

**FACULTAD DE INFORMÁTICA Y MATEMÁTICA
DEPARTAMENTO DE INGENIERÍA INFORMÁTICA**

**SISTEMA PARA LA GESTIÓN DEL PROCESO
DE DISTRIBUCIÓN DE MEDICAMENTOS EN
LA FARMACIA DEL HOSPITAL
UNIVERSITARIO CLÍNICO QUIRÚRGICO
“LUCÍA ÍÑIGUEZ LANDÍN”**

**Trabajo de diploma para optar por el título
de Ingeniero Informático**

Autor: Rasel Agüero Fernández

Holguín, 2018



FACULTAD DE INFORMÁTICA Y MATEMÁTICA

DEPARTAMENTO DE INGENIERÍA INFORMÁTICA

**SISTEMA PARA LA GESTIÓN DEL PROCESO DE DISTRIBUCIÓN
DE MEDICAMENTOS EN LA FARMACIA DEL HOSPITAL
UNIVERSITARIO CLÍNICO QUIRÚRGICO “LUCÍA ÍÑIGUEZ
LANDÍN”**

Trabajo de diploma para optar por el título de Ingeniero Informático

Autor: Rasel Agüero Fernández

Tutores: M.Sc. Camilo Ernesto Velázquez Rodríguez

Lic. Lorenzo Peña Marrero

Consultante: M.Sc. Yunior López Ricardo

Holguín, 2018



No se equivoca el hombre que empieza a cambiar dando pequeños pasos; se equivoca el que por miedo nunca da el primer paso, que inicia el camino que lo puede llevar a dar la vuelta al mundo.

Dedicado a mis padres, hermano y familia en general, así como a mis amigos más allegados, a ellos que me acompañaron, me ayudaron y me aconsejaron mis más sinceros agradecimientos.

Actualmente en la farmacia del Hospital Universitario Clínico Quirúrgico “Lucía Íñiguez Landín”, el trabajo de gestión e integración de los datos se realiza casi en su totalidad de forma manual, dificultando la búsqueda, la integridad y persistencia de la información. Esto conlleva a la falta de eficiencia en el almacenamiento y posterior procesamiento de los datos con el fin de apoyar la labor de los especialistas del Centro.

La presente investigación soluciona estas deficiencias a partir de la implementación de un sistema *web*. Mediante el desarrollo de la solución se centralizará de forma íntegra y persistente la información manejada por los especialistas que intervienen en el proceso de distribución de los medicamentos.

Para desarrollar el sistema se utilizaron herramientas y tecnologías de *software* libre, entre las cuales se encuentran el lenguaje de programación *Python* con su *framework* para el desarrollo *web Django*, el gestor de bases de datos *PostgreSQL* para el manejo con la base de datos y la metodología para el desarrollo de *software XP*.

At present, in the pharmacy from the Clinic-Surgical University Hospital “Lucía Íñiguez Landín”, the work related to data management and integration is mostly done by hand. For that reason, it carries out difficulties in the integrity and constancy of the information. This problem endures a lack of storage efficiency and its later data processing. The main purpose of this work is to support the specialist’s labor of this hospital.

This investigation solves these difficulties from the implementation of a web system. It will centralize the entirely and constant information managed by the specialists who distribute the medicaments.

To develop this system, there were used tools and free software technology like Python programming language with its framework to develop a Django web, the manager of the database PostgreSQL and the methodology to develop XP software.

Tabla de contenido

INTRODUCCIÓN	1
CAPÍTULO 1. MARCO TEÓRICO	1
1.1 Sistema de Distribución de Medicamentos por Dosis Unitaria.....	1
1.2 Empresa Comercializadora y Distribuidora de Medicamentos.....	2
1.3 Hospital Universitario Clínico Quirúrgico	2
1.4 Farmacia	3
1.4.1 Distribución de medicamentos en los hospitales	5
1.5 Gestión de información.....	7
1.6 Tecnologías y herramientas utilizadas	9
1.6.1 Arquitectura cliente servidor	9
1.6.2 Lenguaje de programación Python	11
1.6.3 Lenguaje de programación JavaScript.....	14
1.6.4 Framework de desarrollo web Django.....	15
1.6.5 Editor de Texto Visual Studio Code	18
1.6.6 Gestor de base de datos PostgreSQL	18
1.6.7 Servidor web Apache	20
1.6.8 Metodología de desarrollo de software XP.....	21
1.7 Conclusiones parciales	24
CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA.....	25
2.1 Exploración.....	25
2.1.1 Requerimientos Funcionales.....	25



2.1.2 Requerimientos no Funcionales.....	28
2.1.3 Historias de usuario.....	30
2.2 Planificación	36
2.2.1 Plan de iteraciones	36
2.2.2 Plan de entregas.....	37
2.3 Implementación	37
2.3.1 Iteración 1	38
2.3.2 Iteración 2	38
2.3.3 Iteración 3	41
2.3.4 Iteración 4	43
2.3.5 Iteración 5	45
2.4 Prueba	47
2.4.1 Pruebas de aceptación	48
2.6 Conclusiones parciales	54
CONCLUSIONES GENERALES	55
RECOMENDACIONES.....	57
REFERENCIAS BIBLIOGRÁFICAS	59
ANEXOS	I



Ilustración 1: Arquitectura cliente servidor..... 9



Tabla 1 Historia de Usuario No. 1. Autenticar usuario.....	30
Tabla 2: Historia de Usuario No.2. Gestionar usuario	30
Tabla 3: Historia de Usuario No.3. Gestionar rol de usuario	31
Tabla 4: Historia de Usuario No.4. Gestionar información de sala	31
Tabla 5: Historia de Usuario No.5. Gestionar información de cama.....	32
Tabla 6: Historia de Usuario No.6. Gestionar solicitudes de cama.....	32
Tabla 7: Historia de Usuario No.7. Gestionar despachos de sala	33
Tabla 8: Historia de Usuario No.8. Gestionar solicitudes de farmacia.....	33
Tabla 9: Historia de Usuario No.9. Gestionar despachos de almacén	34
Tabla 10: Historia de Usuario No.10. Gestionar existencia de medicamento.....	34
Tabla 11: Historia de Usuario No.11. Exportar información	35
Tabla 12: Historia de Usuario No.12. Gestionar medicamentos solicitados por servicio	35
Tabla 13: Distribución de las historias de usuario por iteración.....	36
Tabla 14: Cronograma de entregas.....	37
Tabla 15: Tarea 1. Autenticar usuario	38
Tabla 16: Tarea 2. Registrar usuario.....	39
Tabla 17: Tarea 3. Modificar usuario.....	39
Tabla 18: Tarea 4. Eliminar usuario	40
Tabla 19: Tarea 5. Visualizar información de usuario	40
Tabla 20: Tarea 6. Asignar rol de usuario	41
Tabla 21: Tarea 7. Modificar rol a usuario.....	42
Tabla 22: Tarea 8. Visualizar información de un rol	42
Tabla 23: Tarea 9. Agregar sala.....	43

Tabla 24: Tarea 10. Editar sala	43
Tabla 25: Tarea 11. Eliminar sala.....	44
Tabla 26: Tarea 12. Visualizar información de sala.....	44
Tabla 27: Tarea 13. Agregar cama.....	45
Tabla 28: Tarea 14. Editar cama	46
Tabla 29: Tarea 15. Eliminar cama	46
Tabla 30: Tarea 16. Visualizar información de cama	47
Tabla 31: Caso de prueba de aceptación HU1-P1	48
Tabla 32: Caso de prueba de aceptación HU4-P1	49
Tabla 33: Caso de prueba de aceptación HU5-P1	50
Tabla 34: Caso de prueba de aceptación HU6-P1	50
Tabla 35: Caso de prueba de aceptación HU9-P1	51
Tabla 36: Caso de prueba de aceptación HU10-P1	52
Tabla 37: Caso de prueba de aceptación HU11-P1	53
Tabla 38: Caso de prueba de aceptación HU12-P1	53
Tabla 39: Tarea 17. Agregar pedido de cama	I
Tabla 40: Tarea 18. Editar pedido de cama	I
Tabla 41: Tarea 19. Eliminar pedido de cama.....	II
Tabla 42: Tarea 20. Buscar pedido de cama.....	II
Tabla 43: Tarea 21. Visualizar información de pedido de cama.....	III
Tabla 44: Tarea 22. Solicitar pedido de sala	III
Tabla 45: Tarea 23. Agregar devolución de cama.....	IV
Tabla 46: Tarea 24. Editar devolución de cama	IV

Tabla 47: Tarea 25. Eliminar devolución de cama	V
Tabla 48: Tarea 26. Buscar devolución de cama	V
Tabla 49: Tarea 27. Visualizar información de devolución de cama	VI
Tabla 50: Tarea 28. Solicitar devolución de sala.....	VI
Tabla 51: Tarea 29. Despachar pedido de sala.....	VII
Tabla 52: Tarea 30. Eliminar pedido de sala.....	VII
Tabla 53: Tarea 31. Buscar pedido de sala.....	VIII
Tabla 54: Tarea 32. Visualizar información de pedido de sala.....	VIII
Tabla 55: Tarea 33. Despachar devolución de sala	IX
Tabla 56: Tarea 34. Eliminar devolución de sala.....	IX
Tabla 57: Tarea 35. Buscar devolución de sala	X
Tabla 58: Tarea 36. Visualizar información de devolución de sala.....	X
Tabla 59: Tarea 37. Agregar pedido de farmacia	XI
Tabla 60: Tarea 38. Visualizar información de pedido de farmacia.....	XI
Tabla 61: Tarea 39. Solicitar pedido al almacén	XII
Tabla 62: Tarea 40. Agregar devolución de farmacia.....	XII
Tabla 63: Tarea 41. Editar devolución de farmacia	XIII
Tabla 64: Tarea 42. Eliminar devolución de farmacia	XIII
Tabla 65: Tarea 43. Visualizar información de devolución de farmacia	XIV
Tabla 66: Tarea 44. Solicitar devolución al almacén.....	XIV
Tabla 67: Tarea 45. Despachar pedido al almacén.....	XV
Tabla 68: Tarea 46. Eliminar pedido al almacén	XV
Tabla 69: Tarea 47. Buscar pedido al almacén.....	XVI

Tabla 70: Tarea 48. Visualizar información de pedido al almacén	XVI
Tabla 71: Tarea 49. Despachar devolución al almacén	XVII
Tabla 72: Tarea 50. Eliminar devolución al almacén	XVII
Tabla 73: Tarea 51. Buscar devolución al almacén.....	XVIII
Tabla 74: Tarea 52. Visualizar información de devolución al almacén.....	XVIII
Tabla 75: Tarea 53. Agregar medicamento	XIX
Tabla 76: Tarea 54. Editar medicamento	XIX
Tabla 77: Tarea 55. Eliminar medicamento.....	XX
Tabla 78: Tarea 56. Visualizar información de medicamento.....	XX
Tabla 79: Tarea 57. Agregar existencia de medicamento	XXI
Tabla 80: Tarea 58. Editar existencia de medicamento	XXI
Tabla 81: Tarea 59. Eliminar existencia de medicamento	XXII
Tabla 82: Tarea 60. Solicitar aumento de existencia en baja cobertura.....	XXII
Tabla 83: Tarea 61. Exportar pedido de cama	XXIII
Tabla 84: Tarea 62. Exportar despacho de pedido de sala	XXIII
Tabla 85: Tarea 63. Exportar devolución de cama.....	XXIV
Tabla 86: Tarea 64. Exportar despacho de devolución de sala.....	XXIV
Tabla 87: Tarea 65. Exportar medicamentos	XXV
Tabla 88: Tarea 66. Exportar solicitud de aumento de existencia en baja cobertura	XXV
Tabla 89: Tarea 67. Exportar salas	XXVI
Tabla 90: Tarea 68. Buscar medicamentos pedidos y devueltos por centro de costo.....	XXVI

Tabla 91: Tarea 69. Visualizar información de medicamentos pedidos y devueltos por centro de costo.....	XXVII
---	-------



INTRODUCCIÓN

El uso de las Tecnologías de la Información y las Comunicaciones (TIC) en la salud pública constituye una herramienta para mejorar el impacto de las intervenciones en la comunidad. Las mismas, permiten un acceso más equitativo y eficiente a los servicios, lo que mejora la oportunidad de la atención y el costo-efectividad de los tratamientos.

En este sentido, en 1997 la alta dirección del Ministerio de Salud Pública (MINSAP) se planteó desarrollar en el Sistema Nacional de Salud (SNS) una política única, con un modelo integral de informatización a los diferentes niveles del mismo, así como el acceso a la información como proceso que apoye y potencie decisivamente la asistencia médica, la docencia, la investigación, la higiene y la epidemiología, la industria médico farmacéutica, la economía y la administración de salud. El modelo debería extenderse de forma integral a todas las instituciones del país para alcanzar un Sistema Integrado de Gestión de Información y Conocimiento, que será herramienta básica en la materialización de las estrategias y programas de Salud [1].

A partir del año 2003 el MINSAP retoma como una de sus prioridades, el desarrollo de la informatización en todas sus instituciones y áreas de salud con el objetivo de lograr un impacto en la atención a los pacientes, en la optimización de la gestión de la administración y en el uso eficiente de los recursos [1].

La informatización del SNS está dada por el conjunto de métodos, técnicas, procedimientos y actividades gerenciales dirigidas al manejo de la información en salud, la cual comprende la información sobre el estado de salud de la población, la información sobre el conocimiento de las ciencias de la salud y la información en general para la toma de decisiones clínico-epidemiológicas, operativas y estratégicas [2]. Hasta la actualidad se han desarrollado múltiples proyectos orientados al cumplimiento de estos propósitos. Los sistemas automatizados y aplicaciones que se implementan en hospitales, institutos y centros de salud,

actualmente se orientan hacia la informatización de la gestión de pacientes como elemento básico de control para mejorar la atención médica, optimizar el uso del personal, aumentar la calidad de los servicios hospitalarios y disminuir sus costos, con el fin de brindar la información requerida para la actividad gerencial a todos los niveles y la elaboración de reportes estadísticos [3].

Destaca entre los objetivos y principios fundamentales del proceso de informatización en el sector de salud el diseño de aplicaciones orientadas a facilitar la administración y el acceso a los datos administrativos y de los pacientes en las unidades del SNS y la evaluación de resultados y gerencia a los niveles de dirección, cuya meta es mejorar la manera en que se trabaja con el aumento de la eficiencia, la calidad de los datos y el acceso a la información almacenada para garantizar la seguridad y la confidencialidad requeridas para cada caso [4].

Además, para alcanzar la informatización del sector se previó utilizar en su construcción las últimas tecnologías de la información y las comunicaciones para el desarrollo de las aplicaciones, empleando para este fin herramientas basadas en *Software Libre*.

El Hospital Universitario Clínico Quirúrgico (HUCQ) “Lucía Íñiguez Landín” como uno de los engranes de esta gigantesca maquinaria se inserta en el proceso de la informatización de su gestión. El proyecto tiene como objetivo general mejorar la Gestión Hospitalaria y optimizar la capacidad resolutive en sus consultas y servicios a través de la digitalización e integración de los procesos asistenciales y administrativos [5]. Uno de los procesos priorizados es: la gestión de la distribución de medicamentos en su farmacia.

Cada servicio en el hospital genera diariamente un volumen elevado de pedidos de medicamentos a la farmacia del mismo, la cual tiene que revisar que estén cumplimentados los datos establecidos en el “Pedido de Sala”, posteriormente se realizan otros procedimientos para realizar el despacho. Otra de las funciones que ejercen los trabajadores de la farmacia es: Recibir las devoluciones de las salas,

este procedimiento es similar al anterior, la solicitud parte de la sala hacia la farmacia, la misma recepciona los medicamentos en conjunto con el original del “Vale de Entrega o Devolución”, se revisa si los medicamentos están en buen estado para el consumo, de estarlo se recibe la devolución. En ambos procesos se le debe entregar una copia del original del “Vale de Pedido o Devolución” al Departamento de Contabilidad y al servicio que realizó la solicitud.

Cuando la existencia de los medicamentos en la farmacia cumple con la categoría de baja cobertura, provoca la realización de una de las abundantes tareas que poseen los técnicos farmacéuticos: la reposición del Fondo Fijo, que consiste en realizar una solicitud al Almacén de Medicamentos para abastecer los estantes de la farmacia con los medicamentos que están en falta o próximo a ello.

La devolución al área de Almacén en conjunto con los procesos anteriores son algunos de los procesos más engorrosos y reiterativos del cual tienen responsabilidad los trabajadores de las farmacias hospitalarias, esto es provocado por la disminución del Fondo Fijo o en mal estado, vencimiento y retención de los medicamentos y consiste en la realización de una solicitud al mismo mediante el modelo “Transferencia entre Almacenes”

En la actualidad, al Centro se le integran varias especialidades provenientes del Hospital “Vladimir Illich Lenin”, lo cual incrementa la gestión de los procesos de distribución en la farmacia y las siguientes deficiencias:

- Poca integridad de la información
- Poca persistencia de la información
- Falta de agilidad en el proceso de distribución
- Falta de visibilidad de la información

- Poca legibilidad de la información
- Ausencia de análisis estadísticos
- Ineficiente proceso de búsqueda de la información

El centro hospitalario carece de un sistema que brinde almacenamiento, visualización y procesamiento de la información generada. Esta situación provoca que el personal encargado de gestionar estos datos, lo haga de forma manual, lo que genera inconsistencias en el proceder del trabajador y por consiguiente ineficiencia en el proceso.

Todas estas problemáticas dificultan el análisis eficiente de los datos con el fin de acelerar la distribución de los medicamentos, por lo que se enuncia el siguiente **problema de la investigación**: ¿Cómo mejorar el proceso de distribución de medicamentos en la farmacia del Hospital Universitario Clínico Quirúrgico “Lucía Íñiguez Landín”?

El **objeto de la investigación** en el cual se enmarca el problema planteado lo constituye el proceso de distribución de medicamentos en la farmacia del Hospital Universitario Clínico Quirúrgico “Lucía Íñiguez Landín”.

Para dar solución al problema planteado se propone como **objetivo de la investigación**: desarrollar un sistema informático que gestione el proceso de distribución de medicamentos en la farmacia del Hospital Universitario Clínico Quirúrgico “Lucía Íñiguez Landín”.

El objetivo delimita como **campo de acción**: la informatización del proceso de distribución de medicamentos en la farmacia del Hospital Universitario Clínico Quirúrgico “Lucía Íñiguez Landín”.

Para guiar la investigación se trazaron las siguientes **preguntas**:

- ¿Cuáles son los fundamentos teóricos acerca de la gestión del proceso de distribución de medicamentos en las farmacias de los hospitales?
- ¿Cuál es el estado actual de los sistemas de gestión del proceso de distribución de medicamentos en la farmacia del Hospital Universitario Clínico Quirúrgico “Lucía Íñiguez Landín”?
- ¿Cómo diseñar e implementar un producto informático capaz de mejorar el proceso de distribución de medicamentos en la farmacia del Hospital Universitario Clínico Quirúrgico “Lucía Íñiguez Landín”?
- ¿Cómo comprobar que el producto desarrollado sea robusto y fiable?
- ¿Cómo valorar el grado de aceptación del sistema propuesto?

Para dar respuesta a las preguntas anteriormente planteadas se proponen las siguientes **tareas de la investigación**:

1. Realizar un estudio acerca del proceso de distribución de medicamentos en las farmacias de los hospitales en la provincia Holguín.
2. Realizar una revisión bibliográfica que exponga los sistemas de gestión del proceso de distribución de medicamentos en la farmacia del HUCQ “Lucía Íñiguez Landín”.
3. Diseñar el producto informático propuesto.
4. Implementar el producto informático propuesto.
5. Realizar pruebas unitarias al producto informático propuesto.
6. Valorar la efectividad y grado de aceptación de la solución propuesta.



En el desarrollo de la investigación se utilizaron los siguientes **métodos científicos**:

Empíricos:

Entrevista: permitió la recopilación de toda la información necesaria para el desarrollo del proyecto.

Observación científica: se empleó para obtener una concepción más detallada de los procesos a realizar por parte de los trabajadores de la farmacia.

Encuesta: dio a conocer el grado de aceptación de los usuarios al utilizar el sistema.

Criterio de expertos: garantizó que el sistema cumpliera con los requerimientos funcionales y no funcionales a partir de las necesidades planteadas.

Teóricos:

Análisis y síntesis: se utilizó para conocer los sujetos que intervienen en el proceso y descubrir sus roles y características.

Enfoque sistémico: se utilizó para estructurar la base de datos, dependiendo de la jerarquía de los componentes y la relaciones entre ellos.

Modelación: se empleó para visualizar el flujo de los medicamentos entre los sujetos que intervienen en el proceso.

El documento de la investigación se ha estructurado en dos capítulos:

Capítulo 1. Marco teórico. Este capítulo está orientado a exponer los principales fundamentos teóricos correspondientes al objeto de la investigación. Además, se expone la metodología de *software* utilizada, así como las herramientas y tecnologías empleadas para la construcción de la solución propuesta.

Capítulo 2. Diseño y construcción de la solución propuesta. En este capítulo se detalla el uso de la metodología de desarrollo de *software* propuesta.



CAPÍTULO 1. MARCO TEÓRICO

En el presente capítulo, se expondrán las particularidades de la distribución de medicamentos en las farmacias hospitalarias y el Sistema de Distribución de Medicamentos que utilizan, se abordará también acerca de la entidad encargada de la comercialización y distribución mayorista, así como de la Farmacia Hospitalaria y del HUCQ “Lucía Íñiguez Landín” centro donde se desarrollará la investigación. Además, se estudiarán las tecnologías y herramientas más adecuadas para la implementación de la solución propuesta.

1.1 Sistema de Distribución de Medicamentos por Dosis Unitaria

El Sistema de Distribución de Medicamentos por Dosis Unitaria (SDMDU) es un método moderno de distribución, dispensación y control intrahospitalario de medicamentos coordinado por farmacéuticos cuyo objetivo principal es individualizar las dosis que se dispensan a los pacientes ingresados de acuerdo a las órdenes prescritas por el médico. Las solicitudes de medicamentos en este caso son individuales y por un modelo [6].

Este sistema de distribución de medicamentos ofrece la mejor oportunidad para efectuar un adecuado seguimiento farmacoterapéutico del paciente. Esto permite una oportuna intervención antes de la administración del medicamento al paciente, siendo el sistema más seguro para el paciente, el más eficiente desde el punto de vista económico y el método que utiliza efectivamente los recursos profesionales [7].

Este sistema tiene tres elementos básicos:

1. Los medicamentos deben estar contenidos en envases unitarios y permitir su administración al enfermo sin manipulación previa.

2. No deben ser despachadas más dosis de las necesarias para 24 horas.
3. En la Farmacia debe existir el perfil fármaco terapéutico que permita el seguimiento de la terapéutica medicamentosa de cada paciente.

El objetivo general del SDMDU es mantener disponible los medicamentos prescritos por el médico, en el lugar apropiado y en forma oportuna para su administración al paciente en las dosis y formas farmacéuticas prescritas.

1.2 Empresa Comercializadora y Distribuidora de Medicamentos

La Empresa Comercializadora y Distribuidora de Medicamentos (EMCOMED), es la entidad encargada de almacenar, distribuir y comercializar de forma mayorista a las entidades del grupo de las Industrias Biotecnológica y Farmacéuticas y a las entidades del sector de la salud, medicamentos de producción nacional e importados en pesos cubanos en todo el país; así como prestación de los siguientes servicios: transportación de cargas por vía automotor; frigorífico, solo para productos médico farmacéuticos a las entidades del Sistema Nacional de Salud y a otras entidades y re envase de materias primas para uso dispensarial en pesos cubanos [8].

1.3 Hospital Universitario Clínico Quirúrgico

El Hospital General Clínico Quirúrgico "Lucía Íñiguez Landín", lleva este nombre en honor a la madre del Mayor General Calixto García Íñiguez, patriota y luchadora por la libertad de Cuba.

Ubicado en la provincia de Holguín, posee como misión garantizar el mejoramiento de la salud de la población adulta del territorio a través del perfeccionamiento constante de la calidad de la atención médica integral e integrada, la preparación científico-técnica, profesional y profesoral de sus trabajadores y la aplicación

consecuente de lo más avanzado en la tecnología médica y no médica, con una utilización eficiente de los recursos materiales y financieros, ha logrado el más alto grado de satisfacción de nuestra población [9].

La construcción de esta excelente institución se inició en enero de 1983, para responder a las necesidades de la asistencia médica de una población que superaba el 1 000 000 de habitantes, así como el desarrollo de la docencia y las especialidades, interrumpida producto a las afectaciones inducidas por las restricciones económicas originadas por el recrudecimiento del bloqueo impuesto por Estados Unidos y el período especial, a mediados del año 1991 se paralizó la obra. La misma se reanuda en septiembre de 1996, después de que el comandante en jefe en el acto realizado en la plaza Mayor General Calixto García Íñiguez, con motivo del 43 Aniversario de los asaltos a los cuarteles Moncada y Carlos Manuel de Céspedes, da la buena noticia que: junto a otros hospitales, que también se vieron afectados, se reanudaría la obra constructiva “...como un premio y como estímulo a los holguineros...”

1.4 Farmacia

Farmacia Hospitalaria

En el ámbito hospitalario, por las características propias de los pacientes (gravedad, terapias riesgosas, procedimientos especiales, polifarmacia, permanencia controlada) la Farmacia Clínica ha de desempeñar un papel fundamental en la búsqueda de la seguridad, eficacia y eficiencia de las terapias medicamentosas.

Por la complejidad de esta actividad y el rigor científico que implica ha de ser asumida por personal farmacéutico profesional [6].

Esta nueva rama comprende una serie de servicios, entre los que se encuentran:

- Supervisión directa de las terapias medicamentosas en los pacientes

- Fármaco vigilancia.
- Servicio de Información de Medicamentos
- Preparación de mezclas intravenosa.
- Sistemas de distribución de medicamentos.
- Programas de asesoramiento, orientación y educación a médicos, personal de enfermería y pacientes
- Monitorización de fármacos
- Detección y minimización de errores de medicación (EM).

ÁREAS FUNDAMENTALES

Almacén

- Área de Recepción
- Área de Almacenamiento
- Área de Re envase

Área de Dispensación (Área de Despacho)

- Área de Información e Investigación
- Área de Farmacia Clínica
- Área de Ensayos Clínicos

Laboratorio de Producción (Dispensario o Área de Farmacotecnia)

- Área Aséptica
- Área de Fregado y Esterilización
- Área de Producción de Formulaciones Dispensariales
- Área de Producción de Medicina Natural y Tradicional
- Área de Control de la Calidad

- Área de Preparación de Mezclas Intravenosas.
- Área administrativa.

1.4.1 Distribución de medicamentos en los hospitales

Despacho a Salas y Departamentos

En la farmacia se reciben con la periodicidad establecida los Pedidos de Sala, a los mismos se les revisa que tengan los datos establecidos, se folian en orden consecutivo anual y se procede a realizar el despacho.

Al momento de efectuar el despacho de cada medicamento se realizan las anotaciones que correspondan en el modelo SC-2-14 “Tarjeta de Estiba”, en este caso por las salidas, se firma por la persona que realizó la operación y se anota en el “Pedido de Sala” el saldo resultante, posteriormente se archiva por orden consecutivo [10].

En el momento de entregar los medicamentos a la persona encargada de llevarlos a la sala, se revisa por otro técnico de la farmacia que no sea el que preparó el despacho y el “Pedido de Sala” se firma por el último técnico y por la persona que lo recepciona.

El original del “Pedido de Sala” se completa con el precio y el cálculo del importe por medicamento, así como la suma del importe total, una vez realizadas éstas operaciones se envía al área de Contabilidad.

En Contabilidad se pasan las salidas al submayor de inventario por medicamento, momento en que se concilia el saldo de la farmacia con el de Contabilidad. De surgir alguna diferencia en el día, se investiga hasta obtener el cuadro y se procede a depurar la responsabilidad [6].

Devolución de Salas y Departamentos

Las devoluciones de medicamentos de las salas se producen entre otros por: cambios de tratamientos y egresos.

Estas devoluciones se realizan mediante el modelo SC- 2-08 “Vale de Entrega o Devolución”, emitido en la Sala en original y copia, donde se expresa la causa de la devolución. La copia se archiva en la Sala después de ser firmada como recibido por la farmacia.

En la farmacia se reciben los medicamentos conjuntamente con el original del "Vale de Devolución", se le practica la revisión técnica que corresponde con el objeto de depurar aquellos medicamentos no aptos para el consumo por los que se emite el modelo SC-2-16 “Ajuste de Inventario” por el concepto de “Devolución de Sala no Apto para el Consumo”.

A los medicamentos que cumplan todos los requisitos técnicos para poder utilizarse, se le confecciona un modelo SC-2-08 “Vale de Devolución” en original y copias, con el cual se le da entrada en la “Tarjeta de Estiba”.

Los originales del “Vale de Entrega o Devolución” y “Ajuste de Inventario” emitidos por la farmacia una vez completados, se envían al área de Contabilidad donde se realizan los ajustes en la cuenta de gastos y en el centro de costo correspondiente [6].

Reposición

Con la periodicidad que se establezca se procederá a efectuar la reposición del Fondo Fijo de la farmacia mediante el siguiente procedimiento:

Primeramente se cuadra el saldo que muestra el modelo SC-2-14 “Tarjeta de Estiba” con las existencias físicas de cada medicamento; en caso de existir diferencias entre el saldo de la tarjeta y el físico, se revisan las anotaciones y operaciones a partir de la última reposición con el objeto de encontrar el error, de persistir el faltante o

sobrante en el cuadro se efectúa el ajuste correspondiente a fin de llevar el saldo de la tarjeta al físico real, se emite el modelo SC-2-16 “Ajuste de Inventario” y se establecen las responsabilidades pertinentes y la aplicación del Decreto Ley 92 de ser procedente.

Después de efectuado el cuadro del saldo de la “Tarjeta de Estiba” con las existencias reales se emite el modelo SC-2-07 “Solicitud de Entrega” donde se refleja la cantidad que resulte al restar del monto del Fondo Fijo y el saldo en tarjeta.

Posteriormente la farmacia recibe los medicamentos del almacén conjuntamente con el triplicado del modelo SC-2-09 “Transferencia entre Almacenes”, se revisa y se coloca en los anaqueles y (o) gavetas los medicamentos recibidos, posteriormente se actualiza el modelo SC-2-14 “Tarjeta de Estiba”.

El último día hábil de cada mes debe realizarse una reposición total, se debe hacer la solicitud con el tiempo necesario para que el almacén emita la transferencia dentro del mes en curso [6].

Devolución al Área de Almacén

La devolución al almacén se produce por:

- Disminución del Fondo Fijo
- Mal estado, retenidos y vencimiento.

En los dos casos se trasladan al almacén mediante el modelo SC-2-09 “Transferencia entre Almacenes” donde se indica con letra mayúscula el motivo de la devolución. El original de la Transferencia se envía al Área de Contabilidad [6].

1.5 Gestión de información

Definiciones

Woodman plantea en 1985 que la gestión de información es todo lo que tiene que ver con obtener la información correcta, en la forma adecuada, para la persona

indicada, al costo correcto, en el momento oportuno, en el lugar indicado para tomar la acción precisa.

White la denomina en 1985 como la coordinación eficiente y eficaz de la información procedente de fuentes internas y externas.

Proceso mediatizado por un conjunto de actividades que permiten la obtención de información, lo más pertinente, relevante y económica posible, para ser usada en el desarrollo y el éxito de una organización [11].

Objetivos

La gestión de información tiene el objetivo de:

- Maximizar el valor y los beneficios derivados del uso de la información.
- Minimizar el costo de adquisición, procesamiento y uso de la información.
- Determinar responsabilidades para el uso efectivo, eficiente y económico de información.
- Asegurar un suministro continuo de la información.

Sistema de Gestión de Información

Un Sistema de Gestión de Información permite la gestión de los recursos de información tanto internos como externos. Su finalidad es generar servicios y productos que respondan a las necesidades y sobrepasen las expectativas de los usuarios, posibilitando que el sistema trabaje eficientemente y económicamente a la vez. El Sistema de Gestión de Información aprovecha al máximo sus recursos de información en función de la mejora continua y de la toma de decisiones organizacional a todos los niveles jerárquicos desde la cúspide estratégica hasta la base operativa [11].

1.6 Tecnologías y herramientas utilizadas

A través del tiempo tanto las tecnologías como las herramientas han reducido barreras para la realización de los procesos, al escoger las adecuadas, las mismas tienen la capacidad de incrementar la calidad, economizar los tiempos y facilitar el trabajo a quienes las utilizan. A continuación, se analizan las principales características de las empleadas en el desarrollo de este proyecto.

1.6.1 Arquitectura cliente servidor

La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se dividen en dos partes claramente diferenciadas, la primera es el servidor y la segunda un conjunto de clientes. Normalmente el servidor es una máquina bastante potente que actúa de depósito de datos y funciona como un sistema gestor de base de datos (SGBD). Por otro lado, los clientes suelen ser estaciones de trabajo que solicitan varios servicios al servidor. Ambas partes deben estar conectadas entre sí mediante una red [12].

A continuación, una representación gráfica de la arquitectura cliente-servidor:



Ilustración 1: Arquitectura cliente servidor

La arquitectura en cuestión depende básicamente de tres tipos de software para su correcto funcionamiento [13]:

- *Software* de gestión de datos: Encargado de la manipulación y gestión de los datos almacenados y requeridos por las diferentes aplicaciones. Normalmente se aloja en el servidor.
- *Software* de administración de recursos: Se encarga de la organización de los datos necesarios (almacenamiento, indexación y recuperación) para apoyar al *software* que se encarga de la lógica de la aplicación.
- *Software* de interacción con los usuarios: Reside en los clientes y es la aplicación gráfica de los usuarios para la manipulación de datos, permite enviar las operaciones y presentar de forma adecuada los resultados de una aplicación.

Esta arquitectura presenta varias ventajas con respecto a la tradicional no distribuida:

- Centralización del control: Los accesos y los recursos están controlados por el servidor evitando la duplicidad y la pérdida de coherencia en la información.
- Los clientes pueden ejecutarse en ordenadores menos potentes: Al estar centralizada la información en el servidor, los clientes solo se limitan a solicitar servicios mediante el navegador.
- Se separan las distintas partes de la aplicación favoreciendo la reutilización y facilitando el mantenimiento: Al estar distribuidas las funciones entre un conjunto de ordenadores independientes se hace posible la aplicación de transformaciones al servidor sin afectar a sus clientes.

Aunque la arquitectura cliente-servidor es una de las más aplicables y utilizadas en el mundo actual, también presenta desventajas [14]:

- **Gasto**
El servidor central debe ser lo suficientemente potente como para gestionar y compartir recursos con otros ordenadores de la red.
- **Adición**
Si el servidor centralizado se retira del sistema o cae debido a problemas, toda la red no puede funcionar.
- **Congestión**
Los servidores centralizados tienen que manejar la mayor parte del tráfico de la red, como todas las consultas de los recursos se dirigen al servidor. Esto puede causar congestión en la red y tiempos de respuesta lentos para cada equipo disponible.
- **Mantenimiento**
Las redes que poseen la arquitectura cliente-servidor a menudo requieren un personal con al menos un administrador de red única para gestionar y mantener el equipo y la red.

1.6.2 Lenguaje de programación Python

Un lenguaje de programación es una forma de comunicación formal diseñado para realizar procesos que pueden ser llevados a cabo por máquinas como las computadoras. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones.

Entre los lenguajes más populares existente en la actualidad se encuentra *Python*, un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintáxis de *Python* y su tipado dinámico, junto con su naturaleza interpretada, hacen de este un lenguaje ideal para *scripting* y desarrollo rápido de aplicaciones en diversas áreas y plataformas [15]. Sin dudas el popular lenguaje de programación es una apuesta a la simplicidad, versatilidad y rapidez de desarrollo. A continuación, algunas de las características [16]:

- **Propósito general**

Se pueden crear todo tipo de programas. No es un lenguaje creado específicamente para la *web*, aunque entre sus posibilidades sí se encuentra el desarrollo de páginas.

- **Multipataforma**

Hay versiones disponibles de *Python* en muchos sistemas informáticos distintos. Originalmente se desarrolló para *Unix*, aunque cualquier sistema es compatible con el lenguaje siempre y cuando exista un intérprete programado para él.

- **Interpretado**

Quiere decir que no se debe compilar el código antes de su ejecución. En realidad, se realiza una compilación; pero de manera transparente para el programador. En ciertos casos, cuando se ejecuta por primera vez un código, se producen unos *bytecodes* que se guardan en el sistema y que sirven para acelerar la compilación implícita que realiza el intérprete cada vez que se ejecuta el mismo código.

- **Interactivo**

Python dispone de un intérprete por línea de comandos en el que se pueden introducir sentencias. Cada sentencia se ejecuta y produce un resultado visible, que puede ayudarnos a entender mejor el lenguaje y probar los resultados de la ejecución de porciones de código rápidamente.

- **Orientado a Objetos**

La programación orientada a objetos está soportada en *Python* y ofrece en muchos casos una manera sencilla de crear programas con componentes reutilizables.

- **Funciones y librerías**

Dispone de muchas funciones incorporadas en el propio lenguaje, para el tratamiento de *strings*, números, archivos, etc. Además, existen muchas librerías que podemos importar en los programas para tratar temas específicos como la programación de ventanas o sistemas en red o cosas tan interesantes como crear archivos comprimidos en .zip.

- **Sintaxis clara**

Por último, destacar que *Python* tiene una sintaxis muy visual, gracias a una notación indentada (con márgenes) de obligado cumplimiento. En muchos lenguajes, para separar porciones de código, se utilizan elementos como las llaves o las palabras clave *begin* y *end*. Para separar las porciones de código en *Python* se debe tabular hacia dentro, colocando un margen al código que iría dentro de una función o un bucle. Esto ayuda a que todos los programadores adopten unas mismas notaciones y que los programas de cualquier persona tengan un aspecto muy similar.

1.6.3 Lenguaje de programación JavaScript

JavaScript es un lenguaje de programación de *scripts* (secuencia de comandos), actualmente es utilizado principalmente en *internet* junto con las páginas *web* (*HTML* o *XHTML*), el mismo se incluye en la página *web* o en un archivo externo y mejora un sitio *HTML*, al ofrecer interacciones del usuario, animaciones, ayudas a la navegación tales como:

- Mostrar / ocultar texto
- Deslizamiento de imágenes
- Crear presentaciones de diapositivas
- Crear burbujas de información

JavaScript es un lenguaje de programación interpretado; por tanto, el código fuente se mantiene sin cambios y para ejecutar este código se debe proporcionar un intérprete que sea capaz de leer y realizar las acciones solicitadas. Cada navegador posee en su interior un intérprete *JavaScript*, que varía en función del navegador. Al navegar con *Internet Explorer*, se utiliza el intérprete *JScript*, en *Mozilla Firefox* se llama *SpiderMonkey* y el motor *V8* es el de *Google Chrome*. Además, es un lenguaje orientado a objetos, característica que favorece al desarrollador porque posee la capacidad de brindar una programación más lógica e intuitiva para así obtener un código fuente más claro. Aunque *JavaScript* fue diseñado para ser usado en conjunción con *HTML*, el lenguaje ha evolucionado desde entonces hacia otros destinos. El mismo es regularmente utilizado para hacer extensiones para diferentes programas, como los *scripts* codificados en *Lua* o *Python* [17].

En cuanto a limitaciones *JavaScript* fue diseñado de forma que se ejecutara en un entorno muy limitado que permitiera a los usuarios confiar en la ejecución de los *scripts*. A continuación, algunas de sus limitaciones [18]:

- Los *scripts* no pueden comunicarse con recursos que no pertenezcan al mismo dominio desde el que se descargó el *script*.
- Los *scripts* no pueden cerrar ventanas que no hayan abierto esos mismos *scripts*.
- Las ventanas que se crean no pueden ser demasiado pequeñas ni demasiado grandes, ni colocarse fuera de la vista del usuario.
- No pueden acceder a los archivos del ordenador del usuario, ni leer o modificar las preferencias del navegador.
- Si la ejecución de un script dura demasiado tiempo, el navegador informa al usuario de que un script está consumiendo demasiados recursos y le da la posibilidad de detener la ejecución.

1.6.4 Framework de desarrollo web Django

Django es un *framework web* de alto nivel, basado en el paradigma Modelo-Vista-Controlador y en la filosofía *DRY (Don't Repeat Yourself: No te repitas)* que con ayuda de su lenguaje nativo *Python* y su Base de Datos embebida *SQLite* logra la sencillez, la rapidez y la reutilización de código en todo su marco de desarrollo. Es gratuito y de código abierto, posee una comunidad próspera y activa, amplia documentación y varias opciones de soporte gratuito y de pago [19].

A continuación algunas de sus características [20]:

- **Completo**

Django sigue la filosofía "Baterías incluidas" y provee casi todo lo que los desarrolladores quisieran hacer "de fábrica". Porque todo lo que necesitas es parte de un único "producto", todo funciona a la perfección, sigue principios de diseño consistentes y tiene una amplia y actualizada documentación.

- **Versátil**

Django puede ser (y ha sido) usado para construir casi cualquier tipo de sitio *web*, desde sistemas manejadores de contenidos y *wikis*, desde redes sociales a sitios nuevos. Puede funcionar con cualquier *framework* cliente-servidor, y puede devolver contenido en casi cualquier formato (incluyendo *HTML*, *RSS feeds*, *JSON*, *XML*, etc).

También se puede extender internamente para usar otros componentes si es necesario, si bien ofrece opciones para casi cualquier funcionalidad que desee (varios motores de base de datos, motores de plantillas, etc.).

- **Seguro**

Django ayuda a los desarrolladores a evitar varios errores comunes de seguridad al proveer un *framework* que ha sido diseñado para "hacer lo correcto" para proteger el sitio *web* automáticamente. Por ejemplo, *Django*, proporciona una manera segura de administrar cuentas de usuario y contraseñas, evitando así errores comunes como colocar la información de la sesión en *cookies* donde es vulnerable (en lugar de eso las *cookies* solo contienen una clave y los datos se almacenan en la base de datos) o se almacenan directamente las contraseñas en un *hash* de contraseñas.

Un *hash* de contraseña es un valor de longitud fija creado al enviar la contraseña a una *cryptographic hash function*. *Django* puede validar si la contraseña ingresada es correcta enviándola a través de una función *hash* y comparando la salida con el valor *hash* almacenado. Sin embargo, debido a la naturaleza "unidireccional" de la función, incluso si un valor *hash* almacenado se ve comprometido es difícil para un atacante resolver la contraseña original.

Django permite protección contra algunas vulnerabilidades de forma predeterminada, incluida la inyección *SQL*, *scripts* entre sitios, falsificación de solicitudes entre sitios y *clickjacking*.

- **Escalable**

Django usa un componente basado en la arquitectura “*shared-nothing*” (cada parte de la arquitectura es independiente de las otras, y por lo tanto puede ser reemplazado o cambiado si es necesario). Teniendo en cuenta una clara separación entre las diferentes partes significa que puede escalar para aumentar el tráfico al agregar *hardware* en cualquier nivel: servidores de *caché*, servidores de bases de datos o servidores de aplicación. Algunos de los sitios más concurridos han escalado a *Django* para satisfacer sus demandas (por ejemplo, *Instagram* y *Disqus*).

- **Mantenible**

El código de *Django* está escrito usando principios y patrones de diseño para fomentar la creación de código mantenible y reutilizable. En particular, utiliza el principio “No te repitas” para que no exista una duplicación innecesaria, reduciendo la cantidad de código. *Django* también promueve la agrupación de la funcionalidad relacionada en “aplicaciones” reutilizables y en un nivel más bajo, agrupa código relacionado en módulos (siguiendo el patrón *Model View Controller (MVC)*).

- **Portable**

Django está escrito en *Python*, el cual se ejecuta en muchas plataformas. Lo que significa que no está sujeto a ninguna plataforma en particular, y puede ejecutar sus aplicaciones en muchas distribuciones de *Linux*, *Windows* y *Mac OS X*. Además, *Django* cuenta con el respaldo de muchos proveedores de alojamiento *web*, y que a menudo proporcionan una infraestructura específica y documentación para el alojamiento de sitios de *Django*.

1.6.5 Editor de Texto Visual Studio Code

Un editor de código fuente es un editor de texto diseñado específicamente para editar el código fuente de programas informáticos, permite crear y modificar archivos digitales compuestos únicamente por textos sin formato, conocidos también como texto plano. Los mismos tienen características diseñadas para simplificar y acelerar la escritura de código fuente, como resaltado de sintaxis, autocompletar y pareo de llaves. Estos editores proveen un modo conveniente de ejecutar un compilador, intérprete, depurador o cualquier otro programa que sea relevante en el proceso de desarrollo de *software* [21].

Visual Studio Code es una herramienta desarrollada por *Microsoft* que combina en una única plataforma un editor de texto ligero y sencillo que implementa todas las funciones, características y complementos necesarios para adaptarse de la mejor forma a las necesidades de cualquier programador. Algunas de ellas son el ofrecimiento de un soporte para diseño de páginas *web*, con editores avanzados de *HTML*, *CSS* y *JavaScript*, capacidad de depuración, ayuda contextual para varios lenguajes, validadores de código y accesibilidad. Es un *software* gratuito que soporta una cantidad considerable de lenguajes, ya sean propio de *Microsoft* como *C#*, *F#* y *Visual Basic*, o de otros como *PHP*, *Python*, *Perl*, *SQL*, *Shell scripting* en *Bash* y *Java* [22].

1.6.6 Gestor de base de datos PostgreSQL

Un Sistema de Gestión de Bases de Datos (SGBD), es un conjunto estructurado de datos que representa entidades y sus interrelaciones, que le permite a varios usuarios de forma simultánea el almacenamiento, modificación y extracción de la información en una base de datos de forma más eficiente y segura [23].

Es un servidor de base de datos objeto relacional libre, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de

datos, funciones, restricciones, disparadores, reglas e integridad transaccional, liberado bajo la licencia *Berkeley Software Distribution (BSD)*. Como muchos otros proyectos *open source*, el desarrollo de *PostgreSQL* no es manejado por una sola compañía, sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo, dicha comunidad es denominada el *PGDG (PostgreSQL Global Development Group)*. El mismo es muy conocido y usado en entornos de software libre porque cumple los estándares *SQL92* y *SQL99*, y también por el conjunto de funcionalidades avanzadas que soporta [24].

A continuación, algunas de las características más importantes [25]:

Prestaciones

PostgreSQL destaca por su amplia lista de prestaciones que lo hacen capaz de competir con cualquier SGBD comercial, estando aún en mejor nivel que muchos de ellos. A continuación, se muestran algunas de ellas:

- Es una base de datos 100% *ACID (Atomicity, Consistency, Isolation and Durability)*.
- Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (*MAC, IP ...*), cadenas de bits, etc. También permite la creación de tipos propios.
- Incluye herencia entre tablas, por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Copias de seguridad en caliente (*Online/hot backups*)
- *Unicode*
- Juegos de caracteres internacionales
- Regionalización por columna
- *Multi-Version Concurrency Control (MVCC)*

- Múltiples métodos de autenticación
- Acceso encriptado vía *SSL*
- *SE-postgres*
- Completa documentación
- Licencia *BSD*
- Disponible para *Linux* y *UNIX* en todas sus *variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64)* y *Windows 32/64bit*.
- Fácil de Administrar.
- Capacidades de replicación de datos.
- Soporte empresarial disponible.

1.6.7 Servidor web Apache

Un programa que implementa el protocolo *HTTP (hypertext transfer protocol)*. Este protocolo está diseñado para transferir hipertextos, páginas *HTML (hypertext markup language)*: textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos.

El servidor *HTTP Apache* es un servidor *web software* libre desarrollado por la *Apache Software Foundation (ASF)*. El producto obtenido de este proyecto es un servidor de código fuente completo, descargable, gratuito, robusto y con un ciclo de desarrollo rápido gracias a la gran cantidad de colaboradores voluntarios de que dispone.

A continuación otras de las características generales del servidor [26]:

- Cumple el estándar *HTTP/1.1*
- Configurable y extensible mediante módulos
- Proporciona el código fuente con licencia sin restricciones
- Multiplataforma
- Continua actualización y desarrollo

- Base de datos *DBM* para autenticación de usuarios
- Permite crear páginas *web* protegidas
- Permite personalizar las respuestas a errores y problemas del servidor
- Múltiples directivas *DirectoryIndex*
- Sin límite fijo sobre el número de Alias y redirecciones
- Negociación de contenidos
- *Hosts* virtuales, también llamados servidores multi-alojados
- Archivos *logs* configurables

1.6.8 Metodología de desarrollo de software XP

Manifiesto Ágil

En febrero de 2001, convencidos de que era necesario un cambio en las metodologías “clásicas” de desarrollo de *software*, se reunió en las montañas de *Wasatch* de *Utah* un grupo de 17 desarrolladores. Entre ellos se encontraban los creadores de *XP*, *SCRUM*, *DSDM*, *Adaptive Software Development*, *Crystal*, *Feature-Driven Development* y *Pragmatic Programming*. Juntos proclamaron lo que se ha dado a conocer como el “*Manifiesto for Agile Software Development*” [27].

Este manifiesto se basa en los siguientes principios:

- Satisfacer al cliente a través de entregas continuas y tempranas es la mayor prioridad.
- Cambios a los requerimientos en fases tardías del desarrollo.
- Entregar frecuentemente una versión funcional del *software*, en períodos desde un par de semanas a un par de meses, prefiriendo los periodos más cortos.
- Desarrolladores, gerentes y clientes deben trabajar juntos diariamente, a lo largo del proyecto.

- Construir proyectos alrededor de personas motivadas, dándoles el entorno y soporte que necesitan, y confiando en que realizarán el trabajo.
- El método más eficiente y efectivo de transmitir información entre un equipo de desarrolladores es la conversación frontal (cara a cara).
- Tener *software* que funciona es la medida primaria del progreso.
- El proceso ágil promueve el desarrollo sostenible. Los patrocinadores, desarrolladores y usuarios deben ser capaces de mantener un ritmo de trabajo constante en forma permanente a lo largo del proyecto.
- La atención continua a la excelencia técnica y el buen diseño mejoran la agilidad.
- Simplicidad – el arte de maximizar el trabajo que no se debe hacer – es esencial.
- Las mejores arquitecturas, requerimientos y diseños surgen de los equipos auto-organizados.
- A intervalos regulares, el equipo debe reflexionar sobre cómo ser más efectivos, y ajustar su comportamiento de acuerdo a ello.

Extreme Programming (XP)

La metodología de desarrollo de *software XP* surge como una nueva manera de encarar proyectos de *software*, proponiendo una metodología basada esencialmente en la simplicidad y agilidad. Las metodologías de *software* tradicionales (ciclo de vida en cascada, evolutivo, en espiral, iterativo, etc.) frecuentemente se catalogan como pesadas y poco eficientes, a diferencia de las ágiles que buscan un punto medio entre la ausencia de procesos y el abuso de los mismos, proponiendo un proceso cuyo esfuerzo sea viable [27].

La metodología ágil *XP* cambia significativamente algunas de las premisas de las llamadas metodologías clásicas [28]:

- Es adaptable en lugar de predictivo. Los métodos clásicos tienden a intentar planear una gran parte del proceso del *software* manifestando su naturaleza de resistencia al cambio. En cambio, para *XP* es todo lo contrario, el cambio es bienvenido. Intenta ser un proceso que se adapta y crece en el cambio.
- Es orientado a los sujetos que intervienen y no al proceso. El objetivo de los métodos clásicos es definir un proceso que funcionará bien independientemente de quien lo utilice. El método *XP* afirma que ningún proceso nunca podrá disimular las habilidades del equipo de desarrollo, el papel del proceso es apoyar al equipo de desarrollo en su trabajo.
- Se programa en parejas, se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que de esta manera el código presente mayor calidad, ya que el mismo es revisado y discutido mientras se escribe.
- Simplicidad en el código, la programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado que quizás nunca se utilice. La simplicidad y la comunicación son complementarias, con más comunicación resulta más fácil identificar que se debe y qué no se debe hacer.
- Propiedad del código compartida, en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores sean detectados.

1.7 Conclusiones parciales

En este capítulo se han analizado los principales fundamentos teóricos que sustentan esta investigación, luego de realizar este proceso se puede arribar a las siguientes conclusiones:

1. La fundamentación teórica de la investigación evidencia la importancia de la implementación de un sistema *web* que emule y perfeccione los procesos de gestión de la distribución de los medicamentos y la seguridad y pertinencia de los procesos informatizados.
2. Las herramientas y tecnologías de *software* libre constituyen una solución viable para el desarrollo exitoso de un sistema informático y su posterior persistencia en el tiempo.
3. La metodología de desarrollo *XP* resulta una de las más adecuadas para el desarrollo de un sistema informático que presenta como limitante un grupo pequeño de desarrolladores y requerimientos funcionales cambiantes.

CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

En el presente capítulo se abordarán los temas que componen la ingeniería de *software* aplicada a la implementación del Sistema, utilizando como plataforma la metodología de desarrollo *XP*, el ciclo de vida de la misma ha servido para estructurar todo el proceso de creación del sistema, al ser tan dinámicas sus fases de implementación, ha brindado la oportunidad de tratar con requisitos imprecisos y cambiantes.

2.1 Exploración

La etapa de captura de requerimientos es la más importante en el desarrollo de un sistema informático, ya que es la fase donde se define el alcance general del proyecto. Se realiza una reunión con el cliente para analizar e identificar los requisitos que debe poseer el sistema a implementar, los mismos se clasifican en funcionales y no funcionales.

2.1.1 Requerimientos Funcionales

Los requisitos funcionales son los que describen la interacción entre el sistema y su ambiente, independientemente de su implementación. El ambiente incluye al usuario y cualquier otro sistema externo que interactúa con el sistema primario.

A continuación, se muestra la lista de los requerimientos funcionales:

RF1: Autenticar usuario

RF2: Gestionar usuario

RF2.1: Registrar usuario

RF2.2: Modificar usuario

RF2.3: Eliminar usuario

RF2.4: Visualizar información de usuario

CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

RF3: Gestionar rol de usuario

RF3.1: Asignar rol de usuario

RF3.2: Modificar rol a usuario

RF3.3: Visualizar información de un rol

RF4. Gestionar información de sala

RF4.1: Agregar sala

RF4.2: Editar sala

RF4.3: Eliminar sala

RF4.4: Visualizar información de sala

RF5. Gestionar información de cama

RF5.1: Agregar cama

RF5.2: Editar cama

RF5.3: Eliminar cama

RF5.4: Visualizar información de cama

RF6. Gestionar solicitudes de cama

RF6.1: Agregar pedido de cama

RF6.2: Editar pedido de cama

RF6.3: Eliminar pedido de cama

RF6.4: Buscar pedido de cama

RF6.5: Visualizar información de pedido de cama

RF6.6: Solicitar pedido de sala

RF6.7: Agregar devolución de cama

RF6.8: Editar devolución de cama

RF6.9: Eliminar devolución de cama

RF6.10: Buscar devolución de cama

RF6.11: Visualizar información de devolución de cama

RF6.12: Solicitar devolución de sala

RF7. Gestionar despachos de sala



CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

- RF7.1: Despachar pedido de sala
- RF7.2: Eliminar pedido de sala
- RF7.3: Buscar pedido de sala
- RF7.4: Visualizar información de pedido de sala
- RF7.5: Despachar devolución de sala
- RF7.6: Eliminar devolución de sala
- RF7.7: Buscar devolución de sala
- RF7.8: Visualizar información de devolución de sala
- RF8. Gestionar solicitudes de farmacia
 - RF8.1: Agregar pedido de farmacia
 - RF8.2: Visualizar información de pedido de farmacia
 - RF8.3: Solicitar pedido al almacén
 - RF8.4: Agregar devolución de farmacia
 - RF8.5: Editar devolución de farmacia
 - RF8.6: Eliminar devolución de farmacia
 - RF8.7: Visualizar información de devolución de farmacia
 - RF8.8: Solicitar devolución al almacén
- RF9. Gestionar despachos de almacén
 - RF9.1: Despachar pedido al almacén
 - RF9.2: Eliminar pedido al almacén
 - RF9.3: Buscar pedido al almacén
 - RF9.4: Visualizar información de pedido al almacén
 - RF9.5: Despachar devolución al almacén
 - RF9.6: Eliminar devolución al almacén
 - RF9.7: Buscar devolución al almacén
 - RF9.8: Visualizar información de devolución al almacén



RF10. Gestionar existencia de medicamento

RF10.1: Agregar medicamento

RF10.2: Editar medicamento

RF10.3: Eliminar medicamento

RF10.4: Visualizar información de medicamento

RF10.5: Agregar existencia de medicamento

RF10.6: Editar existencia de medicamento

RF10.7: Eliminar existencia de medicamento

RF10.8: Solicitar aumento de existencia en baja cobertura

RF11. Exportar información

RF11.1: Exportar pedido de cama

RF11.2: Exportar despacho de pedido de sala

RF11.3: Exportar devolución de cama

RF11.4: Exportar despacho de devolución de sala

RF11.5: Exportar medicamentos

RF11.6: Exportar solicitud de aumento de existencia en baja cobertura

RF11.7: Exportar salas

RF12. Gestionar medicamentos solicitados por servicio

RF12.1: Buscar medicamentos pedidos y devueltos por centro de costo

RF12.2: Visualizar información de medicamentos pedidos y devueltos por centro de costo

2.1.2 Requerimientos no Funcionales

Los requisitos no funcionales son los que describen aspectos del sistema que son visibles por el usuario y no incluyen una relación directa con el comportamiento funcional del sistema. Generalmente los mismos incluyen restricciones como el tiempo de respuesta, la precisión, recursos consumidos, seguridad, entre otras.

A continuación, se muestra la lista de los requerimientos no funcionales:

CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

Apariencia o Interfaz de Usuario

RNF1: Distribución de interfaces sencillas e intuitivas, permitiendo a personas de poca experiencia, una mayor comodidad a la hora de trabajar con el sistema.

RNF2: Uso de colores agradables.

RNF3: Usar interfaces que se adapten a la resolución de la pantalla del usuario.

Rendimiento

RNF4: Velocidad de procesamiento y tiempo de respuesta acorde con los recursos del servidor donde se encuentra el Sistema.

Usabilidad

RNF5: Interfaz amigable con facilidades para el uso de los usuarios, independientemente sus niveles de conocimiento y habilidades informáticas.

RNF6: Funcionalidades visibles en todo momento que faciliten la navegación.

Portabilidad

RNF7: La aplicación debe trabajar en las plataformas *Windows, Linux y macOS*.

Seguridad

RNF8: La información debe ser actualizada solo por el personal autorizado.

RNF9: La aplicación debe ser utilizada solo por personal autorizado.

Documentación

RNF10: Debe contar con un Manual de Usuario detallado y claro que brinde una adecuada orientación al usuario respecto a las opciones con que cuenta el sistema.

CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

2.1.3 Historias de usuario

Tabla 1 Historia de Usuario No. 1. Autenticar usuario

Historia de Usuario	
No.: 1	Nombre: Autenticar usuario
Usuarios: Todos	
Prioridad de negocio: Alta	Nivel de complejidad: Medio
Estimación: 1 día	Iteración asignada: 1
Descripción: El usuario debe tener la posibilidad de autenticarse en el sistema.	
Información adicional(Observaciones):	

Fuente: elaboración propia

Tabla 2: Historia de Usuario No.2. Gestionar usuario

Historia de Usuario	
No.: 2	Nombre: Gestionar usuario
Usuarios: Administrador	
Prioridad de negocio: Alta	Nivel de complejidad: Medio
Estimación: 2 semanas	Iteración asignada: 2
Descripción: El usuario debe tener la posibilidad de gestionar la información de los otros usuarios en el sistema.	
Información adicional(Observaciones): Gestionar usuario incluye registrar, modificar, eliminar, visualizar.	

Fuente: elaboración propia



CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

Tabla 3: Historia de Usuario No.3. Gestionar rol de usuario

Historia de Usuario	
No.: 3	Nombre: Gestionar rol de usuario
Usuarios: Administrador	
Prioridad de negocio: Alta	Nivel de complejidad: Alto
Estimación: 2 semanas	Iteración asignada: 3
Descripción: El usuario debe tener la posibilidad de gestionar los roles de los otros usuarios en el sistema.	
Información adicional(Observaciones): Gestionar rol de usuario incluye asignar, modificar, visualizar.	

Fuente: elaboración propia

Tabla 4: Historia de Usuario No.4. Gestionar información de sala

Historia de Usuario	
No.: 4	Nombre: Gestionar información de sala
Usuarios: Administrador	
Prioridad de negocio: Alta	Nivel de complejidad: Medio
Estimación: 2 semanas	Iteración asignada: 4
Descripción: El usuario debe tener la posibilidad de gestionar la información de las salas en el sistema.	
Información adicional(Observaciones): Gestionar la información de sala incluye agregar, editar, eliminar, visualizar.	

Fuente: elaboración propia



CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

Tabla 5: Historia de Usuario No.5. Gestionar información de cama

Historia de Usuario	
No.: 5	Nombre: Gestionar información de cama
Usuarios: Secretaria de la sala	
Prioridad de negocio: Alta	Nivel de complejidad: Medio
Estimación: 2 semanas	Iteración asignada: 5
Descripción: El usuario debe tener la posibilidad de gestionar la información de sus camas en el sistema.	
Información adicional(Observaciones): Gestionar la información de cama incluye agregar, editar, eliminar, visualizar.	

Fuente: elaboración propia

Tabla 6: Historia de Usuario No.6. Gestionar solicitudes de cama

Historia de Usuario	
No.: 6	Nombre: Gestionar solicitudes de cama
Usuarios: Secretaria de la sala	
Prioridad de negocio: Alta	Nivel de complejidad: Alto
Estimación: 4 semanas	Iteración asignada: 6
Descripción: El usuario debe tener la posibilidad de gestionar las solicitudes de las camas en el sistema.	
Información adicional(Observaciones): Gestionar solicitudes de cama incluye agregar, editar, eliminar, buscar, visualizar, solicitar.	

Fuente: elaboración propia

CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

Tabla 7: Historia de Usuario No.7. Gestionar despachos de sala

Historia de Usuario	
No.: 7	Nombre: Gestionar despachos de sala
Usuarios: Dpto. de Farmacia	
Prioridad de negocio: Alta	Nivel de complejidad: Alto
Estimación: 3 semanas	Iteración asignada: 7
Descripción: El usuario debe tener la posibilidad de despachar las solicitudes de las salas en el sistema.	
Información adicional(Observaciones): Gestionar despachos de sala incluye despachar, eliminar, buscar, visualizar.	

Fuente: elaboración propia

Tabla 8: Historia de Usuario No.8. Gestionar solicitudes de farmacia

Historia de Usuario	
No.: 8	Nombre: Gestionar solicitudes de farmacia
Usuarios: Jefe de Dpto. de Farmacia	
Prioridad de negocio: Alta	Nivel de complejidad: Alto
Estimación: 4 semanas	Iteración asignada: 8
Descripción: El usuario debe tener la posibilidad de gestionar las solicitudes de la farmacia en el sistema.	
Información adicional(Observaciones): Gestionar solicitudes de farmacia incluye agregar, editar, eliminar, visualizar, solicitar.	

Fuente: elaboración propia



CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

Tabla 9: Historia de Usuario No.9. Gestionar despachos de almacén

Historia de Usuario	
No.: 9	Nombre: Gestionar despachos de almacén
Usuarios: Almacén de medicamentos	
Prioridad de negocio: Alta	Nivel de complejidad: Alto
Estimación: 3 semanas	Iteración asignada: 9
Descripción: El usuario debe tener la posibilidad de despachar las solicitudes de la farmacia en el sistema.	
Información adicional(Observaciones): Gestionar despachos de almacén incluye despachar, eliminar, buscar, visualizar.	

Fuente: elaboración propia

Tabla 10: Historia de Usuario No.10. Gestionar existencia de medicamento

Historia de Usuario	
No.: 10	Nombre: Gestionar existencia de medicamento
Usuarios: Jefe de Dpto. de Farmacia, Jefe de Dpto. de Contabilidad	
Prioridad de negocio: Alta	Nivel de complejidad: Alto
Estimación: 3 semanas	Iteración asignada: 10
Descripción: El usuario debe tener la posibilidad de gestionar la existencia de los medicamentos en el sistema.	
Información adicional(Observaciones): Gestionar existencia de medicamento incluye agregar, editar, eliminar, visualizar, buscar.	

Fuente: elaboración propia



CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

Tabla 11: Historia de Usuario No.11. Exportar información

Historia de Usuario	
No.: 11	Nombre: Exportar información
Usuarios: Todos	
Prioridad de negocio: Alta	Nivel de complejidad: Alto
Estimación: 3 semanas	Iteración asignada: 11
Descripción: El usuario debe tener la posibilidad de exportar información en el sistema.	
Información adicional(Observaciones):	

Fuente: elaboración propia

Tabla 12: Historia de Usuario No.12. Gestionar medicamentos solicitados por servicio

Historia de Usuario	
No.: 12	Nombre: Gestionar medicamentos solicitados por servicio
Usuarios: Dpto. de Contabilidad	
Prioridad de negocio: Alta	Nivel de complejidad: Alto
Estimación: 1 semana	Iteración asignada: 12
Descripción: El usuario debe tener la posibilidad de gestionar los medicamentos solicitados en el sistema.	
Información adicional(Observaciones): Gestionar datos estadísticos incluye buscar, visualizar.	



CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

Fuente: elaboración propia

2.2 Planificación

La planificación es una fase corta, en la que el cliente, los gerentes y el grupo de desarrollo acuerdan el orden en que deberán implementarse las historias de usuario y asociadas a éstas, las entregas.

2.2.1 Plan de iteraciones

Tabla 13: Distribución de las historias de usuario por iteración

Distribución de las historias de usuario por iteración		
Iteraciones	Orden de las historias de usuarios a implementar	Tiempo
1	Autenticar usuario	1 día
2	Gestionar usuario	2 semanas
3	Gestionar rol de usuario	2 semanas
4	Gestionar información de sala	2 semanas
5	Gestionar información de cama	2 semanas
6	Gestionar solicitudes de cama	4 semanas
7	Gestionar despachos de sala	4 semanas
8	Gestionar solicitudes de farmacia	4 semanas
9	Gestionar despachos de almacén	4 semanas

CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

10	Gestionar existencia de medicamento	3 semanas
11	Exportar información	3 semanas
12	Gestionar medicamentos solicitados por servicio	1 semana

Fuente: elaboración propia

2.2.2 Plan de entregas

El cronograma de entregas establece que las historias de usuario serán agrupadas para conformar una entrega según el orden de las mismas. El plan de entregas se realiza en base a las estimaciones de tiempos de desarrollo realizadas por los programadores.

Tabla 14: Cronograma de entregas

Plan de entregas						
1ra Iteración	2da Iteración	3ra Iteración	4ta Iteración	5ta Iteración	6ta Iteración	7ma Iteración
7/9/17	18/9/17	29/9/17	12/10/17	25/10/17	6/12/17	5/1/18
8va Iteración	9na Iteración	10ma Iteración	11na Iteración	12da Iteración		
4/2/18	2/3/18	29/3/18	22/4/18	28/4/18		

Fuente: elaboración propia

2.3 Implementación

La metodología *XP* indica que a la hora de desarrollar se debe planificar un ritmo constante y razonable de trabajo, sin sobrecargar al equipo; pero que disminuya las



CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

posibilidades de retraso del proyecto. Al haber un retraso se debe trabajar tiempo extra, algo que desmotiva al grupo e impacta en la calidad del producto.

A continuación, se muestran las iteraciones propuestas:

2.3.1 Iteración 1

El principal objetivo de esta iteración es proporcionarle al sistema una funcionalidad que le permita al usuario autenticarse en el sistema. Para ello se planificó la siguiente tarea:

1. **Tarea 1:** Autenticar usuario

Tabla 15: Tarea 1. Autenticar usuario

Tarea	
No.: 1	Número de HU: 1
Nombre de la tarea: Autenticar usuario	
Tipo de tarea: Desarrollo	Estimación: 1 día
Fecha de inicio: 6 de septiembre de 2017	Fecha de fin: 7 de septiembre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que permita al usuario autenticarse en el sistema.	

Fuente: elaboración propia

2.3.2 Iteración 2

El principal objetivo de esta iteración es proporcionarle al sistema una funcionalidad que le permita al usuario con el rol adecuado la gestión de la información de los otros usuarios. Para ello se planificaron las siguientes tareas:

1. **Tarea 2:** Registrar usuario

CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

2. **Tarea 3:** Modificar usuario
3. **Tarea 4:** Eliminar usuario
4. **Tarea 5:** Visualizar información de usuario

Tabla 16: Tarea 2. Registrar usuario

Tarea	
No.: 2	Número de HU: 2
Nombre de la tarea: Registrar usuario	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 7 de septiembre de 2017	Fecha de fin: 9 de septiembre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que permita al administrador registrar en la base de datos un usuario.	

Fuente: elaboración propia

Tabla 17: Tarea 3. Modificar usuario

Tarea	
No.: 3	Número de HU: 2
Nombre de la tarea: Modificar usuario	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 10 de septiembre de 2017	Fecha de fin: 14 de septiembre de 2017
Programador responsable: Rasel Agüero Fernández	

CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

Descripción: Implementar una vista que permita al administrador modificar en la base de datos un usuario.

Fuente: elaboración propia

Tabla 18: Tarea 4. Eliminar usuario

Tarea	
No.: 4	Número de HU: 2
Nombre de la tarea: Eliminar usuario	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 15 de septiembre de 2017	Fecha de fin: 17 de septiembre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que permita al administrador eliminar de la base de datos un usuario.	

Fuente: elaboración propia

Tabla 19: Tarea 5. Visualizar información de usuario

Tarea	
No.: 5	Número de HU: 2
Nombre de la tarea: Visualizar información de usuario	
Tipo de tarea: Desarrollo	Estimación: 1 día
Fecha de inicio: 17 de septiembre de 2017	Fecha de fin: 18 de septiembre de 2017

CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

Programador responsable: Rasel Agüero Fernández
Descripción: Implementar una vista que permita al administrador visualizar la información de los usuarios en el sistema.

Fuente: elaboración propia

2.3.3 Iteración 3

El principal objetivo de esta iteración es proporcionarle al sistema una funcionalidad que le permita al usuario con el rol adecuado la gestión de los roles de usuario. Para ello se planificaron las siguientes tareas:

1. **Tarea 6:** Asignar rol de usuario
2. **Tarea 7:** Modificar rol a usuario
3. **Tarea 8:** Visualizar información de un rol

Tabla 20: Tarea 6. Asignar rol de usuario

Tarea	
No.: 6	Número de HU: 3
Nombre de la tarea: Asignar rol de usuario	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 19 de septiembre de 2017	Fecha de fin: 21 de septiembre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al administrador asignarle un rol a un usuario.	

Fuente: elaboración propia



CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

Tabla 21: Tarea 7. Modificar rol a usuario

Tarea	
No.: 7	Número de HU: 3
Nombre de la tarea: Modificar rol a usuario	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 22 de septiembre de 2017	Fecha de fin: 26 de septiembre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que permita al administrador modificarle el rol a un usuario.	

Fuente: elaboración propia

Tabla 22: Tarea 8. Visualizar información de un rol

Tarea	
No.: 8	Número de HU: 3
Nombre de la tarea: Visualizar información de un rol	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 27 de septiembre de 2017	Fecha de fin: 29 de septiembre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que permita al administrador visualizar la información de un rol de usuario.	

Fuente: elaboración propia



CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

2.3.4 Iteración 4

El principal objetivo de esta iteración es proporcionarle al sistema una funcionalidad que le permita al usuario con el rol adecuado la gestión de la información de las salas. Para ello se planificaron las siguientes tareas:

1. **Tarea 9:** Agregar sala
2. **Tarea 10:** Editar sala
3. **Tarea 11:** Eliminar sala
4. **Tarea 12:** Visualizar información de sala

Tabla 23: Tarea 9. Agregar sala

Tarea	
No.: 9	Número de HU: 4
Nombre de la tarea: Agregar sala	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 30 de septiembre de 2017	Fecha de fin: 2 de octubre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que permita al administrador agregar en la base de datos la información de una sala.	

Fuente: elaboración propia

Tabla 24: Tarea 10. Editar sala

Tarea	
No.: 10	Número de HU: 4

CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

Nombre de la tarea: Editar sala	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 3 de octubre de 2017	Fecha de fin: 7 de octubre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que permita al administrador editar en la base de datos la información de una sala.	

Fuente: elaboración propia

Tabla 25: Tarea 11. Eliminar sala

Tarea	
No.: 11	Número de HU: 4
Nombre de la tarea: Eliminar sala	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 8 de octubre de 2017	Fecha de fin: 10 de octubre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al administrador eliminar de la base de datos la información de una sala.	

Fuente: elaboración propia

Tabla 26: Tarea 12. Visualizar información de sala

Tarea



CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

No.: 12	Número de HU: 4
Nombre de la tarea: Visualizar información de sala	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 10 de octubre de 2017	Fecha de fin: 12 de octubre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al administrador visualizar la información de una sala.	

Fuente: elaboración propia

2.3.5 Iteración 5

El objetivo de esta iteración es proporcionarle al sistema una funcionalidad que le permita al usuario con el rol adecuado gestionar la información de las camas. A continuación, las tareas planificadas:

1. **Tarea 13:** Agregar cama
2. **Tarea 14:** Editar cama
3. **Tarea 15:** Eliminar cama
4. **Tarea 16:** Visualizar información de cama

Tabla 27: Tarea 13. Agregar cama

Tarea	
No.: 13	Número de HU: 5
Nombre de la tarea: Agregar cama	
Tipo de tarea: Desarrollo	Estimación: 3 días



CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

Fecha de inicio: 13 de octubre de 2017	Fecha de fin: 15 de octubre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario agregar en la base de datos la información de una cama.	

Fuente: elaboración propia

Tabla 28: Tarea 14. Editar cama

Tarea	
No.: 14	Número de HU: 5
Nombre de la tarea: Editar cama	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 16 de octubre de 2017	Fecha de fin: 20 de octubre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario editar en la base de datos la información de una cama.	

Fuente: elaboración propia

Tabla 29: Tarea 15. Eliminar cama

Tarea	
No.: 15	Número de HU: 5
Nombre de la tarea: Eliminar cama	



CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 21 de octubre de 2017	Fecha de fin: 23 de octubre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario eliminar de la base de datos la información de una cama.	

Fuente: elaboración propia

Tabla 30: Tarea 16. Visualizar información de cama

Tarea	
No.: 16	Número de HU: 5
Nombre de la tarea: Visualizar información de cama	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 23 de octubre de 2017	Fecha de fin: 25 de octubre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario visualizar la información de una cama.	

Fuente: elaboración propia

2.4 Prueba

El sistema fue expuesto a varias pruebas unitarias como parte de la detección y corrección de errores, el mismo presentó un correcto funcionamiento. El rendimiento del *software* en el procesamiento de sus funcionalidades se mantuvo dentro de un rango razonable. La interfaz gráfica tuvo una buena aceptación por parte de los

CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

clientes finales, esto se obtuvo producto a su sencillez y facilidad de uso en conjunto con la gama de colores escogidas y la poca presencia de ellos en la mayoría de las páginas.

2.4.1 Pruebas de aceptación

Las pruebas de aceptación son creadas en base a las historias de usuarios en cada ciclo de la iteración del desarrollo, son consideradas como “pruebas de caja negra”. Los clientes son responsables de verificar que los resultados de éstas pruebas sean correctos.

A continuación, se muestran las pruebas de aceptación realizadas a la solución propuesta:

Tabla 31: Caso de prueba de aceptación HU1-P1

Caso de Prueba de Aceptación	
Código: HU1-P1	Historia de usuario: 1
Nombre: Autenticar usuario.	
Descripción: Prueba la funcionalidad de autenticar un nuevo usuario.	
Condiciones de ejecución: El usuario que se va autenticar debe existir en la base de datos.	
Entrada/Pasos de ejecución: Acceder al formulario de autenticación. Ingresar los datos y hacer clic en el botón “Autenticar” que aparece en la parte inferior del formulario. Posteriormente el sistema verifica si la información ingresada es correcta, de serlo se verifica si el usuario se encuentra en la base de datos. Si el usuario no existe en la base de datos del sistema, este no tendrá acceso. En caso contrario se direccionará al vista inicial del sistema.	

CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

Resultado esperado: Usuario correctamente autenticado.
Evaluación de la prueba: Prueba satisfactoria.

Fuente: elaboración propia

Tabla 32: Caso de prueba de aceptación HU4-P1

Caso de Prueba de Aceptación	
Código: HU4-P1	Historia de usuario: 4
Nombre: Agregar sala.	
Descripción: Prueba la funcionalidad de crear una nueva sala.	
Condiciones de ejecución: Para poder crear una nueva sala no debe existir una sala con el mismo centro de costo en el Sistema.	
Entrada/Pasos de ejecución: Una vez autenticado el usuario, debe acceder a la lista de las salas. Hacer clic en el botón “Agregar” que aparece en la parte inferior derecha de la tabla y llenar los campos correspondientes. Por ultimo hacer clic en el botón “Guardar” que retorna a la interfaz anterior.	
Resultado esperado: Sala correctamente creada.	
Evaluación de la prueba: Prueba satisfactoria.	

Fuente: elaboración propia



CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

Tabla 33: Caso de prueba de aceptación HU5-P1

Caso de Prueba de Aceptación	
Código: HU5-P1	Historia de usuario: 5
Nombre: Editar cama.	
Descripción: Prueba la funcionalidad de editar la información de una cama.	
Condiciones de ejecución: Para poder editar la información de una cama debe existir en el sistema al menos una cama.	
Entrada/Pasos de ejecución: Una vez autenticado el usuario, debe dirigirse a la lista de camas y seleccionar la cama a la cual se le desea editar la información. Hacer clic en el botón “Editar”, llenar los campos correspondientes. Por último hacer clic en el botón “Guardar” que retorna la interfaz anterior.	
Resultado esperado: Información de la cama actualizada.	
Evaluación de la prueba: Prueba satisfactoria.	

Fuente: elaboración propia

Tabla 34: Caso de prueba de aceptación HU6-P1

Caso de Prueba de Aceptación	
Código: HU6-P1	Historia de usuario: 6
Nombre: Eliminar pedido de cama.	
Descripción: Prueba la funcionalidad de eliminar un pedido de cama existente.	

CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

Condiciones de ejecución: Para poder eliminar un pedido de cama debe existir en el sistema al menos un pedido.
Entrada/Pasos de ejecución: Una vez autenticado el usuario, debe dirigirse a la lista de pedidos de cama y seleccionar el pedido de cama que se desea eliminar. Presionar el botón “Eliminar” que retorna a la interfaz anterior.
Resultado esperado: El pedido de cama se elimina correctamente.
Evaluación de la prueba: Prueba satisfactoria.

Fuente: elaboración propia

Tabla 35: Caso de prueba de aceptación HU9-P1

Caso de Prueba de Aceptación	
Código: HU9-P1	Historia de usuario: 9
Nombre: Despachar devolución al almacén.	
Descripción: Prueba la funcionalidad de despachar una solicitud de devolución al almacén.	
Condiciones de ejecución: Para poder despachar una devolución al almacén debe existir al menos una en la base de datos del sistema.	
Entrada/Pasos de ejecución: Una vez autenticado el usuario, debe dirigirse a la interfaz donde se encuentra la lista de solicitudes de devolución a despachar, hacer clic en el botón “despachar” de la devolución escogida que redireccionará a la interfaz donde se encuentra toda la información de la solicitud de despacho.	



CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

Ingresar los datos correspondientes y hacer clic en el botón “despachar” que retorna a la interfaz anterior.
Resultado esperado: Se despacha correctamente la devolución al almacén.
Evaluación de la prueba: Prueba satisfactoria.

Fuente: elaboración propia

Tabla 36: Caso de prueba de aceptación HU10-P1

Caso de Prueba de Aceptación	
Código: HU10-P1	Historia de usuario: 10
Nombre: Solicitar aumento de existencia en baja cobertura.	
Descripción: Prueba la funcionalidad de solicitar el aumento de la existencia de un medicamento en baja cobertura.	
Condiciones de ejecución: Para poder solicitar el aumento de una existencia en baja cobertura debe existir en el sistema al menos una existencia por debajo del 50% de su capacidad total.	
Entrada/Pasos de ejecución: Una vez autenticado el usuario, debe dirigirse a la lista de productos con existencia en baja cobertura y presionar el botón “Solicitar”.	
Resultado esperado: Se solicita correctamente la información.	
Evaluación de la prueba: Prueba satisfactoria.	

Fuente: elaboración propia

CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

Tabla 37: Caso de prueba de aceptación HU11-P1

Caso de Prueba de Aceptación	
Código: HU11-P1	Historia de usuario: 11
Nombre: Exportar pedido de sala.	
Descripción: Prueba la funcionalidad de exportar un pedido de sala existente.	
Condiciones de ejecución: Para poder exportar la información de un pedido de sala debe existir en el sistema al menos un pedido de sala.	
Entrada/Pasos de ejecución: Una vez autenticado el usuario, debe dirigirse a la lista de los pedidos de sala y hacer clic en el botón “observar” del pedido de sala seleccionado, que redireccionará a la interfaz con la información del pedido. Posteriormente hacer clic en el botón “imprimir” y el sistema publicará un archivo pdf.	
Resultado esperado: Se exporta correctamente el pedido de sala.	
Evaluación de la prueba: Prueba satisfactoria.	

Fuente: elaboración propia

Tabla 38: Caso de prueba de aceptación HU12-P1

Caso de Prueba de Aceptación	
Código: HU12-P1	Historia de usuario: 12

CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

Nombre: Buscar medicamentos pedidos y devueltos por centro de costo.
Descripción: Prueba la funcionalidad de buscar los medicamentos solicitados y despachados dependiendo de su centro de costo y una fecha específica.
Condiciones de ejecución: Para poder buscar la información el centro de costo de existir en el sistema y permanecer dentro de las fechas de despachos.
Entrada/Pasos de ejecución: Una vez autenticado el usuario, debe dirigirse a la interfaz de filtrado e ingresar los datos correspondientes y hacer clic en el botón “filtrar” que redireccionará a la misma interfaz y brindara la información deseada.
Resultado esperado: El funcionamiento de la búsqueda es correcto.
Evaluación de la prueba: Prueba satisfactoria.

Fuente: elaboración propia

2.6 Conclusiones parciales

En el presente capítulo se abordaron todos los temas referentes a la metodología de desarrollo de *software* escogida en el diseño y construcción de la solución propuesta, los mismos provocaron las siguientes conclusiones:

1. Se comprobó la pertinencia de la metodología de desarrollo de *software XP* y las tecnologías seleccionadas para la implementación del sistema.
2. Se conformó la distribución de historias de usuario por iteración y el cronograma de entrega del sistema, basado en el desarrollo de las tareas correspondientes al ciclo de trabajo de la metodología.

CAPÍTULO 2. DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA

3. Mediante la implementación de las pruebas de aceptación de la metodología escogida, se comprobó la conformidad del usuario final en cuanto a las funcionalidades del sistema y su aspecto visual.



CONCLUSIONES GENERALES

1. La realización del estudio en el Hospital Universitario Clínico Quirúrgico “Lucía Íñiguez Landín” permitió identificar las deficiencias existentes en la gestión del proceso de distribución de medicamentos, lo cual corroboró la necesidad de informatizar este proceso.
2. Después de realizar una búsqueda en instituciones homólogas, se determinó la inexistencia de un sistema informático que brinde los procesos demandados por el centro hospitalario.
3. Basado en una correcta selección de la metodología de desarrollo de *software*, tecnologías y herramientas empleadas se logró dar cumplimiento a los requisitos para una eficaz gestión de los procesos.
4. Los resultados finales de las pruebas de aceptación de la metodología *XP* evidenciaron la satisfacción de los usuarios finales.



RECOMENDACIONES

1. Incluir una vista que le permita al usuario con el rol adecuado determinar el cuadro básico de medicamentos por servicio.
2. Incluir una vista que le permita al usuario con el rol adecuado determinar la variación anual y mensual de consumo.
3. Extender la utilización del sistema a las demás entidades homólogas de la provincia.



REFERENCIAS BIBLIOGRÁFICAS

- [1] M. Vidal Ledo, *Primera estrategia para la informatización del sector de la Salud Pública Cubana. Una propuesta para el desarrollo.* 2007.
- [2] Delgado Ramos A, "Informática en la salud pública cubana Rev. Cubana Salud Pública," 2006.
- [3] A. R. Díaz, *Informatización en el Sistema Nacional de Salud.* 2013.
- [4] M. Dirección de Informática, *Programa de Informatización del Sector de la Salud en Cuba.* 2003.
- [5] "Proyecto de Informatización," 2018.
- [6] MINSAP, *Normas y Procedimientos Farmacia Hospitalaria.* .
- [7] K. R. Á. Flores, *Análisis del funcionamiento del sistema de distribución de medicamentos por dosis unitaria en el Centro Médico Naval "Cirujano Mayor Santiago Távara."*2005.
- [8] EMCOMED, *Manual de Imagen e Identidad Corporativa de EMCOMED.* .
- [9] HUCQ, *Manual de Identidad de Hospital Universitario Clínico Quirúrgico.* .
- [10] MINSAP, *Manual Farmacia Hospitalaria.* .
- [11] "Gestión de la Información."
- [12] S. Alvarez, "Arquitectura cliente-servidor," 2007.
- [13] M. Haverbeke, *Eloquent JavaScript.* 2013.
- [14] Bysslender, "Desventajas de la arquitectura cliente-servidor," 2016.
- [15] G. Van Rossum, *El tutorial de Python.* 2009.
- [16] M. A. Alvarez, "Qué es Python," 2013.
- [17] T. Navarrete, *El lenguaje JavaScript.* 2007.
- [18] J. E. Pérez, *Introducción a JavaScript.* 2008.

- [19] “Django: guía rápida para desarrollar páginas web con este framework,” 2016.
- [20] J. del Pino, “Introducción a Django,” 2018.
- [21] Y. S. Falcón, “¿Qué es un IDE?,” 2015.
- [22] E. Medina, “Visual Studio Code, editor de código de Microsoft para Windows, OS X y GNU/Linux,” 2015.
- [23] R. C. Paré, *Software Libre. Base de Datos*. 2018.
- [24] O. P. M. Marc Gibert Ginestá, *Bases de datos en PostgreSQL*. 2018.
- [25] J. D. L. Castillo, “Características que debemos de conocer de PostgreSQL,” 2017.
- [26] E. M. Talón, *Apache*. 2011.
- [27] J. Joskowicz, *Reglas y Prácticas en eXtreme Programming*. 2008.
- [28] B. L. Yolanda, *Metodología Ágil de Desarrollo de Software – XP*. 2016.



ANEXOS

Anexo 1. Tareas para iteración 6

Tabla 39: Tarea 17. Agregar pedido de cama

Tarea	
No.: 17	Número de HU: 6
Nombre de la tarea: Agregar pedido de cama	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 26 de octubre de 2017	Fecha de fin: 28 de octubre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario agregar en la base de datos el pedido de una cama.	

Fuente: elaboración propia

Tabla 40: Tarea 18. Editar pedido de cama

Tarea	
No.: 18	Número de HU: 6
Nombre de la tarea: Editar pedido de cama	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 29 de octubre de 2017	Fecha de fin: 2 de noviembre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario editar en la base de datos el pedido de una cama.	

Fuente: elaboración propia

Tabla 41: Tarea 19. Eliminar pedido de cama

Tarea	
No.: 19	Número de HU: 6
Nombre de la tarea: Eliminar pedido de cama	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 3 de noviembre de 2017	Fecha de fin: 5 de noviembre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario eliminar de la base de datos un pedido de cama.	

Fuente: elaboración propia

Tabla 42: Tarea 20. Buscar pedido de cama

Tarea	
No.: 20	Número de HU: 6
Nombre de la tarea: Buscar pedido de cama	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 5 de noviembre de 2017	Fecha de fin: 9 de noviembre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario buscar en la base de datos un pedido de cama.	

Fuente: elaboración propia

Tabla 43: Tarea 21. Visualizar información de pedido de cama

Tarea	
No.: 21	Número de HU: 6
Nombre de la tarea: Visualizar información de pedido de cama	
Tipo de tarea: Desarrollo	Estimación: 1 día
Fecha de inicio: 10 de noviembre de 2017	Fecha de fin: 11 de noviembre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario visualizar la información de un pedido de cama.	

Fuente: elaboración propia

Tabla 44: Tarea 22. Solicitar pedido de sala

Tarea	
No.: 22	Número de HU: 6
Nombre de la tarea: Solicitar pedido de sala	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 11 de noviembre de 2017	Fecha de fin: 14 de noviembre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario agregar en la base de datos un pedido de sala.	

Fuente: elaboración propia

Tabla 45: Tarea 23. Agregar devolución de cama

Tarea	
No.: 23	Número de HU: 6
Nombre de la tarea: Agregar devolución de cama	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 15 de noviembre de 2017	Fecha de fin: 19 de noviembre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario agregar en la base de datos una devolución de cama.	

Fuente: elaboración propia

Tabla 46: Tarea 24. Editar devolución de cama

Tarea	
No.: 24	Número de HU: 6
Nombre de la tarea: Editar devolución de cama	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 20 de noviembre de 2017	Fecha de fin: 22 de noviembre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario editar en la base de datos una devolución de cama.	

Fuente: elaboración propia

Tabla 47: Tarea 25. Eliminar devolución de cama

Tarea	
No.: 25	Número de HU: 6
Nombre de la tarea: Eliminar devolución de cama	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 23 de noviembre de 2017	Fecha de fin: 27 de noviembre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario eliminar de la base de datos una devolución de cama.	

Fuente: elaboración propia

Tabla 48: Tarea 26. Buscar devolución de cama

Tarea	
No.: 26	Número de HU: 6
Nombre de la tarea: Buscar devolución de cama	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 28 de noviembre de 2017	Fecha de fin: 30 de noviembre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario buscar en la base de datos una devolución de cama.	

Fuente: elaboración propia

Tabla 49: Tarea 27. Visualizar información de devolución de cama

Tarea	
No.: 27	Número de HU: 6
Nombre de la tarea: Visualizar información de devolución de cama	
Tipo de tarea: Desarrollo	Estimación: 1 día
Fecha de inicio: 30 de noviembre de 2017	Fecha de fin: 3 de diciembre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario visualizar la información de una devolución de cama.	

Fuente: elaboración propia

Tabla 50: Tarea 28. Solicitar devolución de sala

Tarea	
No.: 28	Número de HU: 6
Nombre de la tarea: Solicitar devolución de sala	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 4 de diciembre de 2017	Fecha de fin: 6 de diciembre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario agregar en la base de datos una devolución de sala.	

Fuente: elaboración propia

Anexo 2. Tareas para la iteración 7

Tabla 51: Tarea 29. Despachar pedido de sala

Tarea	
No.: 29	Número de HU: 7
Nombre de la tarea: Despachar pedido de sala	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 7 de diciembre de 2017	Fecha de fin: 9 de diciembre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario editar en la base de datos un pedido de sala.	

Fuente: elaboración propia

Tabla 52: Tarea 30. Eliminar pedido de sala

Tarea	
No.: 30	Número de HU: 7
Nombre de la tarea: Eliminar pedido de sala	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 10 de diciembre de 2017	Fecha de fin: 14 de diciembre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario eliminar de la base de datos un pedido de sala.	

Fuente: elaboración propia

Tabla 53: Tarea 31. Buscar pedido de sala

Tarea	
No.: 31	Número de HU: 7
Nombre de la tarea: Buscar pedido de sala	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 15 de diciembre de 2017	Fecha de fin: 17 de diciembre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario buscar en la base de datos un pedido de sala.	

Fuente: elaboración propia

Tabla 54: Tarea 32. Visualizar información de pedido de sala

Tarea	
No.: 32	Número de HU: 7
Nombre de la tarea: Visualizar información de pedido de sala	
Tipo de tarea: Desarrollo	Estimación: 1 día
Fecha de inicio: 18 de diciembre de 2017	Fecha de fin: 21 de diciembre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario visualizar la información de un pedido de sala.	

Fuente: elaboración propia

Tabla 55: Tarea 33. Despachar devolución de sala

Tarea	
No.: 33	Número de HU: 7
Nombre de la tarea: Despachar devolución de sala	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 22 de diciembre de 2017	Fecha de fin: 24 de diciembre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario editar en la base de datos una solicitud de devolución de cama.	

Fuente: elaboración propia

Tabla 56: Tarea 34. Eliminar devolución de sala

Tarea	
No.: 34	Número de HU: 7
Nombre de la tarea: Eliminar devolución de sala	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 25 de diciembre de 2017	Fecha de fin: 29 de diciembre de 2017
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario eliminar de la base de datos una devolución de sala.	

Fuente: elaboración propia

Tabla 57: Tarea 35. Buscar devolución de sala

Tarea	
No.: 35	Número de HU: 7
Nombre de la tarea: Buscar devolución de sala	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 30 de diciembre de 2017	Fecha de fin: 1 de enero de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario buscar en la base de datos una devolución de sala.	

Fuente: elaboración propia

Tabla 58: Tarea 36. Visualizar información de devolución de sala

Tarea	
No.: 36	Número de HU: 7
Nombre de la tarea: Visualizar información de devolución de sala	
Tipo de tarea: Desarrollo	Estimación: 1 día
Fecha de inicio: 2 de enero de 2018	Fecha de fin: 5 de enero de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario visualizar la información de una devolución de sala.	

Fuente: elaboración propia

Anexo 3. Tareas para la iteración 8*Tabla 59: Tarea 37. Agregar pedido de farmacia*

Tarea	
No.: 37	Número de HU: 8
Nombre de la tarea: Agregar pedido de farmacia	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 6 de enero de 2018	Fecha de fin: 8 de enero de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario agregar en la base de datos un pedido de farmacia.	

*Fuente: elaboración propia**Tabla 60: Tarea 38. Visualizar información de pedido de farmacia*

Tarea	
No.: 38	Número de HU: 8
Nombre de la tarea: Visualizar información de pedido de farmacia	
Tipo de tarea: Desarrollo	Estimación: 1 día
Fecha de inicio: 9 de enero de 2017	Fecha de fin: 12 de enero de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario visualizar la información de un pedido de farmacia.	

Fuente: elaboración propia

Tabla 61: Tarea 39. Solicitar pedido al almacén

Tarea	
No.: 39	Número de HU: 8
Nombre de la tarea: Solicitar pedido al almacén	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 13 de enero de 2017	Fecha de fin: 15 de enero de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario agregar en la base de datos un pedido al almacén.	

Fuente: elaboración propia

Tabla 62: Tarea 40. Agregar devolución de farmacia

Tarea	
No.: 40	Número de HU: 8
Nombre de la tarea: Agregar devolución de farmacia	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 15 de enero de 2018	Fecha de fin: 20 de enero de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario agregar en la base de datos una devolución de farmacia.	

Fuente: elaboración propia

Tabla 63: Tarea 41. Editar devolución de farmacia

Tarea	
No.: 41	Número de HU: 8
Nombre de la tarea: Editar devolución de farmacia	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 21 de enero de 2018	Fecha de fin: 23 de enero de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario editar en la base de datos una devolución de farmacia.	

Fuente: elaboración propia

Tabla 64: Tarea 42. Eliminar devolución de farmacia

Tarea	
No.: 42	Número de HU: 8
Nombre de la tarea: Eliminar devolución de farmacia	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 24 de enero de 2018	Fecha de fin: 28 de enero de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario eliminar de la base de datos una devolución de farmacia.	

Fuente: elaboración propia

Tabla 65: Tarea 43. Visualizar información de devolución de farmacia

Tarea	
No.: 43	Número de HU: 8
Nombre de la tarea: Visualizar información de devolución de farmacia	
Tipo de tarea: Desarrollo	Estimación: 1 día
Fecha de inicio: 29 de enero de 2018	Fecha de fin: 30 de enero de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario visualizar la información de una devolución de farmacia.	

Fuente: elaboración propia

Tabla 66: Tarea 44. Solicitar devolución al almacén

Tarea	
No.: 44	Número de HU: 8
Nombre de la tarea: Solicitar devolución al almacén	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 30 de enero de 2018	Fecha de fin: 4 de febrero de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario agregar en la base de datos una devolución al almacén.	

Fuente: elaboración propia

Anexo 4. Tareas para la iteración 9*Tabla 67: Tarea 45. Despachar pedido al almacén*

Tarea	
No.: 45	Número de HU: 9
Nombre de la tarea: Despachar pedido al almacén	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 5 de febrero de 2018	Fecha de fin: 7 de febrero de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario editar en la base de datos una solicitud de pedido de farmacia.	

*Fuente: elaboración propia**Tabla 68: Tarea 46. Eliminar pedido al almacén*

Tarea	
No.: 46	Número de HU: 9
Nombre de la tarea: Eliminar pedido al almacén	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 8 de febrero de 2018	Fecha de fin: 12 de febrero de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario eliminar de la base de datos un pedido al almacén.	

Fuente: elaboración propia

Tabla 69: Tarea 47. Buscar pedido al almacén

Tarea	
No.: 47	Número de HU: 9
Nombre de la tarea: Buscar pedido al almacén	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 13 de febrero de 2018	Fecha de fin: 15 de febrero de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario buscar en la base de datos un pedido al almacén.	

Fuente: elaboración propia

Tabla 70: Tarea 48. Visualizar información de pedido al almacén

Tarea	
No.: 48	Número de HU: 9
Nombre de la tarea: Visualizar información de pedido al almacén	
Tipo de tarea: Desarrollo	Estimación: 1 día
Fecha de inicio: 15 de febrero de 2018	Fecha de fin: 16 de febrero de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario visualizar la información de un pedido al almacén.	

Fuente: elaboración propia

Tabla 71: Tarea 49. Despachar devolución al almacén

Tarea	
No.: 49	Número de HU: 9
Nombre de la tarea: Despachar devolución al almacén	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 16 de febrero de 2018	Fecha de fin: 21 de febrero de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario editar en la base de datos una solicitud de devolución al almacén.	

Fuente: elaboración propia

Tabla 72: Tarea 50. Eliminar devolución al almacén

Tarea	
No.: 50	Número de HU: 9
Nombre de la tarea: Eliminar devolución al almacén	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 22 de febrero de 2018	Fecha de fin: 24 de febrero de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario eliminar de la base de datos una devolución al almacén.	

Fuente: elaboración propia

Tabla 73: Tarea 51. Buscar devolución al almacén

Tarea	
No.: 51	Número de HU: 9
Nombre de la tarea: Buscar devolución al almacén	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 24 de febrero de 2018	Fecha de fin: 28 de febrero de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario buscar en la base de datos una devolución al almacén.	

Fuente: elaboración propia

Tabla 74: Tarea 52. Visualizar información de devolución al almacén

Tarea	
No.: 52	Número de HU: 9
Nombre de la tarea: Visualizar información de devolución al almacén	
Tipo de tarea: Desarrollo	Estimación: 1 día
Fecha de inicio: 1 de marzo de 2018	Fecha de fin: 2 de marzo de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario visualizar la información de una devolución al almacén.	

Fuente: elaboración propia

Anexo 5. Tareas para la iteración 10*Tabla 75: Tarea 53. Agregar medicamento*

Tarea	
No.: 53	Número de HU: 10
Nombre de la tarea: Agregar medicamento	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 3 de marzo de 2018	Fecha de fin: 5 de marzo de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario agregar en la base de datos un medicamento.	

*Fuente: elaboración propia**Tabla 76: Tarea 54. Editar medicamento*

Tarea	
No.: 54	Número de HU: 10
Nombre de la tarea: Editar medicamento	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 6 de marzo de 2018	Fecha de fin: 10 de marzo de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario editar en la base de datos un medicamento.	

Fuente: elaboración propia

Tabla 77: Tarea 55. Eliminar medicamento

Tarea	
No.: 55	Número de HU: 10
Nombre de la tarea: Eliminar medicamento	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 11 de marzo de 2018	Fecha de fin: 13 de marzo de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario eliminar de la base de datos un medicamento.	

Fuente: elaboración propia

Tabla 78: Tarea 56. Visualizar información de medicamento

Tarea	
No.: 56	Número de HU: 10
Nombre de la tarea: Visualizar información de medicamento	
Tipo de tarea: Desarrollo	Estimación: 1 día
Fecha de inicio: 13 de marzo de 2018	Fecha de fin: 15 de marzo de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario visualizar la información de un medicamento.	

Fuente: elaboración propia

Tabla 79: Tarea 57. Agregar existencia de medicamento

Tarea	
No.: 57	Número de HU: 10
Nombre de la tarea: Agregar existencia de medicamento	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 16 de marzo de 2018	Fecha de fin: 18 de marzo de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario agregar en la base de datos un medicamento.	

Fuente: elaboración propia

Tabla 80: Tarea 58. Editar existencia de medicamento

Tarea	
No.: 58	Número de HU: 10
Nombre de la tarea: Editar existencia de medicamento	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 18 de febrero de 2018	Fecha de fin: 22 de marzo de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario editar en la base de datos la existencia de un medicamento.	

Fuente: elaboración propia

Tabla 81: Tarea 59. Eliminar existencia de medicamento

Tarea	
No.: 59	Número de HU: 10
Nombre de la tarea: Eliminar existencia de medicamento	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 23 de marzo de 2018	Fecha de fin: 25 de marzo de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario eliminar de la base de datos la existencia de un medicamento.	

Fuente: elaboración propia

Tabla 82: Tarea 60. Solicitar aumento de existencia en baja cobertura

Tarea	
No.: 60	Número de HU: 10
Nombre de la tarea: Solicitar aumento de existencia en baja cobertura	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 25 de febrero de 2018	Fecha de fin: 29 de marzo de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario buscar en la base de datos la existencia de un medicamento.	

Fuente: elaboración propia

Anexo 6. Tareas para la iteración 11*Tabla 83: Tarea 61. Exportar pedido de cama*

Tarea	
No.: 61	Número de HU: 11
Nombre de la tarea: Exportar pedido de cama	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 30 de marzo de 2018	Fecha de fin: 1 de abril de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario exportar un pedido de cama.	

*Fuente: elaboración propia**Tabla 84: Tarea 62. Exportar despacho de pedido de sala*

Tarea	
No.: 62	Número de HU: 11
Nombre de la tarea: Exportar despacho de pedido de sala.	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 2 de abril de 2018	Fecha de fin: 5 de abril de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario exportar un despacho de pedido de sala.	

Fuente: elaboración propia

Tabla 85: Tarea 63. Exportar devolución de cama

Tarea	
No.: 63	Número de HU: 11
Nombre de la tarea: Exportar devolución de cama	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 6 de abril de 2018	Fecha de fin: 8 de abril de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario exportar una devolución de cama.	

Fuente: elaboración propia

Tabla 86: Tarea 64. Exportar despacho de devolución de sala

Tarea	
No.: 64	Número de HU: 11
Nombre de la tarea: Exportar despacho de devolución de sala	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 8 de abril de 2018	Fecha de fin: 12 de abril de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario exportar un despacho de devolución de sala.	

Fuente: elaboración propia

Tabla 87: Tarea 65. Exportar medicamentos

Tarea	
No.: 65	Número de HU: 11
Nombre de la tarea: Exportar medicamentos	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 13 de abril de 2018	Fecha de fin: 15 de abril de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario exportar un medicamento.	

Fuente: elaboración propia

Tabla 88: Tarea 66. Exportar solicitud de aumento de existencia en baja cobertura

Tarea	
No.: 66	Número de HU: 11
Nombre de la tarea: Exportar solicitud de aumento de existencia en baja cobertura	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 15 de abril de 2018	Fecha de fin: 19 de abril de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario exportar una solicitud de aumento de existencia en baja cobertura.	

Fuente: elaboración propia

Tabla 89: Tarea 67. Exportar salas

Tarea	
No.: 67	Número de HU: 11
Nombre de la tarea: Exportar salas	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 20 de abril de 2018	Fecha de fin: 22 de abril de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario exportar las salas existentes.	

Fuente: elaboración propia

Anexo 7. Tareas para la iteración 12

Tabla 90: Tarea 68. Buscar medicamentos pedidos y devueltos por centro de costo

Tarea	
No.: 68	Número de HU: 12
Nombre de la tarea: Buscar medicamentos pedidos y devueltos por centro de costo	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 23 de abril de 2018	Fecha de fin: 25 de abril de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario buscar los medicamentos solicitados y despachados por cada centro de costo.	

Fuente: elaboración propia

Tabla 91: Tarea 69. Visualizar información de medicamentos pedidos y devueltos por centro de costo

Tarea	
No.: 69	Número de HU: 12
Nombre de la tarea: Visualizar información de medicamentos pedidos y devueltos por centro de costo	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 26 de abril de 2018	Fecha de fin: 28 de abril de 2018
Programador responsable: Rasel Agüero Fernández	
Descripción: Implementar una vista que le permita al usuario visualizar los medicamentos solicitados y despachados por centro de costo.	

Fuente: elaboración propia

