**Esto es lo que vamos a ver**

1. [INSTALACIÓN Y CONFIGURACIÓN BÁSICA](http://codigoxules.org/en/creando-informes-java-jasperreports-desde-jaspersoft-studio/#instalacion-y-configuracion)
2. [DEFINIENDO UNA FUENTE DE DATOS CON POSTGRESQL](http://codigoxules.org/en/creando-informes-java-jasperreports-desde-jaspersoft-studio/#fuente-de-datos-postgresql)
3. [TABLA Y CONSULTA PARA EL INFORME](http://codigoxules.org/en/creando-informes-java-jasperreports-desde-jaspersoft-studio/#tabla-y-consulta-para-el-informe)
4. [CREAMOS NUESTRO PRIMER INFORME CON JASPERREPORTS](http://codigoxules.org/en/creando-informes-java-jasperreports-desde-jaspersoft-studio/#creamos-nuestro-primer-informe-con-jasperreports)
5. [DISEÑAMOS NUESTRO PRIMER INFORME](http://codigoxules.org/en/creando-informes-java-jasperreports-desde-jaspersoft-studio/#disenamos-nuestro-primer-informe)

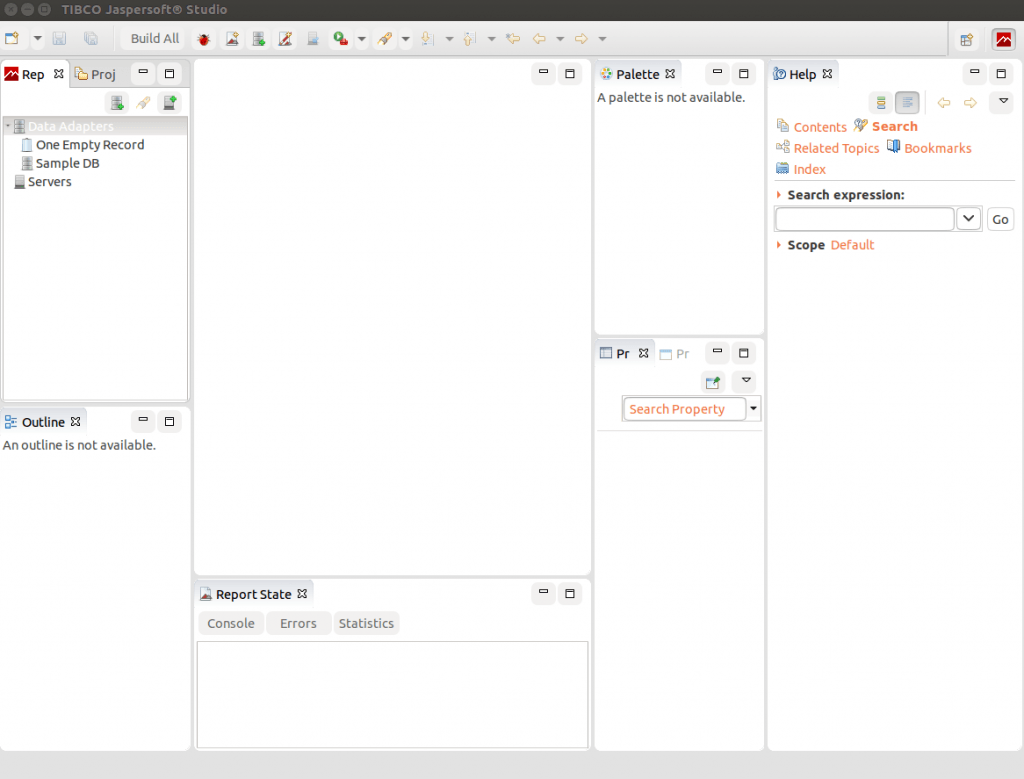
<http://codigoxules.org/en/creando-informes-java-jasperreports-desde-jaspersoft-studio/>

<http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/237>

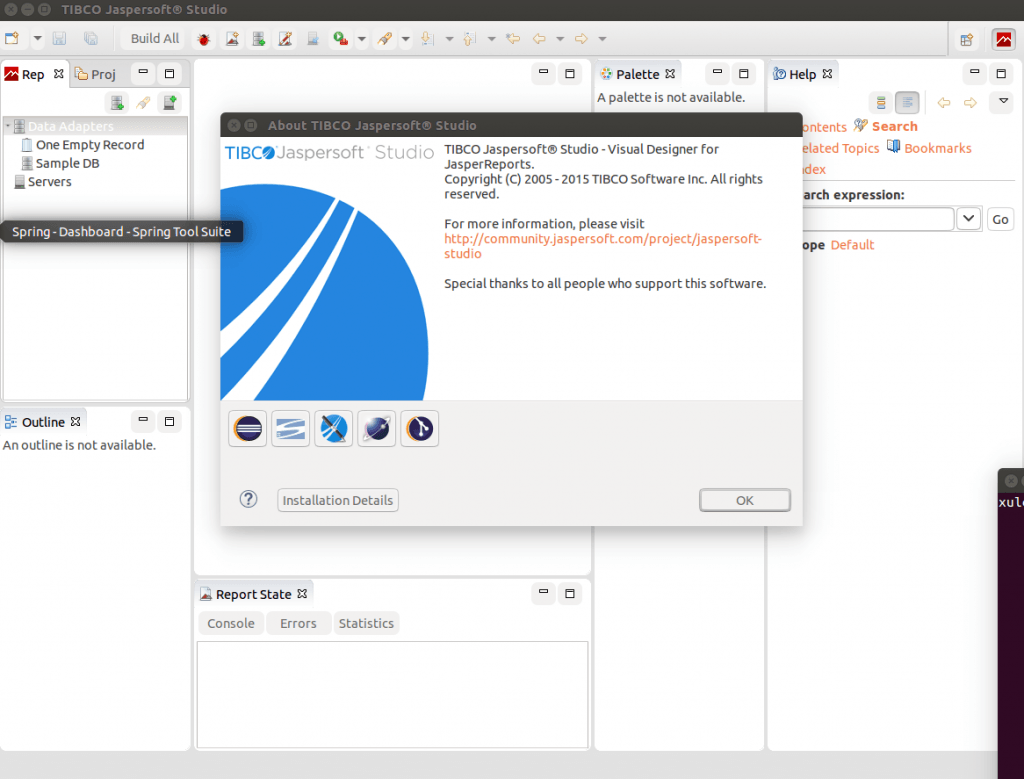
**1. INSTALACIÓN Y CONFIGURACIÓN BÁSICA**

Vamos a instalar Jaspersoft Studio en nuestro equipo, lo podemos descagar en <http://community.jaspersoft.com/project/jaspersoft-studio/releases> solo nos pedirán un registro de usuario.

Una vez en nuestro sistema ejecutamos la instalación y listo, este el aspecto que tiene nuestro generador de informes:



Estos son los detalles de [Tibco](http://www.tibco.com/) la empresa autora de [JasperSoft Studio](http://community.jaspersoft.com/project/jaspersoft-studio) que se engloba dentro de los desarrollos de una de sus filiales [Tibco JasperSoft](https://www.jaspersoft.com/es):

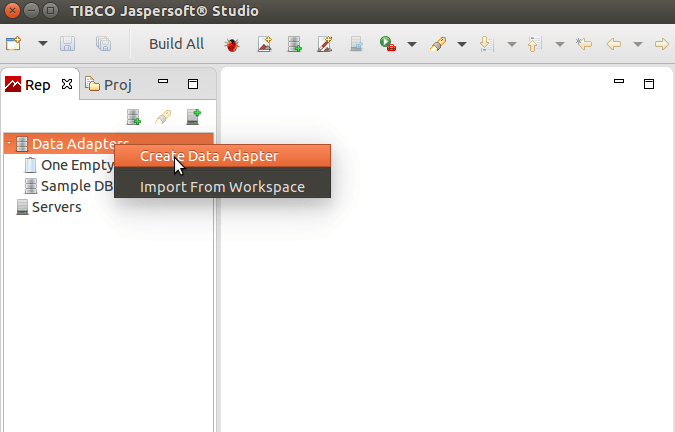


**2. DEFINIENDO UNA FUENTE DE DATOS CON POSTGRESQL**

Si no dispones de una instalación de PostgreSQL no te preocupes puedes utilizar otra base de datos lo único que tendrás que añadir será el driver propio de la conexión a esa base de datos y los datos de la conexión como voy a indicar a continuación para PostgreSQL.

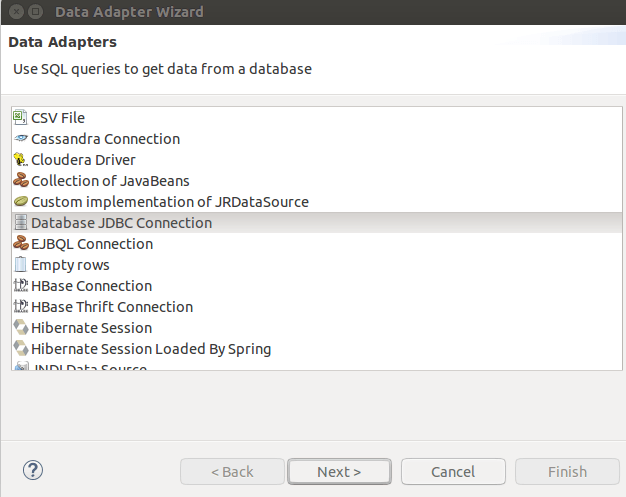
Si quieres instalar y obtener información sobre PostgreSQL la puedes obtener en [mi guía sobre PostgreSQL](http://codigoxules.org/tutorial-postgresql-1-preparando-el-entorno/), la misma información también la tienes disponible en [mi guía sobre MySQL (MariaDB)](http://codigoxules.org/tutorial-mariadb-1-preparacion-del-entorno/).

Echas las presentaciones vamos al grano para eso necesitamos definir en primer lugar la fuente de datos que vamos a utilizar, por eso vamos a ver como creamos un nuevo **Data Adapter JDBC** para nuestra conexión con PostgreSQL:



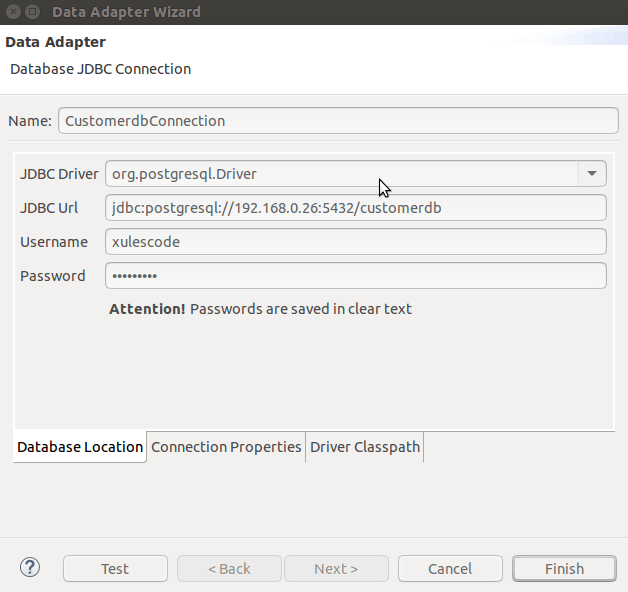
Jasper Reports – Nuevo Data Adapter JDBC

Usamos el Wizard de Jaspersoft Studio seleccionando **Database JDBC Connection**:



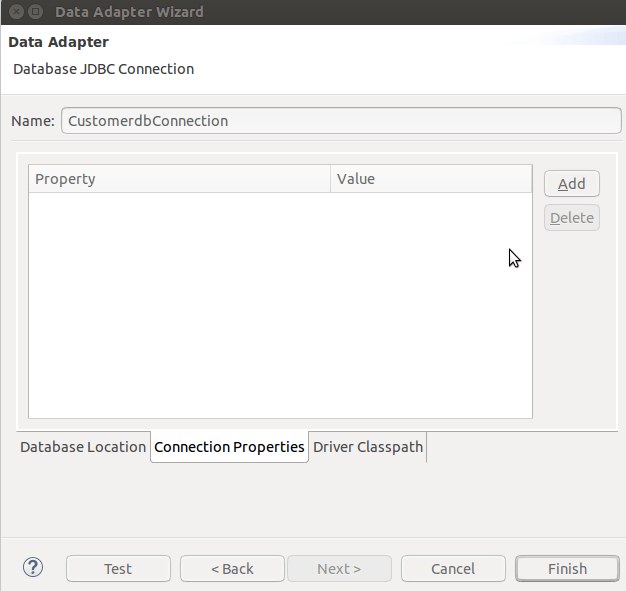
Jasper Reports – Data Adapter JDBC – Wizard

Seleccionamos la conexión **Database JDBC Connection**, donde damos un nombre a nuestra conexión y definimos la localización de la base de datos como se muestra en la imagen:



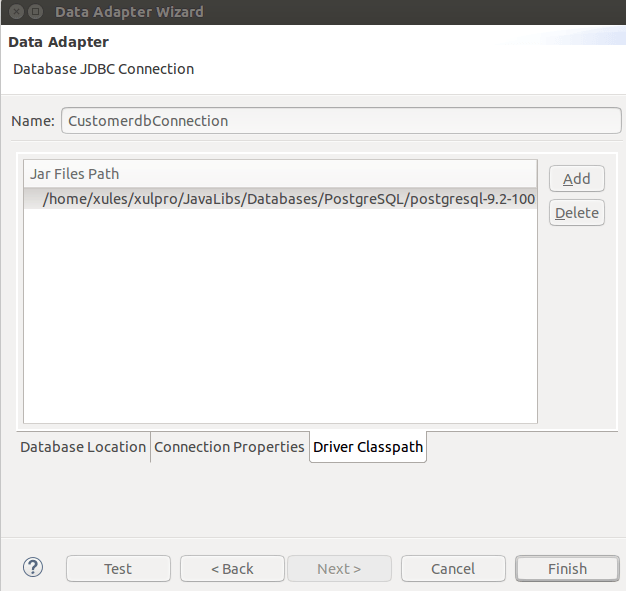
Jasper Reports- Data Adapter JDBC – Database Location

Desde aquí también podemos definir otras propiedades de nuestra conexión en **Connection Properties**:



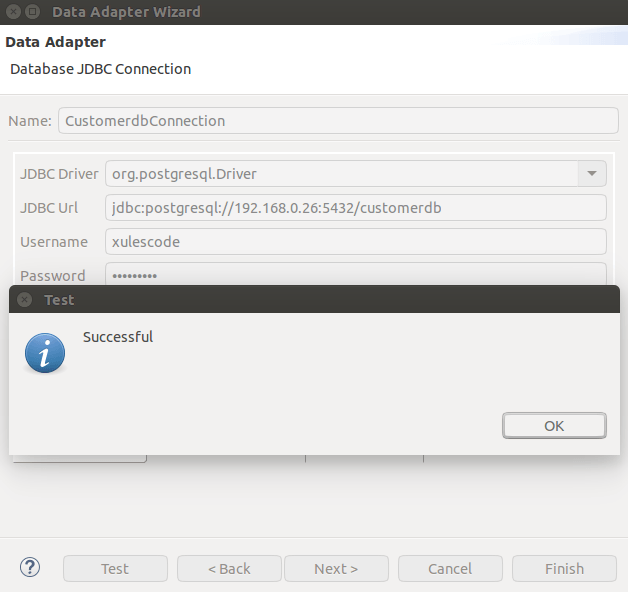
Jasper Reports – Data Adapter JDBC – Connection Properties

Para hacer el test solo nos queda indicar la localización del driver de la base de datos que estamos utilizando en nuestro ejemplo, en mi caso utilizo [PostgreSQL](http://codigoxules.org/postgresql/), y lo añado indicando la carpeta donde se encuentra instalado, si quieres usar otra base de datos deberás descargar el driver e indicar su localización:



Jasper Reports – Data Adapter JDBC

Una vez completados los datos, evaluamos que hemos introducido correctamente todos los parámetros con la opción **Test**:

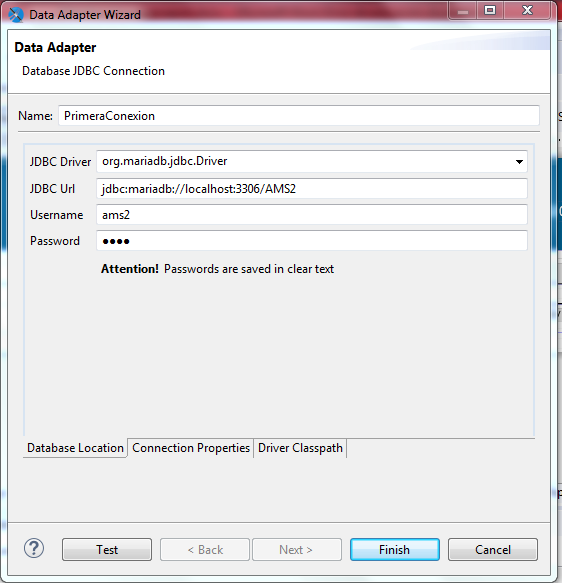


Jasper Reports – Data Adapter JDBC – Database Location Test

Una vez finalizado este último paso ya tenemos configurada una fuente de datos que podremos seleccionar y utilizar para diferentes informes que realizamos sobre esa conexión.

IMPORTANTE:

Para hacer lo mismo con base de datos MariaDB, que por ejemplo es el que lleva XAMPP, ponemos:



Base de datos es AMS2

**3. TABLA Y CONSULTA PARA EL INFORME**

Para este ejemplo utilizo una tabla del desarrollo de [Learning Project](http://codigoxules.org/learning-project/) para [PostgreSQL](http://codigoxules.org/postgresql/), si quieres descargarte la base de datos de 8 tablas que utilizo para este proyecto y los datos para su población, lo puedes hacer aquí:

[**PostgreSQL SQL - Creación de las tablas**](http://codigoxules.org/en/download/postgresql-sql-creacion-de-las-tablas/)

1 file(s) 9.97 KB

[**PostgreSQL Script SQL para poblar la base de datos Customerdb**](http://codigoxules.org/en/download/script-sql-para-poblar-la-base-de-datos-customerdb/)

1 file(s) 126.64 KB

Para hacerlo más fácil para este ejemplo, puedes también simplemente crear esta tabla sin relaciones adicionales, ya que lo que buscamos es introducirnos en el desarrollo de JasperReports con Jaspersoft Studio:

[?](http://codigoxules.org/en/creando-informes-java-jasperreports-desde-jaspersoft-studio/)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | CREATE TABLE cb\_country  (    idcountry serial NOT NULL,    country character varying(100) NOT NULL,    description character varying(255),    countrycode character(2) NOT NULL,    hasregion character(1) NOT NULL DEFAULT 'N'::bpchar,    regionname character varying(60),    expressionphone character varying(20),    displaysequence character varying(20) NOT NULL,    isdefault character(1) DEFAULT 'N'::bpchar,    ibannodigits numeric,    ibancountry character varying(2),    isactive boolean NOT NULL DEFAULT true,    idlanguage character varying(6),    idcurrency integer,    CONSTRAINT pk\_cb\_country PRIMARY KEY (idcountry),    CONSTRAINT un\_cb\_country\_countrycode UNIQUE (countrycode)  );  ALTER TABLE cb\_country  OWNER TO xulescode;  COMMENT ON TABLE cb\_country  IS 'Tabla donde se definen todos los países con sus características principales: idioma, nombre, ..., y diferentes datos íntrinsecos a cada país.'; |

Algunos países para que puedas poblar está tabla de forma sencilla:

|  |  |
| --- | --- |
|  | INSERT INTO cb\_country (idcountry, country, description, countrycode, hasregion, regionname, expressionphone, displaysequence, isdefault, ibannodigits, ibancountry, isactive, idlanguage, idcurrency) VALUES (270, 'United States', 'United States of America', 'US', 'Y', 'State', 'Y', '@C@, @R@ @P@', 'Y', NULL, '', true, 'en\_US', 7);  INSERT INTO cb\_country (idcountry, country, description, countrycode, hasregion, regionname, expressionphone, displaysequence, isdefault, ibannodigits, ibancountry, isactive, idlanguage, idcurrency) VALUES (273, 'Belgium', 'Belgium', 'BE', 'N', '', '', 'B-@P@ @C@', 'N', 16, 'BE', true, NULL, 7);  INSERT INTO cb\_country (idcountry, country, description, countrycode, hasregion, regionname, expressionphone, displaysequence, isdefault, ibannodigits, ibancountry, isactive, idlanguage, idcurrency) VALUES (274, 'Holland', 'Holland', 'NL', 'N', '', 'Y', 'NL @P@ @C@', 'N', 18, 'NL', true, 'nl\_NL', 7);  INSERT INTO cb\_country (idcountry, country, description, countrycode, hasregion, regionname, expressionphone, displaysequence, isdefault, ibannodigits, ibancountry, isactive, idlanguage, idcurrency) VALUES (276, 'Switzerland', 'Switzerland', 'CH', 'N', '', '', 'CH-@P@ @C@', 'N', 21, 'CH', true, 'de\_CH', 7);  INSERT INTO cb\_country (idcountry, country, description, countrycode, hasregion, regionname, expressionphone, displaysequence, isdefault, ibannodigits, ibancountry, isactive, idlanguage, idcurrency) VALUES (277, 'Austria', 'Österreich', 'AT', 'N', '', '', 'A-@P@ @C@', 'N', 20, 'AT', true, 'de\_AT', 7); |

Si quieres más, aquí tiene el script para hacer un INSERT en la tabla creado con 240 Países:

[**Countries for database table cb\_country**](http://codigoxules.org/en/?post_type=wpdmpro&p=1833)

1 file(s) 72.55 KB

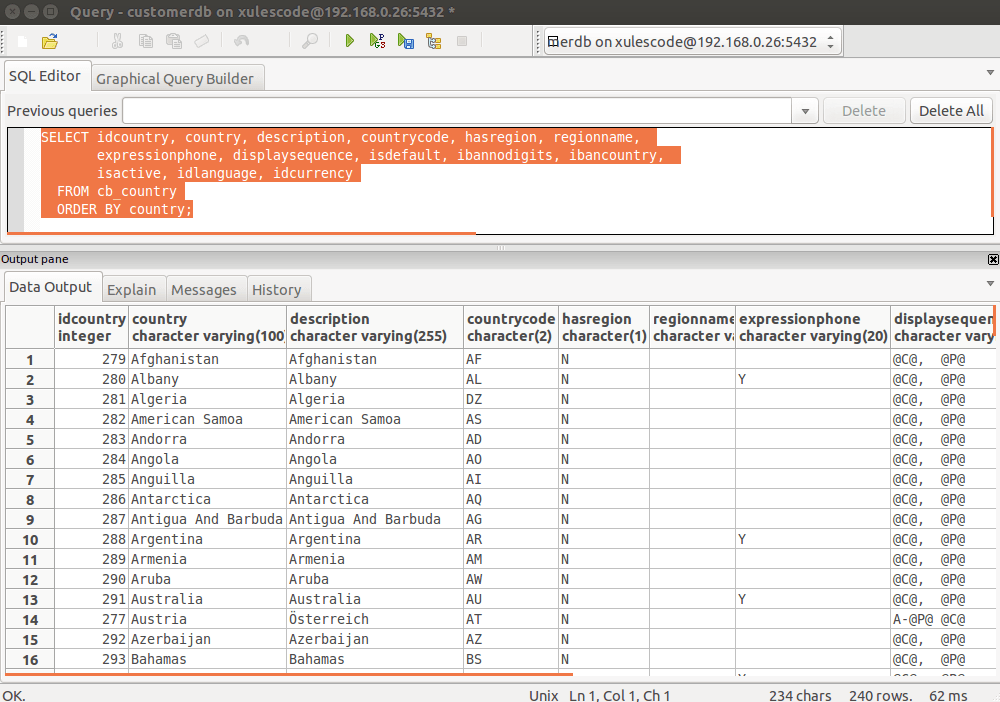
 La consulta que utilizaremos para este informe será un SELECT sobre todos los campos de la tabla, ya que la selección de campos lo haremos al crear el informe. Estamos simplificando lo más posible la consulta para centrarnos en el desarrollo y creación de informes, ya que se puede utilizar la consulta con toda la complejidad que necesitemos, incluyendo la llamada a funciones procedurales.

Esta es la que utilizaremos en la creación del informe en el siguiente apartado:

[?](http://codigoxules.org/en/creando-informes-java-jasperreports-desde-jaspersoft-studio/)

|  |  |
| --- | --- |
| 1  2  3  4  5 | SELECT idcountry, country, description, countrycode, hasregion, regionname,         expressionphone, displaysequence, isdefault, ibannodigits, ibancountry,         isactive, idlanguage, idcurrency    FROM cb\_country    ORDER BY country; |

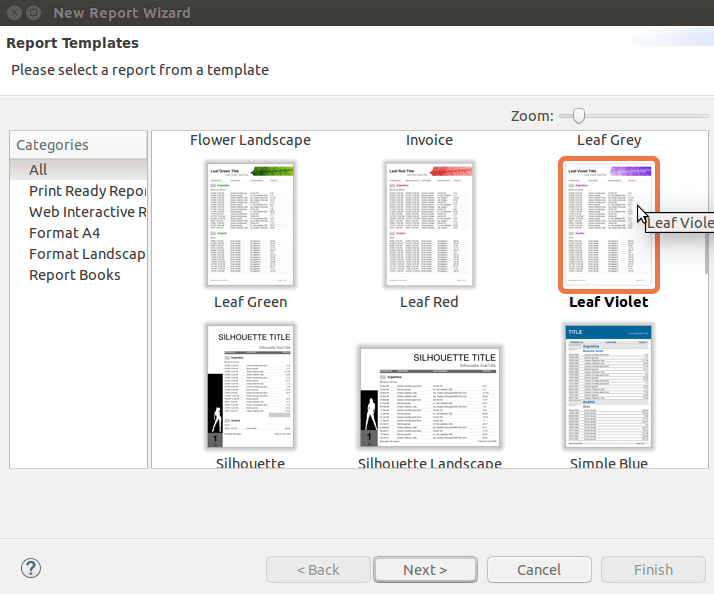
Y estos algunos de los resultados que obtenemos ejecutando la consulta en un gestor de bases de datos en este caso **PgAdmin**:



Jasper Reports – Resultado consulta sobre cb\_country

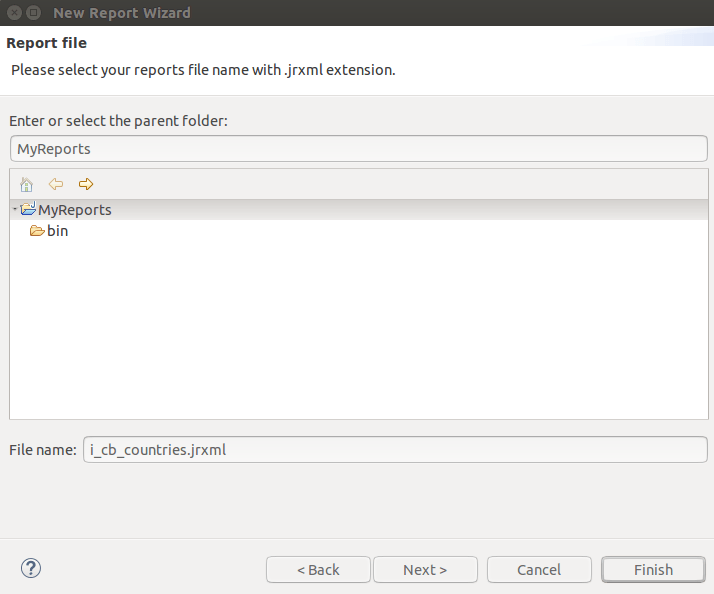
**4. CREAMOS NUESTRO PRIMER INFORME CON JASPERREPORTS**

Empezaremos con nuestro primer informe con JasperSoft Studio seleccionando una de las plantillas base para el diseño, para empezar seleccionamos New Report (nuevo informe) y ya podemos seleccionar la plantilla que utilizaremos:



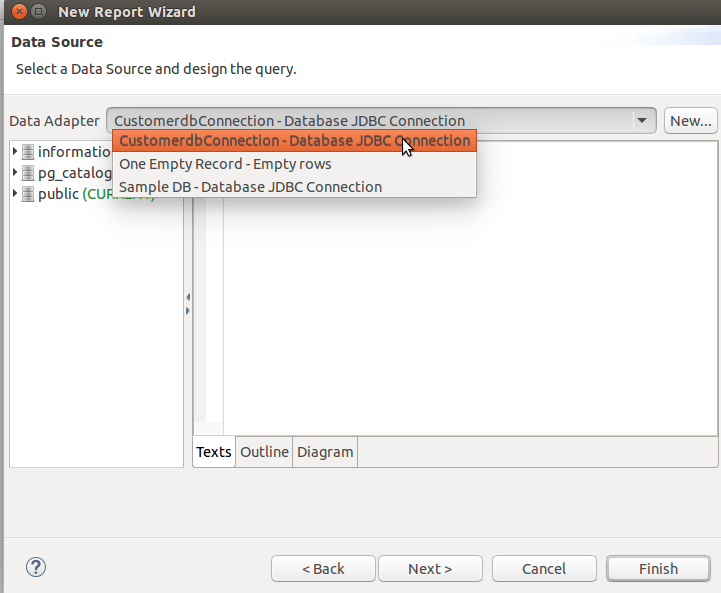
Jasper Reports – New Report – Select Template

En el siguiente paso se nos indica la carpeta donde se guardará nuestro informe que tendrá un formato jrxml, de momento dejamos la configuración por defecto e introducimos el nombre de nuestro informe en mi caso i\_cb\_countries.jrxml:



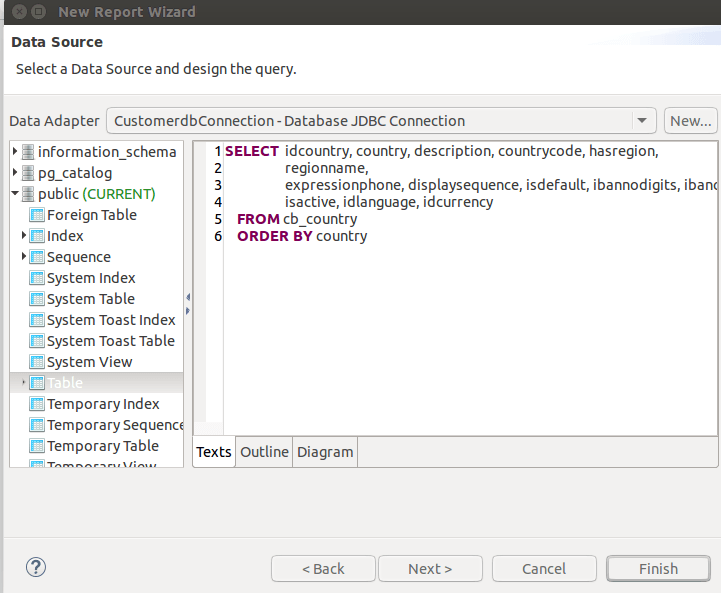
Jasper Reports – New Report – Report name

Ahora el Wizard nos permite que seleccionemos un **Data Source**, aquí seleccionamos el que creamos anteriormente:



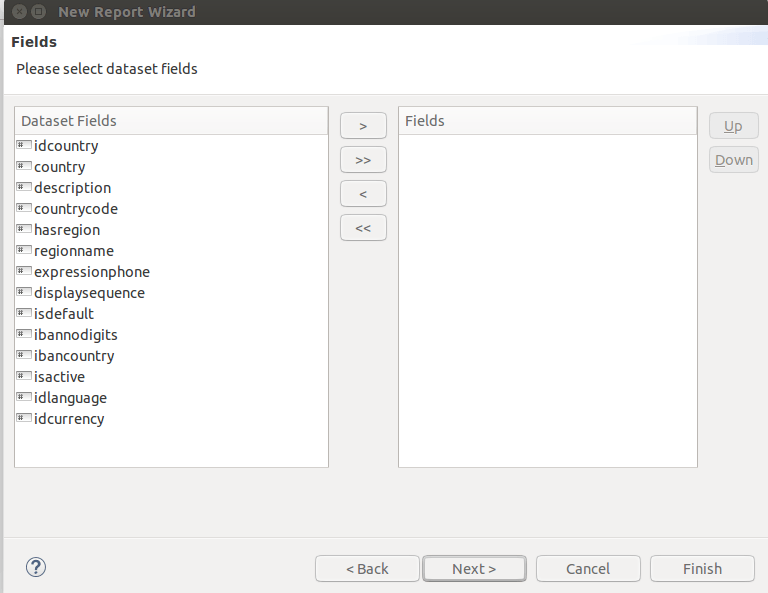
Jasper Reports – New Report – Select Data Source

Añadimos la consulta tal y como se nos indica, puedes copiar directamente el SQL que te proporcioné con anterioridad:



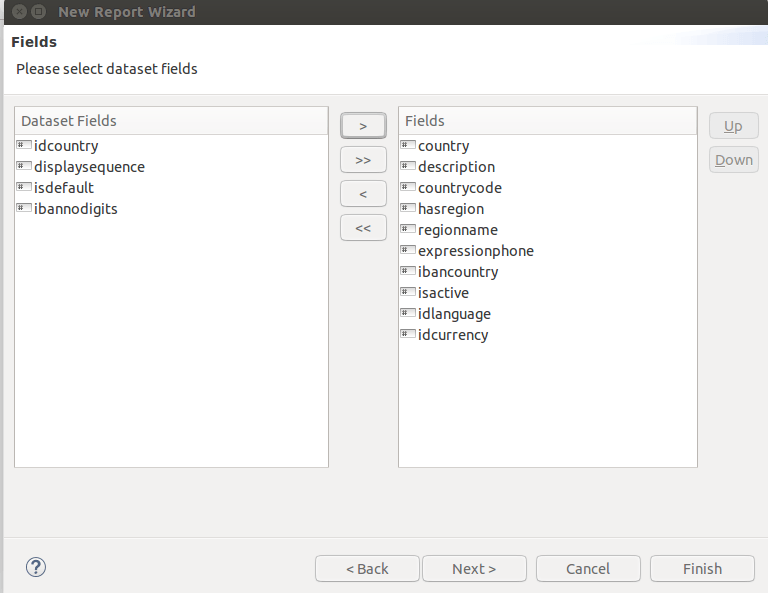
Jasper Reports – New Report – Añadimos la consulta (Add Query)

Es momento de seleccionar los campos que queremos mostrar en el informe, esto es el procedimiento habitual, ya que hay que campos que puedes necesitar en el informe para condicionar como se muestran los resultados, por ejemplo, pero que no quieres que se muestren directamente en el informe:



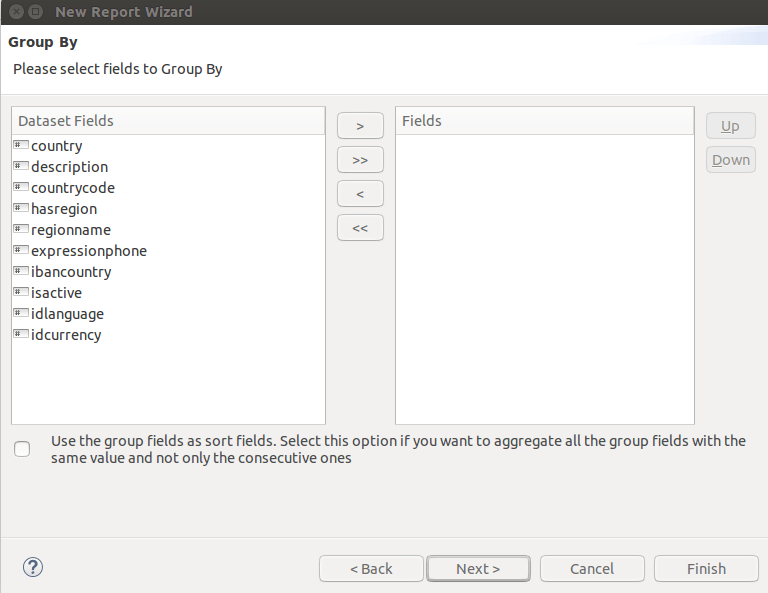
Jasper Reports – New Report – Selección de campos

Aquí puedes ver los campos que yo he seleccionado para este informe:



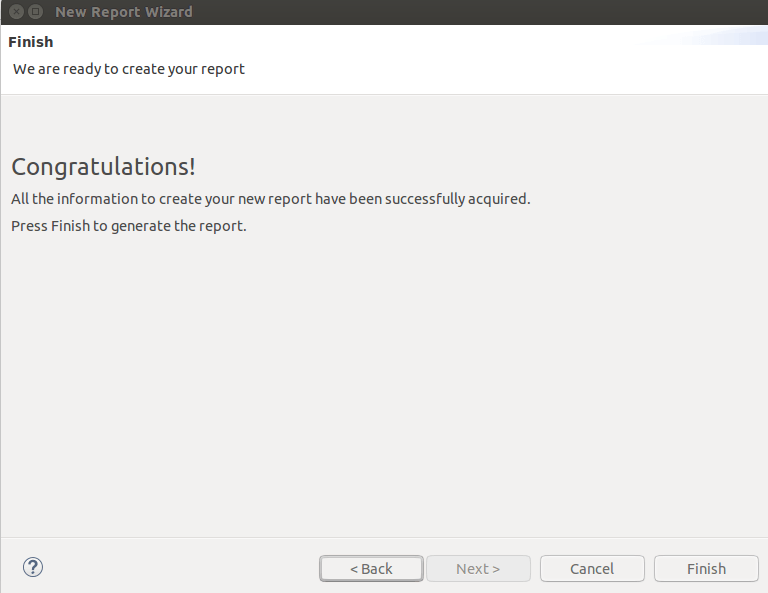
Jasper Reports – New Report – Campos seleccionados

El último apartado del Wizard nos permite crear grupos, esta opción es muy interesante y útil para agrupar resultados por una de las columnas, para este sencillo informe lo dejaremos en blanco y ya entraremos en profundidad más adelantes:



Jasper Reports – New report – No definimos ningún grupo (Group By – Blank)

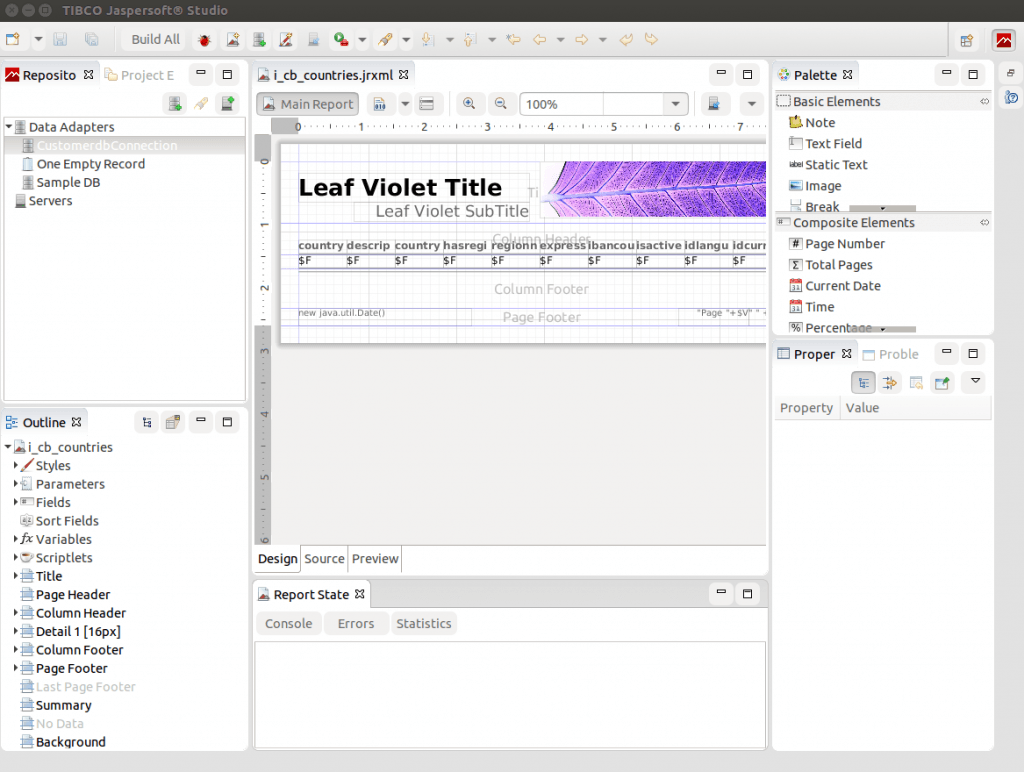
Generamos el informe pulsando el botón de **Finish** y nos saldrá el aviso de la correcta creación de nuestro informe:



Jasper Reports – New Report – FIN

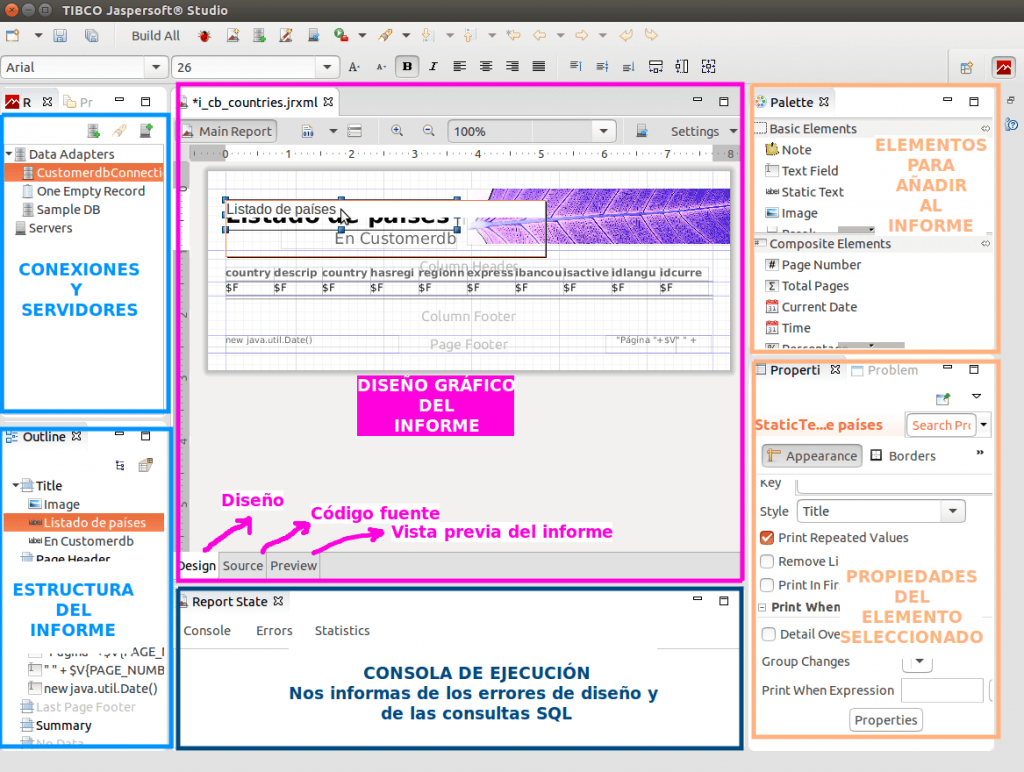
**5.DISEÑAMOS NUESTRO PRIMER INFORME**

Ya hemos creado nuestro informe, ahora mismo puedes pulsar el botón **Preview** y se ejecutará el informe por defecto en la vista previa que viene integrada en Jaspersoft Studio, así es como se ven el informe que hemos creado:



Jasper Reports – New Report – Diseño (Design)

En la siguiente imagen te detallo las partes más importantes del generador de informes:



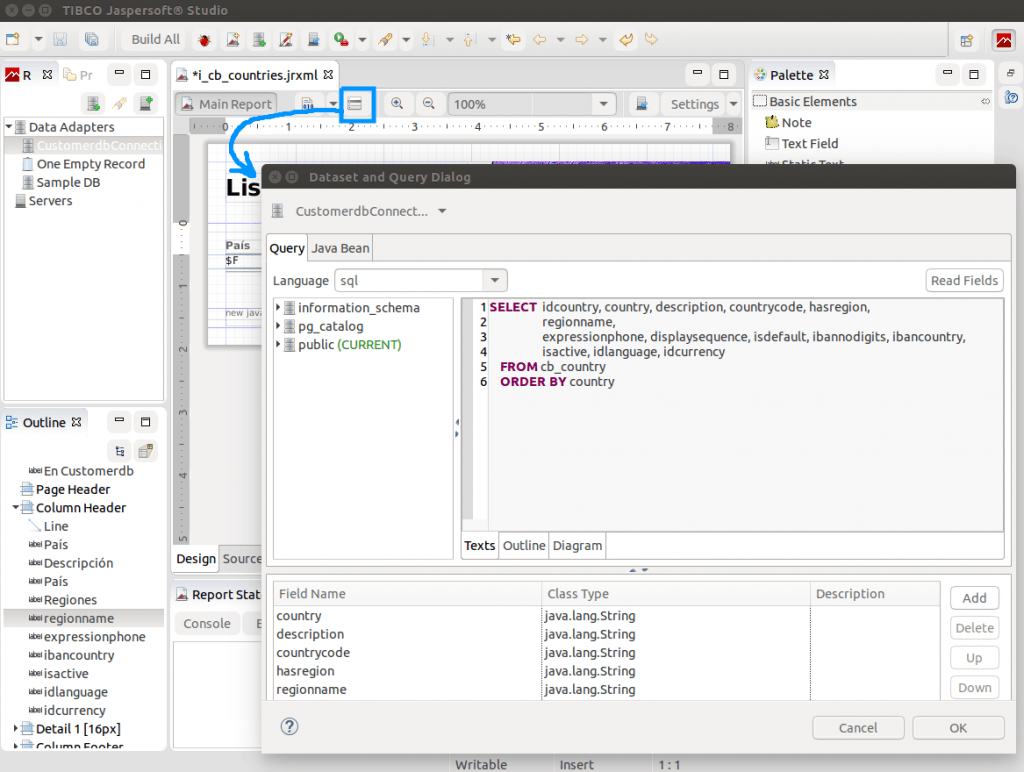
Jasper Reports – Partes principales del generador de informes

Vamos a hacer unos cambios sencillos para ir familiarizándonos con el entorno y aprendiendo como se trabaja con el desarrollo de informe con Jaspersoft Studio.

**Editor de consultas SQL**

Con JasperReports y en concreto desde la aplicación JasperSoft Studio podemos actualizar la consulta en cualquier momento, esto nos va a permitir añadir nuevos campos si se nos olvidó alguno, o cambiar la consulta por completo si es necesario.

Para modificar la consulta pulsa el botón que se indica en la imagen y se abrirá un editor donde podrás editar la consulta:



Jasper Reports – New Report – Diseño – Edición de consulta SQL (Query Editor)

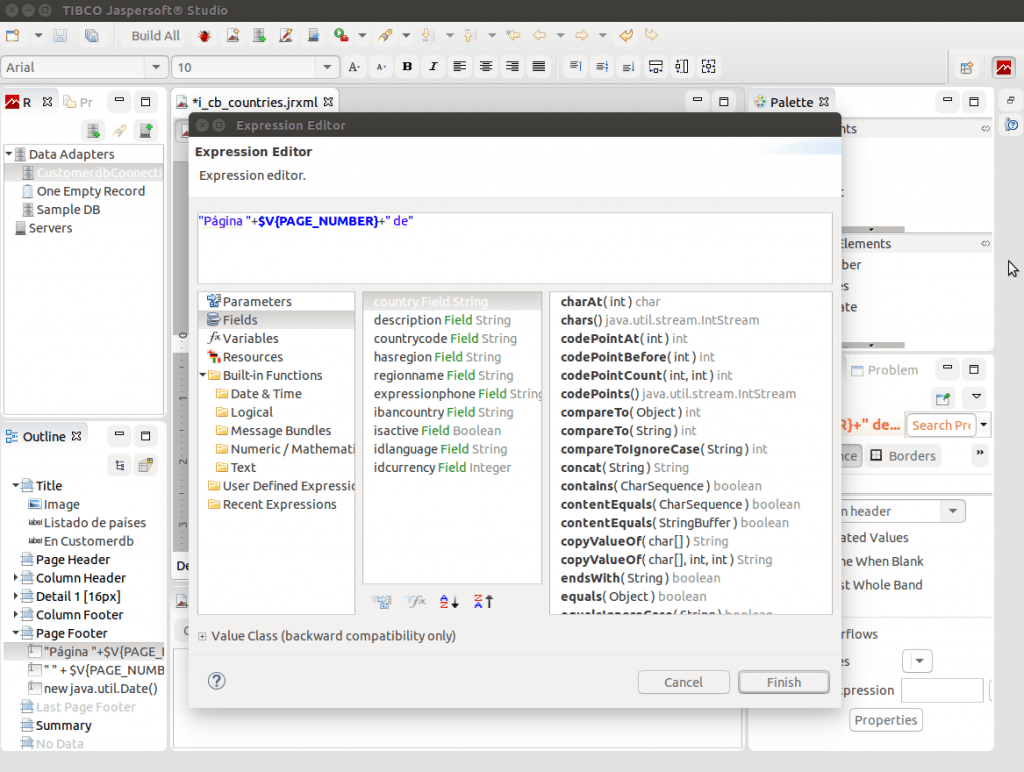
También tiene otras funcionalidades adicionales como: editor gráfico de consulta, añadido de parámetros, ordenación, …, esto lo veremos con más detalle cuando necesitemos profundizar en el diseño de informes con JasperReports.

**Diseño – Editor de expresiones (Expression editor)**

Inicialmente en nuestro informe vamos a tener **etiquetas** (Label) y **expresiones** (Expressions), en la paleta del diseñador los verás como Static Text y Text Field respectivamente.

Para añadir un nuevo Text Field simplemente arrástralo desde la paleta al informe, este añadirá el elemento al informe, este tiene una expresión que nos permitirá hacer referencia a los elementos de la base de datos: fields ($F{database-field}), y también, a otro tipo de expresiones como son **parámetros (parameters: $F{parameter})** y valores calculados ($V{variable}).

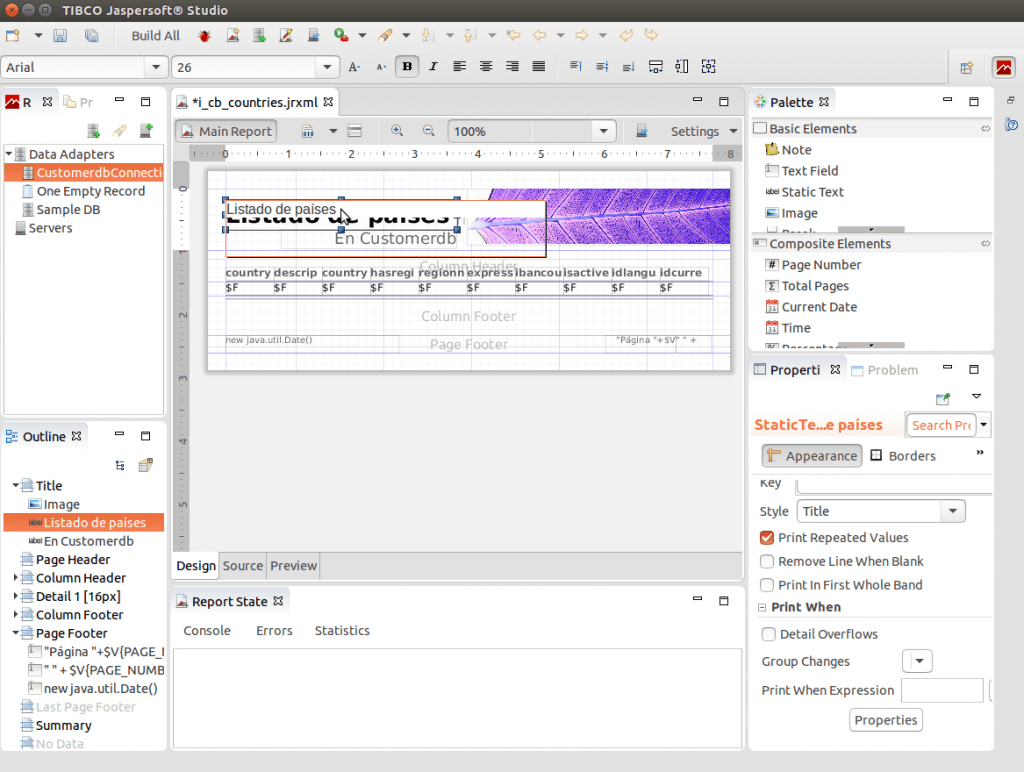
Para abrir el editor puedes hacer click sobre el elemento que quieras modificar y te aparecerá la siguiente ventana:



Jasper Reports – New Report – Diseño – Editor de expresiones (Expression editor)

**Diseño – Editar etiquetas (Label)**

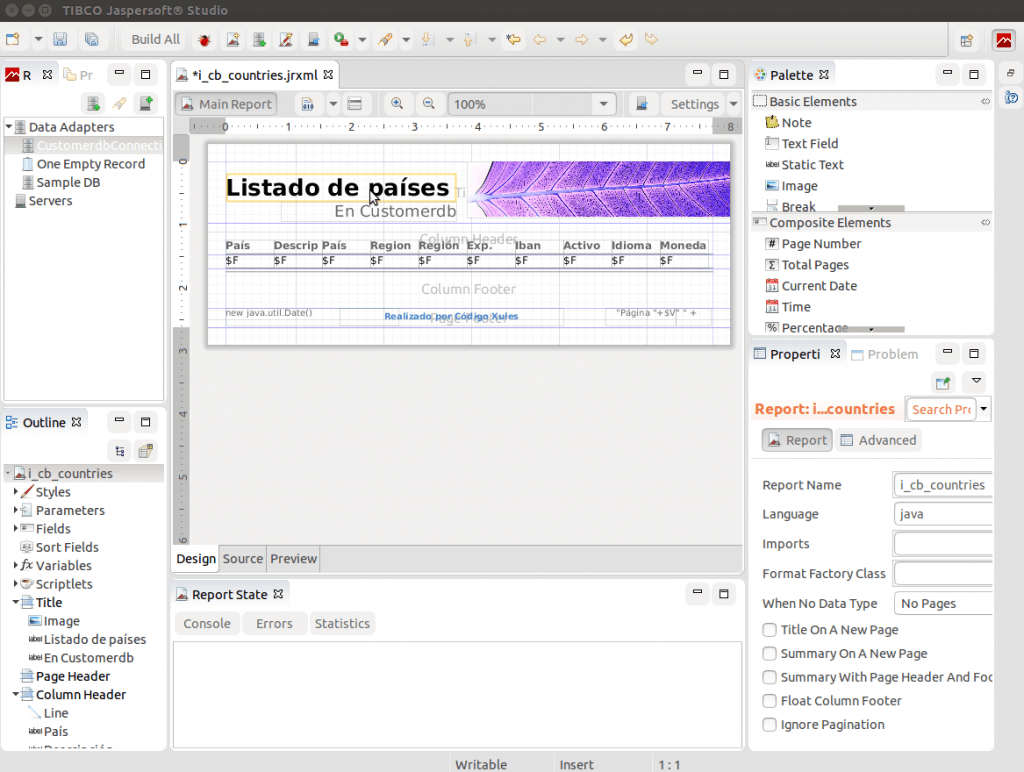
Las **etiquetas** (Label) son los Static Text que nos permitirán añadir texto plano a nuestro informe y enriquecer nuestro diseño mediante la modicación de la fuente, tamaño, colores, … , todo esto se encuentra dentro de las propiedades de Static Text. Una vez añadida la etiqueta a nuestro informe también podemos modificar el texto directamente sobre el generador:



Jasper Reports – New Report – Diseño – Editar etiquetas (Label)

**Fin del Diseño**

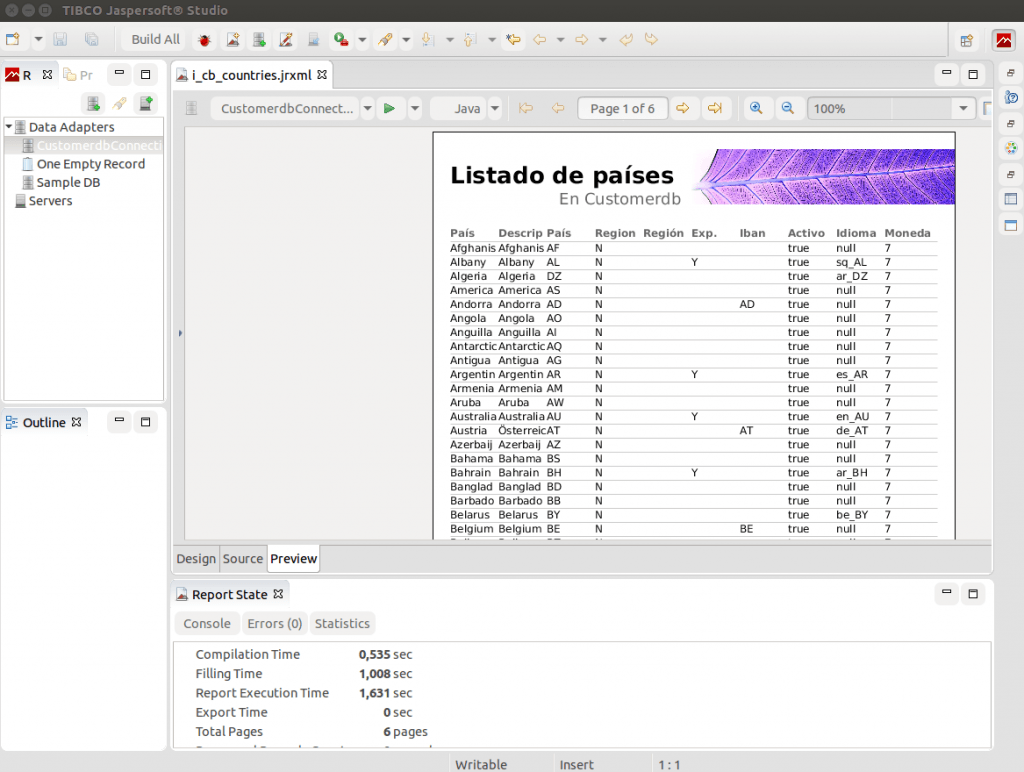
Como podrás ver en mi informe finalizado he modificado los nombre de las columnas que son Static Text y actualizado también los campos de la cabecera y el pié de página, te recomiendo que explores añadiendo y modificando campos para familiarizarte con el programa JasperSoft Studio, aquí está mi informe finalizado:



Jasper Reports – New Report – Diseño – FIN

**Vista previa del informe**

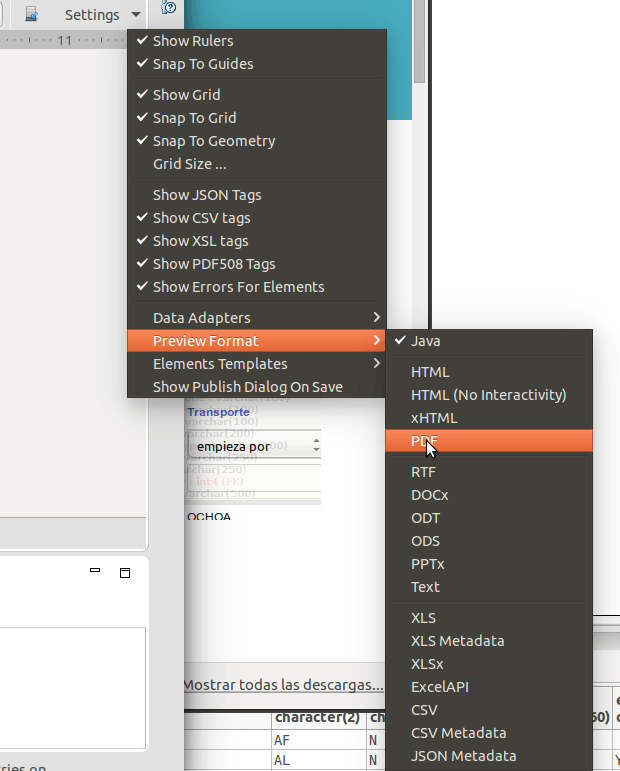
Vamos a ver como queda el informe finalmente, para ello JasperSoft Studio incorpora un botón de **Preview** para visualizar nuestro informe con JasperReports dentro del programa, como se muestra en la imagen.



Jasper Reports – New Report – Diseño – Vista previa (Preview)

**Vista previa en formato PDF**

Este mismo informe lo podemos ejecutar y mostrar en varios formatos simplemente indicando la salida que queremos para la ejecución de nuestro informe, ahora vamos a ver como generar la salida en PDF, como ves en la imagen selecciona esta opción y ejecuta nuevamente el informe pulsando el botón **Preview**:

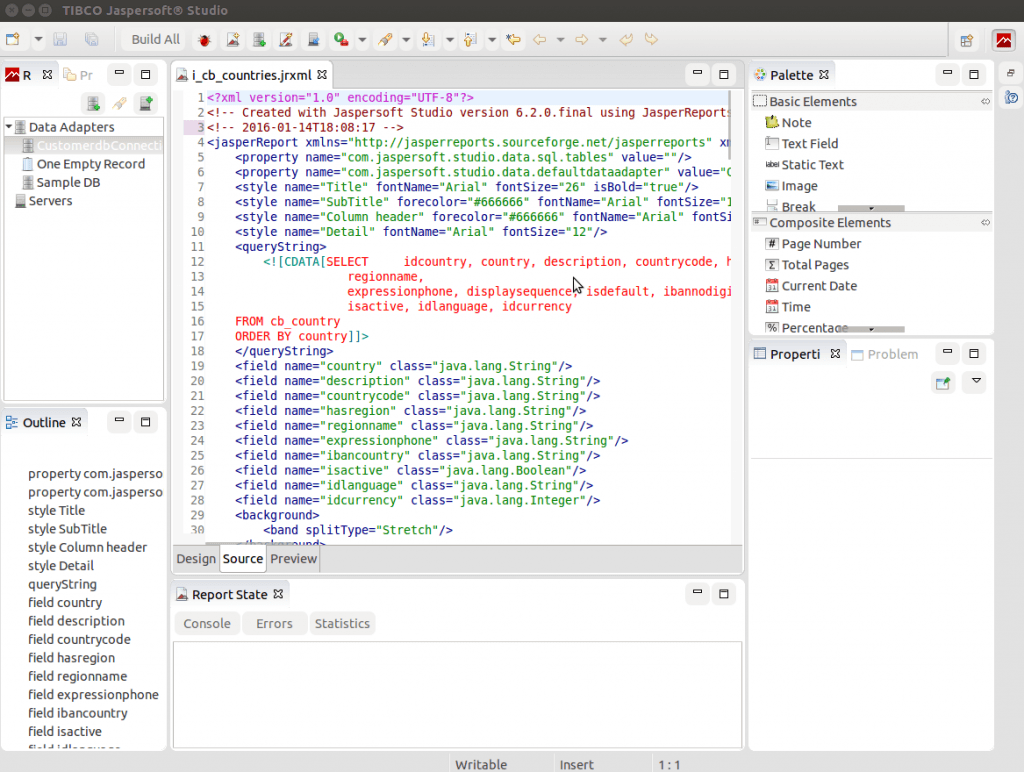


Jasper Reports – New Report – Diseño – Vista previa en PDF

Si lo has realizado verás como se abre automáticamente el informe en PDF, la vista previa también aparece visible tal y como vimos anteriormente. Si tienes curiosidad puedes descargar aquí el resultado final del informe en PDF:

**Código fuente jrxml**

Para finalizar puedes echar un vistazo al código fuente de nuestro informe en formato jrxml que es lo que interpreta el diseñador de informes, cualquier modificación que realices aquí se trasladará directamente al diseñador:

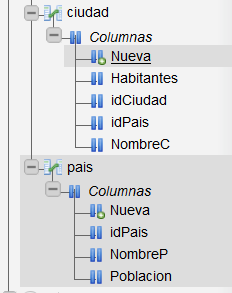


Jasper Reports – New Report – Diseño – Código fuente (Source)

Otras tablas y consultas:

SELECT NombreC, Habitantes , pais.NombreP, pais.Poblacion FROM `ciudad`, pais where ciudad.idPais=pais.idPais;

Uso dos tablas:



Creo un informe con la siguiente consulta:

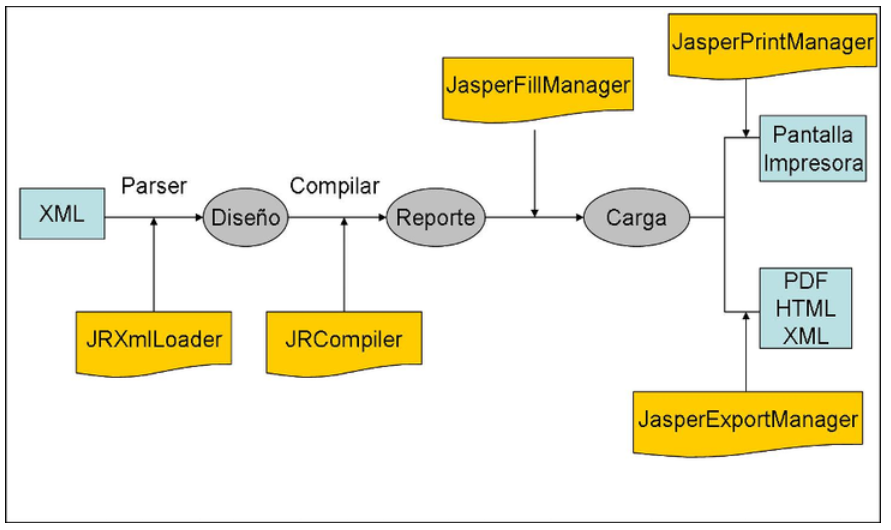
SELECT NombreC, Habitantes , pais.NombreP, pais.Poblacion FROM `ciudad`, pais where ciudad.idPais=pais.idPais;

Con parámetros:

**SELECT** NombreC, Habitantes , pais.NombreP, pais.Poblacion **FROM** `ciudad`, pais

**where** ciudad.idPais=pais.idPais **AND** pais.idPais=’$P{idPais}’

# Mostrar informes en aplicación JAVA



JasperFillManager.

El método principal de esta clase es:



Este método statico devuelve un objeto del tipo [JasperPrint](http://jasperreports.sourceforge.net/api/net/sf/jasperreports/engine/JasperPrint.html)

Los parámetros son:

* Un report complilado con extensión jasper.
* Un HashMap para pasar parámetros al report.
* Una Conexión.

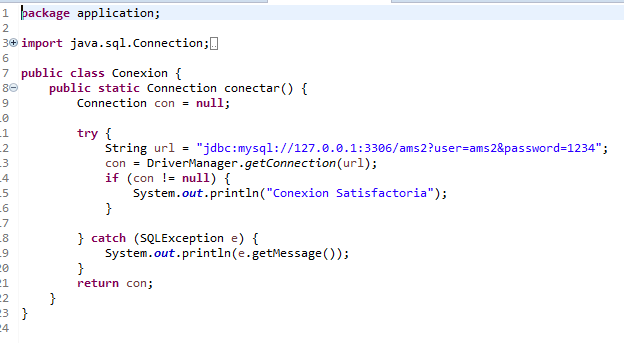
Ejemplo:

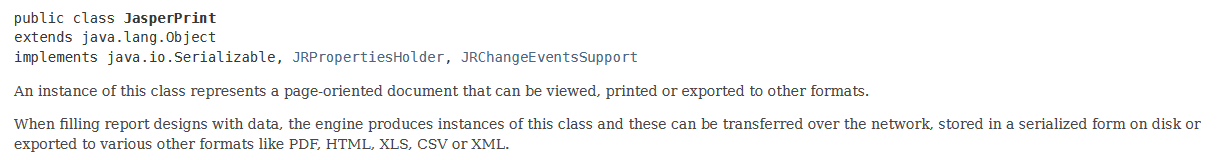


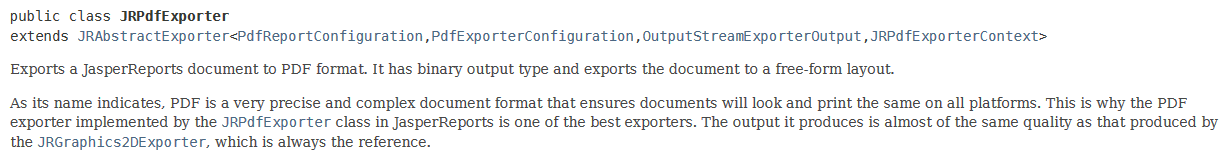


( en este caso no pasamos parámetros )

La clase Conector está en el proyecto:

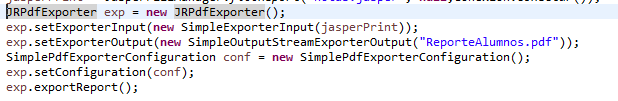






JasperPrint generado a partir del informe

Esta clase sirve para exportar un JasperPrint a pdf



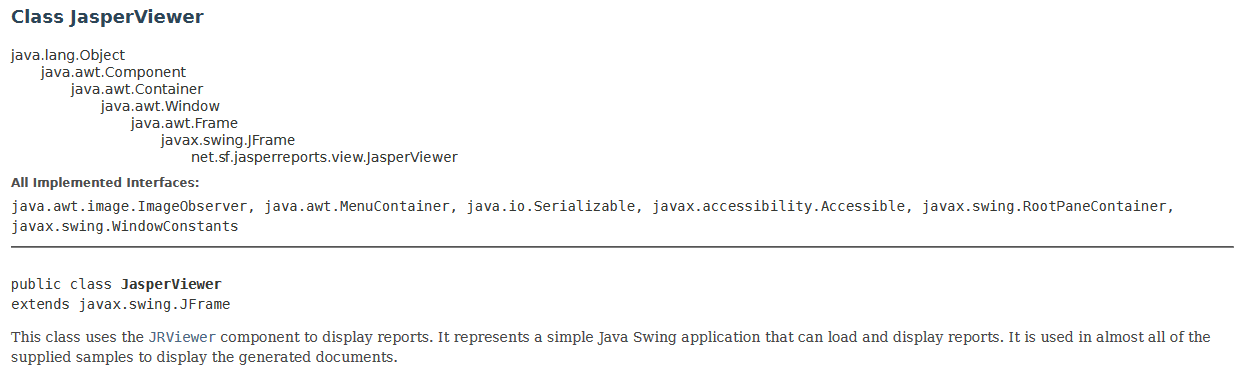
|  |  |
| --- | --- |
|  |  |
| También el parámetro es una interficie | El parámetro es un ExporterInput, que es una interficie implementada por estas tres clases. |
|  | |
|  | |
|  | |

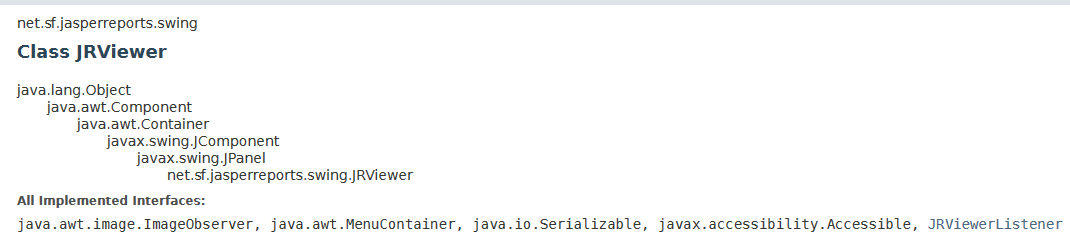
Por otro lado podemos visualizar el informe en una ventana :



Tenemos que tener un JasperPrint y pasarlo a un JasperViewer.

El problema de eso es que es un objeto de SWING:





Se puede crear un JRViewer y añadirlo a un contenerdor Swing.



Por último, para poner un report dentro de una aplicación javaFX…

