

Manual para Generar Reportes en JasperReports desde Netbeans.

Este manual cubre el proceso desde la creación de un reporte sencillo hasta la exportación y visualización del reporte en formato PDF. Además, se indican los puntos críticos a tener en cuenta para evitar los errores comunes.

1. Crear el Reporte en Jaspersoft Studio

1.0. Descargar Jaspersoft Studio

Descarga Jaspersoft Studio desde su web, en su versión community
<https://www.jaspersoft.com/products/jaspersoft-community>

1.1 Crear un nuevo reporte

1. Abre **Jaspersoft Studio** y crea un **Nuevo Proyecto de Reporte**.
2. En el asistente de creación de reporte, selecciona un **Reporte en Blanco**.
3. Define el nombre del archivo `.jrxml`, por ejemplo, `EjemploReporte.jrxml`.

1.2 Definir el Parámetro en el Reporte

Para que el reporte reciba datos desde tu aplicación Java, es necesario crear un **parámetro** en lugar de un campo normal.

1. En el panel de **Outline**, haz clic derecho en **Parameters** y selecciona **Create Parameter**.
2. Nombra el parámetro como `UserInput`.
3. Establece su tipo de datos como `String` (esto es importante para que coincida con el tipo de datos que pasas desde Java).

1.3 Diseñar el Reporte

- Arrastra un **textField** desde la paleta de elementos en el área de diseño y colócalo en la sección **detail**.
- En el campo de expresión del **textField**, usa la expresión `[$P{UserInput}]`, ya que estamos usando un **parámetro** y no un campo (también puedes arrastrar el parámetro directamente).
- Asegúrate de que el parámetro `UserInput` esté bien vinculado en el reporte.

1.4. Exportar el reporte.

- Guarda el reporte en formato `jrxml`.
 - Crea en tu proyecto de Java con Maven una carpeta `resources` en `Proyecto/src/main/resources` y copia ahí el reporte `jrxml`. Si no existe esta carpeta, creala.
-

2. Integración con Java

2.1 Configurar el Proyecto Java

Asegúrate de tener las bibliotecas de **JasperReports** y las librerías necesarias en tu archivo `pom.xml` (para proyectos Maven). Para ello, añade las siguientes dependencias:

```
<dependency>

    <groupId>net.sf.jasperreports</groupId>

    <artifactId>jasperreports</artifactId>

    <version>7.0.1</version> <!-- La versión debe coincidir con
Studio -->

</dependency>
```

Si deseas usar otras exportaciones, como por ejemplo exportación a Excel, también añade las dependencias correspondientes de JasperReports Excel (puedes buscar otras opciones de exportación que ofrece JasperReports):

```
<dependency>

    <groupId>net.sf.jasperreports</groupId>

    <artifactId>jasperreports-extensions</artifactId>

    <version>7.0.1</version>

</dependency>
```

Para que Maven descargue las dependencias, haz clic en el botón **Clean and Build** de Netbeans.

2.2 Código Java para Cargar y Rellenar el Reporte y guardarlo en PDF

El siguiente código (dentro de un evento de un botón que será quien active todo) en Java obtiene el texto de un **JTextField**, lo pasa como parámetro al reporte, y luego genera el PDF:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
    // Obtener el texto del JTextField

    String texto = jTextField1.getText();

    try {

        // Cargar el diseño del reporte

        InputStream reportStream =
getClass().getResourceAsStream("/EjemploReporte.jrxml");

        JasperReport jasperReport =
JasperCompileManager.compileReport(reportStream);

        // Prepara parámetros para enviar al reporte

        Map<String, Object> parameters = new HashMap<>();

        parameters.put("UserInput", texto);

        // Generar el reporte con los parámetros

        JasperPrint jasperPrint =
JasperFillManager.fillReport(jasperReport, parameters, new
JREmptyDataSource());

        // Exportar el reporte a un archivo PDF
```

```
String outputPdfPath = "reporte.pdf";

JasperExportManager.exportReportToPdfFile(jasperPrint,
outputPdfPath);

// Mostrar el reporte en una ventana

JasperViewer.viewReport(jasperPrint, false);

// Confirmación de éxito

JOptionPane.showMessageDialog(null, "El reporte ha sido
generado correctamente como PDF.", "Éxito",
JOptionPane.INFORMATION_MESSAGE);

    } catch (JRException ex) {

        ex.printStackTrace();

        JOptionPane.showMessageDialog(null, "Error al generar el
reporte: " + ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);

    }

}
```

2.3 Asegurarse de la Correcta Configuración de Parámetros

- El parámetro debe ser **UserInput** (con el mismo nombre en Java y en el reporte).
- Si el parámetro no se pasa correctamente o si se usa un **campo** en vez de un **parámetro**, el valor no se mostrará en el reporte y aparecerá como **null**.