

**Universidad Tecnológica del Perú**

**Facultad de Ingeniería Industrial y de  
sistemas**

**Resumen**

**Diseño de un prototipo para el software de  
cajabancos**

**Título Profesional**

**Asesor :**                      **Ing. Luis Zuloaga Rotta**

**Tesista:**                      **Muñoz del Río, Pedro Rodrigo**

**Lima, 2008**

## **1. Introducción**

En la actualidad, la competitividad de las empresas pasa por la informatización de sus actividades. Esto ya ha sido realizado por las grandes empresas, las cuales invirtiendo recursos han automatizado sus procesos. Es así que las medianas y pequeñas empresas se han quedado atrás debido a la falta de recursos materiales y/o humanos para realizar proyectos de envergadura en el área de sistemas.

El proyecto consiste en desarrollar, un sistema que gestione la sección de cajabancos de una empresa en entorno web, el software será desarrollado usando herramientas software libre tanto en sistema operativo, lenguaje de programación, editores, etc.

Mediante este sistema las empresas, especialmente las pequeñas y medianas, podrán ordenar, automatizar y estandarizar el manejo del dinero en la

institución.

Es de suma importancia para la sociedad el mejorar la eficacia de las empresas nacionales, para así poder insertarnos de forma competitiva en el globalizado mundo moderno, que nos exige cada vez más adaptarnos de forma rápida e inteligente a los retos planteados.

## **2. Planteamiento del Problema**

En las empresas, el manejo del dinero es una parte fundamental de las mismas. La gestión de las cuentas bancarias, la calendarización de los pagos, el recibo de los cobros son actividades propias de la actividad de toda empresa.

Una empresa dispone de su dinero en cuentas bancarias y caja chica. Las cuentas bancarias son manejadas de acuerdo a criterios de moneda, intereses, pagos a realizar, etc. La caja chica se utiliza para los gastos menudos del día a día de las diversas unidades de la empresa, estos gastos son fiscalizados por un funcionario especialmente encargado llamado fiscal o algún otro que hace sus veces.

Una adecuada gestión de los fondos (ingresos y egresos), optimizando el manejo de estos, pagando la menor cantidad de intereses y obteniendo el mayor rendimiento posible de estos es la finalidad del área de cajabancos en las empresas.

Las pequeñas y medianas empresas no disponen por lo general de los recursos necesarios para encargar el análisis y desarrollo de software a medida. Es por esto que se desarrolla software vertical, el cual se procura hacer lo más genérico posible para adaptarse a las necesidades de las diferentes tipos de empresas.

Para hacer lo más adaptable posible el software a desarrollar, se utilizara como plataforma software de código abierto y funcionará sobre la intranet de la empresa.

El software de código abierto da la flexibilidad a la empresa para utilizarlo en todas las instalaciones que requiera, actualizar a la última versión, y alcanzar una mayor seguridad entre otras ventajas. El ser desarrollado como Intranet brinda flexibilidad al software al centralizar la administración de este y hacerlo multiplataforma para los clientes.

**Problema:**

Los procesos de cajabancos manuales no son fiables, insumen mayor número de recursos humanos, generan mayor tiempo para la toma de decisiones y no facilitan una buena gestión de los fondos y de la empresa en general, influyendo en el logro de una mayor competitividad y productividad.

### **3. Objetivos del Trabajo**

- Contribuir a la competitividad de la pequeña y mediana empresa.
- Desarrollar un software el cual maneje caja-bancos en entorno web.
- El software debe estar al alcance las empresas nacionales.
- Poner a disposición de los alumnos de la universidad una aplicación que podrán libremente adaptar y desarrollar de acuerdo a sus necesidades. Por lo tanto se declarará de libre disponibilidad el trabajo.

## 4. Alcances

Luego de ver la problemática y alternativas existentes, se decide desarrollar un software que automatice la gestión de cajabancos, no mediante un programa de escritorio sino mediante un desarrollo web, el cual funcione siguiendo el modelo cliente – servidor.

Este software tendría las ventajas inherentes a su condición.

- Multiplataforma, solo se necesita un navegador para poder utilizarlo.
- No requiere hardware especial en los clientes.
- Centralización de la información.
- Mayor facilidad de mantenimiento y migración.
- Parametrizar el software de tal manera que se adapte al tipo de empresa.

Se contemplará la seguridad, mediante el manejo de perfiles de usuarios y protocolos de encriptación de datos y contraseñas.

El software será de libre disponibilidad para las empresas, en especial la pequeña y mediana empresa, necesitadas de apoyo informático a la gestión diaria de sus operaciones.

El software será instalado en un servidor el cual debe disponer de el servidor web Apache instalado con el módulo python habilitado, para poder servir la aplicación. Este servidor además debe tener instalado el motor de base de datos Postgresql, y el framework Django funcionando sobre una plataforma Linux.

Todos los cliente deben tener un navegador web disponible.

Para el adecuado funcionamiento del servidor debe existir por lo menos 1GB de RAM y 2GB de capacidad de almacenamiento por cada cliente , incrementándose estas características con la cantidad de clientes.

La solución diseñada ayudará a las empresas, en especial las pequeñas y medianas a llevar un mejor control sobre sus operaciones financieras. Al tener a mano datos fiables sobre las compras y ventas realizadas por la empresa, los clientes que mayores ingresos proporcionan, rubros de mayor gasto, etc, se encontrarán en la capacidad de efectuar mejores decisiones sobre su futuro.

## 5. Marco Teórico

Para efectuar el análisis y desarrollo de la solución propuesta se utilizara la metodología estructurada propuesta por Edward Yourdon la cual consiste en lo siguiente:

### 4. Análisis de sistemas

**a. Diagrama de Flujo de Datos (DFD):** Es un modelo lógicográfico para representar el funcionamiento de un sistema en un proyecto software mediante flechas, rectángulos cerrados y rectángulos abiertos. Los rectángulos cerrados representan procesos, los rectángulos abiertos almacenes o archivos y la flechas un flujo de datos desde, o hacia, un proceso. Los flujos, entidades externas y los almacenes se etiquetan con un nombre. Los procesos se etiquetan con un número y un verbo en infinitivo con objeto directo. Un diagrama de flujo de datos (DFD) puede ser profundizado expandiendo algunos de sus procesos en subprocesos, en este caso la etiqueta tendrá un número adicional. No hay un límite para el número de procesos.

**b. Diagrama Entidad – Relación (ERD) lógico:** El modelo EntidadRelación es un concepto de modelado de Bases de Datos mediante el cual se visualiza los objetos que pertenecen a la Base de Datos como entidades las cuales tienen ciertos atributos y se vinculan mediante relaciones. El ERD lógico muestra los datos sin especificar su tamaño, tipo de dato y detalles referentes a la implementación física utilizada.

## 6. Conclusiones

- El área de cajabancos en las empresas es de vital importancia. Todo el manejo de las

cuentas, pagos, cheques de la empresa se centraliza en el área de cajabancos.

- El framework usado (Django) permite un desarrollo rápido y de calidad.
- El entendimiento previo de los procesos es fundamental para el diseño y desarrollo del software.
- Se debe parametrizar, ningún dato debe estar incluido en el código fuente.
- El diseño de la base de datos debe realizarse previamente al desarrollo de la aplicación, es el corazón de la misma.
- Se debe tener una idea clara de las reglas de negocio y los procesos antes de iniciar la implementación de un sistema.

## 7. Bibliografía

1. Meigs, Robert F. Contabilidad, la base para decisiones gerenciales.  
  
Undécima edición. Bogotá. Capítulo 1, Contabilidad: Información para la toma de decisiones, p. 5 – 17. Traducido por: Gladys Arango Medina.  
  
Revisado por: John Cardona Arteaga. Fecha publicación 2000.
2. Python Software Foundation. Beginner's Guide. Disponible de World Wide Web: <http://http://wiki.python.org/moin/BeginnersGuide>. Fecha publicación 2004. Fecha de actualización 2007.
3. Adrian Holovaty&Jacob KaplanMoss. Django Book. Capítulo 01: Introduction to Django. Disponible de World Wide Web: <http://www.djangobook.com/en/beta/chapter01/>. Fecha publicación 2006.
4. PostgreSQL Global Development Group. About. Disponible de World Wide Web: <http://www.postgresql.org/about/>. Fecha publicación 1996. Fecha actualización 2007.
5. Apache Foundation. Documentación del Servidor de HTTP Apache.

Disponible de World Wide Web: <http://httpd.apache.org/docs/2.2/>. Versión

2.2. Fecha publicación 2006.

6. PostgreSQL Global Development Group. Introduction. Disponible de World Wide Web: <http://www.pgadmin.org/>. Fecha publicación 2007.
7. Pressman, Roger S. Ingeniería de Software. 5ta Edición. Madrid. Capítulo 17, Técnicas de Prueba del Software, p. 285-294. Capítulo 11, Conceptos y Principios del Análisis, p. 182-186. Traducido por: Rafael Ojeda Martín. Revisado por: Luis Joyanes Aguilar. McGrawHill/Interamericana de España. S.A.U. Fecha publicación 2002.
8. Somerville, Ian. Ingeniería de Software. 6ta edición. México, Capítulo 5, Requerimientos del Software, p. 97 – 109, Traducido por: José Alejandro Domínguez Torres. Revisado por: Sergio Fuenlabrada Velázquez. Pearson Educación. Fecha publicación 2002.
9. Promonegocios. Concepto de Empresa. [en línea]. Disponible de World Wide Web: <http://www.promonegocios.net/empresa/conceptoempresa.html>. Fecha de publicación 2006.

10. Symphony Orchestra Institute. Open System Concepts. Disponible de World Wide Web: <http://www.soi.org/reading/change/concepts.shtml>. Fecha publicación 1999.
11. Extreme Programming.org. The Rules and Practices of Extreme Programming. Disponible de World Wide Web <http://www.extremeprogramming.org/rules.html>. Fecha publicación 2001
12. tweakers.net. Database test. Disponible de World Wide Web:<http://tweakers.net/reviews/657/6>. Fecha de publicación 2007.
13. Ruby on Rails. Ruby on Rails Documentation. Disponible de World Wide Web: <http://www.rubyonrails.org/docs>. Fecha de publicación 2008.
14. Symfony. Symfony Documentaton. Disponible de World Wide Web: [http://www.symfonyproject.org/doc/1\\_0/](http://www.symfonyproject.org/doc/1_0/). Fecha de publicación 2008.
15. php.net. Conceptos básicos. Disponible de World Wide Web: <http://www.php.net/manual/es/gettingstarted.php>. Fecha de publicación 2008.
16. Echenique García, Jose Antonio. Auditoría en Informática. Segunda edición.

México D.F. Capítulo 06: Evaluación de la Seguridad, p 195 – 198.

Revisado por Arturo Gonzáles Maya. McGrawHill/Interamericana. Fecha publicación 2001.

17. MySQL. Documentación MySQL. Disponible de World Wide Web:

<http://www.mysql.com/whymysql/>. Fecha de publicación 2008.

18. Extreme Programming.org. Extreme Programming: A gentle introduction.

Disponible de World Wide Web

<http://www.extremeprogramming.org/what.html>. Fecha publicación 2001.