

INTRODUCCION

En el desarrollo de software, guardar datos es solo una parte del trabajo. La otra parte consiste en mostrarlos de forma útil para la toma de decisiones. En módulos anteriores extraíamos información mediante consultas SQL en consola o tablas simples en el entorno de desarrollo. El objetivo de esta práctica es implementar un sistema de reportes real.

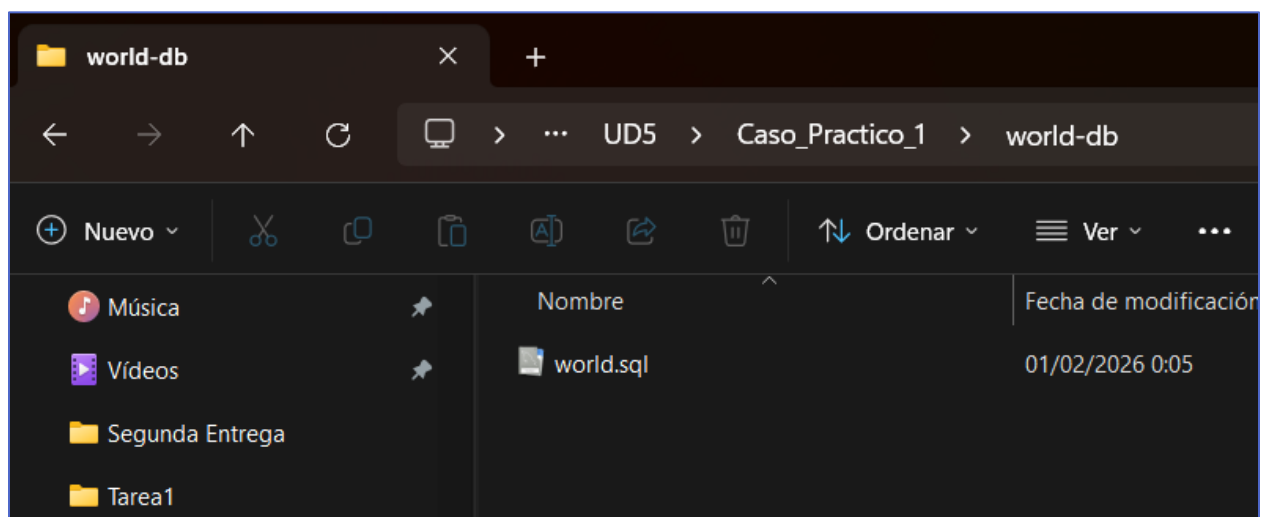
Utilizamos JasperSoft Studio para esta prueba de concepto. Lo elegimos por ser un estándar de código abierto en Java para generar informes. Se integra bien con bases de datos relacionales. Usamos la base de datos “world-db” de MySQL como terreno de pruebas.

Este documento detalla el despliegue paso a paso. Abarca desde la importación de la base de datos hasta la configuración SQL necesaria para un informe limpio.

0 1

FASE 1: IMPORTACIÓN Y PREPARACIÓN DE LA BASE DE DATOS (MYSQL)

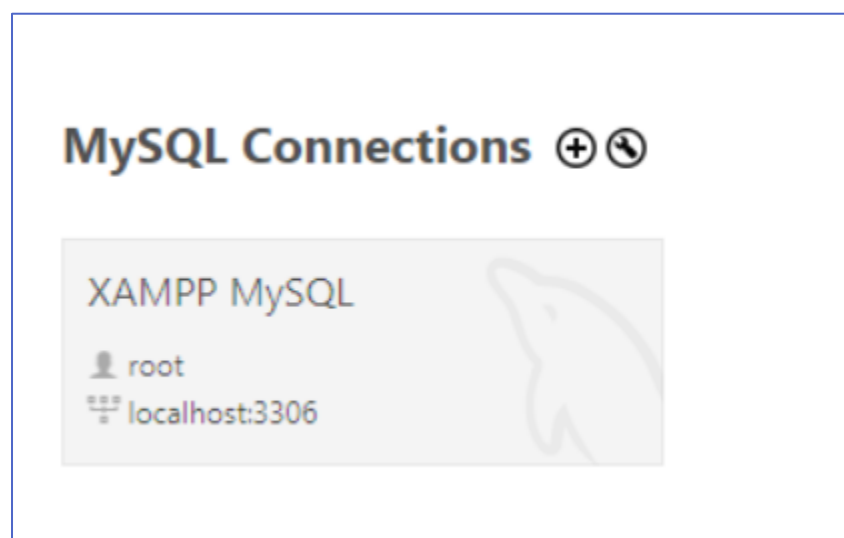
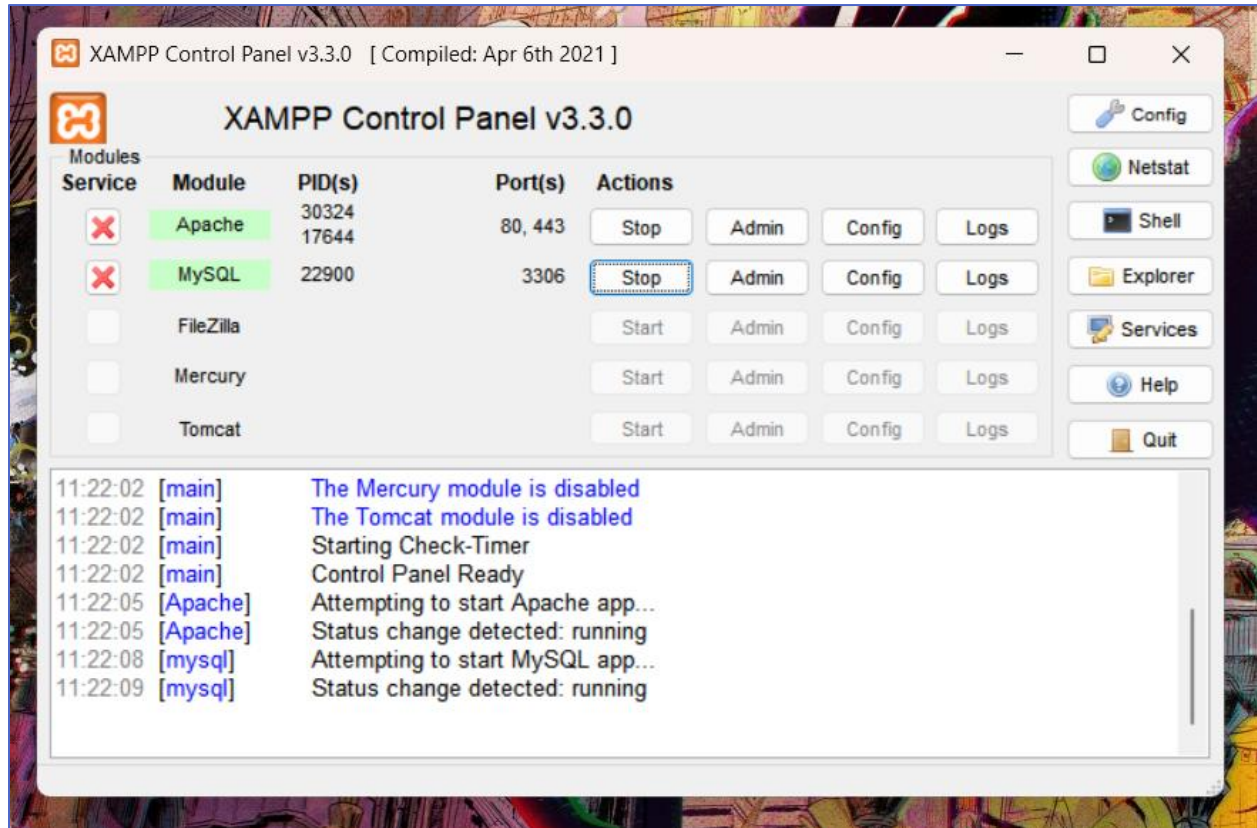
El primer paso lógico era obtener datos de trabajo. Descargamos el archivo **world-db.zip** del repositorio de **MySQL**. Este archivo contiene el **script world.sql**. El archivo incluye todo lo necesario: el DDL para crear las tablas y el DML con las instrucciones para poblarlas.



Es posible hacer esto por línea de comandos. Decidí usar **MySQL Workbench** para controlar el proceso visualmente.

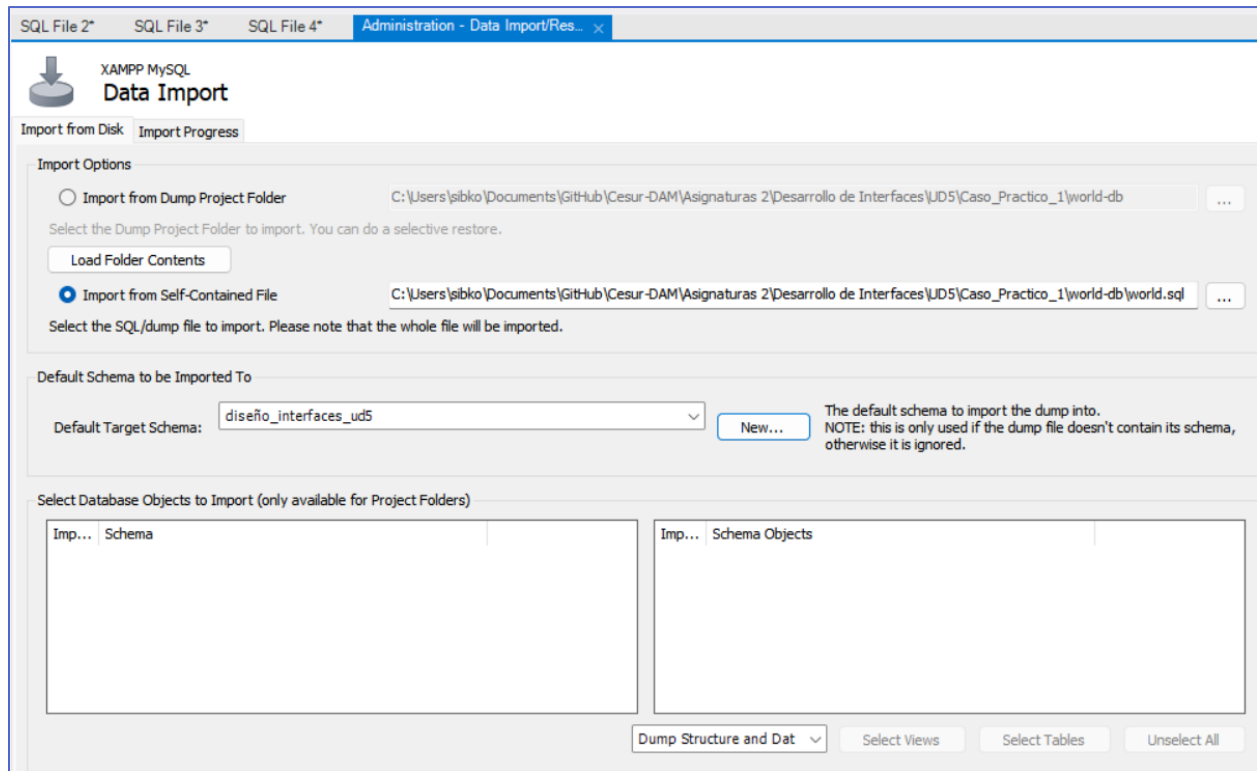
Los pasos técnicos fueron:

Abrí la conexión local en Workbench con el usuario root para tener permisos de creación de esquemas.



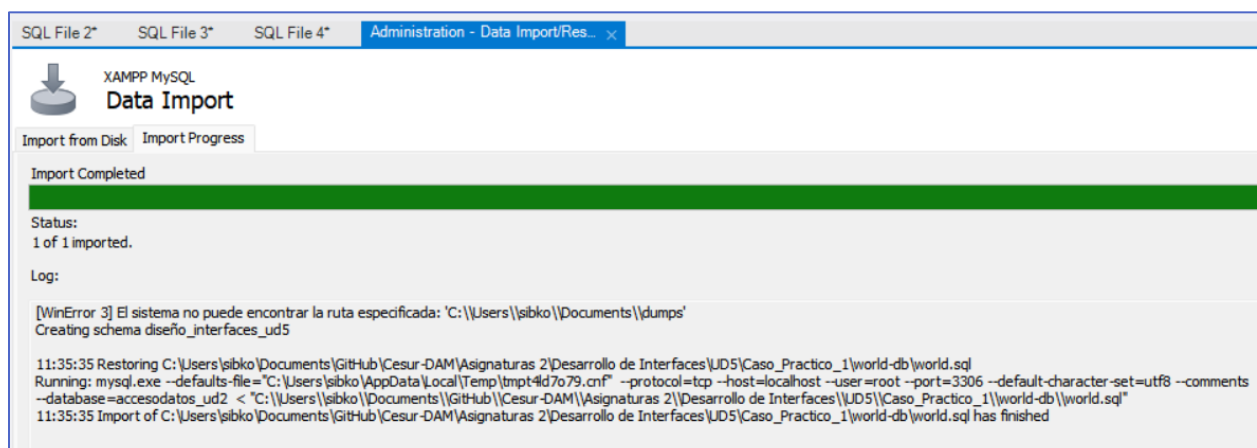
Fui a **Server > Data Import**. Abrir el archivo .sql directamente en el editor puede bloquear el programa si el archivo es grande.

Seleccioné **Import from Self-Contained File** e indiqué la ruta del script.



No tuve que crear el esquema manualmente. El script ya incluía la cláusula **CREATE DATABASE IF NOT EXISTS world;**

Inicié la importación.



Esto comprobó dos puntos vitales **antes de usar JasperSoft**:

- Primero, las tres tablas principales (**city**, **country**, **countrylanguage**) tenían datos.
- Segundo, la **codificación de caracteres era correcta** y los nombres de países se veían bien.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'world' selected. The main window shows a query in the 'SQL File 2*' tab: `SELECT * FROM country LIMIT 5;`. Below the query, the 'Result Grid' displays a table of country data. The 'Output' pane at the bottom shows the execution log with several error messages.

Code	Name	Continent	Region	SurfaceArea	IndepYear	Population	LifeExpectancy	GNP	GNPOld	LocalName
ABW	Aruba	North America	Caribbean	193.00	NA	103000	78.4	828.00	793.00	Aruba
AFG	Afghanistan	Asia	Southern and Central Asia	652090.00	1919	22720000	45.9	5976.00	NA	Afghanistan/Afqanestan
AGO	Angola	Africa	Central Africa	1246700.00	1975	12878000	38.3	6648.00	7984.00	Angola
AIA	Anguilla	North America	Caribbean	96.00	NA	8000	76.1	63.20	NA	Anguilla
ALB	Albania	Europe	Southern Europe	28748.00	1912	3401200	71.6	3205.00	2500.00	Shqipëria

country 1 x

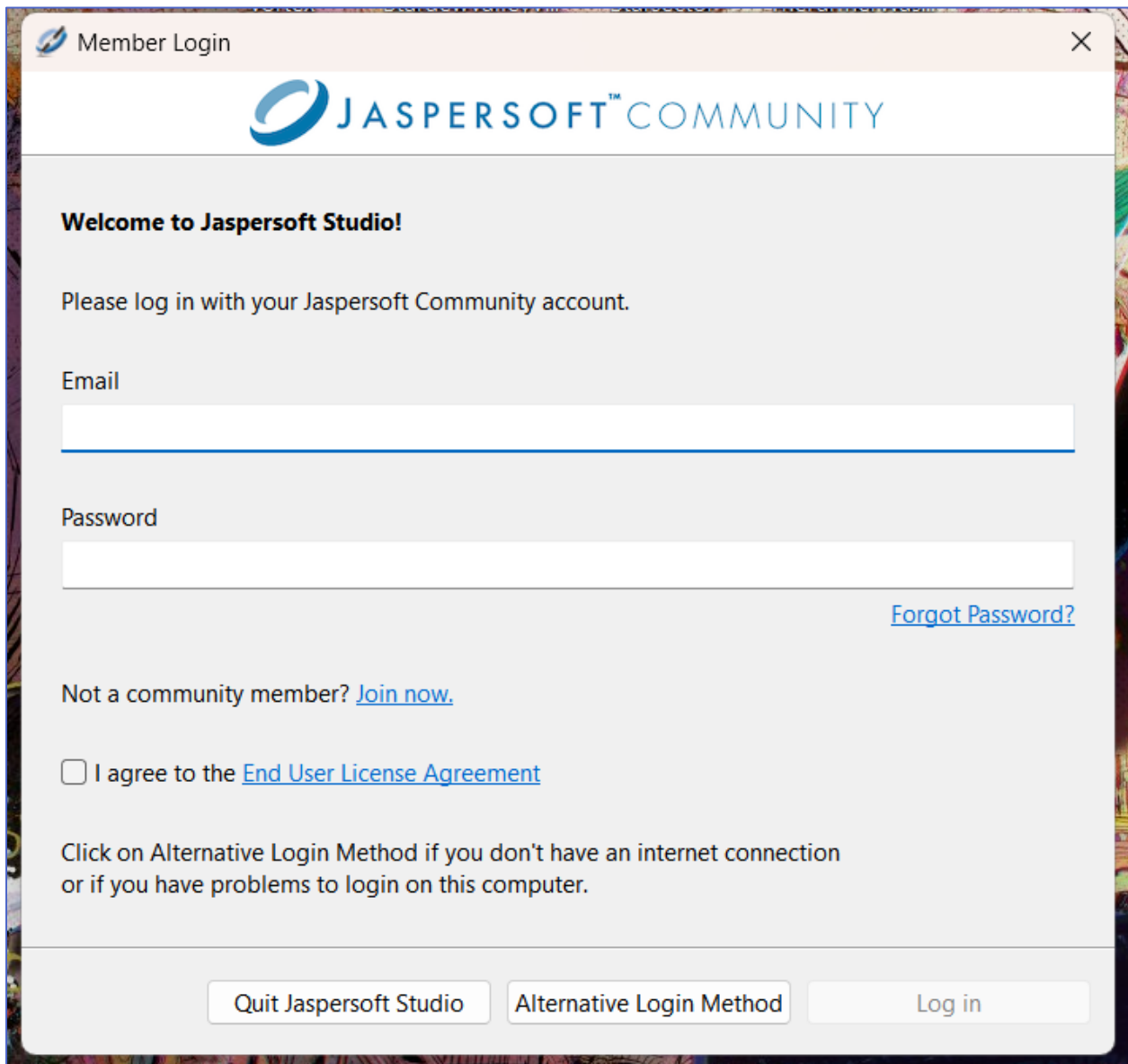
Output

#	Time	Action	Message
1	18:33:27	SELECT * FROM diseño_interfases_ud5 LIMIT 0, 1000	Error Code: 1146. Table 'diseño_interfases_ud5.diseño_interfases_ud5' doesn't exist
2	18:33:44	SELECT tables LIMIT 0, 1000	Error Code: 1054. Unknown column 'tables' in 'field list'
3	18:33:47	SELECT tables LIMIT 0, 1000	Error Code: 1054. Unknown column 'tables' in 'field list'
4	18:33:58	show tables	0 row(s) returned
5	18:34:10	SHOW TABLES	0 row(s) returned
6	18:34:36	USE world	0 row(s) affected
7	18:34:56	show tables	3 row(s) returned
8	18:35:31	SELECT * FROM country LIMIT 5	5 row(s) returned

0 2

FASE 2: CONFIGURACIÓN DEL ENTORNO Y CONEXIÓN (JASPERSOFT)

Los datos estaban listos en **MySQL**. Pasamos a instalar y configurar **TIBCO Jaspersoft Studio**.



The image shows a 'Member Login' dialog box for Jaspersoft Studio. The dialog has a title bar with a close button. Below the title bar is the Jaspersoft Community logo. The main content area includes a welcome message, a login instruction, and two input fields for 'Email' and 'Password'. There is a 'Forgot Password?' link next to the password field. Below the input fields, there is a link to 'Join now' for non-members and a checkbox for agreeing to the 'End User License Agreement'. At the bottom, there is a note about alternative login methods and three buttons: 'Quit Jaspersoft Studio', 'Alternative Login Method', and 'Log in'.

Member Login

JASPERSOFT™ COMMUNITY

Welcome to Jaspersoft Studio!

Please log in with your Jaspersoft Community account.

Email

Password

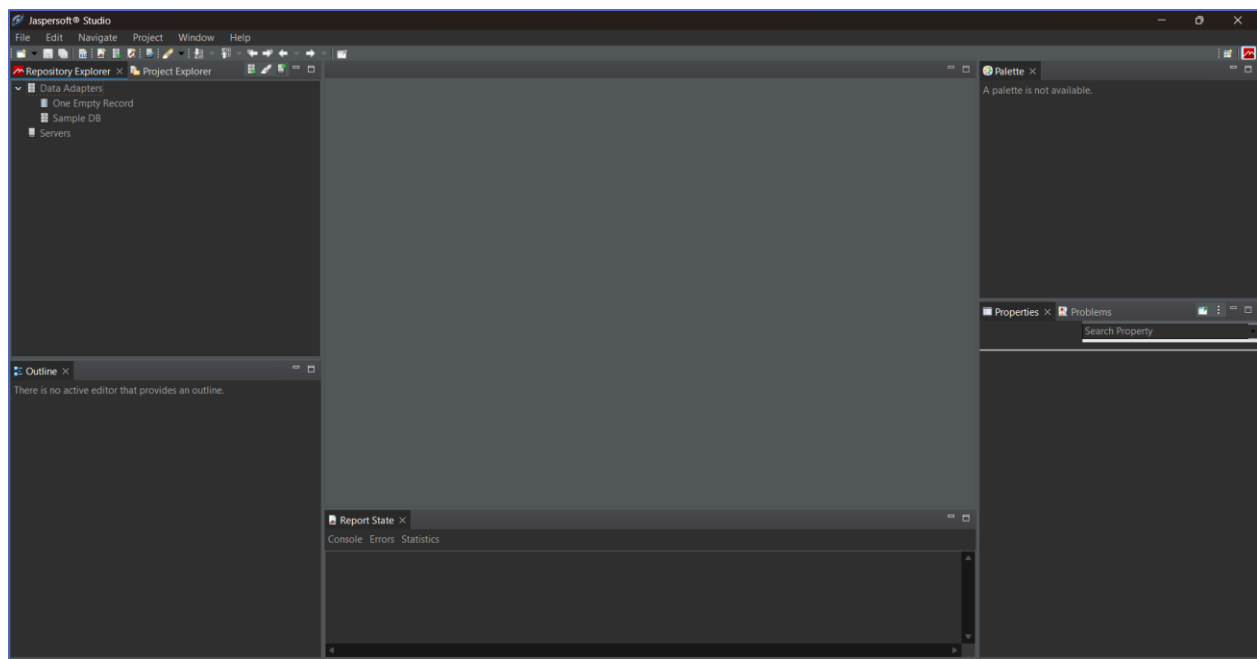
[Forgot Password?](#)

Not a community member? [Join now.](#)

☐ I agree to the [End User License Agreement](#)

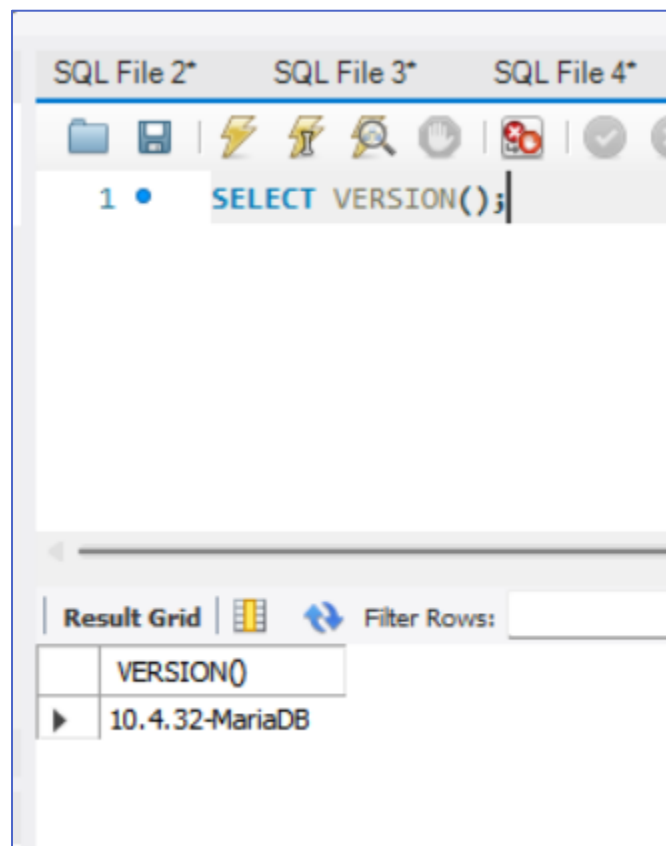
Click on Alternative Login Method if you don't have an internet connection or if you have problems to login on this computer.

Quit Jaspersoft Studio Alternative Login Method Log in



Crear un informe requiere más que seleccionar un archivo nuevo. Debemos conectar el diseñador de informes con la base de datos. Configuré un **Data Adapter** para esto:

Comprobamos la versión de la base de datos: Vemos que tenemos **MariaDB** porque **Xampp dejó de usar MySQL** pero son primos hermanos funcionan igual



Vamos a **Maven repository** y buscamos el **conector MariaDB** para esa versión:



3. MariaDB Java Client

org.mariadb.jdbc » [mariadb-java-client](https://org.mariadb.jdbc/mariadb-java-client)

JDBC driver for MariaDB and MySQL

Last Release on Dec 17, 2025



MariaDB Java Client » 3.5.7

JDBC driver for MariaDB and MySQL

Overview

Dependencies (8)

Changes (3)

Books (50)

LICENSE

LGPL 2.1

CATEGORIES

MySQL Drivers

JDBC Drivers

TAGS

database

sql

jdbc

driver

mariadb

client

rdbms

mysql

ORGANIZATION

mariadb.com

HOMEPAGE

<https://mariadb.com/kb/en/mariadb/about-mariadb-connector-j/>

[Inspect URL](#)

LINKS



DATE

Dec 16, 2025

FILES

pom (34 KB)

jar (745 KB)

[View All](#)

REPOSITORIES

Central

RANKING

#471 in MvnRepository

#3 in MySQL Drivers

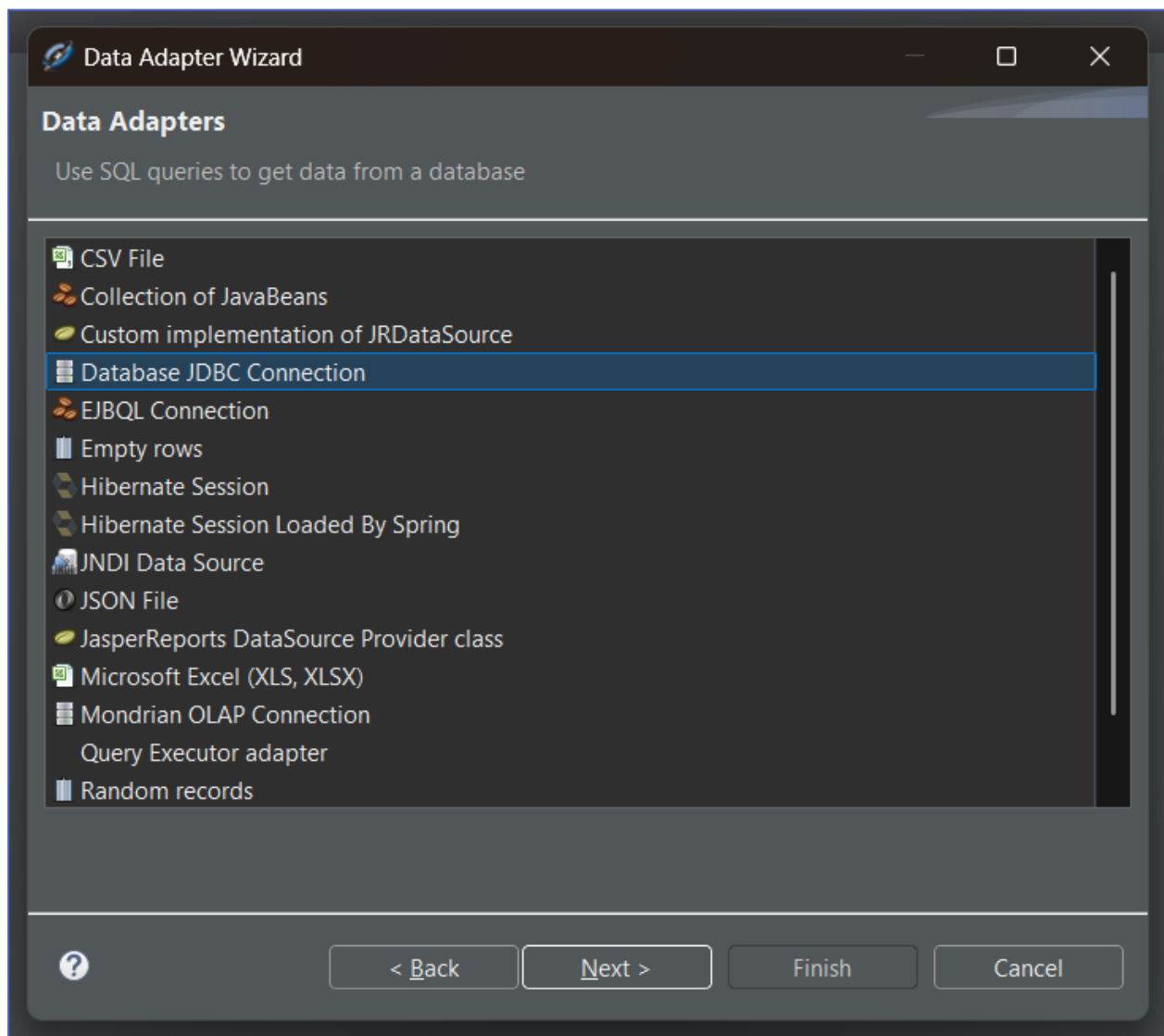
#7 in JDBC Drivers

VULNERABILITIES

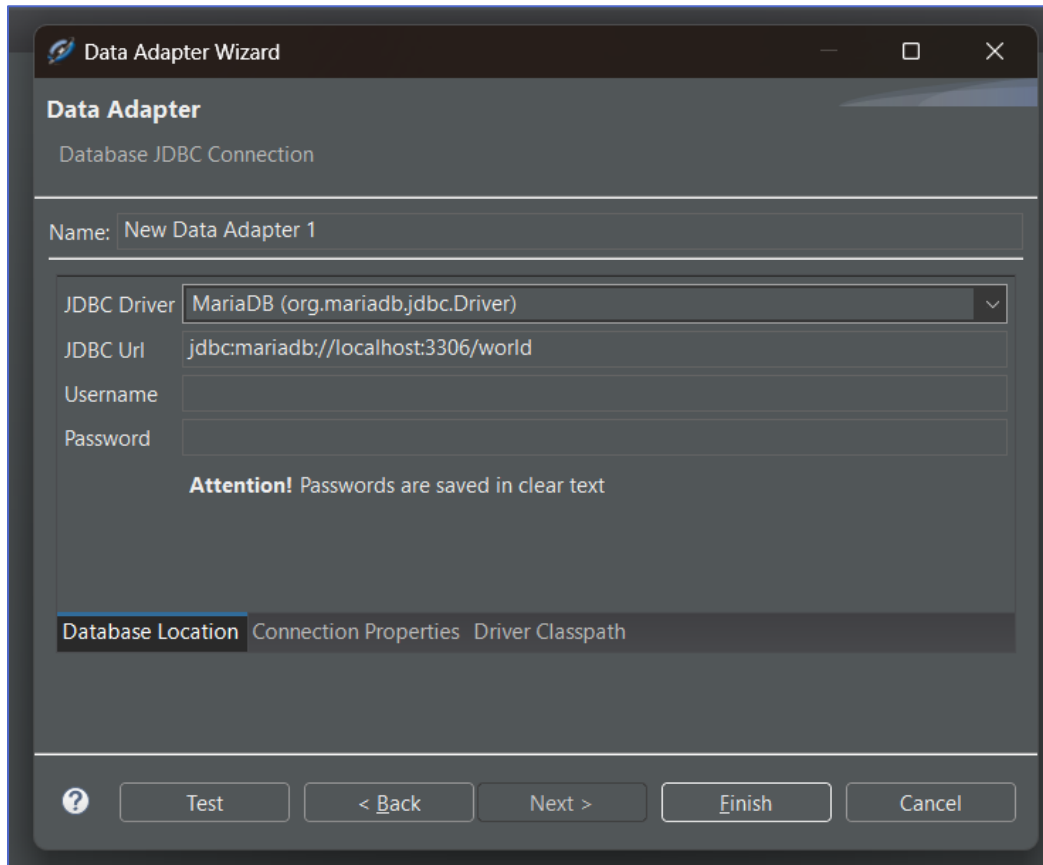
Vulnerabilities from dependencies:

[CVE-2026-24400](#)

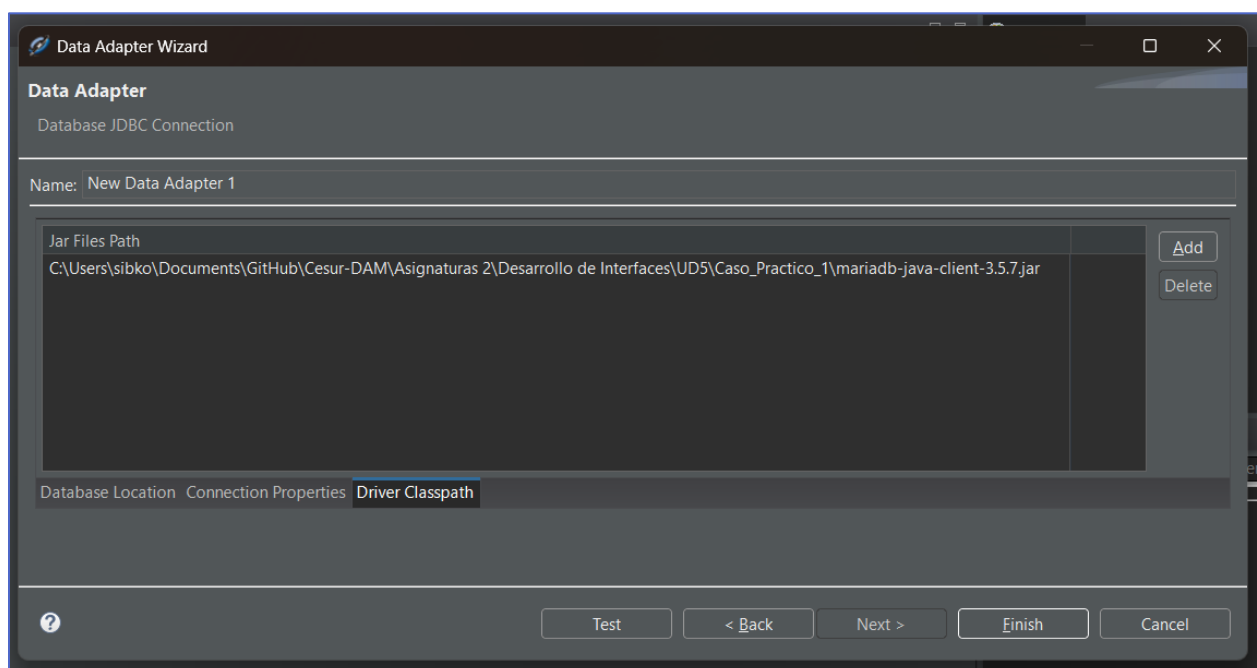
Creé un nuevo **Data Adapter** de tipo **Database JDBC Connection** en el **Repository Explorer**.



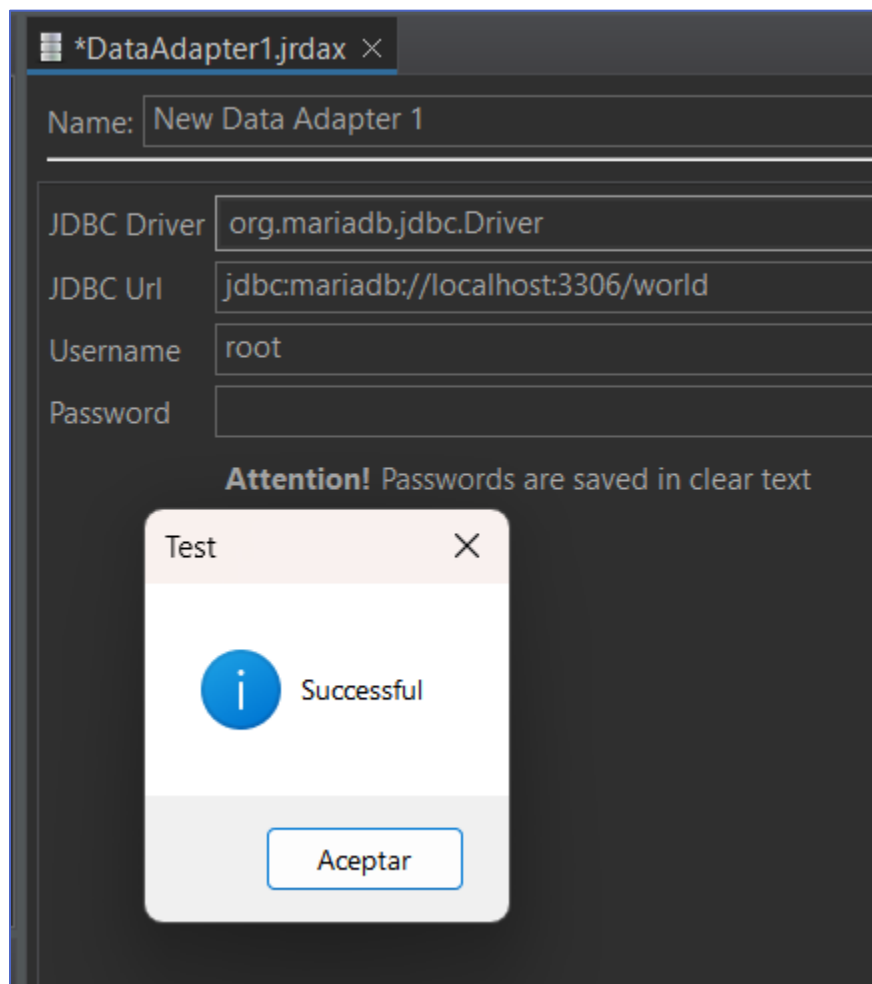
Configuré el controlador de **MariaDB** (**org.mariadb.jdbc.Driver**) y la cadena de conexión estándar: **jdbc:mariadb://localhost:3306/world**.



Añadimos el **conector mariadb-java-client-3.5.7.jar** en dependencias:



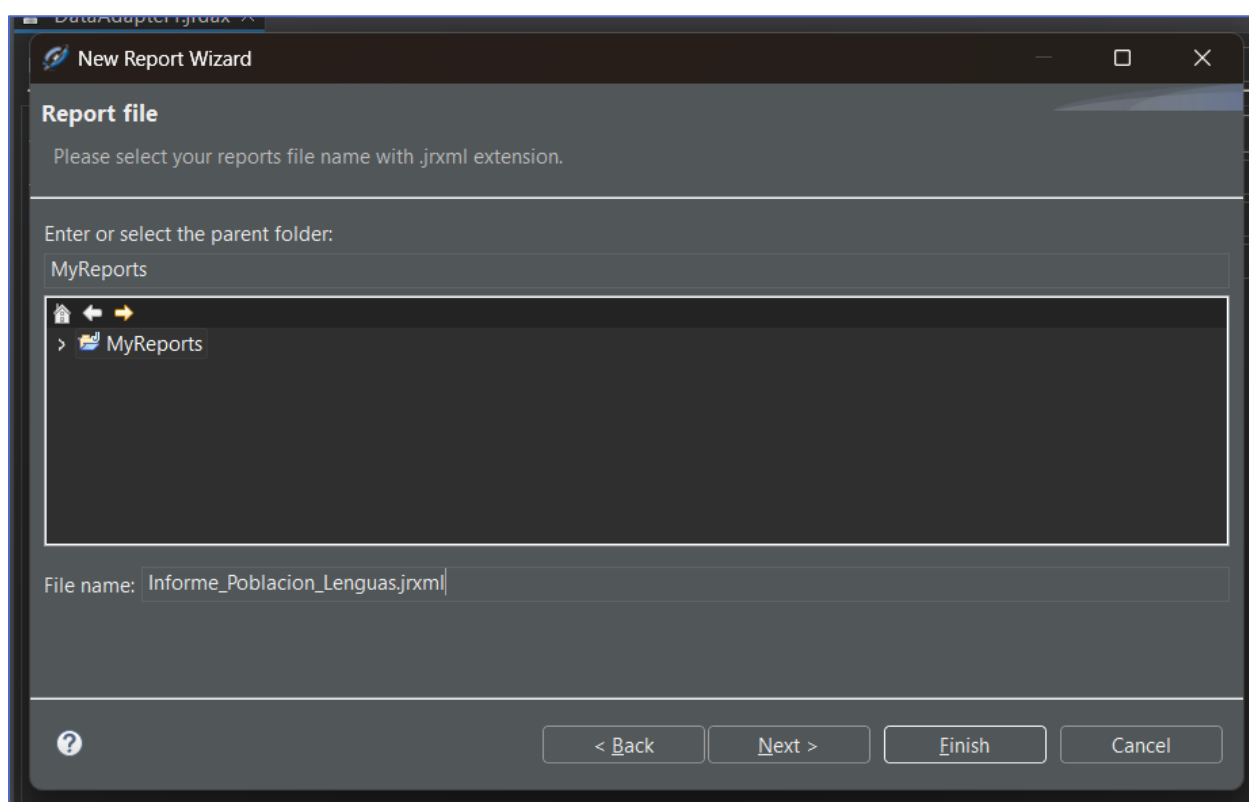
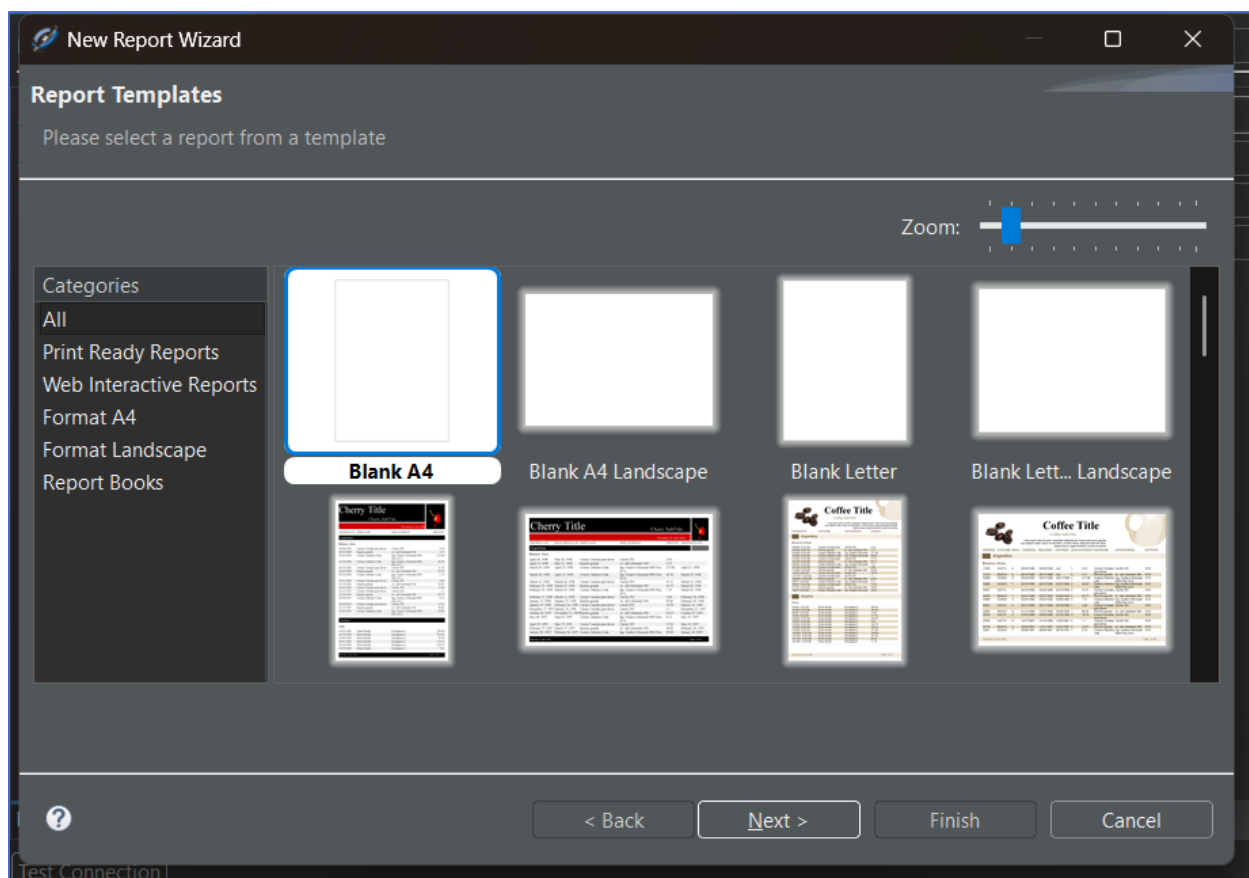
Ingresé usuario y contraseña y probé la conexión. El mensaje "**Successful**" confirmó la **comunicación** entre **JasperSoft** y **MariaDB**.



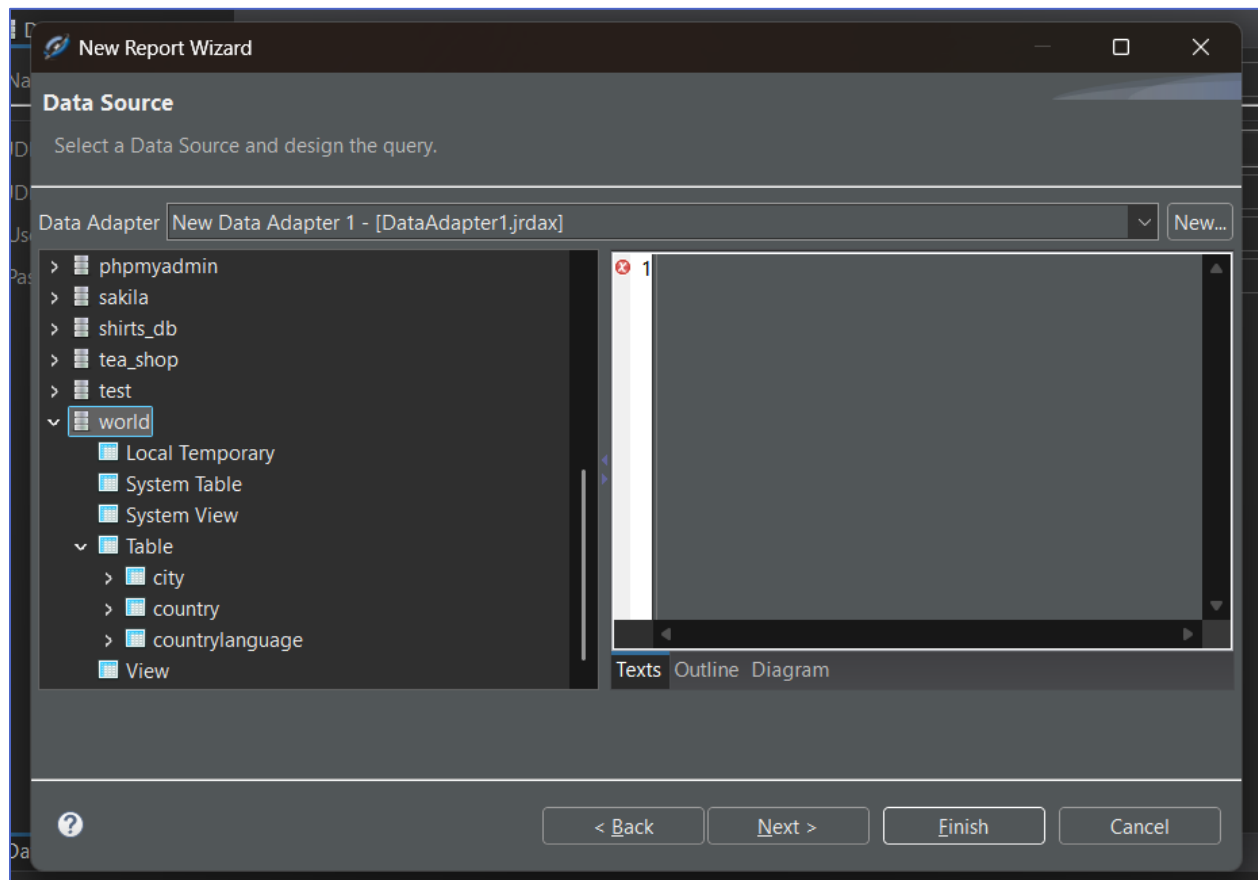
El Informe en Blanco:

Una vez conectado, creé el archivo maestro **Informe_Poblacion_Lenguas.jrxml**. Elegí la plantilla "**Blank A4**" en orientación **vertical**.

El cliente pidió solo **tres columnas: País, Población y Lenguas**. El formato vertical es suficiente y facilita la impresión.

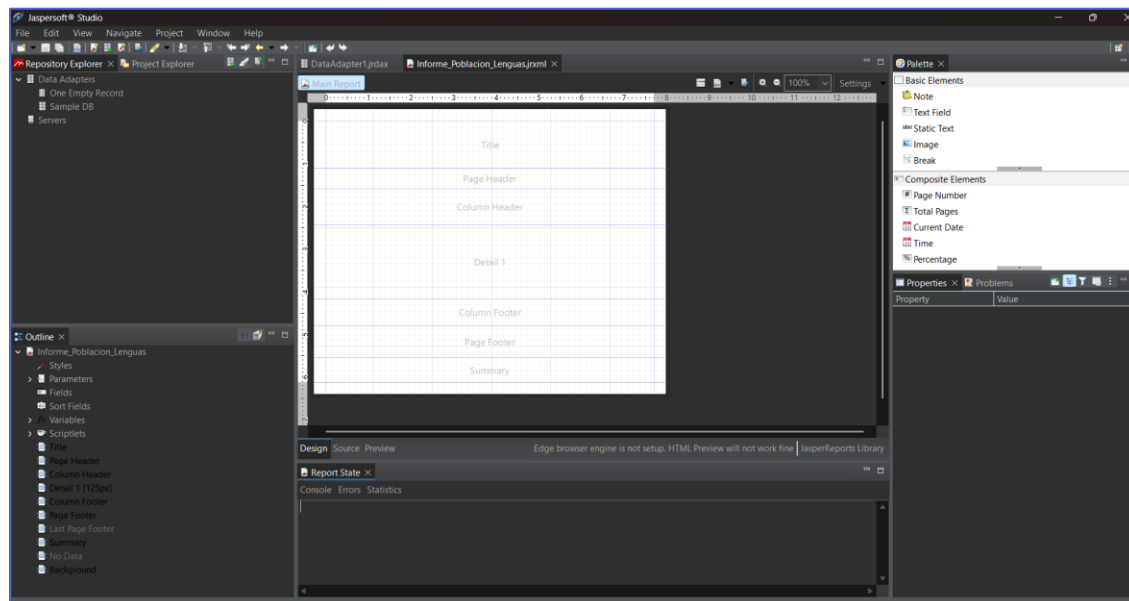


Vemos que conecta con el **data adapter** creado:



Limpieza del Lienzo:

JasperSoft crea varias bandas por defecto, como márgenes y pies de página. Estas sobran para un informe tabular básico.

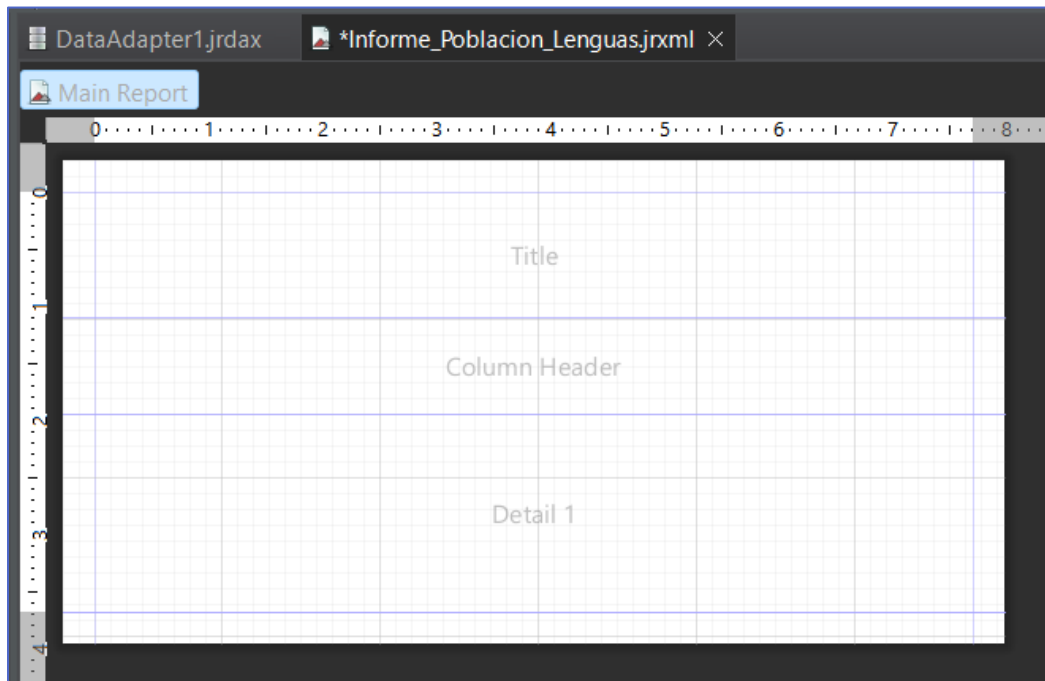
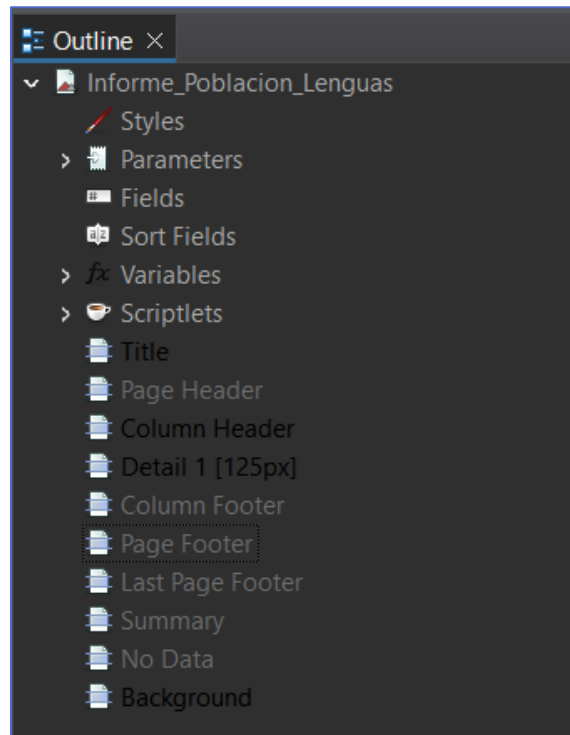


Eliminé las bandas **Page Header**, **Column Footer** y **Summary** para limpiar el área de trabajo. Conservé solo lo necesario:

Title: Para poner el título general.

Column Header: Para los encabezados de la tabla ("País", "Población", "Idiomas").

Detail 1: Esta es la banda mágica. Es la que JasperSoft. El sistema repetirá el proceso automáticamente para cada fila devuelta por nuestra consulta SQL.

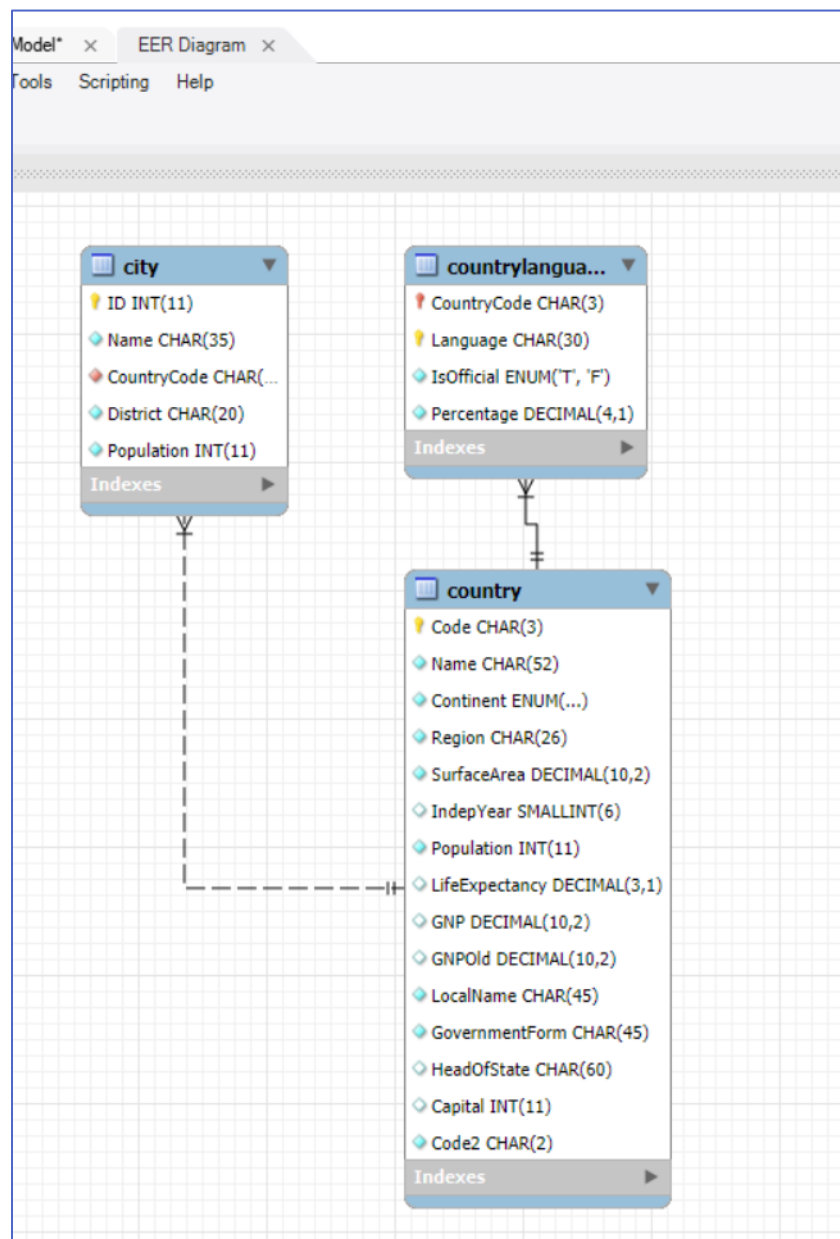


03

FASE 3: INGENIERÍA DE LA CONSULTA SQL (EL CENTRO DEL PROYECTO)

Esta fase fue la más interesante. El **requisito** era un **informe** con el nombre del **país** y su **población** total. También debía incluir las **lenguas** habladas allí.

Observemos el **diagrama Entidad-Relación de world**:



La **tabla country** contiene el país y la población.

La **tabla countrylanguage** contiene las lenguas.

Los **campos Code y CountryCode** relacionan ambas tablas.

El Problema del Producto Cartesiano (Duplicidad):

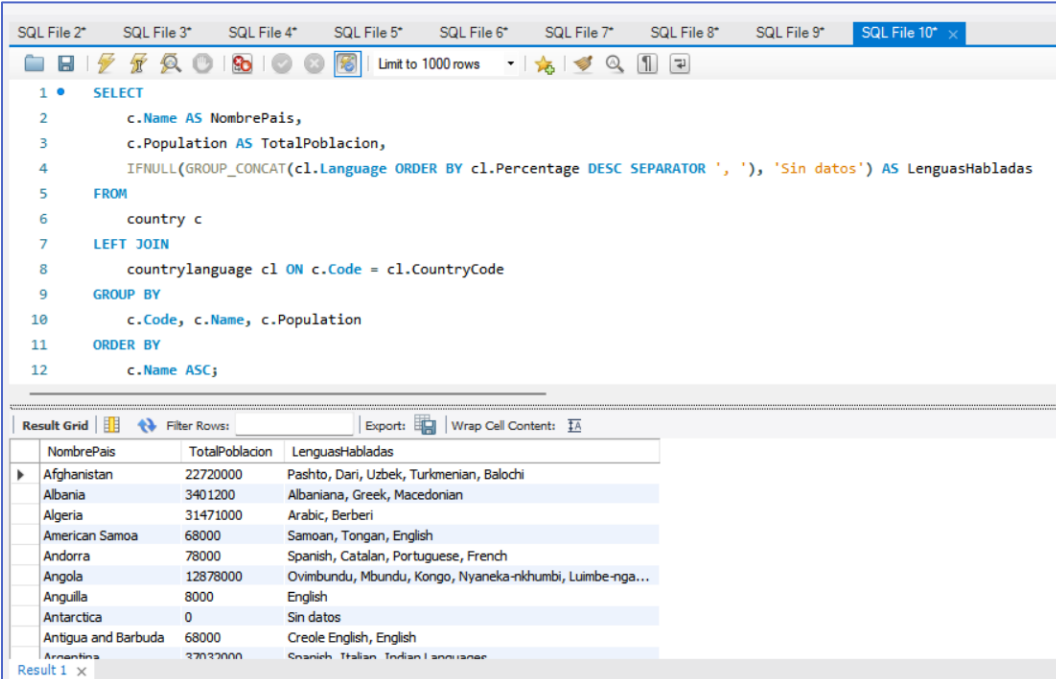
Un **JOIN** normal entre estas tablas en JasperSoft devuelve una **fila por idioma**. La base de datos listaría Canadá dos veces, una por inglés y otra por francés. El dato de población se repetiría en ambas líneas. Esto crea un informe en papel demasiado largo. Resulta difícil de leer. Existían dos soluciones:

Opción A (Desde JasperSoft): Usar el sistema de Grupos del diseñador y agrupar visualmente por país.

Opción B (Desde MariaDB): Resolver el problema en la consulta. Enviamos a JasperSoft los datos ya procesados.

Mi decisión técnica: Elegí la **Opción B**. Preferí delegar el trabajo al motor de MariaDB con la función **GROUP_CONCAT()**. Esta arquitectura es más limpia. La base de datos procesa y **JasperSoft** solo muestra los datos.

La **consulta definitiva** para el diálogo de Dataset and Query fue esta:



The screenshot shows a SQL IDE window with a query editor and a result grid. The query is as follows:

```
1 • SELECT
2     c.Name AS NombrePais,
3     c.Population AS TotalPoblacion,
4     IFNULL(GROUP_CONCAT(c1.Language ORDER BY c1.Percentage DESC SEPARATOR ', '), 'Sin datos') AS LenguasHabladas
5 FROM
6     country c
7 LEFT JOIN
8     countrylanguage c1 ON c.Code = c1.CountryCode
9 GROUP BY
10    c.Code, c.Name, c.Population
11 ORDER BY
12    c.Name ASC;
```

The result grid displays the following data:

NombrePais	TotalPoblacion	LenguasHabladas
Afghanistan	22720000	Pashto, Dari, Uzbek, Turkmenian, Balochi
Albania	3401200	Albanian, Greek, Macedonian
Algeria	31471000	Arabic, Berber
American Samoa	68000	Samoan, Tongan, English
Andorra	78000	Spanish, Catalan, Portuguese, French
Angola	12878000	Ovimbundu, Mbundu, Kongo, Nyaneka-nkhumbi, Lumbu-nga...
Anguilla	8000	English
Antarctica	0	Sin datos
Antigua and Barbuda	68000	Creole English, English
Argentina	37032000	Spanish, Italian, Indian languages

Explicación de la consulta:

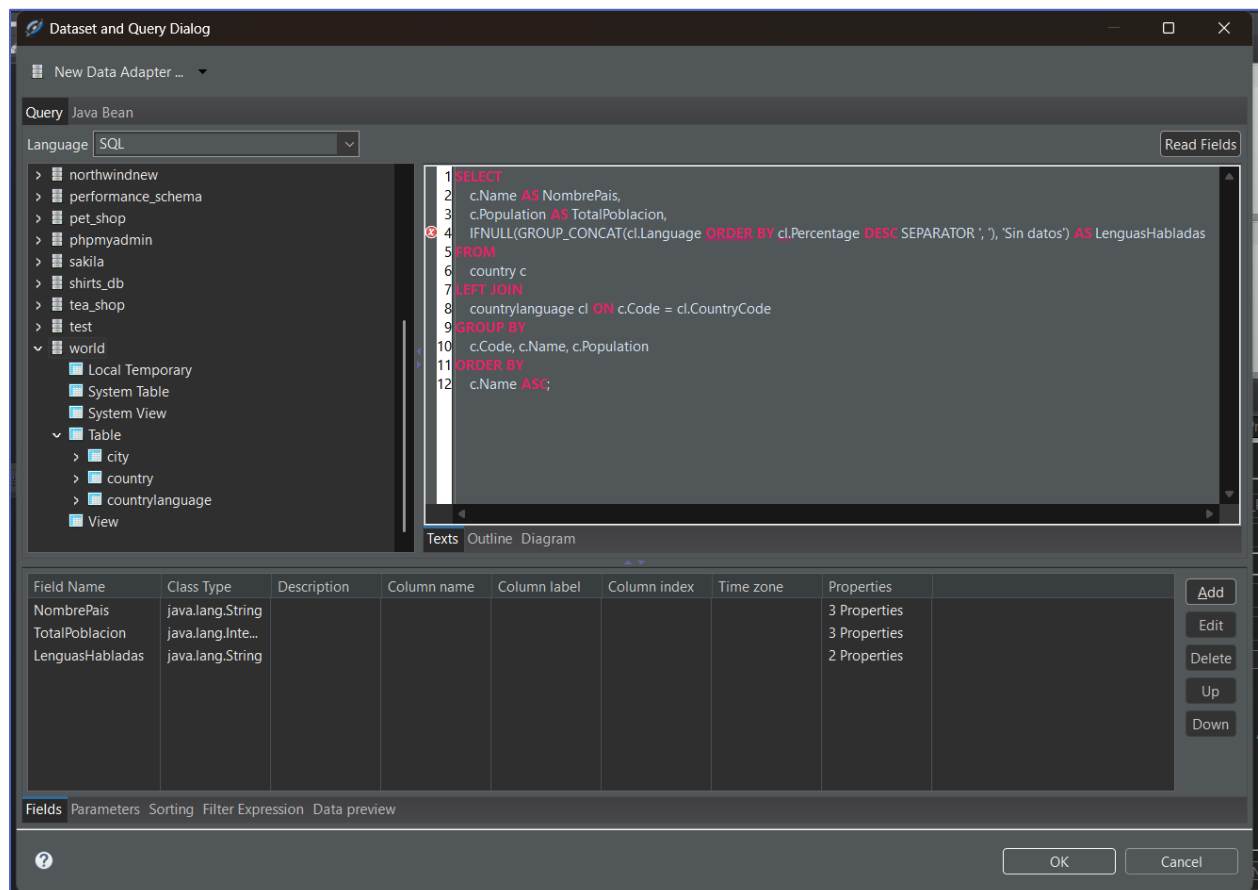
LEFT JOIN frente a INNER JOIN: Algunos lugares, como la Antártida, no tienen registros de idiomas. Un INNER JOIN estricto eliminaría esos países del informe. El resultado estaría incompleto.

La función GROUP_CONCAT: Esta herramienta toma todas las filas de lenguas de un país. Las comprime en una sola celda de texto. Separa los valores con una coma y un espacio.

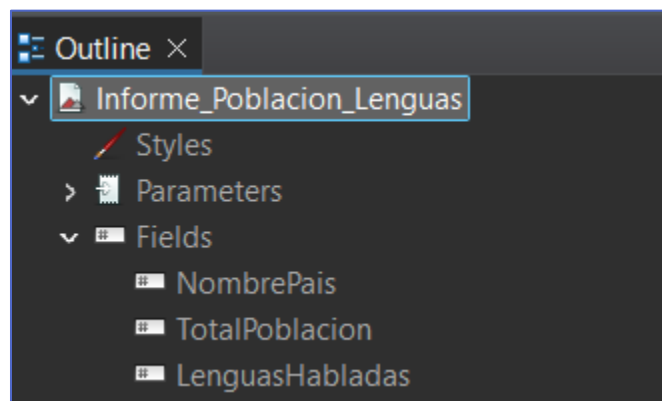
El orden interno: Añadí una mejora de calidad. Los idiomas en la celda no aparecen al azar. Se ordenan de mayor a menor porcentaje de hablantes.

Control de nulos (IFNULL): Esto aplica a casos como la Antártida. Evité que la celda quedara en blanco y pareciera un error. Forcé la aparición del texto "Sin datos".

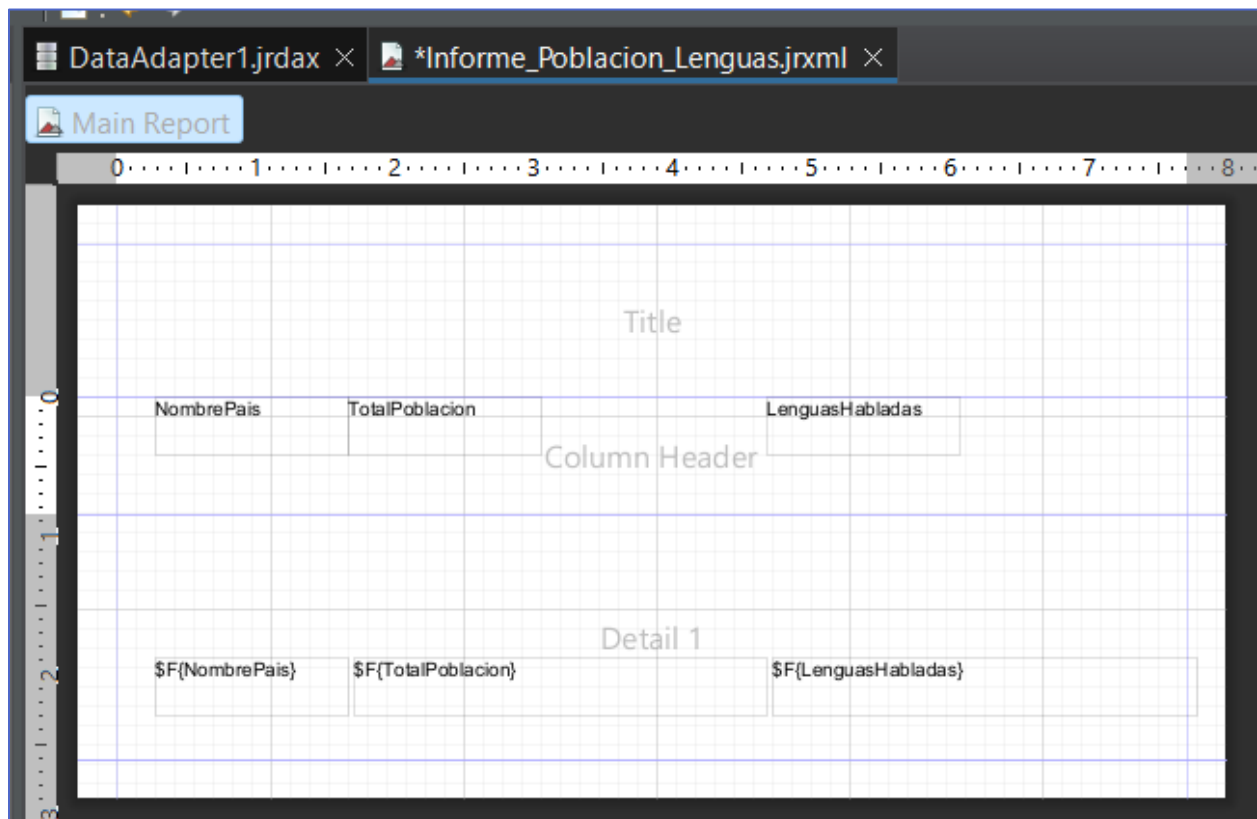
Agrupación y Ordenación Global: Agrupo por los datos del país para activar la concatenación. Luego ordeno alfabéticamente por nombre del país. El informe resulta fácil de consultar.



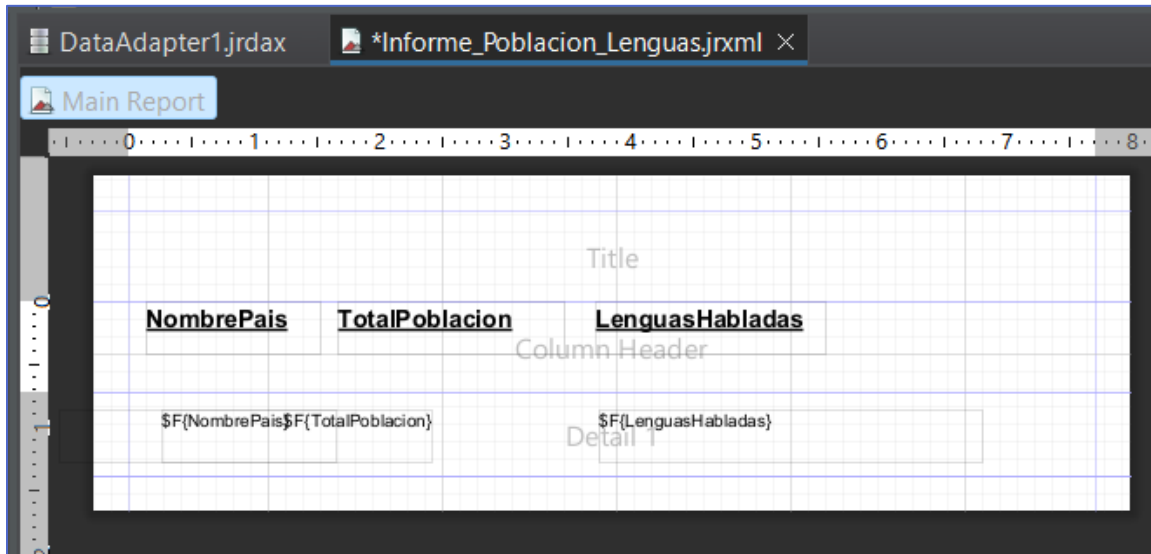
JasperSoft reconoció los campos inmediatamente tras pegar la consulta (\$F{NombrePais}, \$F{TotalPoblacion}, \$F{LenguasHabladas}).



Solo tuve que arrastrarlos a la banda **Detail 1**.



Vamos haciendo **Preview** y lo vamos adaptando para que quede como nos gusta:



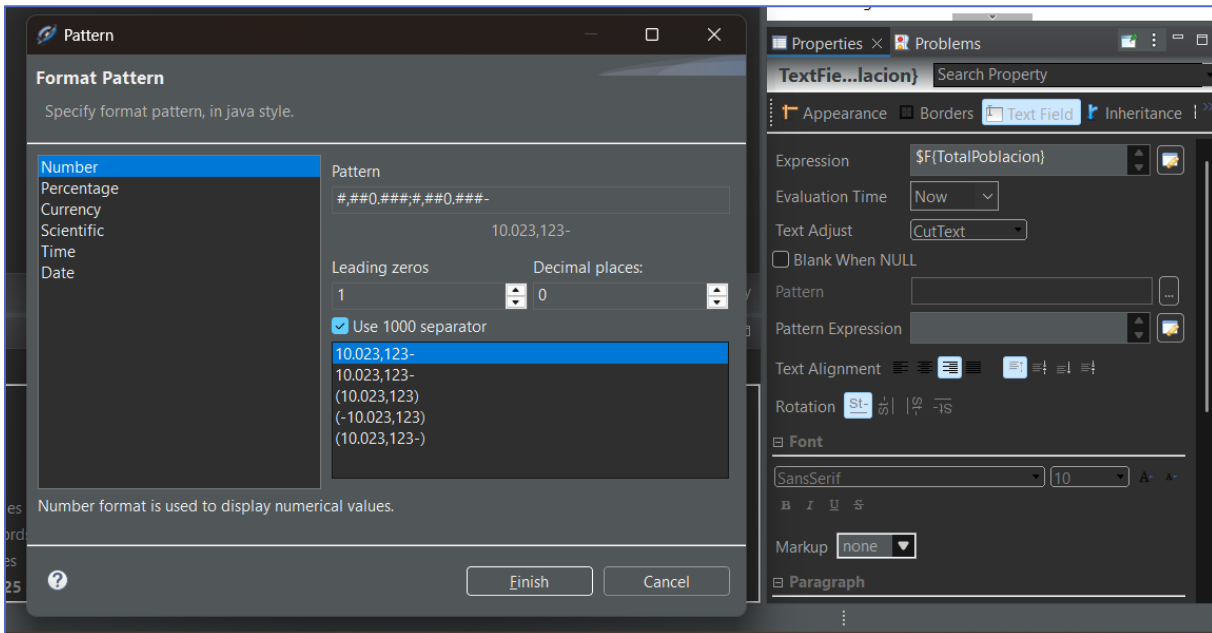
Vista Previa

*Informe_Poblacion_Lenguas.jrxml

Java Page 1 of 16 100%

<u>NombrePais</u>	<u>TotalPoblacion</u>	<u>LenguasHabladas</u>
Afghanistan	22720000	Pashto, Dari, Uzbek, Turkmenian, Balochi
Albania	3401200	Albaniana, Greek, Macedonian
Algeria	31471000	Arabic, Berberi
American Samoa	68000	Samoan, Tongan, English
Andorra	78000	Spanish, Catalan, Portuguese, French
Angola	12878000	Ovimbundu, Mbundu, Kongo, Nyaneka-nkhumbi, Luimbe-nganguela, Chokwe, Luvale, Luchazi,
Anguilla	8000	English
Antarctica	0	Sin datos

Apliqué un **patrón numérico** al campo de población como detalle final. Los puntos de miles aparecen ahora. Leer números largos sin formato es difícil.



*Informe_Poblacion_Lenguas.jrxml

Page 1 of 16

100%

<u>NombrePais</u>	<u>TotalPoblacion</u>	<u>LenguasHabladas</u>
Afghanistan	22.720.000	Pashto, Dari, Uzbek, Turkmenian, Balochi
Albania	3.401.200	Albaniana, Greek, Macedonian
Algeria	31.471.000	Arabic, Berberi
American Samoa	68.000	Samoan, Tongan, English
Andorra	78.000	Spanish, Catalan, Portuguese, French
Angola	12.878.000	Ovimbundu, Mbundu, Kongo, Nyaneka-nkhumbi, Luimbe-nganguela, Chokwe, Luvale, Luchazi,
Anguilla	8.000	English
Antarctica	0	Sin datos

CONCLUSIONES

Integrar JasperSoft con MariaDB es un conocimiento muy útil para el trabajo. Es directamente aplicable. Los gerentes en las empresas evitan el código o la consola. Prefieren un PDF limpio para imprimir o enviar.

La conclusión principal es clara. Un buen informe se diseña en SQL y no en el lienzo. Agrupar elementos con la interfaz gráfica habría sido difícil. Analicé el problema relacional y usé una función avanzada. Solucioné el problema desde la base en el servidor.

El resultado es un informe muy limpio, a Ocupa una sola línea por país. Ahorra papel y se lee con facilidad. El resultado es profesional y no contiene duplicados. Sumaré esta herramienta a mis recursos de desarrollo.

REFERENCIAS

<https://dev.mysql.com/doc/world-setup/en/>

https://mariadb.com/docs/server/reference/sql-functions/aggregate-functions/group_concat

<https://community.jaspersoft.com/documentation/>

<https://mvnrepository.com/artifact/org.mariadb.jdbc/mariadb-java-client>

<https://community.jaspersoft.com/documentation/jaspersoft%C2%AE-studio/tibco-jaspersoft-studio-user-guide/v1000/jss-user-getting-started/>

Introducción a JasperSoft Studio 2025 | Curso de Reportes en Java

<https://www.youtube.com/watch?v=W0sFMzVcCss>

Load and get going with the "world" database using MySQL Workbench

<https://www.youtube.com/watch?v=0L0nDGpQUIU>

<https://www.apachefriends.org/docs/>