



5 Informes

Sumario

5 Informes.....	1
5.1 Convenciones empleadas.....	5
5.2 ¿Qué es un informe?.....	6
5.3 Informes incrustados y no incrustados.....	7
5.4 Generación de informes de forma automática: herramientas.....	7
5.4.1 Jaspersoft Studio Community.....	8
5.4.2 Interfaz de usuario de Jaspersoft Studio Community Edition.....	9
5.4.3 Elementos estructurales de un informe.....	11
5.4.4 Iniciar el origen de datos.....	12
5.4.5 Creación de un informe sencillo.....	12
5.4.6 Gestión de errores.....	23
5.4.7 Formatos de salida.....	24
5.5 Operaciones sobre los informes.....	24
5.5.1 Uso de parámetros en un informe.....	24
5.5.1.1 Filtrado de datos.....	26
5.5.2 Valores calculados.....	27
5.5.2.1 Recuentos y agrupamientos.....	28
5.5.2.2 Modificar encabezados y pies de página.....	31
5.5.3 Gráficos.....	33
5.5.4 Informes sobre consultas complejas.....	34
5.6 Repaso a la librería Jasper Report.....	35
5.6.1 Creación de informe desde una aplicación.....	35
5.7 Análisis del código obtenido.....	36

Índice de figuras










Figura 1: Esquema del proceso de creación de un informe.....	6
Figura 2: Jaspersoft Studio Community.....	9
Figura 3: Ventana de diseño de informes.....	10
Figura 4: Elementos estructurales de un informe.....	11
Figura 5: Asistente de creación de informes.....	13
Figura 6: Ruta de almacenamiento de informes.....	14
Figura 7: Origen de datos del informe.....	15
Figura 8: Consulta de selección de datos.....	16
Figura 9: Campos a seleccionar.....	17
Figura 10: Selección de campos.....	18
Figura 11: Cláusula de agrupamiento.....	19
Figura 12: Finalización del asistente.....	20
Figura 13: Establecimiento de la plantilla para generación del informe.....	21
Figura 14: Selección de campos para el informe.....	22
Figura 15: Esquema del proceso de generación de un informe.....	23
Figura 16: Parámetro USER_HEADER.....	26
Figura 17: Definición de consulta de agrupamiento.....	29
Figura 18: Establecimiento de los parámetros del informe.....	30
Figura 19: Previsualización del informe de agrupamiento.....	31
Figura 20: Generación de pie de página.....	32
Figura 21: Gráfico del precio de los productos.....	33
Figura 22: Proceso de generación de un informe.....	37



Material docente elaborado a partir de la base de los materiales formativos de FP en línea propiedad del Ministerio de Educación e Formación Profesional.

[Aviso Legal](#)

5.1 Convenciones empleadas

	Notas de introducción
	Aclaración
	Archivos de configuración, de registro...
	Casos de uso
	Código fuente
	Avisos o advertencias
	Capturas de pantalla, imágenes
	Actividades
	Enlace recomendado

	Iconos proporcionados por Papyrus Development Team https://github.com/PapyrusDevelopmentTeam/papyrus-icon-theme
---	---

5.2 ¿Qué es un informe?

Un informe es un documento que permite mostrar el contenido de un origen de datos aplicando un formato que permita a los usuarios adquirir **información**. Son un modo eficaz de presentar la información ejerciendo un control bastante preciso sobre cómo se presenta, ya que se permite: ver, formatear y resumir datos relacionados. Por ejemplo, en un informe que muestre como datos la duración media de las estancias en el hotel organizadas por meses, la información deducible por el usuario será los meses en los que cabe esperar una mayor ocupación.

Normalmente, **un informe se genera a partir de un diseño en el que se determina la distribución y configuración de los elementos a mostrar** y que podemos crear utilizando alguna herramienta gráfica. Luego, se combina el diseño con los datos actuales almacenados en el origen de datos volcando el resultado en documentos que faciliten su lectura e interpretación al usuario y posibilitando su impresión y almacenamiento de copias, aunque no se podrán modificar una vez generados. En caso de haber modificaciones en el origen de datos, tendremos que generar de nuevo el informe para reflejar esos cambios.



Esquema del proceso de creación de un informe

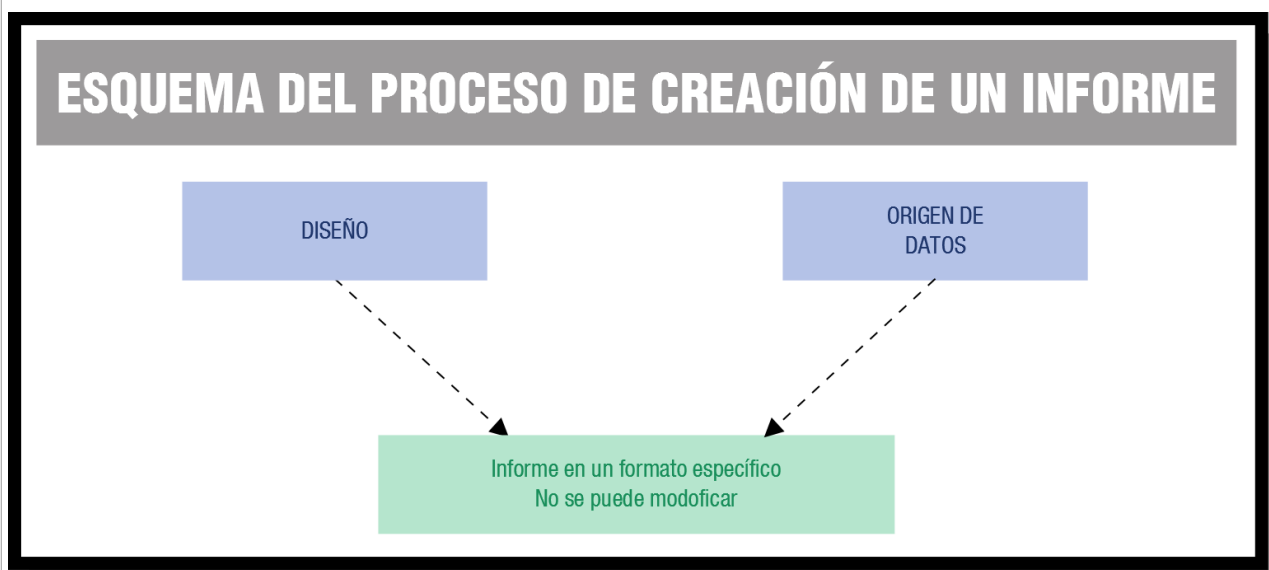


Figura 1: Esquema del proceso de creación de un informe

Ministerio de Educación y Formación Profesional (Elaboración propia)

¿Por qué los informes son un elemento fundamental en cualquier modelo de negocio?

- Porque necesitamos distribuir información a otras personas en un formato que puedan entender.
- Porque necesitamos controlar la distribución de la información cuando la imprimamos.
- Porque necesitamos hacer cálculos y mostrarlos de forma legible.

- Porque precisamos obtener información de los datos para poder tomar decisiones.

5.3 Informes incrustados y no incrustados

La forma de añadir el informe a la aplicación dependerá de cómo se cree el informe. Como hemos visto, es necesario tener la definición del informe y un origen de datos para rellenarlo, sin embargo, podemos crearlo dentro de la aplicación o tenerlo en un archivo independiente e insertarlo después.

- Un **informe incrustado** es un informe que se ha importado al proyecto o que se ha creado directamente en él. Cuando se crea un informe incrustado en una aplicación, se crea una clase contenedora para el informe. Esta clase formará parte del proyecto. Cuando se importa o se crea el informe en el proyecto, se crea una clase contenedora, con el mismo nombre que el informe. Esta clase contiene, o representa, el informe en el proyecto. Cuando ocurre esto, todo el código del proyecto interactúa con la clase del informe que se ha creado para representarlo, en vez de hacerlo con el propio archivo de informe original.

Al compilar el proyecto, tanto el informe como su clase contenedora se incrustan en el ensamblado, lo mismo que ocurriría con cualquier otro recurso del proyecto.

- Un **informe no incrustado** se ha generado con una herramienta específica aparte del proyecto y también se almacena independiente del proyecto. En este caso hay que planificar cómo se va a acceder y cargar el informe para interactuar con él. No existe una clase específica para manejar el informe. A un informe no incrustado siempre se obtiene acceso externamente y el SDK puede tener acceso a él de diversas formas:
 - El informe puede estar en la unidad de disco duro en una ruta de directorio de archivos.
 - El informe puede estar expuesto a través de un servicio web de informes.

Nunca se importan informes no incrustados en el proyecto y, por lo tanto, nunca se crea ninguna clase contenedora de informe, a diferencia de los informes incrustados. En su lugar, se carga el informe no incrustado en tiempo de ejecución.

5.4 Generación de informes de forma automática: herramientas

Los motores de informes permiten, mediante una interfaz gráfica de usuario determinar la posición, aspecto final, y configuración de los elementos que aparecen en el informe, generando automáticamente los ficheros con el diseño final. Hoy día existe gran cantidad de herramientas creadas para tal fin, tanto libres como propietarias, entre las que destacamos las siguientes:

SAP Crystal Reports:

Es la solución para la generación de informes de la empresa SAP, Crystal Solutions. Esta herramienta propietaria viene integrada en Visual Studio .NET, aunque dispone de SDK's para el desarrollo de aplicaciones .NET, Java y DOM. Es compatible con gran variedad de orígenes de datos, desde motores de base de datos, a hojas de cálculo, archivos XML o SAP.

Los informes se almacenan en un archivo de tipo .rpt, que contiene información tanto del origen de datos como del diseño. Admite informes incrustados y no incrustados.

Jaspersoft Studio Community:

Software libre perteneciente a JasperSoft. genera archivos XML (con extensión jrxml) que contienen el diseño del informe, para generar el informe se compila este archivo en otro de extensión.jasper y se rellena con el origen de datos. Tiene su propio lenguaje para la definición de expresiones llamado Groovy, aunque es compatible con Basic y Java.

La librería JasperReport permite combinar el diseño con el origen de datos para obtener el documento final mediante código Java.

Eclipse BIRT:

BIRT son las siglas de Business Intelligence Reporting Tools. Es un sistema de generación de informes para aplicaciones web, basadas en Java o en Java EE. Tiene dos componentes principales: un diseñador de informes basado en Eclipse y un componente que se puede agregar al servidor de aplicaciones y que genera informes en tiempo de ejecución. Ofrece un motor de gráficos y es compatible con gran cantidad de orígenes de datos, bien sea SGBD relacionales, archivos con formato, etc.

5.4.1 Jaspersoft Studio Community

Jaspersoft Studio Community es una herramienta de diseño de informes de código abierto que se utiliza para crear y diseñar informes visuales y tabulares. Desarrollado por Jaspersoft, este software proporciona una interfaz gráfica intuitiva para la creación de informes y ofrece diversas funciones para la generación de informes personalizados.

Entre algunas de las características más destacadas de Jaspersoft Studio Community Edition se incluye la capacidad de conectarse a diversas fuentes de datos, como bases de datos SQL, servicios web y archivos, así como la posibilidad de crear informes complejos con gráficos, tablas y otros elementos visuales. La herramienta es compatible con varios formatos de exportación, lo que permite compartir y distribuir informes en diferentes plataformas.

Además, Jaspersoft Studio Community Edition se integra con el servidor Jaspersoft, lo que facilita la publicación y la visualización de informes en entornos empresariales. Su naturaleza de código abierto fomenta la colaboración y la participación de la comunidad de usuarios en el desarrollo continuo de la herramienta.

Jaspersoft Studio Community Edition dispone de asistentes para la generación de informes, subinformes y plantillas, lo que facilita enormemente el trabajo del desarrollador, sin embargo, también permite la elaboración de informes partiendo desde cero.

Sea cual sea el procedimiento elegido, el objeto de usar una herramienta de diseño es reducir el coste de escribir todo el código necesario para obtener el informe, para ello, usaremos la librería de generación de informes **JasperReports**, que es una de las más usada en Java. Con esta librería, podremos enviar nuestros informes a un documento de texto, PDF, imagen o impresora. Se integra fácilmente en una aplicación java, aunque se puede utilizar de manera independiente.

En resumen, Jaspersoft Studio Community Edition permite diseñar informes mientras que JasperReports ejecuta y genera los informes en una aplicación Java.

5.4.2 Interfaz de usuario de Jaspersoft Studio Community Edition

Al ejecutar Jaspersoft Studio Community, vemos una interfaz parecida a la que aparece a continuación:

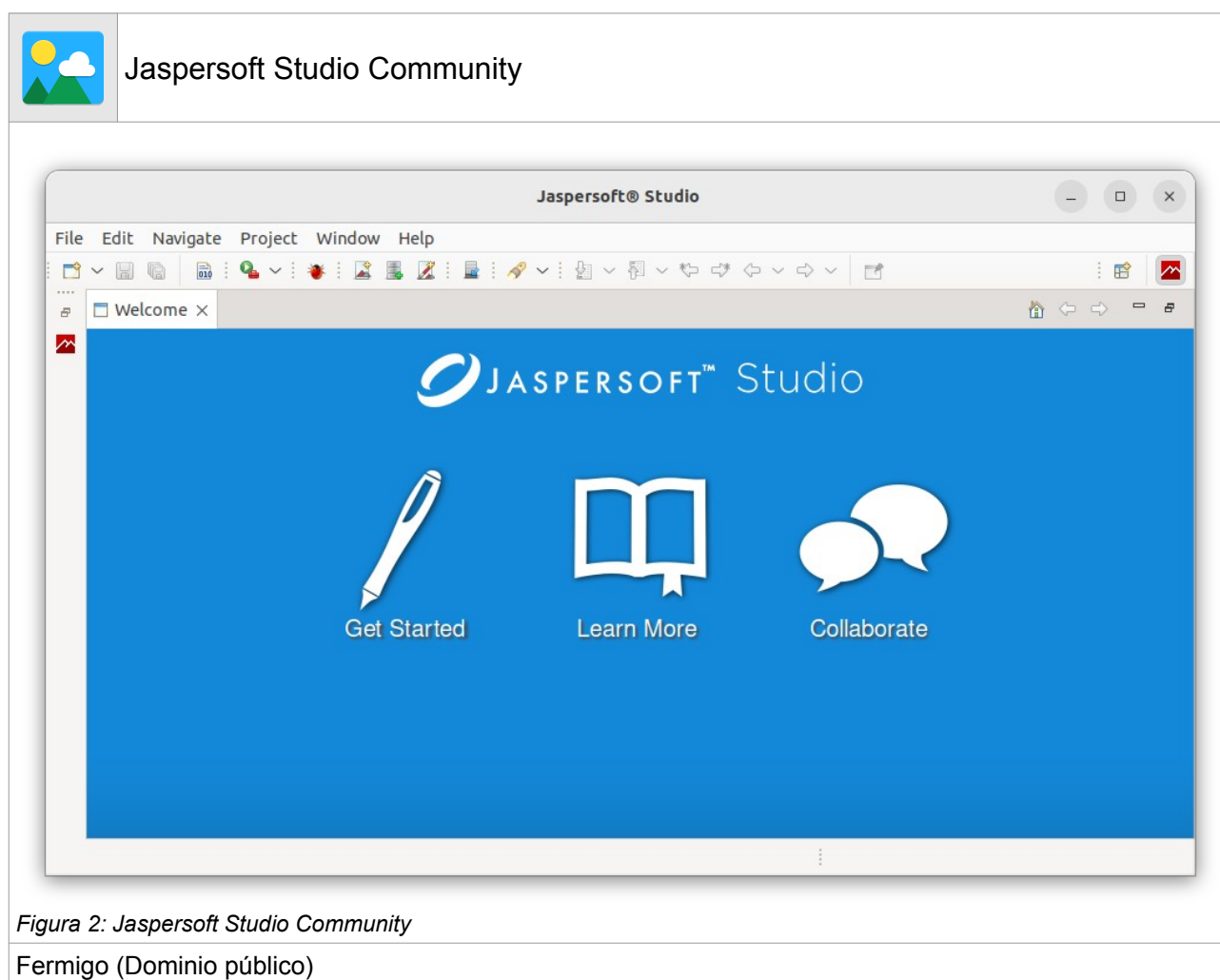


Figura 2: Jaspersoft Studio Community

Fermigo (Dominio público)



Ventana de diseño de informes

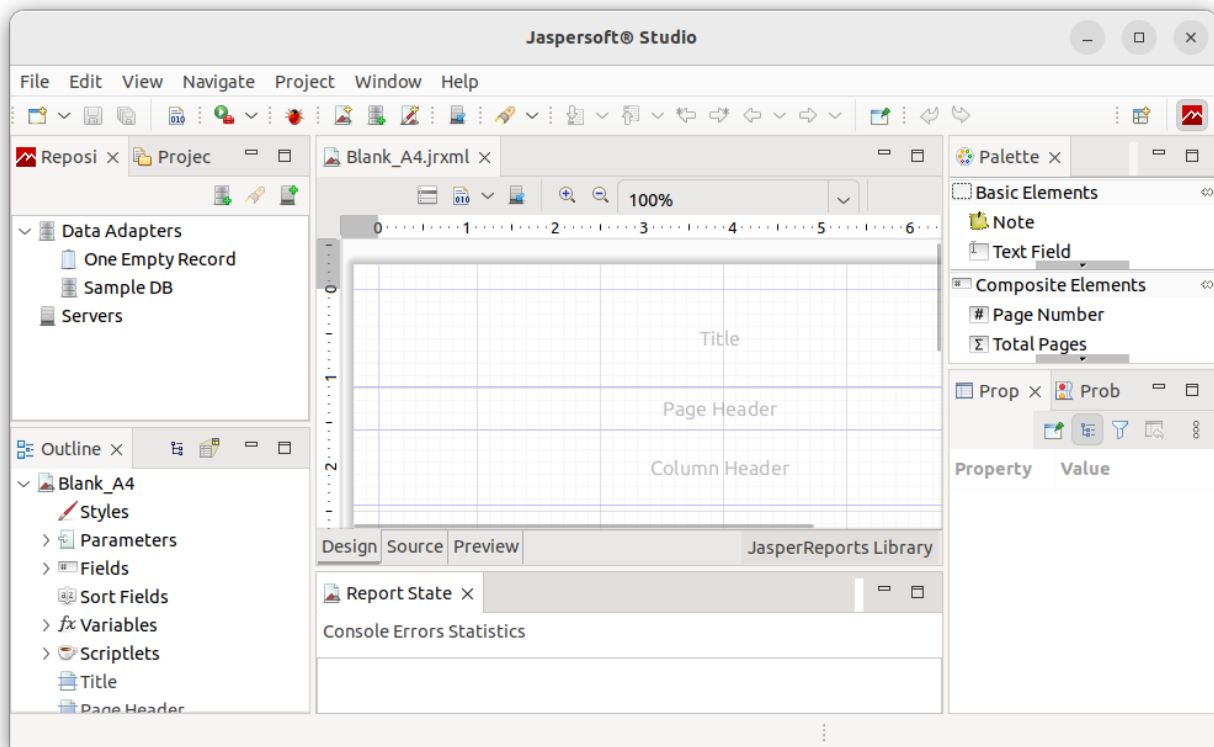


Figura 3: Ventana de diseño de informes

Fermigo (Dominio público)

Inspector de informes: muestra la estructura completa del informe, que se compone de muchos objetos (tales como campos, parámetros y variables), bandas (que son las secciones del documento) y elementos (tales como campos de texto, imágenes o gráficos).

El **diseñador de informes** permite diseñar visualmente el informe de arrastrando, posicionando, alineando y cambiando el tamaño de los elementos del informe.

La **paleta de elementos** contiene los elementos de diseño que pueden ser arrastrados dentro de una banda para mostrar los datos. Para visualizarla hacemos clic en ventana >> paleta.

La **hoja de propiedades** se utiliza para establecer las propiedades del componente seleccionado en el informe (como un campo, elemento, banda, grupo, u otros).

La **ventana de informe de problemas** contiene el listado de los errores encontrados al compilar el informe.

Sobre el diseñador de informes está el selector del origen de datos (selector de conexión), que muestra la conexión activa que se utiliza para ejecutar el informe.

5.4.3 Elementos estructurales de un informe

En la imagen se aprecia un informe vacío. A continuación, se detallan los aspectos estructurales más destacados de cada elemento:



Elementos estructurales de un informe

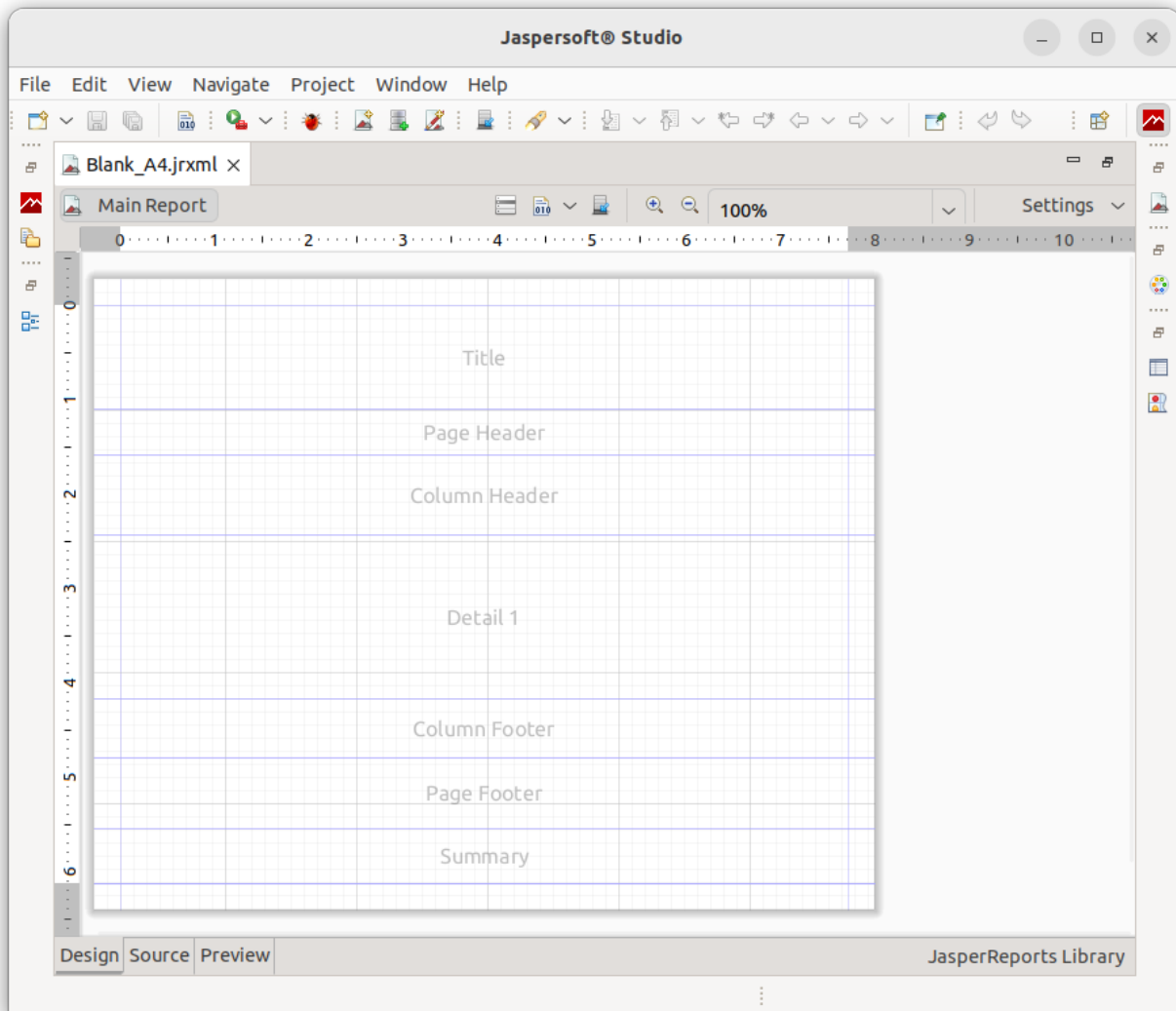


Figura 4: Elementos estructurales de un informe

Fermigo (Dominio público)

Título (Title): aparece sólo al inicio del informe. En esta sección se inserta el título del informe, por ejemplo "Informe de ventas del mes de marzo".

Encabezado de página (Page Header): aparece en la parte superior de cada página. Puede contener información como la fecha y hora, nombre de la organización, etc.

Encabezado de columna (Column Header): se utiliza para listar los nombres de los campos que se van a presentar (desplegar). Por ejemplo: "Producto", "Proveedor", "Precio de compra", "Precio de venta al público", "Beneficio", etc.

Detalle (Detail): En esta sección se despliegan los valores correspondientes a las entradas de campos definidas en la sección anterior. Por ejemplo: "Barra de cortina metálica. 3M", "Cofrilsa distribuidores, S.A.", "2,25", "4,99", "205,30".

Pie de columna (Column Footer): Puede presentar información resumida para cada uno de los campos. Por ejemplo: "Beneficio total del mes: 1245".

Pie de página (Page Footer): Aparece en la parte inferior de cada página. En esta parte podemos incluir, entre otras cosas, un contador de páginas, por ejemplo: "Página 1/7".

Resumen (Summary): Esta sección se usa para proporcionar información resumida de los campos presentes en la sección "detalle". Por ejemplo, para el caso del beneficio por producto se puede definir un objeto gráfico tipo "pie" para tener una mejor comparación y comprensión visual de los datos.

A cada uno de estos elementos estructurales se le denomina **banda**.

5.4.4 Iniciar el origen de datos

El ciclo de vida de un informe, en cualquier caso, pasa por una serie de pasos que se detallan a continuación, pero lo primero que necesitamos, es tener el origen de datos preparado para poder acceder a los datos que vamos a utilizar para construir nuestros informes.

Sin un origen de datos válido el informe no servirá para nada, ya que se compone de la combinación de diseño y datos, de hecho, la estructura del informe depende de los datos a mostrar. Sólo en casos muy particulares se permiten informes con orígenes de datos vacíos. Por eso es necesario tener el motor de base de datos funcionando desde la fase de diseño del informe.

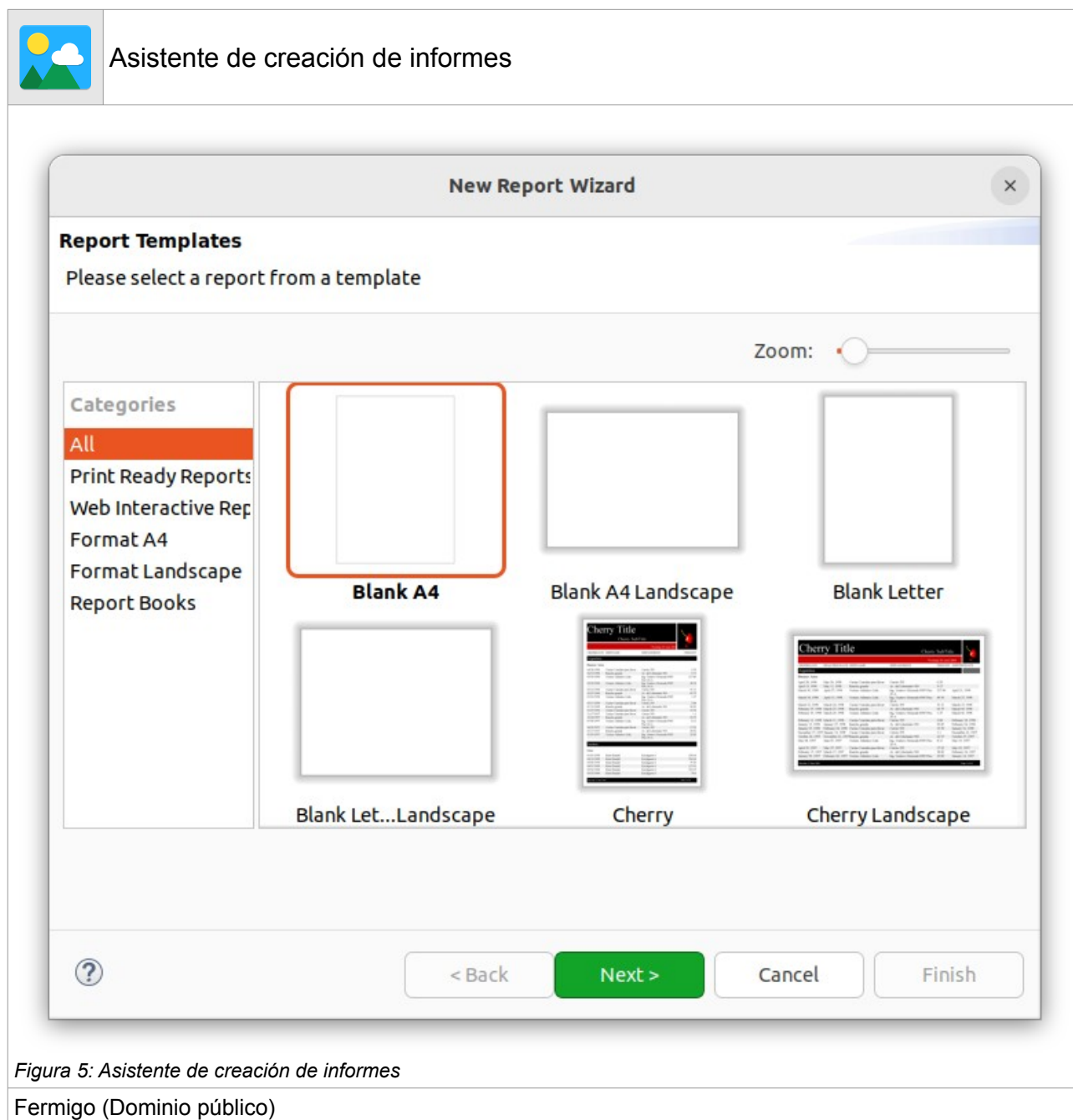
5.4.5 Creación de un informe sencillo

El proceso de creación de informes consta de los siguientes pasos principales:

1. Crear un **origen de datos** o una conexión al origen de datos utilizado para llenar el informe. Un origen de datos es cualquier fuente desde donde la herramienta pueda obtener el conjunto de registros con los que se rellenará el informe.
2. **Crear** el informe nuevo.
3. **Seleccionar los datos** que formarán parte del informe.
4. **Diseñar el informe**, incluyendo la disposición de sus elementos y parámetros para representar los datos.
5. **Ejecutar el informe**, a partir del archivo de origen se genera un archivo compilado y se rellena con los datos para la exportación o en pantalla.

Para crear un informe accederemos en Jaspersoft Studio Community Edition al menú File → New → Jasper Report

Podemos ver que existen diferentes formatos predefinidos que podemos utilizar, En nuestro caso, vamos a proceder a crear un informe vacío. Para ello, seleccionamos Blank A4 y pulsamos Siguiente.



A continuación, introducimos el nombre que va a tener el informe. Podemos indicar en que carpeta podemos almacenar el fichero, el nombre del proyecto en donde queremos crearlo y la ruta absoluta del fichero que se va a crear.



Ruta de almacenamiento de informes

New Report Wizard

Report file
Please select your reports file name with .jrxml extension.

Enter or select the parent folder:

MyReports

> MyReports

File name: Blank_A4.jrxml

? < Back Next > Cancel Finish

Figura 6: Ruta de almacenamiento de informes

Fermigo (Dominio público)

A continuación, seleccionamos la conexión que queremos utilizar, que en nuestro caso se denomina "Conexión MySQL" e introduciremos la orden SQL que utilizaremos para recuperar los datos que se van a presentar en el informe.



Origen de datos del informe

The screenshot shows the 'New Report Wizard' dialog box, specifically the 'Data Source' step. The title bar reads 'New Report Wizard' with a close button (X). Below the title bar, the section is labeled 'Data Source' with the instruction 'Select a Data Source and design the query.' The 'Data Adapter' dropdown menu is set to 'Sample DB - Database JDBC Connection', with a 'New...' button to its right. Below this, a tree view displays the database structure: 'INFORMATION_SCHEMA', 'PUBLIC (CURRENT)', and 'SYSTEM_LOBS'. To the right of the tree view is a large empty text area for writing queries. At the bottom of the text area are tabs labeled 'Texts', 'Outline', and 'Diagram'. The 'Texts' tab is currently selected. At the bottom of the dialog box are four buttons: a help button (question mark icon), '< Back', 'Next >', and a green 'Finish' button.

Figura 7: Origen de datos del informe

Fermigo (Dominio público)

En nuestro caso vamos a realizar una consulta a la base de datos para recuperar el nombre de nuestro cliente junto con la ciudad en la que se encuentran. Para ello, escribiríamos la orden SQL: `SELECT ID, FIRSTNAME, LASTNAME FROM ADDRESS.`



Consulta de selección de datos

New Report Wizard

Data Source

Data Adapter: Sample DB - Database JDBC Connection

Tree View:

- INFORMATION_SCHEMA
- PUBLIC (CURRENT)
 - Global Temporary
 - System Table
 - Table
 - ADDRESS
 - ID (Primary Key)
 - FIRSTNAME**
 - LASTNAME
 - STREET
 - CITY
 - PRODUCT
 - DOCUMENT
 - POSITIONS

SQL Query:

```
1 SELECT ID, FIRSTNAME, LASTNAME
2 FROM ADDRESS
```

Texts | Outline | Diagram

< Back | Next > | Cancel | Finish

Figura 8: Consulta de selección de datos

Fermigo (Dominio público)

A continuación, podemos indicar que campos queremos que aparezcan en el informe. A pulsar Siguiente, podemos agrupar la información que queremos mostrar. De esta manera podemos clasificar los resultados obtenidos a la hora de presentarlos en el informe. Para finalizar el diseño del informe, pulsaremos Finish.



Selección de campos

New Report Wizard [X]

Fields
Please select dataset fields

Dataset Fields		Fields	
<input checked="" type="checkbox"/> ID	>		Up
<input type="checkbox"/> FIRSTNAME	>>		Down
<input type="checkbox"/> LASTNAME	<		
	<<		

[?] < Back Next > Cancel Finish

Figura 9: Campos a seleccionar

Fermigo (Dominio público)



Selección de campos

New Report Wizard [X]

Fields
Please select dataset fields

Dataset Fields		Fields	
	>	ID	Up
	>>	FIRSTNAME	Down
	<	LASTNAME	
	<<		

[?] < Back Next > Cancel **Finish**

Figura 10: Selección de campos

Fermigo (Dominio público)



Cláusula de agrupamiento

New Report Wizard [X]

Group By
Please select fields to Group By

Dataset Fields		Fields	
<input type="checkbox"/> ID	>		Up
<input type="checkbox"/> FIRSTNAME	>>		Down
<input type="checkbox"/> LASTNAME	<		
	<<		

☐ Use the group fields as sort fields. Select this option if you want to aggregate all the group fields with the same value and not only the consecutive ones

[?] < Back Next > Cancel Finish

Figura 11: Cláusula de agrupamiento

Fermigo (Dominio público)



Finalización del asistente

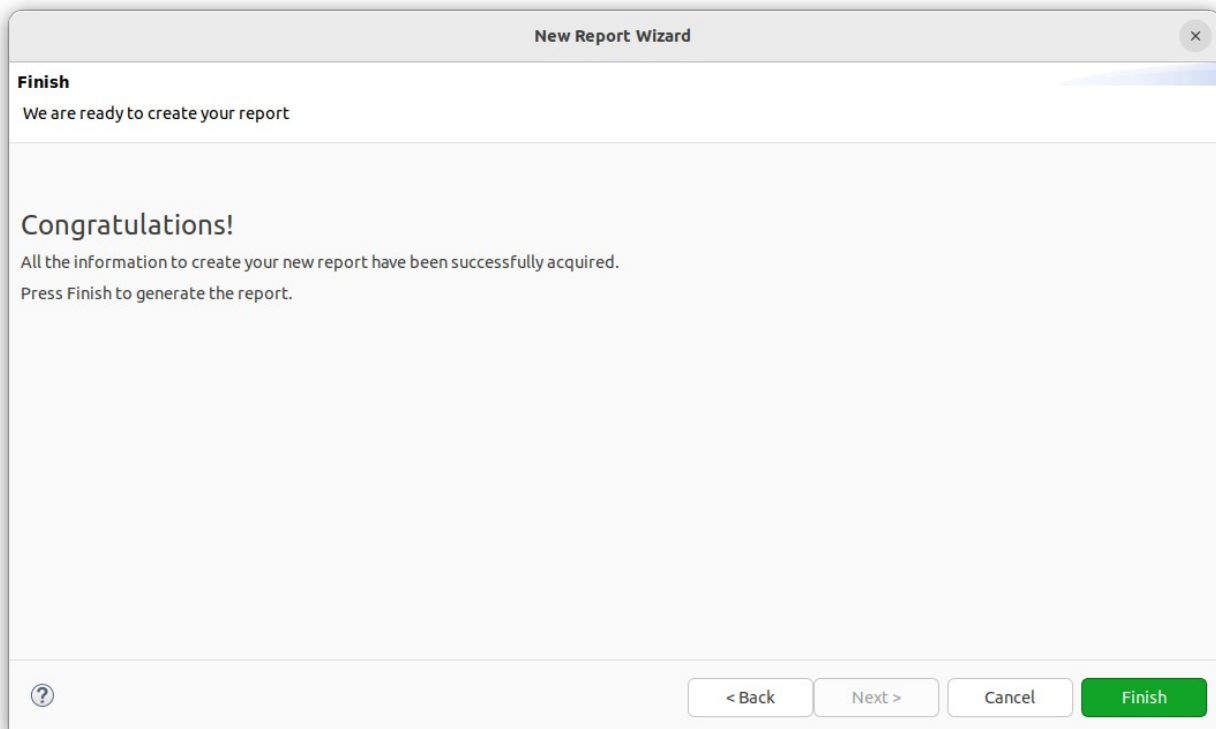


Figura 12: Finalización del asistente

Fermigo (Dominio público)

Llegado este momento, ya tenemos creado el informe vacío y configurado para que pueda recuperar la información que vamos a presentar en el informe.



Establecimiento de la plantilla para generación del informe

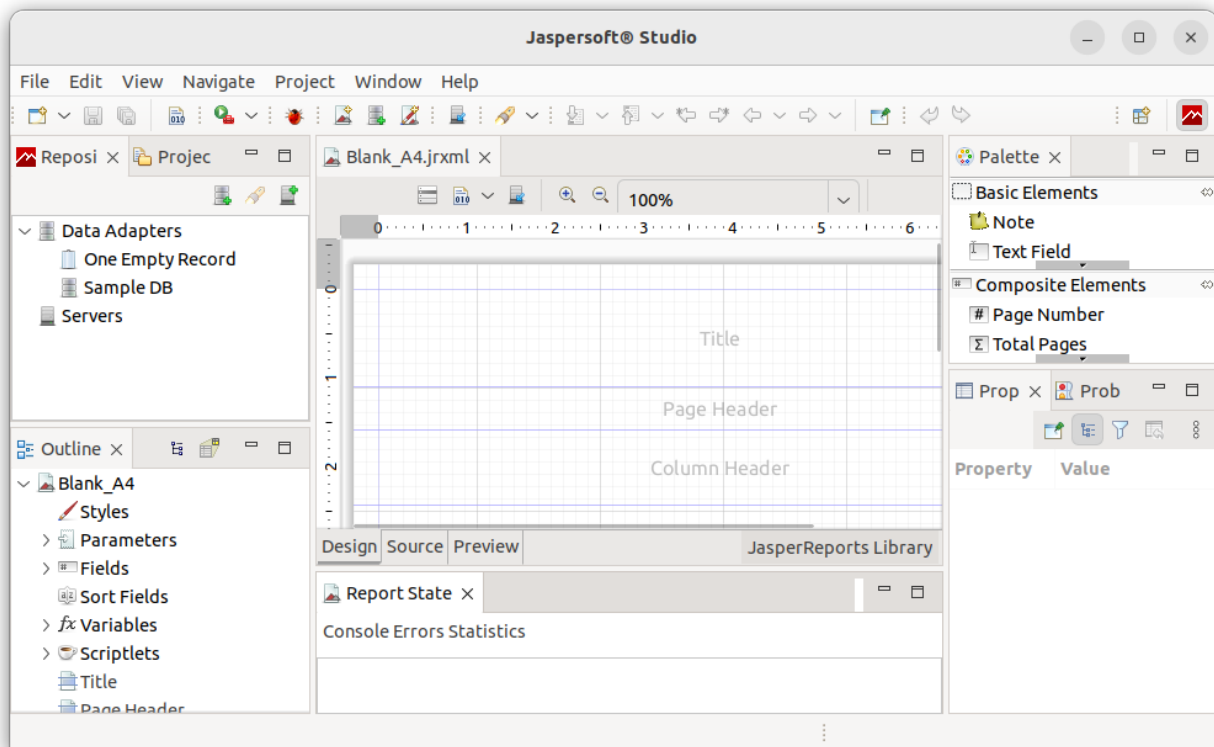


Figura 13: Establecimiento de la plantilla para generación del informe

Fermigo (Dominio público)

Para finalizar, la creación del informe, deberemos de indicar que información queremos que se nos muestre en el informe. Para ello, podemos utilizar el panel Report Inspector para diseñar el informe. A través de este panel, podemos seleccionar los elementos que queramos que aparezcan en el informe. Para ello, solo hay que seleccionarlo y arrastrarlo hacia la parte del informe en donde queramos que se muestren.

Dentro del panel Outline, podemos acceder al apartado **Fields**, en donde se nos mostrará los campos que definimos en la configuración de la consulta a la base de datos. Estos campos deberemos de incluirlos dentro de cada banda del informe en donde queramos que se muestre.



Selección de campos para el informe

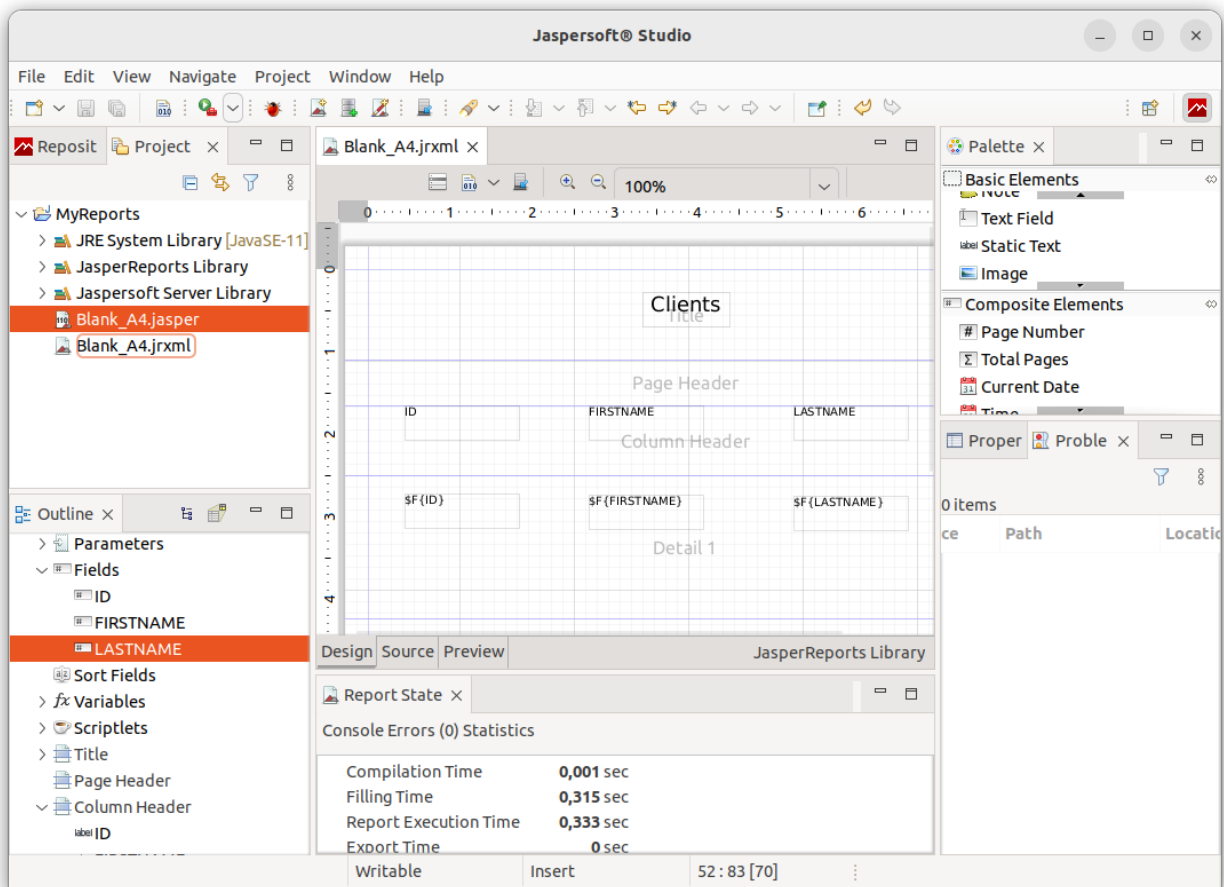


Figura 14: Selección de campos para el informe

Fermigo (Dominio público)

Cuando el objeto de campo se arrastra dentro de la banda de detalle, se crea un elemento de campo de texto y establece el campo de texto la expresión de ese elemento a **\$F{Nombre_Campo}**. Esta es una expresión simple para imprimir el valor del campo Nombre_Campo.

Para añadir otros elementos (tales como líneas o etiquetas), arrástralos desde la Paleta **Reports Elements** en la vista del diseñador para, a continuación, cambiar el tamaño y organizarlos como desee.

Desde la ventana de **Propiedades** se podrán modificar en su aspecto (color, tipo de fuente, tamaño, etc).

En este caso, añadimos una etiqueta en la banda de título para el título del informe, se añade etiquetas de las columnas con elementos de la etiqueta colocada en la banda de cabecera de la columna, y colocamos una fina línea justo debajo de los campos de texto en la banda de detalle.

Por último, cambiamos el tamaño de algunas bandas y eliminamos los demás mediante el establecimiento de una su altura a 0. Cambiaremos la altura de una banda arrastrando su borde inferior. Una forma directa para reducir la altura de la banda es hacer doble clic en su borde inferior, esto se establece en el borde inferior de su elemento más bajo.

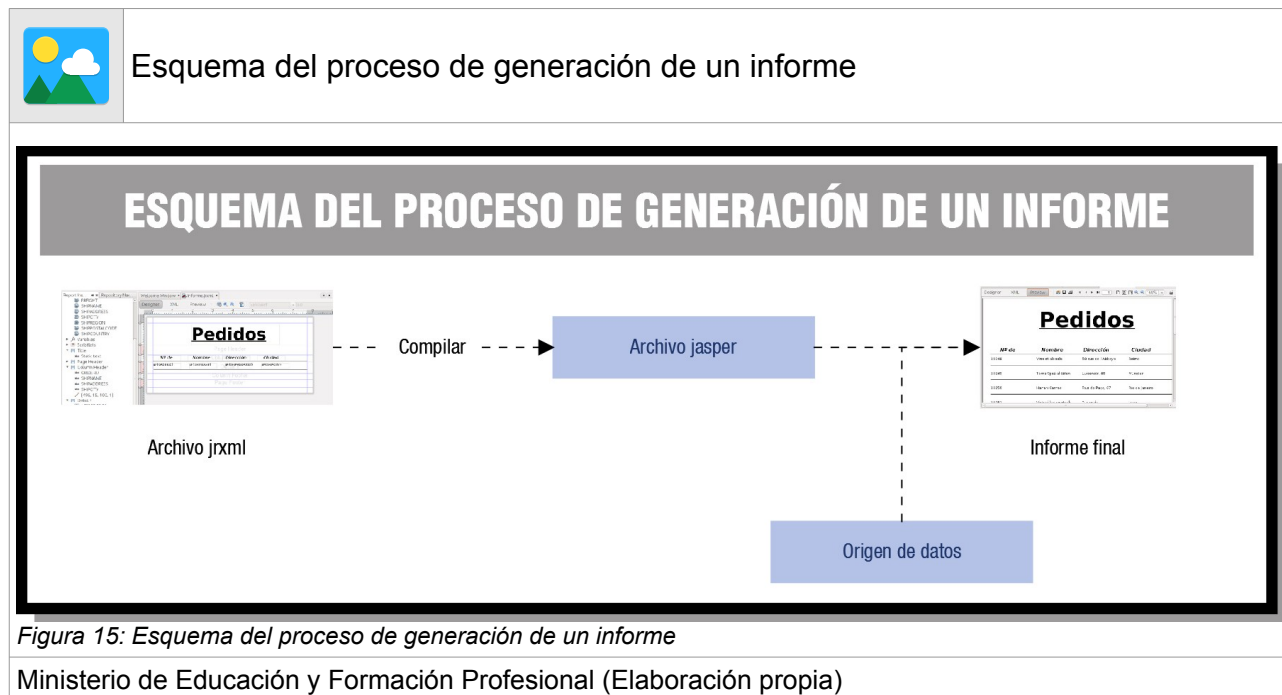
Para ver el resultado del informe que hemos creado solo hay que pulsar el botón **Preview**. Al pulsar este botón, se compila primero el fichero fuente jrxml. Si la compilación es correcta, el archivo producido se carga y se llena con la conexión activa.

5.4.6 Gestión de errores

Al generar la vista previa, Jaspersoft Studio realiza una serie de operaciones para crear el informe final.

La primera operación consiste en compilar el archivo fuente, con extensión .jrxml en un archivo de Jasper, con extensión .jasper. Este primer paso puede fallar si los elementos no están colocados correctamente (por ejemplo, si un elemento se coloca fuera de una banda), o si una expresión en el informe tiene errores y no puede ser compilado.

Si la compilación se ejecuta correctamente, el archivo producido Jasper se carga y se llena con la conexión activa o fuente de datos. Esta segunda operación, otra vez puede conducir a errores, por ejemplo, si la base de datos se hace referencia no está activo, una consulta no válida se ha proporcionado, o en un campo nulo producido un error en una expresión durante el proceso de llenado. Finalmente, si todas las operaciones se completan sin errores, el informe se muestra en el visor integrado.



Los errores se muestran en la ventana de Problems y resultados de la operación se muestran en la salida de Jaspersoft Studio.

5.4.7 Formatos de salida

El objetivo de un informe es presentar información en un formato que sea accesible y fácil de distribuir, por lo que una vez generado se vuelca en archivos de texto, que se puedan almacenar e imprimir. Cualquier motor de informes debe ser capaz que exportar sus documentos a diferentes formatos de uso cotidiano, como PDF o HTML.

En concreto Jaspersoft Studio gestiona esto desde el menú Vista previa (**Preview**), donde podemos seleccionar el formato final del informe.

Para generar de nuevo un informe que no se ha modificado, haga clic en **ejecutar de nuevo** en la barra de herramientas de vista previa. Una nueva ejecución de un informe es útil cuando un subinforme cambia, cuando cambia la fuente de datos, o cuando se desea ejecutar el informe con parámetros de entrada diferentes.

Para almacenar el resultado en un formato diferente al que aparece por defecto, por ejemplo, PDF debemos pulsar el botón Guardar (Save) y elegir el formato de salida que queramos utilizar.

Jaspersoft Studio es capaz de identificar el visor adecuado para cada formato automáticamente en función de los valores por defecto del sistema.

5.5 Operaciones sobre los informes

Efectivamente, Ana tiene razón. Un informe suele ser algo más que la presentación gráfica de los datos de una tabla de la base de datos. Lo habitual es realizar ciertas operaciones sobre el informe para obtener más información o mejor el formato de presentación, entre otras cosas veremos cómo:

- Hacer cálculos sobre los datos para mostrar información resumida, como promedios o sumas.
- Parametrizar el informe para que se pregunte al usuario que datos desea mostrar.
- Filtrar los datos.
- Aplicar encabezados y pies de página a las hojas del informe.
- Etc.

5.5.1 Uso de parámetros en un informe

La utilidad de los **parámetros** en un informe es permitir generar diferentes resultados a partir de un mismo archivo de diseño, en función de un dato que podemos cambiar nosotros directamente o a través de la aplicación final.

Un **parámetro** se define por un **nombre** y una **clase**, cualquier clase de Java es una clase de parámetro válido. Se emplea para pasar información al informe en tiempo de ejecución. Por ejemplo, un parámetro de tipo `java.sql.Connection` se puede utilizar para rellenar un informe integrado, mientras que un parámetro `java.lang.Boolean` se puede utilizar para mostrar u ocultar una sección del informe.

Para administrar los parámetros, utilizamos el apartado Parameters del inspector de informe. Desde aquí es posible agregar y quitar parámetros utilizando el menú contextual. Podemos modificar un parámetro seleccionándolo y editando sus propiedades. Podemos modificar entre otras cosas:

- **Name:** nos permite asignar un nombre a un parámetro.
- **Parameter class:** lugar en donde ubicaremos la clase asociada al parámetro.
- **Default Value Expression:** valor predeterminado asociado al parámetro. Esta expresión es evaluada por JasperReports sólo cuando un valor para el parámetro no ha sido proporcionado por el usuario en tiempo de ejecución. Si el parámetro es de tipo String, el valor que introduzcamos debe de ir encerrado entre comillas.
- **Use as prompt:** activaremos esta opción cuando el valor del parámetro sea asignado durante el instante de tiempo en el cual se está generando el informe. En caso de no activarlo, el parámetro tendrá asignado el valor Default Value Expression.

Los parámetros se pueden clasificar en:

- **Parámetros integrados** están disponibles de forma predeterminada y contienen información en tiempo de ejecución. Algunos de los más importantes son **REPORT_CONNECTION**, que tiene la conexión JDBC para ejecutar la consulta SQL del informe (si el informe está lleno con una conexión JDBC), el **REPORT_DATA_SOURCE** que contiene, en su caso, la fuente de datos utilizada para llenar el informe, o el **REPORT_LOCALE** que contiene la configuración regional utilizada para rellenar el informe, y así sucesivamente. Los parámetros integrados no pueden ser modificados o eliminados. En cuanto al tipo de parámetro podemos encontrar:
- **Parámetros de usuario:** el programador determina su nombre, clase y valor. se pueden configurar para que sean insertados por el usuario cuando se ejecute el informe. Por ejemplo, vamos a crear un parámetro llamado **USER_HEADER** de tipo **String** con la propiedad **"Use as prompt" activa**. Si arrastramos el parámetro desde el inspector de informe a la banda título, Jaspersoft Studio creará un campo de texto para mostrar el valor del parámetro. Por lo tanto, al generar el informe nos solicitará un valor para este parámetro que se visualizará como título del informe.



Ejemplo de parámetro USER_HEADER

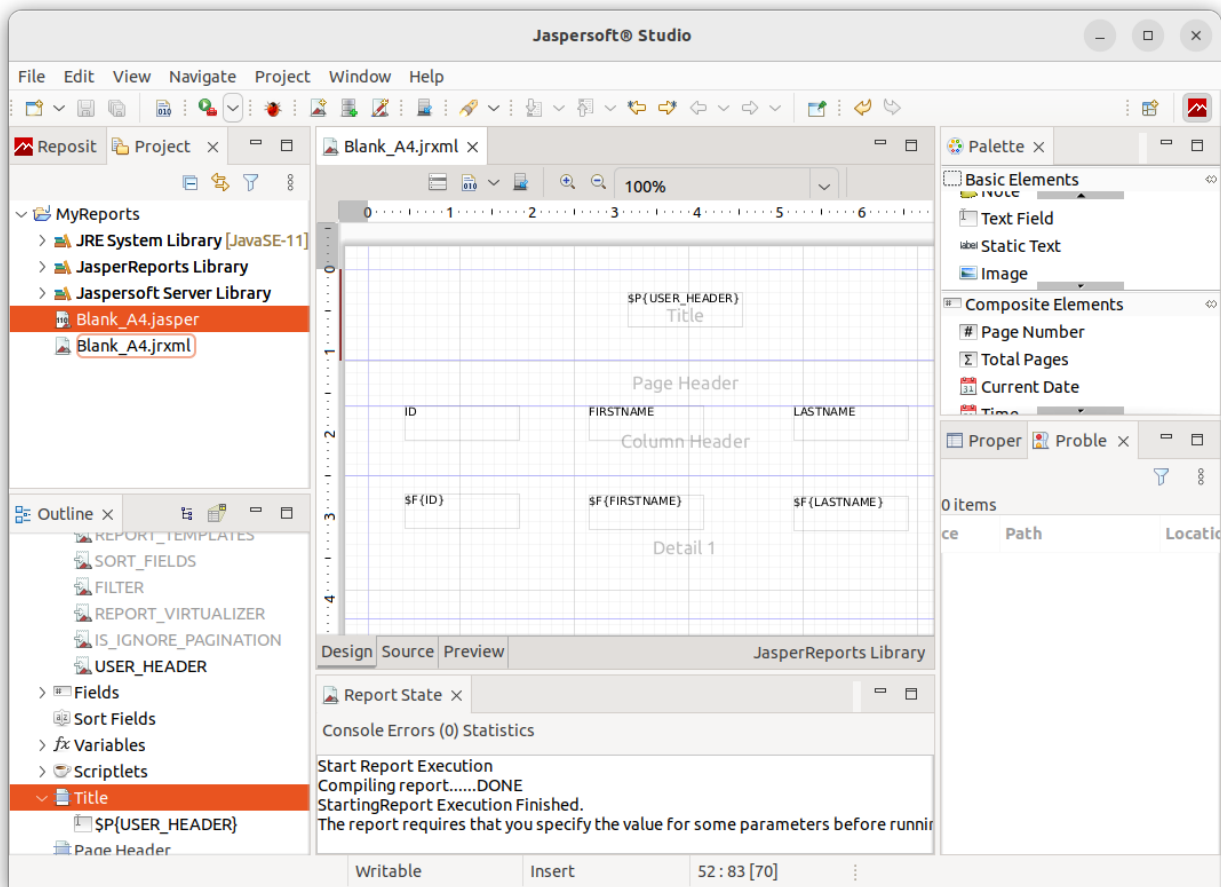


Figura 16: Parámetro USER_HEADER

Fermigo (Dominio público)

Uno de los usos más extendido de los parámetros es el filtrado de datos, que veremos a continuación.

5.5.1.1 Filtrado de datos

Se pueden utilizar parámetros en las consultas SQL para filtrar los registros en la condición where o para añadir o sustituir elementos de la consulta SQL o incluso pasar toda la cadena SQL para ejecutar. Tenemos dos posibilidades:

En el primer caso los parámetros se utilizan como parámetros estándar SQL, por ejemplo:

```
SELECT ID, FIRSTNAME, LASTNAME FROM ADDRESS WHERE LASTNAME = $P{LASTNAME}
```

En este ejemplo, **LASTNAME** es un parámetro de tipo `java.lang.String` (Text) que contiene el apellido a seleccionar. Este parámetro se puede pasar al informe desde la aplicación que lo ejecuta para seleccionar sólo un apellido específico.

El parámetro aquí es un verdadero parámetro SQL, lo que significa que la consulta se ejecutará mediante una sentencia como la siguiente:

```
SELECT ID, FIRSTNAME, LASTNAME FROM ADDRESS WHERE LASTNAME = ?
```

y el valor del parámetro **LASTNAME** entonces pasa a la instrucción. El segundo caso se trata de construcciones como:

```
SELECT * FROM ADDRESS ORDER BY $P!{campos}
```

El parámetro será tratado como un campo de SQL. JasperReports tendrá en cuenta este parámetro como una especie de marcador de posición (ten en cuenta la sintaxis especial de `$P!{}`), que será reemplazado por el valor de texto del parámetro (que en este caso puede ser, por ejemplo, "FIRSTNAME DESC").

Con la misma lógica, una consulta puede pasarse íntegramente mediante un parámetro del siguiente modo:

```
$P! {MYQUERY}
```

El número de parámetros en una consulta es arbitrario. Al pasar un valor utilizando la sintaxis `$P!{}`, el valor del parámetro se toma tal cual, el usuario es responsable de la exactitud del valor pasado: la resolución de la sentencia SQL no se realiza por JasperReports en este caso.

Cuando se utilizan parámetros en una consulta, para que JasperReports pueda recuperar los campos disponibles de la consulta, se debe fijar un valor por defecto para el parámetro.

5.5.2 Valores calculados

Podemos hacer cálculos usando **variables**. Podemos añadir y eliminar variables desde la zona variables del inspector de informe y la hoja de **propiedades** y, al igual que los parámetros, se definen por su **nombre**, **clase** y **valor**. Para editar una variable, se seleccione en el inspector de informe y modificamos la hoja de propiedades.

- Existen **variables predefinidas** que sirven para hacer recuentos de elementos propios del informe, como **PAGE_NUMBER** que contiene el número total de páginas,

o el **REPORT_COUNT** que tiene el número de registros procesados en el momento, **COLUMN_NUMBER** que contiene el número de columnas, **PAGE_COUNT** que contiene el número de registros que han sido procesados por página. Este tipo de variables no pueden ser modificadas o eliminadas.

- También se pueden crear **variables de usuario**, que sí pueden cambiar. Será el programador el encargado de asignar nombre (**NAME**), tipo de datos (**VARIABLE CLASS**) y valor inicial a la variable (**INIT VALUE EXPRESSION**).

A diferencia de los parámetros, que toman un valor establecido por el programador como valor por defecto, o asignado desde la aplicación final en tiempo de ejecución, pero no cambia, el valor de las variables cambia mientras se va creando el informe, y pueden ser evaluadas en diferentes momentos para tomar el valor más adecuado.

Veamos ahora algunos de los usos más comunes de una variable.

5.5.2.1 Recuentos y agrupamientos

Un recuento consiste en aplicar la función resumen suma a un campo concreto de una consulta. Son útiles para calcular totales y subtotales.

Realizaremos un informe sencillo que muestra un listado en donde se muestre el número de clientes que tienen el mismo apellido. La sentencia SQL que utilizaremos es la que aparece a continuación:

```
SELECT COUNT(FIRSTNAME) AS COUNTER, LASTNAME FROM ADDRESS GROUP BY LASTNAME
```

Comenzaremos creando un nuevo informe vacío. Modificaremos la consulta asociada al informe para introducir la consulta anterior.



Definición de consulta de agrupamiento

Dataset and Query Dialog

Sample DB ▾

Query Java Bean

Language **sql** ▾ Read Fields

> INFORMATION_SCHEMA

> PUBLIC (CURRENT)

> SYSTEM_LOBS

```
1 SELECT COUNT(FIRSTNAME) as COUNTER, LASTNAME
2 FROM ADDRESS GROUP BY LASTNAME
```

Texts Outline Diagram

Field Name	Class Type	Description	Time zone	Column nam	Column labe	Column inde	Properties
COUNTER	java.lang.Lor						2 Properties
LASTNAME	java.lang.Str						3 Properties

Add Edit Delete Up Down

Fields Parameters Sorting Filter Expression Data preview

Cancel OK

Figura 17: Definición de consulta de agrupamiento

Fermigo (Dominio público)

Arrastraremos los campos **COUNTER** y **LASTNAME** a la banda de detalle, en la banda Column Header se modificarán las etiquetas de cada columna si fuese necesario.



Establecimiento de los parámetros del informe de agrupamiento

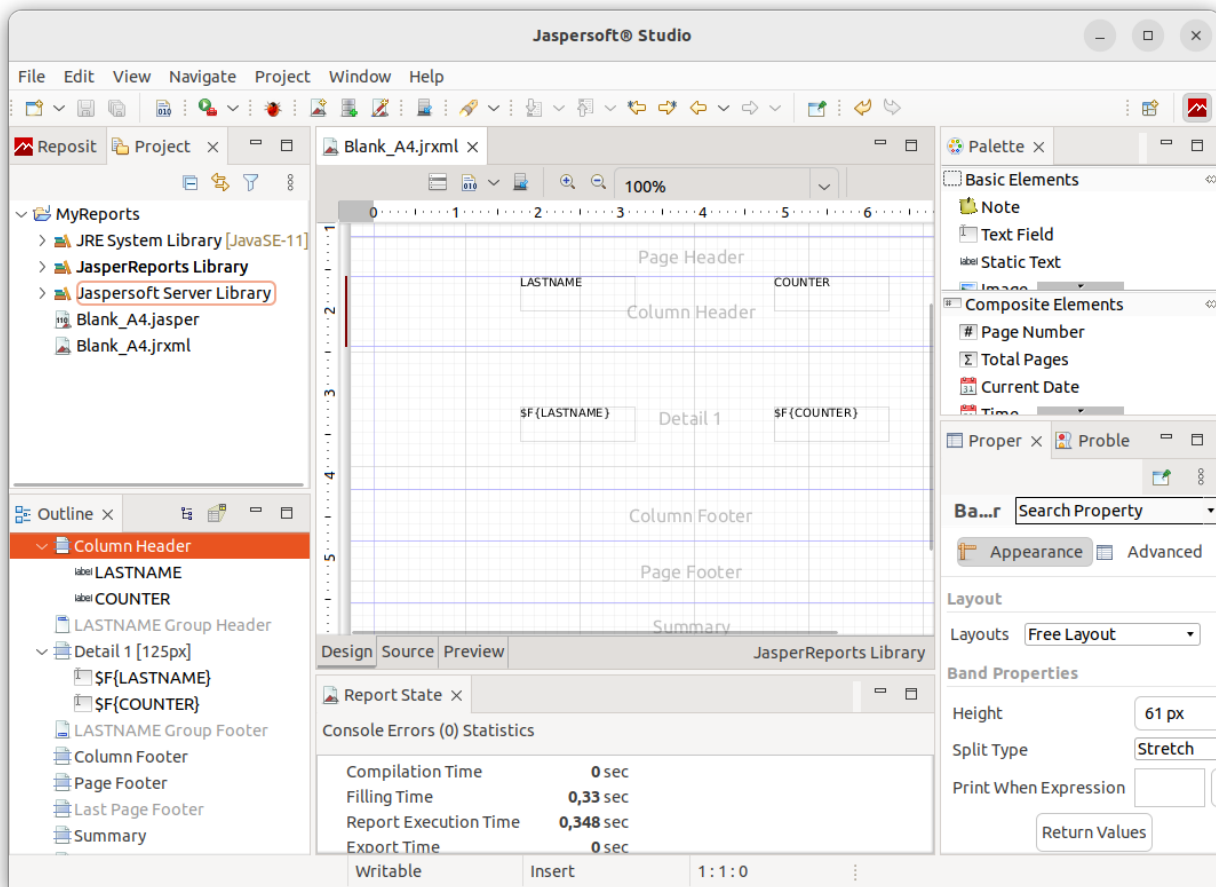


Figura 18: Establecimiento de los parámetros del informe

Fermigo (Dominio público)

En la banda del título (Title) introduciremos una etiqueta (static text) con el texto que se desee. Podemos modificar el color del texto de la etiqueta y el fondo utilizando las propiedades ForeColor y Bakcolor junto con la propiedad Opaque.

Si pulsamos el botón Preview para visualizar el informe, nos aparecerá:



Previsualización del informe de agrupamiento

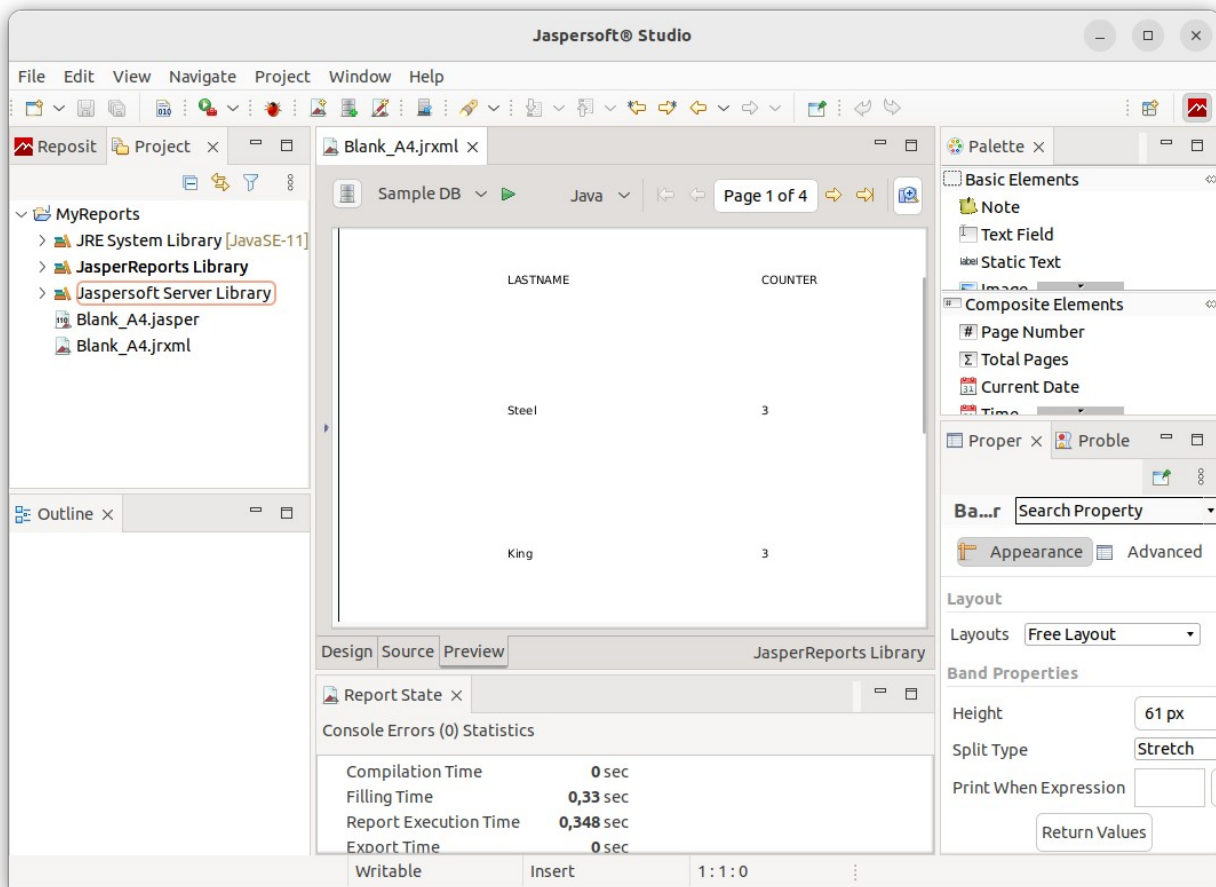


Figura 19: Previsualización del informe de agrupamiento

Fermigo (Dominio público)

5.5.2.2 Modificar encabezados y pies de página

Para añadir la **página X de Y** en el pie de página de un informe, sólo tienes que arrastrar la herramienta página X de Y de la paleta Composite Elements en la banda **PAGE_FOOTER**. Como se escribe Page en inglés, editamos la etiqueta y sustituimos la palabra "Page" por "Página".



Generación de pie de página

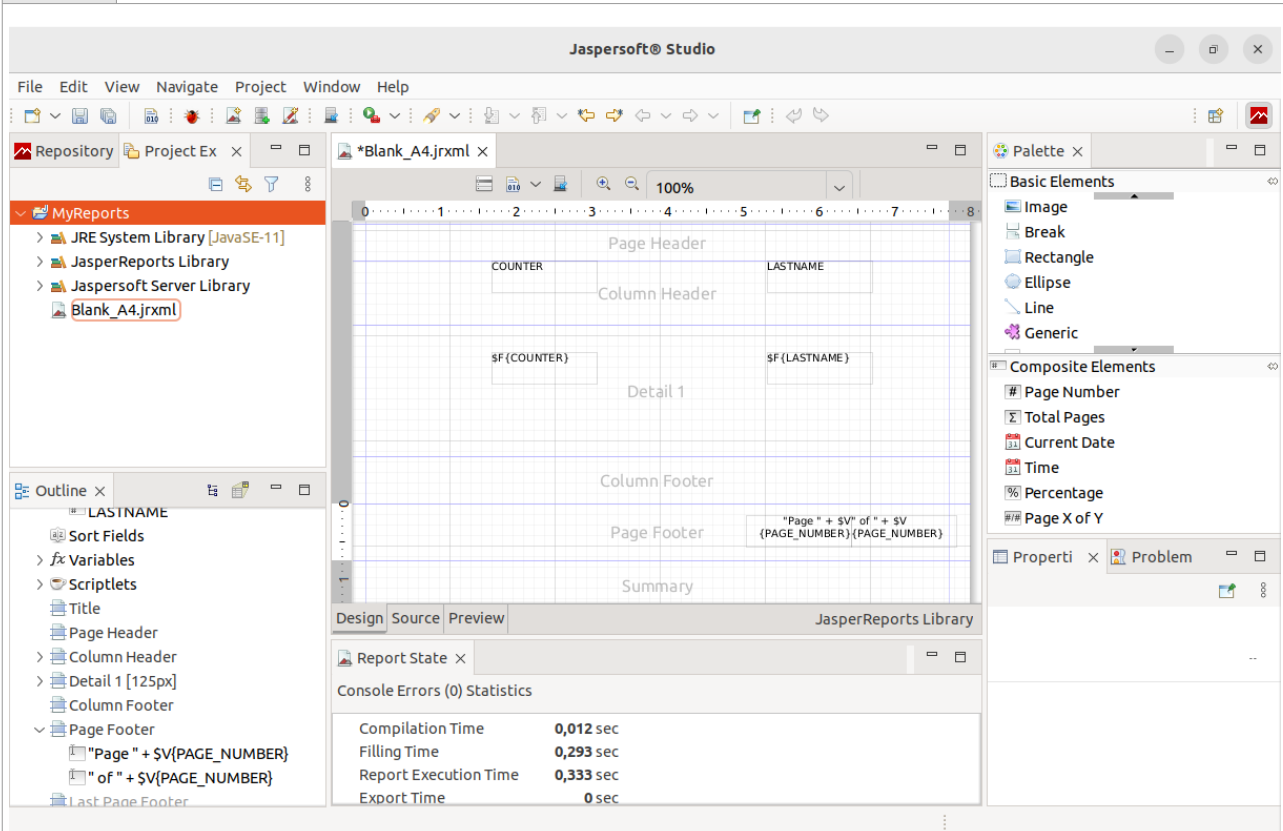


Figura 20: Generación de pie de página

Fermigo (Dominio público)

Esta herramienta crea dos campos de texto que muestran la misma variable: **PAGE_NUMBER**. El campo de texto primero muestra la página actual, el segundo el total de páginas del informe. Esto es posible porque el tiempo de evaluación de cada campo de texto es diferente, en particular, el primer campo de texto tiene el tiempo de evaluación establecido a Now por lo que **PAGE_NUMBER** contiene el valor de la página actual, el segundo lo tiene establecido a Report (en este momento de evaluación, JasperReports ha llegado al final del informe, por lo que **PAGE_NUMBER** contiene el número de la última página).

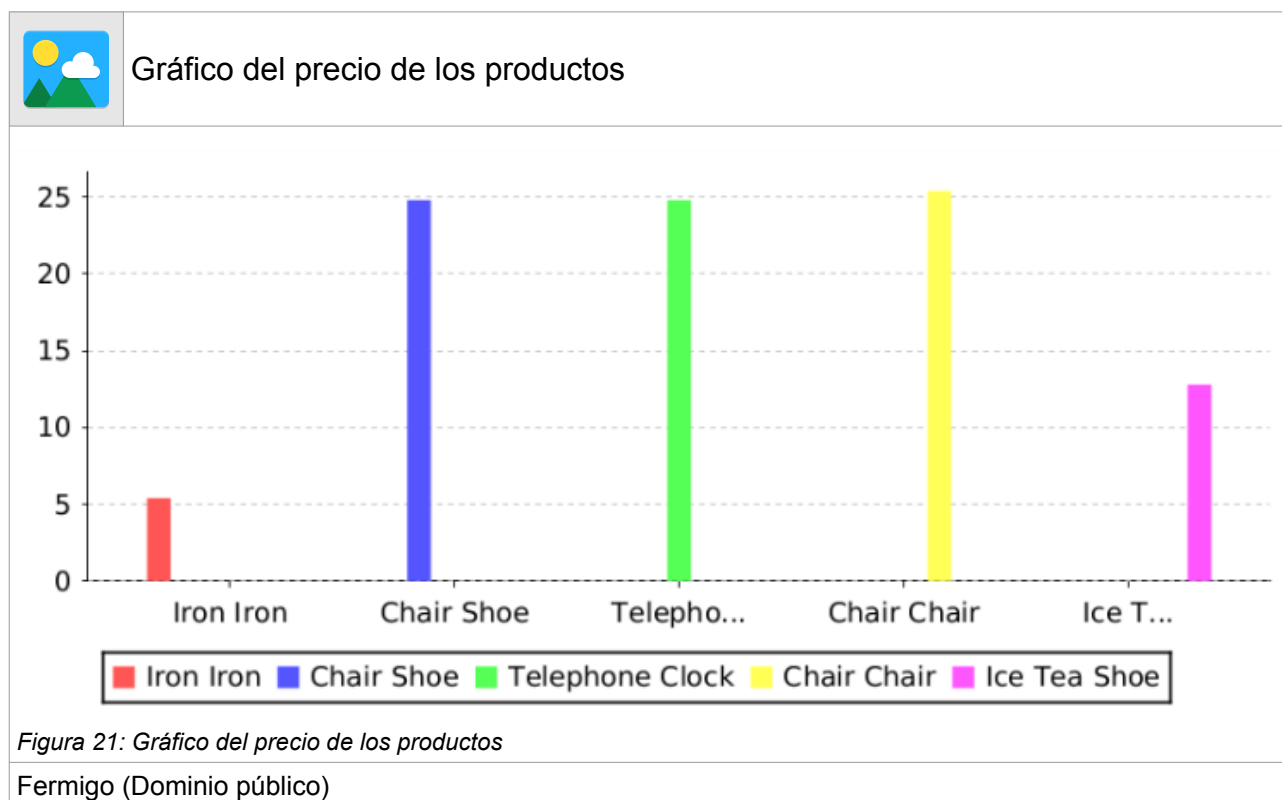
El **tiempo de evaluación** de un campo de texto es muy importante porque nos permite imprimir el valor asumido por una variable en diferentes momentos. Con esta idea, podemos poner la suma total de pedidos, como se ve en el ejemplo anterior, y obtener el valor correcto estableciendo el tiempo de evaluación de ese campo de texto a Report (esto se hace automáticamente por iReport cuando un campo se arrastra a una banda y el usuario elige para mostrar el resultado de una función de agregación).

De igual manera podemos añadir etiquetas, imágenes a la banda PAGE_HEADER para modificar el encabezado de las páginas del informe, por ejemplo, podemos añadir una etiqueta en la que escribamos el nombre de la empresa y arrastrar la fecha (Current date)

del apartado Tools de la paleta. Cuando colocamos el campo de fecha nos preguntará el formato en el que queremos que aparezca.

5.5.3 Gráficos

Un **gráfico** permite representar cierta información de tipo numérico (normalmente) mediante recursos gráficos (líneas, vectores, superficies o símbolos) con el objetivo de hacer más visibles los datos, poner de manifiesto su evolución temporal o espacial, o evidenciar relaciones elementos del sistema.



Existen distintos **tipos de gráficos**, entre los que destacan:

- **Gráficos lineales:** se representan los valores en dos ejes cartesianos ortogonales entre sí. Las gráficas lineales se recomiendan para representar series en el tiempo, y es donde se muestran valores máximos y mínimos; también se utilizan para varias muestras en un diagrama.
- **Gráficos de barras:** contienen barras verticales que representan valores numéricos. Normalmente representan frecuencias dentro de una categoría.
- **Gráficos circulares:** permite ver la distribución interna de los datos que representan un hecho, en forma de porcentajes sobre un total. Se suele separar el sector correspondiente al mayor o menor valor, según lo que se desee destacar.
- **Gráfico simbólico:** con imágenes que sirven para representar el comportamiento o la distribución de los datos cuantitativos de una población, utilizando símbolos de tamaño proporcional al dato representado.

Cuando trabajamos con gráficos, conviene aclarar los conceptos de serie y categoría:

- **Serie:** Es el conjunto de datos numéricos a representar. Podemos tener más de una serie en un gráfico, salvo en los gráficos circulares que sólo tienen una serie. A cada serie se le asigna un color diferente o algún otro identificativo que aparece claramente señalado en la leyenda del gráfico. Cada dato de la serie toma valores en un rango, por lo que si queremos tener más de una serie en el gráfico es conveniente que todas estén dentro del mismo rango.
- **Categoría:** Se corresponde con los datos a representar dentro del eje horizontal de gráfico. Cada dato de la serie toma valores para un dato de la categoría.

Los gráficos se pueden insertar en cualquier banda de nuestro informe, pero lo más lógico y habitual es insertarlo en la banda de resumen (Summary)

En la imagen anterior puede verse un ejemplo, vemos un gráfico con una serie, que representa el precio de los diferentes productos de forma gráfica.

En el siguiente apartado, vamos a ver un ejemplo de cómo construir un gráfico con Jaspersoft Studio.

5.5.4 Informes sobre consultas complejas

A veces es preciso involucrar varias tablas en la consulta para generar un informe, incluyendo uniones, ordenación y agrupaciones. El resultado final es independiente de lo compleja que pueda ser la consulta, ya que una vez que hayamos obtenido los registros, pero para facilitar el proceso de crear la consulta podemos utilizar alguna herramienta visual. Jaspersoft Studio, por ejemplo, proporciona un diseñador de consultas (query designer), accesible desde la ventana de la consulta que puede acceder a la estructura completa de la base de datos para hacer la selección de tablas y campos de manera gráfica.

- En la zona de la izquierda abajo aparece la lista de las tablas de la base de datos seleccionando las opciones **Public** y **Tables**. Podemos desplegar una tabla en el panel de la derecha haciendo doble clic sobre su nombre. Los campos que se marquen se añadirán a la consulta.
- Haciendo clic con el botón secundario sobre la cláusula **WHERE** y seleccionando **add condition** podemos añadir condiciones.
- De igual forma se añaden cláusulas **HAVING**.
- Para añadir cláusulas **GROUP BY** u **ORDER BY** hacemos clic con el botón secundario sobre el campo que va a participar en la cláusula y seleccionamos **add to group by** o **add to order by**.
- Cualquier cláusula se elimina haciendo clic con el botón secundario, seleccionando la opción **remove**.

5.6 Repaso a la librería Jasper Report

La librería **JasperReports** permite la integración de los informes en una aplicación Java. Es posible hacerlo partiendo tanto del archivo .jasper como del archivo .jrxml. Se utiliza para compilar, rellenar, aplicar parámetros y visualizar un informe en diferentes formatos finales. La ventaja es que como podemos pasar parámetros mediante código, basta con conectar el informe a un formulario y el usuario final podrá determinar las características finales del informe o de los datos a mostrar.

Para **pasar parámetros** a un informe es necesario, en primer lugar, tener definido el parámetro en el informe con el tipo adecuado. En el código crearemos una **tabla hash** a la que añadiremos una pareja formada por el nombre del parámetro y la variable que contiene su valor. Una Hashtable Java es una estructura de datos que utiliza una función hash para identificar datos mediante una llave o clave

El despliegue del informe se hace a través de un objeto de la clase JasperPrint y la clase JasperFillManager. Para crearlo usaremos esta sentencia:

```
JasperPrint print = JasperFillManager.fillReport(ArchivoJasper, Parámetros, Conexion);
```

Donde:

- **ArchivoJasper:** es el archivo de Jasper con el informe.
- **Parámetros:** tabla hash con los parámetros que hay que pasar al informe.
- **Conexión:** conexión al origen de datos.

JasperFillManager.fillReport genera el informe en memoria. Para volcar el informe a un archivo, utilizamos la clase **JasperExportManager**, que tiene varios métodos para crear archivos de salida de diferentes tipos, entre otros podremos generar un archivo PDF con el informe con el siguiente código:

```
JasperExportManager.exportReportToPdfFile(print, "informe.pdf");
```

5.6.1 Creación de informe desde una aplicación

Antes de comenzar es necesario tener instalado y configurado NetBeans para poder usar el complemento de JasperReport, y un servidor de base de datos con el repositorio de datos que se empleará

En el proyecto se necesitan las siguientes librerías:

- Driver MariaDB JDBC
- JasperReport 6.30.x, así como sus dependencias.

Lo que pretendemos con este ejemplo, es desde una aplicación Java, construir un informe en formato PDF partiendo de un archivo. Es decir, al ejecutar la aplicación vamos a crear un nuevo fichero, en formato PDF, que contendrá el informe.

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
 */
package demo;

import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Scanner;
import java.util.logging.Level;
import java.util.logging.Logger;
import net.sf.jasperreports.engine.JRException;
import net.sf.jasperreports.engine.JasperCompileManager;
import net.sf.jasperreports.engine.JasperExportManager;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperReport;
import net.sf.jasperreports.view.JasperViewer;

/**
 *
 * @author fermigo
 */
public class Demo {

    static Connection conn = null;
    static String currentDirectory = System.getProperty("user.dir");

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Server IP: ");
        String server = sc.next();
        System.out.print("Server Port: ");
        String port = sc.next();
        System.out.print("Database: ");
        String database = sc.next();
        System.out.print("Username: ");
        String username = sc.next();
        System.out.print("Password: ");
        String password = sc.next();

        try {
            conn = DriverManager.getConnection("jdbc:mariadb://" + server + ":" + port + "/" + database, username, password);
            conn.setAutoCommit(false);
            Files.createDirectories(Paths.get(currentDirectory + "/Reports"));
            JasperReport report = JasperCompileManager.compileReport(Demo.class.getClassLoader().getResourceAsStream("r.jrxml"));
            JasperPrint print = JasperFillManager.fillReport(report, null, conn);
            JasperExportManager.exportReportToPdfFile(print, currentDirectory + "/Reports/Report_5_5.pdf");
            JasperViewer.viewReport(print, false);
        } catch (SQLException e) {
            System.out.println("Connection error: " + e.getMessage());
        } catch (IOException | JRException ex) {
            Logger.getLogger(Demo.class.getName()).log(Level.SEVERE, null, ex);
        } finally {
            try {
                if (conn != null && !conn.isClosed()) {
                    conn.close();
                }
            } catch (SQLException ex) {
                Logger.getLogger(Demo.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
    }
}
```

5.7 Análisis del código obtenido

Los informes de iReport se almacenan en archivos XML con extensión .jrxml. Un archivo jrxml se compone de un conjunto de secciones, algunas de ellas relativas a las características físicas del informe, como las dimensiones de la página, el posicionamiento de los campos, y la altura de las bandas, otras son relativas a las características lógicas, tales como la declaración de los parámetros y variables, y la definición de una consulta para la selección de datos.

La primera operación consiste en compilar el archivo fuente jrxml en un archivo de Jasper. Esta compilación se realiza por motivos de rendimiento. Si la compilación se ejecuta correctamente, el archivo producido Jasper se carga y se llena con la conexión activa o fuente de datos.

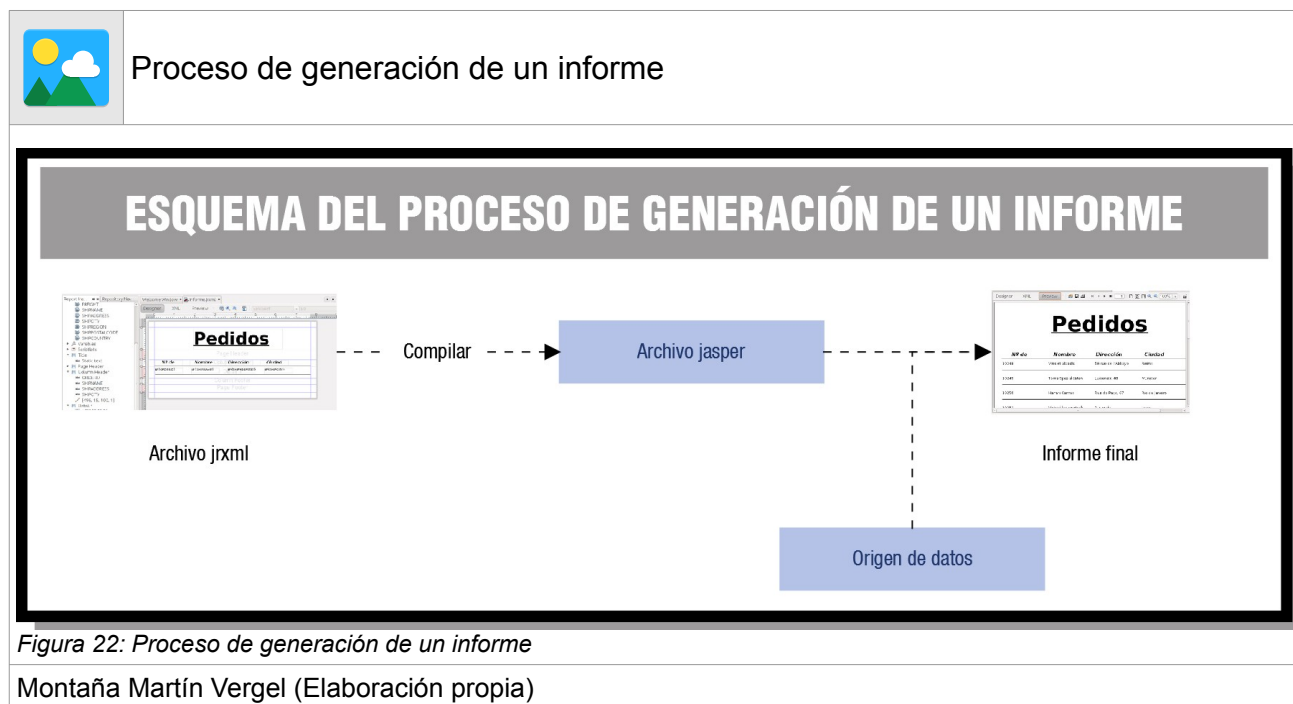


Figura 22: Proceso de generación de un informe

Montaña Martín Vergel (Elaboración propia)

Durante la compilación del archivo jrxml, el archivo XML se analiza y se carga en un objeto JasperDesign, una estructura de datos que permite representar el contenido XML en la memoria. Independientemente del lenguaje que se utiliza para las expresiones dentro del archivo jrxml, JasperReports crea una clase especial de Java que representa la totalidad del informe, se compila, se instancia y se serializa en un archivo Jasper, que será el que se cargue en una aplicación posteriormente.

A la hora de ejecutar un informe necesitaremos este archivo de Jasper y una fuente de datos para JasperReports. Hay muchos tipos de fuentes de datos, es posible llenar un archivo de Jasper de una consulta SQL, un archivo XML, un archivo CSV, una colección de JavaBeans, etc., incluso es posible elaborar una fuente de datos personalizada. Con un archivo de Jasper y un origen de datos, JasperReports es capaz de generar el documento final en el formato que prefiera.

El archivo Jasper no contiene recursos externos, como puedan ser las imágenes utilizadas en el informe, paquetes de recursos para ejecutar el informe en diferentes idiomas, scriptlets extra u hojas de estilos externas. Todos estos recursos deben estar ubicados en tiempo de ejecución y ser proporcionados por la aplicación.