



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

TEMA

ESTUDIO DE LA HERRAMIENTA DE DESARROLLO JASPERREPORTS

EN APLICACIONES WEB CON TECNOLOGÍA JSP

APLICATIVO

SISTEMA DE ADMINISTRACIÓN Y CONTROL DE RECURSOS TECNOLÓGICOS

PARA LA COOPERATIVA DE AHORRO Y CRÉDITO ATUNTAQUI LTDA.

Informe Técnico

ESPAÑOL

AUTOR

WILSON ANÍBAL CÁRDENAS HERNÁNDEZ

DIRECTOR

ING. MARCELO JURADO

“ESTUDIO DE LA HERRAMIENTA DE DESARROLLO JASPERREPORTS EN APLICACIONES WEB CON TECNOLOGÍA JSP”

(Noviembre 2011)

Ibarra - Ecuador

I. LA TECNOLOGÍA JAVA

Java es un lenguaje de programación orientado a objetos, desarrollado por *Sun Microsystems* a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores.

Las aplicaciones Java están típicamente compiladas en un *bytecode*, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el *bytecode* es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del *bytecode* por un procesador Java también es posible.

Entre diciembre de 2006 y mayo de 2007, *Sun Microsystems* liberó la mayor parte de sus tecnologías Java bajo la licencia GNU GPL, de acuerdo con las especificaciones del *Java Community Process*, de tal forma que prácticamente todo el Java es ahora software libre.

El 20 de Abril del 2009, *Oracle Corporation* anuncia la compra de *Sun Microsystems*, por la suma de \$7,400 millones de dólares; tras la adquisición de Sun, Oracle se convierte en guardián absoluto de algunas de las joyas más importantes del código abierto, incluyendo la popular base de datos MySQL, el sistema operativo Solaris y la plataforma de desarrollo Java.

1.1. Características

Las características principales que nos ofrece la tecnología Java, respecto a cualquier otro lenguaje de programación son:

- Simple
- Orientado a objetos
- Distribuido
- Robusto
- Arquitectura neutral
- Seguro
- Portable
- Interpretado
- Multihilo
- Dinámico

1.2. Plataforma Java

La plataforma Java es el entorno de computación originaria de *Sun Microsystems*, capaz de ejecutar aplicaciones desarrolladas usando el lenguaje de programación Java u otros lenguajes que compilen a *bytecode* y un conjunto de herramientas de desarrollo.

En este caso, la plataforma no es un hardware específico o un sistema operativo, sino más bien una máquina virtual encargada de la ejecución de las aplicaciones, y un conjunto de bibliotecas estándar que ofrecen una funcionalidad común.

Máquina Virtual de Java

La Máquina Virtual Java es un programa nativo; es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial, el cual es generado por el compilador del lenguaje Java. La Máquina Virtual de Java es una de las piezas fundamentales de la plataforma Java. Básicamente se sitúa en un nivel superior al hardware del sistema sobre el que se pretende ejecutar la aplicación, y este actúa como un puente que entiende tanto el *bytecode*, como el sistema sobre el que se pretende ejecutar.

API de Java

El conjunto de bibliotecas del lenguaje es conocido como la *Application Programming Interface de Java* y es un conjunto de componentes que proporcionan diferentes herramientas para el desarrollo. La API de Java está formada por un conjunto de paquetes de clases que le proporcionan una gran funcionalidad. El núcleo de la API viene con cada una de las implementaciones de la Máquina Virtual de Java.

Lenguaje Java

Java es un lenguaje de programación orientado a objetos puro, en el sentido de que no hay ninguna variable, función o constante que no esté dentro de una clase. Se accede a los miembros dato y las funciones miembro a través de los objetos y de las clases.

1.3. Ediciones de la plataforma Java

Existen tres distribuciones básicas de la plataforma Java en un intento por cubrir distintos entornos de aplicación.

- Java Platform, Standard Edition o Java SE (antes J2SE).
- Java Platform, Enterprise Edition o Java EE (antes J2EE).
- Java Platform, Micro Edition o Java ME (antes J2ME).

II. JAVA SERVER PAGES

Java Server Pages es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

Esta tecnología es un desarrollo de la compañía *Sun Microsystems*. La Especificación JSP 1.2 fue la primera que se liberó y en la actualidad está disponible la Especificación JSP 2.1.

Las JSP's permiten la utilización de código Java mediante scripts. Además, es posible utilizar algunas acciones JSP predefinidas mediante etiquetas. Estas etiquetas pueden ser enriquecidas mediante la utilización de Bibliotecas de Etiquetas (TagLibs o Tag Libraries) externas e incluso personalizadas.

2.1. Funcionamiento de JSP

Las JSP's son en realidad servlets, un JSP se compila a un programa en Java la primera vez que se invoca y del programa en Java se crea una clase que se empieza a ejecutar en el servidor como un servlet. La principal diferencia entre los servlets y las JSP's es el enfoque de la programación. Un JSP es una página web con etiquetas especiales y código Java incrustado, mientras que un servlet es un programa que recibe peticiones y genera a partir de ellas una página web.

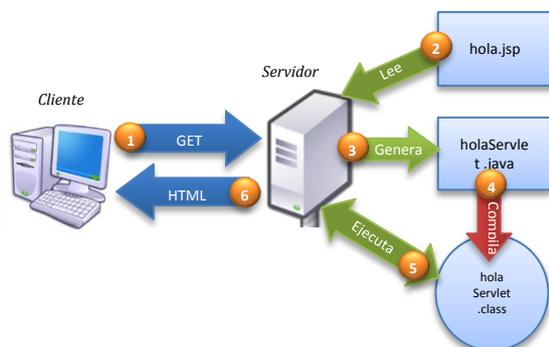


Figura 1.1. Funcionamiento de JSP.

2.2. Componentes JSP

Los componentes que podemos utilizar en una página JSP para hacerla dinámica son:

- Expresiones
- Declaraciones
- Scriptlets
- Directivas
- Etiquetas
- Comentarios

2.3. Objetos implícitos

En JSP disponemos de algunos objetos implícitos, que nos permitirán acceder a diferente información y realizar diversas acciones. En JSP tenemos los siguientes objetos implícitos: *request*, *response*, *out*, *session*, *application*, *config*, *pageContext* y *page*.

III. XML (eXtensible Markup Language)

XML (*eXtensible Markup Language*), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos. Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Algunos de estos lenguajes que usan XML para su definición son XHTML, SVG, MathML, Android.

3.1. Características del estándar XML

El estándar XML se define como "el formato universal para documentos y datos estructurados en Internet" y sus principales características de funcionamiento tal y como la propia W3C lo describe, son:

- XML es un estándar para escribir datos estructurados en un fichero de texto.
- XML parece HTML pero no lo es.
- XML está en formato texto, pero no para ser leído.
- XML consta de una familia de tecnologías.
- XML es prolijo, pero eso no supone un problema.
- XML no requiere licencias.
- Modelo de Objetos de Documento (DOM).

3.2. Ventajas del estándar XML

Las principales ventajas del XML, son las siguientes:

- Es extensible, después de diseñado y puesto en producción, es posible extender XML con la adición de nuevas etiquetas.
- El analizador es un componente estándar, no es necesario crear un analizador específico para cada versión de lenguaje XML.
- Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarla.
- Transformamos datos en información, pues se le añade un significado concreto y los asociamos a un contexto.

3.3. Partes de un documento XML

Un documento XML está formado por el prólogo y por el cuerpo del documento.

Prólogo

Aunque no es obligatorio, los documentos XML pueden empezar con unas líneas que describen la versión XML, el tipo de documento y otras cosas. El prólogo de un documento XML contiene:

- Declaración XML
- Declaración de tipo de documento

Cuerpo

A diferencia del prólogo, el cuerpo no es opcional en un documento XML, el cuerpo debe contener solo un elemento raíz, característica indispensable también para que el documento esté bien formado. El cuerpo puede contener otros elementos como:

- Elementos
- Atributos

IV. JASPERREPORTS

JasperReports es una poderosa herramienta para la generación de reportes en Java, con la habilidad de producir contenido amplio y completo para la pantalla, impresora o inclusive exportarlo a diferentes formatos de archivos como: PDF, XLS, CSV, HTML, XML, entre otros.

La librería está completamente escrita en Java y se puede utilizar en una variedad de aplicaciones Java, incluyendo JEE o aplicaciones web. Su principal objetivo es ayudar a generar de forma sencilla y flexible reportes listos para imprimir.

JasperReports organiza la información de acuerdo a la plantilla de diseño definida en un archivo XML. Dicha información puede provenir de diversas fuentes de datos como: bases de datos relacionales, colecciones o arrays de objetos Java, etc.

Con el fin de llenar un reporte con datos, el diseño en XML debe ser compilado. A través de la compilación, un objeto es generado y luego es serializado con el fin de almacenarlo en un disco o enviarlo a través de la red. Este objeto serializado se utiliza cuando una aplicación necesita llenar el reporte con datos. El objeto generado está listo para ser llenado con los diferentes conjuntos de datos y generar documentos.

Para llenar un diseño, el motor de JasperReports necesita recibir los datos para el reporte. Estos datos pueden provenir de diversas fuentes, algunos se pueden pasar como parámetros del reporte, pero la mayoría de los datos se encuentran en las fuentes de datos definidas en el diseño.

El resultado de esta operación, es un nuevo objeto que representa el documento listo. Este documento se puede visualizar directamente en el visor incorporado de JasperReports, o se puede exportar a otros formatos más populares como: PDF, HTML o XML.

4.1. API de JasperReports

A continuación, se describen las clases e interfaces más importantes que integran esta librería.

- Clase *dori.jasper.engine.design.JasperDesign*

- Clase *dori.jasper.engine.JasperReport*
- Clase *dori.jasper.engine.JasperCompileManager*
- Clase *dori.jasper.engine.JasperPrint*
- Clase *dori.jasper.engine.JRResultSetDataSource*
- Clase *dori.jasper.engine.data.JRTableModelDataSource*
- Clase *dori.jasper.engine.JREmptyDataSource*
- Clase *dori.jasper.engine.JasperFillManager*
- Clase *dori.jasper.engine.JRAbstractScriptlet*
- Clase *dori.jasper.engine.JRDefaultScriptlet*
- Clase *dori.jasper.engine.JasperPrintManager*
- Clase *dori.jasper.engine.JasperExportManager*
- Clase *dori.jasper.engine.JasperRunManager*
- Clase *dori.jasper.view.JRViewer*
- Clase *dori.jasper.view.JasperViewer*
- Clase *dori.jasper.view.JasperDesignViewer*
- Clase *dori.jasper.engine.util.JRLoader*
- Interfaz *dori.jasper.engine.JRDataSource*

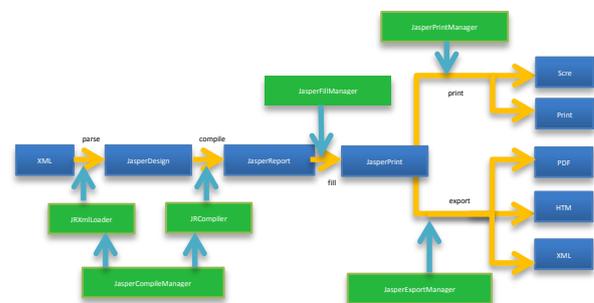


Figura 4.1. Clases e interfaces principales de la librería.

4.2. Tareas y procesos

Aquí se describe lo que hay que saber para poder analizar los diseños en XML; compilarlos, llenarlos, visualizarlos, imprimirlos o exportarlos a otros formatos.

Análisis XML

JasperReports utiliza la API SAX 2.0 para analizar los archivos XML. Sin embargo, no está ligado a una particular implementación de SAX 2.0, sino que es capaz de decidir en tiempo de ejecución que analizador XML utilizará.

Para instanciar una clase de análisis, JasperReports utiliza el método *createXMLReader()* de la clase *org.xml.sax.helpers.XMLReaderFactory*. En este caso, será necesario establecer en tiempo de ejecución la propiedad del sistema Java *org.xml.sax.driver*, el nombre completo de la clase del controlador SAX, como se especifica en la documentación SAX 2.0.

Compilación del diseño

Con el fin de generar un reporte, uno tiene que crear el diseño en primer lugar mediante la edición de un archivo XML, este contenido tiene que ser analizado y cargado en

un objeto *dori.jasper.engine.design.JasperDesign*, para pasar el proceso de compilación antes de estar listo para ser presentado con los datos por el motor de reportes.

El objetivo principal del proceso de compilación es producir y cargar los *bytecode* de la clase que contiene todas las expresiones del reporte. Esta clase creada dinámicamente se utilizará en el llenado del reporte para evaluar todas las expresiones.

Vista previa del diseño

La librería JasperReports no se suministra con una avanzada herramienta GUI para ayudar en la tarea de diseño de reportes. Sin embargo, en la actualidad existen algunas herramientas de desarrollo que conjuntamente con esta librería proporcionan una interfaz simple y amigable para el diseño de reportes. La biblioteca contiene un componente visual muy útil que permite a los desarrolladores realizar una vista previa de los diseños de reportes y ver cómo van quedando los reportes.

Llenado del Reporte

El proceso de llenado del reporte es el más importante de toda la funcionalidad de la librería JasperReports. Este representa el objetivo principal de este componente de software, ya que es el proceso que manipula los conjuntos de datos con el fin de producir documentos de alta calidad. Hay tres cosas que deben ser proporcionadas al proceso de llenado del informe:

- El diseño
- Los parámetros
- La fuente de datos

El resultado es siempre un documento listo para visualizar, imprimir o exportar a otros formatos.

Visualización de reportes

Para ver los reportes generados en el formato propietario o el formato XML propietario producido por el exportador XML interno, JasperReports tiene un visualizador incorporado. Este es un componente basado en swing que puede integrarse fácilmente con otras aplicaciones Java que deseen incorporar esta funcionalidad, sin la utilidad de exportar los documentos en los formatos más populares, para que ellos puedan visualizarse. La clase de *dori.jasper.view.JRViewer* representa este componente visual. Esta puede personalizarse para responder a las necesidades de aplicaciones en particular, de esta manera podríamos agregar o quitar los botones de la barra de herramientas existente que se despliega, o realizar otras modificaciones.

Impresión de reportes

El objetivo principal de la librería JasperReports y el de todas las herramientas destinadas a la generación de reportes, es crear documentos listos para imprimir. La mayoría de reportes que se generan terminan o se supone que terminarían en el papel.

Podemos imprimir los documentos generados por la librería JasperReports utilizando la clase *dori.jasper.engine.JasperPrintManager*.

Exportación de reportes

En algunos ambientes de aplicación, es conveniente transformar el documento JasperReports generado a otros formatos más populares como PDF, HTML o XLS. Haciendo esto, se puede hacer que otras personas puedan ver estos reportes sin necesidad de instalar visualizadores especiales en sus sistemas, especialmente cuando enviamos los documentos a través de la red.

Existe también una clase fachada en JasperReports para este tipo de funcionalidad. Su nombre es *dori.jasper.engine.JasperExportmanager* y puede usarse para obtener contenido PDF, HTML o XLS para los documentos producidos por el proceso de llenado de reportes.

4.3. Diseño del reporte

El diseño del reporte representa una plantilla que será usada por el generador de reportes de JasperReports para entregar un documento listo para la pantalla o impresora.

Generalmente, un diseño contiene toda la información concerniente a la estructura y el aspecto de los documentos que se generarán cuando los datos se proporcionen. Esta información involucra la posición y el contenido de varios elementos de texto o de gráficos que se mostrarán en el documento generado, su apariencia, los cálculos personalizados, datos que se agrupan y la manipulación de los datos que debe realizar al generar los documentos.

Normalmente, los diseños se definen en archivos XML con una estructura especial y son sujetos al proceso de compilación de JasperReports antes de ser llenados con los datos. Pero ellos también pueden construirse en memoria, usando el API de JasperReports.

4.4. Datos del reporte

Durante el proceso de llenado del reporte existen tres cosas que se deben proporcionar: el diseño, los valores de los parámetros y la fuente de datos. Los parámetros y la fuente de datos representan la única fuente de datos que el generador de reportes puede utilizar para el llenado del

informe. Esta información se organizará de acuerdo a la plantilla definida en el diseño y producirá un documento listo para imprimir.

4.5. Secciones del reporte

Las secciones son fragmentos de la plantilla del informe que tienen una determinada altura y ancho y pueden contener elementos del reporte como líneas, rectángulos, imágenes o campos de texto. Las secciones son llenadas en repetidas ocasiones durante el tiempo de generación del reporte y hacer el documento final que se está produciendo. Al declarar el contenido y características de una sección del reporte, en un diseño XML, se debe utilizar el elemento genérico `<band>`.

Toda la estructura del diseño se basa en las siguientes secciones: `<title>`, `<pageHeader>`, `<columnHeader>`, `<groupHeader>`, `<detail>`, `<groupFooter>`, `<columnFooter>` y `<pageFooter>`.

Title

Es la primera sección del reporte y esta se genera sólo una vez durante el proceso de llenado del reporte y va al inicio del documento final.

Siendo la primera sección del reporte significa que va a preceder incluso la sección de encabezado de página.

Page Header

Esta sección aparece en la parte superior de cada página del documento final.

Column Header

Esta sección aparece en la parte superior de cada columna del documento final.

Detail

Para cada registro existente en el origen de datos, el motor generador de reportes trata de generar esta sección.

Column Footer

Esta sección aparece en la parte inferior de cada columna del documento final. Nunca se extiende hacia abajo para visualizar el contenido de los elementos de texto que contiene y será siempre de una altura fija.

Page Footer

Esta sección aparece en la parte inferior de cada página del documento final. Al igual que la sección de la columna de pie de página, nunca el pie de página se extiende hacia abajo para mostrar el contenido de los elementos que contiene y será siempre de una altura fija.

Summary

Esta sección se genera sólo una vez en el reporte y aparece al final del documento final, pero no es necesariamente la última sección que se genera. Eso es porque en algunos casos, el pie de la columna y/o pie de página de la última página puede seguir.

4.6. Elementos del reporte

Los reportes generados estarían vacíos si no se ponen algunos elementos en el diseño. Los elementos del reporte pueden mostrar objetos como: textos estáticos, campos de texto, imágenes, líneas o rectángulos, que se ponen en las secciones de diseño del reporte para que aparezcan en el documento generado.

Elementos de texto

Son los elementos del diseño del reporte que visualizan texto estático y texto generado dinámicamente.

Elementos gráficos

Corresponden a los objetos gráficos insertados en el diseño del reporte como: líneas, rectángulos e imágenes.

Cuando se agrega un elemento del reporte en una de las secciones del reporte, se tiene que especificar la posición relativa del elemento en la sección en particular y su tamaño, junto con otras propiedades generales de los elementos.

4.7. Subreportes

Los subreportes son una característica importante para una herramienta de generación de reportes. Estos permiten y facilitan la creación de reportes más complejos y simplifican el trabajo de diseño. Los subreportes son muy útiles cuando se crea un reporte tipo maestro-detalle o cuando la estructura de un reporte único no es suficiente para describir la complejidad del documento final.

Un subreporte es, de hecho, un reporte normal que se ha incorporado como parte de otro reporte. Se puede montar los subreportes y hacer que un subreporte contenga en sí otro subreporte, el nivel de anidamiento no está limitado.

Por otro lado, un subreporte es también un tipo especial de elemento del reporte que ayuda a introducir un subreporte en el reporte maestro. No hay nada más que decir acerca de los subreportes, pues son vistos como reportes normales, debido a que se compilan y se llena al igual que los reportes normales. De hecho, cualquier diseño de reportes se puede utilizar como un subreporte cuando se lo incorpore en otro diseño de reportes, sin necesidad de cambiar nada en su interior.

V. BASES DE DATOS

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital, que ofrece un amplio rango de soluciones al problema de almacenar datos.

Existen programas denominados Sistemas de Gestión de Bases de Datos o abreviado SGBD, que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos SGBD, así como su utilización y administración, se estudian dentro del ámbito de la informática.

5.1. Sistemas Gestores de Bases de Datos (SGBD)

Los Sistemas de Gestión de Bases de Datos (*Database Management System, DBMS*) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

El propósito general de los sistemas de gestión de bases de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización.

5.2. Objetivos de los SGBD

Existen distintos objetivos que deben cumplir los SGBD:

- Realizar consultas no predefinidas y complejas.
- Flexibilidad e independencia.
- Eliminar la redundancia.
- Integridad de los datos.
- Concurrencia de usuarios.
- Seguridad.
- Abstracción de la información.
- Tiempo de respuesta.

5.3. El Lenguaje SQL

El lenguaje de consulta estructurado o SQL (*Structured Query Language*) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo efectuar consultas con el fin de recuperar, de una forma sencilla, información de interés de una base de datos, así como también hacer cambios sobre ella.

5.4. Administración de SGBD

Los administradores de bases de datos son los responsables del correcto funcionamiento de los Sistemas

de Gestión de Bases de Datos y de garantizar la disponibilidad de los datos. Intervienen en situaciones problemáticas o de emergencia, pero su responsabilidad fundamental es velar para que no se produzcan incidentes. A continuación, enumeramos una lista de tareas típicas del administrador de bases de datos.

- Mantenimiento, administración y control de los esquemas.
- Asegurar la máxima disponibilidad de los datos.
- Resolución de emergencias.
- Garantizar la integridad y la calidad de los datos.
- Diseño físico, estrategia de acceso y reestructuraciones.
- Control del rendimiento y decisiones relativas a las modificaciones.
- Asesoramiento a los usuarios finales sobre la utilización de la base de datos.
- Control y administración de la seguridad.

5.5. Adaptive Server Enterprise

Adaptive Server Enterprise (*ASE*) es el Sistema Gestor de Bases de Datos (*SGBD*) insignia de la compañía Sybase. ASE es un sistema de gestión de bases de datos, altamente escalable, de alto rendimiento, con soporte a grandes volúmenes de datos, transacciones y usuarios, y de bajo costo, que permite:

- Almacenar datos de manera segura.
- Procesar datos de manera inteligente.
- Movilizar datos.

VI. DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

La Cooperativa de Ahorro y Crédito "Atuntaqui" Ltda., es una institución financiera ubicada en la provincia de Imbabura y al norte de Pichincha, ofreciendo a la comunidad una variedad de productos y servicios. Con la finalidad de brindar una adecuada atención a sus socios y clientes, el Departamento Informático de la Cooperativa Atuntaqui Ltda., constantemente actualiza su tecnología, provocando una variedad de cambios y modificaciones en los recursos tecnológicos.

Durante mi permanencia laboral y profesional en la Cooperativa de Ahorro y Crédito "Atuntaqui" Ltda., he podido constatar que en la institución existe una variedad y cantidad considerable de recursos tecnológicos que facilitan el desempeño de las actividades cotidianas de sus empleados; cuya administración y control de forma manual resulta tediosa y compleja, razón por la cual el departamento informático no cuenta con la información clara, precisa y oportuna, complicando de esta manera la administración de los recursos.

Es por esta razón que, en colaboración con el área informática de la Cooperativa de Ahorro y Crédito "Atuntaqui" Ltda., se ha decidido plantear el presente proyecto con la implementación de un sistema que optimice el proceso de administración y control de los recursos tecnológicos.

6.1. Descripción del sistema

Después de haber realizado un estudio de los procesos involucrados en la administración y control de los recursos tecnológicos de la Cooperativa de Ahorro y Crédito Atuntaqui Ltda., se ha visto conveniente clasificarlos, en cinco módulos principales. Mismo que fueron optimizados para una mejor administración de tecnología.



Figura 6.1. Esquema general del sistema (SARTE).

Administración

El sistema está basado en su mayoría en parámetros establecidos por parte del administrador del sistema; mismos que pueden ser configurables a través del sistema de información.

Recursos

Una vez realizada la configuración de los parámetros, esenciales para el correcto funcionamiento del sistema, estos nos permiten ingresar, modificar y organizar la información correspondiente a los recursos tecnológicos, de acuerdo a la categorización establecida por el administrador.

Mantenimientos

Uno de las necesidades principales y motivos por el cual se planteó la implementación del sistema de administración de recursos tecnológicos, fue el poder registrar y llevar un control de todos los mantenimientos y novedades suscitadas durante el tiempo de vida útil de los recursos.

Traslados

Durante el estudio del proceso de control de los recursos y el proceso de recopilación de la información, pudimos determinar que existe carencia de información en lo que respecta al traslado y custodios de equipos, dificultando

el proceso de administración. Es por esto que se planteó la implementación de un módulo de traslado de equipos que nos ayuden con esta actividad.

Reportes

La razón principal por la que se planteó la implementación de este sistema en la cooperativa es el poder disponer de información clara, precisa y oportuna de todas las novedades presentadas con los recursos tecnológicos, con el fin de conocer la realidad institucional en lo que respecta a la tecnología y poder tomar con prontitud las decisiones más acertadas dentro de la administración de estos recursos.

VII.CONCLUSIONES

- La implementación de un sistema informático en la Cooperativa de Ahorro y Crédito "Atuntaqui" Ltda., simplificó notablemente el proceso de administración y control de los recursos tecnológicos con la automatización de actividades repetitivas y registro de manera organizada de todas las novedades presentadas. Además, con la puesta en marcha del sistema en la Cooperativa de Ahorro y Crédito Atuntaqui Ltda., actualmente se dispone de información precisa, clara y oportuna de los recursos tecnológicos existentes y de sus respectivas modificaciones y adaptaciones.
- El uso de una herramienta informática para el diseño y generación de reportes en los sistemas de información web, como JasperReports; disminuye el costo del proyecto ya que permite un ahorro de grandes cantidades de dinero destinado a la adquisición de licencias, a la vez que facilita el diseño de los reportes que visualizan la información de los sistemas disminuyendo el tiempo de desarrollo del proyecto.
- JasperReports es una herramienta de código abierto muy poderosa para producir documentos listos para imprimir, visualizar o exportar a otros formatos como PDF, XLS, HTML, entre otros, que puede ser incorporada a cualquier sistema de información desarrollado en la plataforma Java.
- El desarrollo de aplicaciones utilizando herramientas Open Source, permite que otros programadores puedan modificar el código de los sistemas, permitiendo adaptarlo a sus necesidades.

