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Firma del jurado**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Firma del jurado**

**A Dios y a su divina providencia por la realización de este trabajo de grado. A mi familia por su apoyo constante durante mi tiempo de estudio.**

**AGRADECIMIENTOS**

El autor expresa sus agradecimientos a:

Mi Madre María Daissy Lozano Jiménez.

Mi padre Humberto Santos Salcedo.

A mi hermana Luz Elena y a mi hermano Diego Fernando.

Al Ingeniero Iván Augusto David-Venegas Director de mi Proyecto de Grado, por su constante apoyo y valiosas orientaciones. A todo el equipo docente del Programa de Ingeniería de Sistemas de la Universidad de Ibagué, por el conocimiento y las enseñanzas que me han aportado.

**TABLA DE CONTENIDO**

|  |  |
| --- | --- |
|  | **Pág.** |
| **1. INTRODUCCIÓN** | **13** |
| **2. PROBLEMA** | **14** |
| **3. OBJETIVOS** | **15 - 17** |
| **3.1 Objetivo General** | **15** |
| **3.2 Objetivos Específicos** | **15** |
| **3.3 Alcance** | **15** |
| **3.4 Limitaciones** | **17** |
| **4. JUSTIFICACIÓN** | **18** |
| **5. ANTECEDENTES** | **19** |
| **6. MARCO TEÓRICO** | **20 - 27** |
| **7. MARCO CONCEPTUAL** | **28 - 29** |
| **8. METODOLOGÍA** | **30 - 31** |
| **8.1 Análisis de Requerimientos** | **30** |
| **8.2 Análisis del Sistema** | **30** |
| **8.3 Diseño del Sistema** | **31** |
| **8.4 Implementación del Sistema** | **31** |
| **8.5 Pruebas del Sistema** | **31** |
| **9. ENTREGABLES** | **32 - 67** |
| **9.1 Análisis de Requerimientos** | **32** |
| **9.1.1 RF Requerimientos Funcionales** | **32** |
| **9.1.1.1 Tipos de Cuenta** | **32** |
| **9.1.1.2 Funcionario Administrativo** | **33** |
| **9.1.1.3 Trabajadora Social** | **33** |
| **9.1.1.4 EPS** | **33** |
| **9.1.1.5 Adulto Mayor** | **34** |
| **9.1.1.6 Inscripciones** | **34** |
| **9.1.1.7 Fotos** | **34** |
| **9.1.1.8 Aportes** | **34** |
| **9.1.1.9 Padrino** | **35** |
| **9.1.1.10 Donaciones** | **35** |
| **9.1.1.11 Informes** | **35** |
| **9.1.1.12 Usuarios** | **35** |
| **9.1.2 RNF Requerimientos No Funcionales** | **36** |
| **9.1.2.1 Presentación** | **36** |
| **9.1.2.2 Arquitectura** | **36** |
| **9.1.2.3 Seguridad** | **36** |
| **9.1.2.4 No Funcionales Organizacionales** | **36** |
| **9.1.3 DP Dominio del Problema** | **37** |
| **9.2 Análisis del Sistema** | **38** |
| **9.2.1 AG Arquitectura General** | **38** |
| **9.2.2 Descripción de los Casos de Uso** | **39** |
| **9.2.3 DCU Diagrama de Casos de Uso** | **50** |
| **9.2.4 ECU Escenarios de Casos de Uso** | **50** |
| **9.2.5 MCA Modelo de Clases de Análisis** | **50** |
| **9.2.6 DSA Diagramas de Secuencia de Análisis** | **50** |
| **9.3 Diseño del Sistema** | **51** |
| **9.3.1 DI Diagrama de Infraestructura** | **51** |
| **9.3.2 DC Diagrama de Componentes** | **52** |
| **9.3.3 DCD Diagrama de Clases de Diseño** | **53** |
| **9.3.4 DD Diagrama de Despliegue** | **54** |
| **9.3.5 DIN Diagramas de Interacción** | **55** |
| **9.3.6 DRE Diagrama Relacional** | **55** |
| **9.3.7 DCS Diagrama de Clases de Software** | **56** |
| **9.3.8 Diseño de Base de Datos** | **56** |
| **9.3.9 Diseño de Interfaces** | **58** |
| **9.3.9.1 Página de Inicio** | **59** |
| **9.3.9.2 Menús del Sistema** | **60** |
| **9.4 Implementación del Sistema** | **63** |
| **9.4.1 CF Código Fuente** | **63** |
| **9.4.2 FBD Fichero de Base de Datos** | **63** |
| **9.4.3 MU Manual de Usuario** | **63** |
| **9.5 Pruebas del Sistema** | **64** |
| **9.5.1 PPE Plan de Pruebas de Eficiencia** | **64** |
| **9.5.1.1 Pruebas Transaccionales** | **64** |
| **9.5.1.2 Pruebas Unitarias** | **66** |
| **9.5.1.3 Pruebas de Robustez** | **67** |
| **10. RESUMEN DEL PROYECTO** | **68** |
| **CONCLUSIONES** | **69 - 70** |
| **RECOMENDACIONES** | **71** |
| **ANEXOS** | **71** |
| **BIBLIOGRAFÍA** | **72 - 73** |

**ÍNDICE DE TABLAS**

|  |  |
| --- | --- |
|  | **Pág.** |
| **Tabla 1. Descripción del caso de uso Ingresar al Sistema** | **39** |
| **Tabla 2. Descripción de los casos de uso incluidos en Gestionar EPS** | **40** |
| **Tabla 3. Descripción de los casos de uso incluidos en Gestionar Inscripciones** | **41** |
| **Tabla 4. Descripción de los casos de uso incluidos en Gestionar Adulto Mayor** | **42** |
| **Tabla 5. Descripción del caso de uso incluido en Gestionar Fotos** | **44** |
| **Tabla 6. Descripción de los casos de uso incluidos en Gestionar Aportes** | **44** |
| **Tabla 7. Descripción de los casos de uso incluidos en Gestionar Padrino** | **45** |
| **Tabla 8. Descripción de los casos de uso incluidos en Gestionar Donaciones** | **47** |
| **Tabla 9. Descripción del caso de uso incluido en Gestionar Informes** | **48** |
| **Tabla 10. Descripción de los casos de uso incluidos en Gestionar Usuarios** | **49** |
| **Tabla 11. Descripción de tablas de la base de datos** | **57** |
| **Tabla 12. Pruebas transaccionales aproximadamente al minuto de ejecución** | **65** |
| **Tabla 13. Pruebas transaccionales aproximadamente a los 5 minutos de ejecución** | **65** |
| **Tabla 14. Pruebas transaccionales de datos complementarios** | **66** |
| **Tabla 15. Resultados de pruebas unitarias** | **66** |
| **Tabla 16. Pruebas de robustez servidor local** | **67** |
| **Tabla 17. Pruebas de robustez servidor remoto** | **67** |
| **Tabla 18. Resumen del proyecto** | **68** |

**ÍNDICE DE FIGURAS**

|  |  |
| --- | --- |
|  | **Pág.** |
| **Figura 1. Cronograma de actividades** | **30** |
| **Figura 2. Arquitectura Cliente/Servidor** | **38** |
| **Figura 3. Diagrama de caso de uso Ingresar al Sistema** | **39** |
| **Figura 4. Diagrama de caso de uso Gestionar EPS** | **40** |
| **Figura 5. Diagrama de caso de uso Gestionar Inscripciones** | **41** |
| **Figura 6. Diagrama de caso de uso Gestionar Adulto Mayor** | **43** |
| **Figura 7. Diagrama de caso de uso Gestionar Fotos** | **44** |
| **Figura 8. Diagrama de caso de uso Gestionar Aportes** | **45** |
| **Figura 9. Diagrama de caso de uso Gestionar Padrino** | **46** |
| **Figura 10. Diagrama de caso de uso Gestionar Donaciones** | **47** |
| **Figura 11. Diagrama de caso de uso Gestionar Informes** | **48** |
| **Figura 12. Diagrama de caso de uso Gestionar Usuarios** | **49** |
| **Figura 13. Diagrama general de casos de uso del sistema** | **50** |
| **Figura 14. DI Diagrama de Infraestructura** | **51** |
| **Figura 15. DC Diagrama de Componentes** | **52** |
| **Figura 16. DCD Diagrama de Clases de Diseño** | **53** |
| **Figura 17. DD Diagrama de Despliegue** | **54** |
| **Figura 18. DRE Diagrama Relacional** | **55** |
| **Figura 19. Logo Corporación Jardín de los Abuelos** | **59** |
| **Figura 20. Logo Sistema de Información y Universidad de Ibagué** | **59** |
| **Figura 21. Página de Inicio SIADM** | **60** |
| **Figura 22. Menú principal SIADM para usuario con nivel de acceso 1** | **61** |
| **Figura 23. Menú principal SIADM para usuario con nivel de acceso 2** | **61** |
| **Figura 24. Menú principal SIADM para usuario con nivel de acceso 3** | **62** |

**1. INTRODUCCIÓN**

La Corporación Jardín de los Abuelos, como entidad líder en la atención integral en un ambiente humano-espiritual a la persona mayor, comprometida con el fomento y la difusión del respeto a su dignidad; pretende encontrarse a la vanguardia en cuanto a cambios tecnológicos y de innovación, usando tecnologías computacionales para así agilizar los procesos propios de su funcionamiento, afianzando la confiabilidad de sus actividades.

El Jardín de los Abuelos ha sido exaltado con la distinción Mercurio de Oro por FENALCO Seccional Tolima, por ser una entidad que ha cumplido la maravillosa labor de devolver una esperanza de vida a los adultos mayores y de escasos recursos, brindándoles un hogar, amor y fraternidad, siendo una organización que participa en la construcción de un mejor país.

La implementación de un sistema de información para la Corporación Jardín de los Abuelos, como herramienta de control en el registro de las inscripciones y apadrinamiento del adulto mayor, es imprescindible para el correcto y eficaz desarrollo de las actividades y servicios prestados por esta entidad; ya que para una organización, contar con la información correcta y en el momento en que se necesita es de vital importancia a la hora de tomar decisiones.

**2. PROBLEMA**

La Corporación Jardín de los Abuelos, no tenía a su disposición un Portal Web, que le sirviera como medio para darse a conocer y llegar de una forma más amena a la comunidad en general; en el proceso de práctica empresarial del semestre B de 2011, se dio inicio al desarrollo y gestión de dicho portal.

De igual forma tampoco cuenta con un sistema de información que de soporte a los procesos de registro de inscripciones y apadrinamiento del adulto mayor, el cual permita realizar una gestión más eficiente de la información y que ayude agilizar el servicio al público.

Actualmente el manejo de la información es tedioso lo que desencadena una demora visible en la entrega de información, incluso pueden existir vacios, pérdidas y errores en la información brindada, representando problemas de tiempo en la gestión que realiza el Jardín de los Abuelos, dando la posibilidad de generar una imagen negativa de esta entidad.

Por ende, haciendo uso de las tecnologías de información, se propone desarrollar un sistema de información para el registro y control de los datos, el cual permita su administración y sea de fácil acceso para los funcionarios del Jardín de los Abuelos.

**3. OBJETIVOS**

**3.1 Objetivo General**

Desarrollar un sistema de información, usando tecnologías de la información (TIC), para el registro de inscripciones y apadrinamiento del adulto mayor de la Corporación Jardín de los Abuelos.

**3.2 Objetivos Específicos**

* Modelar, diseñar e implementar una aplicación, siguiendo los patrones de diseño UML para el análisis, y basando el proceso en el Plan Unificado de Desarrollo PDU.
* Diseñar e implementar el esquema de persistencia, que permita administrar la información requerida por el Jardín de los Abuelos en cuanto al proceso de inscripción y apadrinamiento del adulto mayor.
* Añadir un catálogo de productos y un módulo para el apadrinamiento del adulto mayor al Portal Web del Jardín de los Abuelos.

**3.3 Alcance**

En el proceso de desarrollo del sistema de información para el registro de las inscripciones y apadrinamiento del adulto mayor del Jardín de los Abuelos, se tuvieron en cuenta los siguientes lineamientos:

* Se realizó el Análisis y Diseño del sistema de información, basado en los patrones de diseño UML y parámetros del PDU.
* Teniendo en cuenta las etapas de desarrollo que se aplicaron en torno a este proyecto, se desarrollaron los siguientes módulos:
  + **Módulo EPS:** Éste módulo permite registrar la información de la Entidad Prestadora de Salud a la cual está afiliado el adulto mayor (nombre de la EPS y el régimen). Ofrece la posibilidad de modificar la información cuando sea necesario, por ejemplo: corregir datos erróneos, además de agregar un nuevo régimen a una determinada EPS. Adicionalmente, permite consultar a los afiliados que posee una EPS específica.
  + **Módulo Inscripciones:** Éste módulo permite gestionar las inscripciones de los adultos mayores (aprobar o denegar). Adicionalmente, permite asignar el módulo donde residirá el adulto mayor durante su estancia en el Jardín de los Abuelos.
  + **Módulo Adulto Mayor:** Éste módulo permite registrar la información del perfil de los adultos mayores (datos personales, datos de la inscripción y patologías, composición familiar y responsable). Cuenta con las opciones para ver el perfil del adulto mayor, ver el perfil público para el proceso de apadrinamiento, modificar el perfil si se han agregado datos erróneos, ver que padrinos tiene, los aportes realizados por el responsable entre otras funcionalidades.
  + **Módulo Fotos:** Este módulo permite cargar la foto de cada adulto mayor para ser visualizada en su respectivo perfil.
  + **Módulo Aportes:** Este módulo permite registrar los aportes que efectúan los responsables. Ofrece la posibilidad de realizar una búsqueda avanzada, filtrando la información por diferentes parámetros de búsqueda (datos del responsable, módulo, rango de fechas y rango de montos).
  + **Módulo Padrino:** Éste módulo permite registrar la información del perfil de los padrinos (datos personales). Cuenta con las opciones para ver el perfil del padrino, modificar el perfil si se han agregado datos erróneos, ver los aportes y donaciones realizadas y visualizar cuáles son sus apadrinados.
  + **Módulo Donaciones:** Este módulo permite registrar las donaciones que realizan los padrinos. Cuenta con la opción de visualizar de forma detallada los datos de la donación y así diferenciar si fue una donación para el Jardín de los Abuelos o para un adulto mayor. Ofrece la posibilidad de realizar una búsqueda avanzada, filtrando la información por diferentes parámetros de búsqueda (datos del padrino y datos de la donación).
  + **Módulo Informes:** Este módulo permite realizar algunas consultas básicas sobre el estado actual de los datos. Estos informes son generados mediante archivos planos (.csv), ordenados en tablas con diversa información, por ejemplo: los datos de los adultos mayores, las EPS y sus afiliados, entre otros.
  + **Módulo Usuarios:** Este módulo permite gestionar todo lo concerniente a las cuentas de usuario, como ingresar los datos de cada tipo de usuario, modificar y consultar información detallada como la actividad realizada en el sistema.

Fue añadido al Portal Web del Jardín el siguiente componente:

* + **SimpleCaddy:** Este componente sirve como catálogo y permite adicionar los diferentes productos que ofrezca el Jardín de los Abuelos a la comunidad.

**3.4 Limitaciones**

* El desarrollo de esta solución integral, tanto para el Sistema de Información como para Portal Web, comprende los módulos ya especificados, cualquier extensión, o adición de requerimientos, no estarán comprendidos en esta etapa de desarrollo.
* Una vez terminadas satisfactoriamente las fases implementación y pruebas, no se tendrá comprendida ninguna actualización, soporte o mantenimiento del mismo.

**4. JUSTIFICACIÓN**

Con la intención de satisfacer la intención y necesidad de la Corporación Jardín de los Abuelos en continuar creciendo, y ofrecer un mejor servicio al público; esta propuesta está enfocada en incorporar herramientas tecnológicas que les faciliten adaptarse a los cambios tecnológicos que el mundo demanda.

Este hogar geriátrico se sostiene con las donaciones de la comunidad Ibaguereña, con el pago de la mensualidad de los adultos mayores pensionados, con actividades organizadas por la Institución y por convenios interadministrativos con la Alcaldía Municipal y el Departamento.

La gestión del Portal Web y el diseño e implementación de un Sistema de Información, haciendo uso de las tecnologías computacionales, garantiza la optimización de sus procesos para el correcto y eficaz desarrollo de las actividades que actualmente realizan de forma manual; de igual forma asegurar la integridad de la información que maneja esta entidad, dando trámite de manera ágil y eficaz a las solicitudes administrativas y de servicio al público que se requieran.

**5. ANTECEDENTES**

El 23 de agosto de 1977 nació la Corporación Jardín de los Abuelos, empresa de desarrollo social, sin ánimo de lucro. Para el año de 1985 acogió a los adultos mayores del antiguo Asilo Municipal y recibió el grupo de adultos mayores que por la tragedia de Armero, habían quedado sin familia, ni hogar.

El 75% de la población atendida en el Jardín no cuenta con familia y si la tiene es de bajos recursos. La mayoría de las personas mayores se encuentran en las instalaciones gracias a la caridad de la propia entidad y los aportes en dinero para su sostenimiento por parte de los adultos mayores pensionados, los cuales son un número bastante reducido, y los aportes voluntarios de familiares.

Asimismo se está implementado el Plan Padrino, diseñado con el fin de beneficiar a las personas adultas del Jardín que no cuentan con recursos económicos que garanticen el cubrimiento de sus necesidades, el cual contribuye al proceso de cambio social que instituciones como ésta entidad bridan en beneficio de la población menos favorecida del Departamento.

En la actualidad existen 3 maneras en la cual el adulto mayor se inscribe en el Jardín de los Abuelos. La primera opción es la remisión del adulto mayor por parte de otra entidad, proceso en el cual se realiza un estudio para determinar si es viable el traslado del adulto mayor. De igual forma existe una remisión realizada directamente por las entidades afiliadas al Departamento que velan por la dignidad del adulto mayor, y por último la opción de vinculación directa que se realiza mediante una petición al Jardín por parte de los familiares del adulto mayor, para todos los casos se requiere el conceso de la persona mayor. Cabe resaltar que los procesos anteriormente mencionados se realizan de forma manual y sin el debido estándar de seguimiento a los procesos.

**6. MARCO TEÓRICO**

**Software**

La siguiente descripción para software la describe el autor Ian Sommerville:

“Muchas personas asocian el término *software* con los programas de computadora. Sin embargo, yo prefiero una definición más amplia donde el software no son sólo programas, sino todos los documentos asociados y la configuración de datos que se necesitan para hacer que estos programas operen de manera correcta.” [[1]](#footnote-1)

**Sistema de Información**

Un sistema de información se describe como un conjunto de componentes interrelacionados que reúne (u obtiene), procesa, almacena y distribuye información para apoyar la toma de decisiones y el control en una organización. Las actividades del sistema de información son: entrada, procesamiento y salida de datos.

**Proceso Unificado de Desarrollo**

El Proceso Unificado de Desarrollo de software es una metodología que describe una serie de etapas para realizar aplicaciones o programas. Inicia con la exploración del concepto y termina cuando el producto está completamente terminado. Durante este periodo, el producto pasa por una serie de pasos como son los requerimientos, análisis, diseño, implementación e integración.

El Proceso Unificado usa el Lenguaje Unificado de Modelado (UML) en la preparación de todos los planos del sistema. De hecho, UML es una parte integral del Proceso Unificado, fueron desarrollados a la par.

**Lenguaje Unificado de Modelado**

Craig Larman presenta la siguiente definición:

“El Lenguaje Unificado de Modelado (UML por sus siglas en inglés) es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un plano del sistema, incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.” [[2]](#footnote-2)

**Servidor Apache**

Es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo.

Según definición de Miguel Ángel Palomares:

“Su nombre se debe a que Behelendorf quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo, y la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de EEUU, y en esos momentos la preocupación de su grupo era que llegasen las empresas y civilizasen el paisaje que habían creado los primeros ingenieros de internet. Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, a patchy server un servidor parcheado.” [[3]](#footnote-3)

**Netbeans**

Es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

Además es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo.

**PowerDesigner**

La herramienta de modelamiento número uno de la industria, permite a las empresas, de manera más fácil, visualizar, analizar y manipular metadatos, logrando un efectiva arquitectura empresarial de información.

Federico Beigbeder enuncia:

“PowerDesigner para Arquitectura Empresarial también brinda un enfoque basado en modelos, el cual permite alinear al negocio con la tecnología de información, facilitando la implementación de arquitecturas efectivas de información empresarial. Brinda potentes técnicas de análisis, diseño y gestión de metadatos a la empresa.” [[4]](#footnote-4)

**PHP**

Mehdi Achour, Friedhelm Betz, Antony Dovgal y muchos otros señalan:

“PHP, acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje "Open Source" interpretado de alto nivel, especialmente pensado para desarrollos web y el cual puede ser incrustado en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl y es fácil de aprender. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil, aunque se pueda hacer mucho más con PHP.” [[5]](#footnote-5)

**HTML**

HTML, siglas de HyperText Markup Language («lenguaje de marcado de hipertexto»), hace referencia al lenguaje de marcado predominante para la elaboración de páginas web.

Se utiliza para describir y traducir la estructura y la información en forma de texto, así como para complementar el texto con objetos tales como imágenes. El HTML se escribe en forma de «etiquetas», rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

Ángel Cobo, Patricia Gómez, Daniel Pérez y Rocío Rocha enuncian:

“En resumen, es un lenguaje de descripción de hipertexto compuesto por una serie de comandos, marcas o etiquetas, también denominadas "Tags" que permiten definir la estructura lógica de un documento web y establecer los atributos del mismo (color del texto contenidos multimedia, hipervínculos, etc.).” [[6]](#footnote-6)

**JavaScript**

Miguel Ángel Sánchez dice:

“JavaScript se presenta como un lenguaje de desarrollo de aplicaciones, Cliente/Servidor a través de internet.” [[7]](#footnote-7)

Es un lenguaje de programación que puede ser utilizado por profesionales y para quienes se inician en el desarrollo y diseño de sitios web. No requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos. Se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

**Arquitectura Cliente/Servidor**

Esta arquitectura consiste básicamente en un cliente que realiza peticiones a otro programa (el servidor) que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema. La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa.

Ian Sommerville dice:

“Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.” [[8]](#footnote-8)

**Bases de Datos Relacionales**

Una base de datos relacional es una base de datos en donde todos los datos visibles al usuario están organizados estrictamente como tablas de valores, y en donde todas las operaciones de la base de datos operan sobre dichas tablas, estas bases de datos son percibidas por los usuarios como una colección de relaciones normalizadas de diversos grados que varían con el tiempo.

El modelo relacional representa un sistema de bases de datos en un nivel de abstracción un tanto alejado de los detalles de la máquina subyacente, de la misma manera como, por ejemplo, un lenguaje del tipo de PL/1 representa un sistema de programación con un nivel de abstracción un tanto alejado de los detalles de la máquina subyacente.

Fray Osorio Rivera indica:

“De hecho, el modelo relacional puede considerarse como un lenguaje de programación más bien abstracto, orientado de manera específica hacia las aplicaciones de bases de datos.” [[9]](#footnote-9)

**MySQL**

Sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB.

A propósito de MYSQL, Ángel Cobo, Patricia Gómez, Daniel Pérez y Rocío Rocha expresan:

“MySQL fue escrito en C y C++ y se destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en diferentes sistemas operativos. También es muy destacable, que permite "crear la base de datos y los diferentes objetos que la componen, por ejemplo tablas, vistas, índices, etc. La sentencia creada para la creación de objetos es CREATE. La creación de un objeto requiere la asignación de un nombre que lo identifique una vez creado, cualquier referencia posterior a él hace uso de ese identificador.” [[10]](#footnote-10)

**Framework**

En el desarrollo de software, un Framework o infraestructura digital, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

**CodeIgniter**

CodeIgniter es un robusto Framework de desarrollo de aplicaciones, construido para los programadores de PHP que necesitan un conjunto de herramientas simples y selectas para crear todas las funciones de las aplicaciones web.

**Modelo – Vista – Controlador**

CodeIgniter está basado en el patrón de desarrollo Modelo-Vista-Controlador. MVC es un enfoque de software que separa la lógica de la aplicación de la presentación. En la práctica, le permite a las páginas web contener mínimo código ya que la presentación está separada del código PHP.

• El Modelo representa las estructuras de datos. Típicamente las clases del modelo contendrán funciones que ayudan a devolver, insertar y actualizar información de la base de datos.

• La Vista es la información que se presenta al usuario. Una vista será normalmente una página web, pero en CodeIgniter, una vista también puede ser un fragmento de página como el encabezado o pie de página. También puede ser una página RSS, o cualquier otro tipo de "página".

• El Controlador sirve como un intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para procesar la solicitud HTTP y generar una página web.

**jQuery**

jQuery es una librería de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

**jQuery UI**

jQuery UI es una librería de componentes para el Framework jQuery, que contiene un conjunto de plug-ins, widgets y efectos visuales para la creación de aplicaciones web.

**DataTables**

Es una herramienta jQuery muy flexible, basada en los fundamentos de la mejora progresiva, lo que añadirá controles avanzados de interacción para cualquier tabla HTML. Permite darle dinamismo a las tablas de una forma sencilla aumentando su vistosidad.

**960 Grid System**

960 Grid System es un Framework de [CSS](http://es.wikipedia.org/wiki/CSS) enfocado a la maquetación de una página web. Es un sistema de rejilla para hacer páginas con 960 píxeles de ancho. Las columnas que se pueden colocar en la rejilla tendrán distintas anchuras, pero siempre el ancho total de la página será de 960 píxeles. Se ha elegido este valor porque 960 es divisible por una buena cantidad de números, lo que lo hace más versátil para poder alcanzar páginas resultantes de la más variada gama.

**SimpleCaddy**

SimpleCaddy es un componente que brinda una solución de carrito de compras simple pero potente para Joomla! 1.5. Utiliza contenido estándar para mostrar los detalles de los productos y un pequeño plug-in en esa página para presentar la funcionalidad "Añadir al Carro". Los productos tienen que ser creados en la sección de administración del componente y luego se pueden utilizar para la venta. Cuenta con un módulo para ver qué productos contiene el carro de compras.

**7. MARCO CONCEPTUAL**

En el desarrollo del proyecto, se empleo un lenguaje de programación libre como lo es PHP, que es una herramienta robusta para el desarrollo de aplicaciones web, que permite la construcción de páginas dinámicas y eficientes en el momento de su implementación, además de trabajar conjuntamente con un Framework como lo es CodeIgniter.

CodeIgniter es un conjunto de herramientas para personas que construyen aplicaciones web usando PHP. Su objetivo es permitir desarrollar proyectos mucho más rápido que lo que podría hacer si escribiera código desde cero, al proveer un amplio y valioso conjunto de bibliotecas para tareas comúnmente necesarias, tanto como una interfaz sencilla y una estructura lógica para acceder a esas bibliotecas. CodeIgniter permite enfocarse creativamente en su proyecto al minimizar la cantidad de código necesario para una tarea dada, además cuenta con un motor robusto para plantillas.

**¿Por qué se utilizó CodeIgniter para el desarrollo de éste proyecto?**

* **CodeIgniter es Libre**

CodeIgniter está liberado bajo licencias open source del estilo Apache/BSD, así que puede usarlo donde desee. Para mayor información lea el acuerdo de licencia en el siguiente enlace: http://codeigniter.com/user\_guide/license.html

* **CodeIgniter es Liviano**

Es realmente liviano. El núcleo del sistema sólo requiere algunas bibliotecas muy pequeñas. Esto está en marcado contraste con muchos Frameworks que requieren muchos más recursos. Las bibliotecas adicionales se cargan dinámicamente bajo pedido, basado en sus necesidades para un proceso dado, por lo que el sistema base es muy ligero y bastante rápido.

* **CodeIgniter usa M-V-C**

CodeIgniter usa el enfoque Modelo-Vista-Controlador, que permite una separación entre la lógica y la presentación. Es particularmente bueno para proyectos en los que los diseñadores trabajan en sus archivos de plantillas, ya que el código en estos archivos será mínimo.

* **CodeIgniter genera URLs Claras**

Las URLs generadas por CodeIgniter son claras y amigables con los motores de búsqueda. En lugar de usar el enfoque estándar "query string" característico de sistemas dinámicos, CodeIgniter usa el enfoque basado en segmentos.

* **CodeIgniter es Extensible**

El sistema se puede extender fácilmente a través de sus propias bibliotecas, helpers, extensiones de clases o sistema de hooks.

El motor de base de datos que se ha seleccionado para este proyecto es MySQL, el cual es un sistema gestor de bases de datos muy conocido y ampliamente usado por su simplicidad y notable rendimiento. Es una opción atractiva tanto para aplicaciones comerciales, como libres; precisamente por su facilidad de uso y tiempo reducido de puesta en marcha. Esto y su libre distribución le otorgan como beneficios adicionales (no menos importantes) contar con un alto grado de estabilidad, robustez y rápido desarrollo.

**8. METODOLOGÍA**

El Proceso Unificado es un marco de desarrollo de software que se caracteriza por estar dirigido por casos de uso y que junto a un Paradigma Orientado a Objetos forman una sólida metodología, constantemente utilizada para el Análisis, Diseño e Implementación de un sistema de información, garantizando un buen desarrollo de software, ya que dicha metodología funciona de forma iterativa e incremental.

Se ha estimado un tiempo mínimo de 24 semanas (6 meses) para el desarrollo de éste trabajo de grado, y se ha diseñado un cronograma de actividades donde están descritas de forma general las etapas para llevar a cabo el proyecto con sus respectivos tiempos estimados.

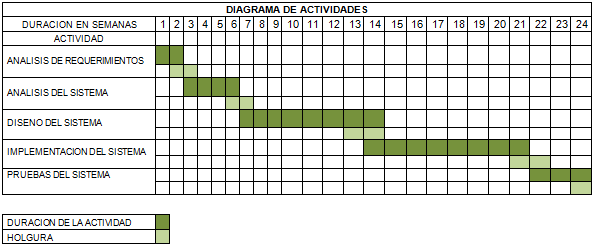


Figura 1. Cronograma de actividades.

**8.1 Análisis de Requerimientos**

* Obtención de requerimientos, tanto funcionales como no funcionales.
* Análisis de datos de entrada y de salida en los procesos que se llevan a cabo en el Jardín de los Abuelos.

**8.2 Análisis del Sistema**

* Análisis de Casos de Uso (diagramas y descripción de escenarios).
* Análisis funcional del sistema (clases de tipo borde, control y entidad).
* Análisis de la arquitectura del sistema.

**8.3 Diseño del Sistema**

* Diseño de la Base de Datos Relacional del Sistema de Información.
* Diseño de la interfaz gráfica de usuario del Sistema de Información con los módulos especificados.

**8.4 Implementación del Sistema**

* Desarrollo del código fuente del Sistema de Información.
* Implementación del Sistema de Información en el Jardín de los Abuelos.

**8.5 Pruebas del Sistema**

* Pruebas de caja negra y caja blanca.
* Correcciones e Iteraciones del Sistema de Información.

**9. ENTREGABLES**

Los archivos y documentos que serán entregados como soporte final del Sistema de Información una vez terminadas las etapas de desarrollo son:

**9.1 Análisis de Requerimientos**

Para realizar el análisis de requerimientos, se llevaron a cabo reuniones periódicas para la identificación de los procesos, funcionalidades y condiciones necesarias con el propósito de que la solución integral de software cumpla con los objetivos planteados.

Dichas reuniones se llevaron a cabo estando presentes todos o parte de los usuarios finales y el responsable del proyecto, analizando diferentes puntos de vista para así forjar los lineamientos a seguir, logrando un consenso entre las partes. Una vez se ha obtenido toda la información necesaria para llevar a cabo un minucioso estudio, se agrupa ordenadamente finalizando este proceso.

Ésta etapa tuvo una duración real de 10 días hábiles (1.4 semanas) desde el 27 de febrero hasta el 12 de marzo de 2012, lo cual quiere decir que se efectuó sin ningún retraso o contratiempo, según lo estimado en el cronograma de actividades.

**9.1.1 RF Requerimientos Funcionales**

**9.1.1.1 Tipos de Cuenta**

El sistema debe permitir la gestión de 3 tipos de usuarios, los cuales son:

* Súper Administrador:Posee el nivel de acceso 1.
* Funcionario Administrativo: Posee el nivel de acceso 2
* Trabajadora Social:Posee el nivel de acceso 3.

Debido a los tipos de usuario, y los requerimientos de la entidad, las restricciones en el sistema estarán dadas por un nivel de acceso, que es un número identificador de ciertos privilegios que un usuario tiene sobre el sistema. Existen 3 tipos de niveles de acceso diferentes, los cuales son:

* Nivel de Acceso 1: Cuenta con todos los privilegios habilitados, por lo tanto posee control total del sistema de información.
* Nivel de Acceso 2: Cuenta con los privilegios para el registro de los aportes de los responsables y la gestión del proceso de apadrinamiento del adulto mayor.
* Nivel de Acceso 3: Posee los privilegios para gestionar todo lo referente al proceso de inscripción del adulto mayor.

**9.1.1.2 Funcionario Administrativo**

* Permitir la gestión de la información relevante de los aportes de los responsables de los adultos mayores.
* Permitir la gestión de la información referente a los padrinos y el proceso de apadrinamiento del adulto mayor.
* Permitir la gestión de la información concerniente a las donaciones efectuadas por los padrinos.
* Permitir la generación de informes de los datos actuales del sistema.
* Permitir consultar la información de su cuenta de usuario.

**9.1.1.3 Trabajadora Social**

* Permitir la gestión de la información de las EPS a las que están afiliados los adultos mayores
* Permitir la gestión de las inscripciones de los adultos mayores.
* Permitir la gestión de la información del perfil de los adultos mayores (datos personales, datos de la inscripción y patologías, composición familiar y responsable).
* Permitir cargar la foto de cada adulto mayor para ser visualizada en su respectivo perfil.
* Permitir la generación de informes de los datos actuales del sistema.
* Permitir consultar la información de su cuenta de usuario.

**9.1.1.4 EPS**

* Permitir el registro de la información relevante de las EPS (nombre de la EPS y régimen) a las cuales están afiliados los adultos mayores. Esta información es esencial en el proceso de inscripción del adulto mayor.
* No está permitido agregar una EPS con el mismo nombre.
* Permitir la visualización de todos los adultos mayores afiliados a una EPS especifica. Incluir una búsqueda de la EPS por nombre o régimen.
* Permitir el registro de un nuevo régimen a cualquier EPS sin modificar el régimen vigente.
* Permitir la consulta y modificación de la información de cada EPS.

**9.1.1.5 Adulto Mayor**

* Permitir el registro de la información del perfil del adulto mayor (datos personales, datos de la inscripción y patologías, composición familiar y responsable).
* Permitir la visualización de los datos personales del adulto mayor de forma pública para que puedan ser apadrinados (adoptados).
* Permitir la visualización de la información del perfil del adulto mayor (datos personales, datos de la inscripción y patologías, composición familiar y responsable).
* Permitir la consulta y modificación de la información del perfil del adulto mayor (datos personales y datos del responsable).
* Permitir la visualización de todos los padrinos (personas naturales o jurídicas) que posee un adulto mayor.
* Permitir la visualización de todos los aportes realizados por el responsable del adulto mayor.
* Permitir eliminar el registro del adulto mayor si este muere o se retira del Jardín de los Abuelos.

**9.1.1.6 Inscripciones**

* Permitir aprobar o denegar la inscripción de un adulto mayor.
* Permitir la asignación del módulo donde residirá el adulto mayor durante su estancia en el Jardín de los Abuelos.

**9.1.1.7 Fotos**

* Permitir cargar la foto de cada adulto mayor para ser visualizada en su respectivo perfil.

**9.1.1.8 Aportes**

* Permitir el registro de los aportes que efectúa el responsable para el sostenimiento del adulto mayor.
* Permitir realizar una búsqueda avanzada, filtrando la información por diferentes parámetros de búsqueda (datos del responsable, módulo, rango de fechas y rango de montos).
* Permitir la consulta y modificación de la información del perfil del responsable (datos personales).

**9.1.1.9 Padrino**

* Permitir el registro de la información del perfil del padrino (persona natural o jurídica).
* Permitir realizar una búsqueda avanzada de los aportes realizados por el padrino, filtrando la información por diferentes parámetros de búsqueda (datos de la persona natural o jurídica, datos del adulto mayor y datos del aporte).
* Permitir la consulta y modificación de la información del perfil del padrino (datos personales).
* Permitir la visualización de todos los aportes realizados por el padrino.
* Permitir la visualización de todas las donaciones efectuadas por el padrino.
* Permitir la visualización de todos los adultos mayores apadrinados por el padrino.
* Permitir el registro de los aportes que realiza el padrino.

**9.1.1.10 Donaciones**

* Permitir realizar una búsqueda avanzada, filtrando la información por diferentes parámetros de búsqueda (datos del padrino y datos de la donación).
* Permitir la consulta y visualización de la información completa de una donación específica que se haya efectuado.
* Permite el registro de las donaciones que efectúen los padrinos (personas naturales o jurídicas), sea directamente para el Jardín de los Abuelos o para un adulto mayor.

**9.1.1.11 Informes**

* Permitir generar informes mediante archivos planos (.csv) separados de forma estructural por categorías, los cuales muestran el estado actual de los datos del sistema de información.

**9.1.1.12 Usuarios**

* Permitir el registro de la información de la cuenta de usuario (datos personales y datos de la cuenta).
* Permitir la consulta y modificación de la información de la cuenta de usuario (datos personales y datos e la cuenta).
* Permitir la visualización de la actividad en el sistema de información. Ofrece la posibilidad de realizar una búsqueda avanzada, filtrando la información por diferentes parámetros de búsqueda (tipo de actividad y rango de fechas).
* Permitir la eliminación una cuenta de usuario inhabilitándola permanentemente.

**9.1.2 RNF Requerimientos No Funcionales**

**9.1.2.1 Presentación**

El sistema de información tendrá un mecanismo de interfaz grafica amigable, de fácil interacción y visualización para los usuarios que lo administren.

**9.1.2.2 Arquitectura**

El sistema está fundamentado en el patrón de diseño MVC (Modelo-Vista-Controlador), con una conexión a base de datos MySQL, logrando un manejo rápido y eficiente en el despliegue y gestión de la información hacia los usuarios, alojado en un Servidor Apache como base primordial para un buen flujo y desarrollo de datos.

**9.1.2.3 Seguridad**

El sistema presenta un mecanismo de seguridad y confidencialidad a través de contraseñas internas de acceso a la base de datos, así como el manejo de cuentas de usuario, las cuales solo los usuarios del sistema o técnicos encargados del mantenimiento tendrán acceso a ellas.

**9.1.2.4 No Funcionales Organizacionales**

Teniendo presente lo mencionado anteriormente, el sistema se ha desarrollado en el lenguaje de programación PHP, es un lenguaje a nivel mundial muy reconocido para aplicaciones Web y con la ventaja principal de ser libre (no tiene ningún costo). Se utilizó un Framework de desarrollo web en PHP llamado CodeIgniter, para facilitar el uso de librerías y clases comúnmente vistas en este tipo de proyectos, además adopta el patrón de arquitectura de software Modelo-Vista-Controlador, el cual es un estándar a nivel mundial y permite una implementación más eficaz y ordenada de los sistemas de información.

**9.1.3 DP Domino del Problema**

El objetivo principal de la Corporación Jardín de los Abuelos es velar por la integridad de los Adultos Mayores que allí residen ofreciéndoles un lugar donde puedan tener una calidad de vida digna, en un entorno de armonía, apoyo espiritual y sana convivencia.

Las inscripciones de los adultos mayores se realizan en 2 etapas, en la primera se realiza en una cita formal con la Trabajadora Social la cual realiza un estudio de los datos trascendentales para este proceso como lo son los datos personales del adulto mayor, el motivo por el cual se quiere realizar la inscripción, los datos personales del responsable, entre otros. La segunda etapa es la admisión en la con el estudio realizado en la primer etapa se toma la decisión de aprobar o denegar la inscripción.

El proceso de apadrinamiento del adulto mayor, se efectúa cuando un padrino siendo una persona natural o jurídica decide adoptar a un adulto mayor y apoyarlo económicamente mediante un aporte o realizando una donación, ésta información puede ser diligenciada propiamente por el padrino o por un Funcionario Administrativo del Jardín de los Abuelos.

El registro de esta información debe realizarse de forma simple y sencilla permitiendo así una visualización clara y precisa de los datos que se almacenarán mediante el sistema. La aplicación debe ser capaz de soportar varios usuarios al mismo tiempo y proveer una gestión mínima de cuentas de usuario, cada cual con su respectivo historial de actividad; también debe manejar permisos de ingreso dependiendo de las funciones que tenga el usuario, ya que existen diferentes tipos de usuario (Súper Administrador, Administrativo y Trabajadora Social).

Por ende, haciendo uso de las tecnologías de información, se propone desarrollar un sistema de información para el registro y control de los datos, el cual permita el procesamiento y gestión de toda la información referente a los procesos de inscripción y apadrinamiento del adulto mayor.

**9.2 Análisis del Sistema**

En esta etapa, se identifican los casos de uso que tendrá el sistema para su implementación final. Previamente con la información obtenida en la etapa de análisis de requerimientos se estudiaron las posibilidades mínimas que existen en la interacción entre el sistema y los usuarios. Cada caso de uso se pensó en base a dichos requerimientos diseñados acorde a las exigencias de la entidad certificando que el sistema cumpla con todo lo estipulado.

Ésta etapa tuvo una duración de 27 días (3.8 semanas) de trabajo, empezando el 13 de marzo finalizando el 9 de abril de 2012, llevándose a cabo sin ningún contratiempo, acorde con el tiempo estipulado en el cronograma de actividades.

**9.2.1 AG Arquitectura General**

Debido a las necesidades de hacer al sistema accesible por varios usuarios y de estar acoplado al Portal Web del Jardín de los Abuelos, se decidió emplear la arquitectura Cliente/Servidor ya que el sistema estará alojado en el servidor que ha comprado la entidad.

Al implementarse el sistema con la arquitectura formulada, los componentes del mismo se organizan según su función y utilizando el patrón de diseño MVC, con el cual cumplirá con todos los estándares de arquitectura de software.

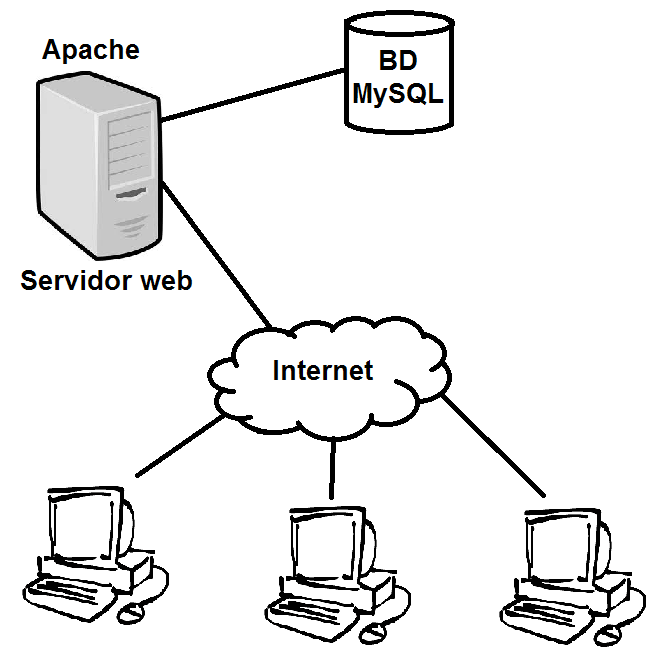
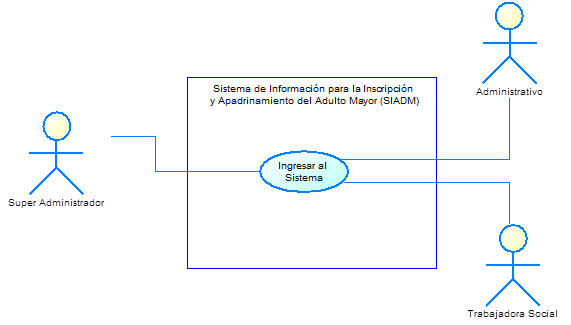
****

Figura 2. Arquitectura Cliente/Servidor.

**9.2.2 Descripción de los Casos de Uso**

|  |  |
| --- | --- |
| **Nombre** | **Descripción** |
| **Ingresar al Sistema** | Permite ingresar al sistema de información mediante el uso de una cuenta de usuario con contraseña. |

Tabla 1. Descripción del caso de uso Ingresar al Sistema.

Figura 3. Diagrama de caso de uso Ingresar al Sistema.

|  |  |
| --- | --- |
| Gestionar EPS | |
| **Nombre** | **Descripción** |
| **Agregar EPS** | Permite agregar la información de la Entidad Prestadora de Salud a la cual está afiliado el adulto mayor (nombre de la EPS y el régimen). |
| **Ver Afiliados** | Permite la visualización de los adultos mayores afiliados a una EPS específica. |
| **Agregar Régimen** | Permite agregar a una EPS específica un nuevo régimen. |
| **Modificar EPS** | Permite modificar la información de una EPS específica. |

Tabla 2. Descripción de los casos de uso incluidos en Gestionar EPS.

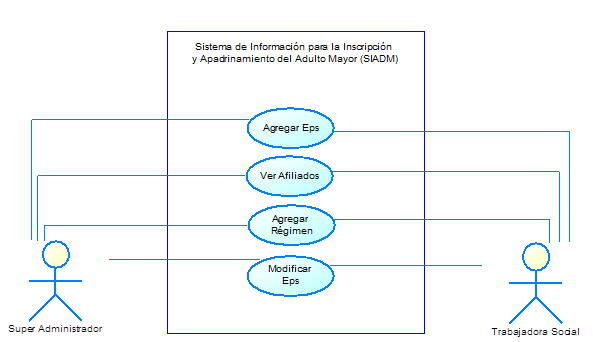


Figura 4. Diagrama de caso de uso Gestionar EPS.

|  |  |
| --- | --- |
| **Gestionar Inscripciones** | |
| **Nombre** | **Descripción** |
| **Aprobar/Denegar Inscripción** | Permite aprobar o denegar la inscripción realizada a un adulto mayor. |
| **Asignar Módulo** | Permite asignar el módulo donde residirá el adulto mayor durante su estancia en el Jardín de los Abuelos. |

Tabla 3. Descripción de los casos de uso incluidos en Gestionar Inscripciones.

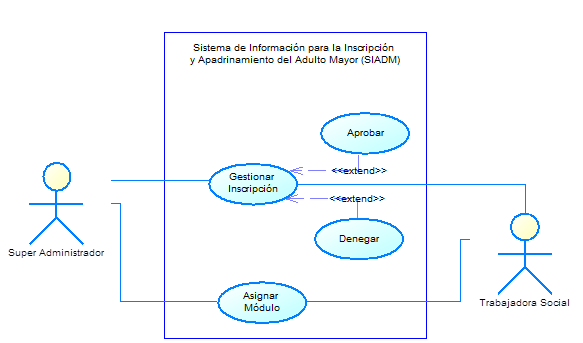


Figura 5. Diagrama de caso de uso Gestionar Inscripciones.

|  |  |
| --- | --- |
| **Gestionar Adulto Mayor** | |
| **Nombre** | **Descripción** |
| **Agregar Adulto Mayor** | Permite agregar la información del perfil de los adultos mayores (datos personales, datos de la inscripción y patologías, composición familiar y responsable). |
| **Ver Perfil Público** | Permite la visualización de los datos personales del adulto mayor para que puedan ser apadrinados (adoptados). |
| **Ver Perfil Adulto Mayor** | Permite visualizar la información del perfil de los adultos mayores (datos personales, datos de la inscripción y patologías, composición familiar y responsable). También ofrece la posibilidad de modificar los datos de los familiares. |
| **Modificar Adulto Mayor** | Permite modificar la información del perfil de los adultos mayores (datos personales y datos del responsable). |
| **Ver Padrinos** | Permite la visualización de los padrinos de los adultos mayores, ordenados por persona natural y jurídica. También ofrece la posibilidad de ver el perfil del padrino, y que aportes y donaciones ha efectuado. |
| **Ver Aportes Responsable** | Permite ver los aportes que mensualmente realiza el responsable del adulto mayor |
| **Eliminar Registro** | Permite eliminar el registro del adulto mayor, cambiando su estado a inactivo. |

Tabla 4. Descripción de los casos de uso incluidos en Gestionar Adulto Mayor.

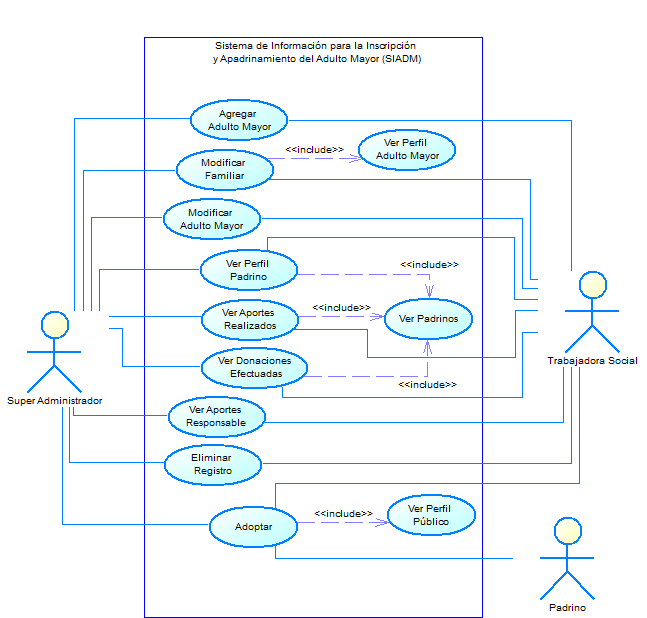


Figura 6. Diagrama de caso de uso Gestionar Adulto Mayor.

|  |  |
| --- | --- |
| **Gestionar Fotos** | |
| **Nombre** | **Descripción** |
| **Cargar Foto** | Permite cargar la foto de cada adulto mayor para ser visualizada en su respectivo perfil. |

Tabla 5. Descripción del caso de uso incluido en Gestionar Fotos.

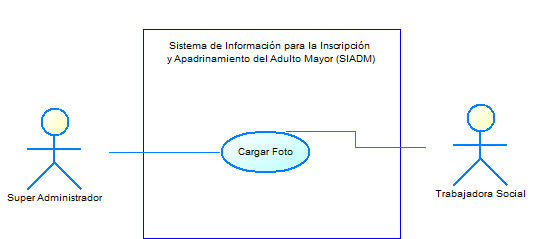


Figura 7. Diagrama de caso de uso Gestionar Fotos.

|  |  |
| --- | --- |
| **Gestionar Aportes** | |
| **Nombre** | **Descripción** |
| **Agregar Aporte Responsable** | Permite agregar los aportes que realiza el responsable para el sostenimiento del adulto mayor. |
| **Buscar Aportes Responsable** | Permite realizar una búsqueda avanzada, filtrando la información por diferentes parámetros de búsqueda (datos del responsable, módulo, rango de fechas y rango de montos). |
| **Modificar Responsable** | Permite visualizar la información del perfil del responsable (datos personales) para su modificación. |

Tabla 6. Descripción de los casos de uso incluidos en Gestionar Aportes.

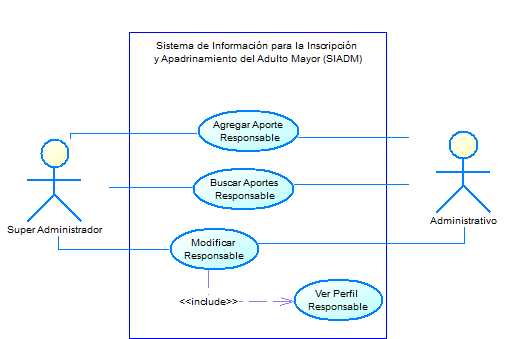


Figura 8. Diagrama de caso de uso Gestionar Aportes.

|  |  |
| --- | --- |
| **Gestionar Padrino** | |
| **Nombre** | **Descripción** |
| **Agregar Padrino** | Permite agregar la información del perfil del padrino (persona natural o jurídica). |
| **Buscar Aportes Padrino** | Permite realizar una búsqueda avanzada, filtrando la información por diferentes parámetros de búsqueda (datos de la persona natural o jurídica, datos del adulto mayor y datos del aporte). |
| **Modificar Padrino** | Permite visualizar la información del perfil del padrino (datos personales) para su modificación. |
| **Ver Aportes** | Permite visualizar todos los aportes que ha realizado el padrino. |
| **Ver Donaciones** | Permite visualizar todas las donaciones que ha efectuado el padrino. |
| **Ver Apadrinados** | Permite la visualización de los adultos mayores que ha apadrinado el padrino. |
| **Agregar Aporte Padrino** | Permite agregar los aportes que realiza el padrino como apoyo económico para el adulto mayor. |

Tabla 7. Descripción de los casos de uso incluidos en Gestionar Padrino.

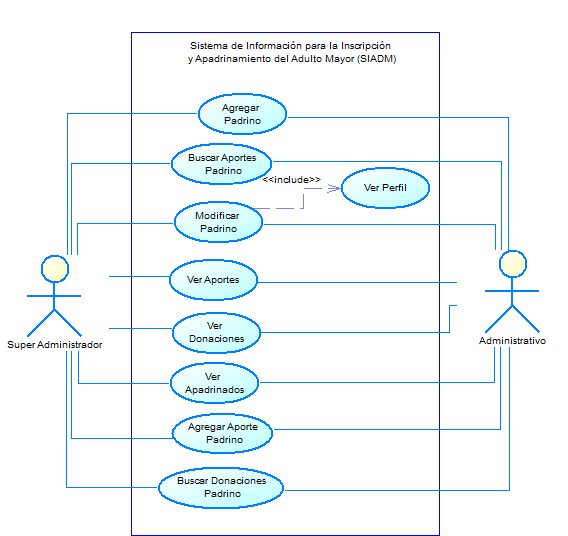


Figura 9. Diagrama de caso de uso Gestionar Padrino.

|  |  |
| --- | --- |
| **Gestionar Donaciones** | |
| **Nombre** | **Descripción** |
| **Buscar Donaciones Padrino** | Permite realizar una búsqueda avanzada, filtrando la información por diferentes parámetros de búsqueda (datos del padrino y datos de la donación) |
| **Ver Detalles Donación** | Permite visualizar la información completa de una donación específica que se haya efectuado. |
| **Agregar Donación** | Permite agregar las donaciones que efectúen los padrinos (personas naturales o jurídicas), sea directamente para el Jardín de los Abuelos o para un adulto mayor. |

Tabla 8. Descripción de los casos de uso incluidos en Gestionar Donaciones.

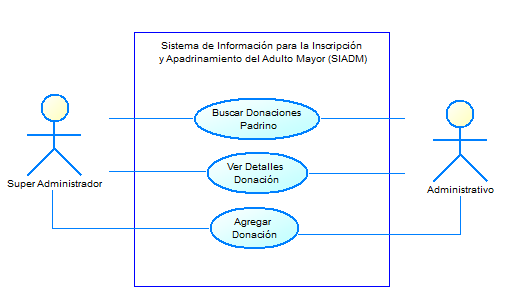


Figura 10. Diagrama de caso de uso Gestionar Donaciones.

|  |  |
| --- | --- |
| **Gestionar Informes** | |
| **Nombre** | **Descripción** |
| **Generar Informes** | Permite generar informes mediante archivos planos (.csv) separados de forma estructural por categorías, los cuales muestran el estado actual de los datos del sistema de información. |

Tabla 9. Descripción del caso de uso incluido en Gestionar Informes.

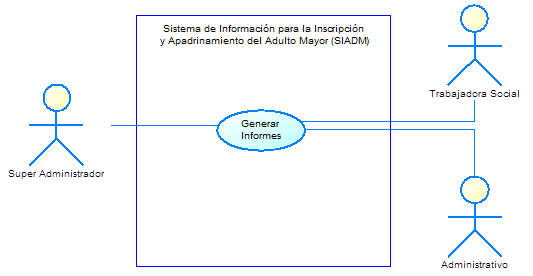


Figura 11. Diagrama de caso de uso Gestionar Informes.

|  |  |
| --- | --- |
| **Gestionar Usuarios** | |
| **Nombre** | **Descripción** |
| **Agregar Usuario** | Permite agregar la información de la cuenta de usuario (datos personales y datos de la cuenta). |
| **Modificar** | Permite modificar la información de la cuenta de usuario (datos personales y datos e la cuenta). |
| **Ver Actividad** | Permite visualizar la actividad en el sistema de información. Ofrece la posibilidad de realizar una búsqueda avanzada, filtrando la información por diferentes parámetros de búsqueda (tipo de actividad y rango de fechas). |
| **Eliminar Cuenta** | Permite eliminar una cuenta de usuario inhabilitándola permanentemente. |

Tabla 10. Descripción de los casos de uso incluidos en Gestionar Usuarios.

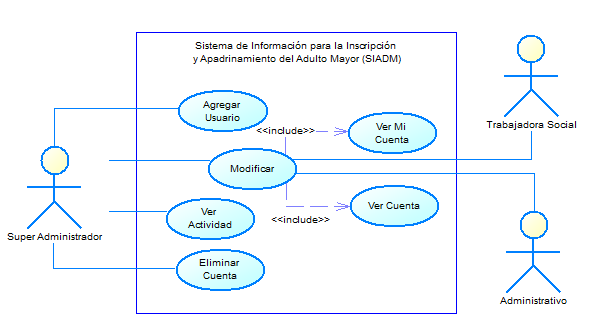


Figura 12. Diagrama de caso de uso Gestionar Usuarios.

**9.2.3 DCU Diagrama de Casos de Uso.**

****

Figura 13. Diagrama general de casos de uso del sistema.

**9.2.4 ECU Escenarios de Casos de Uso.**

**9.2.5 MCA Modelo de Clases de Análisis.**

**9.2.6 DSA Diagramas de Secuencia de Análisis.**

**NOTA IMPORTANTE:** Para más información de los encisos **9.2.3** – **9.2.6** consulte en el documento **Anexo A. Diagramas UML** de la Sección 1 a la 4**.**

**9.3 Diseño del Sistema**

En esta etapa del proyecto se diseñan los diagramas descriptivos del Sistema de Información, utilizando para ello el Lenguaje de Modelado Unificado (UML), el cual dará soporte proveyendo una documentación adecuada al sistema, que servirá como base principal para su análisis en posteriores actualizaciones. Se empezó un desarrollo básico del código fuente del sistema, implementando algunas interfaces y mostrando a los usuarios finales como puede ser el resultado final, para que ellos mismos aporten características importantes sobre la visualización del sistema. También se seleccionó el componente de catalogo de compras para su adición al portal y se estudio la posibilidad de un módulo publico en el sistema de información acoplado al portal, para el proceso de apadrinamiento del adulto mayor. Para finalizar esta etapa se diseñó y desarrolló el script de la base de datos en MySQL y se hicieron varias correcciones y complementaciones según reuniones con los usuarios, donde se les explicaba la información que sería almacenada y como debería ser su apropiada presentación.

Esta etapa tuvo una duración de 55 días (7.8 semanas), comenzando el 10 de abril terminando el 4 de junio de 2012, cumpliendo con el tiempo estipulado según el cronograma de actividades y sin ningún contratiempo significativo.

**9.3.1 DI Diagrama de Infraestructura**

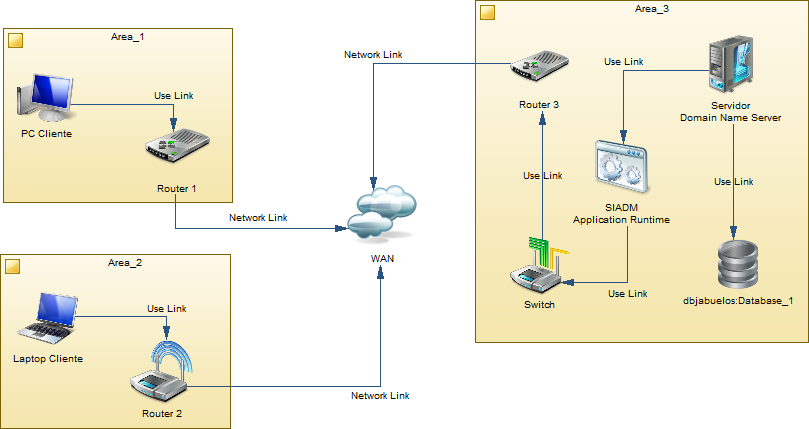
****

Figura 14. DI Diagrama de Infraestructura.

**9.3.2 DC Diagrama de Componentes**

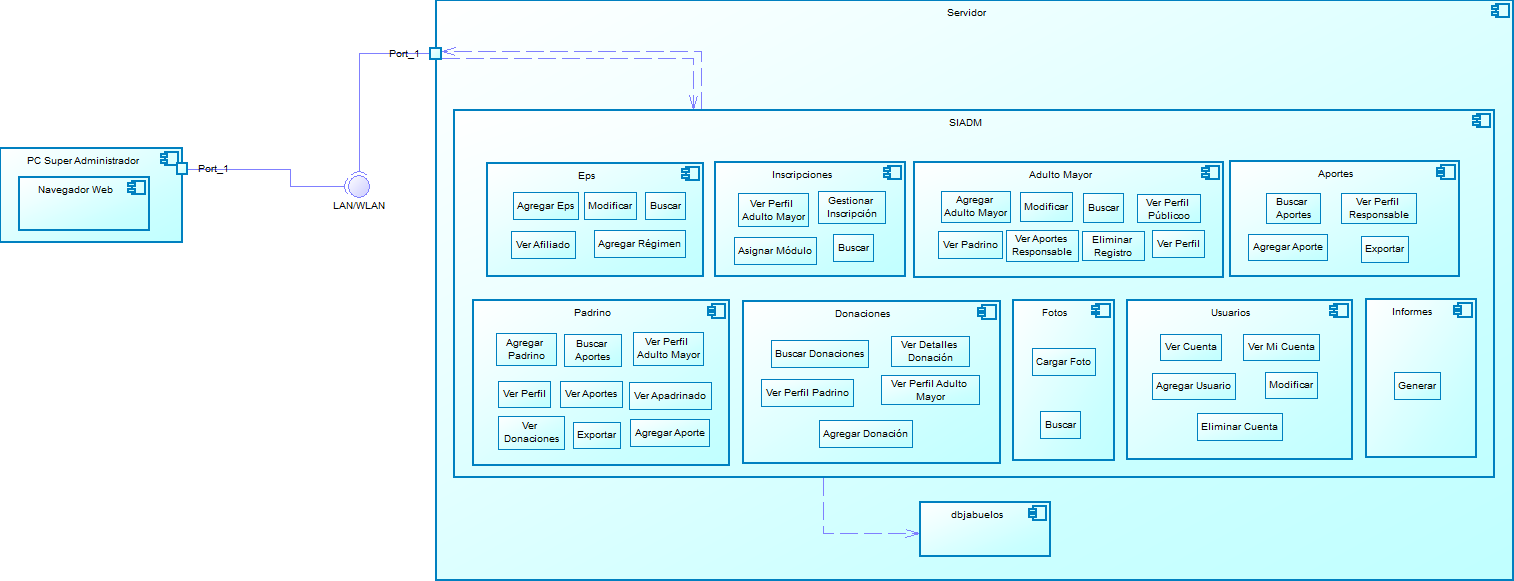
****

Figura 15. DC Diagrama de Componentes.

**9.3.3 DCD Diagrama de Clases de Diseño**

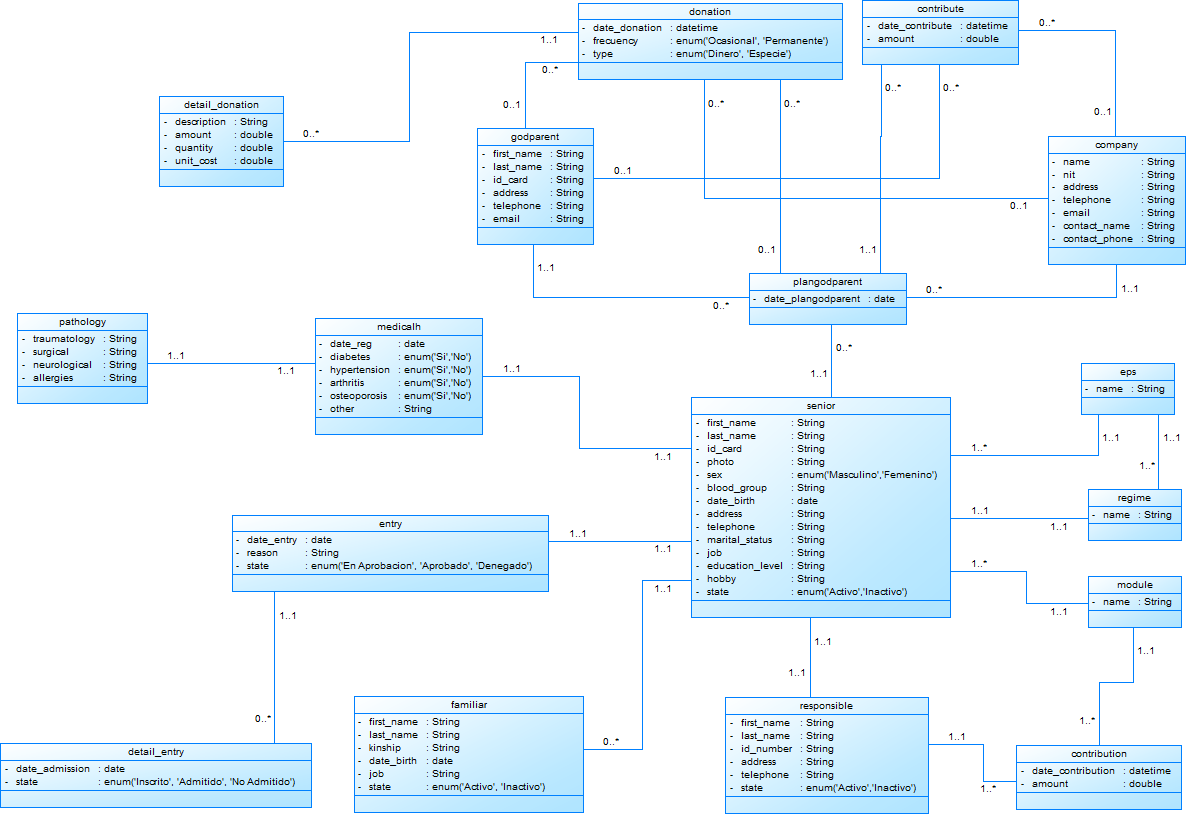
****

Figura 16. DCD Diagrama de Clases de Diseño.

**9.3.4 DD Diagrama de Despliegue**

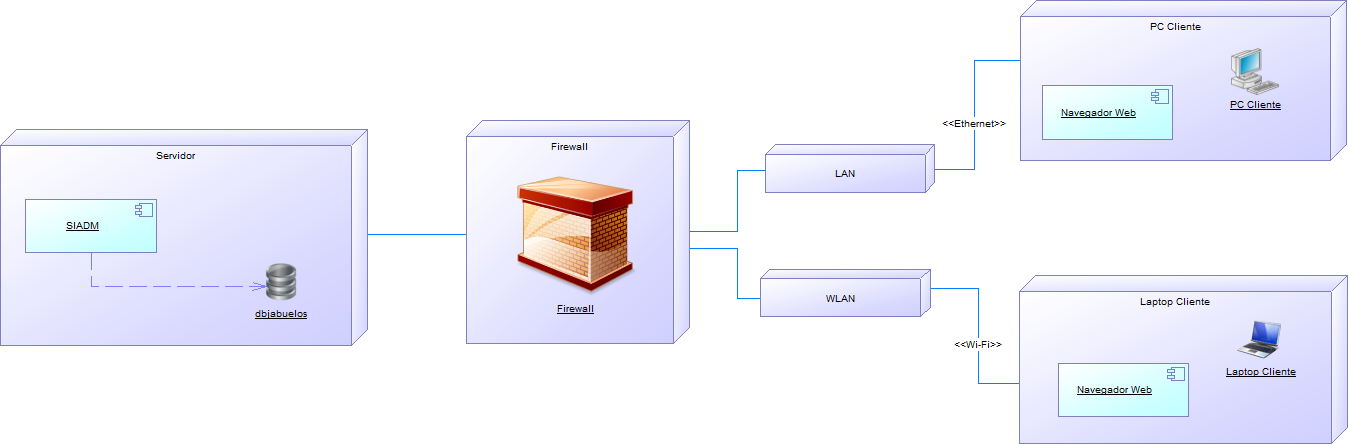
****

Figura 17. DD Diagrama de Despliegue.

**9.3.5 DIN Diagramas de Interacción.**

**9.3.6 DRE Diagrama Relacional**

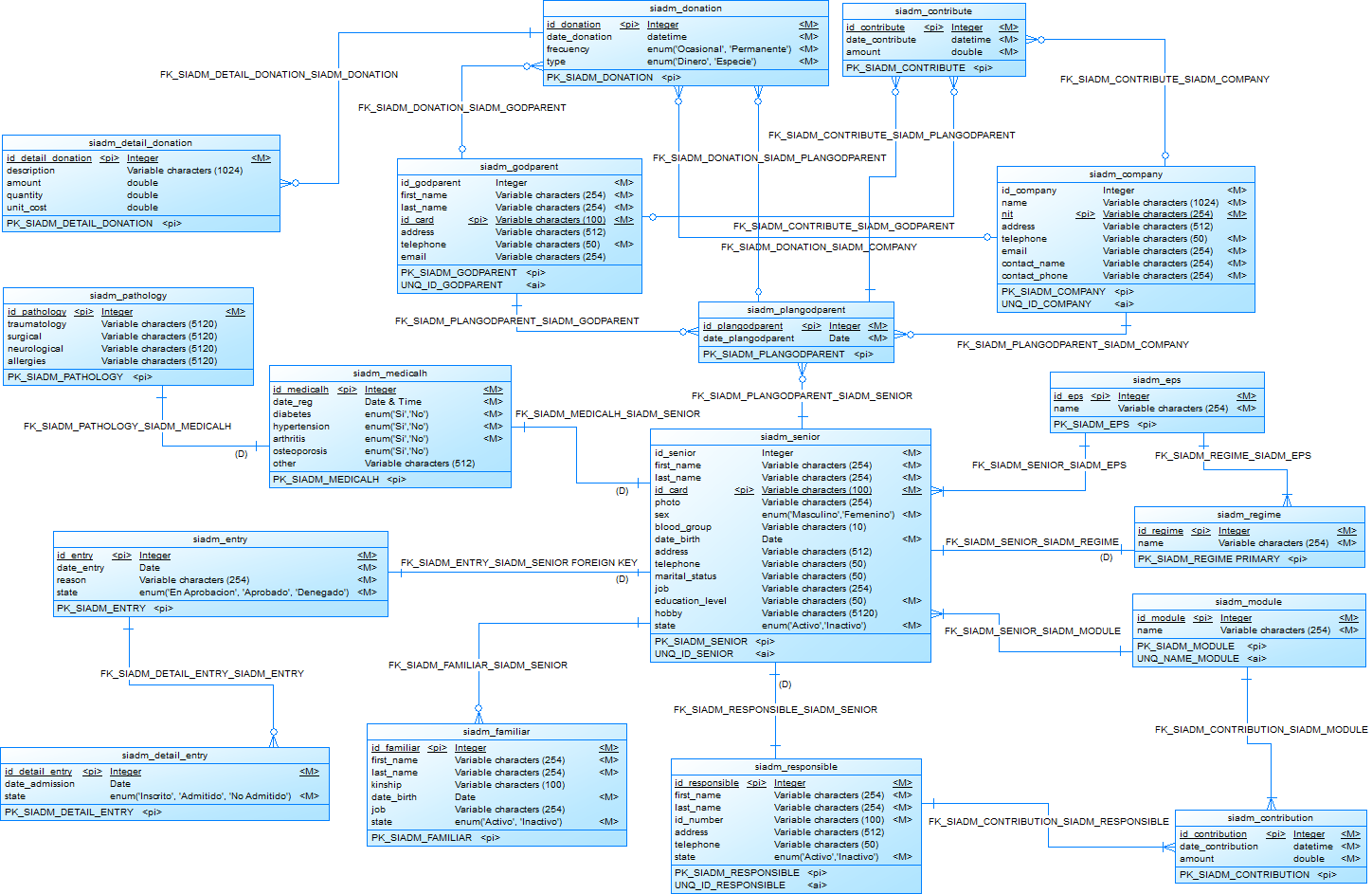
****

Figura 18. DRE Diagrama Relacional.

**9.3.7 DCS Diagrama de Clases de Software.**

**NOTA IMPORTANTE:** Para más información o mejor visualización de los encisos **9.3.1** – **9.3.7** consulte en el documento **Anexo A. Diagramas UML** de la Sección 5 a la 11**.**

**9.3.8 Diseño de Base de Datos**

El diseño fue realizado para un motor de base de datos relacional MySQL versión 4 o superior, para evitar errores de compatibilidad. El script final de la base de datos contiene 19 tablas para gestionar todo lo necesario según los requerimientos iniciales. Teniendo en cuenta esto, los puntos importantes previstos para el desarrollo del script final son descritos a continuación:

* No se diseñaron disparadores o procedimientos almacenados para garantizar compatibilidad con la base de datos del Portal Web; y optimizar el tiempo de ejecución en el servidor.
* Las tablas cuentan con todos los estándares para el diseño de una base de datos, utilizando las formas normales para así evitar la redundancia de datos.
* Todas las tablas poseen un identificador propio (llave primaria), evitando el uso de identificadores compuestos.
* Toda relación entre las tablas se implementó mediante restricción de integridad referencial (llaves foráneas).
* La integridad referencial se actualiza en cascada en todos los casos y restringe la eliminación de datos. Para esto se implementa un borrado lógico el cual se maneja con un atributo de estado o bandera.
* El idioma de los nombres de las tablas y atributos es inglés, sin uso de mayúsculas y caracteres especiales para evitar problemas con el servidor.
* Se utilizo un prefijo para cada tabla (siadm\_tabla) y así poder diferenciar la base de datos del Sistema de Información de la del Portal Web.
* El juego de caracteres que usa la base de datos es UTF-8, debido a la necesidad de almacenar cierta información que puede contener caracteres especiales.

En el cuadro se muestran todas las tablas del script de la base de datos con una breve descripción.

|  |  |
| --- | --- |
| **Nombre de la tabla** | **Descripción** |
| activity | Almacena la información de la actividad que realiza cualquier usuario en el sistema. Se usa para consultar la actividad del usuario en el sistema y está relacionada con la tabla “user”. |
| company | Almacena la información del perfil de la persona jurídica (datos de la entidad). Está relacionada con las tablas “plangodparent”, “contribute” y “donation”. |
| contribute | Almacena la información de todos los aportes que realizan los padrinos (personas naturales o jurídicas) cuando adoptan a un adulto mayor. Está relacionada con las tablas “plangodparent”, “godparent” y “company”. |
| contribution | Almacena la información de todos los aportes que realiza el responsable del adulto mayor. Está relacionada con las tablas “responsible” y “module”. |
| detail\_donation | Almacena la información de todos los detalles de las donaciones que efectúan los padrinos (personas naturales o jurídicas), sea directamente para el Jardín o para un adulto mayor. Está relacionada con la tabla “donation”. |
| detail\_entry | Almacena la información de todos los detalles de las inscripciones, cuando se inscribe a un adulto mayor. Está relacionada con la tabla “entry”. |
| donation | Almacena la información de todas las donaciones que efectúan los padrinos (personas naturales o jurídicas), sea directamente para el Jardín de los Abuelos o para un adulto mayor. Está relacionada con las tablas “company”, “godparent”, “plangodparent” y “detail\_donation”. |
| entry | Almacena la información de todas las inscripciones de los adultos mayores. Está relacionada con las tablas “senior” y “detail\_entry” |
| eps | Almacena la información de la EPS a la cual está afiliado el adulto mayor. Está relacionada con las tablas “regime” y “senior”. |
| familiar | Almacena la información de todos los familiares de los adultos mayores. Está relacionada con la tabla “senior”. |
| godparent | Almacena la información del perfil de la persona natural (datos personales). Está relacionada con las tablas “plangodparent”, “contribute” y “donation”. |
| medicalh | Almacena la información de la historia médica del adulto mayor. Esté relacionada con las tablas “senior” y “pathology”. |
| module | Almacena los módulos donde residen los adultos mayores. Está relacionada con las tablas “senior” y “contribution”. |
| pathology | Almacena la información de las patologías de los adultos mayores. Está relacionada con la tabla “medicalh”. |
| plangodparent | Almacena la información concerniente al plan padrino que se registra en el momento en el que un padrino (persona natural o jurídica) adopta a un adulto mayor. Está relacionada con las tablas “senior”, “godparent”, “company”, “contribute” y “donation”. |
| regime | Almacena los diferentes tipos de régimen que puede tener una EPS. Está relacionada con las tablas “eps” y “senior”. |
| responsible | Almacena la información del perfil del responsable del adulto mayor. Está relacionada con las tablas “senior” y “contribution”. |
| senior | Almacena la información del perfil del adulto mayor. Está relacionada con las tablas “eps”, “regime”, “module”, “entry”, “medicalh”, “familiar” “responsible” y “plangodparent” |
| user | Almacena la información de las cuentas de usuario (datos personales y datos de la cuenta). Está relacionada con la tabla “activity”. |

Tabla 11. Descripción de tablas de la base de datos.

**9.3.9 Diseño de Interfaces**

Las interfaces de usuario se diseñaron utilizando CSS y JavaScript. Para la parte de CSS, se utilizó un Framework de maquetación por columnas llamado 960 Grid System, facilitando la ubicación de los elementos en pantalla. En la parte de JavaScript se empleó el Framework de desarrollo jQuery, el cual posee muchos plug-ins importantes para facilitar el uso de herramientas visuales y en el caso propio del proyecto se hizo uso de jQuery UI y DataTables, proporcionando múltiples ventajas con bajo uso de recursos computacionales.

El diseño principal de las interfaces se pensó usando un contraste entre los colores de la Corporación Jardín de los Abuelos junto con los de la Universidad de Ibagué, se utilizaron los logos de ambas entidades para así darle propiedad y vistosidad al Sistema de Información.



Figura 19. Logo Corporación Jardín de los Abuelos.



Figura 20. Logo Sistema de Información y Universidad de Ibagué.

SIADM significa **“Sistema de Información para la Inscripción y Apadrinamiento del Adulto Mayor”.** Usando un contraste de los colores de los logos antes mencionados y el nombre del sistema de información, se obtiene un resultado agradable, simple y adecuado.

**9.3.9.1 Página de Inicio**

La página de inicio del sistema muestra que los aspectos mencionados lucen de acuerdo a lo planteado en el diseño de interfaces. A continuación se visualiza la página inicial de SIADM.

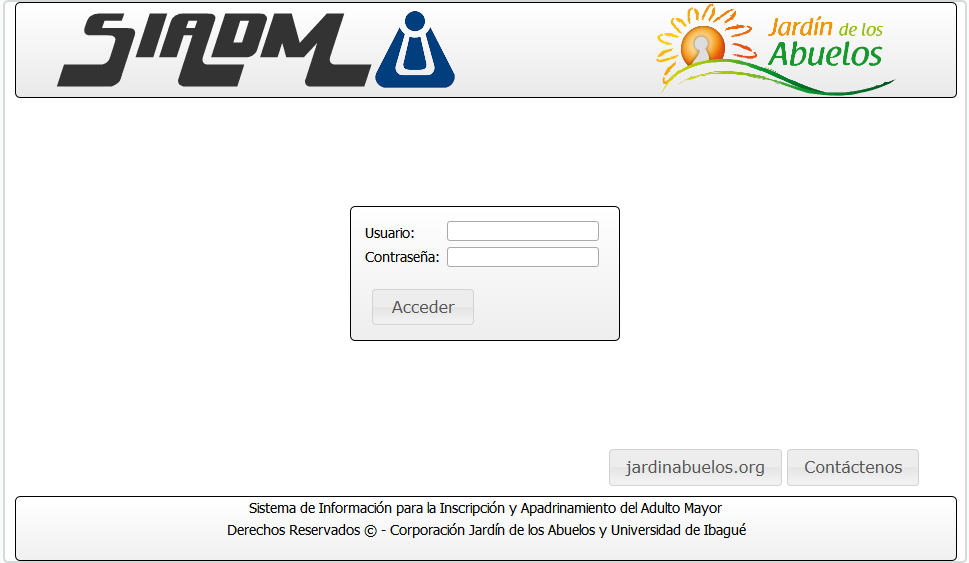
****

Figura 21. Página de Inicio SIADM.

**9.3.9.2 Menús del Sistema**

El diseño de los menús es importante, para cada nivel de acceso debe haber diferentes funcionalidades. Los menús del sistema fueron creados para facilitar la interacción entre el usuario y el sistema siendo lo más sencilla posible.

Utilizando iconos coloridos, agradables y de fácil asociación a las funciones que cumplen; se diseñó un menú principal que estará visible en todo momento y cambiará dinámicamente según el tipo de usuario que acceda al sistema.

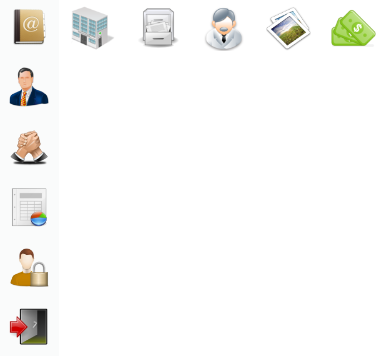


Figura 22. Menú principal SIADM para usuario con nivel de acceso 1.

C:\wamp\www\siadm\user_guide\images\menu2.png

Figura 23. Menú principal SIADM para usuario con nivel de acceso 2.



Figura 24. Menú principal SIADM para usuario con nivel de acceso 3.

**9.4 Implementación del Sistema**

En esta etapa del proyecto, el sistema concluye en su primera versión funcional. Se desarrolló el código fuente en su totalidad y se implementó la base de datos, con lo cual se iniciaron las pruebas a nivel local para identificar posibles fallos y corregirlos en el tiempo en que dura completarse esta etapa. También se añadió el componente SimpleCaddy al Portal Web, el cual funciona como catálogo de productos y carrito de compras.

La duración de esta etapa fue de 62 días (8.8 semanas), iniciando el 5 de junio terminando el 6 de agosto de 2012, con aproximadamente 1 semana de retraso frente a lo estipulado en el cronograma de actividades. Sin embargo, no se presentó ningún contratiempo mayor debido a la holgura temporal prevista de una semana.

Las herramientas utilizadas para el desarrollo de esta etapa de proyecto fueron CodeIgniter, jQuery, jQuery UI, 960 Grid System, como entorno para la base de datos se utilizó WAMPSERVER y NetBeans para el desarrollo del código fuente.

**9.4.1 CF Código Fuente.**

**9.4.2 FBD Fichero de Base de Datos.**

**9.4.3 MU Manual de Usuario. Documento Virtual Adjunto al Sistema de Información.**

**NOTA IMPORTANTE:** Para más información de los encisos **9.4.1** – **9.4.3** consulte en la carpeta **Anexo B. Sistema de Información.**

**9.5 Pruebas del Sistema**

En esta etapa final del proyecto, se hace entrega formal del Sistema de Información para realizar la ejecución del plan de pruebas de eficiencia, que consta de pruebas transaccionales, unitarias y de robustez. Se verificó el funcionamiento del sistema y del componente añadido al portal y se realizó la debida capacitación a los usuarios sobre el manejo adecuado de éste.

La entrega final del Sistema de Información a la Corporación Jardín de los Abuelos contiene los siguientes productos:

* Código fuente del Sistema de Información (SIADM).
* Script de la base de datos utilizado por SIADM con las inserciones necesarias para su funcionamiento.
* Manual de usuario SIADM, que es un documento virtual adjunto al Sistema de Información.

Esta etapa tuvo una duración de 19 días (2.7 semanas), empezando el 8 de agosto terminando el 27 de agosto de 2012, con lo que satisfactoriamente se da finalización al desarrollo del proyecto sin ningún retraso o complicación.

**9.5.1 PPE Plan de Pruebas de Eficiencia**

**9.5.1.1Pruebas Transaccionales**

Se utilizó la herramienta Load Impact que presta un servicio gratuito de pruebas de carga y optimización de sitios web para el proceso de transacciones simulando peticiones realizadas por los usuarios.

En las siguientes tablas se visualizan los resultados de la prueba efectuada:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Segundos** | **0** | **9** | **18** | **27** | **36** | **45** | **54** |
| **Procesos** |  |  |  |  |  |  |  |
| **Usuarios activos** | 0 | 2 | 4 | 5 | 7 | 8 | 10 |
| **Tiempo de carga por usuario en segundos** | 0 | 0 | 0 | 0 | 3.91 | 3.77 | 5.02 |
| **Peticiones por segundo** | 0 | 9 | 3 | 5 | 11 | 8 | 6 |
| **Porcentaje de ocupación de memoria** | 0 | 0.589 | 0.658 | 0.722 | 0.820 | 0.877 | 0.950 |
| **Porcentaje de utilización de la CPU** | 0 | 0.500 | 0.340 | 0.300 | 0.340 | 0.370 | 0.380 |

Tabla 12. Pruebas transaccionales aproximadamente al minuto de ejecución.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Minutos** | **0** | **1** | **2** | **3** | **4** | **5** |
| **Procesos** |  |  |  |  |  |  |
| **Usuarios activos** | 0 | 11 | 20 | 30 | 40 | 50 |
| **Tiempo de carga por usuario en segundos** | 0 | 4.76 | 3.99 | 3.78 | 3.74 | 3.77 |
| **Peticiones por segundo** | 0 | 2 | 27 | 41 | 38 | 66 |
| **Porcentaje de ocupación de memoria** | 0 | 0.985 | 1.427 | 1.910 | 2.396 | 2.889 |
| **Porcentaje de utilización de la CPU** | 0 | 0.360 | 0.540 | 0.730 | 0.960 | 1.150 |

Tabla 13. Pruebas transaccionales aproximadamente a los 5 minutos de ejecución.

|  |  |
| --- | --- |
| **Datos Complementarios** | |
| **Rango de Ancho de Banda** | 132.1 Kbit/s – 22.1Mbit/s |
| **Datos Recibidos** | 260.00MB |
| **Conexiones TCP Activas** | 50 Usuarios |
| **Número Total de Peticiones** | 6666 (21.16req/s) |

Tabla 14. Pruebas transaccionales de datos complementarios.

**9.5.1.2 Pruebas Unitarias**

Se usó la clase Unit Test librería que trae embebida CodeIgniter para llevar a cabo este tipo de pruebas. A continuación se presentan los resultados:

|  |  |  |  |
| --- | --- | --- | --- |
| **Clase** | **Métodos** | **Estado** | **Resultado** |
| Account | display(), add\_account(), update\_account(), delete\_account\_ajax() | Evaluados | Aprobados |
| Godparent | display\_natural\_person(), display\_legal\_person(),  display\_senior(),  add\_np(), add\_lp(), update\_np(), update\_lp() | Evaluados | Aprobados |
| Senior | display(), add\_senior(), update\_senior, update\_familiar(), delete\_senior\_ajax() | Evaluados | Aprobados |

Tabla 15. Resultados de pruebas unitarias.

**9.5.1.3 Pruebas de Robustez**

Empleando la clase Benchmark de CodeIgniter, la cual permite calcular el tiempo de ejecución y la ocupación de espacio en memoria de las clases que componen el sistema de información, se llevaron a cabo éstas pruebas tanto en un servidor local como en el servidor remoto.

|  |  |  |
| --- | --- | --- |
| **Servidor Local** | | |
| **Clase** | **Tiempo Total de Ejecución** | **Espacio en Memoria** |
| Account | 0.0501 s | 3.38 MB |
| Contribution | 0.0526 s | 3.15 MB |
| Dashboard | 0.0366 s | 2.82 MB |
| Donation | 0.0542 s | 3.28 MB |
| Entry | 0.0450 s | 3.04 MB |
| Eps | 0.0467 s | 3.06 MB |
| Godparent | 0.0448 s | 3.35 MB |
| Login | 0.0474 s | 2.84 MB |
| Photo | 0.0477 s | 2.96 MB |
| Report | 0.0537 s | 3.16 MB |
| Senior | 0.0621 s | 3.4 MB |

Tabla 16. Pruebas de robustez servidor local.

|  |  |  |
| --- | --- | --- |
| **Servidor Remoto** | | |
| **Clase** | **Tiempo Total de Ejecución** | **Espacio en Memoria** |
| Account | 0.0195 s | 3.7 MB |
| Contribution | 0.0196 s | 3.44 MB |
| Dashboard | 0.0183 s | 3.12 MB |
| Donation | 0.0194 s | 3.56 MB |
| Entry | 0.0198 s | 3.33 MB |
| Eps | 0.0170 s | 3.35 MB |
| Godparent | 0.0193 s | 3.64 MB |
| Login | 0.0174 s | 3.13 MB |
| Photo | 0.0185 s | 3.26 MB |
| Report | 0.0172 s | 3.46 MB |
| Senior | 0.0190 s | 3.66 MB |

Tabla 17. Pruebas de robustez servidor remoto.

**10. RESUMEN DEL PROYECTO**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actividad** | **Fecha de inicio** | **Fecha de terminación** | **Duración (semanas)** |
| Fecha de entrega de la propuesta al programa de Ingeniería de Sistemas. | Febrero 6 de 2012 | ---- | 0 |
| Fecha de aprobación del proyecto en el programa de Ingeniería de Sistemas. | Marzo 27 de 2012 | ---- | 0 |
| Análisis de requerimientos | Febrero 27 de 2012 | Marzo 12 de 2012 | 1.4 |
| Análisis del sistema | Marzo 13 de 2012 | Abril 9 de 2012 | 3.8 |
| Diseño del sistema | Abril 10 de 2012 | Junio 4 de 2012 | 7.8 |
| Implementación del sistema | Junio 5 de 2012 | Agosto 6 de 2012 | 8.8 |
| Pruebas del sistema | Agosto 8 de 2012 | Agosto 27 de 2012 | 2.7 |
| Tiempo total invertido en el desarrollo del proyecto | Febrero 27 de 2012 | Agosto 27 de 2012 | 24.5 |

Tabla 18. Resumen del proyecto.

**CONCLUSIONES**

* Una vez se determinó el dominio del problema referente a los procesos de inscripción y apadrinamiento del adulto mayor se decidió proceder con la metodología de Proceso Unificado la cual probó ser de mucha utilidad ya que permitió de manera eficiente extraer los requerimientos funcionales para el Sistema de Información y de este modo establecer rápidamente con los usuarios una comprensión bipartita de los mismos y de lo que obtendrían de la solución al final del desarrollo.
* Al finalizar el desarrollo, teniendo en cuenta los requerimientos obtenidos y lo estipulado en las etapas de análisis y diseño el Sistema de Información permite a los diferentes tipos de usuario acceder a los siguientes nueve módulos:
  + Módulo EPS.
  + Módulo Inscripciones.
  + Módulo Adulto Mayor.
  + Módulo Fotos.
  + Módulo Aportes.
  + Módulo Padrino.
  + Módulo Donaciones.
  + Módulo Informes.
  + Módulo Usuarios.

Que permiten agregar, modificar, buscar, realizar eliminaciones, asignaciones, informes, adopciones, aportes, donaciones y demás funcionalidades exigidas por la entidad; referente a los procesos de inscripción y apadrinamiento del adulto mayor. Con la distribución de las funcionalidades en diferentes módulos se brinda la posibilidad al usuario de acceder a la información de una forma más sencilla y amena, además permite la continuidad en el desarrollo de más módulos si así la entidad lo requiere.

* Gracias a la implementación de los patrones de diseño UML y al Proceso Unificado de Desarrollo se determinó optar por una arquitectura Cliente/Servidor apoyada con el patrón de diseño MVC para que así el Sistema de Información esté distribuido en diferentes capas separando la lógica de negocio de la interfaz de usuario. Mediante la persistencia de datos se desarrolló una base de datos relacional normalizada; todo esto para garantizar que la información está protegida de la intervención de cualquier agente externo siendo almacenada de forma segura.
* Finalmente luego de analizar una serie de componentes útiles y de simple interacción con el usuario fue añadido al Portal Web un componente llamado SimpleCaddy, que permite adicionar los diferentes de productos que ofrezca el Jardín de los Abuelos al público, funcionando como una tienda online o carrito de compras. El Módulo Adulto Mayor posee una sección pública la cual se ha acoplado de forma satisfactoria al portal, permitiendo efectuar el proceso de apadrinamiento del adulto mayor.

**RECOMENDACIONES**

* Para realizar un buen uso y aprovechar al máximo la funcionalidad del Sistema de Información para la Inscripción y Apadrinamiento del Adulto Mayor (SIADM), es importante que los usuarios reciban una capacitación que les permita conocer el procedimiento para poder hacer uso del sistema.
* Para un buen rendimiento del Sistema de Información se recomienda utilizar Mozilla Firefox o Google Chrome; Internet Explorer puede presentar problemas de incompatibilidad.

**ANEXOS**

**Anexo A.** Diagramas UML.

**Anexo B.** Sistema de Información.

**Anexo C.** Plan de Pruebas de Eficiencia.

**Anexo D.** Documentos de Soporte.

**BIBLIOGRAFÍA**

**Fuentes Textuales**

Sommerville, Ian. Ingeniería del Software, Prentice Hall, 2007. 687 páginas.

Larman, Craig. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process, Prentice Hall, 1999. 656 páginas.

Schach, Stephen. Análisis y diseño orientado a objetos con UML y el proceso unificado, McGraw-Hill, 2005.

Cobo, Ángel; Gómez, Patricia; Pérez Daniel y Rocha, Rocío. PHP y MySQL: Tecnología para el desarrollo de aplicaciones web. 2005. p. 57.

Aubry, Christopher. DRAMWEAVER CS3: para PC-MAC. Eni ediciones Barcelona, 2008, p. 56.

Beigbeder, Federico. Diccionario Politécnico de lengua española e inglesa 2da Edición. 1997, p. 1396.

Casas Antunez, Carlos. Aplicaciones a la red de internet. Editorial Vértice España, 2da edición 1997, p. 37.

Rivera León, Fray Osorio. Base de datos relacionales. Editorial textos académicos, Colombia 2008, p. 91.

Jorquera, Marcos, Administración de servicios de internet de la teoría a la práctica, Publicaciones Universidad de Alicantes, 2008, p. 105.

Márquez, Andrés; García, Salvador; Quintana, Gregorio. SQL y desarrollo en aplicaciones en Oracle 8. Publicaciones de la UniversitatJaum, 2001, colección No. 9 p. 16.

Palomares Ortega, Miguel Ángel. Manual Práctico: servicio de redes de área local. Editorial Visión Libros. España p. 114.

Pons Capote, Olga. Introducción a la base de datos: el modo relacional. Editorial Thomson, 2005, p. 6.

Sánchez Garreta, José Salvador, Ingeniería de proyectos informáticos: actividades y procedimientos, Publicaciones de la UniversitatJaum, 2003, p. 8.

Sánchez Maza, Miguel Ángel. JAVASCRIPT. Ediciones Innova, España 2001, p. 9.

Villar Varela, Ana maría, Introducción a la informática y al uso y manejo de aplicaciones comerciales, editorial Vigo 1ª edición 2006, p. 16.

Connolly, Thomas. Begg, Carolyn. Database Systems. A practical approach to design, implementation and management. Editorial Addison Wesley, 4ª Edición. Inglaterra 2005.

R. Schach, Stephen. Object – Oriented and Classical Software Engineering. Editorial Mc. Graw Hill, 7ª Edición. Estados Unidos 2007.

Gutiérrez Rodríguez, Abraham. García Ginés, Bravo. PHP5 a través de ejemplos. Editorial Alfaomega, 1ª Edición. España 2005.

**Fuentes Electrónicas**

Pressman, Roger. Ingeniería de software, un enfoque práctico, Mcgraw-Hill, quinta edición, Enero 25 de 2012, http://es.scribd.com/doc/7978336/Ingenieria-de-Software-Un-Enfoque-Practico-Pressman-5th-Ed

Sistemas de información en las empresas, Antonio Muñoz Cañate, Enero 28 de 2012, http://www.hipertext.net/web/pag251.htm.

Procesos de la Ingeniería de Requerimientos, Gonzalo Mena Mendoza, Ingeniería de Requerimientos, Maestría en Ingeniería de Software Distribuido, Universidad Autónoma de Querétaro, Enero 29 de 2012, http://mena.com.mx/gonzalo/maestria/ingreq/presenta/procesos\_ir/

Sistemas Información Conceptos Y Elementos, Profesor Lauro Soto, Ensenada, BC, México, Enero 31 de 2012, http://www.mitecnologico.com/Main/SistemasInformacionConceptosYElementos

Achour, Mehdi; Betz, Friedhelm; Dovgal, Antony y muchos otros. Manual de PHP, Agosto 25 de 2012, http://docs.php.net/manual/es/intro-whatis.php

1. SOMMERVILLE, Ian. Ingeniería del Software. Prentice Hall, 2007, p. 5. [↑](#footnote-ref-1)
2. LARMAN, Craig. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process. Prentice Hall, 1999, p. 5. [↑](#footnote-ref-2)
3. PALOMARES, Miguel Ángel. Manual Práctico: servicio de redes de área local. Visión Libros. España, p. 114. [↑](#footnote-ref-3)
4. BEIGBEDER, Federico. Diccionario Politécnico de lengua española e inglesa 2da Edición. 1997, p. 1396. [↑](#footnote-ref-4)
5. ACHOUR, Mehdi; BETZ, Friedhelm; DOVGAL, Antony y muchos otros. Manual de PHP, Agosto 25 de 2012, http://docs.php.net/manual/es/intro-whatis.php [↑](#footnote-ref-5)
6. COBO, Ángel; GÓMEZ, Patricia; PÉREZ Daniel y ROCHA, Rocío. PHP y MySQL: Tecnología para el desarrollo de aplicaciones web. 2005, p. 57. [↑](#footnote-ref-6)
7. SÁNCHEZ, Miguel Ángel. JAVASCRIPT. Ediciones Innova, España 2001, p. 9. [↑](#footnote-ref-7)
8. SOMMERVILLE, Ian. Ingeniería del Software. Prentice Hall, 2007, p. 226. [↑](#footnote-ref-8)
9. RIVERA, Fray Osorio. Base de datos relacionales. Editorial textos académicos, Colombia 2008, p. 91. [↑](#footnote-ref-9)
10. COBO, Ángel; GÓMEZ, Patricia; PÉREZ Daniel y ROCHA, Rocío. PHP y MySQL: Tecnología para el desarrollo de aplicaciones web. 2005, p. 339. [↑](#footnote-ref-10)