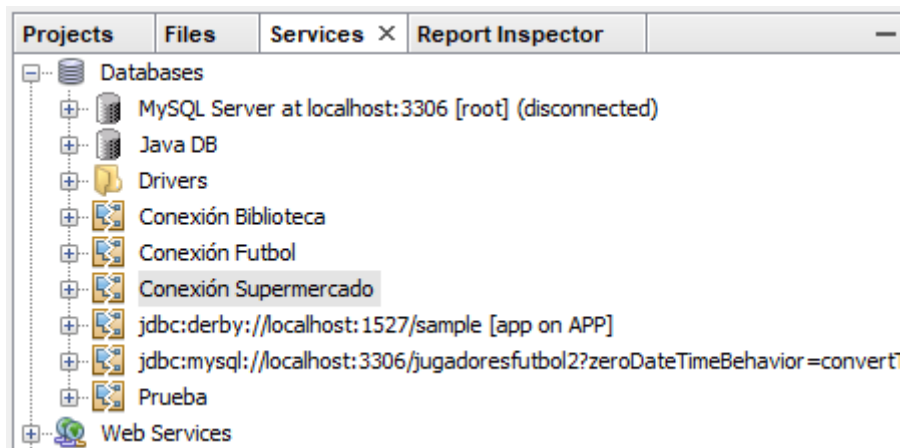


EXAMEN 2º TRIMESTRE DESARROLLO DE INTERFACES

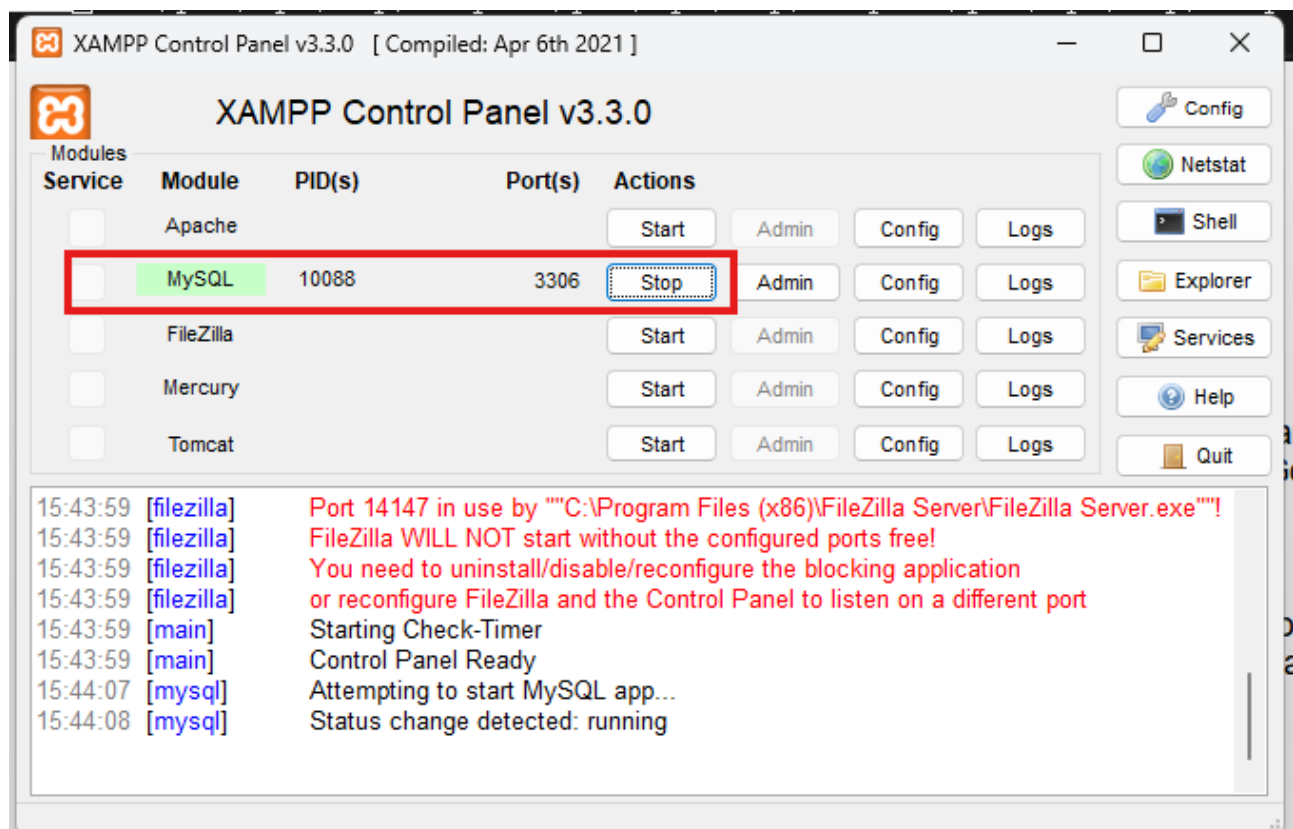
1º Lo primero que vamos a realizar es la búsqueda de la información solicitada para realizar la base de datos en SQL desde NetBeans, yo he encontrado un top en Google con “los mejores digimon” de esta página:

<https://www.hobbyconsolas.com/reportajes/mejores-digimon-112748>

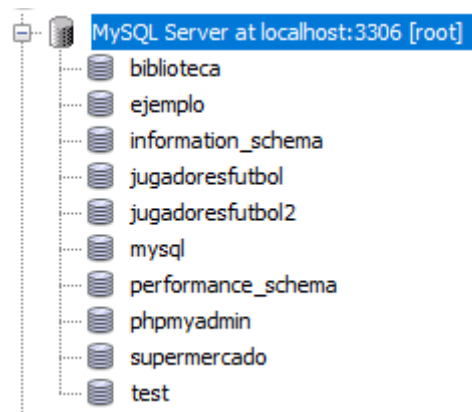
A continuación lo primero que debemos hacer será crear una nueva base de datos en Netbeans y una conexión para conectarnos a ella, para ello accedemos a la pestaña de “Services” y hacemos clic en “Databases”:



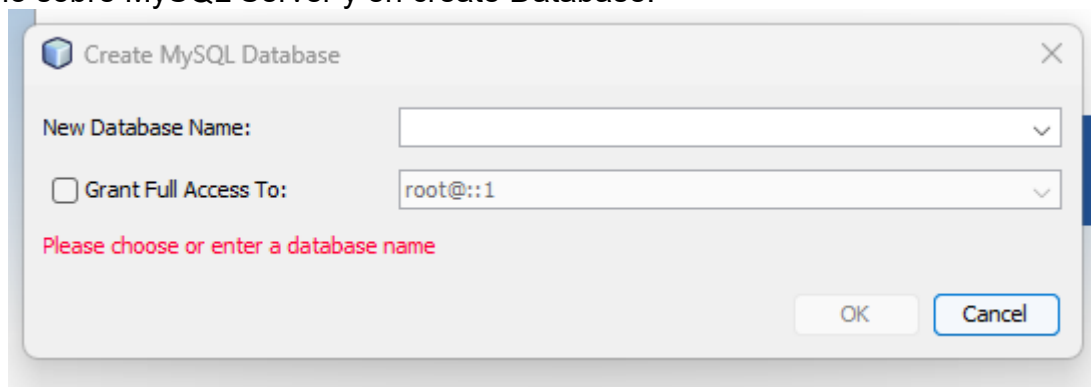
Como vemos nos indica que el servidor de MySQL está desconectado, vamos a conectarlo, primero iniciaremos el XAMPP para iniciar MySQL haciendo clic en start:



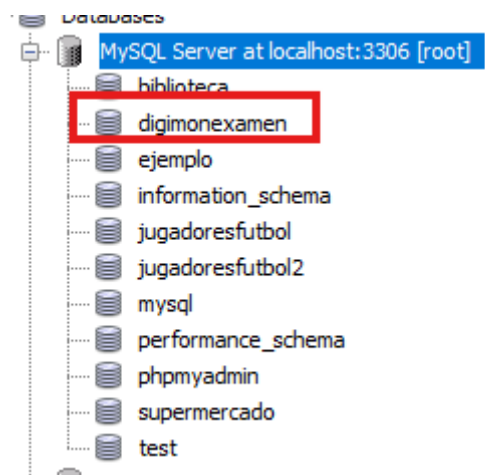
Ahora volvemos a NetBeans y hacemos clic derecho sobre MySQL Server y hacemos clic en start para iniciar el servidor de la base de datos:



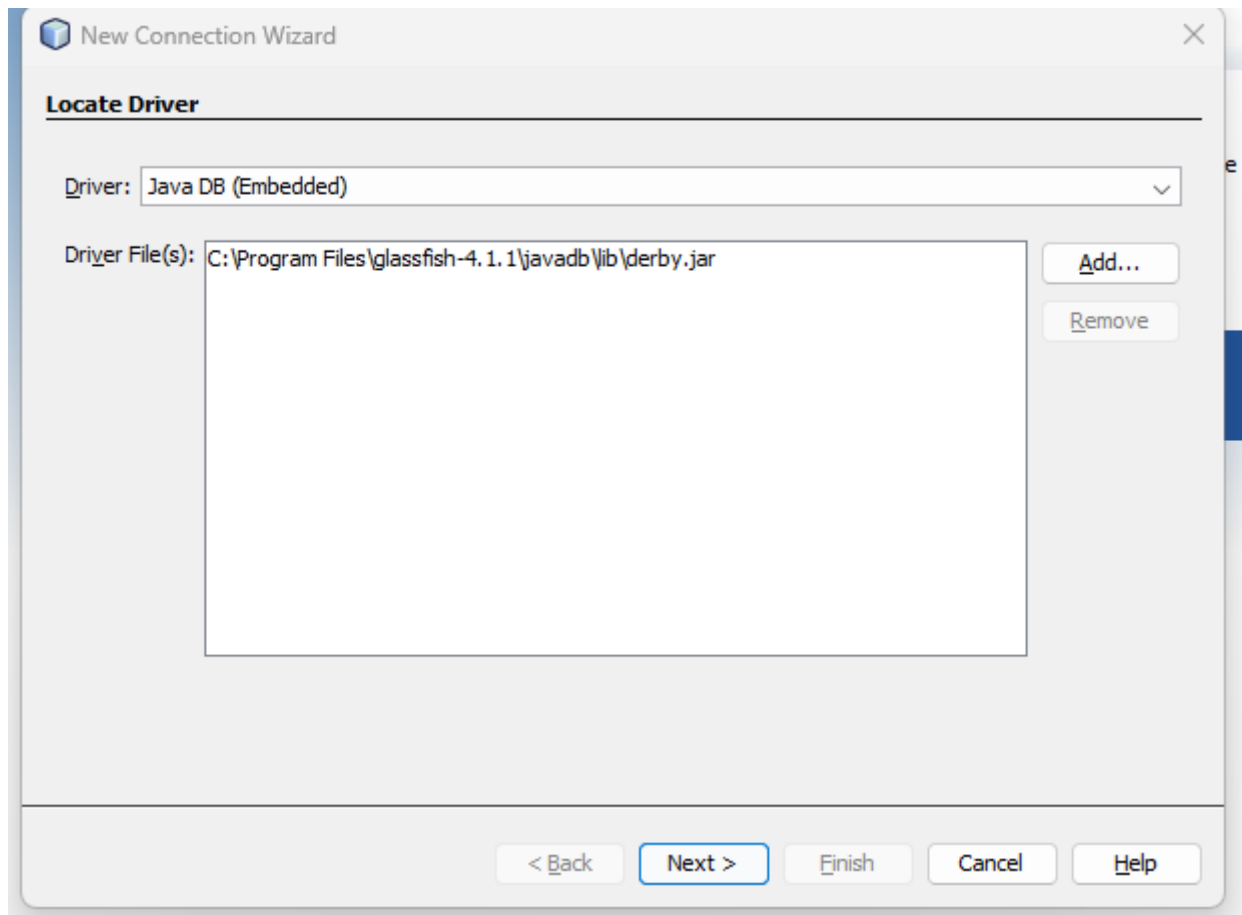
Ahora ya podremos crear una nueva base de datos, simplemente hacemos de nuevo clic derecho sobre MySQL Server y en create Database:



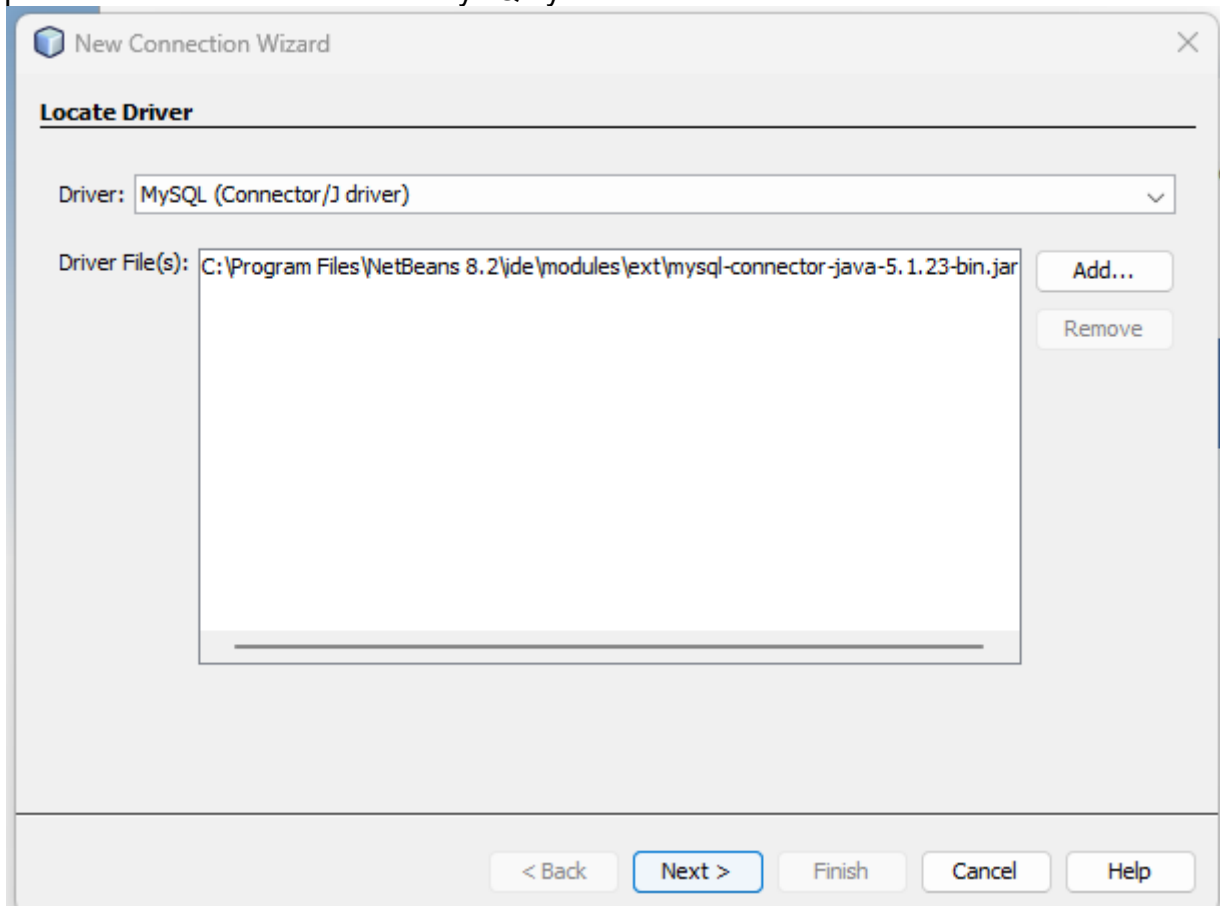
Aquí indicamos el nombre, yo la voy a llamar digimonExamen y hacemos clic en ok:



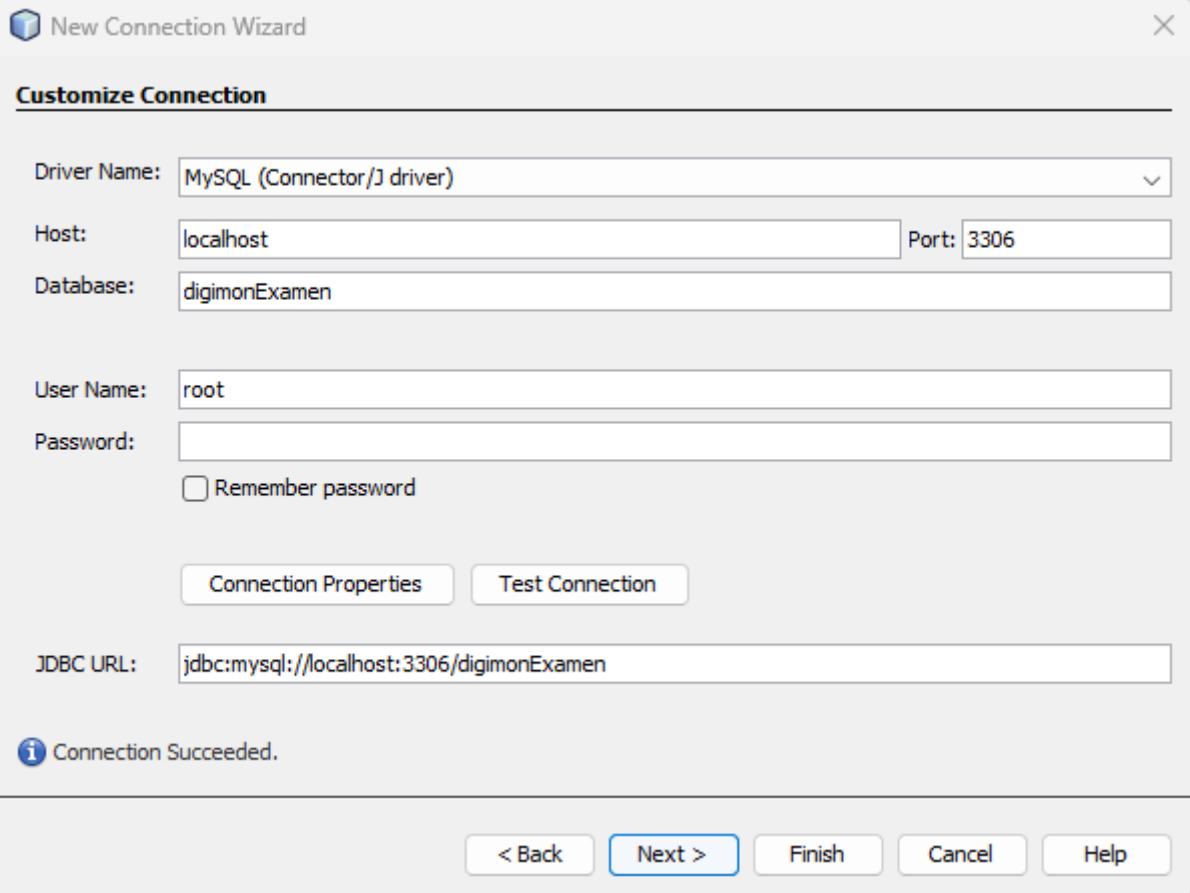
Ahora vamos a realizar la estructura de la base de datos, para ello primero vamos a crear una conexión, hacemos clic derecho en Databases y en New Connection:



Aquí estableceremos el driver a MySQL y hacemos clic en Next:



Aquí nos aseguraremos de que está en el puerto que estemos usando para el servidor de MySQL en mi caso es el 3306 que podemos ver en xampp y establecemos a la base de datos que nos queremos conectar:

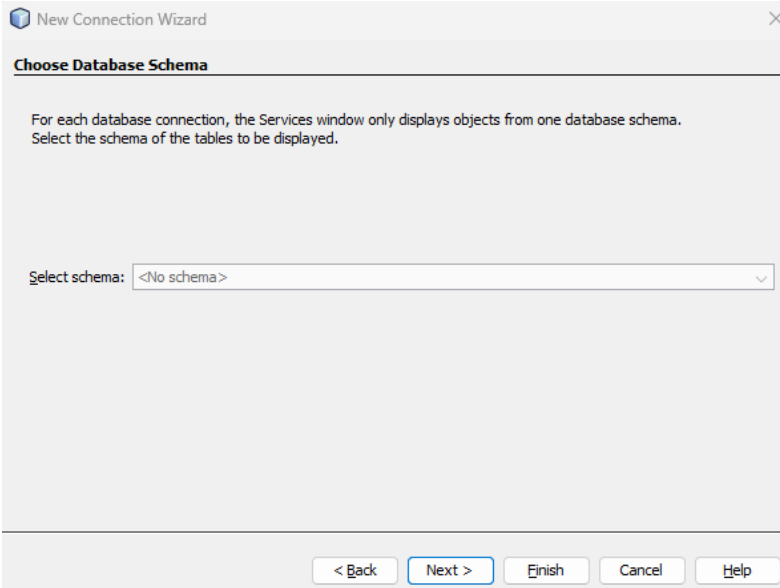


The screenshot shows the 'New Connection Wizard' dialog box, specifically the 'Customize Connection' step. The fields are filled as follows:

- Driver Name: MySQL (Connector/J driver)
- Host: localhost
- Port: 3306
- Database: digimonExamen
- User Name: root
- Password: (empty)
- ☐ Remember password
- Connection Properties
- Test Connection
- JDBC URL: jdbc:mysql://localhost:3306/digimonExamen
- Connection Succeeded.

At the bottom, there are navigation buttons: < Back, Next > (highlighted), Finish, Cancel, and Help.

Dejaremos la contraseña vacía y el usuario como root que es lo que viene por defecto, y en la URL he eliminado lo que viene a partir del nombre de la base de datos ya que no es necesario, ahora hacemos clic en Next:

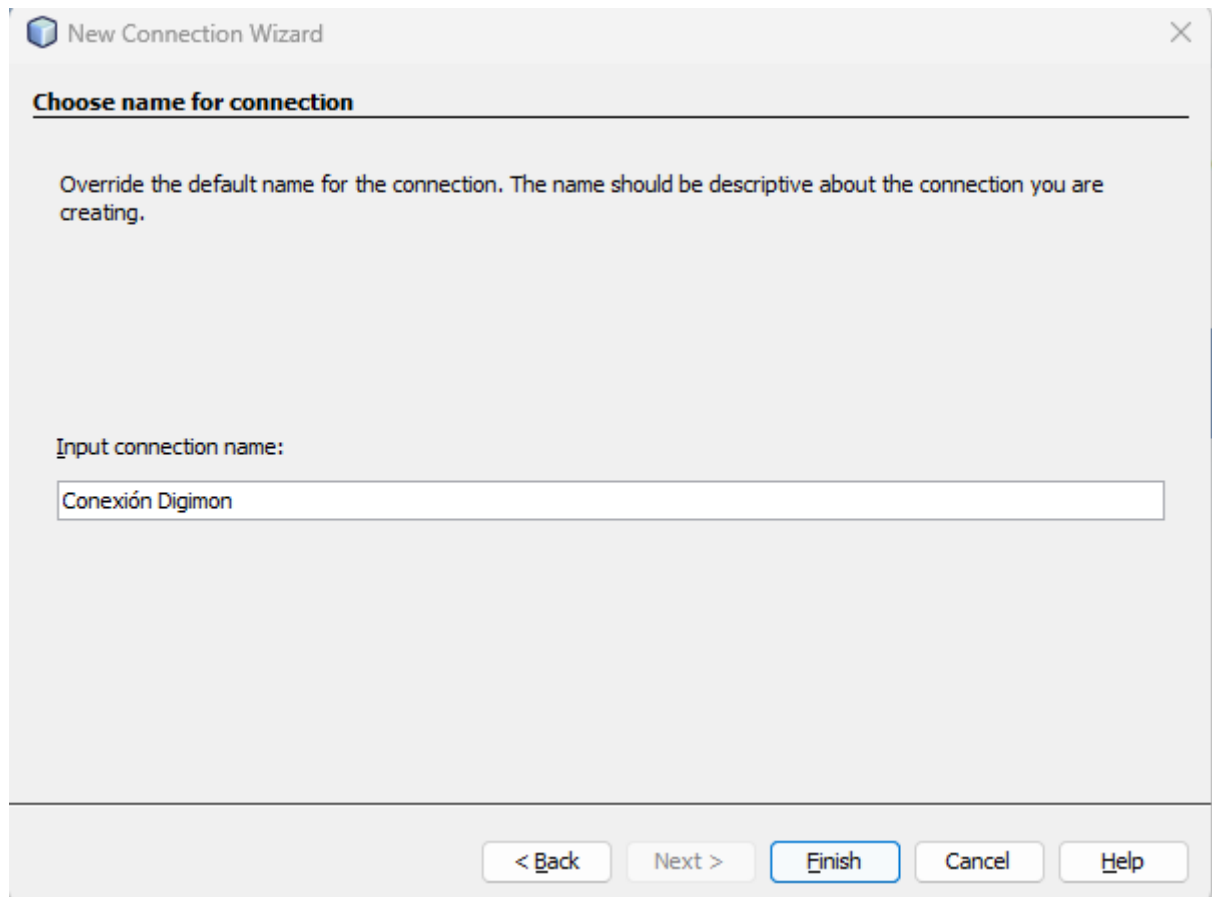


The screenshot shows the 'New Connection Wizard' dialog box, specifically the 'Choose Database Schema' step. The fields are filled as follows:

- For each database connection, the Services window only displays objects from one database schema. Select the schema of the tables to be displayed.
- Select schema: <No schema>

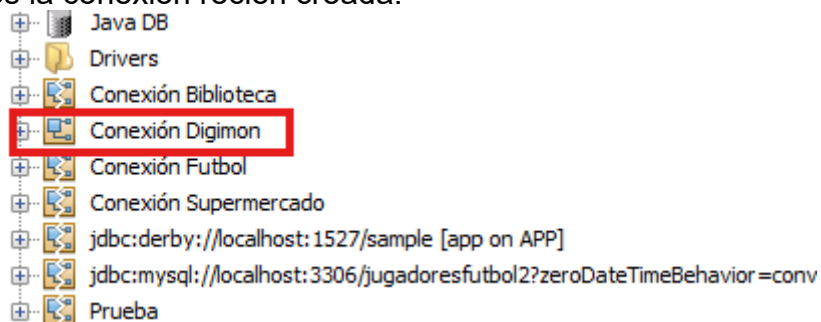
At the bottom, there are navigation buttons: < Back, Next > (highlighted), Finish, Cancel, and Help.

Aquí lo dejamos como está y hacemos clic en Next:



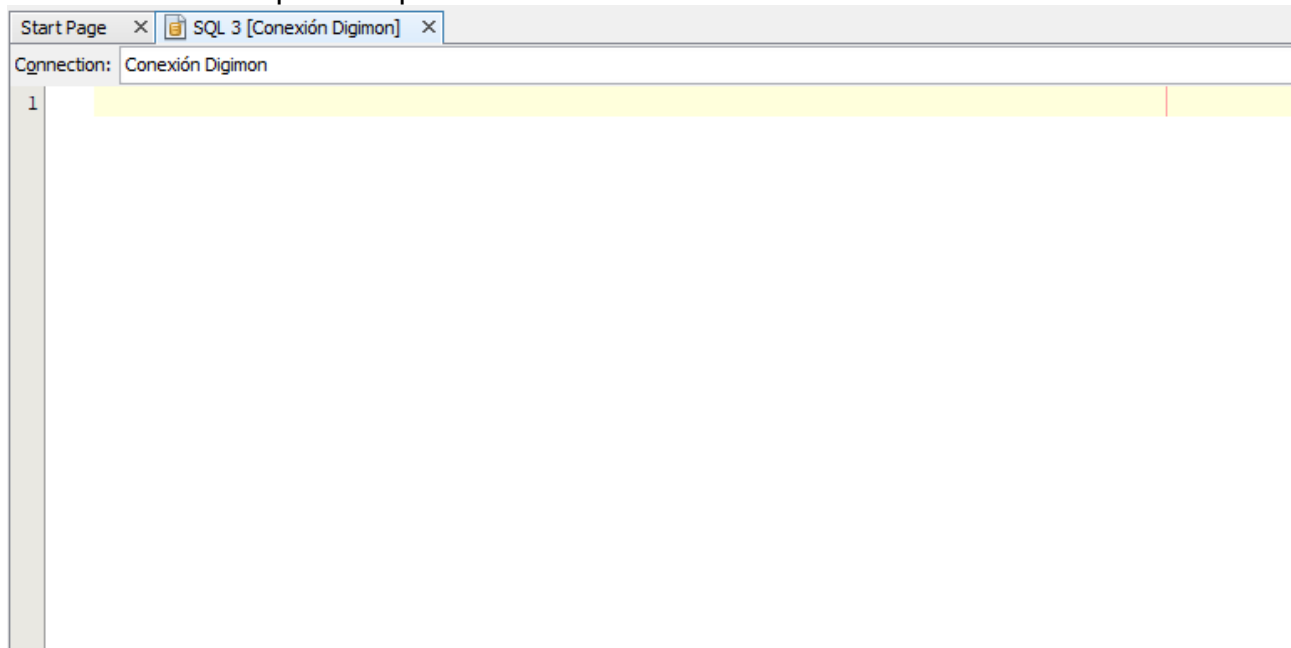
Aquí simplemente ponemos el nombre que queramos que tenga nuestro examen y he puesto Conexión Digimon y hacemos clic en Finish.

Y aquí tendríamos la conexión recién creada:



Ahora vamos a abrir una hoja de trabajo para introducir las consultas SQL, para ello hacemos clic derecho en la conexión y en Execute Command:

Se nos abrirá esta pestaña para realizar nuestra base de datos:



Ahora simplemente vamos a estructurar la base de datos, vamos a crear una tabla con los digimon, para esto pues hacemos una consulta SQL de Create Table:

```
DROP TABLE digimon;

CREATE TABLE digimon(
    codigo int auto_increment primary key,
    nombre VARCHAR(50) UNIQUE,
    tipo VARCHAR (20) NOT NULL,
    color VARCHAR (20) NOT NULL
);
```

Ahora ya podemos insertar datos en nuestra base de datos, vamos a insertar los digimon:

```
INSERT INTO digimon (nombre, tipo, color)
VALUES
    ('Numemon', 'Virus', 'Verde'),
    ('Agumon', 'Vacuna', 'Amarillo'),
    ('SkullGreymon', 'Virus', 'Blanco'),
    ('Leomon', 'Vacuna', 'Amarillo'),
    ('Devimon', 'Virus', 'Negro'),
    ('WereGarurumon', 'Vacuna', 'Azul'),
    ('Andromon', 'Datos', 'Gris'),
    ('Myotismon', 'Virus', 'Gris'),
    ('Wizardmon', 'Datos', 'Morado'),
    ('Etemon', 'Virus', 'Naranja'),
    ('Piedmon', 'Virus', 'Rojo'),
    ('Angemon', 'Vacuna', 'Blanco'),
    ('MetalGarurumon', 'Datos', 'Gris'),
    ('WarGreymon', 'Vacuna', 'Naranja'),
    ('MagnaAngemon', 'Vacuna', 'Blanco');
```

Ahora para consultarlos podemos hacer una consulta básica de tipo `Select * from digimon`:

#	codigo	nombre	tipo	color
1		1 Numemon	Virus	Verde
2		2 Agumon	Vacuna	Amarillo
3		3 SkullGreymon	Virus	Blanco
4		4 Leomon	Vacuna	Amarillo
5		5 Devimon	Virus	Negro
6		6 WereGarurumon	Vacuna	Azul
7		7 Andromon	Datos	Gris
8		8 Myotismon	Virus	Gris
9		9 Wizardmon	Datos	Morado
10		10 Etemon	Virus	Naranja
11		11 Piedmon	Virus	Rojos
12		12 Angemon	Vacuna	Blanco
13		13 MetalGarurumon	Datos	Gris
14		14 WarGreymon	Vacuna	Naranja
15		15 MagnaAngemon	Vacuna	Blanco

Y aquí vemos como nos aparecen todos.

2º Elaboración de reporte:

Ahora a través de la herramienta jasperReports y la base de datos de Digimon vamos a realizar un reporte para el cliente, para ello lo primero que tenemos que hacer es abrir la herramienta jasperReports en NetBeans, para ello si no la tenemos abierta hacemos clic en Window y en Welcome to JasperReport y se nos abrirá esta pestaña:

Recent reports

No recent files

Quick start

Follow these three steps to create your first report

Step 1: Create a database connection or setup a data source. Click on the icon to run the connection setup wizard.

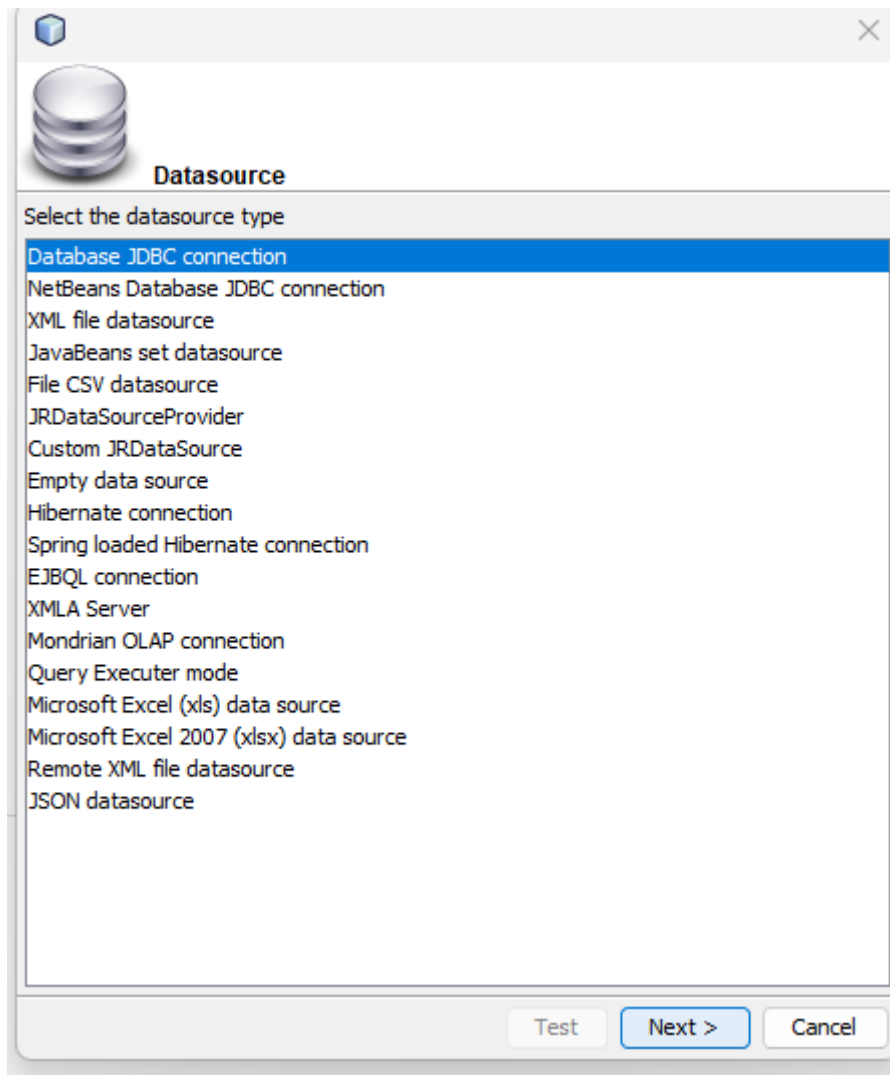
Step 2: Create a new report. Click on the icon to run the report wizard.

Step 3: Press the Preview button (in the designer window tab) to run and preview your report.

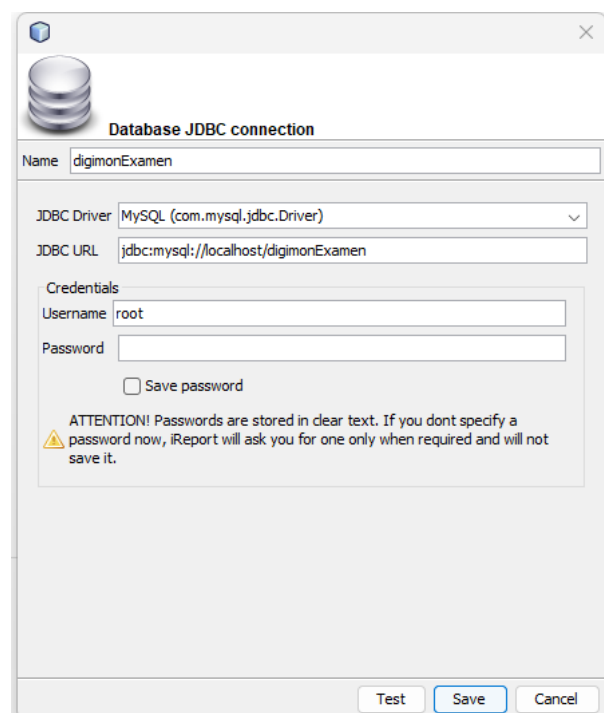
Spotlight

Resource Center

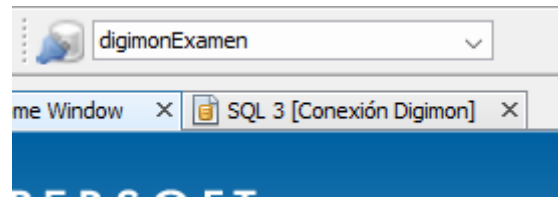
Ahora lo primero que tenemos que hacer es hacer clic donde dice Create Database para vincular la base de datos que acabamos de crear con JasperReports:



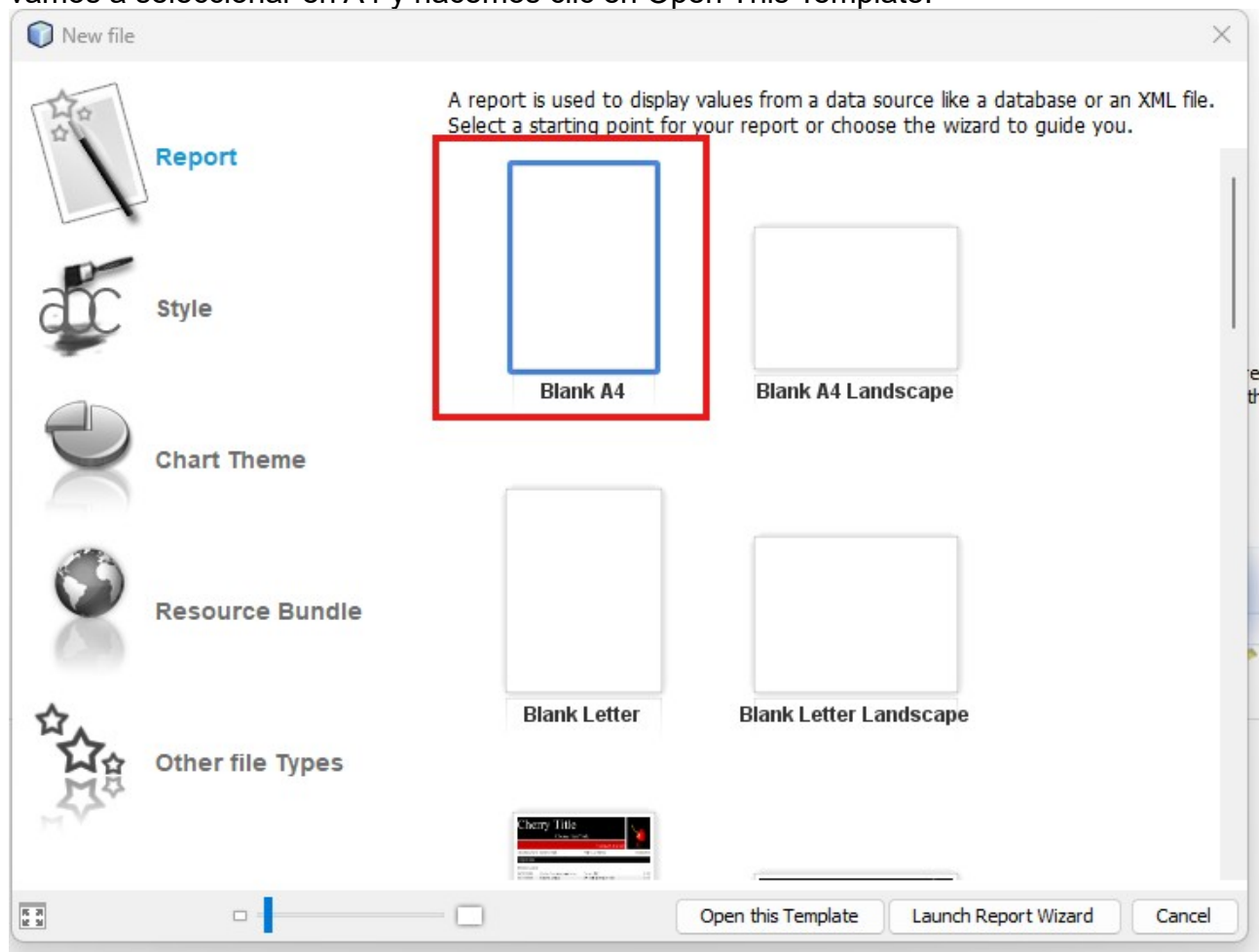
Aquí lo dejamos como está y hacemos clic en Next.
Ahora configuraremos los parámetros para la conexión:



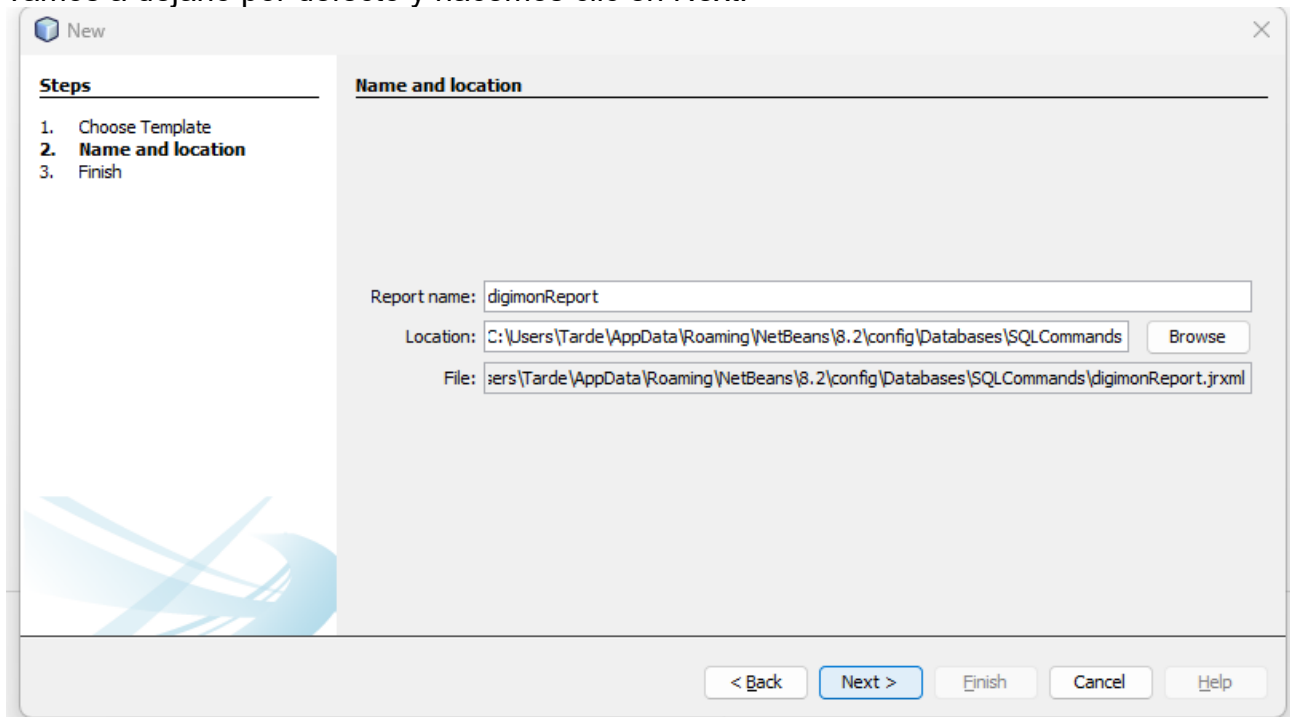
Es muy importante que establezcamos correctamente el nombre de la base de datos en el apartado JDBC URL y que seleccionemos el driver MySQL correcto que son los que tengo puestos, también hay que introducir el usuario y contraseña que en este caso es solo root, hacemos clic en Test para comprobar que está todo bien y en Save:



Ahora nos aparece la conexión para realizar las consultas desde JasperReports. Lo siguiente que debemos hacer ahora es clic en el step 2, para empezar con la creación del informe, lo primero que se nos solicita es en que formato queremos nuestro informe, vamos a seleccionar en A4 y hacemos clic en Open This Template:

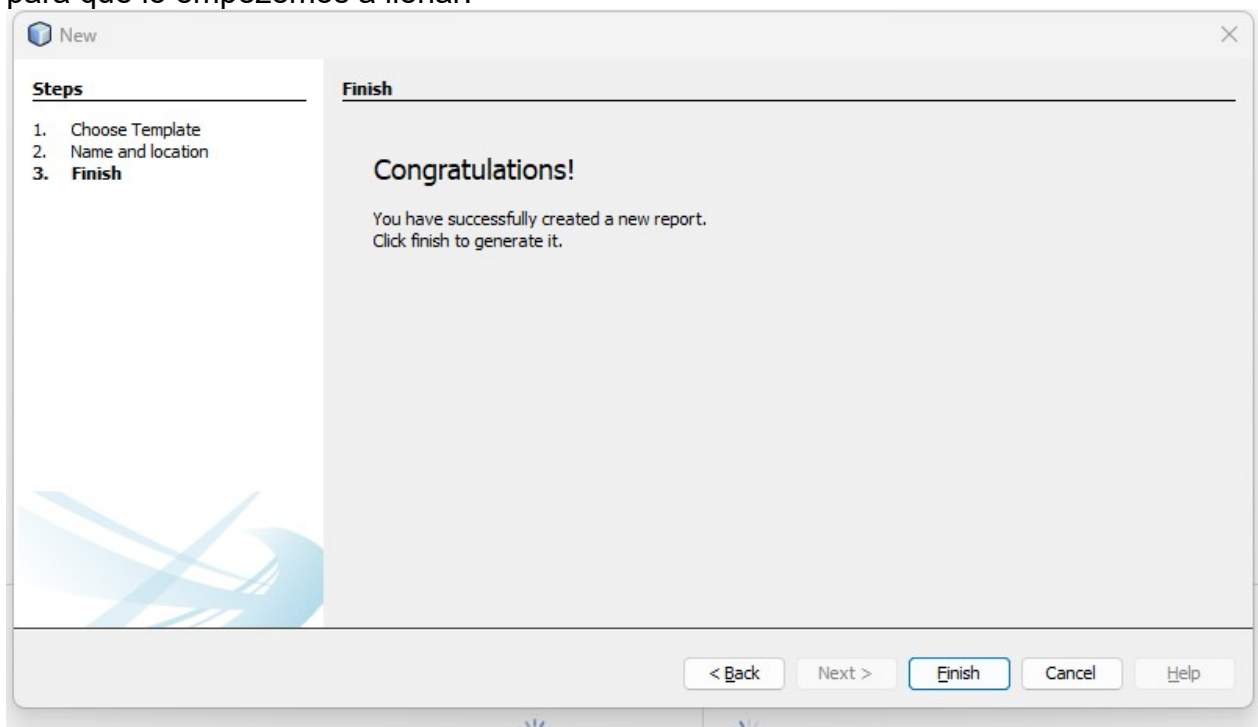


Ahora establecemos un nombre para nuestro report y donde queremos que se guarde, vamos a dejarlo por defecto y hacemos clic en Next:



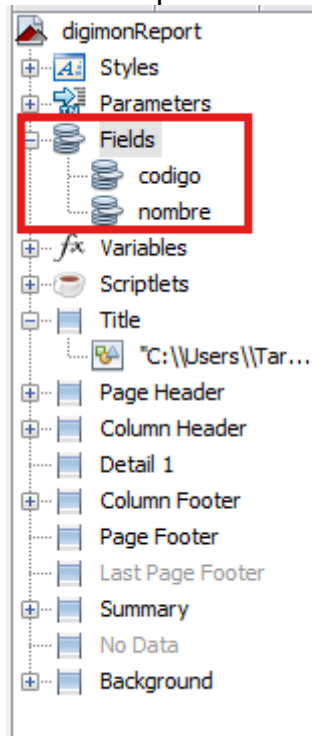
The screenshot shows the 'New' dialog box in NetBeans. On the left, under the 'Steps' section, there is a list: 1. Choose Template, 2. **Name and location**, and 3. Finish. The main area is titled 'Name and location'. It contains three input fields: 'Report name:' with the text 'digimonReport', 'Location:' with the path 'C:\Users\Tarde\AppData\Roaming\NetBeans\8.2\config\Databases\SQLCommands', and 'File:' with the path 'ers\Tarde\AppData\Roaming\NetBeans\8.2\config\Databases\SQLCommands\digimonReport.jrxml'. A 'Browse' button is next to the Location field. At the bottom, there are five buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'. The 'Next >' button is highlighted.

Con esto habremos creado el reporte, hacemos clic en Finish y vemos el reporte vacío para que lo empecemos a llenar:

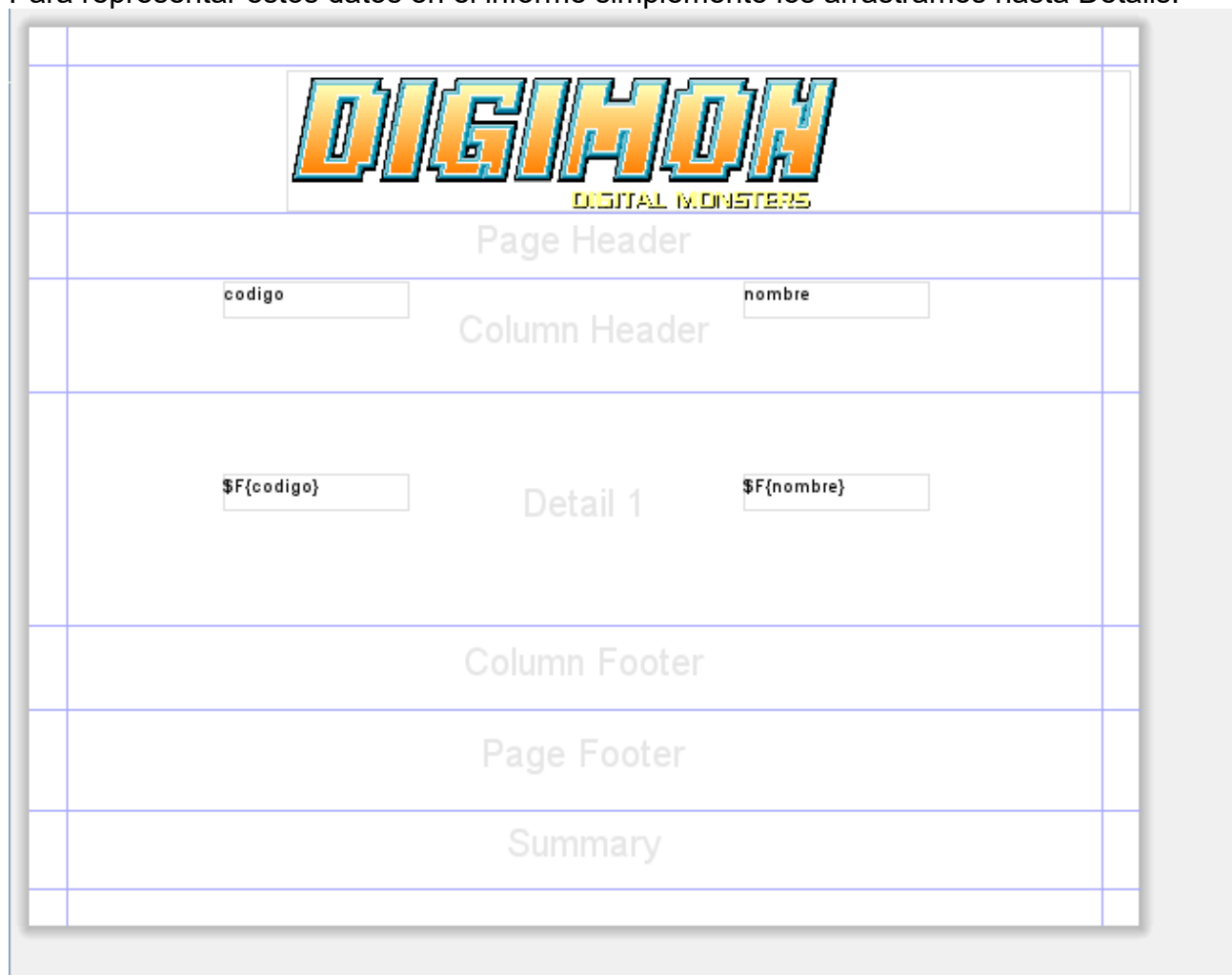


The screenshot shows the 'New' dialog box in NetBeans, now at the 'Finish' step. The 'Steps' list on the left shows: 1. Choose Template, 2. Name and location, and 3. **Finish**. The main area is titled 'Finish' and displays 'Congratulations!' followed by the text: 'You have successfully created a new report. Click finish to generate it.' At the bottom, the buttons are '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'. The 'Finish' button is highlighted.

Aquí como podemos ver hemos recuperado los campos nombre y código de cada digimon, ahora nos aparecerán estos campos en el apartado de la izquierda:



Para representar estos datos en el informe simplemente los arrastramos hasta Details:



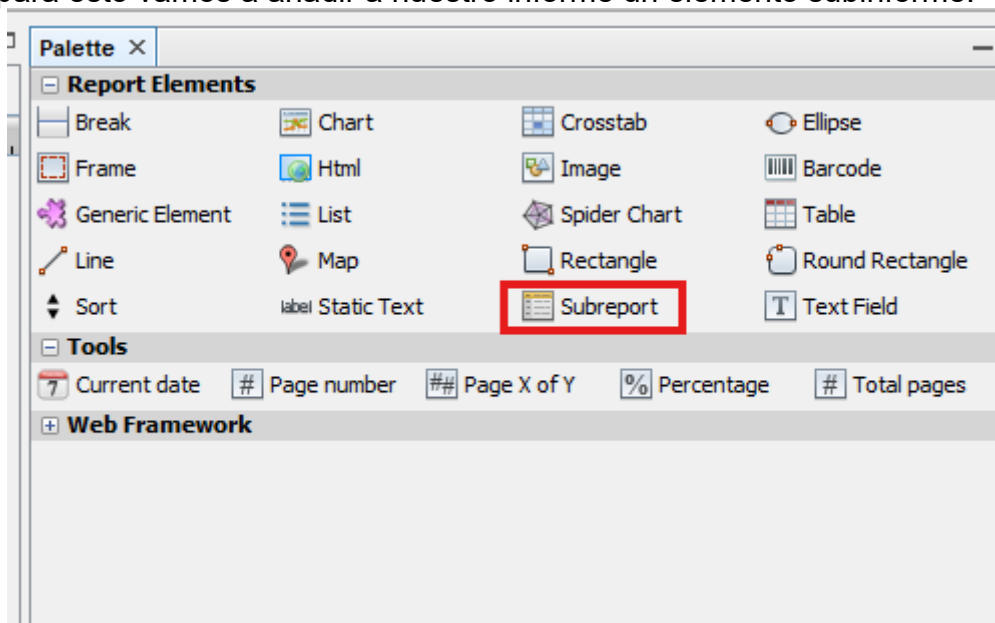
Y si ahora hacemos clic en preview veríamos el informe con la información básica:

codigo	nombre
2	Agumon
7	Andromon
12	Angemon
5	Devimon
10	Etemon
4	Leomon
15	MagnaAngemon
13	MetalGarurumon
8	Myotismon
1	Numemon
11	Piedmon
3	SkullGreymon
14	WarGreymon
6	WereGarurumon
9	Wizardmon

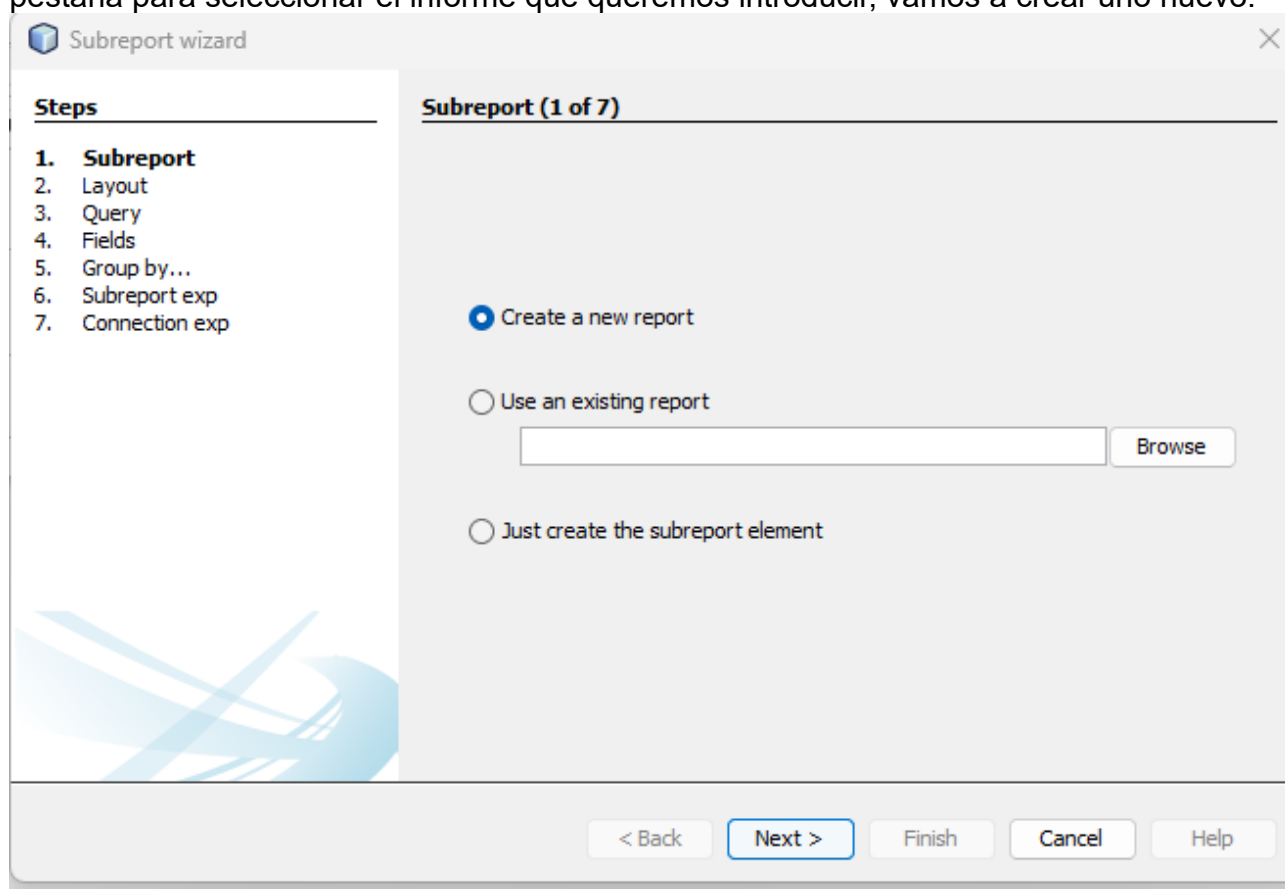
Están ordenados por defecto por nombre, si queremos ordenarlos por código simplemente en la consulta añadimos un order by codigo:

codigo	nombre
1	Numemon
2	Agumon
3	SkullGreymon
4	Leomon
5	Devimon
6	WereGarurumon
7	Andromon
8	Myotismon
9	Wizardmon
10	Etemon
11	Piedmon
12	Angemon
13	MetalGarurumon
14	WarGreymon
15	MagnaAngemon

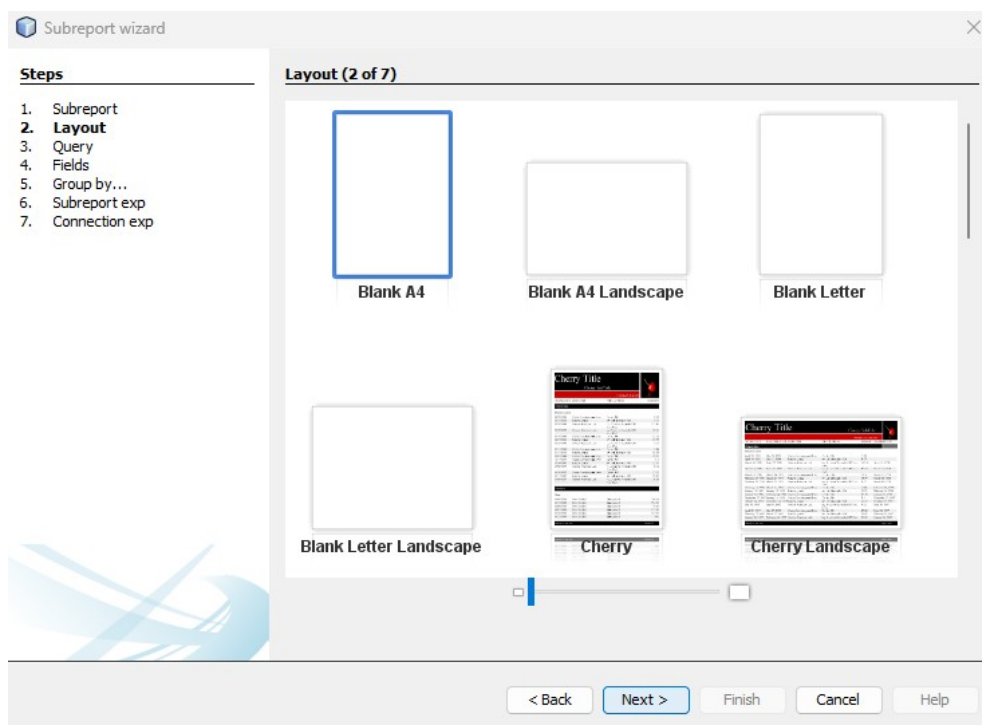
Ahora vamos a crear un subinforme que nos de los datos del tipo y del color de cada digimon, para esto vamos a añadir a nuestro informe un elemento subinforme:



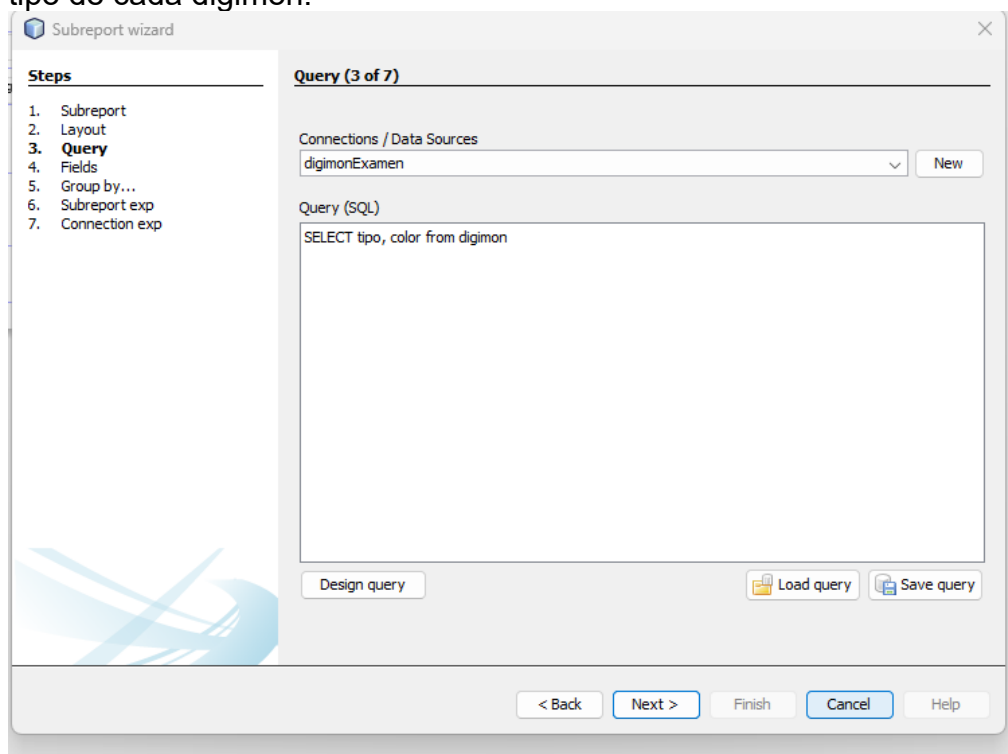
El subinforme debemos introducirlo en el apartado de Page Header para evitar problemas de colocación de los informes, una vez arrastremos el subinforme se nos abrirá una pestaña para seleccionar el informe que queremos introducir, vamos a crear uno nuevo:



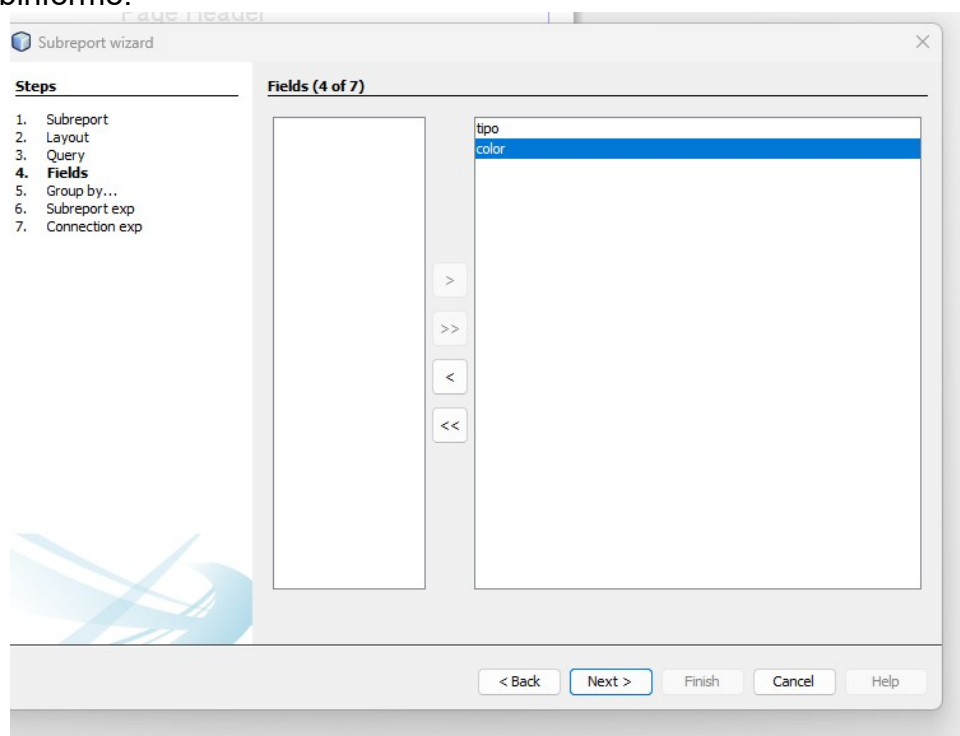
Hacemos clic en Next y seleccionamos de nuevo el formato a4 y hacemos clic otra vez en next:



Aquí ahora podemos establecer la base de datos y la consulta que queremos utilizar para recuperar los datos, como ya sabemos cual es vamos a introducir la consulta para obtener el color y tipo de cada digimon:



Aquí los campos nos aparecerán a la izquierda, los pasamos a la derecha para usarlos en nuestro subinforme.



Aquí nos solicita si lo queremos ordenar por alguna de las dos características, vamos a dejarlo como esta:

Subreport wizard

Steps

1. Subreport
2. Layout
3. Query
4. Fields
- 5. Group by...**
6. Subreport exp
7. Connection exp

Group by... (5 of 7)

Group 1
[Dropdown menu]

Group 2
[Dropdown menu]

Group 3
[Dropdown menu]

Group 4
[Dropdown menu]

< Back Next > Finish Cancel Help

Y aquí configuramos el nombre y la ruta que queremos que tenga nuestro reporte:

Subreport wizard

Steps

1. Subreport
2. Layout
3. Query
4. Fields
5. Group by...
- 6. Subreport exp**
7. Connection exp

Subreport exp (6 of 7)

Report name: digimonReport_subreport1

Location: C:\Users\Tarde\AppData\Roaming\NetBeans\8.2\config\Databases\SQLCommands Browse

File: ppData\Roaming\NetBeans\8.2\config\Databases\SQLCommands\digimonReport_subreport1.jrxml

☒ Store the directory name in a parameter
\$P{SUBREPORT_DIR} + "digimonReport_subreport1.jasper"

☐ Use a static absolute path reference
"C:\\Users\\Tarde\\AppData\\Roaming\\NetBeans\\8.2\\config\\Databases\\..."

< Back Next > Finish Cancel Help

Esto lo dejamos como está y hacemos clic en Finish:

Subreport wizard

Steps

1. Subreport
2. Layout
3. Query
4. Fields
5. Group by...
6. Subreport exp
- 7. Connection exp**

Connection exp (7 of 7)

☒ Use the same connection used to fill the master report

☐ Use another connection

☐ Use a JRDataSource expression

☐ Use an empty data source

☐ Don't use any connection or data source

< Back Next > **Finish** Cancel Help

Y con esto habríamos creado nuestro subinforme:

Title

Page Header

Column Header

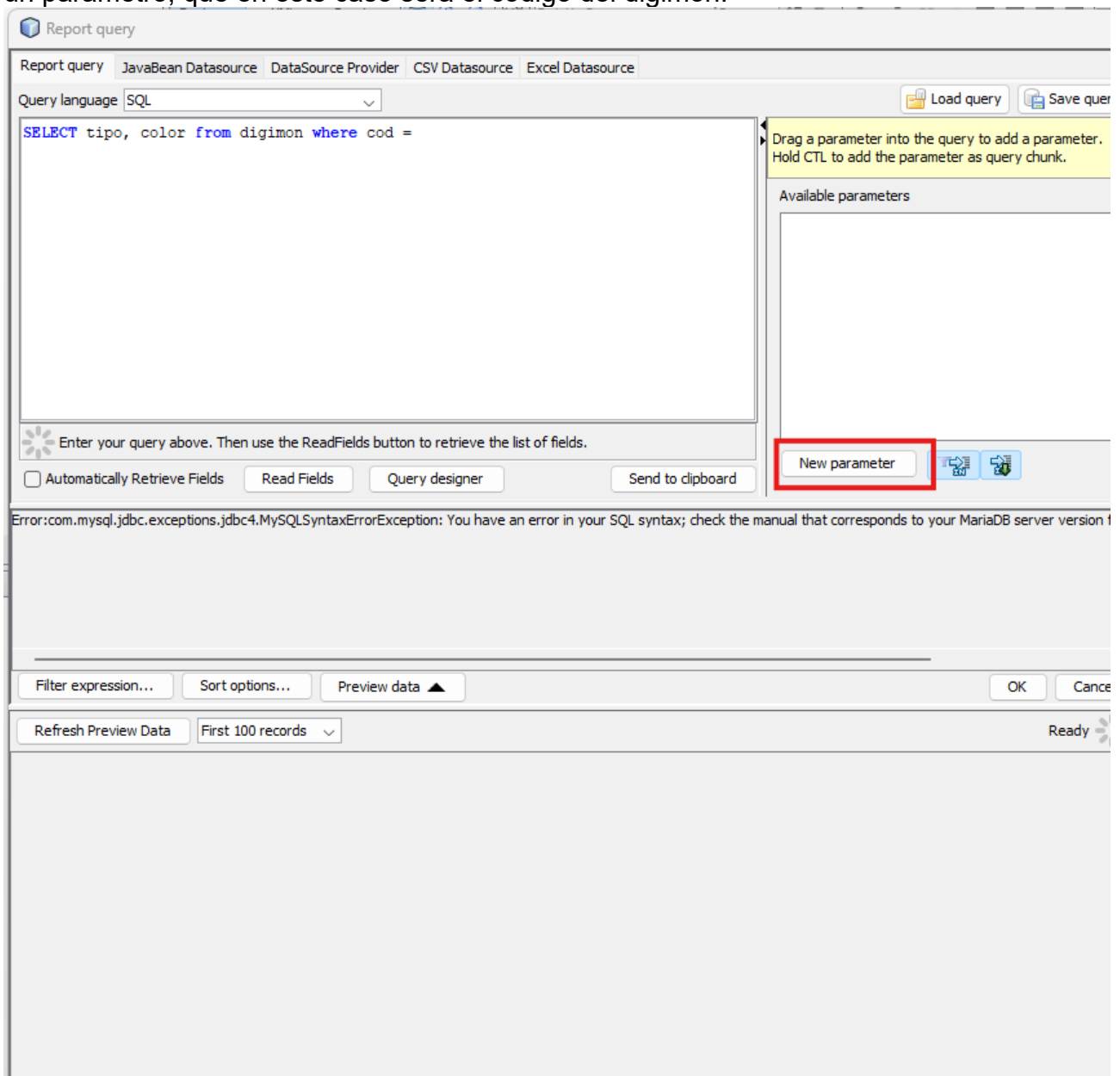
Detail 1

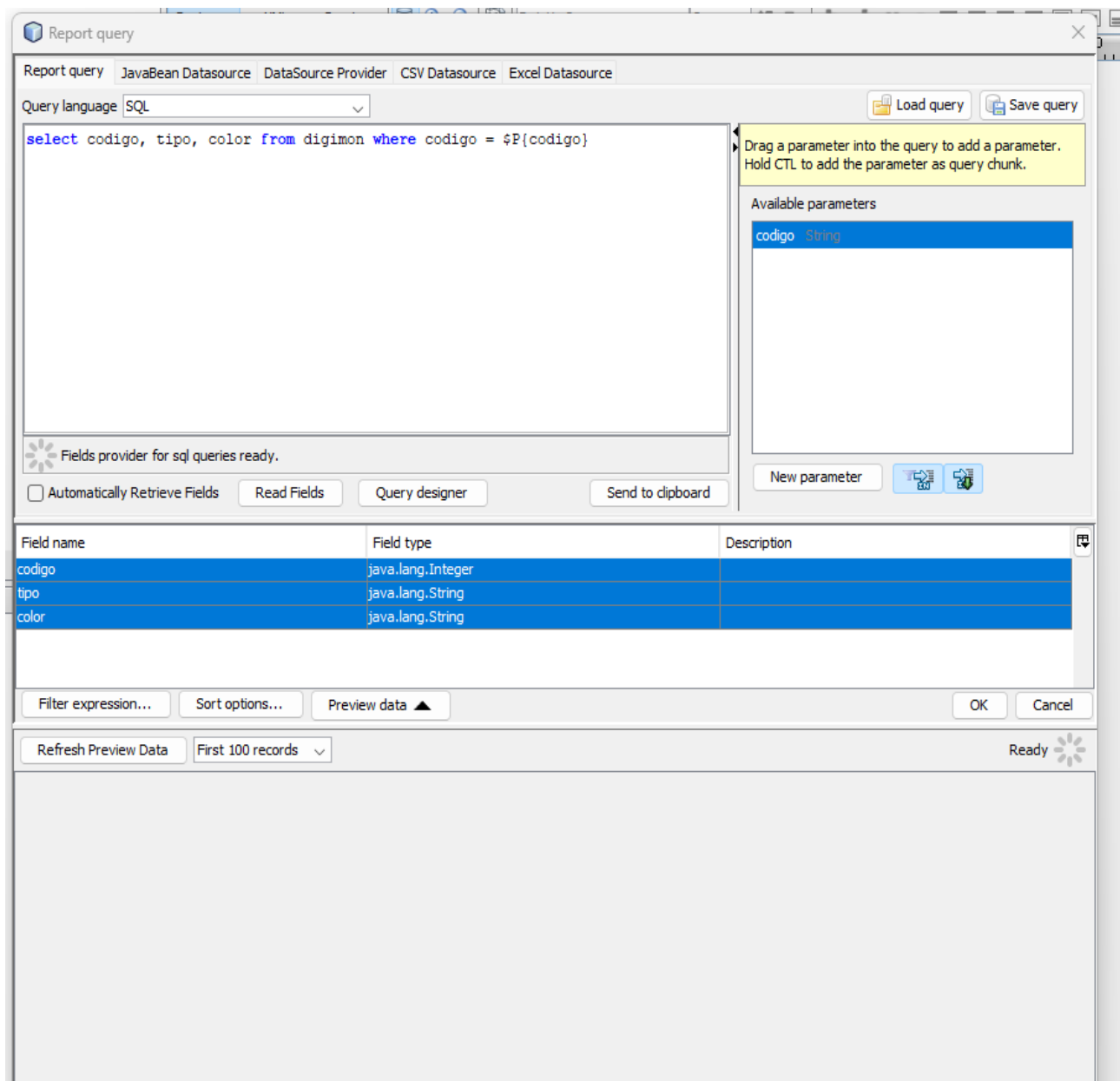
Column Footer

Page Footer

Summary

Ahora vamos a modificar la consulta para obtener el color y el tipo de cada digimon y que no nos de todos los colores y tipos repartidos por el informe, para esto tenemos que crear un parámetro, que en este caso será el código del digimon:

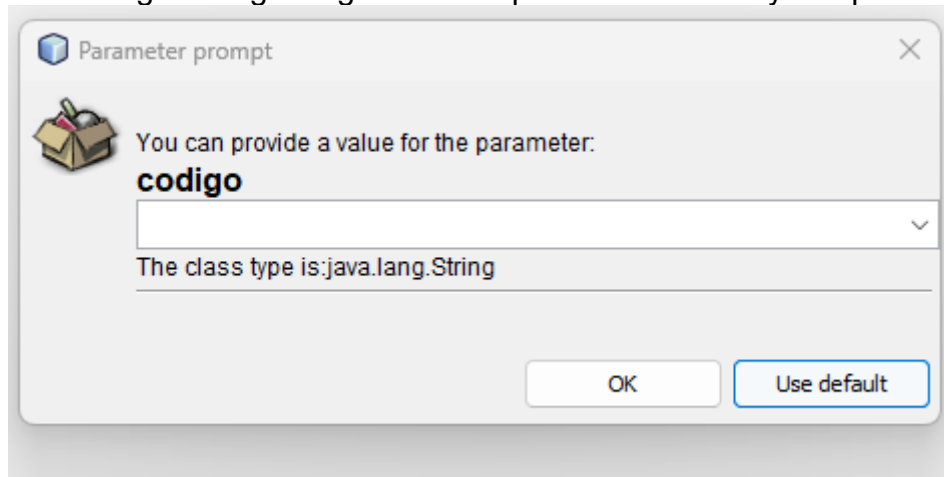




también tendremos que seleccionar el codigo para poder filtrar por el, ahora simplemente arrastramos la información y borramos el resto de bandas del reporte:



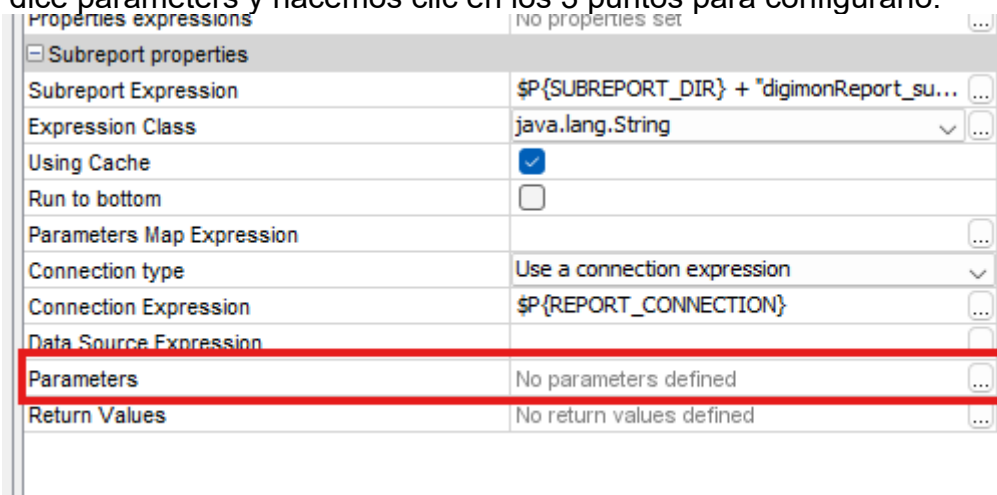
Ahora si hacemos una preview del informe podremos ver que nos solicita un parámetro y si le pasamos el código de algún digimon nos aparecerá su color y su tipo:



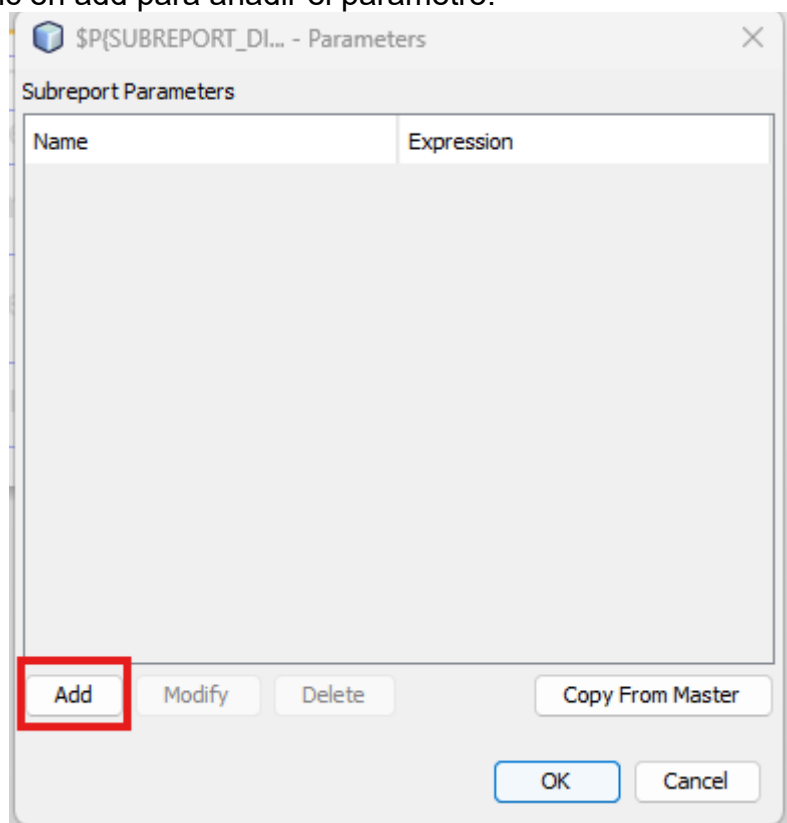
Virus

Verde

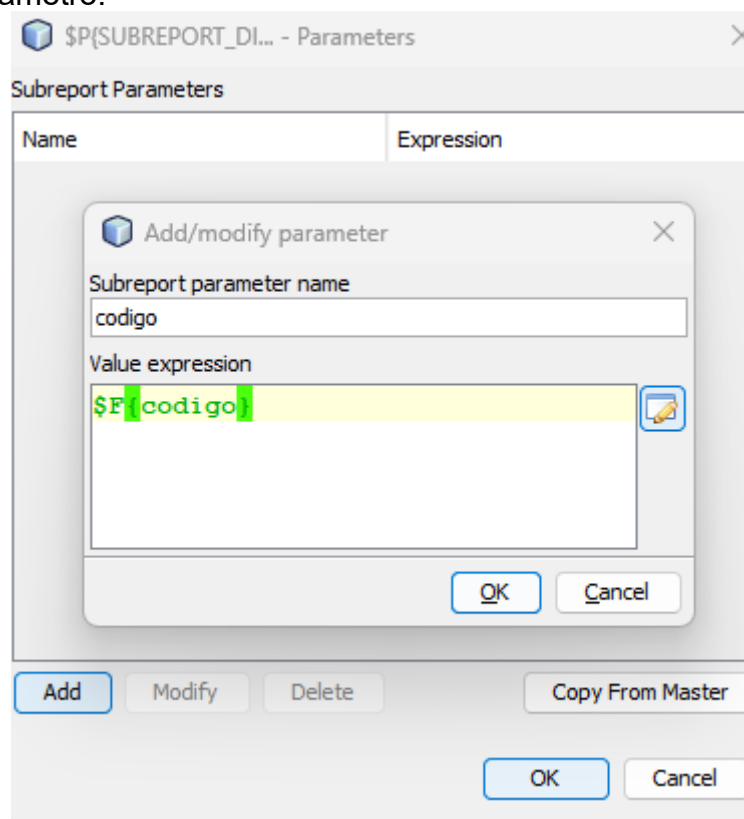
Ahora volvemos al informe principal y tenemos que establecer que para cada subinforme le pasamos el código de cada digimon del informe principal, para esto nos vamos a la parte que dice parameters y hacemos clic en los 3 puntos para configurarlo:



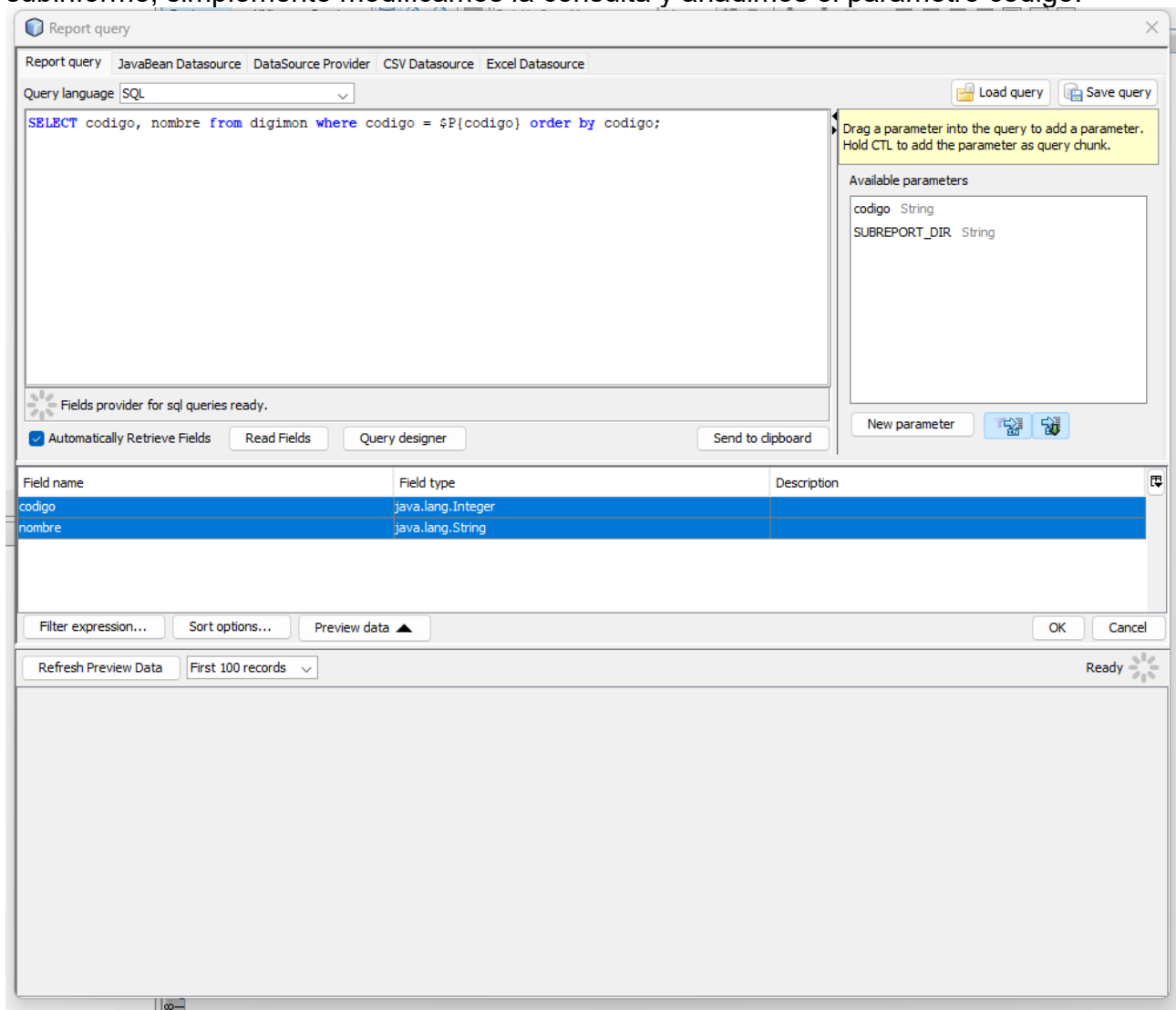
Aquí hacemos clic en add para añadir el parámetro:



Y añadimos el parámetro:



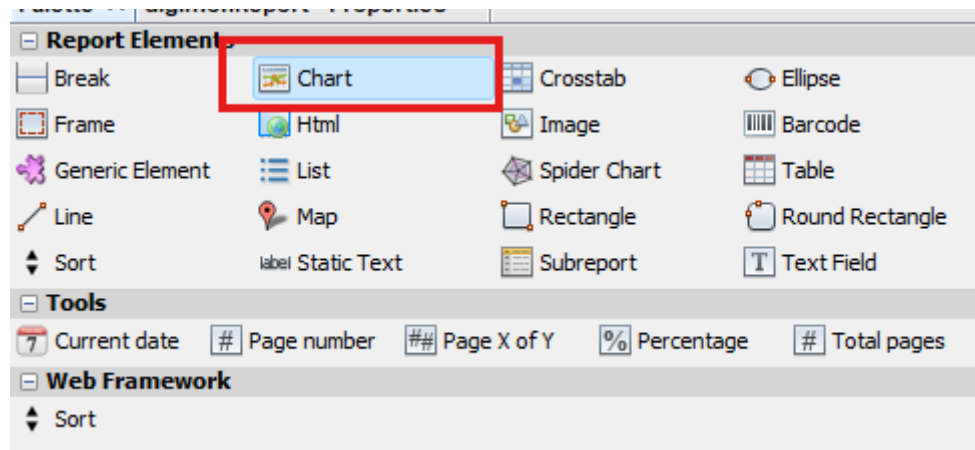
Ahora vamos a cambiar la consulta de nuestro informe principal para añadir un filtrado de digimon por codigo también para que nos aparezca de cada digimon su informe con su subinforme, simplemente modificamos la consulta y añadimos el parámetro codigo.



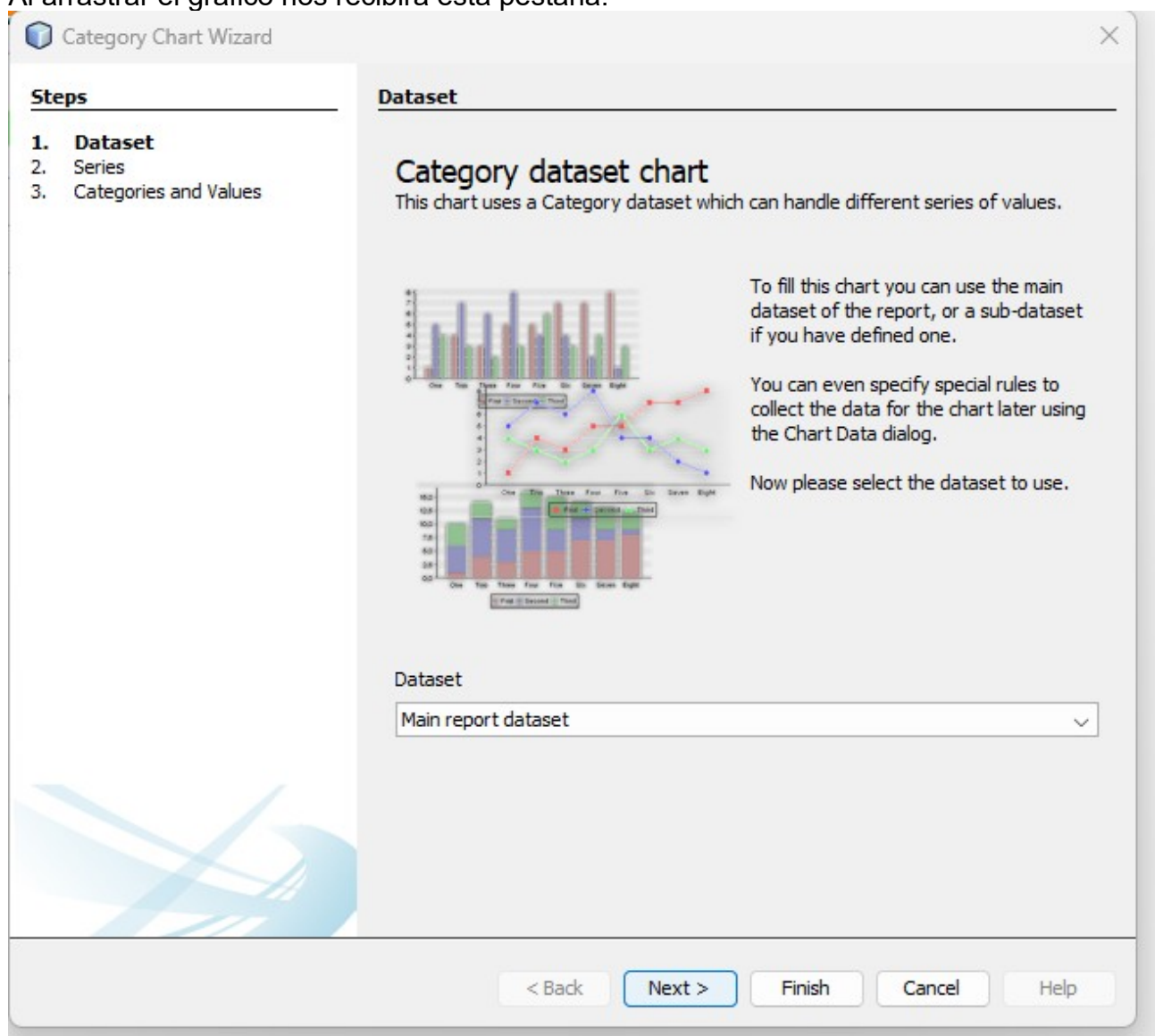
Finalmente así sería nuestro informe, que sería como una hoja de detalle de cada digimon:



Ahora vamos a incluir un gráfico en nuestro informe de cuantos digimon hay por tipo, para esto simplemente vamos a añadir un elemento de gráfico:



Al arrastrar el gráfico nos recibirá esta pestaña:



Hacemos clic en next:

Aquí ponemos el nombre que queramos para nuestro gráfico y otra vez en next:

Category Chart Wizard

Steps

1. Dataset

2. **Series**

3. Categories and Values

Series

Series

A category chart requires three expressions: the first identifies the series, the second the category and the last the value.

The **categories** appear on the X axis.

The **values** appear on the Y axis.

Series are represented by different colors (the series names are shown in the legend).

Category	First	Second	Third
One	1	5	4
Two	4	7	3
Three	3	6	2
Four	5	8	3
Five	5	4	6
Six	7	4	3
Seven	7	2	4
Eight	8	1	3

First

Second

Third

The series expression is useful when you need to represent several sets of data in your chart. If you need a single series chart, set the series expression to a constant value like an empty string ("").

"Numero de tipos de digimon"

< Back

Next >

Finish

Cancel

Help

Category Chart Wizard

Steps

1. Dataset

2. Series

3. Categories and Values

Categories and Values

Category and Value

The categories appear on the X axis. Different colors represent different series. On the Y axis appear the values (specified by the value expression).

Category	First	Second	Third
One	1	5	4
Two	4	7	3
Three	3	6	2
Four	5	8	3
Five	5	4	6
Six	7	4	3
Seven	7	2	4
Eight	8	1	3

The category expression.

\$F{tipo}

The value expression. It must return a numeric value.

\$V{COLUMN_COUNT}

< Back

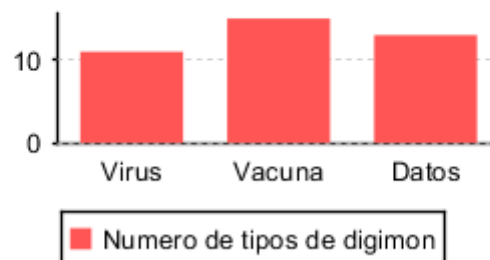
Next >

Finish

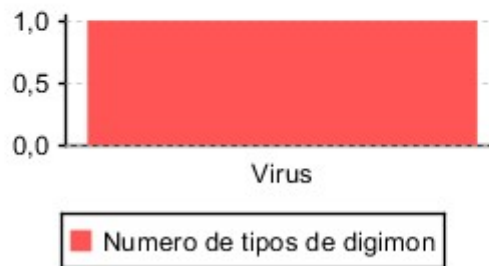
Cancel

Help

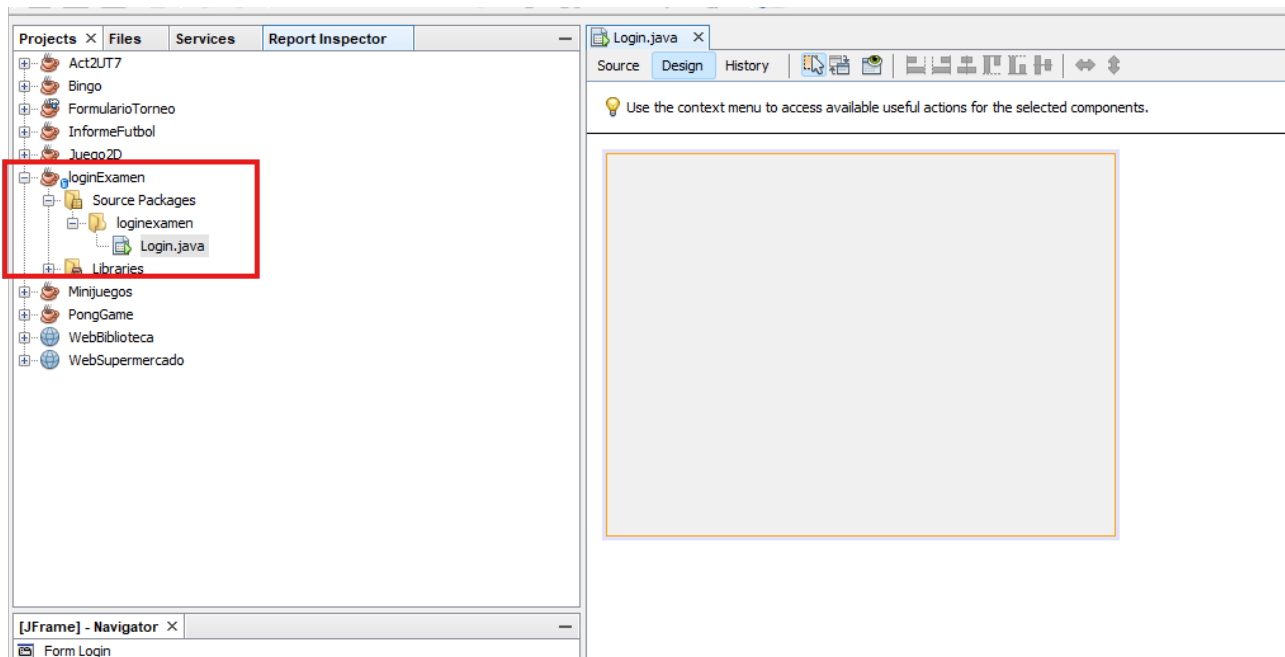
(Ahora tipo sale en rojo porque debemos añadirlo a la consulta del informe principal)
Una vez hagamos clic en finish podremos ir a la preview y consultar nuestro gráfico:



Y si establecemos la consulta para un solo digimon pues nos aparecerá únicamente el tipo de ese digimon:



3ªA continuación se nos solicita realizar una pantalla de login para el inicio de sesión de los usuarios del nuevo juego de Digimon, para esto lo primero que haremos será crear un nuevo proyecto en NetBeans y un nuevo JFrame:



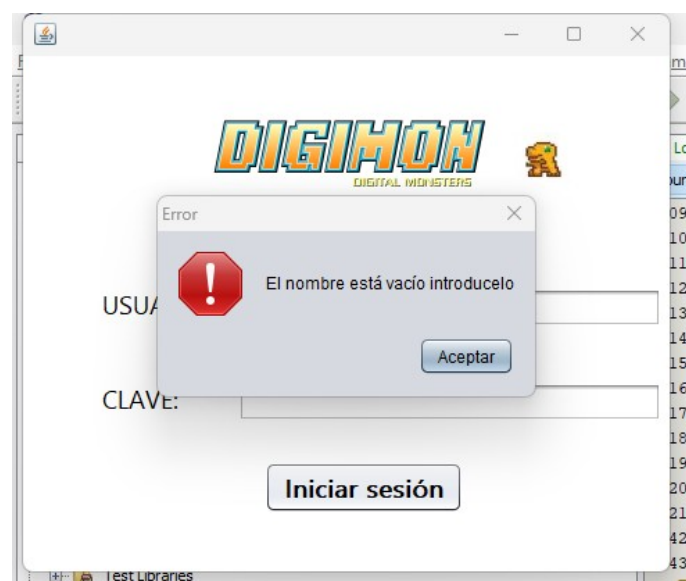
Ahora vamos a diseñar la pantalla de login:



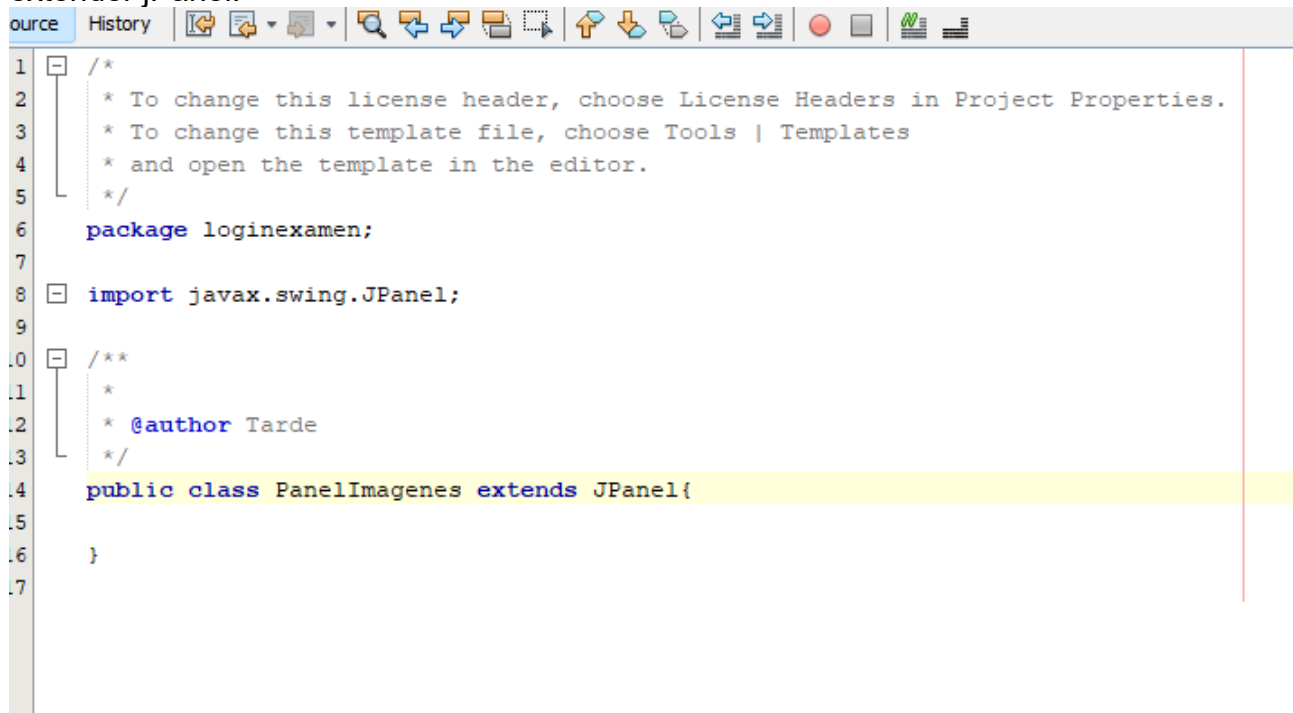
The image shows a login form for 'DIGIMON DIGITAL MONSTERS'. At the top, the 'DIGIMON' logo is in a stylized orange and blue font, with 'DIGITAL MONSTERS' in a smaller green font below it. To the right of the logo is a small icon of a Digimon character. Below the logo, there are two text input fields. The first is labeled 'USUARIO:' and the second is labeled 'CLAVE:'. Below these fields is a button labeled 'Iniciar sesión'.

Ahora vamos a introducir la función de inicio de sesión, para ello el campo de usuario no podrá estar vacío.

```
public boolean validarEntradas() {  
    if (jTextFieldUsuario.getText().equals(null) || jTextFieldUsuario.getText().equals("")) {  
        JOptionPane.showMessageDialog(this, "El nombre está vacío introducelo", "Error", JOptionPane.ERROR_MESSAGE);  
        return false;  
    }  
    return true;  
}
```



Ahora vamos a configurar la pantalla que muestre las imágenes del juego, para ello vamos a crear un nuevo JPanel donde estableceremos las imágenes del juego, para esto vamos a crear una nueva clase de java de nombre PanelImagenes en la cual vamos a extender JPanel:

A screenshot of an IDE window showing a Java file named PanelImagenes.java. The code includes a package declaration 'loginexamen;', an import 'javax.swing.JPanel;', and the start of a class 'PanelImagenes extends JPanel{'. The IDE has a toolbar at the top with various icons for editing and navigation. The code is as follows:

```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package loginexamen;
7
8  import javax.swing.JPanel;
9
10 /**
11  *
12  * @author Tarde
13  */
14 public class PanelImagenes extends JPanel{
15
16 }
17
```

En esta clase vamos a incluir lo siguiente:

```
    * @author Tarde
    */
    public class PanelImagenes extends JPanel{
        private Image backgroundImage;
        private final int WIDTH = 800, HEIGHT = 600;
```

Una imagen para representar de nombre backgroundImage y dos variables enteras para el tamaño de la nueva pestaña del JPanel, ahora vamos a implementar un constructor con los siguientes métodos:

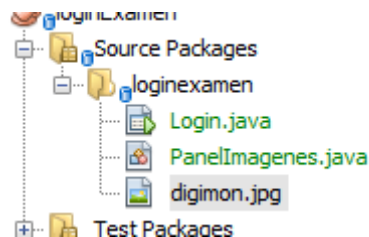
```
    public PanelImagenes() {
        setPreferredSize(new Dimension(WIDTH, HEIGHT));
        loadBackgroundImage();
    }
```

con setPreferredSize establecemos el tamaño que hemos indicado en las variables y con loadBackgroundImage cargaremos la imagen que queremos representar:

Ahora creamos el método loadBackgroundImage:

```
private void loadBackgroundImage() {
    try {
        backgroundImage = ImageIO.read(getClass().getResource("/loginexamen/digimon.jpg"));
    } catch (IOException e) {
        e.printStackTrace();
        setBackground(Color.BLACK);
    }
}
```

Deberemos tener un recurso de nombre digimon en nuestro paquete para que funcione:



Finalmente nos queda introducir el método paint component para pintar automáticamente la imagen del fondo en el jpanel:

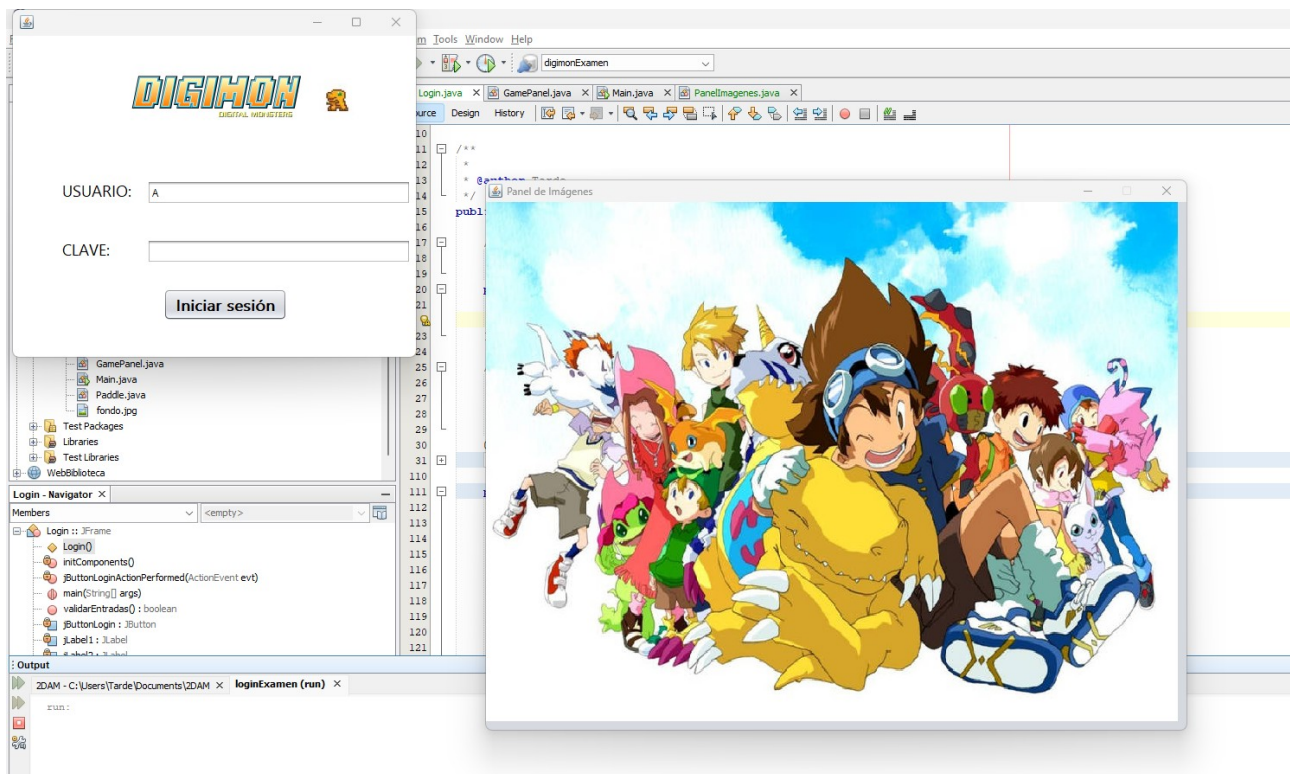
```
@Override
protected void paintComponent(Graphics g) {
    super.paintComponent(g);
    if (backgroundImage != null) {
        g.drawImage(backgroundImage, 0, 0, WIDTH, HEIGHT, this);
    }
}
```

Ahora volvemos a nuestro login y debemos de declarar una nueva instancia de PanelImagen al validar la entrada y un nuevo JFrame para representar el JPanel:

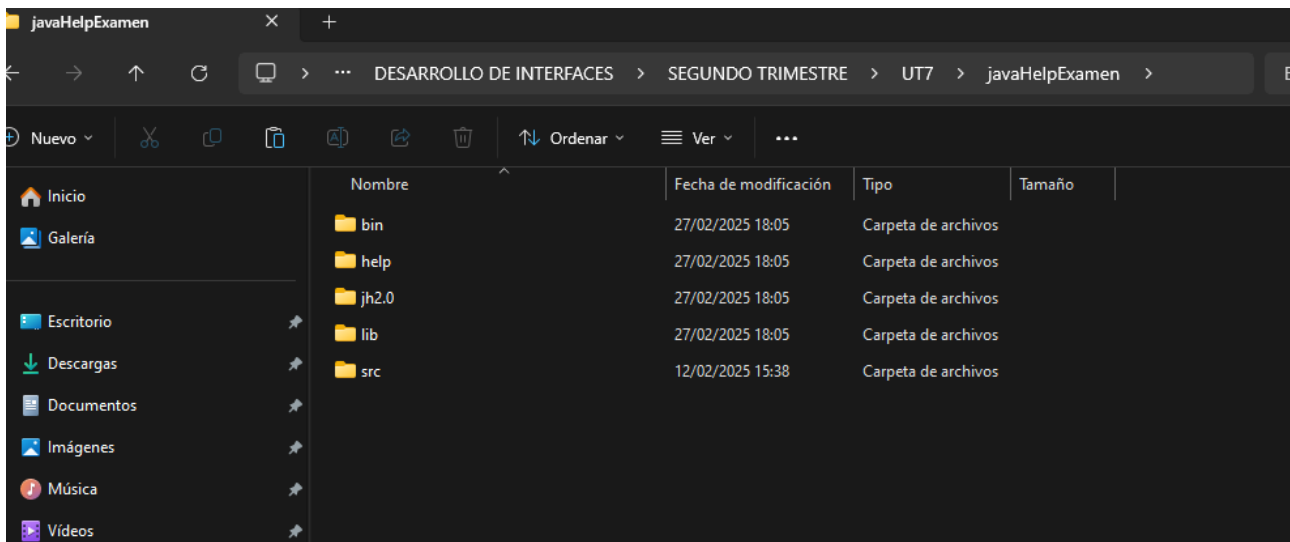
```
private void jButtonLoginActionPerformed(java.awt.event.ActionEvent evt) {
    if(validarEntradas()){
        JFrame frame = new JFrame("Panel de Imágenes");
        PanelImagenes p = new PanelImagenes();

        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setResizable(false);
        frame.add(p);
        frame.pack();
        frame.setLocationRelativeTo(null);
        frame.setVisible(true);
    }
}
```

Con estos métodos le damos un método de cerrado para que se cierre la aplicación al cerrar el panel, no permitimos que se le pueda cambiar el tamaño, lo mostramos en el centro y la hacemos visible, y se vería así:



Ahora vamos a introducir JavaHelp en el proyecto, para esto lo primero que debemos de hacer es en obtener las herramientas de JavaHelp de cualquier otro proyecto y modificar su contenido para adaptarlo a este proyecto:



Ahora en la carpeta help vamos a modificar las distintas páginas que tendrá nuestro manual de ayuda, que se encuentran en html:

html	27/02/2025 18:05	Carpeta de archivos	
JavaHelpSearch	27/02/2025 18:05	Carpeta de archivos	
help_set.hs	12/02/2025 15:38	Archivo HS	2 KB
indice.xml	12/02/2025 15:38	Archivo de origen ...	1 KB
map_file.jhm	12/02/2025 15:38	Archivo JHM	1 KB
toc.xml	12/02/2025 15:38	Archivo de origen ...	1 KB

En este caso yo voy a realizar un manual de ayuda JavaHelp con tres páginas:

html

+

<

>

↑

↺

🖥️

>

...

SEGUNDO TRIMESTRE

>

UT7

>

javaHelpExamen

>

help

>

html

Busca

+

Nuevo ▾

✂️

📄

📄

📄

🗑️

↕ Ordenar ▾

☰ Ver ▾

...

🏠 Inicio

🖼️ Galería

🖥️ Escritorio

⬇️ Descargas

📄 Documentos

🖼️ Imágenes

🎵 Música

Nombre	Fecha de modificación	Tipo	Tamaño
📄 introduccion.html	12/02/2025 15:38	Chrome HTML Do...	1 KB
📄 controles.html	12/02/2025 15:38	Chrome HTML Do...	1 KB
📄 digimon.html	12/02/2025 15:38	Chrome HTML Do...	1 KB

En la página de introducción tendremos una breve información sobre el manual de ayuda y enlaces a las otras páginas:

```

introduccion.html
controles.html

Archivo Editar Ver

<html>
<head>
  <title>Introduccion</title>
</head>
<body>
  <h1>Introduccion</h1>
  <p>¡Bienvenido a digimon PC!.</p>
  <p>En este manual de ayuda te enseñaremos los controles de juego
y como conseguir nuevos digimon.</p>

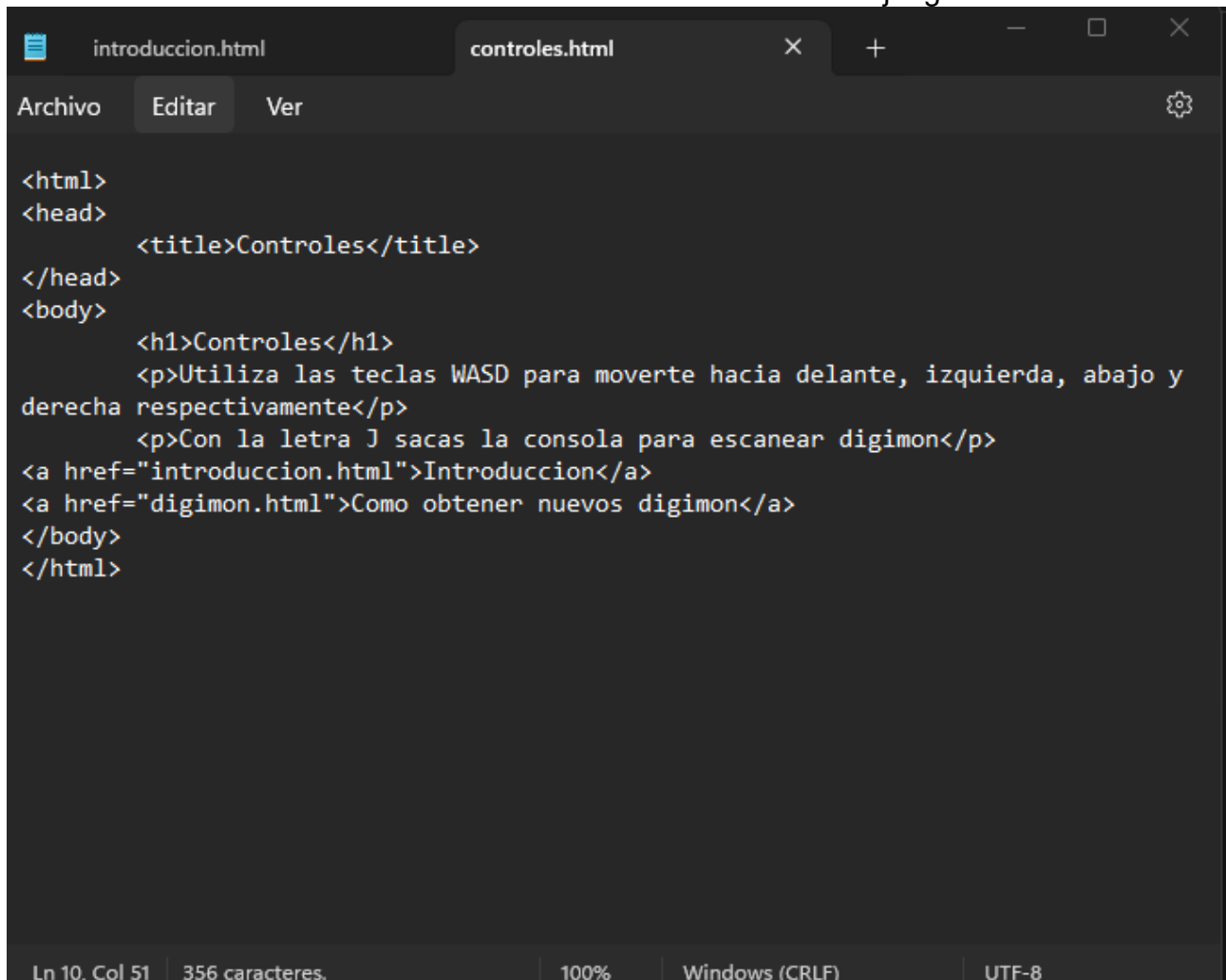
  <p>¡Adelante, la aventura te espera!</p>

  <a href="controles.html">Controles</a>
  <a href="digimon.html">Como obtener nuevos digimon</a>
</body>
</html>

Ln 14, Col 17 372 caracteres. 100% Windows (CRLF) UTF-8

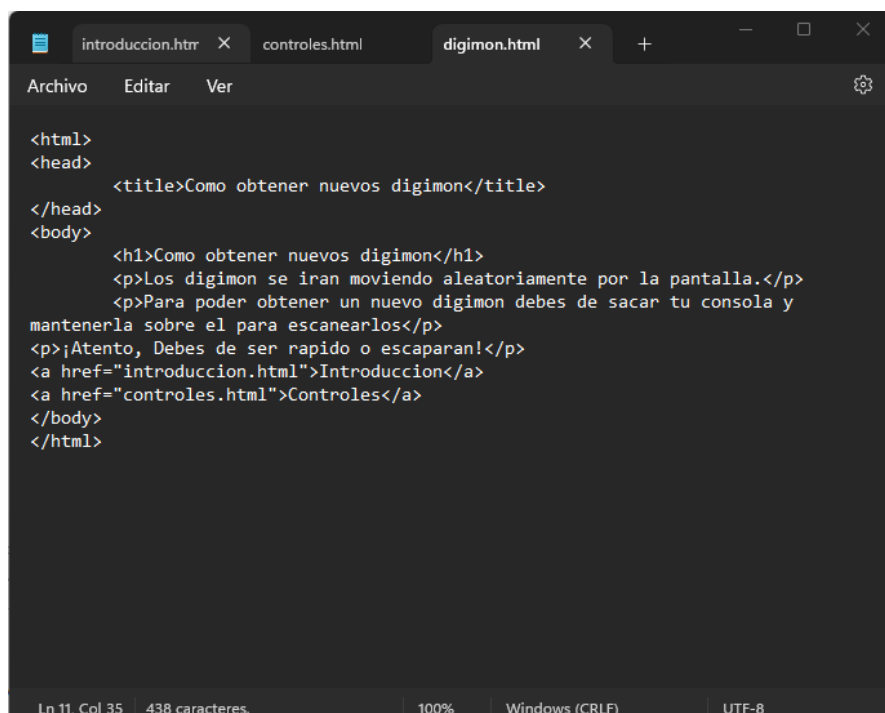
```


En controles tenemos información básica sobre los controles del juego:

A screenshot of a code editor window with two tabs: 'introduccion.html' and 'controles.html'. The 'controles.html' tab is active. The editor shows HTML code for a page titled 'Controles'. The code includes a title, a heading, two paragraphs about movement and console use, and two links to 'introduccion.html' and 'digimon.html'. The status bar at the bottom indicates 'Ln 10, Col 51', '356 caracteres.', '100%', 'Windows (CRLF)', and 'UTF-8'.

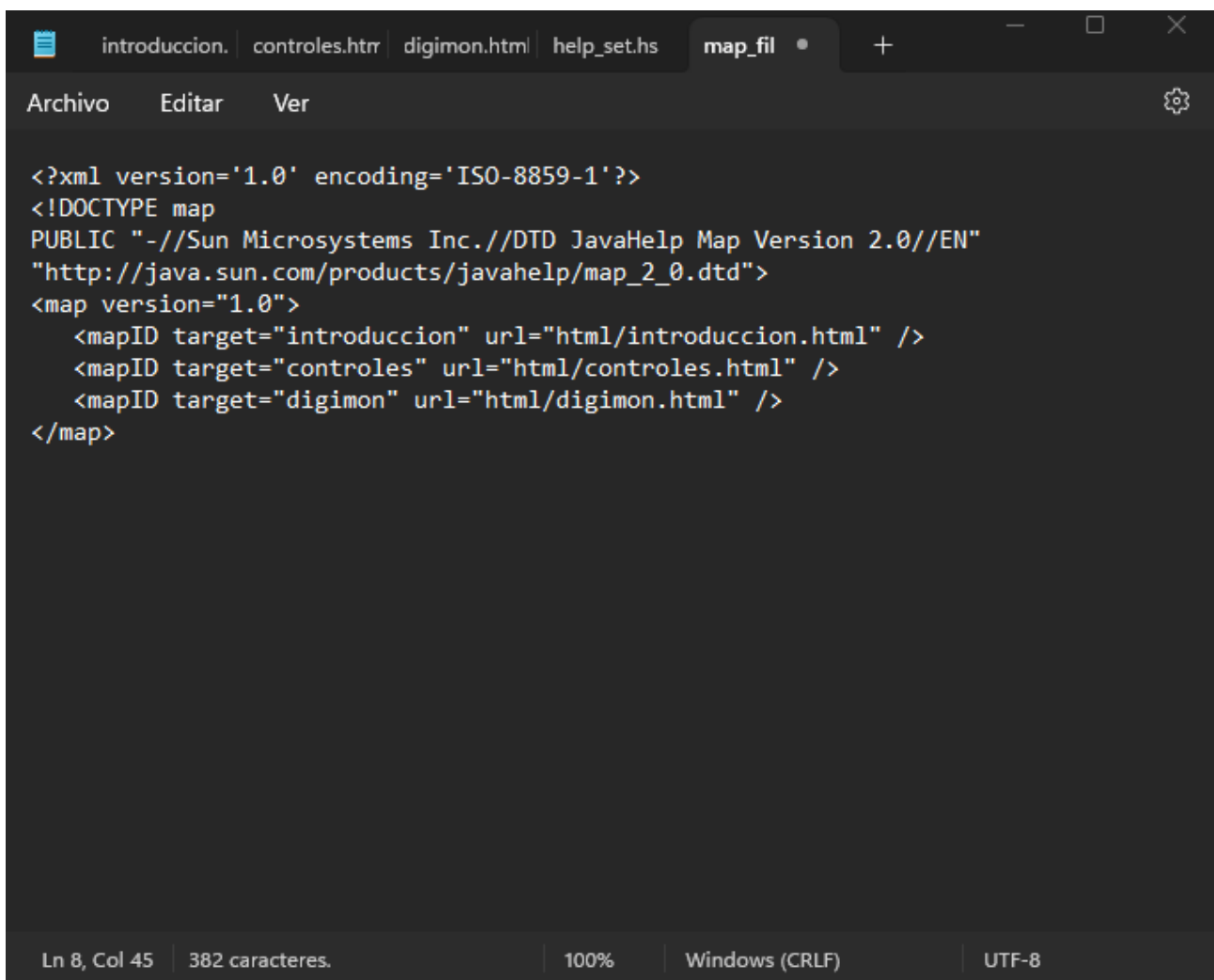
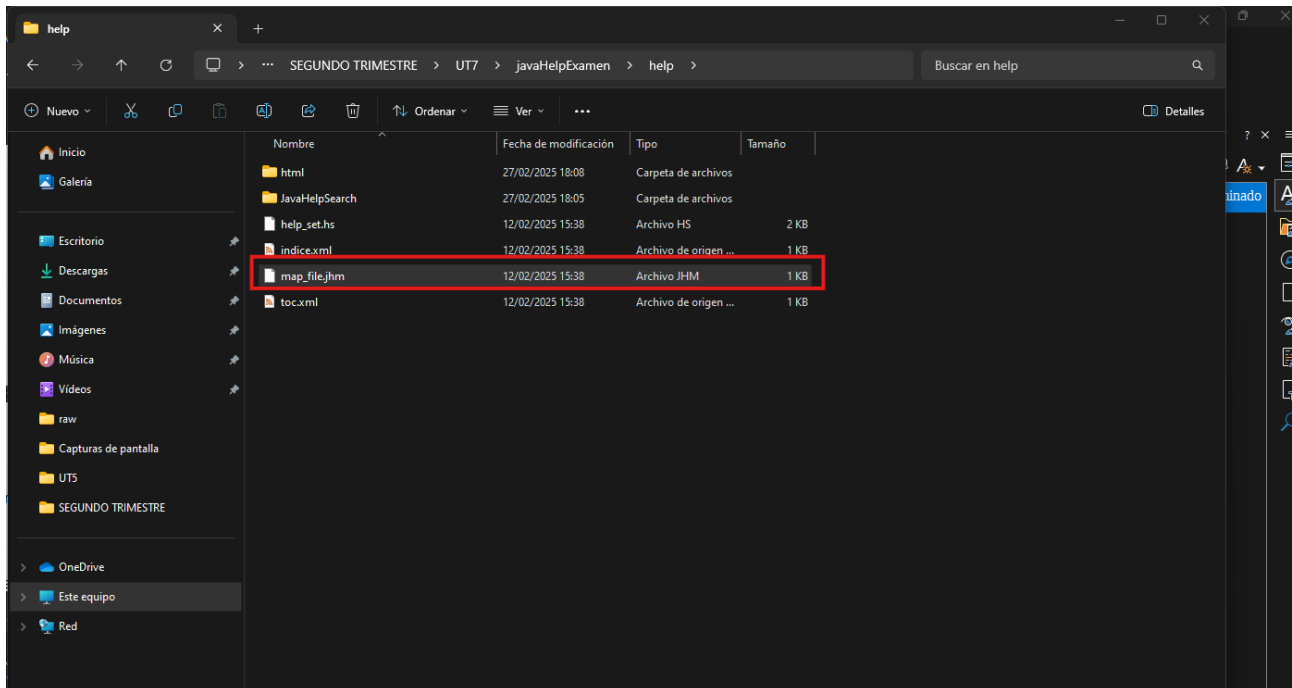
```
<html>
<head>
  <title>Controles</title>
</head>
<body>
  <h1>Controles</h1>
  <p>Utiliza las teclas WASD para moverte hacia delante, izquierda, abajo y
derecha respectivamente</p>
  <p>Con la letra J sacas la consola para escanear digimon</p>
  <a href="introduccion.html">Introduccion</a>
  <a href="digimon.html">Como obtener nuevos digimon</a>
</body>
</html>
```

Finalmente en el apartado de digimon tenemos una breve guía de como obtener nuevos digimon:

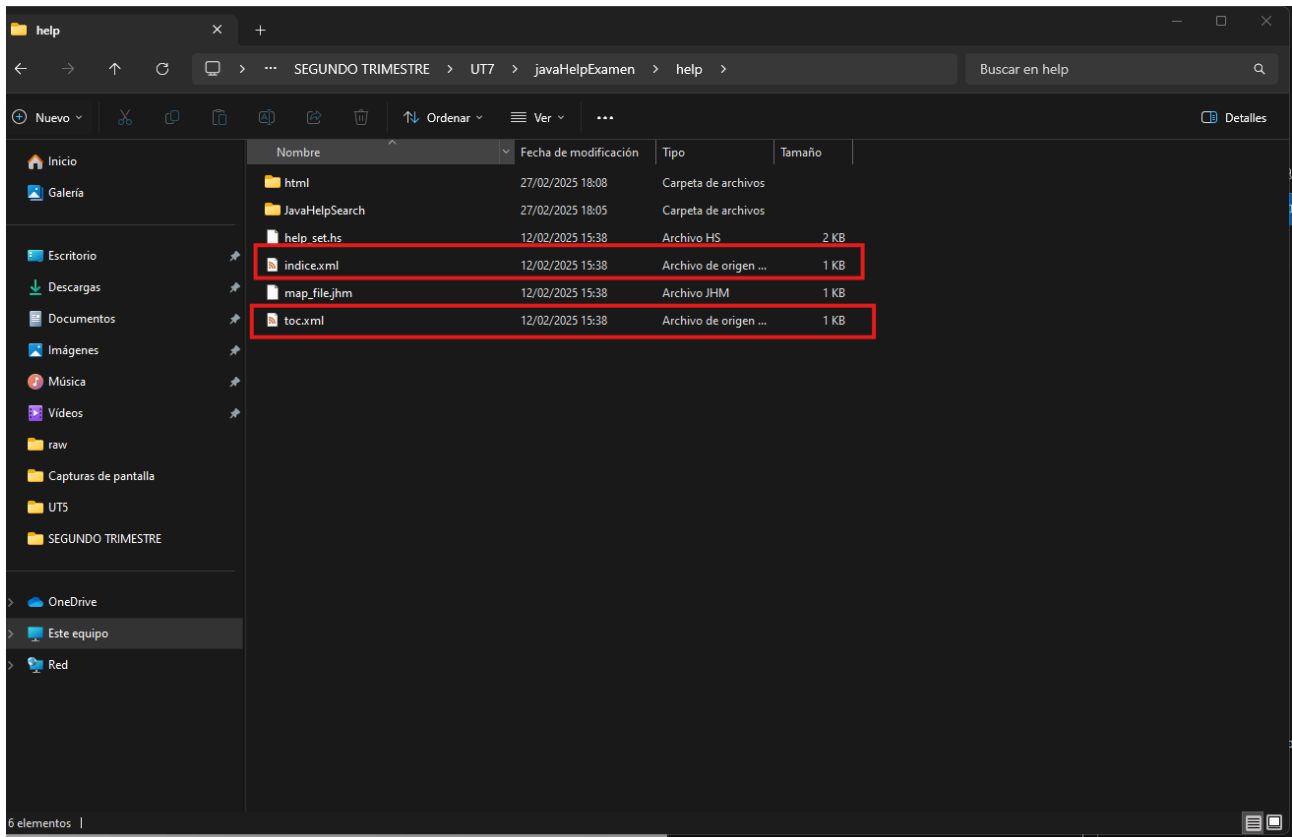
A screenshot of a code editor window with three tabs: 'introduccion.html', 'controles.html', and 'digimon.html'. The 'digimon.html' tab is active. The editor shows HTML code for a page titled 'Como obtener nuevos digimon'. The code includes a title, a heading, three paragraphs about random movement, console use, and speed, and two links to 'introduccion.html' and 'controles.html'. The status bar at the bottom indicates 'Ln 11, Col 35', '438 caracteres.', '100%', 'Windows (CRLF)', and 'UTF-8'.

```
<html>
<head>
  <title>Como obtener nuevos digimon</title>
</head>
<body>
  <h1>Como obtener nuevos digimon</h1>
  <p>Los digimon se iran moviendo aleatoriamente por la pantalla.</p>
  <p>Para poder obtener un nuevo digimon debes de sacar tu consola y
mantenerla sobre el para escanearlos</p>
  <p>¡Atento, Debes de ser rapido o escaparan!</p>
  <a href="introduccion.html">Introduccion</a>
  <a href="controles.html">Controles</a>
</body>
</html>
```

Una vez terminado de editar las diferentes páginas debemos de configurar el mapa, para ello modificaremos el archivo:



Una vez configurado el mapa, configuraremos el índice y la tabla de contenidos para que se representen correctamente, para ello vamos a abrir y editar los archivos:



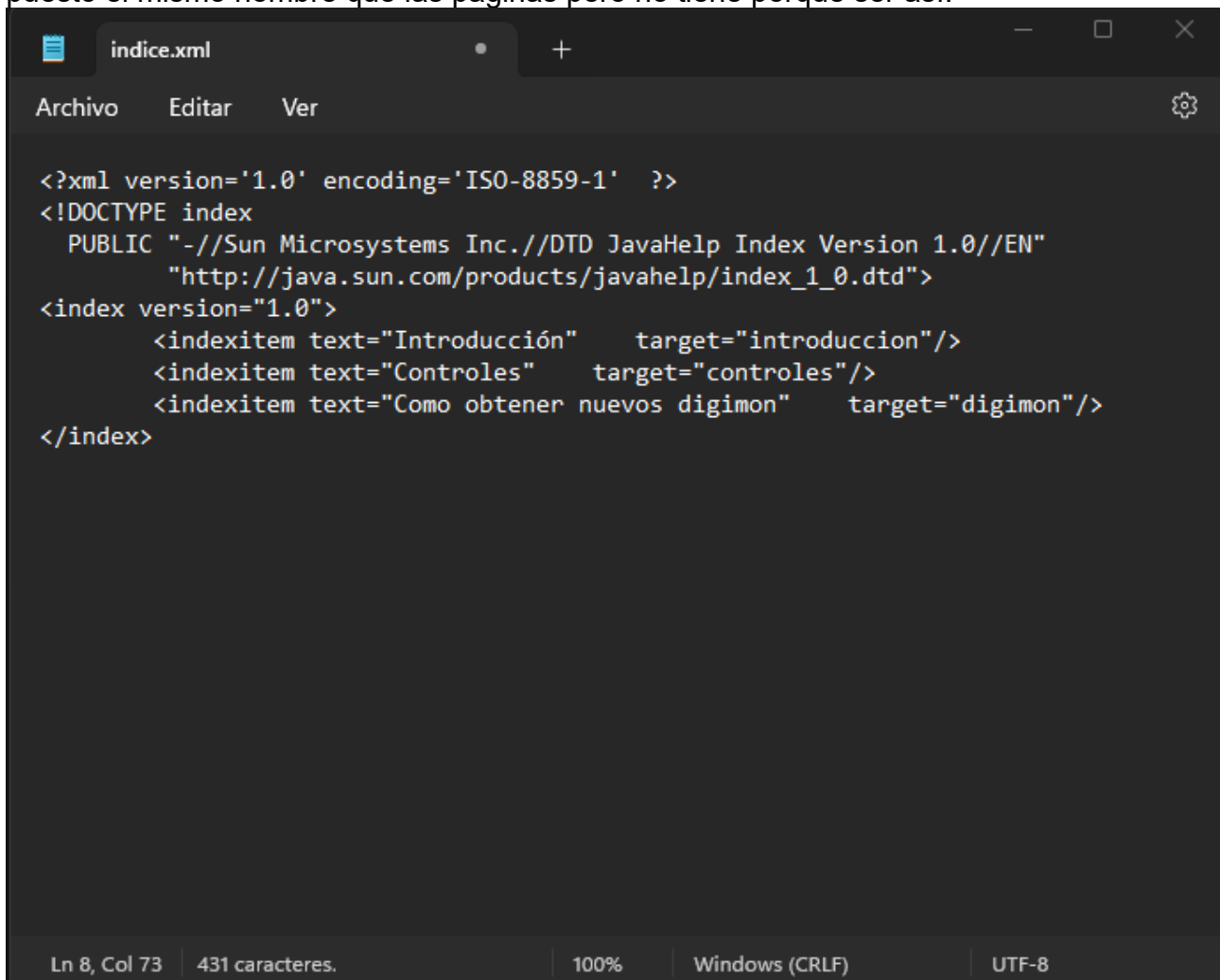
Ahora vemos por ejemplo en indice tenemos distintos elementos de nombre indexitem que hacen referencia a “aplicación”, “principal” y “secundaria”:

```
indice.xml
Archivo  Editar  Ver

<?xml version='1.0' encoding='ISO-8859-1' ?>
<!DOCTYPE index
  PUBLIC "-//Sun Microsystems Inc./DTD JavaHelp Index Version 1.0//EN"
    "http://java.sun.com/products/javahelp/index_1_0.dtd">
<index version="1.0">
  <indexitem text="Ejemplo JavaHelp"    target="aplicacion"/>
  <indexitem text="Ventana principal"   target="ventana_principal"/>
  <indexitem text="Ventana secundaria"  target="ventana_secundaria"/>
</index>

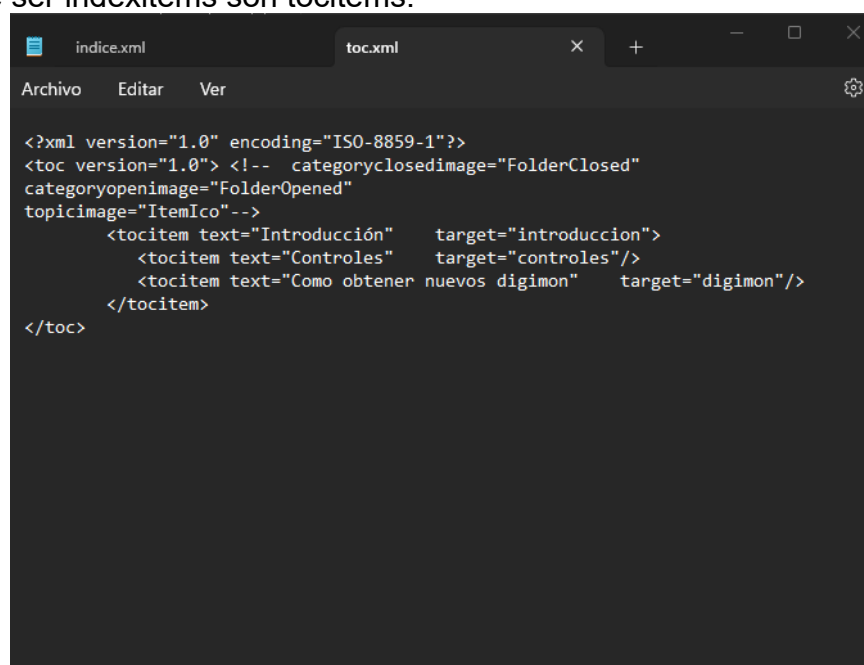
Ln 1, Col 1    451 caracteres.    100%    Windows (CRLF)    UTF-8
```

Estos eran los nombres que tenían antes nuestras pestañas en el mapa, entonces debemos de cambiar esos nombres por los que hemos establecido en el mapa, yo he puesto el mismo nombre que las páginas pero no tiene porque ser así.

A screenshot of a code editor window with a dark theme. The title bar shows 'indice.xml'. The menu bar has 'Archivo', 'Editar', and 'Ver'. The code is XML for an index. It starts with an XML declaration and a DOCTYPE declaration for 'index'. The root element is <index>, which contains three <indexitem> elements. Each item has a 'text' attribute and a 'target' attribute. The status bar at the bottom shows 'Ln 8, Col 73', '431 caracteres.', '100%', 'Windows (CRLF)', and 'UTF-8'.

```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<!DOCTYPE index
  PUBLIC "-//Sun Microsystems Inc.//DTD JavaHelp Index Version 1.0//EN"
  "http://java.sun.com/products/javahelp/index_1_0.dtd">
<index version="1.0">
  <indexitem text="Introducción" target="introduccion"/>
  <indexitem text="Controles" target="controles"/>
  <indexitem text="Como obtener nuevos digimon" target="digimon"/>
</index>
```

Y ahora hacemos lo mismo con el toc que veremos que es prácticamente lo mismo, solo que en vez de ser indexitems son tocitems:

A screenshot of a code editor window with a dark theme. The title bar shows 'indice.xml' and 'toc.xml'. The menu bar has 'Archivo', 'Editar', and 'Ver'. The code is XML for a table of contents. It starts with an XML declaration and a <toc> element. The <toc> element has several attributes and contains three <tocitem> elements. Each item has a 'text' attribute and a 'target' attribute. The status bar at the bottom is not visible in this screenshot.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<toc version="1.0"> <!-- categoryclosedimage="FolderClosed"
categoryopenimage="FolderOpened"
topicimage="ItemIco"-->
  <tocitem text="Introducción" target="introduccion">
    <tocitem text="Controles" target="controles"/>
    <tocitem text="Como obtener nuevos digimon" target="digimon"/>
  </tocitem>
</toc>
```

A continuación vamos a editar el archivo:

html	27/02/2025 18:08	Carpeta de archivos	
JavaHelpSearch	27/02/2025 18:05	Carpeta de archivos	
help_set.hs	12/02/2025 15:38	Archivo HS	2 KB
indice.xml	27/02/2025 18:25	Archivo de origen ...	1 KB
map_file.jhm	27/02/2025 18:22	Archivo JHM	1 KB
toc.xml	27/02/2025 18:26	Archivo de origen ...	1 KB

Que en este caso realmente no necesitamos cambiar nada, pero por ejemplo si nuestro documento indice se llamase de otra manera lo especificaríamos aquí:

```
<?xml version="1.0" encoding='ISO-8859-1' ?>
<!DOCTYPE helpset
  PUBLIC "-//Sun Microsystems Inc.//DTD JavaHelp HelpSet Version 1.0//EN"
    "http://java.sun.com/products/javahelp/helpset_1_0.dtd">

<helpset version="1.0">
  <title>Ejemplo ayuda JavaHelp</title>
  <maps>
    <!-- Pagina por defecto al mostrar la ayuda -->
    <homeID>aplicacion</homeID>
    <!-- Que mapa deseamos -->
    <mapref location="map_file.jhm"/>
  </maps>

  <!-- Las Vistas que deseamos mostrar en la ayuda -->
  <view>
    <name>Tabla Contenidos</name>
    <label>Tabla de contenidos</label>
    <type>javax.help.TOCView</type>
    <data>toc.xml</data>
  </view>

  <view>
    <name>Indice</name>
    <label>El indice</label>
    <type>javax.help.IndexView</type>
    <data>indice.xml</data>
  </view>

  <view>
    <name>Buscar</name>
    <label>Buscar</label>
    <type>javax.help.SearchView</type>
    <data engine="com.sun.java.help.search.DefaultSearchEngine">
      JavaHelpSearch
    </data>
  </view>
</helpset>
```

Finalmente vamos a establecer el JavaHelpSearch para que el usuario pueda filtrar por palabras clave, para ello lo primero que debemos hacer es borrar esta carpeta:

html	27/02/2025 18:08	Carpeta de archivos	
JavaHelpSearch	27/02/2025 18:05	Carpeta de archivos	
help_set.hs	12/02/2025 15:38	Archivo HS	2 KB
indice.xml	27/02/2025 18:25	Archivo de origen ...	1 KB
map_file.jhm	27/02/2025 18:22	Archivo JHM	1 KB
toc.xml	27/02/2025 18:26	Archivo de origen ...	1 KB

Una vez borrada abrimos una nueva consola de comandos en esta carpeta (lo podemos hacer escribiendo cmd en la ruta):

Y ahora debemos de introducir el siguiente comando para ejecutar la aplicación jhindexer que se encuentra en la carpeta de javahelp:

`java -jar "C:\Users\Tarde\Documents\2DAM\DESARROLLO DE INTERFACES\SEGUNDO TRIMESTRE\UT7\javaHelp\jh2.0\javaHelp\bin\jhindexer.jar" "html"` y se nos generará una nueva carpeta JavaHelpSearch actualizada y funcional:

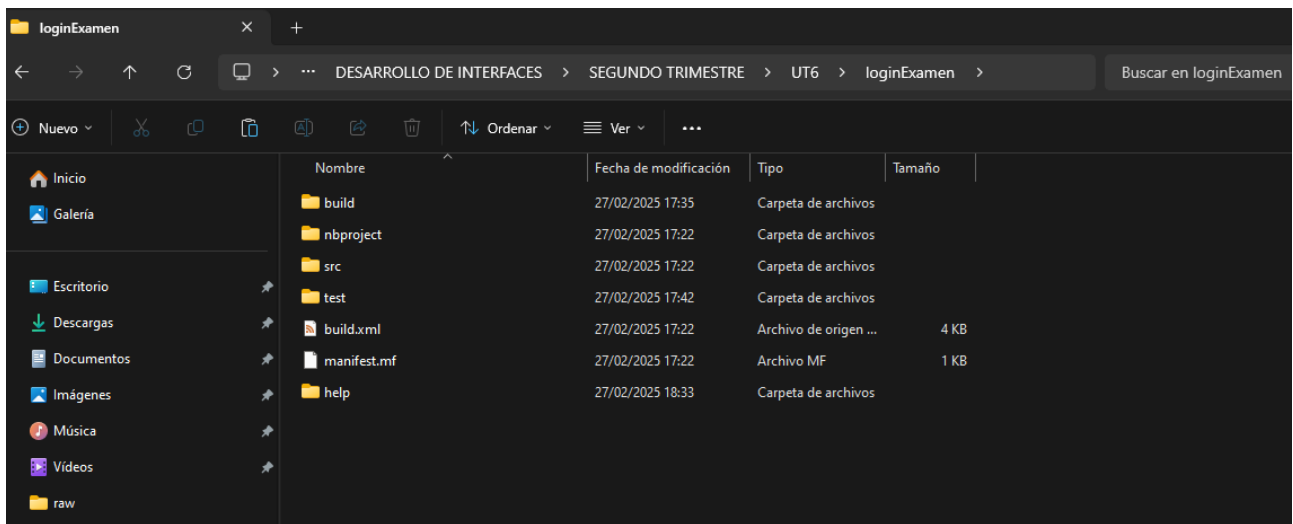
```
C:\Windows\System32\cmd.e  X  +  v
Microsoft Windows [Versión 10.0.22631.4890]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Tarde\Documents\2DAM\DESARROLLO DE INTERFACES\SEGUNDO TRIMESTRE\UT7\javaHelpExamen\help>java -jar "C:\Users\Tarde\Documents\2DAM\DESARROLLO DE INTERFACES\SEGUNDO TRIMESTRE\UT7\javaHelp\jh2.0\javaHelp\bin\jhindexer.jar" "html"

C:\Users\Tarde\Documents\2DAM\DESARROLLO DE INTERFACES\SEGUNDO TRIMESTRE\UT7\javaHelpExamen\help>
```

html	12/02/2025 15:38	Carpeta de archivos	
JavaHelpSearch	12/02/2025 15:38	Carpeta de archivos	
help_set.hs	12/02/2025 15:38	Archivo HS	2 KB
indice.xml	12/02/2025 15:38	Archivo de origen ...	1 KB
map_file.jhm	12/02/2025 15:38	Archivo JHM	1 KB
toc.xml	12/02/2025 15:38	Archivo de origen ...	1 KB

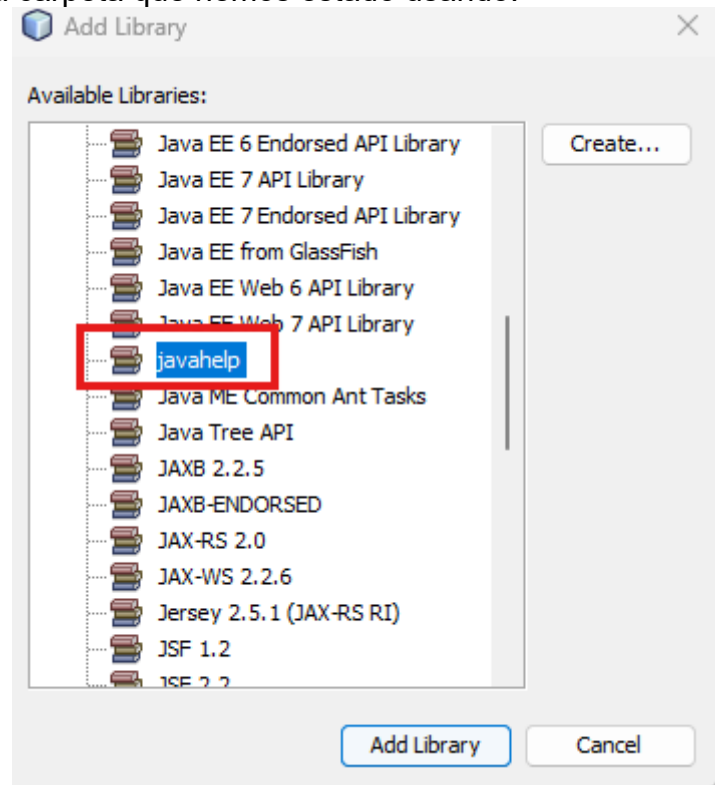
Ahora vamos a implementarlo en nuestro proyecto, para ello lo primero que tenemos que hacer es copiar la carpeta de help a la ruta del proyecto base:



Ahora vamos de nuevo al proyecto para añadir el menú de ayuda y añadimos una menú bar y un menú ítem y le cambiamos el nombre para que nos quede así:



Ahora a nuestro proyecto debemos de importarle las librerías de JavaHelp, que también se encuentran en la carpeta que hemos estado usando:



Una vez añadidas las librerías utilizaremos el siguiente método para vincular la ayuda de JavaHelp al botón del menú que hemos declarado:

```
private void ponLaAyuda() {  
    try{  
        File fichero = new File("help"+File.separator+"help_set.hs");  
        URL hsURL = fichero.toURI().toURL();  
        HelpSet helpset = new HelpSet(getClass().getClassLoader(), hsURL);  
        HelpBroker hb = helpset.createHelpBroker();  
        hb.enableHelpOnButton(jMenuItemAyuda, "introduccion", helpset);  
        hb.enableHelpKey(getRootPane(), "introduccion", helpset);  
    }catch(Exception e){  
        e.printStackTrace();  
    }  
}
```

Así vinculamos la JavaHelp con los botones que hemos creado. Ahora simplemente llamamos a este método aquí:

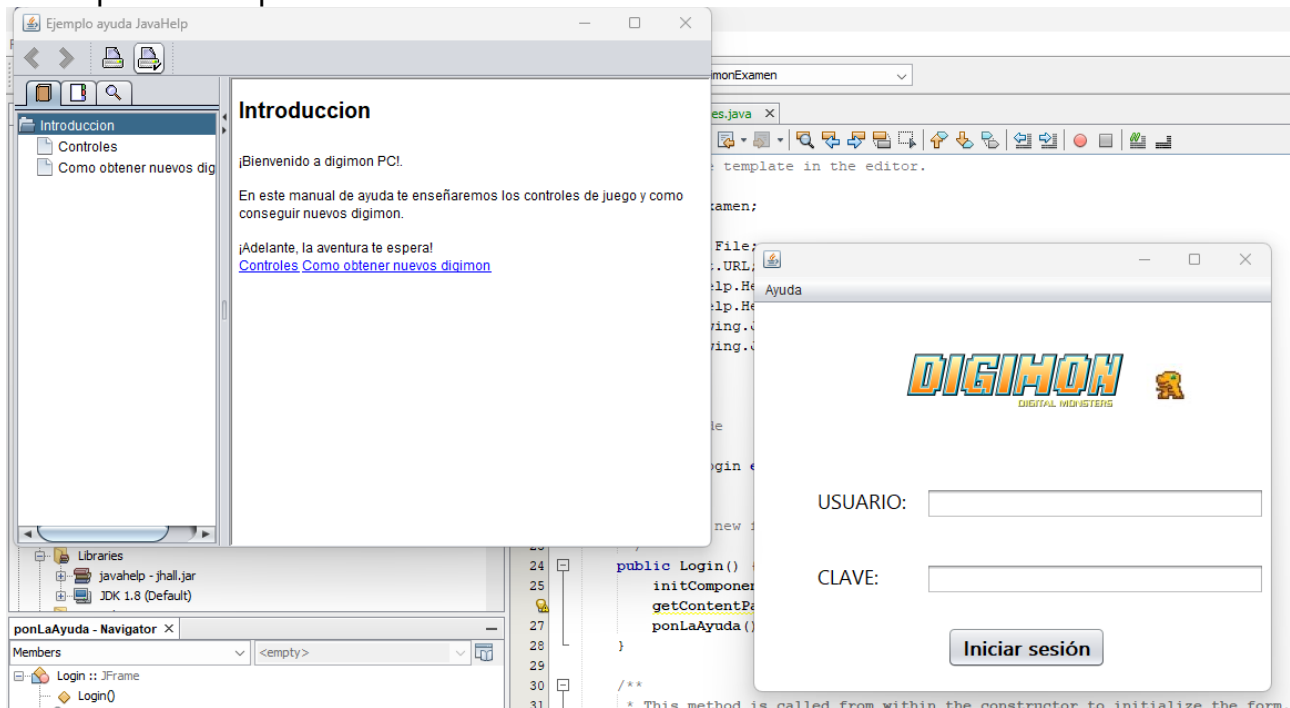

```

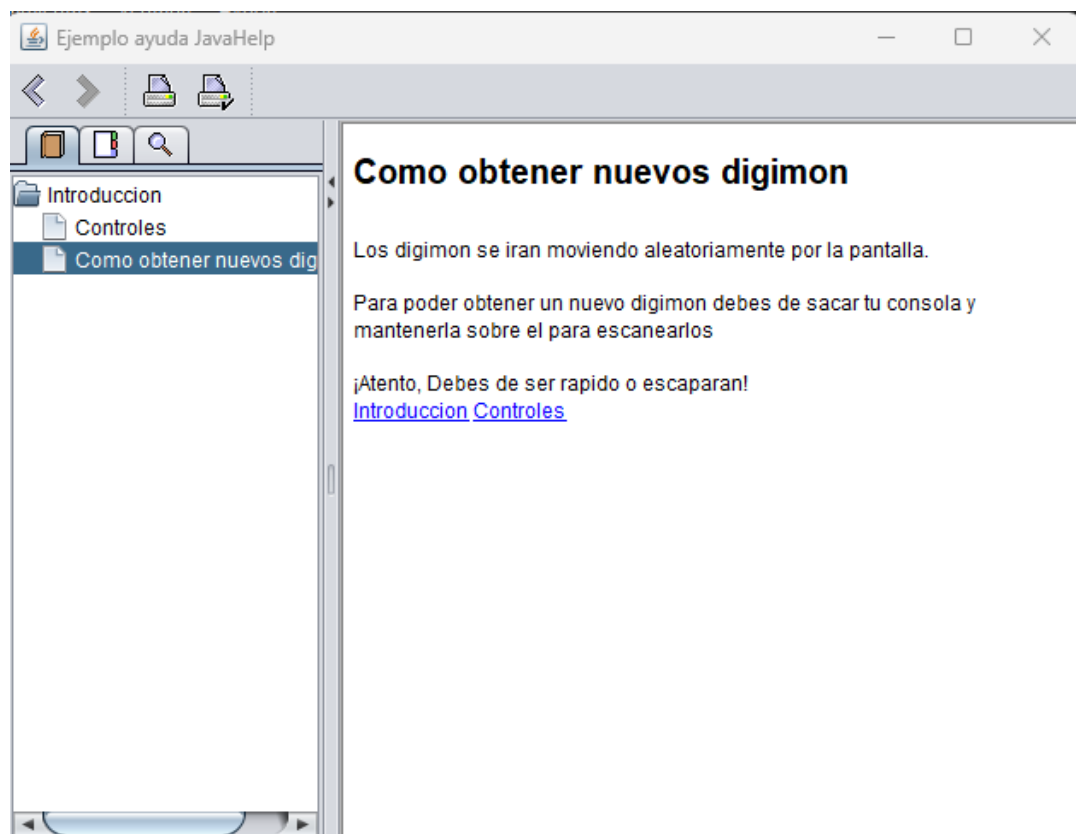
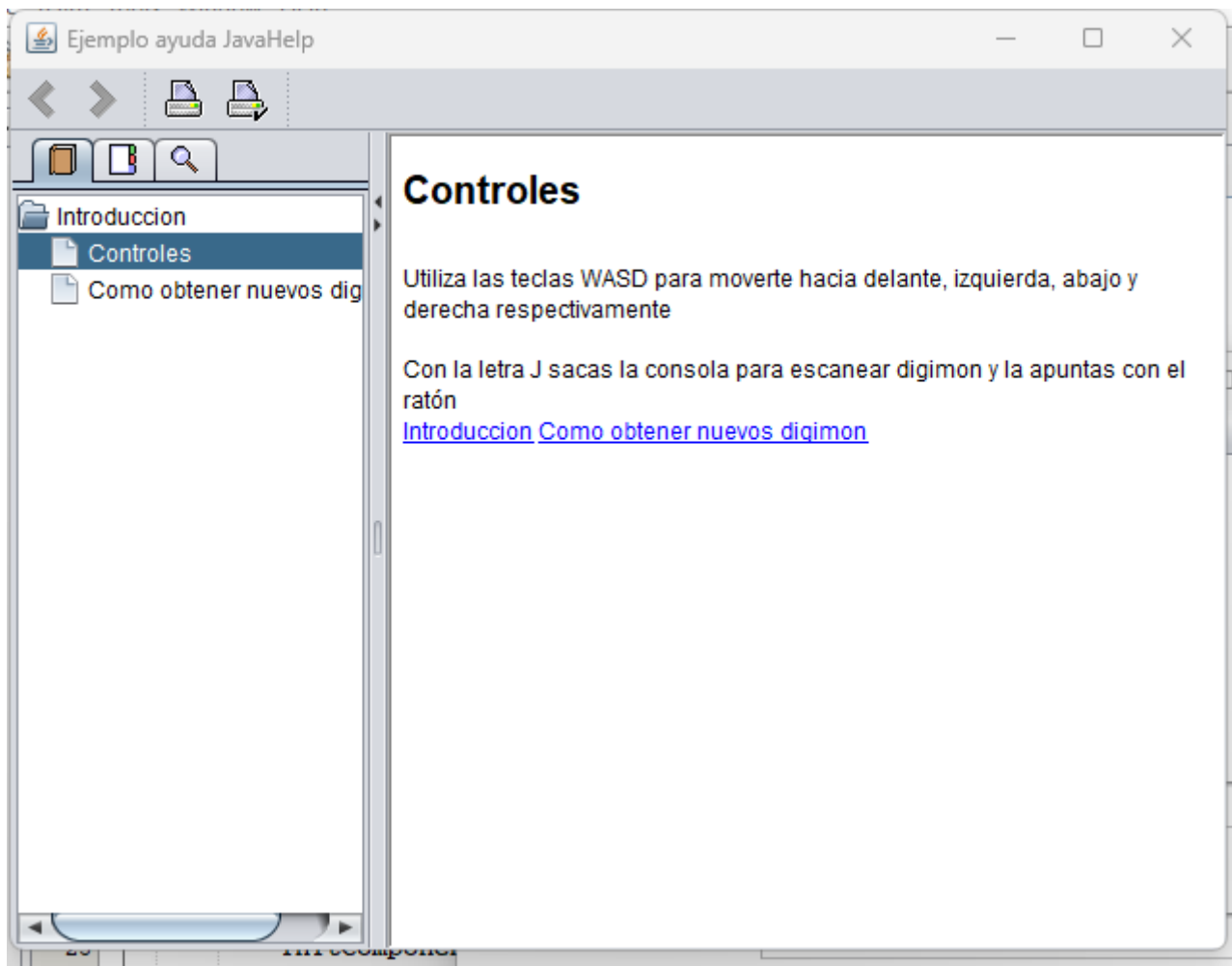
L  */
  public class Login extends javax.swing.JFrame {

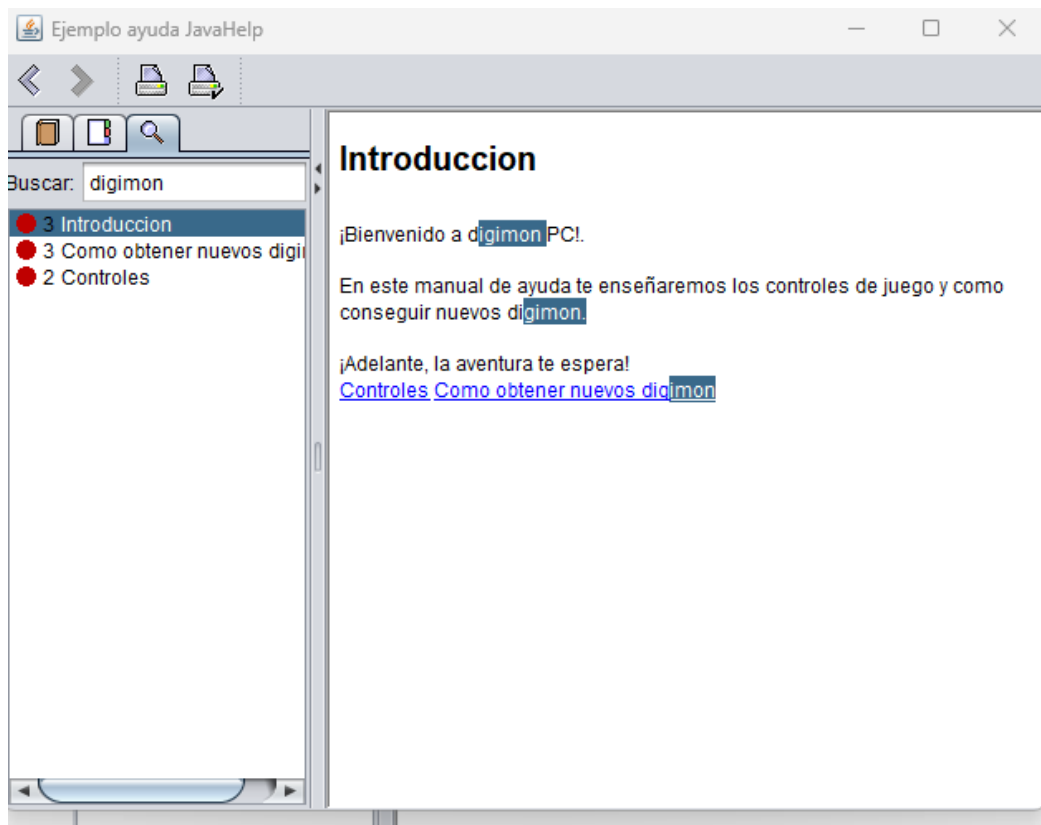
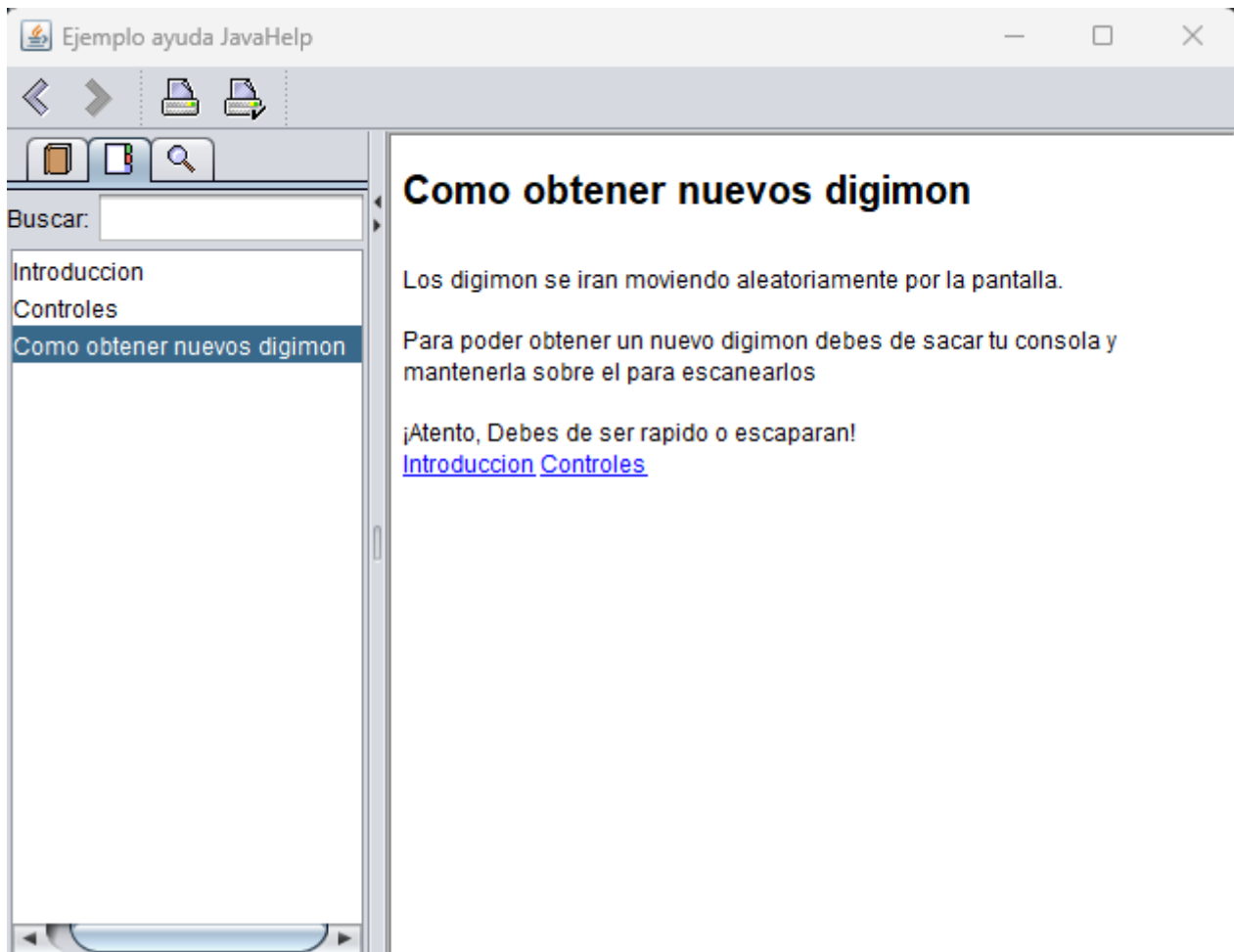
      /**
       * Creates new form Login
       */
      public Login() {
          initComponents();
          getContentPane().setBackground(new java.awt.Color(255, 255, 255));
          ponLaAyuda();
      }
  }

```

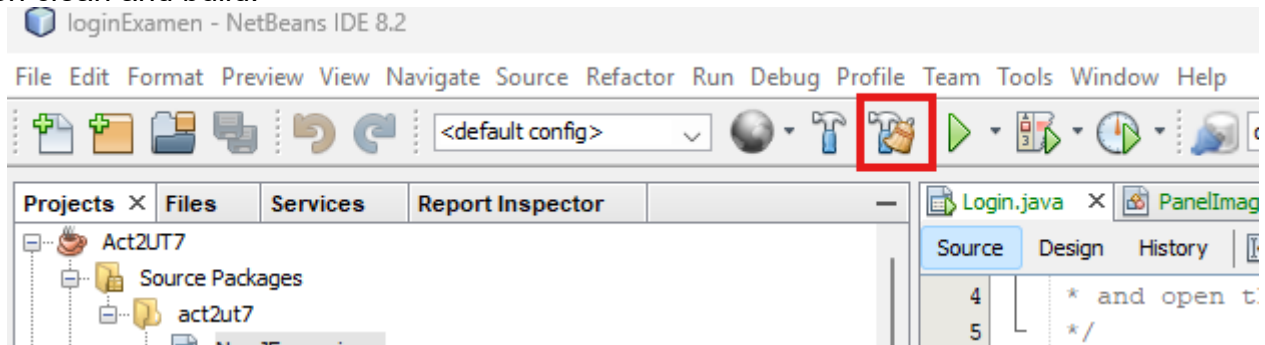
Y comprobemos que funciona:







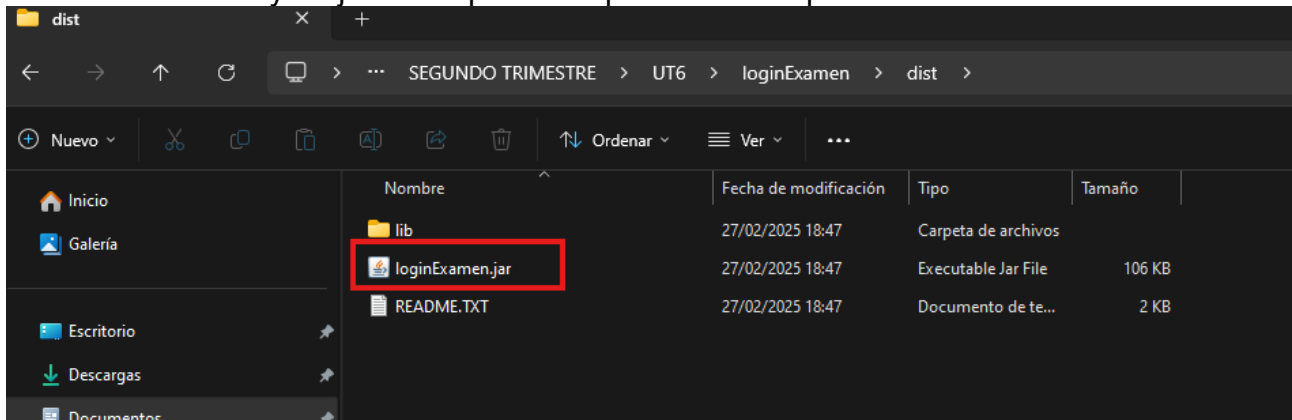
4º Vamos a construir el jar de nuestro proyecto, para esto simplemente vamos a hacer clic en clean and build:



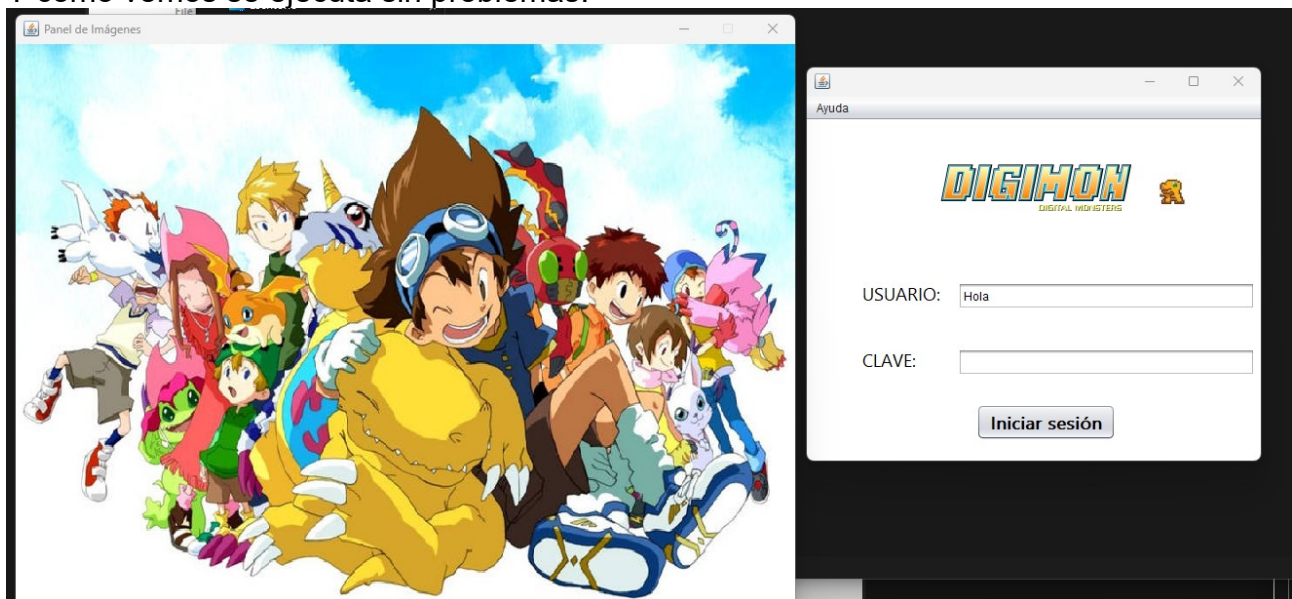
Aquí podemos ver que nuestro jar se ha generado sin problemas en la siguiente dirección:

```
to run this application from the command line without Ant, try:
java -jar "C:\Users\Tarde\Documents\2DAM\DESARROLLO DE INTERFACES\SEGUNDO TRIMESTRE\UT6\loginExamen\dist\loginExamen.jar"
jar:
BUILD SUCCESSFUL (total time: 0 seconds)
```

Vamos a buscarlo y a ejecutarlo para ver que no nos da problemas:



Y como vemos se ejecuta sin problemas:



5º Vamos a desarrollar una aplicación que realizará un tipo test a sus usuarios sobre preguntas generales de Digimon, para asegurarnos de que la aplicación se desarrolla correctamente debemos de elaborar un manual de pruebas detallando las estrategias de diseño que sigue nuestra aplicación, empezando por las fases necesarias para un proyecto así:

- Planificación: Debe ser la primera fase que realicemos, aquí planificaremos con que tecnología vamos a desarrollar la aplicación, cuales serán los mejores métodos para alcanzar los objetivos establecidos tanto por el cliente como por el equipo de diseño, se definirán plazos, herramientas, roles y en general la estructura y organización que debe de tener tanto el equipo como la aplicación en sí.
 - Diseño: Una vez finalice la planificación entraremos en la fase de diseño, donde el principal objetivo es hacer un “boceto” de la aplicación, haremos uso de mockups para definir como queremos que se vea la interfaz de usuario y wireframes para establecer como queremos que funcione la lógica de nuestra aplicación, todo desde un punto de vista de diseño, teniendo en cuenta las implementaciones que requiere la aplicación facilitadas por el cliente.
 - Implementación: Es la fase en la que se empiezan a diseñar los distintos elementos de la aplicación, durante esta fase lo más importante al trabajar en un equipo es asegurarse de que los diferentes miembros comuniquen claramente en que área están trabajando para no pisarse entre ellos, usualmente utilizando herramientas como Git y GitHub para que pueda trabajar todo el equipo a la vez.
 - Evaluación: Es principalmente la fase de pruebas, donde se realizan distintos escenarios y situaciones para poner a prueba la aplicación y comprobar que funciona como se esperaba.
 - Producción: Finalmente cuando se han terminado todas las fases comienza la fase de producción donde la aplicación será presentada a usuarios finales antes de ser presentada al cliente a través por ejemplo de pruebas de fase beta y una vez el cliente aprueba de la aplicación será publicada.
-
- Fase de pruebas: Dentro de la evaluación hay que hacer un inciso y es que es necesario destacar las pruebas que queremos que siga nuestro proyecto y es efectivo hacerlas en un orden para agilizar los tiempos de entrega, lo ideal sería comenzar por las pruebas unitarias para comprobar que los módulos de manera separada funcionan correctamente, una vez finalizadas seguiríamos con las pruebas de integración para comprobar que los módulos responden bien entre sí, seguido de las pruebas de sistema para comprobar que la aplicación hace lo que se supone que debería de hacer, posteriormente deberemos de realizar las pruebas de regresión en caso de haber cambiado algún código que al hacerlo de un nuevo fallo, una vez corregidos, seguiremos con las pruebas funcionales que se encargan de evaluar la aplicación para que cumpla unos estándares como por ejemplo el ISO 25010 que se encarga de que las aplicaciones cumplan con el estándar de idoneidad, exactitud, interoperabilidad y seguridad, seguiríamos con las pruebas de capacidad donde comprobaríamos la capacidad de manejo de volumen de datos de la aplicación seguido de las pruebas de recursos para ver que la aplicación se ejecuta eficientemente y ante situaciones de estrés no se ve superada, seguiríamos con las pruebas de seguridad comprobando que no se filtre información ni haya puertas de seguridad abiertas, continuaríamos con las pruebas de usuario en las que a un grupo cerrado se les ordenaría realizar una serie de pruebas que están relacionadas con el ámbito de uso real de la aplicación, interesa que el grupo sea grande y variado para poder evaluar diferentes perfiles y

diferentes casos aunque tengan la misma instrucción de uso, concluiríamos con las pruebas de aceptación donde haríamos uso de pruebas como la fase beta para que la aplicación sea probada por usuarios reales antes de lanzarla por completo.

Podríamos decir como estimado que la duración de cada prueba sería la siguiente:

- Unitarias 15%
- Integración 10%
- Sistema 10%
- Regresión 5%
- Funcionales 10%
- Capacidad 5%
- Recursos 10%
- Seguridad 15%
- Usuario 10%
- Aceptación 10%

Cada porcentaje del tiempo total.