

# Creación de informes

Guía paso a paso para crear informes con Tibco Jasper Studio y Jasper en Netbeans.

## Creación de un informe sencillo

Para explicar la creación de informes vamos a utilizar un ejemplo sencillo, que muestre un informe de productos.

La generación de informes desde nuestra aplicación se divide en dos fases:

- **Fase 1.** Crear el archivo informe, que es un archivo XML con extensión jrxml.
- **Fase 2.** Integrar el archivo creado en nuestra aplicación.

## Fase 1. Creación del archivo informe

Existen varias formas de crear el informe, incluso podemos hacerlo a mano, pero esto nos llevaría mucho tiempo.

Podemos destacar 2 aplicaciones para crear el informe:

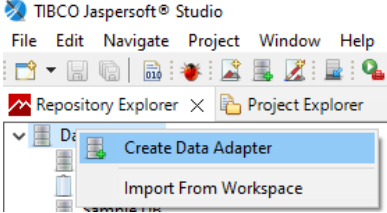
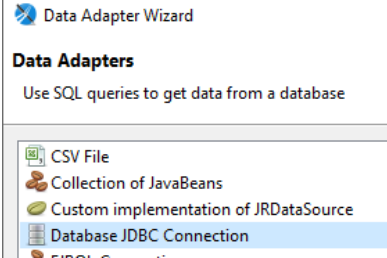
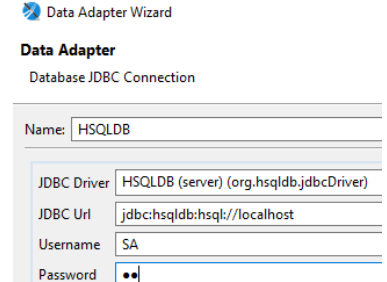
- iReport: tiene versión aplicación y un plug-in para integrarlo en NetBeans.
- Tibco Jasper Studio.

En nuestro caso vamos a utilizar la última, Tibco Jasper Studio.

### 1.0. Conexión a la base de datos

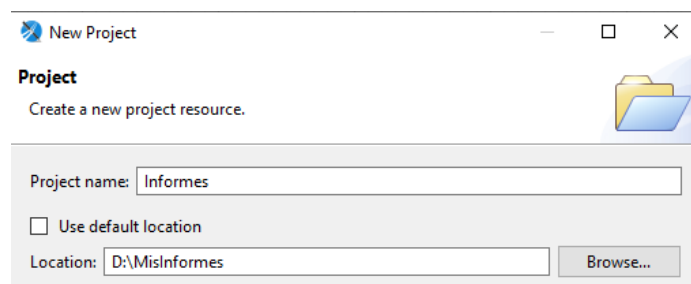
Como paso previo a la creación de un informe, debemos tener una conexión a la base de datos. Si esta conexión no existe, debemos crearla.

Cuando creamos una nueva conexión, nos aparecerá un asistente que nos solicitará los datos para cerrarla. Los datos dependerán de la base de datos que vayamos a utilizar. En nuestro caso vamos a conectarnos a HSQLDB.

		
Pulsamos "Create Data Adapter"	Seleccionamos Database JDBC....	Seleccionamos HSQLDB (Server) y ponemos el resto de información (nombre de la conexión, usuario, etc.)

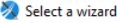
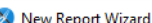
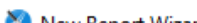
### 1.1. Creación de un nuevo proyecto


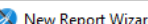
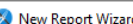
En primer lugar, para tener ordenados los informes que estemos creando para un proyecto, es conveniente crear un proyecto nuevo.



## 1.2. Creación del archivo informe

La creación de un informe se hace a través de un asistente que nos irá pidiendo datos:

 <p><b>Select a wizard</b> Create a new JasperReport</p> <p>Wizards:</p> <p>Type filter text</p> <ul style="list-style-type: none"> <li>General</li> <li>Eclipse Modeling Framework</li> <li>Git</li> <li>Jaspersoft Studio <ul style="list-style-type: none"> <li>Chart Themes</li> <li>Custom Visualization Component</li> <li>Data Adapter</li> <li>Functions Library</li> <li>Jasper Report</li> <li>JasperReports Fonts Extension</li> <li>JasperReports Project</li> <li>JasperReports Samples</li> </ul> </li> </ul>	 <p><b>New Report Wizard</b></p> <p><b>Report Templates</b> Please select a report from a template</p> <p>Categories</p> <ul style="list-style-type: none"> <li>All</li> <li>Print Ready Reports</li> <li>Web Interactive Reports</li> <li>Format A4</li> <li>Format Landscape</li> <li>Report Books</li> </ul> <p>Coffee</p> <p>Coffee Landscape</p> <p>Flower Title</p>	 <p><b>New Report Wizard</b></p> <p><b>Report file</b> Please select your reports file name</p> <p>Enter or select the parent folder:</p> <p>Informes</p> <p>Informes</p> <p>MyReports</p> <p>File name: Productos.jrxml</p>
<p>Para crear un informe debemos crear un "Jasper Report".</p>	<p>Elegimos la plantilla que se ajuste a nuestras necesidades.</p>	<p>Ponemos un nombre a nuestro archivo.</p>

 <p><b>New Report Wizard</b></p> <p><b>Data Source</b> Select a Data Source and design the query.</p> <p>Data Adapter</p> <p>One Empty Record - Empty rows</p> <p>hsqldb - Database JDBC Connection</p> <p>One Empty Record - Empty rows</p> <p>Sample DB - Database JDBC Connection</p>	 <p><b>New Report Wizard</b></p> <p><b>Data Source</b> Select a Data Source and design the query.</p> <p>Data Adapter</p> <p>hsqldb - Database JDBC Connection</p> <p>INFORMATION_SCHEMA</p> <p>PUBLIC (CURRENT)</p> <p>SYSTEM_LOBS</p> <p>1 SELECT *</p> <p>2 FROM productos</p> <p>3 ORDER BY nombre</p>	 <p><b>New Report Wizard</b></p> <p><b>Fields</b> Please select dataset fields</p> <p>Dataset Fields</p> <p>DESCRIPCION</p> <p>PRECIOUNI</p> <p>UNIDEXIST</p> <p>UNIDAVISO</p> <p>SUSPENDIDO</p> <p>CODFAB</p> <p>Fields</p> <p>CODIGO</p> <p>NOMBRE</p>
<p>Elegimos la conexión a nuestra base de datos. (Paso 1.0)</p>	<p>Escribimos la sentencia SQL de consulta a nuestra base de datos y que va a servir para generar informes.</p>	<p>Elegimos los campos que se van a utilizar en nuestro informe.</p>

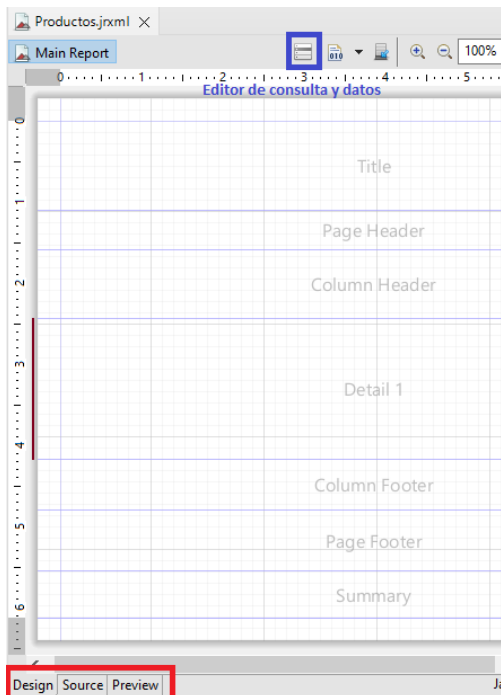
 <p><b>New Report Wizard</b></p> <p><b>Group By</b> Please select fields to Group By</p> <p>Dataset Fields</p> <p>CODIGO</p> <p>NOMBRE</p> <p>Fields</p>	<p>Si queremos, podemos agrupar los registros por algún campo. Esto se puede hacer más adelante.</p>
---	--

La información que hemos introducido desde la fuente de datos podemos modificarla posteriormente, de forma que podemos modificar la consulta SQL, modificar los campos que se van a utilizar en el informe y, también, como ya se ha mencionado, podemos agrupar los registros del resultado por algún otro campo.

Con esto ya tendremos un archivo jrxml y lo siguiente será diseñar el informe.

### 1.3. Creación de un informe sencillo

En este punto vamos a utilizar un ejemplo sencillo para enseñar la forma en la que vamos a acceder a los campos de los registros que ha generado una consulta SQL y en qué lugar debemos colocarlos.



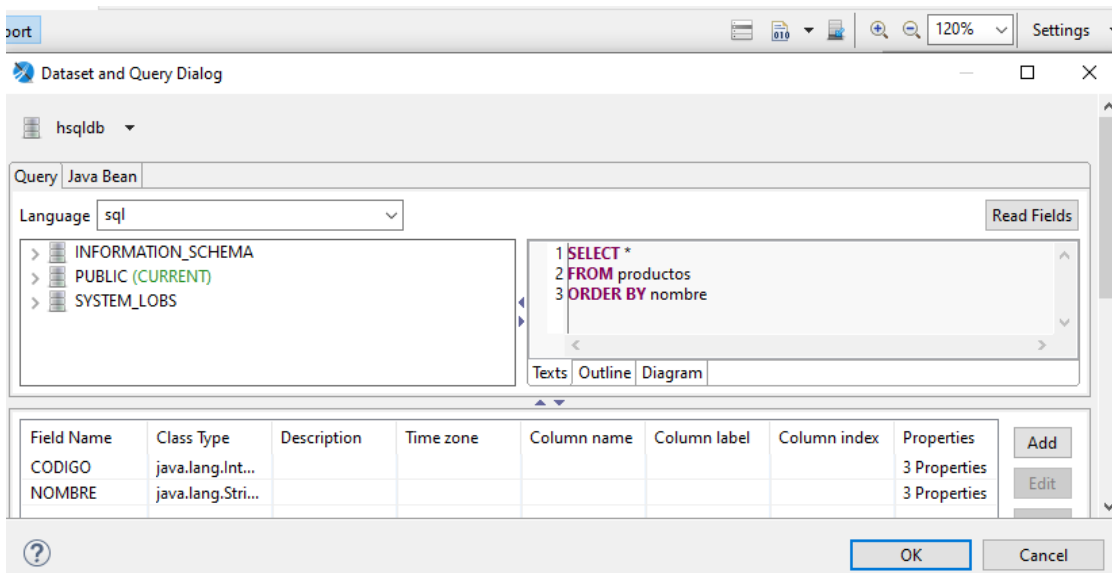
Cada archivo jrxml va a tener asociado tres vistas a las que se accede a través de tres pestañas situadas en la parte inferior derecha de la imagen (recuadro rojo):

Vista de diseño: esta vista nos permite diseñar el informe.

Vista fuente: esta vista contiene el XML que se genera desde la vista de diseño.

Vista previa: en esta vista podemos ver el resultado del informe cruzando nuestro archivo con la consulta realizada a la base de datos.

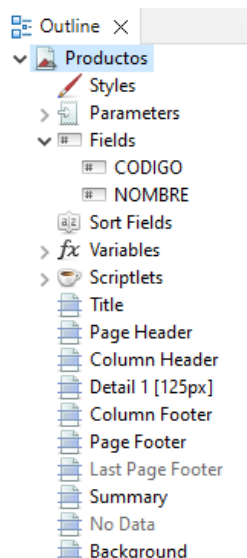
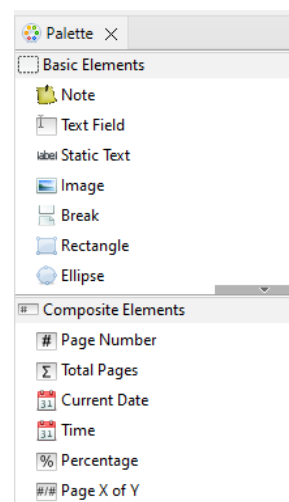
En recuadro azul está el acceso al Editor de consultas y datos desde el cual podemos modificar nuestra consulta SQL, crear parámetros, etc.



En la vista de diseño vemos que aparecen una serie de secciones o bandas. Cada una de ellas tiene un propósito en nuestro informe:

Banda	Descripción
Title	Título del informe. Sólo se muestra en la primera página del mismo.
Page Header	Cabecera de página que puede contener información adicional del informe.
Column Header	En esta sección se muestran los nombres de los campos a presentar (cabecera de los campos).
Detail	Esta sección contendrá los valores obtenidos de la base de datos.
Column Footer	Puede presentar valores totales para algún campo de la sección "detail"
Page Footer	En la parte inferior de cada página para poner fecha, número de página, etc.
Summary	Esta sección se usa para proporcionar información resumida de los campos presentes en la sección "detalle".

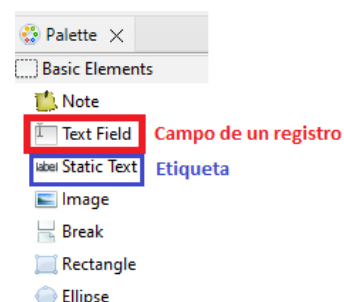
En el lado derecho de la interfaz de la aplicación se encuentra la paleta de componentes que podemos utilizar para diseñar nuestro informe. Esta paleta está dividida en dos secciones, la primera con elementos básicos como etiqueta, cuadro de texto, etc. y la de elementos compuestos que nos permiten añadir a nuestro formulario el número de página, la fecha actual, etc.



En el lado izquierdo de la interfaz de la aplicación tenemos el panel del esquema que contiene las bandas del informe, los campos de nuestra consulta, variables, parámetros, etc.

Dependiendo del informe que vayamos a realizar, es posible que haya bandas que no se van a utilizar, por ejemplo, la banda resumen ("Summary") se utiliza para poner diagramas. Si nuestro informe no fuera a llevar ningún diagrama, podríamos eliminar esta banda sin problemas.

Si queremos hacer un informe que muestre los productos que hay en la base de datos, además de escribir la consulta SQL correspondiente y elegir los campos que se van a mostrar, tenemos que añadir a nuestro informe el componente que va a escribir los campos en el informe. Este componente es el Text Field que se encuentra en la paleta de componentes en el área de Elementos básicos (recuadro rojo). También cabe destacar el elemento Label, que nos permite escribir texto en el informe, como por ejemplo el nombre de una columna.



Código	Nombre del producto
\$F(CODIGO)	\$F(NOMBRE)

Los campos de los registros que se van a mostrar en el informe deben colocarse en la banda Detalle y el elemento que debe utilizarse es el "Text Field". Este elemento debe tener el texto:

`$F{<nombre_campo>}` Recordemos que en Java se distingue entre mayúsculas y minúsculas, por lo que el nombre del

campo debe escribirse igual que el que tenemos en la consulta.

En la banda de Cabecera de columna hemos colocado Labels con el nombre del campo de cada columna del informe.

También podemos añadir los campos de los registros desde el panel izquierdo, arrastrando el componente que ya tiene el nombre del campo.

Si vamos a la Vista previa veremos el resultado:

Con esto ya tendríamos un archivo de informe sencillo que podríamos generar desde nuestra aplicación java.

Una vez creado el archivo de informe, pondremos este dentro de nuestro proyecto de nuestra aplicación y procederemos a integrarlo utilizando las bibliotecas de Jasper.

Código	Nombre del producto
45	Arenque ahumado
30	Arenque blanco del
46	Arenque salado
44	Azúcar negra Malacca
68	Barras de pan de
21	Bollos de Sir
64	Bollos de pan de

## Fase 2. Integración del archivo informe

Una vez que ya tenemos nuestro archivo de informe, vamos a proceder a escribir el código necesario para que desde nuestra aplicación se genere el informe al igual que pasaba al ir a la Vista previa en Tibco Jasper Studio.

Esta segunda fase se divide en 3 pasos. Para poder generar un informe necesitamos:

- Paso 1: Conectarnos a nuestra base de datos.
- Paso 2: Crear un vínculo del archivo del informe.
- Paso 3: Generar el informe con Jasper.

Hay que tener en cuenta que vamos a partir desde cero en este ejemplo. Si ya tuviéramos los pasos 1 y 2 realizados, solamente sería necesario realizar el paso 3.

### Paso 1. Conectarse a la base de datos

En primer lugar debemos conectarnos a la base de datos. Para ello es necesario importar las siguientes librerías:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
```

El código para conectarnos es el siguiente:

Antes de nada debemos cargar el controlador

```
Class.forName("org.hsqldb.jdbcDriver");
```

Una vez registrado el controlador, ya podemos establecer la conexión con la base de datos:

```
Connection conexion = DriverManager.getConnection ("jdbc:hsqldb:hsqldb://localhost", "SA", "SA");
```

El código resultante es el siguiente:

```
Connection conexion = null;
try {
    Class.forName("org.hsqldb.jdbcDriver");
    conexion = DriverManager.getConnection("jdbc:hsqldb:hsqldb://localhost", "SA", "SA");
} catch (ClassNotFoundException ex) {
    Logger.getLogger(principal.class.getName()).log(Level.SEVERE, null, ex);
} catch (SQLException ex) {
    Logger.getLogger(principal.class.getName()).log(Level.SEVERE, null, ex);
}
```

## Paso 2. Establecer un vínculo del archivo jrxml

En este paso debemos establecer un vínculo de lectura con el archivo del informe. Para hacer esto es necesario importar las siguientes bibliotecas:

```
import java.io.File;
import java.io.FileInputStream;
import java.io.InputStream;
import java.io.IOException;
```

Para establecer el vínculo con el archivo utilizaremos la clase InputStream. El archivo hay que tratarlo como si fuera un recurso (getClass().getResourceAsStream).

En Maven éste ejemplo el archivo lo hemos situado en el directorio informes que se encuentra dentro del directorio de "recurso":

```
InputStream vin_arch = getClass().getResourceAsStream("/informes/clientes01.jrxml");
```

El código resultante es el siguiente:

```
InputStream vin_arch = null;
try {
    vin_arch = getClass().getResourceAsStream("/informes/clientes01.jrxml");
} catch (IOException ex) {
    Logger.getLogger(principal.class.getName()).log(Level.SEVERE, null, ex);
}
```

## Paso 3. Generar el informe con Jasper

Para utilizar las siguientes clases de Jasper:

```
import net.sf.jasperreports.engine.JRException;
import net.sf.jasperreports.engine.JasperCompileManager;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperReport;
import net.sf.jasperreports.view.JasperViewer;
```

Para generar el informe es necesario escribir el siguiente código en nuestra clase:

```
//Compila el archivo jrxml
JasperReport jr = JasperCompileManager.compileReport(vin_arch);
```

```
//Carga los datos en el informe desde la BBDD
JasperPrint jasperPrint = JasperFillManager.fillReport(jr, null, conexion);
```

```
//Visualiza el informe
JasperViewer visor = new JasperViewer(jasperPrint,false);
visor.setVisible(true);
```

Además es necesario encerrar las sentencias dentro de clausulas try-catch. El código quedaría de la siguiente manera:

```
JasperReport jr = null;
try {
    jr = JasperCompileManager.compileReport(vin_arch);
    JasperPrint jasperPrint = JasperFillManager.fillReport(jr, null, conexion);
    JasperViewer visor = new JasperViewer(jasperPrint,false) ;
    visor.setVisible(true);
} catch (JRException ex) {
    Logger.getLogger(principal.class.getName()).log(Level.SEVERE, null, ex);
}
```

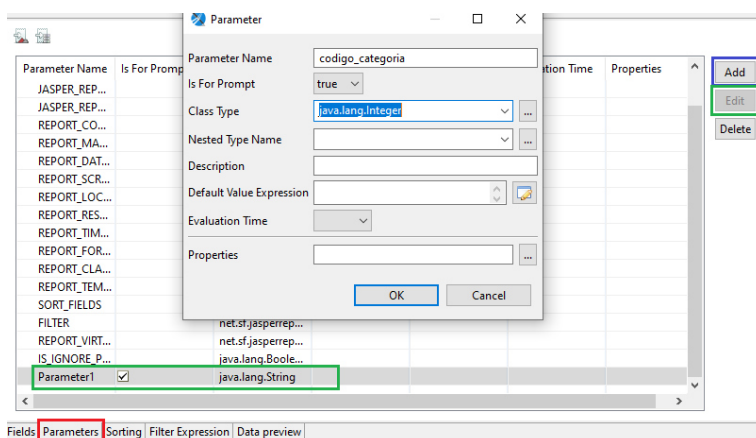
## Otros informes

Una vez visto como generar e integrar un informe con un ejemplo, sencillo vamos a ver cómo sacar mayor partido a los informes.

### 1. Informes con Parámetros

En nuestro ejemplo hemos utilizado una consulta SQL que no admite parámetros, pero esto nos limita mucho y en la realidad se desea generar informes con consultas a base de datos en las que podamos establecer parámetros de búsqueda como por ejemplo generar un informe de los clientes de una ciudad o los productos de una categoría concreta.

Para poder emplear parámetros debemos declarar esos parámetros e introducirlos en la consulta SQL. Esto lo



hacemos accediendo al Editor de consultas y datos. En la parte inferior hay un conjunto de pestañas, demos pulsar la que pone parámetros (recuadro rojo).

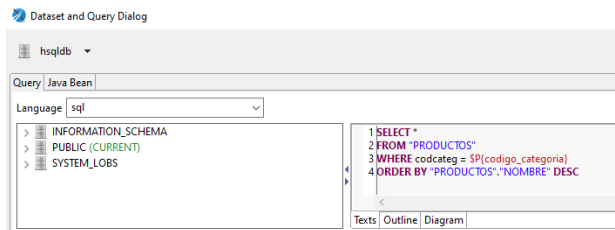
Añadimos un nuevo parámetro (recuadro azul) y después lo editamos para ajustarlo a nuestras necesidades (recuadro verde).

En la ventana que nos aparece vamos a poner el nombre de nuestro parámetro, el tipo del

parámetro (Class Type) y vamos a dejara verdadero “Is For Prompt”, con lo que nos va a mostrar un cuadro de texto para introducir un valor en la Vista Previa cuando vayamos a probar nuestro informe. En nuestro caso, eso es suficiente, pero podemos, por ejemplo, poner un valor por defecto para no tener que escribir uno en la Vista Previa.

codigo_categoria	<input checked="" type="checkbox"/>	java.lang.Integer	1
------------------	-------------------------------------	-------------------	---

El siguiente paso es escribir o modificar la consulta:



Como se muestra en la imagen, se van a mostrar los productos cuyo código de categoría (campo “codcateg” de la tabla) coincida con el que le pasemos como parámetro.

Nótese que el nombre del parámetro está dentro de \$P{<nombre\_parámetro>} y recordemos que Java distingue entre mayúsculas y minúsculas, por lo que debemos escribir el nombre del parámetro exactamente igual.

Una vez que ya tenemos nuestro archivo de informe lo colocamos dentro de nuestro proyecto Java y procedemos a escribir el código necesario para generar el informe.

Desde nuestro código le pasaremos el valor del parámetro mediante la clase MAP de la siguiente manera:

```
Map<String, Object> map = new HashMap<>(); //Declaramos la clase.
map.put("cod_categoria", Integer.valueOf(categoria.getText())); //Mapeamos el parámetro.
.....
JasperPrint jasperPrint = JasperFillManager.fillReport(jr, map, conexion);
```

El código resultante quedaría de la siguiente forma:

```
JasperReport jr = null;
try {
    Map<String, Object> miMapa = new HashMap<>();
    miMapa.put("codigo_categoria", Integer.valueOf(jTextField1.getText()));
    jr = JasperCompileManager.compileReport(vinculoarchivo);
    JasperPrint jasperPrint = JasperFillManager.fillReport(jr, miMapa, conexion);
    JasperViewer visor = new JasperViewer(jasperPrint,false) ;
    visor.setVisible(true);
} catch (JRException ex) {
    Logger.getLogger(principal.class.getName()).log(Level.SEVERE, null, ex);
}
```

En el caso de que el parámetro sea una cadena de texto, en el Report query deberemos utilizar \$P!{<nombre\_parametro>}. Por ejemplo:

```
SELECT * FROM PROVEEDORES $P!{where}
```

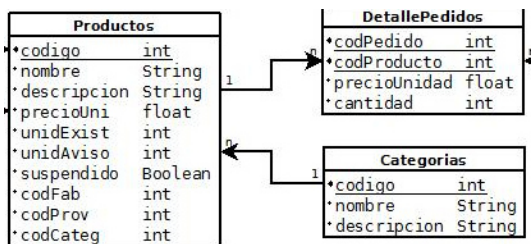
Para más información sobre los parámetros en Jasper:

<https://community.jaspersoft.com/documentation/tibco-jaspersoft-studio-user-guide/v60/using-parameters>



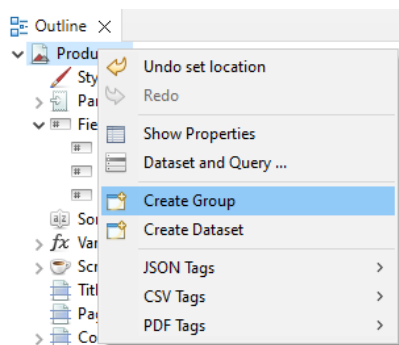
## 2. Agrupación de datos

Podemos generar un informe que agrupe los resultados por un campo concreto. Por ejemplo, si queremos generar un informe que nos muestre los productos agrupados por categorías (bebidas, carnes, etc.) deberemos, en primer lugar crear la consulta SQL y ordenarla por un campo relacionado con la categoría, por ejemplo, por el campo codcateg de productos o código o nombre de la tabla categorías. En nuestro ejemplo lo haremos sobre el nombre de la categoría.



```
SELECT p."CODIGO" AS cod_prod,
       p."NOMBRE" AS producto,
       c."NOMBRE" AS categoria
FROM "PRODUCTOS" p,
     "CATEGORIAS" c
WHERE
  p."CODCATEG" = c."CODIGO"
ORDER BY c."NOMBRE"
```

Recordemos que, durante la creación del archivo jrxml, podemos ya crear la agrupación, pero si no lo hacemos en ese momento, podemos hacerlo yendo al panel del esquema que se encuentra en la parte inferior izquierda y pulsando el botón izquierdo sobre el nombre de nuestro archivo aparecerá el menú contextual. En él elegiremos "Create Group".



<p><b>Group Band</b></p> <p><b>Group</b></p> <p>Create a new Group</p> <p>Group Name: <input type="text" value="Por_Categorias"/></p> <p><input checked="" type="radio"/> Create Group from a report object</p> <p><input type="radio"/> Create Group from expression</p> <p>Report Objects</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> COD_PROD</li> <li><input type="checkbox"/> PRODUCTO</li> <li><input checked="" type="checkbox"/> CATEGORIA</li> <li><input type="checkbox"/> fx PAGE_NUMBER</li> <li><input type="checkbox"/> fx MASTER CURRENT PAGE</li> </ul>	<p><b>Group Band</b></p> <p><b>Group layout</b></p> <p>Please select what bands should we create</p> <p><input checked="" type="checkbox"/> Add the Group Header</p> <p><input checked="" type="checkbox"/> Add the Group Footer</p>
---	--

Nos saldrá un asistente en el que tendremos que poner un nombre a la agrupación, el campo por el que vamos a agrupar los campos y si queremos que aparezcan dos nuevas bandas asociadas a la agrupación de datos.

La banda Group Header aparecerá al principio de cada agrupación y se puede utilizar para poner el campo por el que agrupamos, en nuestro caso pondremos el nombre de la categoría (CATEGORIA).

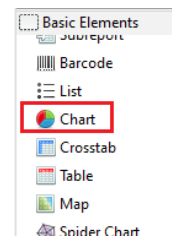
El pie del grupo aparecerá al final de cada agrupación y se puede emplear para sumar cantidades o contar elementos entre otros.

<table><tr><th>Código</th><th>Nombre del producto</th><th>Column Header</th></tr><tr><td colspan="3">Por_Categorias Group Header 1</td></tr><tr><td>\$F{COD_PROD}</td><td>\$F{PRODUCTO}</td><td>Detail 1</td></tr></table>	Código	Nombre del producto	Column Header	Por_Categorias Group Header 1			\$F{COD_PROD}	\$F{PRODUCTO}	Detail 1	<table><tr><td colspan="2">Frutas/Verduras</td></tr><tr><td>28</td><td>Col fermentada Rössle</td></tr><tr><td>14</td><td>Cajada de judías</td></tr><tr><td>51</td><td>Manzanas secas Marjimp</td></tr><tr><td>74</td><td>Queso de soja Longlife</td></tr><tr><td>7</td><td>Papas secas orgánicas del tío Bob</td></tr><tr><td colspan="2">Granos/Cereales</td></tr><tr><td>52</td><td>Cereales para Fllo</td></tr><tr><td>57</td><td>Raviolis Angelo</td></tr><tr><td>64</td><td>Bollos de pan de Wimmer</td></tr><tr><td>42</td><td>Tallarines de Singapur</td></tr><tr><td>56</td><td>Gnocchi de la abuela Alicia</td></tr></table>	Frutas/Verduras		28	Col fermentada Rössle	14	Cajada de judías	51	Manzanas secas Marjimp	74	Queso de soja Longlife	7	Papas secas orgánicas del tío Bob	Granos/Cereales		52	Cereales para Fllo	57	Raviolis Angelo	64	Bollos de pan de Wimmer	42	Tallarines de Singapur	56	Gnocchi de la abuela Alicia
Código	Nombre del producto	Column Header																																
Por_Categorias Group Header 1																																		
\$F{COD_PROD}	\$F{PRODUCTO}	Detail 1																																
Frutas/Verduras																																		
28	Col fermentada Rössle																																	
14	Cajada de judías																																	
51	Manzanas secas Marjimp																																	
74	Queso de soja Longlife																																	
7	Papas secas orgánicas del tío Bob																																	
Granos/Cereales																																		
52	Cereales para Fllo																																	
57	Raviolis Angelo																																	
64	Bollos de pan de Wimmer																																	
42	Tallarines de Singapur																																	
56	Gnocchi de la abuela Alicia																																	
Diseño del informe	Resultado																																	

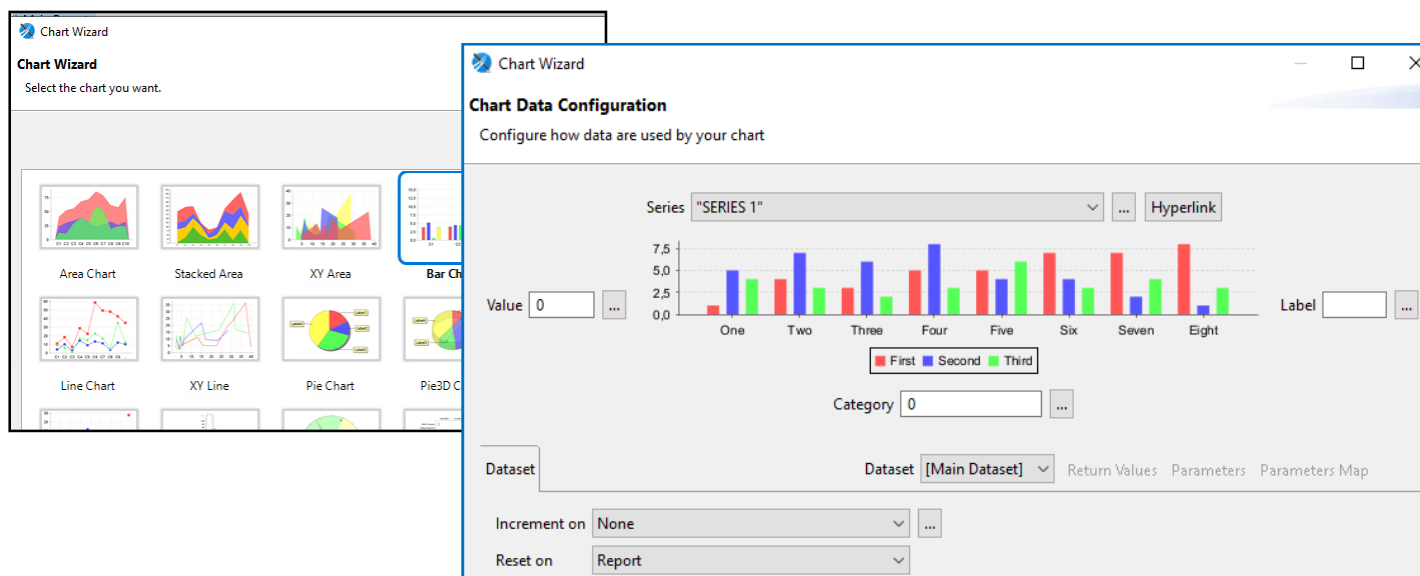
### 3. Inserción de gráficos

Algo muy común en los informes son los gráficos. Recordemos que los gráficos se sitúan en la banda Summary (resumen) y aparecerán en la última hora del informe.

Podemos elegir entre una gran variedad de gráficos y para insertar un gráfico debemos ir al panel de componentes y elegir Chart.

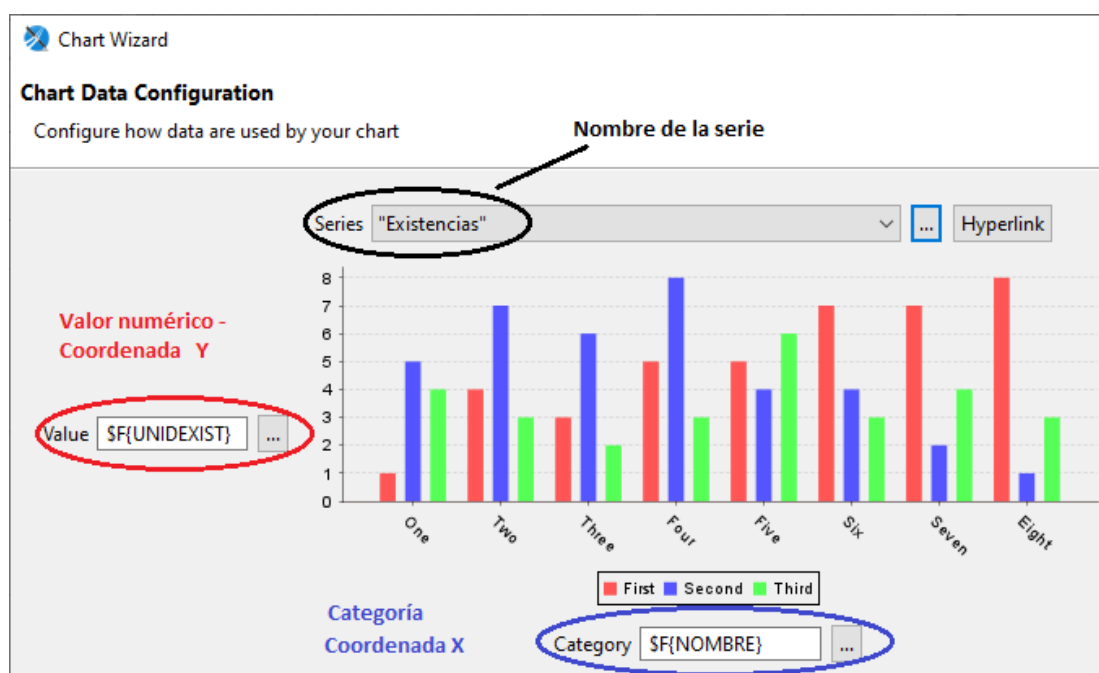


La inserción de un gráfico lleva asociado un asistente que nos permitirá elegir qué tipo de gráfico queremos y permitirá que configuremos los datos del gráfico.



¿Cómo configuramos nuestro gráfico?

En primer lugar hay que tener en cuenta que depende del gráfico que vayamos a utilizar. Aquí vamos a tomar como ejemplo un gráfico de barras.



En la imagen de ejemplo que hay arriba tenemos marcados los siguientes elementos:

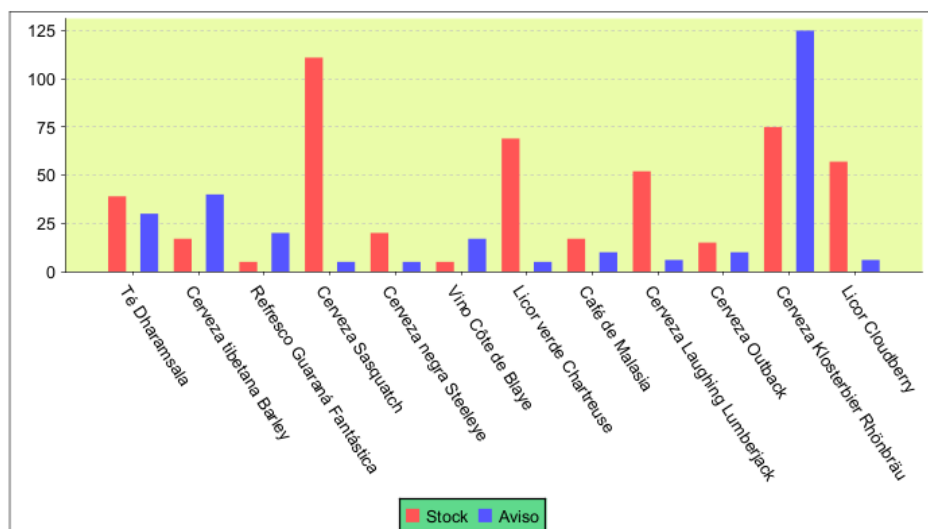
**Nombre de la serie (negro):** en un gráfico de columnas podemos tener por cada elemento a analizar uno o varios datos, por ejemplo, podríamos querer tener por cada producto el número de existencias y el nivel de aviso. Cada uno sería una serie. Este nombre, además, aparecerá en la leyenda.

**Valor numérico (rojo):** debe ser un valor numérico y representa la cantidad. En nuestro ejemplo será el campo unidades en existencia UNIDEXIST.

**Categoría (azul):** es el campo que estamos analizando, en nuestro ejemplo tenemos el campo del nombre del producto NOMBRE.

### Ejemplo

Por cada producto se desea mostrar el stock (unidadexist) y aviso (nivelaviso).



Tendremos 2 series, una de Stock y otra de Aviso y para cada una de ellas elegiremos un valor "Value". Como categoría elegimos los productos.

