

Introducción a JasperReports

JasperReports es la mejor herramienta de código libre en Java para generar reportes. Puede entregar ricas presentaciones o diseños en la pantalla, para la impresora o para archivos en formato PDF, HTML, RTF, XLS, CSV y XML.

Está completamente escrita en Java y se puede utilizar en una gran variedad de aplicaciones de Java, incluyendo J2EE o aplicaciones Web, para generar contenido dinámico.

Requerimientos de JasperReports

- Se requiere **tener instalado** en el equipo el JDK 1.4 (SDK) o posterior. No basta con tener instalado el J2RE (Run Time Environment).
- Las siguientes librerías junto con la de JasperReports deben incluirse en el proyecto en que se desee incluir esta herramienta para generar reportes.

Jakarta Commons Digester Component (versión 1.1 o posterior)
<http://jakarta.apache.org/commons/digester/>
commons-digester.jar

Jakarta Commons BeanUtils Component (versión 1.1 o posterior)
<http://jakarta.apache.org/commons/beanutils/>
commons-beanutils.jar

Jakarta Commons Collections Component (versión 1.0 o posterior)
<http://jakarta.apache.org/commons/collections/>
commons-collections.jar

Jakarta Commons Logging Component (versión 1.0 o posterior)
<http://jakarta.apache.org/commons/logging/>
commons-logging.jar

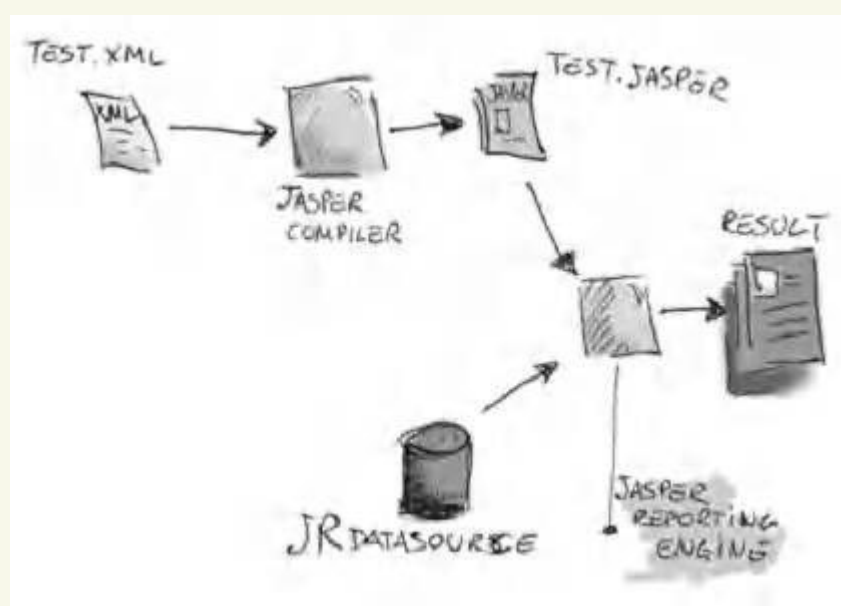
- Driver JDBC 2.0 (Usualmente incluido en el SDK)
- PDF. Librería libre Java-PDF iText por Bruno Lowagie y Paulo Soares

(versión 1.01 o posterior)
<http://www.lowagie.com/iText/>
itext-1.02b.jar

- XLS

Funcionamiento de JasperReports

JasperReports trabaja en forma similar a un compilador y a un intérprete, ver figura 1. El usuario diseña el reporte codificándolo en XML de acuerdo a las etiquetas y atributos definidos en un archivo llamado `jasperreports.dtd` (parte de JasperReports). Usando XML el usuario define completamente el reporte, describiendo donde colocar texto, imágenes, líneas, rectángulos, cómo adquirir los datos, como realizar ciertos cálculos para mostrar totales, etc.

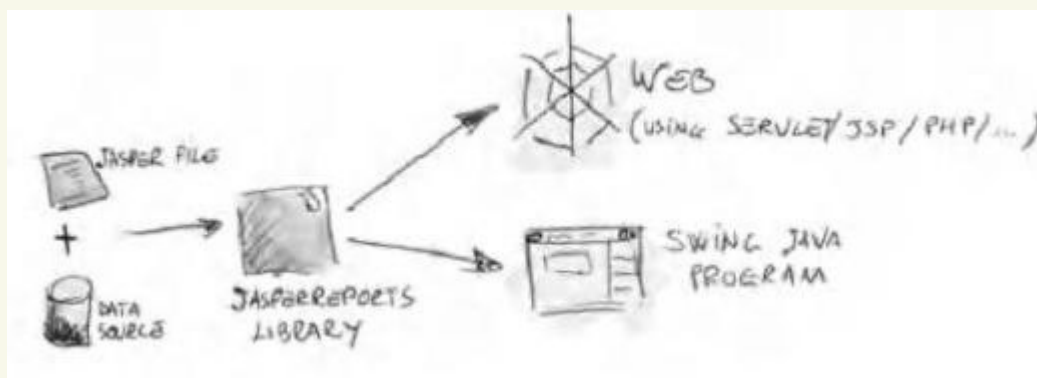


Este archivo fuente XML debe ser compilado para obtener un reporte real. La versión compilada del fuente es nombrada "archivo jasper" (este termina con `.jasper`). Un Archivo jasper es el compilado de un código fuente. Cuando tenemos un archivo jasper, necesitamos otra cosa para producir un reporte: necesitamos datos. Esto no siempre es cierto. En algunos casos querríamos generar un reporte que no mostrara datos dinámicos, solo texto estático por ejemplo, pero esto puede simplificarse a un reporte que tiene solamente un registro vacío. Para proporcionar estos registros al "jasper engine" necesitamos presentarlos usando una interfaz especial específica llamada `JRDataSource`. Una fuente de datos + un Archivo jasper = un "archivo print". Un "archivo print" puede exportarse en muchos formatos como PDF, HTML, RTF, XML, XLS, CVS, etc. La exportación se puede realizar utilizando clases especiales para implementar exportadores específicos.

Compilación, exportación de reportes de JasperReports

Para un novato, diseñar y crear el archivo jasper es la tarea mas dura. Cuando se haya diseñado y compilado el archivo jasper, se puede utilizar la librería JasperReports para llenar dinámicamente el reporte en varios entornos como una aplicación web (Usando un servlet de Java por ejemplo, pero también funciona para generar reportes PDF desde un script PHP).

Jasper tiene disponible un visualizador especial para desplegar la vista previa de un reporte; diseñado para aplicaciones tradicionales de Java basadas en Swing.



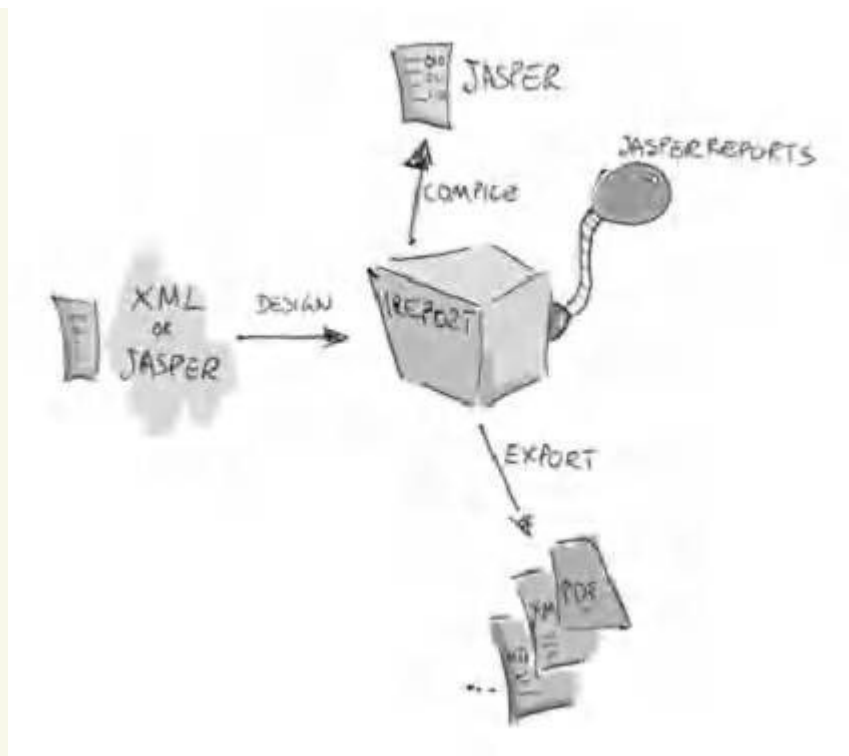
iReport

iReport es un diseñador visual de código libre para JasperReports escrito en Java. Es un programa que ayuda a los usuarios y desarrolladores que usan la librería JasperReports para diseñar reportes visualmente. A través de una interfaz rica y simple de usar, iReport provee las funciones más importantes para crear reportes amenos en poco tiempo.

iReport puede ayudar a la gente que no conoce la sintaxis XML para generar reportes de JasperReports.

Funcionamiento de iReport

iReport provee a los usuarios de JasperReports una interfaz visual para construir reportes, generar archivos "jasper" y "print" de prueba. iReport nació como una herramienta de desarrollo, pero puede utilizarse como una herramienta de oficina para adquirir datos almacenados en una base de datos, sin pasar a través de alguna otra aplicación.



iReport puede leer y modificar ambos tipos de archivo, XML y Jasper. A través de JasperReports, es capaz de compilar XML a archivos Jasper y “ejecutar reportes” para llenarlos usando varios tipos de fuentes de datos (JRDataSource) y exportar el resultado a PDF, HTML, XLS, CSV,...

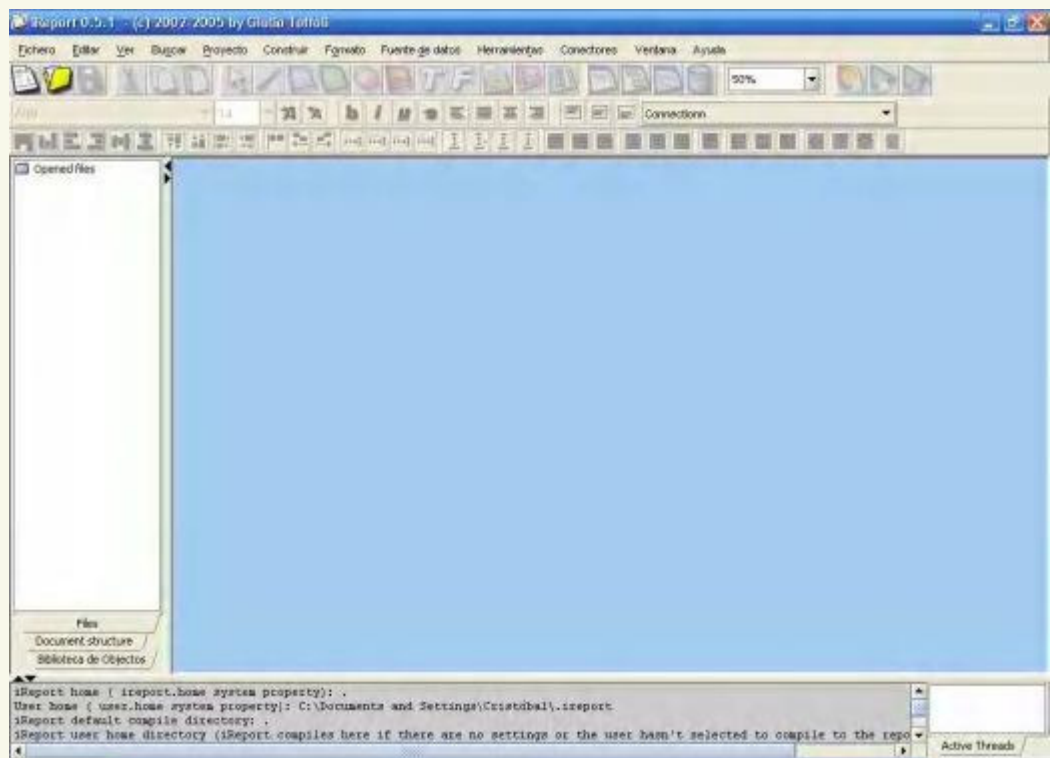
Requerimientos de instalación (Windows 2000, NT, XP)

- Sun JDK 1.4 (SDK) o superior.
- Acrobat 5.0 no es requerido, pero es fuertemente recomendado.
- Si se desea conectar con una base de datos, se debe proporcionar el Driver JDBC correspondiente.

Instalación y configuración ((Windows 2000, NT, XP))

Si tiene instalado en su equipo un JDK (SDK) y no simplemente un JRE, está listo para iniciar la instalación.

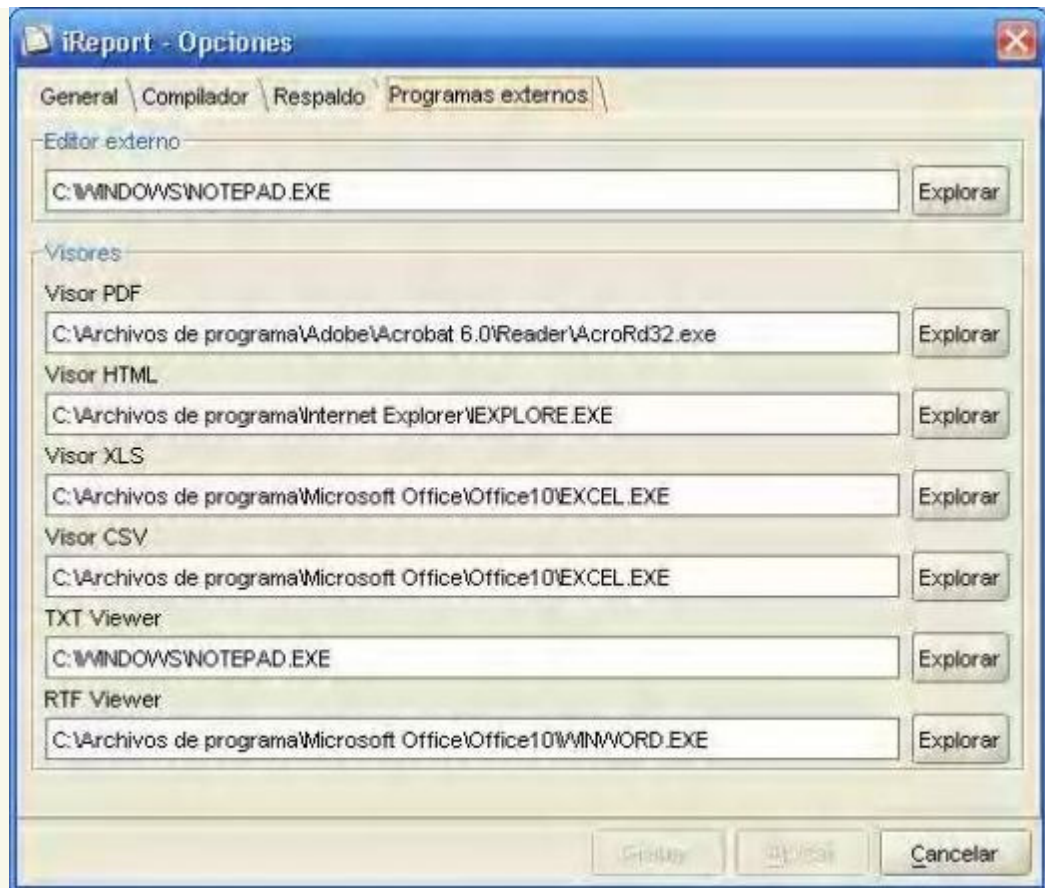
1. Descomprima iReport-x.x.x.zip y copie el directorio extraído al lugar que desee (C:\iReport-x.x.x\ por ejemplo)
2. Busque un archivo llamado tools.jar en su JDK y cópielo en el directorio lib de iReport. (C:\iReport-x.x.x\lib)
3. Ejecute iReport.bat o iReport.sh.



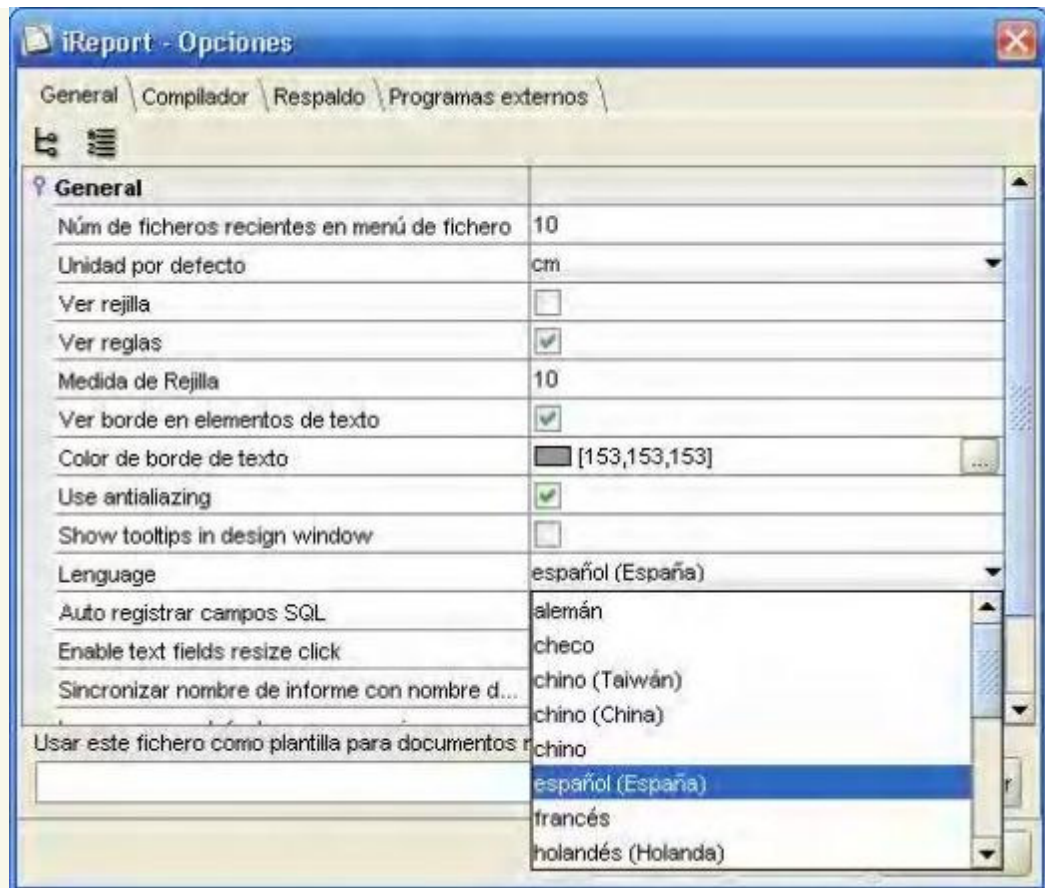
Después de ejecutarse por primera vez, iReport creará un directorio (.ireport) en su directorio principal (home). Aquí se almacenarán todos los archivos de configuración en formato XML.

Proceda a configurar iReport con los siguientes pasos:

1. Vaya a menu->tools->options.
2. Vaya a la pestaña de Programas externos (external programs).
3. Configure los programas visualizadores externos.



4. En la pestaña General puede configurar el idioma



Pruebe si la configuración fue correcta. Cree un nuevo reporte en blanco, haga clic en el botón ejecutar sin conexión (run without connection).



Después de unos segundos aparecerá el reporte con el programa que se haya seleccionado para visualizarse.

¿Qué necesito descargar?

Este es el resumen de las librerías que debe descargar:

- JasperReports
 - **jasperreports-1.0.1.jar**
 - commons-digester.jar
 - commons-beanutils.jar
 - commons-collections.jar

- commons-logging.jar
- itext-1.02b.jar
- poi-2.0-final-20040126.jar
- iReport
 - **iReport-0.5.1 (versión 0.5.1)**

Recuerde que es necesario tener instalado el Jdk 1.4 o superior, no basta con tener instalado el J2RE.

NOTA:

En el directorio /lib de iReport se encuentran versiones de las librerías de jasperreports y librerías adicionales para JasperReports (commons-digester.jar, commons-beanutils.jar, ...). **Es muy importante asegurarse que la librería de jasper utilizada tanto en iReport como en su proyecto, sea la misma, de lo contrario obtendrá errores al intentar llenar dinámicamente su reporte desde otra aplicación Java. Para cualquier otra versión de iReport que descargue, debe verificar la versión máxima de JasperReports soportada si es que considera descargarla por separado.**

Los sitios desde donde puede descargar JasperReports e iReport son los siguientes

<http://jasperreports.sourceforge.net/>

<http://ireport.sourceforge.net/index.php>)

En ambos sitios tiene la opción de descargar la distribución binaria, o bien el código fuente de los proyectos. Es recomendable obtener el fuente de JasperReports, en este se incluye toda la documentación y API's, además de código y ejemplos para realizar reportes. Debes obtener la herramienta ANT si deseas modificar y recompilar los fuentes de los proyectos.

¿Como agregar la librería de JasperReports y las librerías adicionales al proyecto?

Dependiendo del entorno de desarrollo que utilice (NetBeans, JBuilder, Eclipse...), puede agregar estas librerías a su IDE y posteriormente a su proyecto. Acerca del directorio en que deberían colocarse, algunos prefieren colocar las librerías en el directorio \jre\lib\ext\ de su SDK. Personalmente trabajando en Windows prefiero colocar las librerías en un directorio del proyecto, como el \lib. Lo más importante es indicar al compilador el lugar en que estas se localizan.

En JBuilder, las librerías se pueden agregar accediendo al menú tools -> Configure -> Libraries. Seleccionando las carpetas Project o User se oprime el botón New, se pone el nombre de la librería

y se especifica el directorio en que se encuentran. Para agregarlas al proyecto, se va al menú Project->Project Properties y en la pestaña Required libraries se selecciona la de JasperReports que acaba de agregar y listo.

Si la compilación se realiza manualmente, deberá utilizar -classpath en la línea de comando para especificar la ubicación de las librerías. Un ejemplo de línea de comando es el siguiente:

```
-classpath "C:\JasperReports\jasperreports-1.0.1.jar;C:\JasperReports\commons-digester.jar; ... "
```

Recuerde que debe incluir no solo la librería JasperReports; también sus librerías adicionales, de lo contrario podría obtener errores al compilar su proyecto. La librería iReport.jar en el directorio \lib de su iReport, debería agregarse al proyecto solo si diseña el reporte con iReport y pretende compilarlo desde su aplicación Java. En nuestro caso, se diseñará y compilará el reporte con iReport para generar un archivo .jasper, por lo que no es necesario agregar iReport.jar al proyecto.

En la segunda parte del artículo, se mostrará paso a paso la manera de diseñar un reporte con iReport. De la compilación de este reporte, se obtendrá un archivo *.jasper, el cual será llenado y mostrado dinámicamente desde una aplicación Java Swing.

Introducción a JasperReports e iReport (Segunda parte)

En esta segunda parte del artículo se mostrará paso a paso la manera de diseñar, compilar y visualizar un reporte con iReport. Posteriormente se mostrará la manera en que puede ser llenado y mostrado dinámicamente desde una aplicación Java.

Configuración de la conexión a una base de datos.

Para establecer una conexión entre iReport y una base de datos, se debe proporcionar el driver JDBC correspondiente. La versión 0.5.1 de iReport ya proporciona drivers JDBC para establecer conexiones con bases de datos como MySQL y Access. Para nuestro ejemplo se usará una conexión con una base de datos Access, para la cual se ha configurado un origen de datos con nombre DSN. En el artículo “El puente JDBC-ODBC(Ejemplo de conexión entre Access y Java)” se explica detalladamente la manera de configurar un origen de datos.

Suponiendo que se ha configurado un origen de datos nombrado para una base de datos Access con los siguientes valores:

Nombre de driver: PDRV

Nombre de inicio de sesión: cvazquez

Contraseña: vazquez

Procedemos a configurar iReport para establecer la conexión con la base de datos, para ello debe ir a menú->Fuente de datos->Conexiones/Fuente de datos. En la pantalla Connections/Datasources oprima el botón new para agregar una conexión.

La pantalla de conexión debe llenarse tal como se muestra a continuación:



Connections properties

Type of connection / datasource
Database JDBC connection

Name: Conexion

JDBC Driver: sun.jdbc.odbc.JdbcOdbcDriver

JDBC URL: jdbc:odbc:PDRV

JDBC URL Wizard

Server Address:

Database: Wizard

Username: cvazquez

Password: ☒ Save password

ATTENTION! Passwords are stored in clear text. If you dont specify a password now, iReport will ask you for one only when required and will not save it.

Save Cancel Test

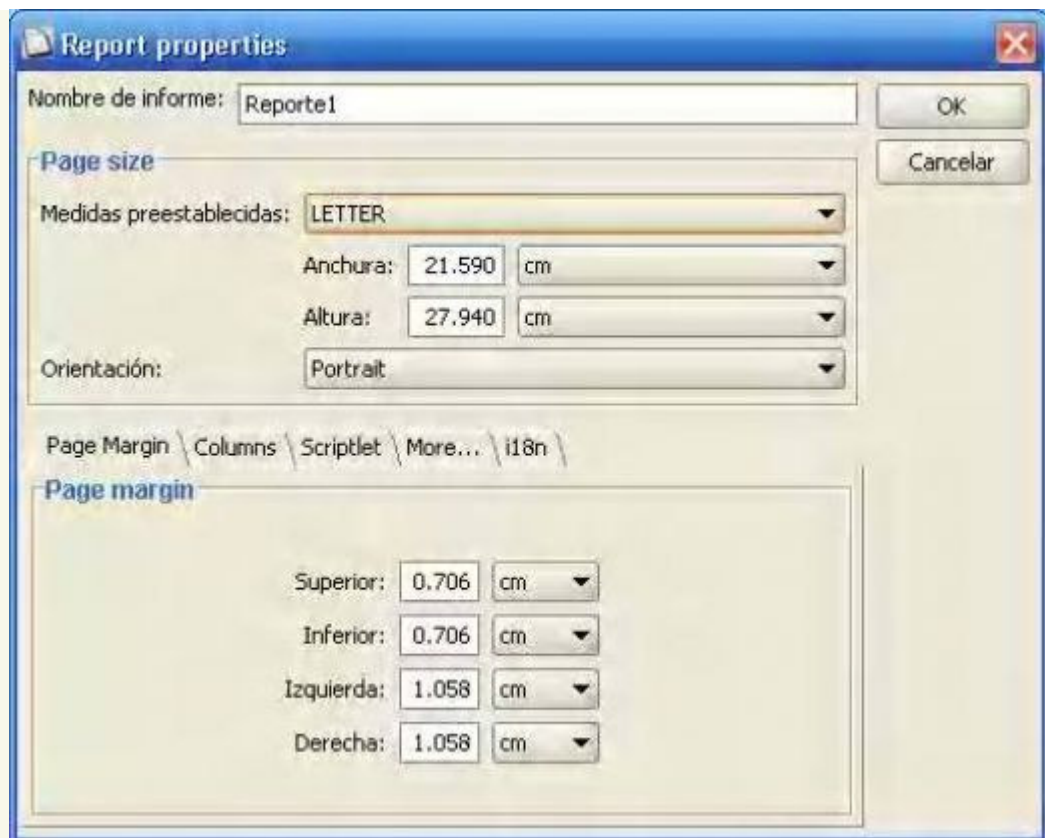
A continuación oprima el botón Test para probar la conexión. Si la conexión fue exitosa, oprima finalmente el botón Save para guardar esta conexión.



Creación del Reporte.

En iReport, se tiene la opción para trabajar por proyecto, el cual puede contener varios reportes, en nuestro caso no se creará un proyecto, se creará solo un reporte de la siguiente manera:

Seleccione nuevo documento del menú Fichero o bien oprima el botón new report de la barra de herramientas, aparecerá una pantalla de propiedades del nuevo reporte que queremos crear:



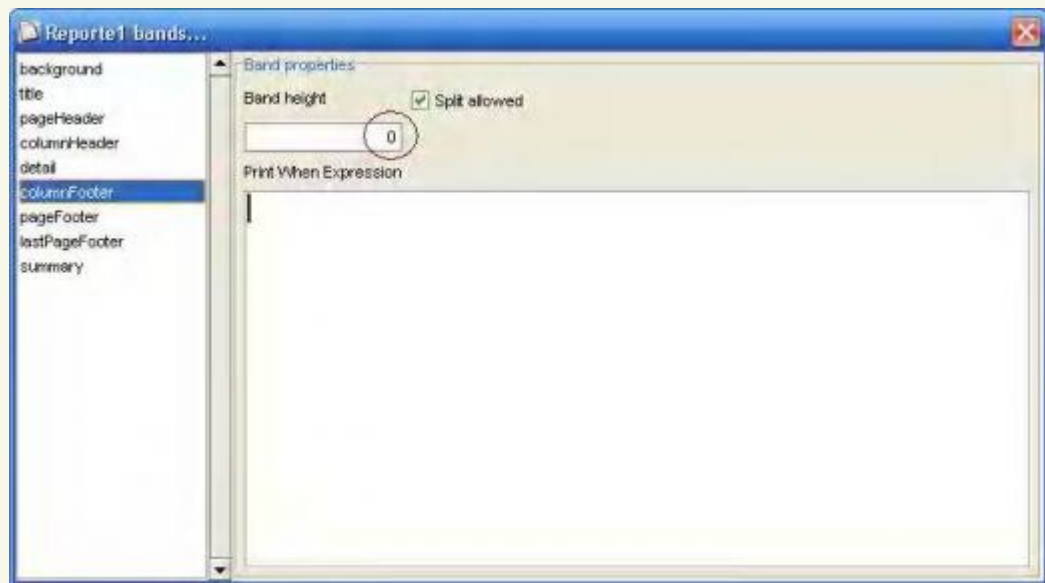
En esta pantalla podemos configurar las propiedades del reporte, en nuestro caso, le llamaremos Reporte1, oprimir el botón OK para crearlo.

Seleccionar la opción Guardar como... del menú fichero o bien el botón Save report de la barra de herramientas, debe seleccionar el nombre y el directorio en que se guardara el reporte. El reporte se guardará con la extensión .xml. Por defecto los archivos de salida de la compilación se crearán en el directorio de instalación de iReport si no especificó uno.

Secciones de un Reporte en iReport.

A continuación se explicará de manera breve, las secciones que componen a un reporte en iReport

bandas, debe seleccionar las bandas no deseadas y colocar su propiedad band height igual a cero como se muestra en la siguiente figura.

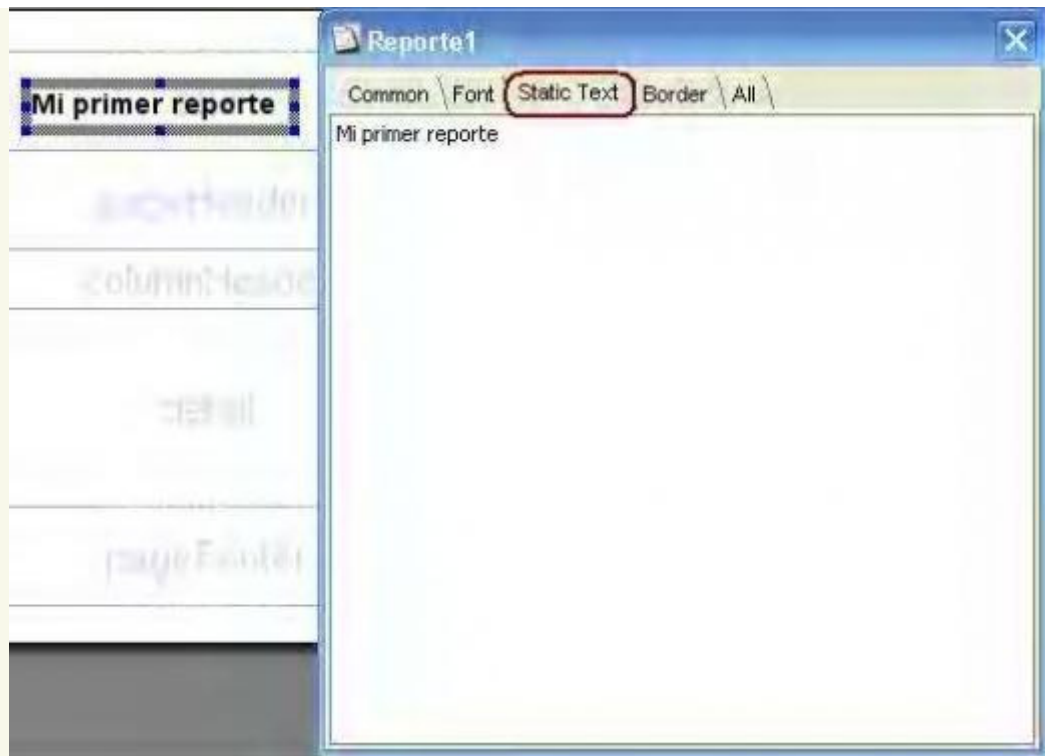


Diseño del Reporte.

Se muestran a continuación los botones principales para el diseño del reporte de la barra de herramientas:



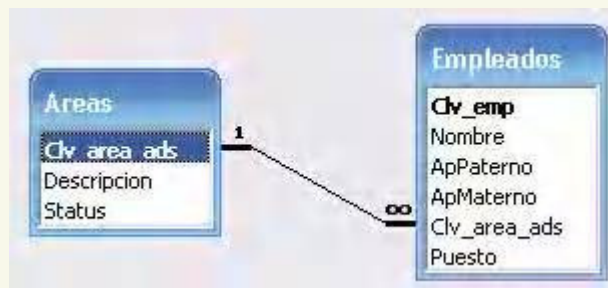
Agreguemos en primer lugar el título de nuestro reporte, para ello seleccione de la barra de herramientas el objeto Static text tool y dibuje una caja sobre la banda title. Haciendo doble click sobre la caja dibujada se mostrará la pantalla de propiedades de la caja de texto estático. Aquí puede configurar el tamaño, ubicación de la caja de texto, tipo de letra para el texto, entre otros; seleccionando la pestaña Static Text puede ingresar el título que desee para el reporte, el resultado debe ser parecido al de la siguiente figura:



En el encabezado de página, pageHeader, podemos colocar una descripción del reporte utilizando también el objeto Static Text tool.

Ahora se agregarán los nombres de los campos que pretendemos mostrar en el reporte, en este caso se recuerda que se configuró una conexión con una base de datos Access a través de un driver u origen de datos con nombre DSN.

Para cuestiones de prueba he agregado las siguientes tablas con los siguientes campos a la base de datos:

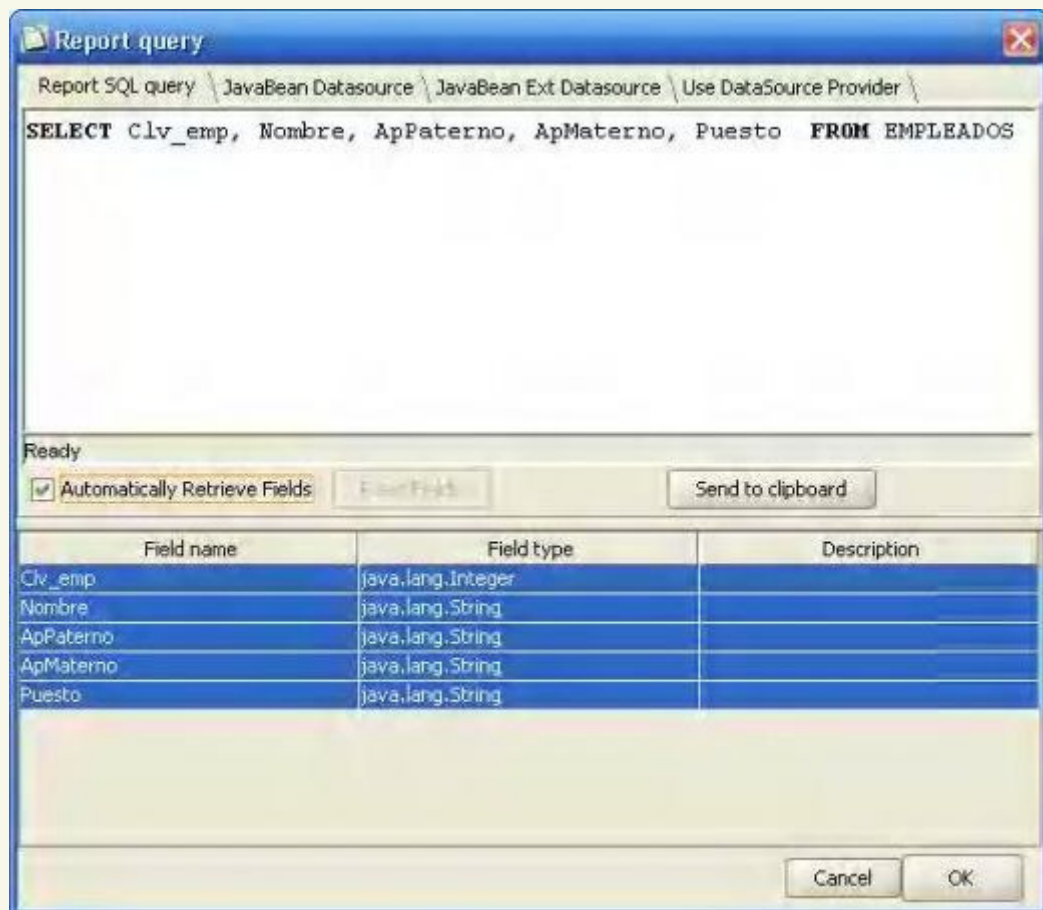


La relación de las tablas como se observa es “uno a muchos”. En este contexto, un empleado puede pertenecer a solo una área de trabajo y una área puede relacionarse con uno o muchos empleados.

Se realiza por ahora una consulta sencilla para el reporte a la tabla de empleados de la siguiente manera:

```
SELECT Clv_emp, Nombre, ApPaterno, ApMaterno, Puesto FROM EMPLEADOS
```

Antes, de agregar los nombres y campos a nuestro reporte, se establecerá la consulta anterior para el reporte. Vaya a la barra de herramientas y seleccione el botón Database, en la pantalla Report Query y pestaña Report SQL query, puede escribirse la sentencia SQL. Si se encuentra seleccionado el check box Automatically Retrieve Fields, nos mostrará automáticamente los campos que se obtienen de la consulta, el tipo y una descripción de estos si es que cuentan con ella. Si la consulta es incorrecta mostrará un mensaje de error. La pantalla debe lucir como sigue:



Debe oprimir el botón OK para guardar esta consulta.

De la misma manera en que se agregó el título al reporte, deberá agregar los nombres de los campos en la banda columnHeader utilizando objetos Static Text tool. El reporte debe lucir hasta ahora como se muestra a continuación:



Ahora solo resta colocar en la sección detail, los campos que se mostrarán en el reporte. Para esto se usará el objeto Text Field, las cajas se pintarán de manera similar a las cajas de texto estático realizadas en la banda columnHeader, sin embargo cada campo debe configurarse de acuerdo al tipo de dato que se quiere mostrar.

A continuación se mostrará la manera de configurar un Text Field. Una vez colocado un campo en la sección detail, haga doble click sobre este para abrir su ventana de propiedades y sitúese en la pestaña Text Field.

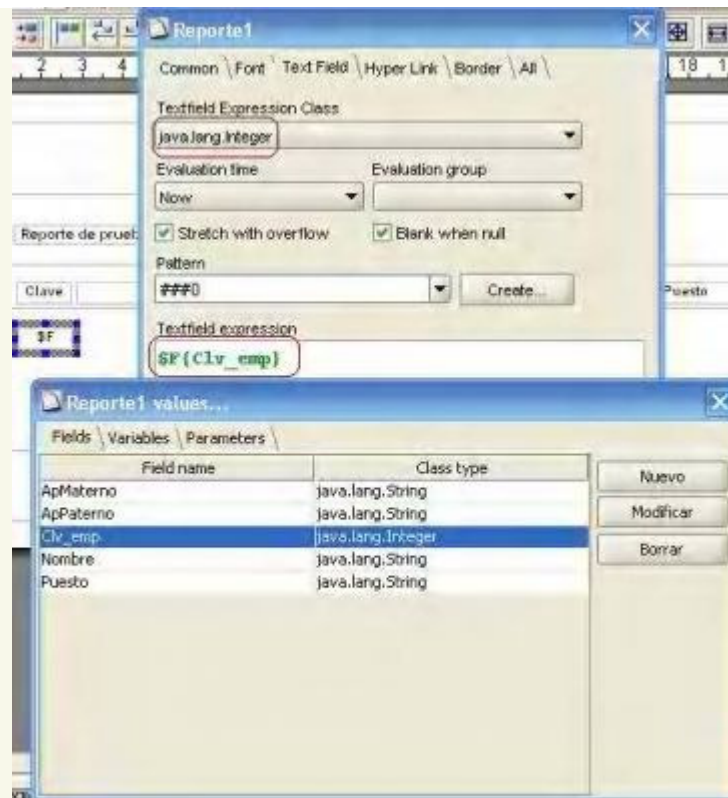
Vaya enseguida al menú Ver y seleccione el item Campos de informe, esto desplegará la pantalla values con los campos de nuestro reporte, los cuales se generaron al establecer la consulta SQL. Esta pantalla muestra adicionalmente los parámetros y variables del reporte, cada uno se distinguirá con la siguiente notación:

Campos: $\$F\{\text{Campo}\}$

Variables: $\$V\{\text{valor}\}$

Parámetros: $\$P\{\text{Parámetro}\}$

Utilice esta pantalla para auxiliarse al configurar un Text Field.



La figura anterior muestra como debe configurarse el campo “clave de empleado”.

En la ventana de propiedades ponga especial atención en seleccionar el tipo correcto del campo en el combo Textfield ExpressionClass (Integer en este caso). En la sección Textfield expression de la misma ventana, cambie la expresión `$F{Field}` por el nombre correcto del campo `$F{Clv_emp}`. Configure así cada uno de los campos restantes.

Adicionalmente, se agregará una línea al inicio de la sección detail para separar los registros con el objeto line tool de la barra de herramientas. **Debe reducir el tamaño de la banda detail al alto de las cajas de los campos para evitar demasiado espacio entre los registros** y listo, con ligeras adecuaciones a los campos, el reporte final debería lucir de la siguiente manera:

Mi primer reporte				
Reporte de prueba en iReport, muestra los datos de empleados				
Clave	Nombre	Apellido Paterno	Apellido Materno	Puesto
\$F	\$F{Nombre}	\$F{ApPaterno}	\$F{ApMaterno}	\$F{Puesto}

Compilación y Ejecución del Reporte.

Las siguientes figuras muestran los botones de la barra de herramientas necesarios para compilar, y ejecutar el reporte con o sin conexión.



Antes que nada, seleccione la vista para el Reporte, vaya al menú Construir y seleccione el ítem vista previa en JRViewer (Vista previa en el Viewer de Jasper). En este menú, puede seleccionar la vista previa para distintos formatos de archivo, siempre y cuando haya configurado los programas externos como se explicó en la primera parte del artículo.

Compile el reporte, el resultado de la compilación aparecerá en la parte inferior de la pantalla. Los errores más frecuentes de compilación se relacionan con los tipos de los campos que pretenden mostrarse. Si la compilación resultó sin errores, está listo para ver su reporte, es recomendable probarlo primero sin usar una conexión a una base de datos. Finalmente, ejecute el reporte ocupando la conexión a la base de datos que se configuró. El resultado dependiendo de sus datos en las tablas debe ser parecido al siguiente:



Mi primer reporte

Reporte de prueba en iReport, muestra los datos de empleados

Clave	Nombre	Apellido Paterno	Apellido Materno	Puesto
144	BENITO	BAHENA Y	LOME	CAJERO GENERAL
2210	LUIS	LOPEZ	VARGAS	AUXILIAR ADMTYO. "A"
10877	FELIPE	RAYON	BELTRAN	OPERARIO 1° TALLER
10884	URBANO	RAMIREZ	CORTES	CHOFER NOCTURNO TALLER
10907	GABINO	BENITO	AGUIRRE	OPERADOR DE TROLEBUS
10917	JORGE DE JESUS	HERNANDEZ	ORTIZ	CHECADOR DE TIEMPO
10922	ARTURO	MENDOZA	LORENZO	OPERADOR DE TROLEBUS
10923	J. DOMINGO	PEREZ	MEDINA	OPERARIO 1RA. "B" CABO
10936	JOSE	SANCHEZ	FLORES	OPERARIO 1° TALLER
10967	ADOLFO	GARCIA	LARA	SUPERVISOR "B" DE OPERACION
10969	ALFONSO ARTEMIO	ROSAS	VILLALVA	OPERADOR DE TROLEBUS
10971	ANTONIO	CASILLAS	CORTES	OPERADOR DE TROLEBUS
10972	JOSE LUIS	RAMIREZ	AYALA	OPERADOR DE TROLEBUS
10978	HUMBERTO	CORDOBA	VEGA	OPERADOR DE TROLEBUS
11014	MARIO	OSORIO	ARENAS	SUPERV.DE TALLER AUTOMOTRIZ
11018	GUILLERMO	SOTO	GARCIA	OPERADOR DE TROLEBUS
11023	MANUEL	VEGA	CAZARES	OPERADOR DE TROLEBUS

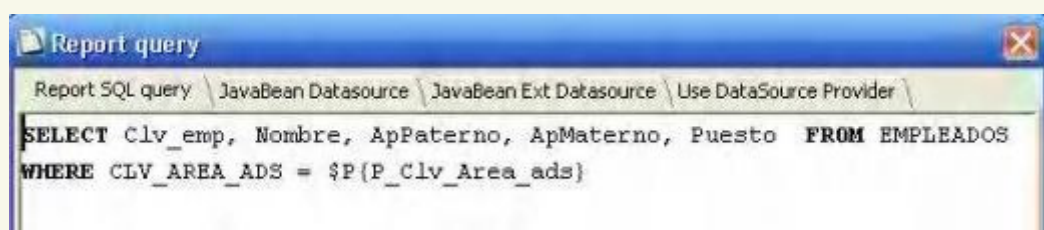
Page 1 of 99

El Viewer de JasperReports, muestra en su barra de herramientas la posibilidad de enviar el reporte directamente a la impresora o bien de guardar el reporte en algún formato de archivo específico: PDF, HTML, XLS, RTF entre otros. La funcionalidad de iReport en este caso es de oficina, obteniendo los datos almacenados en una base de datos y mostrándolos sin pasar a través de ninguna otra aplicación.

Llenar el reporte dinámicamente desde una aplicación Swing.

Algo que resultaría más interesante, es llenar el reporte dinámicamente desde alguna aplicación Java. En este caso, debe hacerse uso del archivo *.jasper generado de la compilación del archivo xml.

Oprima el botón Database de la barra de herramientas. En la ventana Report Query, modifique el Query que se muestra en la pestaña Report SQL query, como se muestra a continuación:



Report query

Report SQL query | JavaBean Datasource | JavaBean Ext Datasource | Use DataSource Provider

```
SELECT Clv_emp, Nombre, ApPaterno, ApMaterno, Puesto FROM EMPLEADOS
WHERE CLV_AREA_ADS = $P{P_Clv_Area_ads}
```

Se ha modificado el query para que se muestre en el reporte, solo a aquellos empleados que pertenezcan a determinada área, dicha área se pasará como parámetro desde una aplicación Swing para llenar el reporte.

Debe agregarse este parámetro al reporte, para esto, oprima el botón Parameters de la barra de herramientas o bien, desde el menú ver, seleccione Parámetros de informe. En la pantalla values, asegúrese de estar ubicado en la pestaña Parameters y oprima el botón Nuevo. Agregue el nuevo parámetro del reporte, como se muestra en la siguiente figura.



El tipo de parámetro se estableció como String aún cuando se sabe de las tablas que debería ser entero, en realidad esto funciona bien, pasando desde la aplicación java al reporte un String. El tipo de parámetro del reporte debe ser del mismo tipo al que se vaya a pasar desde sus aplicaciones en Java, de lo contrario obtendrá errores que le darán muchos dolores de cabeza.

Recompile el proyecto, si se muestra un mensaje de error de compilación mencionando la ausencia del parámetro, vuelva a agregarlo y abra la ventana Report query para asegurarse que se muestran los campos de la consulta. Por alguna razón, la aplicación algunas veces no detecta el nuevo parámetro.

El siguiente hilo es capaz de llenar y exportar el reporte realizado, solo ha de proporcionarse una conexión a la base de datos y la ruta del archivo jasper.

```
import java.util.*;
import java.sql.Connection;
import java.awt.event.*;

/*Librerías necesarias para Jasper Reports*/
import net.sf.jasperreports.engine.*;
import net.sf.jasperreports.view.*;

public class cExport_thread extends Thread {

    cConnection conexion;

    public cExport_thread(String Clv_area) {

    }

    /**
     * Método del hilo
     */
    public void run(){
        try
        {
            //Ruta de Archivo Jasper
            String fileName="C:\\proyecto\\Reportel.jasper";

            //Obtner una conexión a la base de datos
            conexion = new cConnection();
            Connection con = conexion.mkConection();

            //Pasamos parametros al reporte Jasper.
            Map parameters = new HashMap();
            parameters.put("P_Clv_Area_ads",Clv_area);

            //Preparacion del reporte (en esta etapa llena el diseño de reporte)
            //Reporte diseñado y compilado con iReport
            JasperPrint jasperPrint = JasperFillManager.fillReport(fileName,parameters,con);

            //Se lanza el Viewer de Jasper, no termina aplicación al salir
            JasperViewer jviewer = new JasperViewer(jasperPrint,false);
            jviewer.show();
        }
        catch (Exception j)
        {
            System.out.println("Mensaje de Error:"+j.getMessage())
        }
        finally{
```



```
        conexion.closeConecction();  
    }  
}  
}
```

La finalidad del hilo, en mi caso particular, fue para liberar de carga al evento de un botón en una aplicación Swing, dado que la obtención de la conexión y el llenado del reporte puede ser tardado. El hilo debe lanzarse de la siguiente manera:

```
cExport_thread thread_exp = new cExport_thread();  
thread_exp.start();
```

Hasta esta fecha, han salido bastantes versiones de JasperReports e iReport, si utiliza versiones diferentes a las aquí utilizadas, asegúrese que coincidan las librerías de Jasper tanto en iReport como en su proyecto java con el que pretende llenar el reporte.

referencia de herramientas utilizadas:

IDE: JBuilder 2005, iReport 0.5.1

JDK(SDK): j2sdk1.4.2_08

JasperReports 1.0.1

Vr Access: Access 2002

Creación de Gráficos en iReport

JasperReports no maneja directamente gráficos: estos deben crearse independientemente como imágenes, incluso utilizando una de las numerosas librerías de código libre disponibles para la creación de gráficos. La imagen producida será mostrada usando un componente de imagen. La idea es realmente simple, pero la creación de un gráfico en tiempo de ejecución requiere de un buen conocimiento de la programación de JasperReports, y muchas veces es necesario utilizar scriptlets capaces de coleccionar los datos que se mostrarán en el gráfico.

A partir de la versión 0.4.0, iReport proporciona una herramienta para simplificar la construcción de un gráfico. Esta herramienta permite crear un gráfico configurando propiedades y datos principales que serán vistos de manera simple por el usuario final.

La creación de un gráfico se basa en una librería muy conocida de código libre llamada JFreeCharts desarrollada por David Gilbert de Object Refinery Limited. iReport soporta por ahora solamente

un pequeño número de tipos de gráficos presentados en JFreeCharts, y pueden modificarse solamente algunas de las propiedades del gráfico, pero es posible aún así, crear gráficos limpios con un gran impacto a la vista.

Fuente: Documentación y Tutoriales de iReport en: <http://ireport.sourceforge.net/>

Después de la pequeña introducción, se mostrará la manera de crear un grafico en iReport.

La versión de iReport aquí utilizada será la **0.5.1**, para versiones más recientes, la creación de los gráficos puede ser diferente; deben referirse a la documentación correspondiente de su versión.

Supondremos que ya se cuenta con una configuración correcta del iReport y sus librerías, especialmente la librería JFreecharts, la versión que se incluye con iReport 0.5.1, es la jfreechart-1.0.0-rc1.

Además, se necesitará tener configurada una conexión a una base de datos, y por supuesto datos para poder establecer una consulta que alimentará con datos al gráfico.

A partir de los datos de las siguientes tablas, se creará el gráfico en cuestión.



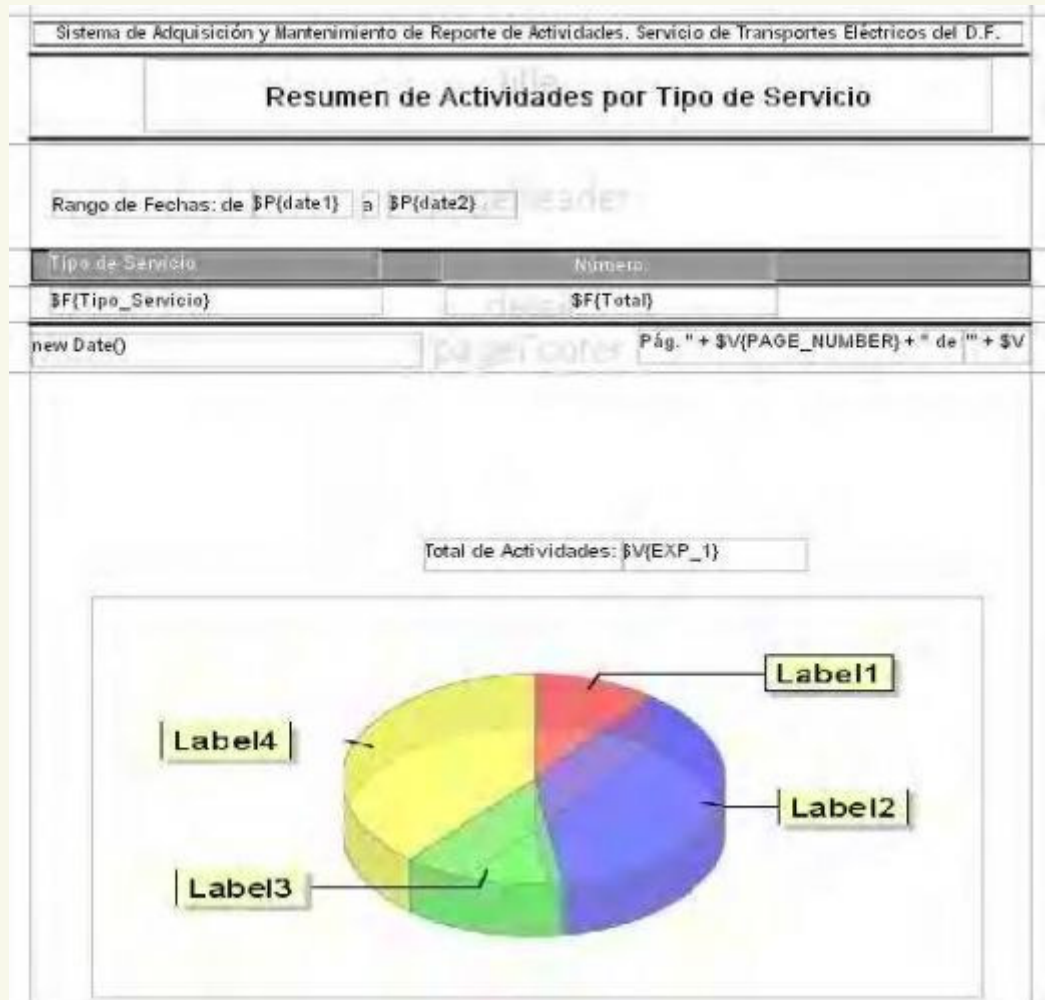
Las tablas anteriores muestran las actividades realizadas en una empresa. Cada una de las actividades se relaciona con un solo servicio. Cada actividad pertenece a un trabajador, que por simplicidad este campo no se muestra en este diseño de tablas.

Supongamos que se desea saber cuantas actividades se realizaron por servicio en un determinado rango de fechas, digamos en un mes. Se debe proceder a crear la consulta SQL correspondiente para obtener estos datos, más aún, podemos reflejarlos gráficamente, la consulta sería como la siguiente:

```
Report query
Report SQL query \ JavaBean Datasource \ JavaBean Ext Datasource \ Use DataSource Provider \
select Servicios.Tipo_Servicio,count(*)As Total
from Actividades,Servicios
where Actividades.IdServicio = Servicios.IdServicio
And Actividades.Fecha_ter >= #1/5/2005#
And Actividades.Fecha_ter <= #25/5/2005#
Group By Servicios.Tipo_Servicio
```

La consulta anterior agrupará las actividades realizadas por tipo de Servicio que se realizaron del 1 de mayo al 25 de mayo del 2005.

El diseño de reporte es el siguiente:



Para insertar un gráfico en un reporte de iReport, seleccione el objeto Chart tool de la barra de herramientas de iReport y dibuje el gráfico sobre la sección summary del reporte:



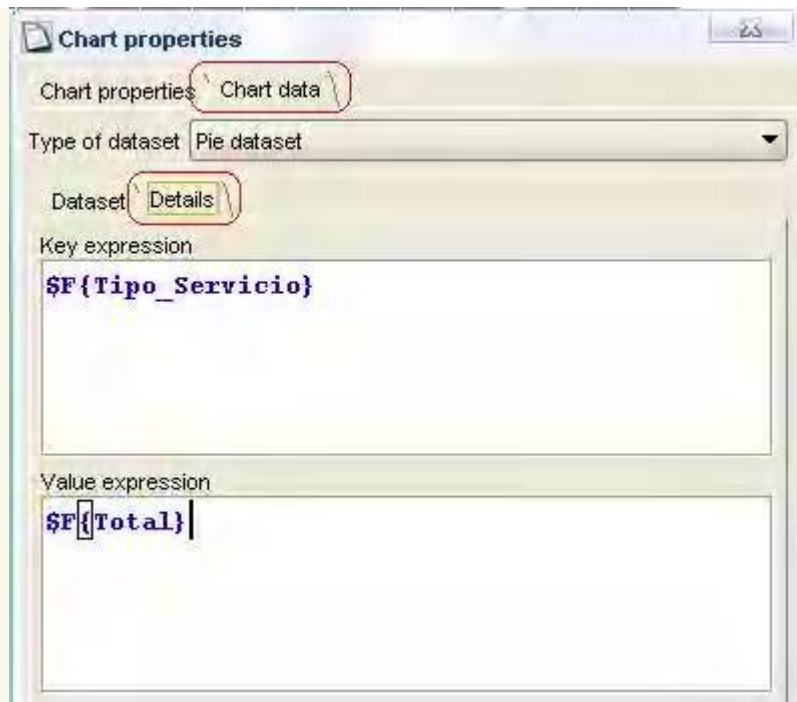
A continuación deberá seleccionar el tipo de gráfico deseado, para este ejemplo seleccionamos el tipo Pie 3D. Oprimir el botón OK y el componente que contendrá al gráfico ya ha sido creado, lo que resta es configurar las propiedades del mismo y los datos que mostrará.

Haciendo doble click sobre el componente del gráfico, se mostrará la pantalla de propiedades de este, sitúese en la pestaña Chart, a continuación oprima el botón Edit chart properties, deberá aparecer una pantalla como la siguiente:



En la pestaña Chart properties de esta pantalla, se encuentran las propiedades que pueden modificarse para el gráfico en turno, como el nombre del gráfico, las fuentes, leyendas, ect. Puede dejar las propiedades actuales por ahora y modificarlas posteriormente. Ahora centraremos la atención en la manera de configurar los datos para el gráfico.

Debe situarse en la pestaña Chart data de la pantalla anterior y enseguida en la pestaña Details. En el apartado Key expression, coloque el campo `$F{Tipo_Servicio}` y en el apartado Value expresión coloque el campo `$F{Total}`. Estos campos son precisamente los que se obtienen de la consulta que se estableció anteriormente, estos a su vez se mostrarán en la sección detail del reporte.



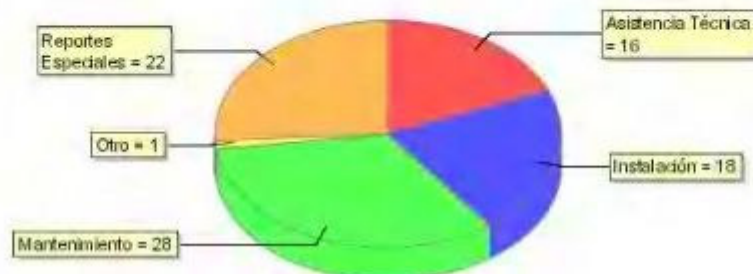
Listo se han configurado los datos que necesita el gráfico. Al compilar y ejecutar el reporte con iReport el resultado puede ser parecido al siguiente:

Resumen de Actividades por Tipo de Servicio

Rango de Fechas: de 01/05/2005 a 25/05/2005.

Tipo de Servicio	Número
Asistencia Técnica	16
Instalación	18
Mantenimiento	28
Otro	1
Reportes Especiales	22

Total de Actividades: 85



Si se desea mostrar el gráfico anterior en tiempo de ejecución desde una aplicación Java, se procede de manera semejante a como se muestran otros reportes. En este caso deben determinarse los parámetros que se pasarán desde la aplicación java al reporte, por lo que la consulta se modifica como se muestra a continuación:

```

Report query
Report SQL query \ JavaBean Datasource \ JavaBean Ext Datasource \ Use DataSource Provider \
select Servicios.Tipo_Servicio,count(*) As Total
from Actividades,Servicios
where Actividades.IdServicio = Servicios.IdServicio
And Actividades.Fecha_ter >= $P{date1}
And Actividades.Fecha_ter <= $P{date2}
Group By Servicios.Tipo_Servicio

```

Lo que se pretende con la consulta anterior, es pasar como parámetros las fechas para las que se desea obtener el total de actividades agrupadas por servicio.

Para la versión 0.5.1 de iReport, solo debe compilarse el reporte para obtener el archivo Jasper que será llenado por la aplicación. En versiones anteriores o quizá recientes, es posible que se necesite incluir un scriptlet para coleccionar los datos del gráfico.

Contando con el archivo Jasper del reporte, la manera en que se manda llenar desde una aplicación Java es la misma que para otros reportes, como se muestra en el siguiente fragmento de código:

```
...
//Ruta de Archivo Jasper
String fileName="C:\\proyecto\\Grafico.jasper";

//Obtner una conexión a la base de datos
conexion = new cConnection();
Connection con = conexion.mkConection();

//Pasamos parametros al reporte Jasper.
Map parameters = new HashMap();
parameters.put("date1",p_date1);
parameters.put("date2",p_date2);

//Preparacion del reporte (en esta etapa llena el diseño de reporte)
//Reporte diseñado y compilado con iReport
JasperPrint jasperPrint = JasperFillManager.fillReport(fileName,parameters,con);

//Se lanza el Viewer de Jasper, no termina aplicación al salir
JasperViewer jviewer = new JasperViewer(jasperPrint,false);
jviewer.show();
...
```

Puede exportar el reporte a diferentes formatos de archivo directamente desde la aplicación sin pasar por el JasperViewer, para esto puede referirse a las API's de su versión correspondiente.