

Implementación de un sistema de información bajo la metodología ágil Open UP para el control de los procesos compra, venta e inventario de la Botica Medicfarma en Tarapoto, San Martín

Persy Quiroz-Menor¹

Recibido 14 de diciembre de 2014, aceptado 21 de enero de 2015

Received: December 14, 2014

Accepted: January 21, 2015

RESUMEN

El presente artículo tiene la finalidad de informar sobre el desarrollo de una solución informática para apoyar los procesos de comercialización de la Botica Medicfarma en Tarapoto, San Martín. Se construyó un sistema de información web que captura la información de los procesos de compra, venta, inventario y caja para ayudar a la administración de esta empresa. Se aplicó la metodología ágil de desarrollo Open UP, que consta de las fases de inicio, elaboración, construcción y transición, y sus respectivas disciplinas que son: modelo de negocio, requerimientos, análisis y diseño, implementación, pruebas, entrega, configuración y administración del proyecto. La solución fue desarrollada con el lenguaje de programación Java, bajo el patrón de diseño Modelo Vista Controlador (MVC), y la información es almacenada en el motor de base de datos PostgreSQL.

Palabras clave: Metodologías ágiles, Open Unified Process, MVC, base de datos, PostgreSQL, Java.

ABSTRACT

This article aims to inform the development of a software solution to support the processes of marketing Botica Medicfarma in Tarapoto, San Martin. A web information system that captures information on the processes of buying, selling, inventory and cash management to help this company was built. Business model, requirements, analysis and design, implementation, testing, delivery, configuration and management: Agile development methodology Open UP, comprising the steps

¹ Bachiller en Ingeniería de Sistemas. Universidad Peruana Unión. Email: persy.quiroz@upeu.edu.pe

of initiation, elaboration, construction and transition, and their respective disciplines are applied the project. The solution was developed with the Java programming language, under the design pattern Model View Controller (MVC), and the information is stored in the engine PostgreSQL database.

Keywords: Agile, Open Unified Process, MVC, database, PostgreSQL, Java.

INTRODUCCIÓN

La información es considerada una parte fundamental para toda empresa, los gerentes toman decisiones basadas en ella. La tecnología, sin lugar a dudas, ya forma parte de la vida y se ha vuelto indispensable para toda organización. Cada día se escucha hablar de nuevas soluciones para satisfacer necesidades de los usuarios o para incrementar la productividad y/o utilidades de las empresas. La tecnología está presente en todo lugar, ahora se cuentan con equipos tecnológicos con gran nivel de procesamiento y con gran capacidad de almacenamiento, cuando antes ni se lo imaginaba. Pero, con todo eso, no se han eliminado los problemas.

En la actualidad hay un sinfín de frameworks y tecnologías que se pueden usar para implementar soluciones robustas y eficientes. En el proceso de los años se han desarrollado herramientas para almacenamientos de datos sofisticados conocidos como gestores de base de datos. Y existe una variedad de lenguajes de programación para la creación de softwares para gestionar datos y presentarlos al usuario final de una forma interactiva, que es de gran ayuda para la toma de decisiones y llevar un control estricto de sus procesos de negocio.

El caso de estudio es una empresa que se dedica a la comercialización de productos farmacéuticos en la ciudad de Tarapoto. Los problemas generales son la falta de control en sus procesos de compra y venta de sus productos y la administración de su inventario, lo cual ocasiona pérdida de dinero y pérdida de tiempo en el proceso, junto con ello se incrementa el nivel de desconfianza hacia sus trabajadores.

MARCO TEÓRICO

Ingeniería de software

La ingeniería de software es el establecimiento y uso de principios de ingeniería robusta, orientada a obtener softwares económicos que sean fiables y funcionen de manera eficiente sobre máquinas reales. Abarca tres elementos claves: métodos, herramientas y procedimientos (Gutiérrez & Payares, 2009).

Metodologías ágiles

Las metodologías ágiles son formas de administrar todo el proceso de implementación de proyectos de software en el menor tiempo posible y entregando un producto de calidad.

Según (Pressman, 2010) afirma que “La agilidad es más que una respuesta efectiva al cambio. Fomenta una buena comunicación entre todas las partes involucradas. Se pone el énfasis en la entrega rápida de software funcional. Adopta al cliente como parte del equipo de desarrollo. Reconoce que los planes del proyecto tienen sus límites y que, al mismo tiempo, es flexible”.

Pressman, en la Tabla 1, presenta los principios generales de agilidad para construir softwares que satisfagan las necesidades del usuario final (cliente).

Tabla 1

Principios de agilidad según (Pressman, 2010)

N.º	Descripción del principio
1	La prioridad más alta es satisfacer al cliente a través de la entrega pronta y continua de software valioso.
2	Son bienvenidos los requerimientos cambiantes, aun en una etapa avanzada de desarrollo. Los procesos ágiles dominan el cambio para provecho de la ventaja competitiva del cliente.
3	Entregar con frecuencia software que funcione, de dos semanas a un par de meses, de preferencia lo más pronto que se pueda.
4	Las personas de negocios y los desarrolladores deben trabajar juntos, a diario y durante todo el proyecto.
5	Hay que desarrollar los proyectos con individuos motivados. Deben darte a estos el ambiente y el apoyo que necesiten, y confiar en que harán el trabajo.
6	El método de una buena comunicación cara a cara.
7	Los indicadores principales del avance es el software que funcione.
8	La atención continua a la excelencia técnica y un buen diseño mejora la agilidad.
9	Es esencial la simplicidad: el arte de maximizar la cantidad de trabajo no realizado.
10	Las mejores arquitecturas, requerimientos y diseños surgen de los equipos con organización propia.
11	Se trabaja constantemente con la mejora continua, se busca ser eficaz.

Existen muchas metodologías ágiles, entre las cuales se encuentran: Programación Extrema (XP), Desarrollo adaptativo de software (DAS), Scrum (uno de los más usados), Método de desarrollo de sistemas dinámicos (MDS), Cristal, Desarrollo impulsado por las características (DIC), Desarrollo esbelto de software (DES), Modelado Ágil (MA), Proceso Unificado Ágil (PUA), Proceso Unificado Abierto (Open UP).

Proceso Unificado Abierto (Open UP)

“Open UP es un proceso unificado que aplica enfoques iterativos e incrementales, dentro de un ciclo de vida estructurado. Open UP abraza una filosofía pragmática y ágil que se centra en la naturaleza colaborativa de desarrollo de software. Se trata de un proceso de baja ceremonia herramientas-agnósticos que se puede ampliar para hacer frente a una amplia variedad de tipos de proyectos” (Eclipse, 2012).

Open UP estructura el ciclo de vida del proyecto en cuatro fases: Inicio, Elaboración, Construcción y Transición. El ciclo de vida del proyecto proporciona las partes interesadas y los miembros del equipo con la visibilidad y la toma puntos a lo largo del proyecto.

Java

“Java se ha convertido en un lenguaje estándar de la industria para el desarrollo de aplicaciones web en internet y es un lenguaje de programación que tiene características excelentes para el aprendizaje” (Joyanes, 2003). Java fue desarrollado por Sun Microsystems en 1995, es un magnífico lenguaje orientado a objetos diseñados para distribuir contenidos a través de una red.

Las características de Java son las siguientes: lenguaje orientado a objetos, lenguaje interpretado, lenguaje seguro, lenguaje orientado al trabajo en red, lenguaje preparado para contenidos multimedia (Vásquez & Balta, 2006).

Modelo Vista Controlador (MVC)

El patrón conocido como Modelo-Vista-Controlador (MVC) separa el modelado del dominio, la presentación y las acciones basados en datos ingresados por el usuario en tres clases diferentes. Donde el Modelo administra el comportamiento y los datos del dominio de aplicación. La Vista maneja la visualización de la información. El Controlador interpreta las peticiones del usuario, informando al modelo y/o a la vista para que cambien según resulte apropiado (Reynoso & Kicillof, 2004).

Programación Orientada a Objetos

La Programación Orientada a Objetos (OOP, por las siglas inglesas de Object-Oriented Programming) es una nueva forma de programar que proliferó a partir de los años ochenta. Se usa los conceptos de objeto, clase, herencia, encapsulamiento, polimorfismo. Así como métodos, mensajes y propiedades (López, 2010).

Base de datos

Según (Oppel, 2010) dice que una base de datos “es un conjunto de elementos de datos interrelacionados, administrados como unidad”. Asimismo, (Sabana, 2006) argumenta que una base de datos es una colección de datos estructurados según

un modelo que refleje las relaciones y restricciones en el mundo real. Asimismo, los procesos que sufran estos datos tendrán que conservar la integridad y seguridad.

SQL (Lenguaje de Consulta Estructurado)

Structured Query Language o Lenguaje de Consulta Estructurado es el que permite la comunicación con el Sistema de Gestión de Base de Datos. Es una herramienta para organizar, gestionar y recuperar datos almacenados en una Base de Datos informática. Las sentencias SQL pertenecen a dos categorías principales: Lenguaje de Definición de Datos (DDL) y Lenguaje de Manipulación de Datos (DML). El DDL permite definir los objetos de la Base de Datos y el DML se utiliza para manejar los datos correspondientes (Sabana, 2006).

Sistemas de Gestión de Bases de Datos (SGDB/DBMS)

Los Sistemas de Gestión de Bases de Datos (SGDB) son un tipo de software muy específico, dedicado a servir de interfaz entre las bases de datos y las aplicaciones que la utilizan, consiguiendo que el acceso a los datos se realice de una forma más eficiente, más fácil de implementar, y sobre todo, más segura (Sabana, 2006).

El sistema de administración de bases de datos (DBMS) es el software proporcionado por el vendedor de la base de datos. Productos de software como Microsoft Acces, Oracle, Microsoft SQL Server, Sybase ASE, DB2, Ingres y MySQL son DBMS (Oppel, 2010).

PostgreSQL

Es un sistema avanzado de administración de base de datos-objeto relacionales (ORDBMS) de código abierto (gratuito y código fuente disponible). Posee interoperabilidad con otros DBMS: SQL. El almacenamiento es confiable, consistente y robusto. La manipulación es potente, flexible y eficiente (Sabana, 2006).

Servidores web

Un servidor web o servidor HTTP es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente y generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo. Generalmente se usa el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación del modelo OSI. Entre los servidores podemos encontrar: Apache HTTP Server, Baby Web Server, Oracle HTTP Server, Xampp, Internet Information Services (IIS), Apache Tomcat y muchos más.

Apache Tomcat

Apache Tomcat es una implementación de código abierto de software de las tecnologías Java Servlet y JavaServer Pages. Las especificaciones Java Servlet y JavaServer Pages son desarrollados bajo la Java Community Process. Apache Tomcat se desarrolla en un entorno abierto y participativo y publicado bajo la licencia Apache versión 2. Apache Tomcat está destinada a ser una colaboración de los desarrolladores mejor de su clase en todo el mundo (Tomcat, 2014).

INSTRUMENTOS

Metodología

La metodología empleada para el desarrollo de esta investigación se basó en el método de desarrollo ágil Open UP, es un método que aplica enfoques iterativos e incrementales. Para ello se elaboró el plan de actividades del proyecto en el diagrama de Gantt.

Nombre de tarea	Duración	Comienzo	Fin
• Desarrollo de la metodología	114 días	mar 24/06/14	lun 01/12/14
• Concepción	3 días	mar 15/07/14	jue 17/07/14
Reunir catálogo de requerimientos	2 días	mar 15/07/14	mié 16/07/14
Especificar los requisitos de software	1 día	jue 17/07/14	jue 17/07/14
• Elaboración	24 días	vie 18/07/14	jue 21/08/14
Elaborar documento de análisis del sistema	5 días	vie 18/07/14	mar 22/07/14
Elaborar doc. de arq. del sistema	2 días	mié 23/07/14	jue 24/07/14
Elaborar doc. del diseño del sistema	6 días	vie 25/07/14	mié 30/07/14
Elaborar doc. de diseño de base de datos	7 días	jue 31/07/14	mié 06/08/14
Elaborar doc. de estándares de programación	5 días	jue 07/08/14	lun 11/08/14
Elaborar doc. de estándares de interfaz gráfica	4 días	mar 12/08/14	vie 15/08/14
Elaborar plan de pruebas	2 días	lun 18/08/14	mar 19/08/14
Confirmar actas de aceptación	2 días	mié 20/08/14	jue 21/08/14

Figura 1. Fase de concepción y elaboración del proyecto.

Nombre de tarea	Duración	Comienzo	Fin
➤ Construcción	72 días	vie 22/08/14	lun 01/12/14
➤ Iteración I	15 días	vie 22/08/14	jue 11/09/14
Implement. Y pruebas del módulo de ventas	14 días	vie 22/08/14	jue 04/09/14
Ejecución de pruebas completas - I	4 días	vie 05/09/14	lun 08/09/14
Versión I - sistema de información	1 día	mar 09/09/14	mar 09/09/14
Elaboración de diagramas de secuencia I	2 días	mié 10/09/14	jue 11/09/14
➤ Iteración II	25 días	vie 05/09/14	jue 09/10/14
Implement. Y pruebas del módulo de gestión de compras	20 días	vie 12/09/14	jue 09/10/14
Ejecución de pruebas completas - II	5 días	vie 05/09/14	mar 09/09/14
Versión II - sistema de información	2 días	mié 10/09/14	jue 11/09/14
Elaboración de diagramas de secuencia II	4 días	vie 12/09/14	lun 15/09/14
➤ Iteración III	19 días	mar 16/09/14	vie 10/10/14
Implement. Y pruebas del módulo de registros e inventarios	16 días	mar 16/09/14	mié 01/10/14
Ejecución de pruebas completas - III	5 días	jue 02/10/14	lun 06/10/14
Versión III - sistema de información	1 día	mar 07/10/14	mar 07/10/14
Elaboración de diagramas de secuencia III	3 días	mié 08/10/14	vie 10/10/14
➤ Iteración IV	19 días	lun 13/10/14	jue 06/11/14
Implement. Y pruebas del módulo de reportes	16 días	lun 13/10/14	mar 28/10/14
Ejecución de pruebas completas - III	3 días	mié 29/10/14	vie 31/10/14
Versión IV - sistema de información	1 día	lun 03/11/14	lun 03/11/14
Elaboración de diagramas de secuencia III	3 días	mar 04/11/14	jue 06/11/14
➤ Iteración V	25 días	vie 07/11/14	lun 01/12/14
Implement. Y pruebas del módulo de seguridad y configuraciones	14 días	vie 07/11/14	jue 20/11/14
Ejecución de pruebas completas - IV	5 días	vie 21/11/14	mar 25/11/14
Versión V - sistema de información	1 día	mié 26/11/14	mié 26/11/14
Elaboración de diagramas de secuencia IV	6 días	jue 27/11/14	lun 01/12/14
➤ Transición	8 días	mar 24/06/14	mar 01/07/14
Implantación del sistema	1 día	mar 24/06/14	mar 24/06/14
Capacitación al personal	3 días	mié 25/06/14	vie 27/06/14
Cierre de la documentación del proyecto	2 días	lun 30/06/14	mar 01/07/14

Figura 2. Fase de Construcción y Transición del proyecto.

Tecnología

El sistema de información fue elaborado en una plataforma web haciendo uso de NetBeans IDE 7.1.2, con código fuente Java, con vista JSP, bajo el patrón de diseño MVC (Modelo Vista Controlador). Además se usaron estilos CSS, JavaScript y Ajax. Para el almacenamiento de los datos se usó la versión de PostgreSQL 9.0.

En la fase de elaboración, tras el levantamiento de la información de los requerimientos funcionales del sistema, se diseñó la base de datos con el modelador Case Studio 2.0. A continuación se presentan la lista de tablas resultantes con el modelo Entidad-Relación:

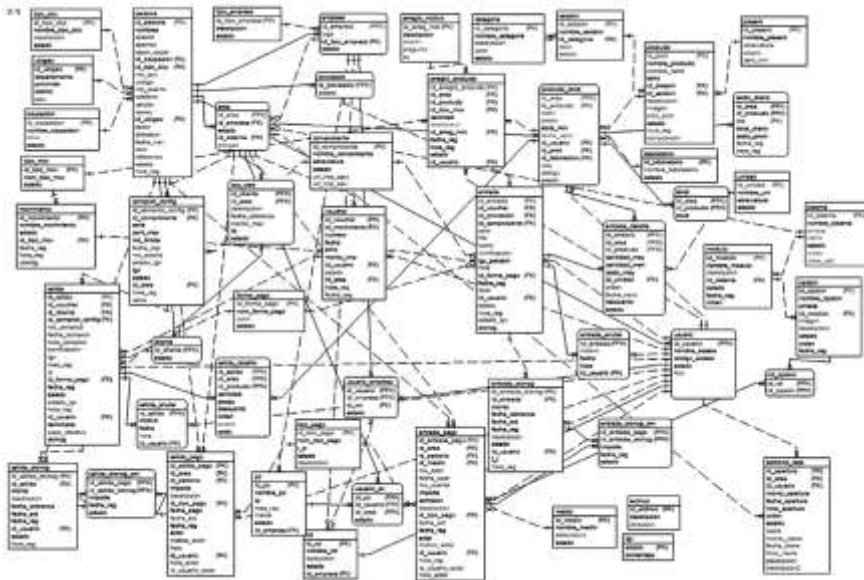


Figura3. Modelado de base de datos por Entidad-Relación.

Arquitectura del sistema de información

Se aplicó el patrón de diseño Modelo Vista Controlador (MVC) con Programación Orientada a Objetos (POO).

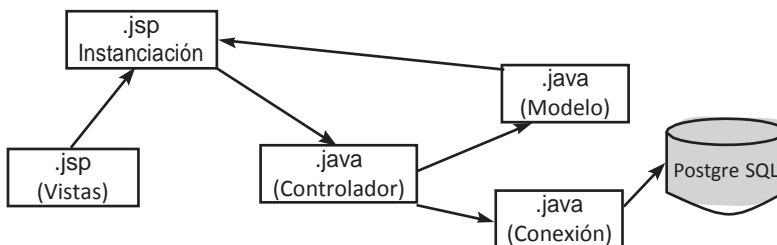


Figura4. Arquitectura del sistema basado en el Modelo Vista Controlador.

RESULTADOS

Los módulos generales del sistema son los siguientes: VENTAS, COMPRAS, REGISTROS, INVENTARIOS Y SEGURIDAD DEL SISTEMA.

Módulo de autenticación: Página de autenticación donde el administrador o trabajadores ingresan con usuario y contraseña para acceder a los módulos del sistema.



Figura 5. Módulo de autenticación.

Módulo menú del sistema: Esta página presenta las opciones a los módulos del sistema, en la parte superior se presentan los subsistemas como servicios, personal y seguridad general.



Figura 6. Módulo menú del sistema.

Módulo de ventas: Esta página permite el proceso de registro de las ventas, primeramente se hace la búsqueda del producto farmacéutico, se buscan o se registran los clientes.

Figura 7. Módulo de ventas.

Módulo de registro de cliente: Esta página permite el registro de los datos del cliente tales como datos personales, procedencia, dirección, edad, entre otros.

Figura 8. Módulo de registro de clientes.

Módulo de compras y transferencias: Esta página permite el registro de compras o entradas de productos a la botica. Se registran la información de los productos, lo que incluye la cantidad al por mayor, costo, número de lote y fecha de vencimiento.



Figura 9. Módulo de compras y transferencias.

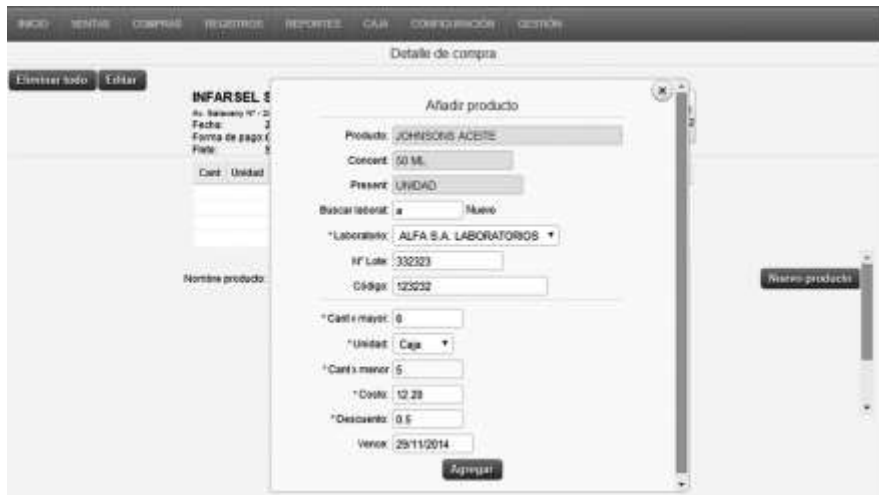


Figura 10. Módulo de detalle de compra.

Módulo de reporte de ventas: Permite el reporte de las ventas hechas diariamente y entre dos fechas. Presenta los totales vendidos por usuarios, con tipo de comprobante, con la forma de pago.

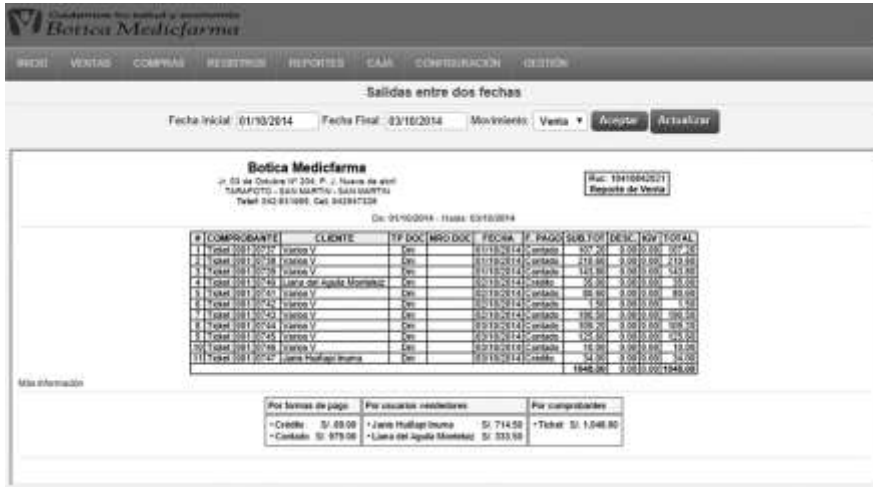


Figura 11. Módulo de reporte de ventas.

Módulo de reporte de compras y transferencias: Permite el reporte de las compras o transferencias recibidas diariamente y entre dos fechas. Presenta los totales de descuentos, IGV, flete.



Figura 12. Módulo de reporte de compras y transferencias.

Módulo reporte de inventario: Permite el reporte de los stock de los productos farmacéuticos agrupándolo por categorías, presentaciones. Se presenta la información general de cada producto, el porcentaje de utilidad, y el total de dinero en productos.

ID	PRODUCTO	LOTE	FECHA VENC	STOCK	COSTO	V. COSTO	INVENTARIO	%
1	AEROSOL							
2	AMPOLLA							
3	AMPOLLA							
4	AMPOLLA							
5	AMPOLLA							
6	AMPOLLA							
7	AMPOLLA							
8	AMPOLLA							
9	AMPOLLA							
10	AMPOLLA							
11	AMPOLLA							
12	AMPOLLA							
13	AMPOLLA							
14	AMPOLLA							
15	AMPOLLA							
16	AMPOLLA							
17	AMPOLLA							
18	AMPOLLA							
19	AMPOLLA							
20	AMPOLLA							

Figura 13. Módulo reporte de inventario.

Módulo reporte de caja: Permite el reporte de los ingresos y egresos del dinero en caja.

#	Fecha	Motivo	Persona referencia	Descripción	Ingresos	Egresos
1	03/11/2014	Compras	COMERCIAL SAC	compra		219.90
2	03/11/2014	Compras	REP AEROSOL SAC	rep aerosol		514.50
3	03/11/2014	Compras	LABORATORIOS MEDICARMA SAC	rep comprobante		219.90
4	03/11/2014	Compras	LABORATORIOS MEDICARMA SAC	rep comprobante		498.24
5	03/11/2014	Compras	LABORATORIOS MEDICARMA SAC	rep comprobante		115.80
6	03/11/2014	Compras	COMERCIALIZADORA DE LA JAMAICA SAC	compra		64.50
7	03/11/2014	Compras	JAMES PASTAGAL INGENIERIA	compra		632.30
8	03/11/2014	Compras	Luz de la Cruz	compra		305.20
9	03/11/2014	Compras	Luz de la Cruz	compra		60.80
10	03/11/2014	Compras	JUAN DE LA CRUZ LLORCA	compra		214.80
11	03/11/2014	Compras	Luz de la Cruz	compra		305.20
12	03/11/2014	Compras	Algo Chavez	compra		800.80
13	03/11/2014	Ventas	Venta V	Pago al comprobante	85.76	
14	03/11/2014	Ventas	Venta V	Pago al comprobante	171.50	
15	03/11/2014	Ventas	Venta V	Pago al comprobante	30.50	
16	03/11/2014	Ventas	Venta V	Pago al comprobante	22.30	
17	03/11/2014	Ventas	Venta V	Pago al comprobante	17.00	
18	03/11/2014	Ventas	Venta V	Pago al comprobante	4.00	
19	03/11/2014	Ventas	Venta V	Pago al comprobante	130.10	
20	03/11/2014	Ventas	Venta V	Pago al comprobante	21.50	
21	03/11/2014	Ventas	Venta V	Pago al comprobante	21.50	
22	03/11/2014	Ventas	Venta V	Pago al comprobante	21.50	

Figura 14. Módulo reporte de caja.

Módulo de seguridad: Permite el registro de usuarios, creación de roles y asignación de accesos a cada rol creado.

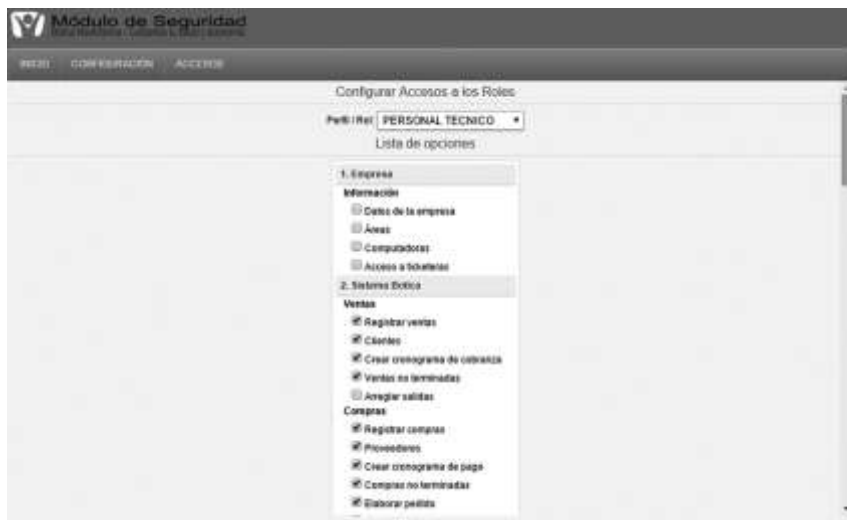


Figura 15. Asignación de accesos a roles, módulo de seguridad.

CONSIDERACIONES FINALES

Al concluir la presente investigación, se cuenta con un sistema automatizado para el control de los procesos compra, venta, inventario y caja de la Botica Medicfarma. Cabe mencionar solo algunos beneficios importantes, tales como:

- Se tiene el inventario disponible en tiempo real.
- Se controla mejor las fechas de vencimiento de los productos, y con ello se evita pérdidas económicas para la empresa.
- Se tiene los precios de los productos al instante.
- Se tiene el control de los ingresos y egresos de caja, para evitar robo de dinero o tal vez desconfianza.
- Mejoró los procesos de compra y venta de productos, y el personal nuevo no tiene problemas con el precio de los productos, porque el sistema le reporta al instante.
- Los totales son calculados automáticamente y se evita el uso de calculadoras, lo cual ocasiona tiempo para el cálculo.
- Se logró tener control de las utilidades generales en la empresa, y esto ayuda para la toma de decisiones administrativas.

RECOMENDACIONES

Para los estudiantes de la especialidad de Ingeniería de Sistemas, considerar lo siguiente: recomiendo tener un buen trato con el cliente, así como cumplir con las entregas del producto software acordado. Es muy importante conocer bien lo que se quiere construir, teniendo los objetivos y requerimientos bien claros. Sea ingenioso en lo posible y no se desanime cuando haya cosas difíciles que hacer. Y, finalmente, domine un lenguaje de programación muy bien y diseñe una buena base de datos.

Referencias

- Eclipse, F. (2012). Open UP. Retrieved from <http://epf.eclipse.org/wikis/openup/index.htm>
- Gutiérrez, P. R., & Payares, V. J. (2009). Desarrollo de un Sistema para la Automatización Distribución y Control del Stock de Materiales de la Fundación Misión Ribas Anzoátegui. Universidad de Oriente.
- Joyanes, L. (2003). Fundamentos de programación. (C. Fernández Madrid & A. Nieva, Eds.) (Tercera ed., 973). Aravaca - Madrid (España).
- López, A. (2010). La programación orientada a objetos y la productividad en el desarrollo de software. Retrieved from http://depto09.sistemasitl.com/index.php?option=com_content&view=article&id=142:articulo-a-lopez&catid=44:informativocomunidad&Itemid=125
- Oppel, A. (2010). Fundamentos de bases de datos. (F. Castellanos Rodríguez, M. Á. Luna Ponce, & Z. García García, Eds.) (Primera ed., 478). Mexico, D.F.
- Pressman, R. S. (2010). Ingeniería de software, un enfoque práctico. (M. Á. Toledo Castellanos, P. Roig Vázquez, M. I. Rocha Martínez, M. T. Zapata Terrazas, & Z. García García, Eds.) (Séptima ed., 749). Mexico.
- Reynoso, C., & Kicillof, N. (2004). Estilos y patrones en la estrategia de arquitectura de Microsoft. Retrieved from <http://carlosreynoso.com.ar/archivos/arquitectura/Estilos.PDF>
- Sabana, M. (2006). Modelamiento e implementación de base de datos. (E. Aburto Correa, M. Sabana Mendoza, & G. Apolinario García, Eds.) (Primera ed., 631). Lima, Perú.
- Tomcat, A. (2014). Apache Tomcat. Retrieved from <http://tomcat.apache.org/>
- Vasquez, J., & Balta, V. (2006). Super Java 2 for Windows (Segunda ed., 586). Lima - Perú.