

Universidad de las Ciencias Informáticas
Facultad 2: "Telecomunicaciones y Seguridad Informática"



Título: "Sistema de Gestión de Inventarios utilizando la tecnología RFID".

Trabajo de Diploma para optar por el título de
Ingeniero Informático.

Autor(es): Esbietta Díaz Matienzo.

Reinier Pérez López

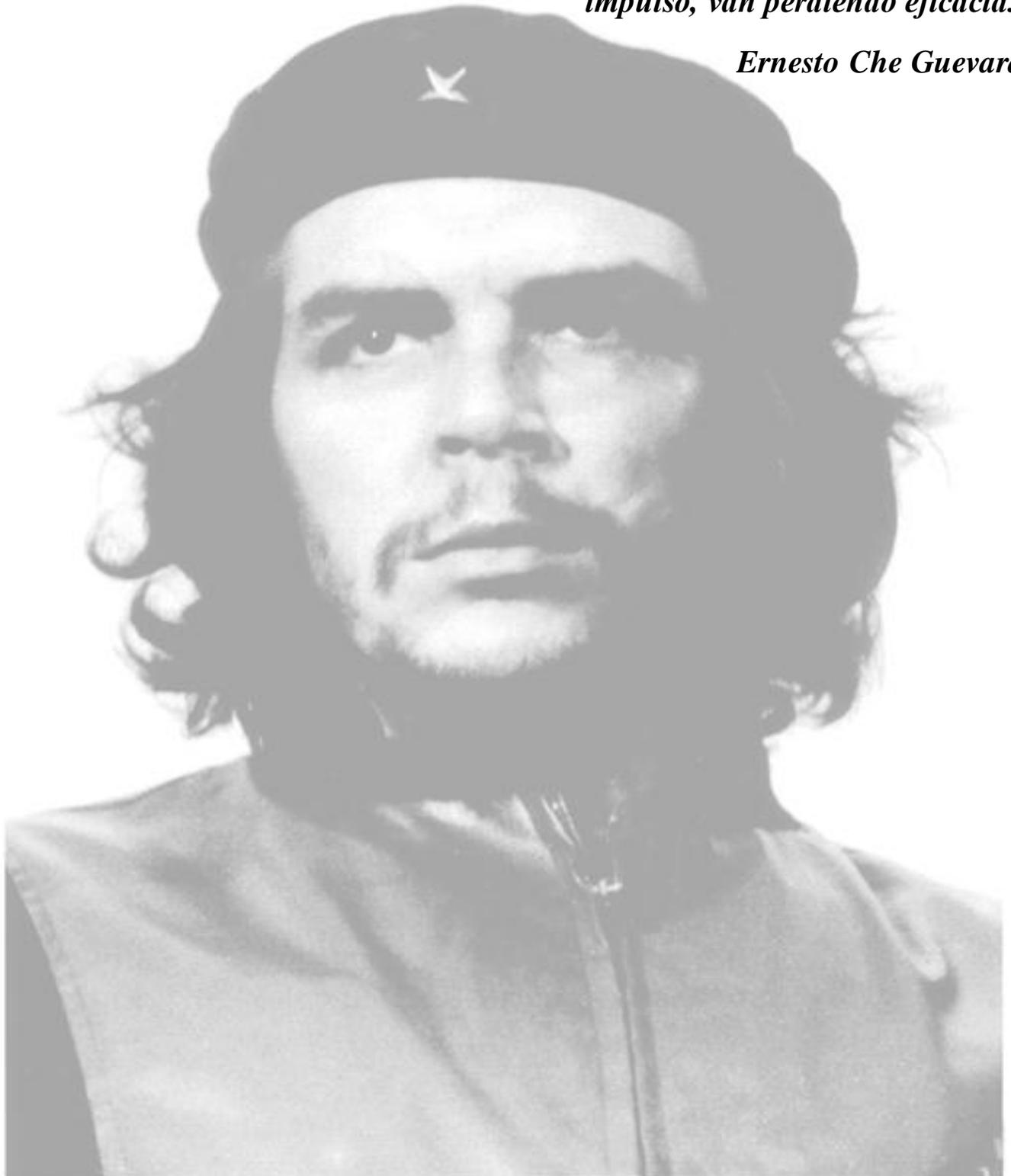
Tutor(es): Ing. Yeilin Pérez Martínez

Co-tutor: Ing. Wilfredo Rosales Romero

Junio del 2013

“Si no existe la organización, las ideas, después del primer momento de impulso, van perdiendo eficacia.”

Ernesto Che Guevara.



DECLARACIÓN DE AUTORÍA

Declaro que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Esbietta Díaz Matienzo

Firma del Autor

Reinier Pérez López

Firma del Autor

Ing. Yeilin Pérez Martínez

Firma del Tutor

Ing. Wilfredo Rosales Romero

Firma del Cotutor

AGRADECIMIENTOS

Esbietta

En los agradecimientos casi siempre se cometen injusticias, pues la memoria es a menudo traicionera. Pero aun sabiendo que no existirá una forma para agradecerles, hoy sin embargo, podemos afirmar sin lugar a dudas, que este trabajo no hubiese sido posible sin la ayuda de:

Mi madre que es lo más grande que tengo y que tanto me ha apoyado para lograr mis objetivos. A ella le debo la vida y espero retribuirle lo que me ha dado. Por ser madre, padre y amiga para mí, quererme, amarme sin medidas y estar siempre conmigo cuando más le he necesitado.

Mi familia que me ha apoyado en las buenas y las malas, mis más grande agradecimiento.

Mis amigas, Eilan, Lisandra, Susana, Yanisleidis, que aunque a muchas no las veré más, no las olvidare.

Reinier

Siempre hay personas que marcan nuestras vidas con solo pequeños detalles, por el solo hecho de que signifiquemos una parte importante de ellos. Hoy quiero agradecer a todas esas personas que se han preocupado por mí, porque mi sueño se haya hecho realidad, en especial a:

Mi mamá y mi papá, que han sido el regalo más grande que la vida me ha dado, gracias por todo lo que han hecho por mí, por preocuparse tanto por mi porvenir, por llorar cuando algo me va mal y reír cuando las cosas salen bien, a ellos les debo mi vida. Los quiero más que a mi propia vida mi mamita y mi papito, gracias por existir.

A mis abuelos que han sido mis padres también, gracias por preocuparse tanto por mí y quererme tanto.

A mis amigos de siempre, por apoyarme y estar conmigo durante estos años, gracias a todos.



DEDICATORIAS

Esbietta

Dedico este trabajo a la persona más importante de mi vida, mi madre María Elena por ser tan comprensiva, darme su apoyo en los momentos más difíciles y disfrutar conmigo las alegrías, por amarme sin condiciones y ser la mejor mami del mundo, estoy muy orgullosa de tenerla.

A mi padre y hermano que a pesar de no encontrarse entre nosotros, tenerlos en mis pensamientos siempre me ha dado la fuerza para ir hacia adelante y darme cuenta que si se puede.

A mi padrastro querido por apoyarme, quererme como una hija, cuidar a mi mami y darme lo mejor de él, siempre estaré agradecida.

Este trabajo es dedicado a ustedes por ser mi razón de ser.

Reinier

Existen muchas personas importantes en mi vida las cuales siempre me ha apoyado en los momentos más difíciles, a todas ellas, gracias por ese apoyo que me ha guiado a ganar confianza en lo que hago cada día, en especial a mis padres, que son los más grandes merecedores de todo mi agradecimiento y orgullo. Este trabajo se lo dedicado a todas esas personas que ocupan un lugar en mi corazón, les juro que no los defraudaré. Gracias por todo.

RESUMEN

La mayoría de las entidades que poseen productos para el desarrollo de ventas o para su consumo necesitan un sistema para el proceso de captura, análisis y consulta de la información de los activos en sus almacenes, para el control e identificación de los recursos. En Cuba se evidencia el registro de los activos de forma manual o mediante herramientas para controlar esta información, aunque no satisfacen todas las necesidades de la entidad.

Después de estudiar las tendencias actuales de las herramientas de inventario a nivel mundial se necesita implementar un sistema que pueda cubrir las necesidades planteadas anteriormente y sobre todo que no precise del personal físico, además de conocer la información relacionada de cada activo en un período corto de tiempo. Dentro del área de gestión de activos y de trazabilidad de éstos, la tecnología de Identificación por Radio Frecuencia (RFID, *Radio Frequency Identification*) se puede aplicar para la gestión del inventario, de forma que se puede fácilmente en un instante dado conocer la localización de cada activo inventariado.

La presente investigación propone la fundamentación de todas las herramientas utilizadas, además de todos los elementos que se necesitan para construir el Sistema de Gestión de Inventarios con tecnología RFID, capaz de inventariar y monitorizar a la vez toda la información de los productos almacenados, lo cual brinda la posibilidad de tomar acciones si ocurre alguna incidencia, ya que con la realización del mismo se pueden cubrir las necesidades expuestas anteriormente para las diferentes entidades del país.

Palabras claves:

Información, gestión, inventario, RFID.

ÍNDICE

INTRODUCCIÓN..... 1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA 4

1.1 Introducción 4

1.2 Conceptos fundamentales 4

 1.2.2 Identificación por Radiofrecuencia (RFID) 4

1.3 Herramientas existentes para la gestión de los inventarios. 4

1.4 Lenguajes, herramientas, técnicas y metodologías de desarrollo. 6

 1.5.1 Metodología de desarrollo utilizada. 6

 1.5.2 Modelación de procesos mediante IDEF0 7

 1.5.3 Lenguaje de programación 8

 1.5.4 Herramienta CASE..... 8

 1.5.5 Entorno de desarrollo 9

 1.5.6 Tecnología del lado del servidor. 9

 1.5.7 Tecnología del lado del cliente 10

 1.5.8 Servidor web. 10

 1.5.9 Servidor de base de datos 10

1.6 Conclusiones 11

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA 12

2.1 Introducción 12

2.2 Problema y situación problemática 12

2.3 Información que se maneja 12

2.4 Solución propuesta 13

2.5 Modelo de negocio 13

2.6 Requisitos no funcionales 15

2.7 Requisitos funcionales 17

2.8 Usuarios relacionados con el sistema 19

2.9 Fase de exploración 19

2.9.1 Historias de usuarios	19
2.9.2 Clasificación de las historias de usuarios.....	20
2.10 Fase de planificación	24
2.10.1 Estimación de esfuerzo por historia de usuario	24
2.10.2 Plan de iteraciones	25
2.10.3 Plan de duración de las iteraciones	25
2.10.4 Plan de entregas	27
2.11 Conclusiones.....	27
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA	28
3.1 Introducción	28
3.2 Arquitectura del sistema	28
3.3 Patrón arquitectónico	¡Error! Marcador no definido.
3.4 Patrones de diseño	29
3.4.1 Patrones para asignar responsabilidades (GRASP)	29
3.4.2 Patrones GoF(del inglés de Gang of Four)	31
3.4.3 Otro patrón utilizado	32
3.5 Tarjetas clase responsabilidad colaborador (CRC)	32
3.6 Diseño de la base de datos	33
3.5.1 Modelo entidad relación	33
3.7 Conclusiones	34
CAPÍTULO 4: IMPLEMETACIÓN Y PRUEBA.....	35
4.1. Introducción	35
4.2. Fase de Implementación	35
4.1. Tareas de la ingeniería.....	35
4.2. Diagrama de despliegue	37
4.3. Pruebas del software	37
4.3.1. Pruebas de Aceptación	38
4.4. Resultado de las pruebas	40
4.5. Conclusiones	41

CONCLUSIONES GENERALES	42
RECOMENDACIONES	43
BIBLIOGRAFÍA	44
ANEXOS	46
Anexo #1 Historias de Usuarios	46
Anexo #2 Tarjetas Clase Responsabilidad	55
Anexo #3 Tareas de Ingenierías	56
Anexo #4 Pruebas de Aceptación	60

ÍNDICE DE TABLAS

Tabla 1: Usuarios relacionados con el sistema	19
Tabla 2: HU: Registrar entrada de productos al almacén.	21
Tabla 3: HU: Dar de baja un producto.	22
Tabla 4: HU: Realizar Inventarios.	22
Tabla 5: HU: Generar inventarios	23
Tabla 6: Estimación de esfuerzo por Historia de Usuario.	24
Tabla 7: Plan de duración de las iteraciones	26
Tabla 8: Plan de entregas	27
Tabla 9: Tarjeta CRC Clase inventarios.	33
Tabla 10: Tarjeta CRC Clase usuario.	33
Tabla 11: Tarea de Ingeniería: Registrar entrada de productos.	36
Tabla 12: Tarea de Ingeniería: Dar de baja un producto	36
Tabla 13: Tarea de Ingeniería: Realizar inventarios.	36
Tabla 14: Caso de prueba de aceptación: Registrar entrada de productos.	38
Tabla 15: Caso de prueba de aceptación: Dar de baja un producto.	39
Tabla 16: Caso de prueba de aceptación: Realizar inventarios.	40

ÍNDICE DE FIGURAS

Figura 1: Componentes básicos de IDEF.	8
Figura 2: Representación del proceso general.	13
Figura 3: Desglose del proceso general.	14
Figura 5: MVC de Spring.	29
Figura 6: Modelo entidad relación.	34
Figura 7: Diagrama de Despliegue.	37
Figura 8 Resultado de pruebas	40

GLOSARIO DE TÉRMINOS

CASE: Herramientas de Ingeniería de Software Asistida por Computadora (*Computer Aided Software Engineering*)

CSS: Hoja de estilo en Cascada (*Cascading Style Sheets*)

CRC: Tarjetas de Cargo o Clase, Responsabilidad y Colaboración

DAO: Objeto de Acceso a Dato (*Data Access Object*)

GOF: (*Gang of Four*)

GRASP: Patron para asignar responsabilidades (*General Responsibility Assignment Software Patterns*)

GNU-LGPL: Licencia Pública General Reducida (*Lesser General Public License*)

HU: Historia de Usuario

HTTP: Protocolo de transferencia de hipertexto (*Hypertext Transfer Protocol*)

HTML: Lenguaje de marcado hipertextual (*Hypertext Transfer Protocol*)

IDE: Entorno Integrado de Desarrollo (*Integrated Development Environment*)

IDEF: Metodología para modelado funcional de procesos (*Integration Definition for Function Modeling*)

IoC: Inversión de Controles (*Inversión of Control*)

MIT: Instituto Tecnológico de *Massachusetts* (*Massachusetts Institute of Technology*)

RFID: Identificación por Radio Frecuencia (*Radio Frequency Identification*)

RUP: Proceso Unificado Racional (*Rational Unified Process*)

SQL: Lenguaje de Consulta Estructurado (*Structured Query Language*)

TCP: Protocolo de Tontrol de Transmisión (*Transmission Control Protocol*)

URL: Localizador de Recursos Uniforme (*Uniform Resource Locator*)

XP: Programación Extrema (*Extreme Programming*).

INTRODUCCIÓN

En las últimas décadas, las tecnologías han tenido un desarrollo vertiginoso, lográndose importantes avances aplicables en diversos campos. Uno de estos campos es el comercio, donde se encuentran numerosas compañías dedicadas a la compra y venta de productos. En cierta medida, el incremento de los intereses de estas entidades, está dado por la cantidad y variedad de productos disponibles para su comercialización. Debido a que dichos productos no se manipulan en su totalidad a diario; los mismos son protegidos en almacenes, los cuales tienden a ser suficientemente grandes para poder almacenar la mayor cantidad de mercancía posible. En este sentido, el avance tecnológico facilita el manejo de los productos almacenados, a través de sistemas para la gestión de los inventarios.

Los sistemas de gestión de inventarios resultan una manera más factible de conocer la cantidad, propiedad y variedad existente de un determinado producto, en un período corto de tiempo; lo que permite una constante información resumida y analizada de los mismos. De esta manera, se reduce el tiempo de operación y se conocen las propiedades y estados de los artículos, así se mantiene un control oportuno de todas las mercancías existentes en la entidad.

Sin embargo, debido al movimiento que se produce en los almacenes, obliga a llevar una trazabilidad en tiempo real para conocer el flujo referente a los artículos y en estos escenarios resultan soluciones de gran impacto los sistemas de gestión de inventarios con tecnología RFID, la cual elimina la necesidad de tener una línea de visión directa o contacto físico. Esta tecnología tiene la capacidad de saber la posición del producto que se está monitoreando; conocer dónde y en qué estado de operación se encuentran los materiales, así como almacenar sus movimientos, en base a un registro histórico de las localizaciones de estos productos y un registro del personal que gestiona los mismos.

En Cuba varias instituciones y almacenes realizan inventarios de forma manual o utilizando un sistema para la gestión. Manualmente se vuelve un proceso engorroso, requiere de mucho tiempo para su realización y aumenta su dificultad cuando se trata de grandes cantidades de productos, lo que posibilita que ocurran incidencias en el futuro y usos indebidos de los mismos. En este transcurso, pueden existir errores tanto en la obtención e introducción de dicha información, por parte del especialista encargado de procesarla, lo que puede dar pie a ilegalidades. En la mayoría de los sistemas con los que cuenta el país, solo controlan las existencias de productos, muchos carecen de alertas o notificaciones en caso de

cambio de productos en las áreas disponibles o cuando se produce alguna irregularidad en el inventario. Los sistemas existentes están desarrollados con poco criterio de seguridad, para un ambiente de escritorio; además de no soportar mecanismos estándares de integración con otras aplicaciones. No siguen las políticas actuales del país de migrar hacia software libre debido a que la mayoría de estos sistemas tienen procedencia extranjera y no satisfacen todas las necesidades para el control de inventario en las entidades.

A partir de lo antes planteado se tiene el siguiente **Problema a resolver**: ¿Cómo lograr un mejor control e identificación del movimiento de los productos en un almacén?

Se define como **Objeto de Estudio** los procesos de gestión de inventarios y se enmarca como **Campo de Acción** los procesos de gestión de inventarios en almacenes utilizando la tecnología RFID.

Para darle solución al problema a resolver se plantea como **Objetivo General** desarrollar un sistema web para el control e identificación del movimiento de productos usando la tecnología RFID.

De ahí se derivan los siguientes **Objetivos Específicos**:

- Identificar Procesos de Negocio relativos al control e identificación del movimiento de productos dentro de plantas de producción y almacenes.
- Realizar Diseño e implementación del sistema propuesto.

Como **Idea a Defender** se plantea que si se desarrolla un sistema de gestión de inventarios, usando la tecnología RFID, se podría lograr un mejor control e identificación de los productos en un almacén.

Para dar cumplimiento al objetivo propuesto se definen las siguientes **Tareas de Investigación**:

- Estudio de las aplicaciones existentes en el mundo sobre el control e identificación del movimiento de productos en almacenes.
- Realizar un estudio de la tecnología RFID, identificando sus principales características y funcionamiento en los sistemas de gestión de inventarios.
- Identificación de los requisitos funcionales y no funcionales del Sistema Web.
- Realización del análisis y diseño del sistema web.
- Validación del diseño del Sistema Web.

Para apoyar el desarrollo de la investigación se emplean los siguientes **Métodos Científicos**:

- **Métodos Teóricos:**

Analítico sintético: Este método permitirá analizar las teorías y los documentos referentes al objetivo de la investigación, facilitando de esta forma la extracción de los elementos más importantes relacionados con el objeto de estudio.

Histórico-Lógico: Este método permite estudiar todo lo relacionado con los inventarios, para así obtener un conocimiento de su desarrollo y comportamiento tanto a nivel internacional como nacional.

El trabajo de diploma se divide en 4 capítulos, los cuales estarán estructurados de la siguiente forma:

Capítulo 1. “Fundamentación Teórica”: La fundamentación teórica de una tesis, es el tratamiento conceptual del tema, se define el asunto o problema de estudio desde determinado punto de vista, con base en datos acumulados en el transcurso de la investigación bibliográfica. Contiene un análisis a partir de la investigación realizada sobre las tecnologías inmersas en la solución. Además se fundamentan los conceptos, metodologías, lenguajes, y herramientas utilizadas.

Capítulo 2. “Características del Sistema”: Este capítulo se enfocará en el problema a resolver mediante el desarrollo del mismo. Se identifican los requisitos funcionales y no funcionales. Se hace énfasis en las características del sistema a desarrollar, la propuesta del mismo, se definen las iteraciones a realizar y la duración que presentarán; se describen también las historias de usuarios y la estimación de esfuerzo que presentan las mismas.

Capítulo 3. “Análisis y Diseño del Sistema”: Este capítulo se basa principalmente en la construcción del sistema a realizar. Se describen las tarjetas: Tarjetas de Cargo o Clase, Responsabilidad y Colaboración (CRC), así como los patrones de diseño, y la estructura que presentará la base de datos a utilizar.

Capítulo 4. “Implementación y Prueba”: En este capítulo se mostrará el sistema implementado exponiendo todo lo relacionado a los procesos de implementación y pruebas del sistema, destacándose la identificación de las tareas de ingeniería y la ejecución de las pruebas a realizar.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

Este capítulo comprende los principales conceptos que serán tratados durante el desarrollo del trabajo, así como algunas herramientas existentes que brindan solución a la gestión de inventarios, demostrando sus ventajas y desventajas. Además se muestra una caracterización de las técnicas, tecnologías y metodologías propuestas para el desarrollo del sistema.

1.2 Conceptos fundamentales

1.2.1 Inventario

Relación ordenada de bienes y existencias de una entidad o empresa, a una fecha determinada. (1) Las empresas realizan inventarios para llevar un registro histórico de los activos y de esta manera conocer el estado de las mercancías, así como sus características, dígame fecha de entrada, número de lote, fecha de caducidad, entre otras.

1.2.2 Identificación por Radiofrecuencia (RFID)

Se denomina RFID al tipo de tecnología utilizada para la identificación de objetos a distancia sin necesidad de contacto físico ni línea de vista. RFID, es un sistema de almacenamiento y recuperación de datos de manera automática. Este utiliza ondas de radio para transmitir la información entre una etiqueta y un lector. Esta etiqueta RFID contiene datos de identificación del objeto al que se encuentra adherido, generando una señal de radio frecuencia con dichos datos. Esta señal es capturada por un lector RFID mediante una antena, que se encarga de extraer la información y transmitírsela al Middleware. El middleware define cómo y qué información se comparte entre RFID y las bases de datos locales, incluyendo referencia de los objetos y su descripción, número de lote, fecha de caducidad o estado del proceso. (2)

1.3 Herramientas existentes para la gestión de los inventarios.

En la actualidad existen diferentes herramientas para la gestión de inventarios y que utilizan para su mejor funcionamiento la tecnología RFID, entre ellas se encuentra la herramienta **Data Suite**, la cual tiene como

principal objetivo el control y la gestión inteligente de sus inventarios, registra los movimientos y las operaciones en tiempo real, sin la intervención humana, para obtener el inventario on-line y vivo, además de actualizar al instante e informarle a la base de alertas. (3) A pesar de todas las cualidades que posee la herramienta se debe destacar que no es una plataforma libre y dentro de sus variadas funciones no da la posibilidad de controlar al personal encargado de registrar el flujo de la mercancía del almacén.

ORION es un sistema que permite planificar los suministros considerados necesarios para poder cumplir con el mantenimiento de los niveles de inventario en el almacén acordados con sus clientes. Utiliza las previsiones de demanda de los productos para planificar sus suministros, y le dará la oportunidad de cumplir y mejorar el nivel de servicio prestado a sus clientes, aumentando así su satisfacción y la fidelización de los mismos (4) Sin embargo a pesar que este software se basa en un histórico de entregas y en los niveles de inventario deseados, solo genera automáticamente pedidos de reposición justo antes de que el inventario del cliente llegue a su estado crítico, sin llevar un control de la trazabilidad de los activos ni del personal que gestiona el flujo de mercancía. Sin dejar de destacar que no es una plataforma libre.

Otra plataforma existente es **CS-Almacenes** que permite controlar el inventario de almacenes. El programa se basa en ir registrando las entradas y salidas de manera que siempre conocen qué cantidad hay de un producto en tiempo real. Controla productos en ilimitados almacenes, llevando a cabo una realización de ajustes manuales si se producen inventarios físicos y conteos, así como realizar traspasos entre almacenes. (5) Es una aplicación fácil de utilizar, ya que es destinada a profesionales que no necesiten controlar trazabilidades ni parámetros complejos, simplemente conocer en todo momento sus existencias. Sin embargo no da la posibilidad de generar una estadística de productos, ni la localización de los mismos dentro del almacén, sin dejar de destacar que es una plataforma privativa.

Otra de las herramientas es **GeSTOCK**, software de Gestión de Inventario Logística y Almacén, compuesto de los módulos agenda, artículos, albaranes¹ de entrada, albaranes de salida, vendedores y proveedores (6). Este software tiene algunas ventajas como es la importación de la información de otras bases de datos y busca registros fácilmente, además de ser un software gratuito, pero no abarca toda la

¹ Albarán: Documento que suele acompañar a una mercancía para acreditar que ha sido entregada satisfactoriamente.

información que se necesita para la gestión de inventarios como la localización de los productos en sus áreas.

1.4 Lenguajes, herramientas, técnicas y metodologías de desarrollo.

En este epígrafe se particularizarán las herramientas y tecnologías utilizadas para el desarrollo del Sistema Web. Las tecnologías fueron escogidas de acuerdo a las ventajas que facilitan un mejor desarrollo de dicho sistema.

1.5.1 Metodología de desarrollo utilizada.

En la calidad de un software, intervienen diferentes cuestiones; entre ellas, el esfuerzo de las personas implicadas en el desarrollo del proyecto, además de la metodología a utilizar. Existen diferentes metodologías, entre ellas se pueden encontrar las metodologías ágiles y las metodologías robustas.

Las robustas están basadas en normas provenientes de estándares seguidos por el entorno de desarrollo; ofrecen por lo general cierta resistencia a los cambios que puedan producirse durante el ciclo de desarrollo del producto, y es un proceso muy bien controlado con muchas políticas y normas. (7)

Para la realización del sistema se decide emplear la metodología ágil Programación Extrema (XP, *Extreme Programming*) por ser este un proyecto pequeño donde todo el trabajo es realizado por una pareja de programadores, en un ambiente de trabajo donde el riesgo de desarrollo es elevado debido al corto tiempo de entrega planteado y a los continuos cambios de requerimientos; además el cliente forma parte del equipo de desarrollo, de esta manera se logra una mejor retroalimentación, corrección de errores y se garantiza la entrega de un producto final con la calidad requerida.

XP centra su atención en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, para esto promueve el trabajo en equipo, se preocupa por el aprendizaje de los desarrolladores, y propicia un buen clima de trabajo para los mismos. Se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, en la comunicación fluida entre todos los participantes y en el coraje para enfrentar los cambios. Está diseñada para mitigar los riesgos en proyectos con corto tiempo de entrega y con continuos cambios de requerimientos. Este enfoque muestra su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente el tiempo de desarrollo pero manteniendo siempre una alta calidad. (8)

Sustituye casos de usos que plantea el Proceso Unificado Racional (RUP, *Rational Unified Process*) por las Historias de Usuario (HU) con el objetivo que cada una no demore más de una semana. Los artefactos que genera son: funcionalidades del sistema, el Plan de Iteraciones que especifica en cada iteración las HU a implementar, el Plan de Entregas que contiene la fecha final de cada iteración, las Tareas de Ingeniería que orientan las actividades a desarrollar para dar respuesta a la implementación de las HU, las tarjetas Clase Responsabilidad Colaborador (CRC) donde se muestran las clases principales y las pruebas que esta metodología propone para realizarle al software. (9)

1.5.2 Modelación de procesos mediante IDEF0

Metodología para modelado funcional de procesos (IDEF, *Integration Definition for Function Modeling*) consiste en una serie de normas que definen la metodología para la representación de actividades o procesos. Es una forma unificada de representar funciones de manera detallada.

El diseño con IDEF0 se muestra mediante Diagramas de caja simple basado en gráficos y flechas; etiquetas de texto para describir cuadros y textos, mediante una estructura jerárquica, con las funciones principales en la parte superior y los sucesivos niveles de subfunciones.

Esta codificación identifica la flecha como una entrada, un control, una salida y un mecanismo (ICOM, *Input, Control, Output, Mechanism*) en la caja principal. (Véase en la figura 1).

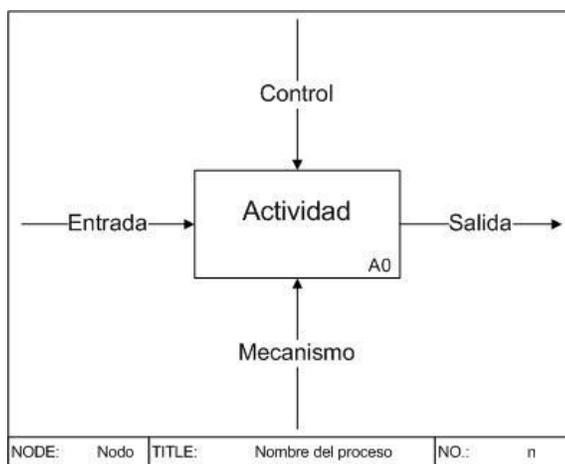


Figura 1: Componentes básicos de IDEF.

La Actividad(o proceso): Se representa mediante cajas.

La Entrada: Representa la información consumida o transformada por una actividad.

La Salida: Representa los objetos producidos por la actividad o proceso.

El Control: Son objetos que gobiernan o regulan cómo, cuándo y si una actividad se ejecuta o no.

Los Mecanismos: Son recursos necesarios para ejecutar un proceso. (10)

1.5.3 Lenguaje de programación

En el desarrollo del Sistema de Inventarios es usado como lenguaje de programación Java, creado por la compañía Sun Microsystems en 1995. Es la tecnología subyacente que permite el uso de programas punteros, como herramientas, juegos y aplicaciones de negocios. (11)

Entre sus diversas características se puede resaltar por ser un lenguaje orientado a objetos ya que agrupan en estructuras encapsuladas tanto sus datos como los métodos (o funciones) que manipulan esos datos; en la medida en que su código fuente se transforma en una especie de código máquina; es interpretado y compilado a la vez; indiferente a la arquitectura: puesto que está diseñado para soportar aplicaciones que serán ejecutadas en los más variados entornos de red, sobre arquitecturas distintas y con sistemas operativos diversos. (12)

1.5.4 Herramienta CASE

Las Herramientas de Ingeniería de Software Asistida por Computadora (CASE, *Computer Aided Software Engineering*) son aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software, con el fin de automatizar los aspectos clave de todo el proceso de desarrollo de un sistema, desde el principio, hasta el final y así incrementar su posición en el mercado competitivo. Estas herramientas pueden ayudar en muchos de los aspectos del ciclo de vida de desarrollo del software, en tareas como la realización del diseño del proyecto, cálculo de costes, documentación o detección de errores.

La herramienta CASE utilizada fue Visual Paradigm, con el fin de automatizar los aspectos claves de todo el proceso de desarrollo del sistema. Este es un estándar ampliamente utilizado para el modelado de software, que además proporciona valiosa ayuda a los desarrolladores visualizando, comunicando y aplicando sus diseños orientados a objetos, además de ser multiplataforma y libre. (13)

1.5.5 Entorno de desarrollo

Un Entorno Integrado de Desarrollo (IDE, *Integrated Development Environment*), es una aplicación o conjunto de estas en las que se combinan las tecnologías a utilizar para el desarrollo del software, permiten entre otras tareas: escribir el código, compilarlo o ejecutarlo y detectar errores.

En la implementación del sistema se utilizó NetBeans, fundado por Sun Microsystems en junio 2000 y de código abierto. Es una herramienta en la que los programadores puedan escribir, compilar, depurar y ejecutar programas. Es utilizado porque está escrito en Java aunque puede servir para cualquier otro lenguaje de programación. (14)

1.5.6 Tecnología del lado del servidor.

- **Spring**

Para el desarrollo del proyecto fue utilizado como framework de desarrollo Spring, siendo este un framework liviano o ligero, ya que no es una aplicación que requiera de muchos recursos para su ejecución. Proporciona la integración del framework con otras herramientas o incluso con otros framework, promueve la reutilización de códigos y un estándar del paradigma orientado a objetos, además es una plataforma que proporciona una infraestructura que actúa de soporte para desarrollar aplicaciones Java. (15)

- **Hibernate**

Hibernate facilita el almacenamiento y la recuperación de objetos de dominio de Java a través de Mapeo objeto / relacional. Hibernate se encarga de la asignación de clases Java a tablas de base de datos y de tipos de datos Java con tipos de datos SQL. Ofrece consulta de datos y recuperación en que reducen significativamente el tiempo de desarrollo. Además está bajo la Licencia Pública General Reducida (GNU

LGPL, *Lesser General Public License*) y se puede descargar gratis y de uso libre para el desarrollo y despliegue de producción. (16)

1.5.7 Tecnología del lado del cliente

- **JQuery**

JQuery es una biblioteca JavaScript creada por John Resig, rápido, pequeño y rico en funciones. JQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia del Instituto Tecnológico de Massachusetts (MIT, *Massachusetts Institute of Technology*) y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados. Entre sus características tiene la facilidad de acceder al documento de Lenguaje de Marcado Hipertextual (HTML, Hypertext Transfer Protocol), modificar la apariencia y contenido de la página con Hojas de estilos en Cascada (CSS, Cascading Style Sheets). (17)

1.5.8 Servidor web.

El servidor web utilizado es Apache, también conocido como simplemente Tomcat o Jakarta Tomcat, es un servidor web multiplataforma que funciona como contenedor de servlets² y que se desarrolla bajo el proyecto denominado Jackarta perteneciente a la Fundación de Software Apache (*Apache Software Foundation*) bajo la licencia Apache versión 2. Además, puede funcionar como servidor web por sí mismo y por esto, es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad. (18)

1.5.9 Servidor de base de datos

Como servidor de bases de datos es utilizado PostgreSQL, por ser una plataforma de código abierto del sistema objeto-relacional de bases de datos, funciona en todos los principales sistemas operativos, convirtiéndola en multiplataforma, además de tener soporte completo para claves foráneas, uniones, vistas y procedimientos almacenados. Otra de sus características es la herencia de tablas, utilizando el paradigma orientado a objetos. PostgreSQL incluye un marco que permite a los desarrolladores definir y

² Los servlets, son objetos que corren dentro del contexto de un servidor de aplicaciones o también conocido como contenedor de servlets

crear sus propios tipos de datos personalizados junto con funciones de apoyo y operadores que definen su comportamiento, además de estar disponible bajo una licencia de código abierto liberal. (19)

1.6 Conclusiones

Durante el capítulo se realizó un estudio sobre los sistemas de gestión de inventarios con tecnología RFID existentes en la actualidad, donde se pudo concluir que algunas presentan características favorables para el presente trabajo pero son privativas y otras no se ajustan a los requerimientos del cliente. Se definieron las tecnologías, herramientas y metodología de desarrollo de software a utilizar ya que cumplieron con las características necesarias para dar solución a la propuesta.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción

En el presente capítulo se describen los procesos de negocio y las funcionalidades del sistema. Se hace alusión a las fases de Exploración y Planificación, propias de la metodología de desarrollo utilizada, donde se confeccionan las historias de usuarios importantes para cada iteración definida por el equipo de desarrollo.

2.2 Problema y situación problemática

La gestión de inventarios de los productos en almacenes es importante porque se monitorea y se controla la cantidad, localización y movimiento de los productos. Esta gestión a menudo es un proceso engorroso, el procesamiento de la información es lento debido a que la obtención de información de cada producto se realiza de forma manual para saber la cantidad y variedad existente, así como las propiedades de los mismos. En este proceso pueden existir errores tanto en la obtención e introducción de dicha información, lo que puede dar pie a ilegalidades. Además en la mayoría de los casos, los robos que puedan ocurrir son detectados mucho tiempo después de haber sucedido.

En estos escenarios las soluciones de Gestión de Inventarios con tecnología RFID tienen gran impacto ya que permiten obtener información de varias etiquetas de identificación rápidamente sin exigir la presencia de personal físico, posibilitando tomar medidas a tiempo ante cualquier incidente o eventualidad. En estas soluciones es necesario obtener la información relacionada a los productos y mantener un control sobre la cantidad, localización y movimiento de los mismos.

2.3 Información que se maneja

Durante la gestión de inventarios se maneja información de los productos almacenados, la cual se encuentra almacenada en bases de datos, por lo que es de suma importancia conocer cómo debe manipularse dicha información. Se dispone, para ello, de un sistema de inicialización con consulta a la base de datos, usando como framework Hibernate para el mapeo de datos.

2.4 Solución propuesta

El objetivo fundamental del proyecto es desarrollar una aplicación para gestionar inventarios mediante la tecnología RFID, para facilitar el control de los productos almacenados, además de permitir generar reportes, conocer estadísticas relacionadas con los productos, localizar los productos en las áreas del almacén y conocer las incidencias o anomalías llevadas a cabo en el proceso de gestión de los productos.

La herramienta propuesta en esta investigación surge a partir de la idea de realizar un sistema con tecnología de punta para así explotar al máximo sus capacidades; confiable; seguro y con interfaces intuitivas.

2.5 Modelo de negocio

A continuación se muestran los procesos de negocios, los cuales han sido modelados mediante IDEF0 ya que simplifica la representación de los mismos. (Véase en la figura 2).

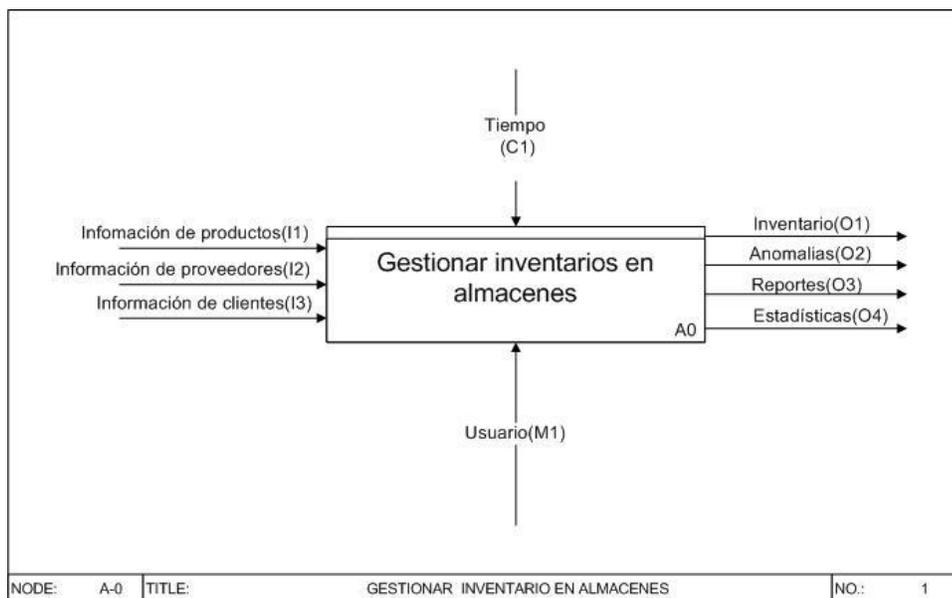


Figura 2: Representación del proceso general.

Descripción

Gestionar inventarios en almacenes: Este proceso permite facilitar y agilizar la realización de inventarios a los productos almacenados. Además comprende otros procesos como administrar productos, generar inventarios y otros que se varan a continuación. (Véase en la figura 3).

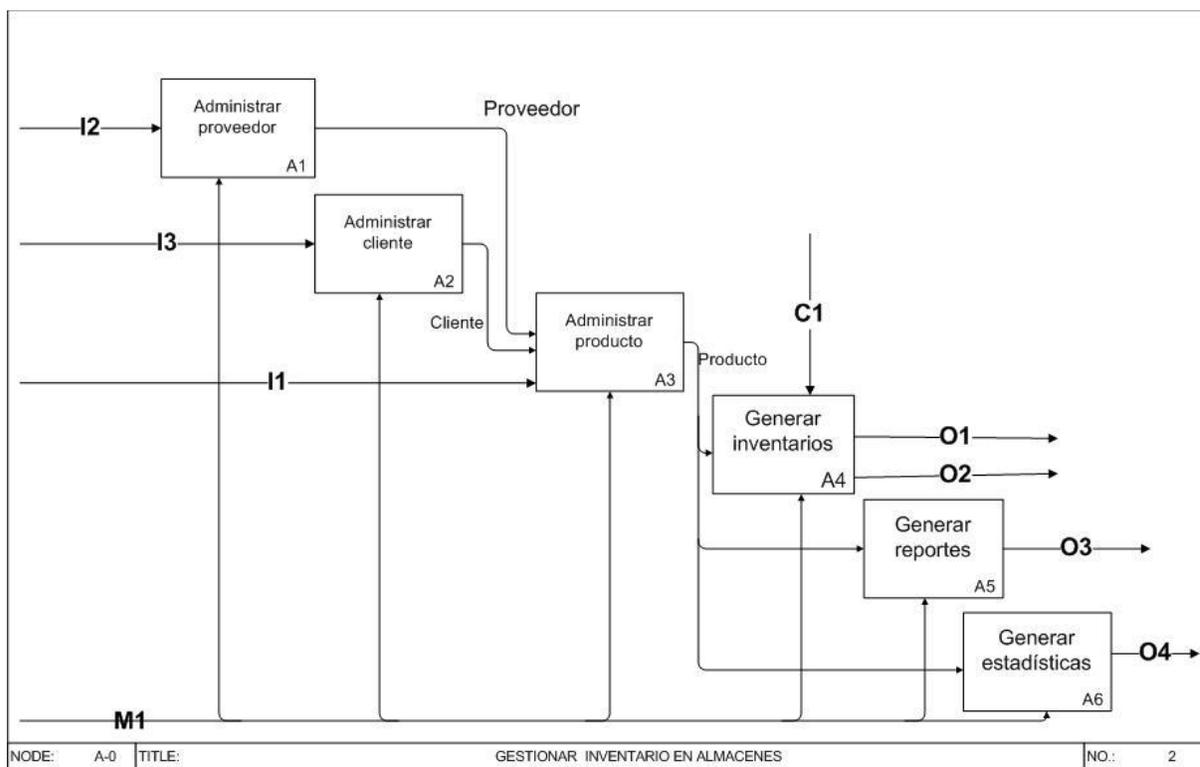


Figura 3: Desglose del proceso general.

Descripción

Administrar productos: este proceso permite administrar la información relacionada con los productos, teniendo como entrada la información del producto y del personal que permite la gestión.

Administrar proveedor: este proceso permite administrar la información del proveedor, teniendo la entrada de dicha información.

Administrar clientes: en este proceso se administra la información del cliente, teniendo la entrada de dicha información.

Generar inventarios: Permite generar los inventarios que revelan los datos y la existencia real de los productos en el almacén, obteniendo como entrada la información de los productos.

Generar reportes: Generar un reporte con la información del producto seleccionado por el usuario, el cual se genera con la información de los productos almacenados.

Generar estadísticas: Generar una estadística de los productos más despachados, obtenidos de la información de los productos almacenados.

2.6 Requisitos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Representan características que hacen al producto atractivo, rápido, confiable, entre otras. En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requerimientos funcionales, es decir una vez que se conozca lo que el sistema debe hacer, se puede determinar cómo ha de comportarse y las cualidades que debe tener.

- **Soporte.**

Se requiere que el sistema cuente con la documentación necesaria para permitir agilizar la gestión y administración de los procesos que automatizarán el sistema en las auditorías informáticas. Se brindarán cursos de capacitación al personal que utilizará el software en caso de ser necesario.

- **Usabilidad.**

Se necesitará una preparación previa para operar con el sistema y un conocimiento anterior de computación. Además el sistema podrá ser utilizado por usuarios con conocimientos básicos en el área de Gestión de Inventarios.

- **Disponibilidad.**

La aplicación podrá ser utilizada siempre que se requiera por aquellas personas con acceso y los mecanismos utilizados para lograr la seguridad, no deben retrasar a los usuarios para obtener los servicios deseados en un momento dado.

- **Eficiencia.**

El tiempo de respuesta a las peticiones tiene cumplir con los estándares a nivel internacional; en este caso el tiempo de respuesta se dará en el orden de milisegundos.

- **Apariencia o Interfaz externa**

El sistema contará con un diseño que muestra en una sola interfaz, el contenido de la sección que el usuario se encuentra utilizando en el momento, evitando las barras de desplazamiento, tanto verticales como horizontales y el uso de un menú claro en todo momento para facilitar el acceso a las distintas funcionalidades del sistema.

- **Seguridad:**

La seguridad del sistema se debe garantizar en todo momento, la cual debe velar por tres aspectos fundamentales:

Confidencialidad

La información manejada en el sistema estará protegida de acceso no autorizado y cada usuario con acceso al sistema sólo debe acceder a los datos que le correspondan por su rol y cuenta registrada. El rol de administrador tendrá acceso a todas las funcionalidades del sistema, mientras que el técnico, solo contará con permisos sobre la información relacionada a los productos existentes en el almacén.

Integridad

El sistema se desplegará en una red local mediante el protocolo de transferencia de hipertexto (HTTP, *Hypertext Transfer Protocol*) donde solo acceden los usuarios de la entidad. El acceso a las funcionalidades estará determinado por los permisos establecidos por el administrador según el tipo de usuario. Autenticarse será obligatorio para el uso de cualquier servicio de la aplicación. Las contraseñas de cada usuario serán protegidas por el algoritmo de encriptación MD5 el cual imposibilita su lectura en claro.

Disponibilidad

Al personal autorizado se le garantizará el acceso al sistema mediante una cuenta de usuario con los privilegios necesarios para poder operar y consultar los datos deseados en el momento establecido.

- **Hardware**

Esta sección incluye los requisitos que describen las interfaces de hardware que deberán ser garantizadas; incluyendo la estructura lógica, rendimiento y el comportamiento esperado.

La estación de trabajo donde se desplegará el sistema requiere una máquina con 512 MB de RAM como mínimo, un procesador igual o superior a Intel Pentium, 160 GB de disco duro mínimo, y una tarjeta de red Ethernet a 100 Mbps o superior.

- **Software**

En las computadoras de los usuarios solo se requiere un navegador Web Mozilla Firefox 3.5 o superior, aunque también se puede utilizar Internet Explorer 6 o superior, además de cualquier distribución de Linux o Windows.

2.7 Requisitos funcionales

Los requisitos funcionales reflejan las capacidades con las que debe contar el sistema. Estos se mantienen invariables sin importar con qué propiedades o cualidades se relacionen. A continuación serán tratados los requerimientos funcionales asociados al Sistema de Gestión de Inventarios:

RF 1 Autenticar

RF 2 Gestionar usuarios.

- **RF 2.1** Adicionar usuarios.
- **RF 2.2** Modificar usuarios.
- **RF 2.3** Desactivar usuarios.
- **RF 2.4** Buscar usuarios.

RF 3 Registrar proveedor.

- **RF 3.1** Buscar proveedor.

RF 4 Registrar cliente.

- **RF 4.1** Buscar clientes.

RF 5 Gestionar productos.

- **RF 5.1** Adicionar productos.
- **RF 5.3** Dar de baja a productos.
- **RF 5.4** Buscar productos.

RF 6 Generar Inventarios.

- **RF 6.1** Realizar inventarios.
- **RF 6.2** Exportar inventarios.

RF 7 Notificar anomalía.

- **RF 7.1** Notificar anomalía al almacenero.
- **RF 7.2** Exportar registro anomalía.

RF 8 Generar reportes.

- **RF 8.1** Realizar reportes.
- **RF 8.2** Exportar reporte.

RF 9 Generar estadísticas.

- **RF 9.1** Graficar los resultados de la estadística.
- **RF 9.2** Exportar estadística.

RF 10 Gestionar áreas del almacén.

- **RF 10.1** Adicionar áreas.
- **RF 10.2** Modificar áreas.
- **RF 10.3** Desactivar áreas.
- **RF 10.4** Buscar áreas.

2.8 Usuarios relacionados con el sistema

Se define como usuarios relacionados con el sistema a aquellos que interactúan de una u otra forma con la plataforma de mensajería y obtienen un resultado de todos los procesos que se ejecutan en ella. (Véase la tabla 1).

Tabla 1: Usuarios relacionados con el sistema.

Nombre	Justificación
Almacenero	El Almacenero representa una persona que tiene la responsabilidad de llevar a cabo los procesos dentro del almacén, todo lo relacionado al registro de productos que tienen entrada y salida del local.
Administrador	El Administrador es el que más privilegios tiene sobre el sistema, es el encargado de gestionar todos los usuarios del sistema, así como mantener el control sobre ellos, además de otorgar permisos y privilegios.

2.9 Fase de exploración

La fase de Exploración es la primera fase definida por la metodología XP. Esta fase comienza por la creación de una serie de historias, llamadas Historias de Usuarios (HU), las cuales definen mediante su redacción qué es lo que verdaderamente necesita el cliente. Es aquí donde se define el alcance real del sistema, permitiendo una familiarización del equipo de desarrollo con las herramientas y tecnologías a usar en la solución.

2.9.1 Historias de usuarios

Las Historias de Usuarios son las técnicas que utiliza la metodología XP para especificar los requisitos del software. Estas son escritas desde la perspectiva del cliente, donde describen de forma breve las características que el sistema debe realizar, aunque también los desarrolladores pueden brindar su ayuda en la identificación de las mismas. El contenido de las historias debe ser concreto, sencillo, dinámico y flexible. Cada una de ellas es lo suficientemente comprensible, como para que los programadores puedan estimar con un reducido margen de riesgo, su tiempo de desarrollo. Es el cliente el encargado de asignarle una prioridad a cada Historia de Usuario y es el equipo de desarrollo el encargado de asignarle

un costo, este se traduce en las semanas que llevará el desarrollo de las mismas. Las historias de usuarios deben poder ser programadas en un tiempo entre una y tres semanas. Si la estimación es superior a tres semanas, debe ser dividida en dos o más historias. Si es menos de una semana, se debe combinar con otra historia. Por otra parte es bueno destacar que las Historias de Usuarios nuevas pueden describirse en cualquier momento, con esto se comprueba la flexibilidad de la metodología.

2.9.2 Clasificación de las historias de usuarios

La prioridad en el negocio:

Alto: Se le otorga a las HU que resultan funcionalidades fundamentales en el desarrollo del sistema, a las que el cliente define como principales para el control integral del sistema.

Medio: Se le otorga a las HU que resultan para el cliente como funcionalidades a tener en cuenta, sin que estas tengan una afectación sobre el sistema que se esté desarrollando.

Bajo: Se le otorga a las HU que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tienen nada que ver con el sistema en desarrollo.

El riesgo en su desarrollo:

Alto: Cuando en la implementación de las HU se consideran la posible existencia de errores que lleven la inoperatividad del código.

Medio: Cuando pueden aparecer errores en la implementación de la HU que puedan retrasar la entrega de la versión.

Bajo: Cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto.

Las HU serán representadas mediante tablas divididas por las siguientes secciones:

- Número: Número de la historia de usuario incremental en el tiempo.
- Nombre de Historia de Usuario: El nombre de la historia de usuario sería para identificarlas mejor entre los desarrolladores y el cliente.
- Usuario: Personas involucradas en el desarrollo de la HU.
- Iteración Asignada: Número de la iteración.

- **Prioridad en negocio:** Alto, Medio o Bajo.
- **Riesgo en Desarrollo:** Alto, Medio o Bajo
- **Puntos estimados:** Tiempo estimado que se demorará el desarrollo de la HU; cada punto es considerado como una semana de trabajo.
- **Puntos Reales:** Tiempo que se demoró en realidad el desarrollo de la HU.
- **Descripción:** Breve descripción de la HU.
- **Observaciones:** Señalamiento o advertencia del sistema.
- **Prototipo de interfaz:** Prototipo de interfaz si aplica.

Se muestran a continuación algunas Historias de Usuarios de prioridad alta, (Véanse las tablas 2, 3, 4 y 5); el resto se describen en el Anexo # 1 del presente documento:

Tabla 2: HU: Registrar entrada de productos al almacén.

Historia de Usuario	
Número: 8	Nombre de Historia de Usuario: Adicionar productos.
Usuario: Esbietta Díaz Matienzo. Reinier Pérez López.	Iteración Asignada: 1
Prioridad en negocio: Alto	Puntos estimados: 2
Riesgo en Desarrollo: Alto	Puntos Reales: 2
Descripción: El sistema te permite adicionar los nuevos productos que fueron encontrados por la tecnología RFID al almacén.	
Observaciones:	
Prototipo de interfaz:	

Tabla 3: HU: Dar de baja un producto.

Historia de Usuario	
Número: 9	Nombre de Historia de Usuario: Dar de baja a productos.
Usuario: Esbietta Díaz Matienzo. Reinier Pérez López.	Iteración Asignada: 1
Prioridad en negocio: Alto	Puntos estimados: 2
Riesgo en Desarrollo: Alto	Puntos Reales: 2
Descripción: El sistema te permite, luego de ser detectado la salida de productos del almacén, darle de baja a dichos productos.	
Observaciones:	
Prototipo de interfaz:	

Tabla 4: HU: Realizar Inventarios.

Historia de Usuario	
Número: 15	Nombre de Historia de Usuario: Realizar Inventarios.
Usuario: Esbietta Díaz Matienzo. Reinier Pérez López.	Iteración Asignada: 1
Prioridad en negocio: Alto	Puntos estimados: 2

Riesgo en Desarrollo: Alto	Puntos Reales: 2
Descripción: El sistema permitirá realizar un inventario en un corto periodo de tiempo a través de la aplicación; mediante un servicio web se solicita la información relacionada con los productos identificados por radio frecuencia y se verifica con la información recogida por el almacenero.	
Observaciones:	
Prototipo de interfaz:	

Tabla 5: HU: Generar inventarios

Historia de Usuario	
Número: 16	Nombre de Historia de Usuario: Exportar inventarios.
Usuario: Esbietta Díaz Matienzo. Reinier Pérez López.	Iteración Asignada: 1
Prioridad en negocio: Alto	Puntos estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: El sistema te permite luego de realizar y mostrar el inventario, la posibilidad de exportar dicho inventario.	
Observaciones: <ul style="list-style-type: none"> El campo URL de la descarga es requerido. 	
Prototipo de interfaz:	

2.10 Fase de planificación

En la fase de planificación en la metodología XP es donde se le concede la prioridad a cada Historia de Usuario, describiendo sus características y funcionalidades requeridas para el desarrollo del software. Durante la planeación se realiza una estimación del esfuerzo que costará implementar cada Historia de Usuario, este se expresa utilizando como medida el punto, el cual es considerado como una semana de trabajo, donde el equipo de desarrollo labora a tiempo completo. Esta estimación incluye todo el esfuerzo asociado a la implementación de la Historia de Usuario.

2.10.1 Estimación de esfuerzo por historia de usuario

A continuación se muestra la estimación del esfuerzo por cada Historia de Usuario descrita para el desarrollo del sistema llegando a los siguientes resultados. (Véase la tabla 6).

Tabla 6: Estimación de esfuerzo por Historia de Usuario.

Historias de Usuarios		Estimación
Autenticar	<ul style="list-style-type: none"> • Autenticar 	1
Registrar proveedor	<ul style="list-style-type: none"> • Registrar proveedor • Buscar proveedor. 	1
Registrar clientes	<ul style="list-style-type: none"> • Registrar clientes • Buscar clientes. 	1
Gestionar productos	<ul style="list-style-type: none"> • Adicionar productos • Dar baja a un productos • Buscar producto. 	2
Realizar inventarios	<ul style="list-style-type: none"> • Crear inventarios • Exportar inventarios 	2
Notificar anomalías	<ul style="list-style-type: none"> • Notificar anomalías al almacenero 	1
Generar reportes	<ul style="list-style-type: none"> • Realizar reportes • Exportar reportes 	2
Generar estadísticas	<ul style="list-style-type: none"> • Graficar resultados de las 	2

	estadísticas <ul style="list-style-type: none"> • Exportar estadísticas 	
Gestionar áreas	<ul style="list-style-type: none"> • Adicionar área • Modificar área • Desactivar área. • Buscar áreas. 	2
Gestionar usuarios	<ul style="list-style-type: none"> • Adicionar usuarios. • Modificar usuarios. • Eliminar usuarios. • Buscar usuarios. 	1

2.10.2 Plan de iteraciones

Luego de ser identificadas y descritas las Historias de Usuarios, además de estimar su esfuerzo, se procede a la planificación de la fase de implementación, donde se realizarán dos iteraciones, las cuales se describen a continuación:

Iteración 1

Esta iteración tiene como objetivo la implementación de las Historias de Usuarios 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 y 15. Esta iteración será el inicio de las funcionalidades para la aplicación que el cliente necesita.

Iteración 2

En esta iteración se tendrán en cuenta las Historias de Usuarios 16, 17, 18, 19, 20, 21, 22, 23, 24 y 25 que facilitarán un mejor funcionamiento de la aplicación para los usuarios que interactúen en ella. Concluida esta iteración se obtendrá la versión 1.0 del producto, listo para ser evaluado y puesto en práctica en un entorno de trabajo real.

2.10.3 Plan de duración de las iteraciones

Este plan se encarga de mostrar las Historias de Usuarios en el orden en que se implementarán en cada iteración, así como la duración estimada. (Véase la tabla 7).

Tabla 7: Plan de duración de las iteraciones

Iteración	Orden de las HU a implementar		Duración total
1	Autenticar	<ul style="list-style-type: none"> • Autenticar 	9 semanas
	Registrar proveedor	<ul style="list-style-type: none"> • Registrar proveedor • Buscar proveedor 	
	Registrar cliente	<ul style="list-style-type: none"> • Registrar cliente • Buscar clientes. 	
	Gestionar producto	<ul style="list-style-type: none"> • Adicionar productos • Dar baja a un producto • Buscar producto. 	
	Gestionar usuarios	<ul style="list-style-type: none"> • Adicionar usuarios • Modificar usuarios • Eliminar usuarios • Buscar usuarios 	
	Realizar inventario	<ul style="list-style-type: none"> • Crear inventarios • Exportar inventarios 	
	Notificar anomalías	<ul style="list-style-type: none"> • Notificar anomalías al almacenero 	
2	Generar reportes	<ul style="list-style-type: none"> • Realizar reportes • Exportar reportes 	6 semanas
	Generar estadísticas	<ul style="list-style-type: none"> • Graficar resultados de las estadísticas • Exportar estadísticas. 	
	Generar áreas	<ul style="list-style-type: none"> • Adicionar área • Modificar área • Desactivar área • Buscar áreas. 	

2.10.4 Plan de entregas

En este plan se detalla la fecha fin de cada iteración, mostrando una versión desarrollada del producto en ese momento hasta lograr el producto final en la fecha establecida. A continuación se muestra el plan de entrega para cada iteración. (Véase la tabla 8).

FPA: Funcionalidades con prioridad alta.

FPM: Funcionalidades con prioridad media

Tabla 8: Plan de entregas

Módulo	Fin de la iteración 1 (30 marzo del 2013)	Fin de la iteración 2 (18 mayo del 2013)
SGIA – RFID	FPA	FPM

2.11 Conclusiones

Como parte del presente capítulo se abordó todo lo referente a las fases de exploración y planificación desarrolladas para la correcta implementación de la aplicación, haciendo una descripción de las Historias de Usuarios divididas por iteraciones y la planificación del esfuerzo dedicado al desarrollo de cada una de ellas en el orden en que se les dará cumplimiento, logrando su terminación de acuerdo a las peticiones del cliente en un periodo de 17 semanas. Además se definieron los procesos de negocio, los cuales fueron modelados con IDEF0 y que serán automatizados para la solución propuesta.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 Introducción

En este capítulo se describe la fase de diseño del sistema propia de la metodología XP. Se define el patrón de arquitectura a seguir y la arquitectura utilizada, así como los patrones de diseño a utilizar para la implementación del sistema. Se propondrá un diseño de la base de datos para satisfacer los requerimientos del software y quedarán definidas las tarjetas CRC (Colaborador - Responsabilidad - Clase) como técnica de diseño.

3.2 Arquitectura del sistema

Un estilo arquitectónico es una transformación impuesta al diseño de todo un sistema. El objetivo es establecer una estructura para todos los componentes de dicho sistema.

Un patrón arquitectónico al igual que un estilo, impone una transformación en el diseño de una arquitectura. Sin embargo un patrón difiere de un estilo en varios elementos fundamentales: el alcance de un patrón es menor debido a que se concentra en un aspecto en lugar de hacerlo en toda la arquitectura, los patrones arquitectónicos tienden a abarcar aspectos específicos del comportamiento dentro del contexto de la arquitectura. Los patrones se usan junto con un estilo arquitectónico para determinar la forma de la estructura general del sistema. (20)

Los términos estilo y patrón arquitectónicos son a menudo confundidos, ya que están ampliamente diseminados. Aunque no se precise de una definición estándar para estos términos, sean estilos o patrones arquitectónicos, las aplicaciones en el desarrollo de software constituyen un elemento elemental.

El sistema presenta la arquitectura Modelo Vista Controlador, la cual está estructurada de acuerdo al framework Spring que sigue dicho patrón para dividir y organizar el código.

El código de la presentación se guarda en la Vista, el código de manipulación de datos se guarda en el Modelo y la lógica de procesamiento de las peticiones constituye el Controlador. Aplicar el patrón MVC a una aplicación resulta bastante útil. El framework Spring incluye varias clases que facilitan el trabajo las cuales son mostradas en la siguiente figura. (Véase la figura 5).

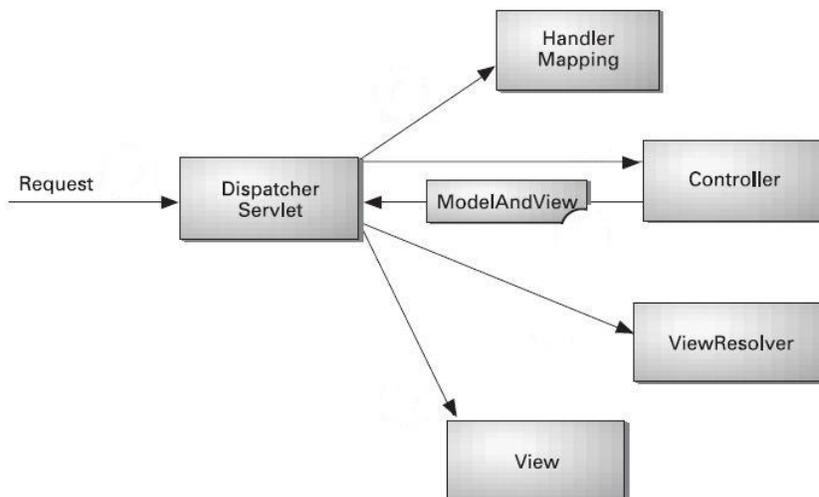


Figura 4: MVC de Spring.

La figura muestra el ciclo de vida de una petición de principio a fin. El proceso se inicia cuando un cliente (típicamente un navegador web) envía una solicitud al componente Controlador Frontal (*DispatcherServlet*) donde una sola controladora delega responsabilidad de una solicitud a otros componentes de una solicitud, para llevar a cabo la transformación real, luego el Controlador (*Controller*) es el componente responsable de manejar la solicitud consultando uno o más Manejadores de Mapeo (*HandlerMappings*) los cuales normalmente realizan su trabajo mediante la asignación de patrones de URL a los objetos del controlador y obtienen como respuesta un Modelo y Vista (*ModelAndView*), de este toma el nombre de la vista y utilizando el Ver y Resolver (*ViewResolver*) encuentra la vista física, a la que le envía los datos extraídos del *ModelAndView* para que sea dibujada, resultado de lo cual se obtiene el código HTML que es enviado como respuesta al cliente. (21)

3.3 Patrones de diseño

3.4.1 Patrones para asignar responsabilidades

La calidad de diseño de la interacción de los objetos y la asignación de responsabilidades presentan gran variación. Las decisiones poco acertadas dan origen a sistemas incompetentes, frágiles y difíciles de mantener. Una implementación hábil se funda en los principios que rigen un buen diseño orientado a objetos. Los patrones para asignar responsabilidades (GRASP, *General Responsibility Assignment Software Patterns*) codifican algunos de ellos, cuando se asignan las responsabilidades.

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones.

Se emplearon en el sistema los siguientes patrones GRASP:

- **Experto, Bajo Acoplamiento y Alta Cohesión**

Estos patrones se encargan de Asignar una responsabilidad a la clase que tiene la información necesaria para cumplirla eliminando la dependencia entre las clases, caracterizando a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme.

En la solución el paquete `acceso_datos.intf` define todas las clases de interfaz **Dao.java** donde se encuentran todos los métodos necesarios para gestionar la base de datos y en el paquete **acceso_datos.impl** se encuentran las clases que implementan estos métodos. Las clases **Impl_Dao.java** representan el acceso a la base de datos jugando el papel de expertas en el acceso al modelo, pero para evitar el acoplamiento o la sobrecarga de información, delega las funciones de acceder a las tablas de la base de datos a las clases **DataAccessUtils.java** e **HibernateUtil.java**.

- **Creador**

Este patrón es el responsable de asignarle a una clase la responsabilidad de crear una instancia de otra. En la solución las clases del negocio **Impl_Dao.java** contienen una instancia de su acceso a datos (**acceso_datos.impl**, **acceso_datos.intf**) correspondiente y cada clase controladora (**controladora.java**) tiene una instancia de su negocio.

- **Controlador**

Este patrón asigna la responsabilidad del manejo de los eventos de un sistema a una o varias clases. Es usado en cada clase controladora del framework Spring. Cada petición que se genera es manejada por una clase **controladora.java** que se encuentra en el paquete **Controladoras**, éstas invocan a las clases de su negocio que necesitan para cumplir con la petición, recibe la respuesta y construye la vista correspondiente para mostrar finalmente la respuesta al usuario.

3.4.2 Patrones GoF

Los patrones GoF (*del inglés de Gang of Four*) complementan a los patrones GRASP y en ocasiones se puede encontrar una contraposición entre este tipo de patrones, e incluso, podría inferirse que algunos patrones GoF son variantes de los patrones GRASP, es por ello que la decisión de utilizar uno u otro debe tomarse con precaución y aplicarse sólo en el ámbito necesario.

Se emplearon en el sistema los siguientes patrones GoF:

- **Comando**

Este patrón soluciona el problema de cuando un objeto o sistema recibe varias peticiones o comandos, para ello cada comando define una clase que lo represente y le asigna la responsabilidad de ejecutarse el mismo, de esta forma reduce la responsabilidad del receptor en el manejo de los comandos, aumenta la facilidad con que pueden agregarse otros comandos y ofrece las bases para registrar los comandos, formar colas de espera con ellos y cancelarlos. Cada petición que se genera es manejada por una clase **controladora.java** que se encuentra en el paquete Controladoras, de esta manera se aplica el patrón Comando.

- **Fachada y Agente**

El patrón Agente explica el comportamiento cuando no se desea o no es posible acceder directamente a un componente, para ello este patrón sugiere definir una clase sustituta de software que represente al componente y asignarle la responsabilidad de comunicarse con el componente real.

El patrón Fachada es generalmente utilizado para resolver problemas de integración, y se aplica cuando se requiere una interfaz común de comunicación con un conjunto de interfaces o funciones de otro subsistema, en este sentido Fachada define una sola clase que unifique las interfaces y le asigna la responsabilidad de colaborar con el subsistema.

La clase **Impl_Dao.java** representa el acceso a la base de datos para los demás componentes del sistema, jugando el rol de fachada, correspondiente con el patrón de mismo nombre. De igual forma actúa como un agente ya que la base de datos solo estará disponible a través de él.

3.4.3 Otro patrón utilizado

- **Inyección de dependencia**

La técnica inyección de dependencias, también conocida como Inversión de control (IoC, *Inversión of Control*), consiste en cargar las propiedades de los objetos (*beans*³) mediante su constructor o sus setters⁴ en el momento de iniciar la aplicación. Sin inyección de dependencias, cada clase llama al objeto que necesita en tiempo de ejecución. Mientras que con inyección de dependencias, cada objeto es cargado en cada clase que lo necesita en tiempo de inicialización.

Para aplicar inyección de dependencias sobre los objetos, Spring permite hacerlo de dos formas distintas, mediante su constructor, o mediante setters. Para ello, a la hora de declarar el bean en el “dispatcher-servlet.xml” de Spring, se debe especificar cuál de los dos métodos vamos a utilizar para la inyección, en este caso mediante el constructor. (22)

3.4 Tarjetas clase responsabilidad colaborador

La metodología de desarrollo XP como parte de la fase de diseño propone el modelado de Clase-Responsabilidad-Colaborador (CRC), lo que constituye un modelo simple de organizar las clases más relevantes para las funcionalidades del sistema. Este modelado CRC utiliza tarjetas, con el objetivo de desarrollar una representación organizada de las clases, las cuales tienen los siguientes datos:

- Nombre
- Responsabilidades
- Colaboraciones

Una responsabilidad es cualquier cosa que la clase sabe o hace. Los colaboradores son aquellas clases que se requieren para que una clase reciba la información necesaria para completar una responsabilidad.

A continuación se muestran las tarjetas CRC correspondientes a las clases más relevantes para las funcionalidades del sistema (*Véanse las tablas 9 y 10*) y el resto se detallan en el Anexo # 2.

³ Un bean es un objeto que es instanciado, ensamblado (cuando sus dependencias son inyectadas), y en general, administrado por el contenedor de IoC.

⁴ Método set que se utiliza para cambiar los valores de los atributos de una clase.

Tabla 9: Tarjeta CRC Clase inventarios.

Clase inventario	
Descripción: Clase que realiza inventarios.	
Responsabilidad:	Colaborador
Realizar inventarios. Detecta anomalías.	IInventario, InventarioImpl, IInventario_Dao, InventarioImpl_Dao

Tabla 10: Tarjeta CRC Clase usuario.

Clase usuario	
Descripción: Clase que gestiona los usuarios.	
Responsabilidad:	Colaborador
Adicionar usuario. Eliminar usuario. Modificar usuario. Buscar usuario. Mostrar usuario.	IUsuario, UsuarioImpl, IUsuario_Dao, UsuarioImpl_Dao.

3.5 Diseño de la base de datos

3.5.1 Modelo entidad relación

El modelo Entidad-Relación facilita tener una estructura de datos de forma tal que se pueda entender fácilmente lo que se está diseñando. A continuación se muestran las distintas entidades y sus relaciones, agregadas al sistema de almacenamiento de datos para cubrir las funcionalidades del sistema. (Véase la figura 6)

Modelo entidad relación

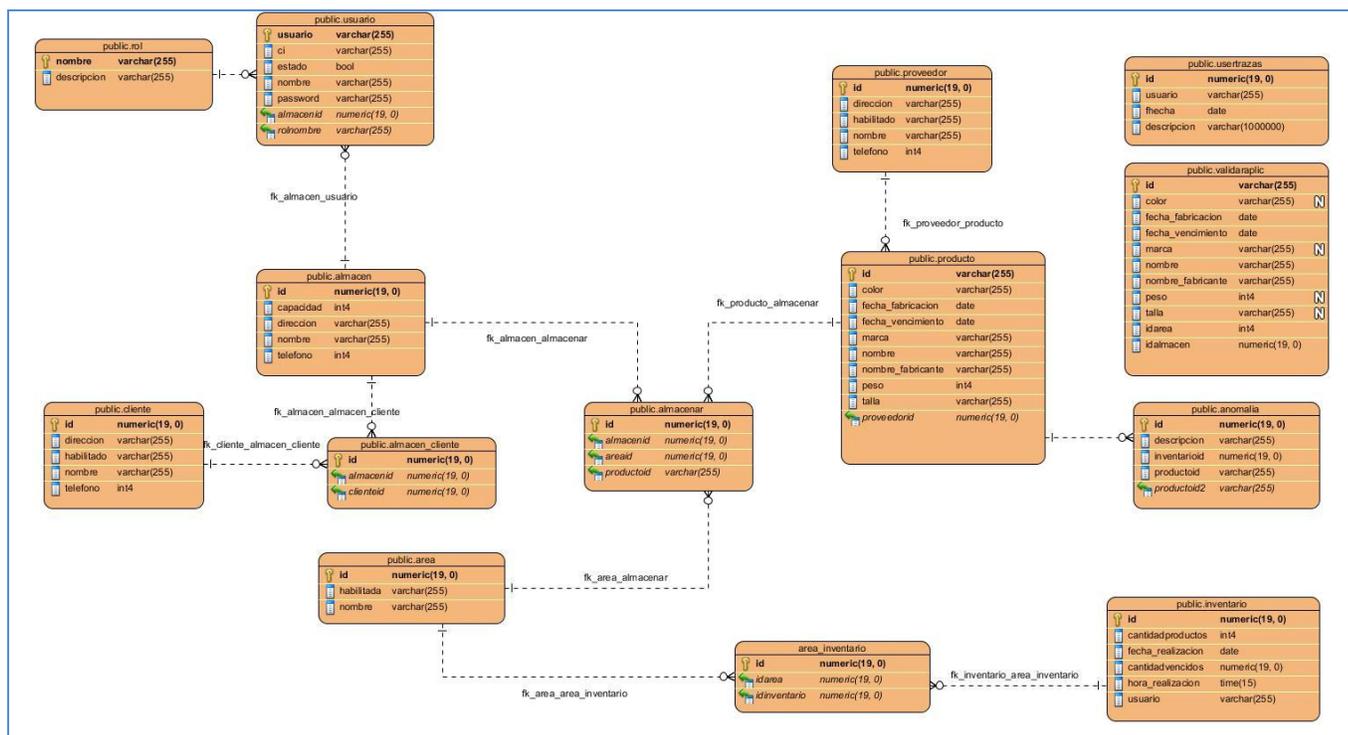


Figura 5: Modelo entidad relación.

3.6 Conclusiones

En el presente capítulo se abordó lo referente a la fase de diseño del proyecto, que contempla los patrones de diseño que favorecen el buen desempeño de la siguiente fase y describe el diseño de la base de datos mediante el modelo físico con todas las entidades que se manejan en el sistema. A partir de este punto se puede continuar con las siguientes fases en el desarrollo del sistema atendiendo a que se cumplan los requerimientos que se proponen implementar.

CAPÍTULO 4: IMPLMETACIÓN Y PRUEBA

4.1. Introducción

En el presente capítulo se describen las tareas de ingeniería generadas por cada HU y las pruebas planteadas por la metodología XP, diseñando los casos de pruebas por cada HU definida, para así evaluar que la aplicación web esté completamente funcional. Además se presenta el diagrama de despliegue y se mostrarán a través de tablas los resultados obtenidos durante la realización de las pruebas de aceptación.

4.2. Fase de Implementación

La metodología XP plantea que la implementación de un software se hace iterativamente, obteniendo al culminar cada iteración un producto funcional, que debe ser probado y mostrado al cliente. Durante el transcurso de las iteraciones, se realiza la implementación de las HU definidas por el cliente y descritas por el equipo de desarrollo en la etapa de Planificación. Como parte de este plan, se descomponen estas HU en tareas de la ingeniería las cuales son asignadas a los programadores para ser implementadas durante la iteración correspondiente. (23)

4.1. Tareas de la ingeniería

Las tareas de la ingeniería serán representadas mediante tablas divididas por las siguientes secciones:

- Número tarea: los números deben ser consecutivos.
- Número historia de usuario: número de la historia de usuario a la que pertenece la tarea.
- Nombre tarea: nombre que identifica a la tarea.
- Tipo de tarea: las tareas pueden ser de: desarrollo, corrección, mejora, otra (especificar)
- Puntos estimados: tiempo estimado en semanas que se le asignará a su desarrollo.
- Fecha inicio: fecha en que inicia el desarrollo de la tarea.
- Fecha fin: fecha en que finaliza el desarrollo de la tarea.
- Programador responsable: nombre y apellidos del programador.
- Descripción: breve descripción de la tarea.

A continuación se muestran algunas de las tareas de ingeniería correspondientes a las HU de prioridad alta (*Véanse las tablas 11, 12 y 13*) y el resto se detallan en el Anexo #3 del presente trabajo.

Tabla 11: Tarea de Ingeniería: Registrar entrada de productos.

Tarea de Ingeniería	
Número de tarea: 8	Número de HU: 8
Nombre de tarea: Adicionar productos al almacén.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio: 17/2/2013	Fecha de fin: 2/3/2013
Programador responsable: Reinier Pérez López Esbieta Díaz Matienzo.	
Descripción: Implementar la funcionalidad de adicionar la información referente a un producto.	

Tabla 12: Tarea de Ingeniería: Dar de baja un producto

Tarea de Ingeniería	
Número de tarea: 9	Número de HU: 9
Nombre de tarea: Dar de baja un producto	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio: 17/2/2013	Fecha de fin: 2/3/2013
Programador responsable: Reinier Pérez López Esbieta Díaz Matienzo	
Descripción: Implementar las funcionalidades de eliminar a un usuario.	

Tabla 13: Tarea de Ingeniería: Realizar inventarios.

Tarea de Ingeniería	
Número de tarea: 15	Número de HU: 15
Nombre de tarea: Realizar inventarios	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio: 10/3/2013	Fecha de fin: 16/3/2013
Programador responsable: Reinier Pérez López Esbieta Díaz Matienzo	

Descripción: Implementar las funcionalidades de crear un inventario.

4.2. Diagrama de despliegue

Un diagrama de despliegue representa la distribución física del sistema en términos de cómo se distribuirán las funcionalidades entre los nodos, cada nodo representa un recurso de cómputo, siendo estos procesadores o dispositivos hardware que se necesitarán para el despliegue del sistema. (Véase la figura 7).

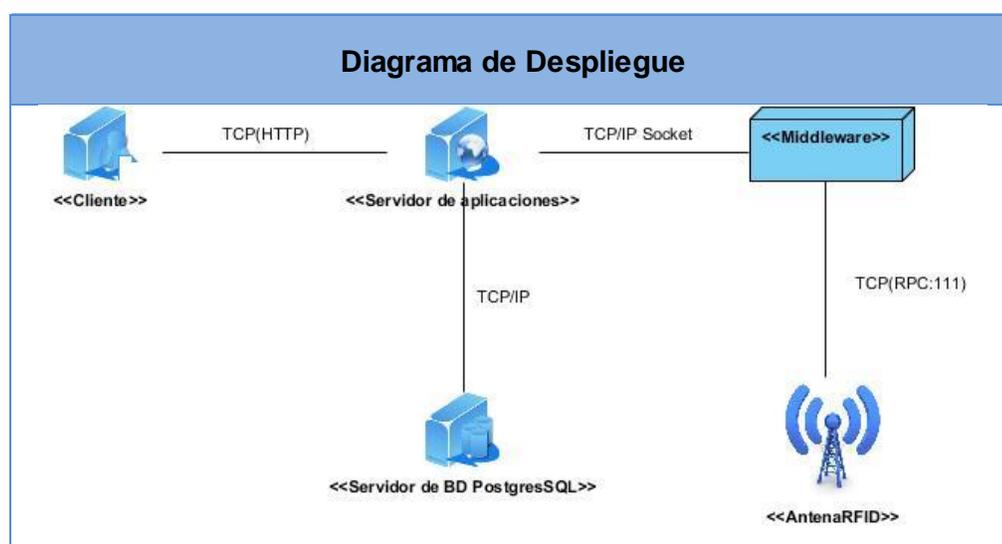


Figura 6: Diagrama de Despliegue.

4.3. Pruebas del software

XP anima a probar constantemente tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones.

XP divide las pruebas en dos grupos: pruebas unitarias, desarrolladas por los programadores y encargadas de verificar el código de forma automática y las pruebas de aceptación, destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida además de comprobar que dicha funcionalidad sea la esperada por el cliente. (24)

4.3.1. Pruebas de Aceptación

Las pruebas de aceptación son pruebas de caja negra que se crean a partir de las HU. En estas serán probadas las funcionalidades exigidas por el cliente, descritas en las HU que se han implementado. Las pruebas de aceptación se llevarán a cabo redactando los casos de prueba, teniendo en cuenta el orden de las HU y la prioridad que ha sido asignada a las funcionalidades.

Las pruebas de aceptación correspondiente a cada una de las funcionalidades serán representadas mediante tablas divididas por las secciones siguientes:

- Código de la prueba de aceptación.
- Número de la historia de usuario a la que se le realiza la prueba.
- Nombre de la funcionalidad.
- Descripción de la funcionalidad.
- Condiciones de ejecución de la funcionalidad.
- Entrada y pasos de ejecución que realiza el usuario con el objetivo de obtener el resultado esperado.
- Resultado esperado.
- Evaluación de la prueba.

A continuación se muestra algunos casos de pruebas de aceptación de HU de prioridad alta (*Véanse las tablas 14, 15 y 16*) y el resto se detallan en el Anexo # 4 del presente trabajo.

Tabla 14: Caso de prueba de aceptación: Registrar entrada de productos.

Caso de prueba de aceptación	
Código: HU8 – P8	Número de HU: 8
Nombre: Adicionar productos.	
Descripción: La aplicación brinda la posibilidad de adicionar productos al almacén.	
Condiciones de ejecución: La solicitud de adicionar productos al almacén.	
Entrada/Pasos: El usuario accede a la interfaz inicial, escoge la opción Producto; seguidamente se muestra una interfaz con la lista de los productos ubicado en las áreas y las opciones Adicionar producto,	

<p>Dar baja y Buscar producto. El usuario selecciona la opción Adicionar producto y se muestra la interfaz con la lista de los nuevos productos con los que cuenta el almacén, además de la opción Aceptar o Denegar y Seleccionar Proveedor.</p>
<p>Resultado esperado</p> <p>Correcto: Si el usuario selecciona el proveedor y la opción Aceptar, se muestra el siguiente mensaje “El producto ha sido añadido correctamente”.</p> <p>Incorrecto: Si el usuario no selecciona el proveedor, el sistema muestra un mensaje de error “Debe introducir un proveedor”.</p> <p>Si el usuario selecciona la opción Denegar, se genera una anomalía.</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla 15: Caso de prueba de aceptación: Dar de baja un producto.

Caso de prueba de aceptación	
Código: HU9 – P9	Número de HU: 9
Nombre: Dar baja a productos.	
Descripción: La aplicación brinda la posibilidad de dar de baja a productos del almacén.	
Condiciones de ejecución: La solicitud de darle de baja a productos del almacén.	
Entrada/Pasos: El usuario accede a la interfaz inicial, escoge la opción Productos; seguidamente se muestra una interfaz con la lista de los productos y las opciones Adicionar productos, Dar baja y Buscar producto. El usuario selecciona la opción Dar baja y se muestra la interfaz con la lista de los productos que no se encuentran en el almacén, además de la opción Aceptar o Denegar y Seleccionar Cliente.	
Resultado esperado	
<p>Correcto: Si el usuario selecciona el cliente y la opción Aceptar, se muestra la interfaz con todos los productos sin mostrar los que han sido dados de baja y el siguiente mensaje “El producto ha sido dado de baja” .</p> <p>Incorrecto: Si el usuario no selecciona el cliente, el sistema muestra un mensaje de error “Debe introducir un cliente”.</p> <p>Si el usuario selecciona la opción Denegar, se genera una anomalía.</p>	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 16: Caso de prueba de aceptación: Realizar inventarios.

Caso de prueba de aceptación	
Código: HU15 – P15	Número de HU: 15
Nombre: Realizar inventarios.	
Descripción: La aplicación debe dar la posibilidad de realizar un inventario en un periodo corto de tiempo.	
Condiciones de ejecución: La solicitud de realizar un inventario.	
Entrada/Pasos: El usuario accede a la interfaz inicial, escoge la opción de realizar inventarios sin existir datos de entrada para esta prueba. El sistema da a conocer la información de los productos en un inventario.	
Resultado esperado	
Correcto: Se muestra el inventario.	
Evaluación de la prueba: Prueba satisfactoria.	

4.4. Resultado de las pruebas

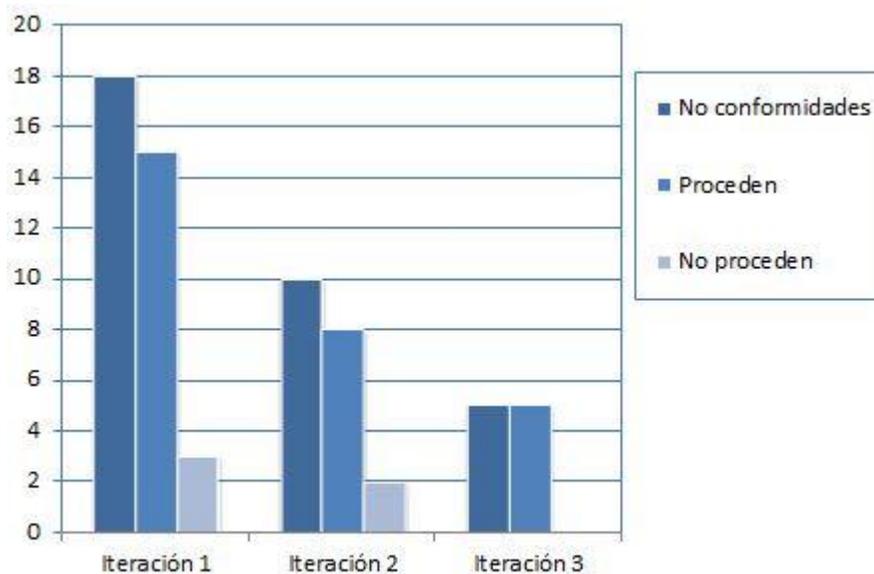


Figura 7 Resultado de pruebas

4.5. Conclusiones

En este capítulo, luego de culminar cada iteración, el producto funcional fue probado y mostrado al cliente. Se realizaron las tareas de la ingeniería correspondientes a cada HU listas para su implementación y quedó diseñado el modelo de despliegue de la aplicación. Se plantearon además las pruebas a seguir durante la elaboración del mismo, donde se definieron las pruebas de aceptación logrando la satisfacción del cliente con el software desarrollado.

CONCLUSIONES GENERALES

El resultado principal de este trabajo es un software funcional para la realización de inventarios utilizando la tecnología RFID, con toda la documentación generada durante su diseño e implementación. Con el propósito de darle cumplimiento al objetivo general y a la problemática planteada en el presente trabajo, se han llevado a cabo satisfactoriamente cada una de las tareas que fueron trazadas al comienzo del mismo.

- Se realizó un estudio del estado actual en el que se encuentra inmersa la gestión de inventarios que utilizan la tecnología RFID para agilizar sus procesos y la exposición de las tendencias mayormente utilizadas en el mundo lo que contribuyó al desarrollo de la solución.
- Durante el desarrollo de los capítulos se hizo un seguimiento de las etapas por las que pasó el software, haciendo uso de la metodología XP. Mediante el uso de la herramienta case Visual Paradigm se generó el diagrama de despliegue y con el uso del NetBeans se desarrolló el sistema en lenguaje Java, empleando las herramientas y tecnologías que se seleccionaron en el capítulo 1 siendo óptimas para la aplicación.
- Se logró un buen levantamiento de requisitos del sistema permitiendo el desarrollo de cada una de las funcionalidades para gestionar los procesos de inventario.
- Se validaron las funcionalidades implementadas a través de la interfaz del sistema, mediante el diseño de pruebas de aceptación. Las pruebas permitieron encontrar inconformidades en la validación de datos y el tratamiento de errores que contribuyeron a aumentar la calidad del sistema.
- Como resultado se obtiene una aplicación web funcional, que contribuye a la gestión de inventarios mediante la tecnología RFID como una nueva guía para el seguimiento y control de objetos en diferentes almacenes del país.

RECOMENDACIONES

A partir del resultado obtenido con la realización del presente trabajo de diploma se hace necesario que queden plasmadas algunas recomendaciones para futuras versiones del sistema.

- Se recomienda que el presente trabajo sea utilizado como referencia para posteriores investigaciones.
- Desarrollar un manual de usuario para familiarizar al cliente en el manejo del sistema.

Bibliografía

1. Org. *Org.* [En línea] [Citado el: 10 de 12 de 2012.] www.definicion.org/inventario..
2. Ferroxtag. [En línea] [Citado el: 1 de 10 de 2012.]
<http://www.ferroxtag.com/Downloads/BasicsRFID-ES.pdf>.
3. Dipole. *Data Suite.* [En línea] [Citado el: 15 de 12 de 2012.]
<http://www.dipolerfid.es/Soluciones/RFID-Stocks-Inventario.aspx>.
4. Orion. *Software Orion.* [En línea] [Citado el: 15 de 12 de 2012.]
<http://www.softwarseleccion.com/orion-p-1151..>
5. CS-Almacenes. *Software CS-Almacenes.* [En línea] [Citado el: 15 de 12 de 2012.]
<http://www.softwarseleccion.com/cs+almacenes-p-3197>.
6. @GeStocks. *Software @GeStocks.* [En línea] [Citado el: 15 de 12 de 2012.]
<http://www.softwarseleccion.com/gestock-p-1113>.
7. Quispe Carita, Vilma, Huamantuco Solorzano, Dante Harry y Vargas Yupanqui, José Luis. 2011. *METODOLOGIA RUP (RATIONAL UNIFIED PROCESS).* Peru : s.n., 2011.
8. Carmen Penadés, María, H. Canós, José y Letelier, Patricio. 2004. *Métodologías Ágiles en el Desarrollo de Software.* Universidad Politécnica de Valencia : s.n., 2004.
9. Fernández Escribano, Gerardo. *Introducción a Extreme Programming.*
10. *Curso de IDEF0, Introducción a IDEF0.*
11. Java. *Java.* [En línea] [Citado el: 17 de 1 de 2013.] <http://www.java.com>.
12. Departamento de Tratamiento de la Información y Codificación(TIC). *Criptonomicón.* [En línea] [Citado el: 17 de 1 de 2013.] <http://www.iec.csic.es/criptonomicon/java/quesjava.html>.
13. Robles, avid Agudo RuanoPablo Niñosles AznarFernando García CussóPablo Delgado de. Scribd. *Herramientas CASE.* [En línea] [Citado el: 15 de 1 de 2013.] <http://es.scribd.com>.
14. NetBeans IDE. *NetBeans IDE.* [En línea] [Citado el: 5 de 2 de 2013.] <https://netbeans.org>.
15. Springhispano. *Spring.* [En línea] [Citado el: 17 de 1 de 2013.] <http://www.springhispano.org>.
16. Hibernate. *Hibernate.* [En línea] [Citado el: 3 de 2 de 2013.] <http://www.hibernate.org/>.
17. jQuery. *jQuery.* [En línea] [Citado el: 5 de 2 de 2013.] <http://jquery.com>.
18. Apache Tomcat. *Apache Tomcat.* [En línea] [Citado el: 20 de 1 de 2013.] <http://tomcat.apache.org>.

19. PostgreSQL. *PostgreSQL*. [En línea] [Citado el: 20 de 1 de 2013.] <http://www.postgresql.org>.
20. Pressman. *Diseño arquitectónico. Parte 1. Capítulo 10*.
21. Craig Walls, Ryan Breidenbach. *SPRING IN ACTION*.
22. [En línea] [Citado el: 6 de 4 de 2013.] <http://www.programacionj2ee.com/inyeccion-de-dependencias-en-spring/>.
23. Pressman., R. S. *Ingeniería del Software: Un Enfoque Práctico (sexta ed.)*.
24. Crispin, T. H. Lisa. *Testing Extreme Programming*. [En línea] [Citado el: 5 de 4 de 2013.] http://www.ebooksbay.org/The_Digital_Library/2008/01/23/Testing_Extreme_Programming__The_XP_Series_.
25. MYRIAM YADIRA LEAL CASTELLANOS, YULY CAROLINA LEAL MOLINA, LEYDI CAROLINA MEDINA CASTIBLANCO. [En línea] [Citado el: 20 de 4 de 2013.] <http://basesii.wikispaces.com/file/view/Caracter%C3%ADsticas+de+la+arquitectura+Cliente.pdf>.
26. Ecured. *Ecured*. [En línea] [Citado el: 1 de 6 de 2013.] http://www.ecured.cu/index.php/Cliente-Servidor#Arquitectura_Cliente-Servidor.

ANEXOS

Anexo #1 Historias de Usuarios

Historia de Usuario	
Número: 1	Nombre de Historia de Usuario: Autenticar
Usuario: Esbietta Díaz Matienzo. Reinier Pérez López.	Iteración Asignada: 1
Prioridad en negocio: Alto	Puntos estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: Los usuarios del sistema se deben autenticar.	
Observaciones: <ul style="list-style-type: none"> • Usuario • Contraseña 	
Prototipo de interfaz:	

Historia de Usuario	
Número: 2	Nombre de Historia de Usuario: Desactivar proveedor.

Usuario: Esbietta Díaz Matienzo. Reinier Pérez López.	Iteración Asignada: 1
Prioridad en negocio: Alto	Puntos estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: Los usuarios del sistema se deben autenticar.	
Observaciones:	
Prototipo de interfaz:	

Historia de Usuario	
Número: 3	Nombre de Historia de Usuario: Registrar proveedor.
Usuario: Esbietta Díaz Matienzo. Reinier Pérez López.	Iteración Asignada: 1
Prioridad en negocio: Alto	Puntos estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: El sistema permitirá registrar los datos del proveedor.	
Observaciones: <ul style="list-style-type: none"> • Identificador • Nombre • Dirección • Teléfono 	

- Estado

Prototipo de interfaz:

Historia de Usuario

Número: 4

Nombre de Historia de Usuario: Buscar proveedor.

Usuario: Esbietta Díaz Matienzo.
Reinier Pérez López.

Iteración Asignada: 1

Prioridad en negocio: Alto

Puntos estimados: 1

Riesgo en Desarrollo: Alto

Puntos Reales: 1

Descripción: El sistema permitirá buscar un proveedor

Observaciones:

- Criterio de búsqueda.

Prototipo de interfaz:

Historia de Usuario

Número: 5

Nombre de Historia de Usuario: Registrar clientes

Usuario: Esbietta Díaz Matienzo.
Reinier Pérez López.

Iteración Asignada: 1

Prioridad en negocio: Alto	Puntos estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: El sistema permitirá registrar los datos del cliente que solicita el producto.	
Observaciones: <ul style="list-style-type: none"> • Nombre • Dirección • Teléfono • Estado • Identificador 	
Prototipo de interfaz:	

Historia de Usuario	
Número: 6	Nombre de Historia de Usuario: Desactivar clientes
Usuario: Esbietta Díaz Matienzo. Reinier Pérez López.	Iteración Asignada: 1
Prioridad en negocio: Alto	Puntos estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: El sistema permitirá registrar los datos del cliente que solicita el producto.	
Observaciones: <ul style="list-style-type: none"> • Nombre • Dirección 	

- Teléfono
- Estado
- Identificador

Prototipo de interfaz:

Historia de Usuario

Número: 7	Nombre de Historia de Usuario: Buscar clientes.
Usuario: Esbietta Díaz Matienzo. Reinier Pérez López.	Iteración Asignada: 1
Prioridad en negocio: Alto	Puntos estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: El sistema permitirá buscar un cliente.	
Observaciones:	
Prototipo de interfaz:	

Historia de Usuario

Número: 10	Nombre de Historia de Usuario: Buscar productos.
Usuario: Esbietta Díaz Matienzo. Reinier Pérez López.	Iteración Asignada: 1

Prioridad en negocio: Alto	Puntos estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: El sistema permitirá buscar productos.	
Observaciones:	
<ul style="list-style-type: none"> • Criterio de búsqueda 	
Prototipo de interfaz:	

Historia de Usuario	
Número: 11	Nombre de Historia de Usuario: Adicionar usuarios.
Usuario: Esbietta Díaz Matienzo. Reinier Pérez López.	Iteración Asignada: 1
Prioridad en negocio: Alto	Puntos estimados: 2
Riesgo en Desarrollo: Alto	Puntos Reales: 2

<p>Descripción: El sistema le permite al administrador adicionar una cuenta.</p>
<p>Observaciones:</p> <ul style="list-style-type: none"> • Nombre • Usuario • Contraseña • Carnet de identidad • Estado
<p>Prototipo de interfaz:</p>

Historia de Usuario	
Número: 12	Nombre de Historia de Usuario: Desactivar usuarios.
Usuario: Esbietta Díaz Matienzo. Reinier Pérez López.	Iteración Asignada: 1
Prioridad en negocio: Alto	Puntos estimados: 2
Riesgo en Desarrollo: Alto	Puntos Reales: 2
Descripción: El sistema le permitirá al administrador eliminar una cuenta.	
Observaciones:	

- Usuario

Prototipo de interfaz:

Historia de Usuario

Número: 13	Nombre de Historia de Usuario: Modificar usuarios.
Usuario: Esbietta Díaz Matienzo. Reinier Pérez López.	Iteración Asignada: 1
Prioridad en negocio: Alto	Puntos estimados: 2
Riesgo en Desarrollo: Alto	Puntos Reales: 2

Descripción: El sistema le permitirá al administrador modificar una cuenta.

Observaciones:

- Nombre.
- Contraseña
- Carnet de identidad
- Estado

Prototipo de interfaz:

Historia de Usuario

--	--

Número: 14	Nombre de Historia de Usuario: Buscar usuarios.
Usuario: Esbietta Díaz Matienzo. Reinier Pérez López.	Iteración Asignada: 1
Prioridad en negocio: Alto	Puntos estimados: 2
Riesgo en Desarrollo: Alto	Puntos Reales: 2
Descripción: El sistema le permitirá al administrador buscar un usuario.	
Observaciones: <ul style="list-style-type: none"> • Criterio de búsqueda 	
Prototipo de interfaz:	

Historia de Usuario	
Número: 17	Nombre de Historia de Usuario: Notificar anomalías al almacenero.
Usuario: Esbietta Díaz Matienzo. Reinier Pérez López.	Iteración Asignada: 1
Prioridad en negocio: Alto	Puntos estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: El sistema permitirá notificarle al almacenero las incidencias encontradas al realizar	

inventarios.
Observaciones:
Prototipo de interfaz:

Anexo #2 Tarjetas Clase Responsabilidad

Clase proveedor	
Descripción: Clase que registra los proveedores.	
Responsabilidad:	Colaborador
Adicionar proveedor. Eliminar proveedor. Modificar proveedor. Obtener proveedor. Listar proveedor.	IProveedor, ProveedorImpl, IProveedor_Dao, ProveedorImpl_Dao.

Clase clientes	
Descripción: Clase que registra los clientes.	
Responsabilidad:	Colaborador
Adicionar clientes. Desactivar clientes. Obtener clientes. Listar clientes Buscar clientes	ICientes, ClientesImpl, ICientes_Dao, ClientesImpl_Dao.

Clase productos

Descripción: Clase que gestiona los productos.	
Responsabilidad:	Colaborador
Adicionar productos. Eliminar productos. Obtener productos. Listar productos. Buscar productos.	IProductos, ProductosImpl, IProductos_Dao, ProductosImpl_Dao.

Anexo #3 Tareas de Ingenierías

Tarea de Ingeniería	
Número de tarea: 1	Número de HU: 1
Nombre de tarea: Autenticar.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 27/12/2013	Fecha de fin: 2/2/2013
Programador responsable: Reinier Pérez López Esbieta Díaz Matienzo.	
Descripción: Implementar la funcionalidad de autenticar los usuarios al sistema con sus roles definidos.	

Tarea de Ingeniería	
Número de tarea: 2	Número de HU: 2
Nombre de tarea: Registrar proveedor.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 3/2/2013	Fecha de fin: 9/2/2013
Programador responsable: Reinier Pérez López Esbieta Díaz Matienzo.	
Descripción: Implementar la funcionalidad de adicionar un proveedor de productos al sistema.	

Tarea de Ingeniería	
Número de tarea: 3	Número de HU: 3
Nombre de tarea: Desactivar proveedor.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 3/2/2013	Fecha de fin: 9/2/2013
Programador responsable: Reinier Pérez López Esbieta Díaz Matienzo.	
Descripción: Implementar la funcionalidad de desactivar un proveedor de productos al sistema.	

Tarea de Ingeniería	
Número de tarea: 4	Número de HU: 4
Nombre de tarea: Buscar proveedor.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 3/2/2013	Fecha de fin: 9/2/2013
Programador responsable: Reinier Pérez López Esbieta Díaz Matienzo.	
Descripción: Implementar la funcionalidad de buscar un proveedor de productos al sistema.	

Tarea de Ingeniería	
Número de tarea: 5	Número de HU: 5
Nombre de tarea: Registrar clientes.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 10/2/2013	Fecha de fin: 16/2/2013
Programador responsable: Reinier Pérez López Esbieta Díaz Matienzo.	

Descripción: Implementar la funcionalidad de adicionar un cliente al sistema, que sería la persona que se lleva la mercancía del almacén.

Tarea de Ingeniería	
Número de tarea: 6	Número de HU: 6
Nombre de tarea: Desactivar clientes.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 10/2/2013	Fecha de fin: 16/2/2013
Programador responsable: Reinier Pérez López Esbieta Díaz Matienzo.	
Descripción: Implementar la funcionalidad de desactivar un cliente al sistema, que sería la persona que se lleva la mercancía del almacén.	

Tarea de Ingeniería	
Número de tarea: 7	Número de HU: 7
Nombre de tarea: Buscar clientes.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 10/2/2013	Fecha de fin: 16/2/2013
Programador responsable: Reinier Pérez López Esbieta Díaz Matienzo.	
Descripción: Implementar la funcionalidad de buscar un cliente al sistema, que sería la persona que se lleva la mercancía del almacén.	

Tarea de Ingeniería	
Número de tarea: 8	Número de HU: 8
Nombre de tarea: Buscar producto.	

Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio: 17/2/2013	Fecha de fin: 2/3/2013
Programador responsable: Reinier Pérez López Esbietta Díaz Matienzo.	
Descripción Implementar la funcionalidad de buscar un producto.	

Tarea de Ingeniería	
Número de tarea: 11	Número de HU: 11
Nombre de tarea: Adicionar usuarios.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio: 3/2/2013	Fecha de fin: 9/3/2013
Programador responsable: Reinier Pérez López Esbietta Díaz Matienzo.	
Descripción: Implementar la funcionalidad de adicionar un usuarios al sistema, donde el encargado de realizar esta funcionalidad seria el administrador del mismo.	

Tarea de Ingeniería	
Número de tarea: 12	Número de HU: 12
Nombre de tarea: Modificar usuarios.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio: 3/2/2013	Fecha de fin: 9/3/2013
Programador responsable: Reinier Pérez López Esbietta Díaz Matienzo.	
Descripción: Implementar la funcionalidad de modificar los datos de un usuarios al sistema, donde el encargado de realizar esta funcionalidad seria el administrador del mismo.	

Anexo #4 Pruebas de Aceptación

Caso de prueba de aceptación	
Código: HU1 – P1	Número de HU: 1
Nombre: Autenticar	
Descripción: La aplicación brinda la posibilidad autenticar a los usuarios de acuerdo al rol correspondiente introduciendo su nombre de usuario y contraseña.	
Condiciones de ejecución: Acceder al sistema.	
Entrada/Pasos: Se muestra en la pantalla Autenticar Usuario solicitando usuario y contraseña, el usuario introduce los datos solicitados y presiona la opción Iniciar Sesión.	
Resultado esperado	
Correcto: El usuario se introduce en el sistema.	
Incorrecto: El usuario no podrá entrar al sistema y este la de la posibilidad de volver a introducir los datos correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba de aceptación	
Código: HU2 – P2	Número de HU: 2
Nombre: Registrar proveedor	
Descripción: La aplicación brinda la posibilidad registrar(o adicionar) el proveedor encargado de realizar la entrega de productos al almacén.	
Condiciones de ejecución: La entrada al sistema.	
Entrada/Pasos: El usuario accede a la interfaz inicial, escoge la opción Proveedor; seguidamente se muestra una interfaz con todos los proveedores existentes, el usuario presiona la opción Adicionar Proveedor y se muestran los siguientes campos: identificador, nombre, dirección, estado y teléfono. Además se encuentran las opciones de Aceptar o Cerrar la petición.	
Resultado esperado	
Correcto: El usuario introduce los datos correctamente, presiona la opción Aceptar y se adiciona un nuevo proveedor al sistema.	

Incorrecto: Si se introduce alguno de los datos incorrectamente se muestra el mensaje de error "Existen campos incorrectos" y da la opción de volver a introducir los datos o Cancelar la petición.

Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación

Código: HU3 – P3

Número de HU: 3

Nombre: Desactivar proveedor

Descripción: La aplicación brinda la posibilidad desactivar un proveedor.

Condiciones de ejecución: La entrada al sistema.

Entrada/Pasos: El usuario accede a la interfaz inicial, escoge la opción Proveedor; seguidamente se muestra una interfaz con todos los proveedores existentes, el usuario presiona la opción Desactivar Proveedor y se muestran el siguiente campo: nombre. Además se encuentran las opciones de Aceptar o Cerrar la petición.

Resultado esperado

Correcto: El usuario selecciona el proveedor, presiona la opción Aceptar y se desactiva el proveedor seleccionado.

Incorrecto: Si no se selecciona el proveedor el sistema muestra el mensaje de error "Seleccione el proveedor" y da la opción de volver a seleccionar el proveedor o Cancelar la petición.

Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación

Código: HU4 – P4

Número de HU: 4

Nombre: Buscar proveedor

Descripción: La aplicación brinda la posibilidad de buscar un proveedor que este registrado en el sistema.

Condiciones de ejecución: La entrada al sistema.

Entrada/Pasos: El usuario accede a la interfaz inicial, escoge la opción Proveedor; seguidamente se

muestra la opción Buscar. Se introduce el criterio de búsqueda y el sistema muestra los proveedores correspondientes al criterio.
Resultado esperado
Correcto: Se muestra los proveedores correspondientes al criterio.
Incorrecto: Si se introduce un criterio de búsqueda incorrecto el sistema no muestra ningún resultado.
Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación	
Código: HU5 – P5	Número de HU: 5
Nombre: Registrar clientes	
Descripción: La aplicación brinda la posibilidad registra(o adicionar) los datos de los clientes que se llevan la mercancía del almacén.	
Condiciones de ejecución: La entrada al sistema.	
Entrada/Pasos: El usuario accede a la interfaz inicial, escoge la opción Clientes; seguidamente se muestra una interfaz con todos los clientes existentes, el usuario presiona la opción Adicionar Cliente y se muestran los siguientes campos: identificador, nombre, dirección, estado y teléfono. Además se encuentran las opciones de Aceptar o Cerrar la petición.	
Resultado esperado	
Correcto: El usuario introduce los datos correctamente, presiona la opción Aceptar y se adiciona un nuevo cliente al sistema.	
Incorrecto: Si se introduce alguno de los datos incorrectamente se muestra el mensaje de error "Existen campos incorrectos" y da la opción de volver a introducir los datos o Cancelar la petición.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba de aceptación	
Código: HU6 – P6	Número de HU: 6
Nombre: Desactivar cliente	
Descripción: La aplicación brinda la posibilidad desactivar un cliente.	

Condiciones de ejecución: La entrada al sistema.
Entrada/Pasos: El usuario accede a la interfaz inicial, escoge la opción Clientes; seguidamente se muestra una interfaz con todos los clientes existentes, el usuario presiona la opción Desactivar Cliente y se muestran el siguiente campo: nombre. Además se encuentran las opciones de Aceptar o Cerrar la petición.
Resultado esperado
Correcto: El usuario selecciona el cliente, presiona la opción Aceptar y se desactiva el cliente seleccionado.
Incorrecto: Si no se selecciona el cliente el sistema muestra el mensaje de error "Seleccione el cliente" y da la opción de volver a seleccionar el cliente o Cancelar la petición.
Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación	
Código: HU7 – P7	Número de HU: 7
Nombre: Buscar clientes	
Descripción: La aplicación brinda la posibilidad de buscar un cliente que este registrado en el sistema.	
Condiciones de ejecución: La entrada al sistema.	
Entrada/Pasos: El usuario accede a la interfaz inicial, escoge la opción Clientes; seguidamente se muestra la opción Buscar. Se introduce el criterio de búsqueda y el sistema muestra los clientes correspondientes al criterio.	
Resultado esperado	
Correcto: Se muestra los clientes correspondientes al criterio.	
Incorrecto: Si se introduce un criterio de búsqueda incorrecto el sistema no muestra ningún resultado.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba de aceptación	
Código: HU10 – P10	Número de HU: 10
Nombre: Buscar producto	

Descripción: La aplicación brinda la posibilidad de buscar un producto.
Condiciones de ejecución: La entrada al sistema.
Entrada/Pasos: El usuario accede a la interfaz inicial, escoge la opción Productos; seguidamente se muestra la opción Buscar. Se introduce el criterio de búsqueda y el sistema muestra los productos correspondientes al criterio.
Resultado esperado
Correcto: Se muestra los productos correspondientes al criterio.
Incorrecto: Si se introduce un criterio de búsqueda incorrecto el sistema no muestra ningún resultado.
Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación	
Código: HU11 – 11	Número de HU: 11
Nombre: Adicionar usuario.	
Descripción: La aplicación brinda la posibilidad al administrador del sistema de adicionar un nuevo usuario en el mismo.	
Condiciones de ejecución: La entrada al sistema siendo un administrador.	
Entrada/Pasos: El administrador accede a la interfaz inicial, escoge la opción Usuarios, seguidamente se muestra una interfaz con todos los usuarios existentes, el administrador presiona la opción Adicionar Usuario y se muestran los siguientes campos: usuario, nombre, carnet de identidad, contraseña, rol y almacén. Además se encuentran las opciones de Aceptar o Cerrar la petición.	
Resultado esperado	
Correcto: Se adiciona un nuevo usuario al sistema.	
Incorrecto: Si el administrador introduce datos incorrectos, el sistema muestra un mensaje de error: “Existen datos incorrectos” y brinda la opción de introducir los datos nuevamente o la opción Cancelar.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba de aceptación	
Código: HU12 – P12	Número de HU: 12

Nombre: Modificar usuarios
Descripción: El sistema brinda la posibilidad al administrador de modificar un usuario.
Condiciones de ejecución: La entrada al sistema.
Entrada/Pasos: El administrador accede a la interfaz inicial, escoge la opción Usuarios, seguidamente se muestra una interfaz con todos los usuarios existentes, el administrador presiona la opción Modificar Usuario y se muestran los siguientes campos: usuario, nombre, carnet de identidad, contraseña, rol y almacén. Además se encuentran las opciones de Aceptar o Cerrar la petición.
Resultado esperado
Correcto: El administrador selecciona el usuario que desea modificar, introduce los datos correctamente y presiona la opción Aceptar.
Incorrecto: El administrador no selecciona el usuario que desea modificar e introduce los datos correctamente y presiona la opción Aceptar. El sistema muestra un mensaje de error: “Seleccione el usuario que desea modificar” y brinda la opción de introducir los datos nuevamente o la opción Cancelar. El administrador selecciona el usuario que desea modificar e introduce los datos incorrectamente y presiona la opción Aceptar. El sistema muestra un mensaje de error: “Existen datos incorrectos” y brinda la opción de introducir los datos nuevamente o la opción Cancelar.
Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación	
Código: HU13 – P13	Número de HU: 13
Nombre: Desactivar usuario.	
Descripción: La aplicación le brinda la opción al administrador de desactivar un usuario.	
Condiciones de ejecución: La entrada del administrador al sistema.	
Entrada/Pasos: El administrador accede a la interfaz inicial, escoge la opción Usuarios; seguidamente se muestra una interfaz con todos los usuarios existentes, el administrador presiona la opción Desactivar Usuario y se muestran el siguiente campo: usuario. Además se encuentran las opciones de Aceptar o Cerrar la petición.	
Resultado esperado	
Correcto: El usuario selecciona el usuario, presiona la opción Aceptar y se desactiva el usuario seleccionado.	

Incorrecto: Si no se selecciona el usuario el sistema muestra el mensaje de error "Seleccione el usuario " y da la opción de volver a seleccionar el usuario o Cancelar la petición.

Evaluación de la prueba: Prueba satisfactoria.