

# SISTEMA WEB PARA EL MANEJO DE INVENTARIO DE MULTIBODEGAS, GESTIÓN CONTABLE, Y CREACIÓN DE UN REPOSITORIO DIGITAL DE INFORMES TÉCNICOS, UTILIZANDO DSPACE, TECNOLOGÍA PRIMEFACES Y HERRAMIENTAS OPEN SOURCE, PARA LA EMPRESA CROSSTRONIK CIA. LTDA.

*Diana Cruz Caiza<sup>1</sup>, Christian Zhamungui Oviedo<sup>2</sup>, Alex Jiménez Vélez<sup>3</sup>, Freddy Dueñas Mora<sup>4</sup>*

1 Departamento de Ciencias de la Computación, Quito-Ecuador, dccc\_18@hotmail.com

2 Departamento de Ciencias de la Computación, Quito-Ecuador, chrisxavi1988@hotmail.com

3 Departamento de Ciencias de la Computación, Quito-Ecuador, afjimenezv@gmail.com

4 Departamento de Ciencias de la Computación, Quito-Ecuador, faduenias@espe.edu.ec

## RESUMEN

*El presente trabajo muestra el desarrollo de un sistema web que permite automatizar los procesos de control de Inventario de multibodegas, gestión de los principales procesos contables, y la publicación de documentos (informes técnicos) en la web, a través de la utilización/acoplamiento del Repositorio Digital DSpace a las necesidades de la empresa Crosstronik Cía. Ltda. El sistema web también toma en consideración los procesos de generación de Órdenes de trabajo, y Cotizaciones.*

*Para estructurar, planificar y controlar el proceso de desarrollo del sistema, se utilizó la metodología de desarrollo ágil de software, Programación Extrema (X.P); porque es una metodología bastante flexible que promueve un desarrollo iterativo e incremental, y la retroalimentación frecuente del código; que son aspectos relevantes para el desarrollo de este proyecto.*

*La herramienta desarrollada lleva el nombre CTKSYS. Se trata de un sistema web distribuido desarrollado en Java, haciendo uso del entorno de desarrollo Eclipse Índigo. Utiliza el motor de base de datos PostgreSQL y como servidor de aplicaciones Jboss AS 6.1. Todas las herramientas utilizadas son software libre, con el fin de utilizar las últimas tecnologías de vanguardia y disminuir costos por licencias.*

*Con la implementación y utilización adecuada de la herramienta desarrollada, se optimiza el manejo de los procesos internos de la empresa, tiempo y organización; todo en su conjunto genera satisfacción al usuario interno y externo, brindando una imagen corporativa robusta.*

**Palabras Clave:** Inventario, DSpace, Java, PostgreSQL, Programación Extrema.

## ABSTRACT

*This paper shows the development of a web system that allows to automate the processes of Inventory control of multibodegas, management of the main accounting processes, and the publication of documents (technical reports) on the web, through the use of the DSpace repository software according to the needs of Crosstronik Cia. Ltda. Company. The web system also takes into account the processes of generating Work Orders, and Quotes.*

*For the structure, planning and control of the development process of the system, there was used the agile software development methodology, Extreme Programming (X.P), because it is a very flexible methodology that promotes an iterative and incremental development, and the frequent feedback of the code, which are relevant aspects to the development of this project.*

*CTKSYS is the name of the tool that was developed. It is a distributed web system developed in Java, using the integrated development environment Eclipse Indigo. It uses the object-relational database management system PostgreSQL and the application Server Jboss AS 6.1. All these tools are Free Software, in order to implement the latest cutting-edge technologies and reduce licensing costs.*

*With the implementation and the appropriate use of the developed tool, optimizes the management of the internal business processes, time and organization; all as a whole generates the satisfaction of the internal and external user, providing a strong corporate image.*

**KeyWords:** Inventory, DSpace, Java, PostgreSQL, Extreme Programming.

## 1. INTRODUCCIÓN

La finalidad de este proyecto es desarrollar un sistema web para la empresa Crosstronik Cía. Ltda., basado en una metodología de desarrollo ágil, la aplicación de tecnologías actuales, e investigación de herramientas de software libre para gestionar archivos digitales, controlar y administrar los movimientos de materiales de cualquier bodega, y llevar un control de la información contable.

Actualmente la empresa Crosstronik Cía. Ltda., no cuenta con una herramienta informática que automatice los procesos internos que se realizan para el desarrollo de proyectos, control de inventario de productos en sus distintas bodegas, gestión contable, y publicación de informes técnicos.

Sería realmente útil para la empresa, reemplazar los archivos de base de datos que tienen en Microsoft Access, los archivos de formatos que tienen en Microsoft Excel, y en medios físicos (hojas), para llevar el control de inventario de productos que se encuentra en sus bodegas; pues existe un manejo erróneo de éstos archivos al no disponer de una almacenamiento organizado ni centralizado de la información, lo que produce que los datos estén desactualizados o sin disponibilidad para cuando se los requiere.

Con el desarrollo de este sistema web se cumple con el objetivo principal de llevar un control de las entradas y salidas de productos pertenecientes a una bodega en cualquier ubicación, así como también la automatización de los procesos para la generación de Órdenes de Trabajo y Cotizaciones. Además de gestionar los procesos de apoyo contable tales como: la generación de comprobantes de ingreso, egreso, facturas y reportes contables (Libro Diario, Libro Mayor, Balance de Comprobación, y Estado de Resultados), finalmente solventa el requerimiento de gestionar la publicación/subida, búsqueda y descarga de informes técnicos que genera la empresa a través de la utilización del repositorio digital Dspace.

En la primera parte de este documento, se muestra la metodología utilizada en el desarrollo del sistema, los modelos de diseño y arquitectura del software, materiales y métodos; luego una descripción de los resultados obtenidos, finalmente se detallan las conclusiones y trabajo futuro.

## 2. METODOLOGÍA

Para la realización del presente proyecto fue necesario escoger una metodología para lo cual se realiza un análisis comparativo como se muestra en la Tabla I.

**Tabla I: Matriz de comparación entre las Metodologías candidatas**

CRITERIOS	METODOLOGÍAS		
	Microsoft Solutions Framework (MSF)	Programación Extrema (XP)	Proceso Unificado Ágil (AUP)
Pocos roles y flexibles	2	3	1
Iterativa	1	3	1
Entorno amplio de proyectos de Software	2	3	2
Permite desarrollar software sobre cualquier tecnología	2	2	3
Nivel de conocimiento del grupo de trabajo	0	3	1
Soporte de Orientación a Objetos	3	3	3
Afinidad con el sistema	2	3	2
No presenta Resistencia a los cambios	1	3	1
Proceso Controlado	2	2	2
Comunicación con el cliente	1	3	1
<b>TOTAL:</b>	<b>16</b>	<b>28</b>	<b>17</b>

Valores:

- Nivel Alto=3
- Nivel Medio=2
- Nivel Bajo=1
- Nivel Nulo=0

Realizado el análisis comparativo entre las tres Metodologías de Desarrollo Ágil; Extreme Programming (X.P) resultó ser la más idónea para el desarrollo de este sistema, debido a que se verían aplicadas las características fundamentales que plantea la metodología, como: el desarrollo iterativo e incremental, realización de pruebas unitarias continuas, programación por parejas, interacción frecuente del equipo de trabajo con el cliente/usuario, corrección de errores antes de añadir nueva funcionalidad, y la refactorización de código [1].

Un punto decisivo durante la selección de metodología fue que la aplicación de ésta es bastante adaptable, no exige la realización de diagramas –sino los que el desarrollador considere que le pueden ser útiles-. [2] Además, era evidente que se debía desarrollar un sistema capaz de adaptarse a los cambios de requisitos en cualquier fase del proyecto siendo un aspecto natural, inevitable e incluso deseable dentro del desarrollo de un sistema.

La evolución de las metodologías de desarrollo en cascada a ciclos iterativos más cortos y a la combinación que hace XP, como se muestra en la Fig. 1, facilita que se asuman de mejor manera los proyectos de cambios continuos, como el presente.

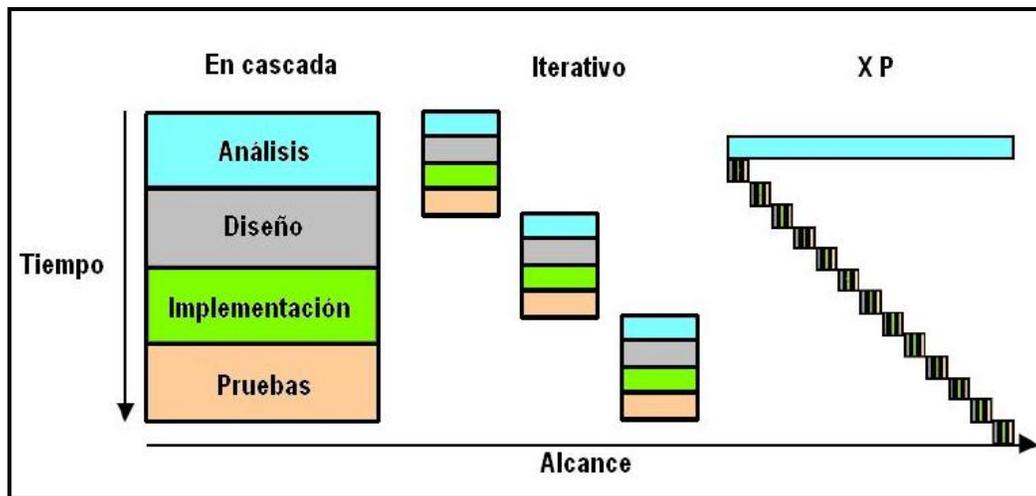


Fig. 1: Evolución de los ciclos de XP.

Las fases que se siguen en la metodología XP (BECK, 2005) son las siguientes:

- **Planificación.** Al iniciar el proyecto, se establecen reuniones con el cliente, en las cuales se planteará y acordará mediante Historias de Usuario todas las necesidades y funcionamiento del sistema; al mismo tiempo el equipo de trabajo comienza a explorar y familiarizarse con la tecnología que se va a utilizar en el desarrollo del sistema.
- **Diseño.** En base a las historias de usuario, se diseñan los modelos necesarios para el desarrollo del sistema, como el de base de datos, diagramas UML necesarios y se elaboran las interfaces, en cada uno de estos elementos se realiza una verificación continua con el cliente. Lo que permitirá obtener un prototipo inicial del software.
- **Desarrollo.** El grupo de trabajo categoriza cada historia de Usuario, para realizar una planificación y división modular del sistema que permita un avance continuo y auditable por parte del cliente; aplicando las sugerencias de desarrollo que propone ésta metodología. Al final del desarrollo de cada Historia de Usuario se evaluará conjuntamente con el cliente, y se efectuarán iteraciones hasta conseguir el resultado deseado.
- **Pruebas.** Una vez aceptado el funcionamiento del sistema, se procederá con pruebas de usuario, para las cuales se utilizan formatos que pueda ir aplicando el cliente, el cual será categorizado dentro del sistema para poder efectuar dichas pruebas; además permitirá al sistema obtener una interface más amigable para el usuario.
- **Finalización del Proyecto.** Una vez satisfechas todas las necesidades del cliente dentro del sistema y aprobados todos los test de pruebas se da por finalizado el sistema web para la empresa.[3]

### 3. MATERIALES Y METODOS

Para el desarrollo del presente proyecto se ha utilizado herramientas de software libre, los cuales se encuentran descritos en la Tabla II.

**Tabla II: Software Utilizado**

<b>JDK 1.7: Java Development Kit.</b>	Conjunto de herramientas de desarrollo para la creación, depuración y control de programas en lenguaje JAVA.[4]
<b>Eclipse Indigo 3.7 SR2:</b>	Es un IDE de código abierto y multiplataforma que proporciona un conjunto completo de herramientas de programación; además de un excelente ambiente para desarrolladores de aplicaciones integradas.[5]
<b>JBoss AS 6.1</b>	Es un servidor de aplicaciones desde Java2EE en adelante, de código abierto, que corre sobre cualquier sistema operativo para el que esté disponible la máquina virtual de Java, implementa por defecto las especificaciones: EJB 3.0, JPA 2.0 y Hibernate.[6]
<b>PostgreSQL 8.4</b>	Es un sistema gestor de bases de datos objeto-relación, de código abierto; utiliza el modelo cliente/servidor y uso de multiprocesos, lo que lo vuelve uno de los más robustos y estables. PostgreSQL maneja sin problema grandes cantidades de datos, además de soportar una alta concurrencia de usuarios.[7]
<b>Primefaces 3.2</b>	Es una librería de componentes visuales Open Source para JSF, que permite la creación o personalización de aplicaciones web; funciona de manera conjunta con tecnologías como Ajax y JPA, además permite a la aplicación ser ejecutada en dispositivos móviles sin programación extra.
<b>Dspace 1.8.2</b>	Es un sistema de repositorio digital, de código abierto, que permite recoger material digital, distribuirlos sobre la web a través de un sistema de búsqueda y recuperación; así como preservarlos a largo plazo.
<b>Jasper Reports 4.7.1</b>	Es una herramienta de creación de informes de tipo dinámico y permite generar ficheros en varias extensiones (PDF, HTML, XLS, CSV y XML) y una impresión de una forma simple y flexible. [8] Puede ser usado en cualquier aplicación Java, se usa comúnmente con un front-end gráfico de código abierto para la edición de informes, en el presente proyecto iReport-4.7.1.
<b>Visual Paradigm for UML Standard</b>	Es una herramienta para modelado UML de datos, arquitectura, diagramas de Entidad-Relación entre, otros; de manera fácil y ágil.

## 4. DISEÑO E IMPLEMENTACIÓN

Como paso previo al diseño de la aplicación se realizó el análisis de requerimientos mediante el levantamiento de requisitos, utilizando técnicas como entrevistas, y reuniones de trabajo; de manera que se pueda entender las necesidades y requerimientos que tiene el usuario final, respecto al software que desea.

Se obtienen cada uno de los requerimientos, tomando en cuenta que la metodología X.P, plantea la utilización de un formato de Historias de Usuario para especificar los requerimientos en la situación actual.

Posteriormente se plasmó en Diagramas de Casos de Uso, Diagrama de clases, diagrama entidad - relación, Diagrama de componentes y Diagrama de despliegue; cada uno de los requerimientos postulados en las Historias de Usuario. Adicionalmente se diseñó un Diagrama de flujo de trabajo para representar de manera más clara la parte del sistema referente a los procesos internos que se llevan a cabo en el área técnica, para la generación de Órdenes de Trabajo y Cotizaciones.

La conceptualización de cada uno de estos diagramas nos permite definir la arquitectura que tendrá el sistema, el mismo que se diseñó como una aplicación Java EE, por tanto, tiene una arquitectura multi-capas, que es una extensión de la arquitectura MVC (Modelo Vista Controlador), como se puede apreciar en la Fig. 2.

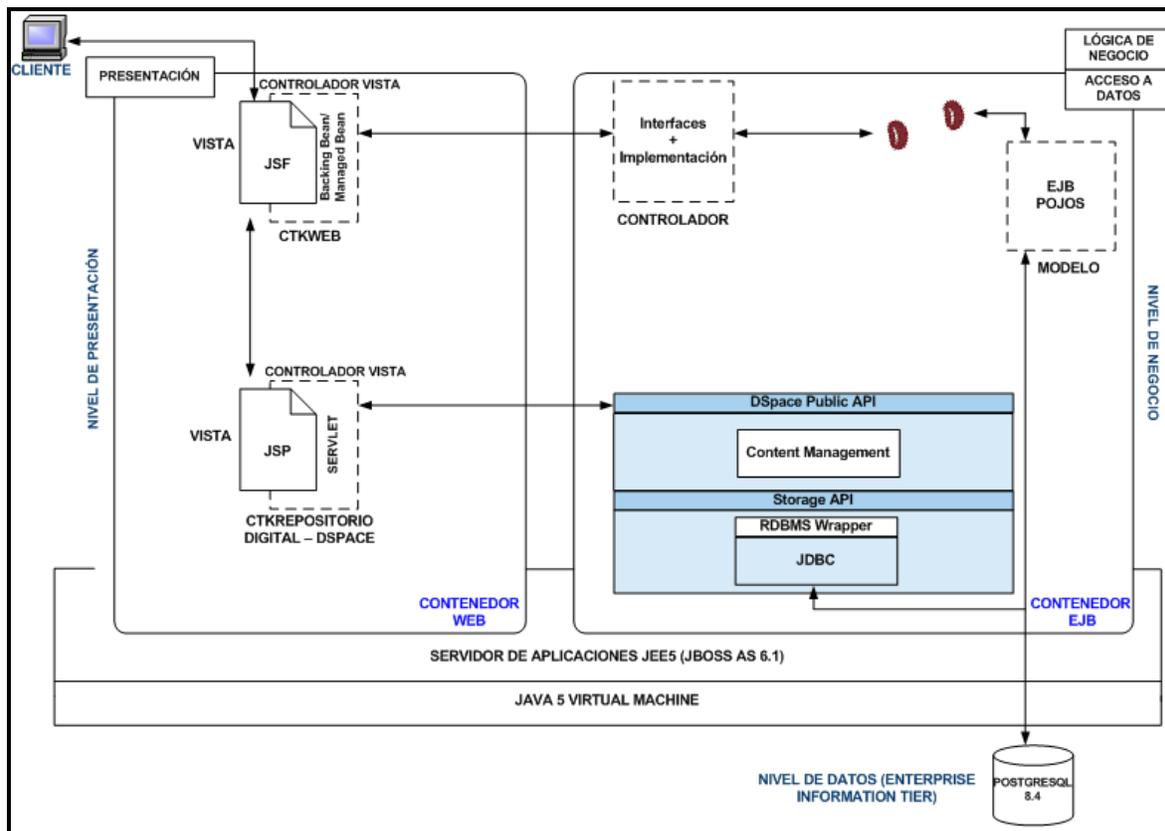


Fig. 2: Arquitectura del Sistema Web CTKSYS

De forma general, la solución de arquitectura propuesta divide a la aplicación en tres componentes: primero el modelo, constituido por las entidades/datos que son persistentes en el sistema y se almacenan dentro de la base de datos; luego el controlador, encargado de obtener, procesar y enviar los datos al cliente. Y finalmente la vista, que muestra la información en el formato adecuado al cliente. De ahí, como se muestra en la Fig.2, los tres componentes se encuentran divididos en varias capas.

De manera más específica, se puede decir que la aplicación CTKSYS (EAR) está formada por dos módulos Web (WAR): CTKREPOSITORIO y CTKWEB que usa el módulo CTKEJB (JAR). De forma conjunta, estos módulos cuentan con tres niveles:

El nivel de Negocio, está constituido por dos capas: Lógica de Negocio y Acceso a Datos. En el módulo CTKWEB, estas capas han sido implementadas con componentes EJB (Enterprise Java Beans v3.0) que hacen uso de JPA (Java Persistence API v2.0) – Hibernate como mecanismos de mapeo Objeto Relacional para proveer de una capa de acceso a datos.

En cuanto al módulo web CTKREPOSITORIO, se hace uso de las APIs: DSpace Public API y Storage API propias de Dspace para las negociaciones entre la vista y el modelo, además de implementar el API JDBC (Java Database Connectivity API) para el acceso a datos.

Para el nivel de Presentación, en el módulo CTKWEB, se implementó la vista basada en JSF (Java Server Faces v2.0) a través de la utilización del framework Primefaces.

En el módulo CTKREPOSITORIO se hace uso de la interface clásica de Dspace (JSPUI) implementada con JSP.

En el nivel de Datos para este sistema se utilizó PostgreSQL versión 8.4.

El modelo de arquitectura utilizado permitió implementar en su totalidad los requerimientos que exige el sistema y además, provee de una infraestructura escalable con un bajo nivel de dependencia entre cada uno de sus componentes.

## 5. RESULTADOS

En la metodología X.P. las pruebas, constituyen una de sus prácticas fundamentales. Esta actividad se realizó completamente en el presente trabajo. Existen dos tipos de pruebas: unitarias y de aceptación.

Las pruebas unitarias son aquellas que realiza el desarrollador, para la verificación de un módulo. Se realizaron estas pruebas de forma continua, cuando la interfaz de un método no era clara o la implementación era complicada. Se contempló para cada módulo del sistema sus posibles fallas. Para estas pruebas, no fue necesario escribir casos de prueba.

En XP las pruebas de aceptación deben consistir en un conjunto mínimo (no pobre ni insuficiente) de casos de prueba que cubran los requerimientos de negocios fundamentales planteados por el cliente [9]. Para escribir los casos de prueba de aceptación se utilizó el formato Test Case, como se lo puede apreciar en la Fig. 3.

Fecha Realización	<i>[ingresar fecha ejecución del test case]</i>				
Título del Caso de Prueba	<i>[ingresar el título del test case]</i>				
Núm. Caso de Prueba	<i>[ingresar el número del testcase, el mismo que ayudará para llevar un mejor control]</i>				
Objetivo	<i>[ingresar el objetivo que se pretende alcanzar con la prueba]</i>				
Resultado Esperado	<i>[ingresar a manejar general los resultados que se esperan de la prueba]</i>				
			<b>Resultado de la Prueba</b>		
<b>Paso</b>	<b>Procedimiento</b>	<b>Resultado Observado</b>	<b>Resultado de la Prueba</b>		
			<b>Pasa</b>	<b>Falla</b>	<b>Comentarios</b>
1	<i>[se detalla los pasos a seguir]</i>	<i>[se detalla el resultado que se espera obtener de la configuración o acción realizada, es decir, como debería funcionar]</i>	<i>[se marca si el resultado fue igual al esperado]</i>	<i>[se marca si el resultado NO es igual al esperado]</i>	<i>[Generalmente se ocupa este campo para describir el error que se ha obtenido, o algún comentario válido]</i>
<b>Start Testing</b>					
2					
3					
<b>** End of Test Case **</b>					

Fig. 3: Formato Test Case

Kent Beck, autor de la metodología, recomienda la exhibición de los resultados que se obtienen al ejecutar las pruebas de aceptación (BECK, 2005), generando reportes y gráficas que desplieguen los porcentajes de efectividad obtenidos. Estos índices permiten evaluar si el equipo de desarrollo está realizando un buen trabajo o no. Para el presente sistema, se muestran los resultados obtenidos, como se puede apreciar en la Tabla III, y en las figuras: Fig. 4, Fig. 5, Fig. 6.

Tabla III: Resultado de Pruebas

Usuario	PARAMETROS				
	Adaptación al Sistema (%)	Gasto Papel en hojas		Tiempo ejecución en minutos	
		Antes del Sistema	Con el Sistema	Antes del Sistema	Con el sistema
<b>Módulo Usuarios</b>		<b>Por operación</b>		<b>Por actividad</b>	
Administrador	90	1-5	0	N/A	N/A
<b>Módulo Inventario</b>		<b>Por Orden de Trabajo</b>		<b>Por actividad</b>	
Bodeguero 1	80	20-30	1-10	30'-45'	5'-7'
Bodeguero 2	80			20'-30'	
Técnico 1	80	30-40	1-10	20'-30'	5'-8'
Técnico 2	75			15'-25'	
<b>Módulo G. Contable</b>		<b>Por Comprobante y Factura</b>		<b>Por actividad</b>	
Contador 1	75	10-13	3-5	9'-15'	3'-5'
Contador 2	90			6'-10'	
<b>Módulo Repositorio</b>		<b>Por Mes</b>		<b>Por actividad</b>	
Personal CTK	80	4	0	5'-10'	2'-3'
Cliente Externo	90	4	0	2'-5'	1'-2'

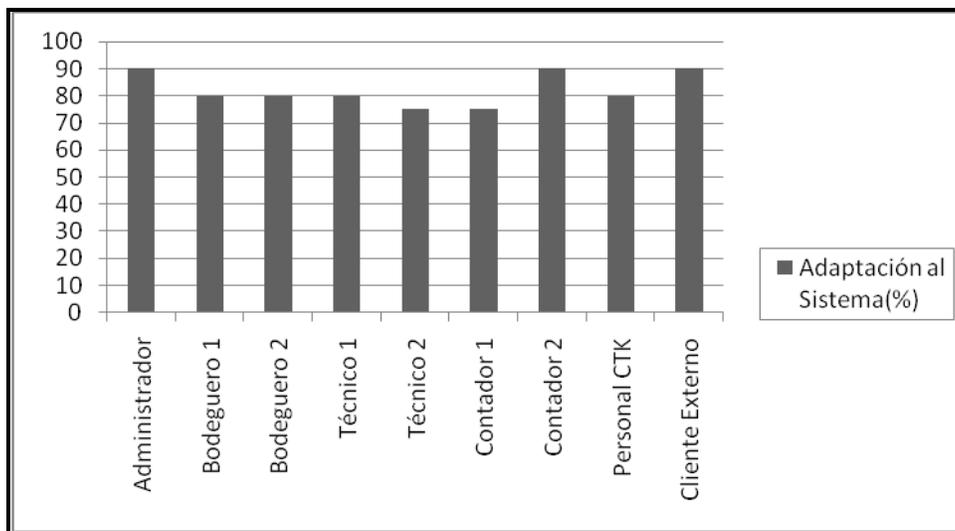
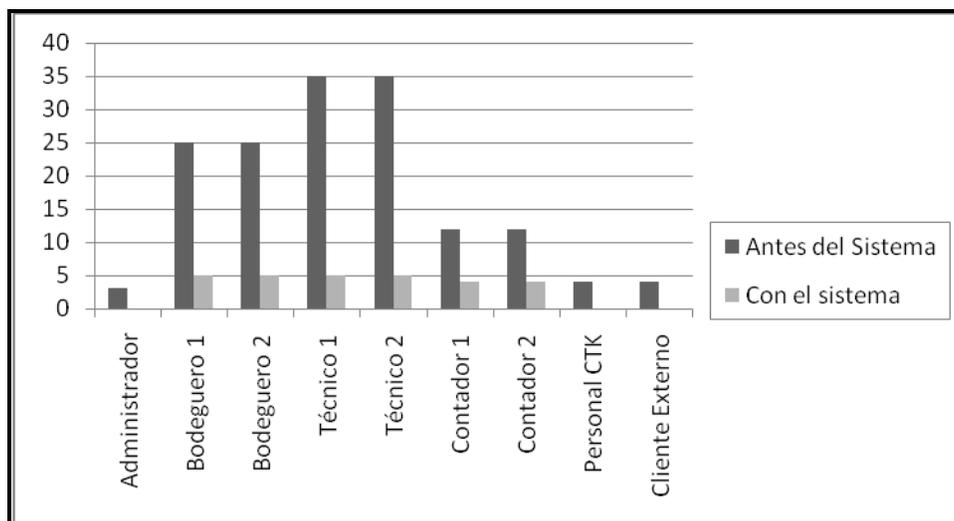
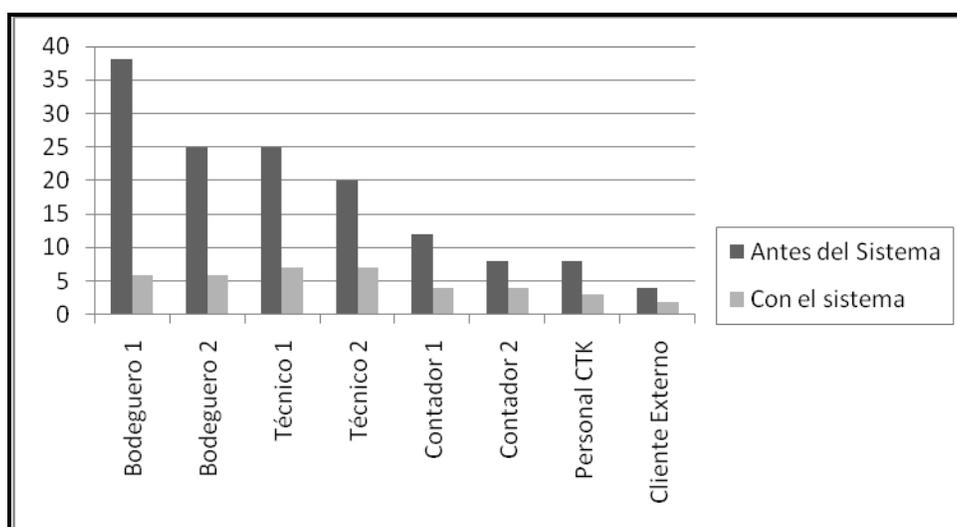


Fig. 4: Porcentaje de adaptación al sistema CTKSYS



**Fig. 5: Gasto de papel en hojas**



**Fig. 6: Tiempo ejecución de actividades en minutos**

De la Fig. 4, se puede determinar que el porcentaje de adaptabilidad del personal de la empresa al sistema, es bastante alto, por su amigable diseño, y fácil manejo.

Como se muestra en la Fig. 5, se evita la pérdida de tiempo en la generación, emisión, recepción y control de archivos, documentos o informes técnicos por medio de correo electrónico, o impresos.

En la Fig. 6, se puede observar que hay una clara disminución en el tiempo de ejecución de las actividades que antes realizaba manualmente el personal técnico y el contador de la empresa, principalmente en lo referente a la administración de Órdenes de Trabajo, Cotizaciones, Inventario y generación de comprobantes y facturas. Con el uso adecuado de la herramienta informática desarrollada se optimiza el manejo de un recurso importante como es el tiempo, permitiendo canalizarlo adecuadamente con el fin de lograr una mejor atención a sus clientes.

## 6. CONCLUSIONES Y TRABAJO FUTURO

- El sistema desarrollado para la empresa Crosstronik Cía. Ltda, utilizando herramientas de software libre permite que dicha empresa cuente con una herramienta informática a la medida, que automatiza los procesos internos de la empresa y que no genera costos por licencias.
- Programar y probar es más rápido que sólo programar. Se puede ganar tiempo de productividad sin hacer pruebas, pero perder mucho tiempo en la depuración. Es por esta razón que se debe probar si lo desarrollado es lo deseado, preguntando al cliente, que es quien necesita la información y quien mejor conoce los problemas de su negocio. La retroalimentación entre el desarrollador y el cliente representan puntos de vista interesantes para el desarrollo y funcionamiento correcto del sistema.
- El uso de metodologías ágiles ha generado un punto de inflexión donde cada día las empresas requieren software flexible y no rígido, que se adapte a los cambios en el menor tiempo posible y donde se evidencie avances gracias al concepto de prototipado.
- El diseño de una interfaz amigable y estandarizada, hacen que el sistema sea fácil de utilizar por cualquier usuario de la empresa. Con esto se evitó largas horas de capacitación, y que los errores cometidos por el usuario durante la realización de alguna tarea con el sistema sean mínimos.
- Al escoger una herramienta de Software Libre es necesario, evaluar el soporte técnico que avala dicha herramienta antes de utilizarla; dado que al escalar a futuro en funcionalidad puede resultar más costosa que una herramienta propietaria al no contar con el soporte técnico adecuado.
- El modelo de arquitectura utilizado para la construcción del sistema, es una infraestructura escalable con un bajo nivel de dependencia entre cada uno de sus componentes. Por tanto, como trabajo futuro se propone complementar cualquiera de sus módulos, siendo el módulo contable, el más extensible, ya que actualmente satisface únicamente la gestión de los procesos contables principales acoplado a las necesidades básicas de una PYME.

## 7. REFERENCIAS BIBLIOGRÁFICAS

- [1] Sommerville, I. *Ingeniería del Software*. Séptima edición. [PDF]. Madrid. 2006
- [2] Canós, J., Letelier, P. Y Penadés, M. *Metodologías Ágiles en el Desarrollo de Software*. [PDF]. Valencia, [www.willydev.net/descargas/prev/TodoAgil.Pdf](http://www.willydev.net/descargas/prev/TodoAgil.Pdf). Consulta: 30 marzo 2012.
- [3] Beck K. *Extreme Programming Explained: Embrace Change*. Second edition. [PDF]. Boston. 2005.
- [4] Richardson, C., Avondolio D., Scanlon J., Mitchell M., *Professional Java JDK*. [PDF]. Toronto. 2007
- [5] Daum, B. *Professional Eclipse for Java Developers*. [PDF]. Indiana, 2005
- [6] Marchioni, F. *Jboss AS Development*. [PDF]. Toronto 2009
- [7] The PostgreSQL Global Development Group, *PostgreSQL 8.4 Documentation*. [PDF]. Boston. 2010
- [8] Jaspersoft, *The JasperReports Ultimate Guide*. Third Edition. [PDF]. U.S.A. 2010
- [9] Malfará, D., Cukerman, D., Cócaro, F., Cassinelli J.P., Séttimo, R., "Testing en eXtreme Programming", *Disertación Gestión de Software*, Facultad de Ingeniería, Universidad de la República – Uruguay, Montevideo, 2006.