

Universidad de las Ciencias Informáticas

Facultad 8



“Sistema de Gestión Información Gineco-Obstétrico”

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

Autores: Yamisleydis Borrero LaO

Yanisvel Suárez Arbolay

Tutores: Ing. Ariel Orta Hernández

Ing. Iliana Pérez Pupo

Ase soras: Esp. Marisol González Vigoa

Lic. Ariagnys Quintana Montalvo

Lic. María Perojo Medrano

Ciudad de La Habana, 2009

“Año del 50 Aniversario del Triunfo de la Revolución”

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 -----.

Yanisvel Suárez Arbolay

Yamisleydis Borrero La O

Firma del autor

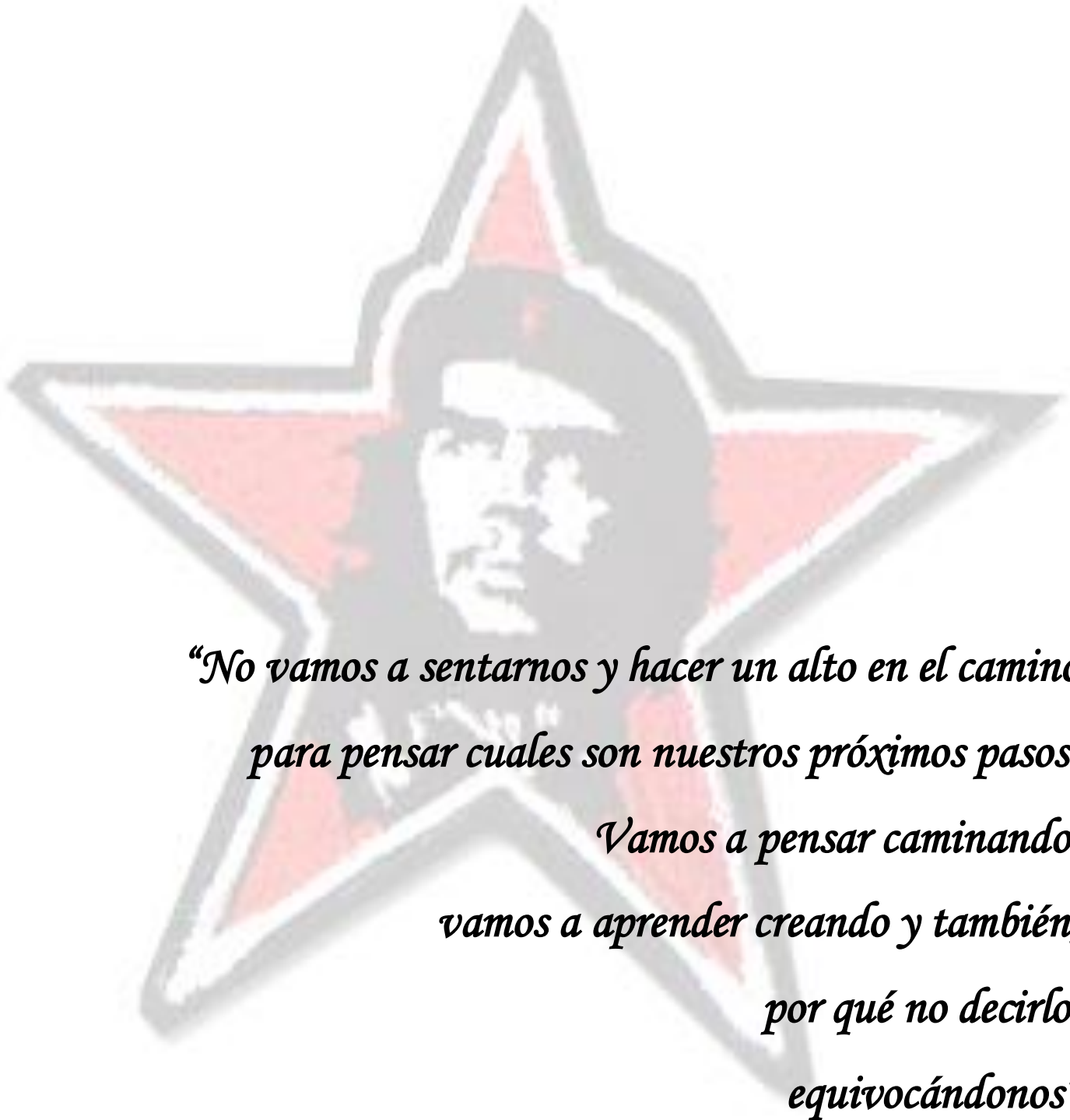
Firma del autor

Ing. Ariel Orta Hernández

Ing. Iliana Pérez Pupo

Firma del Tutor

Firma de la Co-tutora



*“No vamos a sentarnos y hacer un alto en el camino
para pensar cuales son nuestros próximos pasos.*

*Vamos a pensar caminando,
vamos a aprender creando y también,
por qué no decirlo,
equivocándonos”*

Datos de contactos

Inq. Ariel Orta Hernández -> arota@uci.cu

Graduado de Ingeniería Informática en el año 2007 en la Universidad de las Ciencias Informáticas (UCI). Actualmente se desempeña como profesor en la misma universidad donde ha impartido asignaturas como Base de Datos y Gestión del Software. Posee categoría Instructor Recién Graduado.

Inq. Iliana Pérez Pupo -> iperez@uci.cu

Graduada de Ingeniería Informática en el año 2007 en la Universidad de las Ciencias Informáticas (UCI). Actualmente se desempeña como profesora en la misma universidad donde ha impartido asignaturas como Ingeniería del Software, Práctica Profesional, Inteligencia Artificial, curso de pregrado de Analista y Teleinformática. Posee categoría Instructor Recién Graduado. Forma parte del equipo de desarrollo del proyecto SCADA.

Esp. Marisol González Vigoa - >mgonzalezv@infomed.sld.cu

Graduada de medicina en el año 1993 en la facultad de ciencias médicas Ernesto Che Guevara de Pinar del Río. Concluyó la especialidad con Título de Oro en Ginecología y Obstetricia en el año 1998. Diplomante en Ginecología Infanto Juvenil y Salud Reproductiva. Posee categoría docente. Actualmente labora como especialista en Ginecología y Obstetricia en el Policlínico-Hospital Ernesto Che Guevara en la Universidad de las Ciencias Informáticas.

Lic. Ariagnys Quintana Montalvo - >ariagnysqm@infomed.sld.cu

Graduada de Licenciatura en enfermería en el año 2000 en la facultad de medicina número I en Santiago de Cuba. Actualmente se encuentra laborando en el área de ginecología del Policlínico-Hospital Ernesto Che Guevara en la Universidad de las Ciencias Informáticas atendiendo la consulta de ginecología

específicamente y el programa de Citología. En el presente año culmina su maestría en Atención Integrada a la Mujer.

Lic. Maria del Carmen Perojo Medrano -> mperojo@infomed.sld.cu

Graduada el 25 de abril del año 1971 como enfermera especialista en Ginecología y Obstetricia. Actualmente se encuentra laborando en el área de ginecología del Policlínico-Hospital Ernesto Che Guevara en la Universidad de las Ciencias Informáticas atendiendo la consulta de Regulación de la Fecundidad y Regulación Menstrual.

A los tutores Ariel Orta Hernández e Iliana Pérez Pupo por el apoyo brindado durante la investigación y el desarrollo del SIGIGO, además por brindarnos su infinita amistad y confianza.

A la familia de amigos: Nati, Yasniel, Reiniel González, Lienys, Mile, Reimis, Pico, Yadira Bauza, Sahel, Yadira Arguëlles, Rosi, Carlos, Yudito, Dalaiti, Merlin, Lili, Adi, Oslaida, Annia, Yailin, Frank, Maidelys, Mailyn, Dayi, Israel, Yaimara y Yeleynne (las primas) A los varones del grupo, Alexei(jibia), Brayan, Arcel, Rodolphis, Sandro...

Por hacer que nuestros días se tornen "DIFERENTES".

Por estar siempre cuando más los hemos necesitado.

Por lograr que pasen inadvertidos nuestros problemas.

Por brindarnos ese cariño y la alegría que los caracteriza.

Por su incansable preocupación en todo momento y

por la confianza que han depositado en nosotras.

Por aportarnos cada día su granito de arena en aras de convertirnos en mejores estudiantes, mejores profesionales y sobre todo en mejores seres humanos.

A todos los profesores que de una forma u otra han contribuido a nuestra formación como profesionales.

A la Revolución Cubana por permitirnos estudiar y sobretodo en una universidad de excelencia.

Al Comandante en Jefe Fidel Castro Ruz por la genial idea de crear la UCI.

A la UCI:

Por formarnos como profesionales de estos tiempos.

Por darnos la oportunidad de poder contar hoy con una gigantesca, incomparable e inolvidable familia de amigos.

Y por permitirnos hacer realidad un sueño del Comandante Fidel sumándonos a la

"TROPA DE FUTURO".

De Yeni

A mis abuelitos Emilio y Julia que más que mis abuelos junto a mi mamá han sabido ser los mejores padres del mundo.

A mi mamá que en todo momento ha estado conmigo, incluso como una hermana o la mejor de las amigas.

A mis hermanitos Migue y Yedi y a mis primitos Victor, Orbelito, Alianis y Yasmani los cuales recién comienzan a vivir y por los que cada día trato de ser mejor para servirles de guía.

A mi tío Emilíto; a mis tías Belki, Carmita, Tomasita; a mis primos Oilme, Yudit, Taibo que juntos todos han estado presentes en cada paso dado en la Universidad y me han dado todo su apoyo cuando lo he necesitado.

A Gilbe que ha sido otro papá y Amelita que será por siempre mi tía postiza.

A Mary Tania y a Danay por su constante preocupación en estos años de carrera universitaria.

A Eduard, esa persona que conocí gracias a Daniar y a la UCI y que a me ha apoyado tanto cuando mi familia ha estado lejos y por el que me he esforzado tanto para demostrarle que la vida tiene matices diferentes.

A mi papá Toni que gracias a él y a mi mamá hoy estoy viva y que aún cuando no está muy cerca de mí siempre lo llevo presente.

A mi abuelos Cuca y Luis, a Sarita, a mis tías Nilka y Yolanda, a Jorgito y Eylén, a Fredy, Sergio... a toda mi familia de Camajuaní que se que está ahí para lo que necesite.

A Yami mi amiga y compañera de tesis, por el cariño, la confianza y paciencia que ha tenido conmigo.

A la gran familia de hermanos formada por: Lisdaymis, Marliuvis, Diana, Miroslady, Dero, Ada, Diana Beatriz, Ayi, Belí, Yoe, Delvis, Ori, Edgar, Yisel, Ana, Yaneisy, Madelaine que más que los mejores amigos del mundo han sido mis hermanos en mis buenos y sobre todo en mis malos momentos. Y junto a ellos sumar a Daniar, Yinet, Vivi, Arany, Cartaya, Lisset, Yanelis, Lili, Yisel, Yanet, Lida, Ann, Annie, Arelis Yinet, Denise, Erllys...

A esas personas del barrio que hoy estoy segura se gradúan conmigo: Marilín, Cándida, Noelvis, Robe (traquetor), Yoya, Mirelis, Milagrito, Jeo, Mercedes, Nena, Leno, Drulis, Saida, Elia, Ivo, Robe, Yoan, Oneisis, Misnelis, Katy,..... A Bertico, Moñi, Ama, Martha y Florito que aún cuando no estén se que hubiesen disfrutado este momento como si fuese su hija.

A los profes del centro cultural especialmente a Mairela, a Vela y a Jorge por guiarme en el mundo de la actuación y la locución y por su empeño en demostrarme que puedo hacerlo y hacerlo bien.

A los amigos del círculo, la primaria, la secundaria, del inolvidable Tony Santiago. A todos esos primeros amigos UCI del 3104 que ya se graduaron y que nunca olvidaré. Y a otros que se fueron sumando a la interminable lista y que todos juntos hicieron que mi paso por UCI fuese real, inolvidable, incomparable y hasta inevitable.

Mil gracias a todos.

De Yami

A mi mamá por estar siempre a mi lado guiándome por el camino correcto, mi padraastro por comportarse como si fuera mi papá de sangre dándome cariño y apoyo, y mi abuela por aconsejarme siempre y estar pendiente a mis pasos. A todos ellos gracias porque de una forma u otra siempre me han ayudado y apoyando en todo lo que han podido. Que me han permitido ser lo que soy y siempre han tenido confianza en mí.

A mis hermanitos Yumi y Osmani ya que son la fuerza que me impulsa a seguir hacia delante y a sobrepasar todos los malos momentos.

A mis tíos y mis primos que tanto orgullosos se sienten de mí.

A mis amistades por todo el apoyo y cariño que me han brindado en los buenos y malos momentos de mi devenir por la UCI, lo cual apreciaré y valoraré por siempre. Gracias a todos.

A Reiniel por toda la comprensión y apoyo que me ha brindado durante estos 5 años. Por tener tanta paciencia conmigo y ayudarme siempre que lo necesité.

A mis amigas Yadira Bauza, Yadira Arguells, Lienys, Milenis, Yeni, Yaimara, Yelaine, Rosalba, Natalia, Reimis, Dayana, Oslaida, Mayra Merlin, Yailin, Dalayti, Lizandra Díaz y Adi por todo el cariño y la comprensión que me han dado en todo momento siempre que lo he necesitado.

A Luis Ángel por el cariño, atención, preocupación y comprensión que me brindó. Que aunque no esté físicamente desde el lugar donde esté en estos momentos se sentirá muy orgulloso de mí.

A mis amigos Shael, Carlos Pupo, Javier Heredia, Rodolkis, Alexey, Frank, Raydel More, Gustavo, Emilio, Arcel, Hector, Brayan, Albito, Yasniel y Yudislandry, porque siempre estuvieron disponibles para ayudarme cuando más lo necesité.

A todos gracias.

De Yeni

A mi Kimberlee que es el mejor regalo de graduación, mi razón de ser en este momento y por lo tanto quien me impulsa a seguir adelante cuando las circunstancias se toman más difíciles. Lo que más deseo es que este trabajo le sirva guía y ejemplo a seguir.

A mi super abuela chiquitica y a mi adorada tía Juanita que hoy no están físicamente pero estoy completamente segura que estarían muy orgullosas de mí.

A mi toda mi insustituible familia y amigos.

Y a Jehová Dios por permitirme vivir este momento, y por darme una excelente familia y una interminable y envidiable para muchos, familia de amigos.

De Yami:

A mi mamá y a mi padrastro que siempre han sido la luz que guía mi camino, por depositar toda su confianza en que yo iba a ser alguien en la vida, hoy están muy orgullosos de que su Lola ya se graduó.

A mi abuelita que es mi segunda mamá por todos sus consejos, por su amor, su preocupación y por siempre alentarme a seguir adelante y no darme por vencido.

A mis hermanitos porque son los hermanos más bellos del mundo y las personas que más quiero.

Resumen

El presente trabajo plantea una estrategia de cómo resolver los problemas existentes en el departamento de ginecología del hospital Ernesto Che Guevara de la Universidad de las Ciencias Informáticas, en los procesos que complementan el proceso de control y gestión de la información en dicho departamento, a partir del diseño e implementación de una solución de software que satisfaga las necesidades existentes.

Con el objetivo de automatizar los siguientes procesos: gestionar historias clínicas de los pacientes, gestionar remisiones, gestionar hojas de cargo, y la generación de reportes a través de éstas; se analiza cómo se llevan a cabo varias de las actividades que se desarrollan en esta importante área de la universidad. Se presentan las características del sistema que se propone y cómo se construye, según los estándares de desarrollo de software de la metodología Extreme Programming.

El resultado esperado será una versión del sistema capaz de mejorar el desempeño de gran parte de los procesos que tienen lugar hoy en el departamento de ginecología del hospital Ernesto Che Guevara, los cuales se realizan de manera engorrosa, porque se realizan manualmente, lo que provoca un retraso en el trabajos de las especialistas y, por consiguiente, la calidad de la atención de los pacientes no es la adecuada.

Abstract

This paper presents a strategy for solving problems in the department of gynecology hospital Ernesto Che Guevara at the University of Informatics Science, in processes that complement the process control and information management at the department, starting the design and implementation of a software solution that meets the needs.

With the purpose to automate the following processes: managing medical records of patients manage referrals, managing leaves office, and report generation through these; is analyzes how are developed a number of activities that take place on this important area of the university. It is presents the characteristics of the system being proposed and how it is constructed, according the standards of the software development methodology Extreme Programming.

The expected result will be a version of the system capable of improving the performance of much of the processes taking place today in the department of gynecology hospital Ernesto Che Guevara, which are so cumbersome because are performed manually, which causes a delays in the work of specialists and thus the quality of patient care is not proper.

ÍNDICE

INTRODUCCIÓN	1
CAPITULO 1: FUNDAMENTACIÓN TEÓRICA	6
1.1 Introducción	6
1.1 Sistemas Similares vinculados al campo de acción	6
1.1.1 <i>Ejemplos de productos cubanos</i>	7
1.1.2 <i>Ejemplos de productos fuera de Cuba</i>	8
1.3 Metodologías de desarrollo de Software estudiadas	10
1.3.1 <i>Extreme Programming (XP)</i>	11
1.3.2 <i>Rational Unified Process (RUP)</i>	12
1.3.3 <i>Crystal Methodologies</i>	13
1.3.4 <i>Fundamentación de la metodología a utilizar</i>	13
1.4 Herramientas Case (Computer-Aided Software Engineering)	14
1.4.1 <i>Unified Modeling Language (UML)</i>	15
1.4.1.1 <i>Visual Paradigm</i>	15
1.4.1.2 <i>Rational Rose</i>	16
1.4.2 <i>Fundamentación de la herramienta CASE a utilizar</i>	16
1.5 Tipos de aplicaciones estudiadas	17
1.5.1 <i>Aplicaciones Web</i>	17
1.5.2 <i>Aplicaciones de Escritorio</i>	18
1.5.3 <i>Fundamentación del tipo de aplicación a utilizar</i>	18
1.6 Servidores Web	19
1.6.1 <i>Servidor Apache</i>	19
1.6.2 <i>Servidor AOLServer</i>	20
1.6.3 <i>Internet Information Server (IIS)</i>	21
1.6.4 <i>Fundamentación del servidor web a utilizar</i>	21
1.7 Lenguajes de programación	22
1.7.1 <i>Lenguajes del lado del servidor</i>	22
1.7.1.1 <i>Hypertext Pre-processor (PHP)</i>	22
1.7.1.2 <i>Active Server Pages (ASP)</i>	23

1.7.1.3 Java.....	24
1.7.2 Lenguajes del lado del cliente	25
1.7.2.1 HyperText Markup Language (HTML)	25
1.7.2.2 Cascading Style Sheets (CSS)	26
1.7.2.3 JavaScript.....	27
1.7.3 Fundamentación de los lenguajes de programación a utilizar	28
1.8 Lenguaje de Consulta Estructurado (SQL)	28
1.8.1 Sistemas Gestores de Bases de Datos (SGBD)	29
1.8.1.1 MySQL.....	30
1.8.1.2 PostgreSQL.....	30
1.8.2 Fundamentación del sistema gestor de base de datos a utilizar	31
1.9 Fundamentación de la solución propuesta	31
1.10 Conclusiones	32
CAPÍTULO 2: PROPUESTA DEL SISTEMA	33
2.1 Introducción	33
2.2 Arquitectura del Software	33
2.2.1 Patrones de Arquitectura.....	34
2.2.1.1 Arquitectura en capas	34
2.2.1.2 Arquitectura Cliente – Servidor.....	36
2.2.1.3 Patrón Modelo – Vista – Controlador (MVC).....	36
2.3 Patrones de diseño	38
2.3.1 Patrones GRASP.....	38
2.4 Procesos presentes en el negocio	39
2.4.1 Descripción de los procesos presentes en el negocio	39
2.5 Descripción de los procesos que serán objeto de automatización en el sistema	49
2.6 Propuesta del sistema	50
2.6.1 Personal relacionado con el sistema	52
2.7 Conclusiones.....	53
CAPÍTULO 3: EXPLORACIÓN Y PLANIFICACIÓN	54

3.1	Introducción	54
3.2	Fase I: Exploración	54
3.2.1	Historias de Usuario	54
3.3	Fase II: Planificación de la Entrega	64
3.3.1	Estimación de esfuerzos por historias de usuario	65
3.3.2	Plan de iteraciones	66
3.3.3	Plan de duración de las iteraciones	67
3.3.4	Plan de entrega	68
3.4	Conclusiones	70
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA		71
4.1	Introducción	71
4.2	Previo a la implantación (Diseño)	71
4.2.1	Tarjetas Clase, Responsabilidad y Colaboración (CRC)	71
4.3	Fase de Implementación	73
4.3.1	Tareas de programación por iteraciones	74
4.3.1.1	Iteración 1	74
4.3.1.2	Iteración 2	76
4.3.1.3	Iteración 3	¡Error! Marcador no definido.
4.3.2	Arquitectura de hardware (Diagrama de Despliegue)	79
4.4	Fase de prueba	79
4.4.1	Pruebas de aceptación	80
4.5	Conclusiones	95
CONCLUSIONES		96
RECOMENDACIONES		97
REFERENCIAS BIBLIOGRÁFICAS		98
BIBLIOGRAFÍA		101

GLOSARIO DE TÉRMINOS	105
ANEXOS	110
Iteración 1	110
Iteración 2	118
Iteración 3	129

ÍNDICE DE IMÁGENES

Imagen 1: RUP en dos dimensiones.....	12
Imagen 2: Arquitectura en Capas.....	34
Imagen 3: Arquitectura en tres capas	35
Imagen 4: Modelo Cliente-Servidor.....	36
Imagen 5: Modelo-Vista-Controlador	37
Imagen 6: Diagrama de despliegue	79

ÍNDICE DE TABLAS

Tabla 1: Personas relacionadas con el sistema	52
Tabla 2: Representación de una historia de usuario	55
Tabla 3: Representación de la Historia de Usuario Nro. 1	55
Tabla 4: Representación de la Historia de Usuario Nro. 2	56
Tabla 5: Representación de la Historia de Usuario Nro. 3	56
Tabla 6: Representación de la Historia de Usuario Nro. 4	57
Tabla 7: Representación de la Historia de Usuario Nro. 5	57
Tabla 8: Representación de la Historia de Usuario Nro. 6	58
Tabla 9: Representación de la Historia de Usuario Nro. 7	58
Tabla 10: Representación de la Historia de Usuario Nro. 8	59
Tabla 11: Representación de la Historia de Usuario Nro. 9	59
Tabla 12: Representación de la Historia de Usuario Nro. 10	60
Tabla 13: Representación de la Historia de Usuario Nro. 11	60
Tabla 14: Representación de la Historia de Usuario Nro. 12	61
Tabla 15: Representación de la Historia de Usuario Nro. 13	61
Tabla 16: Representación de la Historia de Usuario Nro. 14	62
Tabla 17: Representación de la Historia de Usuario Nro. 15	62
Tabla 18: Representación de la Historia de Usuario Nro. 16	63
Tabla 19: Representación de la Historia de Usuario Nro. 17	63
Tabla 20: Representación de la Historia de Usuario Nro. 18	64

Tabla 21: Estimación de esfuerzos por Historia de Usuario.....	65
Tabla 22: Plan de duración de las iteraciones	67
Tabla 23: Funcionalidades por módulos	69
Tabla 24: Plan de duración entrega	70
Tabla 25: Tarjeta CRC Clase Paciente.....	72
Tabla 26: Tarjeta CRC Clase Historia Clínica.....	72
Tabla 27: Tarjeta CRC Clase Hoja de Cargo.....	72
Tabla 28: Tarjeta CRC Clase Remisión.....	73
Tabla 29: Tarjeta CRC Clase Reporte	73
Tabla 30: Tiempo real de implementación por módulo (iteración 1)	74
Tabla 31: Tarea de programación por historias de usuario (Iteración 1)	74
Tabla 32: Tiempo real de implementación por módulo (iteración 2)	76
Tabla 33: Tarea de programación por historias de usuario (Iteración 2)	76
Tabla 34: Tiempo real de implementación por módulo (iteración 3)	77
Tabla 35: Tarea de programación por historias de usuario (Iteración 3)	78
Tabla 36: Prueba de aceptación Nro.1 (Iteración 1)	80
Tabla 37: Prueba de aceptación Nro.2 (Iteración 1)	81
Tabla 38: Prueba de aceptación Nro.3 (Iteración 1)	82
Tabla 39: Prueba de aceptación Nro.4 (Iteración 1)	82
Tabla 40: Prueba de aceptación Nro.5 (Iteración 1)	83
Tabla 41: Prueba de aceptación Nro.6 (Iteración 1)	84
Tabla 42: Prueba de aceptación Nro.7 (Iteración 1)	84
Tabla 43: Prueba de aceptación Nro.8 (Iteración 1)	85

Tabla 44: Prueba de aceptación Nro.9 (Iteración 2).....	85
Tabla 45: Prueba de aceptación Nro.10 (Iteración 2).....	86
Tabla 46: Prueba de aceptación Nro.11 (Iteración 2).....	87
Tabla 47: Prueba de aceptación Nro.12 (Iteración 2).....	87
Tabla 48: Prueba de aceptación Nro.13 (Iteración 2).....	88
Tabla 49: Prueba de aceptación Nro.14 (Iteración 2).....	89
Tabla 50: Prueba de aceptación Nro.15 (Iteración 2).....	89
Tabla 51: Prueba de aceptación Nro.16 (Iteración 2).....	90
Tabla 52: Prueba de aceptación Nro.17 (Iteración 2).....	91
Tabla 53: Prueba de aceptación Nro.18 (Iteración 2).....	91
Tabla 54: Prueba de aceptación Nro.19 (Iteración 2).....	92
Tabla 55: Prueba de aceptación Nro.20 (Iteración 3).....	93
Tabla 56: Prueba de aceptación Nro.21 (Iteración 3).....	93
Tabla 57: Prueba de aceptación Nro.22 (Iteración 3).....	94
Tabla 58: Prueba de aceptación Nro.23 (Iteración 3).....	95

ÍNDICE DE ANEXOS

Anexo 1: Tarea de Programación: Mostrar interfaz “Crear Perfil”	110
Anexo 2: Tarea de Programación: “Crear perfil”	110
Anexo 3: Tarea de Programación: “Mostrar interfaz autenticarse”	111
Anexo 4: Tarea de Programación: “Insertar Datos”	111
Anexo 5: Tarea de Programación: “Comprobar datos”	112
Anexo 6: Tarea de Programación: “Realizar búsqueda”	113
Anexo 7: Tarea de Programación: “Comprobar datos”	113
Anexo 8: Tarea de Programación: “Insertar datos”	114
Anexo 9: Tarea de Programación: “Mostrar interfaz buscar paciente”	114
Anexo 10: Tarea de Programación: “Mostrar interfaz crear hoja de cargo”	115
Anexo 11: Tarea de Programación: “Crear hoja de cargo”	115
Anexo 12: Tarea de Programación: “Mostrar interfaz buscar paciente”	115
Anexo 13: Tarea de Programación: “Mostrar interfaz crear remisión”	116
Anexo 14: Tarea de Programación: “Crear remisión”	116
Anexo 15: Tarea de Programación: “Mostrar interfaz buscar paciente”	117
Anexo 16: Tarea de Programación: “Mostrar interfaz crear historia clínica”	117
Anexo 17: Tarea de Programación: “Crear historia clínica”	118
Anexo 18: Tarea de Programación: “Mostrar interfaz buscar perfil”	118
Anexo 19: Tarea de Programación: “Realizar búsqueda”	119
Anexo 20: Tarea de Programación: “Mostrar interfaz buscar paciente”	119
Anexo 21: Tarea de Programación: “Realizar búsqueda”	119

Anexo 22: Tarea de Programación: “Comprobar datos”	120
Anexo 23: Tarea de Programación: “Mostrar datos”	120
Anexo 24: Tarea de Programación: “Mostrar interfaz generar reportes”	121
Anexo 25: Tarea de Programación: “Generar reporte”	121
Anexo 26: Tarea de Programación: “Mostrar interfaz buscar hoja de cargo”	122
Anexo 27: Tarea de Programación: “Realizar búsqueda”	122
Anexo 28: Tarea de Programación: “Comprobar datos”	122
Anexo 29: Tarea de Programación: “Mostrar datos”	123
Anexo 30: Tarea de Programación: “Mostrar interfaz buscar paciente”	123
Anexo 31: Tarea de Programación: “Realizar búsqueda”	124
Anexo 32: Tarea de Programación: “Comprobar datos”	124
Anexo 33: Tarea de Programación: “Modificar datos”	125
Anexo 34: Tarea de Programación: “Buscar historia clínica”	125
Anexo 35: Tarea de Programación: “Modificar datos”	126
Anexo 36: Tarea de Programación: “Mostrar interfaz buscar remisión”	126
Anexo 37: Tarea de Programación: “Realizar búsqueda”	126
Anexo 38: Tarea de Programación: “Comprobar datos”	127
Anexo 39: Tarea de Programación: “Mostrar datos”	127
Anexo 40: Tarea de Programación: “Mostrar interfaz buscar paciente”	128
Anexo 41: Tarea de Programación: “Realizar búsqueda”	128
Anexo 42: Tarea de Programación: “Comprobar datos”	128
Anexo 43: Tarea de Programación: “Mostrar datos”	129
Anexo 44: Tarea de Programación: “Mostrar interfaz buscar perfil”	129

Anexo 45: Tarea de Programación: “Realizar búsqueda”	130
Anexo 46: Tarea de Programación: “Comprobar datos”	130
Anexo 47: Tarea de Programación: “Eliminar perfil”	131
Anexo 48: Tarea de Programación: “Mostrar interfaz buscar paciente”	131
Anexo 49: Tarea de Programación: “Realizar búsqueda”	131
Anexo 50: Tarea de Programación: “Comprobar datos”	132
Anexo 51: Tarea de Programación: “Eliminar datos”	132
Anexo 52: Tarea de Programación: “Buscar hoja de cargo”	133
Anexo 53: Tarea de Programación: “Modificar datos”	133
Anexo 54: Tarea de Programación: “Mostrar interfaz buscar paciente”	134
Anexo 55: Tarea de Programación: “Realizar búsqueda”	134
Anexo 56: Tarea de Programación: “Comprobar datos”	134
Anexo 57: Tarea de Programación: “Eliminar datos”	135

Introducción

A partir de 1996 se dieron los primeros pasos para el ordenamiento de un trabajo continuo con el fin de impulsar el uso y desarrollo de las Tecnologías de la Información y las Comunicaciones en el país, estos pasos, aunque discretos, condujeron a la creación del Ministerio de la Informática y las Comunicaciones en enero del 2000 con la misión fundamental de fomentar el uso masivo de las Tecnologías de la Información y las Comunicaciones para satisfacer las necesidades de información y conocimiento de todas las personas y esferas de la sociedad, tomando como sustento los cuatro pilares fundamentales del proceso revolucionario cubano: la educación, la salud, la seguridad social y la cultura. [1]

En sus inicios, trabajar por la informatización de estos cuatro pilares, resultó ser más que un sueño un verdadero reto para el pueblo cubano. Específicamente "La Salud" en nuestro país, es un sector que es paradigma y noticia para numerosos países del mundo, debido al progreso alcanzado desde los inicios de la Revolución. Su informatización, es hoy, una de las tareas más significativas y complejas a las que se enfrentan los profesionales de la informática en Cuba.

En tareas como éstas, se encuentran inmersas empresas como Softel y Copextel donde se implementan disímiles proyectos y promueven soluciones tanto al país como a otras entidades extranjeras. Así como la Universidad de Ciencias Informáticas (UCI), la cual fue creada con el objetivo de formar todo un personal, con un elevado nivel técnico y profesional que apoye y sirva de sostén al proceso de informatización llevado a cabo en nuestra sociedad incluyendo tan importante sector.

La UCI, universidad surgida como programa de la Revolución cuenta con el Policlínico-Hospital Ernesto Che Guevara, el cual queda involucrado dentro de este proceso. Se ha categorizado como Policlínico-Hospital porque presta servicios que no son brindados por un policlínico común pero tampoco son todos los ofrecidos por los hospitales.

El Policlínico-Hospital Ernesto Che Guevara de la Universidad de Ciencias Informáticas contó con el servicio de Ginecología y Obstetricia desde abril del 2004, incluyendo a partir del presente año (2009) el servicio Materno-Infantil. Actualmente estos dos servicios prestan atención a cerca de 7000 pacientes

aproximadamente entre estudiantes, trabajadoras y profesoras internas lo que representa el 100% de la población fémina de la universidad.

Estas pacientes se encuentran distribuidas entre las consultas que conforman a dicho departamento dígase: Regulación de la Fecundidad, Ginecología, Obstetricia (atención a embarazadas) y Regulación Menstrual. Las mismas están estrechamente relacionadas unas con otras, y al mismo tiempo poseen relación con Estomatología y Psicología, con los laboratorios de ultrasonidos y complementarios, los diez consultorios de la familia, las direcciones de las facultades (10 decanatos) y con el departamento de estadísticas de la propia institución. Adicionándole genética y nutrición; consultas con las que no cuenta actualmente el Policlínico-Hospital.

El manejo y gestión de toda la información requerida resulta tedioso y agobiante debido a la forma en que se procede con la misma. Por ejemplo:

Para cada una de estas consultas existe una Hoja de Cargo con un formato específico. Cada vez que acude un paciente a consulta, la doctora introduce los datos correspondientes a la misma. Las Hoja de Cargo, son llevadas al finalizar cada consulta, al departamento de estadísticas donde son archivadas durante cinco años. De ella se extraen datos y se realizan determinados cálculos con el fin de obtener de reportes importantes.

En la consulta de Regulación de la Fecundidad existe una carpeta en forma de libro, donde cada una de sus hojas, es la historia clínica perteneciente a una paciente. En el momento en que la paciente pasa a ser atendida, la enfermera deberá hacer una búsqueda de su historia hoja a hoja en el libro que le pertenezca. Es válido aclarar que existe un libro para las estudiantes (uno para cada facultad), uno para profesoras internas y otro para trabajadoras internas.

En Regulación Menstrual y Ginecología deberá estar presente la historia clínica del consultorio médico para que una paciente sea atendida. En reiteradas ocasiones esta historia está ausente por razones que le son ajenas a la especialista, tomando la decisión de atender a la paciente y escribir la información referente en una hoja de papel corriendo el riesgo de perderla.

En la consulta de Obstetricia: es necesario también la presencia de la historia clínica del consultorio médico solo que además de ésta, la paciente deberá portar su Tarjetón de Embarazada. Tarjetón que igualmente corre el riesgo de perderse o destruirse parcial o totalmente.

Existe de la misma manera extravíos de los resultados de exámenes importantes como ultrasonidos, radiologías, pruebas citológicas y análisis complementarios que se archivan dentro de las historias clínicas de los consultorios supuestamente presilladas o en cajitas denominadas buzones (buzón citológico, buzón de complementarios).

Como se puede apreciar contar con que la recolección de la información, como es la entrevista médica, el examen físico y el empleo de instrumentos complementarios; el registro de la información, como la confección de historias clínicas, la integración sindrómica y la identificación de problemas diagnósticos; la confección y manipulación de la Hoja de Cargo así como de sus reportes derivados y el archivo de los resultados de exámenes se encuentren de forma centralizada, no duplicada y disponible en dicho departamento se ha convertido en un sueño aún no hecho realidad para la especialista y las enfermeras que en el área de ginecología radican.

Debido a que no existe una aplicación que automatice estos procesos; de manera que se agilice el trabajo en dicho departamento y después de haber analizado la problemática expuesta con anterioridad se procede a plantear el siguiente **Problema a resolver:**

¿Cómo resolver los problemas de gestión de la información en el departamento de ginecología en Policlínico-Hospital Ernesto Che Guevara de la Universidad de las Ciencias Informáticas?

Objeto de Estudio: El proceso de gestión de información en el área de Ginecología.

Campo de Acción: El proceso de gestión de información dentro del área de Ginecología del Policlínico-Hospital Ernesto Che Guevara.

Objetivo General: Diseñar e implementar un sistema de gestión de información ginecológica para hospital Ernesto che Guevara de la Universidad de las Ciencias Informáticas.

Objetivos específicos:

- ❖ Realizar un estudio de sistemas similares existentes tanto nacionales como internacionales.
- ❖ Determinar la metodología de desarrollo de software, los lenguajes de modelado y programación así como el servidor web y el sistema gestor de base de datos a utilizar.
- ❖ Diseñar e implementar las funcionalidades del sistema.

Idea a defender:

Con el desarrollo de un sistema que gestione la información en el departamento de ginecología del hospital Ernesto Che Guevara de la Universidad de las Ciencias Informáticas se brindará:

- A los especialistas del departamento una manera de procesar la información de forma automatizada que agilizará su trabajo.

Tareas:

- ❖ Consultar y analizar la documentación de sistemas similares y contactar al equipo de desarrollo siempre que sea posible.
- ❖ Consultar documentos sobre metodologías de desarrollo de software, lenguajes de modelado y programación, herramientas CASE, servidores web y sistemas gestores de base de datos. Especificar los más adecuado para el desarrollo de la aplicación.
- ❖ Consultar documentación sobre arquitectura de software y patrones de diseño. Definir la arquitectura sobre la cual estará basado el producto.
- ❖ Seleccionar y describir las historias de usuarios.
- ❖ Dividir las historias de usuario en tareas de programación y describirlas.
- ❖ Confeccionar las tarjetas Clase-Colaboración-Responsabilidad.

- ❖ Confeccionar el diagrama de despliegue.

El contenido del presente trabajo de diploma se estructura en 4 capítulos:

El Capítulo 1. Fundamentación teórica: Incluye un análisis de algunos sistemas existentes que poseen similitud con el que se desea implementar. Un estudio de las metodologías de desarrollo de software, de los lenguajes de modelado y programación, de las herramientas, servidores web y sistemas gestores de base de datos más utilizados en la actualidad con el fin de proponer lo más adecuadas para la solución del problema.

El Capítulo 2. Descripción y análisis de la solución propuesta: Contiene un minucioso estudio sobre arquitectura de software, así como de los estilos arquitectónicos y los patrones de diseño más usados en la actualidad. Se realiza una descripción detallada del flujo de procesos presentes en el negocio y de los procesos que serán objeto de automatización. Concluyendo con la propuesta del sistema a desarrollar.

El Capítulo 3. Exploración y Planificación: Contendrá los artefactos generados en las fases de exploración y planificación, fases propias de la metodología Extreme Programming.

El Capítulo 4. Implementación y Prueba: Capítulo que engloba los artefactos generados por la metodología de desarrollo del software propuesta en su fase: implementación y prueba.

Capítulo 1: Fundamentación teórica

“Todo en el software cambia. Los requisitos cambian. El diseño cambia. El negocio cambia. La tecnología cambia. El equipo cambia. Los miembros del equipo cambian. El problema no es el cambio en sí mismo, puesto que sabemos que el cambio va a suceder; el problema es la incapacidad de adaptarnos a dicho cambio cuando éste tiene lugar. ”

Kent Beck. [2]

1.1 Introducción

Cuando se va a dar comienzo al desarrollo de un producto software, es de suma importancia definir claramente el tipo de aplicación y su ambiente de desarrollo. En el presente capítulo se abordará la fundamentación teórica para el desarrollo del Sistema de gestión de información para el área de ginecología del Policlínico-Hospital Ernesto Che Guevara de la Universidad de las Ciencias Informáticas. Para ello se realizará un minucioso análisis de los sistemas similares existentes tanto internacionales como nacionales relacionados con el campo de acción, un estudio detallado de las metodologías de desarrollo de software, de los lenguajes de modelado y programación, de las herramientas y de las plataformas de desarrollo que las soportan, de los diferentes tipos de servidores web y de los gestores de base de datos, con la finalidad de proponer la más adecuada para la solución del problema.

1.1 Sistemas Similares vinculados al campo de acción

En la segunda mitad del siglo XX, se perfilan los primeros sistemas de información médica, sistemas que posteriormente dieron lugar a los sistemas de información hospitalaria, tan indispensables en la actualidad. Su impacto en las instituciones de salud es notable, ya que buscan elevar la calidad de la atención al paciente, de los servicios brindados y aplicar la información obtenida a las áreas de la investigación, la clínica, la docencia, la administración y desde luego abatir costos y elevar la

productividad. Las oportunidades que brindan, han provocado que el interés por las soluciones informáticas para el sector de la salud hayan aumentado con creces, donde la comunidad científica internacional juega hoy su papel primordial; dirigiéndose a lograr la integración de dichas aplicaciones o desarrollar una que las englobe a todas.

Existen disímiles y variados productos y versiones de Sistemas de Información Hospitalaria (SIH) en todo el globo terráqueo y Cuba como parte de él no está exenta. A continuación se hace alusión a algunos de éstos y la empresa que los desarrolló.

1.1.1 Ejemplos de productos cubanos

Empresa SOFTEL: Tiene como misión proporcionar soluciones informáticas que eleven la eficiencia de los servicios de salud con personal y tecnologías de avanzada. [3]

Ha desarrollado productos como: Galen Lab, Galen Hospital, Galen Banco de Sangre, Galen Farmacia, Galen HemoRed, Quirófano, Cuerpo de Guardia. El hospital “Hermanos Ameijeiras” cuenta en estos momentos para la inscripción, registro de los pacientes y movimiento hospitalario con el sistema “GALEN HOSPITAL.

- Producto Galen Lab: Dirigido a la gestión de los medios de diagnóstico y consultas, facilita la solicitud de exámenes, registro y evaluación de los resultados obtenidos y la generación de información estadística. Diseñado para ser utilizado por técnicos, médicos y personal administrativo de medios de diagnóstico y consultas para optimizar el trabajo y elevar la eficiencia. [4]
- Este producto cuenta con componentes tales como: Gestión de Servicios, Consulta, Agenda, Auto-analizadores, Imágenes y Trazados Gráficos.
- Producto Galen Hospital: Orientado a la informatización de la gestión de pacientes como elemento básico en la atención médica, optimizar el uso del personal, aumentar la calidad de los servicios hospitalarios y disminuir sus costos. Posee componentes como: Inscripción /Admisión, Archivo,

Estadísticas, Información, Turnos de Consulta ambulatoria, Dietas y Certificados y Gestión de Salas.

Empresa CEDISAP: (Centro de Desarrollo Informático para la Salud Pública): Institución del Sistema Nacional de Salud de la República de Cuba, creada en 1988. Los sistemas creados por este centro se dividen en Sistemas Hospitalarios y Sistemas de Atención Primaria.

Dentro de los Sistemas Hospitalarios se encuentran: el Sistema Integral de Administración Hospitalaria (SIAH), Medisys en sus Versiones: FoxPro 2.6 y Cliente-Servidor.

- **Producto Medisys** (Versión Cliente-Servidor): Permite registrar, controlar y procesar la información necesaria del paciente para la realización de los servicios de salud que requiera, así como la información de la planificación, ejecución y supervisión de los servicios de salud prestados por la institución para facilitar la toma de decisiones con vistas a mejorar la calidad y eficiencia de los mismos.

Este Sistema Informativo Hospitalario funciona a través de una Red Local en plataforma Cliente - Servidor, que enlaza todas las computadoras existentes en la institución en un ambiente que permite comunicarse entre ellas y compartir los ficheros de información desde cualquier punto, teniendo en cuenta las funciones que debe cumplir cada uno de los usuarios, así como los niveles de acceso que le sean permitidos a la misma. [5]

Algunos módulos que lo conforman son: Registros médicos, Información al Público, Turnos de consulta externa, Medicamentos, Laboratorio clínico, Laboratorio de microbiología e Imagenología.

1.2.2 Ejemplos de productos fuera de Cuba

Empresa QSOFT: Nace en 1995 como empresa de Software especializada en el sector de la Salud. El equipo de QSOFT está formado, en su mayoría, por Ingenieros en Informática con amplia experiencia en el campo de la Sanidad.

- Producto GineSalus: Permite la gestión integral de Clínicas Ginecológicas y Centros de Fertilidad. Incluye, por ello, todo la gestión relativa a citación de pacientes (Agendas), información clínica y captura de imágenes (Historiales Clínicos) así como la facturación, cobros y presupuestos (Gestión Económica). El Software ha sido desarrollado desde 1997 por Médicos y expertos en gestión de Clínicas y Centros Ginecológicos, y entre las principales ventajas que aporta destacan el ahorro de tiempo en la gestión diaria, la eliminación de archivos de papel así como facilitar al centro el cumplimiento de la legislación de protección de datos personales de los pacientes. [6]

Empresa Konocer: Dispone de aplicaciones listas para poner en marcha en las siguientes áreas: médica, Comercio electrónico, Prensa, Hoteles entre otros.

- Producto GineControl: Desarrollado por Konocer durante el 2004-2005. Software para los centros de atención ginecológica. Cuenta con módulo como:
 - Control de citas
 - Gestión de historias clínicas.
 - Sistema de facturación a aseguradoras.
 - Control de hospitalización
 - Informes de rendimiento
 - Base de datos de conocimiento (historias, imágenes, tratamientos)
 - Informe de visitas. [7]
- Producto Ginepack: Ofrece un conjunto de aplicaciones informáticas orientadas a ofrecer la ayuda necesaria para la gestión de diferentes áreas dentro de la asistencia ginecológica. Permite la exportación de información para su uso en paquetes estadísticos, así como la integración con otras aplicaciones informáticas. Incluye los módulo:
 - * Reproducción asistida
 - * Planificación familiar

- * Registro de tumores ginecológicos
- * Consulta de menopausia
- * Consulta de ginecología
- * Consulta de suelo pélvico
- * Consulta para diagnóstico en alta resolución
- * Sistema de formación continua a través de internet (Praxis)
- * Módulo de acceso a través de internet [8]

Los Sistemas de Información permiten a las organizaciones sanitarias mejorar drásticamente tanto sus procesos clínicos como administrativos. Sin embargo, el resultado final se relaciona directamente con factores que dependen de los servicios: tiempo y coste de la implementación, adopción y utilización del sistema por parte del usuario, disponibilidad del sistema, y satisfacción final del cliente.

Es evidente que éstos sistemas descritos con anterioridad sirven como apoyo a instituciones dentro del sector de la salud, pero también es real que todos los centros de asistencia médica no poseen las mismas características y por tanto en ocasiones se hace necesario modificar los productos ya existentes o crearlos nuevos en dependencia de las necesidades y expectativas los mismos.

El Policlínico-Hospital Ernesto Che Guevara de La Universidad de Ciencias Informáticas no cuenta con un software que se encargue de la gestión de la información del departamento de ginecología y como consecuencia, surgió la necesidad de desarrollar un nuevo sistema que se haga cargo de dicha gestión.

1.3 Metodologías de desarrollo de Software estudiadas

El desarrollo de un software es sin dudas una tarea difícil y como resultado de esto surgió una alternativa: La Metodología.

Las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. En cualquier proceso de desarrollo de software esta define: “Qué” debe hacer el software, “Quien” debe realizar cada actividad, “Cuándo” hacerla y “Cómo” se debe hacer. El objetivo fundamental del proceso de desarrollo es elevar la calidad del software en cada una de las fases por las que transita mediante una mayor transparencia y control sobre el mismo. No existe una metodología de software universal, sino que cada equipo es responsable de elegirla según las características de su proyecto. [9]

En lo que a proceso de desarrollo respecta se conocen hoy los llamados métodos pesados (conocidos también como tradicionales, robustos o no ágiles) y los llamados métodos ligeros (conocidos como ágiles). Los métodos pesados están dirigidos al control del proceso en toda su magnitud demostrando ser efectivos y necesarios en súper-proyectos (respecto a tiempo y recurso). Mientras que los ágiles tratan de adaptarse a la realidad del propio desarrollo o sea son menos orientados al documento y más al código.

1.3.1 Extreme Programming (XP)

Extreme Programming nació como una nueva disciplina de desarrollo de software en el año 1996 y fue dada a conocer en 1999 en el del libro “Extreme Programming Explained” de Kent Beck causando gran impacto entre el colectivo de programadores del mundo.

XP se encuentra incluida específicamente dentro del grupo de Metodologías Ágiles y se centra fundamentalmente en la retroalimentación sistemática y fluida entre el cliente y el equipo de desarrollo, en la simplicidad en las soluciones implementadas y en el valor y la disposición para enfrentar el cambio. Es utilizada fundamentalmente en proyectos con requisitos imprecisos, es decir que pueden variar con facilidad y que tienen un alto riesgo técnico, que cuentan con un equipo de desarrollo pequeño y multidisciplinario.

Extreme Programming es una metodología eficiente, de bajo riesgo y flexible; cuenta con una serie de principios, valores y 12 prácticas propiciando un aumento en la productividad y generación del producto.

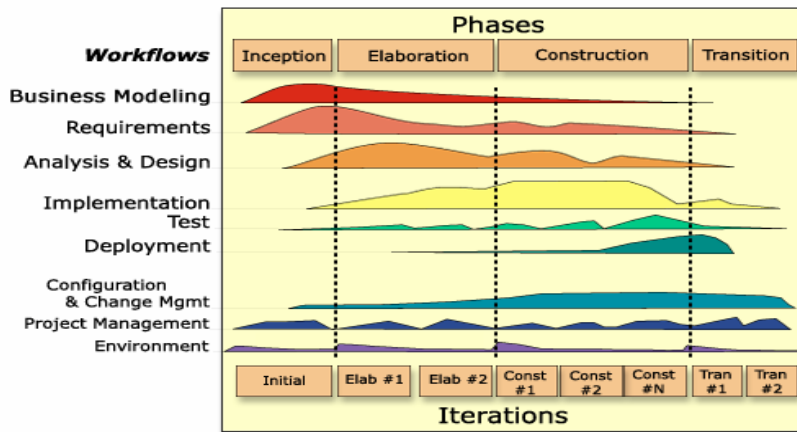
Muchas de estas prácticas han sido reconocidas por la industria como las mejores en años. XP las utiliza al extremo, razón por la cual es conocida como: Programación Extrema.

1.3.2 Rational Unified Process (RUP)

El proceso unificado de desarrollo (RUP) es una metodología para la ingeniería de software que proporciona una familia de técnicas que soportan el ciclo completo de desarrollo de software. El resultado es un proceso basado en componentes, dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental.

RUP en su ciclo de desarrollo cuenta con 4 fases y 9 flujos para sus actividades las cuales se desarrollaran siguiendo un modelo en cascada. A los 6 primeros flujos se les conoce como “Flujos Ingenieriles” o de “Trabajo Básico” y a los 3 restantes de “Apoyo” o “Soporte”.

Imagen 1: RUP en dos dimensiones



1.3.3 Crystal Methodologies

La metodología Crystal desarrollada por Alistair Cockburn, es una de las más ligeras y adaptables a los enfoques de desarrollo de software. Está compuesta por una familia de metodologías por ejemplo Crystal Clear (3 a 8 miembros), Crystal Yellow y Crystal Orange (25 a 50 miembros) cuyas características son impulsadas por factores como: el tamaño del equipo, la criticidad del sistema y las prioridades de los proyectos.

Al igual que otras metodologías ágiles, Crystal los promueve principios siguientes:

- Entrega frecuente de trabajo de software
- Alta participación de los usuarios
- Adaptabilidad
- Eliminación de la burocracia y la o las distracciones

1.3.4 Fundamentación de la metodología a utilizar

Teniendo en cuenta las características de las metodologías de desarrollo antes mencionadas, se pudo apreciar que aún cuando se centran en la producción de software y que el proceso se establece para aumentar la calidad del producto y la eficiencia de los desarrolladores, poseen sus diferencias.

La metodología RUP está destinada para grandes proyectos y equipos de desarrollo en cuanto a tamaño y duración. Basa su trabajo fundamentalmente en la documentación del software, el cliente puede encontrarse lejos de la estación de desarrollo y hay una gran dependencia entre sus grupos de trabajo. Crystal Methodologies está basada en iteraciones cortas, preferiblemente para proyectos pequeños y con un reducido grupo de trabajo, aunque varía en dependencia del color de Crystal que se escoja entre la familia que la conforman.

En cambio, a pesar de que Extreme Programming posee características similares a Crystal Methodologies se decidió utilizar esta metodología debido a que cuenta con buena documentación, a que hay un personal calificado en la universidad con conocimientos sobre la misma y por las peculiaridades presentadas por el proyecto como son:

Destinada a proyectos de corto plazo: el sistema SIGIGO fue concebido en un plazo aproximado a 8 semanas.

Elevado riesgo de desarrollo en cuanto a tiempo y al constante cambio de requisitos: inicialmente la realización de SIGIGO se estimó para un plazo no superior a las 12 semanas, y a pesar de las constantes peticiones de cambio por parte del cliente hubo que reducir su tiempo de entrega.

El cliente forma parte del equipo de desarrollo: la estación de trabajo del cliente de SIGIGO se encuentra dentro de la propia UCI, lo cual propició una constante retroalimentación entre éste y su equipo de desarrollo.

Pequeño equipo de desarrollo: La aplicación contó solamente con 2 desarrolladores, lo que condujo a un intercambio de responsabilidades en determinadas ocasiones.

Propiedad del código es colectiva: los programadores pudieron realizar cambios en cualquier parte del código siempre que fue necesario, propiciando la comunicación entre los mismos.

Se basa en iteraciones pequeñas: SIGIGO contó con 3 iteraciones y la duración de ellas no excedieron las 3 semanas.

1.4 Herramientas Case (Computer-Aided Software Engineering)

Las Herramientas Asistidas por Computadoras para la Ingeniería de Sistemas (CASE) son un conjunto de aplicaciones destinadas a mejorar la productividad del analista, aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero y por lo tanto a mejorar la eficiencia y la calidad del sistema de información.

1.4.1 Unified Modeling Language (UML)

UML es un lenguaje de modelado visual para sistemas, principalmente sistemas de software orientados a objetos. Hoy en día, está considerado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Mediante UML es posible establecer la serie de requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código.

Aunque Unified Modeling Language es un lenguaje, posee más características visuales que programáticas, lo que facilita la fácil participación e intercomunicación de los integrantes de un equipo multidisciplinario, dígase: analistas, diseñadores, especialistas de área y desde luego los programadores. [10]

1.4.1.1 Visual Paradigm

Visual Paradigm es una herramienta CASE que utiliza “UML”: como lenguaje de modelado. Se caracteriza principalmente por su robustez, usabilidad y portabilidad. Soporta el ciclo de vida completo del desarrollo de software dígase análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permitiendo dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

Esta herramienta permite realizar ingeniería tanto directa como inversa, a partir de un modelo relacional en SQL Server, MySQL, PostgreSQL, es capaz de desplegar todas las clases asociadas a las tablas (siguiendo el patrón de diseño Una Clase-Una Tabla). Soporta múltiples usuarios trabajando sobre el mismo proyecto, permite control de versiones y genera la documentación automáticamente en formatos como Web o .PDF. Es libre y multiplataforma.

1.4.1.2 Rational Rose

Rational Rose es una poderosa herramienta software para el Modelado Visual de sistemas software. Utiliza UML como lenguaje de modelado. Permite Especificar, Analizar, Diseñar el sistema antes de Codificarlo. Cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases.

Rational Rose permite que se mantenga la consistencia de los modelos del sistema software, el chequeo de la sintaxis UML, la generación de documentación automática, generación de código a partir de los modelos, realizar la ingeniería inversa (crear modelo a partir código) y trabajo en grupo. Propone la utilización de cuatro tipos de modelo para realizar el diseño de un sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software. Es privativo y no es multiplataforma.

1.4.2 Fundamentación de la herramienta CASE a utilizar

Tanto Visual Paradigm como Rational Rose son herramientas CASE que utilizan el lenguaje UML para el modelado visual de sistemas. Para el modelado de SIGIGO se decidió utilizar la herramienta CASE Visual Paradigm debido a que: cuenta con un amigable entorno de desarrollo, es libre y multiplataforma características de suma importancia ya que la universidad cuenta con su licencia y que en un por cierto elevado de computadoras se encuentran instalados los sistemas operativos Windows y Linux incluso diferentes versiones de éstos.

1.5 Tipos de aplicaciones estudiadas

1.5.1 Aplicaciones Web

A la aplicación a la que los usuarios acceden a un servidor Web a través de alguna Intranet o del propio Internet mediante un navegador Web operando como cliente ligero se le conoce como: Aplicación Web.

En este tipo de aplicación el gasto se reduce, puesto que su mantenimiento se concentra en el servidor. Es su proveedor quien se preocupa por mantenerla siempre disponible y actualizada a cambio de una cuota fija y razonable. A continuación se presentan algunas de sus ventajas:

- Permite una administración de forma centralizada.
- Permite tener la información centralizada eliminando la necesidad de hacer replicas periódicas.
- Una reducción del capital humano para el mantenimiento del sistema.
- Brinda la posibilidad de actualizar y mantener las aplicaciones sin la necesidad de distribuir e instalar software en un variado número de clientes.
- Posibilita que el cliente cuente con las actualizaciones en el instante en que se realizan.
- Permite contar con clientes independientes de tecnología y estaciones de trabajo.
- Necesita solamente de un navegador Web para acceder al sistema.
- Permite la navegación desde cualquier punto.
- Permite ahorrar al no tener que invertir en costosas infraestructuras de comunicación que en muchos casos podrían ser sencillamente imposibles.

Mediante sistemas informáticos sobre la Web se obtienen resultados superiores en cuanto al trabajo con información. Sin duda, las aplicaciones Web tienen aún un largo camino por recorrer, pero son ya una opción muy interesante, especialmente, en tareas dónde las bases de datos juegan un papel predominante y los usuarios se encuentran dispersos.

1.5.2 Aplicaciones de Escritorio

Toda aplicación que ha sido desarrollada para ser ejecutada en una plataforma específica, ya sea Windows, GNU/Linux ó Mac se le denomina: Aplicación de escritorio.

Este tipo de aplicación se desarrolla generalmente para cubrir necesidades específicas de las empresas como la contabilidad o gestión de personal.

Entre las ventajas de las aplicaciones de escritorio se destacan:

- Mayor capacidad gráfica visual
- Menor tiempo de respuesta (aplicación más rápida)
- Mayor personalización

El problema principal radica en que el desarrollo de este tipo de aplicación sobre una plataforma, normalmente, implica que "no" pueda ser ejecutada en otras. El duplicado de los datos por la falta de centralización de los mismos. Implica su instalación en cada uno de los clientes. Trae problemas a la hora de realizar alguna actualización o corrección al programa debido a que las instalaciones están distribuidas. Dificultad para configurar las estaciones ya que cada usuario tiene necesidades diferentes.

Existen aplicaciones de escritorio con tecnología Web y GNU Consultores se ha especializado en su desarrollo basadas en tecnologías Web como: AJAX, DHTML y CSS.

1.5.3 Fundamentación del tipo de aplicación a utilizar

Por parte del equipo de desarrollo se decidió desarrollar una aplicación Web puesto que:

Se reducen los gastos en gran medida: al ser SIGIGO una aplicación Web no existe la necesidad de distribuirlo ni instalarlo.

Solo se necesita de un navegador Web para acceder a la aplicación: solo bastará con tener Internet Explorer o Mozilla Firefox instalado en su PC.

Su mantenimiento se concentra en el servidor: el informático del Policlínico-Hospital Ernesto Che Guevara será su único administrador. Lo que posibilitará que el cliente cuente con las actualizaciones en el instante en que se realizan y que no haya un gran número de personas dedicadas a su mantenimiento.

Permite la navegación desde cualquier punto: lo que facilita el acceso a SIGIGO desde cualquier sitio del Policlínico-Hospital, desde los consultorios médicos de la familia si se restaura la comunicación con éstos incluso desde cualquier sitio de la universidad en caso que se abriera el puerto de comunicación y se decidiera dar acceso a otras personas (pacientes por ejemplo).

1.6 Servidores Web

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 (la versión cifrada y autenticada). Un servidor Web básico cuenta con un esquema de funcionamiento muy simple, basado en ejecutar infinitamente el siguiente bucle:

1. Espera peticiones en el puerto TCP indicado (el estándar por defecto para HTTP es el 80).
2. Recibe una petición.
3. Busca el recurso.
4. Envía el recurso utilizando la misma conexión por la que recibió petición.
5. Vuelve al segundo punto. [11]

1.6.1 Servidor Apache

El Servidor Apache HTTP es un servidor Web de tecnología Open Source sólido y para uso comercial desarrollado por la Apache Software Foundation, además de ser un software libre es robusto, multiplataforma, extensible y popular (fácil conseguir ayuda/suporte). Su implementación se ha realizado y se sigue realizando de forma colaborativa, con prestaciones, características y funcionalidades

equivalentes a las de cualquier servidor comercial. Este servidor Web tiene amplia aceptación en la red: desde 1996, es el servidor HTTP más usado, presenta mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache es un software que está estructurado en módulos. La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Los módulos del Apache se pueden clasificar en tres categorías: Módulos Base, Módulos Multiproceso y Módulos Adicionales.

Las funcionalidades más elementales se encuentran en el módulo base, siendo necesario un módulo multiproceso para manejar las peticiones. Se han diseñado varios módulos multiproceso para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando el rendimiento y rapidez del código. El resto de las funcionalidades del servidor se consiguen por medio de módulos adicionales que se pueden cargar. Para añadir un conjunto de utilidades a este, simplemente hay que añadirle un módulo, de forma que no es necesario volver a instalar el software. [12]

1.6.2 Servidor AOLServer

AOLServer es un proyecto liderado por AOL (América Online) en Source Forge. Actualmente se utiliza como servidor Web de importantes portales con una gran carga de usuarios, desde la propia Web de AOL hasta la de Netscape o Creative Commons. AOLServer es multi-hilo, soporta TCL, es altamente escalable y está particularmente optimizado para sitios Web dinámicos de gran envergadura. [13]

Muchas personas conocen al servidor AOLServer no solo como servidor Web si no también como un servidor de aplicaciones. Este servidor puede mostrar un típico contenido estático, páginas ADP (AOLServer Dynamic Pages) para contenido dinámico, basadas en la instrucción de código TCL en HTML.

1.6.3 Internet Information Server (IIS)

Internet Information Services (IIS) 6.0 es un potente servidor Web que ofrece una infraestructura de gran fiabilidad, capacidad de manejo y escalabilidad para aplicaciones Web sobre todas las versiones de Windows Server 2003. IIS hace posible que las organizaciones aumenten la disponibilidad de sus sitios y aplicaciones Web y a la vez reducir sus costes administrativos. [14]

Los servicios de Internet Information Server son los servicios de software que admiten principalmente la creación, configuración y administración de sitios Web, entre otras funciones de Internet. Incluyen el Protocolo de transferencia de noticias a través de la red (NNTP), el Protocolo de transferencia de archivos (FTP) y el Protocolo simple de transferencia de correo (SMTP). [14].

Es privativo, de la mano de Microsoft.

1.6.4 Fundamentación del servidor web a utilizar

Al referirnos a Internet Information Server nos referimos a un poderoso servidor de páginas web de software privativo; está limitado a entornos Windows, tanto en el Hardware como en Software por lo que necesita licencias como: la del sistema operativo, la de IIS, de SQLserver, para su instalación y utilización se necesita contar con elevados requerimientos de hardware, sobre todo en memoria.

AOLserver es un servidor HTTP de código abierto y fue diseñado conociendo varias deficiencias que existían en el modelo inicial utilizado por Apache, es especialmente adecuado en entornos con una gran carga de usuarios y de transacciones contra base de datos. A pesar de contar con ventajas como éstas y otras que fueron mencionadas con anterioridad no fue seleccionado porque no es muy conocido dentro de la comunidad universitaria (UCI).

Se decidió utilizar como servidor Web el Apache porque es sin dudas, uno de los proyectos de Software Libre más exitosos del momento, por tanto cuenta con muy buena documentación y es bastante popular tanto en la comunidad informática internacional como en la UCI y es muy fácil conseguir soporte y ayuda en caso que sea necesario, es gratuito por lo que no se necesita licencia alguna para su instalación y utilización y multiplataforma, lo que posibilita su utilización en cualquier PC de la universidad en las que se

encuentran instalados sistemas operativos como Linux y Windows incluyendo también varias de sus versiones siendo también compatible con distintas alternativas Hardware: prácticamente todas las arquitecturas actuales.

1.7 Lenguajes de programación

Un lenguaje de programación es una técnica estándar de comunicación que permite expresar las instrucciones que han de ser ejecutadas en una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen un lenguaje informático.

1.7.1 Lenguajes del lado del servidor

Entre los lenguajes del lado del servidor más usados en la actualidad se encuentran: PHP, ASP, ASP.NET, JAVA entre otros. Mediante ellos se implementa la lógica de negocio dentro del servidor y los accesos a los sistemas de gestión de bases de datos.

1.7.1.1 Hypertext Pre-processor (PHP)

En su inicio fue creado por Rasmus Lerdorf en el año 1994; mientras que la implementación principal fue producida por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

PHP es ampliamente usado ya que es un lenguaje interpretado de propósito general y que está diseñado especialmente para desarrollo Web. Generalmente es ejecutado en un servidor Web tomando el código en PHP como su entrada y creando páginas Web como salida. Aun cuando su diseño está orientado a hacer más fácil la creación de páginas Web, permite la creación de aplicaciones con una interfaz gráfica para el usuario utilizando la extensión PHP-Qt o PHP-GTK. También puede ser usado desde la línea de órdenes,

de la misma manera como Perl o Python pueden hacerlo, a esta versión de PHP se la llama PHP CLI (Command Line Interface).

Entre sus ventajas figuran:

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones (desde PHP5).

En cuanto a la seguridad PHP es un poderoso lenguaje e intérprete, ya sea incluido como parte de un servidor Web en forma de módulo, es capaz de acceder a archivos, ejecutar comandos y abrir conexiones de red en el servidor.

1.7.1.2 Active Server Pages (ASP)

Es una tecnología del lado servidor de Microsoft para páginas Web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Server (IIS). La tecnología ASP está

estrechamente relacionada con el modelo tecnológico de su fabricante. Intenta ser solución para un modelo de programación rápida ya que programar en ASP es como programar en Visual Basic.

Es importante conocer que ASP es un sistema con nula portabilidad pues requiere necesariamente de un servidor Windows, con todas las implicaciones de alto costo, poca flexibilidad y escasa seguridad que estos equipos conllevan.

1.7.1.3 Java

Java fue diseñado como lenguaje orientado a objetos, posee una curva de aprendizaje muy rápida, muestra una colección de clases para su uso en aplicaciones de red, que permiten establecer y aceptar conexiones con servidores o clientes remotos, facilitando así la creación de aplicaciones distribuidas, es compilado en la medida en que su código fuente se transforma en una especie de código máquina, fue diseñado para crear software altamente fiable por lo que proporciona numerosas comprobaciones en compilación y en tiempo de ejecución, soporta aplicaciones que serán ejecutadas en los más variados entornos de red, sobre arquitecturas distintas y con sistemas diversos, 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.

El lenguaje Java puede ser usado para crear dos tipos de programas: aplicaciones independientes y applets.

- Las aplicaciones independientes se comportan como cualquier otro programa escrito en cualquier lenguaje
- Por su parte, las applets son pequeños programas que aparecen embebidos en las páginas Web, como aparecen los gráficos o el texto, pero con la capacidad de ejecutar acciones muy complejas, como animar imágenes, establecer conexiones de red, presentar menús y cuadros de diálogo para luego emprender acciones.

1.7.2 Lenguajes del lado del cliente

Entre los lenguajes del lado del cliente más usados en la actualidad se encuentran: HTML, CSS, JavaScript entre otros. Estos lenguajes surgieron por la necesidad de incorporar a las páginas Web elementos dinámicos y comportamientos programados. No sólo para incluir animaciones gráficas, sino también para interactuar con el usuario de la página para, por ejemplo, pedirle e introducir datos, seleccionar opciones o confirmar operaciones.

1.7.2.1 HyperText Markup Language (HTML)

Podemos decir que un científico llamado Tim Berners-Lee inventó HTML (HyperText Markup Language) allá por 1990. Berners-Lee consideraba a HTML una ampliación de "Lenguaje de Marcado Generalizado" (SGML), pero no fue formalmente reconocida como tal hasta la publicación de mediados de 1993, por una organización internacional abierta de normalización nombrada Grupo de Trabajo en Ingeniería de Internet (IETF), en EEUU.

HTML, siglas de HyperText Markup Language (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas Web. Es un lenguaje de hipertexto, es decir, un lenguaje que permite escribir texto de forma estructurada, y que está compuesto por etiquetas, que marcan el inicio y el fin de cada elemento del documento.

Un documento hipertexto no sólo está compuesto por texto si no también puede contener imágenes, sonido, videos, etc., y por lo tanto puede considerarse un documento multimedia. Cuenta con ventajas como:

- HTML es un Standard independiente de fabricantes y marcas, y puede ser visualizado en toda PC que tenga un navegador.
- Es un lenguaje de fácil comprensión, debido al reducido número de etiquetas que usa.
- Permite describir texto presentado de forma estructurada y agradable.

- Las aplicaciones de texto ocupan poco espacio, son rápidas y la mayoría tiene n mucho desarrollo.
- Es el lenguaje usado por los navegadores para mostrar las páginas Web al usuario siendo hoy en día la interface más extendida en la red.
- Permite aglutinar textos, sonidos e imágenes y combinarlos a nuestro gusto.
- Permite la introducción de referencias a otras páginas.
- Está basado en la estructura, no en el aspecto.

1.7.2.2 Cascading Style Sheets (CSS)

Las hojas de estilo en cascada es un lenguaje del lado del cliente usado para definir la presentación de un documento estructurado escrito en HTML o XML. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación. Las CSS mejoran las posibilidades de diseño y presentación de documentos en la red, facilitando además su mantenimiento, ya se trate de un único archivo HTML, o de grandes sitios, con multitud de páginas. Por tanto el modo de funcionamiento de las CSS consiste en definir, mediante una sintaxis especial la forma de presentación que le aplicaremos ya sea a una web entera, un documento html o pagina, una porción del documento o una etiqueta en concreto.

Utilizando hojas de estilo en cascada se puede tener un control centralizado de la presentación de un sitio Web completo con lo que agiliza de forma considerable la actualización del mismo. Además permite que cualquier cambio en la plantilla de estilo se refleje en un cambio inmediato en la apariencia de las páginas relacionadas, sin necesidad de modificar físicamente éstas. Esta tecnología es bastante nueva, por lo que no todos los navegadores la soportan.

CSS nos permite definir aspectos concretos de un documento, facilitando el diseño de los documentos según las necesidades del usuario, reduce el peso de los ficheros disminuyendo el ancho de banda que se consume, otorga a los documentos una mayor usabilidad y accesibilidad ya que permite a los usuarios definir hojas de estilo local, alcanzando así una visualización que se adapte mejor a sus características,

mediante éste se puede indexar los documentos con mayor rapidez, además de lograr la densidad de las palabras clave ya que se reduce la utilización de etiquetas.

1.7.2.3 JavaScript

JavaScript es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.

Al igual que Java, JavaScript es un lenguaje orientado a objetos propiamente dicho, ya que dispone de Herencia, si bien esta se realiza siguiendo el paradigma de programación basada en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.

Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM.

JavaScript proporciona los medios para:

- Controlar las ventanas del navegador y el contenido que muestran
- Programar páginas dinámicas simples de manera eficiente
- Evitar depender del servidor Web para cálculos sencillos
- Capturar los eventos generados por el usuario y responder a ellos sin salir a Internet
- Simular el comportamiento de las macros CGI cuando no es posible usarlas
- Comprobar los datos que el usuario introduce en un formulario antes de enviarlos
- Comunicarse con el usuario mediante diversos métodos

La característica de JavaScript que más simplifica la programación es que, aunque el lenguaje soporta cuatro tipos de datos, no es necesario declarar el tipo de las variables, argumentos de funciones ni valores

de retorno de las funciones. El tipo de las variables cambia implícitamente cuando es necesario, lo que dificulta el desarrollo de programas complejos, pero ayuda a programar con rapidez macros sencillas.

1.7.3 Fundamentación de los lenguajes de programación a utilizar

Después de haberse realizado un estudio previo referente a los lenguajes de programación tanto del lado del servidor como del cliente se arribó a la siguiente conclusión:

Como lenguaje de programación del lado del servidor se decidió utilizar PHP debido a que: está orientado al desarrollo web, es libre y está disponible bajo la licencia GPL, es multiplataforma por lo que no será problema alguno desplegarlo (correrlo) en cualquier PC de la universidad, puede ser ejecutado bajo servidores como IIS, AOLServer y principalmente Apache que fue el elegido como servidor web a utilizar. Su elección se debe también a que se caracteriza además por la simplicidad de su código, por la amplia documentación que posee y por la experiencia que poseen la mayoría de los desarrolladores de web del mundo e incluso en desarrolladores de la UCI, lo que no sería muy difícil encontrar ayuda en caso necesario. En cambio Java es más general y es más adecuado para grandes aplicaciones y se descarta la posibilidad de usar ASP porque a pesar de estar orientado al desarrollo web es privativo.

Como lenguaje de programación del lado del cliente se decidió emplear HTML, CSS y Java Scripts por las características antes mencionadas siendo éstas, de gran utilidad para el desarrollo y la apariencia de SIGIGO.

1.8 Lenguaje de Consulta Estructurado (SQL)

SQL (Structured Query Language): Es un lenguaje de acceso a bases de dato que permite la especificación de diversos tipos de operaciones sobre las mismas. Gracias a base teórica y su orientación al manejo de conjuntos de registros y no precisamente a registros individuales, es considerado un lenguaje de alto nivel permitiendo una alta productividad en el código.

Hoy, en el mundo de la informática existe una diversidad de lenguajes y de bases de datos. Lo que ha traído como consecuencia el surgimiento de estándares que permitan realizar las operaciones básicas de forma universal, de lo contrario la gestión se tornaría realmente complicada.

Structured Query Language es precisamente eso: un lenguaje estándar de comunicación con bases de datos, no queriendo decir que sea el mismo para cada una. Se habla de un normalizado lenguaje que permite trabajar con cualquier tipo de lenguaje (PHP o ASP) así como con cualquier tipo de base de datos (MS Access, SQL Server, MySQL).

Aparte de esta universalidad, el SQL posee otras dos características muy apreciadas. Por una parte, presenta una potencia y versatilidad notable que contrasta, por otra, con su accesibilidad de aprendizaje.

1.8.1 Sistemas Gestores de Bases de Datos (SGBD)

Los Sistemas Gestores de Bases de Datos (SGBD) son un elemento clave dentro del mundo de la Información ya que contienen las rutinas necesarias para el manejo de los datos dígase definición, construcción y manipulación de los datos. Permiten la eliminación y actualización de registros, la combinación con otras bases de datos, generación de informes impresos

Objetivos de los SGBD

- Evitar la redundancia de los datos
- Mejorar los mecanismos de seguridad de los datos y la privacidad
- Asegurar la independencia de los programas y los datos, es decir, la posibilidad de modificar la estructura de la base de datos (esquema) sin necesidad de modificar los programas de las aplicaciones que manejan esos datos
- Mantener la integridad de los datos realizando las validaciones necesarias cuando se realicen modificaciones en la base de datos
- Mejorar la eficacia de acceso a los datos, en especial en el caso de consultas imprevistas

Actualmente existen SGBD líderes del mercado mundial, atendiendo a su clasificación entre software libre y software privativo los más usados son: SQL Server, Oracle, MySQL y PostgreSQL.

1.8.1.1 MySQL

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario, lo que le permite soportar gran cantidad de datos de forma eficiente. Es propiedad y está patrocinado por una empresa privada (GNU), que posee el copyright de la mayor parte del código aunque es software libre.

Existen una variedad de librerías y herramientas que hacen posible su uso mediante gran cantidad de lenguajes de programación. Posee gran rapidez y facilidad de uso. Trayendo consigo una gran aceptación llegando a ser el gestor más usado en el mundo del software libre.

Las principales características de este gestor de bases de datos son las siguientes:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP).
- Gran portabilidad entre sistemas.
- Gestión de usuarios y passwords, manteniendo un buen nivel de seguridad en los datos.
- Buena integración con PHP.

1.8.1.2 PostgreSQL

Es un sistema de gestión de bases de datos objeto-relacional basado en el proyecto POSTGRES, de la universidad de Berkeley.

Ofrece ventajas como:

- Instalación ilimitada
- Mejor soporte que los proveedores comerciales
- Ahorros considerables en costos de operación
- Estabilidad y confiabilidad legendarias
- Extensible
- Multiplataforma
- Diseñado para ambientes de alto volumen
- Herramientas gráficas de diseño y administración de bases de datos

1.8.2 Fundamentación del sistema gestor de base de datos a utilizar

A la hora de elegir el sistema gestor de base de datos se decidió utilizar pese a su mayor lentitud PostgreSQL: debido a que SIGIGO será un software más serio en el que la consistencia de la Base de Datos es un punto fundamental. En cambio MySQL está orientado para sistemas en los que la velocidad y el número de accesos concurrentes sea algo primordial, y la seguridad no sea muy importante.

1.9 Fundamentación de la solución propuesta

A partir de un análisis exhaustivo en este capítulo, como propuesta de solución se ha decidido desarrollar una aplicación Web, utilizando como metodología que guiará el desarrollo del software Extreme Programming (XP), como lenguajes de programación del lado del servidor PHP y del lado del cliente JavaScript, HTML, CSS, como servidor Web el Apache y como sistema gestor de base de datos PostgreSQL.

1.10 Conclusiones

En el presente capítulo se realizó un bosquejo de los sistemas que poseen similitud con el sistema expuesto como propuesta de solución. Se realizó un estudio detallado de las aplicaciones, metodologías, lenguajes de programación, servidores Web y sistemas gestores de bases de datos con el objetivo de seleccionar lo más apropiado para el desarrollo del mismo.

Capítulo 2: Propuesta del Sistema

2.1 Introducción

En el presente capítulo se realiza un minucioso estudio de los patrones arquitectónicos y de diseños más usados hoy por los desarrolladores de software, con el objetivo de seleccionar los más convenientes para el desarrollo del sistema. También se hará una descripción detallada del flujo de procesos presentes en el negocio así como de los posibles procesos a automatizar. Finalizando con la propuesta del sistema a implementar.

2.2 Arquitectura del Software

“La arquitectura de software es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o diferimiento del detalle inherente a la mayor parte de las abstracciones”. [15]

Otra definición de arquitectura se puede encontrar en el documento de la IEEE Std 1471-2000 que plantea:

“La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución”. [15]

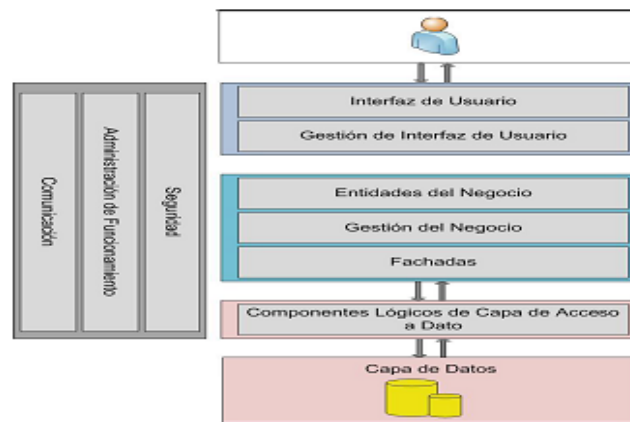
2.2.1 Patrones de Arquitectura

Un patrón de arquitectura de software no es más que un esquema genérico probado para solucionar un problema específico recurrente que surge en un cierto contexto. Este esquema se especifica describiendo los componentes, con sus responsabilidades, relaciones, y las formas en que colaboran. Seguidamente se argumentan algunos de los patrones de arquitectura más utilizados en la actualidad.

2.2.1.1 Arquitectura en capas

En [GS94] Garlan y Shaw definen el estilo en capas como una organización jerárquica tal que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior. [15]

Imagen 2: Arquitectura en Capas

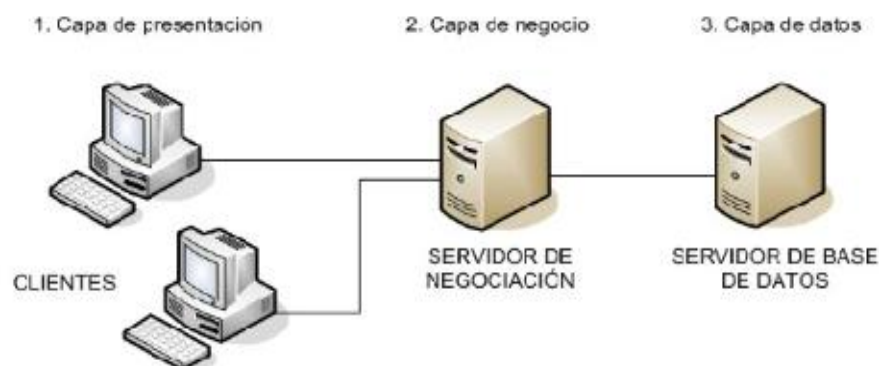


En la actualidad muchos sistemas de información están basados en arquitecturas de dos capas, denominadas generalmente nivel de aplicación y nivel de la base de datos. Este estilo tiene algunos inconvenientes como es la recarga del nivel de aplicaciones y la falta de seguridad, ya que cualquier persona con ciertos conocimientos podría descifrar el código cliente y conocer así todas las dificultades de la aplicación. Por esta razón nace la arquitectura en tres capas.

Las capas que propone este estilo de arquitectura en tres capas son:

1. **Capa de presentación:** Esta capa es la que ve el usuario, también se denomina “Capa de usuario”, presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo proceso realizando un filtrado previo para comprobar que no hay errores de formato. Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener características amigable, entendible y fácil de usar para el usuario, generalmente se presenta como formularios.
2. **Capa de negocio:** Es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se le dice capa de negocio (e incluso de lógica del negocio) pues es aquí donde se establecen todas las reglas que deben dárseles cumplimiento. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con capa de datos para solicitar al gestor de datos para almacenar o recuperar datos de él.
3. **Capa de datos:** Es donde están los datos y es la encargada de acceder a los mismos. Está compuesta por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

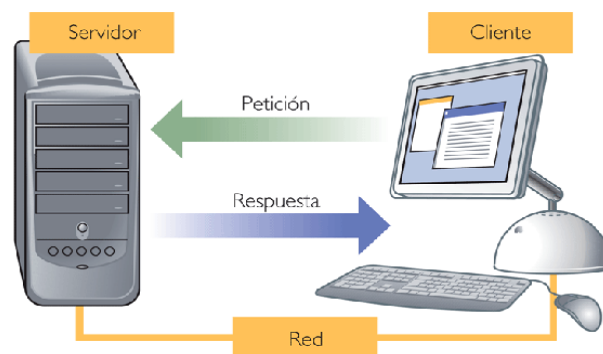
Imagen 3: Arquitectura en tres capas



2.2.1.2 Arquitectura Cliente – Servidor

La arquitectura cliente-servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se llama cliente al proceso que inicia el diálogo o solicita los recursos, y servidor al proceso que responde a las solicitudes del cliente. En este modelo las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario.

Imagen 4: Modelo Cliente-Servidor



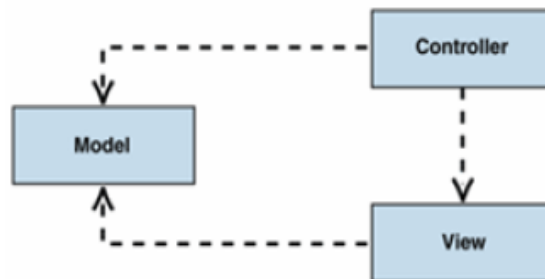
Siempre que un cliente requiere un servicio lo solicita al servidor correspondiente y éste le responde proporcionándole.

2.2.1.3 Patrón Modelo – Vista – Controlador (MVC)

Este modelo fue diseñado para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. Sus características principales son que el Modelo, las Vistas y los Controladores se tratan como entidades separadas; esto hace que cualquier

cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas. En ocasiones este modelo se define más bien como un patrón de diseño como practica recurrente.

Imagen 5: Modelo-Vista-Controlador



Definición por cada una de las partes que componen a un MVC:

El Modelo es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.

La Vista es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al propio Modelo.

El Controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo.

2.3 Patrones de diseño

Los patrones de diseño son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son lenguajes de programación de alto nivel. Estos patrones hacen más fácil reutilizar con éxito los diseños y arquitecturas, ayudan a elegir entre diseños alternativos, hacen a un sistema reutilizable y evitan alternativas que comprometen a la reutilización. Además facilitan el aprendizaje al programador inexperto y ayudan a reutilizar código, facilitando la decisión “herencia o composición”.

Para el diseño de la aplicación se tuvieron en cuenta una serie de patrones, los cuales son de gran utilidad ya que proponen una forma de reutilizar la experiencia de los desarrolladores, para ello clasifican y describen formas de solucionar problemas que ocurren de forma frecuente en el desarrollo.

2.3.1 Patrones GRASP

Los patrones GRASP describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. [16]

Experto: Este patrón es utilizado ya que permite asignar una responsabilidad al experto en información: el módulo que cuenta con la información necesaria para cumplir la responsabilidad. Ofrece facilidad de entender, mantener y manipular. Permite la reutilización.

Para hacer uso del mismo se tuvo en cuenta que la responsabilidad de hacer una labor debe ser de la clase que tiene o puede tener datos involucrados (atributos) a esa labor. Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada.

Creador: Propone que una instancia de un objeto tiene que ser creada por el objeto que tiene la información para ello. Solución que es factible para el diseño del sistema puesto que si se asignan bien el diseño puede soportar un bajo acoplamiento, mayor claridad, encapsulación y reutilización.

Bajo Acoplamiento: Plantea la asignación de responsabilidades de manera que el acoplamiento permanezca bajo, es decir de forma tal que las clases se comuniquen con el menor número de clases que sea posible. Permitiendo tener la mínima conexión entre los módulos de esta aplicación.

Alta cohesión: Cada elemento de nuestro diseño debe hacer una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable, una clase con baja cohesión hace muchas cosas no relacionadas o hace demasiado trabajo.

El uso de este patrón permite asignar a las clases responsabilidades que trabajen sobre una misma área de la aplicación y que no tengan mucha complejidad.

Controlador: El manejo de este patrón facilita establecer la creación de clases que se asocian con operaciones del sistema como respuestas a los eventos del mismo, tales como mensajes, métodos y relaciones entre ellos.

2.4 Procesos presentes en el negocio

En este epígrafe se hace alusión al flujo de procesos, realizando una descripción detallada de cada uno de ellos con el objetivo de lograr ampliar la visión del negocio; para ello se tiene en cuenta el concepto de proceso.

Proceso: Se define como un conjunto de tareas, actividades o acciones interrelacionadas entre sí que, a partir de una o varias entradas de información, materiales o de salidas de otros procesos, dan lugar a una o varias salidas también de materiales (productos) o información con un valor añadido.

2.4.1 Descripción de los procesos presentes en el negocio

Acudir a consulta: El proceso cuenta con tres vías para acudir a las consultas de Ginecología y Regulación de la Fecundidad y solo una para acudir a Obstetricia y a Regulación Menstrual.

- Acudir a las consultas de Ginecología y Regulación de la Fecundidad:

1. Primera vía: Comienza cuando la paciente arriba al consultorio en busca de algún método anticonceptivo, una falta menstrual, amenorrea y alguna patología de cuello o mamas que el médico de la familia no haya logrado solucionar: con los tratamientos indicados con anterioridad. El médico procede a crear y a hacer entrega de la indicación de ultrasonido si es necesario y la remisión a la consulta Ginecología o Regulación de la Fecundidad correspondiente. La paciente recibe dichos documentos y se dirige al laboratorio de ultrasonido en caso de contar con esta remisión (una vez que obtiene los resultados asiste a la consulta de ginecología); en el caso de no contar con alguna indicación asiste directamente a la consulta correspondiente (Ginecología o Regulación de la Fecundidad).
 2. Segunda vía: Inicia cuando la paciente arriba a cualquiera de estas consultas por decisión personal.
 3. Tercera vía: Comienza cuando la paciente una vez estando en la consulta de Ginecología (por cualquiera de las dos vías anteriores) la ginecóloga crea y hace entrega a la misma de la remisión. La paciente arriba a la consulta en la fecha indicada.
- Acudir a las consultas de Obstetricia y Regulación Menstrual:
4. Vía única: Comienza cuando la paciente una vez estando en la consulta de Ginecología (por cualquiera de las dos primeras vías anteriores) la ginecóloga crea y hace entrega a la misma de la remisión a Regulación Menstrual (en caso que vaya a realizarse una interrupción) o a Obstetricia (en caso que se vaya a dar continuidad a un embarazo y la paciente sea profesora o trabajadora interna). La paciente arriba en la fecha indicada.

Acudir a consulta posee relación con el proceso Solicitar Historia Clínica: debido a que se realiza una solicitud cuando la historia de una paciente no se encuentra en consulta, con Enviar Historia Clínica: obteniendo del mismo la Historia Clínica solicitada y con Ingresar información en Hoja de Cargo.

Involucra los documentos Historia Clínica, remisión e indicación y a personas tales paciente, médico del consultorio.

Enviar Historia Clínica: Inicia luego de que una paciente haya sido remitida de su consultorio a la consulta de Ginecología. El médico hace entrega de la historia a la enfermera que allí labora indicándole la consulta a la que debe ser enviada. Esta procede a realizar la entrega.

Enviar Historia Clínica posee relación con el proceso Acudir a consulta: ya que siempre que una paciente sea remitida a alguna consulta, se debe realizar el envío de su historia y con Solicitar Historia Clínica debido a que cuando la historia no se encuentre en la consulta se realizará la solicitud de la misma. Es un posible proceso a automatizar.

Involucra el documento Historia Clínica y a personas tales como al médico y la enfermera del consultorio.

Solicitar Historia Clínica: Comienza una vez que la paciente se encuentra en alguna de las consultas y no está presente su Historia Clínica. La enfermera especialista perteneciente a dicha consulta o la ginecóloga procede a realizar la solicitud mediante:

- Una llamada telefónica al consultorio correspondiente.
- Un correo electrónico a la enfermera del consultorio correspondiente.
- Un mensaje a través de la paciente al consultorio correspondiente.

Solicitar Historia Clínica posee relación con el proceso Acudir a consulta: esperando una solicitud de la historia de la paciente siempre que no se encuentre en consulta y con Enviar Historia Clínica: obteniendo del mismo la Historia Clínica requerida.

Involucra el documento Historia Clínica y a personas tales como la enfermera de la consulta y la enfermera del consultorio.

Crear Hoja de Cargo: Inicia cuando llega a consulta la primera paciente y la ginecóloga o la enfermera se dispone a crear la hoja. Mantiene tiene relación con el proceso Acudir a consulta. Involucra el documento Hoja de Cargo y a personas tales como la ginecóloga y la enfermera de la consulta. Posible proceso a automatizar.

Registrar información en Hoja de Cargo: Inicia cuando la paciente ya se encuentra en consulta. Inmediatamente la ginecóloga da comienzo a un interrogatorio y llena los campos presente en la hoja de cargo (cada consulta posee una con campos propios).

Presenta relación con el Crear Hoja de Cargo. Involucra el documento Hoja de cargo y a personas tales como la paciente y la ginecóloga. Es un posible proceso a automatizar.

Consultar Hoja de Cargo: Comienza cuando la ginecóloga o la enfermera resuelven analizar u obtener información de la misma. Presenta relación con Registrar información en Hoja de Cargo. Involucra a la ginecóloga, la enfermera y la Hoja de Cargo. Es un posible proceso a automatizar.

Modificar Hoja de Cargo: Comienza cuando la ginecóloga comete algún error simple a la hora de crear la hoja o en el registro de la información y se dispone a modificar los datos. Tiene relación con Registrar información en Hoja de Cargo y con Consultar Hoja de Cargo. Involucra a la ginecóloga, la enfermera y la Hoja de Cargo. Es un posible proceso a automatizar.

Eliminar Hoja de Cargo: Comienza cuando la ginecóloga comete algún error garrafal a la hora de crear la hoja o en el registro de la información y se dispone a eliminar la misma. Tiene relación con Registrar información en Hoja de Cargo y con Consultar Hoja de Cargo. Involucra a la ginecóloga, la enfermera y la Hoja de Cargo. Es un posible proceso a automatizar.

Registrar información en Historia Clínica: Primeramente es necesario conocer que las pacientes deben contar con dos Historia Clínicas; una en la consulta de Regulación de la Fecundidad y otra en el consultorio médico al que corresponden.

- Registrar información en la Historia Clínica perteneciente a Regulación de Fecundidad:

Este proceso comienza cuando la paciente llega a dicha consulta. Inmediatamente la enfermera especialista procede a buscar la historia en el libro que le corresponde a la paciente (estudiantes (uno por cada facultad), trabajadoras internas y profesoras).

- Si se encuentra su historia la enfermera procede a realizar el interrogatorio con el objetivo de ir actualizando sus datos.
- Si no se encuentra su historia o la paciente acude por primera vez a la consulta, la enfermera se dispone a crear su historia y para ello realiza un interrogatorio. De esta forma quedan registrados sus datos

➤ Registrar información en la Historia Clínica perteneciente al consultorio médico:

Este proceso inicia una vez que la paciente se encuentra en la consulta, estando presente también su historia. La Ginecóloga procede a consultar la remisión y a comenzar con el interrogatorio para luego ir registrando la información en la historia.

En esta historia clínica del consultorio médico quedan descritas todas las consultas que conforman el departamento de ginecología. Cuando la paciente se encuentra en etapa gestante parte de su información queda recogida en esta historia y el resto en el llamado “Tarjetón de embarazada”.

En el caso que la paciente acuda por primera vez la ginecóloga procede a crear el tarjetón (la paciente porta uno (es el que llena la ginecóloga) y existe otro en su consultorio médico). Mientras que si acude a re-consulta actualiza el ya existente.

Registrar información en Historia Clínica posee relación con el proceso Acudir a consulta: ya que es cuando la paciente asiste a la consulta, cuando en la historia queda recogido todo lo referente esta, con Solicitar Historia Clínica y con Enviar Historia Clínica: obteniendo del mismo la Historia Clínica (refiere historia del consultorio) requerida. Es un posible proceso a automatizar.

Involucra los documentos Historia Clínica, remisión, indicación con los resultados y a personas tales como la Ginecóloga.

Consultar Historia Clínica: Comienza cuando la ginecóloga o la enfermera resuelven examinar o adquirir información de la misma. Presenta relación con Registrar información en Historia Clínica. Involucra a la ginecóloga, la enfermera y la Historia Clínica. Es un posible proceso a automatizar.

Modificar Historia Clínica: Comienza cuando la ginecóloga comete errores sencillo en el instante de crear la historia o en el registro de su información y se dispone a cambiar los datos. Tiene relación con Registrar información en Historia Clínica y con Consultar Historia Clínica. Involucra a la ginecóloga, la enfermera y la Hoja de Cargo. Es un posible proceso a automatizar.

Eliminar Historia Clínica: Comienza cuando la ginecóloga comete algún error grave en el momento de crear la hoja o en el registro de la información y se dispone a eliminar la misma. Tiene relación con Registrar información en Historia Clínica y con Consultar Historia Clínica. Involucra a la ginecóloga, la enfermera y la Hoja de Cargo. Es un posible proceso a automatizar.

Gestionar paciente en la consulta Ginecología: A la consulta de Ginecología asisten las pacientes con trastornos en su ciclo menstrual (retraso por ejemplo de 2 meses, sangramiento en los genitales), amenorreas, alguna patología de cuello o mamas, siempre y cuando no son controladas con el tratamiento indicado por el médico del consultorio.

El proceso da inicio en el instante en que la paciente arriba a la consulta, y ésta es sometida a un interrogatorio (se adicionan sus datos en la Hoja de Cargo), a exámenes físicos, se le indican análisis complementarios, ultrasonidos, radiologías, citologías (programa atendido por la enfermera de la consulta) o se le realiza el manejo sintomático complementario (en el caso que los laboratorios no estén funcionando).

Cuando se diagnostica alguna afección se le establece el tratamiento a seguir, además se programa la re-consulta para dar continuidad o se puede incluso remitir a la paciente a consulta con las que el hospital de la universidad no cuenta dígame patología de cuello o de mamas.

En el caso que a la paciente se le diagnostique un embarazo se pasa a la llamada consulta de Clasificación (es una subconsulta dentro de la de ginecología) para arribar a la conclusión de: continuar o interrumpir el embarazo.

- Cuando se determina continuarlo y la paciente es:

Estudiante: se le informa al médico de la familia y al decano de su facultad mediante una carta (esta entregado por la paciente). Inmediatamente se debe procesar una licencia y la paciente debe ir al área de salud que le pertenece en su provincia. Luego de un año fuera, la estudiante si lo desea puede regresar a la universidad.

Profesora: se informa al médico de la familia el cual realizará la captación, la remisión a las consultas necesarias. La doctora hace entrega la remisión a Obstetricia.

- Cuando se determina realizar la interrupción y la paciente está en tiempo de hacerse la regulación menstrual:

La doctora ginecóloga la remite a la consulta de Regulación Menstrual e indica los siguientes complementarios (Hemoglobina, Serología, VIH), un exudado y un ultrasonido.

- Cuando se determina realizar la interrupción y la paciente cuenta con más tiempo de lo que se pronostica para hacerse la regulación menstrual:

La paciente no puede realizarse una interrupción mediante una Regulación, por lo que debe tramitar un Legrado u otros métodos que no se hacen en la universidad. Cuando se trata de una profesora la misma busca su propia vía de solución. Si es una estudiante, se envía una carta a su decano y el mismo es responsable de hacérselo saber a sus padres o a sus tutores. Debido a que los médicos del hospital ni las direcciones de las facultades se hacen responsable de las consecuencias que esto pueda traer.

Gestionar paciente en la consulta Ginecología posee relación con el proceso Acudir a consulta, con Solicitar Historia y Enviar Historia: en caso que la historia no se encuentre en consulta, con Registrar información en historia: ya que siempre todo lo relacionado con la consulta quedara plasmado la historia y con Registrar información en Hoja de Cargo.

Involucra los documentos Historia Clínica, Hoja de Cargo, carta a los decanos, remisiones e indicaciones y a personas tales como: paciente, ginecóloga y la enfermera de la consulta.

Gestionar paciente en la consulta Obstetricia: Inicia una vez que llega la paciente a la consulta trayendo consigo la remisión (del médico del consultorio o de la ginecóloga), el resultado del ultrasonido y la

Historia clínica de su consultorio. La doctora escribe los datos correspondientes en la Hoja de Cargo y consulta el resultado de los complementarios en el Buzón de Análisis Complementarios.

La paciente se somete al examen físico (siempre que sea necesario) y es remitida a las consultas de nutrición, psicología, estomatología y genética en caso que no haya asistido. Además se le realizan las indicaciones de los ultrasonidos y exámenes complementarios y físicos en dependencia del trimestre en que se encuentre. Mientras la paciente no presente problemas podrá permanecer en la escuela hasta las 37 semanas de gestación. Una vez cumplidas se procede a tramitar su licencia de maternidad.

Gestionar paciente en la consulta Obstetricia posee relación con el proceso Acudir a consulta, con Solicitar Historia y Enviar Historia: en caso que la historia no se encuentre en consulta, con Registrar información en historia: ya que siempre todo lo relacionado con la consulta quedará plasmado en esta y con Registrar información en Hoja de Cargo.

Involucra los documentos Historia Clínica, Hoja de Cargo, Tarjetón de Embarazada, remisiones e indicaciones y a personas tales como: paciente, ginecóloga y la enfermera de la consulta.

Gestionar paciente en la consulta Regulación Menstrual: Inicia cuando la paciente llega a la consulta portando el resultado del ultrasonido. La doctora registra sus datos en la Hoja de Cargo y consulta el resultado de los complementarios. La paciente es sometida al examen físico (se realiza antes de hacerse la interrupción). Mientras resulte:

- Positivo: se realiza la interrupción y una vez concluida se remite al consultorio para darle el seguimiento correspondiente. Además de indicarle un ultrasonido y de establecer un tratamiento.
- Negativo: no se puede realizar la interrupción pasando por todo el proceso de gestión del paciente en la consulta de genealogía para cuando la paciente no puede regularse.

Gestionar paciente en la consulta Regulación Menstrual posee relación con el proceso Acudir a consulta: debido a que es lo primero que la paciente realiza, con Solicitar Historia y Enviar Historia: en caso que la historia no se encuentre en consulta, con Registrar información en historia y con Registrar información en Hoja de Cargo.

Involucra los documentos Historia Clínica, Hoja de Cargo, remisiones e indicaciones y a personas tales como: paciente, ginecóloga y la enfermera de la consulta.

Gestionar paciente en la consulta Regulación de la Fecundidad: Comienza cuando la paciente se presenta en la consulta por algún método anticonceptivo. La enfermera procede registrar sus datos en la Hoja de Cargo y a pesarla y medirla.

Cuando la paciente decide o de cambiar el anticonceptivo debido a que ha hecho alguna reacción, es la encargada de explicarle todo lo relacionado al respecto. El proceso finaliza una vez que han quedado actualizados los datos en la historia y con la entrega de las tabletas, la aplicación de una vacuna, con el implante de algún intrauterino o con una remisión a la especialista, siempre y cuando se presente algún problema que la enfermera no pueda resolver.

Gestionar paciente en la consulta Regulación de la Fecundidad posee relación con el proceso Acudir a consulta: debido a que es lo primero que la paciente realiza, con Registrar información en historia y con Registrar información en Hoja de Cargo.

Involucra los documentos Historia Clínica, Hoja de Cargo, remisiones y a personas tales como: paciente y la enfermera de la consulta.

Crear remisión e indicación: Este proceso comienza cuando se procede a crearle a la paciente una o varias remisiones o indicaciones, en dependencia de su necesidad. Puede ser iniciado por la ginecóloga o el médico del consultorio. Presenta relación con Acudir a Consulta, Gestionar Paciente (en cualquiera de las consultas). Involucra a la paciente, el médico del consultorio, la ginecóloga y los documentos remisión e indicación. Es un posible proceso a automatizar.

Consultar remisión y resultado de indicación: Da comienzo cuando la ginecóloga de detiene a analizar la remisión enviada, el resultado de algún ultrasonido o radiología (portada por la paciente) o el de análisis complementarios (este resultado es enviado del laboratorio correspondiente). Para consultar este último resultado la enfermera realiza una búsqueda en el buzón de Análisis Complementarios.

En el buzón estos resultados se encuentran archivados por el apellido en orden alfabético. En el caso de no encontrarse se verifica con el personal del laboratorio (por llamada telefónica, mensaje personal, por correo, mediante la enfermera, la ginecóloga o la propia paciente). Una vez realizada la verificación, si los

resultados se encuentran en dicho laboratorio, se procede a realizar el envío o la búsqueda, sino se realiza nuevamente la indicación.

El proceso presenta relación con Gestionar paciente (en cualquier consulta) y con Crear remisión e indicación. Involucra a la paciente, la ginecóloga, la enfermera y los documentos remisión e indicación. Es un posible proceso a automatizar.

Eliminar o modificar remisión e indicación: Inicia cuando la ginecóloga comete un error/es en alguna remisión o indicación procede a modificarla o eliminarla en dependencia del error cometido. Presenta relación con Consultar remisión e indicación. Involucra la ginecóloga y los documentos remisión e indicación. Es un posible proceso a automatizar.

Enviar Hoja de cargo: Se da comienzo al proceso cuando la ginecóloga concluye cualquiera de las consultas y procede a llevar la hoja de cargo al departamento de estadística (puede ser llevada por la enfermera), haciendo entrega de la misma.

Enviar Hoja de cargo: posee relación con el proceso Registrar información en Hoja de cargo, Eliminar información en Hoja de cargo, Modificar información en Hoja de cargo. Es un posible proceso a automatizar. Involucra el documento Hoja de Cargo y a personas tales como la Ginecóloga y la enfermera.

Controlar reportes: Empieza cuando la ginecóloga culmina una consulta y se dispone a consultar la hoja de cargo para sacar los reportes que le son necesarios (total de regulaciones menstruales, números de pacientes controlados en planificación familiar, métodos anticonceptivos más usadas, principales motivos de la consulta ginecológica y número de embarazadas).

Controlar reportes: posee relación con el proceso Ingresar información en Hoja de Cargo. Involucra el documento Hoja de Cargo y a personas tales como la Ginecóloga. Es posible proceso a automatizar.

2.5 Descripción de los procesos que serán objeto de automatización en el sistema

Persiguiendo la idea de optimizar el trabajo de las especialistas que laboran en el área ginecológica y partiendo de la descripción del flujo de procesos presentes en el negocio, se arribó a la siguiente conclusión: existen varios procesos que deben ser automatizados, debido a que su ejecución de forma manual resulta tediosa, y está propensa a errores y réplicas innecesarias, además de consumir mayor cantidad de tiempo.

Se automatizarán los procesos siguientes:

Autenticación: Este proceso tiene un carácter obligatorio, puesto que partiendo de él, cada usuario tendrá el acceso necesario según su rol. Primeramente se deberá teclear al nombre de usuario y la contraseña en la interfaz correspondiente, luego el sistema verifica los datos introducidos y denegará o dará el permiso de entrada y de navegación. Siempre que exista un error en la entrada de los datos el sistema lanzará una excepción (mensaje de error). Cuando el usuario no se encuentre en la base de dato, de igual forma lanzará un mensaje de aviso (El usuario no existe) y solo el administrador del sistema tendrá la posibilidad de agregarlo. Se lanzará otro mensaje cuando el administrador introduzca un usuario que ya existe (El usuario existe).

Gestionar Historia Clínica: Para tener acceso a las opciones que este proceso brinda el usuario tiene que pasar primeramente por el proceso de autenticación. Una vez dentro de la aplicación tiene el acceso que necesita. Este proceso cuenta con opciones como crear, eliminar, consultar (buscar) y actualizar la historia clínica de una determinada paciente. Cuando un usuario sin los permisos intenta acceder o un usuario crea una historia con los datos de una paciente y esta existe o cuando se haga la búsqueda de la historia clínica de una paciente que no existe, el sistema lanzará errores como: Acceso denegado, La historia de la paciente existe, La historia de la paciente no existe.

Gestionar Hoja de Cargo: Igualmente se debe pasar por el proceso de autenticación antes de poder acceder a las opciones que brinda el presente proceso. Según el rol del usuario tendrá la oportunidad de crear, eliminar, actualizar y consultar una hoja de cargo determinada, de una consulta dada. Cuando un

usuario sin los permisos intenta acceder o se realiza la búsqueda de una hoja que no exista el sistema lanzará mensajes de error como: Acceso denegado, No existe la hoja de cargo.

Controlar Reportes: De la misma forma el usuario tiene que autenticarse para lograr el acceso y poder obtener (crear), eliminar y consultar un reporte. Estos se obtienen a través de cálculos que se hacen sobre las hojas de cargo. Si intenta acceder un usuario que no posee el permiso o existe un reporte que se está intentando crear o se desea consultar uno no existente; el sistema arrojará excepciones como: Acceso denegado, El reporte se encuentra creado, El reporte no existe.

Gestionar remisiones: El usuario debe pasar por el proceso de autenticación inicialmente. El proceso permite consultar dichos buzones y mostrar la remisión que se desea. Además ofrece la posibilidad de crear y eliminar.

2.6 Propuesta del sistema

Se propone para resolver los problemas de gestión de la información en el área de ginecología del Policlínico-Hospital de La Universidad de Ciencias Informáticas el desarrollo de una aplicación web basada en la arquitectura: tres capas. Esta elección se debe a las ventajas que proporciona este tipo de aplicación y de arquitectura expuestas con anterioridad, ventajas que son sumamente útiles por las constantes peticiones de cambios provenientes del cliente.

El software brindará a los usuarios los servicios siguientes:

Gestión del perfil de usuario: con el objetivo de prever la autenticación.

Autenticación: para proporcionarles acceso a los usuarios, en dependencia del rol que desempeñan en el sistema.

Gestión de Historias Clínicas proporcionándole al usuario un total control de la información archivada en las historias clínicas de las pacientes así como su total gestión.

Gestión de Hojas de Cargo permitiendo al usuario la gestión y control de las 4 hojas de cargo correspondiente a las 4 consultas que pertenecen al área ginecológica.

Reporte el cual obteniendo la información del módulo Hoja de Cargo permite realizar cálculos y conformar reportes que son de interés para los usuarios del sistema.

Gestión de paciente permitiéndole al usuario la gestión de la información de los mismos, y apoyando el proceso de gestión de historias clínicas.

A estos servicios los usuarios podrán acceder siempre y cuando se hayan autenticado con anterioridad implicando, lógicamente que haya sido creado primeramente su perfil de usuario. En dependencia del rol que desempeñen en el sistema obtendrán acceso a los demás servicios.

Con la finalidad de contar con un producto atractivo, usable, rápido y confiable el sistema contará al mismo tiempo con las siguientes cualidades o propiedades:

- ❖ Utilizar como metodología de desarrollo de software Extreme Programming.
- ❖ Aplicación Web escrita sobre el lenguaje de programación PHP 5, CCS, HTML y Java Script.
- ❖ Utilizar como servidor web Apache 2.2 y como gestor de Base de Datos Postgres SQL 8.3.
- ❖ Podrá ser desplegado sobre cualquier sistema operativo (multiplataforma).
- ❖ Presentará un diseño orientado a llamar la atención del usuario con navegación sencilla y escalable.
- ❖ No tendrá animaciones ni imágenes pesadas que obstaculicen la rapidez de las transiciones del ancho de banda con un diseño perfectamente encuadrado para resoluciones de 1024 x 768.
- ❖ Podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente web en sentido general.
- ❖ La eficiencia de esta aplicación será óptima en cuanto a la velocidad de procesamiento, disponibilidad, tiempo de respuesta y aprovechamiento de los recursos.

- ❖ Garantizará que la información sea publicada únicamente por quien tiene derecho a publicarla. Para esto el usuario antes de que pueda realizar cualquier acción sobre la aplicación deberá autenticarse, buscando eliminar el acceso no autorizado.

2.6.1 Personal relacionado con el sistema

Una persona relacionada al sistema es todo aquel individuo que obtiene al menos un resultado del valor de uno o varios procesos que se ejecutan en el mismo, así como los que no obtienen ningún valor pero se encuentran involucradas en dichos procesos, debido a que participan en ellos.

Tabla 1: Personas relacionadas con el sistema

Personas relacionadas con el sistema	Descripción
Administrador	Individuo facultado para la gestión del sistema. Tendrá solamente acceso al módulo "Administración". Además de ser el responsable de darle soporte al software cada vez que el mismo lo requiera.
Ginecóloga	Persona que tendrá acceso al control y gestión toda la información que el sistema contiene, exceptuado el módulo administración.
Enfermeras del departamento	Persona que tendrá acceso al control y gestión toda la información que el sistema contiene, exceptuado el módulo administración.
Especialista de estadísticas	Persona que tendrá solamente acceso al módulo "Reporte".

2.7 Conclusiones

En el capítulo que recién concluyó se realizó un estudio de los patrones arquitectónicos más utilizados hoy en lo que a arquitectura de software respecta, arribando a la conclusión de que la aplicación se desarrollará basada en una arquitectura en 3 capas. También se realizó la descripción del flujo de procesos presentes en el negocio, de los procesos que serán objetos de automatización y de las personas que se relacionarán con el sistema.

Capítulo 3: Exploración y Planificación

“(…) la dirección con éxito del negocio o la ejecución de la misión pública depende del *software* que lo soporta. Estos sistemas *software* se hacen más complejos, su tiempo de salida al mercado necesita reducirse, y su desarrollo, por tanto, se hace más difícil. Por razones como éstas, la industria del *software* necesita un proceso para guiar a los desarrolladores (…)” (JACOBSON 2000) [17]

3.1 Introducción

En el presente capítulo se hace alusión a las fases de exploración y planificación de la metodología de desarrollo propuesta para la implantación del sistema. Para ello se exponen los artefactos generados durante el desarrollo de las mismas.

3.2 Fase I: Exploración

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología. [18]

3.2.1 Historias de Usuario

Las historias de usuarios representan una breve descripción del comportamiento del sistema ya que son la técnica utilizada en XP para especificar los requisitos del software. Son redactadas desde la perspectiva

del cliente aunque los desarrolladores pueden brindar también su ayuda en la identificación de las mismas. Se realiza una historia por cada característica principal del sistema. Su uso reemplaza un gran documento de requisitos aún cuando su contenido debe ser concreto y sencillo. Presiden la creación de las pruebas de aceptación y su implementación debe contar con un periodo ideal de 1 a 3 semanas aproximadamente.

Tabla 2: Representación de una historia de usuario

Historia de Usuario	
Número: <i>(Número de la historia de usuario incremental en el tiempo)</i>	Usuario: <i>(sistema que protagoniza la historia)</i>
Nombre Historia de Usuario: <i>(El nombre de la historia de usuario sería para identificarlas mejor entre los desarrolladores y el cliente)</i>	
Prioridad en Negocio: <i>(La importancia que tiene para el cliente). (Alta / Media / Baja)</i>	Riesgo en Desarrollo: <i>(Qué tan difícil es para el desarrollador). (Alto / Medio / Bajo)</i>
Puntos asignados: <i>(El tiempo que se demorará al desarrollo de la HU)</i>	Iteración Asignada: <i>(La iteración (liberación en nuestro proceso) a la que corresponde)</i>
Programador responsable: <i>(Persona responsable de la programación de la funcionalidad que contiene dicha historia de usuario)</i>	
Descripción: <i>(Se especificará las operaciones del usuario y las respuestas que dará el sistema)</i>	
Observaciones: <i>(Algunas observaciones de interés, como glosario, información sobre usuarios, etc.)</i>	

En esta fase de exploración se llegaron a identificar 18 historias de usuarios las cuales se describen a continuación:

Tabla 3: Representación de la Historia de Usuario Nro. 1

Historia de Usuario	
Número : 1	Usuario : Administrador

Nombre historia: Crear las cuentas de los usuarios del sistema.	
Prioridad en el negocio : Alta	Riesgo en el desarrollo : Baja
Puntos estimados : 0.4	Iteración asignada : 1
Programador responsable : Yanisvel Suárez – Yamisleydis Borrero	
Descripción: El usuario tendrá la posibilidad de crear una cuenta en el sistema.	
Observaciones: El usuario deberá autenticarse con anterioridad.	

Tabla 4: Representación de la Historia de Usuario Nro. 2

Historia de Usuario	
Número : 2	Usuario : Administrador
Nombre historia: Eliminar perfil de usuario	
Prioridad en el negocio : Baja	Riesgo en el desarrollo : Baja
Puntos estimados : 0.2	Iteración asignada : 3
Programador responsable : Yanisvel Suárez – Yamisleydis Borrero	
Descripción: El usuario tendrá la posibilidad de eliminar cualquier perfil del sistema en caso que sea necesario.	
Observaciones: El usuario deberá autenticarse con anterioridad.	

Tabla 5: Representación de la Historia de Usuario Nro. 3

Historia de Usuario	
Número : 3	Usuario : Todos
Nombre historia: Ver perfil de un usuario	
Prioridad en el negocio : Media	Riesgo en el desarrollo : Baja

Puntos estimados : 0.6	Iteración asignada : 2
Programador responsable : Yanisvel Suárez – Yamisleydis Borrero	
Descripción: Cada usuario del sistema podrá visualizar su perfil.	
Observaciones: Para visualizar un perfil el usuario deberá tener la cuenta creada.	

Tabla 6: Representación de la Historia de Usuario Nro. 4

Historia de Usuario	
Número : 4	Usuario : Todos
Nombre historia: Autenticar usuario.	
Prioridad en el negocio : Alta	Riesgo en el desarrollo : Media
Puntos estimados : 0.6	Iteración asignada : 1
Programador responsable : Yanisvel Suárez – Yamisleydis Borrero	
Descripción: Para tener acceso al sistema los usuarios primeramente tendrán que autenticarse. Se verificará en la aplicación si los datos entrados fueron correctos. En caso contrario el sistema lanzará un mensaje de error.	
Observaciones: En la autenticación se entraran datos como: nombre de usuario y contraseña.	

Tabla 7: Representación de la Historia de Usuario Nro. 5

Historia de Usuario	
Número : 5	Usuario : Ginecóloga
Nombre historia: Introducir datos de un paciente	
Prioridad en el negocio : Alta	Riesgo en el desarrollo : Media
Puntos estimados : 0.4	Iteración asignada : 1
Programador responsable : Yanisvel Suárez – Yamisleydis Borrero	

Descripción: Este usuario podrá ingresar en el sistema los datos personales de un paciente.
Observaciones: Los datos a entrar serán solo datos personales del paciente.

Tabla 8: Representación de la Historia de Usuario Nro. 6

Historia de Usuario	
Número : 6	Usuario : Ginecóloga
Nombre historia: Modificar datos de un paciente.	
Prioridad en el negocio : Alta	Riesgo en el desarrollo : Baja
Puntos estimados : 0.2	Iteración asignada : 2
Programador responsable : Yanisvel Suárez – Yamisleydis Borrero	
Descripción: El usuario podrá modificar en el sistema los datos de un paciente específico.	
Observaciones: Los datos del paciente antes que puedan ser modificados, deben estar registrados en la aplicación.	

Tabla 9: Representación de la Historia de Usuario Nro. 7

Historia de Usuario	
Número : 7	Usuario : Ginecóloga
Nombre historia: Eliminar datos de un paciente.	
Prioridad en el negocio : Baja	Riesgo en el desarrollo : Baja
Puntos estimados : 0.2	Iteración asignada : 3
Programador responsable : Yanisvel Suárez – Yamisleydis Borrero	
Descripción: El usuario podrá eliminar del sistema los datos de un paciente en particular cuando sea necesario.	
Observaciones: Los datos del paciente deben estar registrados en la aplicación.	

Tabla 10: Representación de la Historia de Usuario Nro. 8

Historia de Usuario	
Número : 8	Usuario : Ginecóloga
Nombre historia: Buscar datos de un paciente.	
Prioridad en el negocio : Media	Riesgo en el desarrollo : Media
Puntos estimados : 0.4	Iteración asignada : 2
Programador responsable : Yanisvel Suárez – Yamisleydis Borrero	
Descripción: El usuario tendrá la posibilidad de buscar los datos personales de un paciente.	
Observaciones: Se debe introducir el carné de identidad del paciente que se desee buscar.	

Tabla 11: Representación de la Historia de Usuario Nro. 9

Historia de Usuario	
Número : 9	Usuario : Ginecóloga
Nombre historia: Generar reporte estadístico.	
Prioridad en el negocio : Alta	Riesgo en el desarrollo : Media
Puntos estimados : 0.6	Iteración asignada : 2
Programador responsable : Yanisvel Suárez – Yamisleydis Borrero	
Descripción: El usuario podrá generar reportes estadísticos mediante los datos almacenados en las hojas de cargo.	
Observaciones: Los reportes estadísticos se podrán generar por semanas.	

Tabla 12: Representación de la Historia de Usuario Nro. 10

Historia de Usuario	
Número : 10	Usuario : Ginecóloga
Nombre historia: Crear hoja de Cargo	
Prioridad en el negocio : Alta	Riesgo en el desarrollo : Media
Puntos estimados : 0.4	Iteración asignada : 1
Programador responsable : Yanisvel Suárez – Yamisleydis Borrero	
Descripción: El usuario podrá crear hojas de cargo, para ello contará con 4 formatos diferentes que aún cuando tienen campos comunes también poseen sus diferencias.	
Observaciones: Existen cuatro hojas de cargo una por cada consulta, las cuales tendrán datos diferentes y comunes de los pacientes. Las mismas llevan el mismo nombre que las consultas que componen al departamento de Ginecología: hoja de cargo de Regulación de la Fecundidad, Ginecología del Adolescente, Regulación y la perteneciente Obstetricia.	

Tabla 13: Representación de la Historia de Usuario Nro. 11

Historia de Usuario	
Número : 11	Usuario : Ginecóloga
Nombre historia: Modificar Hoja de Cargo.	
Prioridad en el negocio : Media	Riesgo en el desarrollo : Medio
Puntos estimados : 0.4	Iteración asignada : 3
Programador responsable : Yanisvel Suárez – Yamisleydis Borrero	
Descripción: El usuario podrá modificar datos en las hojas de cargo en caso que le sea necesario.	
Observaciones: Debe estar la hoja de cargo insertada en el sistema.	

Tabla 14: Representación de la Historia de Usuario Nro. 12

Historia de Usuario	
Número : 12	Usuario : Ginecóloga
Nombre historia : Buscar Hoja de Cargo	
Prioridad en el negocio : Alta	Riesgo en el desarrollo : Bajo
Puntos estimados : 0.2	Iteración asignada : 2
Programador responsable : Yanisvel Suárez – Yamisleydis Borrero	
Descripción: El usuario tendrá la posibilidad de realizar la búsqueda de cualquier hoja de cargo. En caso de que no se encuentre el sistema le mostrará un mensaje informativo relacionado con el resultado de la misma.	
Observaciones: Para realizar la búsqueda el usuario deberá entrar la fecha en que fue creada la hoja de cargo y a el nombre de la consulta a la que pertenece.	

Tabla 15: Representación de la Historia de Usuario Nro. 13

Historia de Usuario	
Número : 13	Usuario : Ginecóloga
Nombre historia: Crear remisión.	
Prioridad en el negocio : Alta	Riesgo en el desarrollo : Media
Puntos estimados : 0.4	Iteración asignada : 1
Programador responsable : Yanisvel Suárez – Yamisleydis Borrero	
Descripción: El usuario tendrá la posibilidad de elaborar una remisión a otra consulta.	
Observaciones: La paciente debe estar ingresada a la base de datos	

Tabla 16: Representación de la Historia de Usuario Nro. 14

Historia de Usuario	
Número : 14	Usuario: Ginecóloga
Nombre historia: Buscar remisión.	
Prioridad en el negocio : Alta	Riesgo en el desarrollo : Baja
Puntos estimados : 0.2	Iteración asignada : 2
Programador responsable : Yanisvel Suárez – Yamisleydis Borrero	
Descripción: El usuario podrá buscar una remisión, en caso de no encontrarse el sistema procederá a mostrar un mensaje informativo relacionado con la búsqueda.	
Observaciones: Se debe entrar la fecha en que se hizo la remisión y el nombre de la interconsulta a la que se hizo la remisión.	

Tabla 17: Representación de la Historia de Usuario Nro. 15

Historia de Usuario	
Número : 15	Usuario : Ginecóloga
Nombre historia: Crear historia clínica.	
Prioridad en el negocio : Alta	Riesgo en el desarrollo : Media
Puntos estimados : 0.6	Iteración asignada : 1
Programador responsable : Yanisvel Suárez – Yamisleydis Borrero	
Descripción: El usuario podrá crear las historias clínicas necesarias.	
Observaciones: Existen dos tipos de historias clínicas, historia clínica de Regulación de la Fecundidad e historia clínica del consultorio o ginecológica.	

Tabla 18: Representación de la Historia de Usuario Nro. 16

Historia de Usuario	
Número : 16	Usuario : Ginecóloga
Nombre historia: Actualizar historia clínica.	
Prioridad en el negocio : Alta	Riesgo en el desarrollo : Media
Puntos estimados : 0.4	Iteración asignada : 2
Programador responsable : Yanisvel Suárez – Yamisleydis Borrero	
Descripción: El usuario podrá modificar algún campo de la historia clínica en caso que sea necesario. De este modo quedará actualizada la historia.	
Observaciones: Para realizar cambios en una historia el usuario deberá realizar primeramente su búsqueda.	

Tabla 19: Representación de la Historia de Usuario Nro. 17

Historia de Usuario	
Número : 17	Usuario : Ginecóloga
Nombre historia: Eliminar historia clínica.	
Prioridad en el negocio : Baja	Riesgo en el desarrollo : Baja
Puntos estimados : 0.2	Iteración asignada : 3
Programador responsable : Yanisvel Suárez – Yamisleydis Borrero	
Descripción: El usuario podrá eliminar una historia clínica del sistema cuando sea necesario.	
Observaciones: Para eliminar en una historia clínica el usuario deberá realizar primeramente su búsqueda.	

Tabla 20: Representación de la Historia de Usuario Nro. 18

Historia de Usuario	
Número : 18	Usuario : Ginecóloga
Nombre historia: Buscar historia clínica.	
Prioridad en el negocio : Alta	Riesgo en el desarrollo : Baja
Puntos estimados : 0.2	Iteración asignada : 2
Programador responsable : Yanisvel Suárez – Yamisleydis Borrero	
Descripción: El usuario podrá realizar la búsqueda de una historia clínica de un paciente, al mismo tiempo esta se visualizará. En caso de no encontrarse la historia el sistema mostrará un mensaje informativo relacionado con los resultados de la búsqueda.	
Observaciones: El usuario debe ingresar el número del carnet de identidad del paciente a que pertenece dicha historia.	

3.3 Fase II: Planificación de la Entrega

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días. Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos. Por otra parte, el equipo de desarrollo mantiene un registro de la “velocidad” de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración. [18]

La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o

cuánto tiempo tomará implementar un conjunto de historias. Al planificar por tiempo, se multiplica el número de iteraciones por la velocidad del proyecto, determinándose cuántos puntos se pueden completar. Al planificar según alcance del sistema, se divide la suma de puntos de las historias de usuario seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación. [18]

3.3.1 Estimación de esfuerzos por historias de usuario

Con el objetivo de alcanzar un óptimo desarrollo del sistema, se realizó una estimación para cada una de las historias de usuario identificadas, obteniendo los resultados que se muestran a continuación:

Tabla 21: Estimación de esfuerzos por Historia de Usuario

Historia de Usuario	Puntos de Estimación (semanas)
Crear perfil de usuario.	0.4
Eliminar perfil de usuario	0.2
Ver perfil de un usuario.	0.6
Autenticar usuario	0.6
Introducir datos de un paciente	0.4
Modificar los datos de un paciente	0.2
Eliminar los datos de un paciente.	0.2
Buscar datos personales de un paciente	0.4
Generar reporte estadístico	0.6
Crear Hoja de Cargo	0.4
Modificar Hoja de Cargo	0.4

Buscar Hoja de cargo	0.2
Crear remisión	0.4
Buscar remisión	0.2
Crear historia clínica	0.6
Actualizar historia clínica	0.4
Eliminar historia clínica	0.2
Buscar historia clínica	0.2

3.3.2 Plan de iteraciones

Una vez que fueron identificadas y detalladas las historias de usuario y estimado el esfuerzo propuesto para la realización de cada una de éstas, se procede a la planificación de la etapa de implementación del sistema. Este plan define las historias de usuario que serán implementadas en cada iteración y las posibles fechas de sus liberaciones.

Se decidió desarrollar el sistema en tres iteraciones, las cuales se especifican a continuación:

Iteración 1:

Esta iteración tiene como objetivo la implementación de las funcionalidades de mayor prioridad, es decir de las historias de usuario más críticas. Al finalizar se contará con las funcionalidades descritas en las historias de usuario 1, 4, 5, 10, 13 y 15 las cuales hacen alusión a la creación de una cuenta de usuario y a autenticación del mismo como parte de la seguridad de la aplicación, a la introducción de los datos de un paciente, la creación de la hoja de cargo, de la remisión y la historia clínica.

Iteración 2:

Iteración centrada en continuar la implementación de las funcionalidades de alta prioridad así como las de prioridad media. Con la culminación de la misma se tendrán implementadas las peticiones del cliente

descritas en las historias de usuario 3, 6, 8, 9, 12, 14, 16 y 18 en las cuales se hace mención a la posibilidad de ver un perfil de usuario teniendo como precondition que el usuario tiene que haber creado ya su cuenta, modificar un paciente, así como permitir buscar los mismos en el sistema, además de generar los reportes estadísticos y actualizar las historias clínicas. Ofrece la posibilidad también de buscar una hoja de cargo, una remisión y una historia clínica.

Iteración 3:

En esta iteración serán implementadas las funcionalidades de menos prioridad. Estas tienen el propósito de brindar al cliente comodidad en la gestión de otras tareas asociadas a las de alta prioridad, entre ellas tenemos las historias de usuario 2, 7, 11 y 17 respectivamente las cuales van dirigidas a la eliminación de cuentas de los usuarios, a la eliminación de los datos de un paciente, la modificación de hojas de cargo y la eliminación de historias clínicas.

3.3.3 Plan de duración de las iteraciones

En XP se crea un plan de duración de cada una de las iteraciones, como parte de su ciclo de vida. Su objetivo fundamental es mostrar la duración de cada iteración y el orden en que serán implementadas las historias de usuario en cada una de éstas. A continuación se presenta el plan de duración que se trazó el equipo de desarrollo para la implementación de SIGIGO.

Tabla 22: Plan de duración de las iteraciones

Iteraciones	Orden de las historias usuario a implementar	Duración total de las iteraciones
Iteración 1	1- Crear perfil de usuario 2- Autenticar usuario 3- Introducir datos de un paciente	3 semanas

	<ul style="list-style-type: none">4- Crear Hoja de Cargo5- Crear remisión6- Crear historia clínica	
Iteración 2	<ul style="list-style-type: none">1- Ver perfil de un usuario2- Buscar datos personales de un paciente3- Generar reporte estadístico4- Buscar Hoja de cargo5- Buscar remisión6- Buscar historia clínica7- Modificar los datos de un paciente8- Actualizar historia clínica	3 semanas
Iteración 3	<ul style="list-style-type: none">1- Eliminar perfil de usuario2- Eliminar los datos de un paciente3- Modificar Hoja de Cargo4- Eliminar historia clínica	2 semanas

3.3.4 Plan de entrega

A continuación se expone para la fase de implementación el plan de liberaciones elaborado por el equipo de desarrollo, para en el cual primeramente se unieron las funcionalidades pertinentes a un mismo tema en módulos.

Tabla 23: Funcionalidades por módulos

Módulos	Historias de usuario que abarca
administración	<ol style="list-style-type: none">1- Crear perfil de usuario2- Eliminar perfil de usuario3- Autenticar usuario4- Ver perfil de un usuario
paciente	<ol style="list-style-type: none">1- Introducir datos de un paciente2- Modificar los datos de un paciente3- Eliminar los datos de un paciente4- Buscar los datos de un paciente
reporte	<ol style="list-style-type: none">1- Generar reporte estadístico
hoja de cargo	<ol style="list-style-type: none">1- Crear Hoja de Cargo2- Actualizar Hoja de Cargo3- Buscar Hoja de cargo
remisión	<ol style="list-style-type: none">1- Elaborar remisión2- Buscar remisión
historia clínica	<ol style="list-style-type: none">1- Crear historia clínica2- Actualizar historia clínica3- Eliminar historia clínica4- Buscar historia clínica

Tabla 24: Plan de duración entrega

Módulos	Final 1ra iteración 3ra semana de abril	Final 2da iteración 2da semana de mayo	Final 3ra iteración 4ta semana de mayo
administración	v0.1	v1.0	finalizado
paciente			v1.0
reporte	v0.1	v1.0	finalizado
hoja de cargo	v0.1	v1.0	finalizado
remisión	v0.1	v1.0	finalizado
historia clínica	v0.1	v1.0	finalizado

3.4 Conclusiones

En el capítulo se hace referencia a las fases de Exploración y Planificación concernientes al desarrollo del sistema. Para ello se realizaron las descripciones de cada una de las historias de usuarios, la estimación de esfuerzos por historias de usuario, el plan de duración de las iteraciones y el plan de entregas.

Capítulo 4: Implementación y Prueba

4.1 Introducción

La metodología Extreme Programming plantea que la implementación de un software debe realizarse de forma iterativa, obteniendo un producto funcional al finalizar cada iteración. Producto que recomienda debe ser probado y mostrado al cliente con el objetivo de incrementar la visión de los desarrolladores y sustentar además la retroalimentación cliente-desarrollador.

4.2 Previo a la implantación (Diseño)

Para el diseño de las aplicaciones, la metodología XP no requiere la presentación del sistema mediante diagramas de clases utilizando notación UML, en su lugar se usan otras técnicas como las tarjetas CRC (Contenido, Responsabilidad y Colaboración) [19].

No obstante el uso de estos diagramas puede aplicarse siempre y cuando influyan en el mejoramiento de la comunicación, no sea un peso su mantenimiento, no sean extensos y se enfoquen en la información importante [20].

4.2.1 Tarjetas Clase, Responsabilidad y Colaboración (CRC)

El uso de las tarjetas C.R.C (Clase, Responsabilidad y Colaboración) determinan el comportamiento de cada actividad además permiten al programador centrarse y apreciar el desarrollo orientado a objetos. A continuación se describirán las tarjetas CRC por cada uno de los módulos permitiendo definir y simular los escenarios que garantizarán el buen funcionamiento del diseño.

Tabla 25: Tarjeta CRC Clase Paciente

Clase: Paciente	
Responsabilidad	Colaboración
Introducir datos de un paciente	Historia clínica
Modificar los datos de un paciente	Hoja de Cargo
Buscar datos personales de un paciente	Remisiones
Eliminar los datos de un paciente	

Tabla 26: Tarjeta CRC Clase Historia Clínica

Clase: Historia clínica	
Responsabilidad	Colaboración
Crear historia clínica	Paciente
Modificar historia clínica	
Buscar historia clínica	
Eliminar historia clínica	

Tabla 27: Tarjeta CRC Clase Hoja de Cargo

Clase: Hoja de Cargo	
Responsabilidad	Colaboración
Crear Hoja de Cargo	Paciente
Modificar Hoja de Cargo	
Buscar Hoja de cargo	

Tabla 28: Tarjeta CRC Clase Remisión

Clase: Remisión	
Responsabilidad	Colaboración
Crear remisión Buscar remisión	Paciente

Tabla 29: Tarjeta CRC Clase Reporte

Clase: Reporte	
Responsabilidad	Colaboración
Generar reporte estadístico	Hoja de Cargo Paciente

4.3 Fase de Implementación

En esta fase se realiza la implementación de las historias de usuario que fueron seleccionadas por cada iteración llevando a cabo el plan de iteraciones, el mismo puede modificarse siempre que sea necesario. Como parte de este plan de iteraciones las historias de usuario se descomponen en tareas de programación, tareas que no deben demorar más tres días en desarrollo

4.3.1 Tareas de programación por iteraciones

Cada historia de usuario como funcionalidad del sistema estará dividida por una o varias tareas de programación, estas tareas son pasos lógicos a seguir por el programador para realizar la implementación de una historia de usuario. A continuación para una mejor comprensión se detallarán las tareas de programación por cada iteración según corresponda con la historia de usuario relacionada.

4.3.1.1 Iteración 1

Tabla 30: Tiempo real de implementación por módulo (iteración 1)

Módulos	Historias de usuario	Tiempo de implementación (semanas)	
		Estimación	Real
administración	Crear perfil de usuario	0.4	0.2
	Autenticar usuario	0.6	0.6
paciente	Introducir datos de un paciente	0.4	0.2
hoja de cargo	Crear Hoja de Cargo	0.4	0.4
remisión	Crear remisión	0.4	0.2
historia clínica	Crear historia clínica	0.6	0.4

Tabla 31: Tarea de programación por historias de usuario (Iteración 1)

Historias de usuario	Tarea de programación por historias de usuario
Crear perfil de usuario	<ol style="list-style-type: none"> 1. Mostrar interfaz "Crear Perfil" 2. Crear perfil

Autenticar usuario	<ol style="list-style-type: none">1. Mostrar interfaz autenticarse2. Insertar datos3. Comprobar datos
Introducir datos de un paciente	<ol style="list-style-type: none">1. Mostrar interfaz buscar paciente2. Realizar búsqueda3. Comprobar datos4. Insertar datos
Crear Hoja de Cargo	<ol style="list-style-type: none">1. Mostrar interfaz buscar paciente2. Mostrar interfaz crear hoja de cargo3. Crear hoja de cargo
Crear remisión	<ol style="list-style-type: none">1. Mostrar interfaz buscar paciente2. Mostrar interfaz crear remisión3. Crear remisión
Crear historia clínica	<ol style="list-style-type: none">1. Mostrar interfaz buscar paciente2. Mostrar interfaz crear historia clínica3. Crear historia clínica

Ver anexo 1-18

4.3.1.2 Iteración 2

Tabla 32: Tiempo real de implementación por módulo (iteración 2)

Módulos	Historias de usuario	Tiempo de implementación (semanas)	
		Estimación	Real
administración	Ver perfil de un usuario	0.6	0.2
paciente	Buscar datos personales de un paciente	0.4	0.2
reporte	Generar reporte estadístico	0.6	0.2
hoja de cargo	Buscar Hoja de cargo	0.2	0.2
remisión	Buscar remisión	0.2	0.2
historia clínica	Buscar historia clínica	0.2	0.2

Tabla 33: Tarea de programación por historias de usuario (Iteración 2)

Historias de usuario	Tarea de programación por historias de usuario
Ver perfil de un usuario	1- Mostrar interfaz buscar perfil 2- Realizar búsqueda
Buscar datos personales de un paciente	1- Mostrar interfaz buscar paciente 2- Realizar búsqueda 3- Comprobar datos 4- Mostrar datos

Generar reporte estadístico	<ol style="list-style-type: none"> 1- Mostrar interfaz generar reportes 2- Generar reporte
Buscar Hoja de cargo	<ol style="list-style-type: none"> 1- Mostrar interfaz buscar hoja de cargo 2- Realizar búsqueda 3- Comprobar datos 4- Mostrar datos
Buscar remisión	<ol style="list-style-type: none"> 1- Mostrar interfaz buscar remisión 2- Realizar búsqueda 3- Comprobar datos 4- Mostrar datos
Buscar historia clínica	<ol style="list-style-type: none"> 1- Mostrar interfaz buscar paciente 2- Realizar búsqueda 3- Comprobar datos 4- Mostrar datos

Ver anexos 19-44

4.3.1.3 Iteración 3

Tabla 34: Tiempo real de implementación por módulo (iteración 3)

Módulos	Historias de usuario	Tiempo de implementación (semanas)	
		Estimación	Real
administración	Eliminar perfil de usuario	0.2	0.1

paciente	Eliminar los datos de un paciente	0.2	0.2
hoja de cargo	Modificar Hoja de Cargo	0.4	0.4
historia clínica	Eliminar historia clínica	0.2	0.2

Tabla 35: Tarea de programación por historias de usuario (Iteración 3)

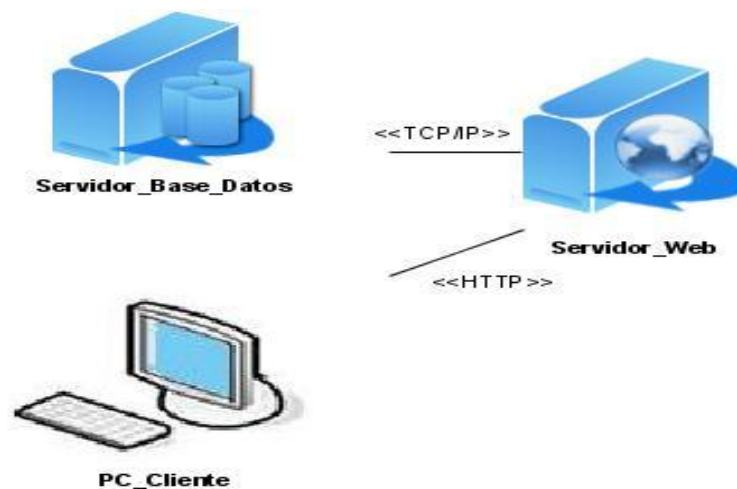
Historias de usuario	Tarea de programación por historias de usuario.
Eliminar perfil de usuario	<ol style="list-style-type: none"> 1- Mostrar interfaz buscar perfil. 2- Realizar búsqueda 3- Comprobar datos 4- Eliminar perfil
Eliminar los datos de un paciente	<ol style="list-style-type: none"> 1- Mostrar interfaz buscar paciente 2- Realizar búsqueda 3- Comprobar datos 4- Eliminar datos
Modificar Hoja de Cargo	<ol style="list-style-type: none"> 1- Buscar hoja de cargo 2- Modificar datos
Eliminar historia clínica	<ol style="list-style-type: none"> 1- Mostrar interfaz buscar paciente 2- Realizar búsqueda 3- Comprobar datos 4- Eliminar datos

Ver anexo 45-58

4.3.2 Arquitectura de hardware (Diagrama de Despliegue)

El diagrama de despliegue es un tipo de diagrama del Lenguaje Unificado de Modelado que se utiliza para modelar el hardware utilizado en la implementación de sistemas y las relaciones físicas entre sus componentes. En el diagrama que se muestra a continuación la aplicación cuenta con la computadora cliente la cual mantiene comunicación con un servidor web (Apache) que su vez se comunica con un sistema gestor de base de datos (PostgreSQL).

Imagen 6: Diagrama de despliegue



4.4 Fase de prueba

Uno de los pilares de la metodología X.P es el uso de test para comprobar el funcionamiento de los códigos que se vayan implementando. Esta metodología divide las pruebas en dos grupos, los cuales se conocen como: pruebas unitarias y pruebas de aceptación.

Las pruebas unitarias son realizadas por los programadores y son las encargadas de verificar el código de forma automática, entre sus ventajas figuran las siguientes:

- Los errores son más fáciles de localizar
- Se reducen los “efectos secundarios”
- Se da más seguridad al programador
- Cada prueba se convierte en un ejemplo de uso

La programación de pruebas unitarias puede ser una tarea laboriosa, ya que requiere un tiempo que no tendríamos que invertir en caso de no hacerlas. Además es preciso percatarse, de que este tipo de prueba no descubrirá todos los errores del código, sino que sólo probará las unidades por sí solas.

Al ser las distintas funcionalidades de la aplicación no demasiado extensas, no se harán test que analicen partes de las mismas, sino que las pruebas se realizarán para las funcionalidades generales que debe cumplir el software.

4.4.1 Pruebas de aceptación

Las pruebas de aceptación son un tipo de caja negra orientadas a evaluar las distintas tareas en las que ha sido dividida una historia de usuario. Para asegurar el funcionamiento final de una determinada historia los test son creados y usados por los clientes para comprobar que éstas cumplen su cometido. Un relato de usuario puede tener una o varias pruebas de aceptación y no está completo hasta no haberlas pasado todas. El cliente es el máximo responsable de verificar cada una de las pruebas y de priorizar la corrección de las pruebas fallidas.

Tabla 36: Prueba de aceptación Nro.1 (Iteración 1)

Caso de Prueba de Aceptación	
Código: HU1_P1	Historia de Usuario: 1
Nombre: Crear perfil	

Descripción: Prueba para funcionalidad crear perfil de usuario.
Condiciones de Ejecución: El usuario primeramente deberá autenticarse, además poseerá el rol de administrador (Marisol González Vigoa -> <i>administradora</i>). Se introducirán datos de un usuario que no se encuentra registrado (Ariagnys Quintana Montalvo).
Entrada / Pasos de ejecución: El administrador hace clic en la opción Crear perfil de usuario. El administrador introduce el nombre usuario para realizar la búsqueda (ariagnysqm). Introduce los restantes datos.
Resultado Esperado: El sistema muestra el siguiente mensaje “El perfil ha sido creado satisfactoriamente”.
Evaluación de la Prueba: Satisfactoria

Tabla 37: Prueba de aceptación Nro.2 (Iteración 1)

Caso de Prueba de Aceptación	
Código: HU1_P2	Historia de Usuario: 1
Nombre: Crear perfil	
Descripción: Prueba para funcionalidad crear perfil de un usuario.	
Condiciones de Ejecución: El usuario primeramente deberá autenticarse y poseerá el rol de administrador (Marisol González Vigoa -> <i>administradora</i>). Se introducirán datos de un usuario ya registrado (Ariagnys Quintana Montalvo).	
Entrada / Pasos de ejecución:	

El administrador hace clic en la opción Crear perfil de usuario. Se muestra la interfaz para realizar la búsqueda. El administrador introduce el nombre de usuario (ariagnysqm).
Resultado Esperado: El sistema muestra el siguiente mensaje: "El usuario ya existe".
Evaluación de la Prueba: Satisfactoria

Tabla 38: Prueba de aceptación Nro.3 (Iteración 1)

Caso de Prueba de Aceptación	
Código: HU4_P4	Historia de Usuario: 4
Nombre: Introducir datos de usuario	
Descripción: Prueba para funcionalidad autenticación de usuario.	
Condiciones de Ejecución: Se insertaran datos válidos.	
Entrada / Pasos de ejecución: Se hace clic en la opción autenticar Se introducen los datos (Maria del Carmen Perojo Medrano -> mperajo, pas-> mperajo).	
Resultado Esperado: Se muestra las opciones correspondientes al rol que desempeña.	
Evaluación de la Prueba: Satisfactoria	

Tabla 39: Prueba de aceptación Nro.4 (Iteración 1)

Caso de Prueba de Aceptación	
Código: HU4_P5	Historia de Usuario: 4
Nombre: Introducir datos de usuario	

Descripción: Prueba para funcionalidad autenticación de usuario.
Condiciones de Ejecución: Se insertaran datos inválidos.
Entrada / Pasos de ejecución: Se hace clic en la opción autenticar Se introducen el nombre de usuario y la contraseña incorrecta (Maria del Carmen Perojo Medrano -> mperogo, pas-> mperogo).
Resultado Esperado: Se muestra el siguiente mensaje “El nombre de usuario o la contraseña son incorrectos”.
Evaluación de la Prueba: Satisfactoria

Tabla 40: Prueba de aceptación Nro.5 (Iteración 1)

Caso de Prueba de Aceptación	
Código: HU5_P5	Historia de Usuario: 5
Nombre: Introducir datos de una paciente	
Descripción: Prueba para funcionalidad de ingresar un paciente en el sistema.	
Condiciones de Ejecución: El usuario primeramente deberá autenticarse y poseerá el rol de ginecóloga, enfermeras o médico (amontalvo -> enfermera de ginecología). Se introducirán datos válidos, solo que la paciente ya fue insertada con anterioridad (Lienis Abreu Hernández)	
Entrada / Pasos de ejecución: Se introducen los datos correspondientes a la paciente pero con un CI que ya existe. (CI: 86111413695)	
Resultado Esperado: Se muestra el siguiente mensaje “La paciente existe”.	
Evaluación de la Prueba: Satisfactoria	

Tabla 41: Prueba de aceptación Nro.6 (Iteración 1)

Caso de Prueba de Aceptación	
Código: HU10_P6	Historia de Usuario: 10
Nombre: Crear hoja de Cargo	
Descripción: Prueba para funcionalidad crear hoja de cargo	
Condiciones de Ejecución: Se introducirán datos inválidos.	
Entrada / Pasos de ejecución: Se introducirá una fecha de creación de la hoja que no se corresponde (fecha: 20/12/2009, consulta: regulación de la fecundidad)	
Resultado Esperado: Se muestra el siguiente mensaje "Error en la fecha".	
Evaluación de la Prueba: Satisfactoria	

Tabla 42: Prueba de aceptación Nro.7 (Iteración 1)

Caso de Prueba de Aceptación	
Código: HU13_P7	Historia de Usuario: 13
Nombre: Crear remisión	
Descripción: Prueba para funcionalidad crear una remisión	
Condiciones de Ejecución: Se introducirán datos válidos.	
Entrada / Pasos de ejecución: Se introducirá un nombre de paciente inválido (Yudislandry Águila González). Se realizará la búsqueda.	
Resultado Esperado: Se mostrara el mensaje siguiente "El paciente no existe". No se visualizará la interfaz de crear remisión.	

Evaluación de la Prueba: Satisfactoria

Tabla 43: Prueba de aceptación Nro.8 (Iteración 1)

Caso de Prueba de Aceptación	
Código: HU15_P8	Historia de Usuario: 15
Nombre: Crear historia clínica.	
Descripción: Prueba para funcionalidad crear historia clínica.	
Condiciones de Ejecución: Se introducirán datos válidos.	
Entrada / Pasos de ejecución: El usuario deberá autenticarse y poseerá el rol de ginecóloga (Marisol González Vigoa - > ginecóloga). El usuario seleccionará la opción crear historia clínica. Introducirá los datos correspondiente a la paciente	
Resultado Esperado: Se mostrara el siguiente mensaje “La Historia clínica ha sido creada satisfactoriamente”.	
Evaluación de la Prueba: Satisfactoria	

Tabla 44: Prueba de aceptación Nro.9 (Iteración 2)

Caso de Prueba de Aceptación	
Código: HU3_P9	Historia de Usuario: 3
Nombre: Buscar perfil	
Descripción: Prueba para funcionalidad buscar perfil	
Condiciones de Ejecución: Se introducirán datos válidos.	

<p>Entrada / Pasos de ejecución:</p> <p>El usuario deberá autenticarse y poseerá el rol de ginecóloga (Marisol González Vigoa - > ginecóloga).</p> <p>El usuario seleccionará la opción buscar perfil .</p> <p>Se introducirá el usuario para realizar la búsqueda.</p>
<p>Resultado Esperado:</p> <p>Se mostrara mostrará el perfil del usuario buscado.</p>
<p>Evaluación de la Prueba: Satisfactoria</p>

Tabla 45: Prueba de aceptación Nro.10 (Iteración 2)

Caso de Prueba de Aceptación	
Código: HU8_P10	Historia de Usuario: 8
Nombre: Buscar datos(comunes) de una paciente	
Descripción: Prueba para funcionalidad buscar datos de una paciente	
<p>Condiciones de Ejecución:</p> <p>El usuario deberá autenticarse y poseerá el rol de ginecóloga, enfermera o médico (Marisol González Vigoa -> ginecóloga).</p> <p>Se introducirán datos válidos.</p>	
<p>Entrada / Pasos de ejecución:</p> <p>El usuario seleccionará la opción buscar datos</p> <p>Se introducirá el CI para realizar la búsqueda.</p>	
Resultado Esperado: Se visualizaran los datos de la paciente.	
Evaluación de la Prueba: Satisfactoria	

Tabla 46: Prueba de aceptación Nro.11 (Iteración 2)

Caso de Prueba de Aceptación	
Código: HU9_P11	Historia de Usuario: 9
Nombre: Generar reporte	
Descripción: Prueba para funcionalidad generar reporte	
Condiciones de Ejecución: El usuario deberá autenticarse y poseerá el rol de ginecóloga. (Marisol González Vigoa -> ginecóloga). Se introducirán datos válidos.	
Entrada / Pasos de ejecución: El usuario seleccionará la opción generar reporte Introducirá una fecha válida.	
Resultado Esperado: Se visualizará el reporte correspondiente.	
Evaluación de la Prueba: Satisfactoria	

Tabla 47: Prueba de aceptación Nro.12 (Iteración 2)

Caso de Prueba de Aceptación	
Código: HU12_P12	Historia de Usuario: 12
Nombre: Buscar hoja de cargo	
Descripción: Prueba para funcionalidad buscar hoja de cargo	
Condiciones de Ejecución: El usuario deberá autenticarse y poseerá el rol de ginecóloga. (Marisol González Vigoa	

-> ginecóloga). Se introducirán datos inválidos.
Entrada / Pasos de ejecución: El usuario seleccionará la opción buscar hoja de cargo Introducirá una fecha no válida (20/11/2009).
Resultado Esperado: Se mostrará el mensaje: Error en la entrada de fecha”.
Evaluación de la Prueba: Satisfactoria

Tabla 48: Prueba de aceptación Nro.13 (Iteración 2)

Caso de Prueba de Aceptación	
Código: HU12_P13	Historia de Usuario: 12
Nombre: Buscar hoja de cargo	
Descripción: Prueba para funcionalidad buscar hoja de cargo	
Condiciones de Ejecución: El usuario deberá autenticarse y poseerá el rol de ginecóloga. (Marisol González Vigoa -> ginecóloga). Se introducirán datos válidos	
Entrada / Pasos de ejecución: El usuario seleccionará la opción buscar hoja de cargo Introducirá una fecha no válida (6/5/2009).	
Resultado Esperado: Se visualizará la hoja de cargo correspondiente.	
Evaluación de la Prueba: Satisfactoria	

Tabla 49: Prueba de aceptación Nro.14 (Iteración 2)

Caso de Prueba de Aceptación	
Código: HU6_P14	Historia de Usuario: 6
Nombre: Modificar datos de una paciente	
Descripción: Prueba para funcionalidad modificar datos de una paciente	
Condiciones de Ejecución: El usuario deberá autenticarse y poseerá el rol de ginecóloga. (Marisol González Vigoa -> ginecóloga). Se introducirán datos inválidos.	
Entrada / Pasos de ejecución: El usuario seleccionará la opción modificar datos de una paciente Introducirá un CI que no es el correspondiente.	
Resultado Esperado: Se mostrará el mensaje: "No se puede modificar el CI".	
Evaluación de la Prueba: Satisfactoria	

Tabla 50: Prueba de aceptación Nro.15 (Iteración 2)

Caso de Prueba de Aceptación	
Código: HU6_P15	Historia de Usuario: 6
Nombre: Modificar datos de una paciente	
Descripción: Prueba para funcionalidad modificar datos de una paciente	
Condiciones de Ejecución: El usuario deberá autenticarse y poseerá el rol de ginecóloga. (Marisol González Vigoa	

-> ginecóloga). Se introducirán datos válidos.
Entrada / Pasos de ejecución: El usuario seleccionará la opción modificar datos de una paciente Se cambiará una nueva dirección particular.
Resultado Esperado: Se mostrará el mensaje: "Modificación satisfactoria".
Evaluación de la Prueba: Satisfactoria

Tabla 51: Prueba de aceptación Nro.16 (Iteración 2)

Caso de Prueba de Aceptación	
Código: HU16_P16	Historia de Usuario: 16
Nombre: Modificar historia clínica	
Descripción: Prueba para funcionalidad modificar historia clínica	
Condiciones de Ejecución: El usuario deberá autenticarse y poseerá el rol de ginecóloga. (Marisol González Vigoa -> ginecóloga). Se introducirán datos válidos.	
Entrada / Pasos de ejecución: El usuario seleccionará la opción modificar historia clínica Se cambiara el tipo de anticonceptivo.	
Resultado Esperado: Se mostrará el mensaje: "Modificación satisfactoria".	
Evaluación de la Prueba: Satisfactoria	

Tabla 52: Prueba de aceptación Nro.17 (Iteración 2)

Caso de Prueba de Aceptación	
Código: HU14_P17	Historia de Usuario: 14
Nombre: Buscar remisión	
Descripción: Prueba para funcionalidad buscar remisión	
Condiciones de Ejecución: El usuario deberá autenticarse y poseerá el rol de ginecóloga. (Marisol González Vigoa -> ginecóloga). Se introducirán datos válidos.	
Entrada / Pasos de ejecución: El usuario seleccionará la opción buscar remisión Se introducirá la fecha (29/4/2009) y la consulta (Obstetricia)	
Resultado Esperado: Se visualizará la remisión correspondiente.	
Evaluación de la Prueba: Satisfactoria	

Tabla 53: Prueba de aceptación Nro.18 (Iteración 2)

Caso de Prueba de Aceptación	
Código: HU18_P18	Historia de Usuario: 18
Nombre: Buscar historia clínica	
Descripción: Prueba para funcionalidad buscar historia clínica	
Condiciones de Ejecución: El usuario deberá autenticarse y poseerá el rol de ginecóloga. (Marisol González Vigoa	

-> ginecóloga). Se introducirán datos válidos.
Entrada / Pasos de ejecución: El usuario seleccionará la opción buscar historia clínica Se introducirá el CI de la paciente (84042913299) y el nombre de la historia (Ginecológica).
Resultado Esperado: Se visualizará la historia clínica correspondiente a (Yanisvel Suárez Arbolay).
Evaluación de la Prueba: Satisfactoria

Tabla 54: Prueba de aceptación Nro.19 (Iteración 2)

Caso de Prueba de Aceptación	
Código: HU18_P19	Historia de Usuario: 18
Nombre: Buscar historia clínica	
Descripción: Prueba para funcionalidad buscar historia clínica	
Condiciones de Ejecución: El usuario deberá autenticarse y poseerá el rol de ginecóloga. (Marisol González Vigoa -> ginecóloga). Se introducirán datos inválidos.	
Entrada / Pasos de ejecución: El usuario seleccionará la opción buscar historia clínica Se introducirá el CI de la paciente (87011826856) y el nombre de la historia (Regulación de la fecundidad)	
Resultado Esperado: Se mostrará el mensaje: “La historia clínica no existe”.	
Evaluación de la Prueba: Satisfactoria	

Tabla 55: Prueba de aceptación Nro.20 (Iteración 3)

Caso de Prueba de Aceptación	
Código: HU2_P20	Historia de Usuario: 2
Nombre: Eliminar perfil de usuario.	
Descripción: Prueba para funcionalidad eliminar perfil de usuario.	
Condiciones de Ejecución: El usuario deberá autenticarse y poseerá el rol de administrador. (Marisol González Vigoa -> administrador). Se introducirán datos válidos.	
Entrada / Pasos de ejecución: El usuario seleccionará la opción buscar historia clínica Se introducirá el nombre de usuario (ariagnysqm)	
Resultado Esperado: Se mostrará el mensaje: "Perfil eliminado satisfactoriamente".	
Evaluación de la Prueba: Satisfactoria	

Tabla 56: Prueba de aceptación Nro.21 (Iteración 3)

Caso de Prueba de Aceptación	
Código: HU7_P21	Historia de Usuario: 7
Nombre: Eliminar paciente.	
Descripción: Prueba para funcionalidad eliminar paciente.	
Condiciones de Ejecución: El usuario deberá autenticarse y poseerá el rol de administrador. (Marisol González	

Vigoa -> administrador). Se introducirán datos válidos.
Entrada / Pasos de ejecución: El usuario seleccionará la opción eliminar paciente. Se introducirá el CI de la paciente (8404291 3299).
Resultado Esperado: Se mostrará el mensaje: “La paciente fue eliminada satisfactoriamente”.
Evaluación de la Prueba: Satisfactoria

Tabla 57: Prueba de aceptación Nro.22 (Iteración 3)

Caso de Prueba de Aceptación	
Código: HU11_P22	Historia de Usuario: 11
Nombre: Modificar Hoja de Cargo	
Descripción: Prueba para funcionalidad modificar Hoja de Cargo	
Condiciones de Ejecución: El usuario deberá autenticarse y poseerá el rol de administrador. (Marisol González Vigoa -> administrador). Se introducirán datos válidos.	
Entrada / Pasos de ejecución: El usuario seleccionará la opción eliminar paciente. Se introducirá un nuevo número de teléfono UCI (837- 3411)	
Resultado Esperado: Se mostrará el mensaje: “Modificación satisfactoria”.	
Evaluación de la Prueba: Satisfactoria	

Tabla 58: Prueba de aceptación Nro.23 (Iteración 3)

Caso de Prueba de Aceptación	
Código: HU17_P23	Historia de Usuario: 17
Nombre: Eliminar historia clínica	
Descripción: Prueba para funcionalidad eliminar historia clínica	
Condiciones de Ejecución: El usuario deberá autenticarse y poseerá el rol de administrador. (Marisol González Vigoa -> administrador). Se introducirán datos válidos.	
Entrada / Pasos de ejecución: El usuario seleccionará la opción eliminar historia clínica Se seleccionara la opción eliminar historia clínica con una historia existente ()	
Resultado Esperado: Se mostrará el mensaje: "Eliminación satisfactoria".	
Evaluación de la Prueba: Satisfactoria	

4.5 Conclusiones

En el presente capítulo se detallan las tres iteraciones llevadas a cabo durante la etapa de implementación del sistema, las tareas de programación generadas por cada historia de usuario y los diagramas de clases persistentes, modelo de datos y despliegue como complementarios para un mejor entendimiento de la aplicación desarrollada.

Conclusiones

Una vez finalizado el análisis, diseño e implementación del Sistema de Gestión de Información Gineco-Obstétrico para el Policlínico-Hospital Ernesto Che Guevara de la Universidad de las Ciencias Informáticas se arribó a las siguientes conclusiones:

Durante el ciclo de vida del producto se realizó un estudio de los principales sistemas de gestión de información hospitalaria existentes en Cuba y el Mundo; llegándose a demostrar la necesidad de desarrollar un sistema capaz de automatizar los procesos de gestión de información Gineco-Obstétrico en dicho Policlínico-Hospital. Esta demostración proporcionó el punto de partida para la realización de un minucioso estudio sobre: tipos de aplicaciones y patrones arquitectónicos, metodologías de desarrollo de software, los lenguajes de modelado y programación, los sistemas gestores de base de datos, servidores web así como de un conjunto de técnicas y herramientas más utilizadas hoy en el campo de la informática con el objetivo de seleccionar las más adecuadas para la realización de este sistema.

El tránsito por cada una de las etapas de la metodología Extreme Programming, propició la generación de un grupo de artefactos que permiten la comprensión del producto desde sus inicios y que dieron lugar una aplicación funcional que responde a las necesidades del cliente.

Recomendaciones

Con el presente trabajo se exhorta a:

- Profundizar el tema abordado y poner a prueba el sistema durante un período de tiempo significativo con el objetivo de detectar posibles debilidades en el desarrollo del sistema en su primera versión.

- Agregar al sistema en su segunda versión los módulos que hacen referencia a otros procesos del negocio que no fueron desarrollados.

- Estudio de un módulo para mantenimiento de la base de datos.

- Valorar la posible de integración del sistema con otros ya existentes en el Policlínico-Hospital.

- Valorar la posibilidad de hacer extensiva esta propuesta a otras áreas del Policlínico-Hospital.

Referencias bibliográficas

1. **ONE.** (s.f.). ONE Oficina Nacional de Estadísticas. República de Cuba. [Citado el 15 de 1 de 2009.] de ONE Oficina Nacional de Estadísticas. República de Cuba

Disponible en : http://www.one.cu/aec2007/esp/17_tabla_cuadro.htm

2. **Beck, kent. 1999.** *"Extreme Programming Explained. Embrace Change.* 1999. [Citado el: 25 de 1 de 2009.]

3. **Softel.** Softel soluciones informaticas. *Softel soluciones informaticas.* [En línea] [Citado el: 25 de 1 de 2009.] Disponible en: <http://www.softel.cu/actual.htm>

4. **Softel soluciones informaticas.** *Softel soluciones informaticas.* [En línea] [Citado el: 25 de 1 de 2009.] Disponible en: <http://www.softel.cu/productos.htm>

5. **CEDISAP.** CEDISAP Informática Médica. *CEDISAP Informática Médica.* [En línea] [Citado el: 26 de 1 de 2009.] Disponible en: <http://www.sld.cu/instituciones/cedisap//Medclien.htm>

6. **GineSalus.** (s.f.). *GineSalus.* Obtenido de GineSalus. [En línea] [Citado el: 26 de 1 de 2009.]

Disponible en: http://www.ginesalus.com/salus_noticia.aspx?noticia=61&info=SALUS

7. **Knocer.** (2008-2009). *Ginepack Solución software integral y modular para el área de .* Obtenido de Ginepack Solución software integral y modular para el área de ginecología. [En línea] [Citado el: 29 de 1 de 2009.]

Disponible en: <http://www.knocer.com/ginepack/intro.php>

8. Konocer. (2004-2005). *Ginecontrol version 2.5 "características generales"*. Obtenido de Ginecontrol version 2.5 "características generales" . [En línea] [Citado el: 29 de 1 de 2009.]

Disponible en: <http://www.konocer.com/ginecontrol/general.html>

9. Jacobson, I., Booch, G., & Rumbaugh, J. El Proceso Unificado de Desarrollo de Software. [Citado el: 2 de 2 de 2009.]

10. OsmosisLatina. (s.f.). *OsmosisLatina*. Recuperado el 2 de 2 de 2009, de OsmosisLatina: <http://www.osmosislatina.com/lenguajes/uml/basico.htm>

11. Cibernetia. Cibernetia. *Cibernetia*. [En línea] [Citado el: 3 de 2 de 2009.]

Disponible en: http://www.cibernetia.com/manuales/instalacion_servidor_web/1_conceptos_basicos.php

12. DesarrolloWeb.com. DesarrolloWeb.com. *DesarrolloWeb.com*. [En línea] [Citado el: 3 de 2 de 2009.]

Disponible en: <http://www.desarrolloweb.com/articulos/1112.php>

13. AOLSERVER. Una alternativa a Apache AOLSERVER. *Una alternativa a Apache AOLSERVER*. [En línea] [Citado el: 3 de 2 de 2009.]

Disponible en: <http://www.linux-magazine.es/issue/12/AOLServer.pdf>.

14. microsoft. microsoft. *microsoft*. [En línea] [Citado el: 3 de 2 de 2009.]

Disponible en: <http://www.microsoft.com/spain/windowsserver2003/technologies/webapp/iis.msp>

15. © 2009 Microsoft. Microsoft Download Center. *Microsoft Download Center*. [En línea] 2009. [Citado el: 5 de 3 de 2009.] Disponible en:

<http://download.microsoft.com/download/4/F/F/4FF88340-43CC-4C5B-8E50-09002969D0DD/20051122-ARC-BA.ppt>

16. @ WillyDev . WillyDev. *WillyDev*. [En línea] 2003 - 2008. [Citado el: 5 de 3 de 2009.]

Disponible en:

<http://www.willydev.net/InsiteCreation/v1.0/WillyCrawler/2008.05.01.Articulo.Introduccion%20a%20la%20arquitectura%20de%20Soft.pdf>

17. JACOBSON, I. 1998. "*Applying UML in The Unified Process. Rational Software.*". 1998. [Citado el: 10 de 3 de 2009.]

18. Penadés, Patricio Letelier y M^a Carmen. @ WillyDev .NET . @ *WillyDev .NET* . [En línea] 2003 - 2008 . [Citado el: 10 de 3 de 2009.]

Disponible en: <http://www.willydev.net/descargas/masyxp.pdf>

19. Beck, K. *Extreme Programming Explained*. s.l.: Addison Wesley, 2000. [Citado el:10 de 3 de 2009.]

20. XP. *A gentle introduction*. [En línea] 2006. [Citado el: 10 de 3 de 2009.]

Disponible en: <http://www.extremeprogramming.org>.

21. **maestrosdelweb**. (9 de 10 de 2007). *maestros del web*. Recuperado el 5 de 4 de 2009, de maestros del web:

Disponible en: <http://www.maestrosdelweb.com/editorial/modelado-de-datos-e-implementacion-de-la-base-de-datos-primer-nivel-l5>

Bibliografía

- **Aikom. Web y un poco mas.** Aikom. Web y un poco mas. *Aikom. Web y un poco mas.* . [En línea]
 - Disponible en: <http://aikon.com.ve/metodologias-desarrollo-software-extreme-programming/>.
- **AOLSERVER.** Una alternativa a Apache AOLSERVER. Una alternativa a Apache AOLSERVER. [En línea] [Citado el: 3 de 2 de 2009.]
 - Disponible en: <http://www.linux-magazine.es/issue/12/AOLServer.pdf>.
- **babylon.com. (s.f.).** *babylon.com*. Recuperado el 2009, de [babylon.com](http://www.babylon.com)
 - Disponible en: <http://www.babylon.com/>
- **Barbone, V. A. (s.f.).** *XP: Extreme Programming* . Recuperado el 28 de 04 de 2009, de XP: Extreme Programming .
 - Disponible en: <http://ie.fing.edu.uy/~nacho/blandos/seminario/XProg1.html>
- **CalinSoft. (2008-2009).** CalinSoft. Aplicaciones web ventajas y desventajas. *CalinSoft. Aplicaciones web ventajas y desventajas.* [En línea] (2008-2009).
 - Disponible en: <http://www.calinsoft.com/2008/08/aplicaciones-web-ventajas-y-desventajas.html> .
- **Desarrollo web.com.** Desarrollo web.com. Arquitectura cliente-servidor. *Desarrollo web.com. Arquitectura cliente-servidor.* [En línea]
 - Disponible en: <http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html>.
- **El mundo informatico .** El mundo informatico. Patrones Grafs. *El mundo informatico. Patrones Grafs.* [En línea]
 - Disponible en: <http://jorgesaavedra.wordpress.com/category/patrones-grasp/>.

- **Flores, J. Tomás Nogales. 1998.** CSS2 (Cascading Style Sheets 2) . *CSS2 (Cascading Style Sheets 2)* . [En línea] 23 de 08 de 1998.
 - Disponible en: http://paginaspersonales.deusto.es/abaitua/konzeptu/sgml/dmg_css.htm.
- **garbage Collector.** garbage Collector. Sistemas Gestores de bases de datos SGBD. *garbage Collector. Sistemas Gestores de bases de datos SGBD.* [En línea]
 - Disponible en: http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php.
- **htmlpoint.com. 1997 -2006.** htmlpoint.com.Aspectos y características generales . *htmlpoint.com.Aspectos y características generales* . [En línea] 1997 -2006.
 - Disponible en: http://www.htmlpoint.com/javascript/corso/js_02.htm.
- **Itera 2009.** Rational Unified Process. *Rational Unified Process.* [En línea] Itera 2009.

Disponible en:

 - http://www.iteraprocess.com/index.php?option=com_content&task=view&id=18&Itemid=42&limit=1&limitstart=1.
- **La Plaga Tux. 2007.** La Plaga Tux.UML en Linux. *La Plaga Tux.UML en Linux.* [En línea] 2007.
 - Disponible en :<http://plagatux.es/2008/08/uml-en-linux/>.
- **La Webera.es.** La Webera.es. Características de PHP. *La Webera.es. Características de PHP.* [En línea] <http://www.lawebera.es/manuales/php/2-2.php>.
- **Metodologías de desarrollo web.** *Metodologías de desarrollo web.* [En línea]
 - Disponible en: <http://www.monografias.com/trabajos60/metodologias-desarrollo-software/metodologias-desarrollo-software.shtml>.
- **Monografia.com.** Monografia.com. Herramienta Case. [En línea]

- Disponible en : http://html.rincondelvago.com/herramientas-case_2.html.
- **mysql arti:** ¿Qué es MySQL? *mysql arti: ¿Qué es MySQL?* [En línea]
 - Disponible en: http://www.netpecos.org/docs/mysql_postgres/x57.html.
- **Patrones Arquitectónicos.** Patrones Arquitectónicos.Ingeniería del Software de Gestión 3: *Patrones Arquitectónicos.Ingeniería del Software de Gestión 3:* [En línea]
 - Disponible en: <http://isg3.pbwiki.com/>.
- **Patrones de diseño .** *Patrones de diseño .* [En línea]
 - Disponible en: <http://mit.ocw.universia.net/6.170/6.170/f01/pdf/lecture-12.pdf>
- **Prieto, Félix. 2008/2009.** Patrones de diseño. *Patrones de diseño.* [En línea] 2008/2009.
 - Disponible en :http://www.infor.uva.es/~felix/datos/priii/tr_patrones-2x4.pdf.
- **Programacionweb.net. 2003-2009.** Programacionweb.net. *Programacionweb.net.* [En línea] 2003-2009.
 - Disponible en: <http://www.programacionweb.net>.
- **servidores apache.** servidores apache. El servidor de web Apache. *servidores apache. El servidor de web Apache.* [En línea]
 - Disponible en: <http://acsblog.es/articulos/trunk/LinuxActual/Apache/html/x31.html>.
- **Sitio de descargas de software.** *Sitio de descargas de software.* [En línea]
 - Disponible en: [Http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/)
- **Sitio Infomed. Red de salud de Cuba.** Sitio Infomed. Red de salud de Cuba. *Sitio Infomed. Red de salud de Cuba.* [En línea] <http://www.sld.cu/>.

- **Sitio Joven Club de Computación y Electrónica. Villa Clara.** *Sitio Joven Club de Computación y Electrónica. Villa Clara.* [En línea] Última actualización abril 2009.
 - Disponible es: <http://www.vcl.jovenclub.cu/prov/adultomayor/docs/tematica1.pdf>.
- **SlideShare Inc.** All Rights Reserved. (s.f.). *Slideshare.* Recuperado el 28 de 04 de 2009, de Slideshare
 - Disponible en : <http://www.slideshare.net/dbaccess/pruebas-unitarias-uso-de-nunit-dentro-de-proyectos-net>
- **Tecnología orientada a procesos de negocio.** Tecnología orientada a procesos de negocio. Tecnología Internet para procesos de negocio. *Tecnología orientada a procesos de negocio. Tecnología Internet para procesos de negocio.* [En línea]
 - Disponible en: <http://www.proactiva-calidad.com/java/patrones/mvc.html>.
- **wordreference.com. (s.f.).** *wordreference.com.* Recuperado el 2009, de wordreference.com
 - Dispononible en: <http://www.wordreference.com/>

Glosario de términos

AJAX (Asynchronous JavaScript And XML): (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas

API: Es una **interfaz de programación** de aplicaciones (del inglés API: Application Programming Interface). Es un conjunto de rutinas que provee acceso a funciones de un determinado **software**.

Applets: Es un componente de una aplicación que se ejecuta en el contexto de otro programa, por ejemplo un navegador web.

Artefacto: Es una de las múltiples formas de resultados tangibles producidos durante el desarrollo de software.

Autenticación: Es el proceso de intento de verificar la identidad digital del remitente de una comunicación como una petición para conectarse.

Automatización: El conjunto de métodos que sirven para realizar tareas repetitivas en un ordenador.

C: Es un **lenguaje de programación** creado en 1972 por Ken Thompson y Dennis M. Ritchie en los Laboratorios Bell como evolución del anterior lenguaje B.

C++: Es un lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup

Copyright: Conjunto de normas y principios que regulan los derechos morales y patrimoniales que la ley concede a los autores por el solo hecho de la creación de una obra literaria, artística o científica, esté publicada o inédita.

DHTML: Designa el conjunto de técnicas que permiten crear sitios web interactivos utilizando una combinación de lenguaje HTML estático, un lenguaje interpretado en el lado del cliente como JavaScript, el lenguaje de hojas de estilo en cascada (CSS) y la jerarquía de objetos de un DOM.

DOM (Document Object Model): Interfaz de programación de aplicaciones que proporciona un conjunto estándar de objetos para representar documentos HTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos.

Extreme Programming Explained: Es un enfoque de la ingeniería de software formulado por Kent Beck, autor del primer libro sobre la materia, *Extreme Programming Explained: Embrace Change* (1999). Es el más destacado de los procesos ágiles de desarrollo de software.

Exudado: Es el conjunto de elementos extravasados en el proceso inflamatorio, que se depositan en el intersticio de los tejidos o cavidades del organismo.

Ginecología: Parte de la medicina que estudia la fisiología y patología del sistema reproductor femenino.

GTK: Es un conjunto de bibliotecas multiplataforma para desarrollar interfaces gráficas de usuario (GUI),

Hemoglobina (hb): Pigmento proteínico colorante de los glóbulos rojos y plasma sanguíneo que permite el transporte de oxígeno en los vertebrados.

Hipertexto: Nombre que recibe el texto que en la pantalla de una computadora te conduce a otro texto relacionado.

Interfaz gráfica de usuario: Es el artefacto tecnológico de un sistema interactivo que posibilita, a través del uso y la representación del lenguaje visual, una interacción amigable con un sistema informático.

Ingeniería de software: Disciplina o área de la informática que ofrece métodos y técnicas para desarrollar y mantener software de calidad.

La Web (World Wide Web): (Red Global Mundial) sistema de documentos interconectados por enlaces de hipertexto, que se ejecutan en Internet.

Lenguajes de programación: Conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina.

Librerías: conjunto de subprogramas utilizados para desarrollar software.

Microsoft Corporation: Empresa multinacional estadounidense, fundada en 1975 por Bill Gates y Paul Allen. Dedicada al sector de la informática.

Microsoft SQL Server: Sistema de gestión de bases de datos relacionales (SGBD) basado en el lenguaje Transact-SQL, y específicamente en Sybase IQ, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.

Multi-hilo : Característica que permite a una aplicación realizar varias tareas concurrentemente

Multimedia: Término que se aplica a cualquier objeto que usa simultáneamente diferentes formas de contenido informativo como texto, sonido, imágenes, animación y video para informar o entretener al usuario.

Multiplataforma: Término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas.

Multi-usuario: Característica de un sistema operativo o programa que permite proveer servicio y procesamiento a múltiples usuarios simultáneamente (tanto en paralelismo real como simulado).

MySQL: Sistema de gestión de base de datos relacional, multihilo y multiusuario.

Obstetricia: Es por definición la rama de la Medicina dedicada al cuidado de la mujer durante el estado de gestación, el parto y el puerperio.

Página Web: Documento multimedia generalmente en formato HTML, que contiene enlaces hacia otros documentos. Se accede a este documento a través de un servidor web, gracias a una dirección única (URL) y puede ser visualizado desde un navegador.

PDF (Portable Document Format): (formato de documento portátil) es un formato de almacenamiento de documentos, desarrollado por la empresa Adobe Systems.

Perl: Es un lenguaje de programación diseñado por Larry Wall en 1987.

PHP-GTK: Es una extensión para el lenguaje de programación PHP que permite la utilización de GTK.

PHP-Qt: Es una extensión para PHP5 que permite escribir programas con el Toolkit de Qt.

Plataformas: Es una combinación de hardware y software usada para ejecutar aplicaciones; en su forma más simple consiste únicamente de un sistema operativo, una arquitectura, o una combinación de ambos.

PostgreSQL: Sistema de gestión de base de datos relacional orientada a objetos de software libre, publicado bajo la licencia BSD.

Programación Orientada a Objetos (POO): Es un [paradigma de programación](#) que usa [objetos](#) y sus interacciones para diseñar aplicaciones y programas de computadora.

Protocolo de red: Conjunto de reglas que especifican el intercambio de datos u órdenes durante la comunicación entre sistemas.

Protocolo HTTP (HyperText Transfer Protocol): Protocolo usado en cada transacción de la Web (WWW).

Protocolo HTTPS: Protocolo seguro de transferencia de hipertexto. Es un protocolo de red basado en el protocolo HTTP, destinado a la transferencia segura de datos de hipertexto, es la versión segura de HTTP.

Python: Es un [lenguaje de programación interpretado](#) creado por Guido van Rossum en el año 1991

Qt: Es una biblioteca multiplataforma para desarrollar interfaces gráficas de usuario.

Serelogía: Estudio de la naturaleza y producción de los antígenos y los anticuerpos presentes en el suero.

Servidor web: Programa que implementa el protocolo HTTP (hypertext transfer protocol) para transferir lo que llamamos hipertextos, páginas web o páginas HTML (hypertext markup language).

Sistema operativo: Software de sistema o conjunto de programas de computación destinados a realizar muchas tareas entre las que destaca la administración eficaz de sus recursos.

Sitio web: Conjunto de *páginas web*, típicamente comunes a un dominio o subdominio en la World Wide Web en Internet.

Software: Conjunto de programas elaborados por el hombre, que controlan la actuación del computador, haciendo que éste siga en sus acciones una serie de esquemas lógicos predeterminados y pueda

desempeñar tareas inteligentes. Suele sustituirse por expresiones tales como programas (informáticos) o aplicaciones (informáticas).

Software libre: Según la Free Software Foundation, el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software

Software propietario: Se refiere a cualquier programa informático en el que los usuarios tienen limitadas las posibilidades de usarlo, modificarlo o redistribuirlo (con o sin modificaciones), o cuyo código fuente no está disponible o el acceso a éste se encuentra restringido.

Sybase IQ: Motor de bases de datos altamente optimizado para inteligencia empresarial, desarrollado por la empresa Sybase.

Tecnología: Conjunto de habilidades que permiten construir objetos y máquinas para adaptar el medio y satisfacer nuestras necesidades.

Ultrasonido: Onda acústica cuya frecuencia está por encima del límite perceptible por el oído humano.

VIH: Agente infeccioso determinante del síndrome de inmunodeficiencia adquirida (SIDA).

Visual Basic: Lenguaje de programación desarrollado por Alan Cooper para Microsoft.

XML(Extensible Markup Language): Lenguaje de marca es un metalenguaje extensible de etiquetas

Anexos

Iteración 1

Anexo 1: Tarea de Programación: Mostrar interfaz “Crear Perfil”

Tarea de programación	
Tarea Nro. 1	Número Historia (Nro. 1): Crear perfil de usuario.
Nombre tarea: Mostrar interfaz “Crear Perfil”	
Tipo de tarea : Desarrollo	Puntos estimados: 0.1
Fecha inicio: 05/04/09	Fecha fin: 05/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
<p>Descripción: Una vez que el usuario accede al sitio como administrador, le es otorgado el permiso para crear los perfiles. Este seleccionará la opción Crear perfil de usuario e inmediatamente se mostrará la interfaz de usuario correspondiente.</p>	

Anexo 2: Tarea de Programación: “Crear perfil”

Tarea de programación	
Tarea Nro. 2	Número Historia (Nro. 1): Crear perfil de usuario.
Nombre tarea: Crear perfil.	
Tipo de tarea : Desarrollo	Puntos estimados: 0.3
Fecha inicio: 05/04/09	Fecha fin: 06/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	

Descripción: Una vez que se muestra la interfaz Crear perfil el usuario deberá insertar los datos en los campos y seleccionará el botón Crear perfil. En caso que haya algún error el sistema lanzará un mensaje relacionado con el mismo. En caso contrario el usuario seleccionará “Guardar perfil”. Se mostrará el mensaje “El perfil ha sido creado satisfactoriamente”.

Anexo 3: Tarea de Programación: “Mostrar interfaz autenticarse”

Tarea de programación	
Tarea Nro. 1	Número Historia (Nro. 4): Autenticar usuario
Nombre tarea: Mostrar interfaz autenticarse	
Tipo de tarea : Desarrollo	Puntos estimados: 0.1
Fecha inicio: 17/04/09	Fecha fin: 17/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El usuario selecciona la opción “Autenticarse” visualizándose inmediatamente la interfaz correspondiente.	

Anexo 4: Tarea de Programación: “Insertar Datos”

Tarea de programación	
Tarea Nro. 2	Número Historia (Nro. 4): Autenticar usuario
Nombre tarea: Insertar datos	
Tipo de tarea : Desarrollo	Puntos estimados: 0.2
Fecha inicio: 17/04/09	Fecha fin: 18/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El usuario procederá a introducir datos tales como: usuario y contraseña y seleccionará el botón enviar.	

Anexo 5: Tarea de Programación: “Comprobar datos”

Tarea de programación	
Tarea Nro. 3	Número Historia (Nro. 4): Autenticar usuario
Nombre tarea: Comprobar datos	
Tipo de tarea : Desarrollo	Puntos estimados: 0.3
Fecha inicio: 18/04/09	Fecha fin: 19/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: Una vez que el usuario haya enviado sus datos el sistema procederá a comprobar los mismos. En caso que haya un error se mostrará un mensaje relacionado con el mismo, “Error en los datos introducidos”. En caso contrario se mostrará un cartel de bienvenida, “Bienvenido a SGIG”.	

Anexo 1: Tarea de Programación: “Mostrar interfaz buscar paciente”

Tarea de programación	
Tarea Nro. 1	Número Historia (Nro. 5): Introducir datos de un paciente.
Nombre tarea: Mostrar interfaz buscar paciente	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 07/04/09	Fecha fin: 07/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El usuario seleccionará la opción “Buscar paciente” e inmediatamente se visualizará la interfaz correspondiente.	

Anexo 6: Tarea de Programación: “Realizar búsqueda”

Tarea de programación	
Tarea Nro. 2	Número Historia (Nro. 5): Introducir datos de un paciente.
Nombre tarea: Realizar búsqueda	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 07/04/09	Fecha fin: 07/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: Una vez que se muestra la interfaz buscar paciente, se introducirá el número de carné de identidad del mismo y se selecciona “Buscar”.	

Anexo 7: Tarea de Programación: “Comprobar datos”

Tarea de programación	
Tarea Nro. 3	Número Historia (Nro. 5): Introducir datos de un paciente.
Nombre tarea: Comprobar datos	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 07/04/09	Fecha fin: 07/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El sistema comprobará el número de identidad introducido, si lo encuentra lanzará un mensaje de error, “El paciente ya existe”. En caso contrario mostrará la interfaz correspondiente a la introducción de los datos.	

Anexo 8: Tarea de Programación: “Insertar datos”

Tarea de programación	
Tarea Nro. 4	Número Historia (Nro. 5): Introducir datos de un paciente.
Nombre tarea: Insertar datos	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 07/04/09	Fecha fin: 07/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: Una vez que se muestra la interfaz correspondiente se procede a la inserción de los datos de la paciente. Seguidamente se seleccionará “Enviar”. Si no ocurre errores el sistema mostrará el mensaje Los datos de la paciente fueron enviados satisfactoriamente”. En caso contrario se visualizará un mensaje de error, “Envió fallido”.	

Anexo 9: Tarea de Programación: “Mostrar interfaz buscar paciente”

Tarea de programación	
Tarea Nro. 1	Número Historia (Nro. 10): Crear hoja de Cargo
Nombre tarea: Mostrar interfaz crear hoja de Cargo	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 09/04/09	Fecha fin: 09/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El usuario seleccionará la opción crear hoja de Cargo e inmediatamente se visualizará la interfaz correspondiente. Deberá introducir la fecha en la que fue creada.	

Anexo 10: Tarea de Programación: “Mostrar interfaz crear hoja de cargo”

Tarea de programación	
Tarea Nro. 2	Número Historia (Nro10): Crear hoja de Cargo
Nombre tarea: Mostrar interfaz crear hoja de cargo	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 09/04/09	Fecha fin: 09/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El usuario seleccionará la opción Crear hoja de cargo e inmediatamente se visualizará la interfaz correspondiente.	

Anexo 11: Tarea de Programación: “Crear hoja de cargo”

Tarea de programación	
Tarea Nro: 3	Número Historia (Nro 10): Crear hoja de Cargo
Nombre tarea: Crear hoja de cargo	
Tipo de tarea : Desarrollo	Puntos estimados: 0.2
Fecha inicio: 9/04/09	Fecha fin: 10/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El usuario luego de la comprobación de los datos debe insertar los datos de la misma. Una vez introducido los datos se selecciona la opción Enviar. En caso que se introduzcan los datos correctamente se mostrará un mensaje informativo “Se ha creado la historia satisfactoriamente”, en caso contrario se mostrará “Intento fallido”.	

Anexo 12: Tarea de Programación: “Mostrar interfaz buscar paciente”

Tarea de programación	
Tarea Nro. 1	Número Historia (Nro13): Crear remisión
Nombre tarea: Mostrar interfaz buscar paciente	

Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 11/04/09	Fecha fin: 11/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El usuario seleccionará la opción buscar paciente e inmediatamente se visualizará la interfaz correspondiente. Deberá introducir el número de carné de identidad. En caso que no se encuentre el sistema lanzará el mensaje “El paciente no existe en la base datos”. En caso contrario selecciona la opción crear remisión.	

Anexo 13: Tarea de Programación: “Mostrar interfaz crear remisión”

Tarea de programación	
Tarea Nro. 2	Número Historia (Nro. 13): Crear remisión
Nombre tarea: Mostrar interfaz crear remisión	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 11/04/09	Fecha fin: 11/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El usuario seleccionará la opción Crear remisión e inmediatamente se visualizará la interfaz correspondiente.	

Anexo 14: Tarea de Programación: “Crear remisión”

Tarea de programación	
Tarea Nro. 3	Número Historia (Nro. 13): Crear remisión
Nombre tarea: Crear remisión	
Tipo de tarea : Desarrollo	Puntos estimados: 0.1
Fecha inicio: 11/04/09	Fecha fin: 11/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	

Descripción: El usuario introducirá los datos y seleccionará la opción Enviar. Si los datos fueron ingresados correctamente el sistema lanzará el mensaje “La remisión fue creada satisfactoriamente”. En caso contrario se mostrará “Envió fallido”.

Anexo 15: Tarea de Programación: “Mostrar interfaz buscar paciente”

Tarea de programación	
Tarea Nro. 1	Número Historia (Nro. 15): Crear historia clínica
Nombre tarea: Mostrar interfaz buscar paciente	
Tipo de tarea : Desarrollo	Puntos estimados: 0.1
Fecha inicio: 12/04/09	Fecha fin: 12/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El usuario seleccionará la opción crear historia clínica e inmediatamente se visualizará la interfaz buscar paciente. Posteriormente deberá introducir el número de carné de identidad. En caso que no se encuentre el sistema lanzará el mensaje “El paciente no existe en la base datos”. En caso contrario selecciona la opción crear historia clínica.	

Anexo 16: Tarea de Programación: “Mostrar interfaz crear historia clínica”

Tarea de programación	
Tarea Nro. 2	Número Historia (Nro. 15): Crear historia clínica
Nombre tarea: Mostrar interfaz crear historia clínica	
Tipo de tarea : Desarrollo	Puntos estimados: 0.1
Fecha inicio: 12/04/09	Fecha fin: 12/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El usuario seleccionará la opción crear historia clínica e inmediatamente se visualizará la interfaz correspondiente.	

Anexo 17: Tarea de Programación: “Crear historia clínica”

Tarea de programación	
Tarea Nro. 3	Número Historia (Nro. 15): Crear historia clínica
Nombre tarea: Crear historia clínica	
Tipo de tarea : Desarrollo	Puntos estimados: 0.2
Fecha inicio: 13/04/09	Fecha fin: 13/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El usuario introducirá los datos y seleccionará la opción Enviar. Si los datos fueron ingresados correctamente el sistema lanzará el mensaje “La historia clínica fue creada satisfactoriamente”. En caso contrario se mostrará “Envió fallido”.	

Iteración 2

Anexo 18: Tarea de Programación: “Mostrar interfaz buscar perfil”

Tarea de programación	
Tarea Nro. 1	Número Historia (Nro. 3): Ver perfil de un usuario
Nombre tarea: Mostrar interfaz buscar perfil.	
Tipo de tarea : Desarrollo	Puntos estimados: 0.1
Fecha inicio: 15/04/09	Fecha fin: 15/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: Una vez que el usuario se loguea selecciona la opción “Buscar Perfil” y se visualizará la interfaz correspondiente.	

Anexo 19: Tarea de Programación: “Realizar búsqueda”

Tarea de programación	
Tarea Nro. 2	Número Historia (Nro. 3): Ver perfil de un usuario
Nombre tarea: Realizar búsqueda	
Tipo de tarea : Desarrollo	Puntos estimados: 0.5
Fecha inicio: 15/04/09	Fecha fin: 17/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: Para realizar la búsqueda se insertará el nombre del usuario que se desea mostrar (este es único). En caso que no se encuentre el sistema lanzará un mensaje de error. En caso contrario se mostrará el perfil buscado.	

Anexo 20: Tarea de Programación: “Mostrar interfaz buscar paciente”

Tarea de programación	
Tarea Nro. 1	Número Historia (Nro. 8): Buscar datos personales de un paciente
Nombre tarea: Mostrar interfaz buscar paciente	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 18/04/09	Fecha fin: 18/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El usuario seleccionará la opción “Buscar paciente” e inmediatamente se visualizará la interfaz correspondiente.	

Anexo 21: Tarea de Programación: “Realizar búsqueda”

Tarea de programación	
Tarea Nro. 2	Número Historia (Nro. 8): Buscar datos personales de un paciente

Nombre tarea: Realizar búsqueda	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 18/04/09	Fecha fin: 18/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: Una vez que se muestra la interfaz buscar paciente, se introducirá el número de carné de identidad del mismo y se selecciona “Buscar”.	

Anexo 22: Tarea de Programación: “Comprobar datos”

Tarea de programación	
Tarea Nro. 3	Número Historia (Nro. 8): Buscar datos personales de un paciente
Nombre tarea: Comprobar datos	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 18/04/09	Fecha fin: 18/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El sistema comprobará el número de identidad introducido, si no lo encuentra lanzará un mensaje de error, “El paciente no existe”.	

Anexo 23: Tarea de Programación: “Mostrar datos”

Tarea de programación	
Tarea Nro. 4	Número Historia (Nro. 8): Buscar datos personales de un paciente
Nombre tarea: Mostrar datos	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 18/04/09	Fecha fin: 18/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	

Descripción: Una vez realizada la búsqueda se mostrará la interfaz con los datos de la paciente.

Anexo 24: Tarea de Programación: “Mostrar interfaz generar reportes”

Tarea de programación	
Tarea Nro. 1	Número Historia (Nro. 9): Generar reporte estadístico
Nombre tarea: Mostrar interfaz generar reportes	
Tipo de tarea : Desarrollo	Puntos estimados: 0.1
Fecha inicio: 20/04/09	Fecha fin: 20/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El usuario seleccionará la opción generar reportes estadístico e inmediatamente se visualizará la interfaz correspondiente.	

Anexo 25: Tarea de Programación: “Generar reporte”

Tarea de programación	
Tarea Nro. 2	Número Historia (Nro. 9): Generar reporte estadístico
Nombre tarea: Generar reporte	
Tipo de tarea : Desarrollo	Puntos estimados: 0.1
Fecha inicio: 20/04/09	Fecha fin: 20/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El usuario introducirá la fecha de la hoja de cargo y seleccionará la opción buscar. En caso que no se encuentren los datos en la base de datos se mostrará el mensaje, “No hay hojas de cargo de esta fecha”; en caso contrario se visualizarán los reportes correspondientes.	

Anexo 26: Tarea de Programación: “Mostrar interfaz buscar hoja de cargo”

Tarea de programación	
Tarea Nro. 1	Número Historia (Nro. 12): Buscar Hoja de Cargo
Nombre tarea: Mostrar interfaz buscar hoja de cargo	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 23/04/09	Fecha fin: 23/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El usuario seleccionará la opción buscar hoja de cargo e inmediatamente se visualizará la interfaz correspondiente.	

Anexo 27: Tarea de Programación: “Realizar búsqueda”

Tarea de programación	
Tarea Nro. 2	Número Historia (Nro. 12): Buscar Hoja de Cargo
Nombre tarea: Realizar búsqueda	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 23/04/09	Fecha fin: 23/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: Una vez que se muestra la interfaz buscar hoja de cargo, se introducirá la fecha y el nombre de la consulta a la que se hizo la remisión, se selecciona la opción “Buscar”.	

Anexo 28: Tarea de Programación: “Comprobar datos”

Tarea de programación	
Tarea Nro. 3	Número Historia (Nro. 12): Buscar Hoja de Cargo

Nombre tarea: Comprobar datos	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 23/04/09	Fecha fin: 23/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El sistema comprobará si la fecha y el nombre de la consulta introducidos se encuentran en la base de datos, si no lo encuentra lanzará un mensaje de error, “La hoja de cargo no existe”.	

Anexo 29: Tarea de Programación: “Mostrar datos”

Tarea de programación	
Tarea Nro. 4	Número Historia (Nro. 12): Buscar Hoja de Cargo
Nombre tarea: Mostrar datos	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 23/04/09	Fecha fin: 23/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: Una vez realizada la búsqueda se mostrará la interfaz con los datos pertenecientes a la hoja de cargo.	

Anexo 30: Tarea de Programación: “Mostrar interfaz buscar paciente”

Tarea de programación	
Tarea Nro. 1	Número Historia (Nro. 6): Modificar datos de un paciente.
Nombre tarea: Mostrar interfaz buscar paciente	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 24/04/09	Fecha fin: 24/04/09

Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO
Descripción: El usuario seleccionará la opción “Buscar paciente” e inmediatamente se visualizará la interfaz correspondiente.

Anexo 31: Tarea de Programación: “Realizar búsqueda”

Tarea de programación	
Tarea Nro. 2	Número Historia (Nro. 6): Modificar datos de un paciente.
Nombre tarea: Realizar búsqueda	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 24/04/09	Fecha fin: 24/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: Una vez que se muestra la interfaz buscar paciente, se introducirá el número de carné de identidad del mismo y se selecciona “Buscar”.	

Anexo 32: Tarea de Programación: “Comprobar datos”

Tarea de programación	
Tarea Nro. 3	Número Historia (Nro. 6): Modificar datos de un paciente.
Nombre tarea: Comprobar datos	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 24/04/09	Fecha fin: 24/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El sistema comprobará el número de identidad introducido, si lo encuentra lanzará un mensaje de error, “El paciente ya existe”. En caso contrario mostrará la interfaz correspondiente a la introducción de los datos.	

Anexo 33: Tarea de Programación: “Modificar datos”

Tarea de programación	
Tarea Nro. 4	Número Historia (Nro. 6): Modificar datos de un paciente.
Nombre tarea: Modificar datos	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 24/04/09	Fecha fin: 24/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: Una vez que se muestra la interfaz correspondiente se procede a la modificación de los datos de la paciente. Seguidamente se seleccionará “Enviar cambio”. Si no ocurre errores el sistema mostrará el mensaje “Datos enviado correctamente”, en caso contrario se visualizará un mensaje de error, “Envío fallido”.	

Anexo 34: Tarea de Programación: “Buscar historia clínica”

Tarea de programación	
Tarea Nro. 1	Número Historia (Nro. 16): Actualizar historia clínica
Nombre tarea: Buscar historia clínica	
Tipo de tarea : Desarrollo	Puntos estimados: 0.2
Fecha inicio: 25/04/09	Fecha fin: 25/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El usuario seleccionará Buscar historia clínica, se muestra la interfaz correspondiente e introduce el número del carné de identidad que desea buscar. Si no se encuentra el sistema mostrará el mensaje “No existe la historia clínica”. En caso contrario se visualizará la historia clínica correspondiente al número del carné de identidad introducido.	

Anexo 35: Tarea de Programación: “Modificar datos”

Tarea de programación	
Tarea Nro. 2	Número Historia (Nro. 16): Actualizar historia clínica
Nombre tarea: Modificar datos	
Tipo de tarea : Desarrollo	Puntos estimados: 0.2
Fecha inicio: 26/04/09	Fecha fin: 26/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El usuario modificará los datos necesarios y seleccionará la opción guardar modificación.	

Anexo 36: Tarea de Programación: “Mostrar interfaz buscar remisión”

Tarea de programación	
Tarea Nro. 1	Número Historia (Nro. 14): Buscar remisión
Nombre tarea: Mostrar interfaz buscar remisión	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 27/04/09	Fecha fin: 27/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El usuario seleccionará la opción Buscar remisión e inmediatamente se visualizará la interfaz correspondiente.	

Anexo 37: Tarea de Programación: “Realizar búsqueda”

Tarea de programación	
Tarea Nro. 2	Número Historia (Nro. 14): Buscar remisión
Nombre tarea: Realizar búsqueda	

Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 27/04/09	Fecha fin: 27/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: Una vez que se muestra la interfaz buscar remisión, se introducirá la fecha y el nombre de la consulta a la que se hizo la remisión, se selecciona la opción “Buscar”.	

Anexo 38: Tarea de Programación: “Comprobar datos”

Tarea de programación	
Tarea Nro. 3	Número Historia (Nro. 14): Buscar remisión
Nombre tarea: Comprobar datos	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 27/04/09	Fecha fin: 27/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El sistema comprobará si la fecha y el nombre de la consulta introducidos se encuentran en la base de datos, si no lo encuentra lanzará un mensaje de error, “La remisión no existe”.	

Anexo 39: Tarea de Programación: “Mostrar datos”

Tarea de programación	
Tarea Nro. 4	Número Historia (Nro. 14): Buscar remisión
Nombre tarea: Mostrar datos	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 27/04/09	Fecha fin: 27/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: Una vez realizada la búsqueda se mostrará la interfaz con los datos pertenecientes a la remisión.	

Anexo 40: Tarea de Programación: “Mostrar interfaz buscar paciente”

Tarea de programación	
Tarea Nro. 1	Número Historia (Nro. 18): Buscar historia clínica
Nombre tarea: Mostrar interfaz buscar paciente	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 28/04/09	Fecha fin: 28/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El usuario seleccionará la opción “Buscar paciente” e inmediatamente se visualizará la interfaz correspondiente.	

Anexo 41: Tarea de Programación: “Realizar búsqueda”

Tarea de programación	
Tarea Nro. 2	Número Historia (Nro. 18): Buscar historia clínica
Nombre tarea: Realizar búsqueda	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 28/04/09	Fecha fin: 28/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: Una vez que se muestra la interfaz buscar paciente, se introducirá el número de carné de identidad del mismo y se selecciona la opción “Buscar”.	

Anexo 42: Tarea de Programación: “Comprobar datos”

Tarea de programación	
Tarea Nro. 3	Número Historia (Nro. 18): Buscar historia clínica

Nombre tarea: Comprobar datos	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 28/04/09	Fecha fin: 28/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El sistema comprobará el número de carné de identidad introducido, si no lo encuentra lanzará un mensaje de error, “El paciente no existe”.	

Anexo 43: Tarea de Programación: “Mostrar datos”

Tarea de programación	
Tarea Nro. 4	Número Historia (Nro. 18): Buscar historia clínica
Nombre tarea: Mostrar datos	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 28/04/09	Fecha fin: 28/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: Una vez realizada la búsqueda se mostrará la interfaz con los datos de la paciente.	

Iteración 3

Anexo 44: Tarea de Programación: “Mostrar interfaz buscar perfil”

Tarea de programación	
Tarea Nro. 1	Número Historia (Nro. 2): Eliminar perfil de usuario.
Nombre tarea: Mostrar interfaz buscar perfil.	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05

Fecha inicio: 29/04/09	Fecha fin: 29/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: Una vez que el usuario se loguea selecciona la opción “Buscar Perfil” e inmediatamente se visualizará la interfaz correspondiente.	

Anexo 45: Tarea de Programación: “Realizar búsqueda”

Tarea de programación	
Tarea Nro. 2	Número Historia (Nro. 2): Eliminar perfil de usuario.
Nombre tarea: Realizar búsqueda.	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 29/04/09	Fecha fin: 29/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: Para realizar la búsqueda se insertará el nombre del usuario (este es único). En caso que no se encuentre el sistema lanzará un mensaje de error, “El usuario no existe”.	

Anexo 46: Tarea de Programación: “Comprobar datos”

Tarea de programación	
Tarea Nro. 3	Número Historia (Nro. 2): Eliminar perfil de usuario.
Nombre tarea: Comprobar datos	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 29/04/09	Fecha fin: 29/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El sistema comprobará el nombre de usuario introducido, si no lo encuentra lanzará un mensaje de error, “El perfil no existe”. En caso contrario mostrará la interfaz correspondiente al perfil.	

Anexo 47: Tarea de Programación: “Eliminar perfil”

Tarea de programación	
Tarea Nro. 4	Número Historia (Nro. 2): Eliminar perfil de usuario.
Nombre tarea: Eliminar perfil	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 29/04/09	Fecha fin: 29/04/09
Programador responsable: Yanisvel Suárez Arbolay – Yamisleydis Borrero LaO	
Descripción: Una vez que el usuario realiza la búsqueda seleccionará la opción “Eliminar Perfil”. En caso que no haya habido ningún error el sistema mostrara un mensaje “Perfil eliminado satisfactoriamente”. En caso contrario mostrará un mensaje de error “Eliminación fallida”.	

Anexo 48: Tarea de Programación: “Mostrar interfaz buscar paciente”

Tarea de programación	
Tarea Nro: 1	Número Historia (Nro 7): Eliminar datos de un paciente.
Nombre tarea: Mostrar interfaz buscar paciente	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 30/04/09	Fecha fin: 30/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El usuario seleccionará la opción “Buscar paciente” e inmediatamente se visualizará la interfaz correspondiente.	

Anexo 49: Tarea de Programación: “Realizar búsqueda”

Tarea de programación	
Tarea Nro. 2	Número Historia (Nro. 7) Eliminar datos de un paciente.

Nombre tarea: Realizar búsqueda	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 30/04/09	Fecha fin: 30/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: Una vez que se muestra la interfaz buscar paciente, se introducirá el número de carné de identidad del mismo y se selecciona “Buscar”.	

Anexo 50: Tarea de Programación: “Comprobar datos”

Tarea de programación	
Tarea Nro. 3	Número Historia (Nro. 7): Eliminar datos de un paciente.
Nombre tarea: Comprobar datos	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 30/04/09	Fecha fin: 30/04/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El sistema comprobará el número de identidad introducido, si lo encuentra lanzará un mensaje de error, “El paciente ya existe”. En caso contrario mostrará la interfaz con los datos de la paciente.	

Anexo 51: Tarea de Programación: “Eliminar datos”

Tarea de programación	
Tarea Nro. 4	Número Historia (Nro. 7): Eliminar datos de un paciente.
Nombre tarea: Eliminar datos	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 30/04/09	Fecha fin: 30/04/09

Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO
Descripción: Una vez que se muestra la interfaz con los datos se selecciona “Eliminar paciente”. Si ocurre algún error el sistema mostrará el mensaje “Eliminación fallida”, en caso contrario mostrará “Eliminación satisfactoria”.

Anexo 52: Tarea de Programación: “Buscar hoja de cargo”

Tarea de programación	
Tarea Nro. 1	Número Historia (Nro. 11): Modificar Hoja de Cargo
Nombre tarea: Buscar hoja de cargo	
Tipo de tarea : Desarrollo	Puntos estimados: 0.2
Fecha inicio: 01/05/09	Fecha fin: 01/05/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El usuario seleccionará Buscar hoja de cargo, se muestra la interfaz correspondiente e introduce la fecha de la hoja que desea buscar. Si no se encuentra el sistema mostrará el mensaje “No existe la hoja de cargo”. En caso contrario se visualizará la hoja de cargo correspondiente a la fecha introducida.	

Anexo 53: Tarea de Programación: “Modificar datos”

Tarea de programación	
Tarea Nro. 2	Número Historia (Nro. 11): Modificar Hoja de Cargo
Nombre tarea: Modificar datos	
Tipo de tarea : Desarrollo	Puntos estimados: 0.2
Fecha inicio: 02/05/09	Fecha fin: 02/05/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El usuario modificará los datos necesarios y seleccionará la opción guardar modificación.	

Anexo 54: Tarea de Programación: “Mostrar interfaz buscar paciente”

Tarea de programación	
Tarea Nro. 1	Número Historia (Nro. 17): Eliminar historia clínica
Nombre tarea: Mostrar interfaz buscar paciente	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 03/05/09	Fecha fin: 03/05/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El usuario seleccionará la opción “Buscar paciente” e inmediatamente se visualizará la interfaz correspondiente.	

Anexo 55: Tarea de Programación: “Realizar búsqueda”

Tarea de programación	
Tarea Nro. 2	Número Historia (Nro. 17): Eliminar historia clínica
Nombre tarea: Realizar búsqueda	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 03/05/09	Fecha fin: 03/05/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: Una vez que se muestra la interfaz buscar paciente, se introducirá el número de carné de identidad del mismo y se seleccionará la opción “Buscar”.	

Anexo 56: Tarea de Programación: “Comprobar datos”

Tarea de programación	
Tarea Nro. 3	Número Historia (Nro. 17): Eliminar historia clínica

Nombre tarea: Comprobar datos	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 03/05/09	Fecha fin: 03/05/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: El sistema comprobará el número de identidad introducido, si lo encuentra lanzará un mensaje de error, "El paciente ya existe". En caso contrario mostrará la interfaz con los datos de la paciente.	

Anexo 57: Tarea de Programación: "Eliminar datos"

Tarea de programación	
Tarea Nro. 4	Número Historia (Nro. 17): Eliminar historia clínica
Nombre tarea: Eliminar datos	
Tipo de tarea : Desarrollo	Puntos estimados: 0.05
Fecha inicio: 03/05/09	Fecha fin: 03/05/09
Programador responsable: Yanisvel Suárez Arbolay– Yamisleydis Borrero LaO	
Descripción: Una vez que se muestra la interfaz con los datos se selecciona "Eliminar paciente". Si ocurre algún error el sistema mostrará el mensaje "Eliminación fallida", en caso contrario mostrará "Eliminación satisfactoria".	