

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD # 8



“CONTINUACIÓN DEL SISTEMA DE GESTIÓN DE INFORMACIÓN PARA EL ÁREA DE GINECOLOGÍA (SIGIGO) DEL POLICLÍNICO-HOSPITAL ERNESTO CHE GUEVÁRA DE LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS”.

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS

Autores: Patricia Caulineau Castro
Luis Felipe Díaz Barrios

Tutor: Ing. Ariel Orta Hernández

Co-Tutor: Ing. Abduly Díaz García

Ciudad de La Habana, 2010.

Año 52 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 _____.

Patricia Cauleau Castro

Firma del Autor

Luis Felipe Díaz Barrios

Firma del Autor

Ariel Orta Hernández

Firma del Tutor

Abduly Díaz García

Firma del Co-Tutor

“Todos y cada uno de nosotros paga puntualmente su cuota de sacrificio consciente de recibir el premio en la satisfacción del deber cumplido, conscientes de avanzar con todos hacia el Hombre Nuevo que se vislumbra en el horizonte. “

A handwritten signature in black ink, appearing to be the name 'J. L.' or similar, written in a cursive style.

DATOS DE CONTACTO

Ing. 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.

Ing. Abduly Díaz García -> abduly@softel.cu Graduado de Ingeniería Informática en el año 2007 en la Universidad de las Ciencias Informáticas (UCI). Actualmente se desempeña como especialista en Informática en la empresa Softel. Posee la categoría de tercerizado.

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.

AGRADECIMIENTOS

A nuestro Comandante en Jefe y a la Revolución por brindarnos la oportunidad de estudiar en esta Universidad y alcanzar nuestro sueño.

A la Universidad de las Ciencias Informáticas por formarnos como profesionales.

A nuestros tutores Ing. Ariel Orta Hernández e Ing. Abduly Díaz García por su paciencia y dedicación.

A todos los profesores que nos guiaron y brindaron los conocimientos necesarios para nuestra formación como Ingenieros Informáticos.

A todas aquellas personas que pusieron un granito de arena en el desarrollo de este trabajo.

De Patricia:

A las personas más importantes de mi vida, mis padres por ser los mejores padres del mundo y estar ahí siempre que los necesité, por darme su apoyo y estar conmigo en todo momento y sobre todo por darme su confianza. Sin ustedes nada de esto hubiese sido posible.

A mis abuelos que con su vida y ejemplo forjaron el amor y los valores que ahora forman parte de mí y continúan guiándome día a día enorgulleciéndose de mis triunfos, gracias por ser tan especiales.

A mi novio por todas sus enseñanzas, por su amor, por aguantarme en los peores momentos, por ser parte de mi vida.

A mis tíos por brindarme tanto amor desde pequeña y estar siempre a mi lado ayudándome en todo lo que estuviera a su alcance.

A mis primos que son como mis hermanos, por compartir conmigo los momentos más lindos de mi vida.

A toda mi familia por la preocupación constante y el apoyo a lo largo de mi vida.

A mis amigas Annia, Lianny, Daily y Danay por ser personas tan especiales y estar siempre a mi lado cuando las necesito.

A mi compañero de tesis, ante todo por su paciencia y dedicación, por siempre brindarme las fuerzas necesarias en estos momentos.

A mis compañeros de aula por siempre brindarme su ayuda y compartir conmigo buenos momentos.

A todos muchas gracias.

De Luis Felipe:

A mis padres por su amor, su apoyo y su confianza.

A mi hermana por creer y apostar por mí.

A mis abuelos por su preocupación.

A mi novia por estar presente en cada momento de mi vida.

A mis compañeros de aula por apoyarme en estos cinco años.

A Reinier Santander, Alberto Menéndez y a los profesores Hector Pérez, Rolando Fernández, Edier García, Arcel Labrada y Marlon Lugo por su ayuda incondicional.

DEDICATORIA

...A mis padres y a mis abuelos por todo el apoyo, la confianza y el amor que siempre me han dado. Sin su ejemplo no hubiera podido lograr lo que soy hoy.

Patricia

... A mi mamá que siempre ha estado pendiente de mí regalándome paciencia y amor, a mi papá que ha sabido darme el consejo oportuno en el momento preciso, a mi hermanita por estar siempre a mi lado, a mis abuelos, a toda mi familia por apoyarme y confiar en mí, a mi novia que siempre ha estado en los momentos difíciles brindándome su ayuda y el soporte espiritual de todas las cosas que he podido lograr.

Luis Felipe

RESUMEN

El policlínico-hospital “Ernesto Che Guevara” de la Universidad de las Ciencias Informáticas requiere de la continuación del Sistema de Gestión de Información para el área de Ginecología, debido a que en la actualidad la aplicación implementada presenta problemas de usabilidad, funcionabilidad y mantenibilidad. Por lo antes mencionado la presente investigación se centró en la continuación de la aplicación web para lograr un mejor funcionamiento del sistema actualmente desplegado (SIGIGO). Para el desarrollo del sistema, se siguieron los pasos que propone la metodología Extreme Programming (XP), en sus distintas fases: exploración, planificación, implementación y pruebas.

Como resultado se obtuvo un producto funcional que satisface las necesidades del cliente, además de la documentación obtenida a lo largo de la investigación.

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	6
1.1 INTRODUCCIÓN.....	6
1.2 SISTEMAS SIMILARES VINCULADOS AL CAMPO DE ACCIÓN.....	6
1.2.1 <i>Ejemplos de productos fuera de Cuba</i>	7
1.3 FRAMEWORK.....	15
1.3.1 <i>Cakephp</i>	16
1.3.2 <i>Zend Framework</i>	17
1.3.3 <i>CodeIgniter</i>	18
1.3.4 <i>Symfony</i>	19
1.3.5 <i>Fundamentación del framework a utilizar</i>	21
1.4 POSTGRESQL VERSIÓN 8.4.....	22
1.4.1 <i>Fundamentación del sistema gestor de base de datos a utilizar</i>	23
1.5 FUNDAMENTACIÓN DE LA SOLUCIÓN PROPUESTA	23
1.6 CONCLUSIONES.....	23
CAPÍTULO 2: PROPUESTA DEL SISTEMA.....	25
2.1 INTRODUCCIÓN.....	25
2.2 PROCESOS PRESENTES EN EL NEGOCIO	25
2.2.1 <i>Descripción de los procesos presentes en el negocio</i>	25
2.3 DESCRIPCIÓN DE LOS PROCESOS QUE SERÁN OBJETO DE AUTOMATIZACIÓN EN EL SISTEMA.....	26
2.5 REQUISITOS NO FUNCIONALES	29
2.6 ARQUITECTURA DE LA INFORMACIÓN.....	30

2.6.1 Solución de la arquitectura de información de SIGIGO.....	33
2.7 PATRONES DE ARQUITECTURA.....	34
2.7.1 Patrón Modelo Vista Controlador (MVC)	34
2.8 PATRONES DE DISEÑO	36
2.8.1 Patrones GOF que implementa Symfony	36
2.8.1.1 Creacionales	36
2.8.1.2 Estructurales	37
2.8.1.3 Comportamiento.....	37
2.9 ESTÁNDARES	38
2.9.1 Estándar de codificación para la Base de Datos	38
2.9.2 Estándar de codificación para PHP	38
2.10 PROPUESTA DEL SISTEMA.....	44
2.11 CONCLUSIONES.....	45
CAPÍTULO 3: DESARROLLO DE LA SOLUCIÓN.....	46
3.1 INTRODUCCIÓN.....	46
3.2 FASE I: EXPLORACIÓN	46
3.2.1 Historias de Usuario	46
3.3 FASE II: PLANIFICACIÓN DE LA ENTREGA	49
3.3.1 Estimación de esfuerzos por historias de usuario	49
3.3.2 Plan de iteraciones	50
3.3.3 Plan de duración de las iteraciones.....	51
3.3.4 Plan de entregas.....	52
3.4 CONCLUSIONES.....	54

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA	55
4.1 INTRODUCCIÓN.....	55
4.2 DISEÑO DEL SISTEMA	55
4.2.1 <i>Tarjetas Clase, Responsabilidad y Colaboración (CRC)</i>	55
4.3 DISEÑO DE LA BASE DE DATOS	58
4.4 FASE DE IMPLEMENTACIÓN	60
4.4.1 <i>Tareas de programación por iteraciones</i>	60
4.4.1.1 Iteración 1	60
4.4.1.2 Iteración 2	62
4.4.1.3 Iteración 3	63
4.5 FASE DE PRUEBAS	64
4.5.1 <i>Pruebas de aceptación</i>	64
4.6 CONCLUSIONES.....	76
CONCLUSIONES	77
RECOMENDACIONES:	78
REFERENCIAS BIBLIOGRÁFICAS	79
BIBLIOGRAFÍA	81
GLOSARIO DE TÉRMINOS	82
ANEXOS	85
HISTORIAS DE USUARIO	85
TAREAS DE PROGRAMACIÓN	89
<i>Iteración 1</i>	89
<i>Iteración 2</i>	97

Iteración 3..... 107

PRUEBAS DE ACEPTACIÓN..... 111

ÍNDICE DE FIGURAS

Figura 1: Módulo Historias Clínicas	8
Figura 2: Módulo de Controles Obstétricos	9
Figura 3: Módulo administrativo-financiero-contable.	10
Figura 4: Protocolo de Ginecología.	11
Figura 5: Protocolo de Obstetricia.	11
Figura 6: Historia Clínica.	12
Figura 7: Agenda de citas.	12
Figura 8: Creación de facturas.	13
Figura 9: Proceso Principal	28

ÍNDICE DE TABLAS

Tabla 1: La implementación del MVC que realiza Symfony.	35
Tabla 2: Representación de una Historia de Usuario.	46
Tabla 3: Representación de la Historia de Usuario Nro. 1.....	47
Tabla 4: Representación de la Historia de Usuario Nro. 2.....	47
Tabla 5: Representación de la Historia de Usuario Nro. 3.....	48
Tabla 6: Representación de la Historia de Usuario Nro. 4.....	49
Tabla 14: Estimación de esfuerzos por Historia de Usuario.	50
Tabla 15: Plan de duración de las iteraciones.	52
Tabla 16: Funcionalidades por módulos.	52
Tabla 17: Plan de duración de entrega.	53
Tabla 18: Tarjeta CRC Clase Configuración	55
Tabla 19: Tarjeta CRC Clase Medicamentos.....	56
Tabla 20: Tarjeta CRC Clase Tratamiento	56
Tabla 21: Tarjeta CRC Clase Complementarios	57
Tabla 22: Tarjeta CRC Clase Notificaciones.....	57
Tabla 23: Tarjeta CRC Clase Calendario	58
Tabla 24: Tiempo real de implementación por módulo (iteración 1)	60
Tabla 25: Tarea de programación por historias de usuario (iteración 1)	61

Tabla 26: Tiempo real de implementación por módulo (iteración 2)	62
Tabla 27: Tarea de programación por historias de usuario (iteración 2)	62
Tabla 28: Tiempo real de implementación por módulo (iteración 3)	63
Tabla 29: Tarea de programación por historias de usuario (iteración 3)	64
Tabla 30: Prueba 1 al Módulo Configuración (iteración 1).....	65
Tabla 31: Prueba 2 al Módulo Configuración (iteración 1).....	65
Tabla 32: Prueba 3 al Módulo Configuración (iteración 1).....	66
Tabla 33: Prueba 4 al Módulo Configuración (iteración 1).....	67
Tabla 34: Prueba 5 al Módulo Configuración (iteración 1).....	67
Tabla 35: Prueba 6 al Módulo Configuración (iteración 1).....	68
Tabla 36: Prueba 7 al Módulo Configuración (iteración 1).....	69
Tabla 37: Prueba 8 al Módulo Configuración (iteración 1).....	69
Tabla 38: Prueba 9 al Módulo Configuración (iteración 1).....	70
Tabla 39: Prueba 10 al Módulo Configuración (iteración 1).....	71
Tabla 40: Prueba 11 al Módulo Configuración (iteración 1).....	72
Tabla 41: Prueba 12 al Módulo Configuración (iteración 1).....	72
Tabla 42: Prueba 13 al Módulo Configuración (iteración 1).....	73
Tabla 43: Prueba 14 al Módulo Configuración (iteración 1).....	74
Tabla 44: Prueba 15 al Módulo Configuración (iteración 1).....	74

Tabla 45: Prueba 16 al Módulo Configuración (iteración 1).....75

ÍNDICE DE ANEXOS

Anexo 1: Representación de la Historia de Usuario Nro. 5.	85
Anexo 2: Representación de la Historia de Usuario Nro. 6.	85
Anexo 3: Representación de la Historia de Usuario Nro. 7.	86
Anexo 4: Representación de la Historia de Usuario Nro. 8.	86
Anexo 5: Representación de la Historia de Usuario Nro. 9.	87
Anexo 6: Representación de la Historia de Usuario Nro. 10.	87
Anexo 7: Representación de la Historia de Usuario Nro. 11.	88
Anexo 8: Tarea de Programación: "Mostrar interfaz Gestionar campo"	89
Anexo 9: Tarea de Programación: "Crear campo"	89
Anexo 10: Tarea de Programación: "Modificar campo"	90
Anexo 11: Tarea de Programación: "Desactivar campo"	91
Anexo 12: Tarea de Programación: "Mostrar interfaz Gestionar formato"	91
Anexo 13: Tarea de Programación: "Crear formato"	92
Anexo 14: Tarea de Programación: "Mostrar interfaz Gestionar formularios médicos"	92
Anexo 15: Tarea de Programación: "Crear registro en la tabla nomencladora"	93
Anexo 16: Tarea de Programación: "Modificar registro en la tabla nomencladora"	94
Anexo 17: Tarea de Programación: "Mostrar interfaz Gestionar tabla nomencladora"	94
Anexo 18: Tarea de Programación: "Crear registro en la lista de valores"	95

Anexo 19: Tarea de Programación: "Modificar registro en la lista de valores"	96
Anexo 20: Tarea de Programación: "Eliminar registro en la lista de valores"	96
Anexo 21: Tarea de Programación: "Mostrar interfaz Gestionar pedido de medicamentos"	97
Anexo 22: Tarea de Programación: "Crear solicitud de medicamentos"	97
Anexo 23: Tarea de Programación: "Modificar solicitud de medicamentos"	98
Anexo 24: Tarea de Programación: "Cancelar solicitud de medicamentos"	99
Anexo 25: Tarea de Programación: "Mostrar interfaz Gestionar tratamiento"	99
Anexo 26: Tarea de Programación: "Asignar tratamiento"	100
Anexo 27: Tarea de Programación: "Registrar cambio de tratamiento"	101
Anexo 28: Tarea de Programación: "Mostrar interfaz Gestionar complementarios"	101
Anexo 29: Tarea de Programación: "Crear orden de complementarios"	102
Anexo 30: Tarea de Programación: "Modificar orden de complementarios"	102
Anexo 31: Tarea de Programación: "Eliminar orden de complementarios"	103
Anexo 32: Tarea de Programación: "Mostrar interfaz Gestionar resultados de complementarios"	104
Anexo 33: Tarea de Programación: "Registrar resultados de complementarios"	104
Anexo 34: Tarea de Programación: "Modificar resultados de complementarios"	105
Anexo 35: Tarea de Programación: "Eliminar resultados de complementarios"	106
Anexo 36: Tarea de Programación: "Mostrar interfaz Gestionar calendario médico"	106
Anexo 37: Tarea de Programación: " Consultar remisiones con fecha menor que la actual"	107

Anexo 38: Tarea de Programación: "Mostrar interfaz Gestionar datos de decano"	107
Anexo 39: Tarea de Programación: "Crear estructura organizativa"	108
Anexo 40: Tarea de Programación: "Modificar estructura organizativa"	108
Anexo 41: Tarea de Programación: "Eliminar estructura organizativa"	109
Anexo 42: Tarea de Programación: "Consultar resultados de ultrasonido"	110
Anexo 43: Tarea de Programación: "Consultar tipo de paciente"	110
Anexo 44: Tarea de Programación: "Enviar notificación"	111
Anexo 45: Prueba 17 al Módulo Medicamentos (iteración 2)	111
Anexo 46: Prueba 18 al Módulo Medicamentos (iteración 2)	112
Anexo 47: Prueba 19 al Módulo Medicamentos (iteración 2)	113
Anexo 48: Prueba 20 al Módulo Tratamiento (iteración 2)	113
Anexo 49: Prueba 21 al Módulo Tratamiento (iteración 2)	114
Anexo 50: Prueba 22 al Módulo Tratamiento (iteración 2)	115
Anexo 51: Prueba 23 al Módulo Tratamiento (iteración 2)	115
Anexo 52: Prueba 24 al Módulo Complementarios (iteración 2)	116
Anexo 53: Prueba 25 al Módulo Complementarios (iteración 2)	117
Anexo 54: Prueba 26 al Módulo Complementarios (iteración 2)	118
Anexo 55: Prueba 27 al Módulo Complementarios (iteración 2)	118
Anexo 56: Prueba 28 al Módulo Complementarios (iteración 2)	119

Anexo 57. Prueba 29 al Módulo Complementarios (iteración 2).....	120
Anexo 58: Prueba 30 al Módulo Complementarios (iteración 2).....	120
Anexo 59: Prueba 31 al Módulo Complementarios (iteración 2).....	121
Anexo 60: Prueba 32 al Módulo Complementarios (iteración 2).....	122
Anexo 61: Prueba 33 al Módulo Complementarios (iteración 2).....	123
Anexo 62: Prueba 34 al Módulo Calendario (iteración 2).....	123
Anexo 63: Prueba 35 al Módulo Configuración (iteración 3)	124
Anexo 64: Prueba 36 al Módulo Configuración (iteración 3)	125
Anexo 65: Prueba 37 al Módulo Configuración (iteración 3)	125
Anexo 66: Prueba 38 al Módulo Configuración (iteración 3)	126
Anexo 67: Prueba 39 al Módulo Configuración (iteración 3)	127
Anexo 68: Prueba 40 al Módulo Notificación (iteración 3)	127

Introducción

La Universidad de las Ciencias Informáticas cuenta desde septiembre del 2002 con el policlínico “Ernesto Che Guevara”, el cual surge con el objetivo de brindarle asistencia médica a la Misión Milagro y a los estudiantes de la Universidad. Comenzó brindando servicios como: Fisioterapia, Rayos X, Optometría, Farmacia, Laboratorio, Ultrasonido, ITS, Psicología, Esterilización, PAMI, Estomatología. Luego en abril del 2004 se le incorpora el servicio de Ginecología y Obstetricia el cual hasta ese momento se realizaba fuera de la Universidad.

El 19 de diciembre del 2006 se inaugura el policlínico-hospital “Ernesto Che Guevara”, el cual se ha categorizado de esta manera porque presta servicios que no son brindados por un policlínico común pero tampoco son todos los ofrecidos por los hospitales, estos servicios son: Traumatología, Podología, Óptica, Vacunación, Atención al viajero, Endoscopia, Psiquiatría, Salas de ingreso y a partir del 2009 el servicio Materno-Infantil, además de todas las interconsultas del hospital militar en el policlínico-hospital (Cirugía, Ortopedia, Dermatología, Clínico).

Actualmente los servicios de Ginecología y Obstetricia prestan atención a un poco menos de 10000 pacientes entre estudiantes y trabajadoras internas, lo que representa el 100% de la población femenina de la universidad.

Estas pacientes se encuentran distribuidas entre las consultas que conforman a dicho departamento: Regulación de la Fecundidad, Ginecología, Obstetricia (atención a embarazadas) y Regulación Menstrual, las que 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.

Con el objetivo de reducir los problemas que trae consigo el trabajo manual, se realizó en el curso 2008-2009 un trabajo investigativo que incluía un sistema a desplegar en las consultas mencionadas anteriormente. Dicho sistema (SIGIGO) automatizó los siguientes procesos: gestionar historias clínicas de los pacientes, gestionar remisiones, gestionar hojas de cargo, y la generación de reportes. En la actualidad la aplicación implementada presenta problemas de usabilidad, funcionabilidad y mantenibilidad. Usabilidad debido a que es difícil entender cómo realizar los procesos con la aplicación, no tiene una interfaz intuitiva y amigable, además de que presenta algunos errores en la arquitectura de la información definida; funcionabilidad debido a que actualmente los procesos han sufrido cambios y el sistema tal como

se encuentra no soporta estos, algunos procesos que se implementaron no funcionan correctamente; mantenibilidad debido a la poca organización del código, la no implementación de un módulo para la administración del sitio y la variedad de tecnologías utilizadas, ejemplo de esto son las distintas formas de usar la tecnología AJAX (librería JQuery o librerías implementadas a la medida por otros usuarios). Además, existen nuevos procesos que nunca se lograron implementar debido al tiempo con que se contaba y la poca experiencia de los tesistas en el desarrollo de un software. Dichos procesos se realizan actualmente de la manera tradicional, lo que trae consigo la presencia de los riesgos identificados en el trabajo anterior: pérdida de información al deteriorarse o extraviarse la planilla, Historia Clínica o papel con los resultados de análisis o remisiones de los pacientes.

Otro elemento importante a tener en cuenta es la poca comunicación que existe entre varios departamentos del policlínico-hospital (Laboratorio Clínico, Farmacia, Psicología, Estomatología, Ultrasonido), consultorios y decanos de cada facultad. Lo que trae consigo demoras en el proceso de atención al paciente y gastos en recursos materiales tales como papeles, provocando además agotamiento físico, al tener la enfermera o doctora que trasladarse a distintos lugares (facultades docentes) para realizar la notificación a los decanos sobre el estado de los pacientes.

Los pacientes que deben acudir a las consultas planificadas, en ocasiones no se presentan debido a olvidos o llegan tarde puesto que no tienen un calendario que los guíe, solo un papel que tiene el riesgo de perderse, además de que la información que deben conocer sobre los peligros del embarazo prematuro o no planificado, las enfermedades o sobre el mismo embarazo, por lo general se encuentra en el policlínico-hospital o áreas relacionadas, lo que puede aportar en alguna medida a que ocurran hechos de este tipo en el centro.

A partir de esta situación problemática se plantea como **problema científico**: ¿Cómo lograr un mejor funcionamiento del sistema SIGIGO desplegado en el policlínico-hospital “Ernesto Che Guevara”?

El **objetivo general** de la presente investigación es desarrollar una segunda versión del sistema SIGIGO para la eliminación de los problemas de usabilidad, funcionabilidad y mantenibilidad e inclusión de nuevos servicios en la aplicación actualmente desplegada en el policlínico-hospital “Ernesto Che Guevara”.

Para el desarrollo de la investigación se define como **objeto de estudio** los procesos de gestión de la información en el área de Ginecología.

Los **objetivos específicos** que se persiguen son:

- Desarrollar un módulo de administración del sistema.
- Mejorar las funcionalidades ya implementadas.
- Implementar nuevas funcionalidades al sistema de los procesos identificados.

Esto conlleva a que el **campo de acción** se encuentre enmarcado en los procesos de gestión de información en el área de Ginecología del policlínico-hospital “Ernesto Che Guevara”.

Para darle cumplimiento a los objetivos trazados se ha decidido desarrollar las siguientes **tareas**:

- Realizar entrevistas con los usuarios.
- Encontrar cambios en los procesos identificados anteriormente.
- Identificar nuevos procesos que hayan surgido en el transcurso del año.
- Detectar fallas en las funcionalidades implementadas en la primera versión.
- Elaborar una nueva propuesta de arquitectura de la información.
- Determinar las principales técnicas y tendencias de diseño de software para desarrollar sistemas similares.
- Identificar frameworks candidatos para implementar el sistema.
- Realizar análisis de factibilidad de la implementación del sistema apoyándose en un framework.
- Adaptar la arquitectura del producto al framework seleccionado.
- Seleccionar y describir las historias de usuario.
- Revisar el esquema de datos y describir el nuevo esquema que se proponga.
- Implementar los módulos que se incluirán en el sistema: módulo de administración, módulo de gestión de calendario del paciente, módulo para la integración con otras áreas del centro.

Idea a defender:

Con la presente investigación se pretende mejorar las características de usabilidad, funcionabilidad y mantenibilidad del sistema actualmente en uso, mejorando así la calidad de los servicios que se brindan en la consulta de Ginecología del policlínico-hospital “Ernesto Che Guevara”.

Métodos Científicos:**Métodos teóricos**

Analítico-sintético: para realizar una revisión bibliográfica de todos los temas asociados a la investigación y extraer los elementos más importantes de la información consultada.

Análisis histórico - lógico: para el análisis histórico de los procesos de gestión hospitalaria y el estudio de la trayectoria concreta de la aplicación implementada.

Métodos empíricos

Observación: utilizada para identificar las funcionalidades de los sistemas de gestión hospitalaria existentes hoy en día y poder definir los requisitos funcionales y no funcionales del sistema a implementar.

Entrevista: a especialistas del área de Ginecología, Estadísticas, Estomatología y Psicología para conocer cómo se comportan cualitativamente los procesos en el policlínico-hospital “Ernesto Che Guevara” y en consultorios médicos, obteniendo así una mejor comprensión de los problemas presentados en la aplicación anterior.

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

Capítulo 1: Fundamentación teórica: Incluye un análisis de algunos sistemas existentes que poseen similitud con el que se desea implementar. Un estudio de las mejoras potenciales que ha sufrido el Sistema Gestor de Bases de Datos PostgreSQL y su última versión estable 8.4, así como los frameworks más utilizados en la actualidad con el fin de proponer el más adecuado para la solución del problema.

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 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, además una nueva propuesta de la arquitectura de información. Concluyendo con la propuesta del sistema a desarrollar.

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

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

1.1 Introducción

Cuando se realiza una nueva versión de un producto de software debe tenerse en cuenta qué parte funciona correctamente, las mejoras potenciales que se le pueden añadir, y las nuevas características que se podrían incluir para lograr una aplicación robusta, funcional, disponible y confiable, por lo que hay que basarse en el estudio realizado en las versiones anteriores, para establecer una línea de evolución del producto.

La realización de la segunda versión del Sistema de Gestión de Información Ginecobstétrico se basa en el estudio realizado en la primera versión e incluye una revisión bibliográfica de aplicaciones similares relacionadas al área de Ginecología, así como el estudio de algunos de los frameworks más usados en la actualidad, además de los elementos que en la etapa de explotación del producto han sufrido mejoras, como es el Sistema Gestor de Bases de Datos PostgreSQL y su última versión estable 8.4.

1.2 Sistemas similares vinculados al campo de acción

Con el auge de las nuevas tecnologías se ha hecho necesario el uso de la informática para automatizar y acelerar los procesos más importantes de cualquier institución, para evitar trabajos innecesarios y volúmenes de información repetitivos, permitiendo mejorar la calidad de los servicios. El sector de la salud no queda exento de ello, en la versión anterior se realizó un estudio de aplicaciones de gestión de información hospitalaria con gran aceptación por parte de la comunidad médica mundial.

Estos sistemas son de gran importancia para las instituciones sanitarias, pero también es innegable que cada centro consta de características individuales, por lo que en su mayoría se hace imprescindible realizar modificaciones a los productos ya existentes, teniendo en cuenta lo antes mencionado es que surge la necesidad de desarrollar un producto que se adapte completamente a la situación real del campo de acción de la investigación. A continuación se hará un estudio de algunos productos relacionados principalmente al área de Ginecología.

1.2.1 Ejemplos de productos fuera de Cuba

Empresa Interactive Solutions Development (ISD): es una empresa que lleva más de 10 años desarrollando sistemas informáticos, brindando servicios para diferentes áreas de la salud, diseño y mantenimiento de sitios web, alojamiento de sitios web en Internet, entre otros.

Producto Gyne-Soft: es un sistema informático íntegramente especializado en la administración de consultorios de Ginecología y Obstetricia.

Gyne-Soft posee los siguientes módulos:

- **Reserva de turnos:** Permite asentar con fecha y hora la cita de un paciente. Visualiza y administra la lista de espera. Asigna sobretornos. Emite un listado de control de turnos reservados, para avisar posible cancelación del día de atención.
- **Ficha médica:** Posee una ficha muy completa y parametrizada donde el profesional puede cargar datos de la Historia Clínica: exámenes realizados, antecedentes, enfermedades padecidas y comentarios u observaciones.
- **Facturación:** Permite tener un absoluto control de las órdenes de prestaciones o estudios recibidos y pagos efectuados por paciente, además de asentar los pagos recibidos por parte de los centros de salud o pacientes que vayan cancelando sus facturas.
- **Exámenes:** Brinda la opción de asentar los resultados de los tests, exámenes o prácticas que se le van realizando a los pacientes. (Isd, 2005)

Empresa BioMedical Developments (BMD): es una empresa con más de 15 años en el mercado, dedicada exclusivamente al desarrollo de software de Historias Clínicas y administración de consultorios médicos.

Producto BMD Gyb & Obs: es un software de gestión de consultorios y centros de atención ginecológicos que ofrece, además de las ventajas de un administrador de Historias Clínicas digitales, un set de herramientas, tablas y gráficas diseñadas especialmente para el médico ginecólogo.

Se distingue por los siguientes aspectos:

- **Ficha del paciente:** incluye datos personales, antecedentes personales, familiares, tocoginecológicos y obstétricos.
- **Ficha resumida:** de consultas y patologías relevantes.
- **Historia Clínica:** generada automáticamente, que el profesional personaliza adaptándola a su propia metodología y dinámica de trabajo.



Figura 1: Módulo Historias Clínicas.

- **Conexión:** de una computadora a un ecógrafo a través de una interfaz de muy fácil manejo. Podrá incorporar de esta manera a la Historia Clínica las imágenes producidas por un ecógrafo y contar con un archivo digital de imágenes de acceso inmediato. Comparar estudios ecográficos realizados en diferentes fechas y verificar así la evolución de patologías, o bien comparar imágenes de diferentes embarazos.
- **Planilla de control obstétrico:** con cálculo automático de edad gestacional y presentación de valores normales de las variables más importantes de seguimiento en el embarazo.

- **Gráficas:** automáticas de incremento de peso materno, altura del cuello uterino, peso fetal; comparación con embarazos anteriores; impresión de las curvas para entregar al paciente.

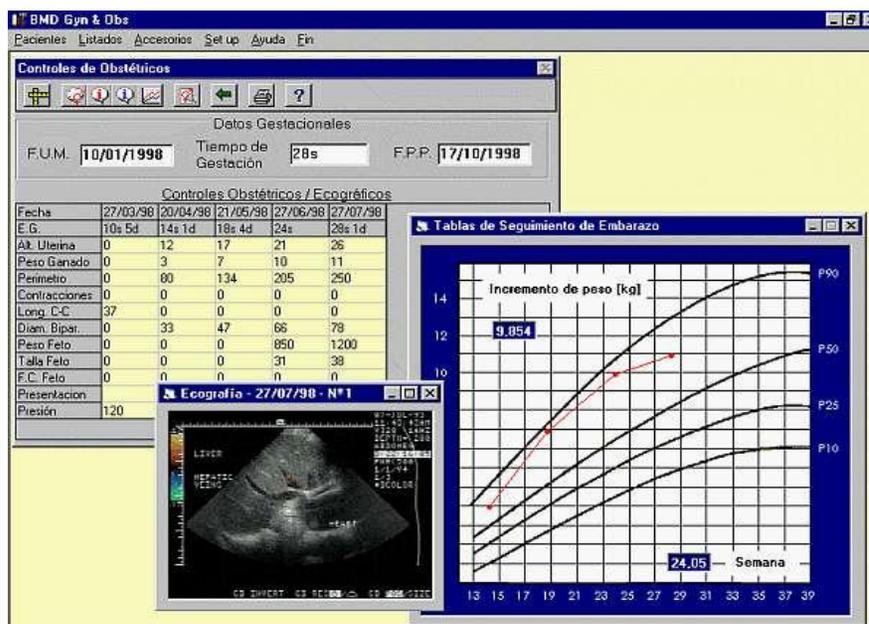


Figura 2: Módulo de Controles Obstétricos.

- **Módulo estadístico (opcional):** podrá realizar estudios estadísticos de su población particular.
- **Herramientas complementarias:** archivo de imágenes y videos explicativos personalizables, algoritmos de diagnóstico, mailing a pacientes, control de órdenes de estudios especiales, generación de recetas, vademécum personalizable, base de datos con indicaciones y explicaciones para el paciente, conexión directa a internet.
- **Agenda de turnos y agenda telefónica:** linkeada con la base de datos de pacientes, con horarios de atención y duración de turnos personalizables; carga de trabajo mensual y control de ausentismo; apto para trabajo en red (secretaria - médico/s) y actualización permanente de turnos del día por orden de llegada, control en tiempo real de demora promedio y cantidad de pacientes en sala de espera.
- **Archivo de estudios especiales:** imágenes radiográficas, ecográficas y tomográficas, audiometrías, exámenes de laboratorio, hemogramas, etc.
- **Módulos de adquisición de imágenes y video:** escaneado de imágenes y textos, y asociación directa y automática a la Historia Clínica del paciente.

- **Procesador de textos incorporado:** generación automática de órdenes, recetas y certificados.
- **Vademécum personalizable:** integrado a la consulta, recordando presentaciones comerciales.
- **Personalización de todos los módulos:** consultas (anamnesis, diagnósticos, tratamientos clínicos, quirúrgicos y exámenes complementarios), plan de vacunación, antropometría, pautas madurativas, alimentación.
- Manual de ayuda on-line interactivo.
- Back-up de seguridad diario automático.
- Módulo administrativo-financiero-contable. (BMD, 2005)



Figura 3: Módulo administrativo-financiero-contable.

Empresa Medigest Consultores: empresa líder en el ámbito de la informática médica, que desarrolla software especialmente diseñado para cubrir las demandas de todo tipo de servicios médicos y especialidades clínicas, tanto para consultas como para hospitales. Es una empresa de servicios con dedicación exclusiva al mundo de la medicina.

Producto Axón: software especializado en Ginecología y Obstetricia, se ha realizado bajo la supervisión y colaboración de prestigiosos miembros de la Sociedad Española de Ginecología y Obstetricia (SEGO), adaptándolo mejor a las necesidades de los ginecólogos.

El programa Axón permite llevar un seguimiento de la historia de las pacientes, y gestionar situaciones clínicas como el control del embarazo y parto, incluyendo apartados específicos para la ecografía, mamografía o para exploraciones del tracto inferior como citologías o biopsias.

Figura 4: Protocolo de Ginecología.

Figura 5: Protocolo de Obstetricia.

Entre los módulos más importantes se pueden mencionar:

Historia Clínica: Muestra la información más importante de su historia: alergias, lista de problemas, resumen de antecedentes, lista de visitas con su resumen. Desde ella también se puede acceder de manera muy ágil al resto de la información. Se ofrecen dos tipos de utilidades: una pantalla de visita general válida para todas las especialidades y otra para optar por trabajar con protocolos específicos para cada especialidad.

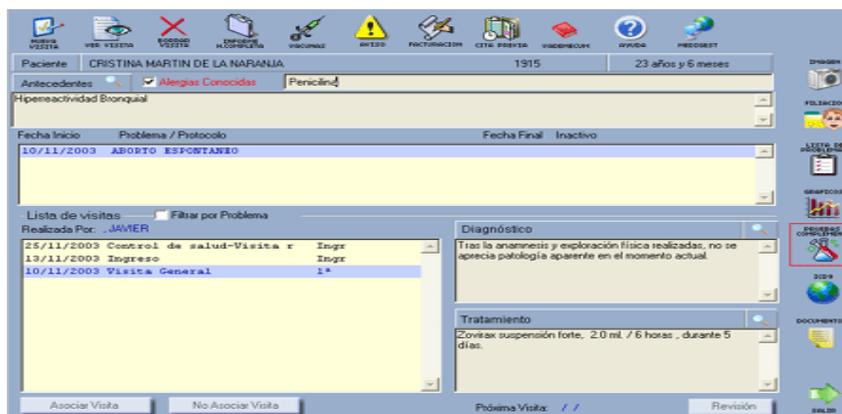


Figura 6: Historia Clínica.

- **Agenda de citaciones:** Permite la asignación automática de citas así como la duración de estas, la búsqueda automática de espacios libres, la reasignación automática de citas, además de la impresión de tickets, como recordatorios de citas.

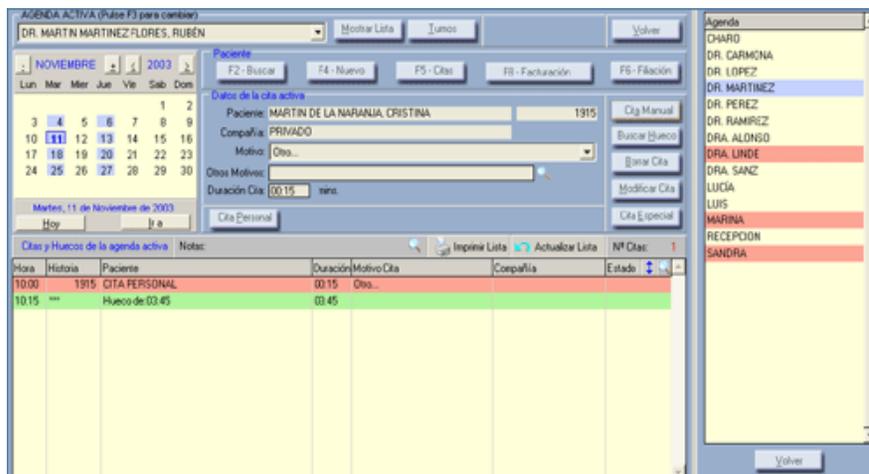


Figura 7: Agenda de citaciones.

- **Facturación:** Posibilita las facturaciones para distintos profesionales, la facturación automática a sociedades (mutuas médicas), el listado de cobros, pendiente de cobro y facturas emitidas, así como la facturación a través de distintas entidades o sociedades emisoras.

Fecha	Concepto	Cantidad	Precio	Fact
10/11/2003	Fisioterapia Consulta	1.0	22.00	Si
10/11/2003	AUX. 1/2 HORA LABORAL	1.0	6.76	Si
10/11/2003	Fisioterapia Neurológica	1.0	25.00	Si
10/11/2003	Rehabilitación Revisión	1.0	33.30	Si

Figura 8: Creación de facturas.

Pruebas complementarias que ofrece:

- Citologías y biopsias.
- Ecografía primer trimestre.
- Ecografía tercer trimestre.
- Ecografía gemelar.
- Ecografía ginecológica detallada.
- Ecografía semana 20.
- Estudio senológico.
- Histeroscopia. (Medigest, 2007)

Empresa Infonet Software: es una empresa de informática con más de 10 de años de experiencia en el desarrollo de software, ubicada en España.

Producto Mediconta: se trata de un programa o software médico para gestión de clínicas médicas. Este permite gestionar clínicas de pediatras, ginecólogos, fisioterapeutas, psicólogos y psiquiatras.

Mediconta está dividido en 6 módulos:

- **Funciones administrativas:** consta de un asistente de importación de datos: permite importar datos procedentes de Excel, creación de múltiples empresas: con los datos totalmente independientes, gestión de usuarios: según ley 15/1999 de Protección de Datos de Carácter Personal, copias de seguridad encriptadas, recuperación de base de datos, descargar actualizaciones automáticas por Internet, sistema de ayuda online: para poder ser asistido a través de Internet ante cualquier problema.
- **Gestión de pacientes:** permite introducir datos de los pacientes, historia médica, diagnósticos, datos del tratamiento, visitas médicas, informes, plantillas de informes, recetas, listados, facturas asociadas al tratamiento, permite además asociar una factura con las visitas que paga la misma.
- **Contabilidad (opcional):** permite llevar la contabilidad de la clínica de una forma transparente para el usuario. Realiza una generación automática de los asientos contables asociados a las facturas, generación de cuentas contables automáticas asociadas a clientes y proveedores. Incluye además diario, cierre ejercicio, balances, modelos de hacienda, listados.
- **Herramientas:** dispone de agenda, calculadora, listín telefónico.
- **Conexiones de red:** permite desde interconectar los equipos de una red cableada o wifi, hasta incluso (si así lo desea) conectar equipos ubicados en un lugar remoto a través de Internet (teletrabajo).
- **Digitalización de historias médicas:** Infonet Software mantiene una estrecha colaboración con Normadat, empresa especializada en servicios de digitalización y custodia de historias médicas. Mediconta es capaz de incorporar automáticamente en su base de datos todas las historias médicas digitalizadas por Normadat. (Infonet, 2005)

Los sistemas de gestión de información hospitalaria y en particular los relacionados con el área de Ginecología permiten lograr una mayor organización de todos los procesos clínicos que se llevan a cabo en las consultas. No obstante, el uso de ellos está estrechamente relacionado con el grado de aceptación

de los clientes y las particularidades de cada institución, por lo que se decide no utilizar ninguno de los productos estudiados, porque no se adaptan completamente a las necesidades reales del entorno de trabajo. Además, estos sistemas son aplicaciones desarrolladas por empresas privadas y se necesita una licencia para usarlos. Generalmente estas licencias no incluyen la modificación del código fuente. De manera general algunos de los sistemas que fueron objeto de estudio en la presente investigación poseen el módulo Facturación (Gyne-Soft, Axón), el cual no es necesario en el entorno de trabajo de la consulta de ginecología del policlínico-hospital “Ernesto Che Guevara”. Además, como el hospital se encuentra localizado en una zona donde la mayoría de las personas son internos, se puede aprovechar las capacidades de intranet para poder brindar información relevante a los pacientes y a su vez que estos puedan ver y gestionar su cronograma de consulta. Existen otras características propias del hospital que no implementan los sistemas antes mencionados como la relación con otros departamentos de la institución como Psicología, Laboratorio Clínico, entre otros.

1.3 Framework

Los framework proporcionan una estructura o marco para desarrollar proyectos. Simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Facilita la programación de aplicaciones, porque encapsula operaciones complejas en instrucciones sencillas. En resumen, poseen una serie de clases, funciones o librerías, que junto a convenciones comunes, se organizan bajo una estructura sobre la cual se desarrollan las aplicaciones. (François Zaninotto y Fabien Potencier, 2008)

Entre las ventajas de usar framework, se pueden contar:

1. Facilita integrar a otras personas a los proyectos pues comparten convenciones de desarrollo comunes.
2. No hay que preocuparse por mantener actualizadas las distintas partes, generalmente los frameworks son abiertos y soportados por una comunidad que actualizan la funcionalidad y corrigen los bugs de manera sostenida.
3. No reinventar la rueda, puesto que se aprovechan los componentes existentes aumentando la velocidad de desarrollo.

4. Reducción en el tiempo de desarrollo de nuevas aplicaciones.
5. Reducción del costo de mantenimiento.
6. Mayor nivel de confiabilidad (comparado con escribir código nuevo), en la medida que hay reuso y el framework se estabiliza.
7. Abstracción de URLs (Localizador de Recursos Uniforme) y sesiones, porque no es necesario manipular directamente las URLs ni las sesiones, el framework ya se encarga de hacerlo.
8. Fácil acceso a datos. Incluyen las herramientas e interfaces necesarias para integrarse con herramientas de acceso a datos, en bases de datos, XML, etc.
9. Autenticación y control de acceso, pues incluyen mecanismos para la identificación de usuarios mediante login y password que permiten restringir el acceso a determinadas páginas a determinados usuarios.

Como desventajas se pueden nombrar:

1. Agrega código adicional que no es elaborado por el programador de la aplicación.
2. Hay que invertir tiempo en aprender a usarlos.
3. En algunos casos una aplicación desarrollada con un framework puede ser más lenta (en cuanto a rendimiento) que una diseñada y desarrollada desde cero. (KICKBILL,2009)

1.3.1 Cakephp

Cakephp provee una base robusta para las aplicaciones web. Puede manejar cualquier aspecto, desde la solicitud inicial del usuario hasta el renderizado final de la página web. Además, como éste sigue los principios Modelo Vista Controlador (en adelante MVC), permite fácilmente personalizar y extender muchos aspectos de ésta.

El framework también suministra una estructura de organización básica, desde los nombres de los archivos hasta los de las tablas de la base de datos, manteniendo toda la aplicación consistente y lógica.

Cakephp permite:

1. Internacionalización y localización.
2. Validación de datos.

3. Limpieza de Datos.
4. Manejo de errores.
5. Depuración.
6. Caching.
7. Logging.
8. Testing.
9. Paginación.
10. Rest.

Está caracterizado por:

1. Compatible con PHP4 y PHP5.
2. CRUD de la base de datos integrado.
3. URLs amigables.
4. Sistema de plantillas rápido y flexible.
5. Ayudas para AJAX, Java script, HTML, forms.
6. Trabaja en cualquier subdirectorío del sitio.
7. Scaffolding (andamiaje) de las aplicaciones.
8. Listas de Control de Acceso.
9. Componentes de seguridad y sesión. (Cookbook, 2009)

1.3.2 Zend Framework

Zend Framework es un framework de código abierto y orientado a objetos para facilitar el desarrollo de aplicaciones web con PHP 5, la implementación del código es 100% orientada a objetos. A menudo, es considerado una biblioteca de componentes, debido a que estos poseen bajo acoplamiento entre sí, lo cual permite reutilizarlos con un alto grado de independencia. Proporciona además una sofisticada implementación del patrón MVC, el cual puede ser utilizado para fijar la estructura básica de las aplicaciones desarrolladas con el framework.

Entre sus principales características se pueden mencionar:

1. Trabaja con MVC.
2. Cuenta con módulos para manejar archivos PDF, canales RSS, Web Services (Amazon, Flickr, Yahoo).
3. El Marco de Zend también incluye objetos de las diferentes bases de datos, por lo que es extremadamente simple para consultar su base de datos, sin tener que escribir ninguna consulta SQL.

4. Una solución para el acceso a base de datos que balancea el Mapeo Relacional de Objetos (en adelante ORM) con eficiencia y simplicidad.
5. Completa documentación y tests de alta calidad.
6. Soporte avanzado para i18n (internacionalización).
7. Posee un buscador compatible con Lucene.
8. Clientes para servicios web, incluidos Google Data APIs y Strikelron.
9. Muchas otras clases útiles para hacerlo tan productivo como sea posible.
10. Abstracción de la base de datos fácil de usar.
11. Componentes que implementan formas de representación de formularios HTML.
12. Validación y filtrado de datos.
13. Servicios de autenticación y autorización de usuario. (Zend, 2009)

1.3.3 CodeIgniter

CodeIgniter contiene una serie de librerías que sirven para el desarrollo de aplicaciones web y además propone una manera de desarrollarlas que debemos seguir para obtener provecho de la aplicación. Este marca una manera específica de codificar las páginas web y clasificar sus diferentes scripts, que sirve para que el código esté organizado y sea más fácil de crear y mantener. CodeIgniter implementa el proceso de desarrollo llamado (MVC), que es un estándar de programación de aplicaciones, utilizado tanto para hacer sitios web como programas tradicionales.

Entre las características más importantes se pueden mencionar:

1. **Compatibilidad:** es compatible con la versión PHP 4, lo que hace que se pueda utilizar en cualquier servidor, incluso en algunos antiguos, también en PHP 5.
2. **Facilidad de instalación:** solo es necesaria una cuenta de FTP para subir CodeIgniter al servidor y su configuración se realiza con la edición de un archivo.

3. Flexibilidad: define una manera de trabajar específica, pero en muchos de los casos podemos seguirla o no, y sus reglas de codificación muchas veces se pueden saltar para trabajar más a gusto. Algunos módulos como el uso de plantillas son totalmente opcionales.
4. Ligereza: el núcleo de CodeIgniter es bastante ligero, lo que permite que el servidor no se sobrecargue interpretando o ejecutando grandes porciones de código.
5. Documentación tutorializada: la documentación de este es fácil de seguir y de asimilar, porque está escrita en modo de tutorial, aunque no facilita mucho la referencia rápida, cuando ya se sabe acerca del framework y se quiere consultar sobre una función o un método en concreto. (Álvarez, 2009)

1.3.4 Symfony

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

Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Es el framework más documentado del mundo, pues cuenta con miles de páginas de documentación distribuidas en varios libros gratuitos y decenas de tutoriales.

Symfony está caracterizado por:

1. Fácil de instalar y configurar en sistemas Windows, Mac y Linux.
2. Funciona con todas las bases de datos comunes (MySQL, PostgreSQL, Oracle, SQL Server de Microsoft).
3. Compatible solamente con PHP 5, para asegurar el mayor rendimiento y acceso a las características más avanzadas de PHP.
4. Basado en la premisa de *"convenir en vez de configurar"*, en la que el desarrollador solo debe configurar aquello que no es convencional.

5. Preparado para aplicaciones empresariales, porque se puede adaptar con facilidad a las políticas y arquitecturas propias de cada empresa u organización.
6. Flexible hasta cualquier límite y extensible mediante un completo mecanismo de plugins.
7. Publicado bajo licencia Massachusetts Institute of Technology (MIT) de software libre y apoyado por una empresa comprometida con su desarrollo.
8. Traducido a más de 40 idiomas y fácilmente traducible a cualquier otro idioma.

Symfony automatiza la mayoría de elementos comunes de los proyectos web, como por ejemplo:

- La capa de internacionalización que incluye Symfony permite la traducción de los datos y de la interfaz, así como la adaptación local de los contenidos.
- La capa de presentación utiliza plantillas y layouts que pueden ser creados por diseñadores HTML sin ningún tipo de conocimiento del framework. Los helpers incluidos permiten minimizar el código utilizado en la presentación, pues encapsulan grandes bloques de código en llamadas simples a funciones.
- Los formularios incluyen validación automatizada y relleno automático de datos ("repopulation"), lo que asegura la obtención de datos correctos y mejora la experiencia del usuario.
- Los datos incluyen mecanismos de escape que permiten una mejor protección contra los ataques producidos por datos corruptos.
- La gestión de la caché reduce el ancho de banda utilizado y la carga del servidor.
- La autenticación y la gestión de credenciales simplifican la creación de secciones restringidas y la gestión de la seguridad de usuario.
- El sistema de enrutamiento y las URL limpias permiten considerar a las direcciones de las páginas como parte de la interfaz, además de estar optimizadas para los buscadores.
- El soporte de e-mail incluido y la gestión de APIs permiten a las aplicaciones web interactuar más allá de los navegadores.
- Los listados son más fáciles de utilizar debido a la paginación automatizada, el filtrado y la ordenación de datos.

- Las interacciones con Ajax son muy fáciles de implementar mediante los helpers que permiten encapsular los efectos JavaScript compatibles con todos los navegadores en una única línea de código. (François Zaninotto y Fabien Potencier, 2008)

1.3.5 Fundamentación del framework a utilizar

Cuando se realiza el estudio de un conjunto de framework se debe tener en cuenta para una selección adecuada, su rendimiento, integración con otros componentes, extensibilidad ya sea a través de plugins o extensiones, pruebas unitarias, soporte, seguridad, documentación; es por eso, que teniendo en cuenta el campo de acción donde se pretende implementar la aplicación, se considera Symfony como el framework que más se adapta a las necesidades del equipo de desarrollo, porque permite realizar el diseño gráfico de la aplicación sin tener conocimiento previo del framework, lo que disminuye el tiempo de desarrollo al diseñar una interfaz amigable a la vez que se estudia a fondo las particularidades de éste, la gestión de los correos de notificaciones de estado de salud de los pacientes al personal autorizado (decanos) es sencillo, pues solo se centra alrededor de un objeto mailer que cuenta con un método `compose()` que contiene como parámetros el remitente, el destinatario, el asunto y el cuerpo del mensaje y un método `send()` que posee como argumento el mensaje a enviar. Cuando se quiere realizar una consulta que muestra grandes volúmenes de información se puede utilizar el método `doSelect()`, este genera una consulta del tipo `SELECT * FROM <<Tabla>>` a la base de datos el que unido a una clase llamada `sfDoctrinePager` realiza la paginación de los resultados a mostrar de manera automática. También posee un soporte nativo para pruebas unitarias y funcionales, lo que reducirá el tiempo de pruebas del producto SIGIGO. Maneja automáticamente las sesiones del usuario, siendo capaz de almacenar datos de forma persistente entre peticiones, utilizando el mecanismo de manejo de sesiones incluido en PHP y lo mejora para hacerlo más configurable y fácil de usar mediante la clase `sfUser`. Esta clase dispone de un contenedor de parámetros que guarda cualquier atributo del usuario en él, lo que permite que esta información esté disponible en otras peticiones hasta terminar la sesión del usuario. En Symfony, el modelo contiene la información necesaria para poder generar de forma automática el código de las operaciones de insertar, obtener, modificar y borrar (CRUD), de forma que se simplifica el desarrollo inicial de la parte de administración de las aplicaciones. Posee licencia de código abierto y es independiente del gestor de bases de datos, gracias a su capa de abstracción y es muy adecuado para metodologías ágiles de desarrollo como XP.

1.4 PostgreSQL versión 8.4

PostgreSQL es un sistema de gestión de base de datos objeto relacional y su código fuente está disponible libremente. Se encuentra publicado bajo la licencia Berkeley Software Distribution (BSD). PostgreSQL 8.4 es la última versión estable, liberada en el 2009. Tiene 293 mejoras relacionadas con la administración, programación y el lenguaje de consulta posibilitando que el trabajo con éste sea más fácil y productivo.

La versión 8.4 incluye nuevas características y mejoras. Entre ellas se pueden mencionar:

1. Funciones ventana.
2. Expresiones de tabla comunes y joins recursivos (Cláusula WITH RECURSIVE)].
3. Funciones con parámetros con valores por defecto y con número de parámetros variable.
4. Restauración en paralelo.
5. Permisos de usuario por columna.
6. Auto ajuste del mapa de espacio libre (FSM).
7. Soporte para certificados SSL de autenticación de usuario.
8. Estadísticas de ejecución por función.
9. Nuevos módulos contrib: pg_stat_statements, auto_explain, Citext, btree_gin.
10. Configuración de idioma y ordenamiento por base de datos.
11. Nuevas herramientas para monitorear las consultas.
12. Implementación de funcionalidades avanzadas de ANSI SQL: 2003.
13. Mejoras en procedimientos de almacenado.
14. Pg-migrator permite actualizaciones desde PostgreSQL 8.3 con un downtime mínimo.
15. Coincidencia parcial en búsqueda de texto. (Casanova, 2009)

1.4.1 Fundamentación del sistema gestor de base de datos a utilizar

Se decide utilizar PostgreSQL 8.4 pues este posee una expresión o subconsulta LIMIT que limita el número de filas devueltas por una consulta. Esto hace que sea más fácil la paginación de los datos de múltiples pantallas de páginas dinámicas. También los permisos a nivel de columnas permiten hacer más fácil la protección de los datos sensibles de la base de datos al conceder los permisos de SELECT y UPDATE a las columnas especificadas. Además, permite que los usuarios se autentiquen usando certificados SSL, lo que representa un esquema de autenticación mucho más flexible y seguro, vital para manejar información valiosa que debe mantenerse íntegra como son los resultados de los complementarios. PostgreSQL 8.4 dispone de la funcionalidad llamada disparador, que representa una acción definida en una tabla de la base de datos y ejecutada automáticamente por una función programada en cualquier lenguaje de procedimientos disponible y posibilita realizar actividades antes y después de realizar operaciones de INSERT, UPDATE y DELETE en las tablas. También esta versión mejora las tareas del optimizador de consultas al crear un plan de ejecución óptimo. Primero combina todas las posibles vías de barrer y cruzar las relaciones que aparecen en una consulta. Todas las rutas creadas conducen al mismo resultado y el optimizador estima el coste de ejecutar cada una de ellas para encontrar cual es la más económica, en el caso de PostgreSQL 8.4 usa tres estrategias.

1.5 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 haciendo uso del frameworks Symfony, manteniendo como metodología de desarrollo de software Extreme Programming (XP), como lenguajes del lado del servidor PHP y del lado del cliente JavaScript, HTML, CSS, como servidor web Apache y utilizando como sistema gestor de base de datos PostgreSQL 8.4.

1.6 Conclusiones

En el presente capítulo se realizó un estudio de algunos sistemas de gestión hospitalaria con mayor aceptación a nivel mundial, pero específicamente relacionados con el área de Ginecología, así como de los frameworks más usados en la actualidad y las mejoras potenciales que durante la explotación del producto sufrió el Sistema Gestor de Bases de Datos PostgreSQL en su última versión estable 8.4, con el

objetivo de establecer las nuevas características de la segunda versión de SIGIGO, trazando así las pautas para el desarrollo del sistema.

Capítulo 2: Propuesta del Sistema

2.1 Introducción

En el presente capítulo se elabora una nueva propuesta de la arquitectura de la información del sistema SIGIGO. Además, se realiza un estudio detallado de los patrones arquitectónicos y de diseños que implementa Symfony para el desarrollo de aplicaciones. También se continúa con la 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, especificando detalladamente los requerimientos no funcionales.

2.2 Procesos presentes en el negocio

2.2.1 Descripción de los procesos presentes en el negocio

1. Solicitar medicamentos: el proceso comienza una vez que la enfermera de la consulta o la ginecóloga realiza una solicitud de medicamentos a la farmacia, llevando un papel (Pedido de medicamentos) con los medicamentos que necesita, en la farmacia archivan ese documento y le entregan las pastillas. Involucra a la enfermera de la consulta, al farmacéutico y al documento Pedido de medicamentos. Es un posible proceso a automatizar.
2. Gestionar tratamiento paciente: el proceso comienza cuando la ginecóloga le indica un tratamiento a seguir a una paciente, en caso que éste no sea efectivo o produzca alguna reacción, la ginecóloga orienta otro tratamiento y registra por qué fue realizado el cambio de tratamiento. Presenta relación con Acudir a Consulta y Gestionar paciente (en cualquiera de las consultas). Involucra a la paciente, la ginecóloga y el documento Historia Clínica. Es un posible proceso a automatizar.
3. Crear orden de complementarios: el proceso comienza cuando la ginecóloga o el médico del consultorio procede a crear la orden de los complementarios que se debe realizar la paciente, en dependencia de su necesidad. Presenta relación con Acudir a Consulta, Gestionar Paciente (en cualquiera de las consultas). Involucra a la paciente, el médico del consultorio, la ginecóloga y el documento Indicación de complementarios. Es un posible proceso a automatizar.

4. Consultar resultados de complementarios: el proceso comienza cuando la ginecóloga o el médico del consultorio necesita examinar los resultados de los complementarios. El proceso presenta relación con Gestionar paciente (en cualquier consulta) y con Crear orden de complementarios. Involucra a la paciente, el médico del consultorio, la ginecóloga y el documento Indicación de complementarios. Es un posible proceso a automatizar.
5. Registrar resultados de complementarios: el proceso comienza cuando el técnico de laboratorio registra los resultados de los complementarios. Presenta relación con Crear orden de complementarios. Involucra al técnico de laboratorio y el documento Indicación de complementarios. Es un posible proceso a automatizar.
6. Notificar a decano: el proceso comienza cuando la ginecóloga le diagnostica a una paciente (estudiante) que está embarazada, inmediatamente se le informa al médico del consultorio y al decano de su facultad mediante una carta entregada por la paciente. Involucra a la paciente, a la ginecóloga y al documento Carta a los decanos. Es un posible proceso a automatizar.

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

1. Gestionar campo: el proceso comienza cuando la enfermera de la consulta o la ginecóloga se autentica. Una vez dentro de la aplicación, en la sección de configuración define un nuevo formato (determina cuales son los nuevos campos que debe tener el formato) para las Hojas de Cargo, Historias Clínicas y/o complementarios. Este proceso brinda las opciones de crear nuevos campos, modificar campos usados o desactivar campos.
2. Gestionar formato: el proceso comienza cuando la enfermera de la consulta o la ginecóloga se autentica. Una vez dentro de la aplicación, en la sección de configuración crea un nuevo formato. Un formato no es más que una estructura que especifica características determinadas de una entidad (Historia Clínica, Hoja de Cargo, etc).
3. Gestionar registro en la tabla nomencladora: el proceso comienza cuando la enfermera de la consulta o la ginecóloga se autentica. Una vez dentro de la aplicación, en la sección de configuración crea un nuevo registro en la tabla nomencladora, modifica un registro ya existente en tabla nomencladora. Una tabla nomencladora existe para clasificar elementos y esas clasificaciones son fijas o varían a largo

plazo. Estos registros son opciones de valores que los campos que se crean pueden tener, para los distintos formatos que maneja el sistema.

4. Gestionar registro en lista de valores: el proceso comienza cuando la enfermera de la consulta o la ginecóloga se autentica. Una vez dentro de la aplicación, en la sección de configuración crea un registro en la lista de valores, elimina o modifica un registro ya existente de la lista de valores. Estas listas de valores son los posibles valores que puede tener determinado campo en un formato dado.
5. Gestionar estructura (decanos facultad): el proceso comienza cuando la ginecóloga o el administrador del sistema se autentica. Una vez dentro de la aplicación por cambios ocurridos en las facultades se decide crear, eliminar o modificar la estructura organizativa de los decanos de cada facultad.
6. Gestionar solicitud de medicamentos: el proceso comienza cuando la enfermera de la consulta o la ginecóloga se autentica. Una vez dentro de la aplicación esta le brinda las opciones de crear, cancelar y modificar una solicitud de medicamentos.
7. Gestionar tratamiento paciente: el proceso comienza cuando la ginecóloga se autentica. Una vez dentro de la aplicación esta brinda las opciones de asignar tratamiento y registrar porque fue realizado un cambio de tratamiento en una paciente determinada.
8. Gestionar complementarios: el proceso comienza cuando la ginecóloga o el médico del consultorio se autentica. Una vez dentro de la aplicación esta brinda las opciones de crear, eliminar, modificar una orden de complementarios, además de consultar (buscar) los resultados de los complementarios.
9. Gestionar resultados de complementarios: el proceso comienza cuando el técnico de laboratorio se autentica. Una vez dentro de la aplicación esta brinda las opciones de registrar, modificar resultados de complementarios.
10. Notificar a decano: el proceso comienza cuando el resultado del ultrasonido es positivo para embarazo, entonces el sistema automáticamente realiza una búsqueda en la base de datos y comprueba si la paciente es estudiante, entonces envía un correo para el decano, el médico del consultorio y la ginecóloga.
11. Consultar calendario: el proceso comienza cuando las pacientes se autentican. Una vez dentro de la aplicación esta le brinda la opción de consultar (buscar) el calendario de sus citas planificadas.

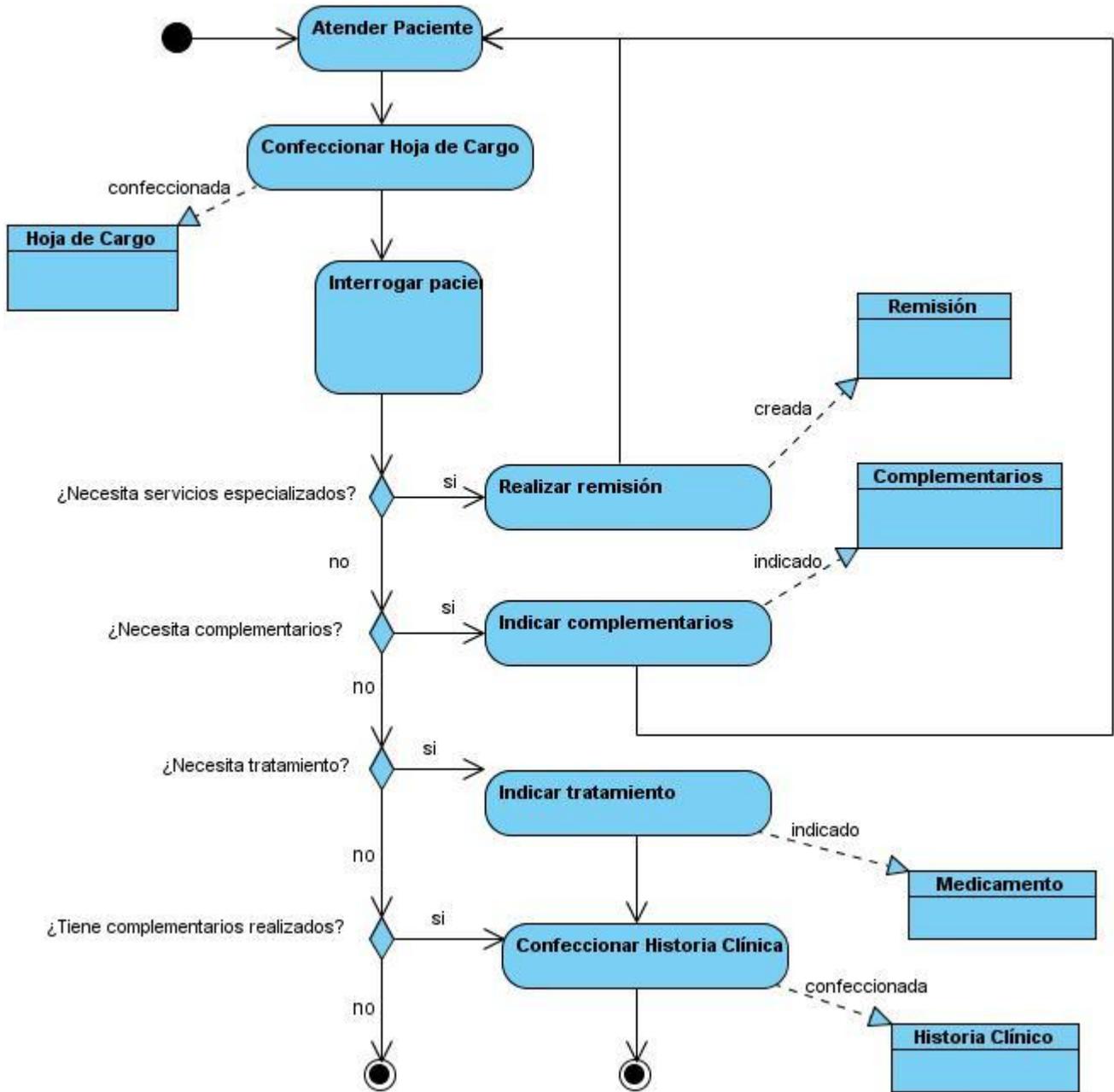


Figura 9: Proceso Principal

2.5 Requisitos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

Rendimiento

- Proporcionar tiempos de respuesta entre 5 y 10 segundos para los procesos en línea del sistema.
- Garantizar velocidad estable de navegación a nivel de aplicación.
- Optimizar la ejecución de procesos de sistema para disminuir la congestión de recursos.

Seguridad

- Confiabilidad: La información manejada por el sistema debe estar protegida de acceso no autorizado.
- Integridad: La información manejada por el sistema debe ser objeto de cuidadosa protección contra la corrupción y estados de inconsistencia.
- Disponibilidad: La aplicación deberá estar disponible en todo momento para aquellas personas con acceso a la información y los mecanismos utilizados para lograr la seguridad no deben ser un obstáculo a los usuarios para obtener los datos deseados en un momento dado.

Fiabilidad

- Garantizar capacidad para capturar excepciones.
- Prever contingencias para eventos de caída del sistema.

Capacidad

- Considerar características técnicas mínimas para la ejecución en clientes.
- Soportar múltiples conexiones al sistema de manera simultánea.
- Considerar el crecimiento esperado en el volumen de datos.

Funcionalidad

- Capacidad de búsqueda con un tiempo menor de 5 segundos. Mínima cantidad de páginas para ejecutar todas las funciones posibles, es decir, agrupar funciones afines en las mismas páginas.

2.6 Arquitectura de la Información

La Arquitectura de la Información es una disciplina que pretende articular un conjunto de técnicas para ayudar al desarrollo y producción de espacios de información (sitios web, aplicaciones multimedia y otros) y tiene como componentes: el sistema de organización y estructuración de la información, el sistema de etiquetado o rotulado y los sistemas de navegación y de búsqueda.

La Información puede organizarse mediante los esquemas de organización de la información (en adelante EOI) y estructuras de organización de la información (en adelante ESOI).

Los EOI delimitan las características compartidas de los elementos que integran el contenido y las influencias del agrupamiento lógico de estos. Pueden ser exactos o ambiguos.

Entre los EOI exactos se pueden mencionar:

- Los esquemas alfabéticos de organización de la información (en adelante EAOI) que organizan el contenido en orden alfabético y sirven como ayuda a otros esquemas, son muy utilizados en directorios y diccionarios.
- Los esquemas cronológicos de organización de la información (en adelante ECOI) que organizan el contenido en orden cronológico. El elemento asociativo e identificador es un dato del calendario.
- Los esquemas geográficos de organización de la información (en adelante EGOI) que organizan el contenido atendiendo a lugares y estos comprenden determinado significado.

Entre los EOI ambiguos, los más comunes son:

- Los que organizan la información según el asunto o la materia, que requiere de algún ordenamiento de las materias que posibilitan el acceso al contenido, así como el dominio de los contenidos que los usuarios esperan hallar dentro de las materias.
- Los que organizan la información atendiendo a determinados procesos o tareas, usados cuando es posible predecir con anticipación un limitado número de tareas de alta prioridad que los usuarios desarrollarán.
- Los que organizan la información según los intereses de la audiencia potencial, que permiten fragmentar el contenido en contenidos más pequeños según el tipo de usuario.

- Los que organizan la información a través de metáforas de la vida cotidiana, lo que permite crear relaciones entre elementos conocidos de la vida cotidiana y elementos novedosos y desconocidos y pueden ayudar a los usuarios a entender la distribución de los contenidos y las funcionalidades de una manera intuitiva.
- Los esquemas híbridos, no son más que la unión de 2 o más EOI ambiguos, pero al utilizar elementos de distintos EOI puede ocurrir cierta desorientación, por lo que se aconseja presentarlo por separado en distintas áreas, de esta manera, cada uno podrá cumplir su objetivo.

Las ESOI son intangibles pero muy importantes en el diseño de productos de información pues definen los caminos primarios sobre los cuales los usuarios podrán moverse, entre las más usadas se pueden mencionar:

- La jerarquía, que permite brindar una manera simple y familiar de organizar la información, con ella el usuario llega a sentirse cómodo, puede hacerse una idea de la estructura del producto sin necesidad de adentrarse en una extensa exploración de los contenidos.
- Los modelos orientados a base de datos, que puede definirse como una colección de registros donde cada registro contiene un número determinado de campos asociados.
- El hipertexto, que es una vía no lineal de estructurar la información, muy usado como complemento de otros ESOI.

Los Sistemas de Navegación se emplean para brindar un sentido del entorno y dar flexibilidad al movimiento dentro de la aplicación. Entre los tipos de navegación se pueden mencionar:

- Sistemas de navegación jerárquicos que permiten el acceso a los diferentes niveles jerárquicos inferiores a partir de la interfaz inicial.
- Sistemas de navegación globales, que complementan a uno jerárquico, permite la navegación horizontal y vertical de la aplicación.
- Sistemas de navegación locales, que complementan a los sistemas globales y son usados en aplicaciones complejas.
- Sistemas de navegación específicos, que utiliza principalmente cuando no es posible clasificar las relaciones entre las diferentes interfaces de la aplicación en algunos de los sistemas de navegación

anteriores. Ejemplo: enlaces dentro de un texto hacia otro documento relacionado, esté o no en la aplicación.

Los sistemas de navegación tienen como elementos integrados fundamentales:

- Las barras de navegación muy usadas en los sistemas de navegación jerárquicos, locales y globales.
- Los menús desplegables, que permiten el acceso a múltiples elementos de navegación de forma compacta.
- Los marcos que permiten definir uno o varios paneles desplazables dentro de una misma ventana del navegador.

Como elementos remotos:

- El mapa del sitio, que es una representación gráfica de la estructura de un sitio web.
- Tablas de contenidos, que muestran los primeros niveles dentro de la jerarquía y facilitan el acceso a segmentos del contenido o al contenido en su totalidad.
- Los índices, que presentan una lista de palabras clave o frases organizadas en orden alfabético que no constituyen una jerarquía.

Los sistemas de etiquetado permiten representar un conjunto de información. Las etiquetas describen o designan los elementos que integran el sistema de navegación, actualmente existen dos formas de crear etiquetas, textualmente o mediante íconos, aunque la combinación de ambas también es frecuente.

Los sistemas de etiquetado pueden dividirse en cuatro tipos:

- Etiquetas del sistema de navegación, que interactúan en un primer momento con el usuario.
- Etiquetas de sistemas de enlaces: que aparecen en el cuerpo de los párrafos y se enlazan con otros textos en función del contexto y su significado.
- Etiquetas del sistema de cabeceras o títulos: que se utilizan para titular los bloques de información. Hacen el papel de títulos o subtítulos, su significado está condicionado por el contexto.
- Etiquetas del sistema de indización: estas etiquetas son "invisibles" para el usuario, no obstante, tienen una función de suma importancia en la representación del contenido de las páginas para la identificación de estas en los motores de búsqueda.

El diseño de un sistema de búsqueda requiere de la consideración de numerosos aspectos que permitan determinar las razones que ameriten su implementación, es decir, cuando el contenido sea muy grande,

cuando buscar resulta más rápido que navegar, o la cantidad de información supera la funcionalidad de la estructura de la aplicación.

Las búsquedas pueden ser de diferentes tipos:

- De elemento conocido, cuando solo tiene una única y específica respuesta.
- De existencia, cuando el concepto está incorrectamente definido o es muy abstracto.
- De exploración, cuando no se tiene una pregunta sino una idea sobre la que se quiere aprender más.
- Global, cuando se requiere toda la información acerca de algo.

Debido a lo difícil que resulta implementar un motor de búsqueda, no por el algoritmo que se utilice sino por la dificultad del diseño de su interfaz, que sea capaz de satisfacer a los múltiples usuarios, es preferible solucionar errores en los sistemas de etiquetado y de navegación, que usar este para mitigar dichos errores.

Si se procede a usar un motor de búsqueda se debe aclarar cómo funciona el buscador, qué parámetros acepta y cómo modificar el alcance, así como proveer a este de una cierta retroalimentación, que le permita saber que obtuvo resultados de su búsqueda, lo que acelerará el aprendizaje asociativo y la búsqueda en sí.

Cuando se desarrolla la Arquitectura de Información de un producto, cualquiera que sea su tipo, se pretende obtener un resultado que sea accesible, usable y efectivo. (Bustamante, 2004)

2.6.1 Solución de la arquitectura de información de SIGIGO

El objetivo de la nueva propuesta de Arquitectura de la Información del Sistema de Gestión de Información Ginecobstétrico es facilitar la interacción entre el usuario y la aplicación así como agrupar, ordenar, etiquetar y organizar gráficamente los contenidos mostrados.

Después de un estudio riguroso y detallado de los principales aspectos que rigen la arquitectura de la información y teniendo en cuenta las mejores prácticas en la creación y diseño de aplicaciones web, concluimos que se organizará la información referente al Calendario Médico mediante el esquema cronológico (SOI exacto), para hacer más fácil el trabajo con el calendario que usarán los pacientes para guiar sus consultas y realizar sus exámenes médicos y el resultado de las búsquedas de pacientes mediante el esquema alfabético, para listar los datos de estos alfabéticamente (EOI exacto). Se organizará la información que tiene que ver con los procesos o actividades que los usuarios realizarán,

según sus privilegios mediante la organización de la información atendiendo a tareas (EOI ambiguo). Se usará la jerarquía como ESOI, para que el usuario siga pasos lógicos en la realización de las actividades o tareas a realizar, tratando siempre de atender la amplitud y profundidad de esta jerarquía, la cual completaremos con el uso de los hipertextos para darle una mayor flexibilidad, tratando siempre de no desorientar al usuario con la información consultada.

Se usarán los sistemas de navegación jerárquicos y globales, para permitir el acceso a la información de manera jerárquica; pero a la vez permitirá navegar de manera horizontal, para esto se hará uso de las barras de navegación y los menús desplegados y el mapa del sitio.

Los sistemas de etiquetado que se utilizarán son: etiquetas del sistema de navegación, etiquetas de sistemas de enlaces y etiquetas de sistema de cabecera.

Las búsquedas serán de elemento conocido, para buscar usuarios, pacientes, Historias Clínicas y otras informaciones, puesto que se sabe de antemano qué es lo que se desea buscar además el tamaño de la información a manejar no es demasiado grande ni variado y la navegación y el etiquetado permitirán encontrar la información que se desea sin dificultad.

Como las Historias Clínicas y Hojas de Cargo son unidades de contenido homogéneas; pueden ser descritas a través de un conjunto de metadatos comunes, por lo que basándose en ello, se puede organizar la aplicación mediante una clasificación facetada. Las clasificaciones facetadas se utilizan para organizar conjuntos de objetos (sitios web, productos, documentos), con una homogeneidad suficiente como para que puedan ser descritos por un número determinado de atributos o propiedades (facetas y categorías).

2.7 Patrones de Arquitectura

2.7.1 Patrón Modelo Vista Controlador (MVC)

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

El modelo: que representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.

La vista: que transforma el modelo en una página web que permite al usuario interactuar con ella.

El controlador: que se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones, además las capas del modelo, la vista y el controlador, se pueden subdividir en más capas para simplificar la programación.

La capa del modelo se divide en las subcapas de acceso a los datos y de abstracción de la base de datos. De esta forma, las funciones que acceden a los datos no utilizan sentencias ni consultas que dependen de una base de datos, sino que utilizan otras funciones para realizar las consultas. La capa de la vista también puede aprovechar la separación de código. Las páginas web suelen contener elementos que se muestran de forma idéntica a lo largo de toda la aplicación: cabeceras de la página, el layout genérico, el pie de página y la navegación global. Normalmente, sólo cambia el interior de la página. Por este motivo, la vista se separa en un layout y en una plantilla. El layout es global en toda la aplicación o al menos en un grupo de páginas. La plantilla sólo se encarga de visualizar las variables definidas en el controlador. El controlador se divide en un controlador frontal, que es único para cada aplicación, y las acciones, que incluyen el código específico del controlador de cada página, lo que permite crear un único punto de entrada a la aplicación y por ende el manejo de las peticiones del usuario y de la seguridad, además de cargar la configuración de la aplicación y otras tareas similares. (François Zaninotto y Fabien Potencier, 2008)

Tabla 1: La implementación del MVC que realiza Symfony.

Capas	Subcapas
Modelo	Abstracción de la base de datos Acceso a los datos
Vista	Vista Plantilla Layout

Controlador	Controlador frontal Acción
-------------	-------------------------------

2.8 Patrones de Diseño

Los patrones de diseño son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos, que se basan en la experiencia y que han demostrado que funcionan. Ofrecen un lenguaje estándar para reconocer, definir, y describir dichas soluciones en la creación de aplicaciones. El conocimiento de los patrones de diseño implementados por el framework Symfony es vital para entender el funcionamiento del mismo y por ende lograr objetivos trazados.

2.8.1 Patrones GOF que implementa Symfony

2.8.1.1 Creacionales

Los patrones creacionales tratan las formas de crear instancias de objetos. El objetivo de estos patrones es el de abstraer el proceso de instanciación y ocultar los detalles de cómo el objeto es creado o inicializado.

➤ **Singleton (Instancia única)**

El patrón Singleton garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. En el controlador frontal hay una llamada a `sfContext::getInstance()`. En una acción, el método `getContext()`, un objeto muy útil que guarda una referencia a todos los objetos del núcleo de Symfony.

➤ **Abstract Factory (Fábrica abstracta)**

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

2.8.1.2 Estructurales

Los patrones estructurales describen cómo las clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades. Estos objetos adicionales pueden ser incluso objetos simples u objetos compuestos.

➤ **Decorator (Envoltorio)**

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

➤ **Composite (Objeto compuesto)**

El patrón Composite permite tratar objetos compuestos como si de uno simple se tratase. Sirve para construir objetos complejos a partir de otros más simples y similares entre sí, gracias a la composición recursiva y a una estructura en forma de árbol. Esto simplifica el tratamiento de los objetos creados, pues al poseer todos ellos una interfaz común, se tratan todos de la misma manera. Symfony utiliza este patrón en su código fuente, como por ejemplo en la clase `composite.class.php` de `sfLog`.

2.8.1.3 Comportamiento

➤ **Observer**

El patrón Observer (Observador) define una dependencia del tipo uno-a-muchos entre objetos, de manera que cuando uno de los objetos cambia su estado, el observador se encarga de notificar este cambio a todos los otros dependientes. El objetivo de este patrón es desacoplar la clase de los objetos clientes del objeto, aumentando la modularidad del lenguaje.

Symfony proporciona un sistema de eventos inspirado en el patrón Observer para suplir las limitaciones actuales de PHP de que una clase no puede heredar de más de una clase. (François Zaninotto y Fabien Potencier, 2008)

2.9 Estándares

2.9.1 Estándar de codificación para la base de datos

1. El nombre de las tablas va antecedido por el prefijo tb_, excepto las que se crean producto del uso de plugins para Symfony.
Ejemplo: tb_area, tb_paciente, tb_campo, sf_guard_group, sf_guard_user.
2. Atributos con minúscula, lo más cercano a la descripción de la característica real del objeto.
Ejemplo: solapin, grupo, anno.
3. Atributos complejos de describir separados por _.
Ejemplo: motivo_cambio, fecha_nacimiento, id_unidad_medida.
4. Tablas complejas de describir separadas por _.
Ejemplo: tb_pedido_anticonceptivo, tb_tipo_formato, tb_unidad_medida.
5. Nombre de la base de datos antecedidos por el prefijo bd_.
Ejemplo: bd_ginecologia.
6. Cuando un atributo especifica un nombre se le añade el nombre de la tabla separado por _.
Ejemplo: nombre_area, nombre_paciente, nombre_decano.
7. Las tablas que no poseen id propio toman como identificador el prefijo id_ y a continuación el nombre de la tabla y las tablas que tengan id propio su identificador será el nombre del atributo.
Ejemplo: id_pedido, id_municipio, solapin.

2.9.2 Estándar de codificación para PHP

1. Indentación

La indentación debe ser a cuatro espacios sin caracteres de tabulación.

Ejemplo de indentación.

```
public function executeBuscar(sfWebRequest $request) {
    $this-> id_formato = $request-> getParameter('id');
    $lista_solapin = Doctrine::getTable('TbPaciente')-> getSolapines('doctrine');
    if ($this-> getRequest()-> getMethod() != sfRequest::POST) {
        return sfView::SUCCESS;
    }
}
```

```
    }  
}
```

2. Comentarios

El estilo de los comentarios debe ser (`/* */` ó `//`), no debe de utilizarse (`#`). Estos deben ser elaborados con un lenguaje formal y siempre teniendo en cuenta la ortografía, además deben de ser simples y directos.

Ejemplo de comentarios.

```
public function executeBuscar(sfWebRequest $request) {  
    $this-> id_formato = $request-> getParameter('id');  
    $lista_solapin = Doctrine::getTable('TbPaciente')-> getSolapines('doctrine');  
    /* Permite mostrar la interfaz si no se han enviado valores por el método  
    POST en el action*/  
    if ($this-> getRequest()-> getMethod() != sfRequest::POST) {  
        // Mostrar el formulario  
        return sfView::SUCCESS;  
    }  
}
```

3. Estructuras de Control

Deben utilizarse las llaves en cualquier caso “{ }”, incluso en situaciones donde son técnicamente opcionales. Su uso incrementa su capacidad de lectura y reduce la probabilidad de errores lógicos que son introducidos cuando líneas nuevas se agregan.

Estas incluyen: if, for, while, switch, etc.

Ejemplo de estructura “switch”.

PHP:

```
<?php  
switch(condition) {  
    case 1:
```

```
    action1;
    break;
    case 2:
    action2;
    break;
    default: defaultaction;
    break;
}
```

?>

Ejemplo de estructura "if".

PHP:

```
<?php
if((condition1) || (condition2)){
    action1;
}
else if((condition3) && (condition4)){
    action2;
}
else{
    defaultaction;
}
?>
```

4. Llamadas de Función

Las funciones deberán ser llamadas sin espacios entre el nombre de la función, el abrir paréntesis y el primer parámetro; y sin espacios entre el último parámetro y el cierre del paréntesis, al final punto y coma. Los parámetros se separarán por coma y sin espacios entre ellos.

Ejemplo:

PHP:

```
<?php
    $var=foo($bar,$baz,$quux);
?>
```

5. Definiciones de Función

En las declaraciones de función, el bracket de apertura empieza al final de la línea de definición.

Ejemplo:

PHP:

```
<?php
function foo_function($arg1,$arg2 = 'x'){
    if (condition){
        statement;
    }
    return $val;
}
?>
```

Los argumentos con valores por default van al final de la lista de argumentos. Siempre tratando de regresar un valor significativo de una función, si una es la apropiada.

Un ejemplo más largo:

PHP:

```
<?php
function connect(&$dsn,$persistent = false){
    if (is_array($dsn)){
        $dsninfo = &$dsn;
    }
    else {
        $dsninfo = DB::parseDSN($dsn);
    }
    if (!$dsninfo || !$dsninfo['phptype']){
        return $this->raiseError();
    }
    return true;
}
?>
```

6. Inclusión de Código

Cuando se incluye un archivo de dependencia incondicionalmente se debe utilizar **require_once** y cuando sea condicionalmente **include_once**.

7. Convención de Nombres de Variables, Funciones, Clases

Variables:

Estas deberán nombrarse con un prefijo de tres letras, el cual define el tipo de dato de la misma, seguido de un guión bajo y el nombre descriptivo de la variable.

Ejemplo:

PHP:

\$txt_descripcion: Campo tipo text

\$chr_nombres: Campo tipo char o varchar

\$int_cantidad: Campo tipo integer

\$flt_total: Campo tipo float

\$dbl_precio : Campo tipo double

\$dtm_fecha: Campo tipo fecha

\$bol_activo: Campo tipo boolean

Funciones:

Estas deberán nombrarse con palabras minúsculas seguido de un guión bajo, para separar palabras en el nombre de la misma.

Ejemplo:

PHP:

```
<?php
function conectar_db($arg1,$arg2 = 'x'){
    if (condition){
        statement;
    }
    return $val;
}
?>
```

Clases:

Estas deberán nombrarse con la primera palabra iniciando con mayúscula y separando las demás palabras con letra mayúscula.

Ejemplo:

PHP:

```
<?php
class EstadoPedido{
    function foo($arg1){
        statement;
    }
}
?>
```

2.10 Propuesta del sistema

El presente trabajo propone desarrollar una segunda versión de la aplicación web SIGIGO para eliminar las deficiencias del sistema anterior. Cada una de las funcionalidades se implementará como parte de los diferentes módulos y tendrá un propósito específico, relacionado con el flujo de trabajo actual.

El sistema permitirá a los usuarios:

Gestionar campo.

Gestionar formulario.

Gestionar registro en la tabla nomencladora.

Gestionar registro en la lista de valores.

Gestionar estructura (decanos facultad).

Gestionar solicitud de medicamentos.

Gestionar tratamiento paciente.

Gestionar orden de complementarios.

Gestionar resultados de complementarios.

Notificar a decanos.

Consultar calendario.

2.11 Conclusiones

En este capítulo se han expuesto las principales características que debe tener el sistema, en función de realizar un sistema capaz de cumplir con las expectativas del cliente. A favor de dicha propuesta se exponen los procesos presentes en el negocio, los procesos que serán objetos de automatización y la nueva propuesta de la arquitectura de la información. Además, se obtiene un listado de requisitos no funcionales que la aplicación debe cumplir.

Después de todo lo manifestado en el transcurso del capítulo, se permite comenzar con la próxima fase de desarrollo del sistema para dar cumplimiento a los requerimientos impuestos por el cliente para lograr resultados satisfactorios.

Capítulo 3: Desarrollo de la Solución

3.1 Introducción

En el presente capítulo se hace referencia a las fases de exploración y planificación, propias de la metodología Extreme Programming utilizada para el desarrollo del sistema. También se detallan las historias de usuario en adelante (HU) para luego establecer el orden en que serán implementadas atendiendo a su prioridad, esto constituye uno de los pasos iniciales de la metodología seleccionada. Además, se exponen otros artefactos generados durante el transcurso de estas fases.

3.2 Fase I: Exploración

En esta etapa el cliente crea las HU que son de interés para la entrega del producto. Además, el equipo de desarrollo se familiariza con las herramientas y tecnologías que se utilizarán en el proyecto.

3.2.1 Historias de Usuario

Las HU tienen el mismo propósito que los casos de uso. Estas se escriben desde la perspectiva del cliente aunque los desarrolladores pueden brindar también su ayuda en la identificación de las mismas. El contenido de estas debe ser concreto y sencillo.

A continuación la plantilla que se usará para definir las HU.

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 11 historias de usuario las cuales se describen a continuación:

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

Historia de Usuario	
Número: 1	Usuario: Ginecóloga, Enfermera.
Nombre Historia de Usuario: Gestionar campo.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Puntos estimados: 1	Iteración Asignada: 1
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: El usuario podrá crear nuevos campos, modificar campos usados o desactivar campos que serán utilizados en los distintos tipos de formatos.	
Observaciones: Para modificar o eliminar un campo este debe estar insertado con anterioridad en la aplicación.	

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

Historia de Usuario	
Número: 2	Usuario: Ginecóloga, Enfermera

Nombre Historia de Usuario: Gestionar formato.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Puntos estimados: 1	Iteración Asignada: 1
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: Se le brinda la posibilidad al usuario de crear un nuevo formato en la aplicación, para esto deberá ingresar los campos por los que estará compuesto.	
Observaciones: Los campos deben estar insertados con anterioridad en la aplicación.	

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

Historia de Usuario	
Número: 3	Usuario: Ginecóloga, Enfermera
Nombre Historia de Usuario: Gestionar registro en la tabla nomencladora.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Puntos estimados: 0.5	Iteración Asignada: 1
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: El usuario podrá crea un nuevo registro en la tabla nomencladora, elimina o modifica un registro ya existente en la aplicación.	
Observaciones: Para modificar o eliminar un registro este debe estar insertado con anterioridad en la aplicación.	

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

Historia de Usuario	
Número: 4	Usuario: Ginecóloga, Enfermera
Nombre Historia de Usuario: Gestionar registro en lista de valores.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Bajo
Puntos estimados: 0.5	Iteración Asignada: 1
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: El usuario podrá crear un registro en la lista de valores, elimina o modifica un registro ya existente de la lista de valores.	
Observaciones: Para modificar o eliminar un registro este debe estar insertado con anterioridad en la aplicación.	

Ver anexo 1 - 7

3.3 Fase II: Planificación de la Entrega

Durante la fase de planificación se realiza una estimación del esfuerzo que costará implementar cada HU. Este se expresa utilizando como medida el punto. Un punto se considera como una semana (6 días) ideal de trabajo donde los miembros de los equipos de desarrollo trabajan el tiempo planeado sin ningún tipo de interrupción. Por regla general, las HU necesitan de una a tres semanas de desarrollo.

3.3.1 Estimación de esfuerzos por historias de usuario

Las estimaciones del esfuerzo para implementar las HU permiten tener una medida real de la velocidad de progreso del proyecto y brindan una guía a la cual ajustarse. Los resultados obtenidos se muestran a continuación:

Tabla 7: Estimación de esfuerzos por Historia de Usuario.

Historia de Usuario	Puntos de Estimación (semanas)
Gestionar campo	1
Gestionar formato	1
Gestionar registro en la tabla nomencladora	0.5
Gestionar registro en lista de valores	0.5
Gestionar estructura (decanos facultad)	0.5
Gestionar solicitud de medicamentos	0.5
Gestionar tratamiento paciente	0.5
Gestionar complementarios	0.5
Gestionar resultados de complementarios	0.5
Notificar a decano	1
Consultar calendario	1

3.3.2 Plan de iteraciones

Después de identificar y detallar las HU y estimar 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 HU que serán implementadas en cada iteración y las posibles fechas de sus entregas.

En el transcurso de la fase de planificación se identificaron tres iteraciones, las cuales se especifican a continuación:

Iteración 1:

En esta iteración se implementan las HU que tienen prioridad alta en el negocio, de esta forma, se van creando las funcionalidades principales del sistema que dan soporte a la implementación de las demás funcionalidades. Estas HU son: 1, 2, 3, 4, las cuales hacen alusión de modo general a la administración y configuración del sistema, como son: gestionar campo, gestionar formato, gestionar registro en la tabla nomencladora, gestionar registro en la lista de valores. Además, se tendrá la primera versión de prueba, la cual será mostrada al cliente con el objetivo de obtener una retroalimentación para el grupo de trabajo.

Iteración 2:

El objetivo de esta iteración es la implementación de las restantes funcionalidades con prioridad alta, además de otras con prioridad media. Con la culminación de la misma se tendrán implementadas las peticiones del cliente descritas en las HU 6, 7, 8, 9, 11 en las cuales se hace referencia a funcionalidades como: gestionar solicitud de medicamentos, gestionar tratamiento paciente, gestionar complementarios, gestionar resultados de complementarios y consultar calendario. Al igual que en la primera iteración, al término de esta se contará también con una versión de prueba del producto, en este caso la número 2.

Iteración 3:

El objetivo fundamental de la última iteración está dado por la implementación de aquellas HU que tienen una prioridad baja dentro del negocio. Con la culminación de la misma se tendrán implementadas las peticiones del cliente descritas en las HU 5, 10 en las cuales se hace referencia a las funcionalidades gestionar estructura y notificar decano. Al término de esta iteración se contará con la versión final del producto. A partir de este momento el sistema será puesto a prueba por un período de tiempo para evaluar el desempeño del mismo.

3.3.3 Plan de duración de las iteraciones

El objetivo fundamental del plan de duración de las iteraciones es mostrar la duración de cada iteración y el orden en que serán implementadas las HU en cada una de éstas.

Tabla 8: Plan de duración de las iteraciones.

Iteraciones	Orden de las historias usuario a implementar	Duración total de las iteraciones
Iteración 1	Gestionar campo Gestionar formato Gestionar registro en la tabla nomencladora Gestionar registro en la lista de valores	3 semanas
Iteración 2	Gestionar solicitud de medicamentos Gestionar tratamiento paciente Gestionar complementarios Gestionar resultados de complementarios Consultar calendario	3 semanas
Iteración 3	Gestionar estructura Notificar decano	1 semana

3.3.4 Plan de entregas

El propósito del plan de entregas es establecer las HU que serán agrupadas para conformar una entrega, y el orden de las mismas. En este plan se unen las funcionalidades referentes a un mismo tema en módulos, esto permite un mayor entendimiento en la fase de implementación.

Tabla 9: Funcionalidades por módulos.

Módulos	Historias de usuario que abarca
Configuración	1. Gestionar campo

	<ol style="list-style-type: none"> 2. Gestionar formato 3. Gestionar registro en la tabla nomencladora 4. Gestionar registro en la lista de valores 5. Gestionar estructura
Medicamentos	<ol style="list-style-type: none"> 1. Gestionar solicitar medicamentos
Tratamiento	<ol style="list-style-type: none"> 1. Gestionar tratamiento paciente
Complementarios	<ol style="list-style-type: none"> 1. Gestionar complementarios 2. Gestionar resultados de complementarios
Notificaciones	<ol style="list-style-type: none"> 1. Notificar decano
Calendario	<ol style="list-style-type: none"> 1. Consulta calendario

Tabla 10: Plan de duración de entrega.

Módulos	Final 1ra Iteración 2da semana de abril	Final 2da Iteración 1ra semana de mayo	Final 3ra Iteración 2da semana de mayo
Configuración	V1.0		V1.5 Final
Medicamentos		V1.1	Finalizado
Tratamiento		V1.2	Finalizado
Complementarios		V1.3	Finalizado
Calendario		V1.4	Finalizado
Notificaciones			V1.6 Final

3.4 Conclusiones

En el desarrollo de este capítulo se realizó todo lo referente a las etapas de Exploración y Planificación que propone la metodología seleccionada. En este se generaron las HU que caracterizan al sistema, así como una estimación de la duración de cada una y los planes de entregas e iteraciones.

Capítulo 4: Implementación y Prueba

4.1 Introducción

La metodología XP plantea que la implementación de un producto debe realizarse de forma iterativa, esta característica trae consigo que después del desarrollo de cada iteración se obtenga un producto funcional que debe ser mostrado al cliente y previamente probado para incrementar la visión de los desarrolladores y clientes de posibles cambios. En el presente capítulo se detallan las tres iteraciones llevadas a cabo durante la construcción del sistema, además se exponen las tareas generadas por cada HU, así como las pruebas de aceptación efectuadas sobre el sistema.

4.2 Diseño del Sistema

4.2.1 Tarjetas Clase, Responsabilidad y Colaboración (CRC)

La tarjeta CRC es una técnica de modelado orientado a objetos que permite identificar las clases y sus responsabilidades. El nombre de la clase se coloca en forma de título en la tarjeta, en la parte izquierda las funcionalidades (responsabilidades) y en la parte derecha las clases que se implican en cada funcionalidad (colaboración).

A continuación se describen por cada uno de los módulos las tarjetas CRC pertenecientes a estos, permitiendo definir y simular los escenarios que garantizan el buen funcionamiento del diseño.

Tabla 11: Tarjeta CRC Clase Configuración

Clase: Configuración	
Responsabilidad	Colaboración
Gestionar campo	TbTipoCampo
Gestionar formato	TbCampo
Gestionar registro en la tabla nomencladora	TbTipoFormato
Gestionar registro en la lista de valores	TbNomencladora

Gestionar estructura	TbFormato TbInstanciaFormatoCampo
----------------------	--------------------------------------

Tabla 12: Tarjeta CRC Clase Medicamentos

Clase: Medicamentos	
Responsabilidad	Colaboración
Gestionar solicitar medicamentos	TbPedido TbPedidoAnticonceptivo TbUnidadMedida TbTipoAnticonceptivo TbAnticonceptivo TbClasificacion

Tabla 13: Tarjeta CRC Clase Tratamiento

Clase: Tratamiento	
Responsabilidad	Colaboración
Gestionar tratamiento paciente	TbPaciente TbInstanciaFormatoAnticonceptivo

Tabla 14: Tarjeta CRC Clase Complementarios

Clase: Complementarios	
Responsabilidad	Colaboración
Gestionar complementarios Gestionar resultados de complementarios	TbTipoCampo TbCampo TbTipoFormato TbNomencladora TbFormato TbInstanciaFormatoCampo

Tabla 15: Tarjeta CRC Clase Notificaciones

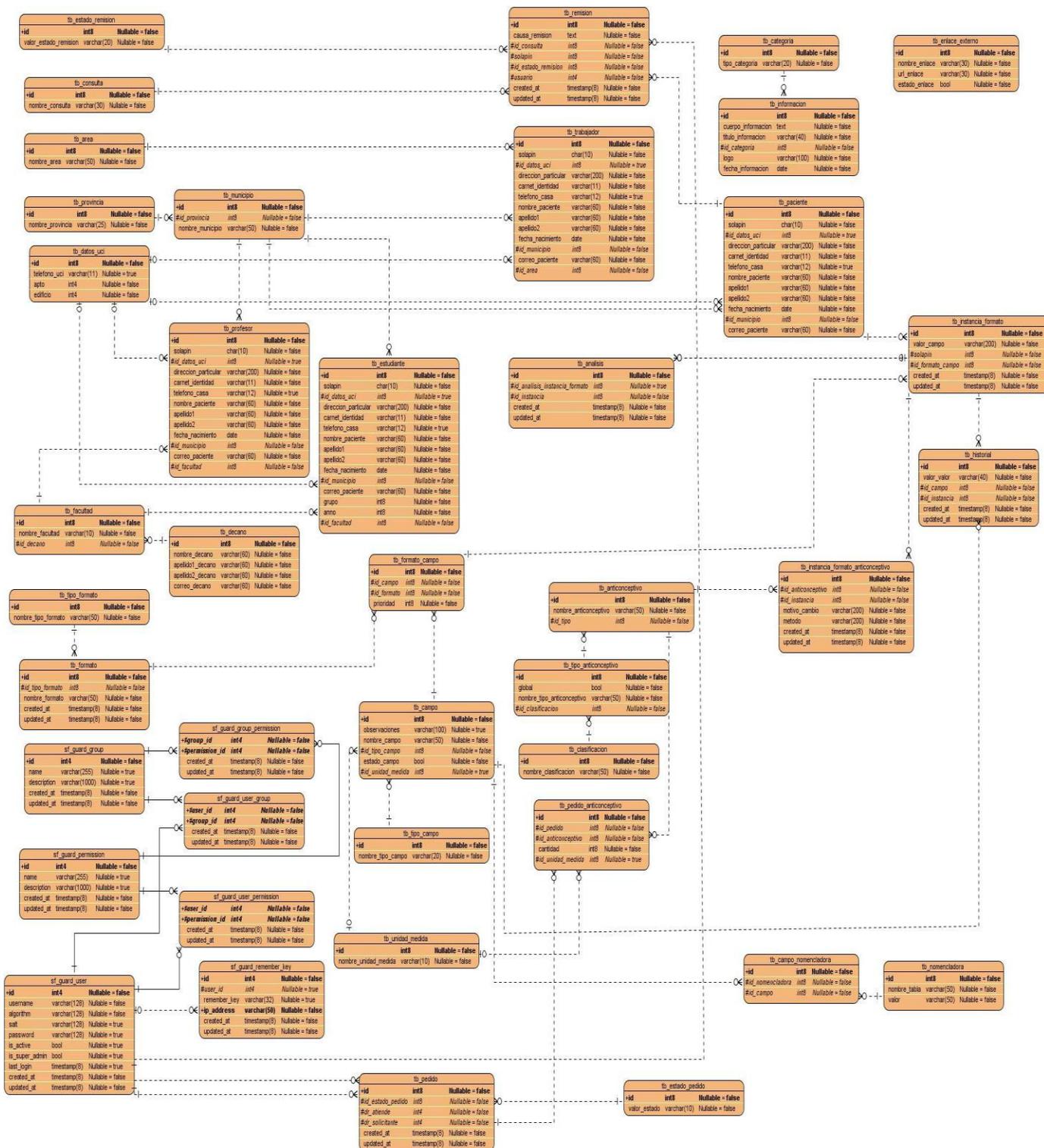
Clase: Notificaciones	
Responsabilidad	Colaboración
Notificar decano	TbInstanciaFormato TbFormato TbTipoFormato TbTipoCampo TbCampo

Tabla 16: Tarjeta CRC Clase Calendario

Clase: Calendario	
Responsabilidad	Colaboración
Consultar calendario	TbRemision TbFormato TbTipoFormato TbCampo TbTipoCampo

4.3 Diseño de la base de datos

Una de las tareas más importantes a la hora de construir una aplicación web es la base de datos. El modelo de datos del sistema SIGIGO posee un nivel de complejidad alto, como consecuencia de que éste está compuesto por numerosas tablas, así como las relaciones entre ellas. A continuación se muestra el modelo de datos que se utilizó:



4.4 Fase de Implementación

En esta fase se realiza la implementación de las HU que fueron seleccionadas por cada iteración. Al inicio se lleva a cabo un chequeo del plan de iteraciones por si es necesario realizar modificaciones. Como parte de este plan se crean tareas de programación para ayudar a organizar la implementación exitosa de las HU.

4.4.1 Tareas de programación por iteraciones

Cada HU como funcionalidad de la aplicación está compuesta por una o varias tareas de programación, éstas no son más que pasos lógicos a seguir por el programador para realizar la implementación de una HU. A continuación se detallan para cada una de las iteraciones las tareas a desarrollar por cada HU.

4.4.1.1 Iteración 1

Tabla 17: Tiempo real de implementación por módulo (iteración 1)

Módulos	Historias de Usuario	Tiempo de Implementación (Semana)	
		Estimación	Real
Configuración	Gestionar campo	1	0.5
	Gestionar formato	1	1
	Gestionar registro en la tabla nomencladora	0.5	0.5
	Gestionar registro en la lista de valores	0.5	0.5

Tabla 18: Tarea de programación por historias de usuario (iteración 1)

Historias de Usuario	Tarea de programación por historias de usuario
Gestionar campo	<ol style="list-style-type: none"> 1. Mostrar interfaz "Gestionar campo" 2. Crear campo 3. Modificar campo 4. Desactivar campo
Gestionar formato	<ol style="list-style-type: none"> 1. Mostrar interfaz "Gestionar formato" 2. Crear formato
Gestionar registro en la tabla nomencladora	<ol style="list-style-type: none"> 1. Mostrar interfaz "Gestionar formularios médicos" 2. Crear registro en la tabla nomencladora 3. Modificar registro en la tabla nomencladora
Gestionar registro en la lista de valores	<ol style="list-style-type: none"> 1. Mostrar interfaz "Gestionar tabla nomencladora" 2. Crear registro en la lista de valores 3. Modificar registro en la lista de valores 4. Eliminar registro en la lista de valores

Ver anexo 8 - 20

4.4.1.2 Iteración 2

Tabla 19: Tiempo real de implementación por módulo (iteración 2)

Módulos	Historias de Usuario	Tiempo de Implementación (Semana)	
		Estimación	Real
Medicamentos	Gestionar solicitud de medicamentos	0.5	0.2
Tratamiento	Gestionar tratamiento paciente	0.5	0.5
Complementarios	Gestionar complementarios	0.5	0.5
	Gestionar resultados de complementarios	0.5	0.5
Calendario	Consultar calendario	1	1

Tabla 20: Tarea de programación por historias de usuario (iteración 2)

Historias de Usuario	Tarea de programación por historias de usuario
Gestionar solicitud de medicamentos	<ol style="list-style-type: none"> 1. Mostrar interfaz "Gestionar pedido de medicamentos" 2. Crear solicitud de medicamentos 3. Modificar solicitud de medicamentos 4. Cancelar solicitud de medicamentos
Gestionar tratamiento paciente	<ol style="list-style-type: none"> 1. Mostrar interfaz "Gestionar tratamiento"

	<ol style="list-style-type: none"> 2. Asignar tratamiento 3. Registrar cambio de tratamiento
Gestionar complementarios	<ol style="list-style-type: none"> 1. Mostrar interfaz "Gestionar complementarios" 2. Crear orden de complementarios 3. Modificar orden de complementarios 4. Eliminar orden de complementarios
Gestionar resultados de complementarios	<ol style="list-style-type: none"> 1. Mostrar interfaz "Gestionar resultados de complementarios" 2. Registrar resultados de complementarios 3. Modificar resultados de complementarios 4. Eliminar resultados de complementarios
Consultar calendario	<ol style="list-style-type: none"> 1. Consultar calendario

Ver anexo 21 - 37

4.4.1.3 Iteración 3

Tabla 21: Tiempo real de implementación por módulo (iteración 3)

Módulos	Historias de Usuario	Tiempo de Implementación (Semana)	
		Estimación	Real
Configuración	Gestionar estructura	0.5	0.2
Notificaciones	Notificar decano	0.5	0.5

Tabla 22: Tarea de programación por historias de usuario (iteración 3)

Historias de Usuario	Tarea de programación por historias de usuario
Gestionar estructura	<ol style="list-style-type: none"> 1. Mostrar interfaz "Gestionar datos de decano" 2. Crear estructura organizativa 3. Modificar estructura organizativa 4. Eliminar estructura organizativa
Notificar decano	<ol style="list-style-type: none"> 1. Consultar resultados de ultrasonido 2. Consultar tipo de paciente 3. Enviar notificación

Ver anexo 38 - 44

4.5 Fase de Pruebas

La metodología XP divide las pruebas en dos grupos: pruebas unitarias, desarrolladas por los programadores, encargadas de verificar el código de forma automática y las pruebas de aceptación, destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente. (Beck, Kent. 2000)

4.5.1 Pruebas de aceptación

Las pruebas de aceptación son pruebas de caja negra que se crean a partir de las HU. Durante las iteraciones las HU seleccionadas serán traducidas a pruebas de aceptación. En ellas se especifican, desde la perspectiva del cliente, los escenarios para probar que una HU ha sido implementada correctamente. Una HU puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. El objetivo final de éstas es garantizar que los requerimientos han sido cumplidos y que el sistema es aceptable. Una HU no se considera completa hasta que no ha pasado por sus pruebas de aceptación. (Beck, Kent. 2000)

Tabla 23: Prueba 1 al Módulo Configuración (iteración 1)

Caso de Prueba de Aceptación	
Código: HU1_P1	Historia de Usuario: 1
Nombre: Crear campo.	
Descripción: Prueba para funcionalidad crear campo.	
Condiciones de Ejecución: El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>) o de Enfermera (Ariagnys Quintana Montalvo-> enfermera). Se introducirán datos válidos.	
Entrada/Pasos de ejecución: El usuario seleccionará la opción crear campo. Introducirá los datos necesarios para crear un campo (ejemplo: observaciones: el campo contendrá un listado de enfermedades, nombre_campo: Enfermedades, id_tipo_campo: selección múltiple, estado_campo: activado, id_unidad_medida: Null).	
Resultados Esperados: Se mostrará el siguiente mensaje “El campo ha sido creado satisfactoriamente”.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 24: Prueba 2 al Módulo Configuración (iteración 1)

Caso de Prueba de Aceptación	
Código: HU1_P2	Historia de Usuario: 1
Nombre: Crear campo.	
Descripción: Prueba para funcionalidad crear campo.	
Condiciones de Ejecución: El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa ->	

<p><i>ginecóloga</i>) o de Enfermera (Ariagnys Quintana Montalvo-> enfermera).</p> <p>Se introducirán datos inválidos.</p>
<p>Entrada/Pasos de ejecución:</p> <p>El usuario seleccionará la opción crear campo.</p> <p>No introducirá los datos necesarios para crear un campo, dejará algunos campos vacíos (ejemplo: observaciones: Null, nombre_campo: Null, id_tipo_campo: selección múltiple, estado_campo: activado, id_unidad_medida: Null).</p>
<p>Resultados Esperados: Se mostrará el siguiente mensaje “Error al crear campo”.</p>
<p>Evaluación de la Prueba: Satisfactoria.</p>

Tabla 25: Prueba 3 al Módulo Configuración (iteración 1)

Caso de Prueba de Aceptación	
Código: HU1_P3	Historia de Usuario: 1
Nombre: Modificar campo.	
Descripción: Prueba para funcionalidad modificar campo.	
<p>Condiciones de Ejecución:</p> <p>El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>) o de Enfermera (Ariagnys Quintana Montalvo-> enfermera).</p> <p>Se introducirán datos válidos.</p>	
<p>Entrada/Pasos de ejecución:</p> <p>El usuario seleccionará la opción modificar campo.</p> <p>Se cambiará alguno de los atributos de campo (ejemplo: nombre_campo: Examen).</p>	
<p>Resultados Esperados: Se mostrará el siguiente mensaje “El campo ha sido modificado satisfactoriamente”.</p>	

Evaluación de la Prueba: Satisfactoria.

Tabla 26: Prueba 4 al Módulo Configuración (iteración 1)

Caso de Prueba de Aceptación	
Código: HU1_P4	Historia de Usuario: 1
Nombre: Modificar campo.	
Descripción: Prueba para funcionalidad modificar campo.	
Condiciones de Ejecución: El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>) o de Enfermera (Ariagnys Quintana Montalvo-> enfermera). Se introducirán datos inválidos.	
Entrada/Pasos de ejecución: El usuario seleccionará la opción modificar campo. Se deja vacío el atributo de campo que va a ser modificado (ejemplo: observaciones: Null, nombre_campo: Null).	
Resultados Esperados: Se mostrará el siguiente mensaje "Error al modificar campo".	
Evaluación de la Prueba: Satisfactoria.	

Tabla 27: Prueba 5 al Módulo Configuración (iteración 1)

Caso de Prueba de Aceptación	
Código: HU1_P5	Historia de Usuario: 1
Nombre: Desactivar campo.	
Descripción: Prueba para funcionalidad desactivar campo.	

<p>Condiciones de Ejecución:</p> <p>El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>) o de Enfermera (Ariagnys Quintana Montalvo-> enfermera).</p> <p>Se introducirán datos válidos.</p>
<p>Entrada/Pasos de ejecución:</p> <p>El usuario seleccionará la opción desactivar campo.</p> <p>Se cambiará el estado del campo (ejemplo: estado_campo: desactivado).</p>
<p>Resultados Esperados: Se mostrará el siguiente mensaje “El campo ha sido modificado satisfactoriamente”.</p>
<p>Evaluación de la Prueba: Satisfactoria.</p>

Tabla 28: Prueba 6 al Módulo Configuración (iteración 1)

Caso de Prueba de Aceptación	
Código: HU2_P1	Historia de Usuario: 2
Nombre: Crear formato.	
Descripción: Prueba para funcionalidad crear formato.	
<p>Condiciones de Ejecución:</p> <p>El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>) o de Enfermera (Ariagnys Quintana Montalvo-> enfermera).</p> <p>Se introducirán datos válidos.</p>	
<p>Entrada/Pasos de ejecución:</p> <p>El usuario seleccionará la opción crear formato.</p> <p>Introducirá los datos necesarios para crear un formato (ejemplo: id_campo: Edad, id_formato: Historia Clínica Ginecología, prioridad: 1).</p>	
Resultados Esperados: Se mostrará el siguiente mensaje “El formato ha sido creado	

satisfactoriamente”.
Evaluación de la Prueba: Satisfactoria.

Tabla 29: Prueba 7 al Módulo Configuración (iteración 1)

Caso de Prueba de Aceptación	
Código: HU2_P2	Historia de Usuario: 2
Nombre: Crear formato.	
Descripción: Prueba para funcionalidad crear formato.	
Condiciones de Ejecución: El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>) o de Enfermera (Ariagnys Quintana Montalvo-> <i>enfermera</i>). Se introducirán datos inválidos.	
Entrada/Pasos de ejecución: El usuario seleccionará la opción crear formato. No introducirá los datos necesarios para crear un formato, dejará algunos campos vacíos (ejemplo: id_campo: Edad, id_formato: Historia Clínica Ginecología, prioridad: Null).	
Resultados Esperados: Se mostrará el siguiente mensaje “Error al crear formato”.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 30: Prueba 8 al Módulo Configuración (iteración 1)

Caso de Prueba de Aceptación	
Código: HU3_P1	Historia de Usuario: 3
Nombre: Crear registro en la tabla nomencladora.	

Descripción: Prueba para funcionalidad crear registro en la tabla nomencladora.
Condiciones de Ejecución: El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>) o de Enfermera (Ariagnys Quintana Montalvo-> enfermera). Se introducirán datos válidos.
Entrada/Pasos de ejecución: El usuario seleccionará la opción crear registro en la tabla nomencladora. Introducirá los datos necesarios para crear un registro en la tabla nomencladora (ejemplo: valor_campo: positivo, solapin: 57972, id_formato_campo: Resultado)
Resultados Esperados: Se mostrará el siguiente mensaje “El registro en la tabla nomencladora ha sido creado satisfactoriamente”.
Evaluación de la Prueba: Satisfactoria.

Tabla 31: Prueba 9 al Módulo Configuración (iteración 1)

Caso de Prueba de Aceptación	
Código: HU3_P2	Historia de Usuario: 3
Nombre: Crear registro en la tabla nomencladora.	
Descripción: Prueba para funcionalidad crear registro en la tabla nomencladora.	
Condiciones de Ejecución: El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>) o de Enfermera (Ariagnys Quintana Montalvo-> enfermera). Se introducirán datos inválidos.	
Entrada/Pasos de ejecución: El usuario seleccionará la opción crear registro en la tabla nomencladora. No introducirá los datos necesarios para crear un registro en la tabla nomencladora, dejará algunos	

campos vacíos (ejemplo: valor_campo: Null).
Resultados Esperados: Se mostrará el siguiente mensaje “Error al crear registro en la tabla nomencladora”.
Evaluación de la Prueba: Satisfactoria.

Tabla 32: Prueba 10 al Módulo Configuración (iteración 1)

Caso de Prueba de Aceptación	
Código: HU3_P3	Historia de Usuario: 3
Nombre: Modificar registro en la tabla nomencladora.	
Descripción: Prueba para funcionalidad modificar registro en la tabla nomencladora.	
Condiciones de Ejecución: El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>) o de Enfermera (Ariagnys Quintana Montalvo-> enfermera). Se introducirán datos válidos.	
Entrada/Pasos de ejecución: El usuario seleccionará la opción modificar registro en la tabla nomencladora. Se cambiará alguno de los atributos de instancia_formato_campo (ejemplo: valor_campo: negativo).	
Resultados Esperados: Se mostrará el siguiente mensaje “El registro ha sido modificado satisfactoriamente”.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 33: Prueba 11 al Módulo Configuración (iteración 1)

Caso de Prueba de Aceptación	
Código: HU3_P4	Historia de Usuario: 3
Nombre: Modificar registro en la tabla nomencladora.	
Descripción: Prueba para funcionalidad modificar registro en la tabla nomencladora.	
Condiciones de Ejecución: El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>) o de Enfermera (Ariagnys Quintana Montalvo-> enfermera). Se introducirán datos inválidos.	
Entrada/Pasos de ejecución: El usuario seleccionará la opción modificar registro en la tabla nomencladora. Se cambiará alguno de los atributos de instancia_formato_campo (ejemplo: valor_campo: Null).	
Resultados Esperados: Se mostrará el siguiente mensaje “Error al modificar registro en la tabla nomencladora”.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 34: Prueba 12 al Módulo Configuración (iteración 1)

Caso de Prueba de Aceptación	
Código: HU4_P1	Historia de Usuario: 4
Nombre: Crear registro en la lista de valores.	
Descripción: Prueba para funcionalidad crear registro en la lista de valores.	

<p>Condiciones de Ejecución:</p> <p>El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>) o de Enfermera (Ariagnys Quintana Montalvo-> enfermera).</p> <p>Se introducirán datos válidos.</p>
<p>Entrada/Pasos de ejecución:</p> <p>El usuario seleccionará la opción crear registro en la lista de valores.</p> <p>Introducirá los datos necesarios para crear un registro en la lista de valores (ejemplo: nombre_tabla: Enfermedades, valor: Sida).</p>
<p>Resultados Esperados: Se mostrará el siguiente mensaje “El registro en la lista de valores ha sido creado satisfactoriamente”.</p>
<p>Evaluación de la Prueba: Satisfactoria.</p>

Tabla 35: Prueba 13 al Módulo Configuración (iteración 1)

Caso de Prueba de Aceptación	
Código: HU4_P2	Historia de Usuario: 4
Nombre: Crear registro en la lista de valores.	
Descripción: Prueba para funcionalidad crear registro en la lista de valores.	
<p>Condiciones de Ejecución:</p> <p>El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>) o de Enfermera (Ariagnys Quintana Montalvo-> enfermera).</p> <p>Se introducirán datos inválidos.</p>	
<p>Entrada/Pasos de ejecución:</p> <p>El usuario seleccionará la opción crear registro en la lista de valores.</p> <p>No introducirá los datos necesarios para crear un registro en la lista de valores, dejará algunos campos vacíos (ejemplo: nombre_tabla: Null, valor: Null).</p>	

Resultados Esperados: Se mostrará el siguiente mensaje “Error al crear registro en la lista de valores”.
Evaluación de la Prueba: Satisfactoria.

Tabla 36: Prueba 14 al Módulo Configuración (iteración 1)

Caso de Prueba de Aceptación	
Código: HU4_P3	Historia de Usuario: 4
Nombre: Modificar registro en la lista de valores.	
Descripción: Prueba para funcionalidad modificar registro en la lista de valores.	
Condiciones de Ejecución: El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>) o de Enfermera (Ariagnys Quintana Montalvo-> enfermera). Se introducirán datos válidos.	
Entrada/Pasos de ejecución: El usuario seleccionará la opción modificar registro en la lista de valores. Se cambiará alguno de los atributos del registro en la lista de valores (Ejemplo: valor: Condiloma).	
Resultados Esperados: Se mostrará el siguiente mensaje “El registro ha sido modificado satisfactoriamente”.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 37: Prueba 15 al Módulo Configuración (iteración 1)

Caso de Prueba de Aceptación	
Código: HU4_P4	Historia de Usuario: 4
Nombre: Modificar registro en la lista de valores.	

Descripción: Prueba para funcionalidad modificar registro en la lista de valores.
Condiciones de Ejecución: El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>) o de Enfermera (Ariagnys Quintana Montalvo-> enfermera). Se introducirán datos inválidos.
Entrada/Pasos de ejecución: El usuario seleccionará la opción modificar registro en la lista de valores. Se deja vacío el atributo del registro en la lista de valores que va a ser modificado (ejemplo: valor: Null).
Resultados Esperados: Se mostrará el siguiente mensaje "Error al modificar registro en la lista de valores".
Evaluación de la Prueba: Satisfactoria.

Tabla 38: Prueba 16 al Módulo Configuración (iteración 1)

Caso de Prueba de Aceptación	
Código: HU4_P5	Historia de Usuario: 4
Nombre: Eliminar registro en la lista de valores.	
Descripción: Prueba para funcionalidad eliminar registro en la lista de valores.	
Condiciones de Ejecución: El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>) o de Enfermera (Ariagnys Quintana Montalvo-> enfermera). Se introducirán datos válidos.	
Entrada/Pasos de ejecución: El usuario seleccionará la opción eliminar registro en la lista de valores.	
Resultados Esperados: Se mostrará el siguiente mensaje "El registro se ha eliminado"	

satisfactoriamente”.

Evaluación de la Prueba: Satisfactoria.
--

Ver anexo 45 - 68

4.6 Conclusiones

En el presente capítulo se trazaron las tareas de programación correspondientes para dar solución a las historias de usuario planteadas, así como el modelo de datos y las pruebas de aceptación que se llevaron a cabo con el objetivo de brindarle al cliente un producto funcional que cumpla con las necesidades especificadas por el mismo. Con la conclusión de este capítulo se considera terminada la propuesta que ofrece el trabajo.

Conclusiones

Durante el proceso de desarrollo del trabajo de diploma se continuó con el estudio de los principales sistemas de gestión de información hospitalaria a nivel mundial, pero específicamente relacionados al área de ginecología, llegándose a demostrar la necesidad de desarrollar un sistema capaz de automatizar siguientes procesos: gestionar campo, gestionar formulario, gestionar registro en la tabla nomencladora, gestionar registro en la lista de valores, gestionar estructura (decanos facultad), gestionar solicitud de medicamentos, gestionar tratamiento paciente, gestionar orden de complementarios, gestionar resultados de complementarios, notificar a decanos, consultar calendario, dando solución así a todos los requisitos exigidos por parte del cliente.

El uso de la metodología XP para guiar el proceso y los artefactos generados en cada una de sus fases propició una mayor comunicación entre el equipo de desarrollo y el cliente, logrando obtener un mejor funcionamiento del sistema SIGIGO y que éste responda a todas las necesidades del cliente. Por lo que se considera concluida la investigación y dado por cumplido todos los objetivos establecidos al comienzo de la investigación.

Recomendaciones:

Con el presente trabajo de diploma se exhorta a:

- 1- Utilizar el presente trabajo como bibliografía para futuras investigaciones que utilicen la misma línea de implementación expuesta.
- 2- Valorar la posibilidad de integración del sistema con otros ya existentes en el policlínico-hospital y hacer extensiva esta propuesta a otras áreas.
- 3- Estudiar la posibilidad de realizar un sistema similar para ser usado en el país, específicamente en las consultas de Ginecología y Obstetricia.

Referencias Bibliográficas

1. **Álvarez, Miguel Angel. 2009.** Desarrolloweb.Codelgniter. *Desarrolloweb.Codelgniter*. [En línea] 23 de noviembre de 2009. [Citado el: 5 de febrero de 2010.] <http://www.desarrolloweb.com/articulos/codeigniter.html>.
2. **Beck, Kent. 2000.** *Extreme Programming Explained*. Addison-Wesley Professional. [En línea] 2009. [Citado el: 8 de marzo de 2010.] <http://books.google.com.cu/books?id=G8EL4H4vf7UC&printsec=frontcover&dq=Beck,+Kent.+2000.&cd=1#v=onepage&q=Beck%2C%20Kent.%202000.&f=false>
3. **BMD. 2005.** BioMedical developments. *BioMedical developments*. [En línea] 2005. [Citado el: 25 de enero de 2010.] <http://biomedical.com.ar/>
4. **Bustamante, Antonio Montes de Oca Sánchez de. 2004.** Arquitectura de información y usabilidad: nociones básicas para los profesionales de la información. *Arquitectura de información y usabilidad: nociones básicas para los profesionales de la información*. [Online] 2004. [Cited: marzo 10, 2010.] http://bvs.sld.cu/revistas/aci/vol12_6_04/aci04604.htm.
5. **Casanova, Jaime. 2009.** Postgre SQL. *Postgre SQL*. [En línea] 15 de abril de 2009. [Citado el: 10 de febrero de 2010.] <http://archives.postgresql.org/pgsql-es-fomento/2009-04/msg00035.php>.
6. **CooKbook. 2009.** El Cookbook. *El Cookbook*. [En línea] 2009. [Citado el: 27 de enero de 2010.] <http://book.cakephp.org/es/>.
7. **François Zaninotto y Fabien Potencier. 2008.** Symfony 1.2, la guía definitiva. *Symfony 1.2, la guía definitiva*. [En línea] 2008. [Citado el: 7 de febrero de 2010.] http://librosweb.es/symfony_1_2/capitulo1/symfony_en_pocas_palabras.html.
8. **Infonet. 2005.** Infonet Software. *Infonet Software*. [En línea] 2005. [Citado el: 15 de febrero de 2010.] <http://www.infonetsoftware.com/>.
9. **Isd. 2005.** ISD-Interactive Solutions Development. *ISD-Interactive Solutions Development*. [En línea] 2005. [Citado el: 25 de enero de 2010.] <http://www.isd.com.ar/>.

10. **Leopoldo, Carlos. 2007.** Techtastico. Zend Framework, una Introducción. *Techtastico. Zend Framework, una Introducción*. [En línea] 27 de noviembre de 2007. [Citado el: 3 de febrero de 2010.] <http://techtastico.com/post/zend-framework-una-introduccion/>.
11. **Medigest. 2007.** Medigest Consultores. *Medigest Consultores*. [En línea] 2007. [Citado el: 26 de enero de 2010.] <http://www.medigest.com/>.
12. **KICKBILL. 2009.** Kickbill.Tus primeros pasos con Zend Framework: Parte 1. *Kickbill.Tus primeros pasos con Zend Framework: Parte 1*. [En línea] [Citado el: 6 de febrero de 2010.] <http://www.kickbill.com/?p=1232>.
13. **Zend, Framework. 2009.** Capítulo 3. Inicio Rápido con Zend Framework. *Capítulo 3. Inicio Rápido con Zend Framework*. [En línea] 2009. [Citado el: 2 de febrero de 2010.] <http://manual.zfdes.com/es>
http://bvs.sld.cu/revistas/aci/vol12_6_04/aci04604.htm

Bibliografía

1. **Bustamante, Antonio Montes de Oca Sánchez de. 2004.** Arquitectura de información y usabilidad: nociones básicas para los profesionales de la información. *Arquitectura de información y usabilidad: nociones básicas para los profesionales de la información*. [En línea] 2004. [Consultado el: marzo 10, 2010.] http://bvs.sld.cu/revistas/aci/vol12_6_04/aci04604.htm.
2. **Gerardo Fernández Escribano. 2002.** Introducción a Extreme Programming. [Consultado el: 28 de enero de 2010.] <http://www.info-ab.uclm.es/asignaturas/42551/trabajosAnteriores/Presentacion-XP.pdf>
3. **Herrera Muñoz, Álvaro Felipe y Casanova Merchán, Jaime Alberto. 2009.** *Curso avanzado de PostgreSQL edición UCI 2009*.
4. **Potencier, Fabien. 2009.** *El tutorial Jobeet*. 2009. [En línea] 2008. [Consultado el: 8 de febrero de 2010.] http://librosweb.es/jobeeet_1_3/.
5. **François Zaninotto y Fabien Potencier. 2008.** Symphony 1.2, la guía definitiva. *Symfony 1.2, la guía definitiva*. [En línea] 2008. [Consultado el: 7 de febrero de 2010.] http://librosweb.es/symfony_1_2/capitulo1/symfony_en_pocas_palabras.html.
6. **SencioLab. 2009.** *Symfony Forms in Action*. 2009.
7. **Yoannis Puente Márquez, Laura Candebat Giraudy. 2007-2008.** Transitando por las migraciones internacionales: Arquitectura de Información. 2007-2008. [Consultado el: 18 de febrero de 2010.] <http://www.uh.cu/centros/cemi/anuario%200708/2008/11%20Laura%20y%20Yoannis%20Al%20en%20el%20Anuario%20CEMI%2007%20-%202008.pdf>.

Glosario de términos

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

Anamnesis: (traer a la memoria) significa recolección, reminiscencia, rememoración. La anamnesis en general apunta a traer al presente los recuerdos del pasado, recuperar la información registrada en épocas pretéritas.

API (application programming interface): (interfaz de programación de aplicaciones), es un conjunto de funciones residentes en bibliotecas.

Bracket: corchete, llave, paréntesis; abrazadera, ménsula; escuadra, soporte; clasificación, categoría, categorización, clase.

BSD (Berkeley Software Distribution): (Distribución de Software Berkeley) se utiliza para identificar un sistema operativo derivado del sistema Unix nacido a partir de los aportes realizados a ese sistema por la Universidad de California en Berkeley.

Bugs: Es un error de software.

CRUD: Es el acrónimo de Crear, Obtener, Actualizar y Borrar (Create, Retrieve, Update y Delete en inglés).

CSS: (Significa hojas de estilo en cascada) es un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML.

FSM (free space map): mapa de espacio libre.

FTP (File Transfer Protocol): (Protocolo de Transferencia de Archivos) es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP (Transmission Control Protocol), basada en la arquitectura cliente-servidor.

HTML (HyperText Markup Language): Lenguaje de Marcado de Hipertexto, es el lenguaje de marcado predominante para la elaboración de páginas web.

ITS: Infecciones de transmisión sexual.

jQuery: Es una biblioteca o framework de Javascript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web.

Licencia BSD: es la licencia de software otorgada principalmente para los sistemas BSD.

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.

MIT (Massachusetts Institute of Technology): (Instituto Tecnológico de Massachussets) es una de las principales instituciones dedicadas a la docencia y a la investigación en Estados Unidos, especialmente en ciencia, ingeniería y economía.

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

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

Normadat S.A: empresa con más de 15 años de experiencia en la gestión externa de documentación.

Oracle: es un sistema de gestión de base de datos relacional (RDBMS por el acrónimo en inglés de Relational Data Base Management System), desarrollado por Oracle Corporation.

ORM (mapeo objeto-relacional): (Object-Relational mapping, o sus siglas O/RM, ORM, y O/R mapping) es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional.

PAMI: Programa de Atención Materno-Infantil.

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

PHP: Hypertext Pre-processor. Es un ambiente script del lado del servidor que permite crear y ejecutar aplicaciones Web dinámicas e interactivas.

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

RSS: es una familia de formatos de fuentes web codificados en XML.

SEGO: Sociedad Española de Ginecología y Obstetricia.

SQL (Structured Query Language): (lenguaje de consulta estructurado) es un lenguaje declarativo de acceso a bases de datos relacionales.

SSL (Secure Socket Layer): es un protocolo de seguridad desarrollado por la empresa Netscape Communications para lograr que la transmisión de datos entre un servidor y un usuario, o viceversa, a través de Internet.

URL (Uniform Resource Locator): Localizador uniforme de recursos, es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet para su localización o identificación.

Vademécum: Libro de consulta inmediata de datos o nociones fundamentales de uso frecuente en determinada materia.

Web service (servicio web): es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

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

Anexos

Historias de usuario

Anexo 1: Representación de la Historia de Usuario Nro. 5

Historia de Usuario	
Número: 5	Usuario: Ginecóloga, Administrador
Nombre Historia de Usuario: Gestionar estructura (decanos facultad).	
Prioridad en Negocio: Baja	Riesgo en Desarrollo: Bajo
Puntos estimados: 0.5	Iteración Asignada: 3
Programador responsable: Patricia Cauleau Castro – Luis Felipe Díaz Barrios	
Descripción: El usuario podrá crear, eliminar o modificar la estructura organizativa de los decanos de cada facultad.	
Observaciones: Para modificar o eliminar una estructura esta debe estar insertada con anterioridad en la aplicación.	

Anexo 2: Representación de la Historia de Usuario Nro. 6.

Historia de Usuario	
Número: 6	Usuario: Ginecóloga, Enfermera
Nombre Historia de Usuario: Gestionar solicitud de medicamentos.	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Medio
Puntos estimados: 0.5	Iteración Asignada: 3

Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios
Descripción: El usuario podrá crear, cancelar y modificar una solicitud de medicamentos.
Observaciones: Para modificar o eliminar una solicitud esta debe estar insertada con anterioridad en la aplicación.

Anexo 3: Representación de la Historia de Usuario Nro. 7.

Historia de Usuario	
Número: 7	Usuario: Ginecóloga
Nombre Historia de Usuario: Gestionar tratamiento paciente.	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Alto
Puntos estimados: 0.5	Iteración Asignada: 2
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: El usuario podrá asignar un tratamiento y registrar porque fue realizado un cambio de tratamiento en una paciente determinada.	
Observaciones: El usuario debe estar autenticado con anterioridad.	

Anexo 4: Representación de la Historia de Usuario Nro. 8.

Historia de Usuario	
Número: 8	Usuario: Ginecóloga, Médico del consultorio
Nombre Historia de Usuario: Gestionar complementarios.	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Alto

Puntos estimados: 0.5	Iteración Asignada: 2
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: El usuario podrá crear, eliminar, modificar una orden de complementarios, además de consultar (buscar) los resultados de los complementarios.	
Observaciones: Para modificar o eliminar una orden de complementarios o mostrar los resultados deben estar insertadas la orden y los resultados respectivamente con anterioridad en la aplicación.	

Anexo 5: Representación de la Historia de Usuario Nro. 9.

Historia de Usuario	
Número: 9	Usuario: Técnico de laboratorio
Nombre Historia de Usuario: Gestionar resultados de complementarios.	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Alto
Puntos estimados: 0.5	Iteración Asignada: 2
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: El usuario podrá registrar, eliminar, modificar resultados de complementarios.	
Observaciones: Para modificar o eliminar un resultado debe estar insertado con anterioridad en la aplicación.	

Anexo 6: Representación de la Historia de Usuario Nro. 10.

Historia de Usuario	
Número: 10	Usuario: Sistema
Nombre Historia de Usuario: Notificar a decano.	

Prioridad en Negocio: Bajo	Riesgo en Desarrollo: Media
Puntos estimados: 0.5	Iteración Asignada: 3
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: El usuario podrá notificar a los decanos de las facultades la situación de una determinada paciente.	
Observaciones: El resultado del ultrasonido tiene que estar insertado en la aplicación.	

Anexo 7: Representación de la Historia de Usuario Nro. 11

Historia de Usuario	
Número: 11	Usuario: Paciente
Nombre Historia de Usuario: Consultar calendario.	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Alto
Puntos estimados: 1	Iteración Asignada: 2
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: El usuario podrá ver su calendario de turnos planificados.	
Observaciones: El usuario debe estar autenticado con anterioridad.	

Tareas de programación

Iteración 1

Anexo 8: Tarea de Programación: “Mostrar interfaz Gestionar campo”

Tarea de programación	
Tarea Nro.1	Número Historia (Nro.1): Gestionar campo
Nombre tarea: Mostrar interfaz "Gestionar campo"	
Tipo tarea: Desarrollo	Puntos estimados: 0.3
Fecha inicio: 25/03/2010	Fecha fin: 26/03/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: Una vez que el usuario se autentica, este selecciona la opción Gestionar campo e inmediatamente se muestra la interfaz de usuario correspondiente.	

Anexo 9: Tarea de Programación: “Crear campo”

Tarea de programación	
Tarea Nro.2	Número Historia (Nro.1): Gestionar campo
Nombre tarea: Crear campo	
Tipo tarea: Desarrollo	Puntos estimados: 0.3
Fecha inicio: 27/03/2010	Fecha fin: 29/03/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	

Descripción: Una vez que se muestra la interfaz “Crear campo” el usuario inserta los datos necesarios para crear un campo y selecciona el botón Salvar si desea insertar los datos de un solo campo o el botón Salvar y Adicionar si desea insertar más de un campo. Si no ocurren errores el sistema muestra el mensaje “El campo ha sido creado satisfactoriamente” o “El campo ha sido creado satisfactoriamente, puede adicionar otro campo” en correspondencia al botón seleccionado, en caso contrario se visualiza un mensaje de error: “Error al crear campo”.

Anexo 10: Tarea de Programación: "Modificar campo"

Tarea de programación	
Tarea Nro.3	Número Historia (Nro.1): Gestionar campo
Nombre tarea: Modificar campo	
Tipo tarea: Desarrollo	Puntos estimados: 0.3
Fecha inicio: 29/03/2010	Fecha fin: 30/03/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: Una vez que se muestra la interfaz correspondiente se procede a modificar un campo. Seguidamente se selecciona “Salvar”. Si no ocurren errores el sistema muestra el mensaje “El campo ha sido modificado satisfactoriamente”, en caso contrario se visualiza un mensaje de error, “Error al modificar campo”.	

Anexo 11: Tarea de Programación: "Desactivar campo"

Tarea de programación	
Tarea Nro.4	Número Historia (Nro.1): Gestionar campo
Nombre tarea: Desactivar campo	
Tipo tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 31/03/2010	Fecha fin: 31/03/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: Una vez que se muestra la interfaz con los datos se selecciona "Editar". Si no ocurren errores el sistema muestra el mensaje "El campo ha sido desactivado satisfactoriamente", en caso contrario se visualiza un mensaje de error, "Error al desactivar campo".	

Anexo 12: Tarea de Programación: "Mostrar interfaz Gestionar formato"

Tarea de programación	
Tarea Nro.1	Número Historia (Nro.2): Gestionar formato
Nombre tarea: Mostrar interfaz "Gestionar formato"	
Tipo tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 1/04/2010	Fecha fin: 3/04/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: Una vez que el usuario se autentica, este selecciona la opción Gestionar formato e	

inmediatamente se muestra la interfaz de usuario correspondiente.

Anexo 13: Tarea de Programación: "Crear formato"

Tarea de programación	
Tarea Nro.2	Número Historia (Nro.2): Gestionar formato
Nombre tarea: Crear formato	
Tipo tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 5/04/2010	Fecha fin: 7/04/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: Una vez que se muestra la interfaz "Nuevo formato" el usuario inserta los datos necesarios para crear un formato y selecciona el botón Salvar si desea insertar los datos de un solo formato o el botón Salvar y Adicionar si desea insertar más de un formato. Si no ocurren errores el sistema muestra el mensaje "El formato ha sido creado satisfactoriamente", en caso contrario se visualiza un mensaje de error, "Error al crear formato".	

Anexo 14: Tarea de Programación: "Mostrar interfaz Gestionar formularios médicos"

Tarea de programación	
Tarea Nro.1	Número Historia (Nro.3): Gestionar registro en la tabla nomencladora
Nombre tarea: Mostrar interfaz "Gestionar formularios médicos"	

Tipo tarea: Desarrollo	Puntos estimados: 0.3
Fecha inicio: 8/04/2010	Fecha fin: 9/04/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: Una vez que el usuario se autentica, este selecciona la opción Gestionar formularios médicos e inmediatamente se muestra la interfaz de usuario correspondiente.	

Anexo 15: Tarea de Programación: "Crear registro en la tabla nomencladora"

Tarea de programación	
Tarea Nro.2	Número Historia (Nro.3): Gestionar registro en la tabla nomencladora
Nombre tarea: Crear registro en la tabla nomencladora	
Tipo tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 10/04/2010	Fecha fin: 10/04/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: Una vez que se muestra la interfaz "Nueva tabla nomencladora" el usuario inserta los datos necesarios para Crear el registro en la tabla nomencladora y selecciona el botón Salvar si desea insertar los datos de un registro o el botón Salvar y Adicionar si desea insertar más de un registro. Si no ocurren errores el sistema muestra el mensaje "El registro ha sido creado satisfactoriamente" o "El registro ha sido creado satisfactoriamente, puede adicionar otro registro" en correspondencia al botón seleccionado, en caso contrario se visualiza un mensaje de error, "Error al crear registro en la tabla nomencladora".	

Anexo 16: Tarea de Programación: "Modificar registro en la tabla nomencladora"

Tarea de programación	
Tarea Nro.3	Número Historia (Nro.3): Gestionar registro en la tabla nomencladora
Nombre tarea: Modificar registro en la tabla nomencladora	
Tipo tarea:	Puntos estimados: 0.1
Fecha inicio: 10/04/2010	Fecha fin: 10/04/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: Una vez que se muestra la interfaz correspondiente se procede a modificar un registro en la tabla nomencladora. Seguidamente se selecciona "Editar". Si no ocurren errores el sistema muestra el mensaje "El registro de la tabla nomencladora ha sido modificado satisfactoriamente", en caso contrario se visualiza un mensaje de error, "Error al modificar registro de la tabla nomencladora".	

Anexo 17: Tarea de Programación: "Mostrar interfaz Gestionar tabla nomencladora"

Tarea de programación	
Tarea Nro.1	Número Historia (Nro.4): Gestionar registro en la lista de valores
Nombre tarea: Mostrar interfaz "Gestionar tabla nomencladora"	
Tipo tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 12/04/2010	Fecha fin: 12/04/2010

Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios
Descripción: Una vez que el usuario se autentica, este selecciona la opción Gestionar tabla nomencladora e inmediatamente se muestra la interfaz de usuario correspondiente.

Anexo 18: Tarea de Programación: "Crear registro en la lista de valores"

Tarea de programación	
Tarea Nro.2	Número Historia (Nro.4): Gestionar registro en la lista de valores
Nombre tarea: Crear registro en la lista de valores	
Tipo tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 12/04/2010	Fecha fin: 13/04/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: Una vez que se muestra la interfaz "Crear registro en la lista de valores" el usuario inserta los datos necesarios para Crear el registro en la lista de valores y selecciona el botón Salvar si desea insertar los datos de un registro o el botón Salvar y Adicionar si desea insertar más de un registro. Si no ocurren errores el sistema muestra el mensaje "El registro ha sido creado satisfactoriamente" o "El registro ha sido creado satisfactoriamente, puede adicionar otro registro" en correspondencia al botón seleccionado, en caso contrario se visualiza un mensaje de error, "Error al crear registro en la lista de valores".	

Anexo 19: Tarea de Programación: "Modificar registro en la lista de valores"

Tarea de programación	
Tarea Nro.3	Número Historia (Nro.4): Gestionar registro en la lista de valores
Nombre tarea: Modificar registro en la lista de valores	
Tipo tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 14/04/2010	Fecha fin: 14/04/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: Una vez que se muestra la interfaz correspondiente se procede a modificar un registro en la lista de valores. Seguidamente se selecciona "Editar". Si no ocurren errores el sistema muestra el mensaje "El registro en la lista de valores ha sido modificado satisfactoriamente", en caso contrario se visualiza un mensaje de error, "Error al modificar un registro en la lista de valores".	

Anexo 20: Tarea de Programación: "Eliminar registro en la lista de valores"

Tarea de programación	
Tarea Nro.4	Número Historia (Nro.4): Gestionar registro en la lista de valores
Nombre tarea: Eliminar registro en la lista de valores	
Tipo tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 14/04/2010	Fecha fin: 14/04/2010

Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios
Descripción: Una vez que se muestra la interfaz con los datos se selecciona “Eliminar”. Si no ocurren errores el sistema muestra el mensaje “El registro en la lista de valores se ha eliminado satisfactoriamente”, en caso contrario se visualiza un mensaje de error, “Error al eliminar el registro en la lista de valores”.

Iteración 2

Anexo 21: Tarea de Programación: "Mostrar interfaz Gestionar pedido de medicamentos"

Tarea de programación	
Tarea Nro.1	Número Historia (Nro.6): Gestionar solicitud de medicamentos
Nombre tarea: Mostrar interfaz "Gestionar pedido de medicamentos"	
Tipo tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 15/04/2010	Fecha fin: 15/04/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: Una vez que el usuario se autentica, este selecciona la opción Gestionar solicitud de medicamentos e inmediatamente se muestra la interfaz de usuario correspondiente.	

Anexo 22: Tarea de Programación: "Crear solicitud de medicamentos"

Tarea de programación	
Tarea Nro.2	Número Historia (Nro.6): Gestionar solicitud de

	medicamentos
Nombre tarea: Crear solicitud de medicamentos	
Tipo tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 15/04/2010	Fecha fin: 16/04/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: Una vez que se muestra la interfaz “Crear solicitud de medicamentos” el usuario inserta los datos necesarios para Crear la solicitud de medicamentos y selecciona el botón Salvar si desea insertar los datos de una solicitud de medicamentos o el botón Salvar y Adicionar si desea insertar más de una solicitud de medicamentos. Si no ocurren errores el sistema muestra el mensaje “La solicitud de medicamentos ha sido creada satisfactoriamente” o “La solicitud de medicamentos ha sido creada satisfactoriamente, puede adicionar otra solicitud” en correspondencia al botón seleccionado, en caso contrario se visualiza un mensaje de error, “Error al crear la solicitud de medicamentos”.	

Anexo 23: Tarea de Programación: "Modificar solicitud de medicamentos"

Tarea de programación	
Tarea Nro.3	Número Historia (Nro.6): Gestionar solicitud de medicamentos
Nombre tarea: Modificar solicitud de medicamentos	
Tipo tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 17/04/2010	Fecha fin: 17/04/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: Una vez que se muestra la interfaz correspondiente se procede a modificar una solicitud	

de medicamentos. Seguidamente se selecciona "Editar". Si no ocurren errores el sistema muestra el mensaje "La solicitud de medicamentos ha sido modificada satisfactoriamente", en caso contrario se visualiza un mensaje de error, "Error al modificar la solicitud de medicamentos".

Anexo 24: Tarea de Programación: "Cancelar solicitud de medicamentos"

Tarea de programación	
Tarea Nro.4	Número Historia (Nro.6): Gestionar solicitud de medicamentos
Nombre tarea: Cancelar solicitud de medicamentos	
Tipo tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 17/04/2010	Fecha fin: 17/04/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: Una vez que se muestra la interfaz con los datos se selecciona "Eliminar". Si no ocurren errores el sistema muestra el mensaje "La solicitud de medicamentos ha sido cancelada satisfactoriamente", en caso contrario se visualiza un mensaje de error, "Error al cancelar la solicitud de medicamentos".	

Anexo 25: Tarea de Programación: "Mostrar interfaz Gestionar tratamiento"

Tarea de programación	
Tarea Nro.1	Número Historia (Nro.7): Gestionar tratamiento paciente

Nombre tarea: Mostrar interfaz "Gestionar tratamiento"	
Tipo tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 19/04/2010	Fecha fin: 19/04/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: Una vez que el usuario se autentica, este selecciona la opción Gestionar tratamiento e inmediatamente se muestra la interfaz de usuario correspondiente.	

Anexo 26: Tarea de Programación: "Asignar tratamiento"

Tarea de programación	
Tarea Nro.2	Número Historia (Nro.7): Gestionar tratamiento paciente
Nombre tarea: Asignar tratamiento	
Tipo tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 19/04/2010	Fecha fin: 20/04/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: Una vez que se muestra la interfaz "Asignar tratamiento" el usuario inserta los datos necesarios para Asignar el tratamiento y selecciona el botón Asignar tratamiento. Si no ocurren errores el sistema muestra el mensaje "Se ha asignado el tratamiento satisfactoriamente", en caso contrario se visualiza un mensaje de error, "Error al asignar tratamiento".	

Anexo 27: Tarea de Programación: "Registrar cambio de tratamiento"

Tarea de programación	
Tarea Nro.3	Número Historia (Nro.7): Gestionar tratamiento paciente
Nombre tarea: Registrar cambio de tratamiento	
Tipo tarea:	Puntos estimados:
Fecha inicio: 21/04/2010	Fecha fin: 22/04/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: Una vez que se muestra la interfaz correspondiente se procede a registrar el cambio de tratamiento. Seguidamente se selecciona "Registrar cambio". Si no ocurren errores el sistema muestra el mensaje "Se ha registrado el cambio de tratamiento satisfactoriamente", en caso contrario se visualiza un mensaje de error, "Error al registrar cambio de tratamiento".	

Anexo 28: Tarea de Programación: "Mostrar interfaz Gestionar complementarios"

Tarea de programación	
Tarea Nro.1	Número Historia (Nro.8): Gestionar complementarios
Nombre tarea: Mostrar interfaz "Gestionar complementarios"	
Tipo tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 22/04/2010	Fecha fin: 23/04/2010

Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios
Descripción: Una vez que el usuario se autentica, este selecciona la opción Gestionar complementarios e inmediatamente se muestra la interfaz de usuario correspondiente.

Anexo 29: Tarea de Programación: "Crear orden de complementarios"

Tarea de programación			
Tarea Nro.2	Número	Historia	(Nro.8): Gestionar complementarios
Nombre tarea: Crear orden de complementarios			
Tipo tarea: Desarrollo	Puntos estimados: 0.1		
Fecha inicio: 24/04/2010	Fecha fin: 24/04/2010		
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios			
Descripción: Una vez que se muestra la interfaz "Crear orden de complementarios" el usuario inserta los datos necesarios para Crear la orden de complementarios y selecciona el botón Crear orden de complementarios. Si no ocurren errores el sistema muestra el mensaje "La orden de complementarios ha sido creada satisfactoriamente", en caso contrario se visualiza un mensaje de error, "Error al crear orden de complementarios".			

Anexo 30: Tarea de Programación: "Modificar orden de complementarios"

Tarea de programación			
Tarea Nro.3	Número	Historia	(Nro.8): Gestionar

	complementarios
Nombre tarea: Modificar orden de complementarios	
Tipo tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 24/04/2010	Fecha fin: 24/04/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: Una vez que se muestra la interfaz correspondiente se procede a modificar la orden de complementarios. Seguidamente se selecciona “Enviar cambio”. Si no ocurren errores el sistema muestra el mensaje “La orden de complementarios ha sido modificada satisfactoriamente”, en caso contrario se visualiza un mensaje de error, “Error al modificar la orden de complementarios”.	

Anexo 31: Tarea de Programación: "Eliminar orden de complementarios"

Tarea de programación			
Tarea Nro.4	Número	Historia	(Nro.8): Gestionar complementarios
Nombre tarea: Eliminar orden de complementarios			
Tipo tarea: Desarrollo	Puntos estimados: 0.1		
Fecha inicio: 26/04/2010	Fecha fin: 26/04/2010		
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios			
Descripción: Una vez que se muestra la interfaz con los datos se selecciona “Eliminar orden de complementarios”. Si no ocurren errores el sistema muestra el mensaje “La orden de complementarios se ha eliminado satisfactoriamente”, en caso contrario se visualiza un mensaje de error, “Error al eliminar			

la orden de complementarios”.

Anexo 32: Tarea de Programación: "Mostrar interfaz Gestionar resultados de complementarios"

Tarea de programación	
Tarea Nro.1	Número Historia (Nro.9): Gestionar resultados de complementarios
Nombre tarea: Mostrar interfaz "Gestionar resultados de complementarios"	
Tipo tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 27/04/2010	Fecha fin: 27/04/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: Una vez que el usuario se autentica, este selecciona la opción Gestionar resultados de complementarios e inmediatamente se muestra la interfaz de usuario correspondiente.	

Anexo 33: Tarea de Programación: "Registrar resultados de complementarios"

Tarea de programación	
Tarea Nro.2	Número Historia (Nro.9): Gestionar resultados de complementarios
Nombre tarea: Registrar resultados de complementarios	
Tipo tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 28/04/2010	Fecha fin: 28/04/2010

Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios
Descripción: Una vez que se muestra la interfaz “Registrar resultados de complementarios” el usuario inserta los datos necesarios para Registrar los resultados de los complementarios y selecciona el botón Registrar resultados de complementarios. Si no ocurren errores el sistema muestra el mensaje “Los resultados de los complementarios han sido registrados satisfactoriamente”, en caso contrario se visualiza un mensaje de error, “Error al registrar los resultados de los complementarios”.

Anexo 34: Tarea de Programación: "Modificar resultados de complementarios"

Tarea de programación	
Tarea Nro.3	Número Historia (Nro.9): Gestionar resultados de complementarios
Nombre tarea: Modificar resultados de complementarios	
Tipo tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 28/04/2010	Fecha fin: 28/04/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: Una vez que se muestra la interfaz correspondiente se procede a modificar los resultados de los complementarios. Seguidamente se selecciona “Enviar cambio”. Si no ocurren errores el sistema muestra el mensaje “Los resultados de los complementarios han sido modificados satisfactoriamente”, en caso contrario se visualiza un mensaje de error, “Error al modificar los resultados de complementarios”.	

Anexo 35: Tarea de Programación: "Eliminar resultados de complementarios"

Tarea de programación	
Tarea Nro.4	Número Historia (Nro.9): Gestionar resultados de complementarios
Nombre tarea: Eliminar resultados de complementarios	
Tipo tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 29/04/2010	Fecha fin: 29/04/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: Una vez que se muestra la interfaz con los datos se selecciona "Eliminar resultados de complementarios". Si no ocurren errores el sistema muestra el mensaje "Los resultados de los complementarios han sido eliminados satisfactoriamente", en caso contrario se visualiza un mensaje de error, "Error al eliminar los resultados de los complementarios".	

Anexo 36: Tarea de Programación: "Mostrar interfaz Gestionar calendario médico"

Tarea de programación	
Tarea Nro.1	Número Historia (Nro.11): Consultar calendario
Nombre tarea: Mostrar interfaz "Gestionar calendario médico"	
Tipo tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 30/04/2010	Fecha fin: 3/05/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	

Descripción: Una vez que el usuario se autentica, este selecciona la opción Gestionar calendario médico e inmediatamente se muestra la interfaz de usuario correspondiente.

Anexo 37: Tarea de Programación: " Consultar remisiones con fecha menor que la actual"

Tarea de programación	
Tarea Nro.2	Número Historia (Nro.11): Consultar calendario
Nombre tarea: Consultar remisiones con fecha menor que la actual.	
Tipo tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 4/05/2010	Fecha fin: 6/05/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: Consiste en crear una consulta para obtener un listado con todas las remisiones que tienen una fecha menor que la actual.	

Iteración 3

Anexo 38: Tarea de Programación: "Mostrar interfaz Gestionar datos de decano"

Tarea de programación	
Tarea Nro.1	Número Historia (Nro.5): Gestionar estructura
Nombre tarea: Mostrar interfaz "Gestionar datos de decano"	
Tipo tarea: Desarrollo	Puntos estimados: 0.1

Fecha inicio: 7/05/2010	Fecha fin: 7/05/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: Una vez que el usuario se autentica, este selecciona la opción Gestionar datos de decano e inmediatamente se muestra la interfaz de usuario correspondiente.	

Anexo 39: Tarea de Programación: "Crear estructura organizativa"

Tarea de programación	
Tarea Nro.2	Número Historia (Nro.5): Gestionar estructura
Nombre tarea: Crear estructura organizativa	
Tipo tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 7/05/2010	Fecha fin: 8/05/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: Una vez que se muestra la interfaz "Crear estructura organizativa" el usuario inserta los datos necesarios para Crear la estructura organizativa y selecciona el botón Salvar si desea insertar una estructura organizativa o el botón Salvar y Adicionar si desea insertar más de una estructura organizativa. Si no ocurren errores el sistema muestra el mensaje "La estructura organizativa ha sido creada satisfactoriamente" o "La estructura organizativa ha sido creada satisfactoriamente, puede adicionar otra estructura" en correspondencia al botón seleccionado, en caso contrario se visualiza un mensaje de error, "Error al crear la estructura organizativa".	

Anexo 40: Tarea de Programación: "Modificar estructura organizativa"

Tarea de programación

Tarea Nro.3	Número Historia (Nro.5): Gestionar estructura
Nombre tarea: Modificar estructura organizativa	
Tipo tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 10/05/2010	Fecha fin: 10/05/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: Una vez que se muestra la interfaz correspondiente se procede a modificar la estructura organizativa que se desea. Seguidamente se selecciona "Editar". Si no ocurren errores el sistema muestra el mensaje "La estructura organizativa ha sido modificada satisfactoriamente", en caso contrario se visualiza un mensaje de error, "Error al modificar la estructura organizativa".	

Anexo 41: Tarea de Programación: "Eliminar estructura organizativa"

Tarea de programación	
Tarea Nro.4	Número Historia (Nro.5): Gestionar estructura
Nombre tarea: Eliminar estructura organizativa	
Tipo tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 10/05/2010	Fecha fin: 10/05/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: Una vez que se muestra la interfaz con los datos se selecciona "Eliminar". Si no ocurren errores el sistema muestra el mensaje "La estructura organizativa ha sido eliminada satisfactoriamente", en caso contrario se visualiza un mensaje de error, "Error al eliminar la estructura organizativa".	

Anexo 42: Tarea de Programación: "Consultar resultados de ultrasonido"

Tarea de programación	
Tarea Nro.1	Número Historia (Nro.10): Notificar decano
Nombre tarea: Consultar resultados de ultrasonido	
Tipo tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 11/05/2010	Fecha fin: 11/05/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: Consiste en crear una consulta para obtener el resultado del ultrasonido realizado a una determinada paciente.	

Anexo 43: Tarea de Programación: "Consultar tipo de paciente"

Tarea de programación	
Tarea Nro.2	Número Historia (Nro.10): Notificar decano
Nombre tarea: Consultar tipo de paciente	
Tipo tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 11/05/2010	Fecha fin: 11/05/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: Consiste en crear una consulta para obtener el tipo de paciente (estudiante o profesora).	

Anexo 44: Tarea de Programación: "Enviar notificación"

Tarea de programación	
Tarea Nro.3	Número Historia (Nro.10): Notificar decano
Nombre tarea: Enviar notificación	
Tipo tarea: Desarrollo	Puntos estimados: 0.3
Fecha inicio: 12/05/2010	Fecha fin: 13/05/2010
Programador responsable: Patricia Caulineau Castro – Luis Felipe Díaz Barrios	
Descripción: El sistema envía un correo de notificación al decano y al médico de consultorio con el nombre del paciente y la información referente al resultado de los complementarios.	

Pruebas de Aceptación

Anexo 45: Prueba 17 al Módulo Medicamentos (iteración 2)

Caso de Prueba de Aceptación	
Código: HU5_P1	Historia de Usuario: 5
Nombre: Crear solicitud de medicamentos.	
Descripción: Prueba para funcionalidad crear solicitud de medicamentos.	
Condiciones de Ejecución: El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>) o de Enfermera (Ariagnys Quintana Montalvo-> enfermera). Se introducirán datos válidos.	

<p>Entrada/Pasos de ejecución:</p> <p>El usuario seleccionará la opción crear solicitud de medicamentos.</p> <p>Introducirá los datos necesarios para crear una solicitud de medicamentos (id_estado_pedido: creado, dr: atiende: Marisol, dr: solicitud: Alicia).</p>
<p>Resultados Esperados: Se mostrará el siguiente mensaje “La solicitud de medicamentos ha sido creada satisfactoriamente”.</p>
<p>Evaluación de la Prueba: Satisfactoria.</p>

Anexo 46: Prueba 18 al Módulo Medicamentos (iteración 2)

Caso de Prueba de Aceptación	
Código: HU5_P2	Historia de Usuario: 5
Nombre: Modificar solicitud de medicamentos.	
Descripción: Prueba para funcionalidad modificar solicitud de medicamentos.	
<p>Condiciones de Ejecución:</p> <p>El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>) o de Enfermera (Ariagnys Quintana Montalvo-> enfermera).</p> <p>Se introducirán datos válidos.</p>	
<p>Entrada/Pasos de ejecución:</p> <p>El usuario seleccionará la opción modificar solicitud de medicamentos.</p> <p>Se cambiará alguno de los atributos de la solicitud de medicamentos (ejemplo: dr_atiende: Lianny).</p>	
<p>Resultados Esperados: Se mostrará el siguiente mensaje “La solicitud de medicamentos ha sido modificado satisfactoriamente”.</p>	
<p>Evaluación de la Prueba: Satisfactoria.</p>	

Anexo 47: Prueba 19 al Módulo Medicamentos (iteración 2)

Caso de Prueba de Aceptación	
Código: HU5_P3	Historia de Usuario: 5
Nombre: Cancelar solicitud medicamentos.	
Descripción: Prueba para funcionalidad cancelar solicitud medicamentos.	
Condiciones de Ejecución: El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>) o de Enfermera (Ariagnys Quintana Montalvo-> enfermera). Se introducirán datos válidos.	
Entrada/Pasos de ejecución: El usuario seleccionará la opción cancelar solicitud de medicamentos. Se cambiará el estado del pedido (valor_estado: cancelado).	
Resultados Esperados: Se mostrará el siguiente mensaje "La solicitud de medicamentos ha sido cancelada satisfactoriamente"	
Evaluación de la Prueba: Satisfactoria.	

Anexo 48: Prueba 20 al Módulo Tratamiento (iteración 2)

Caso de Prueba de Aceptación	
Código: HU6_P1	Historia de Usuario: 6
Nombre: Asignar tratamiento.	
Descripción: Prueba para funcionalidad asignar tratamiento.	
Condiciones de Ejecución:	

<p>El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>).</p> <p>Se introducirán datos válidos.</p>
<p>Entrada/Pasos de ejecución:</p> <p>El usuario seleccionará la opción asignar tratamiento.</p> <p>Introducirá los datos necesarios para asignar un tratamiento (ejemplo: id_anticonceptivo: Cipresta, motivo_cambio: aumento de peso, método: Tabletas).</p>
<p>Resultados Esperados: Se mostrará el siguiente mensaje “Se ha asignado el tratamiento satisfactoriamente”.</p>
<p>Evaluación de la Prueba: Satisfactoria.</p>

Anexo 49: Prueba 21 al Módulo Tratamiento (iteración 2)

Caso de Prueba de Aceptación	
Código: HU6_P2	Historia de Usuario: 6
Nombre: Asignar tratamiento.	
Descripción: Prueba para funcionalidad asignar tratamiento.	
<p>Condiciones de Ejecución:</p> <p>El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>).</p> <p>Se introducirán datos inválidos.</p>	
<p>Entrada/Pasos de ejecución:</p> <p>El usuario seleccionará la opción asignar tratamiento.</p> <p>No introducirá los datos necesarios para asignar un tratamiento, dejará algunos campos vacíos (ejemplo: id_anticonceptivo: Cipresta, motivo_cambio: Null, método: Null).</p>	

Resultados Esperados: Se mostrará el siguiente mensaje “Error al asignar el tratamiento”.
Evaluación de la Prueba: Satisfactoria.

Anexo 50: Prueba 22 al Módulo Tratamiento (iteración 2)

Caso de Prueba de Aceptación	
Código: HU6_P3	Historia de Usuario: 6
Nombre: Registrar cambio tratamiento.	
Descripción: Prueba para funcionalidad registrar cambio tratamiento.	
Condiciones de Ejecución: El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>). Se introducirán datos válidos.	
Entrada/Pasos de ejecución: El usuario seleccionará la opción registrar cambio de tratamiento. Introducirá los datos necesarios para registrar la asignación de un nuevo tratamiento (ejemplo: id_anticonceptivo: Triquilar, motivo_cambio: mareos, metodo: Tabletas).	
Resultados Esperados: Se mostrará el siguiente mensaje “Se ha registrado el cambio de tratamiento satisfactoriamente”.	
Evaluación de la Prueba: Satisfactoria.	

Anexo 51: Prueba 23 al Módulo Tratamiento (iteración 2)

Caso de Prueba de Aceptación

Código: HU6_P4	Historia de Usuario: 6
Nombre: Registrar cambio tratamiento.	
Descripción: Prueba para funcionalidad registrar cambio tratamiento.	
Condiciones de Ejecución: El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>). Se introducirán datos inválidos.	
Entrada/Pasos de ejecución: El usuario seleccionará la opción registrar cambio de tratamiento. No introducirá los datos necesarios para registrar la asignación de un nuevo tratamiento, dejará algunos campos vacíos (ejemplo: id_anticonceptivo: Triquilar, motivo_cambio: Null, metodo: Null).	
Resultados Esperados: Se mostrará el siguiente mensaje "Error al registrar cambio de tratamiento".	
Evaluación de la Prueba: Satisfactoria.	

Anexo 52: Prueba 24 al Módulo Complementarios (iteración 2)

Caso de Prueba de Aceptación	
Código: HU7_P1	Historia de Usuario: 7
Nombre: Crear orden complementarios.	
Descripción: Prueba para funcionalidad crear orden complementarios.	
Condiciones de Ejecución: El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>) o de Médico del consultorio. Se introducirán datos válidos.	

<p>Entrada/Pasos de ejecución:</p> <p>El usuario seleccionará la opción crear orden de complementarios.</p> <p>Introducirá los datos necesarios para crear una orden de complementarios (ejemplo: id_campo: resultado , id_formato: Complementarios Citología, prioridad: 2).</p>
<p>Resultados Esperados: Se mostrará el siguiente mensaje “La orden de complementarios ha sido creada satisfactoriamente”.</p>
<p>Evaluación de la Prueba: Satisfactoria.</p>

Anexo 53: Prueba 25 al Módulo Complementarios (iteración 2)

Caso de Prueba de Aceptación	
Código: HU7_P2	Historia de Usuario: 7
Nombre: Crear orden complementarios.	
Descripción: Prueba para funcionalidad crear orden complementarios.	
<p>Condiciones de Ejecución:</p> <p>El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>) o de Médico del consultorio.</p> <p>Se introducirán datos inválidos.</p>	
<p>Entrada/Pasos de ejecución:</p> <p>El usuario seleccionará la opción crear orden de complementarios.</p> <p>No introducirá los datos necesarios para crear una orden de complementarios, dejará algunos campos vacíos (ejemplo: id_campo: resultado, id_formato: Complementarios Citología, prioridad: Null).</p>	
Resultados Esperados: Se mostrará el siguiente mensaje “Error al crear la orden de complementarios”.	
Evaluación de la Prueba: Satisfactoria.	

Anexo 54: Prueba 26 al Módulo Complementarios (iteración 2)

Caso de Prueba de Aceptación	
Código: HU7_P3	Historia de Usuario: 7
Nombre: Modificar orden complementarios.	
Descripción: Prueba para funcionalidad modificar orden complementarios.	
Condiciones de Ejecución: El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>) o de Médico del consultorio. Se introducirán datos válidos.	
Entrada/Pasos de ejecución: El usuario seleccionará la opción modificar orden de complementarios. Se cambiará alguno de los atributos de formato_campo (ejemplo: id_campo: resultado , id_formato: Complementarios Básicos, prioridad: 3).	
Resultados Esperados: Se mostrará el siguiente mensaje “La orden de complementarios ha sido modificada satisfactoriamente”.	
Evaluación de la Prueba: Satisfactoria.	

Anexo 55: Prueba 27 al Módulo Complementarios (iteración 2)

Caso de Prueba de Aceptación	
Código: HU7_P4	Historia de Usuario: 7
Nombre: Modificar orden complementarios.	
Descripción: Prueba para funcionalidad modificar orden complementarios.	
Condiciones de Ejecución:	

<p>El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>) o de Médico del consultorio.</p> <p>Se introducirán datos inválidos.</p>
<p>Entrada/Pasos de ejecución:</p> <p>El usuario seleccionará la opción modificar orden de complementarios.</p> <p>Se deja vacío el atributo de campo que va a ser modificado (ejemplo: id_campo: resultado, id_formato: Complementarios Básicos, prioridad: Null).</p>
<p>Resultados Esperados: Se mostrará el siguiente mensaje “Error al modificar la orden de complementarios”.</p>
<p>Evaluación de la Prueba: Satisfactoria.</p>

Anexo 56: Prueba 28 al Módulo Complementarios (iteración 2)

Caso de Prueba de Aceptación	
Código: HU7_P5	Historia de Usuario: 7
Nombre: Eliminar orden complementarios.	
Descripción: Prueba para funcionalidad eliminar orden complementarios.	
<p>Condiciones de Ejecución:</p> <p>El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>) o de Médico del consultorio.</p> <p>Se introducirán datos válidos.</p>	
<p>Entrada/Pasos de ejecución:</p> <p>El usuario seleccionará la opción eliminar orden de complementarios.</p>	
<p>Resultados Esperados: Se mostrará el siguiente mensaje “La orden de complementarios ha sido eliminada satisfactoriamente”.</p>	

Evaluación de la Prueba: Satisfactoria.

Anexo 57. Prueba 29 al Módulo Complementarios (iteración 2)

Caso de Prueba de Aceptación	
Código: HU8_P1	Historia de Usuario: 8
Nombre: Registrar resultados de complementarios.	
Descripción: Prueba para funcionalidad registrar resultados complementarios.	
Condiciones de Ejecución: El usuario primeramente deberá autenticarse y poseerá el rol de Técnico de laboratorio. Se introducirán datos válidos.	
Entrada/Pasos de ejecución: El usuario seleccionará la opción registrar resultados de complementarios. En dependencia del formato del complementario seleccionado, se introducirán los datos necesarios para registrar los resultados de los complementarios.	
Resultados Esperados: Se mostrará el siguiente mensaje “Los resultados de los complementarios han sido registrados satisfactoriamente”.	
Evaluación de la Prueba: Satisfactoria.	

Anexo 58: Prueba 30 al Módulo Complementarios (iteración 2)

Caso de Prueba de Aceptación	
Código: HU8_P2	Historia de Usuario: 8
Nombre: Registrar resultados de complementarios.	
Descripción: Prueba para funcionalidad registrar resultados complementarios.	
Condiciones de Ejecución: El usuario primeramente deberá autenticarse y poseerá el rol de Técnico de laboratorio. Se introducirán datos inválidos.	
Entrada/Pasos de ejecución: El usuario seleccionará la opción registrar resultados de complementarios. En dependencia del formato del complementario seleccionado, no se introducirán los datos necesarios para registrar los resultados de los complementarios (dejará algunos campos vacíos).	
Resultados Esperados: Se mostrará el siguiente mensaje "Error al registrar los resultados de los complementarios".	
Evaluación de la Prueba: Satisfactoria.	

Anexo 59: Prueba 31 al Módulo Complementarios (iteración 2)

Caso de Prueba de Aceptación	
Código: HU8_P3	Historia de Usuario: 8
Nombre: Modificar resultados complementarios.	
Descripción: Prueba para funcionalidad modificar resultados complementarios.	
Condiciones de Ejecución: El usuario primeramente deberá autenticarse y poseerá el rol de Técnico de laboratorio. Se introducirán datos válidos.	

<p>Entrada/Pasos de ejecución:</p> <p>El usuario seleccionará la opción modificar resultados de complementarios.</p> <p>En dependencia del formato del complementario seleccionado, se introducirán los datos necesarios para modificar los resultados de los complementarios.</p>
<p>Resultados Esperados: Se mostrará el siguiente mensaje “Los resultados de los complementarios han sido modificados satisfactoriamente”.</p>
<p>Evaluación de la Prueba: Satisfactoria.</p>

Anexo 60: Prueba 32 al Módulo Complementarios (iteración 2)

Caso de Prueba de Aceptación	
Código: HU8_P4	Historia de Usuario: 8
Nombre: Modificar resultados complementarios.	
Descripción: Prueba para funcionalidad modificar resultados complementarios.	
<p>Condiciones de Ejecución:</p> <p>El usuario primeramente deberá autenticarse y poseerá el rol de Técnico de laboratorio.</p> <p>Se introducirán datos inválidos.</p>	
<p>Entrada/Pasos de ejecución:</p> <p>El usuario seleccionará la opción modificar resultados de complementarios.</p> <p>En dependencia del formato del complementario seleccionado, no se introducirán los datos necesarios para modificar los resultados de los complementarios (dejará algunos campos vacíos).</p>	
<p>Resultados Esperados: Se mostrará el siguiente mensaje “Error al modificar los resultados de los complementarios”.</p>	
<p>Evaluación de la Prueba: Satisfactoria.</p>	

Anexo 61: Prueba 33 al Módulo Complementarios (iteración 2)

Caso de Prueba de Aceptación	
Código: HU8_P5	Historia de Usuario: 8
Nombre: Eliminar resultados complementarios.	
Descripción: Prueba para funcionalidad eliminar resultados complementarios.	
Condiciones de Ejecución: El usuario primeramente deberá autenticarse y poseerá el rol de Técnico de laboratorio. Se introducirán datos válidos.	
Entrada/Pasos de ejecución: El usuario seleccionará la opción eliminar resultados de los complementarios.	
Resultados Esperados: Se mostrará el siguiente mensaje “Los resultados de los complementarios han sido eliminados satisfactoriamente”.	
Evaluación de la Prueba: Satisfactoria.	

Anexo 62: Prueba 34 al Módulo Calendario (iteración 2)

Caso de Prueba de Aceptación	
Código: HU9_P1	Historia de Usuario: 9
Nombre: Consultar calendario.	
Descripción: Prueba para funcionalidad consultar calendario.	
Condiciones de Ejecución: El usuario primeramente deberá autenticarse por el dominio UCI. Se introducirán datos válidos.	

<p>Entrada/Pasos de ejecución:</p> <p>El usuario seleccionará la opción consultar calendario.</p>
<p>Resultados Esperados: El sistema muestra los datos de las remisiones a consultas e indicaciones de complementarios de la paciente.</p>
<p>Evaluación de la Prueba: Satisfactoria.</p>

Anexo 63: Prueba 35 al Módulo Configuración (iteración 3)

Caso de Prueba de Aceptación	
Código: HU10_P1	Historia de Usuario: 10
Nombre: Crear estructura organizativa.	
Descripción: Prueba para funcionalidad crear estructura organizativa.	
<p>Condiciones de Ejecución:</p> <p>El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>) o de Administrador.</p> <p>Se introducirán datos válidos.</p>	
<p>Entrada/Pasos de ejecución:</p> <p>El usuario seleccionará la opción crear estructura organizativa.</p> <p>Introducirá los datos necesarios para crear una estructura organizativa (ejemplo: nombre_decano: Basulto, apellido1_decano: López, apellido2_decano: Pérez, correo_decano: basulto@uci.cu).</p>	
<p>Resultados Esperados: Se mostrará el siguiente mensaje “La estructura organizativa ha sido creada satisfactoriamente”.</p>	
<p>Evaluación de la Prueba: Satisfactoria.</p>	

Anexo 64: Prueba 36 al Módulo Configuración (iteración 3)

Caso de Prueba de Aceptación	
Código: HU10_P2	Historia de Usuario: 10
Nombre: Crear estructura organizativa.	
Descripción: Prueba para funcionalidad crear estructura organizativa.	
Condiciones de Ejecución: El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>) o de Administrador. Se introducirán datos inválidos.	
Entrada/Pasos de ejecución: El usuario seleccionará la opción crear estructura organizativa. No introducirá los datos necesarios para crear una estructura organizativa, dejará algunos campos vacíos (ejemplo: nombre_decanao: Null, apellido1_decanao: Null, apellido2_decanao: Null, correo_decanao: basulto@uci.cu).	
Resultados Esperados: Se mostrará el siguiente mensaje "Error al crear una estructura organizativa".	
Evaluación de la Prueba: Satisfactoria.	

Anexo 65: Prueba 37 al Módulo Configuración (iteración 3)

Caso de Prueba de Aceptación	
Código: HU10_P3	Historia de Usuario: 10

Nombre: Modificar estructura organizativa.
Descripción: Prueba para funcionalidad modificar estructura organizativa.
Condiciones de Ejecución: El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>) o de Administrador. Se introducirán datos válidos.
Entrada/Pasos de ejecución: El usuario seleccionará la opción modificar estructura organizativa. Se cambiará alguno de los atributos de la estructura organizativa (ejemplo: nombre_decano: Alcides, apellido1_decano: Campos, apellido2_decano: Pérez, correo_decano: alcides@uci.cu).
Resultados Esperados: Se mostrará el siguiente mensaje “La estructura organizativa ha sido modificada satisfactoriamente”.
Evaluación de la Prueba: Satisfactoria.

Anexo 66: Prueba 38 al Módulo Configuración (iteración 3)

Caso de Prueba de Aceptación	
Código: HU10_P4	Historia de Usuario: 10
Nombre: Modificar estructura organizativa.	
Descripción: Prueba para funcionalidad modificar estructura organizativa.	
Condiciones de Ejecución: El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>) o de Administrador. Se introducirán datos inválidos.	
Entrada/Pasos de ejecución:	

El usuario seleccionará la opción modificar estructura organizativa.
Se deja vacío el atributo de la estructura organizativa que va a ser modificado (ejemplo: nombre_decanao: Null, apellido1_decanao: Null, apellido2_decanao: Null, correo_decanao: alcides@uci.cu).
Resultados Esperados: Se mostrará el siguiente mensaje “Error al modificar una estructura organizativa”.
Evaluación de la Prueba: Satisfactoria.

Anexo 67: Prueba 39 al Módulo Configuración (iteración 3)

Caso de Prueba de Aceptación	
Código: HU10_P5	Historia de Usuario: 10
Nombre: Eliminar estructura organizativa.	
Descripción: Prueba para funcionalidad eliminar estructura organizativa.	
Condiciones de Ejecución: El usuario primeramente deberá autenticarse y poseerá el rol de Ginecóloga (Marisol González Vigoa -> <i>ginecóloga</i>) o de Administrador. Se introducirán datos válidos.	
Entrada/Pasos de ejecución: El usuario seleccionará la opción eliminar estructura organizativa.	
Resultados Esperados: Se mostrará el siguiente mensaje “La estructura organizativa ha sido eliminada satisfactoriamente”.	
Evaluación de la Prueba: Satisfactoria.	

Anexo 68: Prueba 40 al Módulo Notificación (iteración 3)

Caso de Prueba de Aceptación	
Código: HU11_P1	Historia de Usuario: 11
Nombre: Enviar notificación.	
Descripción: Prueba para funcionalidad enviar notificación.	
Condiciones de Ejecución: El resultado del complementario ultrasonido debe ser positivo.	
Entrada/Pasos de ejecución: Verifica si el resultado del ultrasonido corresponde a una estudiante y que éste es positivo.	
Resultados Esperados: El sistema envía un correo de notificación al decano y al médico de consultorio con el nombre del paciente y la información referente al resultado de los complementarios.	
Evaluación de la Prueba: Satisfactoria.	