Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

APLICACIÓN DE LA INGENIERÍA DE SOFTWARE PARA LA AUTOMATIZACIÓN DEL CONTROL DE INSUMOS EN BODEGAS DE LA DIRECCIÓN DE ÁREA DE SALUD DE ESCUINTLA

Edving David Morales Arana
Asesorado por la Inga. Mirna Ivonne Aldana Larrazábal

Guatemala, septiembre de 2013

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

APLICACIÓN DE LA INGENIERÍA DE SOFTWARE PARA LA AUTOMATIZACIÓN DEL CONTROL DE INSUMOS EN BODEGAS DE LA DIRECCIÓN DE ÁREA DE SALUD DE ESCUINTLA

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

EDVING DAVID MORALES ARANA
ASESORADO POR LA INGA. MIRNA IVONNE ALDANA LARRAZÁBAL

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, SEPTIEMBRE DE 2013

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Ing. Alfredo Enrique Beber Aceituno
VOCAL II	Ing. Pedro Antonio Aguilar Polanco
VOCAL III	Inga. Elvia Miriam Ruballos Samayoa
VOCAL IV	Br. Walter Rafael Véliz Muñoz
VOCAL V	Br. Sergio Alejandro Donis Soto
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. Marlon Antonio Pérez Türk
EXAMINADORA	Inga. Floriza Felipa Ávila Pesquera
EXAMINADORA	Inga. Inga. Sonia Yolanda Castañeda Ramírez
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

APLICACIÓN DE LA INGENIERÍA DE SOFTWARE PARA LA AUTOMATIZACIÓN DEL CONTROL DE INSUMOS EN BODEGAS DE LA DIRECCIÓN DE ÁREA DE SALUD DE ESCUINTLA

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha marzo de 2011.

Edving David Morales Arana

Escuintla, 17 de enero de 2011

Inga. Norma Ileana Sarmiento Zeceña
Directora EPS
Facultad de Ingeniería
Universidad de San Carlos de Guatemala

Por medio de la presente hago de su conocimiento que el estudiante Edving David Morales Arana, con carnet 2001-17517 ha finalizado su proyecto de EPS de 6 meses, titulado "Aplicación de la Ingeniería de Software para la automatización del control de insumos en bodegas de la Dirección de Área de Salud de Escuintla".

Este fue realizado en la bodega central de la Dirección del Área de Salud de Escuintla, cumpliendo satisfactoriamente con los objetivos propuestos.

Además, he revisado el informe final del estudiante, correspondiente a su trabajo de graduación, y a mi criterio, cumple con los objetivos, según el protocolo.

Agradeciendo su atención a la presente, me suscribo.

Atentamente,

Mirna Ivonne Aldana Larrazabal
INGENIERA EN CIENCIAS Y SISTEMAS
Colegiada No. 9567

Inga. Mirna Ivonne Aldana Larrazabal
Colegiado Activo No. 9567

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA

FACULTAD DE INGENIERÍA

UNIDAD DE EPS

Guatemala, 18 de mayo de 2012.
REF.EPS.DOC.734.05.2012.

Inga. Norma Ileana Sarmiento Zeceña de Serrano
Directora Unidad de EPS
Facultad de Ingeniería
Presente

Estimada Ingeniera Sarmiento Zeceña.

Por este medio atentamente le informo que como Supervisora de la Práctica del Ejercicio Profesional Supervisado, (E.P.S) del estudiante universitario de la Carrera de Ingeniería en Ciencias y Sistemas, **Edving David Morales Arana** Carné No. **200117517** procedí a revisar el informe final, cuyo título es **"APLICACIÓN DE LA INGENIERÍA DE SOFTWARE PARA LA AUTOMATIZACIÓN DEL CONTROL DE INSUMOS EN BODEGAS DE LA DIRECCIÓN DE ÁREA DE SALUD DE ESCUINTLA".**

En tal virtud, **LO DOY POR APROBADO**, solicitándole darle el trámite respectivo.

Sin otro particular, me es grato suscribirme.

Atentamente,

"Id y Enseñad a Todos"

Inga. Floriza Felipa Avila Pesquera de Medinilla
Supervisora de EPS
Área de Ingeniería en Ciencias y Sistemas

Universidad de San Carlos de Guatemala
SUPERVISOR (A) DE EPS
Unidad de Prácticas de Ingeniería y EPS
Facultad de Ingeniería

FFAPdM/RA

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA

FACULTAD DE INGENIERÍA

UNIDAD DE EPS

Guatemala, 18 de mayo de 2012.
REF.EPS.D.525.05.2012.

Ing. Marlon Antonio Pérez Turk
Director Escuela de Ingeniería Ciencias y Sistemas
Facultad de Ingeniería
Presente

Estimado Ingeniero Perez Turk.

Por este medio atentamente le envío el informe final correspondiente a la práctica del Ejercicio Profesional Supervisado, (E.P.S) titulado **"APLICACIÓN DE LA INGENIERÍA DE SOFTWARE PARA LA AUTOMATIZACIÓN DEL CONTROL DE INSUMOS EN BODEGAS DE LA DIRECCIÓN DE ÁREA DE SALUD DE ESCUINTLA"**, que fue desarrollado por el estudiante universitario **Edving David Morales Arana** carné No. **200117517** quien fue debidamente asesorado por la Inga. Mirna Ivonne Aldana Larrazabal y supervisado por la Inga. Floriza Felipa Ávila Pesquera de Medinilla.

Por lo que habiendo cumplido con los objetivos y requisitos de ley del referido trabajo y existiendo la aprobación del mismo por parte de la Asesora y la Supervisora de EPS, en mi calidad de Directora apruebo su contenido solicitándole darle el trámite respectivo.

Sin otro particular, me es grato suscribirme.

Atentamente,
"Id y Enseñad a Todos"

Inga. Norma Ileana Sarmiento Zeceña de Serrano
Directora Unidad de EPS

Universidad de San Carlos de Guatemala
DIRECCION
Unidad de Practicas de Ingeniería - EPS
Facultad de Ingeniería

NISZ/ra

Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 30 de Mayo de 2012

Ingeniero
Marlon Antonio Pérez Turk
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación-EPS del estudiante **EDVING DAVID MORALES ARANA,** carné **2001-17517,** titulado: **"APLICACIÓN DE LA INGENIERÍA DE SOFTWARE PARA LA AUTOMATIZACIÓN DEL CONTROL DE INSUMOS EN BODEGAS DE LA DIRECCIÓN DE ÁREA DE SALUD DE ESCUINTLA"**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,

Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
INGENIERIA EN CIENCIAS Y SISTEMAS
COORDINACION TESIS

ESCUELA DE CIENCIAS Y SISTEMAS

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA

FACULTAD DE INGENIERÌA
ESCUELA DE CIENCIAS Y SISTEMAS
TEL: 24767644

El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación ***"APLICACIÓN DE LA INGENIERÍA DE SOFTWARE PARA LA AUTOMATIZACIÓN DEL CONTROL DE INSUMOS EN BODEGAS DE LA DIRECCIÓN DE ÁREA DE SALUD DE ESCUINTLA",*** *realizado por el estudiante EDVING DAVID MORALES ARANA, aprueba el presente trabajo y solicita la autorización del mismo.*

"ID Y ENSEÑAD A TODOS"

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DIRECCION DE INGENIERIA EN CIENCIAS Y SISTEMAS

Ing. Marlon Antonio Pérez Turk
Director, Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 28 de agosto 2013

Universidad de San Carlos
de Guatemala

Facultad de Ingeniería
Decanato

DTG. 595.2013

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **APLICACIÓN DE LA INGENIERÍA DE SOFTWARE PARA LA AUTOMATIZACIÓN DEL CONTROL DE INSUMOS EN BODEGAS DE LA DIRECCIÓN DE ÁREA DE SALUD DE ESCUINTLA,** presentado por el estudiante universitario: **Edving David Morales Arana,** autoriza la impresión del mismo.

IMPRÍMASE:

Ing. Murphy Olympo Paiz Recinos
Decano

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DECANO
FACULTAD DE INGENIERIA

Guatemala, 28 de agosto de 2013

/gdech

ACTO QUE DEDICO A:

Dios	Porque con su luz ha colmado de bendiciones mi vida.
Mis padres	Edving Morales y Marta Arana, pilares de este logro. Siempre estaré agradecido, y retribuiré su apoyo y esfuerzo.
Mis hermanos	Henry y Hasell Morales Arana, por su apoyo incondicional cuando los necesité. Pueden contar con el mío, y les deseo éxitos en sus vidas personales y profesionales.
Mis abuelas	Su esfuerzo y sacrificio en pos de sus hijos, se ve reflejado en este logro.
Mis tíos y primos	Quienes se congratulan con este triunfo, y a quienes les deseo alcancen sus metas.
Mis amigos y compañeros	Porque en algún momento me ayudaron; espero que yo haya hecho lo mismo por ustedes cuando lo requirieron.

AGRADECIMIENTOS A:

Inga. Ivonne Aldana	Por su dinamismo, apoyo y confianza depositada en mi persona al realizar este trabajo.
Carlos Fuentes	Por su ayuda para buscar un lugar donde hacer mi práctica de Ejercicio Profesional Supervisado.
Juan Carlos Samayoa, Patricia Cruz y personal del Área de Salud de Escuintla	Por su colaboración abierta para el desarrollo del proyecto de Ejercicio Profesional Supervisado.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES

FIGURAS

GLOSARIO

Aplicación	En informática, una aplicación es un programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de trabajo.
Bug	En software, es un defecto en la ejecución del mismo.
Cárdex	Tarjeta utilizada en bodega para registrar los movimientos en inventario para cada insumo.
Checkout	En control de versiones de software, comando para obtener en un entorno de trabajo local, el código fuente de un repositorio.
Commit	En control de versiones de software, comando para integrar el código fuente en un entorno de trabajo local, con el código fuente en un repositorio.
DASE	Dirección de Área de Salud de Escuintla.

Framework	Conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar.
IDE	Entorno de desarrollo integrado, por sus siglas en inglés, es una aplicación con herramientas para desarrollo de software.
Iteración	Acción y efecto de repetir.
Repositorio	Sitio creado para guardar información centralizada de un producto de software en desarrollo.
RUP	Rational Unified Process. Conjunto de metodologías, adaptables al contexto y necesidades de cada organización.
Stakeholder	Dentro de una organización, es el interesado en que el proyecto se lleve a cabo.

RESUMEN

Este proyecto fue desarrollado como respuesta a la necesidad de la Dirección del Área de Salud de Escuintla, de tener control de los insumos que se manejan en las bodegas de la institución.

Para construir apropiadamente la herramienta que diera solución a esta necesidad, se planeó trabajar el proyecto, basado en la metodología RUP, orientando el desarrollo con métodos de ingeniería de software que garantizaran la transición ordenada entre las fases del ciclo de vida de desarrollo del producto.

De las fases de RUP, se tomaron las actividades de las que más provecho se obtendría al ponerlas en práctica, tomando en cuenta que se esperaba generar documentación que servirá aún después de terminado el proyecto, como soporte técnico para la institución.

OBJETIVOS

General

Aplicar una metodología de desarrollo de software, para elaborar en seis meses, un sistema que automatice los procesos de registro de entradas y salidas, así como el control en inventarios de bodegas de la Dirección del Área de Salud de Escuintla, reduciendo el tiempo de recopilación de la información.

Específicos

1. Elaborar, con base en la metodología RUP, al menos un artefacto de la fase de elaboración, uno de la fase de construcción y dos de la fase de transición, los que servirán como base técnica en poder de la institución.

2. Finalizar en el tiempo propuesto, con un sistema que tenga la capacidad de almacenar el 100% de las existencias de los insumos en las bodegas de la Dirección Departamental de Salud de Escuintla.

3. Finalizar en el tiempo propuesto, un sistema con la capacidad de generar automáticamente todos los reportes que actualmente se realizan de forma manual, en las bodegas, tomando como base a la bodega central de la institución.

4. Construir la arquitectura del sistema propuesto, empleando tecnología actual y cuyo uso no represente costo ni problemas legales a la institución.

5. Recabar en la institución y manejar con reserva, la información que sea proporcionada, para entender los requerimientos que debe satisfacer el sistema.

6. Coordinar con las autoridades y personal de la institución, la implementación de la solución dentro de un período de 3 semanas, tras finalizado el tiempo propuesto.

7. Capacitar en una semana, tras finalizada la implementación, a usuarios del sistema, mostrándoles el uso del mismo y proveyéndolos de los manuales respectivos.

INTRODUCCIÓN

En este documento se detalla la forma en que se llevó a cabo la construcción del sistema para el control de insumos de bodegas de la Dirección del Área de Salud de Escuintla (DASE).

La DASE, como institución encargada de la gestión de servicios de prevención y curación en el área de salud, para el departamento de Escuintla, tiene bajo su cargo bodegas a nivel departamental, las cuales gestionan los insumos y entre estos, los medicamentos que deben distribuirse en el área, por lo que una buena forma de administrarlos, sería automatizando los procesos que están involucrados en dicha gestión.

El sistema resultante para la automatización de tales procesos, es el producto de la aplicación de los ya establecidos, para el desarrollo de aplicaciones de software. Una metodología como orientación para la selección de métodos y técnicas para lograr objetivos, es importante como base en un proyecto, ya que permite conocer el camino hacia la finalización del trabajo. En este caso, se consideró la metodología RUP como la base de la administración del proyecto de software planteado.

El contenido de este trabajo, describe las actividades de dicha metodología, empleadas para el desarrollo del proyecto. En el primer capítulo, se describe a la institución beneficiada; además, se hace un repaso que justifica la estructuración en los procesos de creación de un producto de software, basándose en los estudios de ingeniería.

El segundo capítulo trata sobre la primera fase de la metodología utilizada, la de Iniciación, describiéndose las actividades que se tomaron en cuenta.

El capítulo tres se basa en la fase de elaboración de la metodología RUP, explicando por medio de qué actividades se diseñó el producto propuesto. El capítulo cuatro, explica las actividades que conformaron las fases de construcción y transición.

En cada uno de los capítulos, se describen los artefactos resultantes, de acuerdo con los objetivos propuestos. La documentación elaborada, se adjunta a este trabajo como apéndice.

1. LA DIRECCIÓN DEL ÁREA DE SALUD DE ESCUINTLA

La Dirección del Área de Salud de Escuintla (DASE), es una dependencia del Ministerio de Salud Pública y Asistencia Social, que tiene a su cargo, la coordinación del sistema de salud de la región departamental escuintleca, contando para esto con otras dependencias, a las que se les denomina “unidades”.

1.1. Las bodegas en la institución

Se describe la organización de las bodegas del Área de Salud escuintleca, para conocer mejor la lógica de distribución de los medicamentos.

1.1.1. Bodega central

En las instalaciones de las oficinas centrales de la DASE, se encuentra la bodega central de la región, la que almacena los insumos que se reparten en el área. Esta surte a las diferentes bodegas de las unidades municipales en el departamento escuintleco, siendo así, el eje de la distribución de medicamentos y otros artículos.

La adquisición de los insumos que llegan a la bodega, está a cargo del departamento de compras de la dirección, aunque puede darse el caso que estos provengan de alguna donación.

1.1.2. Bodegas distritales

Las bodegas en los municipios, son los almacenes de insumos para los centros de salud; cumplen con la tarea de abastecer a estos, según su unidad.

Estas bodegas dependen de la central del Área de Salud; aunque puede darse el caso que la bodega de un municipio, abastezca a otra de un municipio diferente, o bien, que una bodega reciba donaciones por parte de una persona o institución.

1.2. Control de los insumos en las bodegas de la DASE

Se explica el problema que representa la gestión de los insumos, al no contar con un sistema automatizado.

1.2.1. Situación actual

La información de los insumos en bodegas de la institución no se encuentra centralizada ni automatizada. Un jefe que desee información de la bodega, obtiene los datos de forma manual, o en hojas de cálculo, para luego realizar su trabajo respectivo, significando un retraso en sus quehaceres laborales.

La lentitud en el control de los insumos, se complica cuando se requiere la información del resto de bodegas del departamento, debido a que la falta de automatización de los procesos involucrados no permite la obtención de dicha información en tiempo real.

El control de ingresos y egresos de los insumos, es gestionado de forma manual, por medio de tarjetas cárdex, para realizar reportes, se debe consultar dichas tarjetas e ingresar la información a una hoja cálculo.

1.3. Marco conceptual

A continuación se describen los temas base sobre los que el proyecto de desarrollo fue realizado.

1.3.1. Sistema

El origen de la palabra, proviene del griego *systema.* Según el Diccionario de la RAE (Real Academia Española), un sistema es un “conjunto de reglas o principios sobre una materia racionalmente enlazados entre sí”, lo define además como un “conjunto de cosas que relacionadas entre sí ordenadamente, contribuyen a determinado objeto”.

Para el área de la informática, la RAE proporciona una definición, que complementa las dos anteriores, dejando claro el propósito de la construcción de un sistema informático, o bien, sistema experto, indicando que se trata de un “programa de ordenador o computadora que tiene capacidad para dar respuestas semejantes a las que daría un experto en la materia”.

Un programa de computadora, realiza tareas como lo haría cualquier experto en la materia que el sistema cubre, y para esto, es necesario conocer cuáles son las reglas bajo las que debe desenvolverse la aplicación, para asegurar que los elementos que conformarán el sistema proporcionen resultados fiables, de acuerdo con los objetivos esperados antes de su construcción.

1.3.2. Automatización

Se refiere al uso de sistemas de control y tecnologías de la información sin la intervención de un humano, o bien, una intervención parcial del mismo. Al automatizar, se buscan cumplir los siguientes objetivos:

- Mejorar la productividad de la empresa, reduciendo los costes de la producción y mejorando la calidad de la misma.

- Mejorar las condiciones de trabajo del personal, suprimiendo los trabajos penosos e incrementando la seguridad.

- Realizar las operaciones imposibles de controlar intelectual o manualmente.

- Mejorar la disponibilidad de los productos, pudiendo proveer las cantidades necesarias en el momento preciso.

- Simplificar el mantenimiento de forma que el operario no requiera grandes conocimientos para la manipulación del proceso productivo.

- Integrar la gestión y producción.

1.3.3. Ingeniería de software

Es descrita por Ian Sommerville (2005), como “una disciplina de la ingeniería, que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de este después que se utiliza.”

Por su parte, Benet Campderrich (2003) interpreta a la ingeniería de software, como un amplio campo que comprende “los métodos y las técnicas que se utilizan en el desarrollo profesional del software”.

1.3.4. Ciclo de vida del desarrollo de software

Se trata de un conjunto de actividades realizadas en la elaboración de un producto de software, que tienen el propósito de formalizar los procesos involucrados en dicha elaboración.

1.3.4.1. Toma de requerimientos

En esta actividad del ciclo, se recaba la información de los requisitos que debe satisfacer el software a desarrollar. La importancia en esta fase, radica en la necesidad de obtener la suficiente información, que servirá para el éxito en las siguientes fases.

Existen dos tipos de requerimientos: funcionales y no funcionales. Los requerimientos funcionales, son los que tienen que ver con lo que el cliente espera que haga el software; en tanto que los no funcionales, tienen que ver con la calidad del producto final.

1.3.4.2. Análisis

En esta actividad, se revisan los requerimientos recabados en la fase anterior, para analizarlos y determinar la necesidad que el cliente tiene.

Determinada la necesidad que se va a satisfacer, se analiza qué arquitectura es la apropiada para el software a desarrollar, así como el o los *frameworks* que se podrían utilizar, para la construcción de la solución.

1.3.4.3. Diseño

En la fase de diseño, se toma en cuenta lo analizado en la fase anterior, y se esboza la arquitectura del software que se va a desarrollar. De esta fase se obtienen los planos de la solución final, siendo así, una actividad muy importante para el éxito del proyecto.

El éxito del proyecto en el desarrollo del software dependerá de la solidez en el diseño planteado en esta actividad, el cual podría estar inmune por cambios radicales en los requisitos que se habían recabado en la primera actividad del ciclo, lo que significaría retraso en la entrega del producto y pérdidas económicas.

1.3.4.4. Desarrollo

En esta actividad, se codifica el software propuesto, con base en el diseño elaborado previamente.

1.3.4.5. Pruebas

Cada actividad en el ciclo de desarrollo del software, exige la calidad del producto, y la fase de pruebas es el punto en el que se audita el producto, para comprobar que la calidad del mismo se cumpla. En esta fase se asegura que el producto que se va a desarrollar, satisfaga tanto los requerimientos funcionales, como los no funcionales.

1.3.4.6. Implementación

En la actividad final en el ciclo de vida de desarrollo de software, se da por terminado el desarrollo de la solución propuesta y esta se instala para el cliente, según la arquitectura planeada en la fase de diseño.

1.3.5. Arquitectura del software

Representa toda la estructura de un sistema informático. Esta no solo abarca la parte funcional del sistema, sino que además la parte física, como la disposición del equipo en el que el sistema funciona.

En relación con lo anterior, Garlan y Shaw (1994) declaran sobre la arquitectura de software: "más allá de los algoritmos y estructuras de datos de la computación, el diseño y especificación de la estructura global del sistema es un nuevo tipo de problema".

La construcción de una arquitectura de software, es orientada por conjuntos de patrones establecidos con base en la experiencia del desarrollo de aplicaciones en los últimos años, existiendo diferentes soluciones, de acuerdo con los requerimientos que deba cumplir el sistema a implementar.

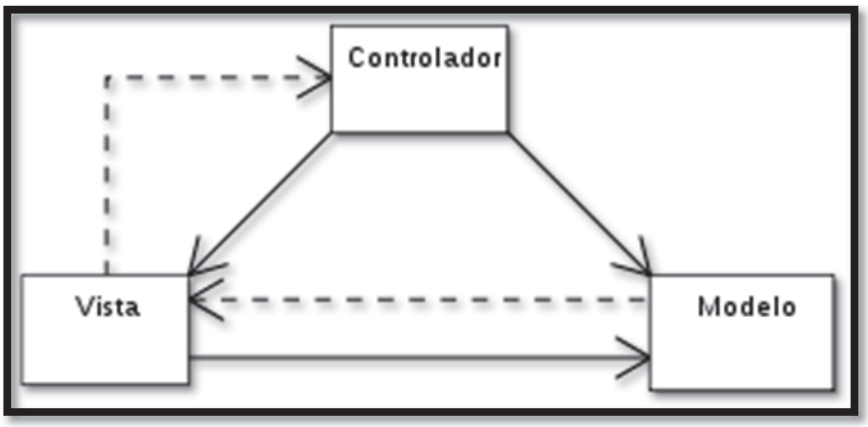
1.3.5.1. Patrón de arquitectura modelo vista controlador

Esta arquitectura, común en las aplicaciones web, separa las partes lógica (modelo) y de usuario (vista) en el sistema, permitiendo la comunicación de estas capas a través de peticiones realizadas por medio de componentes conocidos como controladores.

La capa de modelo, o de negocio, encapsula la información con la que opera el sistema. Por medio de este, se gestionan las reglas de negocio y los datos con los que el sistema trabaja.

La capa de la vista, presenta el modelo en un formato adecuado a través de una interfaz de usuario. Por último, es la capa del controlador la que Invoca peticiones al modelo y a la vista, a través de eventos, los cuales generalmente, son accionados por el usuario.

Figura 1. **Patrón modelo-vista-controlador**



Fuente: Wikipedia. http://tinyurl.com/7xofn2q. Consulta: 25 de enero de 2012.

1.3.6. Metodología de desarrollo de software

Una metodología, como conjunto de procedimientos, técnicas y herramientas para la consecución de un objetivo, permite la administración efectiva y eficaz de las actividades involucradas en determinado proyecto.

La metodología de desarrollo de software, por lo tanto, es útil para la administración de proyectos de software.

1.3.6.1. Modelo de Proceso Racional Unificado (RUP)

La metodología RUP (Rational Unified Process), trata de un conjunto de métodos adaptables al contexto y necesidades de cada organización. Se basa en 6 principios clave (Wikipedia):

- Adaptar el proceso: debido a la importancia de la interacción con el cliente, se debe adaptar el proceso de desarrollo a las necesidades del mismo.

- Equilibrar prioridades: se debe buscar un equilibrio donde las necesidades de todos los involucrados sean satisfechas.

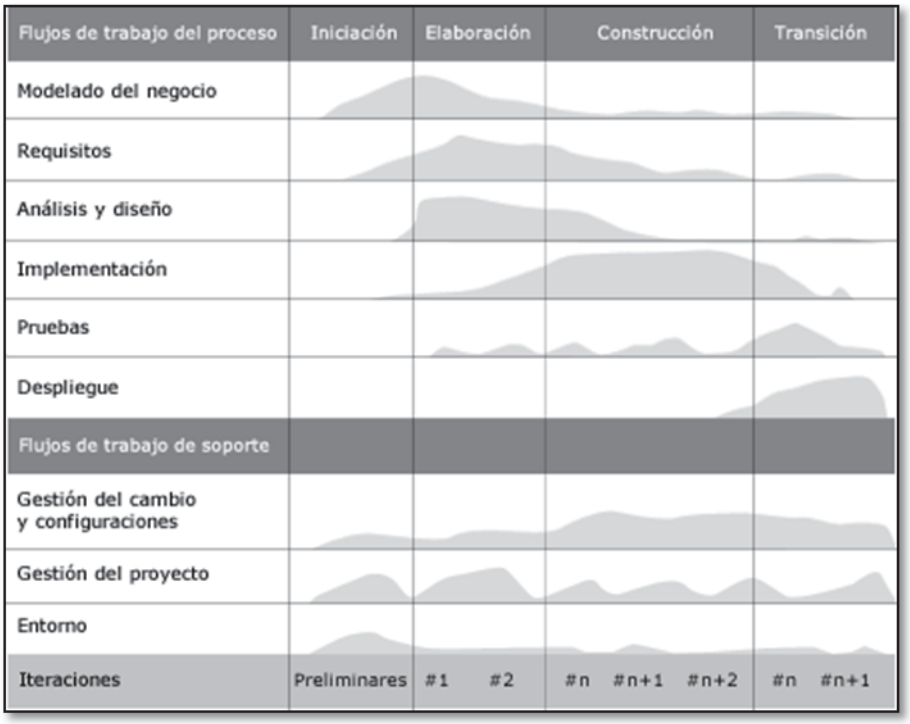
- Demostrar valor iterativamente: se trabajan los proyectos de forma iterativa, aunque sea internamente. De esta forma, en cada etapa se analiza la opinión de los inversores, la estabilidad y calidad del producto, además de refinar la dirección del proyecto y evaluar riesgos.

- Colaboración entre equipos: debe haber una comunicación fluida entre los integrantes de equipo, para coordinar los detalles concernientes en cada etapa del desarrollo del software.

- Elevar el nivel de abstracción: existen conceptos tales como patrones de diseño y *frameworks*, que permiten satisfacer las necesidades del cliente sin necesidad de codificar desde cero.

- Enfoque a la calidad: debe evaluarse la calidad del producto en todos los aspectos de su construcción, y no solo al finalizar cada iteración.

Figura 2. **Actividades y fases en la metodología RUP**



Fuente: Wikipedia. http://tinyurl.com/7fvdpsa. Consulta: 25 de enero de 2012.

2. MODELADO DEL NEGOCIO, TOMA DE REQUERIMIENTOS Y GESTIÓN DE LA CONFIGURACIÓN

2.1. Modelado del negocio

Modelar la lógica del negocio, es de utilidad para conocer los elementos que deben ser automatizados por el sistema.

2.1.1. Ingresos de insumos a las bodegas

La distribución de los insumos, inicia generalmente desde la bodega central de la Dirección del Área de Salud. Es desde esta bodega donde se realiza una distribución en cascada. Así, la bodega central reparte a las bodegas distritales, y estas a su vez, a bodegas de los centros de salud municipales. Puede darse el caso de que una bodega distrital, o la misma bodega central, reciba donación de insumos.

Un ingreso a la bodega central, generalmente se da cuando los proveedores entregan los insumos solicitados a través de órdenes de compra. En este caso, un operario recibe la entrega del proveedor, basándose en una orden de compra, y registra el ingreso en el cárdex respectivo.

Un proveedor es evaluado por el operario de la bodega, de acuerdo con la fecha de solicitud en la orden de compra y la fecha de entrega; la evaluación se registra en el documento: Reevaluación de proveedores del MSPAS.

Luego de registrado el ingreso, un operario asignado en bodega, debe llenar una constancia que detalle la información del mismo. Este detalle es autorizado por el jefe de bodega antes de ser enviado al gerente financiero de la institución, quien aprueba o desaprueba la constancia; de ser aprobada, el operario de bodega envía la información a contabilidad, adjuntando las facturas relacionadas, junto con otros documentos; de lo contrario, el formulario se modifica, según los errores encontrados y el procedimiento después de llenar la constancia, se repite.

Tras recibir la constancia, el departamento de contabilidad la revisa; en caso de que esté bien, termina el proceso del ingreso, de lo contrario, dicho departamento envía una boleta de rechazo en la que se indica el porqué de dicho rechazo, debiendo el operario realizar las correcciones necesarias y enviar la información corregida al departamento de contabilidad, donde repetirán la inspección de la constancia.

La bodega lleva el control de la entrega de esta papelería al departamento de contabilidad.

Como se mencionó, las entradas a una bodega que no es la central, se suceden cuando esta última abastece a la primera.

2.1.2. Salidas de insumos en las bodegas

Las salidas de insumos en la bodega central, se realizan cuando abastece una bodega de distrito bajo su jurisdicción. Para que esto suceda, las bodegas llenan un BRES, en el cual se detallan los insumos que solicitan. Cuando un operario en bodega central recibe el BRES, este despacha los insumos.

Las salidas en bodegas de los distritos, se realizan cuando abastecen a las instituciones que les corresponde.

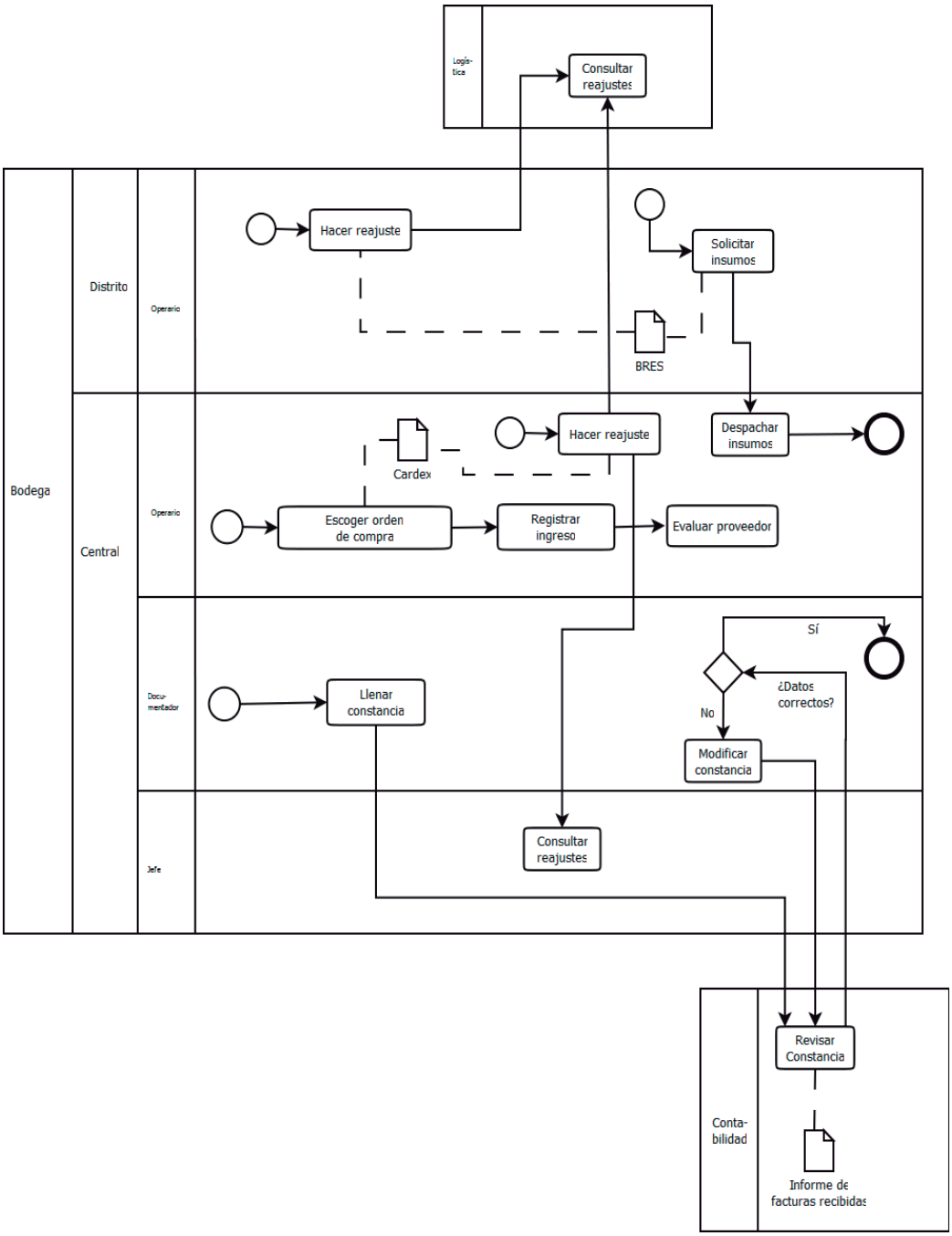
2.1.3. Control del inventario de insumos en las bodegas

Cada operario tiene a su cargo insumos agrupados por categorías, para llevar el control de los mismos, utilizan documentos llamados *cárdex*, en los cuales detallan los movimientos realizados en bodegas.

Una situación que puede darse en el control del inventario, es la de reajustar cantidades, esto es, restar o sumar unidades a la cantidad actual de determinado insumo. Cuando esto sucede, el operario que realiza el reajuste debe escribir una justificación del porqué del reajuste, para notificar al departamento de logística.

Al finalizar un mes, el operario debe notificar al departamento financiero, un inventario de lo que actualmente se encuentra en la bodega; de esta forma, la DASE mantiene un control para conocer qué se necesita adquirir.

Figura 3. **Modelo de negocio de administración de bodegas en la DASE**



Fuente: elaboración propia.

2.2. Toma de requerimientos

Para la toma de requerimientos, se realizaron visitas a los usuarios finales, así como a los interesados en el desarrollo de la aplicación.

Se tomó a la bodega central de la institución, como referencia para conocer los procesos involucrados en todas las bodegas del departamento escuintleco, además de conocer otros procesos, propios de dicha bodega.

Se entrevistó al personal que labora en el lugar, quienes explicaron las funciones que realizan, detallando los procesos involucrados para llevar a cabo sus tareas y acordando qué debería efectuar el sistema a desarrollar, para facilitar la realización de dichas tareas.

La información recabada en cada reunión, era analizada con la jefa de bodega, para confirmar los requerimientos, y especificarlos en una minuta que era firmada para dar como aceptados los mismos.

Así también, se obtuvieron muestras de reportes y papelería que manejan en la bodega, para conocer más a fondo, el formato de presentación de la información que manejan. Dicho material se utilizó únicamente para el desarrollo de la aplicación, no divulgándose nada de lo adquirido, tal y como se especifica en los objetivos del proyecto.

En caso de ser necesario, debido a la naturaleza de los procesos analizados, se recurría al asesor de la institución, como *stakeholder* del proyecto, con quien se corroboraban algunos aspectos de los requerimientos, que implicaban personal fuera de la bodega.

2.3. Gestión de la configuración del software

Para la gestión del desarrollo del software, se decidió trabajar con un control automático de las versiones. De esta forma, se aseguraba que cualquier cambio se realizara en una copia del trabajo, no afectara el producto en desarrollo.

Para este proyecto, se utilizó el software de código libre *subversion*[1], el cual se encuentra bajo licencia Apache/BSD, lo que significó cero costo para su instalación y posterior uso, satisfaciendo las necesidades para el desarrollo del proyecto. Este se encuentra disponible para varios sistemas operativos.

2.3.1. Establecimiento del entorno de gestión de la configuración

Se creó una estructura de directorios que sirvieran para albergar el repositorio de versiones del sistema a desarrollar; debido a la escasez de equipo, ya que este proyecto fue realizado por solo una persona; tanto el repositorio como el espacio de trabajo para el desarrollo, se encontraban en la misma máquina.

Uno de los propósitos de la gestión de la configuración por medio del control de versiones, es centralizar el resultado del trabajo de los desarrolladores en un solo espacio, por lo que si se trabaja con más de una máquina, se debe emplear una como servidor, en el que se integren los cambios al avanzar el desarrollo.

[1] http://subversion.apache.org/

2.3.2. Artículos de la configuración

Para gestionar los cambios realizados durante el desarrollo del proyecto, se definieron cuáles serían los componentes del sistema que estarían bajo el control del sistema de versiones, en el repositorio que almacenará la información de los cambios realizados.

El criterio para especificar los elementos de configuración, se tomó con base en la definición dada por Gerardo Díaz Carrillo en su blog: “Un elemento de configuración es un elemento físico o lógico, o cualquiera de sus subdivisiones discretas, que cumple una función determinada dentro de un producto”.

De acuerdo con lo anterior, se gestionaron los artefactos de RUP, cuyas versiones finales sirven como documentación técnica del producto final.

Así también, se gestionó en el repositorio de versiones, el código fuente del sistema desarrollado, para evitar algún problema conforme se avanzaba en el proyecto, lo que seguramente se daría al no controlar los cambios.

2.3.3. Control de cambios en la configuración

Para asegurar estabilidad en el producto en construcción, se estableció no guardar cambios al repositorio de trabajo, hasta comprobar que estos no entraran en conflicto con lo ya desarrollado.

Por medio de un *check-out* al repositorio, se creó un espacio de trabajo para desarrollo, el que consistía en una copia de la versión final, sirviendo dicho espacio para trabajar con libertad sobre el proyecto.

Teniendo el espacio de trabajo, se realizaban los cambios y se comprobaba por medio de pruebas, si los mismos satisfacían los requerimientos para los casos de uso, y luego de pasar las pruebas, se hacía un *commit* al servidor de versiones.

3. ANÁLISIS Y DISEÑO DEL SISTEMA

Con la actividad del análisis y diseño, se preparó la construcción del sistema a desarrollar, estudiando las necesidades que se debían satisfacer con el producto final terminado, tomando como referencia los requisitos recabados. El diseño del sistema se basó en dichas necesidades estudiadas, concibiéndose por medio del análisis anterior, la arquitectura del producto final, que satisficiera a usuarios finales y *stakeholders*.

3.1. Artefactos elaborados

Se detallan los artefactos elaborados en esta fase, de acuerdo a la metodología implementada.

3.2. Documento de arquitectura

Este documento es un artefacto en la fase de elaboración en RUP. Provee un extenso panorama de la arquitectura del sistema, usando diferentes vistas arquitecturales, para describir los distintos aspectos del sistema. Este se elaboró, orientado a servir como documentación técnica para la institución.

3.3. Definición de la arquitectura

Para definir la arquitectura del producto del software a desarrollar, se analizaron los requerimientos recabados en la fase de incepción, siendo estos,

en cualquier proyecto de desarrollo, la base para idear de qué manera se va a construir una aplicación.

Los requerimientos que influyeron en la decisión para la arquitectura adoptada, fueron los relacionados con la disponibilidad de la información y el uso de *frameworks* que no representaran costos de licenciamiento.

3.3.1. Modelado de casos de uso base

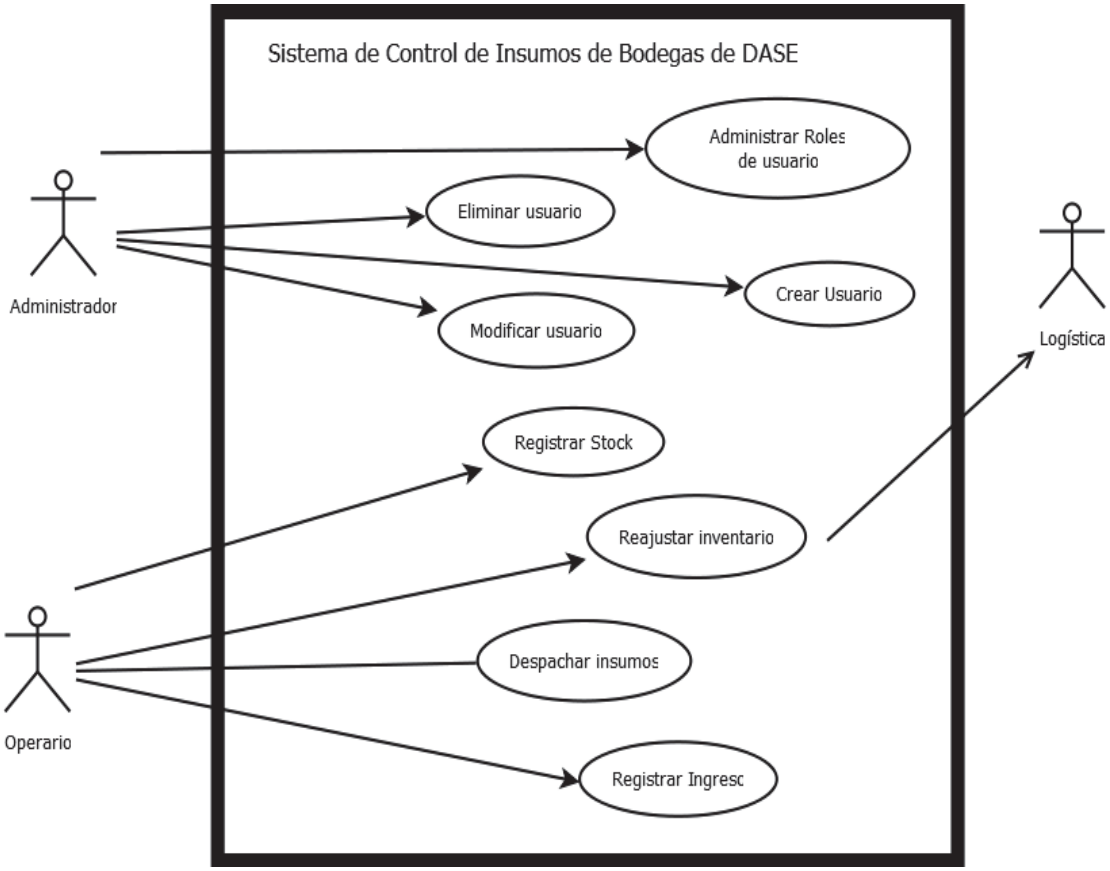
Inicialmente, se modelaron los casos de uso que servirían como base para el sistema a desarrollar. Para llegar a estos, se tomaron los requerimientos con mayor prioridad.

Los casos de uso base representan el núcleo de la aplicación, ya que consisten en las principales funcionalidades de esta, las que tienen más valor para el negocio y para el funcionamiento adecuado del sistema.

Los siguientes, fueron los casos de uso identificados como base:

- Crear usuario
- Modificar usuario
- Eliminar usuario
- Administrar roles de usuario
- Registrar stock
- Registrar ingreso
- Despachar insumos
- Reajustar inventario

Figura 4. **Diagrama de casos de uso base**



Fuente: elaboración propia.

3.4. Análisis y diseño de casos de uso

Para la siguiente iteración correspondiente a la fase de elaboración, se refinaron los casos de uso base, y se elaboró el resto de casos de uso para el sistema; producto de esto fue la vista de los mismos, que se incluyó en el documento de arquitectura.

La presentación de estos casos de uso, es dividida por módulos, siendo identificados cuatro:

- Administración del sistema
- Entradas a bodega
- Salidas a bodega
- Control de bodega

3.4.1. Módulo de administración del sistema

Este módulo consiste en las funcionalidades propias del sistema, que no están relacionadas directamente con las acciones que se realizan en las bodegas de la institución.

3.4.2. Módulo de entradas a bodega

Las funcionalidades de este módulo, conforman la automatización de las operaciones que se realizan en bodegas, para registrar entradas correspondientes a facturas por órdenes de compra; en el caso de la bodega central, por despacho de una bodega superior, por donaciones, y transferencias entre bodegas.

3.4.3. Módulo de salida de bodegas

En este se agrupan las funcionalidades del sistema que se relacionan con el registro de información de salidas de las bodegas, por medio de despachos.

3.4.4. Módulo de control de bodega

En este módulo se encuentran funcionalidades para el manejo, modificación y lectura de la información que se administra por medio del sistema.

3.5. Análisis y diseño de clases

Para la construcción de la aplicación, se diseñaron las clases que desempeñan las tareas de automatización en el sistema.

El análisis que llevó al diseño de estas clases, se basó en los casos de uso previamente realizados. Se tomaron las funcionalidades que estos describían, y se procedió a modelar el sistema, orientado a objetos.

3.6. Análisis y diseño del modelo de base de datos

Para la persistencia de la información que se administraría por medio del sistema, se construyó una base de datos relacional.

Para esta, se modeló un diagrama entidad-relación; siendo útil para la posterior construcción de la base de datos, y como parte de la documentación técnica que quedaría en poder de la institución.

3.7. Refinamiento de la arquitectura

Debido a que el ciclo de desarrollo del proyecto fue planeado de forma iterativa, la arquitectura sufrió cambios mientras se avanzaba.

Una de las motivaciones para refinar la arquitectura, fue el incremento de las funcionalidades, luego de desarrollados los casos de uso base, lo que dio lugar a revisar nuevamente los modelos y agregar información.

Otro factor que motivó a realizar el refinamiento se debió a la realización de cambios en el sistema, luego de realizar pruebas y comprobar que no satisfacían los requerimientos.

4. IMPLEMENTACIÓN, PRUEBAS Y DESPLIEGUE

Las siguientes fases a seguir en el desarrollo del sistema fueron: construcción y transición. Se terminó de escribir el código fuente de los diseños ideados en el proceso de análisis y diseño.

La implementación de los modelos se tradujo en el producto final esperado para cada iteración.

Las pruebas permitieron asegurarse que las funcionalidades desarrolladas satisfacían los requerimientos recabados, además de la calidad del producto. Posteriormente, el despliegue consistió en entregar el producto terminado y los artefactos que resultaron de las fases, además de capacitar a los usuarios finales.

4.1. Artefactos elaborados

Se detallan los artefactos elaborados en las fases de construcción y transición, de acuerdo con la metodología implementada.

4.1.1. El producto

El propósito de todo proyecto de desarrollo de software, también es tomado como artefacto en RUP (Malmö, University). Aparte de entregar a la Dirección del Área de Salud de Escuintla los archivos binarios del producto, construido y preparado para su funcionamiento inmediato al instalarlo, se entregó el código fuente del mismo, junto a la documentación respectiva.

4.1.2. Notas de lanzamiento

Este documento identifica las especificaciones técnicas generales del producto que se está entregando, incluye también *bugs* conocidos en la versión entregada.

4.1.3. Material de soporte para el usuario final

Fueron elaborados manuales de uso del sistema, que ayudarán en el aprendizaje del mismo, y servirán a las autoridades de la institución, como documentación para posterior capacitación de nuevos usuarios.

4.2. Implementación de componentes

Los componentes implementados basados en los diseños, se desarrollaron tomando cada caso de uso planteado y codificando las clases que servirían para realizar las funcionalidades que el caso de uso especificaba.

4.2.1. Provisión de retroalimentación al diseño

Al desarrollar las funcionalidades, se dieron casos en los que se notaron errores en el diseño, ya sea por no tomar en cuenta operaciones o atributos en una clase, o porque se debieron agregar más clases.

Estos cambios se debían reflejar en el diseño, para llevar documentada la vista lógica de la aplicación, por lo que se modificaban los modelos cuando era necesario.

4.2.2. IDE y *frameworks* utilizados

Para trabajar el código fuente de la aplicación, se utilizó el IDE *Netbeans* 6.8. Este es gratuito y de código abierto, provee las herramientas necesarias para crear aplicaciones profesionales para escritorio, web, y móviles, con la plataforma *Java*, así como C/C++, PHP, *Javascript* y *Groovy* (*Netbeans*).

El *framework* utilizado fue JSF2, junto a *Primefaces* 2, para los componentes de la interfaz gráfica.

La tecnología JSF establece el estándar para construir interfaces de usuario del lado del servidor (Oracle); en tanto que Primefaces es una librería de código abierto para JSF, que contiene gran variedad de componentes con soporte AJAX.

4.3. Pruebas

Las pruebas se realizaron cada vez que se desarrollaba un caso de uso, y al finalizar el sistema.

Al superar las pruebas, el código fuente se integraba a la versión estable del sistema, contenida en el repositorio de versiones.

4.3.1. Pruebas de integración

Estas se realizaron para asegurarse que las funcionalidades implementadas o cambios realizados en las actividades de desarrollo, no afectaran el funcionamiento de implementaciones anteriores del código.

La práctica regular de esta actividad, evita integrar las nuevas funcionalidades y cambios sin haber sido probados y asegura un sistema estable al final del desarrollo.

4.3.2. Casos de prueba

Estos se elaboraron para comprobar que el sistema cumplía con los requerimientos recabados al inicio del proyecto, al pasar las pruebas, se daba por aceptado el sistema; de resultar errónea alguna de las pruebas, se realizaban los cambios necesarios y se volvía a realizar el caso de prueba.

Los casos se elaboraron para que los usuarios finales hicieran las pruebas, por lo que se enfocaron en los casos de uso que especificaban las funcionalidades que satisfacían los requerimientos del cliente.

De esta forma, se aseguraba que el producto final cumplía con las expectativas de los interesados, dando la oportunidad a los usuarios finales para que conocieran el producto.

Para esto, se instaló en la institución un entorno de pruebas, luego de finalizado el desarrollo del sistema, y se les proporcionaron los casos de prueba.

4.4. Instalación del sistema en el entorno de producción

Finalizado el desarrollo y las pruebas, se procedió a instalar el producto en el servidor de la institución.

4.5. Capacitación y materiales de soporte

Se planeó realizar reuniones en las que se debería explicar al usuario el uso del sistema en escenarios reales, mientras este trabajaba.

Además, como ya se mencionó, para mejorar el aprendizaje y asegurar que el nuevo personal se capacite por medio de documentación, se elaboraron manuales que explican el uso de la aplicación.

CONCLUSIONES

1. El uso del sistema desarrollado, mejora la administración de los recursos en bodega, permitiendo acceso en tiempo real a la información de existencias disponibles.

2. Además del acceso inmediato a la información, el sistema permite llevar el control automático de los movimientos en inventario para cada insumo, facilitando al personal de bodega generar los reportes correspondientes, ahorrando tiempo.

3. El desarrollo de los artefactos, de acuerdo con la metodología RUP, permite obtener una documentación técnica del producto, que servirá de apoyo a la institución.

4. El desarrollo del producto como aplicación web, con tecnología java, cuyo uso no implica un desembolso en licencias, permite a la institución implementarlo, para ser utilizado desde las demás bodegas bajo la Dirección del Área de Salud de Escuintla, centralizando la información.

RECOMENDACIONES

1. Según requerimientos recabados, la información de órdenes de compra en el sistema, depende de la base de datos del Departamento de Compras, por lo que deberá mantenerse actualizada, para que la aplicación muestre las órdenes que se deben despachar en bodega.

2. El usuario por defecto de la aplicación, posee permisos de administración; por seguridad, debe cambiarse la contraseña original de este.

3. Proporcionar los manuales de usuario a quienes vayan a utilizar la aplicación. El manual que se proporcione dependerá del rol que dicho usuario vaya a tener en el sistema.

4. No perder los documentos que se proporcionen junto con el producto; estos pueden ser de utilidad en cualquier situación en la que terceros deban intervenir el código fuente.

BIBLIOGRAFÍA

1. CAMPDERRICH FALGUERAS, Benet. *Ingeniería de software.* Barcelona, España: UOC, 2003. 320 p. ISBN-84-8318-997-6.

2. DÍAZ, Gerardo. Apuntes sobre Gestión de la Configuración (CPM) y Product Lifecycle Management (PLM). *Fundamentos de Gestión de la Configuración (2).* [en línea] <http://cmyplm.wordpress.com/>. [Consulta: 6 de agostode 2013].

3. GARLAN, David; SHAW, Mary. *An introduction to Software Architecture*. [en línea] Carnegie Mellon University, 1994. <http://repository.cmu.edu/cgi/viewcontent.cgi?article=1720&context=compsci>. [Consulta: 10 de noviembre de 2011].

4. GRUNDMANN, Gesa; STAHL, Joachim. *Como la sal en la sopa: conceptos, métodos y técnicas para profesionalizar el trabajo en las organizaciones de desarrollo.* 3a ed. Quito, Ecuador: Abya Yala, 2002. 300 p. ISBN-9978-22-236-7.

5. *Rational Unified Process (Overview).* [en línea] Malmö (Suecia): Malmö University. <http://www.ts.mah.se/RUP/RationalUnifiedProcess/>. [Consulta: 10 de noviembre de 2011].

6. *Netbeans IDE - The smarter and faster way to code*. Netbeans [en línea] <http://netbeans.org/features/index.html>. [Consulta: 28 de noviembre de 2011].

7. *Java Server Faces* Technology. Oracle. [en línea] <http://www.oracle.com/technetwork/java/javaee/javaserverfaces139869.html>.[Consulta: 28 de noviembre de 2011].

8. Real Academia Española. [en línea] *Sistema*. <http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=sistema>. [Consulta: 4 de abril de 2011].

9. SOMMERVILLE, Ian. *Ingeniería del software.* 7a ed. Madrid, España: Pearson Educación, 2005. 691 p. ISBN-84-7829-074-5.

10. *Artifact: Product* [en línea] Malmö University. Suecia. <http://www.ts.mah.se/RUP/RationalUnifiedProcess/process/artifact/ar_pduct.htm>. [Consulta: 29 de noviembre de 2011].

11. Wikipedia. *Proceso unificado de rational*. [en línea] <http://es.wikipedia.org/wiki/RUP>. [Consulta: 8 de julio de 2011].

12. _____. *Modelo Vista Controlador*. [en línea] <http://es.wikipedia.org/wiki/Modelo_Vista_Controlador>. [Consulta: 3 de abril de 2011].

13. _____. *Arquitectura de software*. [en línea] <http://es.wikipedia.org/wiki/Arquitectura_de_software>. [Consulta: 8 de julio de 2011].

APÉNDICES

Apéndice 1. **Notas de lanzamiento**

El documento de notas de lanzamiento, sirve de presentación de la versión final del sistema.

A continuación se presenta el documento que se elaboró para este proyecto.

INTRODUCCIÓN

- Exclusión de garantía: el desarrollador de este producto, deja de brindar soporte sin cobro, después de 3 meses de finalizado el proyecto. Dicho período, permitirá corregir errores en el producto entregado, suponiendo que existieran, y no se añadirán nuevas funcionalidades. El desuso de la aplicación, luego de instalada en el servidor de la institución, no aplaza los meses de soporte.

- Propósito: el propósito de este documento de notas de lanzamiento, es describir las características principales del sistema de bodegas de la Dirección del Área de Salud de Escuintla, así como problemas conocidos en el uso del mismo.

- Perspectiva general: en este documento, se conocerá la utilidad que el sistema de bodegas de la Dirección del Área de Salud de Escuintla le proporciona, así como las especificaciones técnicas mininas para que este

funcione satisfactoriamente. Además, conocerá detalles de problemas menores que se han encontrado, y que no fueron resueltos, debido a dependencia de la tecnología usada; estos problemas pueden ser evadidos, por lo que no representan riesgo en el desempeño de la aplicación.

A. Acerca de este lanzamiento

Este producto fue desarrollado a la medida, para ser usado exclusivamente en las bodegas de la Dirección del Área de Salud de Escuintla, limitándose su desempeño a los procesos que realizan en las bodegas de dicha institución, y que fueron tomados en cuenta, de acuerdo con los requerimientos presentados por los interesados en la aplicación. El uso del sistema, le permite almacenar la existencia de los insumos que actualmente se encuentren en bodega. Esta información es esencial para el manejo de la futura información que se produzca con el uso del sistema.

Las entradas a la bodega central son almacenadas cuando un usuario registra información de insumos entregados por proveedores, por medio de una orden de compra, o al recibir una donación. En el caso de bodegas distritales, aparte de entradas por donación, el sistema registra como entradas los despachos que dichas bodegas reciban de aquellas que son sus proveedoras. Para solicitar un despacho, la bodega que pide, debe ingresar previamente el pedido en el sistema.

Por medio del sistema, un usuario podrá ver de manera electrónica el detalle de movimientos para cada insumo a su cargo, como sucedería físicamente con el manejo de una tarjeta cárdex, permitiéndole imprimir esa información sobre una tarjeta movible. Además, podrá generar reportes sobre las existencias disponibles e ingresos de insumos.

El control de bodegas permite reajustar información de los inventarios de cada una de ellas, permitiendo al jefe de logística estar al tanto, pudiendo revisar el detalle de dichos reajustes y las justificaciones que el sistema exige a un usuario para realizar tal acción.

B. Productos compatibles

Este producto ha sido probado en los siguientes sistemas operativos:

- Windows 7
- Linux Fedora 15

El funcionamiento del sistema está condicionado por la instalación de los siguientes productos:

- Glassfish, versión 3
- Mysql 5.1

Este producto ha sido probado con el explorador Firefox 7, por lo que se recomienda que las terminales que accedan al sistema tengan instalado dicho explorador web con la misma versión o superior.

Los requerimientos mínimos para el funcionamiento óptimo del sistema son:

- Procesador Intel Celeron 430
- 1 GB de RAM
- 100 MB de espacio de disco duro

C. *Bugs* conocidos y limitaciones:

- Problemas con librería Mojarra 2: el sistema fue desarrollado con el *framework* JSF 2, junto con Primefaces 2.2, para el uso de componentes de la interfaz web; la librería Mojarra 2 presenta problemas con dicha versión de Primefaces, mostrando un mensaje de advertencia al usar la aplicación, indicando que la declaración de un componente no fue hecha dentro de un *form*. Esto se soluciona instalando un lanzamiento anterior o reciente de la librería Mojarra.

- Comportamiento extraño de las listas desplegables: en ocasiones, al desplegar un *combobox*; la lista se "despega" del componente, provocando que se dificulte la navegación en dicha lista. Se soluciona presionando el botón direccional hacia abajo.

Apéndice 2. **Casos de prueba**

Los casos de prueba se elaboraron para que los usuarios finales hicieran sus pruebas, y comprobar que el sistema satisfacía los requerimientos. A continuación se detalla el formato utilizado para los casos de prueba; en este ejemplo, para comprobar que los reajustes se realizan correctamente, deben seguirse los siguientes pasos:

- Paso 1: ingresar como usuario con rol de operario de bodega;
- Paso 2: dar clic en la pestaña “cárdex” en “Operaciones en bodega”;
- Paso 3: dar clic en la opción “Reajustar inventario”;
- Paso 4: escoger el tipo de reajuste entre “Sumar cantidad” o “Restar cantidad”;
- Paso 5: escoger un rubro del listado desplegable (se muestran rubros que le fueron asignados previamente);
- Paso 6: buscar y escoger un insumo (se le muestra insumos que se hayan registrado previamente en inventario);
- Paso 7: buscar y escoger el número de lote;
- Paso 8: escribir la cantidad a reajustar;
- Paso 9: escribir una justificación para el reajuste;
- Paso 10: presione el botón “Aceptar”.

Aceptado: ___
Falló: ___ Descripción del error:

Fuente: elaboración propia.

Apéndice 3. **Documento de arquitectura del software**

El documento de arquitectura fue elaborado durante el desarrollo del proyecto, pensado para ser la documentación técnica que dé soporte cuando se necesite conocer la estructura de la aplicación. Se presenta en las siguientes páginas el documento que resultó de elaborar este artefacto de RUP.

A. Introducción

- **Propósito:** el propósito de este documento es mostrar la estructura del software desarrollado, y que sirva para futuros cambios o nuevas implementaciones. Se debe considerar leer este documento antes de realizar cualquier modificación al sistema, o bien, al querer ampliar las funcionalidades que lo conforman. Se le presentarán los objetivos de la arquitectura, y se detallan las siguientes vistas, que representan su estructura

    - Vista de casos de uso
    - Vista lógica
    - Vista de despliegue
    - Vista de datos

- **Alcance:** se cubre la perspectiva lógica de la aplicación, mostrando la estructura de almacenamiento de la información en una base de dato. Se conocerá el funcionamiento del producto, a través de los diagramas que resultaron del análisis de los requerimientos que dieron origen a la necesidad de desarrollar este sistema.

B. Representación arquitectónica

El sistema se construyó con base en el patrón Modelo Vista Controlador (MVC), que permite bajo acoplamiento entre los componentes que conforman la funcionalidad de la lógica de negocio y de datos, con la interfaz de usuario, gestionados a través de controladores. De esta manera, si se decidiera cambiar aspectos a la parte del modelo, estos se pueden realizar sin afectar la estructura que administra la interfaz de usuario, y viceversa.

El código fuente, fue dividido en paquetes, para delimitar la arquitectura a nivel de 3 capas del sistema; los paquetes principales son: Negocio para la capa del modelo y Controlador, que contiene las clases controlador; el directorio "Web", contenido en el código fuente, y las vistas del sistema, a las cuales se accede por medio de los controladores.

- **Vista de casos de uso:** en esta vista, se describen los casos de uso que sirvieron para idear la funcionalidad de la aplicación; estos parten de los requerimientos recabados. Se modelaron los casos de uso separados por los módulos que conforman la aplicación, para simplificar la presentación de las funcionalidades.

- **Vista lógica:** para esta vista se modelaron las clases que se codificaron para el funcionamiento de la aplicación. Estas se muestran según paquetes.

- **Vista de despliegue:** esta vista muestra cómo debería ser la implementación de la aplicación para su funcionamiento, describiendo el entorno en el que correrá la misma y cómo se accederá a esta.

- **Vista de datos:** esta vista muestra la estructura en la que persiste la información que se maneja en el sistema, siendo la parte esencial para el manejo de dicha información. El modelo para esta vista, se muestra por medio de un diagrama Entidad-Relación.

C. Objetivos y restricciones de la arquitectura

- **Arquitectura basada en tecnología web:** la necesidad de los interesados en la institución, era la de poder tener en tiempo real, la información del manejo de insumos en las bodegas de la Dirección del Área de Salud de Escuintla. La mejor manera para lograr esto, fue ideando una arquitectura orientada como aplicación web, lo que permite el acceso desde cualquier computadora que tenga conexión a internet, no debiendo instalarse aplicaciones de escritorio en cada terminal.

- **Sistema desarrollado con tecnología JAVA:** para evitar costos en compra de licencia por uso de un lenguaje de programación, se optó por desarrollar la aplicación con tecnología JAVA, por lo que la misma funciona sobre un servidor Glassfish, versión 3.

- **Conexión a base de datos del departamento financiero:** debido a que el sistema debe acceder a información de las órdenes de compra que se generan desde la aplicación del departamento financiero, se desarrollaron disparadores que actúan sobre tablas DASF0071, DASF008 y DASF004 de la base de datos “dase”. Los disparadores están identificados como:
  - registraOrdenCompra
  - actualizaOrdenCompra

- eliminaOrdenCompra
- registraProveedor
- actualizaProveedor
- eliminaProveedor
- registraInsumo
- actualizaInsumo
- eliminaDetalleOrdenCompra
- actualizaDetalleOrdenCompra
- eliminaInsumo

Por eso la base de datos “BodegaDASE”, correspondiente a esta aplicación, debe encontrarse en el mismo manejador de base de datos donde se encuentre la del departamento financiero.

- **Recurso de conexión a base de datos desde el servidor:** para que el sistema funcione correctamente, debe estar configurado desde el servidor *glassfish*, la conexión a la base de datos del sistema. Debe asegurarse que en las librerías del servidor, se encuentre *mysqlconnector* para java. Ingresar desde un explorador web, escribiendo en la barra de direcciones: http://<dirección del servidor>*:4848*. En la sección de tareas comunes, se debe buscar y expandir el nodo “Recursos”, luego expandir el nodo “JDBC” y escoger “Conjunto de conexiones JDBC”. Presionar el botón “Nuevo” e ingresar la siguiente información**:**

  - Nombre del conjunto: mysql_BodegaDASE_rootpool
  - Tipo de recurso: javax.sql.DataSource
  - En propiedades del conjunto de conexión ingresar:
    - driverClass: *com.mysql.jdbc.Driver*
    - user: el usuario que tiene acceso a la base de datos

- password: escribir el *password* del usuario
- portNumber: el puerto para *mysql*, por defecto es 3306
- databaseName: BodegaDASE
- servername: la dirección ip del servidor en el que se encuentra la base de datos

Luego de creado el conjunto de conexiones, debe hacerse lo mismo con el recurso JDBC, por medio del cual se conectará el sistema a la base de datos. Escoger el nodo “Recursos JDBC” y presionar el botón “Nuevo”, como nombre del recurso escribir: “*jdbc/DASE*”, y seleccionar el conjunto de conexiones creado antes.

- **Sobre el usuario por defecto:** el usuario por defecto tiene el nombre de “admin”, este se crea automáticamente en la base de datos, la primera vez que se inicia la aplicación.

D. Vista de casos de uso

- **Administrar roles de usuario:** un usuario con rol de administrador puede asignarle y/o desasignarle roles a los demás usuarios del sistema. Los roles definen el acceso que los usuarios tienen a las funcionalidades del sistema.

- **Crear usuario:** un administrador puede crear un usuario que tendrá acceso a las funcionalidades del sistema, de acuerdo con sus roles.

- **Modificar usuario:** un administrador puede modificar la información de un usuario previamente creado en el sistema.

- **Eliminar usuario:** un administrador puede eliminar a un usuario

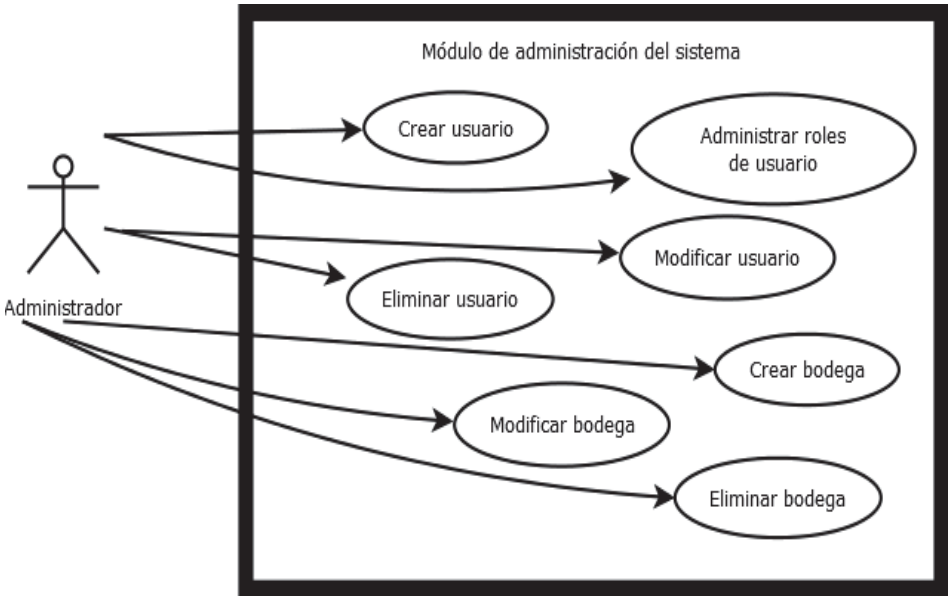
previamente creado en el sistema.

- **Crear bodega:** un administrador puede crear una bodega; las bodegas son necesarias para asignarlas a los usuarios creados, de acuerdo con el municipio en el que labore.

- **Modificar bodega:** el administrador puede editar información de las bodegas registradas en el sistema.

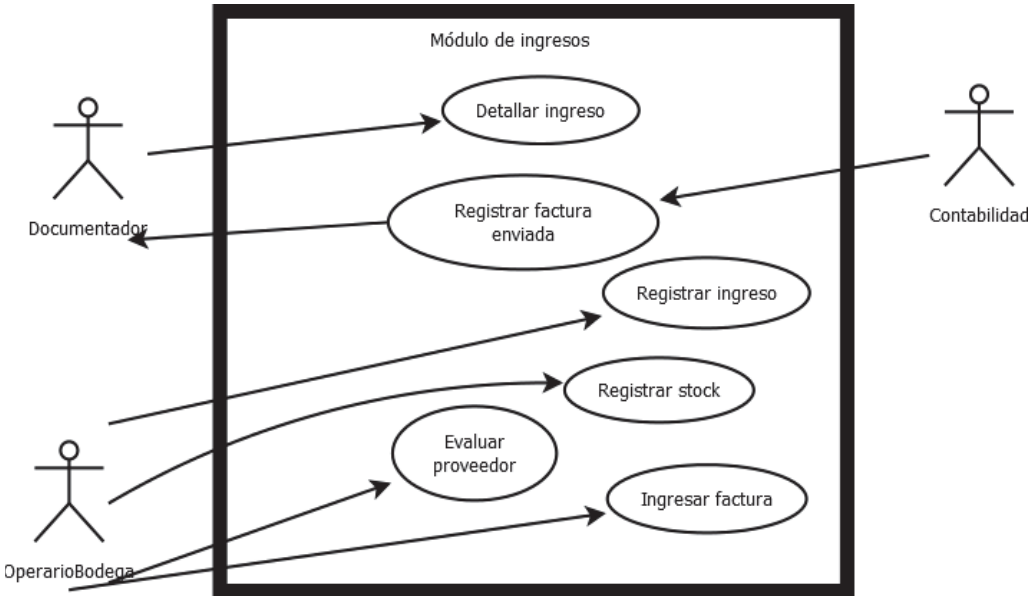
- **Eliminar bodega:** el usuario puede eliminar una bodega registrada en el sistema, siempre que no existan usuarios asignados a esta, o información de insumos que referencien a la bodega.

Apéndice 3a. **Módulo de administración del sistema**



Fuente: elaboración propia.

Apéndice 3b. **Módulo de entradas a bodega**



Fuente: elaboración propia.

- **Detallar ingreso:** un documentador puede detallar el ingreso de insumos por medio de factura o donación, e imprimir una constancia.

- **Registrar factura enviada:** un usuario con rol contabilidad deberá indicar en el sistema, cuando reciba de un documentador las facturas correspondientes a un detalle de ingreso.

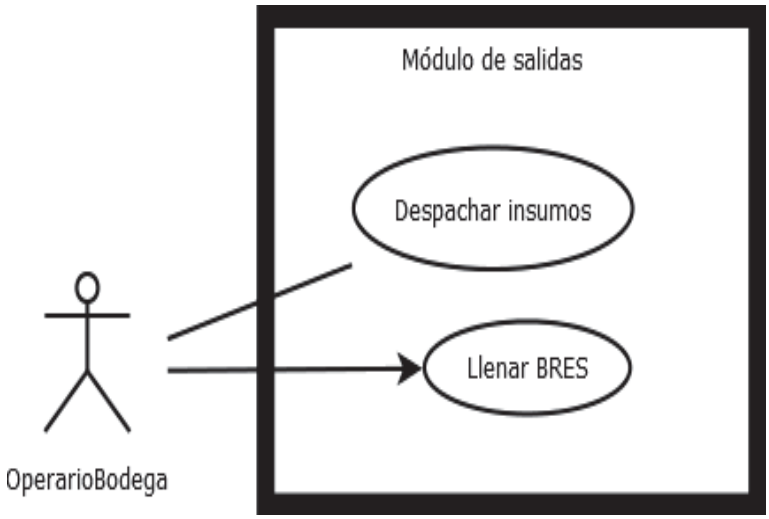
- **Registrar ingreso:** un operario de bodega registrará el ingreso a la misma, cuando deba recibir un pedido a un proveedor, o una entrada por donación.

- **Registrar stock:** el operario debe ingresar las existencias de los insumos que actualmente se encuentran en la bodega.

- **Evaluar proveedor:** un operario evalúa a un proveedor, luego de recibir un pedido hecho por medio de una orden de compra.

- **Ingresar factura:** el operario deberá registrar en el sistema una factura, cuando detalle una entrada por una entrega de proveedor.

Apéndice 3c. **Módulo de salida de bodegas**

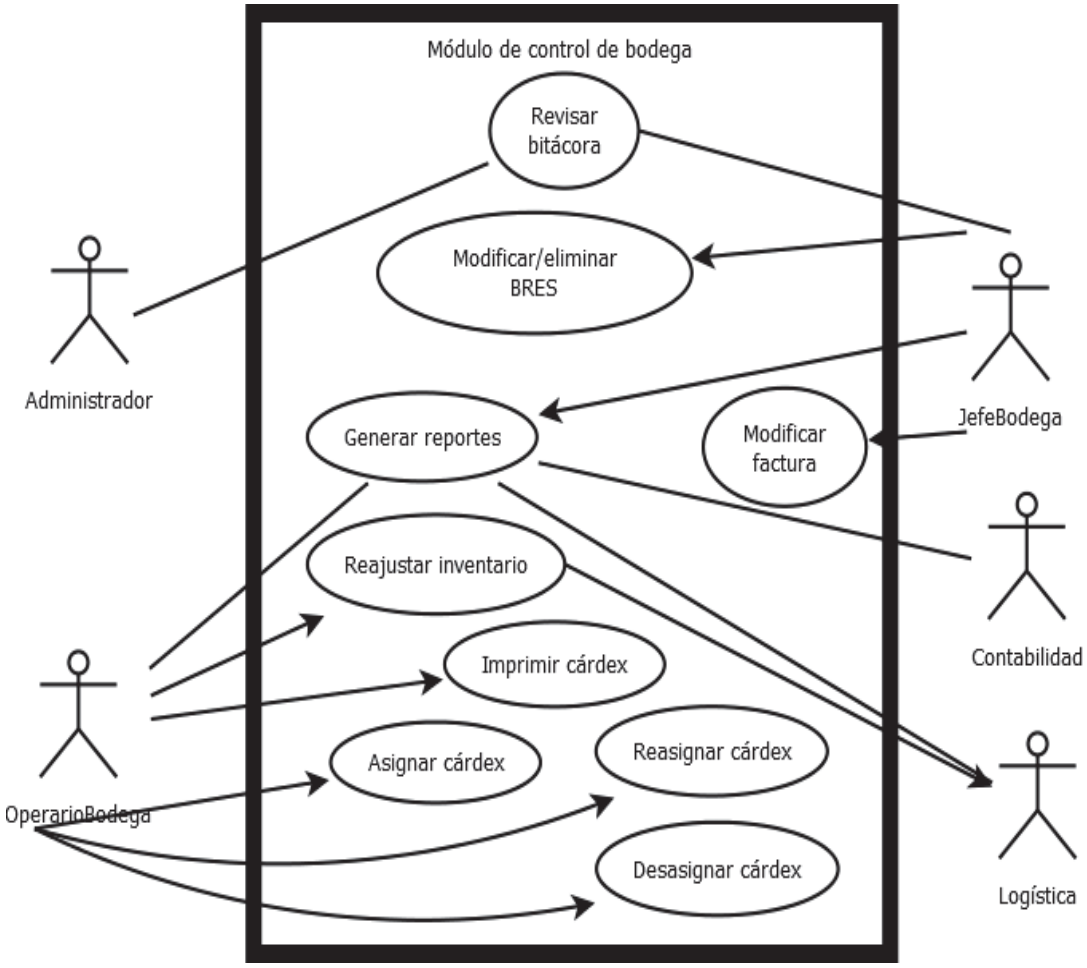


Fuente: elaboración propia.

- **Despachar insumos:** el operario de bodega despacha insumos a otra bodega.

- **Llenar BRES:** el operario de bodega llena un pedido para solicitar insumos a una bodega proveedora.

Apéndice 3d. **Módulo de control de bodegas**



Fuente: elaboración propia.

- **Revisar bitácora:** un administrador y un jefe de bodega pueden revisar detalles de acciones realizadas en el sistema por los usuarios. El jefe de bodega únicamente puede ver la información de los usuarios asignados a la bodega a su cargo.

- **Modificar/eliminar BRES:** un jefe de bodega puede editar información de un pedido de insumos registrado en el sistema, o bien, eliminar la información si aún no se han registrado despachos para el pedido.

- **Modificar factura:** un jefe de bodega puede editar la información de una factura.

- **Generar reportes:** los usuarios de bodega pueden generar reportes basados en los movimientos de sus respectivos inventarios.

- **Reajustar inventario:** un operario puede reajustar información de las existencias de insumos.

- **Imprimir cárdex:** un operario de bodega puede imprimir la información mensual de su cárdex, correspondiente a los rubros que tiene a su cargo.

- **Asignar cárdex:** un operario debe asignar el número de cárdex de cada insumo, para que el sistema lleve un control automático.

- **Reasignar cárdex:** un operario puede reasignar cárdex a un insumo que ya tenía asignado un número.

- **Desasignar cárdex:** un operario puede desasignar cárdex a un insumo, si aún no hay movimientos registrados en este.

E. Vista lógica

- **Perspectiva general:** el código fuente del sistema, se compone principalmente de los paquetes Negocio y Controlador, los cuales

contienen las principales funcionalidades del sistema. La funcionalidad gráfica de la aplicación, se conserva en la carpeta Web, que contiene las páginas que conforman la interfaz de usuario. Adicionalmente, hay paquetes complementarios, delimitados según la lógica de funcionamiento de la aplicación. A continuación se enumeran todos los paquetes que componen el sistema:

- **Beans:** contiene clases que aportan funcionalidad para la persistencia de datos por medio del mapeo de la base de datos en clases java.

- **Controlador:** paquete que contiene las clases que sirven como controladores entre la vista de la aplicación y la lógica del negocio que esta desempeña.

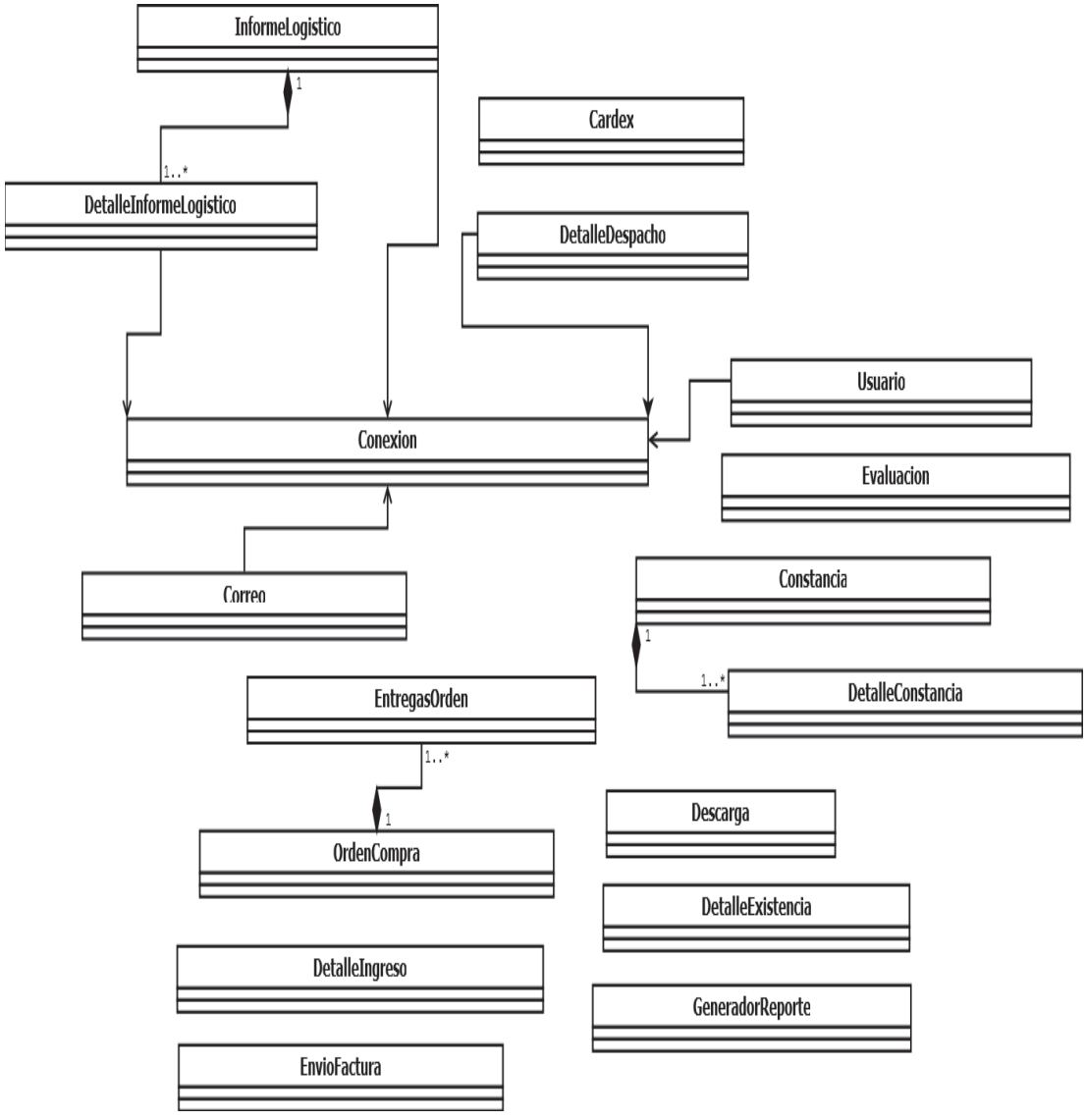
- **Controlador.util:** es un paquete dentro del paquete Controlador, provee funcionalidad extra a las clases que sirven como controladores.

- **Negocio:** este paquete contiene clases que poseen la lógica del negocio, es decir, la automatización de los procesos de las bodegas.

- **Negocio.mapeo:** paquete dentro del paquete Negocio; encapsula la lógica de datos de la aplicación, conteniendo las clases en las que se mapea la base de datos.

- ***Resources:*** no contiene clases, guarda el *script* de creación de la base de datos del sistema y el archivo *Bundle.properties,* útil para centralizar los mensajes que el sistema mostrará.

Apéndice 3e**.** **Diseño de paquetes significativos en la arquitectura (Negocio)**

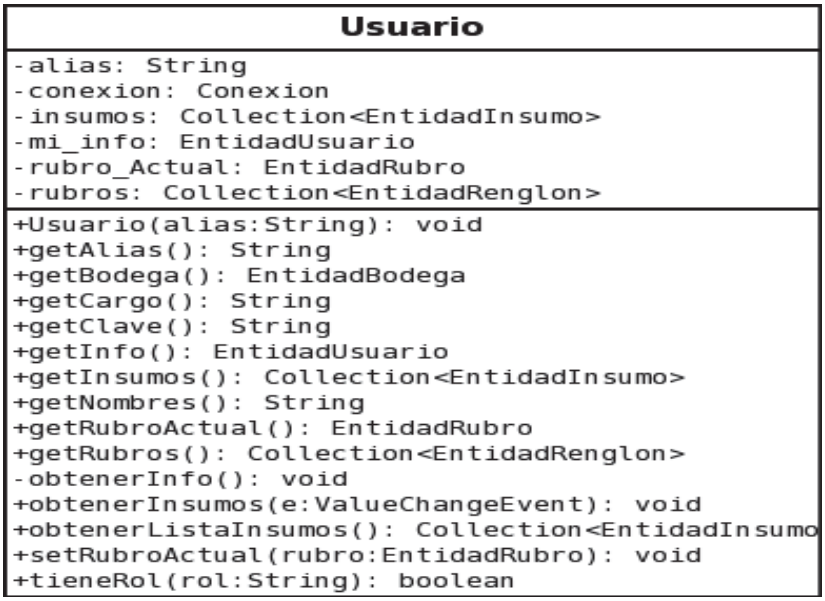


Fuente: elaboración propia.

F. Descripción de clases

- **Usuario:** clase que posee información del usuario que ingresa al sistema, tal como los roles que este tiene y su alias. El atributo más significativo para esta clase, es mi_info, el cual posee la información del usuario, obtenida de la base de datos, y del que se obtiene la información para el resto de atributos que tienen que ver con dicha información.

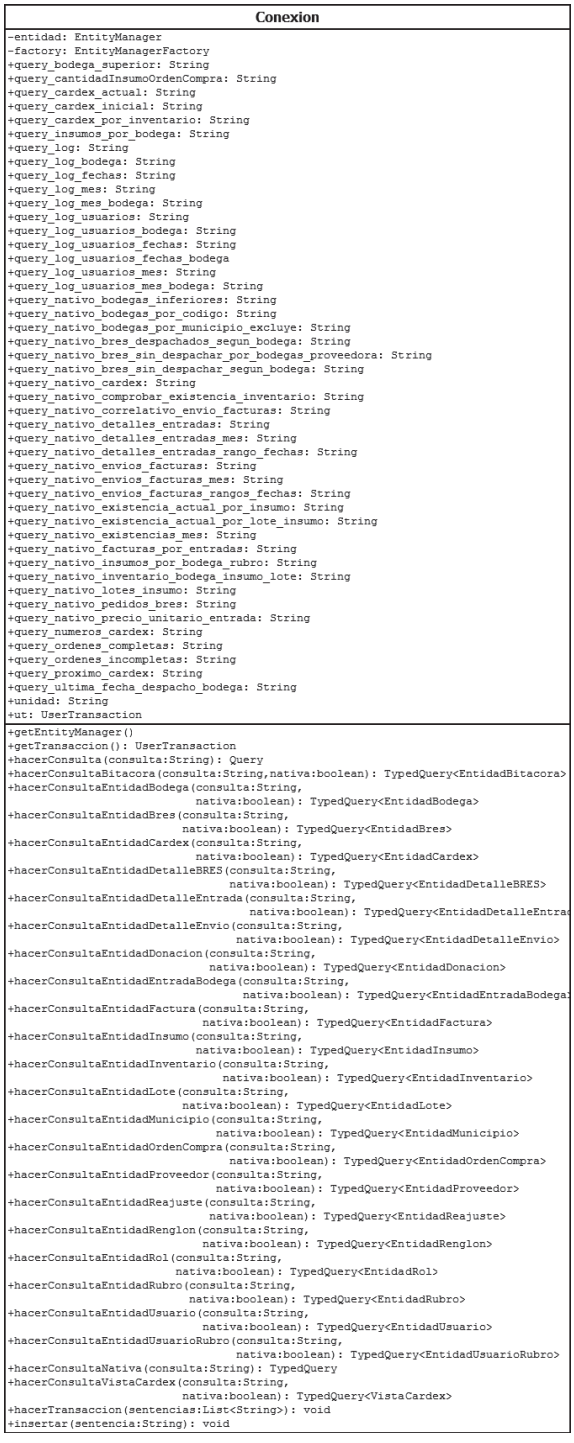
Apéndice 3f. **Usuario**



Fuente: elaboración propia.

- **Conexión:** esta clase proporciona funcionalidad de acceso a la información contenida en la base de datos. Centraliza acciones DML, como la inserción o actualización nativa de datos y las consultas a la base de datos, mapeándolas a su respectiva clase, o devolviendo un valor, en caso de consultas nativas.

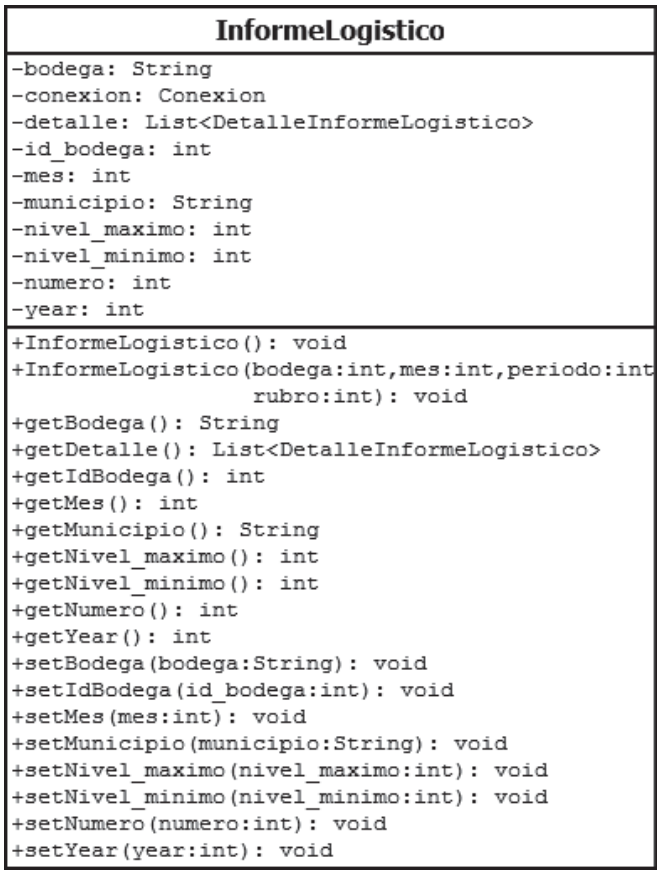
Apéndice 3g. **Conexión**



Fuente: elaboración propia.

- **InformeLogístico:** clase para la preparación del informe logístico de la bodega central. Contiene el encabezado de dicho informe.

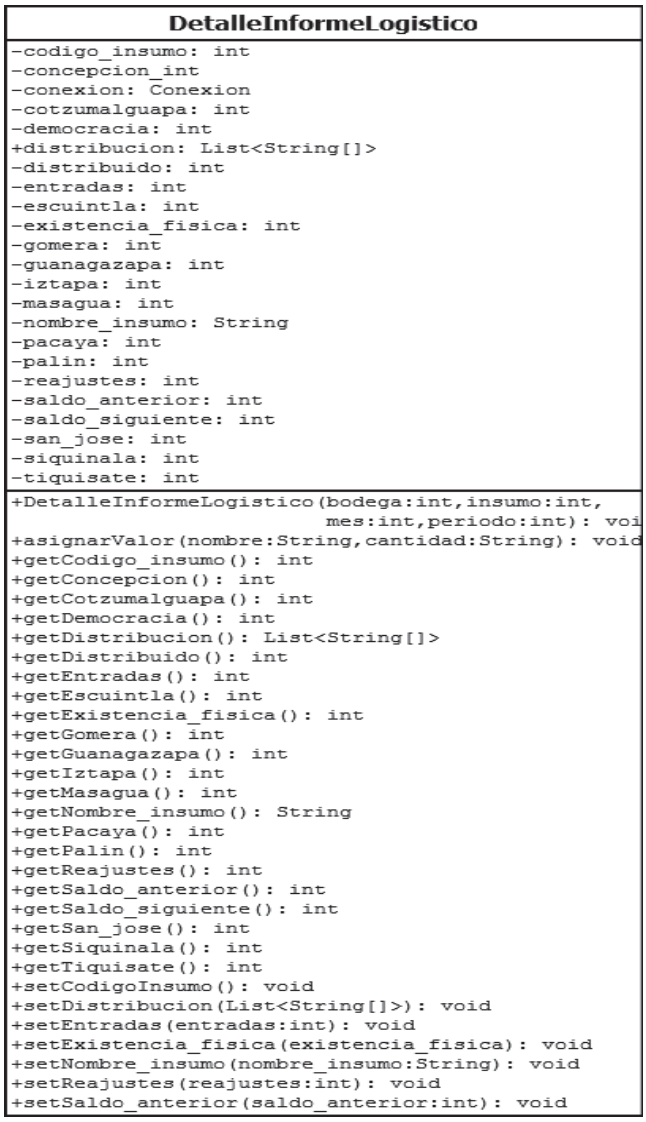
Apéndice 3h. **InformeLogístico**



Fuente: elaboración propia

- **DetalleInformeLogistico:** esta clase sirve para detallar cada fila que deberá ir en el informe logístico cuando este se imprima en el formato adecuado.

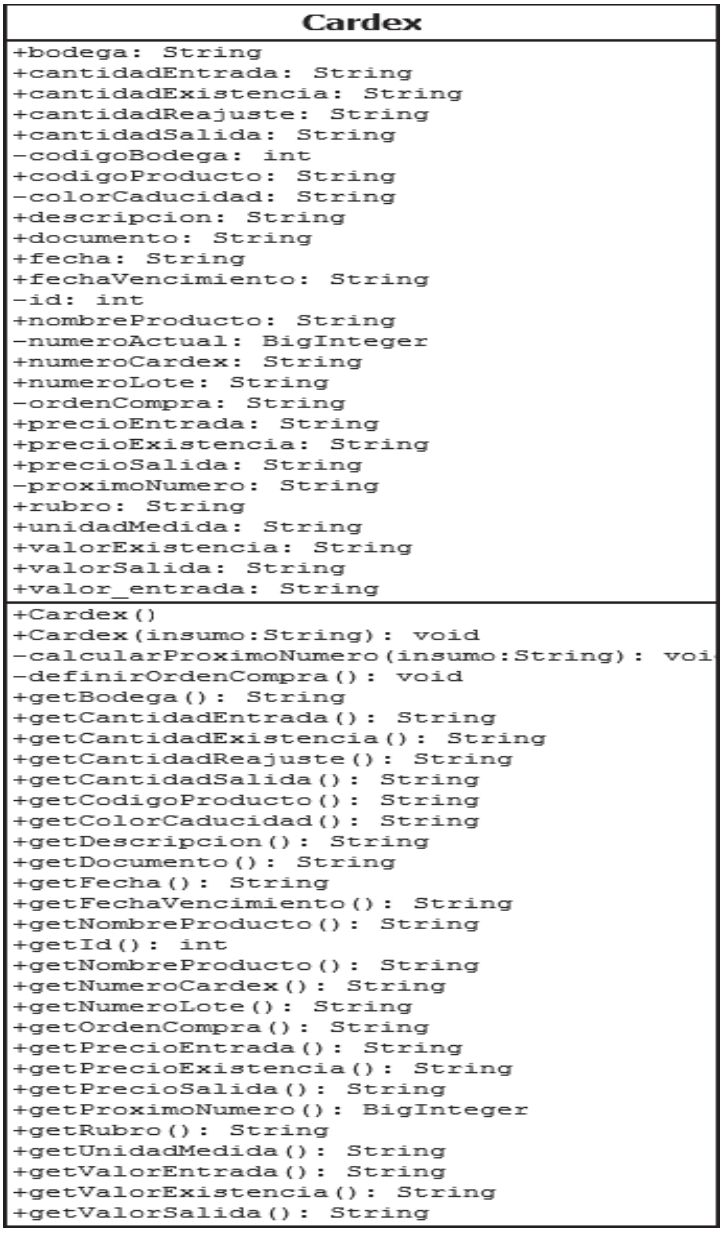
Apéndice 3i. **DetalleInformeLogistico**



Fuente: elaboración propia.

- **Cárdex:** esta clase sirve para recabar la información de cárdex, cuando un operario realiza una consulta de los movimientos en bodega para determinado mes.

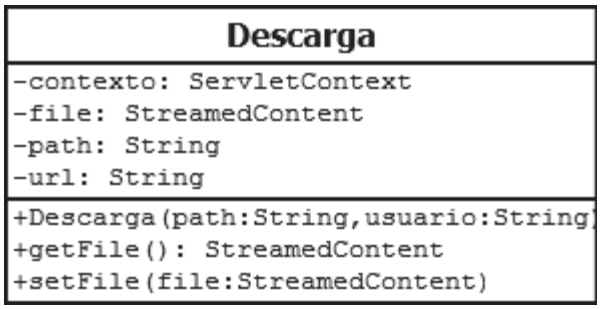
Apéndice 3j. **Cárdex**



Fuente: elaboración propia.

- **Descarga:** encapsula la funcionalidad de obtención de un archivo a descargar. En este caso, sirve para la descarga de informes generados en el sistema.

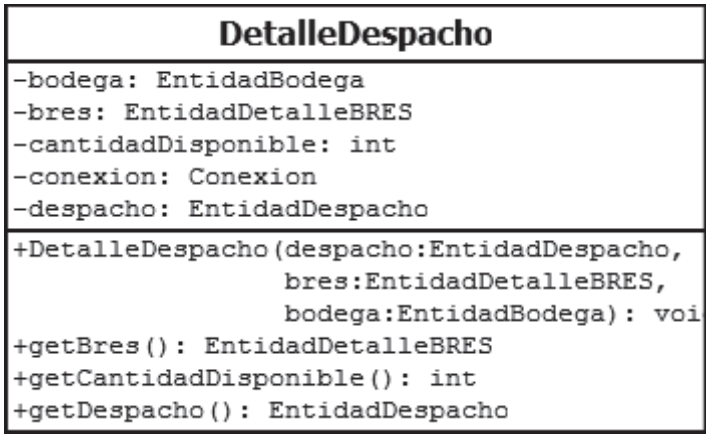
Apéndice 3k. **Descarga**



Fuente: elaboración propia.

- **DetalleDespacho:** esta clase sirve para recabar la información de despachos, tras una consulta realizada por un usuario desde el sistema.

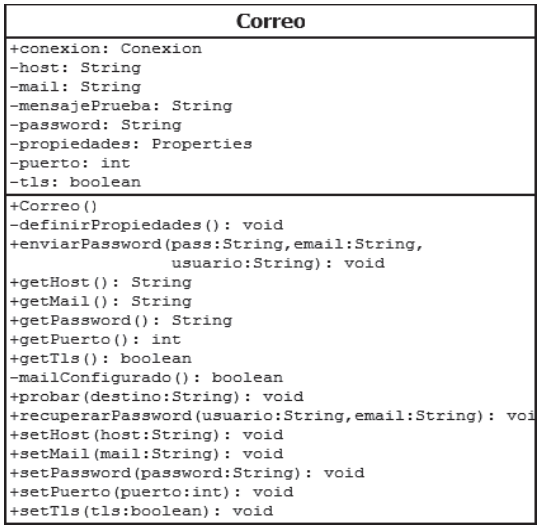
Apéndice 3l. **DetalleDespacho**



Fuente: elaboración propia.

- **Correo:** proporciona funcionalidad para la configuración de la dirección de correo electrónico que servirá al sistema, cuando deba enviar correos por recuperación de *password*. También proporciona funcionalidad para enviar dichos correos electrónicos.

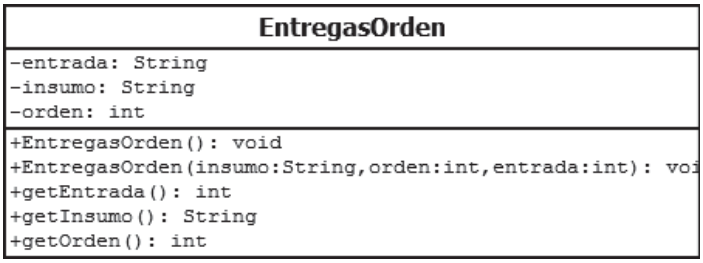
Apéndice 3m. **Correo**



Fuente: elaboración propia.

- **EntregasOrden:** clase que encapsula información de entregas de una orden de compra.

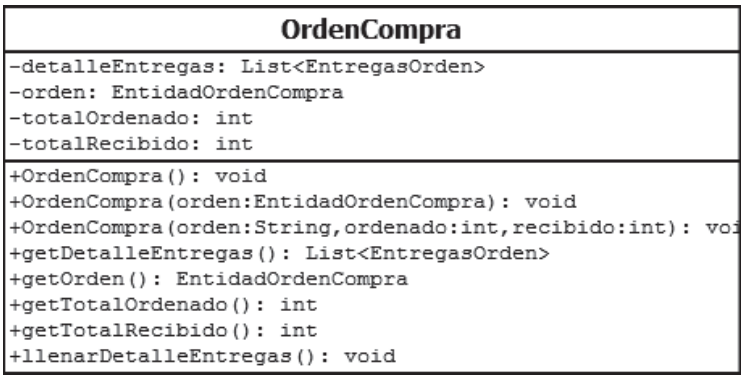
Apéndice 3n, **EntregasOrden**



Fuente: elaboración propia.

- **OrdenCompra:** su funcionalidad es contener el encabezado de una orden de compra. A su vez, posee el detalle de las entregas hechas por proveedores de la orden de compra.

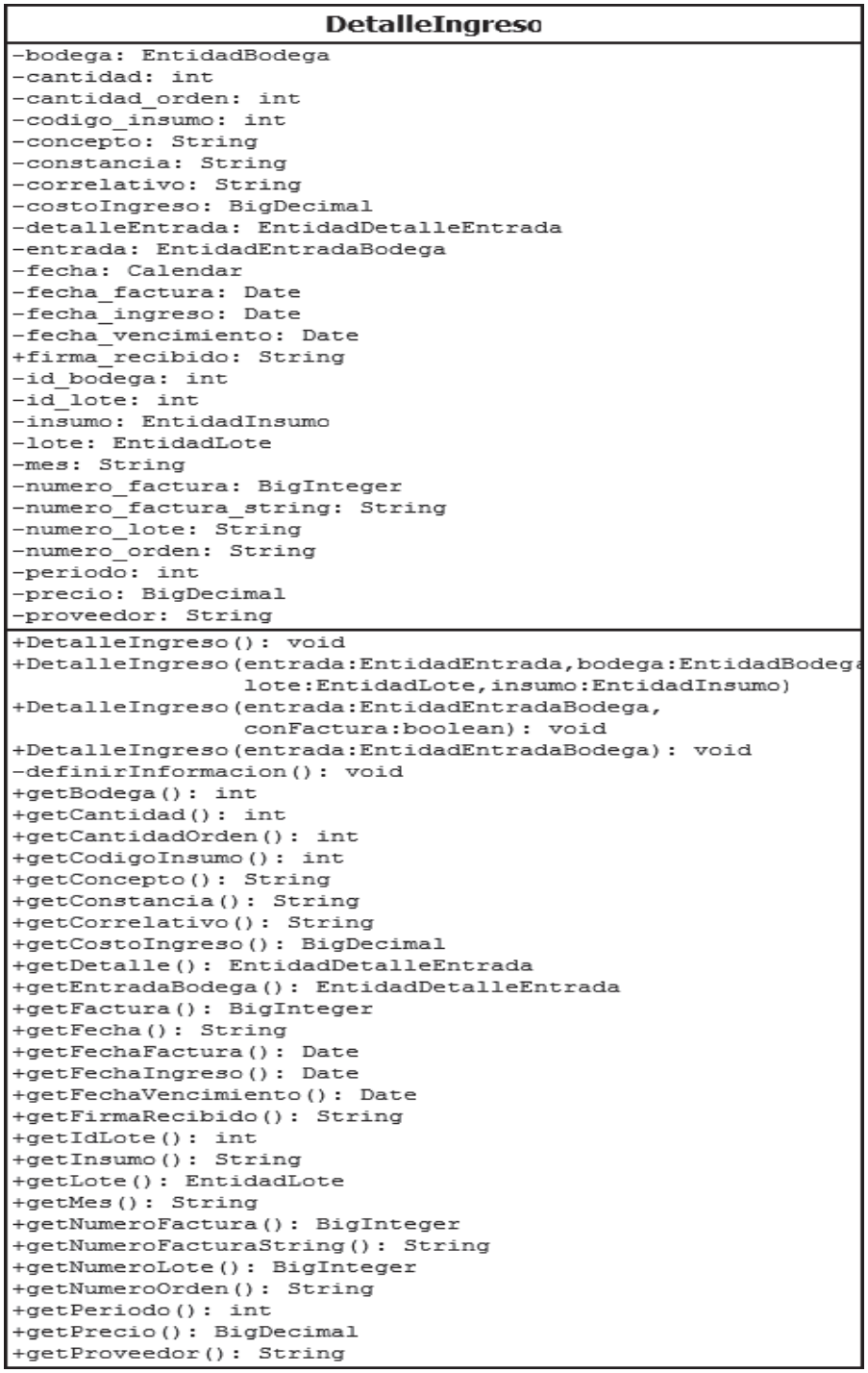
Apéndice 3ñ. **Orden Compra**



Fuente: elaboración propia.

- **DetalleIngreso:** esta clase encapsula el manejo de los datos al registrar una entrada a bodega. También es útil en la generación de informes relacionados con ingresos a bodega.

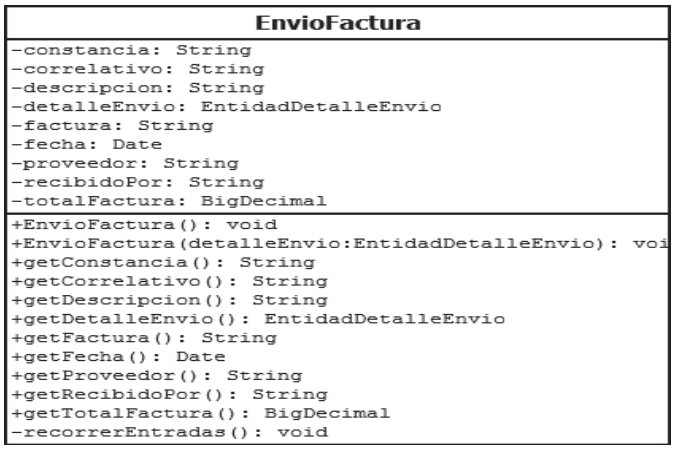
Apéndice 3o. **DetalleIngreso**



Fuente: elaboración propia.

- **EnvioFactura:** esta clase proporciona funcionalidad para el manejo de la información de envíos de factura al departamento de Contabilidad.

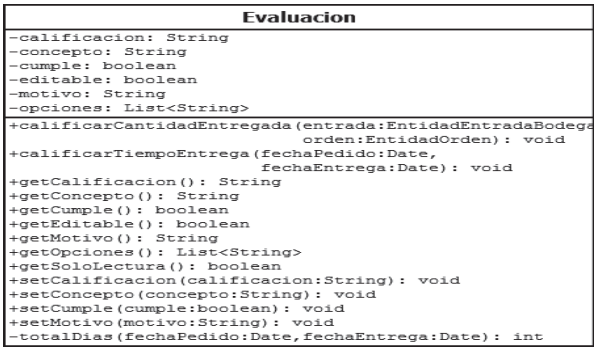
Apéndice 3p. **EnvioFactura**



Fuente: elaboración propia.

- **Evaluación:** permite contener la información generada por un usuario, al momento de realizar una evaluación de proveedor; es útil también para la posterior generación del documento que se imprimirá para la evaluación.

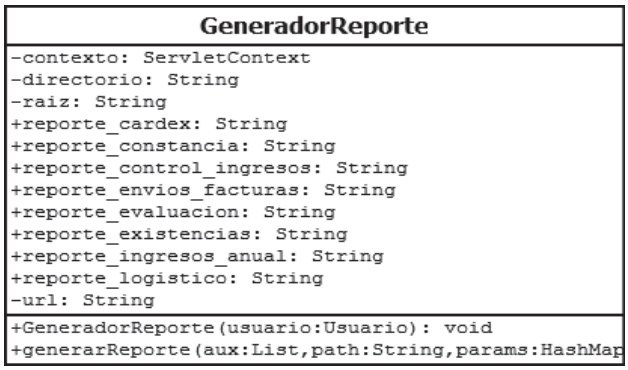
Apéndice 3q. **Evaluación**



Fuente: elaboración propia.

- **GeneradorReporte:** encapsula la funcionalidad para generar en formato pdf un reporte.

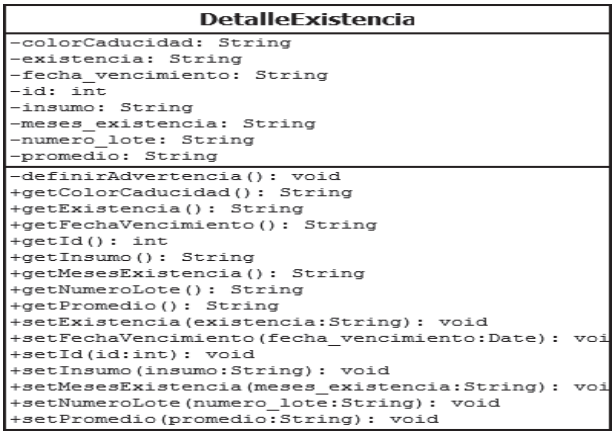
Apéndice 3r. **GeneradorReporte**



Fuente: elaboración propia.

- **DetalleExistencia:** esta clase encapsula la información que se generará en un reporte sobre las existencias en bodega, según la consulta que realice un usuario.

Apéndice 3s. **DetalleExistencia**



Fuente: elaboración propia.

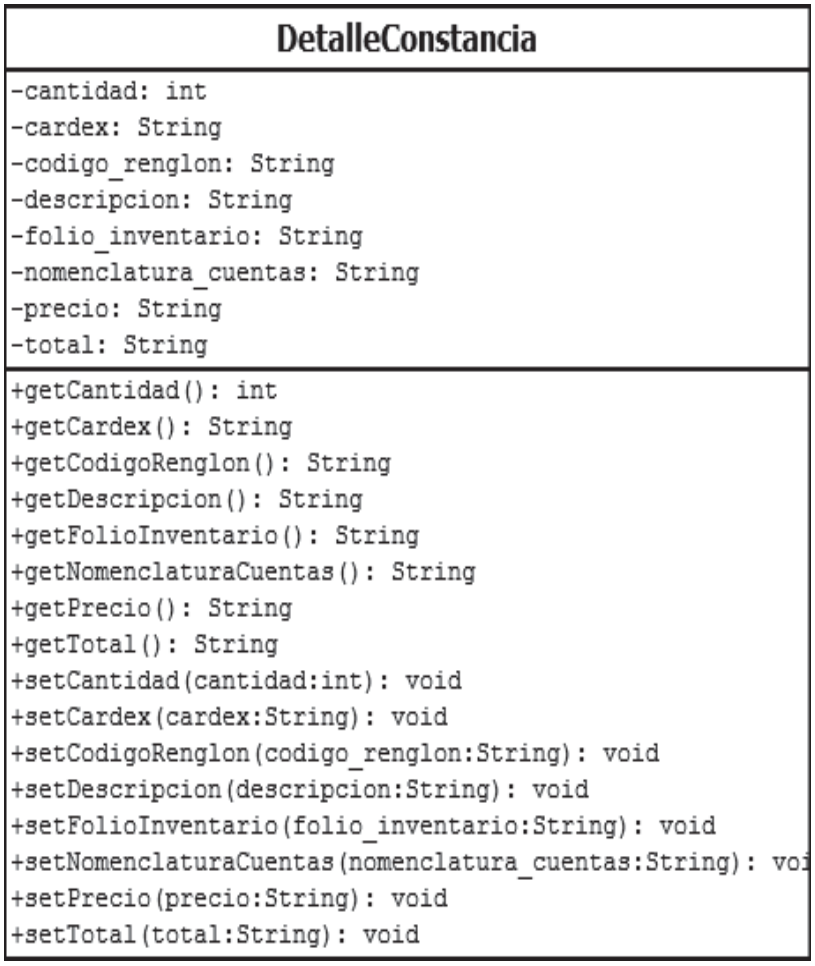
- **Constancia:** es útil para obtener la información de una constancia de ingreso y posteriormente generar dicha constancia.

Apéndice 3t. **Constancia**

Fuente: elaboración propia.

- **DetalleConstancia:** contiene el detalle que se mostrará en una constancia de ingreso.

Apéndice 3u. **DetalleConstancia**



Fuente: elaboración propia.

- **Controlador**: descripción de clases**.**

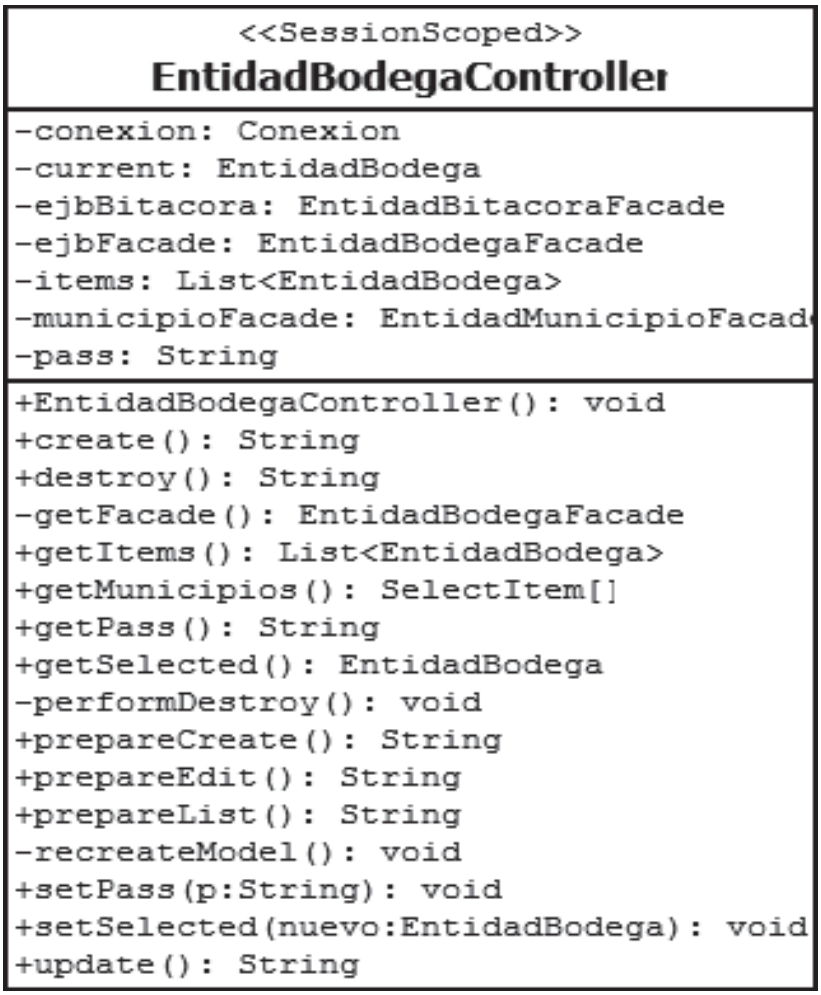
    - **ControladorSesion:** esta clase funciona para recabar información referente a la sesión actual de un usuario.

Apéndice 3v. **ControladorSesion**

Fuente: elaboración propia.

- **EntidadBodegaController:** clase para gestionar la vista y las funcionalidades concernientes a bodegas.

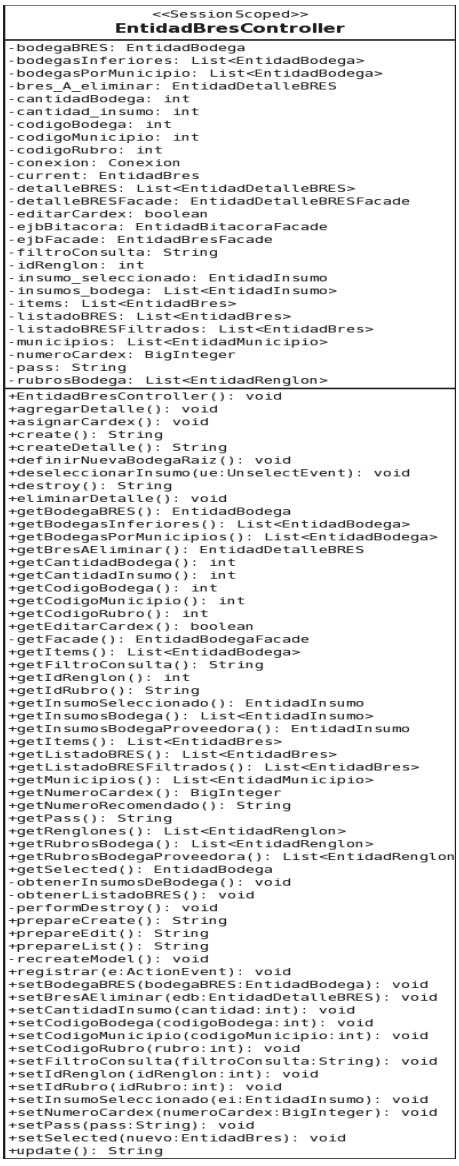
Apéndice 3w. **EntidadBodegaController**



Fuente: elaboración propia.

- **EntidadBresController**: esta clase provee funcionalidad para gestionar en la vista, información y funcionalidad relacionada con BRES.

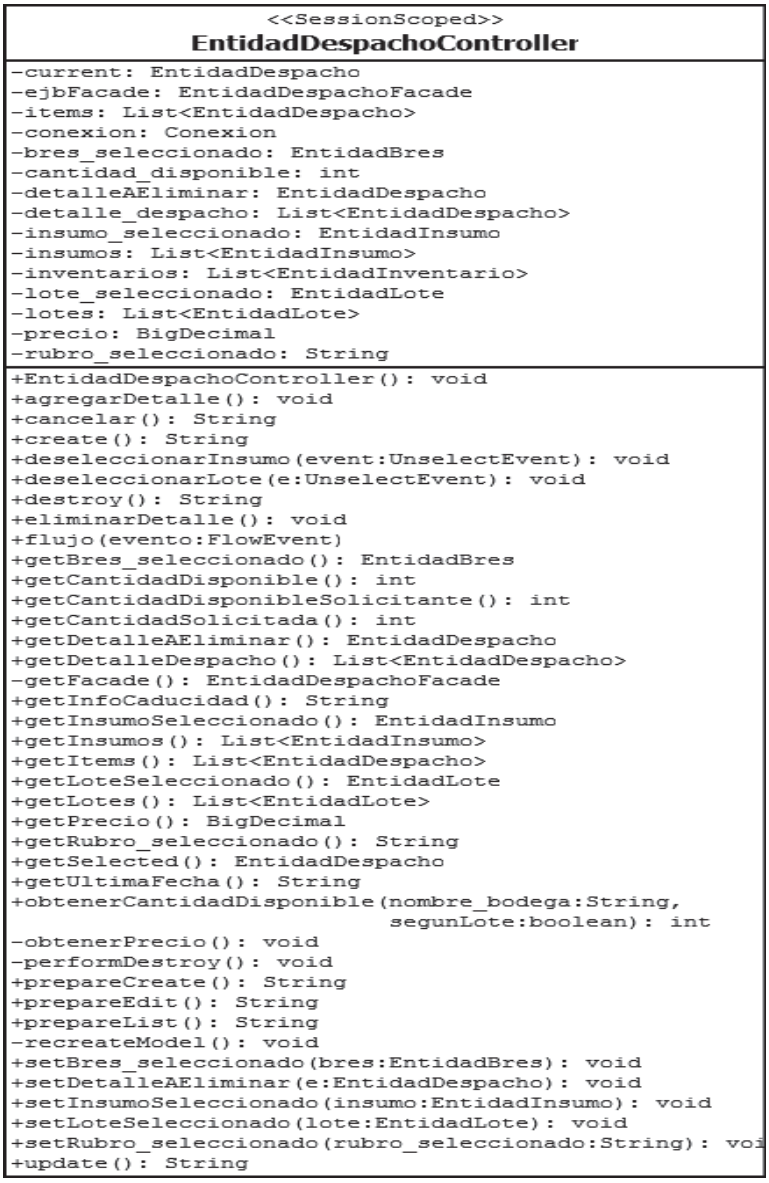
Apéndice 3x. **EntidadBresController**



Fuente: elaboración propia.

- **EntidadDespachoController**: clase de controlador para la funcionalidad de despacho de insumos.

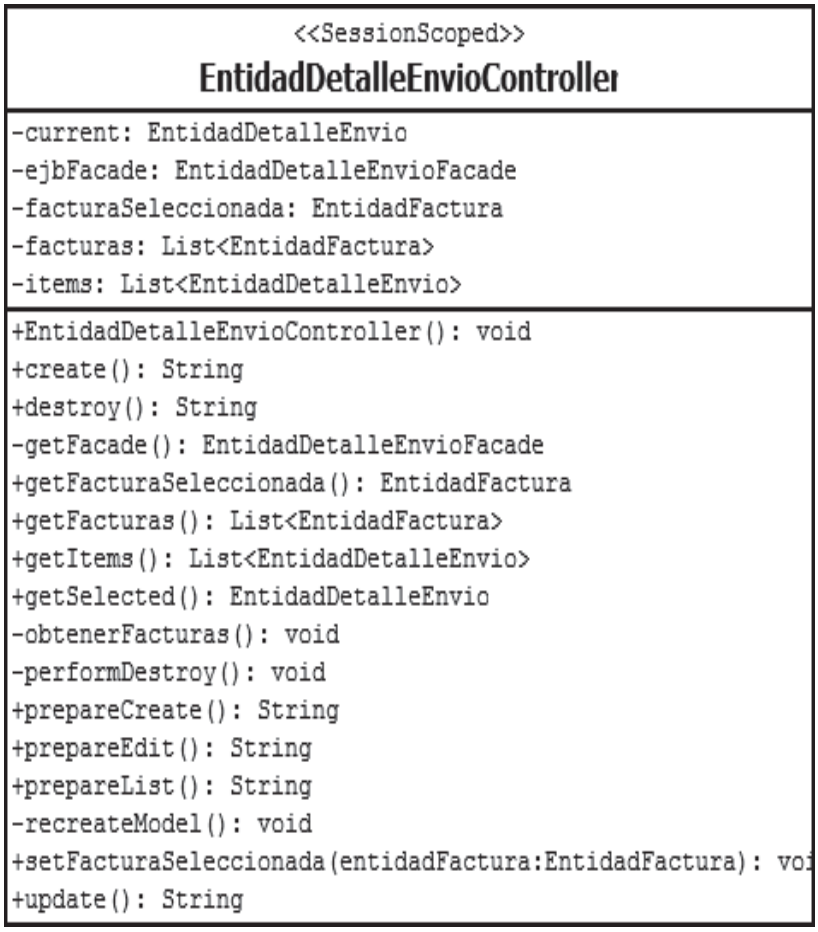
Apéndice 3y. **EntidadDespachoController**



Fuente: elaboración propia.

- **EntidadDetalleEnvioController**: clase de controlador para manejar la funcionalidad e información de entrega de factura.

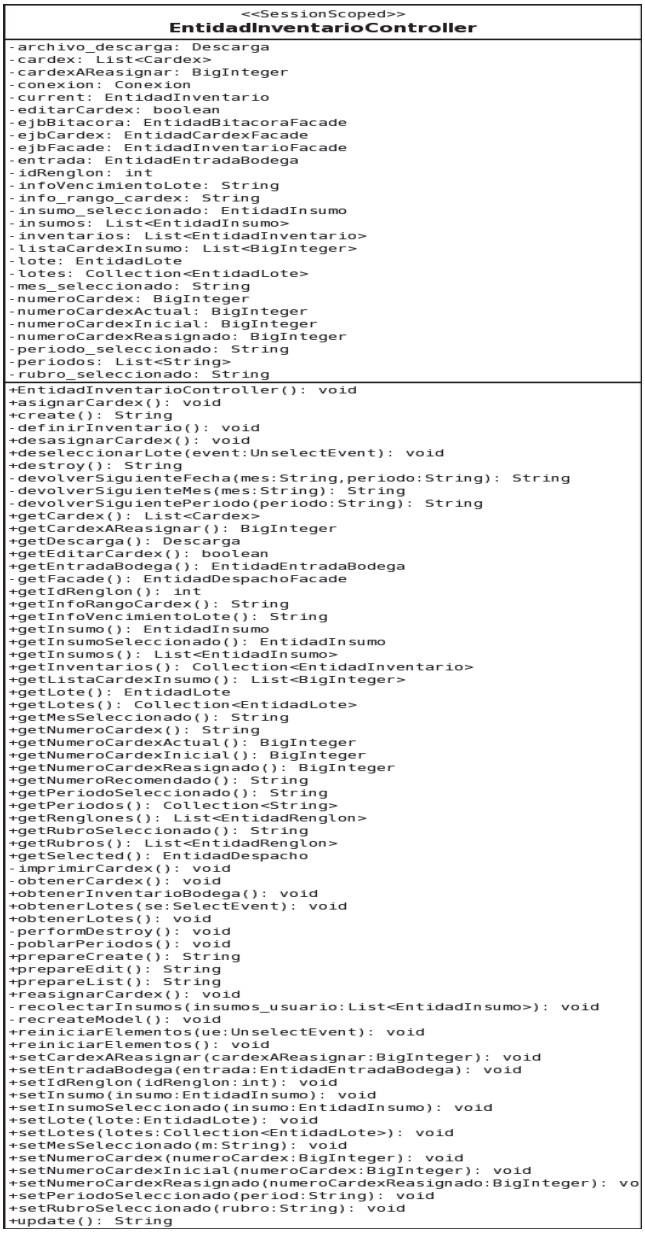
Apéndice 3z. **EntidadDetalleEnvioController**



Fuente: elaboración propia.

- **EntidadInventarioController**: gestiona acceso a la información de inventarios y su presentación en la interfaz de usuario.

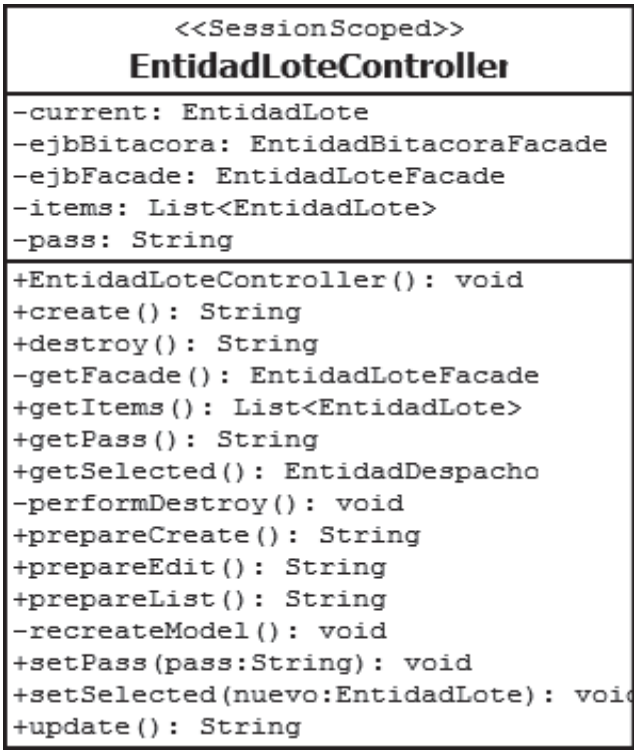
Apéndice 3a$_1$. **EntidadLoteController**



Fuente: elaboración propia.

- **EntidadLoteController**: esta clase funciona como controlador en la gestión de la información y funcionalidades concernientes al manejo de lotes registrados en el sistema.

Apéndice $3a_2$. **EntidadLoteController**



Fuente: elaboración propia.

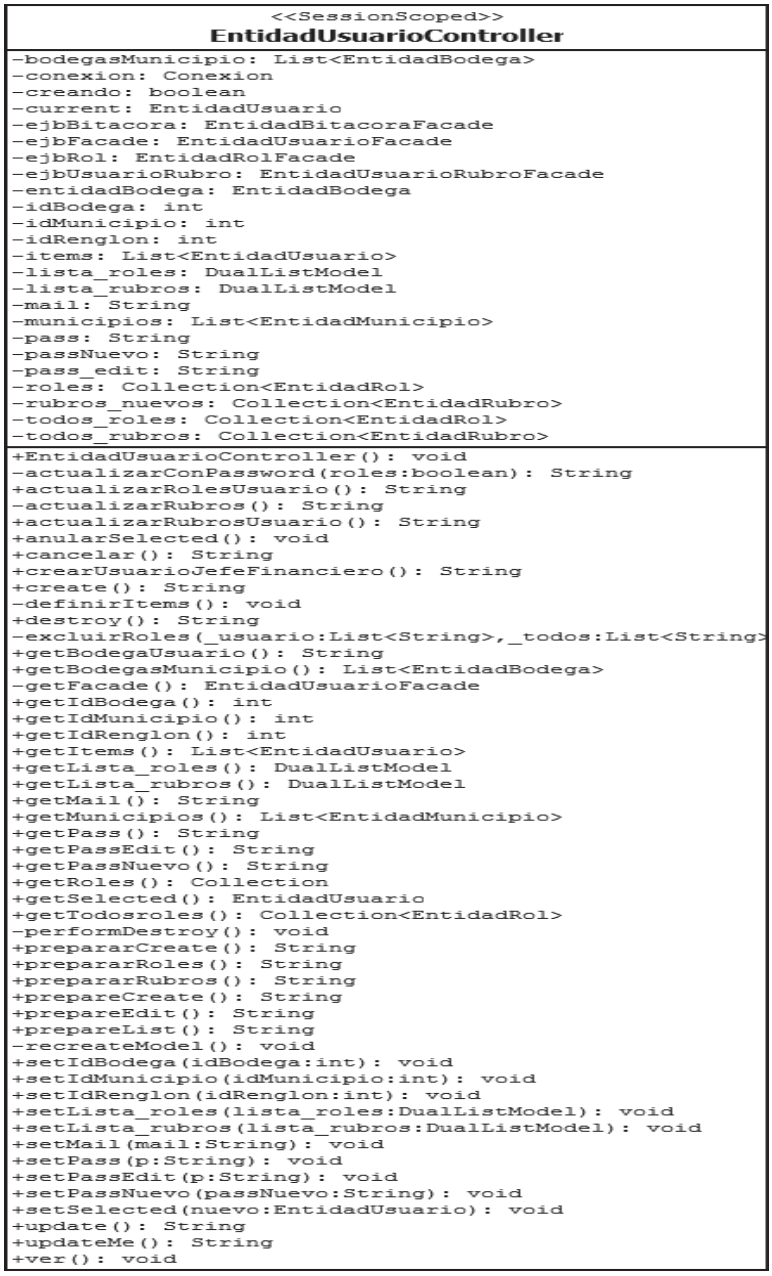
- **EntidadReajusteController**: clase de controlador para la funcionalidad de reajustes en inventario.

Apéndice 3a$_3$. **EntidadReajusteController**

Fuente: elaboración propia.

- **EntidadUsuarioController**: esta clase encapsula las funcionalidades para la gestión de usuarios del sistema.

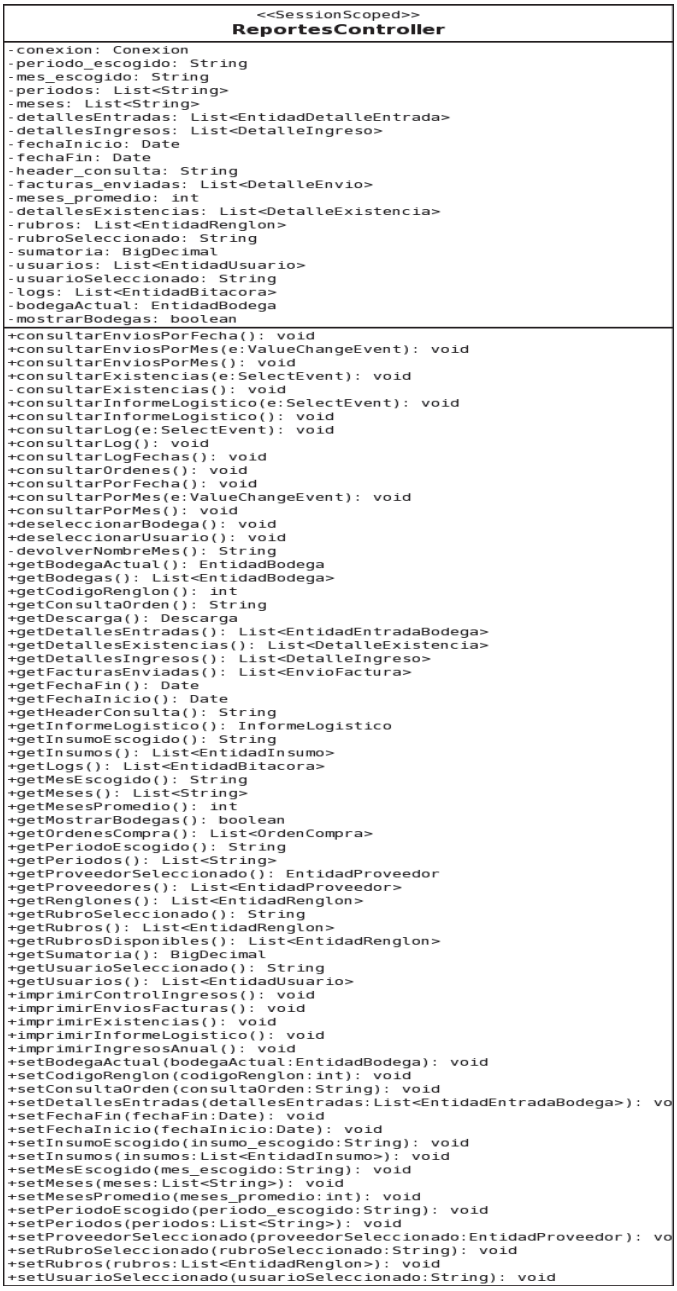
Apéndice 3a$_{4.}$ **EntidadUsuarioController**



Fuente: elaboración propia.

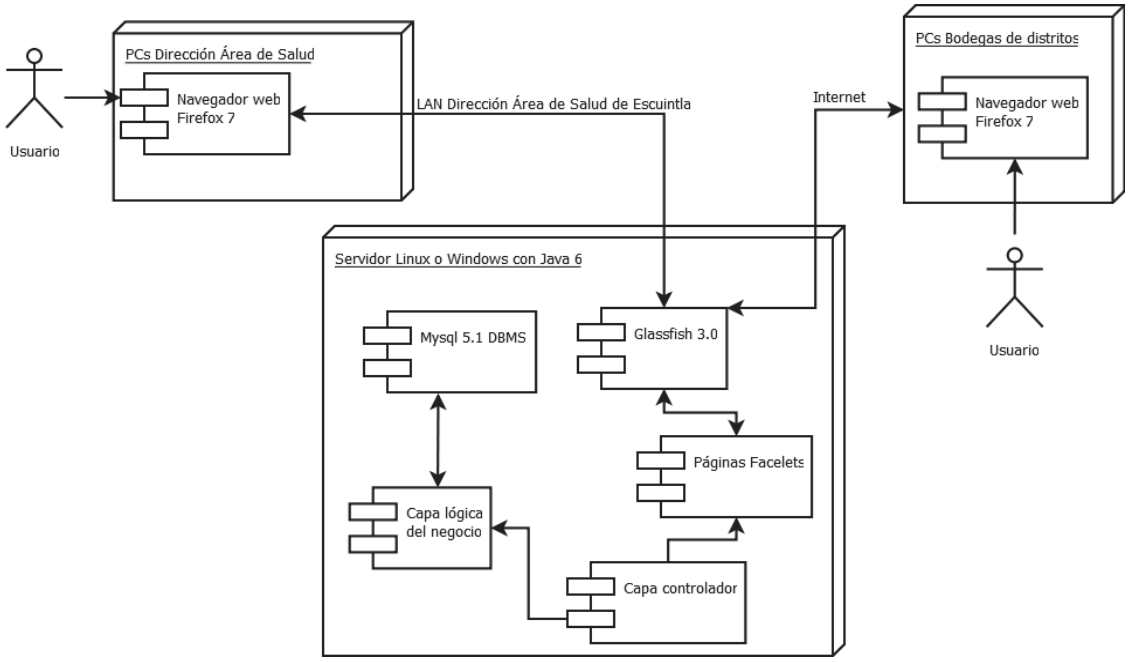
- **ReportesController:** clase para gestionar la presentación de informes en la aplicación y la generación de los mismos a formato pdf.

Apéndice 3a$_{5.}$ **ReportesController**



Fuente: elaboración propia.

Apéndice 4**.** **Vista de despliegue**



Fuente: elaboración propia.

- **Descripción de los componentes**
  - Servidor Windows o Linux con Java 6: la ventaja de trabajar con tecnología JAVA es que la aplicación es portable, siendo independiente el sistema operativo en el que se instale. El servidor de aplicaciones Glassfish 3.0, necesita Java EE 6 o superior para funcionar. La interfaz de usuario está desarrollada con JSF 2 y Primefaces 2. La capa de controlador se encarga de gestionar los eventos de la interfaz del usuario, para que la capa de la lógica del negocio funcione. La capa de la lógica del negocio contiene las funcionalidades propias de la gestión de la información, que está contenida en una base de datos gestionada a su vez con el DBMS Mysql 5.1.

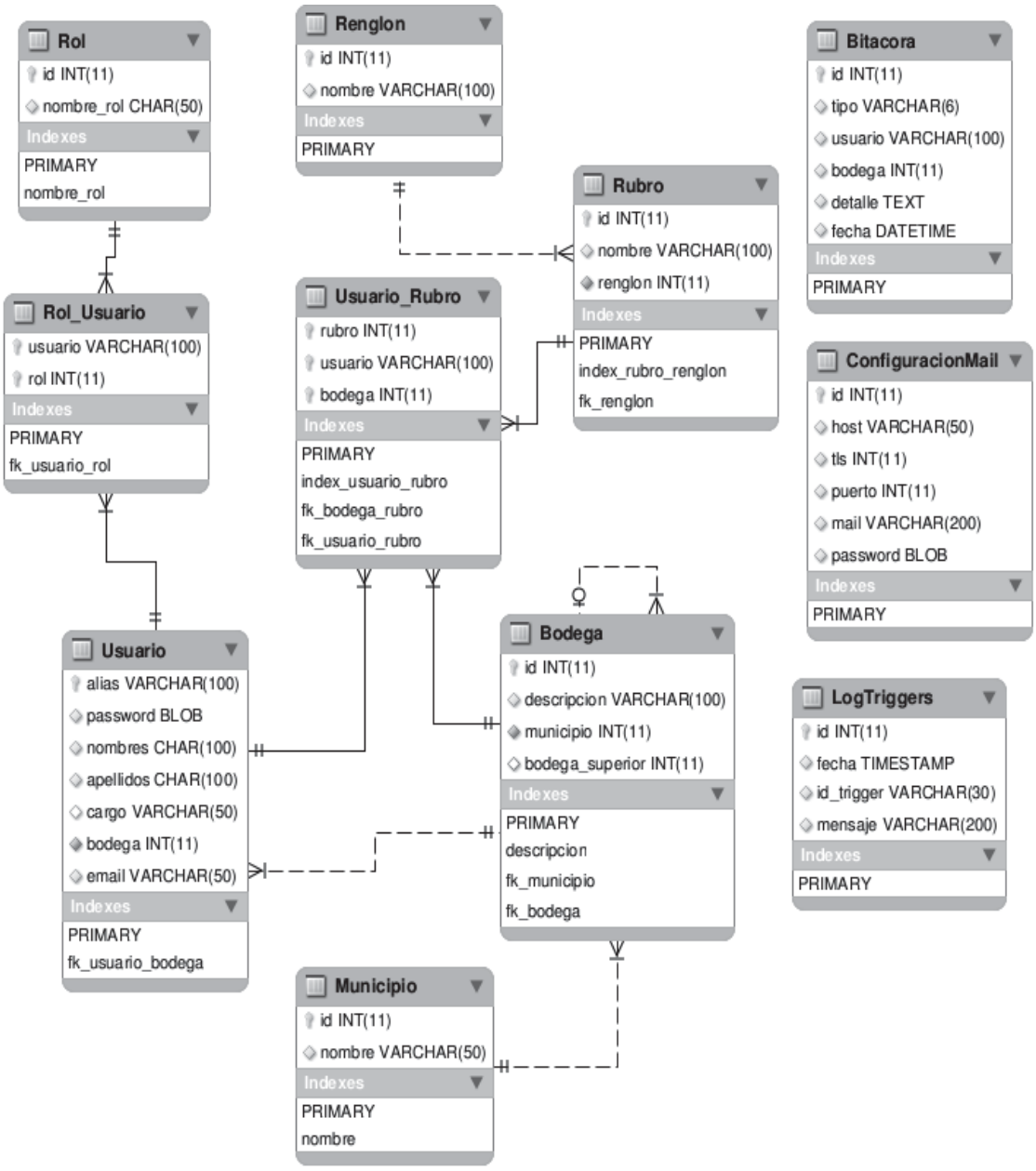
- PCs Dirección Área de Salud: son las computadoras que tendrán acceso al sistema a través de la red LAN con la que cuenta la institución. Para poder acceder al mismo no es necesario instalar software especial, únicamente un navegador web. Se recomienda Firefox 7 o superior.

- PCs bodegas de distrito: son las computadoras de los distritos; al no estar conectados a una red LAN, accederán al sistema por medio de Internet, necesitando únicamente, al igual que en las computadoras de la Dirección, la instalación de un navegador web.

G. Vista de datos

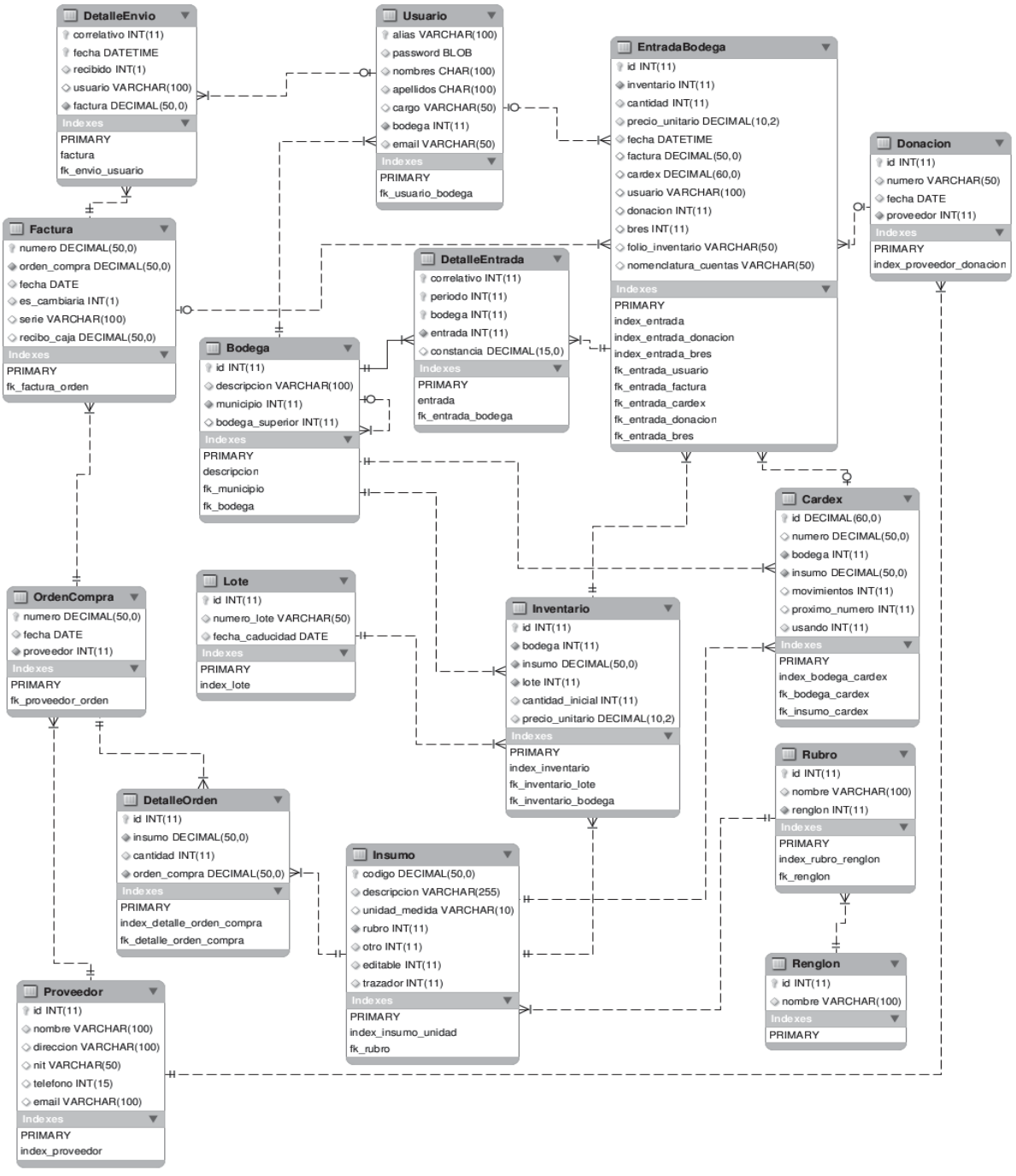
Para presentar de una forma ordenada el diseño de la estructura de la base de datos, se dividió según los módulos descritos en la vista de casos de uso, mostrando las entidades significativas en cada uno de estos.

Apéndice 5. **Administración del sistema**



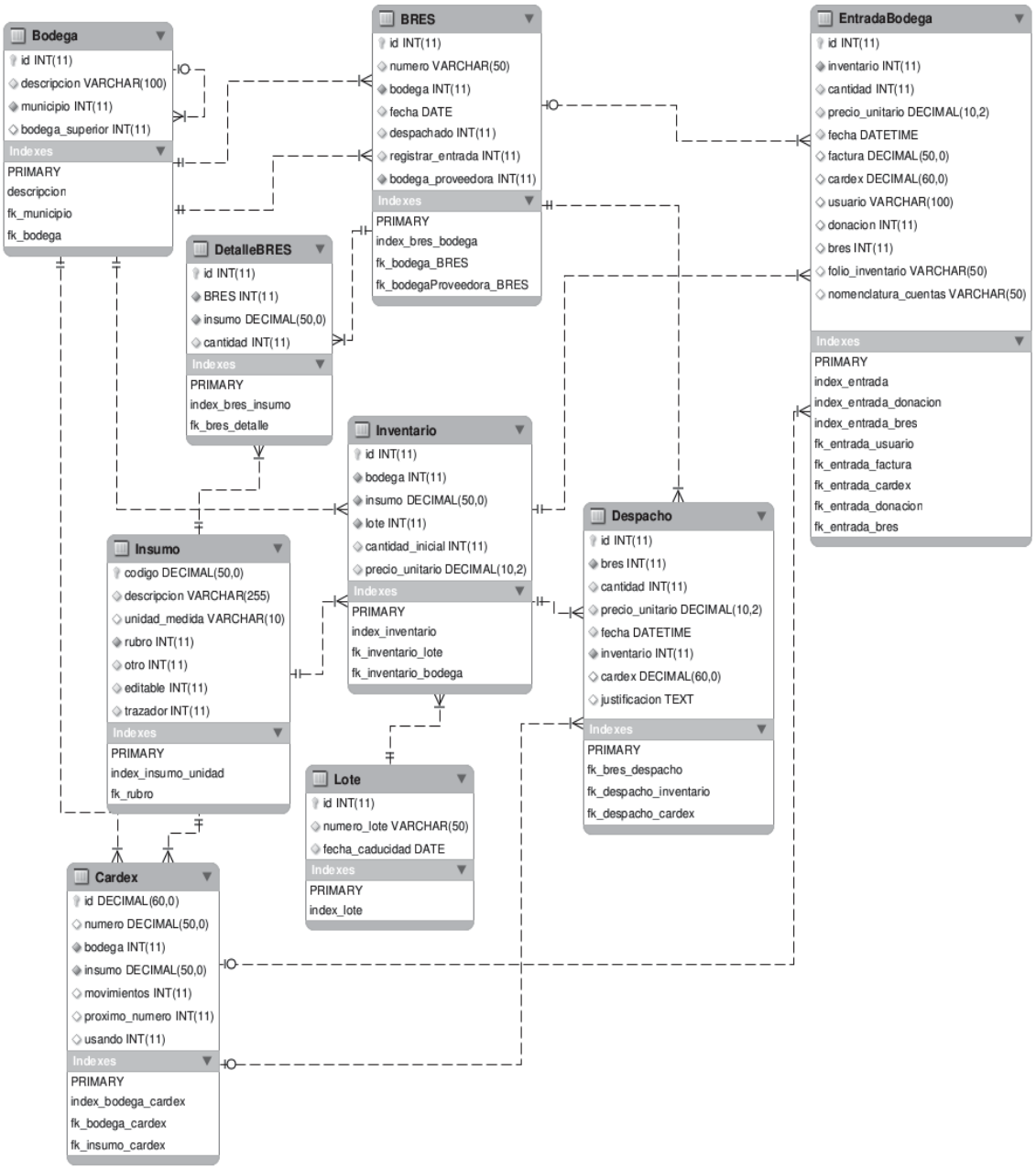
Fuente: elaboración propia.

Apéndice 6. **Entradas a bodega**



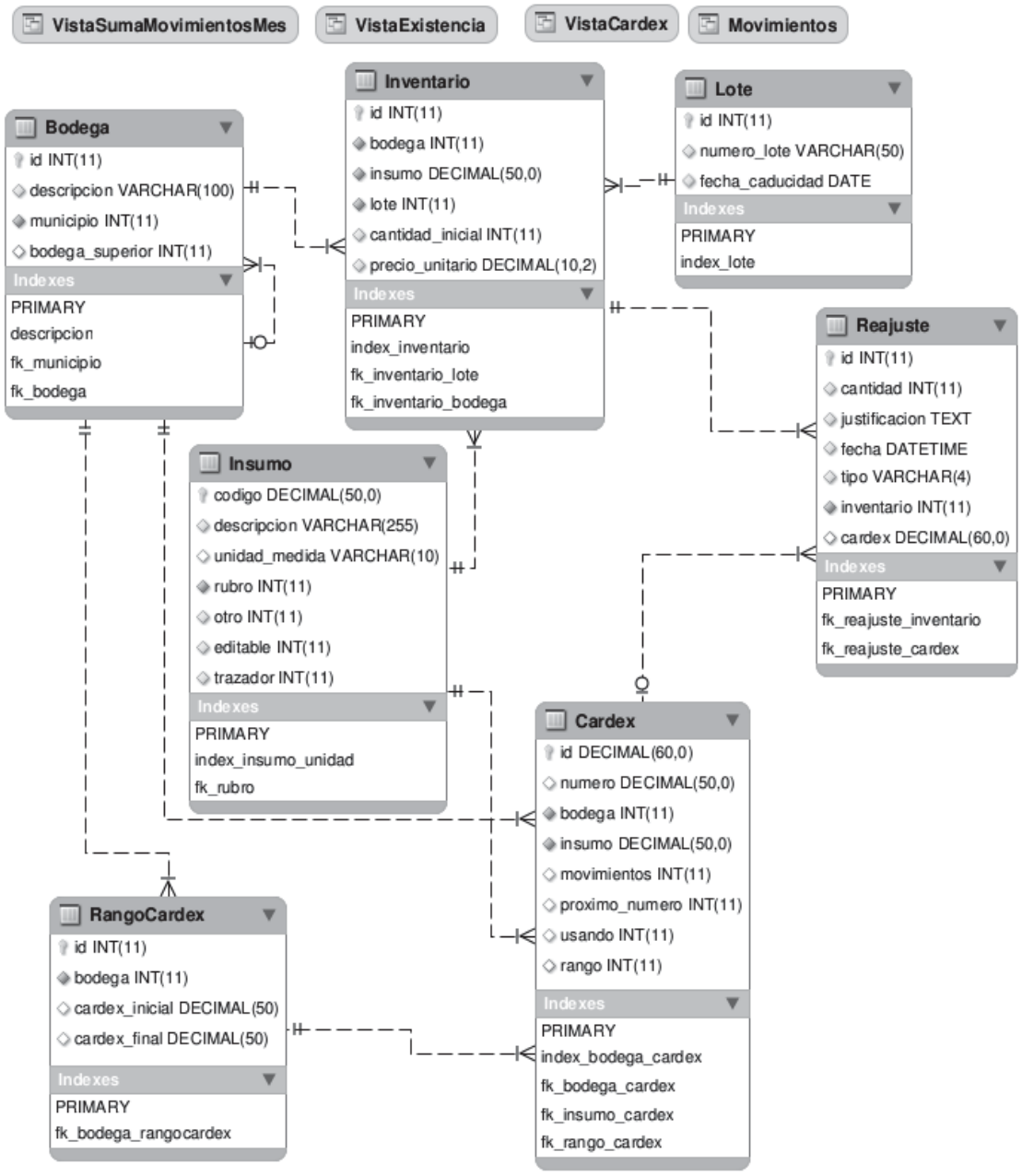
Fuente: elaboración propia.

Apéndice 7. **Salidas de bodega**



Fuente: elaboración propia.

Apéndice 8. **Control de bodegas**



Fuente: elaboración propia.

Apéndice 9. **Descripción de las entidades de datos**

- Bitácora: guarda información de las acciones que los usuarios realizan en el sistema.
- Bodega: almacena la información de las bodegas que existen en los distritos.
- BRES: entidad que almacena la información de encabezado de un BRES.
- Cárdex: esta entidad almacena información de los números de cárdex asignados a los insumos por bodegas.
- ConfiguracionMail: almacena la información del correo electrónico del sistema.
- Despacho: almacena los despachos que se realizan.
- DetalleBRES: almacena la información detallada de un BRES en específico.
- DetalleEntrada: almacena información del correlativo de ingreso a la bodega.
- DetalleEnvio: almacena información de envíos de factura al departamento de Contabilidad.
- DetalleOrden: especifica los insumos que se solicitaron para una orden de compra. Esta información se recupera de la base de datos de Compras.

- Donación**:** almacena información de las donaciones recibidas en bodega.

- EntradaBodega**:** almacena información de las entradas a la bodega.

- Factura: entidad con la información del encabezado de una factura.

- Inventario: entidad que almacena información de los insumos con lotes, según bodegas.

- Insumo: guarda información de los insumos. Dicha información es recuperada desde la base de datos de Compras.

- LogTriggers: esta entidad recaba información de los *triggers* utilizados para recabar información de la base de datos de Financiero.

- Lote: información del lote correspondiente para cada insumo.

- Movimientos: es una vista que proporciona el detalle de todos los movimientos (entradas, salidas y reajustes) en la bodega.

- Municipio: almacena el nombre de cada uno de los 13 distritos municipales de Escuintla.

- OrdenCompra: guarda información de encabezado de una orden de compra. Esta información se recupera de la base de datos de compras.

- Proveedor: entidad para almacenar información de los proveedores.

- Reajuste: entidad que almacena la información de reajustes de insumos

realizados.

- Renglón: entidad para guardar información de los renglones a los que pertenecen los insumos.

- Rol: información de los roles de acceso al sistema.

- Rol_Usuario: específica los roles que posee un usuario.

- Rubro: información de la clasificación para cada insumo.

- Usuario Información de los usuarios creados en el sistema. Usuario admin creado por *default*.

- Usuario_Rubro: guarda información de los rubros asignados a los usuarios.

- VistaCardex: proporciona los datos de cárdex para obtenerlos de una forma parecida a como los muestra la tarjeta físicamente.

- VistaExistencia: vista con recabación de las existencias de insumos por cada mes, según bodegas.

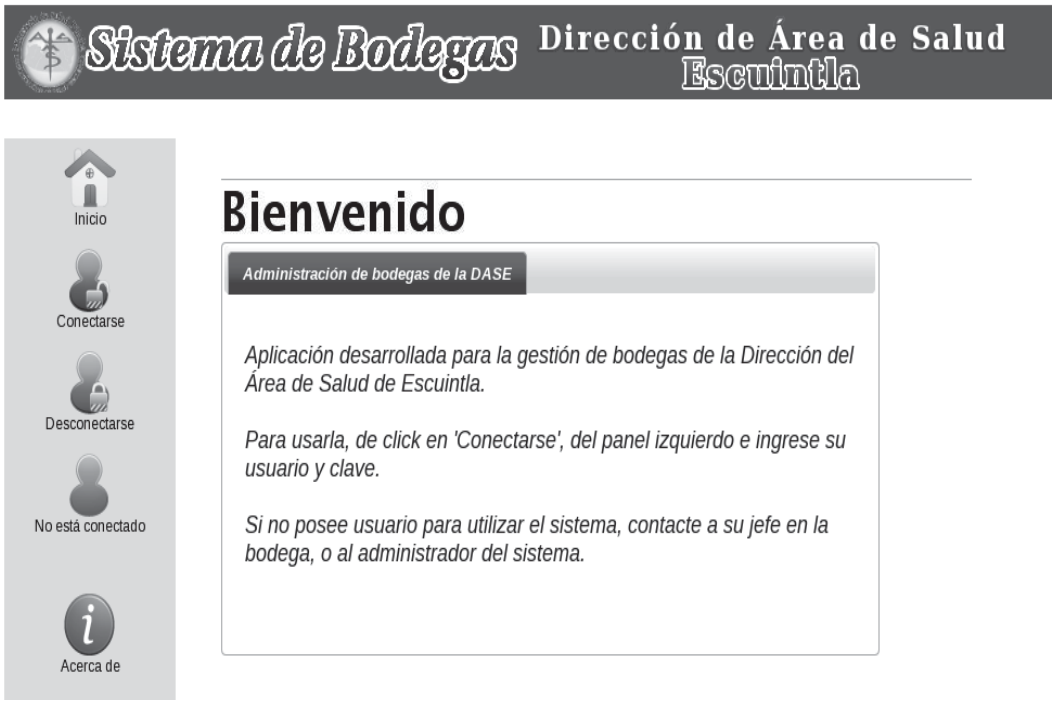
- VistaSumaMovimientosMes: vista que guarda la sumatoria de cantidades obtenidas en la vista Movimientos, agrupándolas por mes.

Apéndice 10. **El producto**

En esta sección se presentan imágenes de las principales funcionalidades de la aplicación desarrollada en este proyecto.

- **Pantalla de inicio:** la aplicación fue desarrollada como tecnología web, por lo que para utilizarla, es necesario poseer un explorador web. Únicamente un usuario registrado en el sistema puede acceder a sus funcionalidades, y para esto, deberá ingresar su nombre de usuario y clave en la pantalla de inicio.

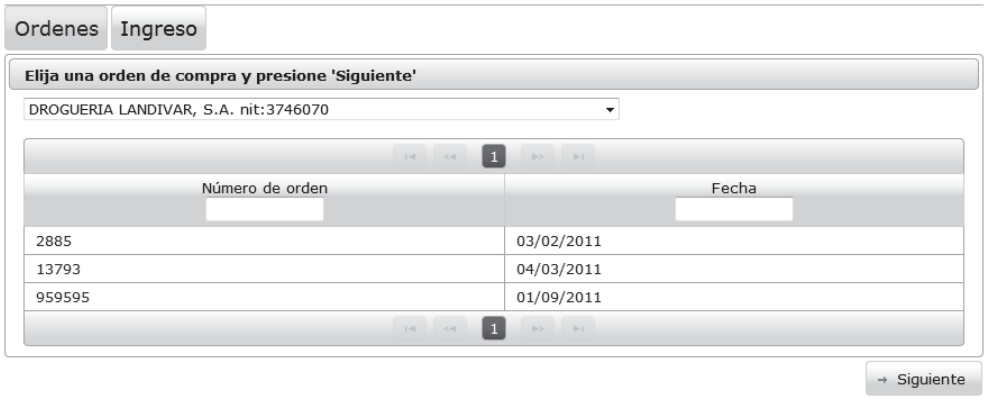
Apéndice 10a. **Pantalla de inicio**



Fuente: elaboración propia.

- **Entradas:** para registrar las entradas, el usuario deberá escoger la orden de compra del proveedor que entrega.

Apéndice 10b. **Pantalla de inicio**



Fuente: elaboración propia.

Luego de escoger la orden de compra, el usuario deberá especificar el detalle de los insumos que está recibiendo, para posteriormente registrar la entrada.

Apéndice 10c. **Detalle de insumos**

Fuente: elaboración propia.

Despachos: para registrar un despacho, el usuario deberá escoger un BRES previamente registrado en el sistema por la bodega que solicita**.** Posteriormente, el usuario deberá indicar el detalle de las cantidades despachadas para cada insumo solicitado.

Apéndice 10d. **Detalle del despacho**

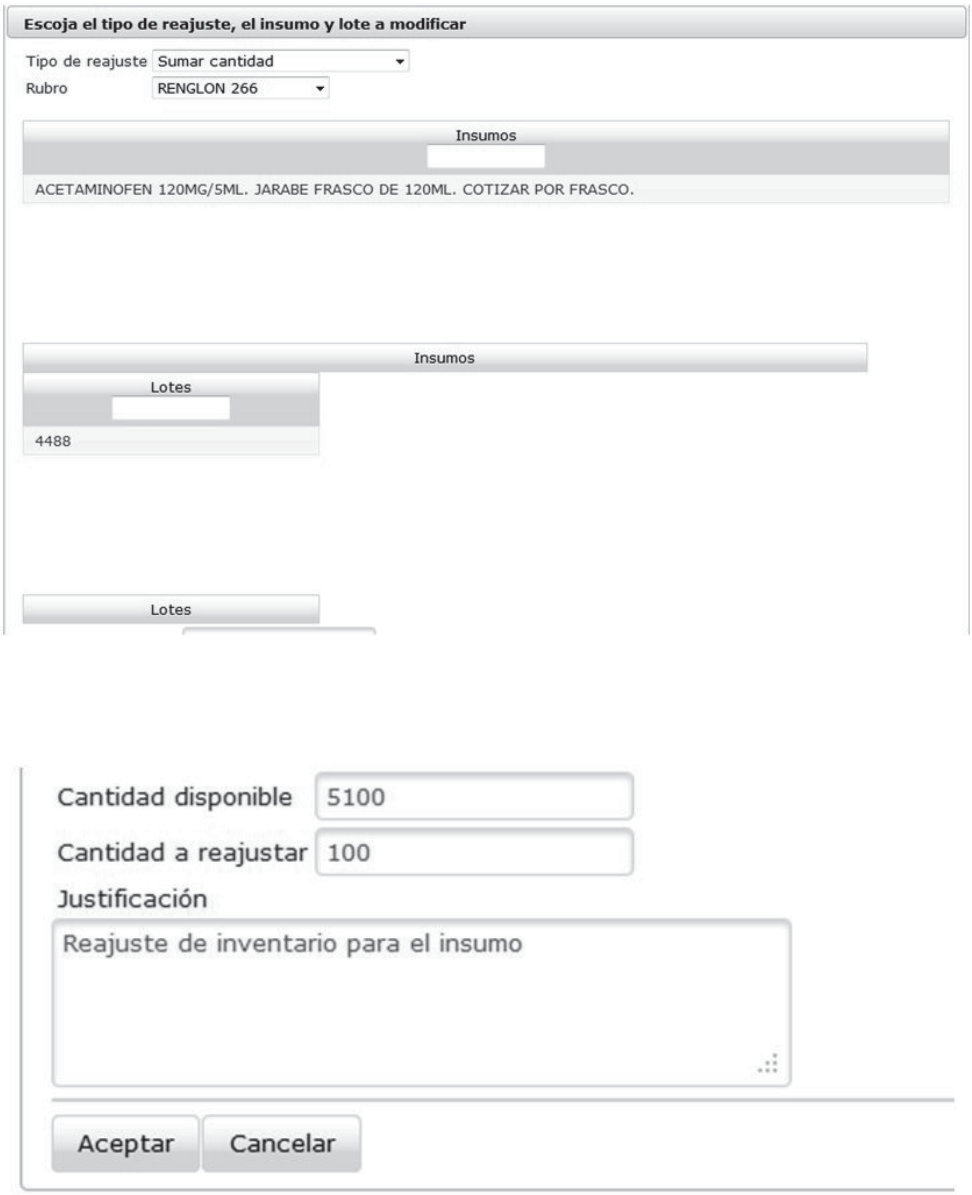
Fuente: elaboración propia.

Apéndice 10e. **Cantidades de insumo**

Fuente: elaboración propia.

- **Reajustes:** para realizar reajustes, el usuario deberá indicar qué tipo de reajuste será, para posteriormente escribir la justificación para la operación, antes de registrarla en el sistema.

Apéndice 10f. **Reajustes**



Fuente: elaboración propia.

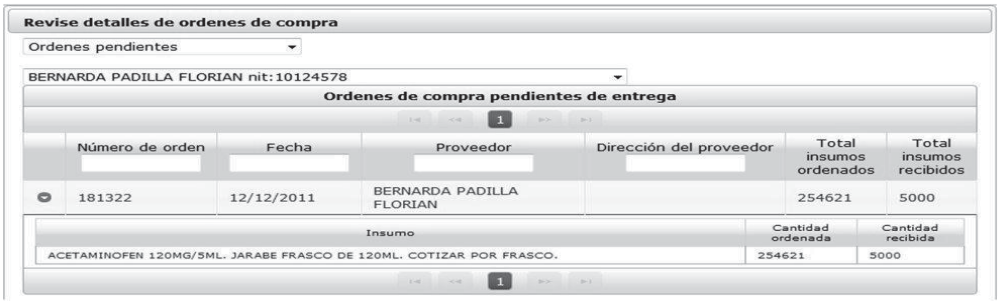
- ***Cárdex* electrónico:** manualmente, los usuarios manejan el control de los movimientos por medio de una tarjeta denominada "*Cárdex*". La aplicación automatiza dicho control, permitiendo ver los movimientos por mes para cada insumo registrado, e imprimir los resultados.

Apéndice 10g. ***Cárdex* electrónico**

Fuente: elaboración propia.

- **Consulta de órdenes de compra:** por medio de la consulta de las órdenes de compra, un usuario puede verificar el estado de dichas órdenes, pudiendo comparar cuántos insumos se han recibido, de acuerdo con los solicitados.

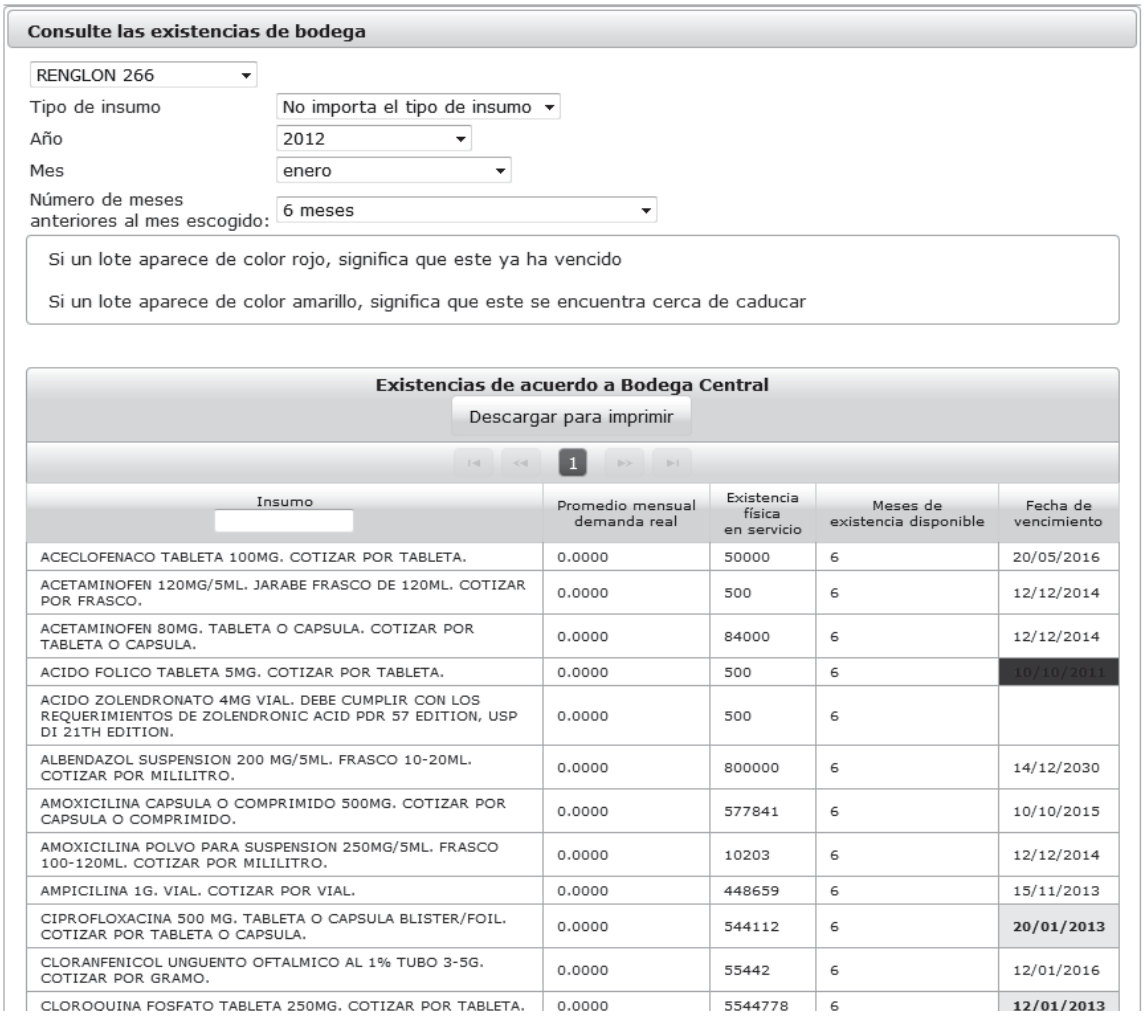
Apéndice 10h. **Consulta de órdenes de compra**



Fuente: elaboración propia.

- **Reporte de existencias:** el reporte de existencias, permite consultar las existencias para los insumos, agrupados por rubros.

Apéndice 10i. **Reporte de existencias**



Consulte las existencias de bodega

RENGLON 266

Tipo de insumo: No importa el tipo de insumo

Año: 2012

Mes: enero

Número de meses anteriores al mes escogido: 6 meses

Si un lote aparece de color rojo, significa que este ya ha vencido

Si un lote aparece de color amarillo, significa que este se encuentra cerca de caducar

Existencias de acuerdo a Bodega Central

Descargar para imprimir

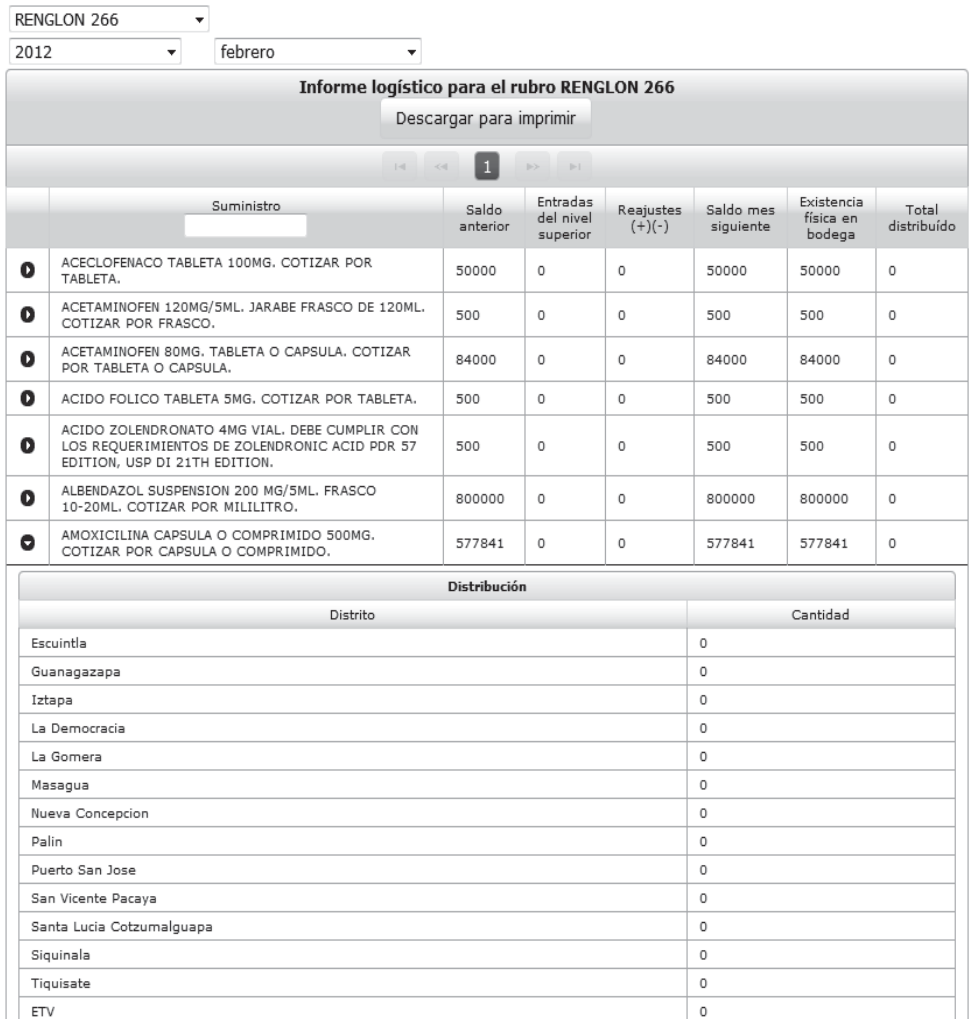
1

Insumo	Promedio mensual demanda real	Existencia física en servicio	Meses de existencia disponible	Fecha de vencimiento
ACECLOFENACO TABLETA 100MG. COTIZAR POR TABLETA.	0.0000	50000	6	20/05/2016
ACETAMINOFEN 120MG/5ML. JARABE FRASCO DE 120ML. COTIZAR POR FRASCO.	0.0000	500	6	12/12/2014
ACETAMINOFEN 80MG. TABLETA O CAPSULA. COTIZAR POR TABLETA O CAPSULA.	0.0000	84000	6	12/12/2014
ACIDO FOLICO TABLETA 5MG. COTIZAR POR TABLETA.	0.0000	500	6	10/10/2011
ACIDO ZOLENDRONATO 4MG VIAL. DEBE CUMPLIR CON LOS REQUERIMIENTOS DE ZOLENDRONIC ACID PDR 57 EDITION, USP DI 21TH EDITION.	0.0000	500	6	
ALBENDAZOL SUSPENSION 200 MG/5ML. FRASCO 10-20ML. COTIZAR POR MILILITRO.	0.0000	800000	6	14/12/2030
AMOXICILINA CAPSULA O COMPRIMIDO 500MG. COTIZAR POR CAPSULA O COMPRIMIDO.	0.0000	577841	6	10/10/2015
AMOXICILINA POLVO PARA SUSPENSION 250MG/5ML. FRASCO 100-120ML. COTIZAR POR MILILITRO.	0.0000	10203	6	12/12/2014
AMPICILINA 1G. VIAL. COTIZAR POR VIAL.	0.0000	448659	6	15/11/2013
CIPROFLOXACINA 500 MG. TABLETA O CAPSULA BLISTER/FOIL. COTIZAR POR TABLETA O CAPSULA.	0.0000	544112	6	20/01/2013
CLORANFENICOL UNGUENTO OFTALMICO AL 1% TUBO 3-5G. COTIZAR POR GRAMO.	0.0000	55442	6	12/01/2016
CLOROQUINA FOSFATO TABLETA 250MG. COTIZAR POR TABLETA.	0.0000	5544778	6	12/01/2013

Fuente: elaboración propia.

- **Informe logístico:** la aplicación permite consultar e imprimir un informe logístico mensual de los insumos, de acuerdo a los rubros disponibles.

Apéndice 10j. **Informe logístico**



RENGLON 266

2012 febrero

Informe logístico para el rubro RENGLON 266

Descargar para imprimir

1

Suministro	Saldo anterior	Entradas del nivel superior	Reajustes (+)(-)	Saldo mes siguiente	Existencia física en bodega	Total distribuído
ACECLOFENACO TABLETA 100MG. COTIZAR POR TABLETA.	50000	0	0	50000	50000	0
ACETAMINOFEN 120MG/5ML. JARABE FRASCO DE 120ML. COTIZAR POR FRASCO.	500	0	0	500	500	0
ACETAMINOFEN 80MG. TABLETA O CAPSULA. COTIZAR POR TABLETA O CAPSULA.	84000	0	0	84000	84000	0
ACIDO FOLICO TABLETA 5MG. COTIZAR POR TABLETA.	500	0	0	500	500	0
ACIDO ZOLENDRONATO 4MG VIAL. DEBE CUMPLIR CON LOS REQUERIMIENTOS DE ZOLENDRONIC ACID PDR 57 EDITION, USP DI 21TH EDITION.	500	0	0	500	500	0
ALBENDAZOL SUSPENSION 200 MG/5ML. FRASCO 10-20ML. COTIZAR POR MILILITRO.	800000	0	0	800000	800000	0
AMOXICILINA CAPSULA O COMPRIMIDO 500MG. COTIZAR POR CAPSULA O COMPRIMIDO.	577841	0	0	577841	577841	0

Distribución

Distrito	Cantidad
Escuintla	0
Guanagazapa	0
Iztapa	0
La Democracia	0
La Gomera	0
Masagua	0
Nueva Concepcion	0
Palin	0
Puerto San Jose	0
San Vicente Pacaya	0
Santa Lucia Cotzumalguapa	0
Siquinala	0
Tiquisate	0
ETV	0

Fuente: elaboración propia.