

Universidad de las Ciencias Informáticas



Facultad 2

Trabajo de diploma para optar por el título de Ingeniero en
Ciencias Informáticas.

*Herramienta informática para la
determinación de los métodos de
destrucción en EMCOMED*

Autores:

Dayma Naranjo López

Luis Daniel Castro Durán

Tutores: Ing. Maidelis Milanés Luque

MSc. Yunia Reyes González

Co-Tutor: Ing. Roberto Antonio Infante Milanés

La Habana, junio 2017

Declaración de Autoría:

Declaramos ser autores de la presente tesis, con el título: Herramienta informática para la determinación de los métodos de destrucción en EMCOMED y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmamos la presente a los __ días del mes de _____ del _____.

Dayma Naranjo López

Firma del autor

Luis Daniel Castro Durán

Firma del autor

Ing. Maidelis Milanés Luque

Firma del tutor

MSc. Yunia Reyes González

Firma del tutor



*“El mérito verdadero es el que el hombre adquiere
con su voluntad, con su esfuerzo, con su
constancia”.*

Fidel Castro Ruz

Dedicatoria

A mi mamita linda Neisy por apoyarme y cuidarme siempre.

A mi abuelita Pili por su amor y paciencia.

A mi tía Odalys por estar siempre a mi lado.

A mi papá y a mi tía Deysi por ayudarme en todo.

A todos los presentes, los que creyeron, nunca me abandonaron y confiaron en mí.

Luís Daniel

A mi mamá.

A mi familia.

A todos los que me ayudaron durante estos 5 años.

Dayma

Agradecimientos

Es increíble que aunque no le hagamos caso, que aunque entremos en un camino errado o porque seamos diferente a los demás, ellos nos siguen queriendo... amando. Mamí, para tí va todo este agradecimiento, por tí soy la persona que soy hoy, capaz de amar, de respetar y de comprender, no sería tal persona si no me hubieras educado tan bien, dándome todo tu amor, capaz de quitarte lo que tengas para que no pase necesidades, aunque pelees mucho sé que me amas y que darías la vida por mí. De tí he recibido tanto: tu apoyo, tu amor, tu educación... lo mínimo que puedo ofrecerte es el orgullo que hoy te doy. Gracias por tu sacrificio a diario para que yo estudie y concluya mi carrera. No sé qué será de mí el día que no estés, eres la persona más importante de mi vida, gracias mamá, te amo.

Que sería de mí sin una personita que según ella sólo tiene la mente sana, que la pobre ha pasado por tantas cosas a la edad de 80 años y aún tiene tantos deseos de vivir que hacen que la admire cada día más. Mi abuela Pili, si antes dije que mi mamá lo era todo en mi vida, a tí te digo que no tengo forma de comparar lo que siento por tí. Quisiera decirte a diario tantas cosas que siento pero por desgracia sólo uso momentos como estos para decirte: Te amo.

Luís Daniel

A todos por su apoyo y cariño.

Dayma

Resumen

El incremento de la producción de medicamentos a nivel mundial, hace que se torne sumamente importante el tratamiento que se le deben dar a los residuales que se generan diariamente en este sector, por lo que la selección del método de destrucción adecuado a cada residuo presenta un importante impacto medioambiental, económico y social. En Cuba la Empresa Comercializadora y Distribuidora de Medicamentos (EMCOMED) es la encargada de gestionar el proceso de tratamiento de medicamentos rechazados.

El objetivo de la presente investigación es desarrollar una herramienta informática para la determinación de los métodos de destrucción asociados a un medicamento. Esta herramienta permitirá obtener a partir de reglas, para cada medicamento rechazado los métodos de destrucción y el costo asociado a cada uno de ellos. Finalmente, será seleccionado el método a emplear a criterio de la especialista de calidad.

Palabras claves: medicamentos rechazados, método de destrucción, tratamiento de medicamentos rechazados, herramienta informática, reglas

Índice de Contenidos

Introducción	8
Capítulo 1: Fundamentación teórica	13
1.1. Introducción	13
1.2. Proceso de tratamiento de los medicamentos rechazados en EMCOMED. ..	13
1.3. Impacto social, ambiental y económico	15
1.4. Análisis de sistemas existentes para el tratamiento de medicamentos.....	16
1.4.1. Sistemas nacionales	17
1.4.2. Sistemas internacionales	18
1.4.3. Resultado del análisis	18
1.5. Técnica a emplear	18
1.6. Metodologías de desarrollo	19
1.7. Tecnologías y herramientas de desarrollo.....	23
1.7.1. Lenguajes empleados.....	23
1.7.2. Framework utilizado	23
1.7.3. Herramienta CASE	24
1.7.4. Entorno de desarrollo integrado	24
1.7.5. Sistema gestor de bases de datos.....	24
1.8. Conclusiones parciales	25
Capítulo 2: Propuesta de solución	26
2.1. Introducción	26
2.2. Modelo conceptual del negocio	26
2.3. Variables lingüísticas.	27
2.4. Propuesta de solución	27
2.5. Especificación de los requisitos del sistema	30
2.5.1. Requisitos funcionales del sistema.....	30
2.5.2. Requisitos no funcionales del sistema.....	32
2.6. Definición de los actores.....	33
2.7. Diagrama de casos de uso del sistema	34
2.7.1. Descripción de los casos de uso del sistema	34
2.8. Diseño de la arquitectura.....	36
2.9. Etapa de diseño	37
2.9.1. Diseño de interfaz.....	37
2.9.2. Diagrama de paquetes	40
2.9.3. Diagrama de clases de diseño utilizando estereotipos web	40
2.10. Modelo físico de los datos.....	41

2.11. Patrones de diseño	42
2.12. Conclusiones parciales.....	44
Capítulo 3: Implementación y prueba.....	46
3.1. Introducción	46
3.2. Etapa de implementación.....	46
3.2.1. Diagrama de componente.....	46
3.3. Estándares de codificación empleados.....	47
3.4. Pruebas	47
3.4.1. Pruebas unitarias.....	47
3.4.2. Pruebas de validación	48
3.4.3. Pruebas de despliegue.....	50
3.4.4 Resultados de las pruebas.....	52
3.5. Conclusiones parciales	53
Conclusiones generales.....	54
Recomendaciones	55
Referencias bibliográficas	56
Bibliografía.....	59
Anexos.....	62
Anexo I: Descripción de los casos de uso del sistema.	62
Anexo II: Diagramas de clases de diseño	72
Anexo III: Ejemplo de casos de prueba	75

Índice de Tablas

Tabla 1: Fases Variación AUP-UCI	21
Tabla 2: Requisitos funcionales del sistema	30
Tabla 3: Actor relacionado con el sistema.....	33
Tabla 4: Descripción CU4. Determinar métodos de destrucción.....	34
Tabla 5: Descripción CU5. Generar reporte.....	35
Tabla 6: Caso de prueba de partición equivalente del RF1 Autenticar usuario	49
Tabla 7: Pruebas de Despliegue.....	51
Tabla 8: Resultado de las pruebas de caja negra.....	52
Tabla 9: Descripción CU1. Gestionar usuario.	62
Tabla 10: Descripción CU2. Gestionar Medicamento.....	64
Tabla 11: Descripción CU3. Gestionar regla.....	69
Tabla 12: Caso de prueba de partición equivalente del RF11 Insertar regla.....	75
Tabla 13: Caso de prueba de partición equivalente del RF12 Modificar regla	76

Índice de Figuras

Ilustración 1: Modelo Conceptual.....	27
Ilustración 2: Flujo de la propuesta de solución	28
Ilustración 3: Diagrama de casos de uso del sistema	34
Ilustración 4: Arquitectura Cliente –Servidor.....	36
Ilustración 5: Funcionamiento del MTV en Django.	37
Ilustración 6: Logo de la herramienta informática	38
Ilustración 7: Autenticar usuario.....	38
Ilustración 8: Interfaz de la herramienta informática DEMED	40
Ilustración 9: Diagrama de paquetes.....	40
Ilustración 10: Diagrama de clases de diseño RF Determinar métodos de destrucción	41
Ilustración 11: Modelo físico de la base de datos	42
Ilustración 12: Diagrama de componentes del sistema.....	46
Ilustración 13: Prueba unitaria a la funcionalidad Determinar métodos de destrucción	48
Ilustración 14: Resultado de las pruebas unitarias	48
Ilustración 15: Gráfico correspondiente al caso de prueba del RF9 Determinar métodos de destrucción.....	50
Ilustración 16: Gráfico correspondiente al caso de prueba del RF10 Generar reporte.	50
Ilustración 17: Diagrama de clases de diseño RF: Autenticar y Editar usuario.....	72
Ilustración 18: Diagrama de clases de diseño RF: Buscar, Insertar, Modificar, Eliminar medicamento y Seleccionar método de destrucción	73
Ilustración 19: Diagrama de clases de diseño RF: Gestionar reporte.....	74
Ilustración 20: Diagrama de clases de diseño RF: Insertar, Modificar y Eliminar regla	75

Introducción

Con la industrialización y el desarrollo tecnológico en el mundo se ha evidenciado un aumento de la producción de residuos. Se considera residuo a: *“toda sustancia o artículo que se convierta en desecho y que, por sus características físicas, biológicas o químicas, pueda representar un peligro para el medio ambiente y la salud humana”* (Miyar, 2009). Un inadecuado tratamiento de residuales conlleva a graves consecuencias sociales, económicas y medioambientales. La creciente conciencia de la humanidad referente a este tema, ha llevado a la concepción de normas, leyes y regulaciones internacionales, con el objetivo de controlar y asegurar una adecuada política con respecto al tratamiento de dichos residuales.

La serie de normas ISO¹ 14000 y la Ley 81 del Medio Ambiente que contribuyen a resolver el problema económico para adaptarse a las normas regulatorias del sistema de gestión ambiental, son un ejemplo de la importancia que se le da en el mundo a esta temática. En la Estrategia Ambiental Nacional del Ministerio de Ciencia, Tecnología y Medio Ambiente de Cuba (CITMA) para el período 2016-2020, se identifica al fenómeno de la contaminación como uno de los problemas ambientales del país y al deficiente manejo de los residuales como uno de los factores que contribuyen a su generación.

Existen diferentes tipos de residuos:

- Sólidos urbanos: son aquellos que componen la basura doméstica.
- Agrarios: proceden de la agricultura, la ganadería, la pesca, las explotaciones forestales y la industria alimentaria.
- Radiactivos: sustancias que emiten radiactividad.
- Médicos y de laboratorios: son los restos del trabajo químico y de laboratorio, dentro de los cuales se insertan los medicamentos.

Aproximadamente en la década del 40, con el desarrollo farmacéutico gracias al avance de la ciencia y la técnica los medicamentos comenzaron a producirse en grandes cantidades, lo que aceleró paralelamente el aumento de residuos en

¹ International Organization for Standardization (ISO, siglas en inglés).

esta área. Desde entonces las maneras en que se presentan los fármacos han evolucionado y la diversidad que se encuentra en el mercado es muy amplia. Los residuos definidos como medicamentos rechazados, provenientes de las industrias biotecnológicas y farmacéuticas, se incrementan y al producirse más de estos productos para la sociedad consumidora, se asume el riesgo de que aumenten estos desechos.

“Los medicamentos rechazados son aquellos vencidos o que fueron mal producidos” (López, 2016). Mientras se consumen los medicamentos que nos liberan de enfermedades, estos desechos contribuyen al deterioro del entorno. Ello trae consigo la necesidad de ofrecer un tratamiento adecuado a estos residuos, de forma tal que se pueda evitar un posible daño ambiental, económico y social.

Para ello es creada por el Ministerio de Economía y Planificación la Empresa Comercializadora y Distribuidora de Medicamentos (EMCOMED). Esta se encarga de comercializar productos médicos y responde por el cumplimiento con los requisitos implícitos y explícitos para cada uno, según lo establecido por las disposiciones legales y reglamentarias.

Desde el momento que ingresan los medicamentos a la empresa pasan por un proceso de calidad donde se determina si están aptos para la venta o son rechazados. En el caso de los medicamentos rechazados, es necesario darle un tratamiento residual que puede ser su destrucción o almacenamiento. Esta temática constituye hoy un tema de investigación en EMCOMED que lleva por título *“La gestión integrada de los residuales sólidos en la plataforma logística de las industrias biotecnológica y farmacéutica”*, resultados que se han presentado en foros investigativos a nivel nacional e internacional, con el objetivo de trazar una metodología para garantizar un proceso eficiente que logre la correcta destrucción de medicamentos rechazados.

EMCOMED es la única empresa en el país encargada de comercializar este tipo de producto, brindando el debido tratamiento a aquellos que se consideran rechazados. Diariamente se reciben grandes volúmenes de estos medicamentos, haciendo que esta actividad se torne compleja, pues solo cuenta con una especialista de calidad encargada de procesar toda la información

referente, para determinar el método de destrucción adecuado para un medicamento.

Para la determinación del método de destrucción adecuado la especialista de calidad debe primeramente realizar una clasificación farmacológica, apoyándose en el Formulario de Medicamentos. Posteriormente se procede según el manual de Cuadros Básico a catalogarlos en Clase 1, Clase 2 o Clase 3, y en el mismo documento se determinan los posibles métodos de destrucción para cada clase. Finalmente se procede al cálculo del costo asociado a cada método de destrucción según los medicamentos analizados, con el fin de seleccionar el más conveniente, apoyándose en la ficha de costo asociada a cada método de destrucción. Es válido destacar que existen medicamentos que pueden traer restricciones para su tratamiento en dependencia de la empresa por la que fue producido o sus características físicas, lo cuál debe ser consultado también por la especialista de calidad.

Este proceso es realizado de forma manual, y en ocasiones de manera empírica, confiando en la experiencia acumulada de la especialista de calidad, lo que puede conllevar a una toma de decisiones inadecuada y por ende a un posible daño económico, social y medioambiental. En Cuba, uno de los principales daños asociados a la incorrecta determinación de los métodos de destrucción de los medicamentos rechazados está dado por la contaminación de las aguas y la vegetación que se encuentran situadas en el perímetro de los vertientes de estos residuos, trayendo consigo enfermedades en los animales y seres humanos además del gasto económico que esto implica para el estado.

Atendiendo a todo lo anteriormente especificado se identifica como **problema a resolver**: la dificultad en la adecuada selección del método de destrucción a emplear en el proceso de tratamiento a los medicamentos rechazados en EMCOMED.

Se define como **objeto de estudio**: el tratamiento a los medicamentos rechazados en EMCOMED.

En este sentido el **objetivo general** que se propone alcanzar es: Desarrollar una herramienta informática para la determinación de los métodos de destrucción de

los medicamentos rechazados en EMCOMED, que facilite la toma de decisiones en la selección del método adecuado.

Atendiendo al objetivo declarado se ha definido como **campo de acción:** la determinación de los métodos de destrucción de los medicamentos rechazados en EMCOMED.

Para dar cumplimiento al objetivo general se plantean las siguientes **tareas de investigación:**

1. Construcción de las bases teóricas de la investigación partiendo del análisis del proceso de tratamiento de los medicamentos rechazados en EMCOMED.
2. Análisis de sistemas que tratan medicamentos, para verificar la necesidad de la construcción de la herramienta e identificar elementos necesarios para el desarrollo del sistema.
3. Análisis de la técnica a emplear para la determinación de los métodos de destrucción de medicamentos.
4. Análisis de las metodologías de desarrollo de software que posibilite la selección de la más idónea para la construcción del sistema.
5. Análisis de las herramientas, tecnologías y lenguajes que posibilite la selección de los más adecuados para el desarrollo de la herramienta informática.
6. Definición de los componentes arquitectónicos y de diseño necesarios, para una correcta implementación de la herramienta.
7. Confección de las pruebas que permitan verificar el correcto funcionamiento de la herramienta.
8. Descripción de futuros desarrollos sobre la herramienta implementada.

Métodos teóricos:

- **Analítico Sintético:** se evidencia en el análisis de la documentación asociada al proceso de tratamiento a los medicamentos rechazados en la empresa EMCOMED, así como el estudio de leyes y normas permitiendo la extracción de los elementos más significativos para el desarrollo de la herramienta.

- **Análisis Documental:** se evidencia al realizar un análisis de la documentación proporcionada por la empresa EMCOMED referente al proceso de clasificación de medicamentos rechazados.
- **Modelación:** se utiliza en el diseño de la propuesta de solución, así como en los artefactos generados según la metodología seleccionada.

Métodos empíricos:

- **Entrevista:** se emplea para conocer las necesidades del cliente, determinar cómo funciona el proceso de clasificación de medicamentos rechazados para poder definir los requisitos y características de la solución propuesta.
- **Observación:** se utiliza para la determinación de los puntos claves del proceso de tratamiento a los medicamentos rechazados desde la perspectiva del cliente, para conocer las clases, los métodos de destrucción asociados a cada clase, además del tiempo que tarda el proceso de determinación del método de destrucción a emplear.

Para una mejor comprensión, el presente documento consta de tres capítulos:

Capítulo 1: “Fundamentación teórica”, donde se realiza un estudio relacionado con los principales conceptos y características del proceso de tratamiento de los medicamentos rechazados, específicamente en la determinación de los métodos de destrucción asociados a cada medicamento. Se explican las herramientas y metodologías seleccionadas para realizar la implementación del sistema.

Capítulo 2: “Propuesta de solución”, en el que se describen las características de la solución que se propone como resultado de la presente investigación. Se aborda todo lo referente al análisis y diseño de la herramienta informática para la determinación de los métodos de destrucción en la empresa EMCOMED, así como la definición de los requisitos funcionales y no funcionales.

Capítulo 3: “Implementación y prueba”, donde se describe el proceso de implementación de la propuesta de solución, así como los principales resultados obtenidos en la etapa de pruebas, para garantizar el correcto funcionamiento y el cumplimiento con los requisitos definidos por el cliente.

Capítulo 1: Fundamentación teórica

1.1. Introducción

En el presente capítulo se abordan los elementos teóricos necesarios para darle soporte a la presente investigación. Se describe el proceso de tratamiento de medicamentos rechazados y se definen los conceptos asociados al campo de acción. Se realiza un estudio del impacto social que tienen actualmente los sistemas que tratan medicamentos. Además, se muestra una descripción de las herramientas, tecnologías, metodologías y lenguaje de programación a emplear en el desarrollo del sistema.

1.2. Proceso de tratamiento de los medicamentos rechazados en EMCOMED.

El tratamiento de los medicamentos rechazados tiene un enfoque teórico-financiero, partiendo de las necesidades de cada empresa. Esta política está trazada a partir de un estudio de las características físicas, químicas y organolépticas de los medicamentos y el territorio que enclava la empresa.

Existe la serie de normas ISO 14000, la Ley 81 del Medio Ambiente, así como la Resolución 136/ 2009 que define por manejo integral de los desechos peligrosos: *“la ejecución de todas las operaciones asociadas a cada una de las etapas del ciclo de vida de estos desechos que incluyen la prevención de su generación, la manipulación, almacenamiento, transporte, tratamiento y la disposición final”*. (Miyar, 2009)

“El plan de manejo con los desechos está constituido por las actividades y técnicas en el tratamiento de los residuos sólidos. Ellas involucran la manipulación, acondicionamiento, transporte, transferencia, tratamiento y disposición final o cualquier otro procedimiento técnico operativo utilizado desde su generación. La introducción, fundamentación y objetivos específicos del plan en los cuales se tiene en cuenta las características de la entidad, las regulaciones vigentes en el país en el sector, las regulaciones del CITMA y las Normas Cubanas referidas a esa temática” (López, 2016)

La fase de tratamiento: *“Es la modificación de las características físicas, químicas o biológicas de los desechos, con el objeto de reducir su nocividad,*

controlar su agresividad ambiental y facilitar su gestión. Existen diferentes tipos de tratamientos de los desechos los que se pueden realizar a nivel de entidad o ya en lugares específicos. Los tipos de tratamientos pueden ser:

- *Método de destrucción por incineración.*
- *Método de destrucción por trituración y tratamiento por recursos hídricos.*
- *Método de destrucción por encapsulación”.* (López, 2016)

El proceso de tratamiento a los medicamentos rechazados en EMCOMED comienza cuando llegan a la especialista de calidad los medicamentos rechazados.

Entiéndase por **Medicamento rechazado**: *“los desechos conformados por los medicamentos vencidos o con problemas de calidad que no pueden ser utilizados por el hombre.”* (López, 2016)

Primeramente se categorizan estos medicamentos en tres grupos, denominados clase1, clase 2 o clase 3, para llevar a cabo esta tarea, la especialista se rige y apoya en el *“Formulario de medicamentos de Cuba²”* y en el *“Cuadro básico de clases³”* por componentes, que documentan estas clasificaciones. A continuación se procede a la identificación de los diferentes métodos de destrucción asociados a cada clase (incineración, encapsulación y recursos hídricos). Se tiene como restricción que en los medicamentos tipo spray o tubo solo pueden ser encapsulados.

Posteriormente, la especialista consulta mediante una ficha de costo asociada a cada método de destrucción, el precio por kg que representa su realización. El objetivo que persigue es calcular el costo total que conlleva destruir X cantidad de medicamentos rechazados por un método en específico, cuestión que le

² Formulario de medicamentos de Cuba: documento que contiene todos los medicamentos con el grupo farmacológico asociado.

³ Cuadro básico de clases: documento que contiene todos los grupos farmacológicos asociados a clases (1,2 y 3) con los métodos de destrucción asociado.

permite tomar una decisión sobre cuál de ellos seleccionar. Finalmente son trasladados a su destino final y se procede a la ejecución del método.

Es válido destacar que aparejado al proceso anterior, se deben emitir una serie de reportes en función de las necesidades que presente el país, la empresa EMCOMED o los diferentes suministradores.

1.3. Impacto social, ambiental y económico

“Un motivo de preocupación, es que se ha encontrado en el medio ambiente prácticamente todos los tipos de medicamentos: antibióticos, analgésicos, anticancerosos, ansiolíticos, antiinflamatorios, antisépticos, hormonas, entre otros”. (Tilló Barrufet)

“Con los resultados obtenidos de nuevos estudios de campo y de laboratorio se puede afirmar que los medicamentos deben ser considerados una de las clases de contaminantes del medio ambiente”. (Tilló Barrufet)

Impacto social-ambiental:

“Se han detectado residuos de varios tipos de medicamentos en diferentes compartimentos ambientales (principalmente en aguas residuales, superficiales y profundas, pero también en suelo, aire y biota)”. (Argitalpen, 2016)

“Los grupos farmacológicos cuyos residuos suscitan mayor preocupación son aquellos que matan sus organismos o células diana: antibióticos, antiparasitarios y antimicóticos (por la posibilidad de que se desarrollen cepas resistentes); medios de contraste radiológicos (por su elevada persistencia) y antineoplásicos (por sus propiedades carcinogénicas, mutagénicas y teratógenas). Otros grupos preocupan por su elevado uso: AINE⁴, betabloqueantes, hipolipemiantes y estrógenos”. (Argitalpen, 2016)

En marzo del año 2016 se evidenció un incidente en la zona cercana donde se encuentra una planta para tratamiento de residuales. En este ejemplo específicamente se trataba a un medicamento tratado por el método de encapsulación, esta selección no fue la adecuada y trajo consigo la rotura del

⁴ Antiinflamatorios No Esteroideos (AINE) son un grupo variado y químicamente heterogéneo de fármacos principalmente antiinflamatorios, analgésicos y antipiréticos

envase donde se encontraba el residuo. Las consecuencias fueron desastrosas pues contaminó la tierra y toda la vegetación del lugar, causando la muerte de un total de 49 animales, 30 de estos pertenecientes a una ganadería cercana al lugar, y el resto a un campesino dedicado a la producción de leche de chiva. Estos animales se alimentaban de las plantas como hábito diario y consumían agua de los ríos cercanos.

Impacto económico:

El incremento de la población trae aparejado la producción de millones de medicamentos, muchos de estos son consumidos por el hombre, pero una gran parte son desechados, ya sea por vencimiento, problemas de calidad o sobreproducción. Esto representa una pérdida económica elevada, pues se gastan miles de millones de pesos en fármacos que no serán consumidos, además se le suma la cifra monetaria que equivale a la destrucción de estos medicamentos rechazados, aumentando considerablemente en gastos económicos y pérdidas para los países compradores.

Alrededor de hace 2 años las cifras monetarias dedicadas a la destrucción de los residuos se convirtió en una gran pérdida, pues a la mayoría de los medicamentos rechazados que llegaban a EMCOMED, se les daba como tratamiento la encapsulación. Económicamente se obtuvo una pérdida de 1,02 cuc o 0,9 cuc por cada kg destruido por encapsulación y no por incineración o recursos hídricos. Durante alrededor de 4 años se destruyeron más de 7000 kg de residuos, propiciando una pérdida económica aproximada de 6 300 cuc a 7 140 cuc al no ser destruido por otros de los métodos.

1.4. Análisis de sistemas existentes para el tratamiento de medicamentos.

En una investigación el análisis de herramientas similares, permite la identificación de elementos que pueden ser de utilidad para el desarrollo adecuado de la misma.

Para la selección de los sistemas que debían ser analizados, se tuvo en cuenta como indicadores:

- Los sistemas están enmarcados en el área del tratamiento a los medicamentos.
- Los sistemas tienen como base la utilización de normas, leyes, formularios o libros referentes a su tratamiento.

1.4.1. Sistemas nacionales

En la Universidad de las Ciencias Informáticas (UCI), el Centro de soluciones de informática médica (CESIM) se especializa en el desarrollo de productos de software para la salud, entre sus principales objetivos se encuentra la implementación de sistemas de apoyo a los servicios clínicos y administrativos.

Sistema para el control farmacológico (Synta):

Fue creado con el objetivo de identificar la semejanza de los medicamentos según la reacción adversa a medicamentos (RAM) que estos causan, a través de una funcionalidad que determina la semejanza existente entre los medicamentos atendiendo a las reacciones adversas que ocasionan.

El desarrollo de esta funcionalidad se realizó aplicando la selección de rasgos y la clasificación no supervisada. Para ello se agruparon los medicamentos, según la RAM que causan, mediante el algoritmo de LC-Conceptual. Se calculó el peso informacional asociado a cada rasgo de la RAM calculando los testores típicos, mediante el algoritmo Fast-BR. Se determinó la semejanza entre los medicamentos atendiendo a los efectos adversos que causan.

Formulario nacional de medicamentos de Cuba:

Este sistema contiene una base de datos con 750 fármacos y 500 principios activos empleados en Cuba. La información está organizada por orden alfabético, presentaciones, categorías farmacológicas y riesgos para grupos especiales, entre otras modalidades. Incluye un sistema de escaneo digital para la identificación de los fármacos por su codificación de barras, y brinda información sobre las categorías de riesgo para lactancia y embarazo.

Versión para android del formulario nacional de medicamentos de Cuba:

Hecha en Cuba y destinada a profesionales, técnicos y estudiantes de la salud, con información sobre los medicamentos empleados en el país. Una herramienta que además ha revelado su utilidad entre la población que recibe prescripciones.

El nuevo formato surge como complemento del trabajo realizado por los especialistas del Comité Editorial del Formulario, disponible en formato de libro impreso y digital online en la Biblioteca Virtual de Salud de Cuba, perteneciente al Centro Nacional de Información de Ciencias Médicas (Infomed).

1.4.2. Sistemas internacionales

Herramienta informática para mejorar el control de caducidades en medicamentos:

Tiene como objetivo facilitar el control de caducidades y garantizar la adecuada trazabilidad de los medicamentos existentes en los botiquines. El proceso se inicia con el registro de entrada de los medicamentos (marca y principio activo), lotes, caducidades y cantidad en la aplicación informática. Al finalizar cada mes, se imprime el informe de caducidades por lotes procediendo a su retirada del botiquín.

El programa también permite obtener un informe con los lotes de los medicamentos almacenados, logrando una rápida localización y retirada en caso de alerta farmacéutica de la Agencia Española de Medicamentos y Productos Sanitarios. Este sistema agiliza el control y retirada de los medicamentos, ya que ahora se revisan lotes y no unidades de ampollas o comprimidos.

1.4.3. Resultado del análisis

De la bibliografía consultada, no se encontraron sistemas que traten medicamentos rechazados, ni que permitan determinar los métodos de destrucción. Los sistemas encontrados a nivel nacional e internacional que se dedican al tratamiento de medicamentos, se enmarcan en la gestión de los mismos.

Solamente los sistemas Synta y Formulario Nacional de Medicamentos tienen implementado sus funcionalidades en función de normas regulatorias.

1.5. Técnica a emplear

Plantea (Sancho Caperrini, 2015) que *“una regla en este contexto es una proposición lógica que relaciona dos o más objetos del dominio e incluye dos partes, la premisa y la conclusión, que se suele escribir normalmente como “Si*

premisa, entonces conclusión". Cada una de estas partes es una expresión lógica con una o más afirmaciones objeto-valor conectadas mediante operadores lógicos (y, o, o no)".

Se define utilizar la clasificación supervisada, pues se conocen las clases de salida (métodos de destrucción). Debido a que la estructura del negocio está descrita en base de reglas, que definen los métodos de destrucción asociados a cada medicamento y por las cuales la especialista se guía, se decide utilizar el empleo de reglas como técnica de solución.

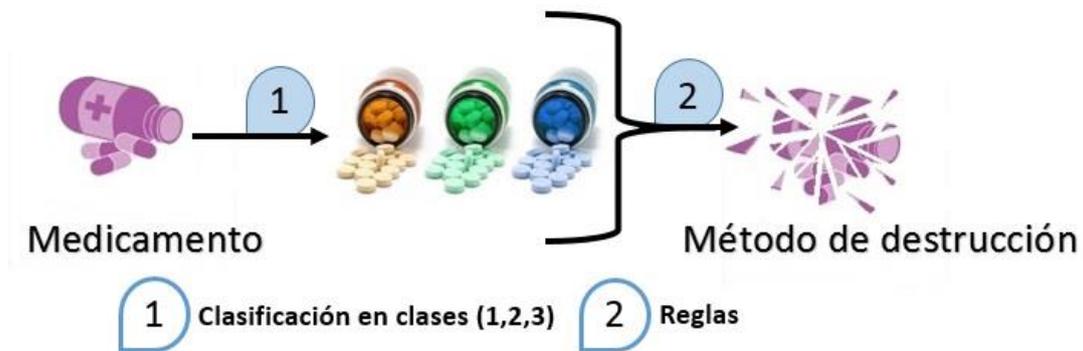


Ilustración 1: Procedimiento para la utilización de la técnica

1.6. Metodologías de desarrollo

En el presente trabajo se hace necesario el empleo de una metodología para guiar el desarrollo de las funcionalidades del sistema.

Las metodologías ágiles se caracterizan por hacer énfasis en la comunicación cara a cara, es decir, se basan en una fuerte y constante interacción, donde clientes y desarrolladores trabajan constantemente juntos, estableciéndose así una estrecha comunicación. Estas metodologías están orientadas al resultado del producto y no a la documentación; exige que el proceso sea adaptable, permitiendo realizar cambios de último momento. Mientras que las metodologías robustas o tradicionales están guiadas por una fuerte planificación. Centran su atención en llevar una documentación exhaustiva de todo el proceso de desarrollo y en cumplir con un plan de proyecto, definido en la fase inicial del mismo.

“AUP es una metodología de desarrollo ágil que heredera de otros paradigmas como la programación extrema (XP⁵) y Rational Unified Process (RUP⁶). Consta de principios y prácticas influyentes en la construcción del software en armonía con la documentación esencial de entregables específicos para el entendimiento de la solución. Entre sus objetivos destaca la reducción del costo del cambio en el proyecto en base a procedimientos iterativos (característica propia de RUP), donde la codificación y pruebas del software se llevan a cabo paralelamente (según XP)” (Galindo, 2012)

Al igual que en RUP, en AUP se establecen cuatro fases que transcurren de manera consecutiva (Flores):

Inicio: permite obtener una comprensión común sobre el alcance del nuevo sistema, entre el cliente y el equipo de desarrollo, definiendo una o varias arquitecturas candidatas para el mismo.

Elaboración: el objetivo es que al equipo de desarrollo se le facilite la comprensión de los requisitos del sistema y pueda validar la arquitectura definida.

Construcción: es la fase en la que el sistema es desarrollado y probado por completo en el entorno de desarrollo.

Transición: el sistema es llevado a entornos de preproducción donde se somete a determinadas pruebas, tanto de validación como de aceptación, siendo desplegado finalmente en los sistemas de producción.

Para el desarrollo del sistema se decide utilizar la variación que propone la UCI de la metodología Proceso Unificado Ágil (AUP, siglas en inglés), ya que a través de los artefactos que propone, permite lograr una clara comprensión del negocio y realizar una descripción más detallada de los requerimientos del sistema a desarrollar.

Para seleccionar esta metodología se tuvo en cuenta que además:

- Existe un intercambio directo con el cliente.
- El equipo de desarrollo es pequeño (2 integrantes).
- Se cuenta con poco tiempo para el desarrollo de la herramienta.

⁵ eXtreme Programming (XP, siglas en inglés)

⁶ Rational Unified Process (RUP, siglas en inglés)

De las 4 fases que propone la metodología AUP (Inicio, Elaboración, Construcción, Transición), la variación AUP-UCI mantiene la fase de Inicio, pero modificando su objetivo. Une las restantes 3 fases de AUP en una sola, llamada Ejecución, y se agrega la fase Cierre. A continuación, para un mejor entendimiento se muestra la siguiente tabla.

Tabla 1: Fases Variación AUP-UCI

Fases AUP	Fases Variación AUP-UCI	Objetivos de las fases Variación AUP-UCI
Inicio	Inicio	En esta fase se realiza un estudio inicial de la organización cliente, que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, Esfuerzo, costo y decidir si se ejecuta o no el proyecto.
Elaboración	Ejecución	En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto, considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, se obtienen los requisitos, se elabora la arquitectura y el diseño y se implementa y se libera el producto.
Construcción		
Transición		
	Cierre	En esta fase se analizan tanto los resultados del

		proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.
--	--	---

AUP-UCI posee varias ventajas, una de ellas se encuentra en la estimación de tiempo y en la forma de planificar el proyecto, elemento importante pues se cuenta con poco tiempo para el desarrollo del sistema.

La metodología AUP-UCI plantea 4 escenarios para la disciplina Requisitos.

Escenario 1: proyectos que modelen el negocio con Caso de Uso del Negocio (CUN) solo pueden modelar el sistema con Caso de Uso del Sistema (CUS).

$CUN + Modelo\ Conceptual\ (MC) = CUS$

Escenario 2: proyectos que modelen el negocio con MC solo pueden modelar el sistema con CUS.

$MC = CUS$

Escenario 3: proyectos que modelen el negocio con Diagrama de Procesos del Negocio (DPN) solo pueden modelar el sistema con Descripción de Requisitos por Proceso (DRP).

$DPN + MC = DRP$

Escenario 4: proyectos que no modelen el negocio solo pueden modelar el sistema con Historias de Usuario (HU).

El sistema propuesto por la presente investigación se centra en la gestión y presentación de la información. Basándose en la importancia que requieren los resultados obtenidos en la búsqueda del método de destrucción apropiado a cada medicamento, se decide utilizar el escenario 2, luego del estudio de estos 4 escenarios.

1.7. Tecnologías y herramientas de desarrollo

A continuación, se muestra una breve descripción de las herramientas utilizadas para la implementación del sistema informático para la determinación de los componentes y métodos de construcción.

1.7.1. Lenguajes empleados

Lenguaje de programación

Un lenguaje de programación permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis.

Para la solución propuesta se utiliza el lenguaje de programación Python en su versión 3.5, permitiendo la adaptabilidad con varios sistemas operativos y el desarrollo de aplicaciones web rápidas y fáciles.

Lenguaje de modelado

El Lenguaje de Modelado Unificado (UML⁷) *“es una de las herramientas más utilizadas en el mundo actual del desarrollo de software, esto se debe a que permite a los desarrolladores crear diseños que engloben sus propósitos de manera sencilla y fácil de comprender para otras personas. El UML está compuesto por diversos elementos gráficos que se combinan para formar diagramas. Proporciona características que permiten organizar y extender los diagramas. Es necesario resaltar que UML indica qué es lo que supuestamente hará el sistema, pero no cómo lo hará”* (Schmuller).

1.7.2. Framework utilizado

Un framework es un ambiente de trabajo que contiene librerías de códigos y módulos que pueden ser reutilizados para el rápido desarrollo de aplicaciones. Se definió para el desarrollo del sistema la utilización de Django en su versión 1.11a1, pues según (Mariños Urquiaga, 2016) *“Django es un framework web de alto nivel que fomenta el desarrollo rápido y el diseño limpio y pragmático”*.

⁷ Unified Modeling Language (UML, siglas en inglés).

Para los archivos CSS y JavaScript se utiliza Materialize en su versión 0.98.0., la cual requiere poca o ninguna configuración.

1.7.3. Herramienta CASE

“Las herramientas CASE⁸ son un conjunto de herramientas y métodos asociados que proporcionan asistencia automatizada en el proceso de desarrollo del software a lo largo de su ciclo de vida. Fueron desarrolladas para automatizar esos procesos y facilitar las tareas de coordinación de los eventos que necesitan ser mejorados en el ciclo de desarrollo de software”. (Mesa, 2011)

Se selecciona Visual Paradigm for UML en su versión 8.0 como herramienta CASE para el modelado de la propuesta de solución, permitiendo representar todo tipo de diagramas en el ciclo de vida del desarrollo de software.

1.7.4. Entorno de desarrollo integrado

Un entorno de desarrollo integrado o IDE (siglas provenientes del inglés Integrated Development Environment), brinda un conjunto de componentes que hacen más fácil la programación y conforma un ambiente favorable para los desarrolladores. El IDE utilizado es Pycharm en su versión 2016.3.2, el cual soporta intérpretes de Python 3.5 y permite la integración con el framework Django.

1.7.5. Sistema gestor de bases de datos

“SQLite es una biblioteca escrita en lenguaje C que implementa un Sistema de gestión de bases de datos transaccionales SQL auto-contenido, sin servidor y sin configuración. El código de SQLite es de dominio público y libre para cualquier uso, ya sea comercial o privado”. (Velazsco, 2014)

Algunas de las características de SQLite son las siguientes (Velazsco, 2014):

1. SQLite soporta múltiples tablas, índices, triggers y vistas.
2. Lee y escribe directamente sobre archivos que se encuentran en el disco duro.

⁸ Ingeniería de Software Asistida por Computadora (CASE, siglas en inglés).

3. El formato de la base de datos es multiplataforma y se puede utilizar el mismo archivo en un sistema de 32 y 64 bits.
4. Utiliza el espacio en disco que es realmente necesario en cada momento.
5. SQL realiza operaciones de manera eficiente y es más rápido que MySQL y PostgreSQL.
6. Cuenta con diversas interfaces API, lo que permite trabajar con C++, PHP, Python, Groovy, entre otros.
7. Es totalmente auto contenida es decir que no tiene dependencias externas.
8. Cuenta con librerías de acceso para muchos lenguajes de programación.
9. Soporta funciones SQL definidas por el usuario.
- 10.El código fuente es de dominio público y se encuentra muy bien documentado.

Se utiliza SQLite en su versión 3.11.0 para el manejo de la base de datos, permitiendo esta versión, bases de datos de hasta 2 Terabytes de tamaño.

1.8. Conclusiones parciales

Luego de realizar un estudio de los principales conceptos asociados al tema de la investigación, el análisis de sistemas que traten medicamentos y el estudio de la técnica a emplear para el desarrollo de la herramienta se arriba a las siguientes conclusiones parciales:

- El estudio de ejemplos concretos donde se haya ejecutado un método de destrucción inadecuado, permitió verificar la necesidad de la problemática planteada.
- Los sistemas existentes consultados no tratan medicamentos rechazados, ni ninguna actividad incluida dentro de este proceso, por lo que infiere que no representan una vía de solución para nuestro problema.
- La determinación de las reglas como técnica de solución es la adecuada según la naturaleza del problema.
- La metodología AUP-UCI es la adecuada para guiar el desarrollo del software, ya que el equipo es pequeño y se cuenta con poco tiempo. Las tecnologías y herramientas a utilizar son las apropiadas para el desarrollo de la herramienta.

Capítulo 2: Propuesta de solución

2.1. Introducción

En el presente capítulo se exponen los elementos que permiten describir la propuesta de solución, se definen y diseñan todos los artefactos teniendo en cuenta la metodología de desarrollo AUP-UCI en el escenario 2. Se describe el modelo conceptual y el diseño de la herramienta informática propuesta. Se muestran los diagramas de clases del diseño, el diagrama de paquetes y el diseño de la base de datos. Se define el estilo arquitectónico y los patrones de diseño utilizados.

2.2. Modelo conceptual del negocio

En el modelo conceptual del negocio se definen cuáles son y cómo se relacionan los conceptos relevantes en la descripción del problema, en este caso se describen los conceptos relacionados con el negocio del sistema a desarrollar.

Conceptos que componen el modelo:

Especialista de calidad: persona encargada de realizar todas las operaciones para clasificar el medicamento y determinar su método de destrucción.

Método de destrucción: métodos de destrucción asociados a los medicamentos.

Reporte de medicamentos destruidos: reportes de los medicamentos destruidos con sus costos asociados.

Medicamentos rechazados: datos referentes a los medicamentos rechazados.

Reglas: reglas que se derivan de los documentos que utiliza la especialista de calidad para determinar el método de destrucción.

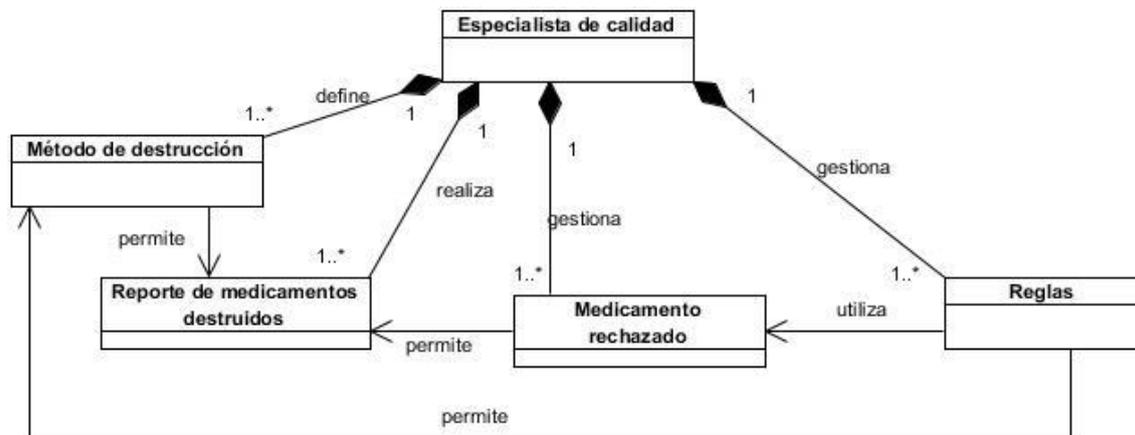


Ilustración 1: Modelo Conceptual

2.3. Variables lingüísticas.

Para el tratamiento y determinación del método de destrucción se seleccionaron las siguientes variables lingüísticas:

Presentación: compuesto por varios rasgos que definen la composición física del medicamento.

Grupo farmacológico: define el grupo farmacológico (analgésico, vitamina, etcétera).

Clase: define la clase a la cual pertenece el medicamento en dependencia de su grupo farmacológico.

Método de destrucción: define el método de destrucción asociado al medicamento.

Los valores de los rasgos clase y método de destrucción son generados por el sistema. Este determina la clase de un medicamento y los métodos de destrucción asociados a él.

2.4. Propuesta de solución

La propuesta de solución está dividida en dos fases, una primera fase que permite la clasificación inicial del medicamento, y una segunda fase para determinar los métodos de destrucción asociados a cada medicamento, con su correspondiente costo, para finalmente proceder a la sección del más adecuado. Ver ilustración 2.

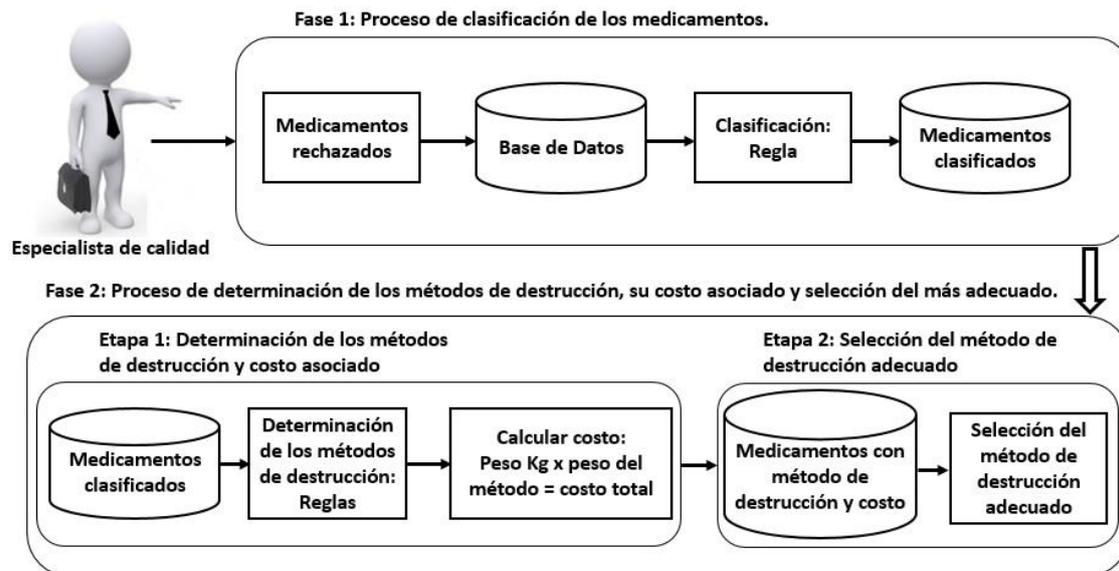


Ilustración 2: Flujo de la propuesta de solución

Fase 1: Proceso de clasificación de medicamentos.

Entrada: Medicamento rechazado.

Salida: Base de Datos con medicamentos catalogados en clase 1, clase 2 o clase 3.

Paso 1: Insertar el medicamento rechazado en la Base de datos.

Paso 2: Clasificar medicamento según su grupo farmacológico.

Ejemplo de reglas utilizadas para la clasificación del medicamento rechazado atendiendo al grupo farmacológico para clasificarlo en clases:

R1: IF grupo farmacológico = (Analgésicos, Antiarrítmico, Anestésico, Antihipertensivo, Antiulceroso, Antianginoso, Antifúngico, Antiemético, Antihistamínico, Antianémico, Anticoagulante, entre otros) THEM clase = 1.

R2: IF grupo farmacológico = (Antibióticos, Antiparasitarios, Antiherpéticos, Antimicóticos Antidiabético, Antídoto, Hormona, Pre-anestésico, Estrógenos, Endometriosis Anticonceptivos, entre otros) THEM clase = 2.

R3: IF grupo farmacológico = (Citostáticos o Antiparkinsoniano) THEM clase = 3.

Paso 3: Registrar los medicamentos rechazados con la clase asociada en la base de datos.

Fase 2: Proceso de determinación de los métodos de destrucción, su costo asociado y selección del más adecuado.

Etapa 1: Determinación de los métodos de destrucción asociados a cada medicamento y su correspondiente costo.

Entrada: Base de Datos con los medicamentos clasificados.

Salida: Base de Datos con los métodos de destrucción asociados a cada medicamento y el costo económico de cada método.

Paso 1: Determinar los métodos de destrucción asociados a cada medicamento.

Ejemplo de algunas reglas utilizadas para la determinación de los métodos de destrucción:

R1: IF clase = 1 AND presentación = (ámpula, bulbo, frasco o colirio) THEM métodos =encapsulación o recursos hídricos.

R2: IF clase = 2 AND presentación = (tableta o polvo) THEM métodos = incineración o encapsulación.

R3: IF presentación = (spray o tubo) THEM métodos = encapsulación.

R4: IF clase = 3 AND presentación= (tableta o polvo) THEM métodos = encapsulación.

Paso 2: Calcular el costo de los métodos de destrucción por medicamento.

Conversión del peso del medicamento a Kg.

Peso Kg x costo método de destrucción = costo total.

Paso 3: Registrar los resultados en la base de datos.

Etapa 2: Selección del método de destrucción adecuado.

Entrada: Base de Datos con los métodos de destrucción asociados a cada medicamento y el costo económico de cada método.

Salida: Medicamento con el método seleccionado.

Paso 1: Mostrar los métodos de destrucción asociados a cada medicamento con su correspondiente costo.

Paso 2: Seleccionar un método de destrucción.

2.5. Especificación de los requisitos del sistema.

Según (Sommerville, 2005), *“los requisitos del sistema especifican qué es lo que debe hacer (sus funciones) y sus propiedades esenciales y deseables”*.

A continuación, se especifican los requisitos funcionales y no funcionales del sistema informático a desarrollar.

2.5.1. Requisitos funcionales del sistema

Los requisitos funcionales de un software definen los servicios que debe proporcionar el sistema.

Tabla 2: Requisitos funcionales del sistema

Número	Nombre	Descripción
RF1	Autenticar usuario	El sistema permitirá que la especialista de calidad pueda autenticarse para acceder a las funcionalidades.
RF2	Editar usuario	El sistema permitirá que la especialista de calidad pueda cambiar sus datos dentro de los que se incluye la contraseña.
RF3	Buscar medicamento	El sistema permitirá que la especialista de calidad pueda buscar un medicamento.
RF4	Insertar medicamento	El sistema permitirá que la especialista de calidad pueda insertar un medicamento.
RF5	Identificar la clase a la que pertenece un medicamento	El sistema permitirá catalogar el medicamento en dependencia de la clase a la que pertenezca, según su grupo farmacológico.

RF6	Modificar medicamento	El sistema permitirá que la especialista de calidad pueda modificar un medicamento.
RF7	Eliminar medicamento	El sistema permitirá que la especialista de calidad pueda eliminar un medicamento.
RF8	Seleccionar método de destrucción	El sistema permitirá que la especialista de calidad pueda seleccionar el método de destrucción más adecuado para un medicamento, teniendo en cuenta el costo asociado.
RF9	Determinar métodos de destrucción	El sistema permitirá que la especialista de calidad obtenga los métodos de destrucción correspondientes al medicamento, y además calcular el costo asociado.
RF10	Generar reporte	El sistema permitirá que la especialista de calidad obtenga un reporte de medicamentos. Permitiendo exportar el mismo para el manejo de la información.
RF11	Insertar regla	El sistema permitirá que la especialista de calidad pueda insertar una regla.
RF12	Modificar regla	El sistema permitirá que la especialista de calidad pueda modificar una regla.

RF13	Eliminar regla	El sistema permitirá que la especialista de calidad pueda eliminar una regla.
RF14	Ayuda	El sistema permitirá que la especialista de calidad se instruya para acceder a las funcionalidades de la herramienta informática.

2.5.2. Requisitos no funcionales del sistema

Las características del funcionamiento del sistema son descritas por los requisitos no funcionales, pues permiten que la aplicación sea confiable, rápida, atractiva y usable.

Disponibilidad

- Para el usuario el sistema debe estar disponible en el momento que lo necesite.

Usabilidad

- La interfaz de la aplicación debe permitirle al usuario sin experiencia en la herramienta informática DEMED poder interactuar fácilmente con la misma y así adaptarse rápidamente.

Apariencia e Interfaz

- La aplicación deberá poseer una interfaz fácil de usar por los usuarios.
- Todos los mensajes y textos en pantalla aparecerán en idioma español.

Hardware

- Para las PC clientes:

1. Requerimientos mínimos 512MB de RAM recomendada o superior.
2. Tarjeta de red para establecer la conexión.

- Para el servidor:

1. Computador con procesador Dual Core que es el tipo de microprocesador que utilizan los servidores de la empresa EMCOMED, 1 GB de memoria RAM, 500 GB de disco duro.

2. Tarjeta de red para establecer la conexión.

Software

- Para las PC clientes:

1. Ubuntu mate con interfaz gráfica y soporte para conectarse a la red y Sistema Operativo Windows 7 o Windows 8.

2. Se requerirá el uso de un navegador web, preferentemente Mozilla Firefox en su versión 30.0 o superior para la utilización del sistema.

- Para el servidor:

Se requiere el Sistema Operativo Windows 7, Windows 8 o Ubuntu mate. Como framework Django en su versión 1.11a1, las bibliotecas necesarias como datatable, entre otras. Se utiliza Sqlite en su versión 3.11.0 para el manejo de la base de datos.

2.6. Definición de los actores

Para (Merseguer, 2010), un actor es *“un agente, alguien o algo que solicita un servicio al sistema o actúa como catalizador para que ocurra algo. Un actor representa un rol que es jugado por una persona, un dispositivo hardware, incluso otro sistema”*.

Tabla 3: Actor relacionado con el sistema

Actor	Objetivo
Especialista de Calidad	La especialista de calidad podrá subir un medicamento al sistema y obtener los métodos de destrucción con su costo asociado, así como seleccionar el método que se va a emplear. Además podrá solicitar reportes de medicamentos.

2.7. Diagrama de casos de uso del sistema

El diagrama de casos de uso permite describir la secuencia de eventos que los actores utilizan para completar un proceso a través del sistema.

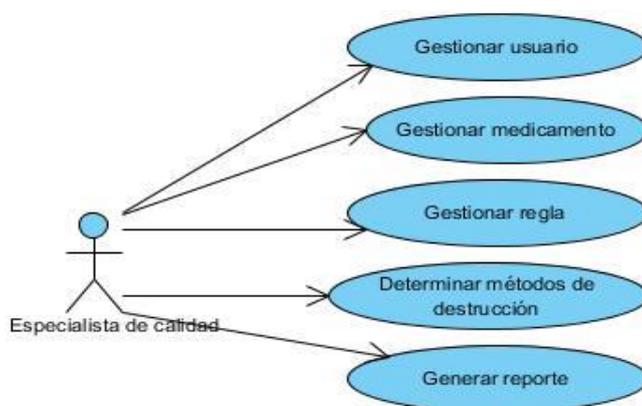


Ilustración 3: Diagrama de casos de uso del sistema

2.7.1. Descripción de los casos de uso del sistema

A continuación se realiza una descripción de los casos de uso del sistema, para consultar el resto de las descripciones ver **Anexo I: Descripción de casos de usos del sistema.**

Tabla 4: Descripción CU4. Determinar métodos de destrucción.

Objetivo	Determinar los métodos de destrucción asociados a un medicamento con su costo asociado para que la especialista de calidad decida cual seleccionar.
Actores	Especialista de Calidad
Resumen	El caso de uso comienza cuando la especialista de calidad desea conocer los métodos de destrucción de los medicamentos y selecciona la opción Método de destrucción.
Complejidad	Baja
Prioridad	Alta
Precondiciones	
Post-condiciones	Se determina el método de destrucción del medicamento.

Flujo de eventos		
Flujo básico		
No.	Actor	Sistema
1.	Selecciona la opción Método de destrucción.	
2.		Determina y guarda en la BD los métodos de destrucción asociados al medicamento con su costo asociado.

Tabla 5: Descripción CU5. Generar reporte.

Objetivo	Exportar un reporte de los medicamentos.	
Actores	Especialista de Calidad	
Resumen	El caso de uso comienza cuando la especialista de calidad desea solicitar un reporte de los medicamentos y selecciona la opción Solicitar reporte.	
Complejidad	Baja	
Prioridad	Alta	
Precondiciones		
Post-condiciones		
Flujo de eventos		
Flujo básico		
No.	Actor	Sistema
1.	Selecciona los medicamentos y las columnas que quiere aparezcan en el reporte.	
2.	Selecciona la opción Solicitar reporte.	

3.		Exporta los medicamentos y sus valores a un documento txt.
----	--	--

2.8. Diseño de la arquitectura

Apunta (Cervantes, 2010) que la arquitectura de software “*constituye un modelo comprensible de cómo está estructurado el sistema y cómo trabajan juntos sus componentes*”

Se empleará la arquitectura cliente-servidor para el desarrollo del sistema informático, ya que el usuario podrá acceder a la aplicación web desde cualquier PC cliente, pues esta se encontrará alojada en el servidor de aplicaciones.

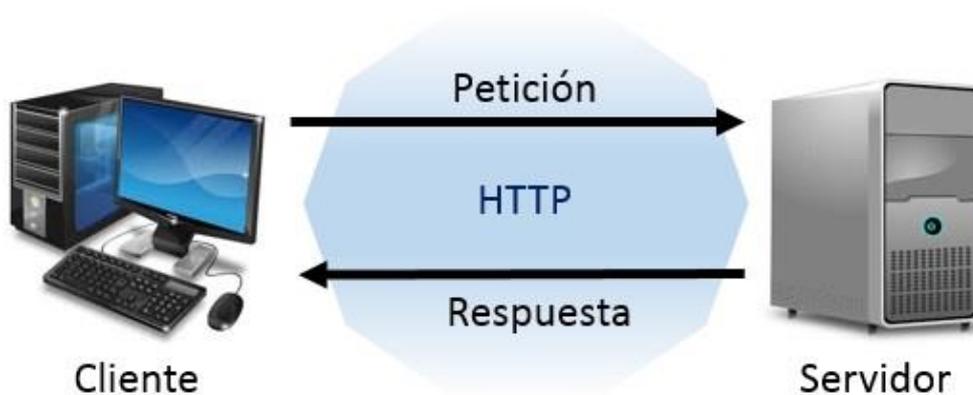


Ilustración 4: Arquitectura Cliente –Servidor

El framework Django define una transformación de la arquitectura Modelo-Vista-Controlador (MVC), llamada MTV (Model-Template-View). El modelo en Django sigue siendo Modelo, el controlador se llama Vista y la vista se llama plantilla (Template). Esta forma de trabajar permite que sea pragmático⁹.

- ❖ Capa Modelo (models): Define los datos almacenados; se encuentra en forma de clases de python, cada tipo de dato que debe ser almacenado se encuentra en una variable con ciertos parámetros, posee métodos también. Todo esto permite indicar y controlar el comportamiento de los datos.
- ❖ Capa Plantilla (templates): Recibe los datos de la vista y luego los organiza para la presentación al navegador web.

⁹ Actúa dando prioridad a las consideraciones prácticas.

- ❖ Capa Vista (views): Determina qué datos serán visualizados sin encargarse del estilo de la presentación de los mismos.

La siguiente imagen describe el funcionamiento del patrón arquitectónico MTV en Django.



Ilustración 5: Funcionamiento del MTV en Django.

2.9. Etapa de diseño

En esta etapa, para obtener la correcta implementación de la solución propuesta, se hace necesario precisar los elementos imprescindibles para lograrlo. Se define y muestra el diseño de la interfaz, el diagrama de paquetes y los diagrama de clases del diseño.

2.9.1. Diseño de interfaz

Para definir los colores de la interfaz de la herramienta informática se realizó un análisis de los más utilizados en la medicina. El resultado de este análisis arrojó que los colores se usan en la medicina según el significado de cada color y lo que se desee lograr. Se solicita por el cliente que la herramienta contemple los significados correspondientes a los colores azul y verde.

Azul: brinda seguridad y responsabilidad.

Verde: brinda riqueza, salud y prestigio.

Para el estilo y diseño se hace uso de CSS 3, el cual junto con HTML 5 y JavaScript, posibilitan la creación páginas visualmente atractivas e interfaces de usuario para aplicaciones web.

CSS está diseñado principalmente para marcar la separación del contenido del documento y la forma de presentación de este, características tales como las capas, los colores y las fuentes. Esta separación busca mejorar la accesibilidad del documento, proveer más flexibilidad y control en la especificación de características de presentación, permitiendo que varios documentos HTML compartan un mismo estilo usando una sola hoja de estilos separada en un archivo CSS para reducir la complejidad y la repetición de código en la estructura del documento.

Se define además un logo identificativo de la herramienta informática, el cual lleva por nombre DEMED (Destrucción de medicamentos o Método de destrucción de medicamentos), haciendo alusión a cualquiera de los dos nombres antes descrito.



Ilustración 6: Logo de la herramienta informática

La herramienta informática contiene varias funcionalidades:

➤ **Autenticar:**

Este es un formulario de autenticación con un fondo blanco y un borde gris. En la parte superior central se encuentra el logo de DEMED. Debajo del logo, el texto "Sistema para la Destrucción de Medicamentos" aparece en un color verde. El formulario contiene dos campos de entrada: el primero está etiquetado "Nombre de usuario" y contiene el texto "administrator"; el segundo está etiquetado "Contraseña" y muestra caracteres ocultos por puntos. Debajo de estos campos hay un checkbox con el texto "Recordar credenciales". En la parte inferior del formulario, hay un botón rectangular de color azul con el texto "ENTRAR" en blanco.

Ilustración 7: Autenticar usuario

➤ **Menú superior:**

Buscar: funcionalidad que permite buscar medicamentos.

Insertar: funcionalidad que permite insertar un medicamento o regla en dependencia de en qué tabla (medicamento o regla) se encuentre el usuario.

Mostrar\Ocultar: funcionalidad que permite mostrar u ocultar las columnas de la tabla.

Exportar reporte: funcionalidad que permite exportar un reporte realizado por la especialista.

➤ **Menú principal a la izquierda:**

Medicamentos: funcionalidad que contiene todos los medicamentos insertados en la BD. Aparece por cada medicamento la funcionalidad de eliminar, modificar y seleccionar método de destrucción.

Reglas: funcionalidad que contiene todas las reglas insertadas en la BD. Aparece por cada regla la funcionalidad de eliminar y modificar.

Método de destrucción: funcionalidad que permite determinar los métodos de destrucción asociados a los medicamentos de la BD. Además se calcula el costo asociado de cada método por medicamento.

Editar usuario: funcionalidad que permite editar los datos del usuario.

Salir: funcionalidad que permite salir del sistema.

Ayuda: funcionalidad que muestra una ayuda al usuario explicándole cada funcionalidad y como moverse por el sistema.

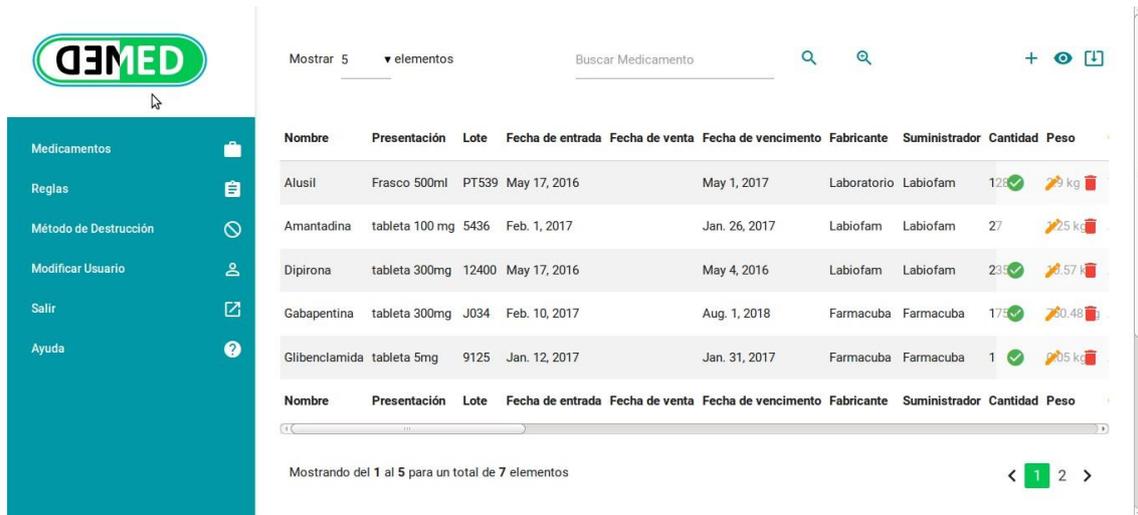


Ilustración 8: Interfaz de la herramienta informática DEMED

2.9.2. Diagrama de paquetes

Tiene como objetivo conformar la estructura lógica del sistema.

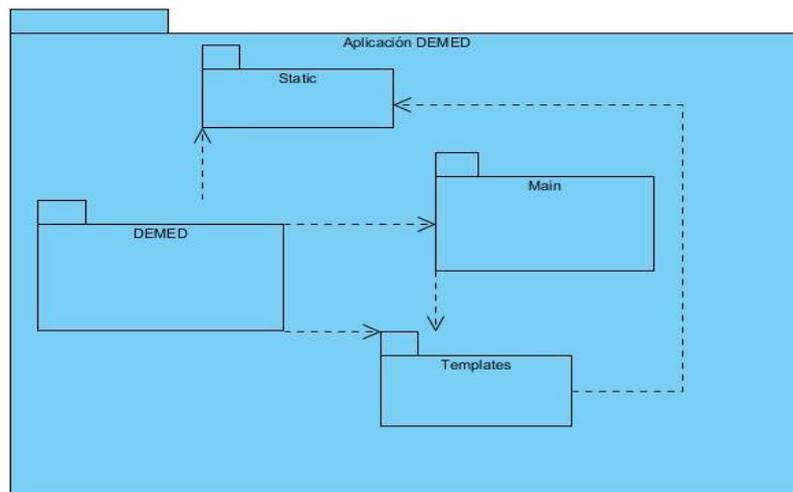


Ilustración 9: Diagrama de paquetes

2.9.3. Diagrama de clases de diseño utilizando estereotipos web

Los diagramas de clases forman una estructura estática, en la cual, la representación de los requisitos se lleva a cabo a través de las clases del sistema y sus interrelaciones.

Para consultar el resto de los diagramas de clases de diseño utilizando estereotipos web ver **Anexo II: Diagramas de clases de diseño.**

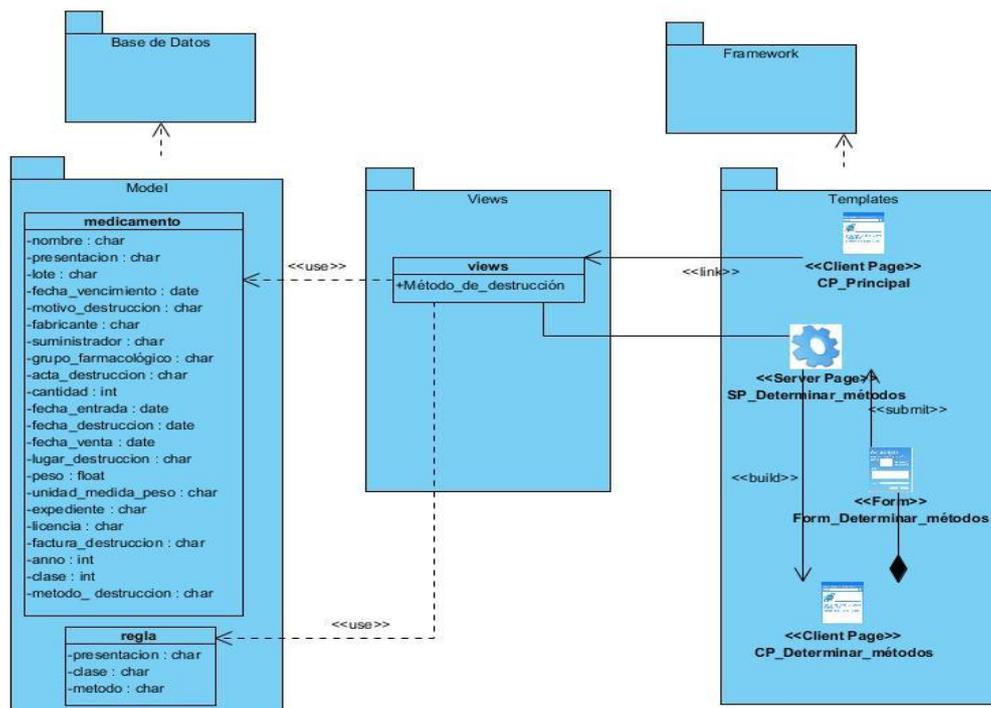


Ilustración 10: Diagrama de clases de diseño RF Determinar métodos de destrucción

2.10. Modelo físico de los datos

En esta fase se describen las representaciones físicas de los datos utilizados en la herramienta informática que serán almacenados en la base de datos. En el diagrama que representa el modelo físico de los datos los elementos esenciales son las entidades, los atributos y las relaciones entre las entidades.

- ❖ Entidades: objetos que el sistema necesita guardar su información, en este caso, las entidades son los medicamentos, el método de destrucción y las reglas.
- ❖ Atributos: características de las entidades, las cuales se clasifican en obligatorios, opcionales, claves foráneas y claves primarias.
- ❖ Relaciones: muestra la relación entre las entidades.

Para el desarrollo de las nuevas funcionalidades propuestas, se utiliza la tabla *Medicamento*, *Metodo_destruccion* y *Regla* de la base de datos que posee el sistema DEMED.

Medicamento: contiene toda la información referente a los almacenados para hallarles los métodos de destrucción asociado.

Metodo_destruccion: esta tabla contiene toda la información de los métodos de destrucción.

Regla: contiene toda la información referente a las reglas que definen la asignación y relación de métodos con medicamentos.

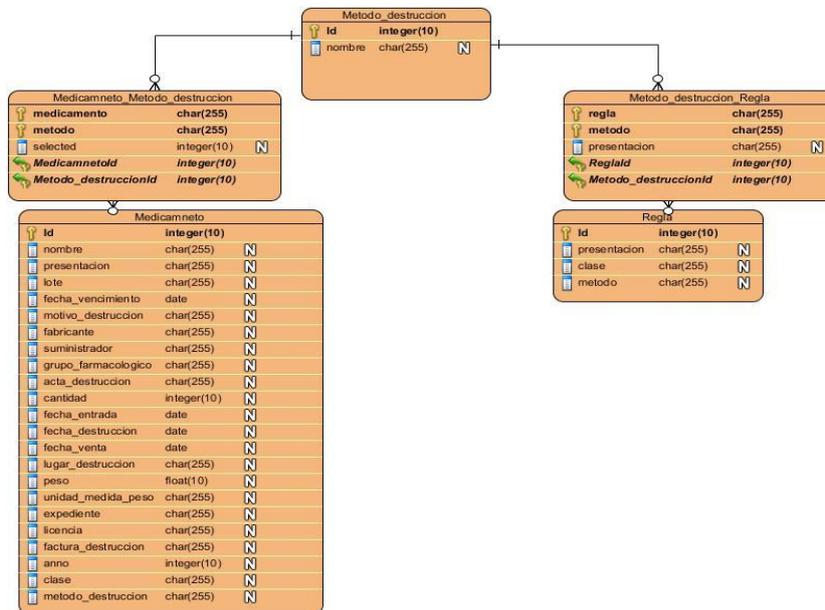


Ilustración 11: Modelo físico de la base de datos

2.11. Patrones de diseño

Los patrones de diseño según (Larman, 1999) *“brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares”*.

Patrones GRASP

Patrones Generales de Software para Asignación de Responsabilidades (GRASP¹⁰, siglas en inglés). Para (Larman, 1999) los patrones GRASP *“describen los principios fundamentales de la asignación de responsabilidades a objetos expresados en forma de patrones”*.

Existen 9 patrones GRASP: experto, creador, controlador, bajo acoplamiento, alta cohesión, polimorfismo, fabricación pura, indirección y variaciones protegidas. Django que es el framework que se utiliza para el desarrollo de las

¹⁰ GRASP, acrónimo de General Responsibility Assignment Software Patterns.

funcionalidades del sistema DEMED, implementa 5 de los patrones antes mencionados (alta cohesión, experto, bajo acoplamiento, controlador y creador).

Alta cohesión: *“es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realizan un trabajo enorme”* (Pressman, 2005).

Ejemplo: de manera coherente y precisa, las clases del modelo describen la estructura de las tablas de la base de datos.

Experto: según (Pressman, 2005), el patrón experto *“es el encargado de asignar la responsabilidad de la creación de un objeto o la implementación de un método a una clase que contenga toda la información necesaria para cumplir con dicha responsabilidad”*.

Ejemplo: en el modelo la clase Medicamento se encarga de la lógica de todo lo relacionado con un medicamento.

Bajo acoplamiento: para (Pressman, 2005) *“el acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con qué las conoce y con qué recurre a ellas. En tal sentido, el término bajo acoplamiento significa que una clase no depende de muchas clases”*.

Ejemplo: las vistas (views) implementan funciones y métodos que son llamados por las URLs, pero cualquier cambio que se realice a una función o método, no afecta la URL.

Controlador: se implementa como parte del patrón arquitectónico utilizado, sirve de intermediario entre una determinada interfaz y el algoritmo que la implementa recibiendo los datos del usuario y enviándolos a las distintas clases según el método llamado.

Para (Larman, 1999) *“este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, para aumentar la reutilización de código y a la vez tener un mayor control. Se recomienda dividir los eventos del sistema en el mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento”*.

Ejemplo: Views y index.

Creador: el mismo tiene como objetivo según (Larman, 1999) *“asignar a la clase B la responsabilidad de crear una instancia de clase A”*.

Ejemplo: en la *views*, la función *add* es responsable de la creación de objetos de tipo medicamento del *model*.

Patrones GOF

“Los patrones GOF acrónimo de Gang Of Four, o Pandilla de los Cuatro por su traducción al español se definen como el conjunto de tareas o reglas para resolver problemas que surgen durante el desarrollo del software”. (Larman, 1999)

Decorador: *“es un patrón de tipo estructural encargado de asociar responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la especialización mediante herencia, cuando se trata de añadir funcionalidades. Brinda mayor flexibilidad que la herencia estática, permitiendo, entre otras cosas, añadir una funcionalidad dos o más veces. Propicia concentrar en lo alto de la jerarquía de clases guiadas por las responsabilidades. De esta forma las nuevas funcionalidades se componen de piezas simples que se crean y se combinan con facilidad, independientemente de los objetos cuyo comportamiento extienden”*. (Wesley, 2006).

El uso de este patrón en la aplicación se evidencia en el uso del decorador `login_required` el cual es el encargado de garantizar que el usuario este autenticado en el sistema al acceder a alguna URL.

Ejemplo: `@login_required(login_url='/login')`

`Def index(request): ...`

2.12. Conclusiones parciales

- El análisis del negocio mediante el modelo conceptual confirmó que fue acertada la decisión de aplicar el escenario 2 de la metodología de desarrollo utilizada.

- La elaboración de los artefactos generados según la metodología seleccionada, garantizó la guía adecuada para el entendimiento y desarrollo de la propuesta de solución.
- Django como framework de desarrollo permitió el uso de patrones que aumentan la seguridad del sistema

Capítulo 3: Implementación y prueba

3.1. Introducción

En este capítulo se describen aspectos relacionados con la implementación y validación de la herramienta informática desarrollada. Se exponen los principales resultados obtenidos durante la etapa de prueba, para garantizar su correcto funcionamiento y el cumplimiento con los requisitos definidos por el cliente.

3.2. Etapa de implementación

En esta etapa, según apunta (González, 2007) “se realiza la codificación, se realiza la programación de la solución diseñada, en el lenguaje de programación y plataforma elegida a tal efecto teniendo en cuenta las restricciones obtenidas en la etapa de análisis”.

3.2.1. Diagrama de componente

Para (Jacobson, 2000) un componente “muestra varios componentes de un sistema, describiendo sus elementos físicos y sus dependencias”.

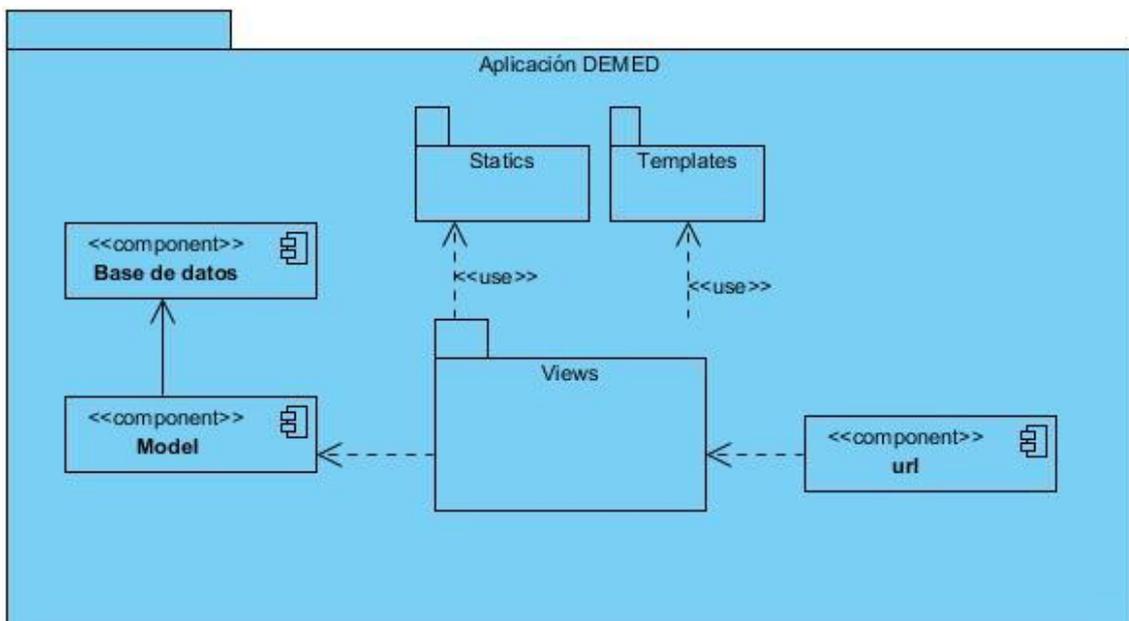


Ilustración 12: Diagrama de componentes del sistema

3.3. Estándares de codificación empleados

Los estándares de codificación permiten establecer una forma de programación única, haciendo más entendible el código fuente para otros desarrolladores, facilitando el mantenimiento posterior de las aplicaciones desarrolladas.

Estándar de codificación utilizado:

- Al inicio de cada clase se realiza un comentario explicando el objetivo de la misma. Cada línea de un comentario empieza con # (numeral).
- Iniciar el nombre de las variables de las clases con letra inicial mayúscula y en caso de ser un nombre compuesto se utiliza underscore (_).
- Iniciar el nombre de las variables y los métodos con letra inicial minúscula y en caso de ser un nombre compuesto se utiliza underscore (_).
- Entre operadores lógicos y aritméticos se utiliza espacios en blanco.
- Antes y después de la declaración de una clase, de una estructura o de la implementación de una función, se dejan dos líneas en blanco.

3.4. Pruebas

Las pruebas son *“un componente importante de calidad del software, proceso de ejecutar un programa para detectar errores”* (Jovanović, 2008).

En esta fase se realizan un conjunto de actividades para verificar la calidad del producto, siendo el principal objetivo detectar y solucionar los errores que presente la herramienta desarrollada según los requisitos definidos.

Se realizan pruebas unitarias para el análisis del correcto funcionamiento del código, de validación para verificar que sean correctas las entradas y salidas de datos en el sistema y de despliegue para la ejercitación del software en el entorno que debe operar.

3.4.1. Pruebas unitarias

Se realizaron pruebas unitarias, para comprobar el correcto funcionamiento de la implementación de la herramienta informática desarrollada. La definición de las pruebas se realizaron dentro del archivo tests.py, ya creado por el framework Django, además se utilizó la biblioteca de python (unit testing). Para ejecutar los test o iniciar el servidor de prueba se hace uso del comando `“python manage.py test”`.

Se realizaron pruebas unitarias en las views.

```

639 @login_required(login_url='/login')
640 def calc_method(request):
641     for medicamento in Medicamento.objects.all():
642         if not Medicamento_Metodo_destruccion.objects.filter(medicamento=medicamento):
643             for regla in Regla.objects.all():
644                 if medicamento.clase == regla.clase:
645                     match = False
646                     for palabra in regla.presentacion.split(';'):
647                         if palabra.strip().lower() in medicamento.presentacion.lower():
648                             match = True
649                             metodos = Regla_Metodo_destruccion.objects.filter(regla=regla)
650                             for metodo in metodos:
651                                 Medicamento_Metodo_destruccion(
652                                     medicamento=medicamento,
653                                     metodo=metodo.metodo
654                                 ).save()
655                             break
656                     if match:
657                         break
658     return HttpResponseRedirect('/')

```

Ilustración 13: Prueba unitaria a la funcionalidad Determinar métodos de destrucción

Module ↓	statements	missing	excluded	coverage
main.py	0	0	0	100%
main/admin.py	1	0	0	100%
main/apps.py	3	0	0	100%
main/migrations/0001_initial.py	6	0	0	100%
main/migrations/0002_auto_20170311_2156.py	5	0	0	100%
main/migrations/0003_auto_20170325_1643.py	6	0	0	100%
main/migrations/0004_auto_20170427_0024.py	5	0	0	100%
main/migrations/0005_auto_20170427_1329.py	6	0	0	100%
main/migrations/0006_auto_20170427_1503.py	5	0	0	100%
main/migrations/0007_auto_20170427_1508.py	5	0	0	100%
main/migrations.py	0	0	0	100%
main/models.py	56	4	0	93%
main/views.py	479	301	0	37%
Total	577	305	0	47%

Ilustración 14: Resultado de las pruebas unitarias

3.4.2. Pruebas de validación

Las pruebas de validación se realizaron a través de casos de prueba, los cuales son según (Aristegui O, 2010) “*un conjunto de acciones con resultados y salidas previstas basadas en los requisitos de especificación del sistema*”. La partición equivalente y gráfico de prueba fueron las técnicas utilizadas para comprobar el funcionamiento del sistema.

Para (Ruiz Tenorio, 2010) *“la partición equivalente es un método de prueba de caja negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba”*. Para representar los diferentes estados posibles en cada condición de entrada, calificando cada estado en válidos o inválidos, se utilizó la técnica de partición equivalente.

El gráfico de prueba es otra de las técnicas empleadas en el desarrollo de las pruebas de caja negra, basada según (Pressman, 2005) en *“una colección de nodos que representan objetos, enlaces que representan la relación entre objetos, pesos de nodo que describen las propiedades de un nodo (como un valor de datos o un comportamiento de estado específico) y pesos de enlace que describen algunas características de un enlace”*. La utilización de esta técnica permite validar funcionalidades que no requieren entrada de datos.

Para verificar el cumplimiento de los requisitos funcionales, a continuación, se muestran los casos de prueba realizados utilizando las técnicas antes mencionadas. Para consultar el resto de los casos de pruebas ver **Anexo III: Ejemplo de casos de prueba.**

Tabla 6: Caso de prueba de partición equivalente del RF1 Autenticar usuario

Escenario	Descripción	Datos	Respuesta del sistema	Flujo central
EC 1.1	El usuario selecciona la opción entrar con usuario y contraseña correctos.	V	Concede la entrada al usuario al sistema y sus funcionalidades.	El usuario introduce su usuario y contraseña, selecciona la opción aceptar.
	El usuario introduce el usuario y la contraseña para acceder al sistema y selecciona la opción entrar.	Administrador		
		Administrador		

EC 1.2	El usuario introduce el usuario y la contraseña para acceder al sistema y selecciona la opción entrar.	I Ldcastro Demed.@ .17	Muestra un mensaje en dependencia del valor del campo que se haya introducido incorrectamente.	El usuario introduce su usuario y contraseña, selecciona la opción aceptar.
EC 1.3	El usuario no introduce ni el usuario y ni la contraseña para acceder al sistema y selecciona la opción entrar.	N/A	Muestra un mensaje en dependencia del valor del campo que se haya introducido incorrectamente.	El usuario no introduce su usuario y contraseña, selecciona la opción aceptar.

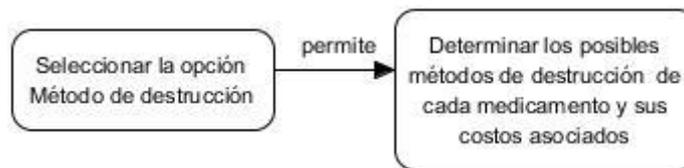


Ilustración 15: Gráfico correspondiente al caso de prueba del RF9 Determinar métodos de destrucción

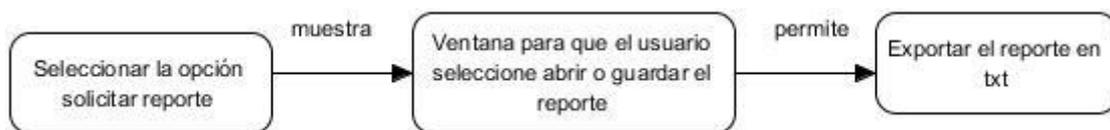


Ilustración 16: Gráfico correspondiente al caso de prueba del RF10 Generar reporte

3.4.3. Pruebas de despliegue

“La prueba de despliegue, en ocasiones llamada prueba de configuración, ejercita el software en cada entorno en el que debe operar. Además, examina todos los procedimientos de instalación y el software de instalación especializado (por ejemplo, “instaladores”) que usarán los clientes, así como toda la

documentación que se usará para introducir el software a los usuarios finales”.
 (Pressman, 2010)

Tabla 7: Pruebas de Despliegue

No	Hardware			
1	Procesador: Core 2 Duo	HDD: 200GB	RAM: 512MB	
2	Procesador: i3	HDD: 300GB	RAM: 2GB	
3	Procesador: i5	HDD: 500GB	RAM: 3GB	
4	Procesador: i7	HDD: 1T	RAM: 3GB	
Entorno a operar				
Navegador web	Sistema Operativo	Hardware	Resultado	Problemas
Mozilla Firefox	Ubuntu mate	1	Aceptado	No
		2,3,4	Aceptado	No
Google Chrome	Windows 7	1	Aceptado	No
		2,3,4	Aceptado	No
Opera	Windows 8	1	Aceptado	No
		2,3,4	Aceptado	No

3.4.4 Resultados de las pruebas

Tabla 8: Resultado de las pruebas de caja negra

Pruebas realizadas	Iteración	No conformidades
Autenticar usuario	Iteración 1	Desarrollada satisfactoriamente.
Editar usuario	Iteración 1	Desarrollada satisfactoriamente.
Buscar medicamento	Iteración 1	Cuando el usuario no introducía un valor en el campo de búsqueda y daba click en buscar, el sistema no mostraba el mensaje “Debe introducir un valor para la búsqueda”
	Iteración 2	Desarrollada satisfactoriamente.
Insertar medicamento	Iteración 1	Desarrollada satisfactoriamente.
Modificar medicamento	Iteración 1	Desarrollada satisfactoriamente.
Eliminar medicamento	Iteración 1	Desarrollada satisfactoriamente.
Seleccionar método de destrucción	Iteración 1	Desarrollada satisfactoriamente.
Determinar métodos de destrucción	Iteración 1	Cuando el usuario seleccionaba un medicamento y daba click en método de destrucción, el sistema no determinaba los métodos de destrucción asociados al medicamento seleccionado.
	Iteración 2	Cuando el usuario seleccionaba un medicamento y daba click en método de destrucción, el sistema no le permitía seleccionar uno de estos.
	Iteración 3	Desarrollada satisfactoriamente.
Generar reporte	Iteración 1	Cuando el usuario no seleccionaba ningún medicamento y daba click en

		solicitar reporte, el sistema no mostraba el mensaje “Debe seleccionar los medicamentos que desee aparezcan en el reporte”.
	Iteración 2	Desarrollada satisfactoriamente.
Insertar regla	Iteración 1	Desarrollada satisfactoriamente.
Modificar regla	Iteración 1	Desarrollada satisfactoriamente.
Eliminar regla	Iteración 1	Desarrollada satisfactoriamente.
Ayuda	Iteración 1	Desarrollada satisfactoriamente.

Al finalizar las pruebas se resolvió el 100% de las no conformidades detectadas, lo que define que la herramienta cumple con los requisitos definidos.

3.5. Conclusiones parciales

- La definición de un estándar de codificación asegura el entendimiento del código, a futuros desarrolladores de la herramienta.
- La identificación y solución de no conformidades confirmó la importancia de aplicar pruebas al software.
- La solución oportuna de las no conformidades identificadas, garantizó la calidad de la propuesta de solución.

Conclusiones generales

Una vez culminada la presente investigación y el desarrollo de la herramienta informática para la determinación de los métodos de destrucción, se puede arribar a las siguientes conclusiones:

- ✓ Se determinó que los sistemas analizados que tratan medicamentos no ofrecen vías, ni técnicas, que permitan solucionar el problema planteado, lo que justifica la necesidad de la investigación.
- ✓ El desarrollo de la herramienta informática DEMED, facilita la toma de decisiones en la selección del método de destrucción adecuado para un medicamento en particular.
- ✓ El desarrollo de la herramienta garantiza que se minimicen los errores al identificar los posibles métodos de destrucción asociado a un medicamento, y por ende disminuye el impacto negativo en el orden económico, social y medioambiental.
- ✓ El adecuado uso de la metodología AUP-UCI garantizó el cumplimiento del objetivo propuesto, con la calidad requerida.

Recomendaciones

Una vez concluido el desarrollo del sistema informático para la determinación de los métodos de destrucción para los medicamentos rechazados en EMCOMED, se recomienda realizar una segunda versión del sistema que permita:

- ✓ Insertar los medicamentos a través del reconocimiento del código de barra.
- ✓ Recomendar el método de destrucción a utilizar en un medicamento, a partir de un análisis de la información almacenada.
- ✓ Alertar a la especialista de calidad cuando existan medicamentos sin procesar.
- ✓ Extender la herramienta al idioma inglés, para que pueda ser utilizada por la comunidad internacional.

Referencias bibliográficas

Argitalpen, Eusko Jaurlaritzaren Zerbitzu Nagusia. 2016. *Farmacontaminación. Impacto ambiental de los medicamentos.* s.l. : Servicio Central de Publicaciones del Gobierno Vasco, 2016. Vol. 24. No10.

Aristegui O, José Luis. 2010. *Test cases in software test.* 2010.

Cervantes, Dr.Humberto. 2010. *Arquitectura de Software. Arquitectura.* [En línea] abril de 2010. [Citado el: 2 de marzo de 2017.] <http://sg.com.mx/content/view/922>.

Galindo, Raúl Miguel Romero. 2012. *Análisis, diseño e implementación de un sistema de información aplicado a la gestión educativa en centros de educación especial.* Perú : s.n., 2012.

González, Carlos Caballero. 2007. *Desarrollo de software con calidad para una empresa.* 2007.

Guijarro, Álvaro Primo. 2012. *Protocolo HTTP.* 2012.

Jovanovi´c, Irena. 2008. *Software testing methods and techniques.* 2008.

Larman, Craig. 1999. *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* Primera edición. México : s.n., 1999. ISBN 970-1 7-0261-1.

López, Tania. 2016. *La gestión integrada de los residuales sólidos en la Plataforma Logística de las Industrias Biotecnológica y Farmacéutica.* Ciudad de La Habana : s.n., 2016.

Mariños Urquiaga, Jean Carlos. 2016. ¿Por qué usar Django ? [En línea] 2016. [Citado el: 22 de febrero de 2017.] <https://devcode.la/blog/por-que-usar-django/>.

Merseguer, José. 2010. *Diagramas de casos de uso.* Zaragoza : s.n., 2010.

Mesa, Mirna. 2011. *Herramientas Case.* [En línea] 2 de abril de 2011. [Citado el: 22 de febrero de 2017.] <http://fds-herramientascase.blogspot.com/>.

Miyar, J.M. 2009. *Resolución 136.* CITMA Cuba : s.n., 2009.

Moine, Juan Miguel. 2013. *Metodologías para el descubrimiento de conocimiento en bases de datos: un estudio comparativo.* 2013.

Pressman, Roger S. 2010. *ingeniería de Software. Un enfoque práctico.* 7 Edición. Madrid : McGraw-Hill, 2010. ISBN 978-607-15-0314-5.

Pressman, Roger S. 2005. *Ingeniería del software. Un enfoque práctico.* Sexta edition. 2005.

Ruiz Tenorio, Roberto. 2010. *Las pruebas de software y su importancia en las organizaciones.* Universidad Veracruzana : s.n., 2010.

Rumbaugh, James, Jacobson, Ivar y Booch, Grady. *El Lenguaje Unificado de Modelado.*

Sancho Caperrini, Fernando. 2015. Sistemas basados en reglas. [En línea] 25 de julio de 2015. [Citado el: 21 de marzo de 2017.] <http://www.cs.us.es/~fsancho/?e=103>.

Schmuller, Joseph. *Aprendiendo UML en 24 horas.* México : Editorial División Computación.

Software, Ingeniería del. 2005. *Introducción al Modelo Conceptual.* 2005.

SOLUS, S.A. 2000-2016. Spark Systems. [En línea] 2000-2016. [Citado el: 3 de marzo de 2017.] http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html.

Sommerville, Ian. 2005. *Ingeniería del Software.* Séptima edición. Madrid : s.n., 2005. 84-7829-074-5.

Tilló Barrufet, Teresa. *Los medicamentos y su influencia en el medio ambiente.* Universidad de Barcelona : s.n.

Velazco, D., Cuadrado M., Sampedro C. 2014. SQLite Data Base. [En línea] 2014. [Citado el: 31 de febrero de 2017.] <http://sqlitedatabasegrupo.blogspot.com/p/caracteristicas-de-sqlite.html>.

Wesley, Addison. 2006. *Patrones de Diseño*. 2006.

Bibliografía

Alarcón de Quesada, Ricardo. 1996. *Ley 81 del Medio Ambiente CITMA.* República de Cuba : s.n., 1996.

CIM, Centre d'Informació de Medicaments de Catalunya. 2007. ¿Qué es un medicamento? . [En línea] 2007. [Citado el: 3 de marzo de 2017.] <http://www.cedimcat.info/index.php?lang=es&Itemid=474>.

Dario, Rubén. 2014. Metodologías de Desarrollo Ágiles Vs. Metodologías Tradicionales. [En línea] 20 de marzo de 2014. [Citado el: 22 de febrero de 2017.] <http://rdsoporteymantenimientodepc.blogspot.com/2014/03/metodologias-de-desarrollo-agiles-vs.html>.

Enviromental management, s. 2004. *ISO 14000. Organización Internacional de Normalización.* Suiza : s.n., 2004.

Flores, Ervin y Cordero, Jorge Luis. *Metodologías Ágiles, Proceso Unificado Ágil (AUP).* Bolivia : s.n.

García, M. B., R. 1997. *El empleo del razonamiento basado en casos en el desarrollo de Sistemas basados en el conocimiento para el diagnóstico. Trabajo de Tesis en opción al grado científico de Doctor en Ciencias Técnicas.* UCLV : s.n., 1997.

Garre M., Cuadrado J.J., Sicilia,M.A., Charro M, Rodríguez D. 2005. *Segmented Parametric Software Estimation Models: Using the EM algorithm with the ISBSG 8 database,Information Technology Interfaces.* Croacia : s.n., 2005.

Hernández Leon, Rolando Alfredo, Cuello González, Saida. 2011. *Proceso de investigacion cientifica.* La habana : Editorial Universitaria, 2011. 978-959-16-1307-3.

Jacobson, Ivar , Booch , Grady y Rumbaugh, James. 2000. *El Proceso Unificado De Desarrollo de Software.* Madrid : Addison Wesley, 2000. ISBN 84-7829-036-2.

Jorin, Ing. Michael González. 2007. *Tesis de maestría. Proceso de pruebas para la liberación de productos software.* Ciudad de La Habana : s.n., 2007.

Martínez, Raúl Pérez. *Manual de Metodología de la Investigación Científica.*

Martínez-Trinidad, J. F. a. G. S.-D. 2001. *LC: A Conceptual Clustering Algorithm, in Proceedings of the Second International Workshop on Machine Learning and Data Mining in Pattern Recognition 2001.* s.l. : Springer-Verlag, 2001. págs. 117 -127.

Oré, Alexander. Calidad y Software. [En línea] [Citado el: 20 de febrero de 2017.] http://www.calidadyssoftware.com/testing/pruebas_unitarias1.php.

Ramos Alvariño, C. 2006. *Los residuos en la industria farmacéutica.* s.l. : CENIC, Ciencias Biológicas, 37, 2006.

Reyes, Yunia. 2014. *Modelo para la adaptación de las soluciones en un Sistema Basado en Casos utilizando el agrupamiento conceptual. Trabajo de Tesis en opción de Título de Máster en Informática Aplicada.* UCI : s.n., 2014.

Rodriguez-Piñero, Piedad Tolmos. Introducción a los algoritmos genéticos y sus aplicaciones. [En línea] [Citado el: 21 de marzo de 2017.] <https://www.uv.es/asepuma/X/J24C.pdf>.

Ruiz-Shulcloper, J. 2013. *RECONOCIMIENTO LÓGICO COMBINATORIO DE PATRONES: TEORÍA Y APLICACIONES. TESIS EN OPCIÓN AL GRADO CIENTÍFICO DE DOCTOR EN CIENCIAS.* Centro de Aplicaciones de Tecnologías de Avanzada : s.n., 2013.

Saladrigas, María Verónica. 2004. El sistema de clasificación ATC de sustancias farmacéuticas para uso humano. [En línea] 2004. [Citado el: 25 de febrero de 2017.] http://www.medtrad.org/panacea/IndiceGeneral/n15_tribuna-Saladrigas.pdf.

Shulcloper, J. R. A., E. y Lazo, M. 1995. *Introducción al Reconocimiento de Patrones. Grupo de Reconocimiento de Patrones Cuba-México. Centro de*

Investigación y de Estudios Avanzados del I.P.N. Dpto de Ingeniería Eléctrica.
México : Serie Verde, 1995. No. 51.

Rodríguez, Tamara. 2015. *Metodología de desarrollo para la Actividad productiva de la UCI.* La Habana: s.n., 2015.

Tedeschi, Nicolás. 2010. ¿Qué es un patrón de diseño? [En línea] 2010. [Citado el: 27 de febrero de 2017.] <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.

Vallejos, Sofia J. 2006. *Diseño y Administración de Datos.* Argentina : s.n., 2006.

Velasco, Ximena. *Métodos de Investigación Científica.*

Anexos

Anexo I: Descripción de los casos de uso del sistema.

Tabla 9: Descripción CU1. Gestionar usuario.

Objetivo	Gestionar toda la información del usuario	
Actores	Especialista de Calidad	
Resumen	El caso de uso comienza cuando la especialista de calidad desea autenticarse o editar sus datos.	
Complejidad	Baja	
Prioridad	Alta	
Precondiciones		
Post-condiciones		
Flujo de eventos		
Flujo básico		
No.	Actor	Sistema
1.	Selecciona una de las opciones: a)Autenticarse para acceder al sistema b)Editar sus datos	
2.		Ejecuta una de las siguientes acciones según la opción seleccionada: a)Si decide autenticarse, ir al escenario 1: "Autenticar usuario" b)Si decide editar sus datos, ir al escenario 2: "Editar usuario"
Escenario 1: "Autenticar usuario"		
No.	Actor	Sistema

3.	Abre el sistema para entrar en él.	
4.		Brinda los campos usuario y contraseña que deben ser llenados obligatoriamente.
5.	Llena los campos.	
6.	Selecciona la opción Entrar.	
7.		Comprueba que los campos fueron llenados correctamente.
8.		Accede a las funcionalidades del sistema.

Flujos alternos

No.	Actor	Sistema
5.1	Llena los campos incorrectamente.	
5.2		Muestra un mensaje en dependencia del valor del campo que se haya introducido incorrectamente.

Escenario 2: "Editar usuario"

No.	Actor	Sistema
3.	Selecciona la opción Editar usuario.	
4.		Brinda los campos del usuario para que puedan ser modificados.
5.	Cambia los campos deseados.	
6.	Selecciona la opción Guardar.	

7.		Comprueba que los campos fueron llenados correctamente y guarda los cambios.
Flujos alternos		
No.	Actor	Sistema
5.1	Cambia el valor de los campos incorrectamente.	
5.2		Muestra un mensaje en dependencia del valor del campo que se haya introducido incorrectamente.

Tabla 10: Descripción CU2. Gestionar Medicamento

Objetivo	Gestionar toda la información del medicamento	
Actores	Especialista de Calidad	
Resumen	El caso de uso comienza cuando la especialista de calidad desea buscar, insertar, modificar o eliminar un medicamento.	
Complejidad	Baja	
Prioridad	Alta	
Precondiciones		
Post-condiciones		
Flujo de eventos		
Flujo básico		
No.	Actor	Sistema
1.	Selecciona una de las opciones: a) Buscar medicamento b) Insertar medicamento	

	<p>c) Modificar medicamento</p> <p>d) Eliminar medicamento</p> <p>e) Seleccionar método de destrucción</p>	
2.		<p>Ejecuta una de las siguientes acciones según la opción seleccionada:</p> <p>a) Si decide buscar un medicamento, ir al escenario 1: “Buscar medicamento”</p> <p>b) Si decide insertar un medicamento, ir al escenario 2: “Insertar medicamento”</p> <p>c) Si decide modificar un medicamento, ir al escenario 3: “Modificar medicamento”</p> <p>d) Si decide eliminar un medicamento, ir al escenario 4: “Eliminar medicamento”</p> <p>d) Si decide seleccionar un método de destrucción, ir al escenario 4: “Seleccionar método de destrucción”</p>
Escenario 1: “Buscar medicamento”		
No.	Actor	Sistema
3.	Coloca algún valor en el campo buscar medicamento.	
4.	Selecciona la opción buscar.	

5.		Comprueba que los campos fueron llenados correctamente.
6.		Realiza la búsqueda por el valor proporcionado.
7.		Muestra los medicamentos encontrados al realizar la búsqueda.

Flujos alternos

No.	Actor	Sistema
3.1	Selecciona la opción búsqueda avanzada.	
3.2		Brinda los campos por los cuales se puede realizar la búsqueda del medicamento.
3.3	Llena los campos por los cuales desea realizar la búsqueda.	
3.4		Comprueba que los campos fueron llenados correctamente.
3.5		Realiza la búsqueda por el valor proporcionado.
3.6		Muestra el resultado de la búsqueda.

Escenario 2: "Insertar medicamento "

No.	Actor	Sistema
3.	Selecciona la opción Insertar medicamento.	
4.		Brinda los campos que deben ser llenados del medicamento.

5.	Llena los campos.	
6.	Selecciona la opción Insertar.	
7.		Comprueba que los campos fueron llenados correctamente.
8.		Inserta el medicamento y lo clasifica según su clase (1,2 o 3), guardándolo en la base de datos.

Flujos alternos

No.	Actor	Sistema
5.1	Llena los campos incorrectamente.	
5.2		Muestra un mensaje en dependencia del valor del campo que se haya introducido incorrectamente.

Escenario 3: "Modificar medicamento"

No.	Actor	Sistema
3.	Selecciona la opción Modificar medicamento.	
4.		Brinda los valores de los campos del medicamento para que pueda ser modificado.
5.	Modifica los campos deseados.	
6.	Selecciona la opción Modificar.	
7.		Comprueba que los campos fueron llenados correctamente.
8.		Modifica el medicamento en la base de datos.

Flujos alternos		
No.	Actor	Sistema
5.1	Modifica el valor de los campos incorrectamente.	
5.2		Muestra un mensaje en dependencia del valor del campo que se haya introducido incorrectamente.
Escenario 4: "Eliminar medicamento "		
No.	Actor	Sistema
3.	Selecciona la opción Eliminar medicamento.	
4.		Lanza un mensaje de confirmación para que el usuario confirme su acción.
5.	Selecciona la opción Sí.	
6.		Elimina el medicamento del sistema.
Escenario 5: "Seleccionar método de destrucción"		
No.	Actor	Sistema
3.	Selecciona la opción Seleccionar método de destrucción.	
4.		Brinda los métodos por los cuales el usuario puede seleccionar.
5.	Selecciona un método y selecciona el botón aceptar.	

6.		Selecciona el método elegido por el usuario y lo guarda en la base de datos.
----	--	--

Tabla 11: Descripción CU3. Gestionar regla

Objetivo	Gestionar las reglas que definen los métodos de destrucción.	
Actores	Especialista de Calidad	
Resumen	El caso de uso comienza cuando la especialista de calidad desea insertar, modificar o eliminar una regla.	
Complejidad	Baja	
Prioridad	Alta	
Precondiciones		
Post-condiciones		
Flujo de eventos		
Flujo básico		
No.	Actor	Sistema
1.	Selecciona una de las opciones: a) Insertar medicamento b) Modificar medicamento c) Eliminar medicamento	
2.		Ejecuta una de las siguientes acciones según la opción seleccionada: a) Si decide insertar una regla, ir al escenario 1: "Insertar regla" b) Si decide modificar una regla, ir al escenario 2: "Modificar una regla"

		c)Si decide eliminar una regla, ir al escenario 3: “Eliminar regla”
Escenario 1: “Insertar regla”		
No.	Actor	Sistema
3.	Selecciona la opción Insertar regla.	
4.		Brinda los campos que deben ser llenados de la regla.
5.	Llena los campos.	
6.	Selecciona la opción Insertar.	
7.		Comprueba que los campos fueron llenados correctamente.
8.		Inserta la regla en la base de datos.
Flujos alternos		
No.	Actor	Sistema
5.1	Llena los campos incorrectamente.	
5.2		Muestra un mensaje en dependencia del valor del campo que se haya introducido incorrectamente.
Escenario 2: “Modificar regla”		
No.	Actor	Sistema
3.	Selecciona la opción Modificar regla.	
4.		Brinda los valores de los campos de la regla para que pueda ser modificada.
5.	Modifica los campos deseados.	

6.	Selecciona la opción Modificar.	
7.		Comprueba que los campos fueron llenados correctamente.
8.		Modifica la regla en la base de datos.

Flujos alternos

No.	Actor	Sistema
5.1	Modifica el valor de los campos incorrectamente.	
5.2		Muestra un mensaje en dependencia del valor del campo que se haya introducido incorrectamente.

Escenario 3: "Eliminar regla"

No.	Actor	Sistema
3.	Selecciona la opción Eliminar regla.	
4.		Lanza un mensaje de confirmación para que el usuario confirme su acción.
5.	Selecciona la opción Sí.	
6.		Elimina la regla del sistema.

Anexo II: Diagramas de clases de diseño

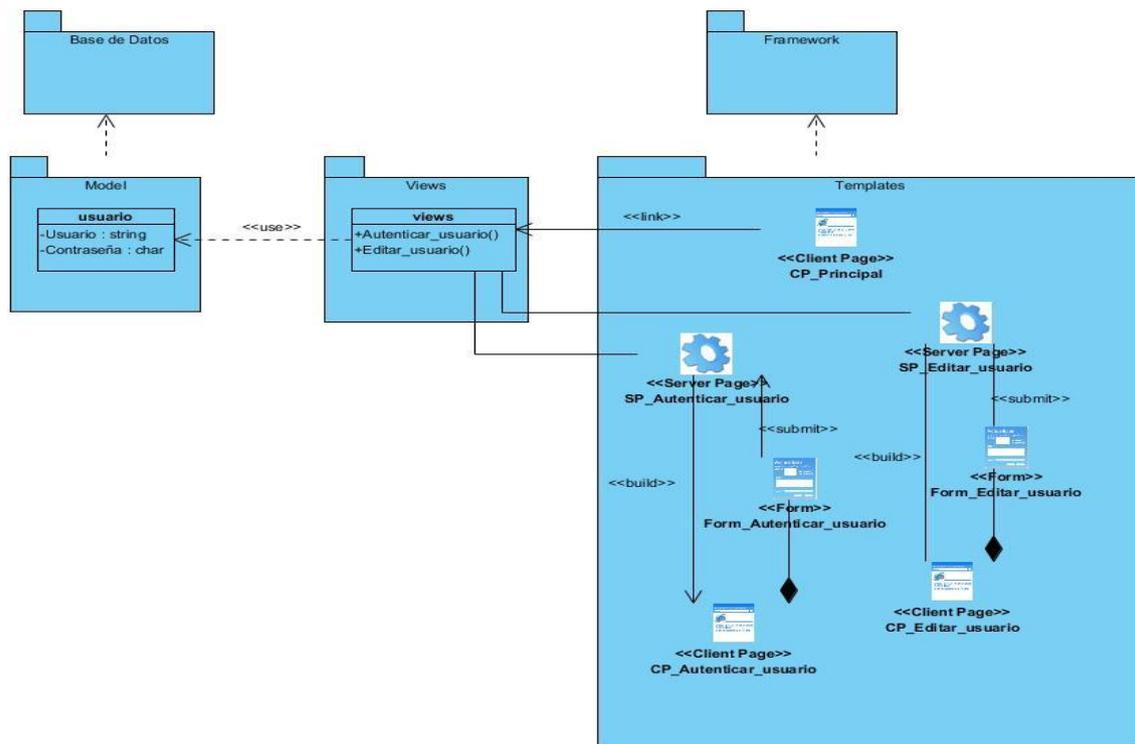


Ilustración 17: Diagrama de clases de diseño RF: Autenticar y Editar usuario

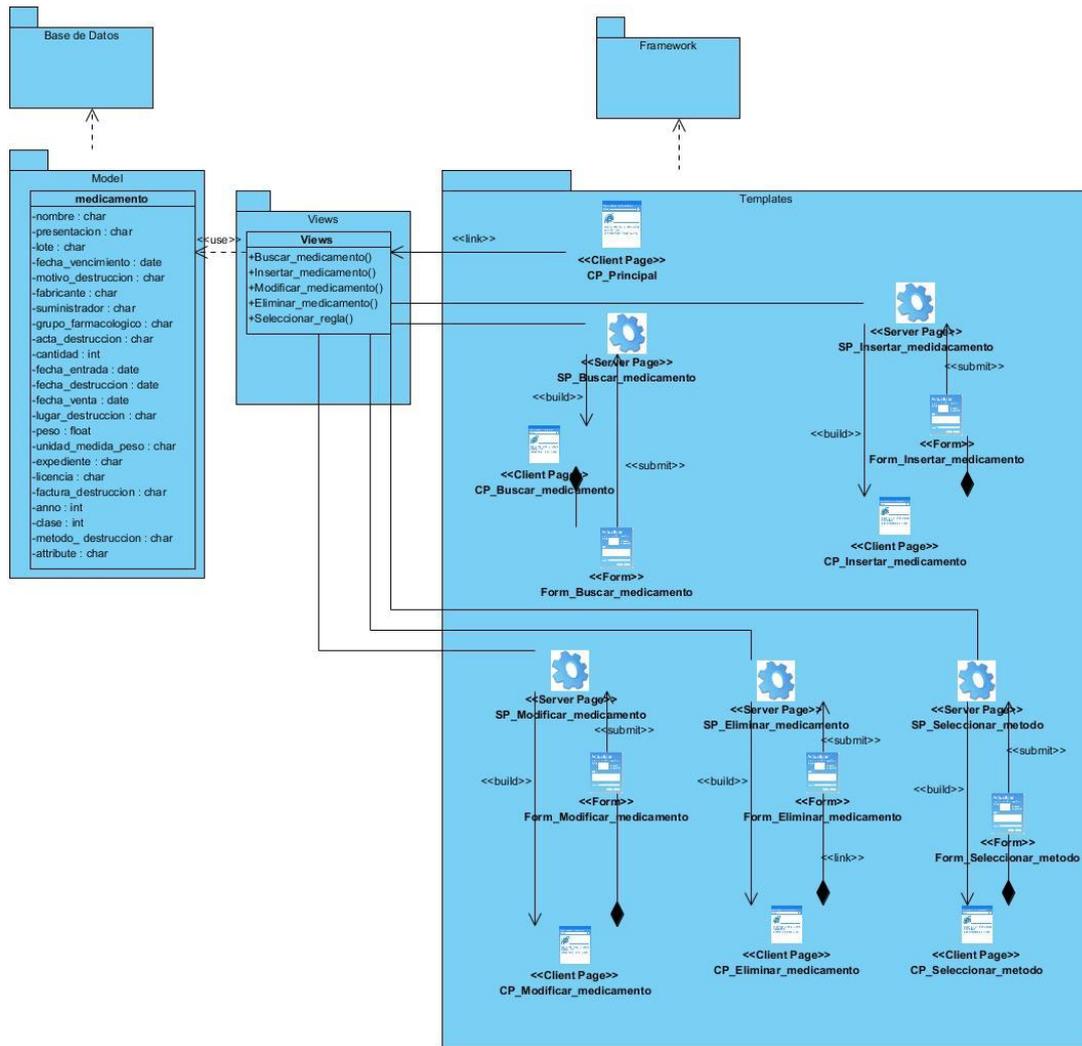


Ilustración 18: Diagrama de clases de diseño RF: Buscar, Insertar, Modificar, Eliminar medicamento y Seleccionar método de destrucción

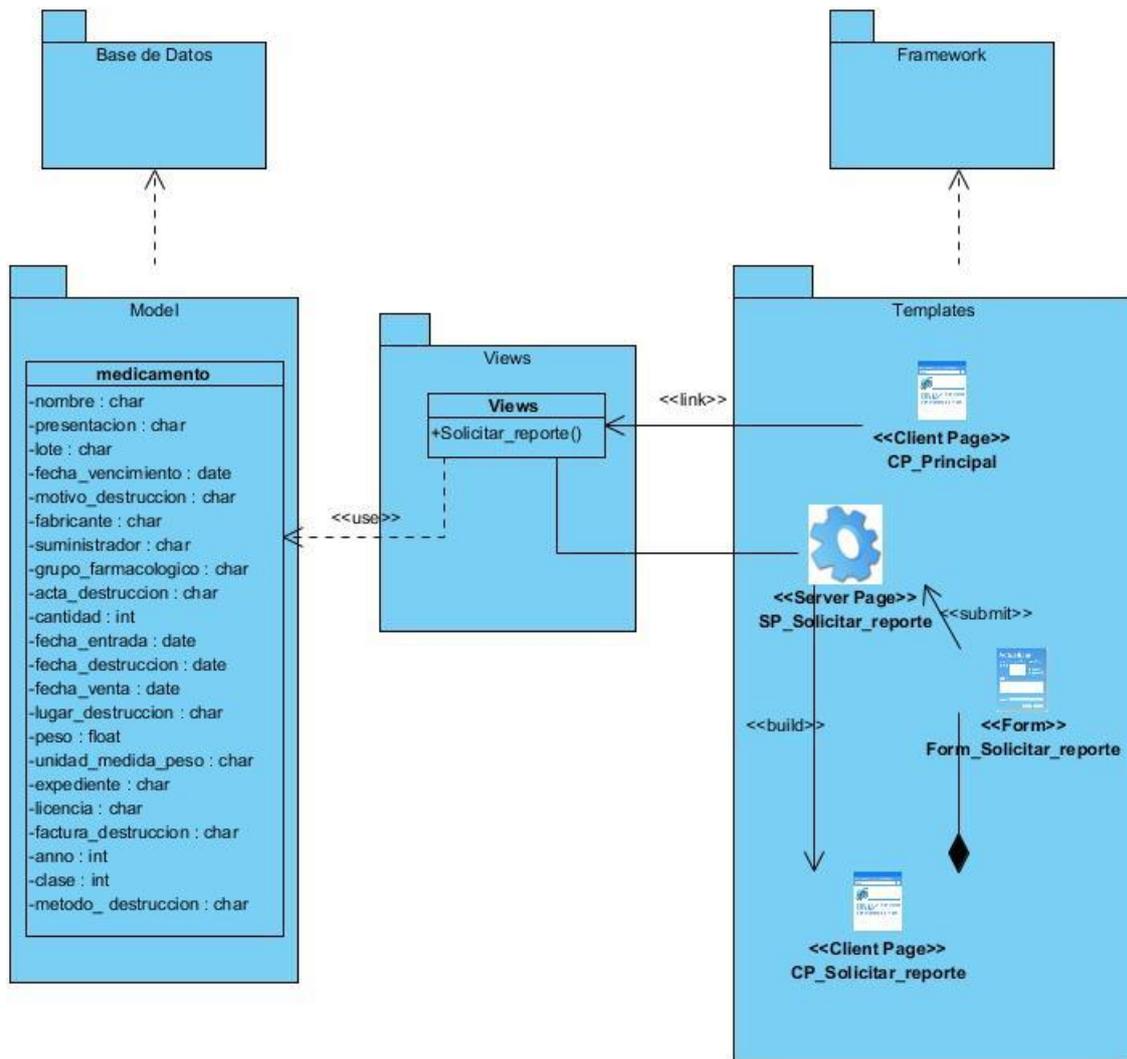


Ilustración 19: Diagrama de clases de diseño RF: Gestionar reporte

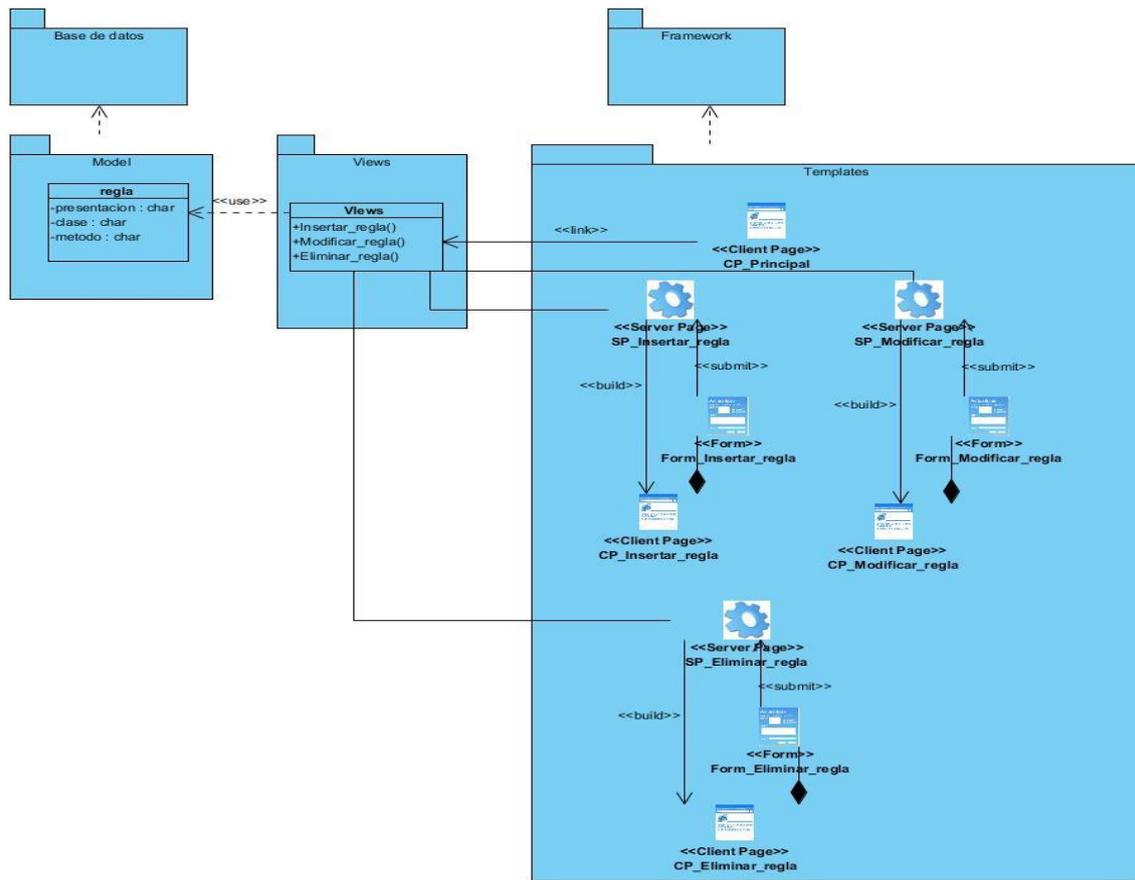


Ilustración 20: Diagrama de clases de diseño RF: Insertar, Modificar y Eliminar regla

Anexo III: Ejemplo de casos de prueba

Tabla 12: Caso de prueba de partición equivalente del RF11 Insertar regla

Escenario	Descripción	Datos	Respuesta del sistema	Flujo central
EC 1.1	El usuario selecciona la opción insertar regla con valores correctos.	V	Inserta en la BD la regla introducida por el usuario.	El usuario inserta los valores de la regla, selecciona la opción insertar.
	inserta el valor de la regla y selecciona la opción insertar.	I		
		polvo		

EC 1.2 Seleccionar la opción insertar regla con valores incorrectos.	El usuario inserta el valor de la regla y selecciona la opción insertar.	I	Muestra un mensaje en dependencia del valor del campo que se haya introducido incorrectamente.	El usuario inserta los valores de la regla, selecciona la opción insertar.
		VI		
		12-1-2017		
EC 1.3 Seleccionar la opción insertar regla y deja los valores de los campos vacíos.	El usuario deja el valor de los campos de la regla a insertar en vacíos y selecciona la opción insertar.	N/A	Muestra un mensaje en dependencia del valor del campo que se haya introducido incorrectamente.	El usuario deja el valor de los campos de la regla a insertar en vacíos y selecciona la opción insertar.

Tabla 13: Caso de prueba de partición equivalente del RF12 Modificar regla

Escenario	Descripción	Datos	Respuesta del sistema	Flujo central
EC 1.1 Seleccionar la opción modificar regla con valores correctos.	El usuario modifica el valor de los datos de la regla y selecciona la opción guardar.	V	Guarda los cambios realizados por el usuario.	El usuario modifica el valor de los datos de la regla, selecciona la opción guardar.
		III		
		tableta		
EC 1.2 Seleccionar	El usuario modifica el	I	Muestra un mensaje en	El usuario modifica el

la opción modificar regla con valores incorrectos.	valor de los datos de la regla y selecciona la opción guardar.	IV 23-5-2017	dependencia del valor del campo que se haya introducido incorrectament e.	valor de los datos de la regla, selecciona la opción guardar.
EC 1.3 Seleccionar la opción modificar regla y deja los valores de los campos vacíos.	El usuario deja el valor de los campos de la regla en vacíos y selecciona la opción guardar.	N/A	Muestra un mensaje en dependencia del valor del campo que se haya introducido incorrectament e.	El usuario modifica el valor de los campos de la regla por vacíos, selecciona la opción guardar.