



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 7

Trabajo de Diploma para optar por el Título de
Ingeniero en Ciencias Informáticas

Componente para gestionar la trazabilidad de los medicamentos en el
Sistema de Información Hospitalaria xavia HIS utilizando la tecnología de
Identificación por Radiofrecuencia

Autores: Dunia Rodríguez Jiménez

Celio Manuel Frías López

Tutores: Ing. Diana Rosa Alfonso Espinosa

Ing. Alfredo Rodríguez Ruiz.

Co-tutor: Reinier Milian Pérez

La Habana, junio del 2013

Año 55 del Triunfo de la Revolución.

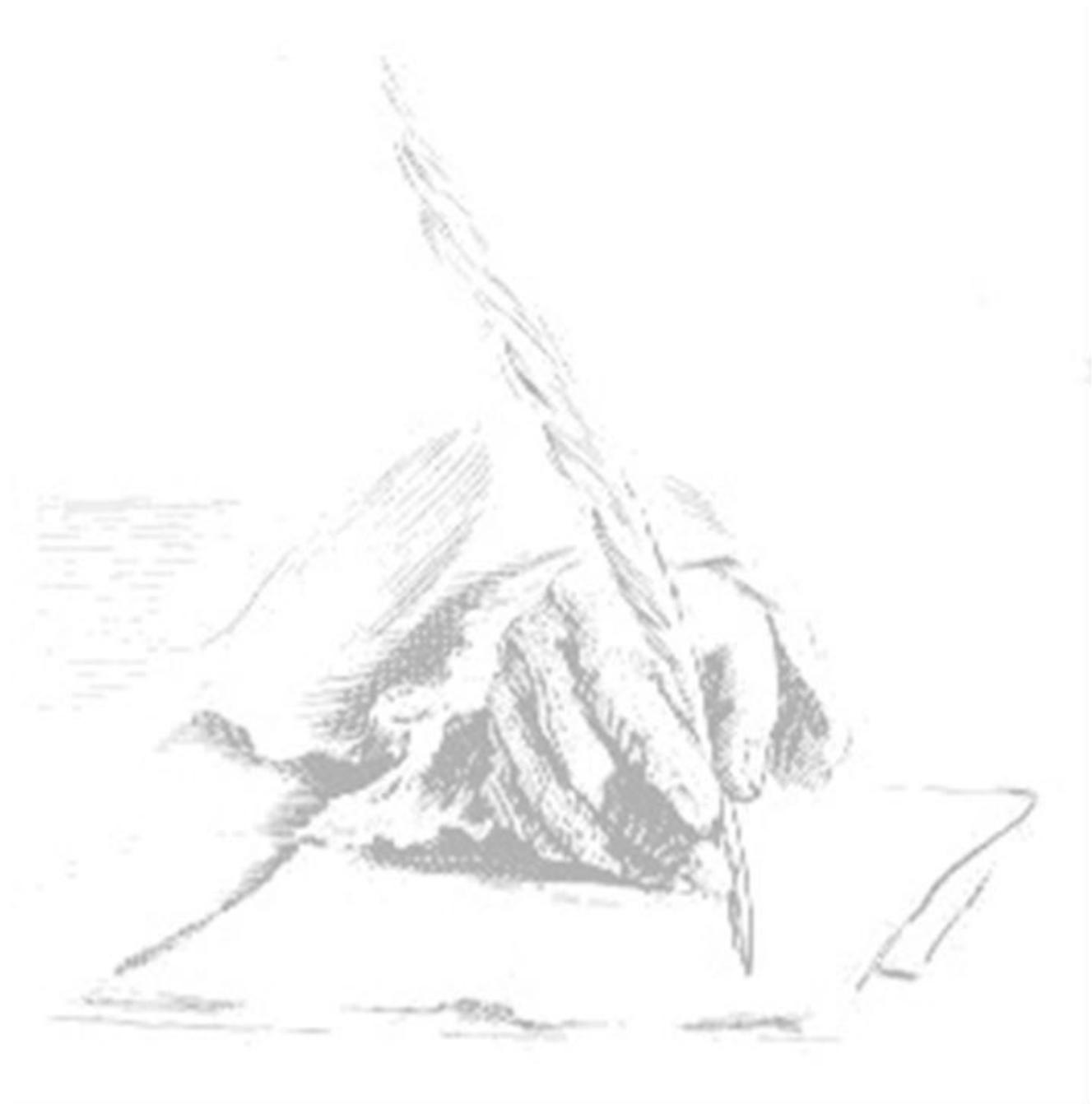
Datos De Contacto

Ing. Diana Rosa Alfonso Espinosa: Graduada en el año 2008 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Posee categoría docente de Instructor. Ha impartido las asignaturas de Pruebas y evaluación de software, Administración de calidad, CMMI, Aplicaciones informáticas en el sector de la salud e Introducción a las pruebas de software. Se desempeña como Analista del departamento Atención Primaria a la Salud en el Centro de Informática Médica (CESIM). Correo: dralfonso@uci.cu.

Ing. Alfredo Rodríguez Ruiz: Graduado en el año 2008 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Posee categoría docente de Instructor. Ha impartido las asignaturas de Base de Datos, Programación web, Programación 2 y 3, y Gráficos por computadora. Se desempeña como Jefe del departamento Atención Primaria a la Salud en el Centro de Informática Médica (CESIM). Correo: arruiz@uci.cu.

Ing. Reinier Milian Pérez: Graduado en el año 2012 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Se desempeña como desarrollador en el departamento de Atención Primaria a la Salud en el Centro de Informática Médica (CESIM). Correo: rmilianp@uci.cu.

Agradecimientos



Dedicataria



Resumen

Los sistemas de gestión hospitalaria proporcionan nuevas oportunidades de mejora en los procesos asistenciales mediante la integración de tecnologías de información y médicas, buscan elevar la calidad de los servicios prestados y de la atención al paciente. El Sistema de Información Hospitalaria (xavia HIS), desarrollado en el Centro de Informática Médica no es la excepción, está diseñado para tener accesible toda la información generada, automatizando y simplificando los principales procesos clínicos y financieros de los hospitales. El avance tecnológico a nivel mundial ha ido en constante aumento, han surgido nuevas tecnologías que pueden aumentar aún más el alcance de este sistema y solucionar algunas desventajas que presenta en la actualidad en cuanto a la trazabilidad de los medicamentos. La presente investigación tiene como objetivo desarrollar un componente, que utilice la tecnología de Identificación por Radiofrecuencia, para la gestión de la trazabilidad de los medicamentos en el Sistema de Información Hospitalaria (xavia HIS).

Para el desarrollo del componente se emplea como servidor de aplicación JBoss y Eclipse como plataforma de desarrollo, Java como lenguaje de programación, PostgreSQL como Sistema Gestor de Bases de Datos, Hibernate como ORM (*Object Relational Mapping*) para la persistencia de los datos y Seam como el marco de trabajo integrador.

Como resultado se espera que la aplicación de este componente simplifique la gestión de la información relacionada con los medicamentos, y logre mantener una correcta trazabilidad de todo el recorrido de los mismos dentro del hospital, asegurando su autenticidad y evitando que puedan ser robados o trasladados de lugar sin permiso.

Palabras Clave: gestión de la trazabilidad, identificación, radiofrecuencia.

Tabla De Contenidos

Introducción 1

Capítulo 1. Fundamentación teórico metodológica de la trazabilidad de los medicamentos5

 1.1 Conceptos de interés 5

 1.1.1 Trazabilidad 5

 1.1.2 Tecnología de Identificación por Radiofrecuencia 7

 1.1.3 Protocolos de comunicación en la tecnología de Identificación por Radiofrecuencia 10

 1.1.4 Beneficios generales de la identificación por radiofrecuencia 13

 1.2 Sistemas para la trazabilidad basados en Radiofrecuencia 14

 1.3 Herramientas y tecnologías utilizadas 18

Capítulo 2. Características y diseño del componente propuesto.....28

 2.1 Modelado del negocio 28

 2.2 Características del componente propuesto 31

 2.3 Especificación de los requerimientos del software 32

 2.4 Modelo de Diseño 37

Capítulo 3. Implementación del componente propuesto 53

 3.1 Modelo de datos 53

 3.2 Diagrama de despliegue 55

 3.3 Gestión de la trazabilidad en el componente 57

 3.4 Análisis de equipos y costos 57

 3.5 Integración del componente con el xavia HIS 59

 3.6 Estándares de codificación y tratamiento de excepciones y errores 60

Beneficios esperados 64

Conclusiones 65

Recomendaciones 66

Trabajos citados 67

Bibliografía 73

Glosario de Términos 77

Anexos 79

Introducción

Las tecnologías a nivel mundial han tenido un gran auge en el último siglo llegando a convertirse en parte imprescindible de la existencia del ser humano. El desarrollo tecnológico de un país se basa en iniciativas propias y sus resultados se observan en la simplicidad al realizar tareas relacionadas con la economía, la educación y otros sectores de la sociedad (1).

En la actualidad todo este avance es aprovechado principalmente, y con grandes resultados en el campo de la salud. Es común la existencia de un médico trabajando en el ciberespacio, buscando alguna evidencia para la obtención de un diagnóstico o un tratamiento. Se ha establecido una mejor comunicación entre la comunidad médica, ya no hay fronteras físicas que les impidan compartir o adquirir conocimientos (2), debido a que los hospitales cada día poseen equipos modernos que facilitan el trabajo de los mismos. Son innumerables los ejemplos de estas nuevas tecnologías que se aplican con éxito en este importantísimo sector, entre las que se destaca la Identificación por Radiofrecuencia (RFID por sus siglas en inglés).

A nivel mundial la tecnología RFID es utilizada principalmente en la cadena de suministros de los productos farmacéuticos. Esto se debe principalmente a la urgente necesidad que tiene dicho sector de poder verificar en tiempo real la autenticidad de sus productos y de realizar un seguimiento de la trazabilidad de los mismos para evitar que puedan ser robados o falsificados. Es por ello que la gestión de la trazabilidad de los productos farmacéuticos y principalmente de los medicamentos se ha convertido en una problemática actual, que requiere una solución urgente.

En la mayoría de los países están presentes estos problemas con los medicamentos, en unos con cifras más alarmantes que en otros, pues constituye una situación que genera afectación de manera global. Por todas las ventajas que ofrece sobre las demás tecnologías aplicadas en la gestión de la trazabilidad de los medicamentos son cada día más las compañías farmacéuticas y de otros sectores involucrados en la gestión hospitalaria que se acogen a RFID (3).

Los Sistemas de Información Hospitalaria (HIS, por sus siglas en inglés) son aplicaciones que buscan elevar la calidad de la atención del paciente y de los servicios brindados, aplicar la información obtenida a

las áreas de la investigación, la clínica, la docencia y la administración, abatir costos y elevar la productividad. Son sistemas orientados a satisfacer las necesidades de generación de información, para almacenar, procesar y reinterpretar datos médico-administrativos de la cualquier institución hospitalaria. Permitiendo la optimización de los recursos humanos y materiales (4).

El Centro de Informática Médica (CESIM) perteneciente a la Universidad de las Ciencias Informáticas (UCI) desarrolla el Sistema de Información Hospitalaria xavia HIS con el objetivo de mejorar la calidad en la atención del paciente. El mismo se encarga de todos los aspectos clínicos y financieros en un hospital, está compuesto por varios módulos que se corresponden con las distintas áreas de la institución y abarcan dos áreas o vertientes fundamentales: los procesos asistenciales y los procesos administrativos. Entre los módulos pertenecientes al área asistencial se mencionan: Consulta externa, Admisión, Citas, Hospitalización, entre otros. Mientras que en los administrativos se encuentran: Estadísticas, Almacén, Epidemiología, Facturación, Configuración y Farmacia.

La aplicación informática xavia HIS aún tiene debilidades en el proceso de gestión de los medicamentos que le restan competitividad como se mencionan a continuación:

- Carece de mecanismos que le permitan gestionar la trazabilidad de los medicamentos dentro del hospital, hasta su consumo por los pacientes.
- Carece de mecanismos que le permitan garantizar la autenticidad de los medicamentos y evitar su falsificación.
- Carece de una herramienta que automatice procesos importantes y que pueden ser objeto de errores humanos, como son la verificación y control de las fechas de vencimiento de los productos.
- El registro de la información sobre los medicamentos a la hora de recepcionarlos se hace tediosa y carece de mecanismos que le posibiliten agilizarla de manera segura.

Para fortalecer este proyecto y lograr que iguale o supere a sistemas existentes en el mercado internacional con funcionalidades similares, se hace necesario que incluya la tecnología más avanzada en la solución de estos problemas. Por este motivo se plantea el siguiente **problema a resolver**: ¿Cómo integrar la tecnología de Identificación por Radiofrecuencia al proceso de trazabilidad de los

medicamentos en el Sistema de Información Hospitalaria xavia HIS para elevar la seguridad y el control de los medicamentos?

Se define como **objeto de estudio**: el proceso de trazabilidad de los medicamentos en los Sistema de Información Hospitalaria.

Específicamente el **campo de acción** la gestión de la trazabilidad de los medicamentos en el Sistema de Información Hospitalaria xavia HIS.

Para darle respuesta al problema planteado se establece como **objetivo general**: desarrollar un componente que utilice la tecnología de Identificación por Radiofrecuencia para el proceso de trazabilidad de los medicamentos en el Sistema de Información Hospitalaria xavia HIS que contribuya a elevar su seguridad y el control.

Por esta razón se plantea la siguiente **idea a defender**: si para el proceso de trazabilidad de los medicamentos en el Sistema de Información Hospitalaria xavia HIS se desarrolla un componente que utilice la tecnología de Identificación por Radiofrecuencia, se contribuirá a elevar su seguridad y control.

Para darle cumplimiento al objetivo propuesto se definen las siguientes **tareas de investigación**:

1. Establecimiento de los fundamentos teóricos de la tecnología de Identificación por Radiofrecuencia y sus componentes.
2. Caracterización de los diferentes sistemas que utilizan la tecnología de Identificación por Radiofrecuencia para el control de los productos farmacéuticos.
3. Desarrollo del componente aplicando las pautas de diseño establecidas por el CESIM.
4. Integración del componente obtenido al Sistema de Información Hospitalaria xavia HIS.

Para la realización de la investigación según las tareas definidas se utilizan **métodos científicos** y **técnicas de investigación**, entre los que se encuentran:

Histórico-Lógico: es un método teórico que se utilizó en la búsqueda de los antecedentes del objeto de la investigación. Permitiendo crear una línea evolutiva para conocer su desarrollo y las principales etapas de su desenvolvimiento, así como las soluciones existentes.

Modelación: este método teórico se utilizó para la representación de los diferentes modelos del negocio y el sistema.

La entrevista: se le realizaron entrevistas no estructuradas con preguntas abiertas a los trabajadores del departamento de Sistema de Información Hospitalaria del CESIM, para obtener la información del negocio que permitiera avanzar en la investigación. Asimismo se entrevistaron diferentes directivos y trabajadores de las áreas de farmacia y almacén de medicamentos del hospital Ernesto Guevara, de la UCI, con el objeto de conocer como gestionaban su negocio.

La observación: con el objetivo de tener un registro visual de todo lo que tiene que ver con el problema planteado y que pueda ser de utilidad en la realización de la investigación.

El trabajo se dividió en cuatro capítulos donde se incluye toda la investigación, el diseño e implementación como se describe a continuación:

Capítulo 1. Fundamentación teórico metodológica de la trazabilidad de los medicamentos: contiene todos los conceptos generales de la investigación así como el resultado de los estudios realizados sobre los sistemas informáticos para la gestión de la trazabilidad de los medicamentos, realizados con tecnología RFID, que existen a nivel mundial y nacional. También se encuentra una descripción de las herramientas y tecnologías escogidas para la realización del componente propuesto.

CAPÍTULO 2. Características y diseño del componente propuesto: en este capítulo se aborda todo lo relacionado con el negocio propuesto. Se describen detalladamente los procesos que lo componen y las actividades que se automatizarán, así como los requisitos funcionales y no funcionales. Se muestran los elementos básicos del diseño mediante la justificación del uso de patrones y los diagramas de clases con sus respectivas descripciones. Igualmente se describe la concepción arquitectónica del sistema con el objetivo de entender la estructura y dinámica de la organización del mismo.

CAPÍTULO 3. Implementación del componente propuesto: este capítulo contiene los principales componentes implementados, así como el proceso del que son resultado y una breve descripción de los estándares utilizados para la codificación y el tratamiento de errores, así como el Modelo de datos y el Diagrama de despliegue.

Capítulo 1. Fundamentación teórico metodológica de la trazabilidad de los medicamentos

En este capítulo se especifican los conceptos principales vinculados a la presente investigación, con el objetivo de lograr un mejor entendimiento del campo de acción. Se realiza el estudio del arte de los sistemas, tanto a nivel internacional como nacional, que utilizan la Identificación por Radiofrecuencia para conocer la trazabilidad de los productos farmacéuticos y que tienen como objetivo mejorar procesos relacionados con los planteados en la situación problemática, y se describen las tecnologías, herramientas y lenguaje utilizado.

1.1 Conceptos de interés

En la investigación se tratan conceptos de interés que pueden ser de difícil comprensión para el lector, y que poseen una gran importancia para fundamentar en teoría el alcance del componente propuesto. Por esto serán explicados detalladamente a continuación.

1.1.1 Trazabilidad

En Vocabulario Internacional de Términos Básicos y Generales de Metrología (*International Vocabulary of Basic and General Terms in Metrology*) de la Organización Internacional para la Estandarización (ISO, por sus siglas en inglés) la trazabilidad es conceptualizada como:

"La propiedad del resultado de una medida o del valor de un estándar donde este pueda estar relacionado con referencias especificadas, usualmente estándares nacionales o internacionales, a través de una cadena continua de comparaciones todas con incertidumbres especificadas" (5).

Los autores de este trabajo plantean que la trazabilidad se basa en seguir las huellas que deja un elemento mientras transita por una cadena antes de llegar al consumidor final, es la capacidad de identificar la ubicación y condición pasada o actual de un producto u objeto, así como conocer su historia. Se aplica con el objetivo de lograr mejoras en los distintos negocios como por ejemplo, minimizar errores o lograr ofertar un mejor servicio al cliente. Existen dos tipos básicos de trazabilidad:

Trazabilidad interna: se refiere al proceso para obtener los datos trazables dentro de la propia empresa o institución.

Trazabilidad externa: se refiere al proceso para obtener los datos trazables entre empresas es decir entre eslabones de una cadena de suministro.

Aplicaciones de la trazabilidad

La trazabilidad tiene infinidad de aplicaciones, sería muy difícil establecer un límite para ellas. En la actualidad está presente en casi todos los sectores, en el agropecuario en el control de los animales, el agrícola en el seguimiento de los alimentos, en la construcción para mantener el historial de los materiales utilizados y especialmente en el de la salud. En este sector la trazabilidad tiene amplios campos, se le puede dar seguimiento a una persona (profesional de la salud o paciente), un activo fijo tangible o a los medicamentos.

Dentro de las nuevas tecnologías de la información y las comunicaciones (TIC) existen varias que permiten describir la ruta o historia de un producto, así como sus características actuales y que se han venido despuntando como posibles soluciones a los problemas actuales. En las secciones siguientes se describen cada una de ellas.

Principales tecnologías en la trazabilidad

Entre las herramientas más utilizadas con el propósito de gestionar, de manera estandarizada, la trazabilidad de un objeto o producto en una cadena de distribución, se encuentran las tecnologías de identificación automática GS1-128, Datamatrix y RFID.

Código de barras GS1-128 (código lineal): es un sistema estándar de identificación mediante código de barras utilizado internacionalmente para la identificación de mercancías en entornos logísticos, se basa en el sistema de códigos de barras *Code-128*, el cual permite una flexibilización a la hora de componer los códigos con diferentes informaciones.

GS1-128 se organiza a través de los denominados Identificadores de Aplicación (IA), que indican el significado del dato que viene a continuación y que formato tiene, es decir, dice si es peso, longitud u otro.

así como cuántos decimales tiene o en que unidades de medida está expresado. Estos identificadores son estándar, de este modo una empresa puede leer las etiquetas generadas por otra sin previamente haberse puesto de acuerdo en el formato, lo que simplifica muchísimo el trabajo. Una etiqueta GS1-128, debe contener obligatoriamente tres campos:

- Anagrama o razón social de la empresa: información general de formato libre en el que se pueden incluir datos como la dirección, el registro de sanidad, textos fijos entre otros.
- Información humanamente legible: como mínimo tienen que estar indicados en caracteres legibles por el hombre, todos aquellos datos simbolizados en el código de barras. Junto a este tipo de información obligatoria se puede incluir cualquier otro tipo de información aclaratoria.
- Símbolo/s. Información de código de barras: se mostrará el código junto con su representación en barras.

El requisito indispensable de la etiqueta GS1-128 es que toda la información representada en el código de barras debe figurar como información humanamente legible, con el fin de facilitar la introducción manual de los datos en caso de fallo del sistema de lectura, de forma que el operario no tenga que conocer obligatoriamente cuál es el significado de cada identificador de aplicación (6).

Código de barras Datamatrix (código 2D): esta es una tecnología óptica, evolución del código de barras convencional. A diferencia de su antecesor, permite la grabación de un gran volumen de información en un formato reducido. Además del número de serie que identifica unívocamente al producto, permite la grabación de otros datos que mejoran el rendimiento y aportan valiosa información a la gestión del sistema de trazabilidad. La captura de datos se realiza con un scanner óptico 2D (7).

1.1.2 Tecnología de Identificación por Radiofrecuencia

La tecnología de Identificación por Radiofrecuencia ha sido utilizada desde 1940, cuando los militares estadounidenses la usaron para identificar sus aviones, y desde entonces su desarrollo no se ha detenido, todo lo contrario su crecimiento ha sido sostenido y cada vez más acelerado (8). Con los años se ha perfeccionado técnicamente disponiendo actualmente de estándares internacionalmente aceptados para las bandas de frecuencia de trabajo más habituales con mayor número de aplicaciones, y con la aceptación de las Administraciones Públicas responsables de la asignación de frecuencias (9).

Funcionamiento básico de la tecnología

Es una tecnología esencialmente de captura e identificación automática, (aunque igualmente permite la modificación), de información, contenida en etiquetas electrónicas (tags o transpondedores). Cuando estas etiquetas entran en el área de cobertura de un lector RFID, éste envía una señal para que la etiqueta le transmita la información almacenada en su memoria, habitualmente un código de identificación. Una de las claves de esta tecnología es que la recuperación de la información contenida en la etiqueta se realiza vía radiofrecuencia y sin necesidad de que exista contacto físico o visual (línea de vista) entre el dispositivo lector y las etiquetas, aunque en muchos casos se exige una cierta proximidad de esos elementos. Se pueden crear sistemas de baja, alta, ultra alta frecuencia y frecuencia microondas (10).

A continuación en la Figura 1.1 se muestra el esquema del funcionamiento de esta tecnología.



Figura 1. 1. Esquema de funcionamiento RFID

Fuente: Tecnologías de identificación (11)

Componentes de la tecnología de Identificación por Radiofrecuencia.

Los componentes de esta tecnología son básicamente cuatro: las etiquetas conocidas comúnmente como tags o transpondedores, los lectores o interrogadores, un ordenador y una capa intermedia (*middleware*).

Etiquetas: las etiquetas contienen la información del objeto, persona o animal que la porte. Están compuestas por un microchip encargado de almacenar la información y una antena para comunicarse con el lector. Se pueden clasificar de muchas formas, por el protocolo, la frecuencia entre otras, pero las principales clasificaciones son:

- Por su forma de alimentación: de acuerdo a esta clasificación existen dos tipos básicos de tags: los pasivos y los activos.

Tags pasivos: son aquellas etiquetas que no necesitan batería interna para emitir su señal, y que reciben la energía necesaria del campo que genera el lector.

Tags activos: estas incorporan una batería interna, lo cual reduce su tiempo de vida pero amplía su campo de acción (radio de cobertura). Dentro de este tipo de etiquetas se encuentran las que permanecen apagadas y se activan cuando entran al campo de acción del lector y son interrogadas por el mismo, y las que emiten su señal periódicamente sin ser interrogadas.

- Por su capacidad de programación: se encuentran las de solo lectura, las de una escritura y muchas lecturas y las de lectura/escritura.

Lector: es el encargado de interrogar a las etiquetas y leer la información enviada por estas, para ello tiene que proporcionarles la energía suficiente para que puedan transmitir. Consta principalmente de una antena, una unidad de control y un módulo de radiofrecuencia que incluye el emisor y el receptor.

Ordenador (Host): en él se desarrolla la aplicación, es el que transmite las órdenes al lector y gestiona la información recibida y se la envía al sistema de información. Puede recibir información de más de un lector.

Capa Intermedia (Middleware): es necesario para recoger, filtrar y manejar los datos. Es el software que se encarga de gestionar la conexión entre el hardware de RFID y los sistemas de información existentes en la aplicación. Se ocupa, entre otras cosas, del encaminamiento de los datos entre los lectores, las etiquetas y los sistemas de información, y es el responsable de la calidad y usabilidad de las aplicaciones basadas en RFID. Tiene 4 funciones básicas:

- Adquisición de datos.
- Encaminamiento de los datos.
- Gestión de procesos.
- Gestión de dispositivos (12).

Estos componentes interactúan entre sí mediante los protocolos de comunicación.

1.1.3 Protocolos de comunicación en la tecnología de Identificación por Radiofrecuencia

Un protocolo es un método estandarizado, un conjunto de reglas que permite la comunicación entre varios procesos.

Con la evolución de la tecnología RFID, cada país fue estableciendo sus propios protocolos de comunicación para las distintas frecuencias existentes. Esto funcionaba perfectamente mientras que los productos no se comercializaran con otro país, de ser así aparecían problemas a la hora de la comunicación porque para una misma frecuencia se usaban diferentes estándares. Esto trajo consigo la necesidad de establecer un solo estándar que permitieran la comunicación a nivel mundial. Esta tarea la asumieron principalmente la ISO y el Auto-ID Center (Tecnologías de Auto Identificación).

Como resultado actualmente existen muchos protocolos de comunicación para RFID y están recogidos en los diferentes estándares de la misma. Algunos de estos protocolos y estándares son:

Código Electrónico del Producto (RFID-EPC): fue desarrollado en el Auto-ID Center del MIT (*Massachusetts Institute of Technology*). Define un código universal genérico y un protocolo RFID común. El código definido tiene la forma (para un código ePC de 96 bits) como se muestra a continuación en la Figura 1.2:



Figura 1. 2. Formato del código electrónico del producto (EPC)

Fuente: El estándar epc (13)

Esta estructura le permite identificar el fabricante, el producto y igualmente lo provee de un número que lo identifica unívocamente. El Código Electrónico de Producto (EPC, por sus siglas en inglés) define, asimismo, una estructura de red mediante la cual el ID del tag se utiliza como clave primaria para acceder a un sistema global de base de datos de productos (14).

Comunicación de Campo Cercano y Protocolo (NFCIP-1 *Near Field Communication and Protocol* por sus siglas en inglés): es un protocolo de interfaz inalámbrica; la comunicación se realiza entre dos entidades (punto a punto): aplicaciones de red y dispositivos electrónicos. Opera en la banda de los 13,56 MHz y tiene un alcance de funcionamiento de 20 cm. En este protocolo siempre hay uno que inicia la conversación, y este es el que la vigilará. Este rol es intercambiable entre las dos partes implicadas (15).

Comunicación de Campo Cercano y Protocolo (NFCIP-2 *Near Field Communication and Protocol*): la interfaz de comunicaciones de campo cercano (NFC) y el Protocolo-2, especifican el mecanismo de selección de modo de comunicación (ECMA 352). Este protocolo distribuye la ubicación de todos los dispositivos NFCIP-1, ISO 14443 e ISO 15693 que operen a 13,56 MHz, pero con diferentes protocolos. Está especificado en NFCIP-2 que los dispositivos puedan entrar en uno de los tres modos de comunicación y son diseñados para no perturbar otros campos de Radiofrecuencia a 13,56 MHz (16).

Organización Internacional de Normalización 18000 (ISO 18000): esta norma define la interfaz aérea, los mecanismos de detección de colisión y el protocolo de comunicación para una etiqueta en diferentes bandas de frecuencia (17).

Tarjetas de proximidad (ISO 14443): esta norma especifica dos estándares tipo A y B para el protocolo de transmisión (inicialización, técnica de anticolisión e interfaz aérea) en la capa enlace. Opera a una frecuencia portadora de 13,56 MHz; especifica una velocidad predeterminada de 106 kilobits por segundo, suficiente para evitar colisiones (18).

Tarjetas vecinas (ISO 15693): el estándar describe el protocolo de transmisión, la técnica de anticolisión y la interfaz aérea en la capa de enlace. A diferencia de las tarjetas de proximidad establece las características físicas del tag y sus protocolos de transmisión. Opera a una frecuencia de modulación de 13,56 MHz, especifica una velocidad predeterminada de 26 kilobits por segundo, suficiente para evitar colisiones (19).

Después del análisis previo de cada protocolo, y basado en la gran aceptación obtenida a nivel mundial por el Low-Level Reader Protocol, se escoge este como el encargado de realizar la comunicación en el desarrollo propuesto.

Protocolo de lectura de bajo nivel

A pesar de la gran cantidad de protocolos y estándares existentes para RFID, el más usado a nivel mundial es sin dudas el estándar EPCGlobal. Perteneciente a este estándar se encontró el *Low-Level Reader Protocol (LLRP)*.

El *Low-Level Reader Protocol* es un protocolo de lectura de bajo nivel que especifica una interfaz de red entre los lectores RFID y clientes. Facilita la administración de dispositivos lectores, mitigando las interferencias lector-etiqueta y lector-lector, y maximiza la eficiencia de señalización y operaciones con datos sobre las etiquetas (20).

Soporta el protocolo aire (*air protocol*) *Ultra High Frequency Class-1 Generation-2* (UHF C1G2), estándar que fue desarrollado por la EPCglobal para estandarizar el protocolo de interfaz aire vía radiofrecuencia

entre las etiquetas y los lectores RFID y el cual define los requerimientos físicos y lógicos para la comunicación entre las etiquetas y los lectores RFID (21).

El LLRP provee el control de todas las funcionalidades básicas de cualquier lector que soporte el *air-protocol* UHF C1G2, y posibilita la creación de extensiones sobre el mismo protocolo, valiéndose de un marco de trabajo que ofrece (22).

Funcionamiento del *Low-Level Reader Protocol*

- Se configura el lector para satisfacer las necesidades del usuario.
- Envía comandos de operaciones (*Reader Operation Specification* o ROSpec) al lector y estos contienen una lista de comandos secuenciales (*Antenna Inventory Specifications* o AISpec) para la lectura.
- Envía comandos (*Acces Specifications* o AccessSpec) al lector para indicarle que operaciones debe realizar sobre el tag.
- Obtención de los reportes de información por parte del lector RFID (23).

1.1.4 Beneficios generales de la identificación por radiofrecuencia

La tecnología tiene innumerables ventajas, las facilidades que ofrece de dar seguimiento en tiempo real de cualquier objeto permitiendo su identificación y localización, la recuperación de información vía radiofrecuencia sin la necesidad de que exista el contacto visual entre sus componentes y la posibilidad de controlar y restringir el acceso a determinados locales, la ubican muy por encima de sus competidoras en el mercado y le abren posibilidades de utilización en casi todos los sectores de la sociedad actual. En un mundo tan agitado como el de hoy, esta tecnología representa en la mayoría de los casos un modo de aligerar el trabajo, evitando a la vez la ocurrencia de errores humanos.

En el componente propuesto se aprovecharon estas ventajas para poder determinar en cada momento la localización de los medicamentos dentro del hospital y poder reconstruir de ser necesario el historial de los mismos, haciendo uso de las trazas que la tecnología permite almacenar. De esta forma se evita el robo o la falsificación de los productos aumentando la seguridad del sistema.

1.2 Sistemas para la trazabilidad basados en Radiofrecuencia

Los sistemas de información hospitalaria buscan mantener una trazabilidad de los productos farmacéuticos y principalmente de los medicamentos, en su recorrido dentro del hospital para poder garantizar su autenticidad y uso correcto. Por ello se hace necesario utilizar sistemas basados en RFID, que les ofrezcan estas garantías.

Sistemas internacionales

A nivel internacional se encontraron diversos sistemas que utilizan RFID para gestionar la trazabilidad de los productos farmacéuticos. Entre ellos se encuentran:

Trazamed

País: Argentina.

Es un sistema desarrollado para llevar a cabo el control y seguimiento de productos medicinales ya sea en procesos productivos o logísticos, respetando y cumpliendo con lo dispuesto en la normativa de la Administración Nacional de Medicamentos, Alimentos y Tecnología Médica (ANMAT) (24).

El sistema permite realizar lecturas de códigos basados en el estándar Global GS1 Datamatrix y poder mostrar información mediante el GTIN (Número Mundial de Artículo Comercial), Lote, Vencimiento, Serie, Laboratorio, Presentación y Dosis unitarias. Está basado en estándares de GS1. Tiene una arquitectura flexible, lo que le posibilita funcionar en pequeñas operaciones y expandirse rápidamente de ser necesario. Permite realizar una serie de operaciones por ejemplo:

- Autorización de movimientos.
- Consultas y reportes.
- Impresión de series.
- Permite administrar localmente los productos, los agentes, la seguridad y la empresa.

Farmatrack®

País: Argentina.

Con Farmatrack cada etapa en el ciclo de vida de un producto farmacéutico es gestionada y verificada, desde la fabricación hasta su consumo, y en cada instancia del proceso de distribución, el historial del medicamento puede ser reconstruido. Cada “actor”, o responsable de un producto, puede verificar, en tiempo real, el origen e historial del producto que tienen en sus manos, garantizando autenticidad y calidad del medicamento al paciente. Este sistema además de garantizar la disminución del factor “riesgo de vida” del paciente, asegura el circuito legal del medicamento (25).

La etiqueta de trazabilidad utilizada tiene 11 medidas de seguridad con soporte en múltiples tecnologías como el código de barras GS1-128; la codificación de datos bidimensional (Datamatrix), o el sistema RFID de identificación por radiofrecuencia.

El sistema es único en su tipo en Argentina porque no solo es el profesional de la salud quien realiza la verificación del medicamento sino que posibilita que también lo haga el paciente (26).

TS-RFID: Trazabilidad de Medicamentos

País: Uruguay.

Este sistema de trazabilidad de medicamentos permite identificar múltiples artículos a metros de distancia mediante la identificación por radiofrecuencia. Para ello se coloca un TAG RFID (un chip) a cada unidad, el cual quedará asociado en el software que recibirá las lecturas de las diferentes antenas que estarán ubicadas en los distintos puntos del hospital o sanatorio.

Se realiza la trazabilidad del medicamento desde el momento que ingresa al depósito hasta que son suministrados al paciente.

Entre sus características principales se encuentran:

- Permite su total integración con sistemas de hospitales o sanatorios.
- Fácil y rápida asociación de tags a los medicamentos.
- Permite un eficiente control sobre los movimientos de los medicamentos, tanto por pedidos como fuera de ellos.
- Reportes detallados de ingresos y egresos de vehículos.

- Gestión del stock de medicamentos (27).

Gador

País: Argentina.

El sistema se basa en etiquetar los medicamentos para su identificación tanto de forma individual como a nivel de pack y de pallet. Incluye todas las etapas del proceso de manufactura. Opera con estándares internacionales EPC Generación 2, basada en el equipamiento de lectura y chips desarrollados por la firma Impinj. No afecta la velocidad del proceso de envasado y a pesar de la misma permite leer y grabar los datos. Puede realizar una lectura masiva de 360 unidades lo que permite identificar los packs que conforman un pallet y así poder mantener su trazabilidad tanto internamente como a través de la cadena de abastecimiento (28).

Kit Check

País: Maryland, EEUU.

Este es un sistema de trazabilidad para los medicamentos contenidos en bandejas en los carros de paradas, los cuales son estaciones de emergencia que contienen equipo médico y bandejas de medicamentos. Estas bandejas están equipadas con tecnología RFID, lo que garantiza que los medicamentos están ahí y que no han vencido, y facilita el trabajo de los farmacéuticos. Asimismo posibilita reemplazar medicamentos cuando hay disponibilidad de los mismos. El etiquetado de los medicamentos y bandejas aún se realiza en la universidad por parte de los farmacéuticos.

Funcionamiento básico del sistema:

- A cada una de 25-50 drogas en una bandeja del carro de paradas se le adjunta una etiqueta RFID pasiva o no energizada.
- En el hospital se rotan cientos de bandejas por los carritos desplegados en el hospital.
- El departamento de farmacia del hospital escanea cada bandeja y detectan los medicamentos que faltan o que están vencidos (29).

GLT - Sistema de Gestión Logística y Trazabilidad

País: España

GLT es un sistema de gestión de almacenes ampliable y adaptable al requerimiento de cada cliente. Trabaja con todos los tipos estándar de codificación, tales como EPC, GS1-128 u otros códigos, cuya elección dependerá de los requerimientos del propio cliente y de los mercados que desee abarcar. Incorpora el módulo de trazabilidad en el paquete estándar. Ofrece el 100% de cobertura en la trazabilidad de todos los productos dados de alta en el sistema GLT.

Características:

- Identificación y control de mercancía por múltiples códigos de barras 1D y 2D y por medio de RFID.
- Gestión multi-almacén, multi-área y multi-empresa.
- Gestión de ubicaciones multiartículo, multicontenedor, multiformato, monoformato y clases.
- Gestión total de la volumetría por cajas y contenedores, asegurando un aprovechamiento máximo del tipo de caja o pallet en cada pedido.
- Gestión total de la trazabilidad en todo el proceso productivo y/o de distribución y las fechas de caducidad.
- Planificación, optimización, gestión y ejecución de rutas en los flujos de la mercancía.
- Sistema de monitorización mediante el uso de gráficas y estadísticas.
- Gestión de existencias y reaprovisionamientos automáticos.
- Permite administrar un almacén de forma automática, por radio o por documentos (en papel).
- Administración y gestión de lotes por precaducidad y caducidad.
- Integración con los ERP's más prestigiosos del mercado que aseguran plena compatibilidad en todas nuestras instalaciones.
- Soporte a proveedores, revisiones y clasificaciones de rechazos (30).

Estos sistemas se rechazan como posibles soluciones al problema planteado por las siguientes razones:

- Los sistemas Trazamed, Farmatrack, TS-RFID: Trazabilidad de Medicamentos, el Caso Gador y GLT abarcan procesos innecesarios para la solución propuesta, ya sean la fabricación, comercialización a lo largo de la cadena de suministro o el etiquetado de los medicamentos.

- Los sistemas Trazamed, Farmatrack y GLT, están diseñados para utilizar otras tecnologías alternativas a RFID como el código de barras GS1-128 y la codificación de datos bidimensional (Datamatrix), que no son objetivos en la solución propuesta, porque a pesar de tener ciertas ventajas no superan las ofrecidas por RFID.
- Los sistemas Trazamed, Farmatrack, el Caso Gador y Kit Check están diseñados para responder a las necesidades específicas de la institución.
- El sistema Kit Check no implementa todas las funcionalidades necesarias para la solución buscada, y no realiza todos los procesos requeridos.
- Además los sistemas TS-RFID: Trazabilidad de Medicamentos y GLT son software propietario.

Todas estas características hacen imposible la integración de estos sistemas con el xavia HIS, sin embargo su estudio aportó una guía para la implementación de las principales funcionalidades, así como una visión de los procesos más importantes para gestionar la trazabilidad de los medicamentos y que son necesarios para la solución propuesta.

Sistemas nacionales

Después de haber realizado un estudio del mercado nacional no se encontró ningún sistema que utilice la tecnología de Identificación por Radiofrecuencia para la trazabilidad de medicamentos.

1.3 Herramientas y tecnologías utilizadas

Para poder satisfacer las necesidades del mercado actual es necesario buscar alternativas para aumentar la calidad de los productos ofertados y así poder seguir en la competencia, por tanto se hace necesario mejorar en cuanto a eficiencia e integración entre las diferentes tecnologías. No cabe duda que la calidad del proceso de desarrollo de un sistema repercute grandemente en el resultado final del producto. Por eso se hace tan importante definir correctamente el ambiente de desarrollo a la hora de emprender cualquier proyecto.

Guía para el desarrollo de software

Modelo de Madurez de Capacidad Integrado (CMMI): pertenece a la familia de modelos desarrollados por el SEI (Software Engineering Institute) para la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software de una empresa. Es un modelo descriptivo que detalla los atributos esenciales que deberían caracterizar a una organización en un determinado nivel de maduración. Es una colección estructurada de elementos, que describe características de procesos que han demostrado, por experiencia, ser exitosos. Es recomendado para organizaciones que quieren incrementar la capacidad de su proceso de desarrollo y desarrollar software con calidad.

El modelo de CMMI plantea 5 niveles de madurez. Cada nivel es un escalón bien definido de mejora de proceso y estabiliza una parte importante de los procesos organizacionales.

- Nivel 1. Inicial.
- Nivel 2. Administrado.
- Nivel 3. Definido.
- Nivel 4. Cuantitativamente administrado.
- Nivel 5. Optimizado (31).

Cada nivel se divide en áreas de proceso, en total consta de 22 áreas.

Cada organización de desarrollo de software, es sometida a una evaluación que arroja como resultado una certificación a nivel internacional según el nivel de CMMI que tenga vencido. La UCI estuvo inmersa en este proceso de mejora y alcanzó el nivel 2, lo cual implica que todos los proyectos realizados en la universidad tengan una mayor calidad de desarrollo, una mayor solidez metodológica y un aval internacional a la hora de comercializarlos, al haber transitado por todas las áreas del nivel. Además la información en cada producto está registrada basándose en la experiencia de utilización y buenas prácticas de los expertos de la metodología Proceso Unificado de Desarrollo (RUP).

El nivel 2 consta de 7 áreas de proceso las cuales son:

- REQM - Gestión de Requisitos
- PP - Planificación de Proyectos
- PMC - Seguimiento y Control de Proyectos

- SAM - Acuerdos con Proveedores
- MA - Medición y Análisis
- PPQA - Aseguramiento de la Calidad de Procesos y Productos
- CM - Gestión de la Configuración

Java

Java V1.6: el componente será desarrollado en Java. Este es un lenguaje de programación orientado a objetos (POO), de alto nivel y semicompilado que funciona con una máquina virtual. Puede ser ejecutado en la mayoría de los sistemas operativos, es de un software de distribución libre, es rápido, multitarea, seguro y fiable.

La plataforma Java está basada únicamente en software que corre por encima de las plataformas basadas en hardware. Consta de dos componentes:

- La Máquina Virtual de Java (JVM)
- La Interfaz de Programación de Aplicaciones de Java (API Java) (32).

Máquina Virtual de Java (*Java Virtual Machine*): es el programa que interpreta el código Java (*bytecode*) y lo traduce al lenguaje del sistema. Es el núcleo de Java y es lo que hace que sea portable, debido a que si una máquina virtual Java se ha implementado en una determinada plataforma, cualquier programa Java se puede ejecutar en esa plataforma (33).

Librerías, plataformas y componentes

Las librerías son subprogramas que se usan como parte de algunos programas y que le proporcionan códigos de determinados servicios para poder desarrollar el software.

Las plataformas a su vez son soluciones que contienen herramientas y motores de ejecución, que permiten reutilizar el código que ya tienen facilitando así el desarrollo de los sistemas.

En este epígrafe se caracterizan las diferentes librerías necesarias en cada capa, para la realización del componente propuesto.

Java Enterprise Edition V5.0: la plataforma Java Enterprise Edition (Java EE) es una especificación que engloba dentro de sí misma a un conjunto de especificaciones (APIs) relacionadas. Ofrece un conjunto de especificaciones y técnicas que proporcionan soluciones completas, seguras, estables y escalables para el desarrollo, despliegue y gestión de aplicaciones de múltiples niveles de funcionalidad basadas en servidores. Permite reducir el costo y complejidad de desarrollo, lo cual resulta en servicios que se pueden desplegar y extender fácilmente (34).

La plataforma Java EE 5 está basada en J2SE 5.0 (*Java Standard Edition*). Implica un modelo de aplicaciones distribuidas en diversas capas o niveles (tier). Contiene tecnologías de componentes, servicios, comunicaciones, Servicios Web y XML, y define como coordinarlos (35).

Principales características de JEE5

- ✓ Provee múltiples plataformas.
- ✓ Simplifica el proceso de desarrollo y despliegue.
- ✓ Integración con sistemas de información existentes.
- ✓ Tecnología estándar en el mercado (36).

RichFaces V 3.1: en el desarrollo del componente se utiliza la versión 3.1 la cual brinda la posibilidad de crear interfaces de usuario modernas de una manera más eficiente y rápida. Es una librería de componentes visuales para JSF, con un avanzado marco de trabajo para la integración de funcionalidades Ajax en dichos componentes visuales, mediante el soporte de la librería Ajax4JSF. Se integra perfectamente en el ciclo de vida de JSF, soporta Facelets y es un proyecto de código abierto (37). Implementa completamente la característica skinnability del marco Ajax4jsf incluyendo un gran número de aspectos predefinidos (38).

Facelets V1.1: Java Server Facelets 1.1 permite diseñar páginas web de forma libre y luego asociarle los componentes JSF. Es un marco de trabajo para plantillas centrado en la tecnología JSF (Java Server Faces), por lo cual se integran de manera muy fácil. Entre sus ventajas más significativas están:

- La posibilidad de separar los componentes de la interfaz de usuario en archivos diferentes.
- Cuenta con un buen sistema de reporte de errores.

- No depende de un contenedor Web.
- En resumen, su uso reduce el tiempo y esfuerzo que se necesita para el desarrollo y despliegue de aplicaciones (39).

Lenguaje de Marcado de Hipertexto Extensible (Xhtml por sus siglas en inglés): lenguaje extensible de marcado de hipertexto en inglés eXtensible Hyper Text Markup Language, es la combinación de HTML y XML. Básicamente tiene las mismas funcionalidades pero cumple las especificaciones más estrictas de XML (40).

Java Server Faces (JSF) V1.2: es un marco de trabajo para crear aplicaciones java J2EE basadas en el patrón Modelo Vista Controlador de tipo 1. Para generar las vistas utiliza páginas JSF y añade una biblioteca de etiquetas propia para crear los elementos de los formularios HTML. Divide el procesamiento de una petición en una serie de etapas como son la validación, reconstrucción de la vista, entre otras. (41).

Ajax4JSF: Ajax4jsf es una librería de código abierto que se integra totalmente en la arquitectura de JSF y extiende la funcionalidad de sus etiquetas dotándolas con tecnología Ajax de forma limpia y sin añadir código JavaScript. Mediante este marco de trabajo se puede variar el ciclo de vida de una petición JSF, recargar determinados componentes de la página sin necesidad de recargarla por completo, realizar peticiones al servidor automáticas, control de cualquier evento de usuario, entre otras operaciones. (42).

JBoss Seam V2.1.1: es una potente plataforma de desarrollo, de código abierto, para crear sofisticadas aplicaciones web en Java. Integra tecnologías como Asynchronous JavaScript y XML (AJAX), Java Server Faces (JSF), Java Persistence (JPA), Enterprise Java Beans (EJB 3.0) y Business Process Management (BPM) en un sistema unificado completo conjunto de soluciones. Permite a los desarrolladores ensamblar aplicaciones web complejas con simples clases Java anotadas, y muy poco XML (43).

Además se puede integrar perfectamente a otros marcos de trabajos como RichFaces e Hibernate entre otros.

Hibernate V3.3: es una herramienta que realiza el mapping entre el mundo orientado a objetos de las aplicaciones y el mundo entidad-relación de las bases de datos en entornos Java. El término utilizado es

ORM (Mapeo Objeto Relacional) y consiste en la técnica de realizar la transición de una representación de los datos de un modelo relacional a un modelo orientado a objetos y viceversa.

Hibernate no solo realiza esta transformación sino que nos proporciona capacidades mediante una extensión de las consultas SQL (HQL), para la obtención y almacenamiento de datos de la base de datos que nos reducen el tiempo de desarrollo. Además permite crear aplicaciones y distribuirlas de forma gratuita (44).

Java Persistence API (JPA) : *Java Persistence API* más conocida por sus siglas JPA, proporciona un estándar para gestionar datos relacionales en aplicaciones Java SE o Java EE, de forma que además se simplifique el desarrollo de la persistencia de datos. Es una API de persistencia de POJOs (*Plain Old Java Object*), objetos simples que no heredan ni implementan otras clases (como los EJBs).

Ha combinado ideas y conceptos de los principales marcos de trabajos de persistencia, como Hibernate, Toplink y JDO, y de las versiones anteriores de EJB (45).

Enterprise JavaBeans (EJB) V3.0: la especificación de *Enterprise JavaBeans* define una arquitectura para un sistema transaccional de objetos distribuidos basado en componentes. Manda un modelo de programación; es decir, convenciones o protocolos y un conjunto de clases e interfaces que crean el API EJB. Este modelo de programación proporciona a los desarrolladores de Beans y a los vendedores de servidores EJB un conjunto de contratos que definen una plataforma de desarrollo común. Una aplicación EJB puede ser desplegada en cualquier servidor de aplicaciones que soporte J2EE. (46)

La especificación de EJB3 reconoce las ventajas de *Hibernate* y hace eco de mantener un mapeo objeto/relacional transparente. Estandariza los APIs de persistencia JPA y JDO (*Java Persistence API* y *Java Data Objects*), de esta forma la nueva especificación se convierte en una especie de clon de las mejores características de Hibernate a las que se sumaron lo mejor y más rescatable de JPA y JDO (47).

Servidor de aplicaciones

Servidos de aplicaciones JBoss (*JBoss Application Server*) 4.2: un servidor de aplicaciones es un software en red dedicado a facilitar la ejecución de otras aplicaciones, por lo general mediante internet a través del protocolo HTTP.

JBoss Application Server (o JBoss AS) es un software libre Java EE de código abierto, implementado en Java puro. Debido a que es basado en Java, puede ser utilizado en cualquier sistema operativo que soporte este lenguaje y además es ideal para aplicaciones en la web (48).

Herramientas de modelado

Lenguaje Unificado de Modelado (UML) 2.1: es el lenguaje que se definió para el trabajo en el componente propuesto, proporciona del vocabulario y las reglas necesarias para permitir una comunicación, se centra en la representación gráfica de un sistema, no especifica que metodología se debe usar, porque se puede utilizar con cualquiera que se escoja. Permite crear y visualizar los diagramas: de clases, de objetos, de casos de uso, de secuencia, de colaboración, de estados, de actividades, de componentes y de despliegue (49).

Visual Paradigm: Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Ayuda a construir con mayor rapidez aplicaciones de calidad, mejores y a un menor coste (50).

En el desarrollo del componente se utiliza la versión 8.0 de esta herramienta para la realización de todos los diagramas necesarios durante el desarrollo del software.

Herramienta de desarrollo

Eclipse SDK: Eclipse es un entorno de desarrollo integrado, de código abierto multiplataforma, para desarrollar Aplicaciones de Cliente Enriquecido.

Dispone de un editor de texto con resaltado de sintaxis. La compilación es en tiempo real. Tiene pruebas unitarias, control de versiones, clases, entre otros.

El SDK de Eclipse incluye las herramientas de desarrollo de Java, ofreciendo un IDE con un compilador de Java interno y un modelo completo de los archivos fuente de este lenguaje. Esto permite técnicas avanzadas de refactorización y análisis de código. El IDE hace uso de un espacio de trabajo (*frameworks*), muy ricos para desarrollar aplicaciones gráficas, y que permiten además modificaciones externas a los archivos en tanto se refresque el espacio de trabajo correspondiente (51).

Para la implementación del componente se utiliza la versión 3.4.2, porque asegura el rendimiento y la fortaleza de los sistemas.

Emulador de la tecnología

Emulador RFID: Rifidi es un conjunto de herramientas de código abierto para el desarrollo integrado de soluciones RFID, es una plataforma middleware. Comprende dos enfoques, uno más dirigido a la creación de prototipos y el otro centrado en el desarrollo y despliegue de aplicaciones. (52).

Estos emuladores son conocidos generalmente como simuladores, y se utilizan con el objetivo de simular el funcionamiento de un sistema RFID, cuando no se posee la tecnología necesaria para probarlo. Con este objetivo se utiliza en el componente desarrollado.

Marco de trabajo para reportes

iReport: la herramienta iReport es un constructor / diseñador de informes visual, poderoso, intuitivo y fácil de usar. Permite que los usuarios corrijan visualmente informes complejos con cartas, imágenes, subinformes, entre otros. Está escrito totalmente en Java y es de código libre, maneja el 98% de las etiquetas de JasperReports y permite diseñar con sus propias herramientas (53).

En la implementación del componente se hace uso de esta herramienta para la generación de los reportes necesarios.

Sistemas de gestión de base de datos

PostgreSQL: PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, de código abierto. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema de manera que si ocurre un fallo en un proceso no afectara al resto, y el sistema continuará funcionando. La conexión de la aplicación cliente puede ocurrir vía TCP/IP o sockets locales. Su principal proceso es el demonio postmaster, el cual es el encargado de escuchar esperando las conexiones entrantes, y es el que crea los procesos hijos que se encargan de autenticar estas peticiones, gestionar las consultas y mandar los resultados a las aplicaciones clientes (54).

Para el desarrollo del componente se utiliza la versión 8.4 la cual incorpora una gran cantidad de características, entre las principales se encuentran:

- Restauración de la base de datos usando procesos paralelos, acelerando la recuperación de un respaldo hasta en 8 veces respecto a la versión anterior.
- Configuración de idioma y ordenamiento por base de datos.
- Nuevas herramientas para monitorear las consultas, entregando mayor información a los administradores para saber lo que está sucediendo en la base de datos.
- Mejoras en procedimientos almacenados (55).

Pgadmin3: es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux, FreeBSD, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3 ejecutándose en cualquier plataforma.

Está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas.

La aplicación incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados, soporte para el motor de replicación Slony-I y mucho más (56).

Esta es la herramienta seleccionada para todo el proceso de creación y gestión de la base de datos del componente realizado.

Mediante el trabajo realizado hasta el momento se pudo arribar a las siguientes conclusiones:

- La tecnología RFID se postula a nivel mundial como la solución más eficaz para enfrentar los problemas actuales del sector farmacéutico. Brinda la seguridad y la trazabilidad necesitadas para asegurar la confianza en el sector de la salud.

- Para poder comprender mejor el alcance y los objetivos del componente propuesto se estudiaron diversos sistemas similares a nivel mundial. Permitiendo una mayor comprensión de las posibilidades ofrecidas por la tecnología y lo innovador del proyecto deseado.
- Como primeros pasos en el desarrollo del componente propuesto se realizaron las discusiones necesarias para definir las herramientas, metodologías y el lenguaje que se utilizaría.

Capítulo 2. Características y diseño del componente propuesto

En este capítulo se abordan los aspectos relacionados con el negocio y los principales procesos a desarrollar. Se plantean detalladamente los requerimientos funcionales y no funcionales que cumplirá la aplicación. Se abordan los temas relacionados con el diseño de la aplicación con el objetivo de modelarla de forma tal que soporte dichos requisitos. Igualmente se describe la arquitectura necesaria para la implementación y se incluyen los diagramas de clases del diseño, diagrama de interacción.

2.1 Modelado del negocio

En el proceso de desarrollo de software la segunda fase es el modelado del negocio, en esta el equipo de trabajo estudia el funcionamiento del negocio propuesto, sus principales características, necesidades, procesos y la interrelación existente entre ellos. Después de familiarizarse, se encuentra en condiciones de comenzar a desarrollar una propuesta del componente deseado.

Flujo actual de los procesos del negocio

Actualmente los procesos que se desarrollan en el negocio, en los hospitales de Cuba, son los siguientes:

- **Recepcionar medicamentos en el almacén**

El proceso se inicia cuando el almacén del hospital recibe del proveedor una entrega de medicamentos acompañada de una factura con todos los datos referentes al contenido de la entrega. Los medicamentos deben venir en cajas selladas y ser recibidos por el jefe o el dependiente del almacén. Estos son los encargados de verificar los paquetes y comprobar contra la factura si existe algún error. Si ocurre algún problema, ya sea que falta algún paquete o hay alguno defectuoso la persona que recibió la entrega debe notificar al proveedor el problema encontrado y dar entrada a los restantes medicamentos. En caso de no haber ningún problema simplemente se le dará entrada a todos los medicamentos recibidos mediante la creación del informe de recepción y la creación y actualización de las tarjetas de estiba.

A continuación en la Figura 2.1 se muestra el diagrama de clases del negocio perteneciente a este proceso, donde se modela el flujo de las acciones anteriormente descritas:

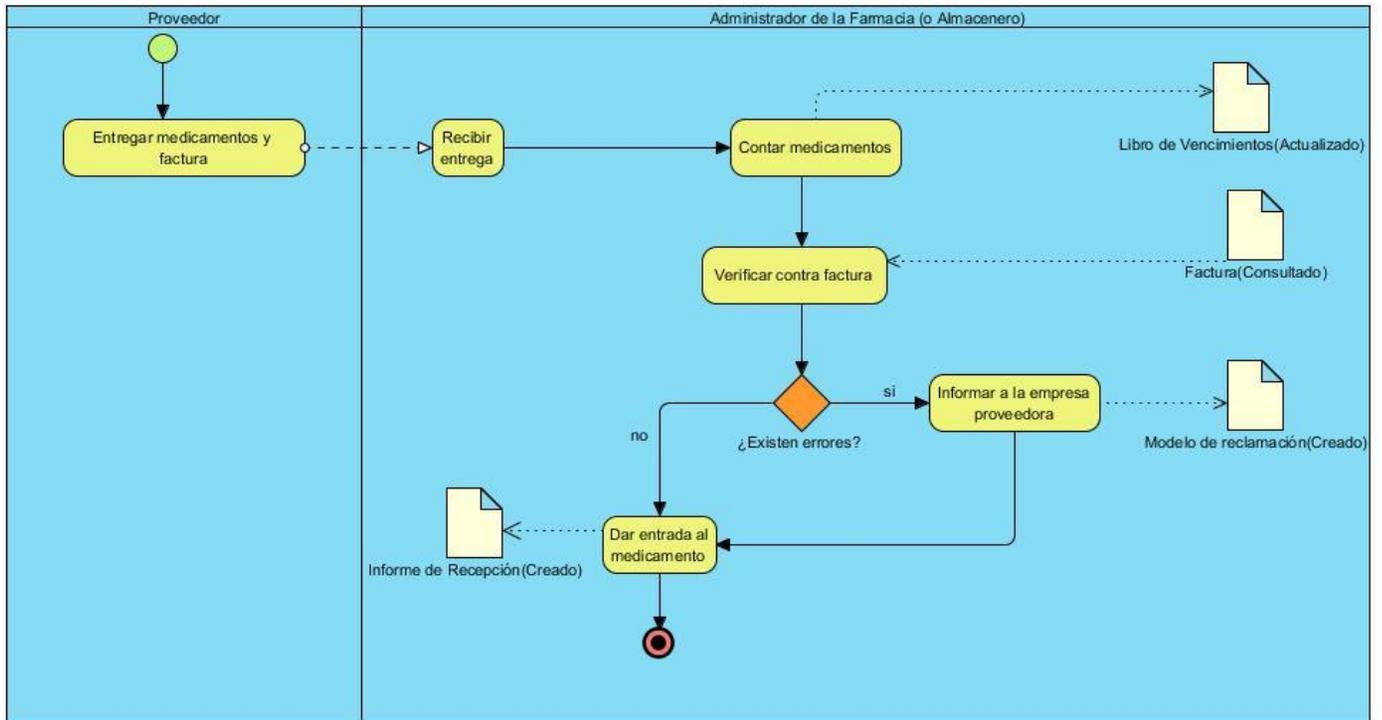


Figura 2. 1. Diagrama de clases del negocio Recepcionar medicamentos en el almacén

Fuente: Elaboración propia

- **Distribución de medicamentos a un eslabón posterior**

El responsable de uno de los centros de costo hace una solicitud de medicamentos al almacén mediante un modelo de solicitud de materiales. El jefe o el dependiente del almacén reciben esta petición, verifican los datos de la solicitud y de haber algún error la rechazan. En caso contrario, se comprueba la existencia del medicamento, y de existir, se realiza la entrega, creando como constancia un vale de salida y actualizando el saldo de los medicamentos en el almacén. Si no hay existencia del medicamento solicitado en esos momentos, se le notifica al responsable de centro de costo y se deniega la solicitud.

El diagrama de clases del negocio perteneciente a este proceso se puede ver en el Anexo 7.

- **Devolución de un producto a un eslabón anterior**

CAPÍTULO II: Características y diseño del componente

El responsable de uno de los centros de costo del hospital, solicita hacer una devolución de medicamentos mediante un modelo de devolución al almacén, que es recibido por el jefe o el dependiente de este. Se verifican los datos de la devolución y de no existir ningún error es aceptada. La persona que recibió la devolución es la encargada de actualizar el saldo del medicamento en el almacén.

El diagrama de clases del negocio perteneciente a este proceso se puede ver en el Anexo 8.

- **Administración del medicamento**

El trabajador de la salud busca el paciente dentro de su centro de costo y revisa su indicación médica. Si hay existencia del medicamento necesitado en el centro de costo en esos momentos, se le aplica y se actualiza la prescripción médica. De no haber el medicamento, se le notifica al responsable del centro de costo para que lo solicite al almacén.

El diagrama de clases del negocio perteneciente a este proceso se puede ver en el Anexo 9.

Estos procesos están explicados más detalladamente en el documento APS_SIAPS_0115_MPN-BPM2v1.0.

Actores del negocio

Los actores del negocio son cualquier persona u organización que interactúa con los procesos del negocio. A continuación en la Tabla 2.1 se muestran los involucrados en el negocio estudiado.

Tabla 2. 1. Actores del negocio

Fuente: Elaboración propia

Actor	Objetivo
Proveedor	Entregar medicamentos al hospital.

Jefe o dependiente del almacén	<ul style="list-style-type: none">• Recepcionar los medicamentos.• Actualizar las tarjetas de estiba de los medicamentos.• Entregar medicamentos a las áreas de costo del hospital.• Recibir las devoluciones hechas por las áreas de costo del hospital.
Responsable del área de costo	<ul style="list-style-type: none">• Hacer solicitud de medicamentos.• Hacer devolución de medicamentos.
Trabajador de la salud	Administrar el medicamento al paciente.

2.2 Características del componente propuesto

Como solución al problema planteado y teniendo en cuenta los procesos anteriormente descritos se propone la realización de un componente que utilice la tecnología RFID y permita integrarlo al xavia HIS. Sus principales características estarán centradas en:

- Permitir gestionar la trazabilidad de los medicamentos dentro del hospital.
- Garantizar la autenticidad de los medicamentos y evitar su falsificación.
- Agilizar de manera segura el proceso de recepción de los medicamentos.
- Permitir la generación de los reportes necesarios.

Toda la información manejada será persistida de manera segura en la base de datos del sistema, y solo podrá ser accedida por las personas autorizadas.

Además el componente debe cumplir con ciertos requerimientos que satisfagan las necesidades del cliente.

2.3 Especificación de los requerimientos del software

Los requerimientos son las cualidades que necesita el sistema para que funcione correctamente. La captura de los requerimientos se realiza directamente con el cliente con el objetivo de lograr un entendimiento entre este y los desarrolladores sobre las funcionalidades que deberá cumplir el sistema. Es imprescindible que se logre este objetivo debido a que de él depende el trabajo del equipo de desarrollo. (57)

Requerimientos funcionales

Los requerimientos funcionales (RF) determinan la capacidad del sistema para satisfacer las necesidades del cliente, o las condiciones de un contrato.

Después de un estudio del negocio se determinan los siguientes requerimientos funcionales para el componente propuesto:

Gestionar tipo de medicamento: estos requisitos son los encargados de gestionar la información relacionada con los medicamentos que entran al sistema por primera vez.

- RF 1 Adicionar tipo de medicamento
- RF 2 Eliminar tipo de medicamento
- RF 3 Modificar tipo de medicamento
- RF 4 Buscar tipo de medicamento

Gestionar antenas: se encargan de adicionar, modificar o eliminar las antenas RFID en el sistema, las cuáles son las que permiten leer la información contenida en las etiquetas de los medicamentos.

- RF 5 Adicionar antena
- RF 6 Eliminar antena
- RF 7 Modificar antena
- RF 8 Buscar antena

Gestionar área de costo: se encargan de adicionar, modificar o eliminar las áreas de la institución en el sistema, las cuáles son las que pueden solicitar medicamentos al almacén.

- RF 9 Adicionar área
- RF 10 Eliminar área
- RF 11 Modificar área
- RF 12 Buscar área

Gestionar unidad de medida: se encargan de adicionar, modificar o eliminar las unidades de medida en el sistema, las cuáles son las que identifican el tipo de medicamento, ya sea tabletas, ámpulas u otro.

- RF 13 Adicionar unidad de medida
- RF 14 Eliminar unidad de medida
- RF 15 Modificar unidad de medida
- RF 16 Buscar unidad de medida

Gestionar lectores: se encargan de adicionar, modificar o eliminar los lectores RFID en el sistema, los cuáles son los que interrogan a las antenas para obtener la información de los medicamentos.

- RF 17 Adicionar lector
- RF 18 Eliminar lector
- RF 19 Modificar lector
- RF 20 Buscar lector

Para los procesos y reportes son necesarios los requerimientos:

- RF 21 Registrar medicamentos del proveedor: se encarga de recepcionar los medicamentos y darle entrada en la base de datos.
- RF 22 Despachar a los centros de costo los medicamentos solicitados: distribuye los medicamentos del almacén hacia las áreas del hospital que los necesiten.
- RF 23 Administrar medicamento a un paciente: permite seleccionar el paciente y la cantidad de medicamento que se le desea aplicar.

- RF 24 Registrar medicamentos internos: permite recibir una devolución de medicamentos al almacén, o una entrega a otro almacén interno de la institución.
- RF 25 Monitorización de medicamentos por áreas: permite verificar si los medicamentos están en el área a la que pertenecen o si faltan en la misma.
- RF 26 Reporte de los medicamentos vencidos: crea un reporte con los medicamentos que han vencido en una fecha determinada.
- RF 27 Reporte de la utilización de los medicamentos: crea un reporte con información de la utilización de un medicamento desde su llegada al hospital.
- RF 28 Reporte del Informe de Recepción: crea un reporte con toda la información de los medicamentos recepcionados.
- RF 29 Reporte del Vale de Salida: crea un reporte como constancia de la entrega de medicamentos a un área determinada de la institución.
- RF 30 Reporte de la trazabilidad de un medicamento: crea un reporte con las trazas del medicamento y toda la información almacenada sobre él.

Requerimientos no funcionales

Los requerimientos no funcionales (RNF) son los que determinan las propiedades del sistema, como la usabilidad, confiabilidad, la rapidez de respuesta o ejecución. Generalmente se vinculan con los requerimientos funcionales (58). Existen muchas categorías para clasificarlos, a continuación se muestran las más importantes para el componente propuesto:

Usabilidad

Dentro de la usabilidad se encuentran los requerimientos que deben cumplir los usuarios que interactúen con el sistema, los cuales se muestran a continuación en la Tabla 2.2:

Tabla 2. 2. Datos de los requerimientos para los usuarios finales

Fuente: APS_SIAPS_0113_ERS2v1.0

No.	Sexo	Edad	Nivel de Escolaridad	Ocupación	Experiencia profesional	Experiencia con la Aplicación Informática	Tipo de discapacidad
1	F o M	20-50 años	Técnico medio.	Dependiente o Jefe del almacén.	Mínimo 6 meses.	Mínimo 1 mes.	Ninguna.
2	F o M	20-50 años	Universitario	Personal de salud	Mínimo 6 meses.	Mínimo 1 mes.	Ninguna.

RNF 1 Tipo de aplicación

Será una aplicación web, fácil de entender y usar aún para una persona con bajos conocimientos informáticos. El componente estará diseñado de manera que los usuarios adquieran las habilidades necesarias para explotarlo en un tiempo reducido.

RNF 2 Finalidad de la aplicación

La aplicación tiene como finalidad gestionar la trazabilidad de los medicamentos una vez que entran al hospital y hasta que son consumidos por los pacientes.

Confiabilidad

RNF 3 Políticas de respaldo

Se deben crear políticas de respaldo sobre toda la información obtenida, evitando así pérdidas en caso de la ocurrencia de algún incidente.

RNF 4 Recuperación de la información

Se podrá recuperar la información almacenada en la base de datos mediante la utilización de las copias de respaldo.

Eficiencia

RNF 5 Tiempo de respuesta

El tiempo de respuesta debe estar en correspondencia con las operaciones realizadas siendo este el mínimo posible.

RNF 6 Optimización de recursos

El componente garantizará la optimización de los recursos críticos y la disminución del volumen de datos en las peticiones.

Soporte

RNF 7 Mantenimiento

Una vez desplegado se debe contar con un manual de soporte que permita realizar mantenimientos periódicos al sistema en caso de ser necesarios.

RNF 8 Capacitación

Se debe realizar una capacitación del personal, en un tiempo aproximado de 1 mes.

Restricciones de diseño

RNF 9 Capa de presentación

La capa de presentación contendrá todas las interfaces del sistema y será la que mantendrá una comunicación con el negocio.

RNF 10 Capa de negocio

La capa del negocio contendrá los procesos a realizar con la información recibida desde la capa de presentación y a su vez procesa la información devuelta por la capa de datos.

RNF 11 Capa de acceso

La capa de acceso a datos contendrá las entidades y los objetos de acceso a datos correspondientes a las mismas.

Interfaz

RNF 12 Información

Se contara con una información correctamente organizada, facilitando así la correcta interpretación de la misma por parte de los usuarios.

RNF 13 Equipo necesario

El equipo de hardware debe contar con las siguientes características:

- Tipo de procesador: Intel Pentium IV o superior.
- Velocidad del procesador: 3.00 GHz.
- Memoria RAM: 2 GB o superior.
- Disco Duro: 80 GB para servidor de aplicación
- Se requiere tarjeta de red.
- Lector RFID: lectores que soporten la comunicación con las etiquetas a través del protocolo aire Ultra High Frequency Class-1 Generation-2.
- Etiquetas RFID: etiquetas que soporten la comunicación con los lectores a través del protocolo aire Ultra High Frequency Class-1 Generation-2.

2.4 Modelo de Diseño

El objetivo de este modelo es transformar los requerimientos obtenidos en un modelo lógico del sistema que se desea implementar. Se tiene en cuenta que soporte todos los requerimientos y reglas identificadas con anterioridad. Es utilizado como entrada para la fase de implementación. Está compuesto por diferentes diagramas, clases, interfaces, paquetes, protocolos entre otros.

Patrones de diseño

Los patrones son una solución a un problema recurrente en un contexto determinado, codifican conocimiento específico acumulado por la experiencia en un dominio, describen sus componentes, responsabilidades y relaciones. Un mismo patrón puede ser ejecutado por diversos sistemas sin que sea empleado dos veces de la misma forma. En general los patrones son un intento de rehusar la experiencia y formalizar la comunicación (59).

En la realización del componente propuesto se utilizaron algunos de los patrones GRASP o patrones de asignación de responsabilidades como se evidencia a continuación:

- **Experto:** este patrón indica que la responsabilidad de crear un objeto recae sobre la clase que posee la información necesaria para crearlo. Por esto fue utilizado para asignar responsabilidades a las clases que poseían la información necesaria para cumplirlas. En la Figura 2.2 se ejemplifica mediante la clase CCCodigoMedicamento (Clase Crear Código de Medicamento), la que posee el código y la descripción de un medicamento, datos que son suficientes para la inserción del mismo, por lo cual en la clase se encuentra el método Adicionar_Código ().

```
public class CCCodigoMedicamento {  
  
    private int codigo;  
    private String descripcion;  
  
    EntityManager entityManager;[]  
  
    FacesMessages facesMessages;[]  
  
    public String AdicionarCodigo() {[]  
  
        //Propiedades  
        public int getCodigo() {[]  
  
        public void setCodigo(int codigo) {[]  
  
        public String getDescripcion() {[]  
  
        public void setDescripcion(String descripcion) {[]  
  
    }  
}
```

Figura 2. 2. Fragmento de código perteneciente a la clase Crear Código de Medicamento

Fuente: Elaboración propia

- **Creador:** patrón responsable de asignar a las clases la responsabilidad de crear instancias de clases u objetos de entidades. La Figura 2.3 muestra un ejemplo de su uso en el desarrollo del componente mediante la creación de un objeto de la entidad TnCodigoMedicamento con el objetivo de persistir el mismo en la base de datos.

```
public class CCCodigoMedicamento {  
  
    private int codigo;  
    private String descripcion;  
  
    EntityManager entityManager;  
  
    FacesMessages facesMessages;  
  
    public String AdicionarCodigo() {  
        TnCodigoMedicamento codigo2 = new TnCodigoMedicamento();  
        codigo2.setDescripcion(this.descripcion);  
        codigo2.setCodigo(this.codigo);  
        TnCodigoMedicamento aux= new TnCodigoMedicamento();  
        try{  
            aux = (TnCodigoMedicamento) entityManager  
                .createQuery(  
                    "select a from TnCodigoMedicamento a where a.codigo=:para1 or a.descripcion=:para2")  
                .setParameter("para1", this.codigo).setParameter("para2",  
                    this.descripcion).getSingleResult();  
        }catch (Exception e) {  
  
        }  
        if (aux.getDescripcion()==null || aux.getDescripcion().equals("")) {  
            entityManager.persist(codigo2);  
            entityManager.flush();  
            facesMessages.add("Código adicionado correctamente");  
            return "succeed";  
        }  
        else  
        {  
  
        }  
    }  
}
```

Figura 2. 3. Fragmento de código para persistir entidad en la base de datos

Fuente: Elaboración propia

- **Bajo Acoplamiento:** este patrón indica que las clases deben estar lo menos relacionadas entre sí que sea posible, de forma que un cambio en una de ellas tenga la mínima repercusión en el resto. Se utilizó con el objetivo reducir el impacto de los cambios en las clases además de hacerlas más reutilizables.
- **Alta Cohesión:** para emplear este patrón se crearon las clases con el mínimo número de métodos posibles, delegando algunos a otras si la tarea es muy grande, de este modo el diseño se hace

más claro y fácil de comprender. El ejemplo de la Figura 2.4 se refiere a la funcionalidad Gestionar Código de Medicamento, la que se separa en cuatro clases, Crear, Eliminar, Modificar y Listar, cada una de estas cumple con su objetivo sin necesidad de estar sobrecargada, así como se menciona es más fácil entender y mantener el código sin crear afectaciones en otras funcionalidades.



Figura 2. 4. Paquete Código de Medicamento

Fuente: Elaboración propia

Descripción de la arquitectura

Arquitectura de software según su definición oficial, la IEEE Std 1471-2000 es: la organización fundamental de un sistema formada por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán, y los principios que orientan su diseño y evolución (60).

De manera sencilla y a modo de resumen se puede decir que la arquitectura es un conjunto de patrones que aportan una visión abstracta y coherente que sirve como marco para un entendimiento común entre los desarrolladores. Define un estilo o combinación de estilos para una solución. Por esto está estrechamente relacionada con el diseño y la implementación de las estructuras de software.

Es el diseño de más alto nivel de la estructura de un sistema, programa o aplicación y tiene la responsabilidad de:

- Definir los módulos principales
- Definir las responsabilidades que tendrá cada uno de estos módulos

- Definir la interacción que existirá entre dichos módulos:
 - Control y flujo de datos
 - Secuenciación de la información
 - Protocolos de interacción y comunicación
 - Ubicación en el hardware (61).

Patrón arquitectónico Modelo Vista Controlador

Para el diseño de este componente se utiliza el patrón arquitectónico Modelo-Vista-Controlador. Este patrón separa los datos, la lógica de control y la interfaz de usuario en tres componentes diferentes. El modelo no sabe nada sobre las vistas y la vista no sabe nada sobre el controlador.

Modelo: representa simplemente los datos, las operaciones y la lógica del negocio de la aplicación. Es independiente de la manera en la que se diseñan y desarrollan los componentes de software que muestran los datos y producen los reportes.

Vista: es la interfaz a través de la cual el modelo muestra los datos al usuario y genera reportes. Pueden existir varias vistas del modelo y cada una tiene asociada un componente controlador.

Controlador: es un componente de software que controla las interacciones entre el modelo y sus vistas. Traduce las interacciones del usuario a acciones que serán implementadas por el modelo (62).

En el componente se evidencia el uso de este modelo en su aplicación a las distintas funcionalidades donde todas las vistas, son páginas Xhtml desarrolladas con *Java Server Faces* y utilizando librerías como *RichFaces* y *Ajax*, en la Figura 2.5 se muestra un ejemplo de una ellas.

Recepción de medicamentos externos

Datos del medicamento a recibir

Medicamento: Cantidad por unidad: Unidad medida:

Precio venta: Fecha recepción: Fecha de vencimiento:

No. de factura: Lote:

Datos del area que recibe

Área:

Medicamentos encontrados

Cantidad	Medicamento	Unidad de medida
1	Dipirona	Tableta

Figura 2. 5. Vista de la interfaz recepción de medicamentos externos.

Fuente: Elaboración propia

Estas páginas envían sus datos a las clases controladoras, Figura 2.6, a través del método POST y estas persisten o eliminan los mismos del servidor de base de datos a través de la capa de modelo, Figura 2.7, la cual hace uso de *Hibernate* y la API para la persistencia de Java.

```
public class CCRRegistrarMedicamentos {  
  
    private TbArea tbArea;  
    private TnUnidadMedida tnUnidadMedida;  
    private String unidad_medida_seleccionada;  
    private TnCodigoMedicamento tnCodigoMedicamento;  
    private String codigo_medicamento_seleccionado;  
    private String idTag;  
    private Integer cantUnidades;  
    private Double precioVenta;  
    private String lote;  
    private Date fechaVencimiento;  
    private Date fechaRecepcion;  
    private Boolean enCirculacion;  
    private Boolean agotado;  
    private Boolean vencido;  
    private long no_factura;  
  
    private String antena_seleccionada;  
    private int no_antena;  
    private String area_seleccionada;  
    List<TbAntena> listaAntenasArea = new ArrayList<TbAntena>();  
    List<String> listaAntenasAreaString = new ArrayList<String>();  
  
    private List<TnUnidadMedida> listaunidadmedida = new ArrayList<TnUnidadMedida>();  
    private List<TnCodigoMedicamento> listacodigosmedicamentos = new ArrayList<TnCodigoMedicamento>();  
    private List<TnLector> listatnLector = new ArrayList<TnLector>();  
    private List<TbArea> listatnArea = new ArrayList<TbArea>();  
  
    private List<String> lista_tag = new ArrayList<String>();  
}
```

Figura 2. 6.Clase controladora registrar medicamentos

Fuente: Elaboración propia

```
@Entity
@Table(name = "tb_medicamento", schema = "modrfid")
public class TbMedicamento implements java.io.Serializable {

    private long id;
    private TbArea tbArea;
    private TnUnidadMedida tnUnidadMedida;
    private TnCodigoMedicamento tnCodigoMedicamento;
    private String idTag;
    private Integer cantUnidades;
    private Double precioVenta;
    private String lote;
    private Date fechaVencimiento;
    private Date fechaRecepcion;
    private Boolean enCirculacion;
    private Boolean agotado;
    private Boolean vencido;
    private Usuario usuario;
    long no_factura;

    private Set<TbUtilizacionMedicamentos> tbUtilizacionMedicamentos = new HashSet<TbUtilizacionMedicamentos>();
    private Set<TbSalida> tbSalidas = new HashSet<TbSalida>();

    public TbMedicamento() {}

    public TbMedicamento(long id, String idTag) {}

    public TbMedicamento() {}

    public long getId() {}
}
```

Figura 2. 7. Modelo para la creación de una entidad de la base de datos.

Fuente: Elaboración propia

El uso de este patrón trajo numerosas ventajas para el desarrollo del componente, por ejemplo:

- Separa la lógica del negocio, y la presentación creando una total independencia de funcionamiento.
- Permite la reutilización de los componentes.
- Simplifica el mantenimiento del sistema.
- El usuario interactúa con el modelo solamente a través de controladores.

- Permite que la aplicación sea escalable, en caso de ser requerido.

En la Figura 2.8 se muestra un ejemplo de la interacción de las capas de este patrón, haciendo uso de las imágenes anteriormente expuestas.

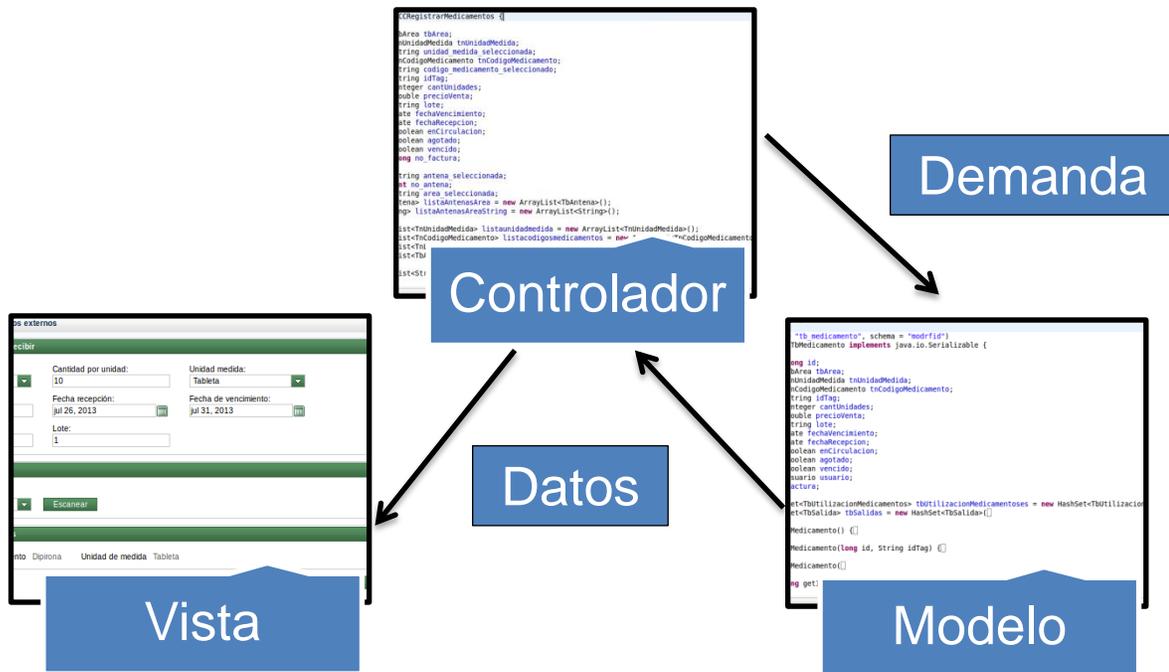


Figura 2. 8. Esquema del patrón Modelo Vista Controlador.

Fuente: Elaboración propia

Diagramas de clases del diseño

Cada requisito se transforma en una clase del diseño y se denotan como casos de uso del diseño. Seguidamente se muestran los diagramas de clases del diseño de los requisitos más significativos.

La Figura 2.9 muestra el diagrama de clases del diseño perteneciente al proceso registrar medicamentos externos, en el cual el actor recibe y recepciona los medicamentos entregados por el proveedor adicionándolos a la base de datos.

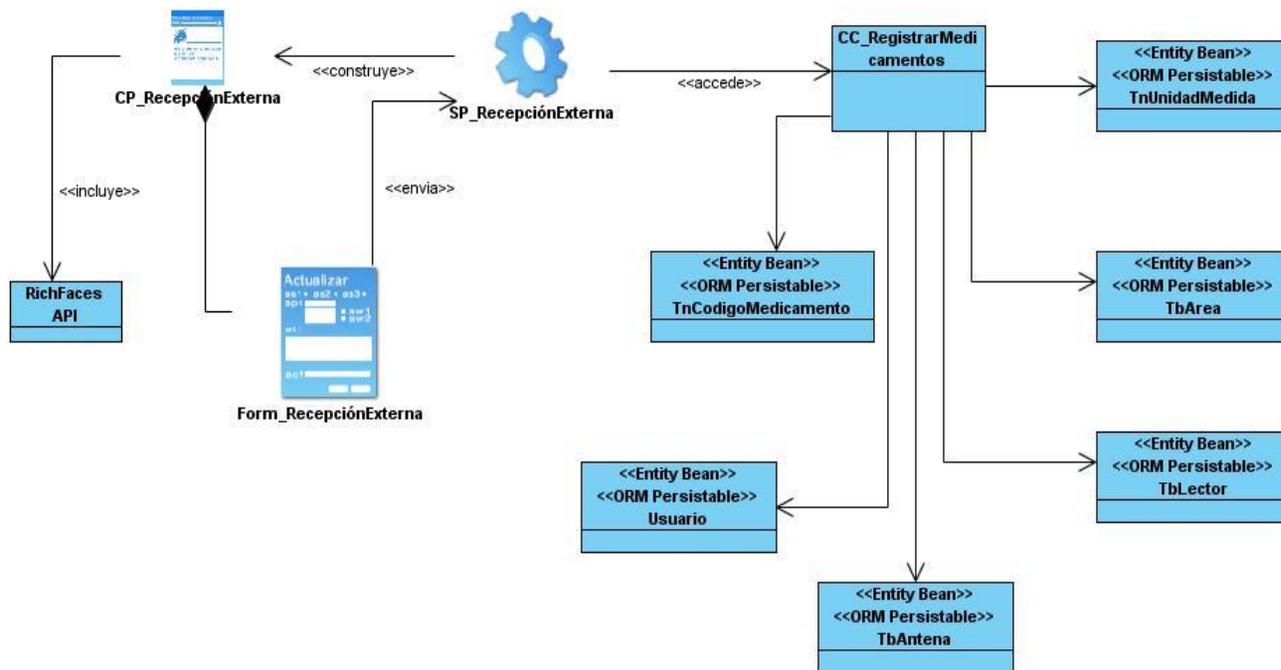


Figura 2. 9. Diagrama de clases del diseño registrar medicamentos externos

Fuente: Elaboración propia

La Figura 2.10 muestra el diagrama de clases del diseño perteneciente al proceso entregar los medicamentos a los centros de costo, en el cual el actor distribuye los medicamentos existentes en el almacén a las distintas áreas del hospital según la necesidad de estas.

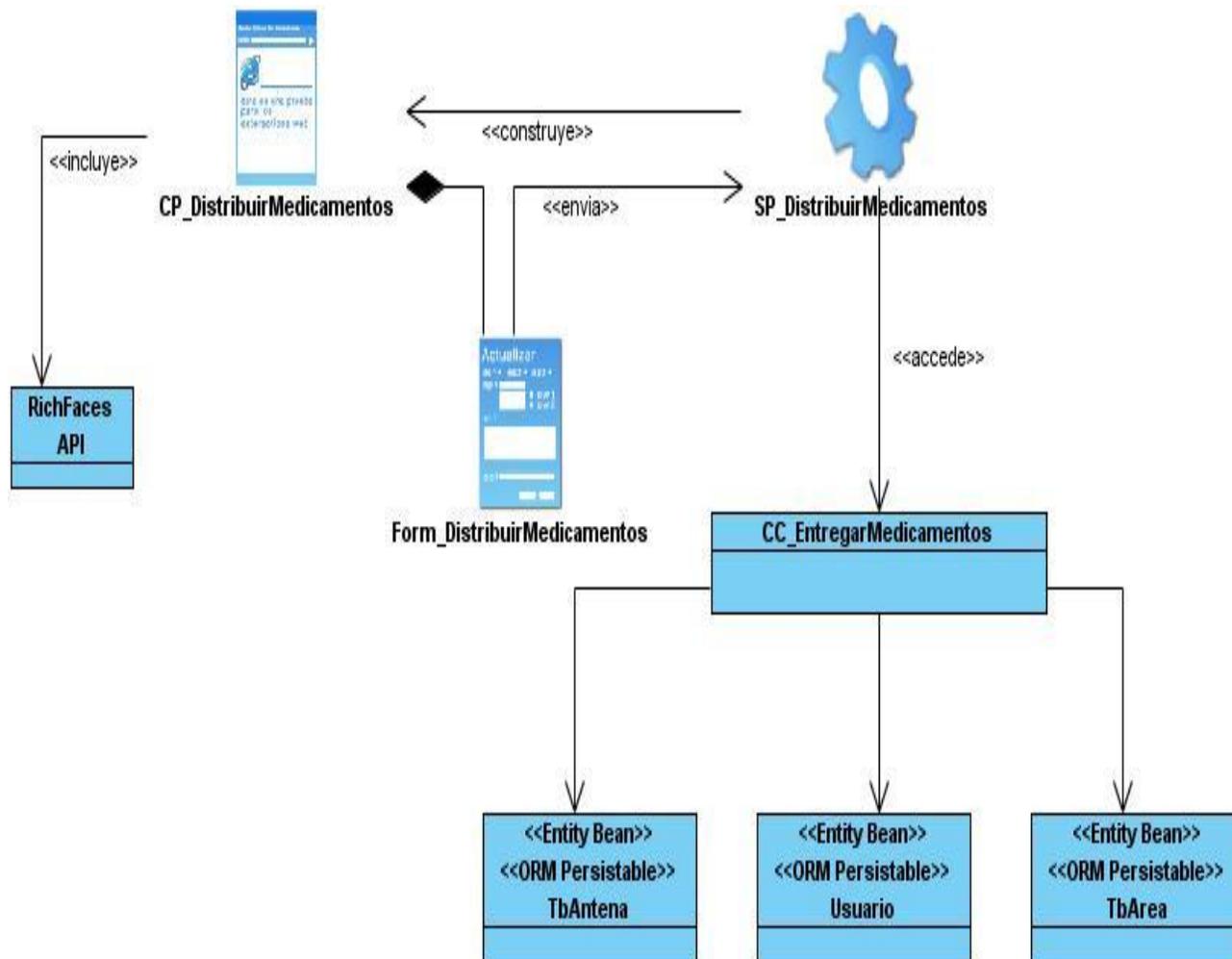


Figura 2. 10. Diagrama de clases del diseño entregar los medicamentos a los centros de costo

Fuente: Elaboración propia

La Figura 2.11 muestra el diagrama de clases del diseño perteneciente al proceso registrar medicamentos internos, mediante el cual el actor recibe los medicamentos provenientes de otra área del hospital y los recepciona internamente.

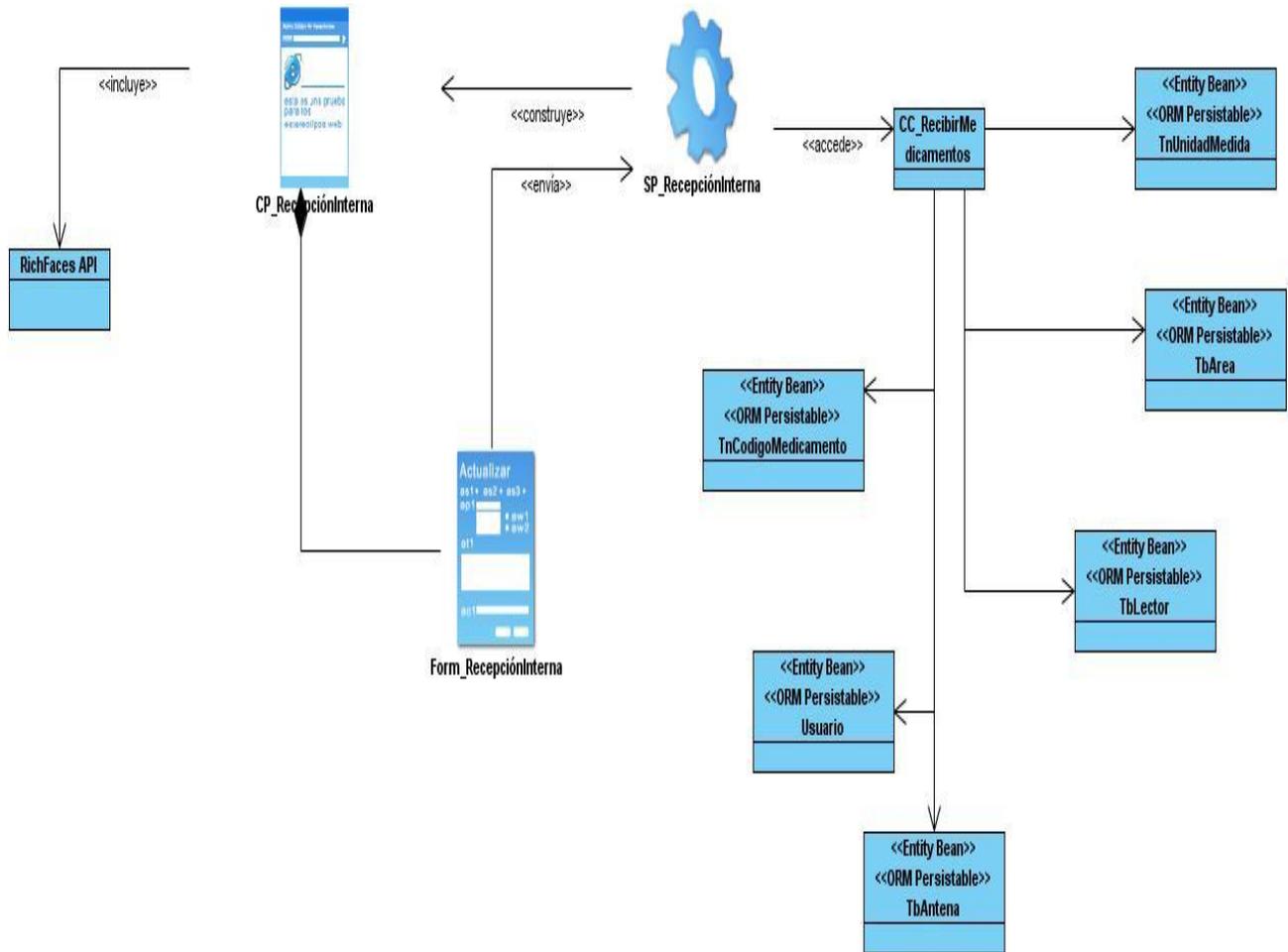


Figura 2. 11. Diagrama de clases del diseño registrar medicamentos internos

Fuente: Elaboración propia

La Figura 2.12 muestra el diagrama de clases del diseño perteneciente al proceso administrar medicamentos a un paciente donde el actor busca el paciente en el sistema y luego especifica la cantidad de medicamentos que se le aplicará.

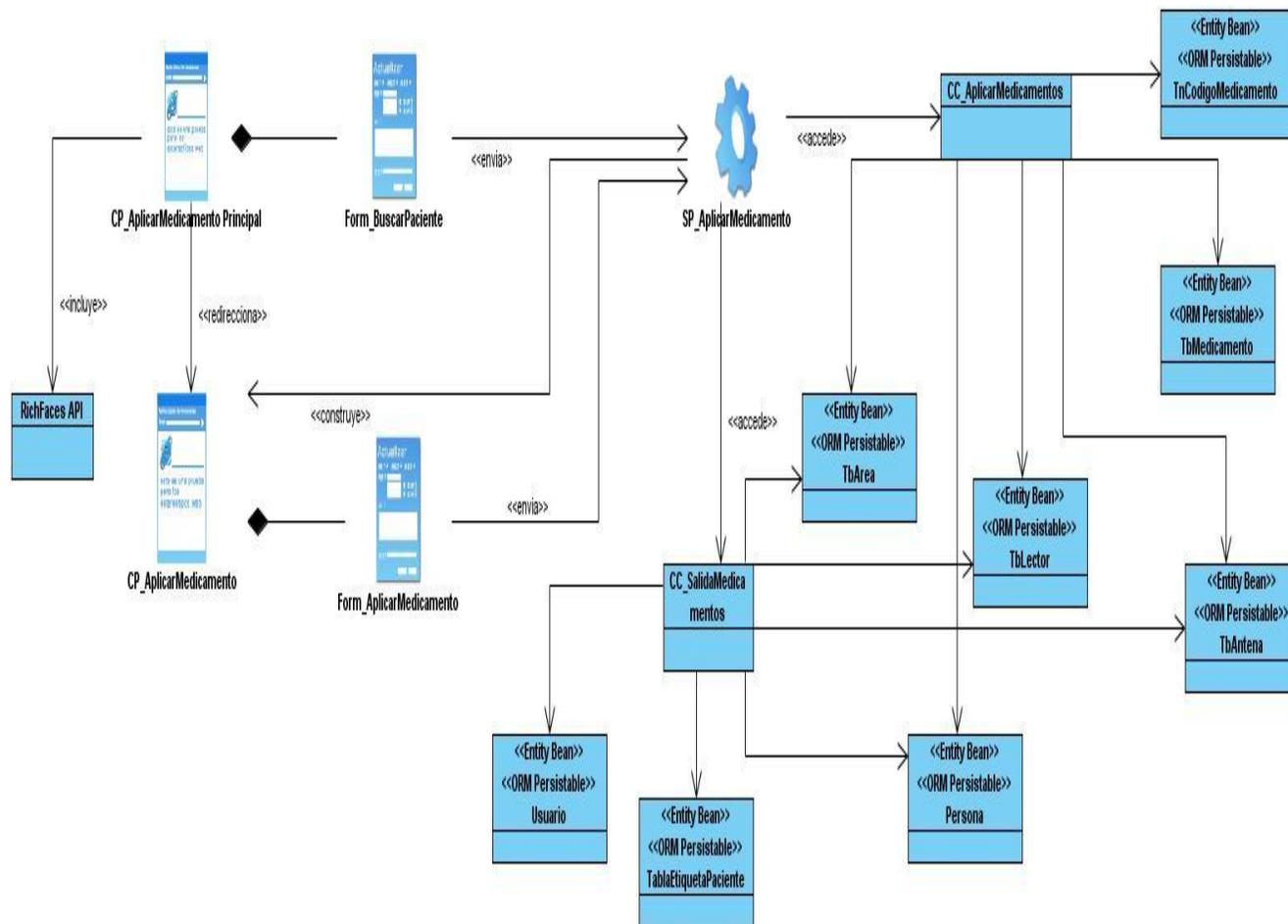


Figura 2. 12. Diagrama de clases del diseño administrar medicamento a un paciente

Fuente: Elaboración propia

La Figura 2.13 muestra el diagrama de clases del diseño perteneciente al proceso monitorizar medicamentos por área, donde el actor hace un reconocimiento del área para verificar los medicamentos que se encuentran en la misma y determinar si deben o no encontrarse en esa área.

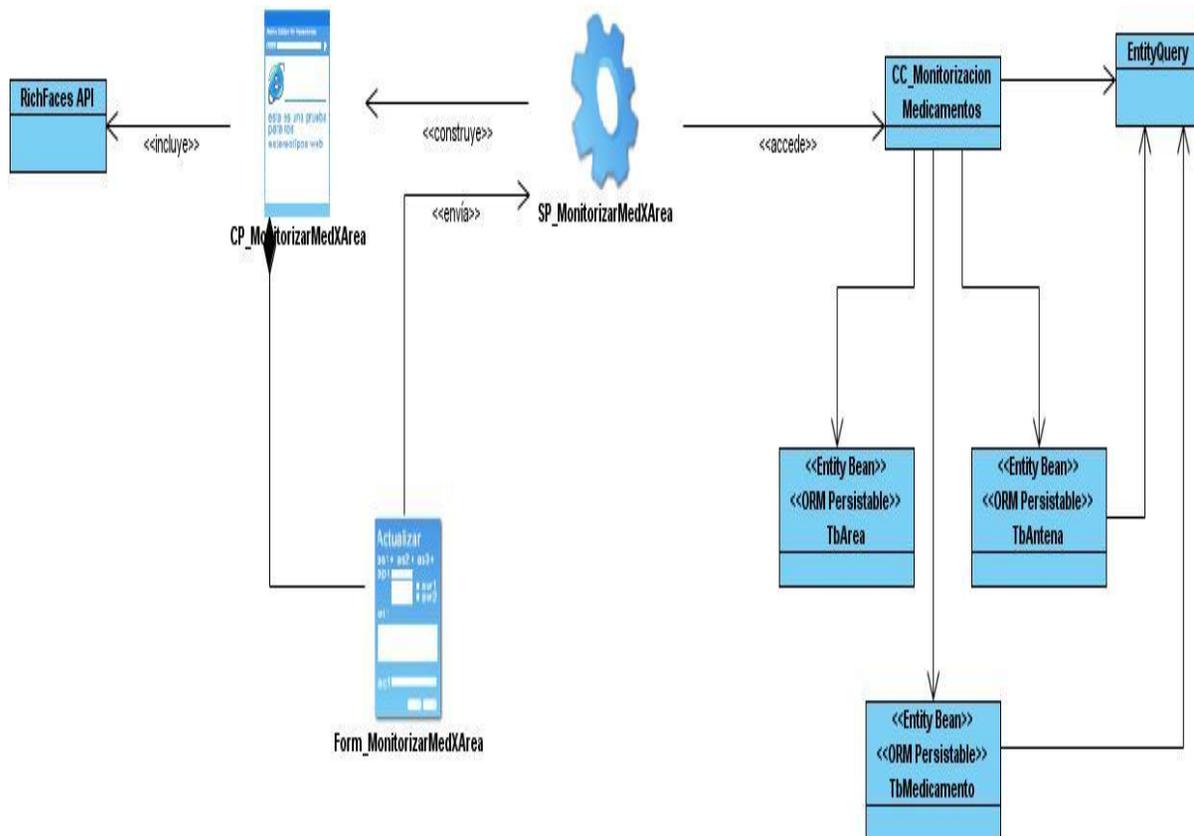


Figura 2. 13. Diagrama de clases del diseño monitorizar medicamentos por área

Fuente: Elaboración propia

Para la realización de las clases se hace uso de UML para el modelado de aplicaciones como se describen a continuación.

Descripción de las clases del diseño

Clase CP_Client Page

Representan el acceso del usuario al sistema, le permiten interactuar con la aplicación y realizar peticiones. Cada página tiene asociados sus propios formularios. Son páginas web que mezclan datos, presentación y lógica. Pueden ser construidas por una sola página de servidor. A continuación la Figura 2.14 muestra el estereotipo web perteneciente a una clase cliente:



Figura 2. 14. Estereotipo web para una página cliente

Fuente: Elaboración propia

La clase CP_RecepciónDeMedicamentos es la encargada de representar la interacción del usuario con la interfaz Recepción de Medicamentos, y de mostrarle los formularios que necesita llenar.

Clase SP_Server Page

Son páginas web que contienen el código que se ejecuta del lado del servidor, son las encargadas de procesar las peticiones del usuario y de realizar los cambios de ser necesarios en la vista o el modelo. A continuación la Figura 2.15 muestra el estereotipo web perteneciente a una clase servidora:



Figura 2. 15. Estereotipo web para una página servidor

Fuente: Elaboración propia

La clase SP_Recepcionar, es la encargada de construir la interfaz con la que interactúa el usuario para realizar la recepción y valida los datos introducidos por este.

Formularios

Representan la colección de datos de entrada y de salida del sistema. Forman parte de una página cliente. No tienen operaciones. A continuación la Figura 2.16 muestra el estereotipo web perteneciente a una clase formulario:

Capítulo 3. Implementación del componente propuesto

A partir del diseño realizado en el capítulo anterior, en este se presenta el modelo de la implementación, teniendo en cuenta cada uno de los elementos que lo integran, en términos de componentes y clases. Se describe el Modelo de datos sobre el cual se va a realizar la implementación y el Diagrama de despliegue para representar los nodos en que se divide el componente.

3.1 Modelo de datos

El Modelo de datos permite describir la estructura de la Base de datos, en cuanto a los tipos de datos que incluyen, y las relaciones entre las diferentes tablas que la componen. Es una representación abstracta de la forma de interactuar de los elementos de una realidad dada. Incluye las restricciones necesarias para que los datos reflejen la realidad deseada, y las posibles operaciones sobre estos datos. En la Figura 3.1 se representa el modelo definido para el componente deseado, el cual es un modelo de datos relacional.

El nombre de relacional procede del hecho de que cada registro de la base de datos contiene información relacionada con un tema específico. Los datos de dos clases de información (por ejemplo libros y autores) pueden ser manipulados como una única entidad basada en los valores de los datos relacionales. Todos los datos se tratan en tablas. Cada una de estas almacena información sobre un tema y dispone de columnas (campos) que contiene los tipos de información sobre ese tema y de filas (tuplas) que describen, los atributos de una única instancia del tema. Las tablas interactúan mediante relaciones (63).

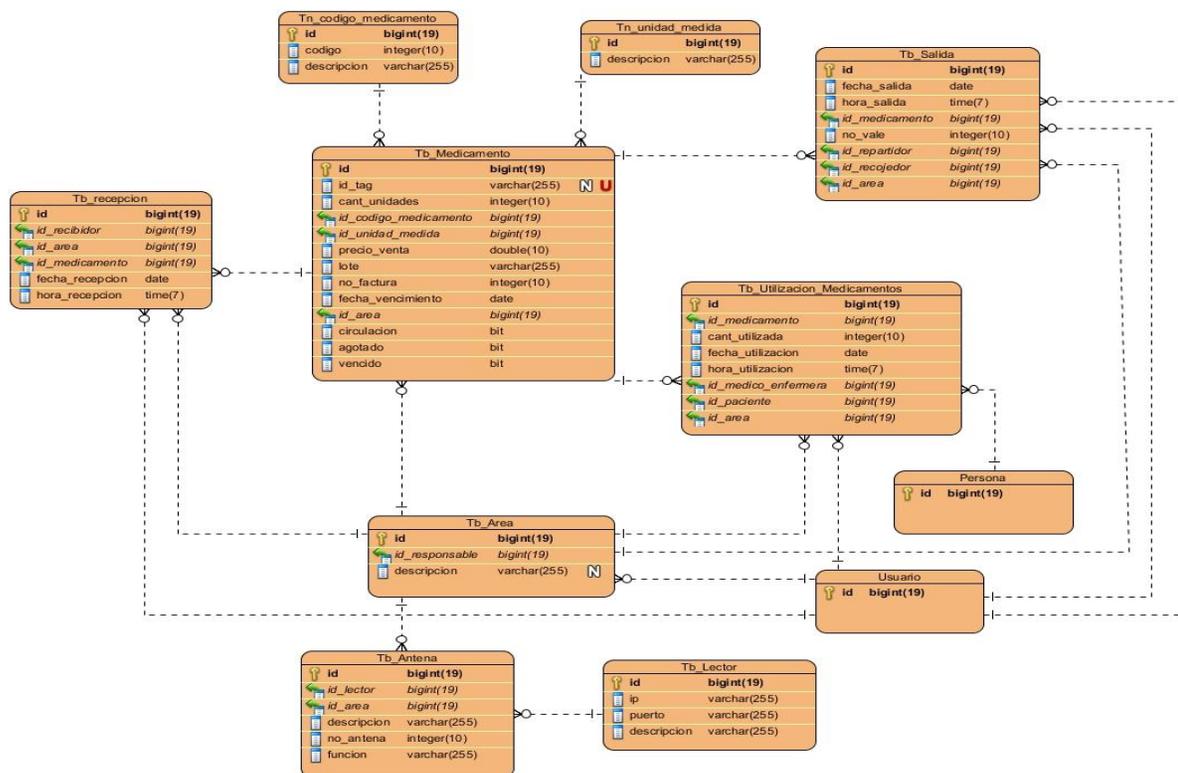


Figura 3. 1. Modelo de datos relacional

Fuente: Elaboración propia

Descripción de las tablas de la base de datos

A continuación en la Tabla 3.1 se describen las entidades más representativas del modelo anterior.

Tabla 3. 1. Descripción de las principales entidades del Modelo de datos

Fuente: Elaboración propia

Entidades del Modelo de datos	Descripción
Tb_medicamento	Entidad que recoge toda la información de los medicamentos: la cantidad, el precio de venta, el lote, la fecha en que vence, el estado en que se encuentra (circulación, agotado o vencido), el número de la factura en la que se recibió. Contiene un identificador único para ese medicamento y los identificadores de las relaciones con las tablas Tn_codigo_medimento, Tn_unidad_medida y

	Tb_Area.
Tb_recepción	Esta entidad almacena toda la información relacionada con una recepción de medicamentos: la fecha en que se realizó, la hora, y los identificadores de las relaciones con las tablas Usuario, Tb_Area y Tb_medicamento para definir la persona que recibió la entrega en que área y que recibió.
Tb_Salida	Esta entidad recoge la información sobre la entrega de los medicamentos a los centros de costo: la fecha de la entrega, la hora, y el número del vale de salida. Contiene los identificadores de su relación con las tablas Tb_Area, Tb_medicamento y Usuario para definir a que área se realizó la entrega, de qué medicamentos, quién los entrego y quién los recibió.
Tb_Utilizacion_Medicamentos	Esta entidad recoge toda la información sobre la utilización de un medicamento: la cantidad utilizada, en qué fecha y hora. Contiene los identificadores de las tablas Tb_Area, Tb_medicamento, Persona y Usuario para definir en qué área se utilizó, cuál fue el medicamento, que médico o enfermera lo aplicó y a qué paciente.

3.2 Diagrama de despliegue

El Diagrama de despliegue describe la distribución física de los distintos nodos que componen el componente, y la relación que se establece entre ellos. Estos nodos representan los elementos de software que entran en la realización del componente y se relacionan mediante soportes bidireccionales. En la Figura 3.2 se muestra el Diagrama de despliegue perteneciente al componente propuesto.

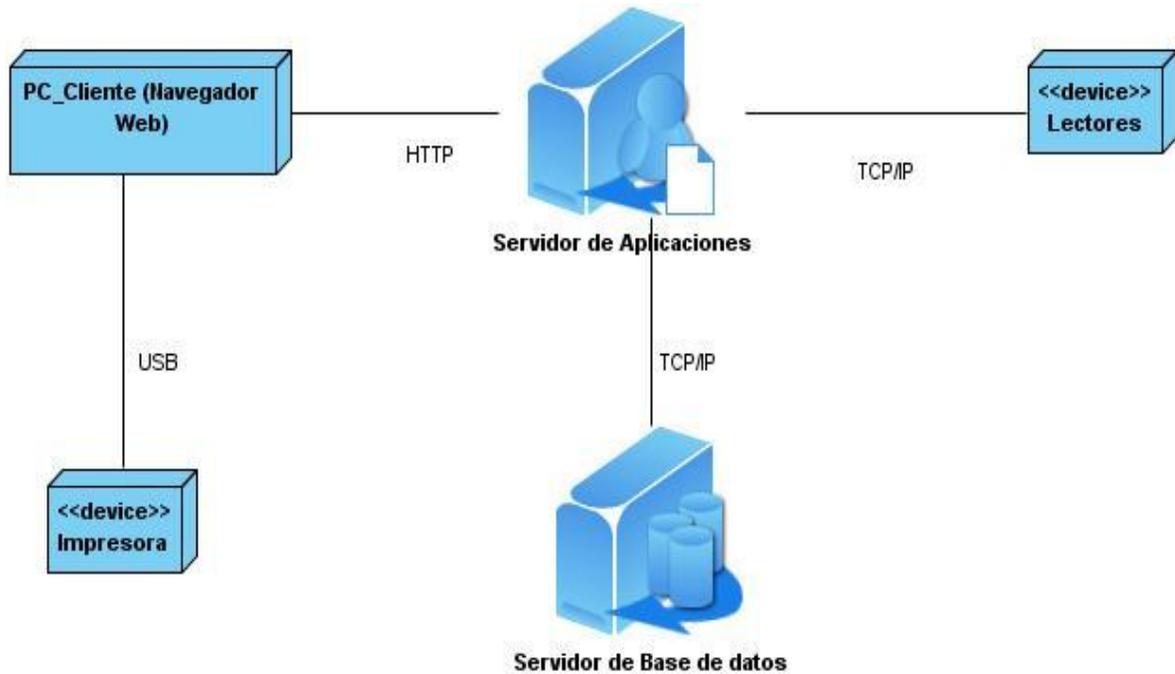


Figura 3. 2. Diagrama de despliegue

Fuente: Elaboración propia

Descripción de los componentes del diagrama de despliegue

PC Cliente: ordenador desde donde los usuarios acceden a la aplicación para utilizar sus servicios. Puede ser un usuario normal u otra aplicación.

Servidor de Base de Datos: servidor donde se encuentra alojada la base de datos que contiene toda la información necesitada por el sistema para brindar sus servicios.

Servidor de Aplicaciones: servidor que contiene las aplicaciones y servicios RFID del sistema.

HTTP: es el protocolo que se utiliza para realizar la conexión entre la PC Cliente y el servidor de aplicaciones y servicios web.

TCP/IP: protocolo que se utiliza para realizar la conexión entre el servidor de la base de datos y el servidor web.

3.3 Gestión de la trazabilidad en el componente

Cada una de las funcionalidades del componente va dejando una traza de la ocurrencia de sus eventos, permitiendo gestionar la trazabilidad de un medicamento en cualquier momento, desde la fecha de recepción hasta la actual, ya sea el día que fueron registrados, distribuidos o administrados a un paciente, quien realizo esta operaciones, en que área se realizaron las mismas, que cantidad de medicamentos interactuaron con que paciente, entre otros datos. Toda la información referente a la trazabilidad de los medicamentos, se muestra en el componente en forma de reportes, como se evidencia en el Anexo 6, capaces de ser exportados o impresos en caso de que se requiera obtener copias tangibles de los estos.

Análisis de equipos y costos

Para lograr una buena comunicación entre los componentes RFID del sistema es necesario que todos soporten el mismo protocolo y se comuniquen en la misma frecuencia. Con este objetivo y teniendo en cuenta las cualidades necesarias para cada uno de ellos, en cuanto a distancia y costo, se realiza un estudio para seleccionar los más indicados. En el caso de las etiquetas, estas no serán analizadas, debido a que el etiquetado de los medicamentos no es objetivo de la solución propuesta, y solamente será necesario tenerlas en cuenta a la hora de seleccionar el proveedor, de forma tal que se correspondan con el resto de la tecnología.

Se determinó como propuesta los siguientes equipos:

- Lector 216010 Gen2 de largo alcance

Tabla 3. 2. Lector 216010 Gen2 de largo alcance

Fuente: Gaorfid (64)

Lector 216010 Gen2	Características
Frecuencia de operación	860 MHz a 960 MHz

Protocolos	EPC Class1 Gen1 and Gen2
Compatibilidad antenas	Hasta 12 dBi
Rango Lectura/Escritura	Hasta 15m
Tamaño	460 x 460 x 50 mm
Compatibilidad antenas	Hasta 12 dBi
Precio	\$980

- Lector USB RFID UHF de escritorio

Tabla 3. 3. Lector USB RFID UHF de escritorio

Fuente: Dhgate (65)

Lector USB RFID UHF	Características
Frecuencia de operación	860 MHz a 960 MHz
Protocolos	ISO18000-6B, ISO18000-6C(EPC C1G2)
Compatibilidad antenas	Hasta 10 dBi
Rango Lectura/Escritura	Hasta 200 mm

Tamaño	15 x 10 x 6.0 cm
Precio	\$252.11

- Antena Alien Circular ALR-9611-CR

Tabla 3. 4. Antena Alien Circular ALR-9611-CR

Fuente: Barmax (66)

Antena Alien Circular ALR-9611-CR	Características
Frecuencia de operación	890 MHz a 940 MHz
Polarización	Circular
Ganancia	7 dBi
Tamaño	240*240*30mm
Precio	\$295

3.4 Integración del componente con el xavia HIS

El componente RFID es el resultado de una necesidad del xavia HIS, ha sido desarrollado con el objetivo de mejorar debilidades en determinados procesos de este sistema. Por esto y para evitar la ocurrencia de

errores o reprocesos en el trabajo con los mismos datos es necesario integrar ambas aplicaciones, minimizando así los costes y mejorando la eficiencia lo cual contribuye a elevar el valor de la información. Al integrar el componente este establece una estrecha relación con los módulos existentes en el sistema especialmente con el de Configuración y el de Monitorización. El primero le aporta información necesaria al segundo y este le brinda servicios que son utilizados por componente implementado.

3.5 Estándares de codificación y tratamiento de excepciones y errores

Los estándares tiene como objetivo lograr la homogeneidad entre las aplicaciones que conformen un mismo sistema, en este trabajo específicamente se ha utilizado el *Stanford University Network* (SUN Microsystems) desarrollado para el lenguaje Java y haciendo uso de la Notación Camello. Todo esto se ha tenido en cuenta para lograr la uniformidad a la hora de desarrollar el componente deseado, como se muestra a continuación.

Estándares de codificación

En el desarrollo del componente se utilizan estándares para definir:

El idioma

Debe utilizarse el español, y no se deben acentuar las palabras.

Los comentarios

Todos los ficheros fuentes deben iniciar con un comentario en el que se lista el nombre de la clase, información de la versión, fecha y *copyright*.

Las líneas en blanco

Se debe usar una línea en blanco siempre:

- Entre las variables locales de un método y su primera sentencia.
- Entre las secciones de un método.
- Entre métodos.
- Antes de un comentario.

Se debe usar dos líneas en blanco siempre:

- Entre las definiciones de interfaces y clases.
- Entre las secciones de un fichero fuente.

Los espacios en blanco

Una palabra clave del lenguaje seguida por un paréntesis deben separarse por un espacio en blanco.

La longitud de líneas

Las líneas no deben tener más de 80 caracteres, porque de ser así no son bien manejadas por muchas herramientas y terminales.

Variables y constantes

- Todas las instancias y variables de clases o métodos empezarán con minúscula. Las palabras internas que los conformen, de ser compuestas, empezarán con mayúscula. Los nombres de las variables no deben comenzar con los caracteres signo de peso "\$" o subguión "_".
- Los nombres de las variables deben ser cortos pero significativos aún para un observador casual. Deben evitarse los nombres de un solo carácter, excepto para variables índices temporales.
- Los nombres de las variables declaradas como constante deben aparecer totalmente en mayúscula, separadas las palabras con un subguión "_".

Clases y métodos

- Los nombres de las clases deben ser sustantivos. De ser compuestos cada palabra que lo conforme comenzará con mayúscula. Se deben usar palabras completas y evitar acrónimos y abreviaturas.
- Los nombres de los métodos deben ser verbos. Cuando son compuestos, la primera palabra que lo conforme comenzará con minúscula y las siguientes con mayúscula.

Tratamiento de excepciones errores

Una excepción es una situación anómala o lo que es lo mismo el fracaso de una rutina, y requiere un tratamiento especial. No necesariamente tiene que ser un error, y si se logran controlar hacen que el sistema sea más confiable y robusto.

En el desarrollo del componente además de las excepciones predefinidas por el marco de trabajo, se definen todas aquellas que se cree puedan dar lugar al fallo de alguna rutina, con el objetivo de evitarlo.

Para el tratamiento de errores, se le informará al usuario mediante mensajes, en el momento de que ocurra la equivocación. Estos mensajes se emitirán en el caso de que el usuario llene mal los campos de datos o deje vacíos campos requeridos.

Para esto se hace uso de los bloques de código **try** y **catch**, para la detección del fallo se utiliza el try, y el catch se encarga de manejar las excepciones mediante los mensajes como se evidencia en la Figura 3.3:

```
try {
    lista_tag = new ConexionLLRP(ip, puerto)
                .ObtenerListaTagsAntena(no_antena);
} catch (Exception e) {
    facesMessages.clear();
    facesMessages.addFromResourceBundle("Conexión rechazada, verifique la conexión " +
    "con las direcciones IP:Puerto instaladas.");
}
```

Figura 3. 3. Segmento perteneciente al bloque de código try / catch

Fuente: Elaboración propia

Seguridad del componente

Debido a la gran importancia de la información manejada en el sistema es necesario establecer políticas para gestionar su seguridad. Como el componente desarrollado se integra al xavia HIS el cual ya implementa cierta cantidad de estos requerimientos, se hace uso de estos. Entre ellos se encuentran:

- El sistema mantendrá la seguridad a nivel de usuarios haciendo una diferenciación entre los mismos a través de la asignación de roles, de forma tal que cada rol solamente tendrá acceso y

visualizará las funcionalidades necesarias para su desempeño. Asegurando así el manejo de la información solamente se realizara por parte del personal autorizado.

- Las contraseñas solo podrán ser cambiadas por el propio usuario o por el administrador del sistema.
- Se registrarán trazas de la navegación de los usuarios dentro del sistema, almacenando toda esta información en la base de datos y brindando la posibilidad de recuperarla mediante las copias de respaldo o salvadas realizadas.

Con la realización de este capítulo:

- Se concluye la descripción del proceso de desarrollo del componente propuesto.
- Se especifican todos los detalles desde el modelo de datos y las pautas del diseño utilizadas hasta las condiciones necesarias para el despliegue del proyecto.

Beneficios esperados

Con la realización de este componente basado en la tecnología de Identificación por Radiofrecuencia para la gestión de la trazabilidad de los medicamentos se espera obtener los siguientes beneficios:

- Se aumentará la seguridad y el control de los medicamentos en el sistema xavia HIS con el propósito de evitar la falsificación o robo de los mismos.
- Se reducirá la posibilidad de ocurrencia de errores humanos en el manejo de la información y la gestión de los medicamentos, facilitando el trabajo del personal involucrado en los procesos del sistema.
- Se fortalecerá el sistema xavia HIS elevando su competitividad en el mercado internacional para su comercialización.

Conclusiones

Luego de culminada la investigación, se arribó a las siguientes conclusiones

- La tecnología RFID constituye a nivel mundial una solución eficaz para enfrentar los problemas actuales en el sector de la cadena de suministros de productos farmacéuticos.
- El estudio de los sistemas basados en la tecnología RFID desarrollados a nivel mundial para gestionar la trazabilidad de los medicamentos sirvió de base para comprender el alcance del proyecto propuesto y de guía para la realización de sus principales funcionalidades.
- A partir de entrevistas no estructuradas efectuadas al cliente, se definieron las principales funcionalidades, dando paso a la propuesta del sistema, obteniéndose el modelo necesario para su despliegue.
- Las tecnologías empleadas, basándose en la arquitectura y el patrón de diseño descrito, facilitaron la obtención de los artefactos necesarios con la calidad requerida.
- El componente desarrollado permite gestionar la trazabilidad de los medicamentos de manera segura así como agilizar la introducción de la información, aligerando el trabajo de todo el personal involucrado en los procesos informatizados. Por lo cual resulta una herramienta innovadora y potente.

Recomendaciones

Con el fin de contribuir al éxito en la continuidad de investigación se hace la siguiente recomendación:

- Continuar con el desarrollo del componente hasta abarcar el proceso de dispensación de medicamentos a los pacientes externos desde la farmacia.

Trabajos citados

1. Gutiérrez, Gibran. La Importancia Y El Impacto Del Desarrollo Tecnológico. [En línea] [Citado el: 17 de Agosto de 2013.] <http://gibraine.com/2005/08/la-importancia-y-el-impacto-del-desarrollo-tecnologico.html>.
2. Imbiomed. [En línea] [Citado el: 4 de Diciembre de 2012.] http://www.imbiomed.com/1/1/articulos.php?method=showDetail&id_articulo=38059&id_seccion=1923&id_ejemplar=3925&id_revista=119Fragmento.
3. *Magnitud del problema*. [En línea] [Citado el: 20 de noviembre de 2012.] <http://www.who.int/mediacentre/factsheets/fs275/es/index.html>.
4. Mtra. Florina Gatica Lara. UNAM- Facultad de Medicina. *Sistema de Información Hospitalaria*. [En línea] [Citado el: 20 de Enero de 2013.] <http://www.facmed.unam.mx/emc/computo/ssa/HIS/his.pdf> .
5. SIQWEB División Difusión y Comunicaciones. *Trazabilidad*. [En línea] [Citado el: 10 de Enero de 2013.] <http://www.sigweb.cl/biblioteca/Trazabilidad.pdf>.
6. Iacotelera. *El código 128 (GS1-128)*. [En línea] [Citado el: 11 de Enero de 2013.] http://codigos_de_barras.iacotelera.net/post/2010/02/07/el-codigo-128-gs1-128.
7. Tecnologías de identificación. [En línea] [Citado el: 10 de Diciembre de 2012.] <http://www.bdev.com.ar/rfid.html>.
8. García, Javier I. Portillo. *Tecnología de identificación por radiofrecuencia (RFID): aplicaciones en el ámbito de la salud*. [En línea] [Citado el: 1 de Octubre de 2012.] www.madrimasd.org/tic/Informes/Downloads_GetFile.aspx?id=7913.
9. [En línea] [Citado el: 6 de Diciembre de 2012.] http://webcache.googleusercontent.com/search?q=cache:Dla-W0AgpVIJ:www.ametic.es/CLI_AETIC/ftpportalweb/documentos/RFIDCOMPLETO.pdf+influencia+de+la+tecnolog%C3%ADa+RFID+en+los+procesos+de+autenticidad,+control+y+seguimiento+de+productos+farmac%C3%A9uticos.

10. Ref.8.

11. Ref.6.

12. García, Javier I. Portillo. Ref. 7.

13. [En línea] [Citado el: 10 de enero de 2013.] <http://www.rfidpoint.com/fundamentos/el-estandar-epc/>.

14. IberWave. *Iberwave Ingeniería Especialistas en RFID*. [En línea] [Citado el: 20 de Enero de 2013.] <http://www.iberwave.com/epcrfid.html>.

15. Bateman, Javier. *Diseño De Un Protocolo RFID Propietario Para Una Aplicación*. [En línea] [Citado el: 5 de Diciembre de 2012.] <http://ingenieriayuniversidad.javeriana.edu.co/Vol13nr2Protocolo.pdf>.

16. Bateman, Javier. *Ref. 12*.

17. Bateman, Javier. *Ref. 12*.

18. Bateman, Javier. *Ref. 12*.

19. Bateman, Javier. *Ref. 12*.

20. msdn. *Introducción al proveedor LLRP*. [En línea] [Citado el: 10 de Diciembre de 2012.] <http://msdn.microsoft.com/es-es/library/bb970584.aspx>.

21. Dipole Soluciones de Trazabilidad y RFID. [En línea] [Citado el: 10 de Diciembre de 2012.] <http://www.dipolerfid.es/Productos/Lectores-RFID/Protocolo-LLRP.aspx>.

22. *Ref. 18*.

23. *Ref. 18*.

24. TrazaMed Sistema de Trazabilidad de Medicamentos. *Trazabilidad de Medicamentos*. [En línea] [Citado el: 7 de Enero de 2013.] <http://www.trazamed.com/index.php/trazabilidad-de-medicamentos>.

25. FARMAtrack. *QUE ES FARMATRACK*. [En línea] [Citado el: 20 de Enero de 2013.] <http://www.farmatrack.com.ar/que-es-farmatrack.php>.

26. RFID-LATINO.com. *Lanzaron en Argentina un nuevo sistema de trazabilidad de medicamentos* . [En línea] [Citado el: 20 de Enero de 2013.] <http://www.rfid-spain.com/articulo/69915/rfid/farmaceutica/lanzaron-en-argentina-un-nuevo-sistema-de-traz>.
27. TodoSoft Uruguay. *TS-RFID:: Trazabilidad de Medicamentos*. [En línea] [Citado el: 21 de Enero de 2013.]
http://www.todosoft.com.uy/sitio/index.php?option=com_content&view=article&id=108&Itemid=100036.
28. [En línea] [Citado el: 22 de Enero de 2013.] <http://www.slideshare.net/RFIDPOINT/trazabilidad-de-medicamentos-con-tecnologia-rfid>.
29. [En línea] [Citado el: 20 de Enero de 2013.] <http://www.baltimoresun.com/health/maryland-health/bs-hs-rfid-for-crash-carts-20120727,0,4851936.story>.
30. [En línea] [Citado el: 20 de Enero de 2013.] <http://www.grupov10.com/sga.html>.
31. [En línea] [Citado el: 20 de Enero de 2013.] www.vates.com/cmmi/que-es-cmmi.html.
32. *Ref. 33.*
33. [En línea] [Citado el: 21 de Enero de 2013.] <http://searchsoa.techtarget.com/definition/Java-virtual-machine>.
34. [En línea] [Citado el: 20 de Enero de 2013.] <http://www.jatun.com/web/company/training/javaee5>.
35. [En línea] [Citado el: 22 de Enero de 2013.] Java Enterprise Edition
<http://www.sicuma.uma.es/sicuma/independientes/argentina08/Badaracco/j2ee.htm>.
36. *Ídem 37.*
37. Reyes, Alexeis Joel Ochoa. *Vista de Análisis usando la Técnica de Agrupamiento para el Sistema Integral para la Atención Primaria de Salud*. Habana : s.n.
38. *RichFaces Live Demo*. [En línea] [Citado el: 22 de Enero de 2013.]
<http://livedemo.exadel.com/richfaces-demo/>.

39. [En línea] [Citado el: 25 de Enero de 2013.] <http://blogs.antartec.com/desarrolloweb/2008/12/facelets-y-jsf-uso-de-templates/>.
40. *Qué es XHTML.* [En línea] [Citado el: 25 de Enero de 2013.] <http://www.webtaller.com/construccion/lenguajes/html/lecciones/que-es-xhtml.php> .
41. *Java EE. Oracle Technology Network.* [En línea] [Citado el: 26 de Enero de 2013.] <http://www.oracle.com/technetwork/java/javaee/overview-140548.html>.
42. Krishna Srinivasan. *Introduction to Ajax4JSF.* *DZone.* [En línea] [Citado el: 18 de Enero de 2013.] <http://java.dzone.com/articles/do-not-publish-introduction-aj>.
43. *The Seam Framework - Next generation enterprise Java development.* [En línea] [Citado el: 19 de Enero de 2013.] <http://www.seamframework.org/> .
44. *About Hibernate.* [En línea] [Citado el: 26 de Enero de 2013.] <http://www.hibernate.org/about>.
45. Meza, Ing. Ronald Cuello. *JAVA PERSISTENCE API (jpa).* [En línea] [Citado el: 10 de Enero de 2013.] <http://www.slideshare.net/maxmouse/java-persistence-api-jpa> .
46. *Introducción a la tecnología EJB.* [En línea] <http://www.jtech.ua.es/j2ee/2003-2004/abierto-j2ee-2003-2004/ejb/sesion01-apuntes.htm>.
47. *EJBs Enterprise javaBeans.* [En línea] [Citado el: 15 de Enero de 2013.] <http://di002.edv.uniovi.es/~dflanvin/docencia/dasdi/teoria/Transparencias/13.%20Introducci%C3%B3n%20EJBs.pdf>.
48. *What is JBoss ?* [En línea] [Citado el: 29 de Enero de 2013.] <http://www.mastertheboss.com/jboss-configuration/what-is-jboss> .
49. [En línea] [Citado el: 21 de Enero de 2013.] <http://resumenuml.bloggratis.es/> .
50. [En línea] [Citado el: 29 de Enero de 2013.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.

51. *Eclipse*. [En línea] [Citado el: 29 de Enero de 2013.] <http://www.infor.uva.es/DesignSmells/index.php/herramientas-analizadas/eclipse.html>.
52. *Emuladores*. [En línea] [Citado el: 26 de Enero de 2013.] <https://sites.google.com/site/portalfid/emuladores> .
53. Pedraza, Emmanuel Alejandro Castañeda. UNIVERSIDAD TECNOLÓGICA EMILIANO ZAPATA. [En línea] [Citado el: 27 de Enero de 2013.] <http://www.utez.edu.mx/tesinas-utez/datic/ing/10464.pdf>.
54. Introducción a PostgreSQL. [En línea] [Citado el: 21 de Enero de 2013.] http://www.postgresql.org.es/sobre_postgresql .
55. *Ídem 54*.
56. PgAdmin III. [En línea] [Citado el: 18 de Enero de 2013.] http://www.guia-ubuntu.com/index.php?title=PgAdmin_III.
57. Rodríguez, Yenier Vega. *Componente web para el módulo Salud Materno-Infantil del Sistema Integral para la Atención Primaria de Salud*. La Habana : s.n., 2011.
58. *Idem Ref 56*.
59. Reynoso, Billy. Scribd. *Architect Academy: Seminario de Arquitectura de Software*. [En línea] [Citado el: 23 de febrero de 2013.] <http://es.scribd.com/doc/101401231/050608-Architect-Academy-Webcast-1>.
60. Casanovas, Josep. Usabilidad y arquitectura del software. *Desarrollo Web*. [En línea] [Citado el: 20 de febrero de 2013.] <http://www.desarrolloweb.com/articulos/1622.php>.
61. —. *Ref.58*.
62. Scribd. *PATRONES ARQUITECTÓNICOS PARA SISTEMAS INTERACTIVOS*. [En línea] <http://es.scribd.com/doc/34279909/Puntos-Examen-Patrones-Arquitectonicos-Secc01>.
63. Rivera, Fray León Osorio. *Base de datos relacionales*. s.l. : ITM. 9588351421, 9789588351421.

64. Gaorfid. [En línea] [Citado el: 24 de Enero de 2013.] http://www.gaorfid.com/index.php?main_page=product_info&cPath=128&products_id=831.

65. [En línea] [Citado el: 24 de Enero de 2013.] <http://es.dhgate.com/usb-rfid-uhf-desktop-reader-writer-free-sample/p-ff8080813dc3508a013de34e140c761e.html#s1-8-1>.

66. Barmax. [En línea] [Citado el: 25 de Enero de 2013.] <http://www.barmax.com/proddetail.asp?prod=ATS6528>.

Bibliografía

About Hibernate. [En línea] [Citado el: 26 de Enero de 2013.] <http://www.hibernate.org/about>.

Barmax. [En línea] [Citado el: 25 de Enero de 2013.]
<http://www.barmax.com/proddetail.asp?prod=ATS6528>.

Bateman, Javier. Diseño De Un Protocolo RFID Propietario Para Una Aplicación. [En línea] [Citado el: 5 de Diciembre de 2012.] <http://ingenieriayuniversidad.javeriana.edu.co/Vol13nr2Protocolo.pdf>

. Casanovas, Josep. Usabilidad y arquitectura del software. Desarrollo Web. [En línea] [Citado el: 20 de febrero de 2013.] <http://www.desarrolloweb.com/articulos/1622.php>.

Dipole Soluciones de Trazabilidad y RFID. [En línea] [Citado el: 10 de Diciembre de 2012.]
<http://www.dipolerfid.es/Productos/Lectores-RFID/Protocolo-LLRP.aspx>.

[En línea] [Citado el: 6 de Diciembre de 2012.]
http://webcache.googleusercontent.com/search?q=cache:Dla-W0AgpVIJ:www.ametic.es/CLI_AETIC/ftpportalweb/documentos/RFIDCOMPLETO.pdf+influencia+de+la+tecnolog%C3%ADa+RFID+en+los+procesos+de+autenticidad,+control+y+seguimiento+de+productos+farmac%C3%A9uticos.

[En línea] [Citado el: 20 de Enero de 2013.] <http://www.baltimoresun.com/health/maryland-health/bs-hs-rfid-for-crash-carts-20120727,0,4851936.story>.

[En línea] [Citado el: 25 de Enero de 2013.] <http://blogs.antartec.com/desarrolloweb/2008/12/facelets-y-jsf-uso-de-templates/>.

Eclipse. [En línea] [Citado el: 29 de Enero de 2013.]
<http://www.infor.uva.es/DesignSmells/index.php/herramientas-analizadas/eclipse.html>.

[En línea] [Citado el: 24 de Enero de 2013.] <http://es.dhgate.com/usb-rfid-uhf-desktop-reader-writer-free-sample/p-ff8080813dc3508a013de34e140c761e.html#s1-8-1>.

[En línea] [Citado el: 29 de Enero de 2013.]
http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.

[En línea] [Citado el: 20 de Enero de 2013.] <http://www.grupov10.com/sga.html>.

[En línea] [Citado el: 20 de Enero de 2013.] <http://www.jatun.com/web/company/training/javaee5>.

EJBs Enterprise javaBeans. [En línea] [Citado el: 15 de Enero de 2013.]

<http://di002.edv.uniovi.es/~dflanvin/docencia/dasdi/teoria/Transparencias/13.%20Introducci%C3%B3n%20EJBs.pdf>.

[En línea] [Citado el: 22 de Enero de 2013.] Java Enterprise Edition

<http://www.sicuma.uma.es/sicuma/independientes/argentina08/Badaracco/j2ee.htm>.

Emuladores. [En línea] [Citado el: 26 de Enero de 2013.] <https://sites.google.com/site/portalrfid/emuladores>

[En línea] [Citado el: 10 de Enero de 2013.] <http://www.rfidpoint.com/fundamentos/el-estandar-epc/>.

[En línea] [Citado el: 21 de Enero de 2013.] <http://searchsoa.techtarget.com/definition/Java-virtual-machine>.

[En línea] [Citado el: 22 de Enero de 2013.] <http://www.slideshare.net/RFIDPOINT/trazabilidad-de-medicamentos-con-tecnologia-rfid>.

[En línea] [Citado el: 20 de Enero de 2013.] www.vates.com/cmami/que-es-cmami.html.

FARMAtrack. QUE ES FARMATRACK. [En línea] [Citado el: 20 de Enero de 2013.]

<http://www.farmatrack.com.ar/que-es-farmatrack.php>.

Gaorfid. [En línea] [Citado el: 24 de Enero de 2013.]

http://www.gaorfid.com/index.php?main_page=product_info&cPath=128&products_id=831.

García, Javier I. Portillo. Tecnología de identificación por radiofrecuencia (RFID): aplicaciones en el ámbito de la salud. [En línea] [Citado el: 1 de Octubre de 2012.]

www.madrimasd.org/tic/Informes/Downloads_GetFile.aspx?id=7913.

Gutiérrez, Gibran. La Importancia Y El Impacto Del Desarrollo Tecnológico. [En línea] [Citado el: 17 de Agosto de 2013.]

<http://gibraine.com/2005/08/la-importancia-y-el-impacto-del-desarrollo-tecnologico.html>.

IberWave. Iberwave Ingenieria Especialistas en RFID. [En línea] [Citado el: 20 de Enero de 2013.]

<http://www.iberwave.com/epcrfid.html>.

Imbiomed. [En línea] [Citado el: 4 de Diciembre de 2012.]

http://www.imbiomed.com/1/1/articulos.php?method=showDetail&id_articulo=38059&id_seccion=1923&id_ejemplar=3925&id_revista=119Fragmento.

Introducción a la tecnología EJB. [En línea] <http://www.jtech.ua.es/j2ee/2003-2004/abierto-j2ee-2003-2004/ejb/sesion01-apuntes.htm>.

Introducción a PostgreSQL. [En línea] [Citado el: 21 de Enero de 2013.]
http://www.postgresql.org.es/sobre_postgresql .

Java EE. Oracle Technology Network. [En línea] [Citado el: 26 de Enero de 2013.]
<http://www.oracle.com/technetwork/java/javaee/overview-140548.html>.

Krishna Srinivasan. Introduction to Ajax4JSF. DZone. [En línea] [Citado el: 18 de Enero de 2013.]
<http://java.dzone.com/articles/do-not-publish-introduction-aj>.

Iacoctelera. El código 128 (GS1-128). [En línea] [Citado el: 11 de Enero de 2013.]
http://codigos_de_barras.iacoctelera.net/post/2010/02/07/el-codigo-128-gs1-128.

Magnitud del problema. [En línea] [Citado el: 20 de noviembre de 2012.]
<http://www.who.int/mediacentre/factsheets/fs275/es/index.html>.

Meza, Ing. Ronald Cuello. JAVA PERSISTENCE API (jpa). [En línea] [Citado el: 10 de Enero de 2013.]
<http://www.slideshare.net/maxmouse/java-persistence-api-jpa> .

msdn. Introducción al proveedor LLRP. [En línea] [Citado el: 10 de Diciembre de 2012.]
<http://msdn.microsoft.com/es-es/library/bb970584.aspx>.

Mtra. Florina Gatica Lara. UNAM- Facultad de Medicina. Sistema de Información Hospitalaria. [En línea] [Citado el: 20 de Enero de 2013.] <http://www.facmed.unam.mx/emc/computo/ssa/HIS/his.pdf> .

Pedraza, Emmanuel Alejandro Castañeda. UNIVERSIDAD TECNOLÓGICA EMILIANO ZAPATA. [En línea] [Citado el: 27 de Enero de 2013.] <http://www.utez.edu.mx/tesinas-utez/datic/ing/10464.pdf>.

PgAdmin III. [En línea] [Citado el: 18 de Enero de 2013.] http://www.guia-ubuntu.com/index.php?title=PgAdmin_III.

Qué es XHTML. [En línea] [Citado el: 25 de Enero de 2013.]
<http://www.webtaller.com/construccion/lenguajes/html/lecciones/que-es-xhtml.php> .

Reyes, Alexeis Joel Ochoa. Vista de Análisis usando la Técnica de Agrupamiento para el Sistema Integral para la Atención Primaria de Salud. Habana : s.n.

Reynoso, Billy. Scribd. Architect Academy: Seminario de Arquitectura de Software. [En línea] [Citado el: 23 de febrero de 2013.] <http://es.scribd.com/doc/101401231/050608-Architect-Academy-Webcast-1>.

RFID-LATINO.com. Lanzaron en Argentina un nuevo sistema de trazabilidad de medicamentos . [En línea] [Citado el: 20 de Enero de 2013.] <http://www.rfid-spain.com/articulo/69915/rfid/farmaceutica/lanzaron-en-argentina-un-nuevo-sistema-de-traz>.

RichFaces Live Demo. [En línea] [Citado el: 22 de Enero de 2013.] <http://livedemo.exadel.com/richfaces-demo/>.

Rivera, Fray León Osorio. Base de datos relacionales. s.l. : ITM. 9588351421, 9789588351421.

Rodríguez, Yenier Vega. Componente web para el módulo Salud Materno-Infantil del Sistema Integral para la Atención Primaria de Salud. La Habana : s.n., 2011.

Scribd. PATRONES ARQUITECTÓNICOS PARA SISTEMAS INTERACTIVOS. [En línea] <http://es.scribd.com/doc/34279909/Puntos-Examen-Patrones-Arquitectonicos-Secc01>.

SIQWEB División Difusión y Comunicaciones. Trazabilidad. [En línea] [Citado el: 10 de Enero de 2013.] <http://www.sigweb.cl/biblioteca/Trazabilidad.pdf>.

Tecnologías de identificación. [En línea] [Citado el: 10 de Diciembre de 2012.] <http://www.bdev.com.ar/rfid.html>.

The Seam Framework - Next generation enterprise Java development. [En línea] [Citado el: 19 de Enero de 2013.] <http://www.seamframework.org/> .

TodoSoft Uruguay. TS-RFID:: Trazabilidad de Medicamentos. [En línea] [Citado el: 21 de Enero de 2013.] http://www.todosoft.com.uy/sitio/index.php?option=com_content&view=article&id=108&Itemid=100036.

TrazaMed Sistema de Trazabilidad de Medicamentos. Trazabilidad de Medicamentos. [En línea] [Citado el: 7 de Enero de 2013.] <http://www.trazamed.com/index.php/trazabilidad-de-medicamentos>.

What is JBoss ? [En línea] [Citado el: 29 de Enero de 2013.] <http://www.mastertheboss.com/jboss-configuration/what-is-jboss> .

Glosario de Términos

API (*Application Programming Interface*): conjunto de funciones y procedimientos que poseen algunas librerías con el objetivo de ser utilizadas por otro software como una capa de abstracción.

Aplicación Web: es una aplicación informática que los usuarios utilizan accediendo a un servidor Web a través de un navegador o browser. Permiten actualizar y mantener la información manipulada sin distribuir e instalar el software en miles de potenciales clientes.

Caso de Uso: descripción de un conjunto de secuencias de acciones, incluyendo variaciones, que un sistema lleva a cabo y que conduce a un resultado observable de interés para un actor determinado.

Centro de costo: áreas del hospital que pueden solicitar medicamentos a los almacenes de los mismos.

Componente: parte física y reemplazable de un sistema que proporciona la realización de un conjunto de interfaces.

Diagrama: representación gráfica de un conjunto de elementos y sus relaciones.

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

HTTP (*HyperText Transfer Protocol* o *Protocolo de Transferencia de Hipertexto*): protocolo usado en cada transacción de la Red Global Mundial.

Informatizar: proceso de aplicar s o equipos informáticos al tratamiento de la información.

JavaBeans: modelo de componentes para la construcción de aplicaciones en Java. Se usan para encapsular varios objetos en un único objeto.

JSP (*JavaServer Pages*): tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

Middleware: es un software que conecta los componentes de aplicaciones para que puedan intercambiar datos entre sí.

Multiplataforma: término usado para referirse a los programas, sistemas operativos, u otra clase de software, que puedan funcionar en varias plataformas.

Proyecto: esfuerzo de desarrollo para llevar un sistema a lo largo de un ciclo de vida.

Trazabilidad: asociación distinguible entre dos o más entidades lógicas como requisitos, elementos del sistema, verificaciones, o tareas.

Anexos

Anexo 1. Interfaz correspondiente al proceso de recepción de medicamentos externos.

Recepción de medicamentos externos Q Buscar...

Datos del medicamento a recibir

Medicamento: Cantidad por unidad: Unidad medida:

Precio venta: Fecha recepción: Fecha de vencimiento:

No. de factura: Lote:

Datos del area que recibe

Area:

Medicamentos encontrados

Cantidad	Medicamento	Unidad de medida
1	Dipirona	Tableta

Anexo 2: Interfaz correspondiente al proceso de entregar los medicamentos a los centros de costo.

Distribuir medicamentos Q Buscar...

Datos de la entrega

Area: Nombre responsable:

Datos del distribuidor

Area:

Listado de medicamentos

Lote	Tipo de medicamento	Cantidad	Unidad de medida
1	Paracetamol	1	Tableta

Anexo 3. Interfaz correspondiente al proceso de registrar medicamentos internos.

Recepcion de medicamentos internos Q Buscar..

Datos del medicamento a recibir

Fecha recepción
jul 15, 2013  Escanear

Medicamentos encontrados

Lote	Medicamento	Cantidad	Unidad medida	
1	Dipirona	1	Tableta	

Aceptar Cancelar

Anexo 4. Interfaces correspondiente al proceso de administrar medicamentos a un paciente.

Administrar medicamento Q Buscar..

Buscar paciente

Nombre: Área: ▼ Buscar

Listado de pacientes encontrados

Foto	Nombre	Primer apellido	Segundo apellido	
	Hector	Ramirez	Romero	

Cancelar

Administrar medicamento Q Buscar...

Datos generales del paciente



Nombre:	Hector	Cédula:	12345678978	Teléfono:	-
Primer apellido:	Ramirez	Fecha de nacimiento:	14/12/1942	Edad:	70 años
Segundo apellido:	Romero	Sexo:	Masculino		

Medicamento

Tipo de medicamento:

Cantidad disponible:

Unidad de medida:

Cantidad a utilizar:

Anexo 5. Interfaz correspondiente al proceso de Monitorización de medicamentos por áreas.

Monitorización de medicamentos por áreas Q Buscar...

Buscar paciente

Área

Listado de medicamentos

Lote	Tipo medicamento	Unidad medida	Área		
1	Dipirona	Tableta	Almacen Medicamentos	✓	🔍
1	Dipirona	Tableta	Almacen Medicamentos	✓	🔍
1	Dipirona	Tableta	Salon1	✗	🔍

Anexo 6. Interfaz correspondiente al proceso de reporte de trazabilidad de un medicamento.





alas HIS
SISTEMA DE INFORMACIÓN HOSPITALARIA

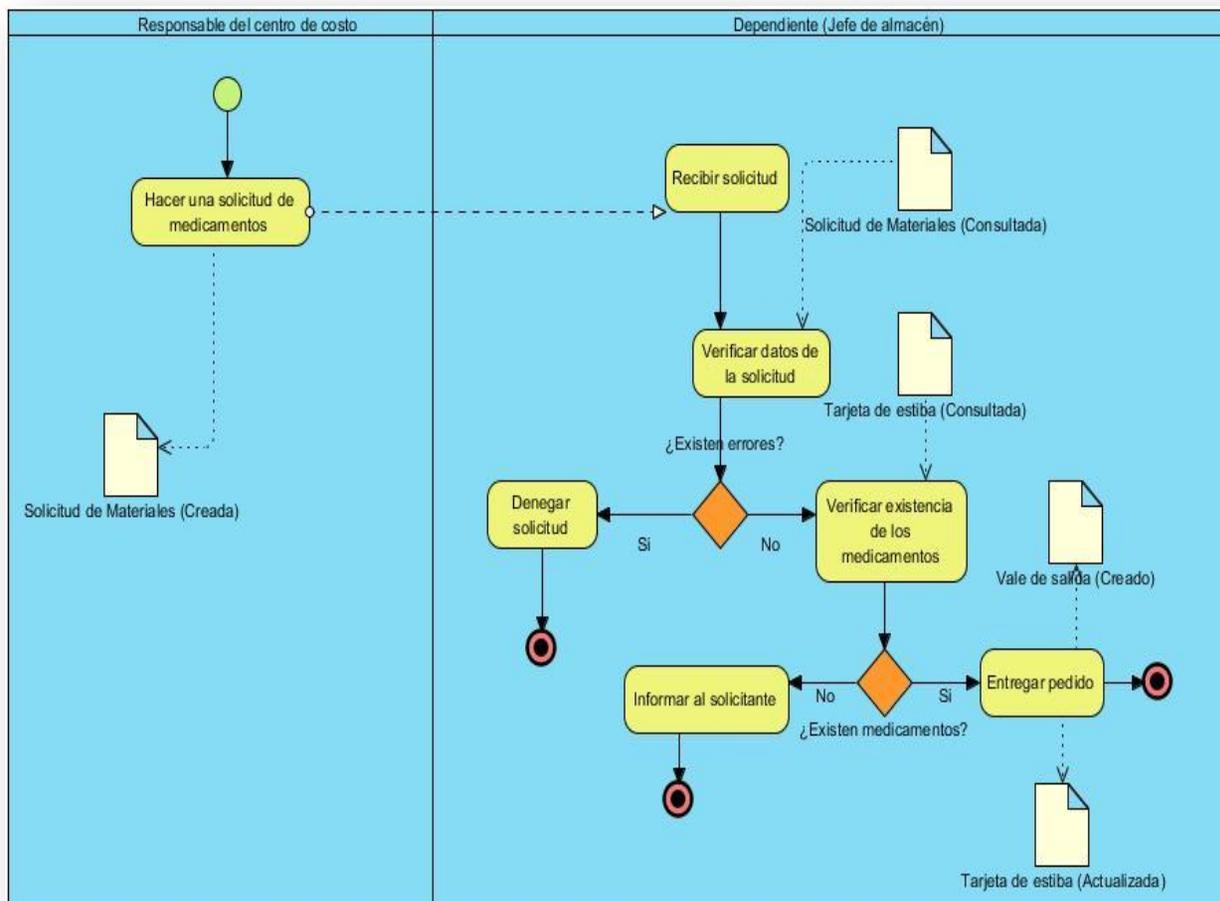
Identificación por Radiofrecuencia
Administrador Administrador
10/07/2013

Reporte de la trazabilidad de un medicamento

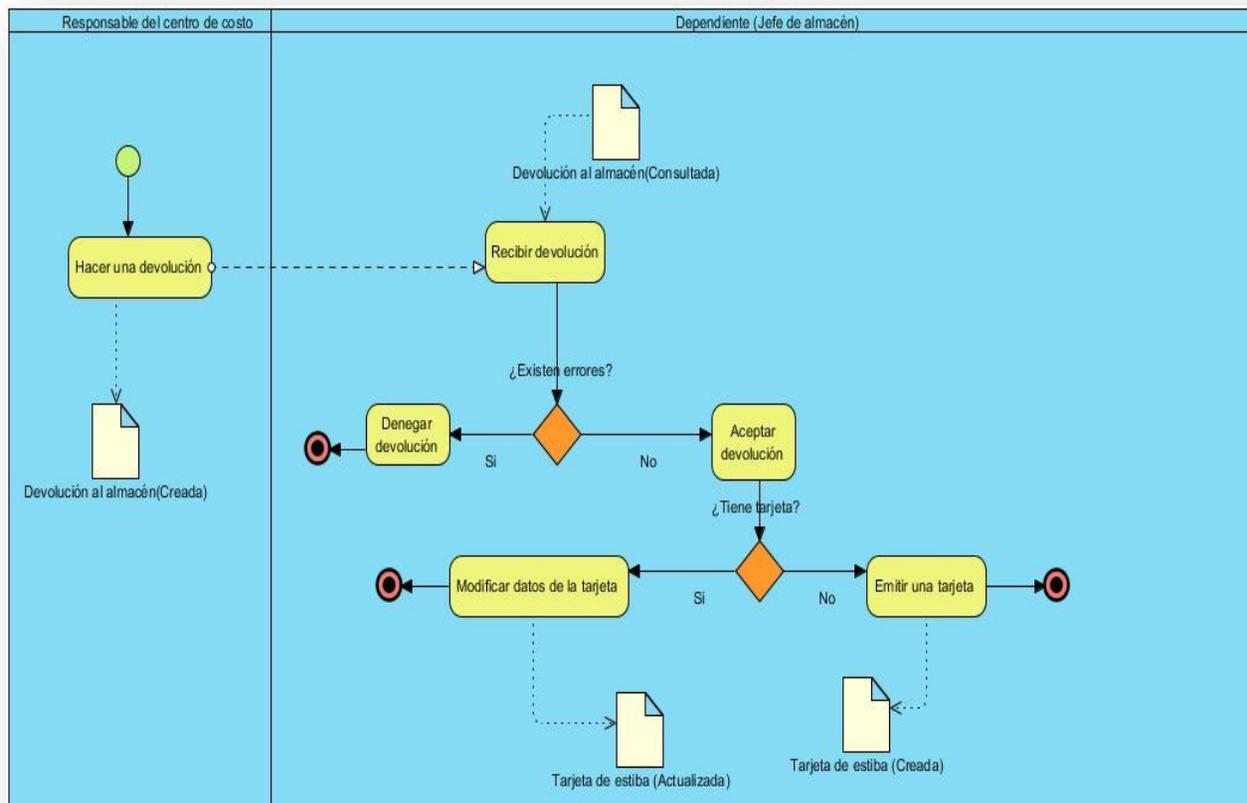
Listado de trazabilidad de un medicamento

Lote	Descripción	U/M	Acción	Cantidad Utilizada	Cantidad Restante	Entregado	Recibido por:	Paciente	Area	Fecha
1	Diasepan	Tableta	Recibido	0	10	--	Administrador	--	Almacen Medicamentos	01/05/2013
1	Diasepan	Tableta	Entregado	0	10	Administrador	Mario	--	Salon1	01/05/2013
1	Diasepan	Tableta	Administrado	1	9	Administrador	--	Julián	Salon1	01/05/2013
1	Diasepan	Tableta	Recibido	0	9	--	Administrador	--	Almacen Medicamentos	01/05/2013

Anexo 7. Diagrama de clases del negocio perteneciente al proceso distribución de medicamentos a un eslabón posterior



Anexo 8. Diagrama de clases del negocio perteneciente al proceso devolución de un producto a un eslabón anterior



Anexo 9. Diagrama de clases del negocio perteneciente al proceso administración del medicamento

