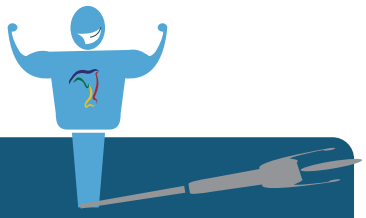
autentia

Avenida de Castilla,1 - Edificio Best Point - Oficina 21B
28830 San Fernando de Henares (Madrid)
tel./fax: +34 91 675 33 06
info@autentia.com - **www.autentia.com**

¿**Qué ofrece** Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

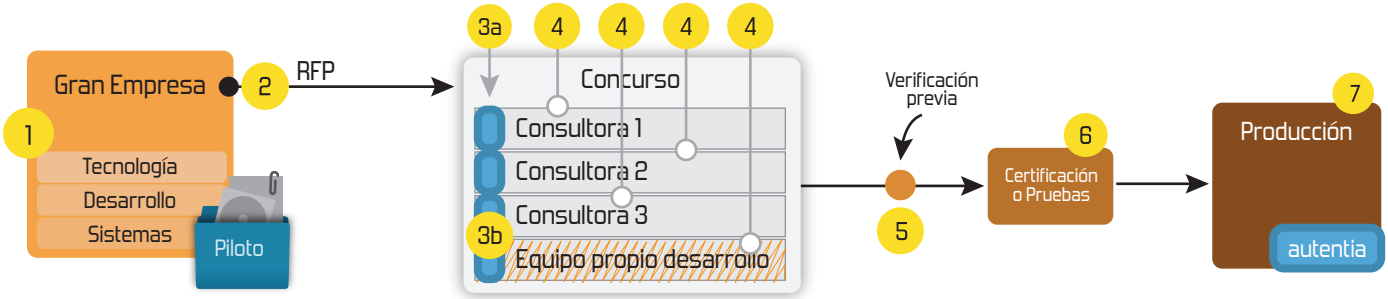
1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces, HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
Gestor de contenidos (Alfresco)
Aplicaciones híbridas

Control de autenticación y acceso (Spring Security)
UDDI
Web Services
Rest Services
Social SSO
SSO (Cas)

JPA-Hibernate, MyBatis
Motor de búsqueda empresarial (Solr)
ETL (Talend)

Dirección de Proyectos Informáticos.
Metodologías ágiles
Patrones de diseño
TDD

Tareas programadas (Quartz)
Gestor documental (Alfresco)
Inversión de control (Spring)

BPM (jBPM o Bonita)
Generación de informes (JasperReport)
ESB (Open ESB)

Compartimos nuestro conocimiento en:
www.adictosaltrabajo.com

Para más información visítenos en:
www.autentia.com

Los años enseñan las cosas que los días jamás llegan a conocer
Adictos AlTrabajo
Powered by autentia
Hosting patrocinado por enredados

Ultimas Noticias

» Autentia patrocina el primer Agile Open Spain
» Nuevas funcionalidades en la web de www.adictosaltrabajo.com
» Disponible la primera versión de los plugins para integrar Maven y Bugzilla.
» Historia de la Informática. Capítulo 78. 2001-2002
» Comentando el llibro: "No he venido aquí a hacer amigos, Desventuras de un consultor IT"
» Lanzamiento del nuevo Web de Autentia
» Si se pregunta ¿Qué ofrece este Web?
» Grupo XING
» Autentia cumple 6 años

+Noticias Destacadas

» Autentia patrocina el primer Agile Open Spain
» Lanzamiento del nuevo Web de Autentia
» Autentia cumple 6 años
» Grupo XING

+Comentarios Cómic

+Enlaces

Tutorial desarrollado por



Víctor Javier Madrid

Consultor tecnológico de desarrollo de proyectos informáticos.

Ingeniero en Informática por la Universidad de Alcalá de Henares.

Puedes encontrarme en Autentia

Somos expertos en Java/J2EE

Catálogo de servicios de Autentia

Descargar (6,2 MB)

Descargar en versión comic (17 MB)

AdictosAlTrabajo.com es el Web de difusión de conocimiento de Autentia.

autentia real business solutions

Catálogo de cursos

Descargar este documento en formato PDF: iReportParametrosDinamicos.pdf

Fecha de creación del tutorial: 2009-09-28

iReport : Solución al problema de los parámetros dinámicos.

0. Índice de contenidos.

- 1. Introducción.
- 2. Entorno.
- 3. Requerimientos.
- 4. Funcionamiento.
- 5. Caso 1 : Ejecución de consulta de base de datos.
- 6. Caso 2 : Ejecución de consulta de base de datos.
- 7. Ejemplo.
- 8. Conclusiones.

1. Introducción

Con este tutorial no se pretende enseñar a utilizar la herramienta de iReport Para ello ya existen varios tutoriales muy buenos enwww.adictosaltrabajo.com , donde se explican detenidamente como generar diferentes tipos de informes mediante el uso de esta herramienta, asi como el uso de muchas de sus funcionalidades.Algunos de los tutoriales son los siguientes :

1. Introducción a iReport
2. Subinformes con iReport
3. Gráficos con iReport
4. Existen más tutoriales...

El objetivo principal de este tutorial es enseñar la forma de solucionar el problema que se produce a la hora de generar un informe cuando se incorpora un parámetro y posteriormente se incorpora otro que es usado por el primero (un poco liosos pero ya lo veréis más claro en el ejemplo).El problema reside en que iReport es incapaz de darse cuenta de esta situación (problema en el orden de declaración de los parámetros) y no dejará generar un informe (ni incluso mostrará mensaje de error alguno ..cosa que no es muy lógica ;-))

Por otro lado y como objetivo secundario de este tutorial se va a explicar como se construye internamente un informe con iRepor, para ello se analizará el código fuente generado en el fichero jrxml para cada uno de los pasos realizados durante este tutorial.Con esta parte, se pretende perder el miedo a este tipo de ficheros y que se sepa modificar directamente sobre el código.

Consejo : Es conveniente tener una idea del código generado en cada informe.

2. Entorno.

El tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil Asus G50Vseries (Core Duo P8600 2.4GHz, 4GB RAM, 320 GB HD).
- Sistema operativo: Windows Vista Ultimate.
- iReport 3.0.0
- MySQL 5.1.32
- MySQL Workbench OSS for Windows versión 5.1.18 (Para el diseño del modelo de datos)

3. Requerimientos.

Para la realización de este tutorial se ha utilizado la herramienta de generación de diagramas de entidad-relación MySQL Workbench.Con ella se ha generado un modelo de datos formado por 3 tablas :

1. **Clientes :** Tabla que almacenará la información de los usuarios del sistema creado
2. **Cuentas :** Tabla que almacenará la información asociada a la cuenta de los usuarios
3. **Perfiles :** Tabla que almacenará los perfiles de las cuentas

Requisito 1 : Un cliente puede tener más de una cuenta.

Requisito 2 : Un perfil puede estar asociado a más de una cuenta.

Diagrama de Entidad Relación

Representa las tablas anteriores y la información a almacenar.

Diagrama Entidad Relación

Script de creación del modelo de datos

Nota: Es necesario tener creado el esquema en el que se guardaran.

Catálogo de servicios Autentia (PDF 6,2MB)

En formato comic...

Acceso de usuarios registrados:

E-mail:
Contraseña:
Entrar

Deseo registrarme

He olvidado mis datos de acceso

Registra tu empresa:

Descubre las ventajas de registrar tu empresa en AdictosAlTrabajo...

Registrar mi empresa

Listado de empresas ya registradas

Google

Web
www.adictosaltrabajo.com
Buscar

Últimos tutoriales

2009-09-28
iReport : Solución al problema de los parámetros dinámicos.

2009-09-27
Mavenizar Liferay SDK

2009-09-27
Crear un plugin con Liferay SDK

2009-09-23
EJB 3.0 y pruebas unitarias con Maven, JUnit 4 y Apache Open EJB.

2009-09-21
Primeros pasos con JBoss Seam

2009-09-21
Integración con sistemas de Bug Tracking desde NetBeans 6.7

2009-09-14
EJB 3.0 y pruebas de persistencia con Maven, JUnit 4 y Embedded JBoss sobre Java 6.

2009-09-12
Instalación de Liferay en Tomcat existente

2009-09-11
Release Bugzilla Maven Plugin

2009-09-11
Enlazar Bugzilla con MavenChangesPlugin

2009-09-08
Sobre las reglas de codificación o... ¿de dónde salen esos caracteres "raros"?.

2009-08-28
Cómo hacer deploy del site de Maven en SourceForge

2009-08-26
Ordenación por cantidades en informe cruzado

2009-08-20
Selenium IDE-Incorporando while en los test

2009-08-14
Blender y JMonkeyEngine. Exportación de archivos Blender y uso de los mismos en JMonkeyEngine

2009-08-14
5º tutorial TNT Concept Versión 0.16.1

Anuncios Google Tutorial Power Point Manual 3D Studio Steinberg Software Programación Java Ejemplos 3D Max

Copyright 2003-2009 © All Rights Reserved | Texto legal y condiciones de uso | Banners | Powered by Autentia | Contacto

```
view plain print ?

-- -----------------------------------------------------
-- Table `Clientes`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `Clientes` ;

CREATE  TABLE IF NOT EXISTS `Clientes` (
  `idClientes` INT NOT NULL ,
  `nombre` VARCHAR(45) NULL ,
  `apellidos` VARCHAR(45) NULL ,
  `direccion` VARCHAR(45) NULL ,
  PRIMARY KEY (`idClientes`) )
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `Perfiles`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `Perfiles` ;

CREATE  TABLE IF NOT EXISTS `Perfiles` (
  `idPerfiles` INT NOT NULL ,
  `descripcion` VARCHAR(45) NULL ,
  PRIMARY KEY (`idPerfiles`) )
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `Cuentas`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `Cuentas` ;

CREATE  TABLE IF NOT EXISTS `Cuentas` (
  `idCuentas` INT NOT NULL ,
  `usuario` VARCHAR(45) NULL ,
  `contraseña` VARCHAR(45) NULL ,
  `fecha_alta` DATETIME NOT NULL ,
  `fecha_baja` DATETIME NULL ,
  `baja_logica` INT DEFAULT 0 ,
  `Clientes_idClientes` INT NOT NULL ,
  `Perfiles_idPerfiles` INT NOT NULL ,
  PRIMARY KEY (`idCuentas`) )
ENGINE = InnoDB;

CREATE INDEX `fk_Cuentas_Clientes` ON `Cuentas` (`Clientes_idClientes` ASC) ;

CREATE INDEX `fk_Cuentas_Perfiles1` ON `Cuentas` (`Perfiles_idPerfiles` ASC) ;
```

Script para la carga de datos

Script de carga de datos de la tabla Clientes :

```
view plain print ?

insert into `clientes`(`idClientes`,`nombre`,`apellidos`,`direccion`) values (1,'Víctor','Apellidos 1','Dirección 1');
insert into `clientes`(`idClientes`,`nombre`,`apellidos`,`direccion`) values (2,'Juan Pablo','Apellidos 2','Dirección 2');
insert into `clientes`(`idClientes`,`nombre`,`apellidos`,`direccion`) values (3,'Fernando','Apellidos 3','Dirección 3');
insert into `clientes`(`idClientes`,`nombre`,`apellidos`,`direccion`) values (4,'Fran','Apellidos 4','Dirección 4');
insert into `clientes`(`idClientes`,`nombre`,`apellidos`,`direccion`) values (5,'Tamara','Apellidos 5','Dirección 5');
insert into `clientes`(`idClientes`,`nombre`,`apellidos`,`direccion`) values (6,'Marta','Apellidos 6','Dirección 6');
```

Script de carga de datos de la tabla Perfiles :

```
view plain print ?

insert into `perfiles`(`idPerfiles`,`descripcion`) values (1,'Administrador');
insert into `perfiles`(`idPerfiles`,`descripcion`) values (2,'Gerente');
insert into `perfiles`(`idPerfiles`,`descripcion`) values (3,'Comercial');
insert into `perfiles`(`idPerfiles`,`descripcion`) values (4,'Administrativo');
insert into `perfiles`(`idPerfiles`,`descripcion`) values (5,'Usuario');
```

Script de carga de datos de la tabla Cuentas :

```
view plain print ?

insert into `cuentas`(`idCuentas`,`usuario`,`contraseña`,`fecha_alta`,`fecha_baja`,`baja_logica`,`Clientes_idClientes`,`Perfiles_idPerfiles`)
values (1,'VíctorAdministrador','contraseña1','2009-07-01 00:00:00',null,'0',1,1);

insert into `cuentas`(`idCuentas`,`usuario`,`contraseña`,`fecha_alta`,`fecha_baja`,`baja_logica`,`Clientes_idClientes`,`Perfiles_idPerfiles`)
values (2,'VíctorUsuario','contraseña2','2009-07-18 00:00:00',null,'0',1,5);

insert into `cuentas`(`idCuentas`,`usuario`,`contraseña`,`fecha_alta`,`fecha_baja`,`baja_logica`,`Clientes_idClientes`,`Perfiles_idPerfiles`)
values (3,'JuanPabloComercial','contraseña3','2009-07-18 00:00:00',null,'0',2,3);

insert into `cuentas`(`idCuentas`,`usuario`,`contraseña`,`fecha_alta`,`fecha_baja`,`baja_logica`,`Clientes_idClientes`,`Perfiles_idPerfiles`)
values (4,'JuanPabloUsuario','contraseña4','2009-07-18 00:00:00',null,'0',2,5);

insert into `cuentas`(`idCuentas`,`usuario`,`contraseña`,`fecha_alta`,`fecha_baja`,`baja_logica`,`Clientes_idClientes`,`Perfiles_idPerfiles`)
values (5,'FernandoAdministrativo','contraseña5','2009-07-18 00:00:00',null,'0',3,4);

insert into `cuentas`(`idCuentas`,`usuario`,`contraseña`,`fecha_alta`,`fecha_baja`,`baja_logica`,`Clientes_idClientes`,`Perfiles_idPerfiles`)
values (6,'FranAdministrativo','contraseña7','2009-07-18 00:00:00',null,'0',4,4);

insert into `cuentas`(`idCuentas`,`usuario`,`contraseña`,`fecha_alta`,`fecha_baja`,`baja_logica`,`Clientes_idClientes`,`Perfiles_idPerfiles`)
values (7,'FranUsuario','contraseña7','2009-07-18 00:00:00',null,'0',4,5);

insert into `cuentas`(`idCuentas`,`usuario`,`contraseña`,`fecha_alta`,`fecha_baja`,`baja_logica`,`Clientes_idClientes`,`Perfiles_idPerfiles`)
values (8,'TamaraGerente','contraseña8','2009-07-18 00:00:00',null,'0',5,2);

insert into `cuentas`(`idCuentas`,`usuario`,`contraseña`,`fecha_alta`,`fecha_baja`,`baja_logica`,`Clientes_idClientes`,`Perfiles_idPerfiles`)
values (9,'TamaraUsuario','contraseña9','2009-07-18 00:00:00',null,'0',5,5);

insert into `cuentas`(`idCuentas`,`usuario`,`contraseña`,`fecha_alta`,`fecha_baja`,`baja_logica`,`Clientes_idClientes`,`Perfiles_idPerfiles`)
values (10,'MartaComercial','contraseña10','2009-07-18 00:00:00',null,'0',5,3);
```

4. Funcionamiento.

En este punto se va a enseñar a utilizar de forma básica la herramienta iReport,con ello se va a generar el informe con el que después se enseñará el ejemplo. Pasos a seguir:

Paso 1 : Cargar la aplicación iReport

Gestión de informes, vacaciones y utilidades

2009-08-14
Joomla 1.5. Instalación y configuración

2009-08-13
Introducción a los diagramas EPC (Event-Driven Process Chain)

2009-08-10
Blender. Animaciones avanzadas y renderización

2009-08-10
Gestión de Calidad, tablón y seguimiento en TNT Concept Versión 0.16.1

2009-08-10
Cómo hacer una página web

2009-08-06
Tips And Tricks JUnit Spring

2009-08-03
Instalación de VirtualBox PUEL

2009-08-03
Gestión de contactos y pedidos en TNT Concept versión 0.16.1

2009-08-03
Comentando el libro: La estrategia del océano azul

2009-07-30
Funciones esenciales para crear un juego.

2009-07-30
2º tutorial TNT Concept versión 1.16.1

2009-07-29
Hibernate Search, Bridges, Analizadores y más

2009-07-24
Migración de EJB3 a JPA y Spring.

2009-07-20
Directorio de ejemplos de jMonkey Engine

2009-07-19
JSR-179 Location API para J2ME: Posicionamiento geográfico en nuestras aplicaciones.

2009-07-16
Gestión de Usuarios en TNT Concept versión 0.16.1

2009-07-16
Continuación del Tutorial: JMonkeyEngine, Creación de nuestro primer juego.

2009-07-16
Como implementar el Scene Monitor para analizar las escenas en jMonkeyEngine

2009-02-26
Transformaciones de escena en JMonkeyEngine

2009-07-15
Detalles del juego de la moto en jMonekyEngine.

2009-07-14
JMonkeyEngine, Creación de nuestro primer juego.

2009-07-13
Ajax tests con Selenium: prototype.js e ICEfaces.

2009-07-08
AOP con AspectJ y Maven

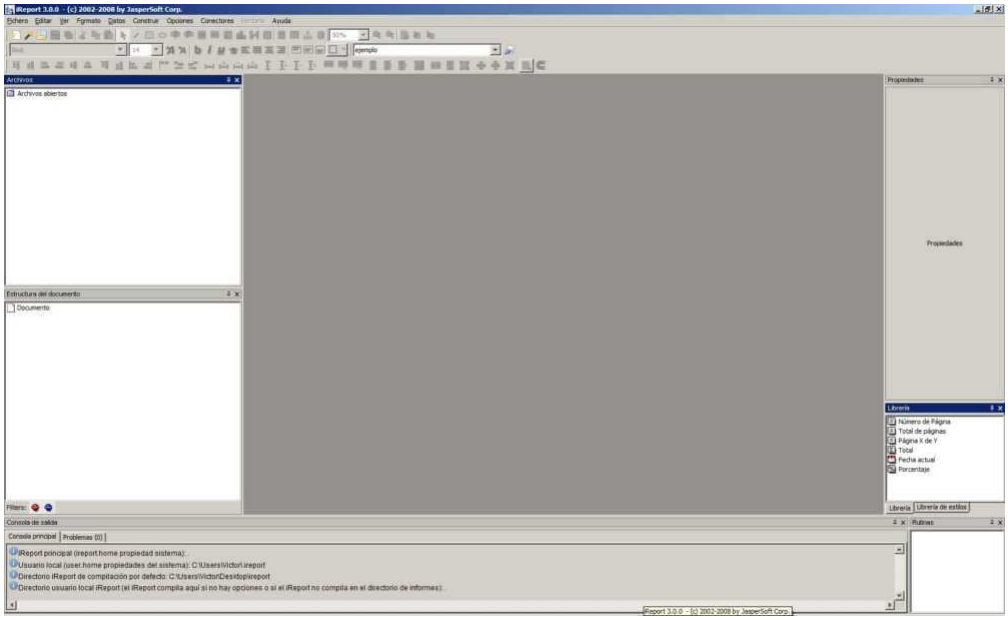
2009-07-07
Instalación y configuración de Eclipse Galileo

2009-07-07
Iniciarse en el manejo de JME, Creación de un Cloth.

2009-07-06
Primeros pasos con Blender: Pintando nuestra mascota en 3D

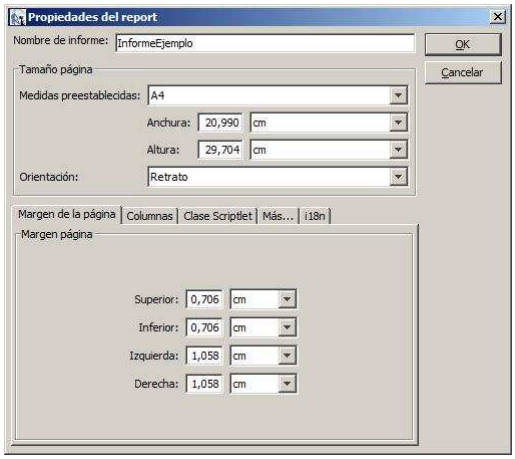
2009-07-06
DBUnit-Exportar e Importar BBDD

2009-07-05
JMeter, Pruebas de stress sobre aplicaciones web: Grabando y reproduciendo navegaciones

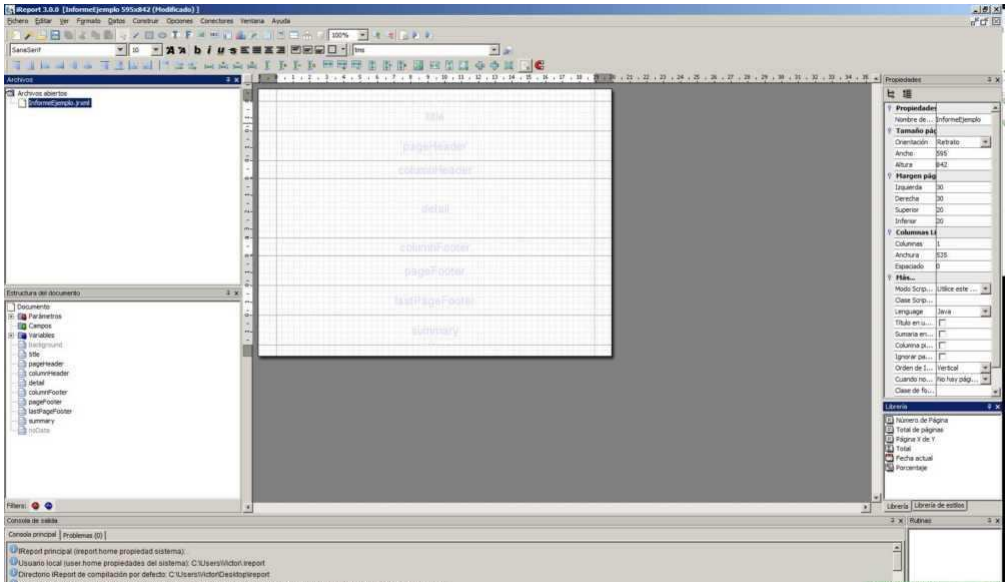


Paso 2 : Crear un nuevo informe (Fichero -> Nuevo Documento)

Darle un nombre al informe, en este caso "InformeEjemplo".



Se generará el informe con las bandas activas.



Si guardamos el fichero y lo visualizamos con un editor de texto podremos ver el código del fichero jrxml generado :

2009-07-02
Axis2: Invocación de Servicios Web usando distintos MEP

2009-07-02
Instalación OpenOffice

2009-07-02
Juegos 3D en Java: Blender y JMonkeyEngine

2009-06-20
StAX (Xml Pull Parser): Streaming API para XML

2009-06-15
Configuración de la desconexión de usuarios con ICEFaces

2009-06-10
LWUIT: Una librería gráfica tipo AWT o Swing para J2ME

Últimas ofertas de empleo

2009-07-31
T. Información - Operador (dia / noche) - BARCELONA.

2009-06-25
Atención a cliente - Call Center - BARCELONA.

2009-06-19
Otras - Ingenieria (minas, puentes y puertos) - VALENCIA.

2009-06-17
Comercial - Ventas - ALICANTE.

2009-06-03
Comercial - Ventas - VIZCAYA.

Anuncios Google

```
view plain  print  ?

<!-- Created with iReport - A designer for JasperReports --
<!DOCTYPE jasperReport PUBLIC "//JasperReports//DTD Report Design//EN" "http://jasperreports.sourceforge.net/dtds/jasperreport.dt>
<jasperReport
        name="InformeEjemplo"
        columnCount="1"
        printOrder="Vertical"
        orientation="Portrait"
        pageWidth="595"
        pageHeight="842"
        columnWidth="535"
        columnSpacing="0"
        leftMargin="30"
        rightMargin="30"
        topMargin="20"
        bottomMargin="20"
        whenNoDataType="NoPages"
        isTitleNewPage="false"
        isSummaryNewPage="false">
    <property name="ireport.scriptlethandling' value="2" />
    <property name="ireport.encoding" value="UTF-8" />
    <import value="java.util.*" />
    <import value="net.sf.jasperreports.engine.*' />
    <import value="net.sf.jasperreports.engine.data.*' />


        <background>
            <band height="0"  isSplitAllowed="true" >
            </band>
        </background>
        <title>
            <band height="50"  isSplitAllowec="true" >
            </band>
        </title>
        <pageHeader>
            <band height="50"  isSplitAllowec="true" >
            </band>
        </pageHeader>
        <columnHeader>
            <band height="30"  isSplitAllowec="true" >
            </band>
        </columnHeader>
        <detail>
            <band height="100"  isSplitAllowed="true" >
            </band>
        </detail>
        <columnFooter>
            <band height="30"  isSplitAllowec="true" >
            </band>
        </columnFooter>
        <pageFooter>
            <band height="50"  isSplitAllowec="true" >
            </band>
        </pageFooter>
        <lastPageFooter>
            <band height="50"  isSplitAllowec="true" >
            </band>
        </lastPageFooter>
        <summary>
            <band height="50"  isSplitAllowec="true" >
            </band>
        </summary>
</jasperReport>
```

Como se puede observar en el código fuente del informe, se trata de un fichero XML en el cual se establecen las características del informe como son : nombre,orientación, codificación, etc. ,así como detalles de la maquetación del informe : márgenes, si se mostrará el sumario,etc..

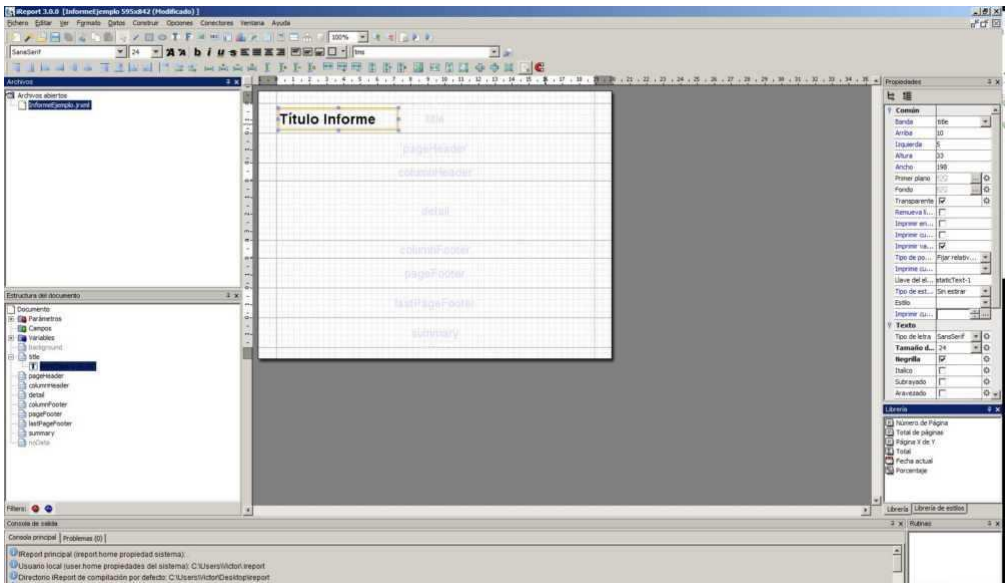
Nota : Observar las etiquetas correspondientes a las bandas del informe.

Las características o las propiedades pueden ser modificadas directamente desde el código fuente y refrescando el informe desde la aplicación se pueden hacer efectivos los cambios.

Por otro lado se definen todas y cada una de las bandas que formarán en informe donde aparece reflejado su tamaño por defecto y si esta activo el reparto permitido.

Paso 3 : Introducir un texto estático en el título

Se insertará el texto estático "Título informe" en el título desde el icono que tiene una T.



Por lo tanto, en el código fuente se incorporará el siguiente código en la banda de título:

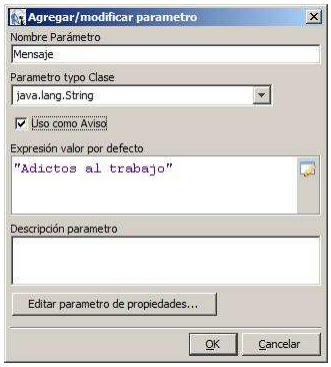
```
view plain  print  ?

.
.
.
<staticText>
    <reportElement
        x="5"
        y="10"
        width="198"
        height="33"
        key="staticText-1"/>
    <box></box>
    <textElement>
        <font pdfFontName="Helvetica-Bold" size="24" isBold="true"/>
    </textElement>
<text><![CDATA[Titulo Informe]]></text>
</staticText>
.
.
.
```

En el código se muestra la posición del texto,altura/anchura del campo y las características de la fuente utilizada para mostrar la información.

Paso 4 : Introducir un parámetro

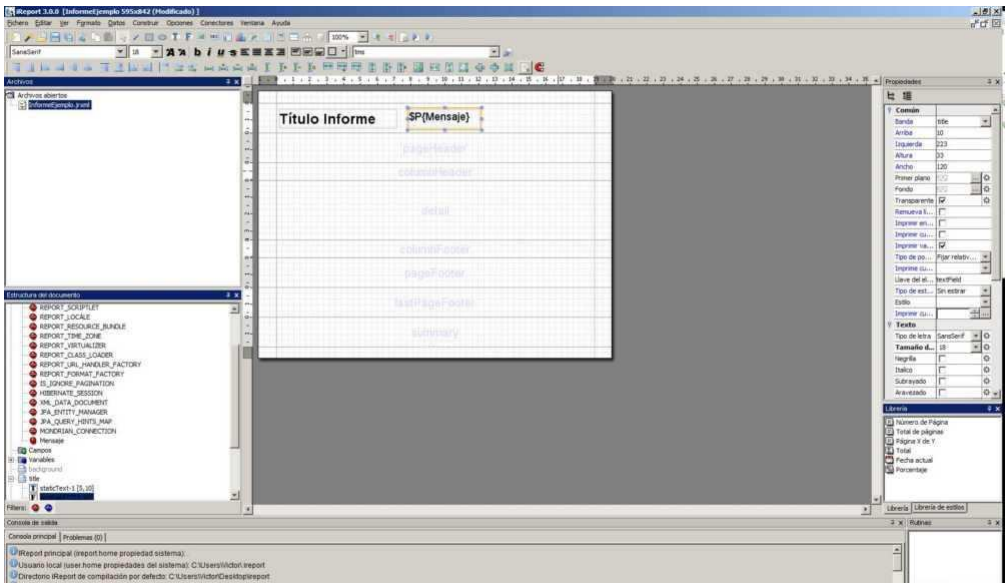
Insertar un parámetro en el informe del tipo String,con identificador "Mensaje" y valor por defecto "Adictos al trabajo".



En el fichero jrxml se incorporará :

```
view plain  print  ?
01. .
02. .
03. .
04. <parameter name="Mensaje" isForPrompting="true" class="java.lang.String">
05.     <defaultValueExpressior ><![CDATA["Adictos al trabajo"]]></defaultValueExpressior>
06. </parameter>
07. .
08. .
09. .
```

Visualizar que se ha incorporado un campo de texto con la referencia al parámetro $P{NOMBRE_PARAMETRO}.

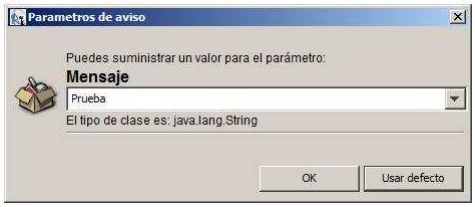


En el código fuente se incorporará la siguiente entrada a continuación de la anterior:

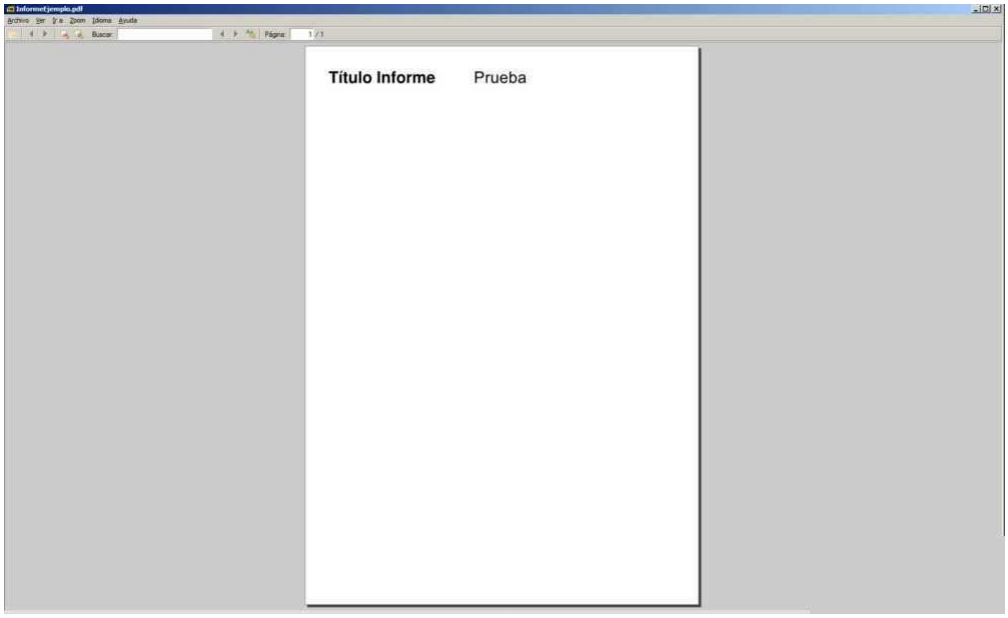
```
view plain  print  ?
01. .
02. .
03. .
04. <textField isStretchWithOverflow="false" isBlankWhenNull="false" evaluationTime="Now" hyperlinkType="None"  hyperlinkTarget="Self" >
05.     <reportElement
06.         x="223"
07.         y="10"
08.         width="145"
09.         height="33"
10.         key="textField"/>
11.     <box></box>
12.     <textElement>
13.         <font size="24"/>
14.     </textElement>
15. <textFieldExpressior   class="java.lang.String"><![CDATA[$P{Mensaje}]]></textFieldExpressior>
16. .
17. .
18. .
```

Ejecutar el informe.

- Introducir el parámetro.



- Visualización de la ejecución.



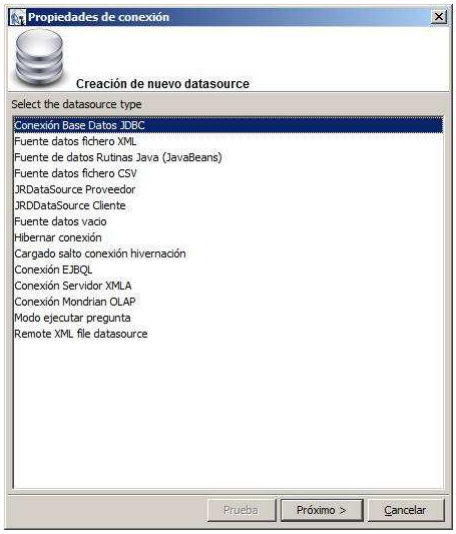
Entonces se mostrará el informe con los elementos que se han incorporado en la banda de título.

Paso 5 : Establecer conexión con la base de datos

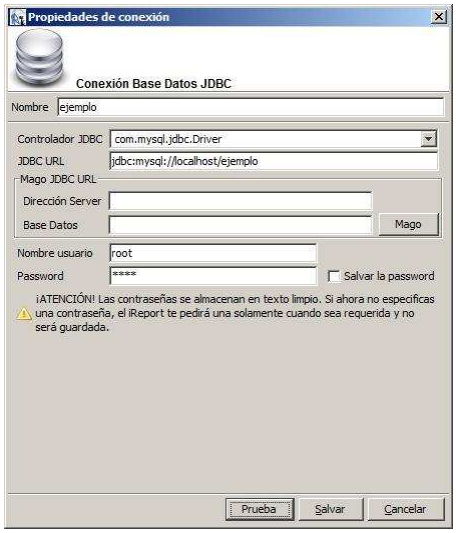
Seleccionar Datos -> Conexiones / Fuentes de datos



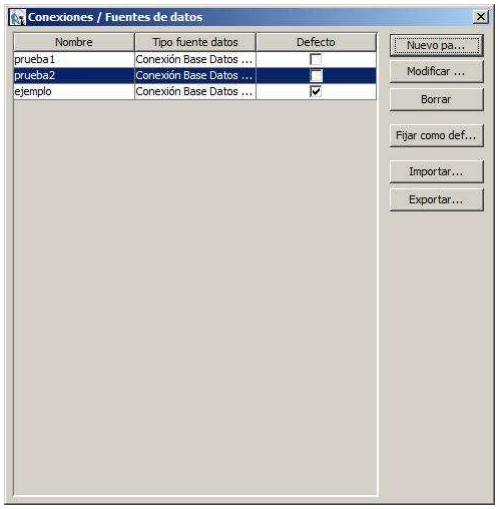
Crear una nueva conexión con las referencias al esquema donde se haya cargado el modelo.



Introducir los parámetros asociados al esquema en el que se ha creado el modelo de datos.



Verificar que se ha creado la conexión.



Con esto se tendrá todo preparado para ejecutar los siguientes casos.

5. Caso 1 : Ejecución de consulta de base de datos.

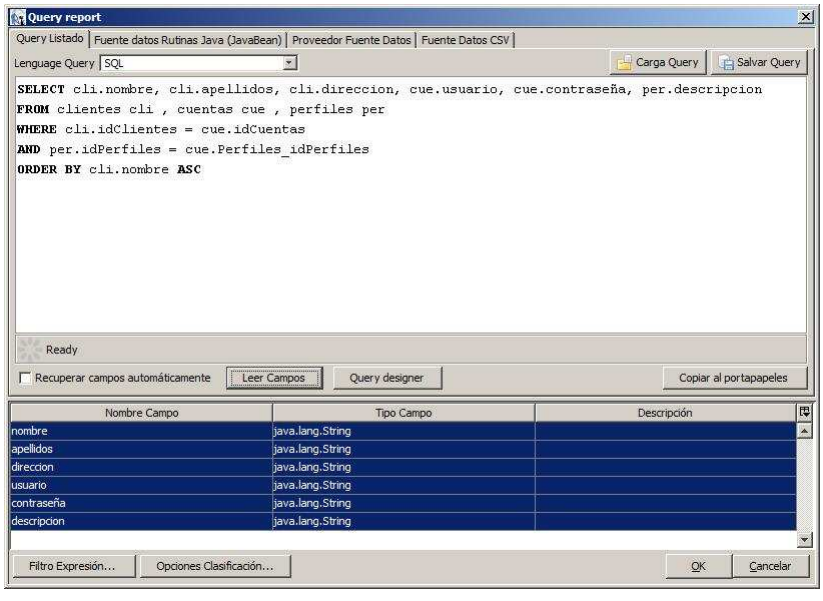
En este caso se ejecutará la consulta desde la opción de **Datos -> Consulta de informe**. Por lo que habrá que definir la consulta a realizar, en esta consulta se mostrará la información considerada de la tabla clientes en relación a las otras dos.

Consulta SQL que mostrará la información :

```
view plain print ?
01. SELECT cli.nombre, cli.apellidos, cli.direccion, cue.usuario, cue.contraseña, per.descripcion
02. FROM clientes cli , cuentas cue , perfiles per
03. WHERE cli.idClientes = cue.idCuentas
04. AND per.idPerfiles = cue.Perfiles_idPerfiles
05. ORDER BY cli.nombre ASC
```

Una vez introducida la consulta y como se puede observar en la imagen inferior los campos devueltos son reconocidos. Para ello se activará la opción de que recuperar los campos de forma automática si la consulta es identificada.

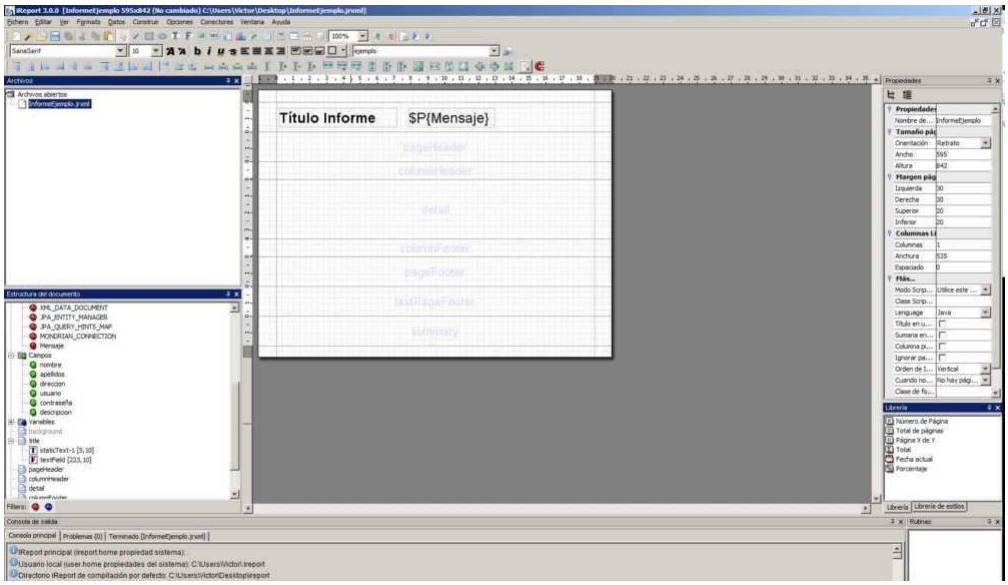
La tipología de los campos se puede cambiar desde la tabla de los campos recuperados.



Con este cambio modificará el fichero jrxml para incorporar la consulta (entre los parámetros y las bandas) :

```
view plain  print  ?
01.  .
02.  .
03.  .
04.  <queryString><![CDATA[
05.  SELECT cli.nombre, cli.apellidos, cli.direccion, cue.usuario, cue.contraseña, per.descripcion
06.  FROM clientes cli , cuentas cue , perfiles per
07.  WHERE cli.idClientes = cue.idCuentas
08.  AND per.idPerfiles = cue.Perfiles_idPerfiles
09.  ORDER BY cli.nombre ASC]]></queryString>
10.
11.  .
12.  .
13.  .
```

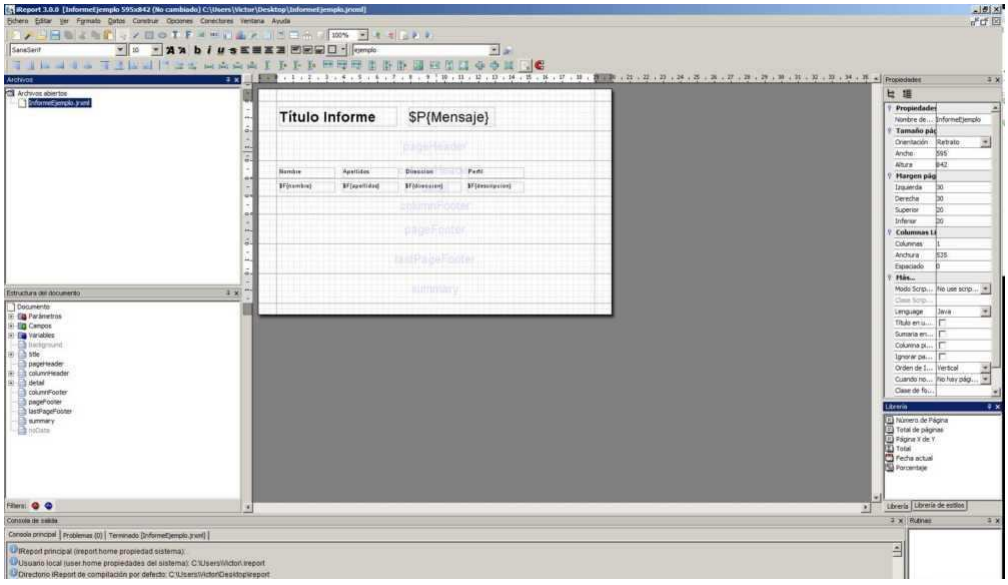
Se podrá visualizar los campos anteriores de la consulta en la ventana de la estructura del documento (Abajo a la izquierda).



Además se modificará el fichero jrxml para incorporar los campos y sus tipos :

```
view plain  print  ?
01.  .
02.  .
03.  .
04.      <field name="nombre" class="java.lang.String"/>
05.      <field name="apellidos" class="java.lang.String"/>
06.      <field name="direccion" class="java.lang.String"/>
07.      <field name="usuario" class="java.lang.String"/>
08.      <field name="contraseña" class="java.lang.String"/>
09.      <field name="descripcion" class="java.lang.String"/>
10.  .
11.  .
12.  .
```

Incoporamos los datos a mostrar en el informe. Para ello utilizaremos 2 bandas : cabecera y detalle.



En la banda de **cabecera** se introducirán los textos estáticos que darán nombre a las columnas. Por lo que se generará el siguiente código dentro del fichero jrxml.

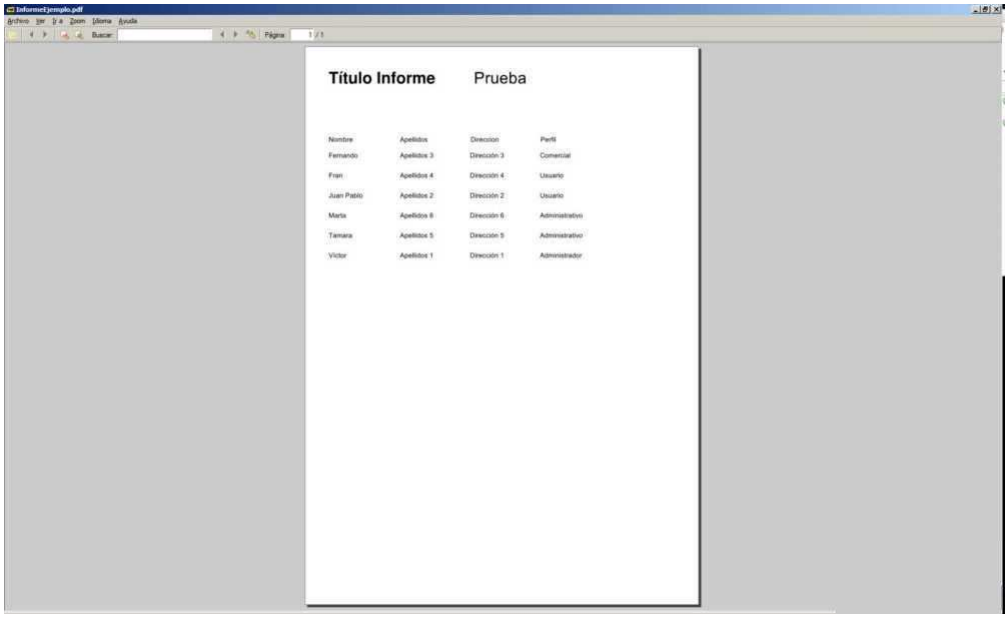
```
view plain  print  ?
01. .
02. .
03. .
04. <columnHeader>
05.     <band height="30"  isSplitAllowec="true" >
06.         <staticText>
07.             <reportElement
08.                 x="5"
09.                 y="12"
10.                 width="94"
11.                 height="18"
12.                 key="staticText-2"/>
13.             <box></box>
14.             <textElement>
15.                 <font/>
16.             </textElement>
17.         <text><![CDATA[Nombre]]></text>
18.         </staticText>
19.         <staticText>
20.             <reportElement
21.                 x="112"
22.                 y="12"
23.                 width="94"
24.                 height="18"
25.                 key="staticText-3"/>
26.             <box></box>
27.             <textElement>
28.                 <font/>
29.             </textElement>
30.         <text><![CDATA[Apellidos]]></text>
31.         </staticText>
32.         <staticText>
33.             <reportElement
34.                 x="218"
35.                 y="12"
36.                 width="94"
37.                 height="18"
38.                 key="staticText-4"/>
39.             <box></box>
40.             <textElement>
41.                 <font/>
42.             </textElement>
43.         <text><![CDATA[Direccion]]></text>
44.         </staticText>
45.         <staticText>
46.             <reportElement
47.                 x="323"
48.                 y="12"
49.                 width="94"
50.                 height="18"
51.                 key="staticText-5"/>
52.             <box></box>
53.             <textElement>
54.                 <font/>
55.             </textElement>
56.         <text><![CDATA[Perfil]]></text>
57.         </staticText>
58.     </band>
59. </columnHeader>
60. .
61. .
62. .
```

En la banda de **detalle** se introducirán los textos dinámicos asociados a los campos de la consulta. Formato : $F{NOMBRE_DEL_CAMPO}.

Se generará el siguiente código en el fichero jrxml.

```
view plain  print  ?
01. .
02. .
03. .
04. <detail>
05.     <band height="30"  isSplitAllowec="true" >
06.         <textField isStretchWithOverflow="false" isBlankWhenNull="false" evaluationTime="Now" hyperlinkType="None"  hyperlinkTarget="Self" >
07.             <reportElement
08.                 x="5"
09.                 y="6"
10.                 width="94"
11.                 height="18"
12.                 key="textField"/>
13.             <box></box>
14.             <textElement>
15.                 <font/>
16.             </textElement>
17.         <textFieldExpression   class="java.lang.String"><![CDATA[$F{nombre}]]></textFieldExpression>
18.         </textField>
19.         <textField isStretchWithOverflow="false" isBlankWhenNull="false" evaluationTime="Now" hyperlinkType="None"  hyperlinkTarget="Self" >
20.             <reportElement
21.                 x="112"
22.                 y="6"
23.                 width="94"
24.                 height="18"
25.                 key="textField"/>
26.             <box></box>
27.             <textElement>
28.                 <font/>
29.             </textElement>
30.         <textFieldExpression   class="java.lang.String"><![CDATA[$F{apellidos}]]></textFieldExpression>
31.         </textField>
32.         <textField isStretchWithOverflow="false" isBlankWhenNull="false" evaluationTime="Now" hyperlinkType="None"  hyperlinkTarget="Self" >
33.             <reportElement
34.                 x="217"
35.                 y="6"
36.                 width="95"
37.                 height="18"
38.                 key="textField"/>
39.             <box></box>
40.             <textElement>
41.                 <font/>
42.             </textElement>
43.         <textFieldExpression   class="java.lang.String"><![CDATA[$F{direccion}]]></textFieldExpression>
44.         </textField>
45.         <textField isStretchWithOverflow="false" isBlankWhenNull="false" evaluationTime="Now" hyperlinkType="None"  hyperlinkTarget="Self" >
46.             <reportElement
47.                 x="322"
48.                 y="6"
49.                 width="95"
50.                 height="18"
51.                 key="textField-1"/>
52.             <box></box>
53.             <textElement>
54.                 <font/>
55.             </textElement>
56.         <textFieldExpression   class="java.lang.String"><![CDATA[$F{descripcion}]]></textFieldExpression>
57.         </textField>
58.     </band>
59. </detail>
60. .
61. .
62. .
```

Tras ejecutar el informe haciendo uso de la conexión a la base de datos anterior se mostrarán todos los clientes.

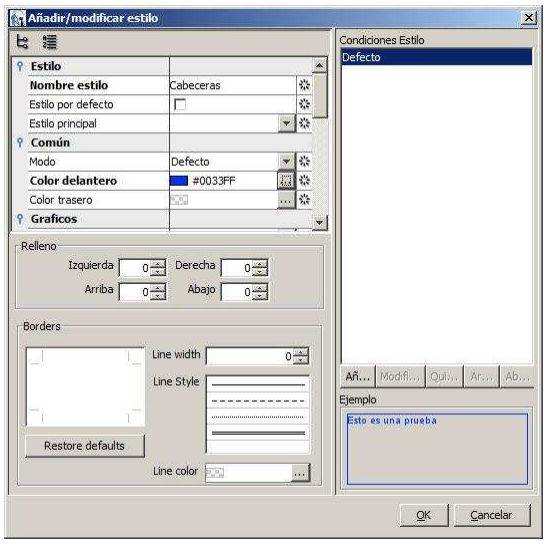


Anexo 1 : Definir un estilo con iReport.

Accederemos a los estilos desde **Fomato -> Estilos**.



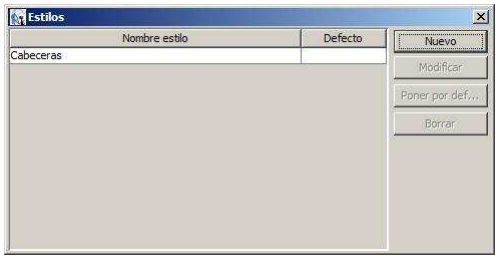
Generaremos un estilo sencillo para las cabeceras donde únicamente cambiaremos el color de la letra.Por lo que sólo introduciremos el nombre y el color elegido.

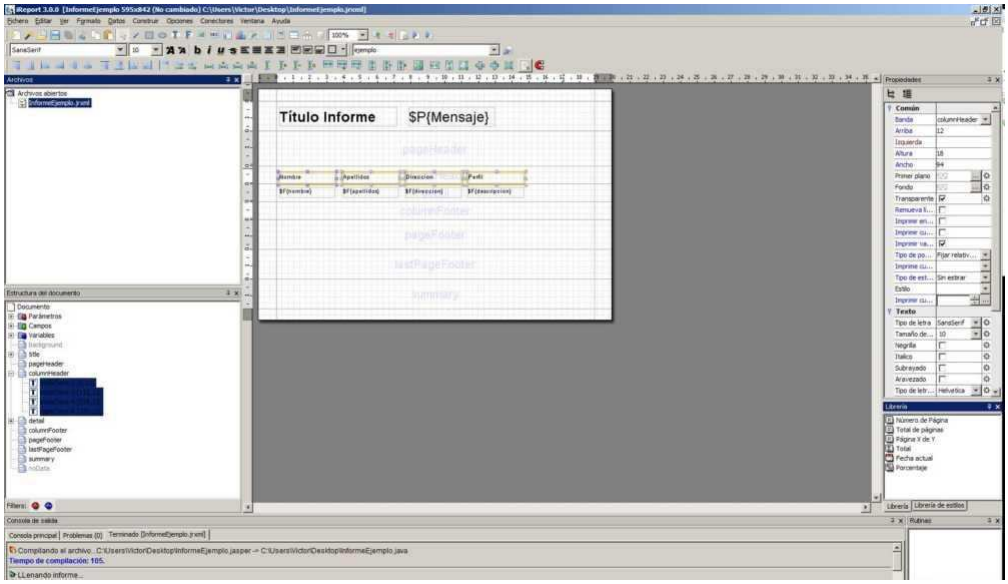


En el fichero jrxml se añadirá la siguiente línea (entre los import y los parámtros) :

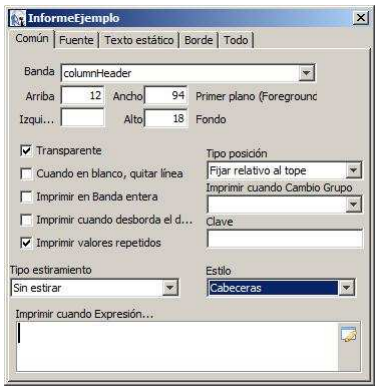
```
view plain  print  ?
01.  .
02.  .
03.  .
04.  <style
05.      name="Cabeceras"
06.      forecolor="#0033FF"
07.  >
08.  </style>
09.  .
10.  .
11.  .
```

Una vez creado el estilo seleccionaremos todas las cabeceras, aunque también se puede ir haciendo una a una.





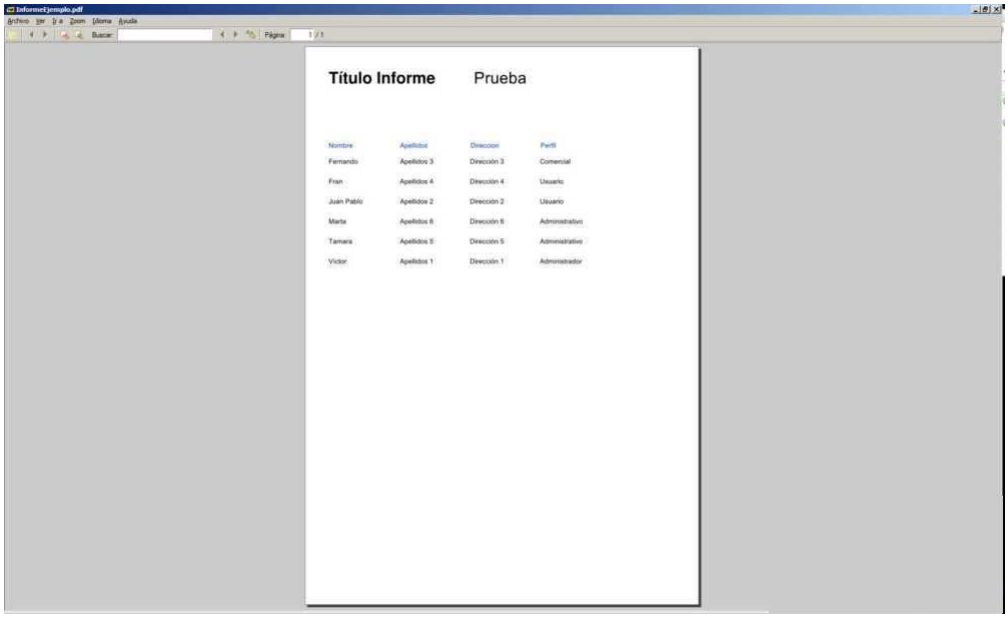
Una vez seleccionadas las cabeceras pulsaremos sobre el botón derecho y modificaremos el estilo para asociarlo a las cabeceras.



Por lo que se modificará el jrxml para los datos estáticos incorporando el estilo :

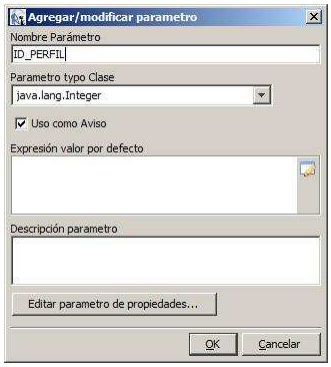
```
view plain  print  ?
01.  .
02.  .
03.  .
04.  <staticText>
05.      <reportElement
06.          style="Cabeceras"
07.          x="5"
08.          y="12"
09.          width="94"
10.          height="18"
11.          key="staticText-2"/>
12.      <box></box>
13.      <textElement>
14.          <font/>
15.      </textElement>
16.  <text><![CDATA[Nombre]]></text>
17.  </staticText>
18.
19.  .
20.  .
21.  .
```

Tras ejecutar el informe haciendo uso de la conexión a la base de datos anterior se mostrarán todos los clientes.Pero con el estilo de las cabeceras.



Anexo 2 : Indroducir un parámetro que filtrará la consulta.

Añadiremos un nuevo parámetro que se utilizará para filtrar la consulta.Utilizaremos de indentificador ID_PERFIL y será del tipo Integer.



Se incluirá la siguiente línea en el fichero jrxml (tras los import y el área de los estilos):

```
view plain print ?
01. .
02. .
03. .
04. <parameter name="ID_PERFIL" isForPrompting="true" class="java.lang.Integer"/>
05. .
06. .
07. .
```

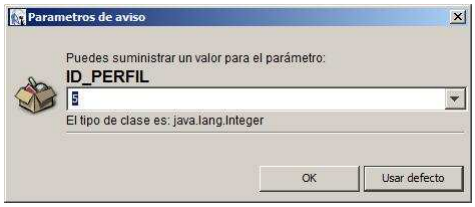
Cambiamos la consulta anterior para que incluya el nuevo filtro : AND per.idPerfiles = $P{ID_PERFIL}

```
view plain print ?
01. SELECT cli.nombre, cli.apellidos, cli.direccion, cue.usuario, cue.contraseña, per.descripcion
02. FROM clientes cli , cuentas cue , perfiles per
03. WHERE cli.idClientes = cue.idCuentas
04. AND per.idPerfiles = cue.Perfiles_idPerfiles
05. AND per.idPerfiles = $P{ID_PERFIL}
06. ORDER BY cli.nombre ASC
```

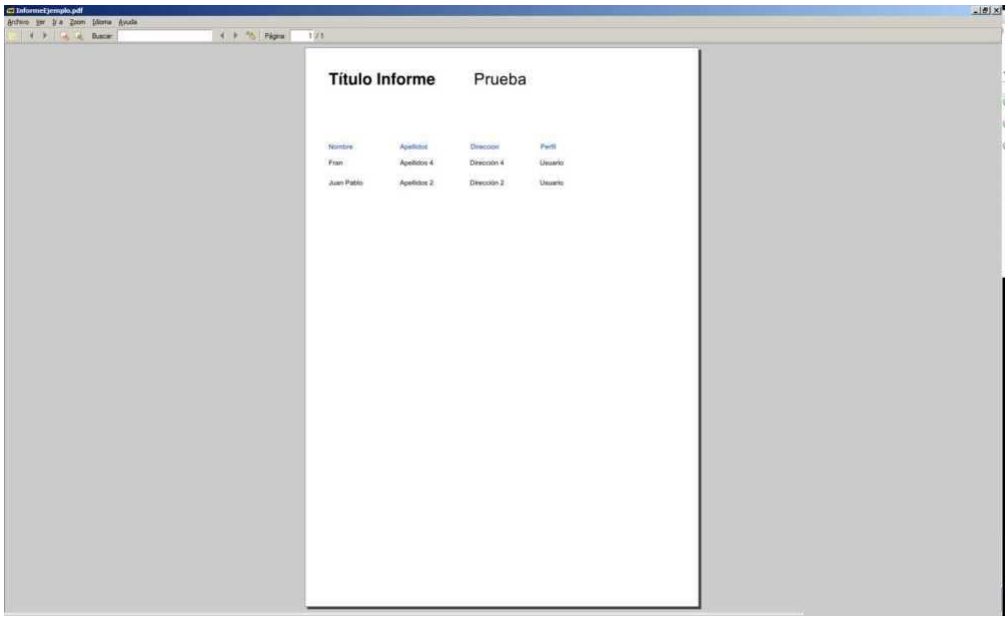
Se modificará el fichero jrxml :

```
view plain print ?
01. .
02. .
03. .
04. <queryString><![CDATA[
05. SELECT cli.nombre, cli.apellidos, cli.direccion, cue.usuario, cue.contraseña, per.descripcion
06. FROM clientes cli , cuentas cue , perfiles per
07. WHERE cli.idClientes = cue.idCuentas
08. AND per.idPerfiles = cue.Perfiles_idPerfiles
09. AND per.idPerfiles = $P{ID_PERFIL}
10. ORDER BY cli.nombre ASC]]></queryString>
11. .
12. .
13. .
```

Tras ejecutar el informe haciendo uso de la conexión a la base de datos anterior se solicitará obtener el nuevo parámetro.



Una vez introducido se generará el siguiente informe:



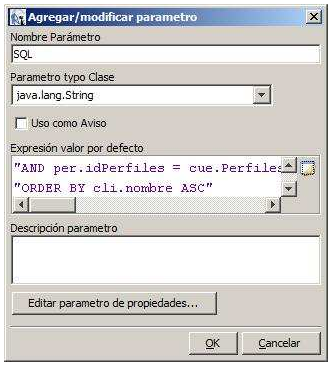
5. Caso 2 : Ejecución de consulta dinámica de base de datos.

Nota : Para crear este ejemplo se aconseja haber realizado el caso 1 (ya que nos basaremos en muchos de sus elementos definidos).

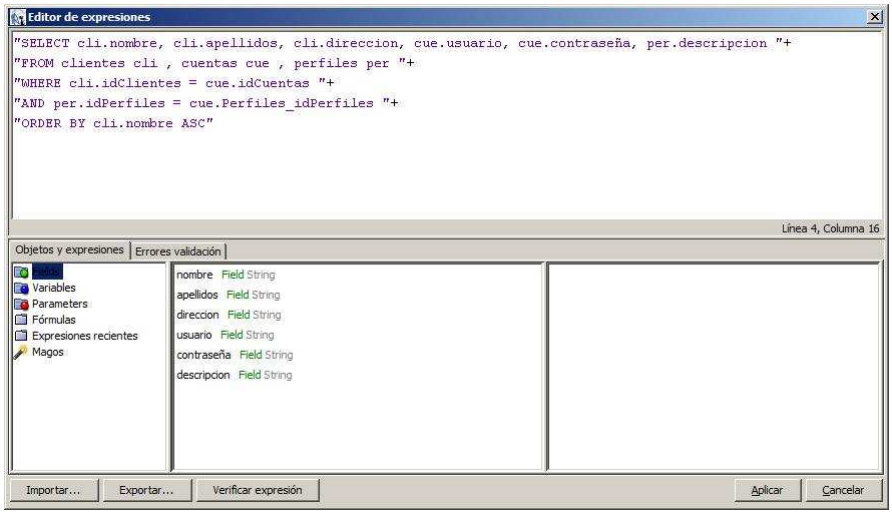
Eliminar el parámetro ID_PERFIL.

En este caso se ejecutará la consulta desde la opción de **Datos -> Consulta de informe** pero de una forma un poco especial.Para ello se generará un nuevo parámetro que contendrá la consulta, lo que nos permitirá ganar en cierta forma dinamismo sobre nuestras consultas.

Generaremos un nuevo parámetro al que denominaremos SQL. Este parámetro tendrá la peculiaridad que será del tipo String y que tendrá como valor por defecto la consulta a ejecutar.(OJO : la consulta que mostraba todos los clientes).



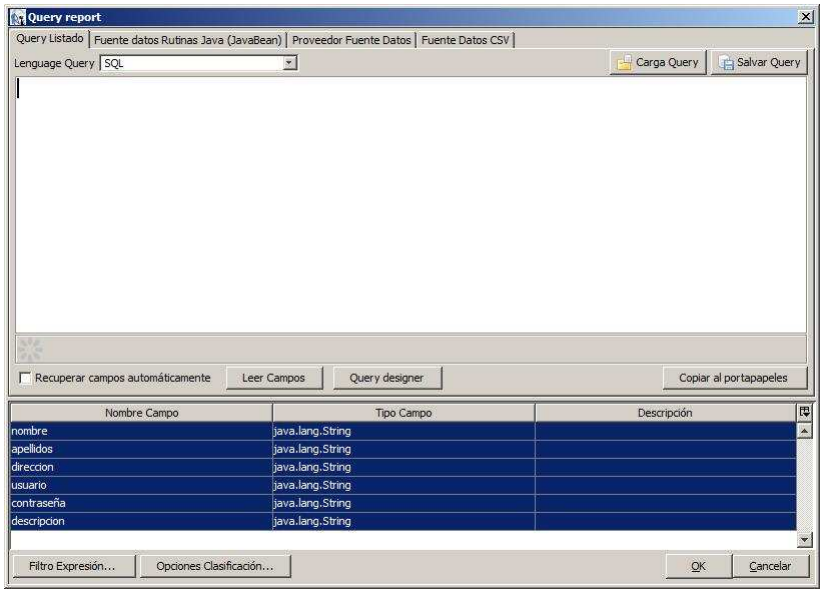
Para ello editaremos la consulta anterior de la siguiente forma y la copiaremos como valor por defecto del parámetro :



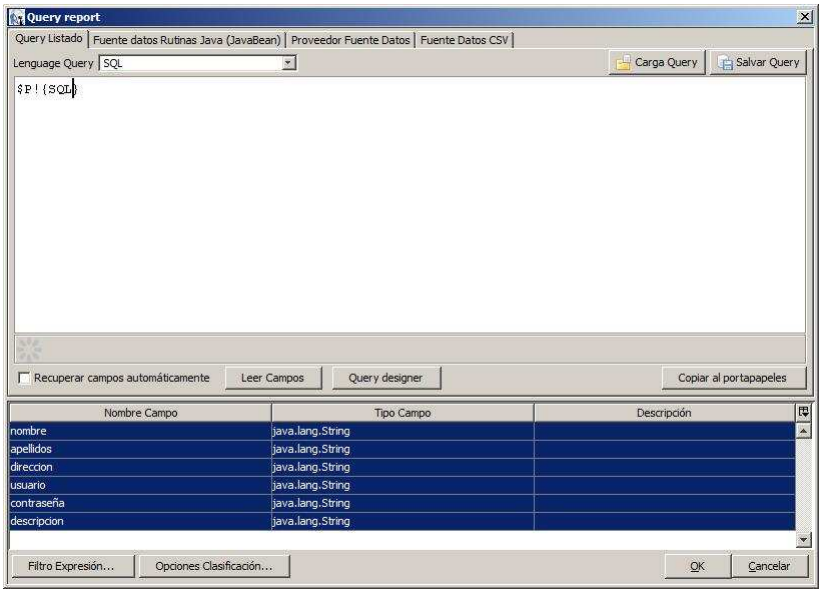
Por lo que se generará en el jrxml el parámetro :

```
view plain print ?
01. .
02. .
03. .
04. <parameter name="SQL" isForPrompting="false" class="java.lang.String">
05.         <defaultValueExpressior ><![CDATA["SELECT cli.nombre, cli.apellidos, cli.direccion, cue.usuario, cue.contraseña, per.descripcion "+
06. "FROM clientes cli , cuentas cue , perfiles per "+
07. "WHERE cli.idClientes = cue.idCuentas "+
08. "AND per.idPerfiles = cue.Perfiles_idPerfiles "+
09. "ORDER BY cli.nombre ASC"]]></defaultValueExpressior>
10. </parameter>
11. .
12. .
13. .
```

Eliminaremos la consulta.



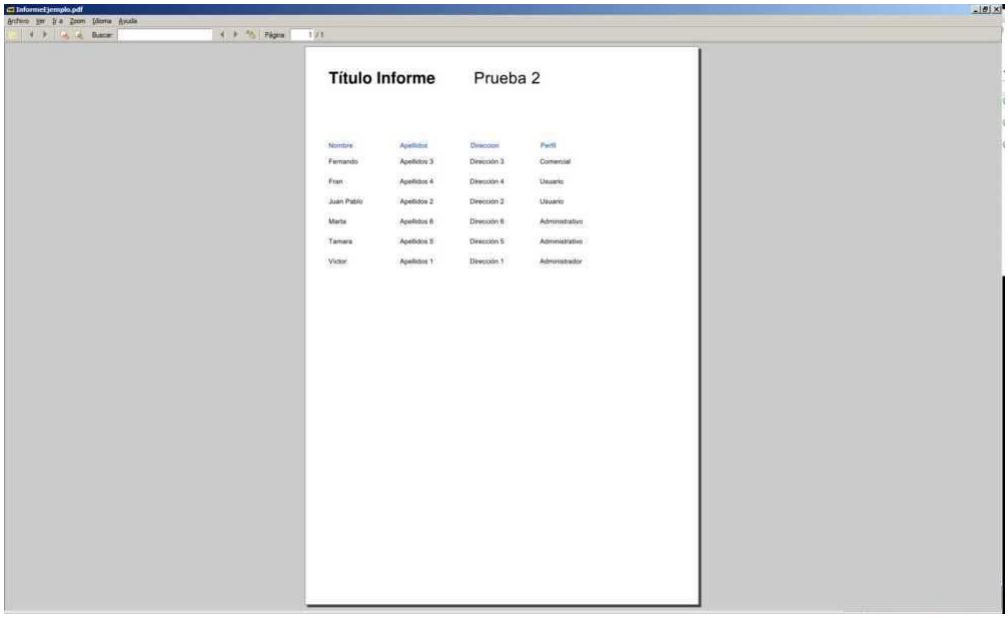
E introduciremos la llamada al parámetro SQL.



Por lo que se modificará en el jrxml el elemento querystring.

```
view plain  print  ?
01.  .
02.  .
03.  .
04.  <parameter name="SQL" isForPrompting="false" class="java.lang.String">
05.      <defaultValueExpressior ><![CDATA["SELECT cli.nombre, cli.apellidos, cli.direccion, cue.usuario, cue.contraseña, per.descripcion "+
06.  "FROM clientes cli , cuentas cue , perfiles per "+
07.  "WHERE cli.idClientes = cue.idCuentas "+
08.  "AND per.idPerfiles = cue.Perfiles_idPerfiles "+
09.  "ORDER BY cli.nombre ASC"]]></defaultValueExpressior>
10.  </parameter>
11.  <queryString><![CDATA[$P!{SQL}]]></queryString>
12.  .
13.  .
14.  .
```

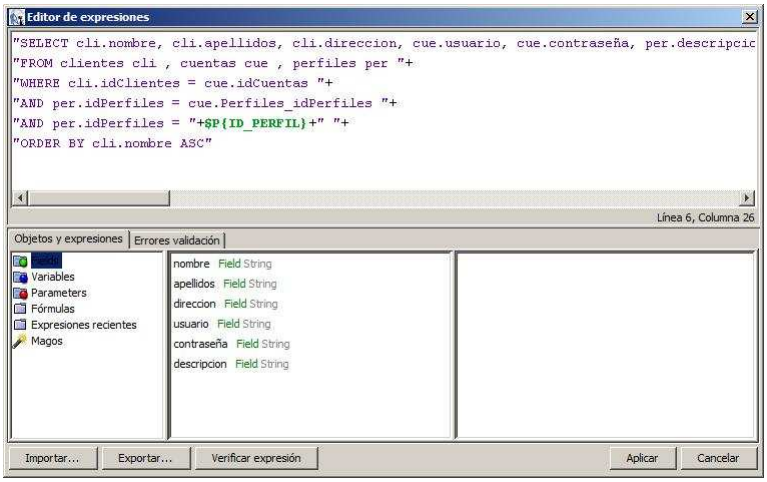
Tras ejecutar el informe haciendo uso de la conexión a la base de datos anterior se mostrarán todos los clientes, como en el caso 1.



Anexo 1 : Indroducir un parámetro que filtrará la consulta.

El problema que trata de solucionar este tutorial viene en esta parte.

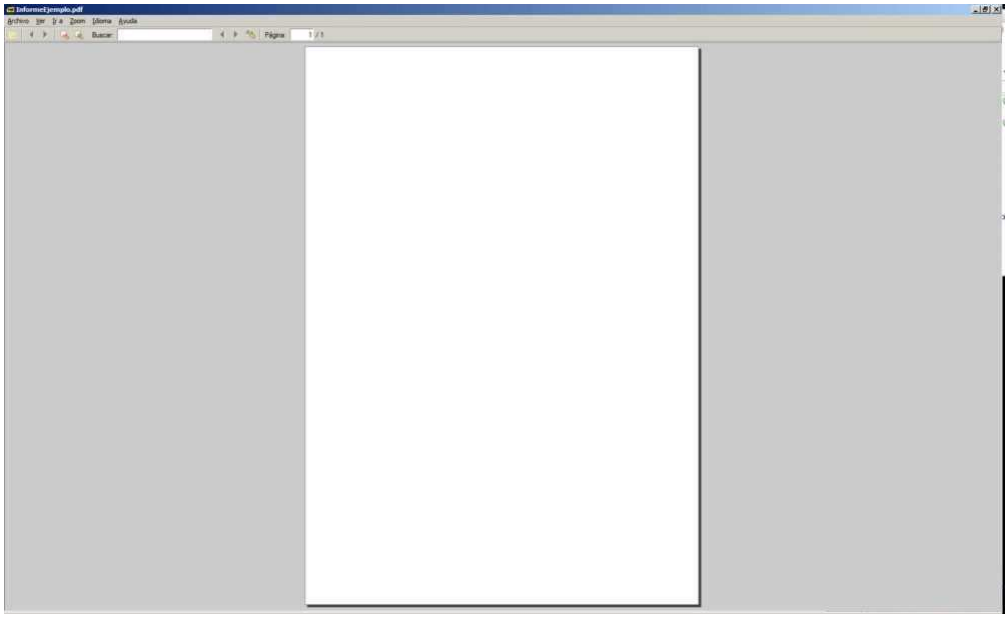
Si volvemos a crear el parámetro para filtrar la consulta como en el caso 1 y modificamos la consulta para que acepte el parámetro.



Se producirá la siguiente modificación en el jrxml.

```
view plain  print  ?
01.  .
02.  .
03.  .
04.      <parameter name="SQL" isForPrompting="false" class="java.lang.String">
05.          <defaultValueExpressior ><![CDATA["SELECT cli.nombre, cli.apellidos, cli.direccion, cue.usuario, cue.contraseña, per.descripcion "+
06.  "FROM clientes cli , cuentas cue , perfiles per "+
07.  "WHERE cli.idClientes = cue.idCuentas "+
08.  "AND per.idPerfiles = cue.Perfiles_idPerfiles "+
09.  "AND per.idPerfiles = "+$P{ID_PERFIL}+" "+
10.  "ORDER BY cli.nombre ASC"]]></defaultValueExpressior>
11.      </parameter>
12.      <parameter name="ID_PERFIL" isForPrompting="true" class="java.lang.Integer"/>
13.      <queryString><![CDATA[$P!{SQL}]]></queryString>
14.  .
15.  .
16.  .
```

El problema radica en que se ha definido un parámetro depués de uno que hace uso del que que acaba de incoporar por lo que al ejecutarlo se mostrará el siguiente informe.

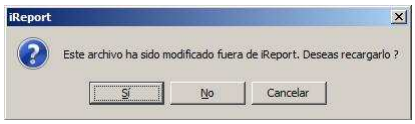


No se produce ningun mensaje de error y simplemente muestra el informe en blanco.

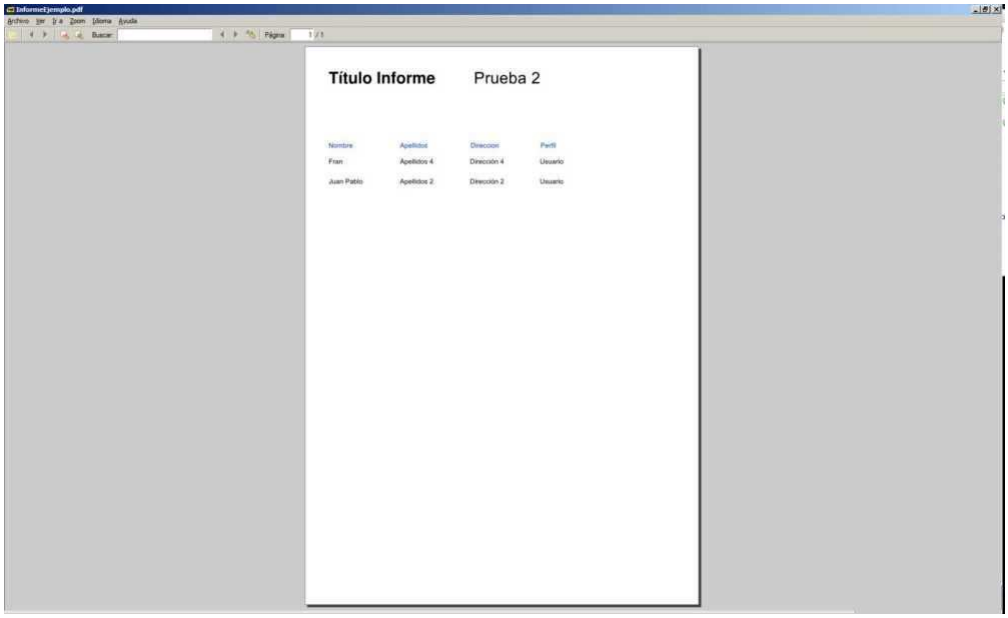
Esto se puede solucionar borrandon los parámetros y declarandolos con el orden adecuado o bien acceder al jrxml y modificar su orden

```
view plain  print  ?
01.  .
02.  .
03.  .
04.  <parameter name="ID_PERFIL" isForPrompting="true" class="java.lang.Integer"/>
05.      <parameter name="SQL" isForPrompting="false" class="java.lang.String">
06.          <defaultValueExpressior ><![CDATA["SELECT cli.nombre, cli.apellidos, cli.direccion, cue.usuario, cue.contraseña, per.descripcion "+
07.  "FROM clientes cli , cuentas cue , perfiles per "+
08.  "WHERE cli.idClientes = cue.idCuentas "+
09.  "AND per.idPerfiles = cue.Perfiles_idPerfiles "+
10.  "AND per.idPerfiles = "+$P{ID_PERFIL}+" "+
11.  "ORDER BY cli.nombre ASC"]]></defaultValueExpressior>
12.      </parameter>
13.
14.      <queryString><![CDATA[$P!{SQL}]]></queryString>
15.  .
16.  .
17.  .
```

Una vez modificado el informe desde el jrxml, el iReport nos pedirá actualizarlo y ya podremos ejecutarlo correctamente



Tras ejecutarlo desde la conexión a la base de datos veremos que muestra la información correctamente



6. Conclusiones.

Espero haberos ayudado a resolver este error y demostrar que NO ES MÁGICO como podría parecer de primeras, sino que se debe a un problema en el orden de declaración de parámetros dinámicos. Por otro lado, espero haberos introducido en el maravilloso mundo de "modificar" los ficheros de informe a mano (esta parte sólo se la aconsejo a los más puristas ;-)), ya que no es algo que se suela hacer a diario o que la gente trabaje con ellos directamente.

Un saludo.

Víctor

mailto:vjmadrid@autentia.com

¿Qué te ha parecido el tutorial? Déjanos saber tu opinión y ¡vota!

Muy malo Malo Regular Bueno Muy bueno

Votar

(Sólo para usuarios registrados)

» Registrate y accede a esta y otras ventajas «

Anímate y coméntanos lo que pienses sobre este tutorial

Puedes opinar o comentar cualquier sugerencia que quieras comunicarnos sobre este tutorial; con tu ayuda, podemos ofrecerte un mejor servicio.

Enviar comentario

(Sólo para usuarios registrados)

» Registrate y accede a esta y otras ventajas «

Autor **Mensaje de usuario registrado**

- Puedes inscribirte en nuestro servicio de notificaciones haciendo clic aquí.
- Puedes firmar en nuestro libro de visitas haciendo clic aquí.
- Puedes asociarte al grupo AdictosAlTrabajo en XING haciendo clic aquí.
- Añadir a favoritos Technorati. ADD THIS BLOG TO MY Technorati FAVORITES

Creative Commons License Esta obra está licenciada bajo licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5

Recuerda

Autentia te regala la mayoría del conocimiento aquí compartido (Ver todos los tutoriales). Somos expertos en: J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ... y muchas otras cosas.

¿Nos vas a tener en cuenta cuando necesites consultoría o formación en tu empresa?, ¿Vas a ser tan generoso con nosotros como lo tratamos de ser con vosotros?

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos ...

Autentia = Soporte a Desarrollo & Formación.

info@autentia.com

Formación en nuevas tecnologías

Tutoriales recomendadosxx

Nombre	Resumen	Fecha	Visitas	Valoración	Votos	Pdf
iReport : Solución al problema de los parámetros dinámicos.	En este tutorial acerca de iReport, tratamos la solución al problema de los parámetros dinámicos e introducción a los ficheros "jrxml"	2009-09-28	5	-	-	
Ordenación por cantidades en informe cruzado	Nico nos explica en ese tutorial cómo lograr ordenar por cantidades en informes cruzados usando JasperReports e iReport	2009-08-26	655	-	-	
Joomla 1.5. Instalación y configuración	Veamos en detalle cómo instalar Joomla 1.5 y aplicar algunas configuraciones posteriores de las disponibles en este CMS	2009-08-14	2624	Bueno	2	
Hibernate Search, Bridges, Analizadores y más	Detalles del uso de Hibernate Search con bases de datos Lucene	2009-07-29	1140	-	-	
DBUnit-Exportar e Importar BBDD	DBUnit como complemento de los test unitarios con carga a una base de datos	2009-07-06	1479	-	-	
Informes Crosstab con iReport	Con este tutorial vamos a ampliar el nivel de conocimiento sobre iReport enseñando como hacer un informe usando crosstab o tablas dinámicas	2009-04-16	4847	-	-	
Registro de Web Services con Apache jUDDI. Configuración y ejemplo	Veamos como podemos catalogar y buscar web services bajo la especificación UDDI (Universal Description, Discovery and Integration)	2009-04-13	3062	-	-	
JasperReports Maven Plugin	JasperReports Maven Plugin es un plugin de Maven que nos permite compilar nuestros informes JasperReports.	2009-03-20	3234	-	-	
Vistas materializadas	En este tutorial vamos a ver lo que son las vistas materializadas (materialized views) y vamos a hacer un ejemplo de creación de una vista materializada y su posterior uso	2009-02-26	3812	-	-	
Maven JXR Plugin: publica el código fuente en el site	JXR es un plugin para maven, de tipo reporting, que genera en el site un informe con los fuentes java de tu aplicación: "Source Xref".	2009-02-26	1958	-	-	

Nota:

Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento. Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores. En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo. Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.