

**DESARROLLO, AJUSTE E IMPLEMENTACIÓN DE UN MODULO DE
SOFTWARE PARA EL MANEJO DE REPORTES UTILIZANDO JAVA
PARA LA EMPRESA MAYASOFT INGENIERÍA LTDA.**

DANILO JOSÉ DÍAZ GÓMEZ

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2011

**DESARROLLO, AJUSTE E IMPLEMENTACIÓN DE UN MODULO DE
SOFTWARE PARA EL MANEJO DE REPORTES UTILIZANDO JAVA
PARA LA EMPRESA MAYASOFT INGENIERÍA LTDA.**

DANILO JOSÉ DÍAZ GÓMEZ

Trabajo de grado para optar al título de Ingeniero de Sistemas

Director

Ing. Emiro Muñoz Jerez

Tutor

Ing. Jesús David Rueda Polo

Mayasoft Ingeniería Ltda.

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2011

AGRADECIMIENTOS

A Dios gracias por darme fuerza, paciencia y sabiduría necesaria para aprovechar cada oportunidad y perseverar ante cada dificultad.

Al Ingeniero Jesús Rueda por depositar su confianza y servir de guía en el proceso de aprendizaje, su apoyo y consejos fueron indispensables para alcanzar la culminación de éste proyecto.

A mis compañeros de la empresa Mayasoft que nos mostraron el inicio del camino y fueron un libro abierto permitiéndome tomar las bases necesarias para emprender esta meta. Gracias a su constante apoyo fue posible superar cada una de las dificultades que surgieron en el trascurso del desarrollo de este proyecto.

A mis amigos y compañeros que siempre confiaron en mí.

A aquellas personas que directa e indirectamente aportaron su valioso granito de arena.

GLOSARIO

ARQUITECTURA CLIENTE-SERVIDOR: Es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en procesos independientes que cooperan entre si para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos, y servidor al proceso que responde a las solicitudes.

INTERNET: Red global de comunicaciones que interconecta computadores y bases de datos distribuidas por todo el planeta.

IP(Internet Protocol, Protocolo de Internet): En un conjunto de normas que provee las funciones basicas de direccionamiento en Internet y en cualquier red TCP/IP (Transfer Contro Protocol/Internet Protocol, Protocolo de Contros de Transferencia/Protocolo de Internet). Este protocolo se encarga de poner una etiqueta con la direccion adecuada a cada paquete, ya que cada computador conectado a la red tiene una direccion de Internet única que lo distingue de cualquier otra computadora en el mundo.

JAVA: Es un lenguaje de programación orientado a objetos desarrollado por SUN Microsystems. Está diseñado para usarse en entorno distribuido de Internet.

JDBC: (Java DataBase Connectivity,Conectividad de Bases de Datos de Java): Es una especificación de la interfaz de programa de Aplicación (API), para conectar los programas escritos en Java a los datos en la base de datos.

ESCALABILIDAD: Es la posibilidad de aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).

LINUX: Sistema Operativo. ES una implementación de libre distribución UNIX para computadoras personales, servidores y estaciones de trabajo. Consta de componentes GNU y el kernel (núcleo) desarrollado por Linus Torvalds.

MINT WEBASSISTANT: Aplicativo para el manejo de recursos y solicitudes que desarrolla la empresa Mayasoft y sobre el cual se encuentra implementado el modulo de reportes en JasperReports.

PLANTILLA: Es es el esqueleto de un reporte y se diseña usando IReport. En esta se colocan los campos y se configura el comportamiento de un reporte. Estan en formato .jrxml.

PORTABLE: La portabilidad de un software se define como su grado de dependencia de la plataforma en la que se ejecuta. La portabilidad es mayor cuanto menor es su dependencia del software de plataforma.

SCRIPT: Es una aplicación informática escrito en un lenguaje específico de programación que tiene un conjunto de instrucciones y normalmente funciona sobre otras aplicaciones que ya están en funcionamiento.

UML (Unified Modeling Language, Lenguaje de Modelamiento Unificado): Es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. UML entrega una forma de modelar cosas conceptuales como los procesos de negocio u funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de bases de datos y componentes de software reusable.

VPN: Red privada virtual.

Tabla de contenido

INTRODUCCIÓN	13
PRESENTACIÓN DEL PROYECTO.....	15
1.1 ORIENTACIÓN SOBRE EL CONTENIDO DEL INFORME.....	15
1.1.1 DESCRIPCIÓN DE LA EMPRESA.....	16
1.2 OBJETIVOS	18
1.3 JUSTIFICACION	19
1.4 ALCANCES Y LIMITACIONES	20
2. MARCO TEÓRICO	21
2.1 ARQUITECTURA CLIENTE-SERVIDOR	21
2.2 TECNOLOGIA USADA: J2SE – JAVA STANDARD EDITION	26
2.3 BASES DE DATOS	31
2.4 PROGRAMACIÓN UTILIZADA	36
3. MARCO METODOLÓGICO	38
3.1 PROTOTIPADO EVOLUTIVO.....	39
3.2 LENGUAJE DE MODELADO UNIFICADO	41
3.3 ESTÁNDARES DE PROGRAMACIÓN	45
4. DESARROLLO DEL MODULO DE REPORTES USANDO JASPERREPORTS47	
4.1 Arquitectura	47
4.2 Requerimientos funcionales	47
4.3 Diseño de plantillas y conexion de IReport con el aplicativo	51
4.4 Documentación técnica	57
Figura 20. Workflow de la ejecución de un reporte	60
5. MANUAL DE USUARIO.....	64
Importar plantilla al MINT WEBASSISTANT	64
Crear nodo reporte	65
Llenado de datos.....	65
Búsqueda sistema archivos.....	66
Menú contextual	67
Opción vista preliminar.....	68
Eliminar un reporte del árbol de archivos.	72
Editar información del reporte	73
Guardar resultado del reporte en un formato determinado	74
6. PRUEBAS REALIZADAS	76
6.1 PRUEBAS DE VERIFICACIÓN.....	76
7.CONCLUSIONES.....	80
8. BIBLIOGRAFIA.....	82

LISTA DE FIGURAS

Figura 1. Modelo Cliente-Servidor.....	22
Figura 2. Esquema de Arquitectura Cliente-Servidor de Dos Capas.....	24
Figura 3. Esquema de Arquitectura Cliente-Servidor de Tres Capas.....	24
Figura 4. Función del DBMS.....	35
Figura 5. Prototipado Evolutivo.....	39
Figura 6. Diagramas de Casos de Uso.....	44
Figura 7. Diagrama de Secuencias.....	46
Figura 8. Importar plantilla.....	47
Figura 9. Importar plantilla sobre nodo existente.....	48
Figura 10. Exportar plantilla.....	49
Figura 11. Generar vista previa.....	50
Figura 12. Carpeta bin de IReport.....	51
Figura 13. Configuración Classpath.....	52
Figura 14. Abrir dialogo de fuentes de datos.....	53
Figura 15. Lista tipos fuentes de datos.....	53
Figura 16. Seteo de ruta de clase.....	54
Figura 17. Agregar campos.....	55
Figura 18. Listar campos.....	56
Figura 19. Crear plantilla.....	57
Figura 20. Ejecución de un reporte.....	60
Figura 21. Arquitectura de reportes en MWA.....	63
Figura 22. Crear nodo reporte.....	64
Figura 23. Llenado de datos.....	65
Figura 24. Búsqueda sistema archivos.....	66
Figura 25. Menú contextual.....	67
Figura 26. Opción vista preliminar.....	68

Figura 27. Dialogo de filtros 1 y 2.....	69
Figura 28. Resultado reporte.....	70
Figura 29. Eliminar plantilla	71
Figura 30. Dialogo confirmación.....	71
Figura 31. Editar reporte.....	72
Figura 32. Dialogo de propiedades del reporte.....	73
Figura 33. Guardar resultado reporte	74
Figura 34. Escoger formato	75

RESUMEN

TÍTULO: DESARROLLO, AJUSTE E IMPLEMENTACIÓN DE UN MODULO DE SOFTWARE PARA EL MANEJO DE REPORTES UTILIZANDO JAVA PARA LA EMPRESA MAYASOFT INGENIERÍA LTDA

AUTORES: DANILO JOSÉ DÍAZ GÓMEZ

PALABRAS CLAVE: JasperReports, Virtual Staff, Mayasoft, Módulo, Mint Media-Interactive.

DESCRIPCIÓN

Actualmente la tecnología avanza a pasos agigantados y con ella la demanda de personas capacitadas para mantener esta tecnología. En los países de primer mundo saben que los profesionales capacitados no son suficientes y han buscado estos profesionales en los países en vía de desarrollo. Mayasoft Ingeniería Ltda es una empresa que, al descubrir esta falencia, se ha dedicado a prestar el servicio de Virtual Staff¹ (equipos de trabajo virtuales) a estas empresas extranjeras. Mayasoft Ingeniería Ltda presta el servicio de Virtual Staff a Mint Media-Interactive, una conocida empresa de desarrollo alemana, con la cual se trabaja de la mano con el propósito de realizar algunos desarrollos requeridos por distintos clientes. Actualmente los desarrolladores de ambas empresas laboran en los mismos proyectos, comunicándose vía SKYPE y utilizando VPN(Red Privada Virtual) para unificar las implementaciones realizadas por ambos equipos de trabajo.

Hoy día nace la necesidad de mejorar la forma como los usuarios obtienen la información de los distintos sistemas de software que ofrece la empresa. Muchas veces un usuario quiere obtener la información en un formato específico o diseñar la forma exacta en que quiere ver las consultas que realiza al sistema.

La implementación de JasperReports como motor generador de reportes es una salida acertada a la necesidad de personalizar la forma en que los usuarios ven la información que les brinda el sistema.

* Proyecto de grado

** Facultad de Ingeniería fisicomecánicas, Escuela de Ingeniería de Sistemas e Informática, Director: Emiro Muñoz Jerez.

ABSTRACT

TITLE: DEVELOPMENT, ADJUSTMENT AND IMLPEMENTATION OF A SOFTWARE MODULE FOR REPORTS MANAGEMENT USING JAVA FOR THE COMPANY MAYASOFT INGENIERIA LTDA.

AUTHORS: DANILO JOSÉ DÍAZ GÓMEZ

KEYWORDS: JasperReports, Virtual Staff, Mayasoft, Module, Mint Media-Interactive.

DESCRIPTION

Currently the technology is advancing by leaps and bounds and with it the demand for people trained to maintain this technology. In first world countries, they know that trained professionals are not enough and have searched for these professionals in other countries. Mayasoft Engineering Ltd is a company that, upon discovering this flaw, is dedicated to provide the Virtual Staff² service(virtual teams) to these foreign companies. Mayasoft Engineering Ltd provides the Virtual staff service to Mint Media-Interactive, a famous German development company, with which it works closely with the purpose of carrying out some development required by different customers. Currently, developers of both companies are working in the same projects, communicating via SKYPE and using VPN (Virtual Private Network) to standardize the implementations made by both teams.

Today, there is the need to improve the way users get information from different software systems offered by the company. Many times a user wants to obtain information in a specific format or design exactly how he wants to see the queries it makes to the system.

The implementation of JasperReports as a generator of reports is a successful exit to the need of customize the way users see the information system gives.

* Thesis of Degree

** Physicomechanical Engineering Faculty, School of Engineering and Computer Systems, Director: Emiro Muñoz Jerez.

INTRODUCCIÓN

Mayasoft Ingenieria Ltda es una empresa de desarrollo software que se ha encargado de la creación de sistemas de información orientados al manejo de recursos para Aerolíneas, el cual desde sus inicios hasta hoy ha estado evolucionando, hasta hacer de este un instrumento cada vez más útil, agradable y de fácil uso para sus usuarios. Con el objeto de llevar a cabo ésta labor se ha contado con herramientas software como lo son Java, ORACLE, entre otras herramientas que han permitido que el aplicativo sea dinámico y fácil de usar.

Aunque el aplicativo cuenta con una gran cantidad de módulos para ayudar al usuario a hacer diferentes tareas de una manera automatizada se contaba con una falencia a la hora de mostrar la información, la información siempre se mostraba en forma de tablas y el usuario lo más que podía hacer era imprimir la información de dichas tablas.

Después de estudiar detenidamente lo que no se podía hacer con el sistema actual y hacer varias entrevistas con los usuarios, se realizó un estudio de posibilidades para encontrar la manera mas optima de brindarle al usuario la posibilidad de escoger como quiere ver organizada la información, agregarle colores, datos adicionales, hacer nuevas consultas con la información disponible e incluso escoger el formato en que debería estar esta información, es decir dejar al usuario escoger entre una documento en PDF, una pagina web, una tabla de Excel, documento de Word, etc.

La herramienta seleccionada para lograr esto fue Jasperreports, con la cual existe la posibilidad de integrar IReport con los aplicativos desarrollados en JAVA con que cuenta la empresa. El usuario de este modo podría crear la forma en que

quiere los datos y simplemente solicitar la información requerida de la forma deseada.

En este documento se presenta el soporte teórico, metodológico y técnico del desarrollo del módulo de reportes para la empresa Mayasoft Ingeniería Ltda.

PRESENTACIÓN DEL PROYECTO

1.1 ORIENTACIÓN SOBRE EL CONTENIDO DEL INFORME

CAPITULO 1. Presentación del Proyecto: Se realiza un análisis de los antecedentes del proyecto, definición del problema, objetivos generales, específicos, justificación, alcances y limitaciones del proyecto.

CAPITULO 2. Marco Teórico: Se presentan los conceptos utilizados en el desarrollo técnico del proyecto.

CAPITULO 3. Marco Metodológico: En este capítulo se menciona el procedimiento metodológico que se siguió para la elaboración del proyecto y las razones por la que se escogió dicha metodología.

CAPITULO 4. Desarrollo de la Herramienta y Labores de Administración y Mantenimiento: Se presenta el análisis de requisitos y el diseño de la herramienta.

CAPITULO 5. Manual de Usuario: Se presenta una guía de uso de los servicios creados dentro del proyecto, y sobre como usar las herramienta implicadas en este desarrollo.

CAPITULO 6. Documentación de Pruebas del Sistema: se presenta un informe de las pruebas realizadas a la herramienta desarrollada y los resultados obtenidos.

CAPITULO 7. Conclusiones del Trabajo Realizado.

1.1.1 DESCRIPCIÓN DE LA EMPRESA

Mayasoft Ing. es una compañía conformada por profesionales especializados en el desarrollo e integración de soluciones informáticas capaces de satisfacer las más diversas necesidades a través de software amigable y de fácil manejo para el apoyo de los procesos empresariales.

Desde la constitución de la empresa, se ha dedicado exclusiva y permanentemente a la investigación, difusión y desarrollo de proyectos de informática, que tienen como características especiales la utilización de conceptos modernos de gerencia de proyectos e ingeniería de información. Esta política nos ha permitido alcanzar un alto grado de especialización convirtiéndose en nuestra principal fortaleza.

El objetivo de la empresa es convertirse en el mejor aliado de negocios ofreciendo soluciones integradas, desarrollos a la medida y consultoría que le permitan contar con información confiable para la toma de decisiones oportunas y acertadas orientadas a garantizar la sostenibilidad de el negocio del contratante.

1.1.2 Misión

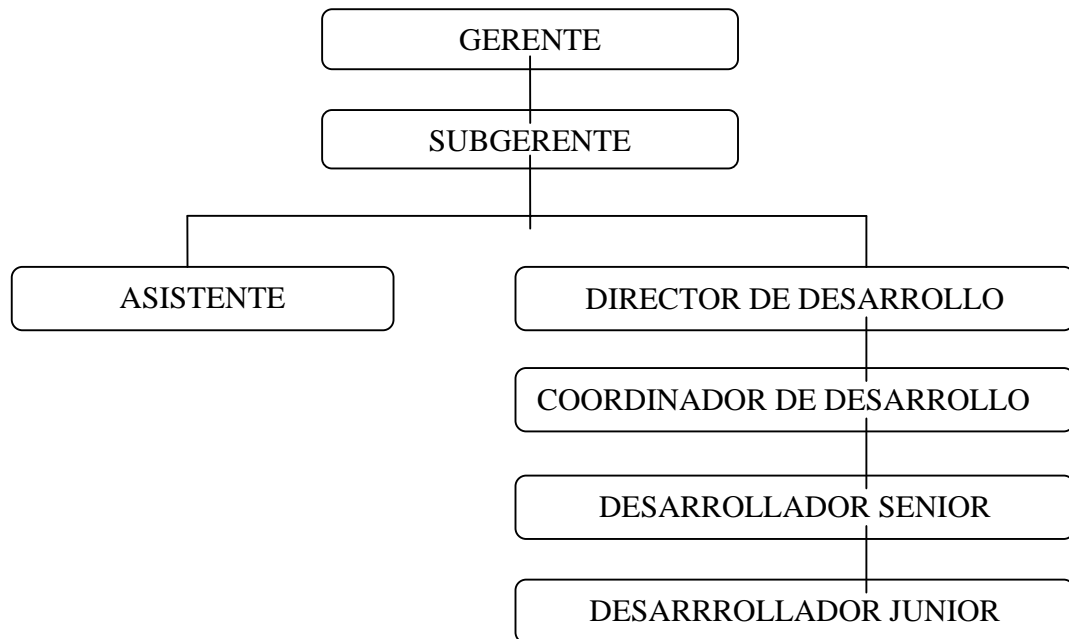
Construir soluciones informáticas capaces de satisfacer las necesidades de los clientes, proporcionándoles diversas herramientas que les permitan ser más eficientes y así obtener la diferencia que les haga más fuerte frente a sus competidores. Mantenernos a la delantera en temas de tecnología invirtiendo en ello los recursos necesarios en investigación y desarrollo que nos permitan conocer los avances que se presentan en el mundo.

1.1.3 Visión

Construir soluciones informáticas capaces de satisfacer las necesidades de los clientes, proporcionándoles diversas herramientas que les permitan ser más eficientes y así obtener la diferencia que les haga más fuerte frente a sus competidores. Mantenernos a la delantera en temas de tecnología invirtiendo en

ello los recursos necesarios en investigación y desarrollo que nos permitan conocer los avances que se presentan en el mundo.

1.1.4 Estructura Organizacional



1.1.5 Situación Actual

Mayasoft Ing. ha estado trabajando en diferentes proyectos. Pero hay uno en especial, es un proyecto que esta siendo desarrollado de la mano de una empresa alemana y que este a medida que pasa el tiempo este proyecto va creciendo, y con este crecimiento la forma de poder controlar todos los procesos es cada vez más difícil, a su vez el nivel de complejidad también crece, gracias a la forma en la cual se llevan los procesos internamente, del cual muchos métodos están entrelazados entre si, por esto también se hace complejo el seguimiento de los posibles errores. Es por esto que surgió la necesidad de tener una forma de testeo,

que pudiera ser comprensible, fácil de llevar un seguimiento de las pruebas, obtener de una forma clara los resultados y que sea robusto para que perdure a medida que el proyecto avanza y además debe ser flexible para que se pueda modificar si las circunstancias lo requieren.

1.2 OBJETIVOS

1.2.1 Objetivo General.

Realizar la implementación de nuevas funcionalidades a los módulos de manejo de recursos y manejo de solicitudes que ofrece la empresa Mayasoft Ingeniería Ltda. A través de la implantación de un generador de reportes con tecnología Java.

1.2.2 Objetivos Específicos:

- Adquirir conocimientos en los estándares de desarrollo de software de la empresa usados para garantizar la máxima calidad a la hora de desarrollar un proyecto.
- Realizar el diseño y el desarrollo de un modulo generador de reportes, el cual deberá permitir:
 - ✓ La inserción de nuevas plantillas de reportes por parte de cualquier usuario, las cuales deberán quedar almacenadas en la base de datos.
 - ✓ La modificación y borrado de plantillas que pertenezcan al usuario conectado al sistema.
 - ✓ La asignación de datos necesarios para la compilación de cada reporte.

- ✓ La importación de plantillas de reportes desde el sistema de archivos de Windows a la base de datos
- ✓ La exportación de plantillas de reportes desde la base de datos al sistema de archivos de Windows.
- Documentar cada una de las etapas de desarrollo del módulo a implementar
- Ejecutar y documentar las pruebas realizadas al módulo generador de reportes de la empresa Mayasoft Ltda, que permitan identificar fallos en el mismo.
- Realizar la documentación técnica y de usuario del módulo a entregar, para posterior revisión o modificación por parte de los desarrolladores vinculados a la empresa.
- Aprender el manejo de la herramienta IReport ³ para facilitar el diseño de las plantillas de cada reporte.

1.3 JUSTIFICACION

Actualmente la tecnología avanza a pasos agigantados y con ella la demanda de personas capacitadas para mantener esta tecnología. En los países de primer mundo saben que los profesionales capacitados no son suficientes y han buscado estos profesionales en los países en vía de desarrollo. Mayasoft Ingeniería Ltda es una empresa que, al descubrir esta falencia, se ha dedicado a prestar el servicio de Virtual Staff (equipos de trabajo virtuales) a estas empresas extranjeras.

Mayasoft Ingeniería Ltda presta el servicio de Virtual Staff2 a Mint Media-Interactive, una conocida empresa de desarrollo alemana, con la cual se trabaja de la mano con el propósito de realizar algunos desarrollos requeridos por distintos clientes. Actualmente los desarrolladores de ambas empresas laboran en

³ IReport es un constructor / diseñador de informes visual

los mismos proyectos, comunicándose vía SKYPE y utilizando VPN 3(Red Privada Virtual) para unificar las implementaciones realizadas por ambos equipos de trabajo. Hoy día nace la necesidad de mejorar la forma como los usuarios obtienen la información de los distintos sistemas de software que ofrece la empresa. Muchas veces un usuario quiere obtener la información en un formato específico o diseñar la forma exacta en que quiere ver las consultas que realiza al sistema.

La implementación de JasperReports4 como motor generador de reportes es una salida acertada a la necesidad de personalizar la forma en que los usuarios ven la información que les brinda el sistema.

Para esta práctica, se procedió a desarrollar e implantar un módulo para la generación de reportes apoyándonos en un generador de reportes como JasperReports, que permitirá a los usuarios obtener la información que necesite sobre algún recurso en particular. Estos reportes estarán disponibles bajo el formato deseado por el usuario, entre los que se proponen PDF, XLS, CSV, RTF, ODT, TXT.

1.4 ALCANCES Y LIMITACIONES

La implantación del modulo de reportes para la empresa Mayasoft Ingenieria permitirá que esta pueda responder a la demanda diaria de los usuarios, quienes podrán pedir la información de la manera que ellos crean conveniente a través de la creación de sus propias plantillas de resultados.

Por otra parte, al terminar el desarrollo y abandonar la empresa, el proyecto y estructura será enseñado a otro ingeniero escogido para brindar soporte en algún tiempo futuro.

2. MARCO TEÓRICO

2.1 ARQUITECTURA CLIENTE-SERVIDOR

En el esquema Cliente-Servidor dos o más procesos actúan autónomamente, pero en una forma coordinada y cooperativa, de este modo una aplicación solicita datos a otra e inmediatamente se recibe la petición, se procede a elaborar la respuesta y se devuelve a la aplicación demandante. Los principales componentes de esta arquitectura son los Clientes, los Servidores y la infraestructura de comunicaciones.

Las aplicaciones del lado del cliente interactúan con el usuario, normalmente usando una interfaz gráfica. Con frecuencia se comunican con procesos auxiliares que establecen una conexión con el servidor, enviar el pedido, recibir la respuesta, manejar las fallas y realizar actividades de sincronización y de seguridad.

Las aplicaciones del lado del servidor no tienen interfaz gráfica, sin embargo proporcionan un servicio al cliente y devuelven los resultados. En algunos casos existen procesos auxiliares que se encargan de recibir las solicitudes del cliente, verificar la protección, activar un proceso servidor para satisfacer el pedido, recibir su respuesta y enviarla al cliente.

Para que las aplicaciones del lado del cliente y del servidor se comuniquen, se hace necesaria una infraestructura de comunicaciones que proporciona los mecanismos básicos de direccionamiento y transporte (Interfaz de comunicaciones).

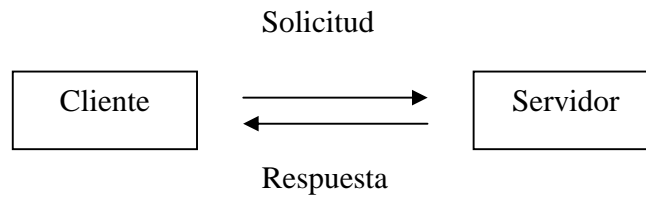


Figura 1. Modelo Cliente-Servidor

2.1.1 Características de la Arquitectura Cliente-Servidor.

- Las tareas de las aplicaciones de lado del cliente y del servidor tienen diferentes exigencias en cuanto a recursos de cómputo como velocidad del procesador, memoria, velocidad y capacidades del disco.
- Se establece una relación entre procesos distintos, los cuales pueden ser ejecutados en la misma máquina o en máquinas diferentes distribuidas a lo largo de la red.
- Las aplicaciones del lado del cliente corresponden a procesos con carácter activo porque hacen peticiones de servicios a los servidores, que tienen un carácter pasivo ya que esperan las peticiones de las aplicaciones del lado del cliente.
- El ambiente es heterogéneo. La plataforma de hardware y el sistema operativo del cliente y del servidor no son siempre la misma.
- El concepto de escalabilidad tanto horizontal como vertical es aplicable a cualquier sistema Cliente-Servidor. La escalabilidad horizontal permite agregar más estaciones de trabajo activas sin afectar significativamente el rendimiento. La escalabilidad vertical permite mejorar las características del servidor o agregar múltiples servidores.

2.1.2 Clasificación de las Arquitecturas Cliente-Servidor.

Los sistemas cliente servidor se clasifican de acuerdo al nivel de abstracción del servicio que se ofrece. Se distinguen tres componentes básicos de software:

- **Presentación:** Muestra al usuario un conjunto de objetos visuales y realiza el procesamiento de datos producidos por el mismo y los que son devueltos por el servidor.
- **Lógica de aplicación:** Es responsable del procesamiento de la información que tiene lugar en la aplicación.
- **Base de datos:** Esta compuesta por los archivos que contienen los datos de la aplicación.

2.1.2.1 Arquitectura Cliente-Servidor de Dos Capas.

- El sistema se separa en dos partes fijas: Las aplicaciones del lado del cliente y las aplicaciones del lado del servidor.
- La lógica de las aplicaciones debe estar en el cliente o en el servidor.
- La comunicación con el servidor es transparente para el usuario: El cliente solicita recursos y el servidor responde directamente a la solicitud, con sus propios recursos.

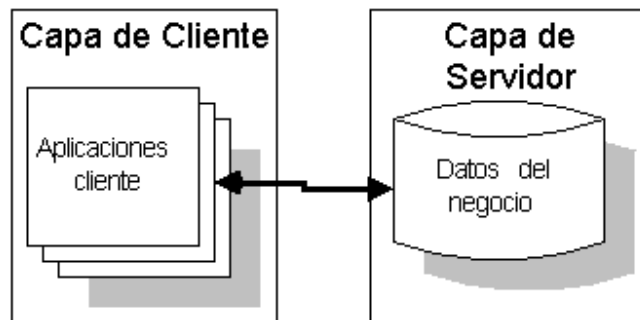


Figura 2. Esquema de Arquitectura Cliente-Servidor de Dos Capas

2.1.2.2 Arquitectura Cliente-Servidor de Tres Capas. Esta compuesta de:

- Un equipo cliente con una interfaz de usuario (normalmente se utiliza un navegador Web), que solicita los recursos.
- El servidor de aplicaciones (también es llamado software intermedio), cuya tarea es proporcionar los recursos solicitados, pero que requiere de otro servidor para hacerlo.
- El servidor de datos, que almacena y proporciona, al servidor de aplicaciones, los datos que requiere.

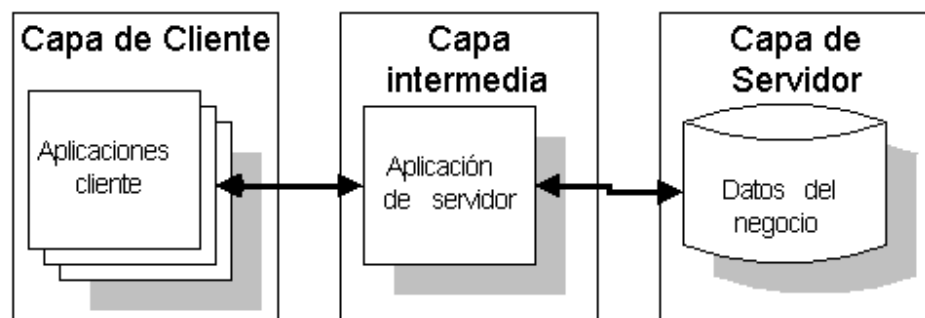


Figura 3. Esquema de Arquitectura Cliente-Servidor de Tres Capas

2.1.3 Arquitectura Cliente-Servidor usada.

Para el desarrollo de este proyecto, se utiliza arquitectura de tres capas, esta es la arquitectura que maneja Mayasoft en sus proyectos debido a las ventajas que ofrece como escalabilidad, facilidad de mantenimiento y el manejo de un mayor número de usuarios que la arquitectura Cliente-Servidor de dos capas. La arquitectura es aplicada de la siguiente forma:

- Capa de Cliente: Interfaz con el usuario, en este caso se ejecuta un jar que tiene contenido el aplicativo como tal
- Capa Intermedia: Para los servicios del negocio se utiliza una computadora configurada como servidor, en el cual se almacenan las capas restantes conformadas por clases especializadas y JavaBeans. Allí se realizan los procesos complejos, y se solicitan los servicios del servidor de datos cuando es necesario acceder a la información almacenada en la base de datos.
- Capa de Servidor: Se utiliza el motor de bases de datos ORACLE, el cual se encuentra en el mismo servidor Web.

2.1.4 Ventajas del Esquema Cliente-Servidor

- La arquitectura Cliente-Servidor facilita la integración entre sistemas heterogéneos y comparte información permitiendo, por ejemplo, que las máquinas ya existentes puedan ser usadas con interfaces más amigables al usuario.
- Al favorecer el uso de interfaces gráficas interactivas, los sistemas contruidos bajo este esquema son más intuitivas para el usuario.

- Proporciona, a los diferentes departamentos de una organización, soluciones locales, pero permitiendo la integración de la información principal globalmente.

2.1.5 Desventajas del Esquema Cliente-Servidor

- El mantenimiento de los sistemas es algo complicado sin la debida documentación, pues implica la interacción de diferentes partes de hardware y de software, distribuidas por distintos proveedores, lo cual dificulta el diagnóstico de fallas.
- Se cuenta con pocas herramientas para la administración y ajuste del desempeño de los sistemas, además, se debe tener estrategias para el manejo de errores y para mantener la consistencia de los datos.
- La seguridad de un esquema Cliente-Servidor es un factor importante a tener en cuenta. Por ejemplo, se deben hacer validaciones y verificaciones tanto en el cliente como en el servidor.
- Un inadecuado desempeño en una arquitectura de este tipo puede ocasionar congestión en la red, dificultad de tráfico de datos, etc.

2.2 TECNOLOGIA USADA: J2SE – JAVA STANDARD EDITION

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, como la manipulación directa de punteros o memoria.

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.

Orientado a Objetos

La primera característica, orientado a objetos (“OO”), se refiere a un método de programación y al diseño del lenguaje. Aunque hay muchas interpretaciones para OO, una primera idea es diseñar el software de forma que los distintos tipos de datos que usen estén unidos a sus operaciones. Así, los datos y el código (funciones o métodos) se combinan en entidades llamadas objetos. Un objeto puede verse como un paquete que contiene el “comportamiento” (el código) y el “estado” (datos). El principio es separar aquello que cambia de las cosas que permanecen inalterables. Frecuentemente, cambiar una estructura de datos implica un cambio en el código que opera sobre los mismos, o viceversa. Esta separación en objetos coherentes e independientes ofrece una base más estable para el diseño de un sistema software. El objetivo es hacer que grandes proyectos sean fáciles de gestionar y manejar, mejorando como consecuencia su calidad y reduciendo el número de proyectos fallidos. Otra de las grandes promesas de la programación orientada a objetos es la creación de entidades más genéricas (objetos) que permitan la reutilización del software entre proyectos, una de las premisas fundamentales de la Ingeniería del Software. Un objeto genérico “cliente”, por ejemplo, debería en teoría tener el mismo conjunto de comportamiento en diferentes proyectos, sobre todo cuando estos coinciden en cierta medida, algo que suele suceder en las grandes organizaciones. En este sentido, los objetos podrían verse como piezas reutilizables que pueden emplearse en múltiples proyectos distintos, posibilitando así a la industria del software a construir proyectos de envergadura empleando componentes ya

existentes y de comprobada calidad; conduciendo esto finalmente a una reducción drástica del tiempo de desarrollo. Podemos usar como ejemplo de objeto el aluminio. Una vez definidos datos (peso, maleabilidad, etc.), y su “comportamiento” (soldar dos piezas, etc.), el objeto “aluminio” puede ser reutilizado en el campo de la construcción, del automóvil, de la aviación, etc.

La reutilización del software ha experimentado resultados dispares, encontrando dos dificultades principales: el diseño de objetos realmente genéricos es pobremente comprendido, y falta una metodología para la amplia comunicación de oportunidades de reutilización. Algunas comunidades de “código abierto” (open source) quieren ayudar en este problema dando medios a los desarrolladores para diseminar la información sobre el uso y versatilidad de objetos reutilizables y bibliotecas de objetos.

Independencia de la plataforma

La segunda característica, la independencia de la plataforma, significa que programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware. Este es el significado de ser capaz de escribir un programa una vez y que pueda ejecutarse en cualquier dispositivo, tal como reza el axioma de Java, “write once, run everywhere”.

Para ello, se compila el código fuente escrito en lenguaje Java, para generar un código conocido como “bytecode” (específicamente Java bytecode)—instrucciones máquina simplificadas específicas de la plataforma Java. Esta pieza está “a medio camino” entre el código fuente y el código máquina que entiende el dispositivo destino. El bytecode es ejecutado entonces en la máquina virtual (JVM), un programa escrito en código nativo de la plataforma destino (que es el que entiende su hardware), que interpreta y ejecuta el código. Además, se suministran bibliotecas adicionales para acceder a las características de cada dispositivo (como los gráficos, ejecución mediante hebras o threads, la interfaz de red) de forma unificada. Se debe tener presente que, aunque hay una etapa explícita de

compilación, el bytecode generado es interpretado o convertido a instrucciones máquina del código nativo por el compilador JIT (Just In Time).

Hay implementaciones del compilador de Java que convierten el código fuente directamente en código objeto nativo, como GCJ. Esto elimina la etapa intermedia donde se genera el bytecode, pero la salida de este tipo de compiladores sólo puede ejecutarse en un tipo de arquitectura.

La licencia sobre Java de Sun insiste que todas las implementaciones sean “compatibles”. Esto dio lugar a una disputa legal entre Microsoft y Sun, cuando éste último alegó que la implementación de Microsoft no daba soporte a las interfaces RMI y JNI además de haber añadido características “dependientes” de su plataforma. Sun demandó a Microsoft y ganó por daños y perjuicios (unos 20 millones de dólares) así como una orden judicial forzando la acatación de la licencia de Sun. Como respuesta, Microsoft no ofrece Java con su versión de sistema operativo, y en recientes versiones de Windows, su navegador Internet Explorer no admite la ejecución de applets sin un conector (o plugin) aparte. Sin embargo, Sun y otras fuentes ofrecen versiones gratuitas para distintas versiones de Windows.

Las primeras implementaciones del lenguaje usaban una máquina virtual interpretada para conseguir la portabilidad. Sin embargo, el resultado eran programas que se ejecutaban comparativamente más lentos que aquellos escritos en C o C++. Esto hizo que Java se ganase una reputación de lento en rendimiento. Las implementaciones recientes de la JVM dan lugar a programas que se ejecutan considerablemente más rápido que las versiones antiguas, empleando diversas técnicas, aunque sigue siendo mucho más lento que otros lenguajes.

La primera de estas técnicas es simplemente compilar directamente en código nativo como hacen los compiladores tradicionales, eliminando la etapa del bytecode. Esto da lugar a un gran rendimiento en la ejecución, pero tapa el camino a la portabilidad. Otra técnica, conocida como compilación JIT (Just In Time, o “compilación al vuelo”), convierte el bytecode a código nativo cuando se ejecuta

la aplicación. Otras máquinas virtuales más sofisticadas usan una "recompilación dinámica" en la que la VM es capaz de analizar el comportamiento del programa en ejecución y recompila y optimiza las partes críticas. La recompilación dinámica puede lograr mayor grado de optimización que la compilación tradicional (o estática), ya que puede basar su trabajo en el conocimiento que de primera mano tiene sobre el entorno de ejecución y el conjunto de clases cargadas en memoria. La compilación JIT y la recompilación dinámica permiten a los programas Java aprovechar la velocidad de ejecución del código nativo sin por ello perder la ventaja de la portabilidad en ambos.

La portabilidad es técnicamente difícil de lograr, y el éxito de Java en ese campo ha sido dispar. Aunque es de hecho posible escribir programas para la plataforma Java que actúen de forma correcta en múltiples plataformas de distinta arquitectura, el gran número de estas con pequeños errores o inconsistencias llevan a que a veces se parodie el eslogan de Sun, "Write once, run anywhere" como "Write once, debug everywhere" (o "Escríbelo una vez, ejecútalo en cualquier parte" por "Escríbelo una vez, depúralo en todas partes")

El concepto de independencia de la plataforma de Java cuenta, sin embargo, con un gran éxito en las aplicaciones en el entorno del servidor, como los Servicios Web, los Servlets, los Java Beans, así como en sistemas empotrados basados en OSGi, usando entornos Java empotrados.

El recolector de basura

En Java el problema de las fugas de memoria se evita en gran medida gracias a la recolección de basura (o automatic garbage collector). El programador determina cuándo se crean los objetos y el entorno en tiempo de ejecución de Java (Java runtime) es el responsable de gestionar el ciclo de vida de los objetos. El programa, u otros objetos pueden tener localizado un objeto mediante una referencia a éste. Cuando no quedan referencias a un objeto, el recolector de basura de Java borra el objeto, liberando así la memoria que ocupaba previniendo

posibles fugas (ejemplo: un objeto creado y únicamente usado dentro de un método sólo tiene entidad dentro de éste; al salir del método el objeto es eliminado). Aun así, es posible que se produzcan fugas de memoria si el código almacena referencias a objetos que ya no son necesarios—es decir, pueden aún ocurrir, pero en un nivel conceptual superior. En definitiva, el recolector de basura de Java permite una fácil creación y eliminación de objetos, mayor seguridad y puede que más rápida que en C++ [cita requerida].

2.3 BASES DE DATOS

Una base de datos es un conjunto de datos que pertenecen a un mismo contexto y que son almacenados porque se consideran necesarios para una determinada organización o negocio.

2.3.1 Modelos de Bases de Datos.

Las bases de datos se pueden clasificar de acuerdo a su modelo de administración de datos. Algunos modelos con frecuencia utilizados en las bases de datos son:

2.3.1.1 Bases de Datos Jerárquicas.

Éstas son bases de datos que almacenan los datos de una manera similar a un árbol (invertido), en donde un *nodo padre* de información puede tener varios *hijos*. El nodo que no tiene padres es llamado *raíz*, y a los nodos que no tienen hijos se los conoce como *hojas*.

2.3.1.2 Base de Datos de Red.

En este modelo se permite que un mismo nodo tenga varios padres. Ofrece una solución eficiente al problema de redundancia de datos; sin embargo, la dificultad para administrar los datos en una base de datos de red ha conllevado a que sea un modelo usado más por programadores que por usuarios finales.

2.3.1.3 Base de Datos Relacional.

Éste es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Su principal idea es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos, también llamados tuplas. Cada relación es una tabla que está compuesta por registros (las filas de una tabla), que representan las tuplas, y campos (las columnas de una tabla). Los datos pueden ser recuperados o almacenados mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

El lenguaje más habitual para construir las consultas a bases de datos relacionales es el Lenguaje Estructurado de Consultas (Structured Query Language, SQL), un estándar implementado por los principales manejadores de bases de datos relacionales.

2.3.2 Acceso a Base de Datos.

Para desarrollar aplicaciones que conecten bases de datos, se utilizan interfaces y programas estándar que envían demandas escritas en SQL, y procesan los resultados. Para conectarse a un motor de bases de datos determinado, se necesita una interfaz estándar o controlador (en inglés: driver) que medie entre la aplicación y la base de datos.

2.3.2.1 Conectores más Utilizados

- **ODBC.** Es un programa de interfaz de aplicaciones (API) para acceder a datos en sistemas manejadores de bases de datos tanto relacionales como no relacionales, utilizando para ello el lenguaje de consulta estructurado (SQL). Se administran a través de la ventana ODBC del *Panel de Control*, En computadoras con sistema operativo Microsoft Windows.

- **MDB.** Servidor de bases de datos casi profesional. Esta aplicación permite trabajar con tablas de base de datos creadas en Microsoft Access 97/2000. Es posible abrir tablas en Lenguaje de consulta estructurado, visualizarlas, navegar, crear y borrar índices, fijar relaciones, copiar, etc.
- **JDBC.** La conectividad de bases de datos Java (Java Database Connectivity, JDBC) es una especificación de la interfaz de aplicación de programa (Application Programming Interface, API) para conectar los programas escritos en Java a los datos en bases de datos de mayor uso.

Para el desarrollo del modulo de reportes se empleó el conector JDBC, tal y como ya venía usando en los demás aplicativos desarrollados por la empresa Mayasoft Ingenieria. Uno de los mayores beneficios de usar el API JDBC es la capacidad para crear aplicaciones cuya programación sea independiente de la base datos, es decir, la mayoría de las aplicaciones que usan JDBC pueden ser migradas a otro servidor de bases de datos sin mayores complicaciones. Sin embargo, dos elementos siguen estando ligados a una base de datos en particular, el nombre de la clase que se usa para cargar el controlador (driver) JDBC y la dirección (Universal Resource Locator, URL) para acceder a la base de datos.

Las operaciones básicas realizadas durante la ejecución de un controlador JDBC son:

- Cargar un controlador JDBC.
- Utilizar ese controlador para abrir una conexión con la base de datos.
- Emitir instrucciones SQL a través de la conexión.
- Procesar los conjuntos de resultados devueltos por las operaciones SQL.

2.3.3 Manejadores o Gestores de Bases de Datos.

Son un tipo de software específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan, para almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las funciones principales de un gestor de bases de datos (DataBase Manager System, DBMS) son:

- Crear y organizar la Base de datos.
- Establecer y mantener las trayectorias de acceso a la base de datos de tal forma que los datos se puedan acceder rápidamente.
- Manejar los datos de acuerdo a las peticiones de los usuarios.
- Registrar el uso de las bases de datos.
- Interacción con el manejador de archivos. Esto a través de las sentencias en Lenguaje Manipulador de Datos (Data Manipulation Language, DML) al comando del sistema de archivos. Así el Manejador de base de datos es el responsable del verdadero almacenamiento de los datos.
- Respaldo y recuperación. Consiste en contar con mecanismos implantados que permitan la recuperación fácilmente de los datos en caso de ocurrir fallas en el sistema de base de datos.
- Control de concurrencia. Consiste en controlar la interacción entre los usuarios concurrentes para no afectar la inconsistencia de los datos.
- Seguridad e integridad. Consiste en contar con mecanismos que permitan el control de la consistencia de los datos evitando que estos se vean perjudicados por cambios no autorizados o previstos.

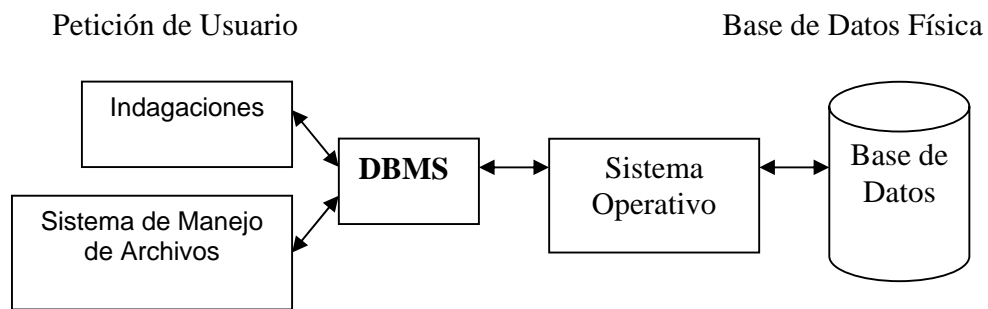


Figura 4. Función del DBMS

La figura 4 muestra el DBMS como interfaz entre la base de datos física y las peticiones del usuario. El DBMS interpreta las peticiones de entrada-salida del usuario y las manda al sistema operativo para la transferencia de datos entre la unidad de memoria secundaria y la memoria principal.

Un sistema manejador de base de datos es como el cerebro de la base de datos porque se encarga del control total de los posibles aspectos que la puedan afectar.

Existen diferentes manejadores de bases de datos como MySQL, ORACLE, FoxPro, Microsoft Access y PowerBuilder.

La empresa Mayasoft Ingenieria utiliza ORACLE.

2.3.3.1 ORACLE

Oracle es básicamente una herramienta cliente/servidor para la gestión de Bases de Datos. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hace que sólo se vea en empresas muy grandes y multinacionales, por norma general.

2.3.3.2 Ventajas de ORACLE

- Oracle es el motor de base de datos relacional más usado a nivel mundial.

- Puede ejecutarse en todas las plataformas, desde una Pc hasta un supercomputador
- Oracle soporta todas las funciones que se esperan de un servidor "serio": un lenguaje de diseño de bases de datos muy completo (PL/SQL) que permite implementar diseños "activos", con triggers y procedimientos almacenados, con una integridad referencial declarativa bastante potente.
- Permite el uso de particiones para la mejora de la eficiencia, de replicación e incluso ciertas versiones admiten la administración de bases de datos distribuidas.

2.4 PROGRAMACIÓN UTILIZADA

Para el desarrollo de este proyecto se usó la programación orientada a objetos (POO). La POO se basa en objetos y sus interacciones para el diseño de las aplicaciones, intenta simular el mundo real a través del significado de objetos que contienen características y funciones. La POO abstrae algunas características de sistemas naturales complejos como son:

- Atributos: Estado del objeto.
- Métodos: Comportamiento del objeto.
- Herencia: Comportamientos comunes entre objetos relacionados para hallar relaciones de especialización y generalización de comportamientos.

2.4.1 Clases.

Son colecciones de objetos de características idénticas. Cuando se programa un objeto y se definen sus características y funcionalidades, realmente lo que se programa es una clase. Por lo tanto, para realizar la abstracción de sistemas naturales, observamos y analizamos un grupo de cosas que tengan características

comunes, el resultado de esta abstracción será válido para todas y cada una de éstas cosas, y al conjunto de todas ellas es llamado “clase”.

2.4.2 Objetos.

Un objeto es cualquier cosa, real o abstracta, que posee atributos y un conjunto de operaciones que manipulan esos atributos; atributos y métodos que le dan al objeto un comportamiento particular. Un objeto es una instancia de una clase, el estado del objeto se determina por el estado (valor) de sus propiedades o características (atributos). Por ejemplo, al considerar un reloj suizo como objeto, sus atributos son, presión de agua que resiste, la hora que marca, etc.

2.4.3 Atributos.

Los atributos son las características de un objeto. Son un conjunto de datos (valores) y calificadores para aquellos datos. Estos atributos pueden ser desde tipos de datos simples (enteros, caracteres, cadenas de texto) hasta otros objetos.

2.4.4 Métodos.

Son funciones o procedimientos propios de la clase que pueden tener acceso a los atributos de la misma para realizar las operaciones para los que son programados.

2.4.5 Herencia.

Consiste en usar una clase ya creada para tomar sus características en clases más especializadas o derivadas de ésta para reutilizar el código que sea común con la clase base y solamente definir nuevos métodos o redefinir algunos de los existentes para ajustarse al comportamiento particular de esta subclase.

2.4.6 Beneficios de la POO.

- Permite obtener aplicaciones modificables y fácilmente extendibles a partir de componentes reutilizables.

- Disminución en el tiempo de desarrollo gracias a la reutilización del código.
- El desarrollo del software es más intuitivo porque la gente piensa naturalmente en términos de objetos más que en términos de algoritmos de software.

A continuación se presenta una breve descripción de JAVA, el lenguaje de programación orientado a objetos que se usó en el desarrollo de este proyecto:

2.4.7 Java Development Kit (JDK)

Para trabajar con Java se necesita un equipo (kit) de desarrollo que proporciona:

- Un compilador: *javac*
- Un intérprete: *java*
- Un generador de documentación: *javadoc*
- Otras herramientas complementarias.

Java es un lenguaje desarrollado por Sun Microsystems que permite el desarrollo de aplicaciones que pueden ejecutarse en casi cualquier plataforma. Java cuenta con una característica denominada “recolección de basura”, este programa examina la memoria y libera cualquier variable u objeto que no se esté usando, esto es de gran ayuda para los programadores aunque no le quita la responsabilidad de hacer programas limpios. El JDK es el entorno de desarrollo de JAVA.

3. MARCO METODOLÓGICO

3.1 PROTOTIPADO EVOLUTIVO

Para realizar la implementación del modulo de reportes se propone como metodología de desarrollo el Prototipado Evolutivo.

Desarrollo Evolutivo

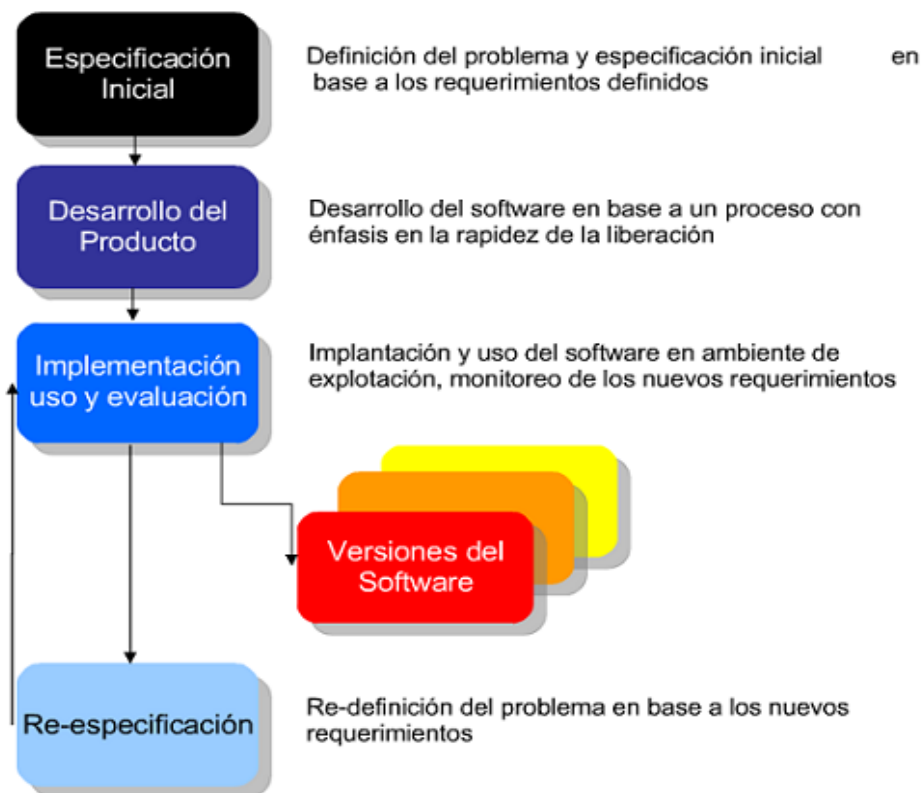


Figura 5. Prototipado Evolutivo

Fuente: es.wikipedia.org/wiki/Modelo_de_prototipos

En Ingeniería de software el desarrollo con prototipos, también llamado modelo de prototipos que pertenece a los modelos de desarrollo evolutivo, se inicia con la definición de los objetivos globales para el software, luego se identifican los requisitos conocidos y las áreas del esquema en donde es necesaria más definición. Entonces se plantea con rapidez una iteración de construcción de prototipos y se presenta el modelado (en forma de un diseño rápido). El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles para el cliente o el usuario final (por ejemplo, la configuración de la interfaz con el usuario y el formato de los despliegues de salida). El diseño rápido conduce a la construcción de un prototipo, el cual es evaluado por el cliente o el usuario, así se refinan los requisitos del software que se desarrollará. La iteración ocurre cuando el prototipo se ajusta para satisfacer las necesidades del cliente. Esto permite que al mismo tiempo el desarrollador entienda mejor lo que se debe hacer y el cliente vea resultados a corto plazo.

Para esta práctica el papel de evaluador de prototipos lo tomara el tutor de la práctica empresarial, siendo esta la persona que dará la información necesaria sobre los cambios pertinentes al prototipo.

Procedimiento a seguir para la Metodología Planteada:

- La construcción de prototipos comienza con la Recolección de los Requisitos.
- El desarrollador y usuario se reúnen y definen los objetivos globales para la aplicación, identifican todos los requisitos conocidos y perfilan las áreas en donde será necesaria una mayor definición.
- Luego se produce el Diseño del Prototipo que se enfoca sobre la representación de los aspectos del software visibles al usuario (por

ejemplo, métodos de entrada y formatos de salida) y se continúa con su desarrollo.

- El prototipo es evaluado por el usuario y se utiliza para refinar los requisitos del software a desarrollar.
- Se produce un proceso interactivo en el que el prototipo es “afinado” (Refinamiento del prototipo) para que satisfaga las necesidades del usuario, al mismo tiempo que facilita al desarrollador una mejor comprensión de lo que hay que hacer y poder entregar el producto final requerido o Producto de Ingeniería.

3.2 LENGUAJE DE MODELADO UNIFICADO

Un lenguaje es cualquier tipo de código semiótico organizado, para el que existe un argumento de uso y ciertos principios combinatorios formales, que al interactuar permiten representar o expresar algo.

El lenguaje de modelado es la notación (principalmente gráfica) que utilizan los métodos para llegar a un diseño, que permita comprender un sistema.

El Lenguaje Unificado de Modelado (UML) es un lenguaje gráfico que se usa para el modelado (visual) de sistemas de software, que permite especificar pero no describir métodos o procesos. Se usa para definir, visualizar, construir y documentar dichos sistemas. En otras palabras, es el lenguaje en el que está descrito el modelo.

UML no es un método de desarrollo, ya que no indica los pasos que se deben seguir para llegar al código, es decir, no especifica como pasar del análisis al diseño y de este al código.

Al UML no ser un método de desarrollo resulta ser independiente del ciclo de desarrollo que se siga, puede encajar en un ciclo en cascada, en un evolutivo, ciclo en espiral o en métodos de desarrollo rápido de aplicaciones (RDA).

Un diagrama es la representación gráfica de un conjunto de elementos con sus relaciones, ofreciendo así, una vista del sistema a modelar desde varias perspectivas. Para poder representar correctamente un sistema, UML ofrece una amplia variedad de diagramas:

- Diagrama de Casos de Uso.
- Diagrama de Clases.
- Diagrama de Objetos.
- Diagrama de Secuencia.
- Diagrama de Colaboración.
- Diagrama de Estados.
- Diagrama de Actividades.
- Diagrama de Componentes.
- Diagrama de Despliegue.

UML fue el lenguaje de modelado utilizado en el desarrollo de este proyecto, debido a que permite visualizar, especificar, construir y documentar un sistema a medida que este evoluciona en su ciclo de desarrollo.

3.2.1 Diagramas de UML.

Los diagramas de UML utilizados en el desarrollo de este proyecto fueron: diagramas de casos de uso y diagramas de secuencias. Las principales razones por las cuales se optó por UML como el lenguaje de modelado son:

- UML facilita el entendimiento de la información, la función y el comportamiento de un sistema, haciendo así más fácil y sistemático el análisis de los requerimientos, ya que sirve de apoyo en los procesos de análisis de un problema.
- UML permite a los desarrolladores de sistemas realizar diseños que facilitan la comunicación a otras personas de manera convencional.
- UML permite generar un punto de comparación entre lo que se ha logrado y lo planificado.
- UML tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas de clases, objetos, etc., hasta la implementación y configuración con los diagramas de despliegue.

3.2.1.1 Diagramas de Casos de Uso

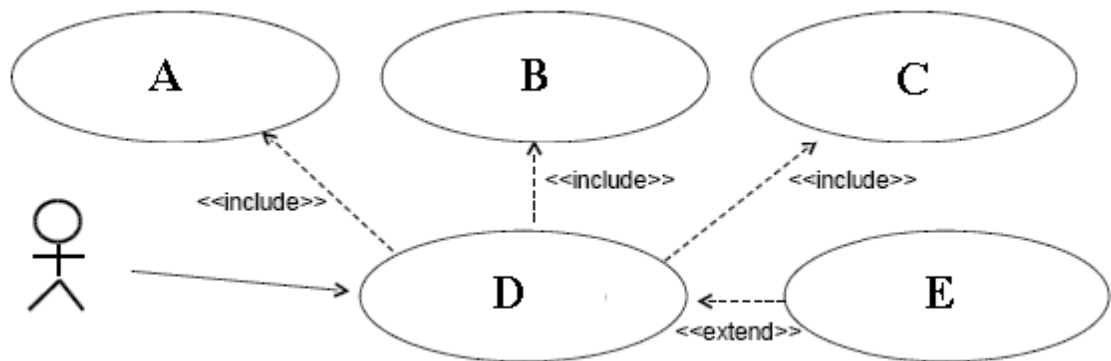


Figura 6. Diagrama de Casos de Uso

Fuente: es.wikipedia.org/wiki/Diagrama_de_casos_de_uso

Un Diagrama de Casos de Uso es una representación gráfica del entorno del sistema (actores) y su funcionalidad principal (casos de uso). Un Diagrama de Casos de Uso describe lo que hace un sistema desde el punto de vista de un observador externo; concentrándose en expresar lo que hace el sistema, y no en dar respuesta a un cómo lograr su comportamiento.

Inclusión (Include): Es una forma de interacción, un caso de uso dado puede "incluir" otro. Una inclusión es utilizada para indicar que un caso de uso depende de otro, es decir, la funcionalidad de determinado caso de uso se requiere para realizar las tareas de otro. En la figura 7 el caso de uso "D" depende de los casos de uso "A", "B" y "C".

Extensión (Extend): Es otra forma de interacción. Una extensión representa una variación de un caso de uso a otro, es decir, una dependencia específica entre los casos de uso, a través de la cual un caso de uso (la extensión) puede extender a otro.

3.2.1.2 Diagramas de secuencias.

Un diagrama de secuencia es un diagrama de interacción que muestra los objetos como líneas de vida y sus interacciones en el tiempo representadas como mensajes dibujados como flechas desde la línea de vida origen hasta la línea de vida destino. Los diagramas de secuencia sirven para mostrar qué objetos se comunican con qué otros objetos y qué mensajes transmiten esas comunicaciones.

A veces un diagrama de secuencia tendrá una línea de vida con un símbolo del elemento actor en la parte superior, que al igual que en el diagrama de casos de uso, es el usuario que interactúa de alguna manera con el sistema. Éste usualmente sería el caso si un diagrama de secuencia es contenido por un caso de uso.

Los elementos entidad, control y límite de los diagramas de robustez también pueden contener líneas de vida, donde el elemento límite es el lugar donde se almacenan los datos, en este caso la Base de Datos, el elemento control hace referencia al proceso de interacción Interfaz – Base de Datos y el elemento entidad es la interfaz con la que interactúa el usuario.

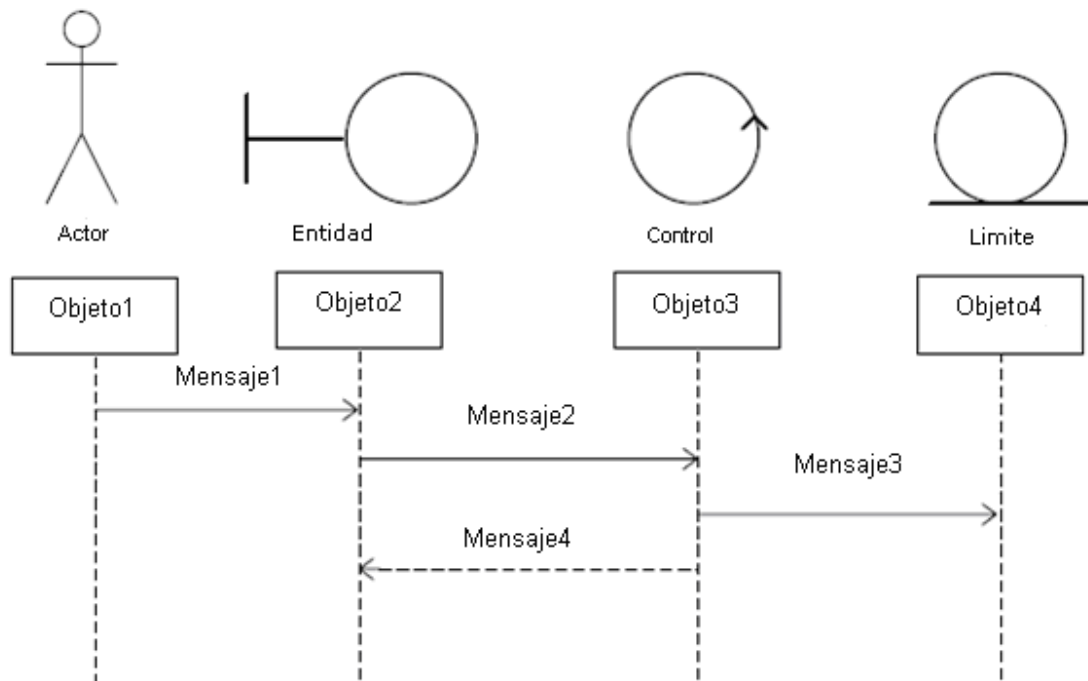


Figura 7. Diagrama de Secuencias

3.3 ESTÁNDARES DE PROGRAMACIÓN

3.3.1 Modelo de datos.

Los nombres de las tablas de la base de datos y sus respectivos campos se escriben con mayúscula inicial en cada palabra que conforme su nombre, las demás letras se escriben en minúscula.

3.3.2 Nombres de las tablas.

Los nombres de las tablas de la base de datos son usados en su forma plural.

3.3.3 Clases.

Los nombres de las clases deben ser sustantivos en singular y en inglés, la primera letra de cada palabra que lo componga debe ser mayúscula y debe ir precedida por el prefijo Mint. Éstos deben ser simples, descriptivos y en lo posible evitar el uso de abreviaciones y acrónimos. Ejemplo: MintEmployeeTable.java.

4. DESARROLLO DEL MODULO DE REPORTES USANDO JASPERREPORTS

A continuación se explicara como se planteo que debería ser la arquitectura del desarrollo que iba a ser entregado al cliente y las funcionalidades y características con que este modulo de reportes iba a contar

4.1 Arquitectura

La arquitectura del modulo de reportes constaría de:

- Un repositorio de plantillas (template repository) en la base de datos.
- Métodos API para transportar los reportes, la información de estos y los resultados ya formateados obtenidos al ejecutar dichos reportes.
- Un modulo dentro del aplicativo de la empresa (Mint WebAssistant) para manipular las plantillas (Crear, editar, ejecutar o borrar).
- Un visor de reportes, en el cual poder obtener los resultados y realizar las exportaciones en los formatos deseados por el usuario (PDF, Word, etc)

4.2 Requerimientos funcionales

4.2.1 Caso de uso: Importar una plantilla diseñada en IReport

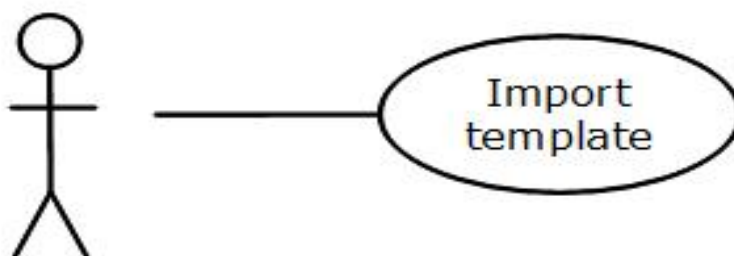


Figura 8. Importar plantilla

Descripción

El usuario puede importar cualquier plantilla al repositorio de archivos. Estos archivos se importan a los nodos de un árbol que ayuda a la organización de dichas plantillas.

Nota: Solo un usuario con los suficientes privilegios puede realizar esta tarea.

Descripción paso a paso

- El usuario selecciona el nodo del árbol donde va a agregar la nueva plantilla
- El usuario hace click derecho sobre el nodo del árbol
- Se despliega un menú contextual.
- Se escoge la opción “Nuevo”
- El aplicativo muestra un diálogo para la configuración de las propiedades que tendrán juego sobre la plantilla
- El usuario asigna las propiedades a dicha plantilla
- Si toda la información obligatoria para la plantilla está completa, entonces el aplicativo guardará la información junto con la plantilla en la base de datos.

4.2.2 Caso de uso: Importar plantilla a un nodo ya existente

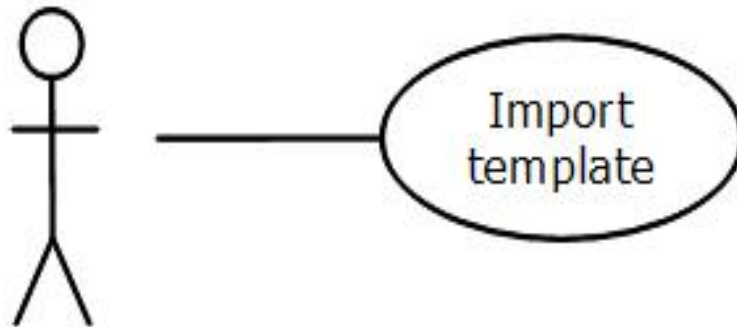


Figura 9. Importar plantilla sobre nodo existente

Descripción:

El usuario puede importar cualquier plantilla sobre un nodo ya existente del árbol. El usuario puede decidir en cual nodo hará la importación

Nota: Solo un usuario con los suficientes privilegios puede agregar una plantilla a un nodo existente

Descripción paso a paso:

- El usuario selecciona el nodo existente y hace clic derecho sobre este.
- El menú contextual es desplegado
- Se selecciona la opción de editar
- El usuario agrega la plantilla buscándola con el explorador de archivos
- Si todos los campos obligatorios están correctamente diligenciados, entonces la plantilla y la información registrada será almacenada en la base de datos.

4.2.3 Caso de uso: Exportar plantilla

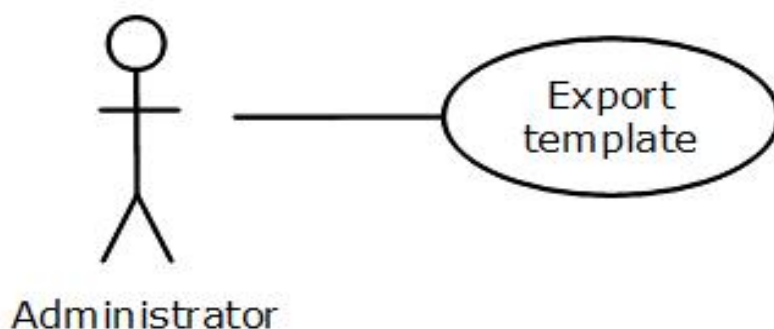


Figura 10. Exportar plantilla

Descripción:

El usuario puede exportar cualquier plantilla desde el repositorio de archivos a un sistema de archivos externo. El usuario escoge cual plantilla va a guardar y donde va a hacerlo.

Nota: Solamente un usuario que tiene los suficientes privilegios para administrar las plantillas puede hacer este procedimiento.

Descripción paso a paso

- El usuario selecciona el nodo que contiene la plantilla que desea exportar a su sistema de archivos y hace clic derecho sobre la plantilla.
- El menú contextual es desplegado.

- El usuario escoge “Exportar plantilla”. Esta opción solo será ofrecida si el usuario selecciona un nodo que contenga una plantilla
- El explorador de archivos es desplegado.
- El usuario escoge el lugar donde va a salvar la plantilla.
- La plantilla es salvada en el folder escogido.

4.2.4 Caso de uso: Generar vista previa

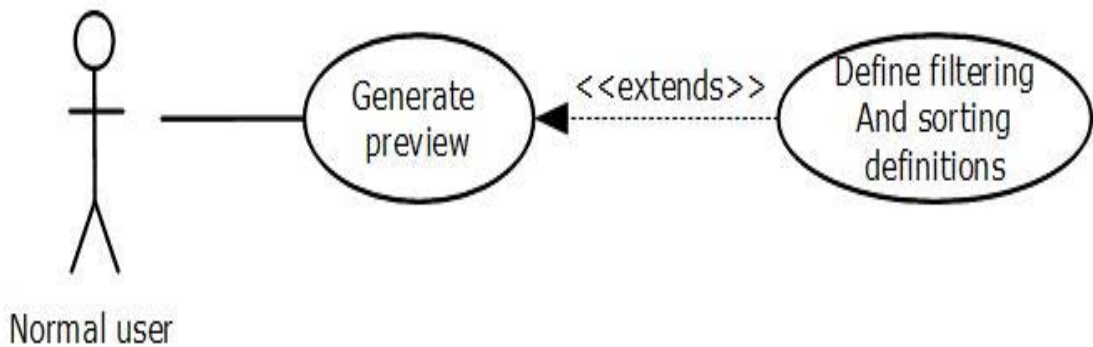


Figura 11. Generar vista previa

Descripción:

El usuario puede generar/ejecutar cualquier reporte y visualizarlo en la ventana de vista previa que tiene el modulo de reportes

Descripción paso a paso:

Primer caso:

- El usuario selecciona una plantilla, la cual es la que desea ejecutar y ver su vista previa
- El usuario hace clic derecho sobre la plantilla y se despliega el menu contextual.
- El usuario selecciona “generar vista previa”
- El modulo despliega un wizard con dos pasos:
 1. Parámetros: En el primer paso el usuario selecciona las propiedades que desea usar como parámetros en la ejecución del reporte, sus operadores y valores. El usuario puede agregar y remover propiedades a su gusto.

2. Orden: En el Segundo paso el usuario selecciona las propiedades que quiere que sean las que lleven el orden de los resultados
- El usuario hace clic en el botón “Cerrar” y el wizard se cierra sin ejecutarse todavía.
 - El usuario seguidamente hace clic sobre el botón aplicar y se prosigue con la ejecución del reporte con la configuración establecida.
 - El modulo genera la imagen previa de la ejecución del reporte en la parte derecha de la pantalla.

4.3 Diseño de plantillas y conexión de IReport con el aplicativo

A continuación se da una descripción de cómo se debe configurar la herramienta IReport para que esta sea capaz de conectarse al MINT WebAssistant (Aplicativo sobre el cual se desarrolla el modulo de reportes).

Los pasos son los siguientes:

1. Descargar de jasperforge.org/projects/ireport la **version 3.5.1 de IReport**. Esta version es la recomendada debido a que los JARS que tiene el Aplicativo fueron tomados de esta para lograr la compatibilidad.
2. En la carpeta que se descarga se encuentra el ejecutable. Esta es el folder: ireport/bin/.

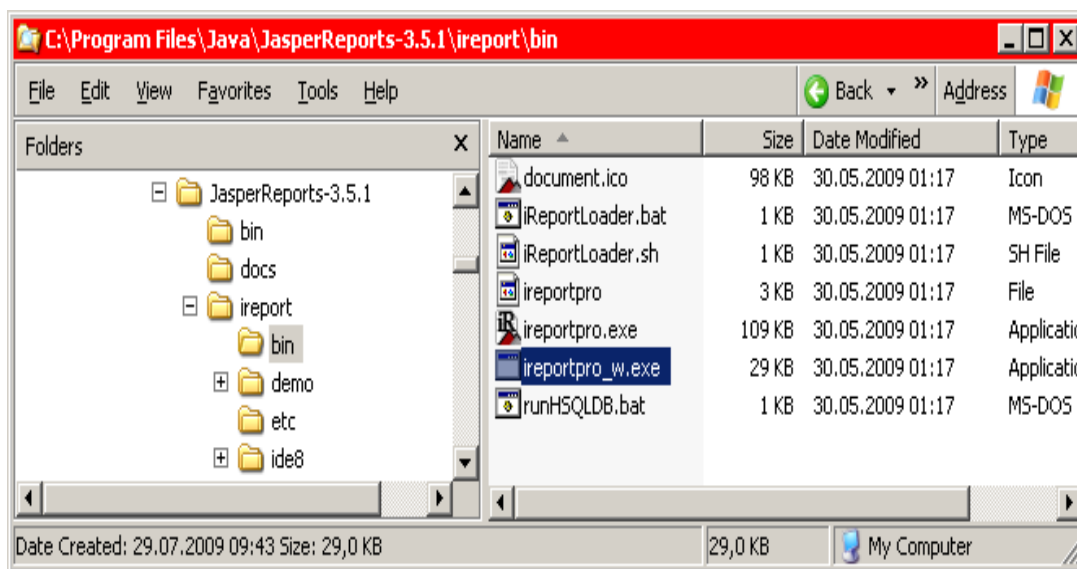


Figura 12. Carpeta bin de IReport

3. Ahora hay que dirigirse a *Extras --> Options* del menu principal y agregar los JARS correspondientes al classpath de Ireport.

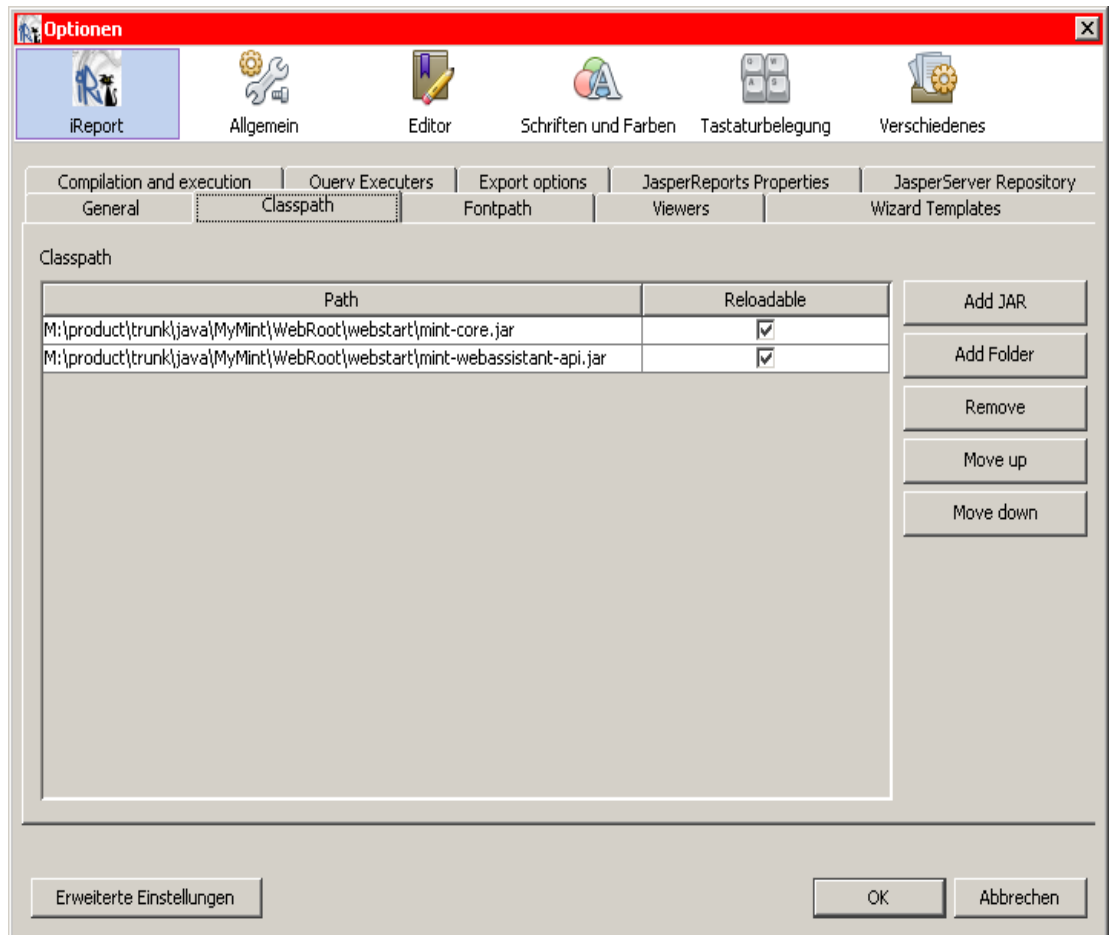


Figura 13. Configuración Classpath

4. Lo siguiente es agregar la fuente de datos correspondiente. Para la ilustración agregaremos la fuente de datos que contiene la información de los empleados del sistema.
 - o En la barra de herramientas principal se hace clic sobre el icono señalado para agregar una nueva fuente de datos:

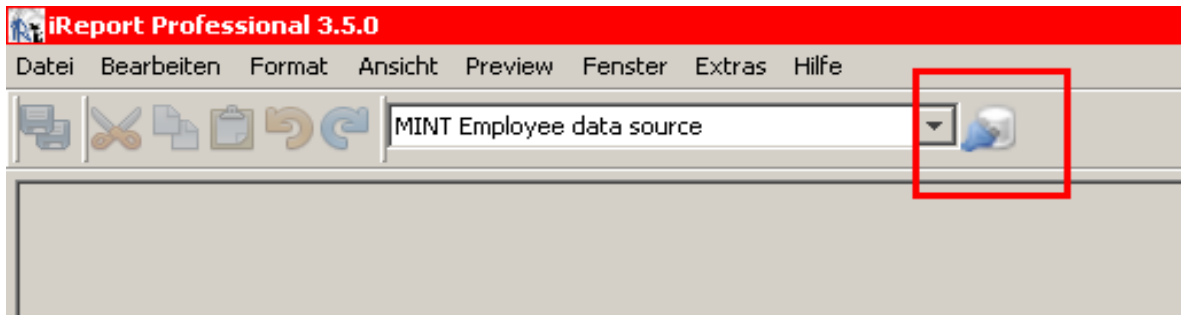


Figura 14. Abrir dialogo de fuentes de datos

- Se selecciona el botón “New” para agregar la nueva fuente de datos

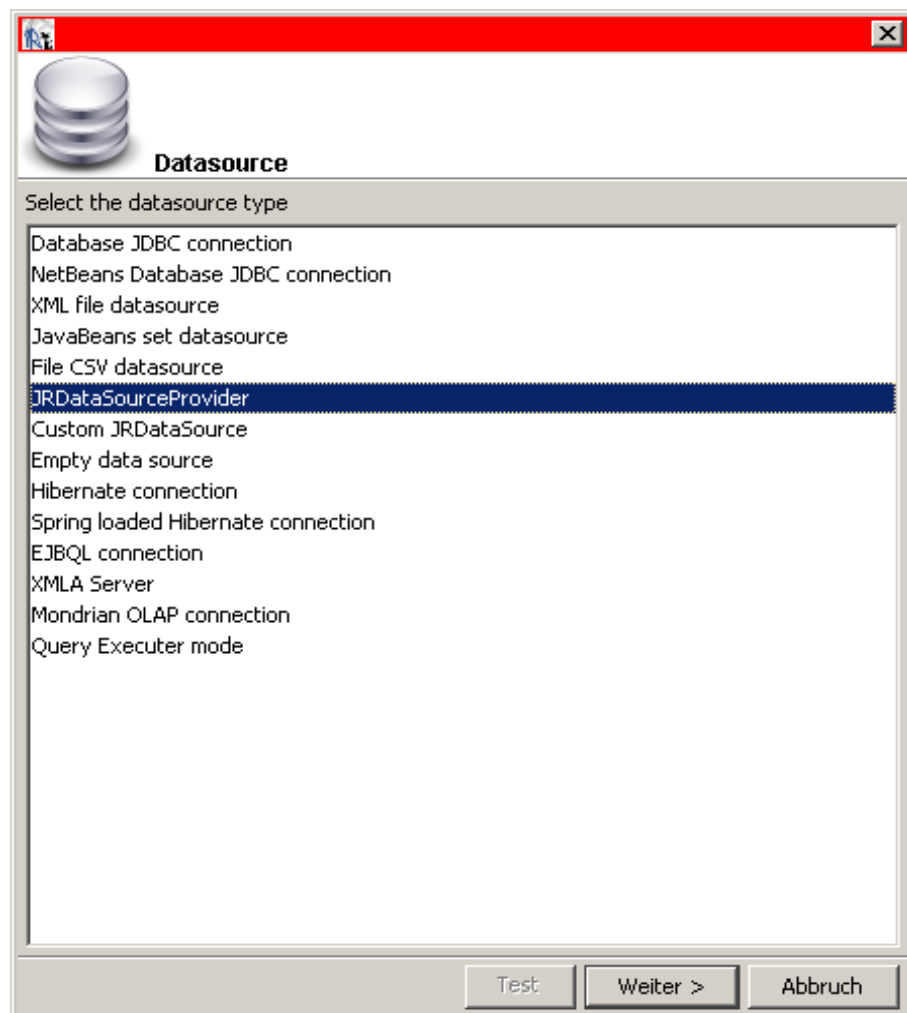


Figura 15. Lista tipos fuentes de datos

- Se selecciona JRDataSourceProvider de la lista de fuentes de datos desplegada
- Ahora se tiene que definir la clase que contiene la fuente de datos. Aquí es necesario colocar el class path de la clase que tiene la fuente de datos para empleas.
En este caso es `mint.api.domain.report.MintEmployeeDataSource`.

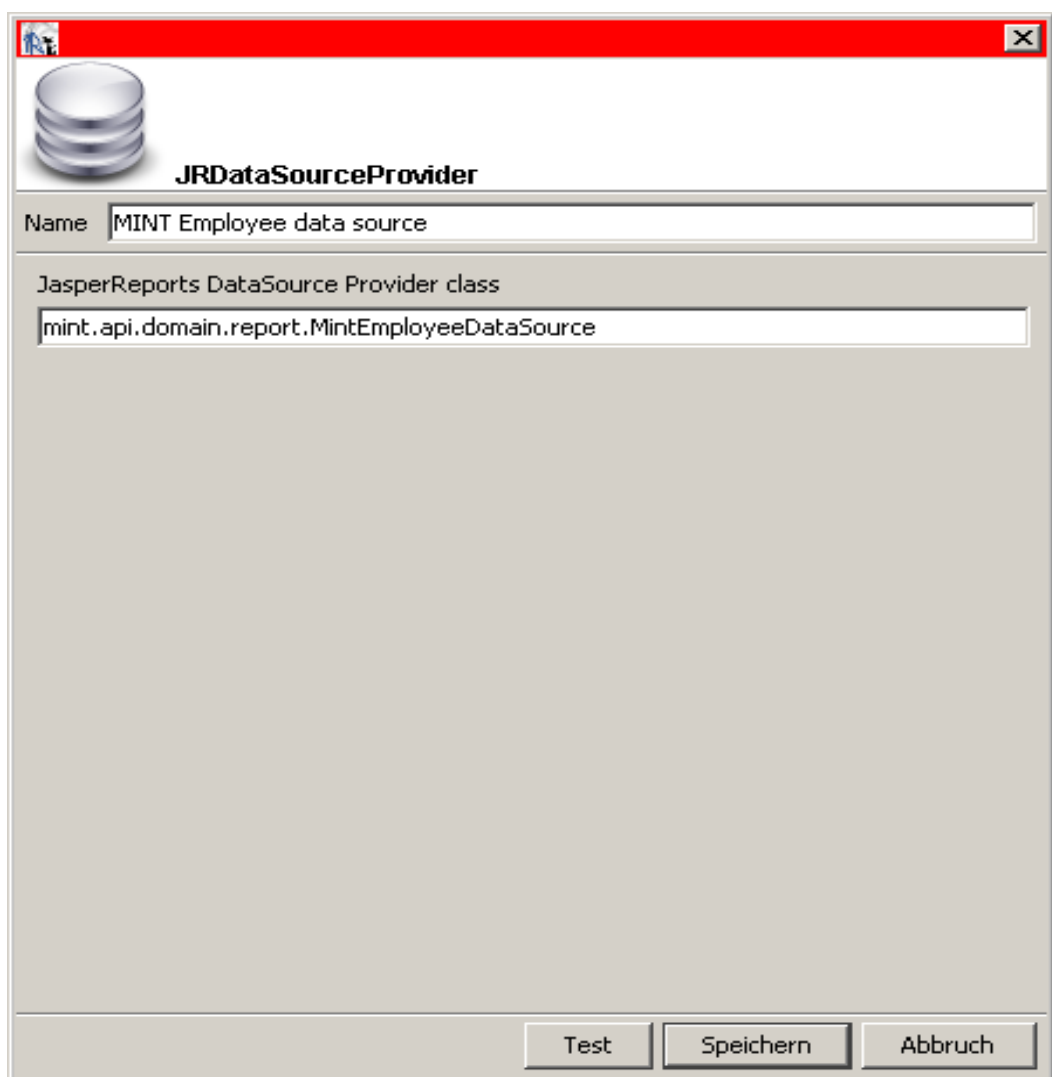


Figura 16. Seteo de ruta de clase

5. Para poder usar campos en el reporte que provienen del entorno del modulo de reports (es decir que desde IReport se puedan ver las propiedades existentes de los objetos del aplicativo de la empresa), debemos cargarlos primero. Para esto se hace clic en el icono de las fuentes de datos:

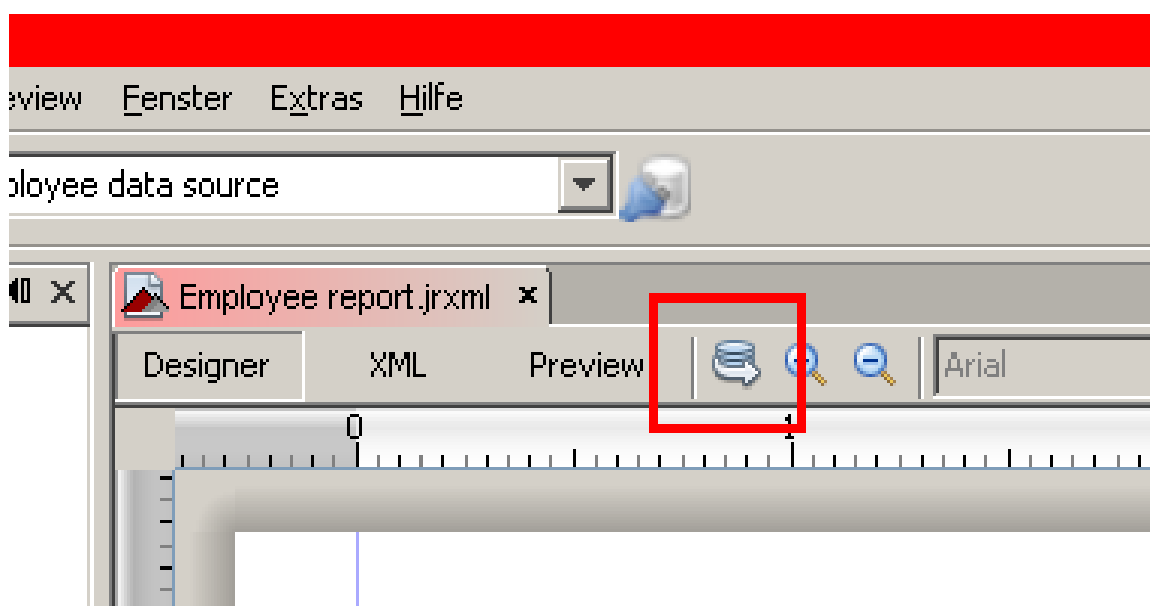


Figura 17. Agregar campos

6. En el dialogo que se despliega, haga clic en la pestaña "DataSource Provider" y seguidamente en el botón "Get fields form Datasource". Con esto lo que se logra es listar todos los atributos que se configuraron como disponibles dentro de nuestra fuente de datos seleccionada. Estos son los campos que se pueden usar para configurar nuestra plantilla.

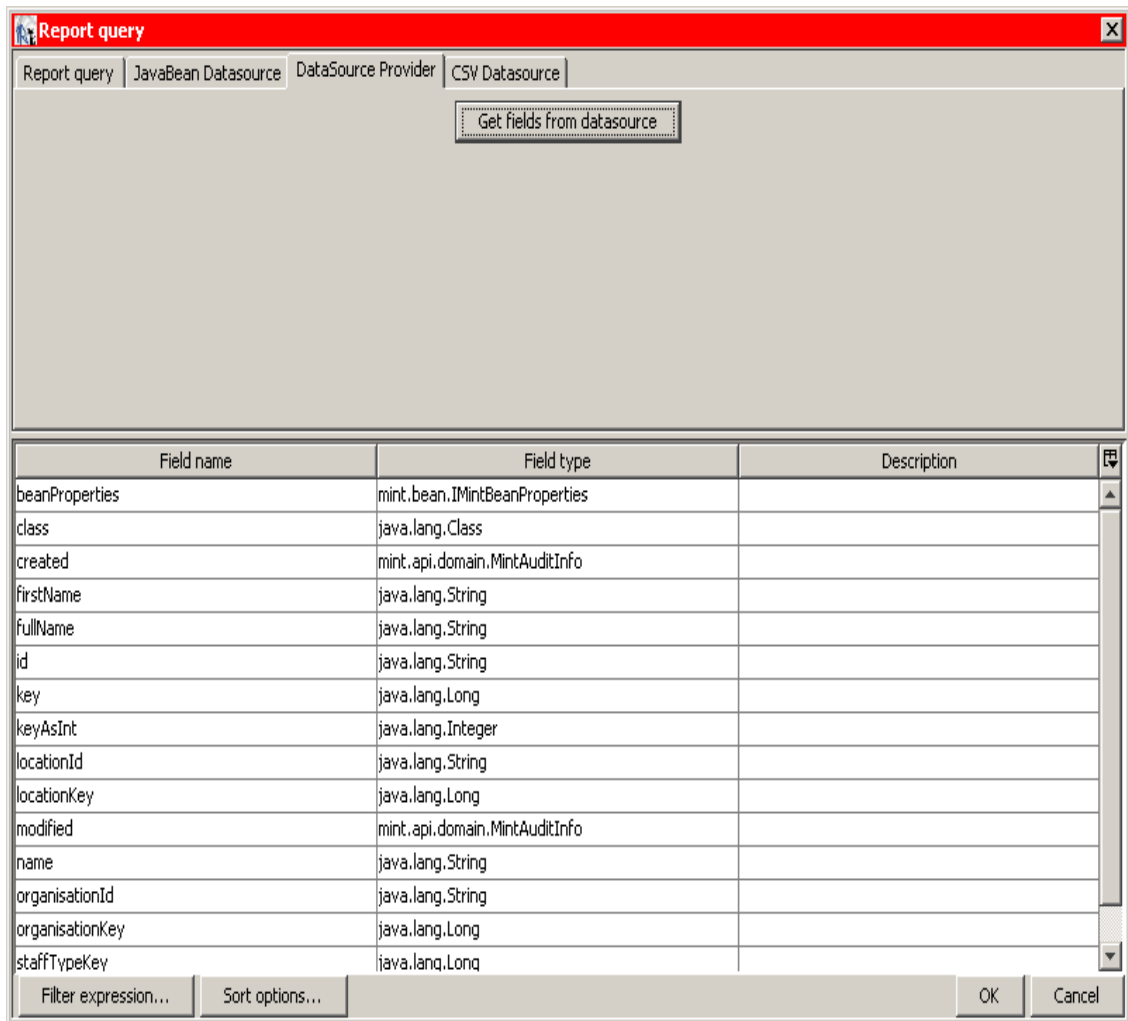


Figura 18. Listar campos

7. Luego de hacer clic en "OK", estos campos son cargados en el árbol de campos al lado de la plantilla sobre la que se está trabajando. Estos atributos se pueden usar haciendo un simple arrastre de ellos sobre la plantilla y se colocan y configuran a decisión del usuario

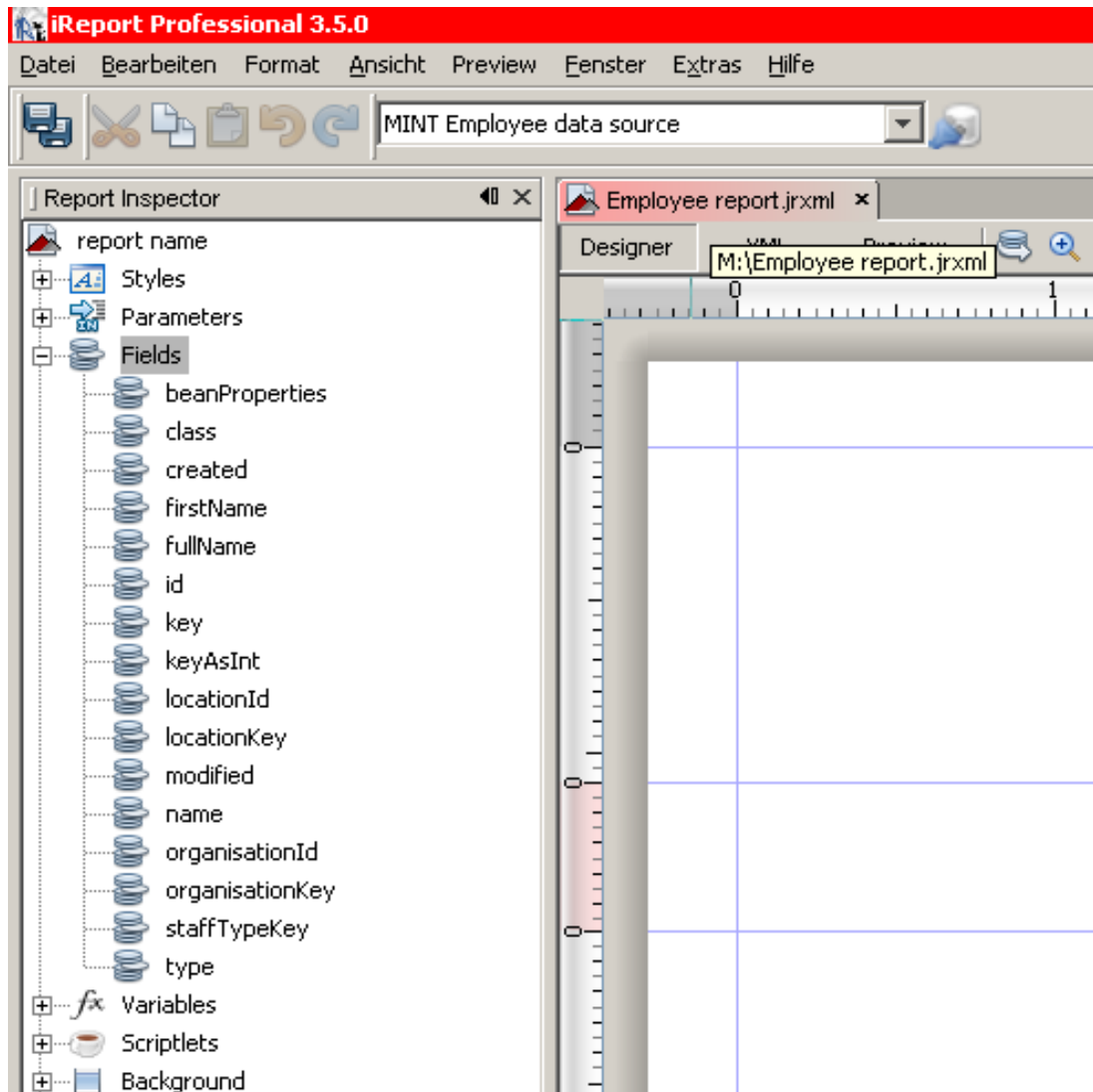


Figura 19. Crear plantilla

4.4 Documentación técnica

4.4.1 Tabla de plantillas en la base de datos:

La siguiente es la lista de campos que tiene la tabla que mantiene las plantillas en la base de datos del aplicativo:

report_template_key	NUMBER(11) NOT NULL	-- primary key
id	VARCHAR2(255)	-- id unico

```

name VARCHAR2(255) NOT NULL          -- nombre unico
description VARCHAR2(4000)          -- descripción opcional
data_source_name VARCHAR2(255) NOT NULL  --nombre fuente de datos
is_public NUMBER(1) NOT NULL        -- plantilla publica/privada
owner NUMBER(11) NOT NULL           -- identificador del usuario dueño
template CLOB NOT NULL              -- el xml de la plantilla
created_by NUMBER(11) NOT NULL
created_date DATE NOT NULL
modified_by NUMBER(11) NOT NULL
modified_date DATE NOT NULL

```

4.4.2 Javabeans creados para soportar el manejo de reportes

MintReportTemplate: Este bean es el que se encarga de representar un objeto plantilla que se ha obtenido o que se va a guardar en la base de datos.

Su estructura es la siguiente:

```

public class MintReportTemplate extends MintNamedKeyObject implements IMintIdObject {
    private String id;
    private String description;
    private EMintReportTemplateType type;
    private String dataSourceName;
    private boolean isPublic;
    private boolean isSystemTemplate;
    private boolean isSystemDataSource;
    private String owner; // the username of the owner
    private String template;
}

```

MintReportTemplateFilter: Este objeto sirve para crear un filtro por el cual es posible realizar una búsqueda mas precisa a la hora de realizar una consulta de reportes con ciertas características particulares.

Su estructura es la siguiente:

```

public class MintReportTemplateFilter {
    private Set<EMintReportTemplateType> types;
    private Set<String> dataSourceNames;
}

```

```

private Boolean publicReportTemplates;
private Boolean systemReportTemplates;
private Boolean systemReportDataSources;
private Set<String> owners;
}

```

MintReportResult: Este objeto se usa para guardar el contenido de la ejecución de un reporte. En sus atributos guardamos los bytes que contienen el reporte en el formato deseado. Los formatos soportados son:

- HTML
- PDF
- XLS
- CSV
- RTF
- ODT
- TXT

```

public class MintReportResult extends MintNamedKeyObject implements IMintIdObject
{
private String username;
private MintTimestamp timestamp;
private Set<EMintReportFormat> initializedFormats;
private String html;
private byte[] pdf;
private byte[] xls;
private String csv;
private String rtf;
private byte[] odt;
private String txt;
}

```

public boolean saveAs(String path, EMintReportFormat format) throws IllegalArgumentException:

El objeto MintReportResult contiene un método llamado SaveAs(), el cual se encarga de salvar el resultado del reporte en el atributo correspondiente según el formato de reporte que se desee.

Este metodo consulta ejecuta directamente el reporte desde el cliente, este a su vez solicita la información al servidor donde ocurre la compilación y ejecución de dicho reporte:

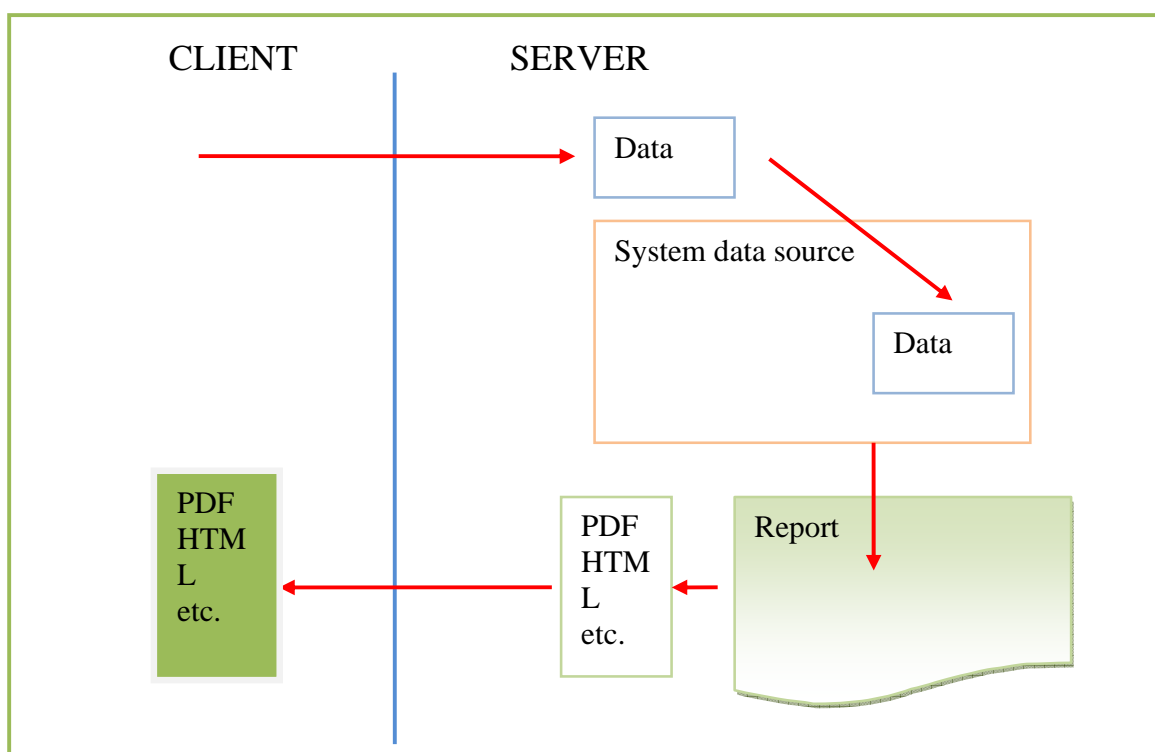


Figura 20. Workflow de la ejecución de un reporte

Fuente: jasperforge.org/jasperreports

4.4.3 Metodos de API implementados

public static MintReportTemplate getReportTemplate(Long key, String id) throws MintUnprivilegedException:

Este método obtiene un reporte usando la llave/id que se pasen como parámetros, si ambos argumentos son dados entonces la key será la única usada para identificar el reporte.

La excepción `MintUnprivilegedException` será lanzado en caso de que un usuario que no sea dueño de dicho reporte este tratando de accederlo.

`public static List<MintReportTemplate> getReportTemplates(List<Long> keys, List<String> ids) throws MintUnprivilegedException:`

Este método obtiene uno o varios reportes usando las llaves/ids que se pasen como parámetros, si ambos argumentos son dados entonces las keys será la única usada para identificar los reportes.

La excepción `MintUnprivilegedException` será lanzado en caso de que un usuario que no sea dueño de dicho reporte este tratando de accederlo. Tambien se producirán excepciones si alguna key no existe o si existe keys duplicadas.

`public static List<MintReportTemplate> getReportTemplatesByFilter(MintReportTemplateFilter filter) throws MintUnprivilegedException:`

Este método obtiene todos los reportes de que concuerden con el filtro que se pasa como argumento en el método. Si el filtro es nulo, se entiende que se desean traer todos los reportes contenidos en la base de datos.

La excepción `MintUnprivilegedException` será lanzado en caso de que un usuario este en la lista de usuarios y este no corresponda al usuario que está haciendo la llamada al método.

`public static MintReportTemplate insertReportTemplate(MintReportTemplate reportTemplate):`

Este método inserta en la base de datos un nuevo reporte. Al terminar la inserción el objeto devuelto por el método ya cuenta con la llave con que quedo almacenado en la base de datos.

El método lanza una `IllegalArgumentException` si alguno de los campos obligatorios en la tabla de templates no está presente dentro del vean que se pasa como parámetro al método.

public static void updateReportTemplate(MintReportTemplate reportTemplate):

Este método actualiza un reporte ya existente en la base de datos usando la información contenida en el bean que se pasa como parámetro.

El método lanza una `IllegalArgumentException` si la llave contenida en el bean no existe, ya que sin esta no se puede encontrar el reporte a ser actualizado.

public static void deleteReportTemplate(Long key, String id):

Este método elimina un reporte de la base de datos usando una llave/id para identificarlo.

Si la llave/id no corresponde a ningún reporte en la base de datos entonces es lanzada una `IllegalArgumentException`.

4.4.4 Logica y organización de la arquitectura de reportes dentro del MINT WebAssistant

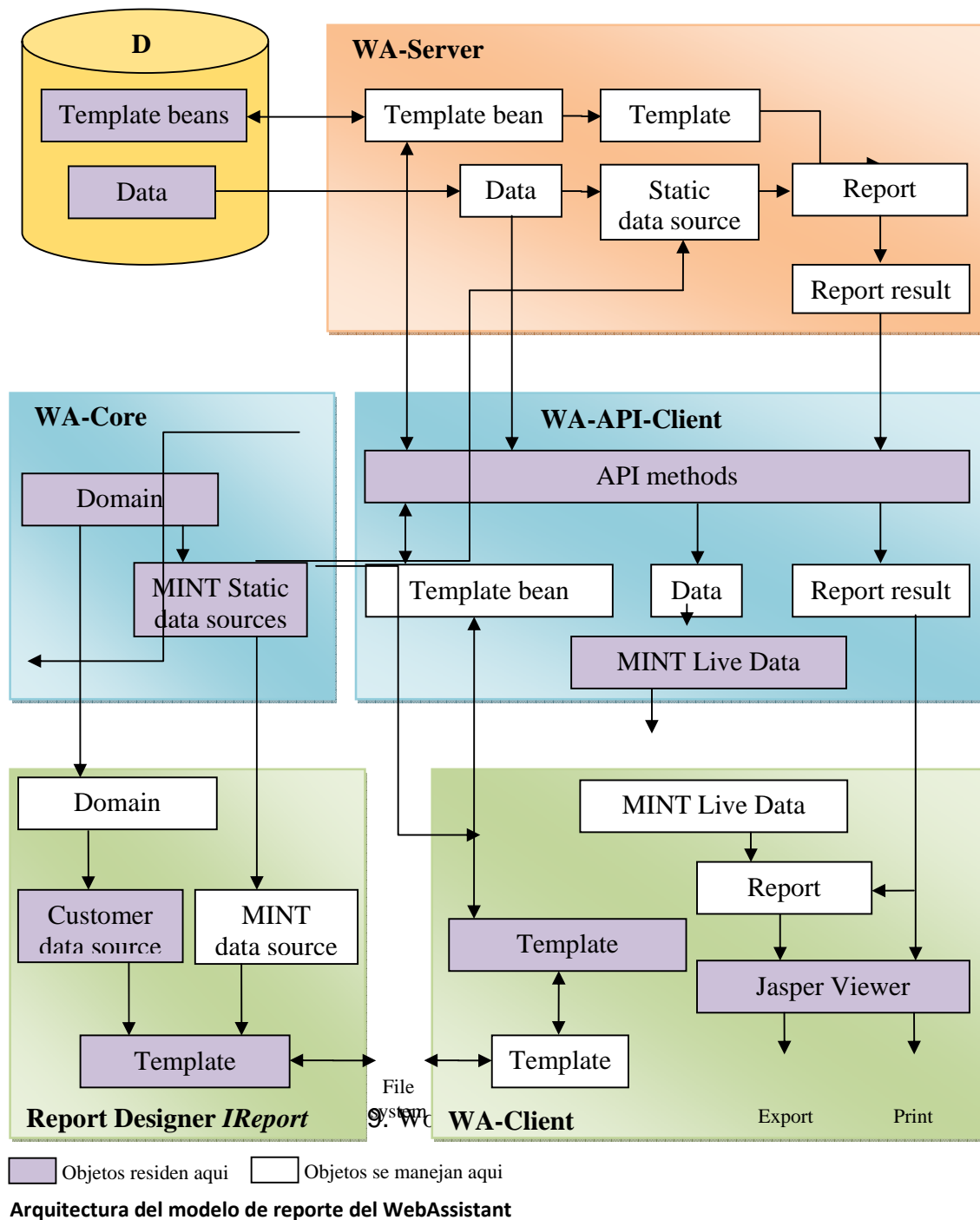


Figura 21. Arquitectura de reportes en MWA

Fuente: www.media-interactive.de

5. MANUAL DE USUARIO

A continuación se presenta un manual que sirve de guía al usuario, aquí se muestra que debe hacer el usuario para importar una plantilla creada en IReport para luego ejecutar dicho reporte.

Importar plantilla al MINT WEBASSISTANT

Lo primero es ir a la pagina del MINT WEBASSISTANT que tiene el repositorio de archivos, esta en la que tiene el árbol de archivos que a su vez es el árbol donde se almacenas las plantillas. Cuando se esta en esta pagina, se debe crear un nodo nuevo que servirá para guardar nuestra plantilla:

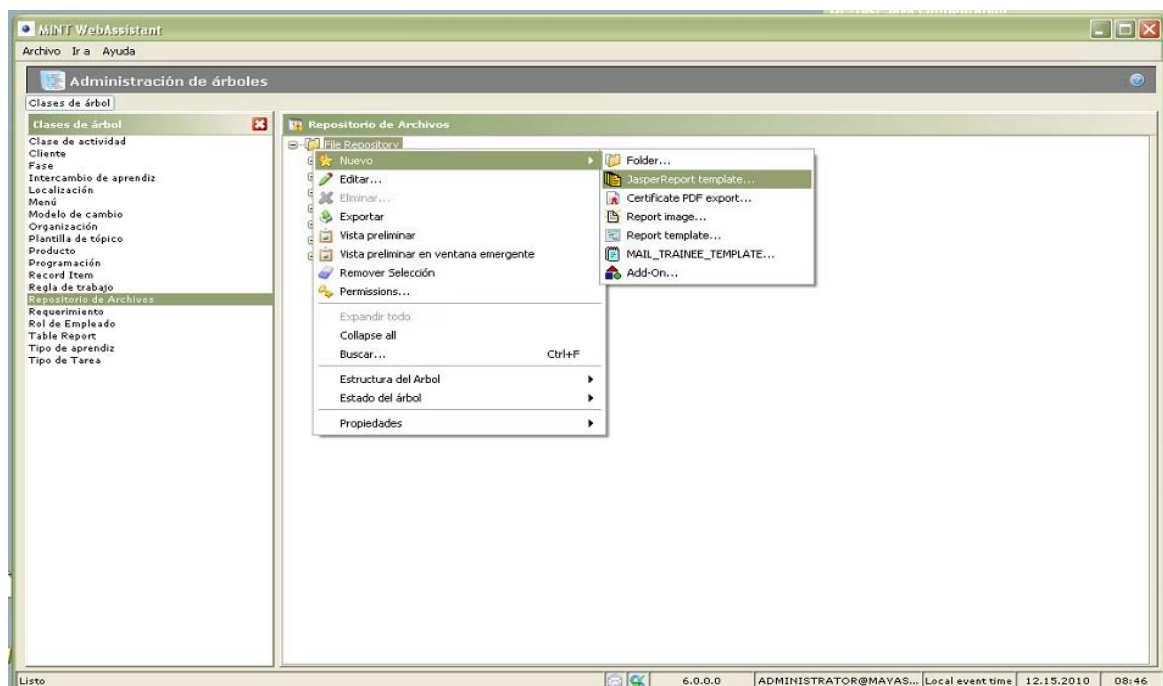


Figura 22. Crear nodo reporte

Seguidamente debemos ponerle un nombre a nuestra nueva plantilla de reportes, el tipo de la fuente de datos.

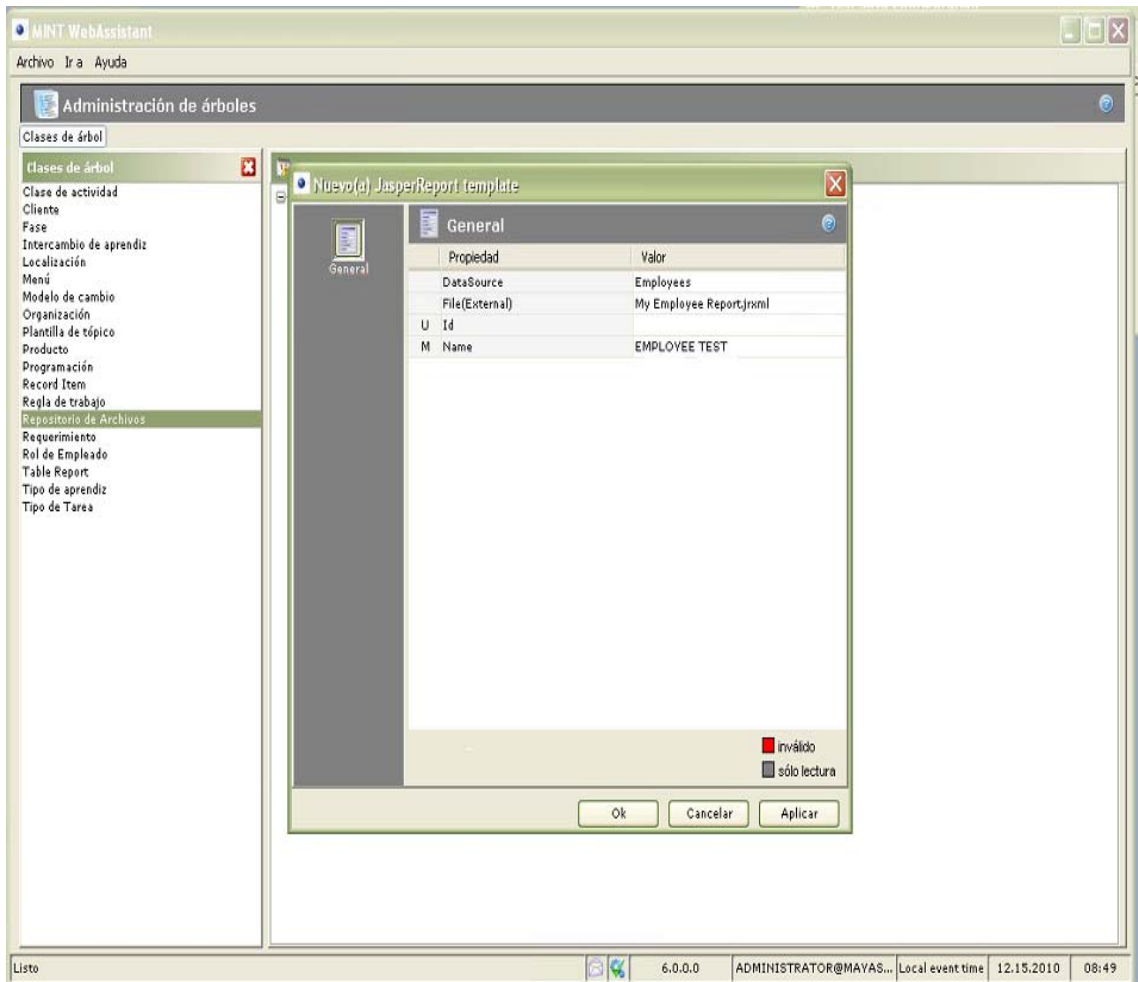


Figura 23. Llenado de datos

También debemos agregar la plantilla que creamos en anteriormente usando IReport. Esta la buscamos usando el explorador de archivos desplegado recordando que dichos archivos tienen una extensión jrxml:

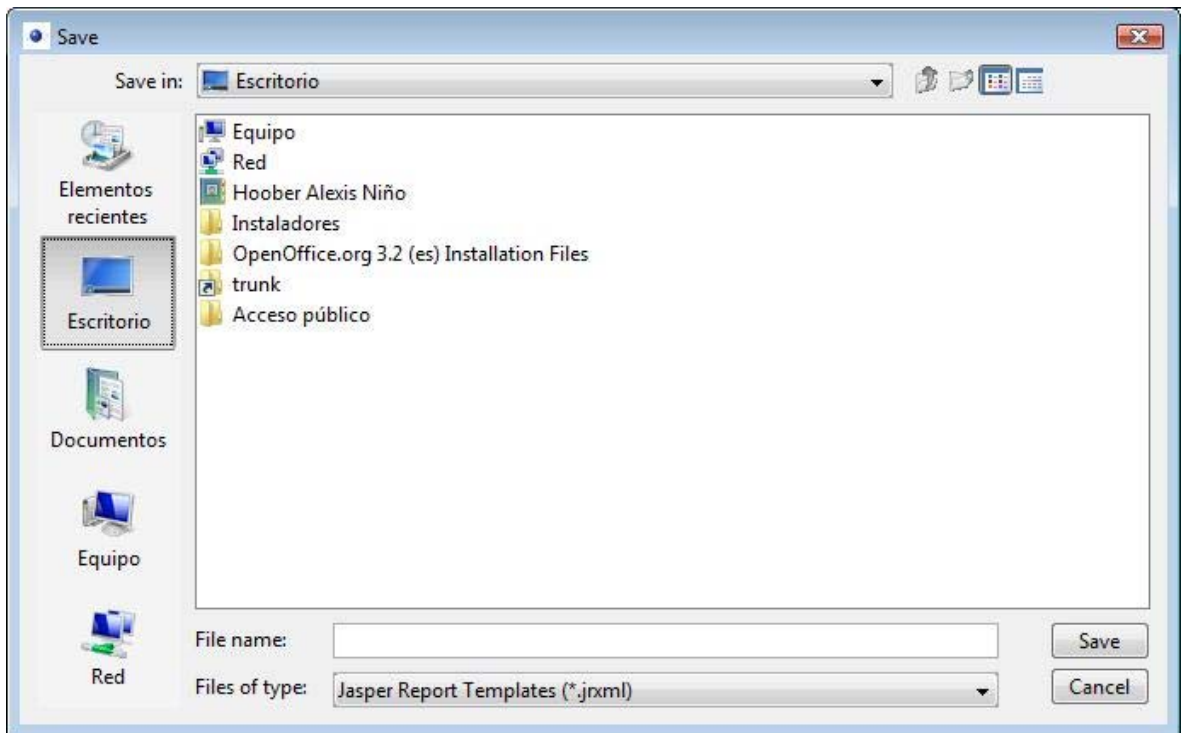


Figura 24. Búsqueda sistema archivos

Luego de tener los datos correctamente diligenciados, se aplican los cambios dentro del dialog de propiedades del nodo y se procede ahora a lo que tiene que ver con lo que tiene que ver con generar la vista previa de nuestro reporte.

Seleccionamos nuevamente el nodo que contiene el reporte y hacemos clic derecho para desplegar el menú contextual:



Figura 25. Menú contextual

Aquí se debe seleccionar la opción “Vista preliminar”. Nótese que también hay una opción llamada “Vista preliminar en ventana emergente”, esta funciona de la misma manera solo que presenta el dialogo en una ventana externa a la ventana normal del aplicativo.

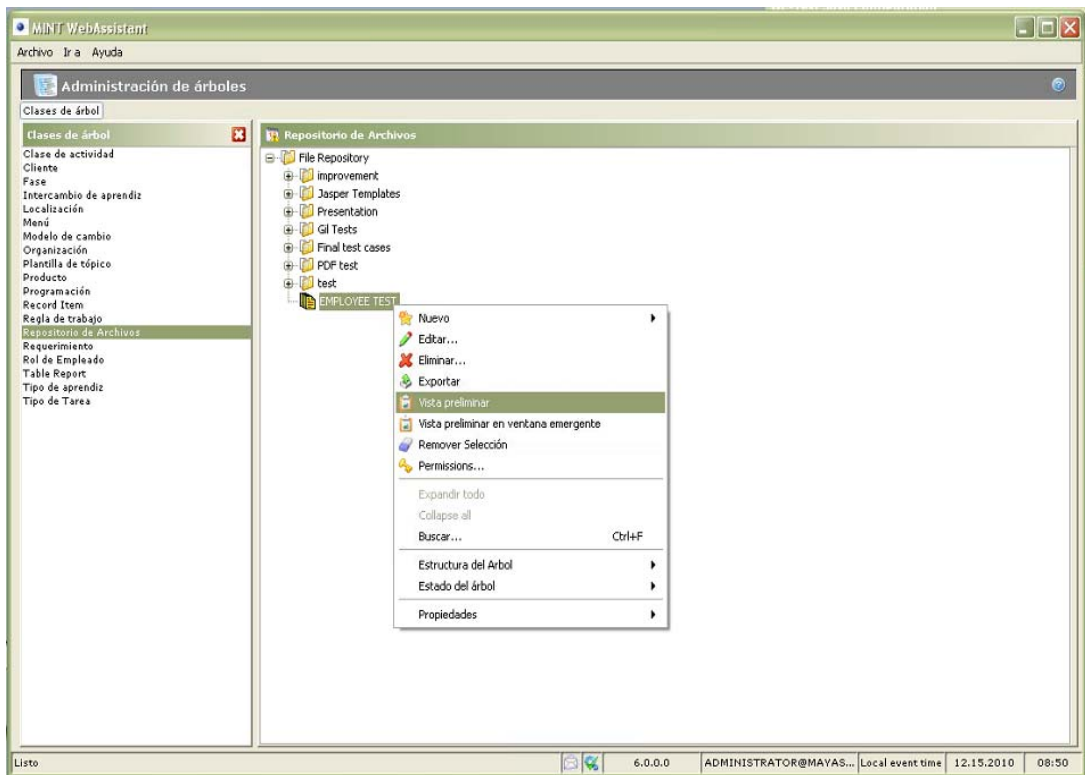


Figura 26. Opción vista preliminar

Al ejecutarse la opción de vista previa, se abre una nueva ventana con un wizard que permite darle los parámetros con los que va a ser ejecutado el reporte en caso de que este deba recibir parámetros.

En esta sección también se pueden ajustar los filtros que juegan un papel importante sobre que datos deben y cuales no deben ir dentro del resultado de ejecutar un reporte particular, con esto se garantiza mostrar diferente información usando una misma plantilla.

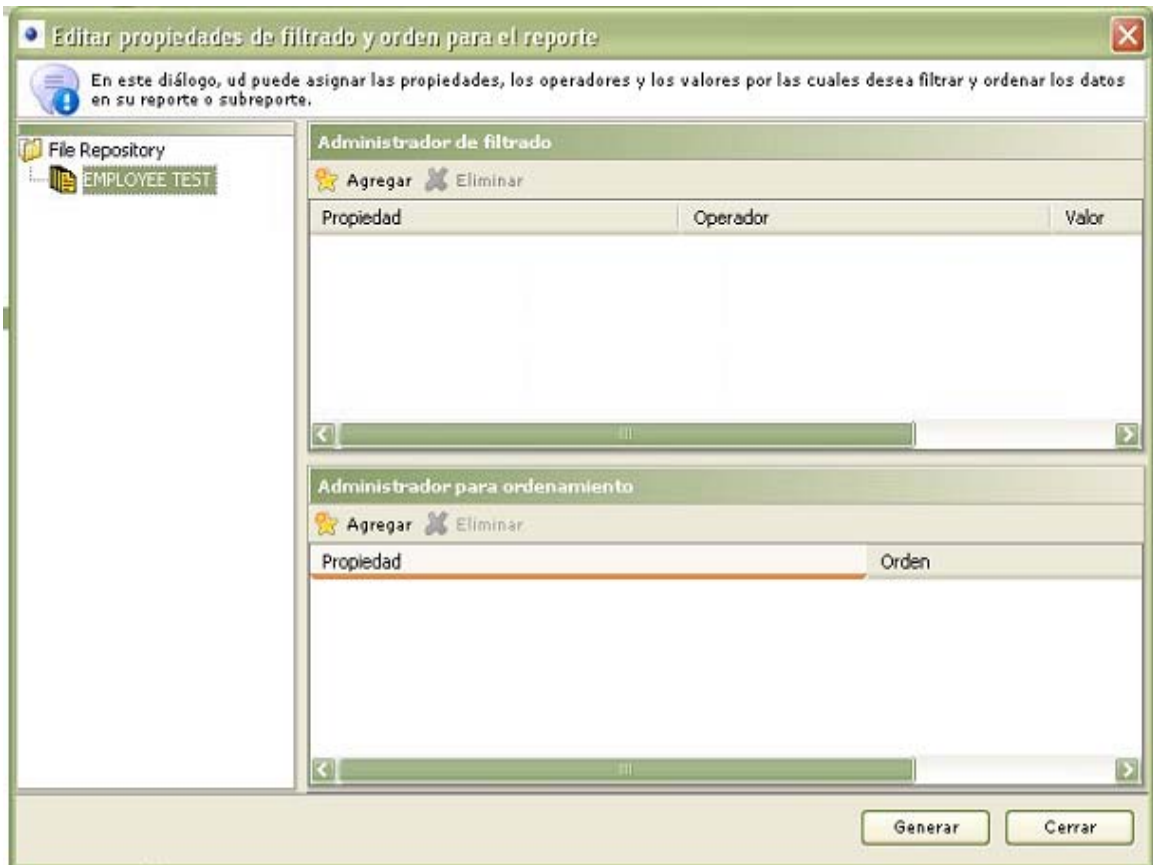
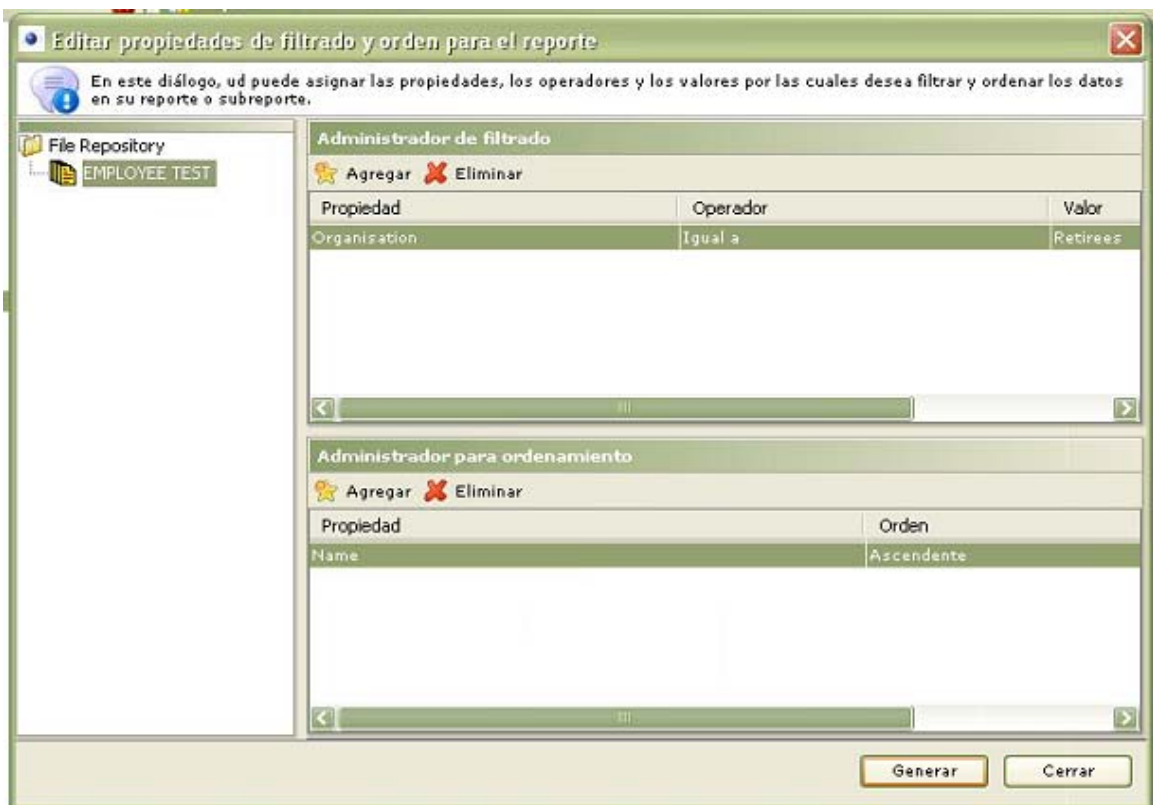
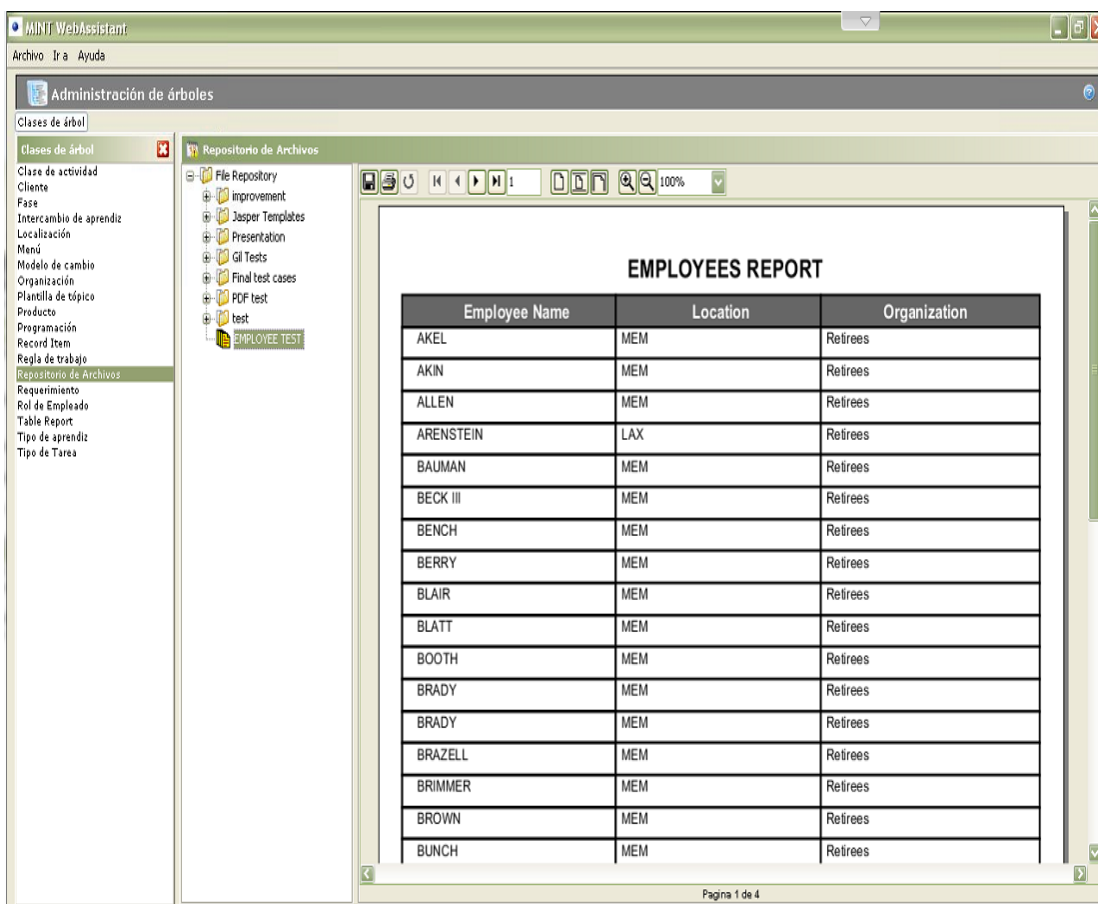


Figura 27. Dialogo de filtros 1 y 2



En la imagen anterior lo que se hizo fue crear el filtro donde los empleados que se van a mostrar pertenezcan a la organización RETIREES y van a estar ordenados de manera ascendente por el atributo "Name".

Luego de configurar al gusto los filtros y el orden que deseemos, hacemos clic sobre el botón generar para ejecutar el reporte:



The screenshot shows the MINT WebAssistant interface. On the left, there is a sidebar with 'Clases de árbol' and 'Repositorio de Archivos'. The main area displays a report titled 'EMPLOYEES REPORT'. The report is a table with three columns: 'Employee Name', 'Location', and 'Organization'. The data is as follows:

Employee Name	Location	Organization
AKEL	MEM	Retirees
AKIN	MEM	Retirees
ALLEN	MEM	Retirees
ARENSTEIN	LAX	Retirees
BAUMAN	MEM	Retirees
BECK III	MEM	Retirees
BENCH	MEM	Retirees
BERRY	MEM	Retirees
BLAIR	MEM	Retirees
BLATT	MEM	Retirees
BOOTH	MEM	Retirees
BRADY	MEM	Retirees
BRADY	MEM	Retirees
BRAZELL	MEM	Retirees
BRIMMER	MEM	Retirees
BROWN	MEM	Retirees
BUNCH	MEM	Retirees

At the bottom of the window, it says 'Página 1 de 4'.

Figura 28. Resultado reporte

Eliminar un reporte del árbol de archivos.

Para esto simplemente hay que seleccionar el nodo correspondiente a un reporte y presionar clic derecho para abrir el menú contextual. Una vez ahí escogemos la opción “Eliminar”.

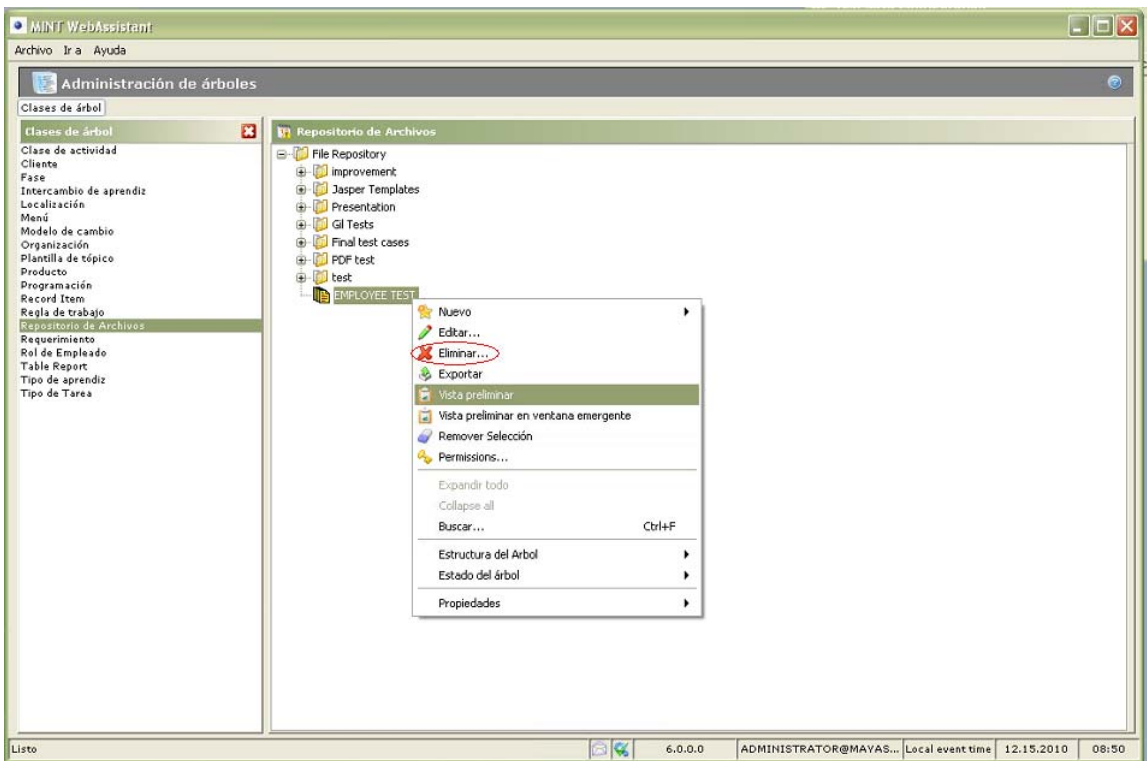


Figura 29. Eliminar plantilla

Se le pedirá confirmación al usuario para realizar esta acción.

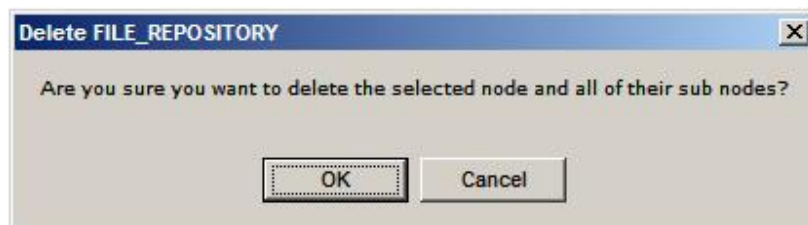


Figura 30. Dialogo confirmación

Editar información del reporte

Para editar la información asociada a un reporte se tiene que seleccionar el nodo contenedor y presionar clic derecho para desplegar el menú contextual. Allí se selecciona la opción “Editar”.

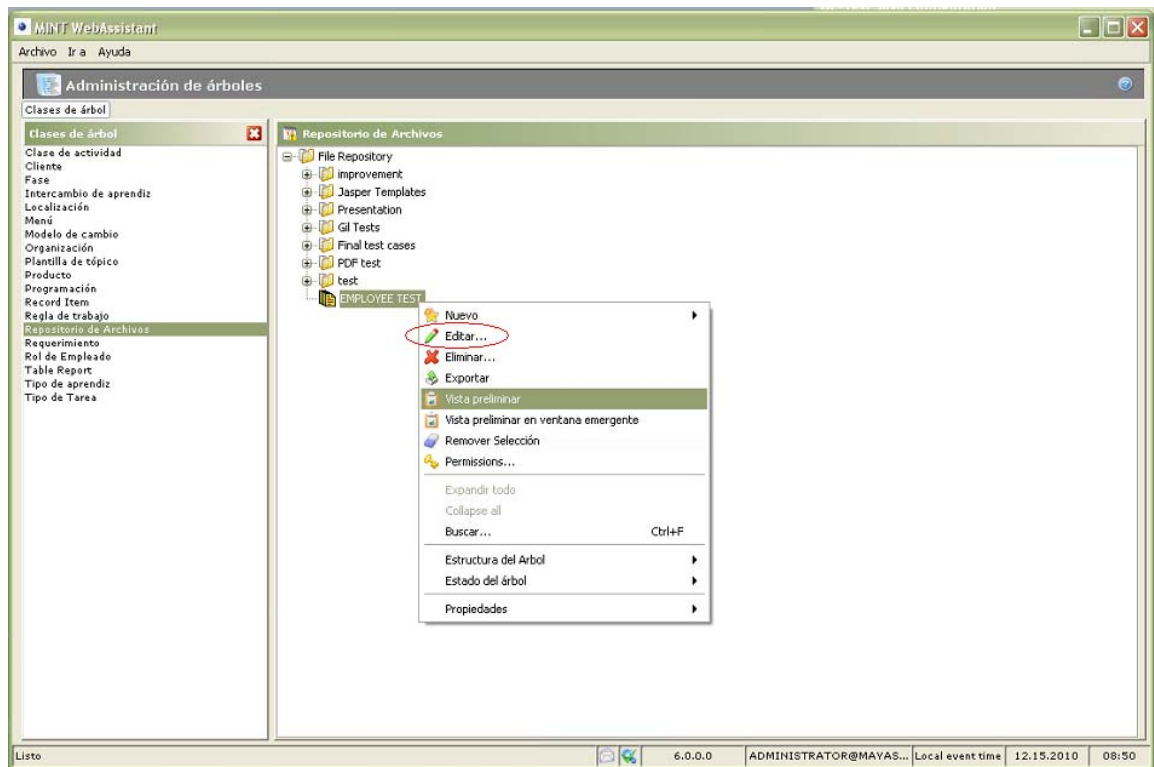


Figura 31. Editar reporte

Al ejecutar la opción de editar el reporte se desplegará el diálogo de propiedades donde se pueden modificar todas las opciones relacionadas con el reporte seleccionado, se modifican y seguidamente se aplican los cambios.

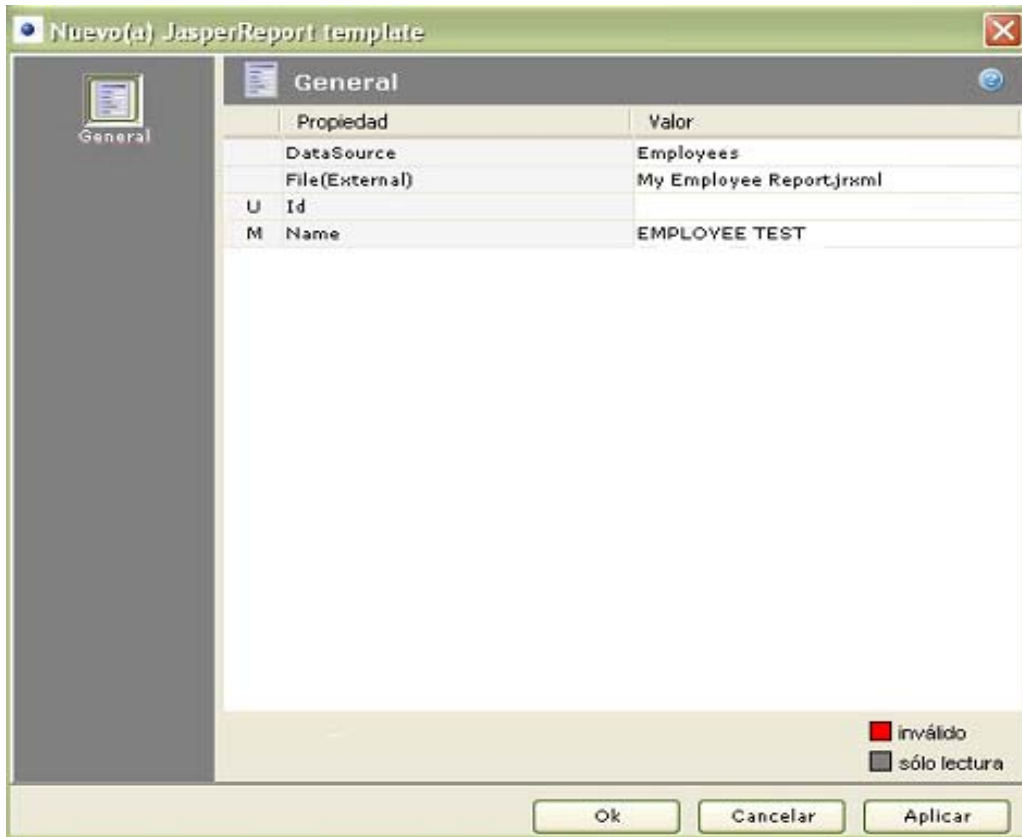


Figura 32. Dialogo de propiedades del reporte

Guardar resultado del reporte en un formato determinado

Para guardar el resultado de un reporte en un formato determinado, solo es necesario haber ejecutado la vista previa del resultado del reporte dentro del MINT WebAssistant y nos ubicamos sobre la barra de herramientas en la parte superior de la ventana del reporte.

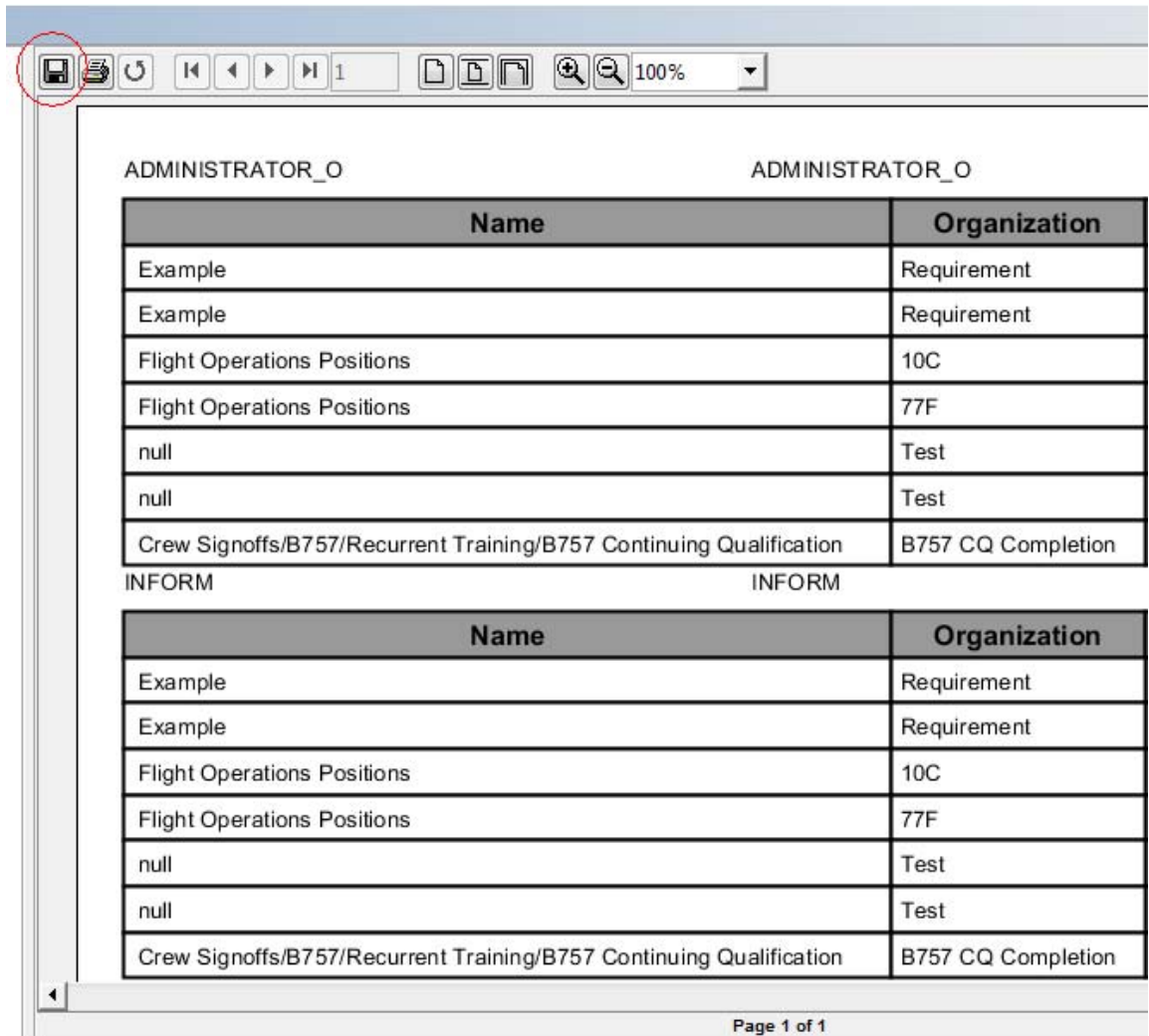


Figura 33. Guardar resultado reporte

Luego de hacer esto, se abrirá un dialogo en el cual podremos seleccionar la ubicación en la que vamos a guardar el resultado del reporte y el formato con el que lo vamos a hacer.

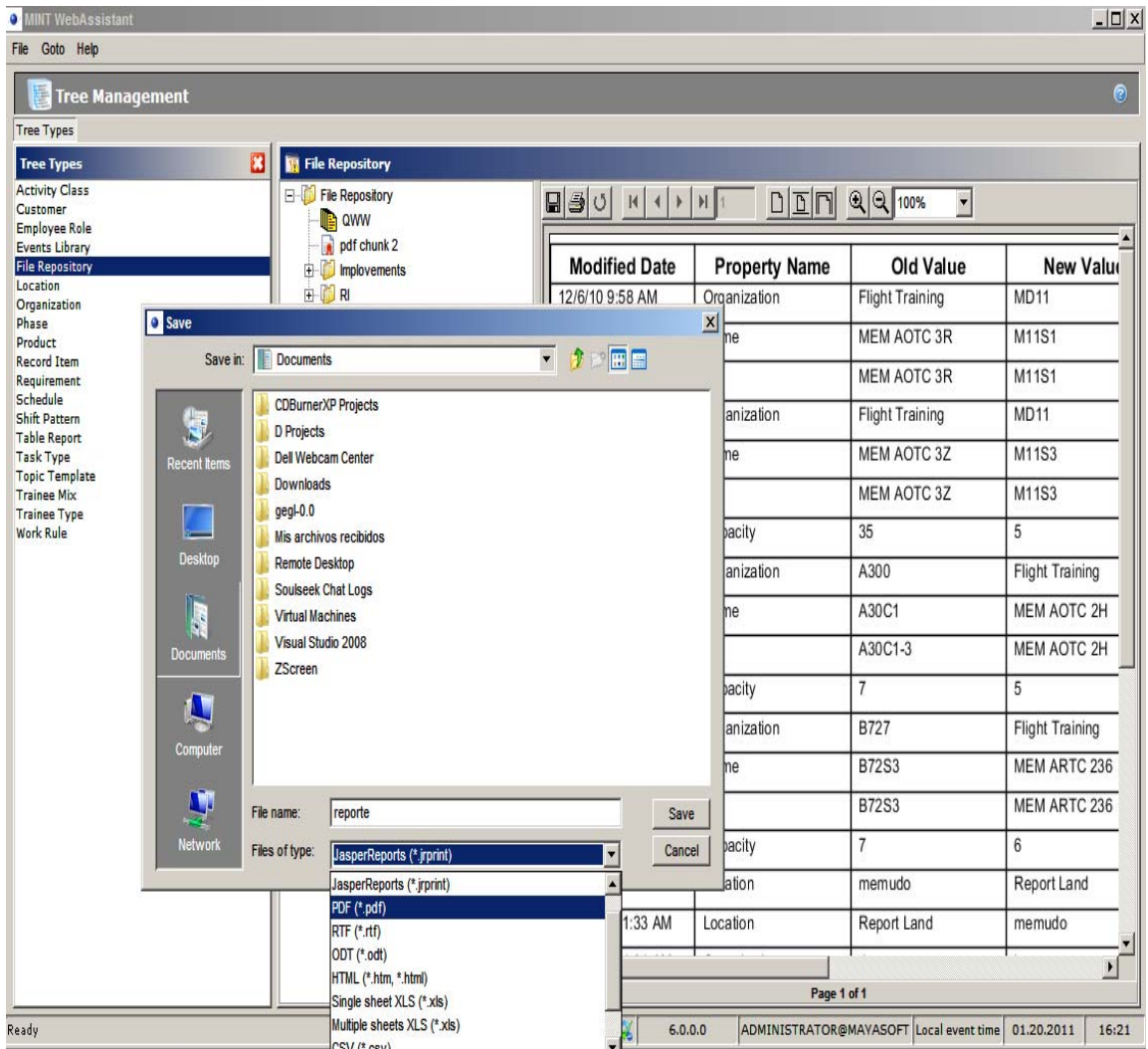


Figura 34. Escoger formato

6. PRUEBAS REALIZADAS

Al desarrollar cada uno de los servicios mencionados en los objetivos, es necesaria la realización de las pertinentes pruebas que permitan garantizar que cada uno de los subsistemas desarrollados funcione correctamente, sin errores tanto de diseño como de programación; para lo cual se realizan pruebas de verificación y de validación. A continuación se listan y describen los resultados obtenidos en cada uno de los tipos de prueba.

6.1 PRUEBAS DE VERIFICACIÓN

Estas pruebas se realizan para comprobar que lo implementado esté acorde a lo que se plantea inicialmente y que presentan el funcionamiento que se espera.

Estas pruebas se realizan obteniendo información de las bases de datos y comparándola con las actividades realizadas utilizando los respectivos servicios desarrollados, se verifica que los datos necesarios (obligatorios) no estén vacíos, las respectivas validaciones que dependen del tipo de dato que se esté utilizando y la estabilidad del sitio al ocurrir algún evento no esperado.

Accion	Prueba Realizada	Resultado
Conectar IReport a las fuentes de datos creadas dentro del Mint WebAssistant	Dentro de IReport verificar que se pueda lograr la conexión con dichas clases	✓
Listar las propiedades creadas para los recursos asociados a una fuente de datos a	Verificar qque se puedan obtener las propiedades creadas en el MINT WebAssistant dentro de IReport	✓

travez de IReport		
Ejecutar un reporte desde IReport usando una fuente de datos creada en el MINT WebAssistant	Crear una plantilla con una fuente de datos propia del MWA, ejecutarla y que se vean los resultados sin errores en IReport	✓
Importar una plantilla al MINT WebAssistant	Crear una plantilla en IReport y luego desde el MINT WebAssistant importarla a la base de datos	✓
Editar datos de una plantilla importada dentro del MINT WebAssistant	Verificar que al editar los cambios sobre una plantilla ya guardada, estos se actualicen con los nuevos valores en la base de datos	✓
Ejecutar un reporte para visualizar su vista previa	Verificar los datos mostrados al ejecutar un reporte importado dentro del MINT WebAssistant	✓
Ejecutar un reporte pero seteandole previamente un criterio de ordenamiento	Verificar que los datos mostrados luego de la ejecución estén ordenados de manera correcta	✓
Ejecutar un reporte utilizando propiedades para realizar el filtrado de	Los datos mostrados en el resultado estarán definidos por el filtro anteriormente dado.	✓

datos		
Imprimir el resultado de un reporte usando el botón de impresión del panel de vista previa	Se debe imprimir el reporte	✓
Guardar el resultado del reporte en PDF en el sistema de archivos	Se confirma que se cree el archivo con el resultado del reporte en el formato especificado.	✓
Guardar el resultado del reporte en excel en el sistema de archivos	Se confirma que se cree el archivo con el resultado del reporte en el formato especificado.	✓
Guardar el resultado del reporte en PDF en el sistema de archivos	Se confirma que se cree el archivo con el resultado del reporte en el formato especificado.	✓
Guardar el resultado del reporte en Word en el sistema de archivos	Se confirma que se cree el archivo con el resultado del reporte en el formato especificado.	✓
Guardar el resultado del reporte en ODT en el sistema de archivos	Se confirma que se cree el archivo con el resultado del reporte en el formato especificado.	✓
Guardar el resultado del reporte en HTML en el sistema de archivos	Se confirma que se cree el archivo con el resultado del reporte en el formato especificado.	✓
Guardar el resultado	Se confirma que se cree el archivo con el	✓

del reporte en RTF en el sistema de archivos	resultado del reporte en el formato especificado.	
Eliminar una plantilla previamente importada a la base de datos	Verificar que el nodo del árbol que contiene el reporte desaparezca y que en el base de datos no esté el registro de la tabla de plantillas	•
Ejecutar un reporte con diferentes parámetros	Verificar que dichos resultados sean diferentes y que estén acordes a los filtros asignados	•

7. CONCLUSIONES

- El modelo del ciclo de vida escogido en el desarrollo de un proyecto es un factor principal para lograr los objetivos propuestos. Para la realización de este proyecto se consideró necesario hacer uso del modelo de prototipado evolutivo debido a que en éste se comienza diseñando y construyendo las partes más importantes de la aplicación en un prototipo que posteriormente se enriquece y amplía hasta obtener el prototipo final, que es el software de entrega final.
- Java es un lenguaje de programación que permite realizar cualquier tipo de programa, es independiente de la plataforma, permitiendo así, que un programa en Java funcione en cualquier computadora, ya que cuenta con una Máquina de Java para cada sistema que hace de puente entre el sistema operativo y el programa de Java posibilitando que este último se entienda perfectamente.
- El desarrollo de este proyecto, contribuye a mejorar el manejo de información de los recursos existentes. Los usuarios tienen la facilidad de disponer de la información disponible en la manera en que les perezca conveniente
- Para el modulo de reportes fue importante capacitar a alguien para que quedara encargado de su mantenimiento y posibles nuevas implementaciones, ya que como cualquier software esta expuesto a

fallos, los cuales deben ser resueltos en lo posible de manera inmediata.

8. BIBLIOGRAFÍA

TEXTOS BIBLIOGRAFICOS

BRUEGGE BY DUTOIT A. Ingeniería de Software Orientado a Objetos. Prentice Hall. 2002. En este libro se encuentra información sobre las metodologías de desarrollo software

PRESSMAN, R. Ingeniería de Software, Un enfoque práctico. Quinta Edición. McGraw-Hill. 2002. En este libro se encuentra información sobre las metodologías de desarrollo software.

DOCUMENTACION DISPONIBLE EN INTERNET

<http://www.desarrolloweb.com>

Este sitio tiene un tutorial amplio sobre JavaScript con ejemplos y CSS.

<http://www.geocities.com/SiliconValley/Hardware/6503/ReglasNegocio.htm>

Arquitectura cliente servidor de dos y tres capas

<http://es.kioskea.net/cs/cs3tier.php3>. En este sitio se encuentra teoría sobre el modelo Cliente - Servidor.

<http://mysql.conclase.net/curso/index.php>. Sitio web que contiene el instructivo sobre la instalación, configuración, y conceptos básicos de bases de datos.

http://www.wikilearning.com/tutorial/tutorial_de_java/3938. Tutorial básico de Java.