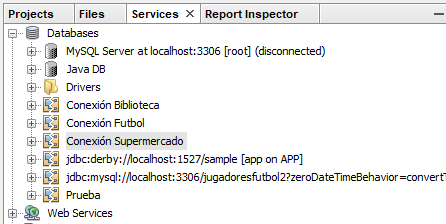
Alejandro Sánchez Gil

EXAMEN 2º TRIMESTRE

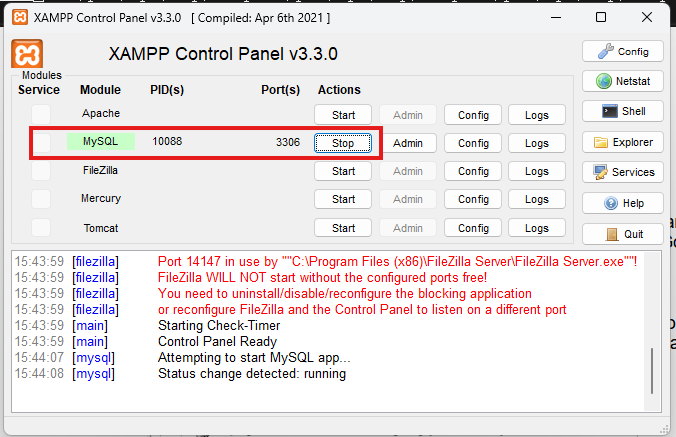
DESARROLLO DE INTERFACES

1ºLo 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>

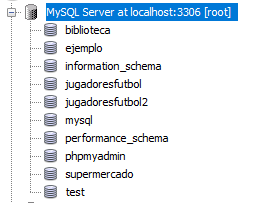
A continuación lo primero que debemos hacer será crear una nueva base de datos en Netbeans y una conexión para conectarnos a ella, para ello accedemos a la pestaña de “Services” y hacemos clic en “Databases”:



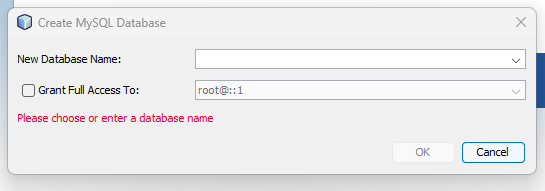
Como vemos nos indica que el servidor de MySQL está desconectado, vamos a conectarlo, primero iniciaremos el XAMPP para iniciar MySQL haciendo clic en start:



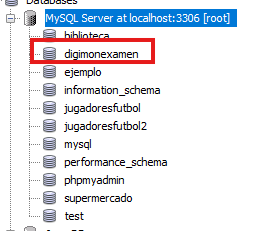
Ahora volvemos a NetBeans y hacemos clic derecho sobre MySQL Server y hacemos clic en start para iniciar el servidor de la base de datos:



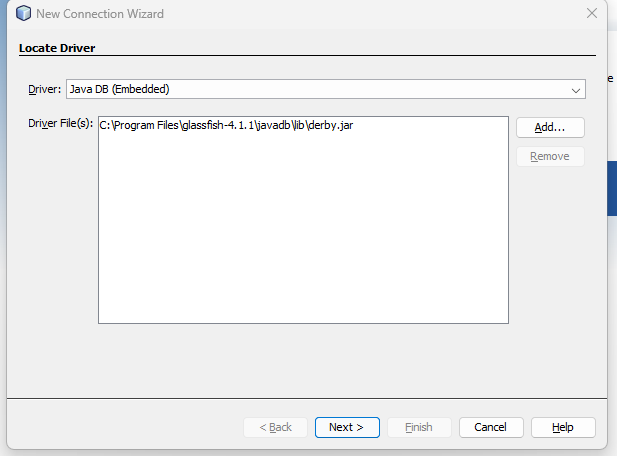
Ahora ya podremos crear una nueva base de datos, simplemente hacemos de nuevo clic derecho sobre MySQL Server y en create Database:



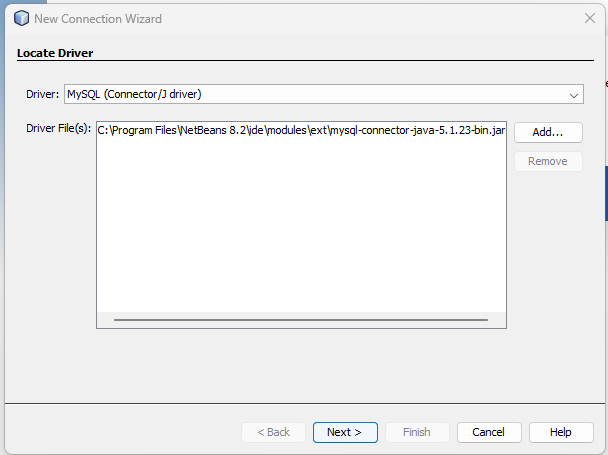
Aquí indicamos el nombre, yo la voy a llamar digimonExamen y hacemos clic en ok:



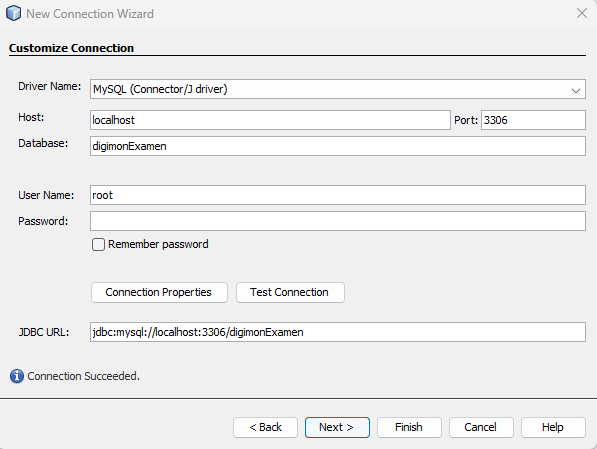
Ahora vamos a realizar la estructura de la base de datos, para ello primero vamos a crear una conexión, hacemos clic derecho en Databases y en New Connection:

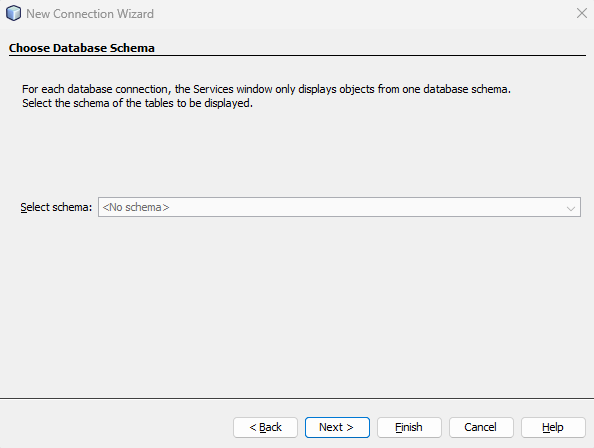


Aquí estableceremos el driver a MySQL y hacemos clic en Next:

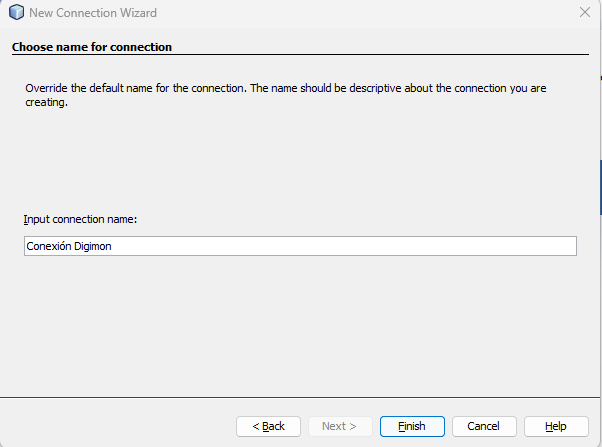


Aquí nos aseguraremos de que está en el puerto que estemos usando para el servidor de MySQL en mi caso es el 3306 que podemos ver en xampp y establecemos a la base de datos que nos queremos conectar:



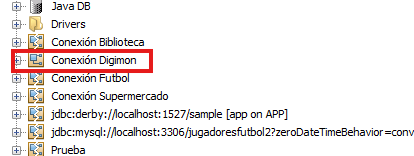
Dejaremos la contraseña vacía y el usuario como root que es lo que viene por defecto, y en la URL he eliminado lo que viene a partir del nombre de la base de datos ya que no es necesario, ahora hacemos clic en Next:

Aquí lo dejamos como está y hacemos clic en Next:



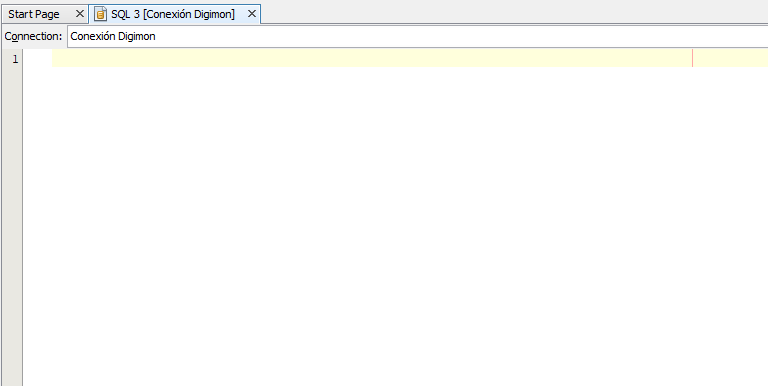
Aquí simplemente ponemos el nombre que queramos que tenga nuestro examen yo he puesto Conexión Digimon y hacemos clic en Finish.

Y aquí tendríamos la conexión recién creada:

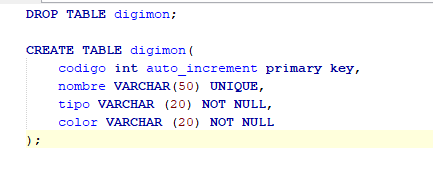


Ahora vamos a abrir una hoja de trabajo para introducir las consultas SQL, para ello hacemos clic derecho en la conexión y en Execute Command:

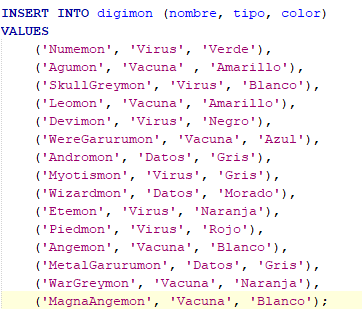
Se nos abrirá esta pestaña para realizar nuestra base de datos:



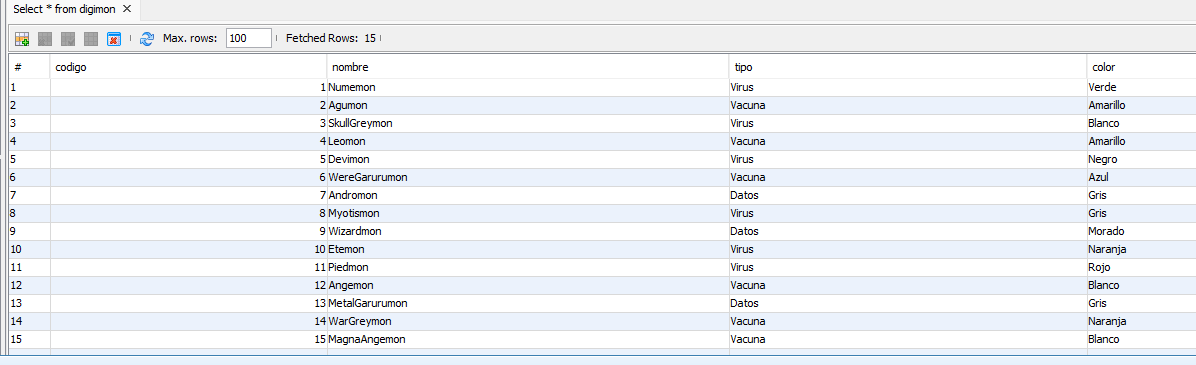
Ahora simplemente vamos a estructurar la base de datos, vamos a crear una tabla con los digimon, para esto pues hacemos una consulta SQL de Create Table:



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



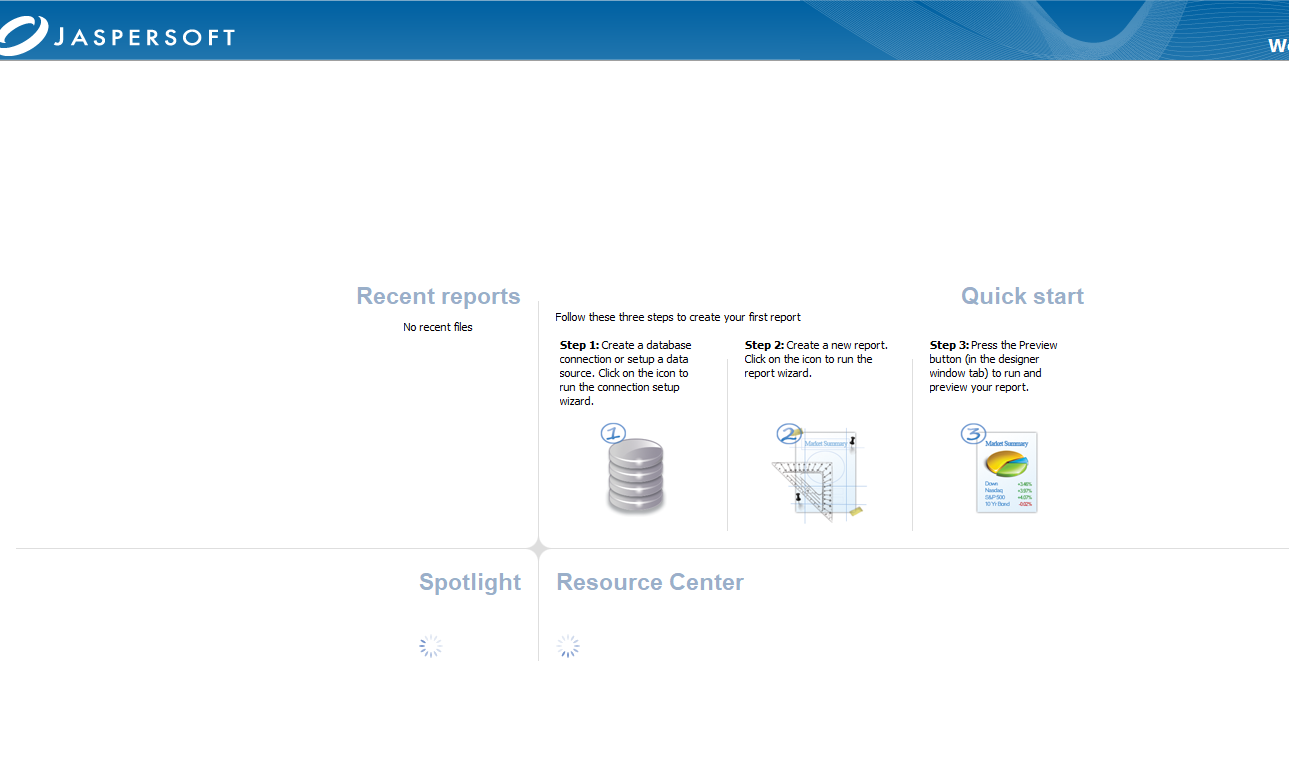
Ahora para consultarlos podemos hacer una consulta básica de tipo Select \* from digimon:



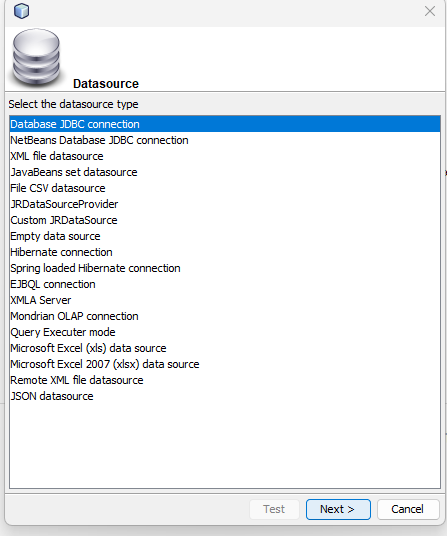
Y aquí vemos como nos aparecen todos.

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:

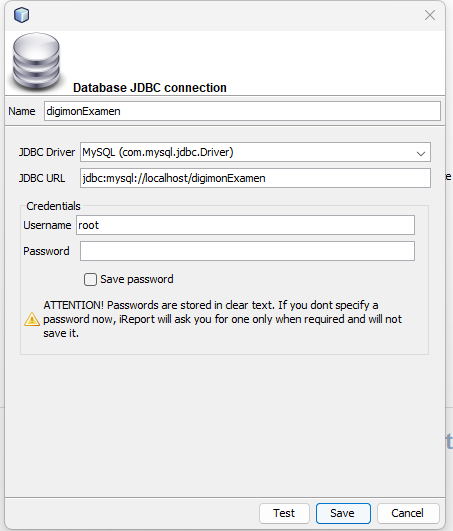


Ahora lo primero que tenemos que hacer es hacer clic donde dice Create Database para vincular la base de datos que acabamos de crear con JasperReports:

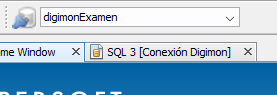


Aquí lo dejamos como está y hacemos clic en Next.

Ahora configuraremos los parámetros para la conexión:

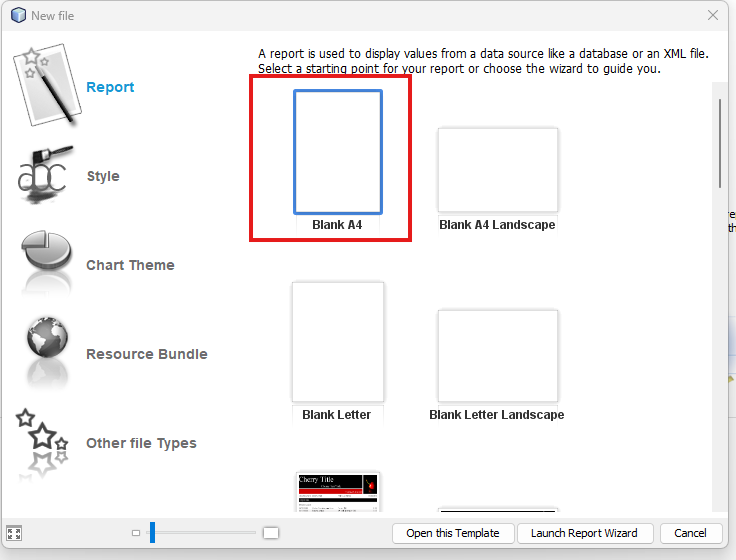


Es muy importante que establezcamos correctamente el nombre de la base de datos en el apartado JDBC URL y que seleccionemos el driver MySQL correcto que son los que tengo puestos, también hay que introducir el usuario y contraseña que en este caso es solo root, hacemos clic en Test para comprobar que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

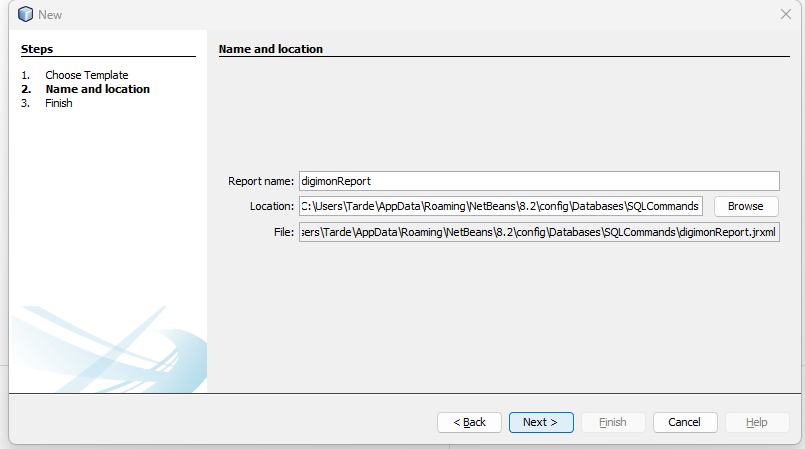


Ahora nos aparece la conexión para realizar las consultas desde jasperReports.

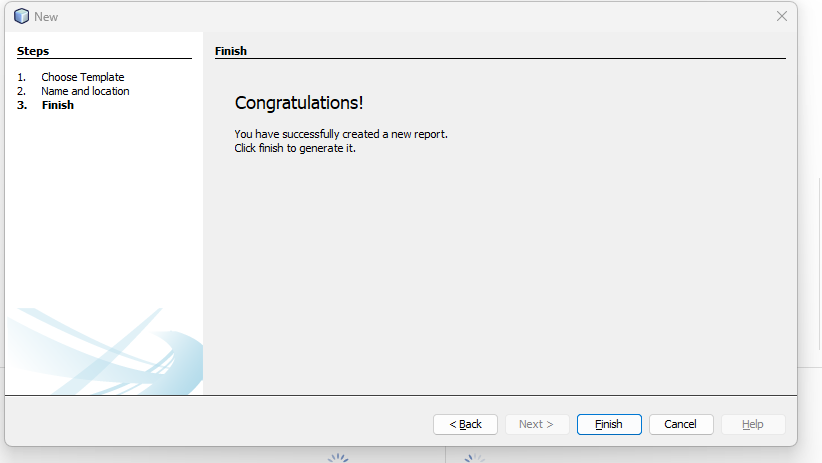
Lo siguiente que debemos hacer ahora es clic en el step 2, para empezar con la creación del informe, lo primero que se nos solicita es en que formato queremos nuestro informe, vamos a seleccionar en A4 y hacemos clic en Open This Template:

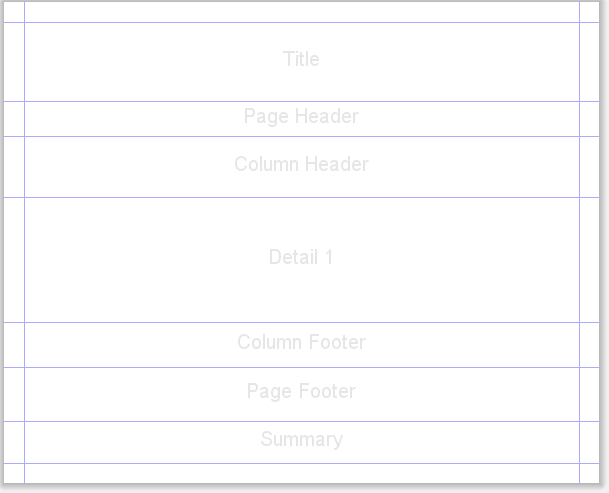


Ahora establecemos un nombre para nuestro report y donde queremos que se guarde, vamos a dejarlo por defecto y hacemos clic en Next:



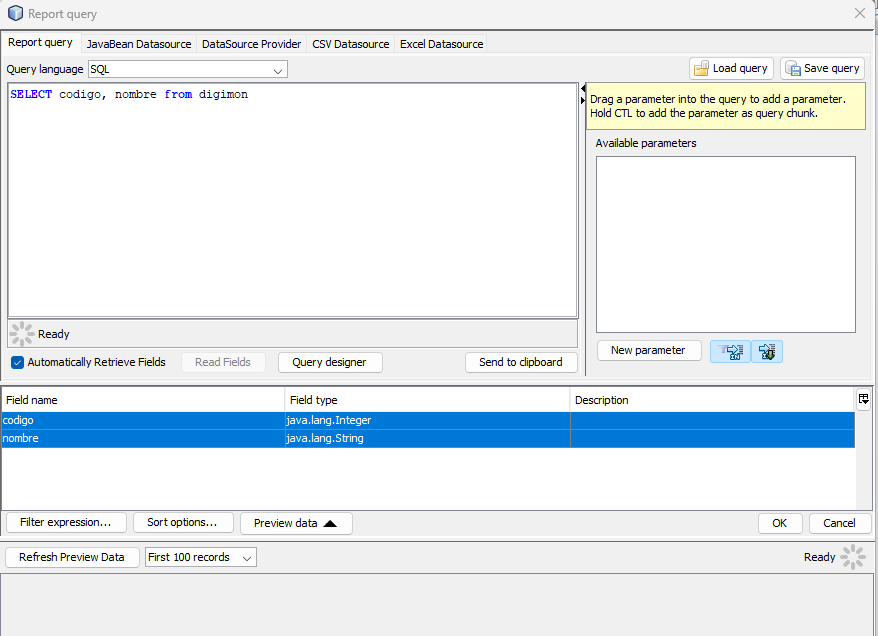
Con esto habremos creado el reporte, hacemos clic en Finish y vemos el reporte vacío para que lo empezemos a llenar:



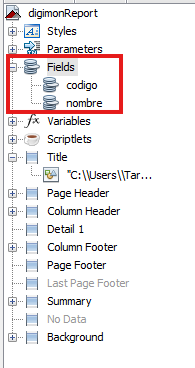
Ahora vamos a rellenar este informe con información y un subinforme que muestre el tipo y el color de cada digimon y lo decoramos un poco, primero tenemos que establecer la consulta de la que queremos que sacar los datos, para ello haremos clic en este botón:



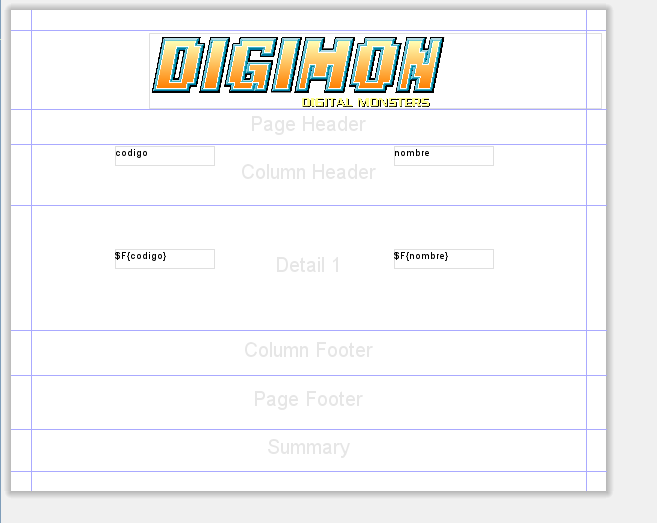
Aquí estableceremos la consulta para obtener los campos a representar en nuestro informe:



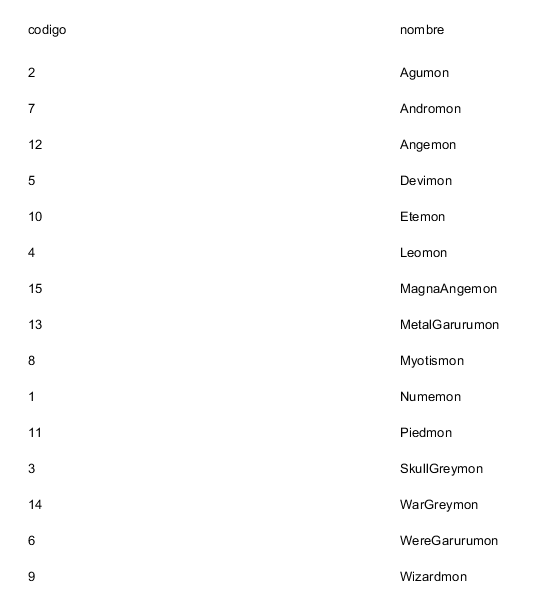
Aquí como podemos ver hemos recuperado los campos nombre y código de cada digimon, ahora nos aparecerán estos campos en el apartado de la izquierda:



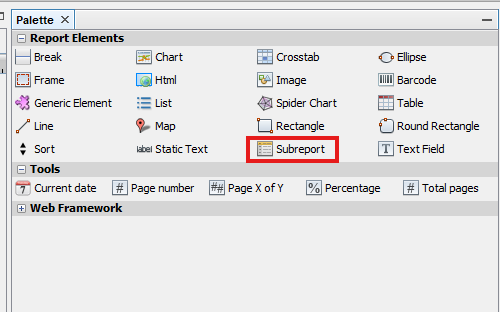
Para representar estos datos en el informe simplemente los arrastramos hasta Details:



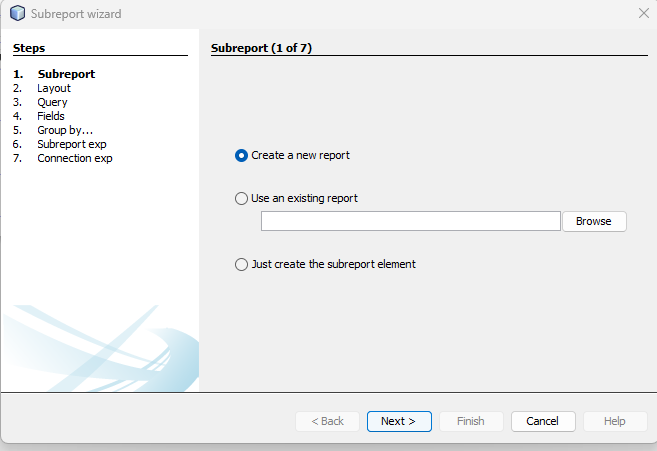
Y si ahora hacemos clic en preview veríamos el informe con la información básica:

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

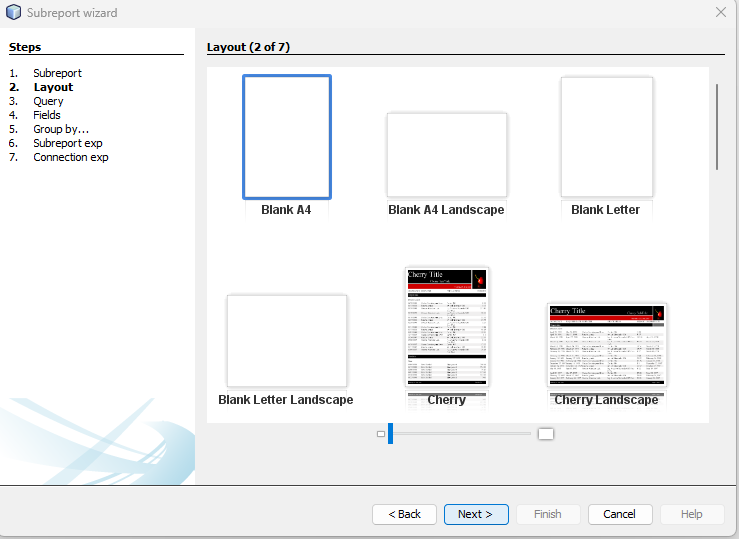
Ahora vamos a crear un subinforme que nos de los datos del tipo y del color de cada digimon, para esto vamos a añadir a nuestro informe un elemento subinforme:



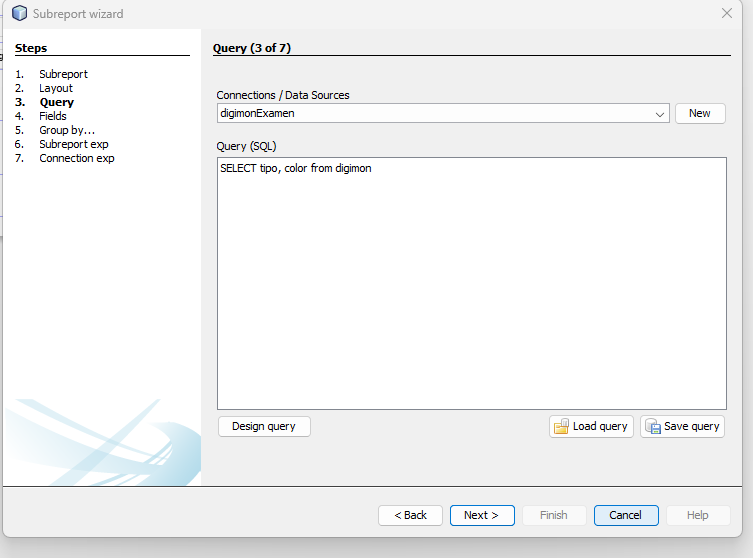
El subinforme debemos introducirlo en el apartado de Page Header para evitar problemas de colocación de los informes, una vez arrastremos el subinforme se nos abrirá una pestaña para seleccionar el informe que queremos introducir, vamos a crear uno nuevo:



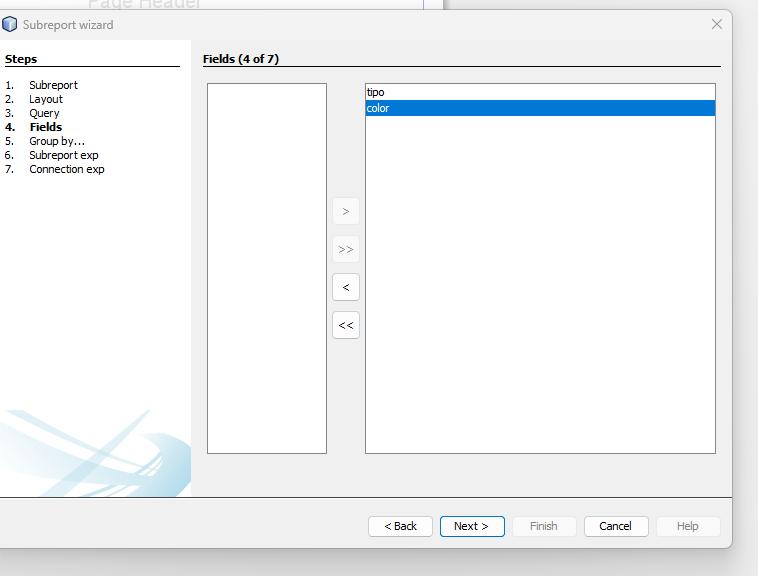
Hacemos clic en Next y seleccionamos de nuevo el formato a4 y hacemos clic otra vez en next:



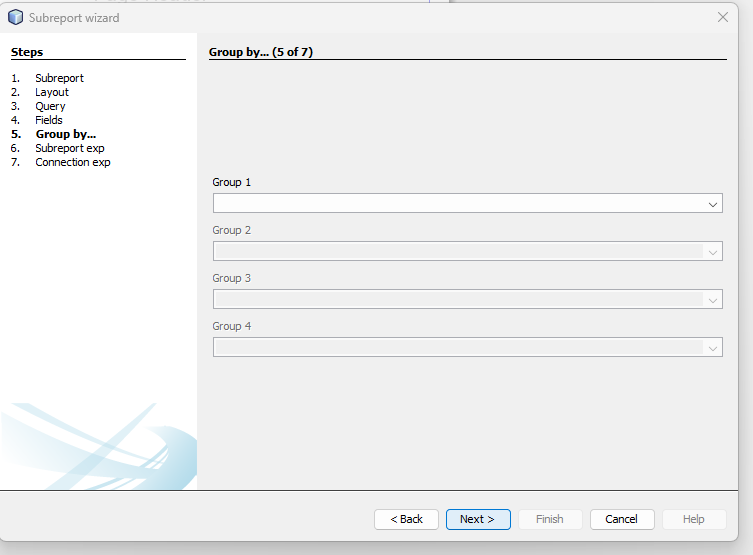
Aquí ahora podemos establecer la base de datos y la consulta que queremos utilizar para recuperar los datos, como ya sabemos cual es vamos a introducir la consulta para obtener el color y tipo de cada digimon:



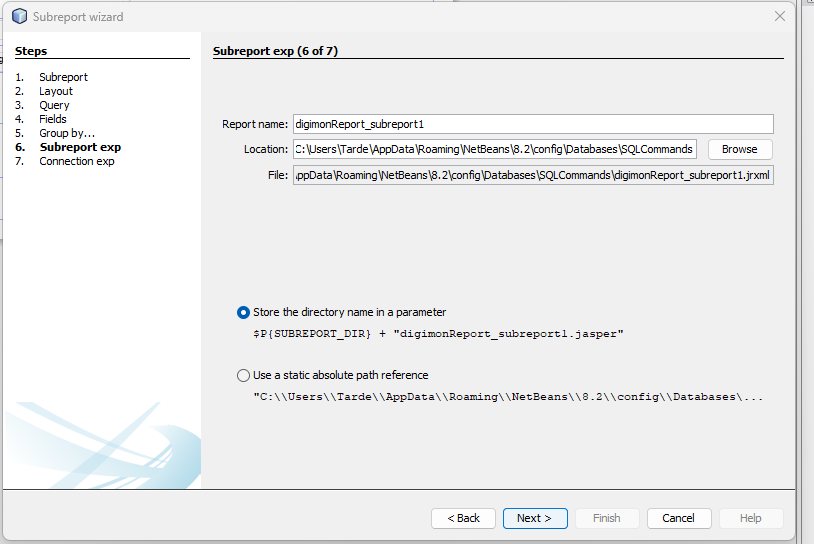
Aquí los campos nos aparecerán a la izquierda, los pasamos a la derecha para usarlos en nuestro subinforme.

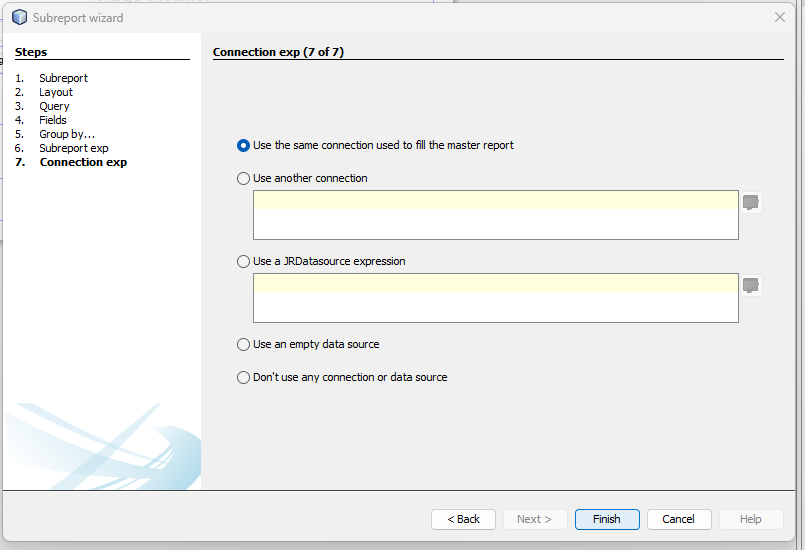


Aquí nos solicita si lo queremos ordenar por alguna de las dos características, vamos a dejarlo como esta:

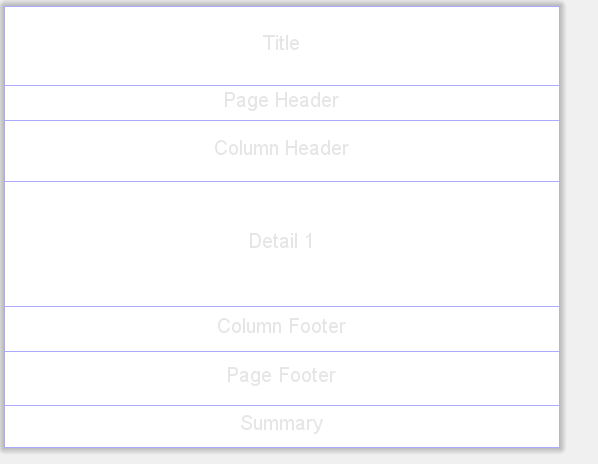


Y aquí configuramos el nombre y la ruta que queremos que tenga nuestro reporte:

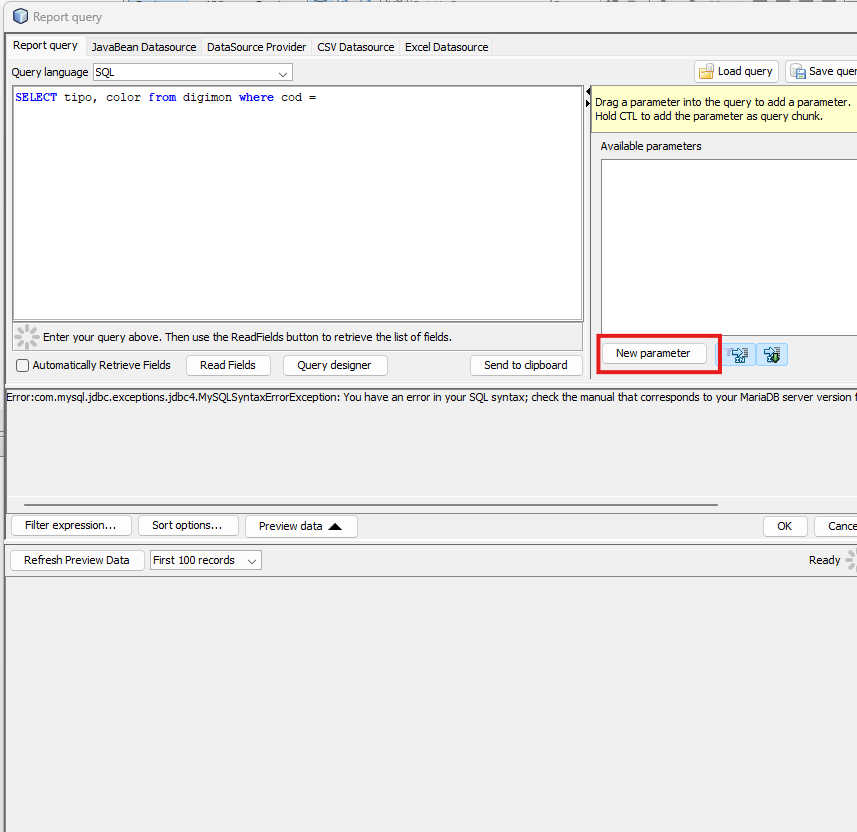


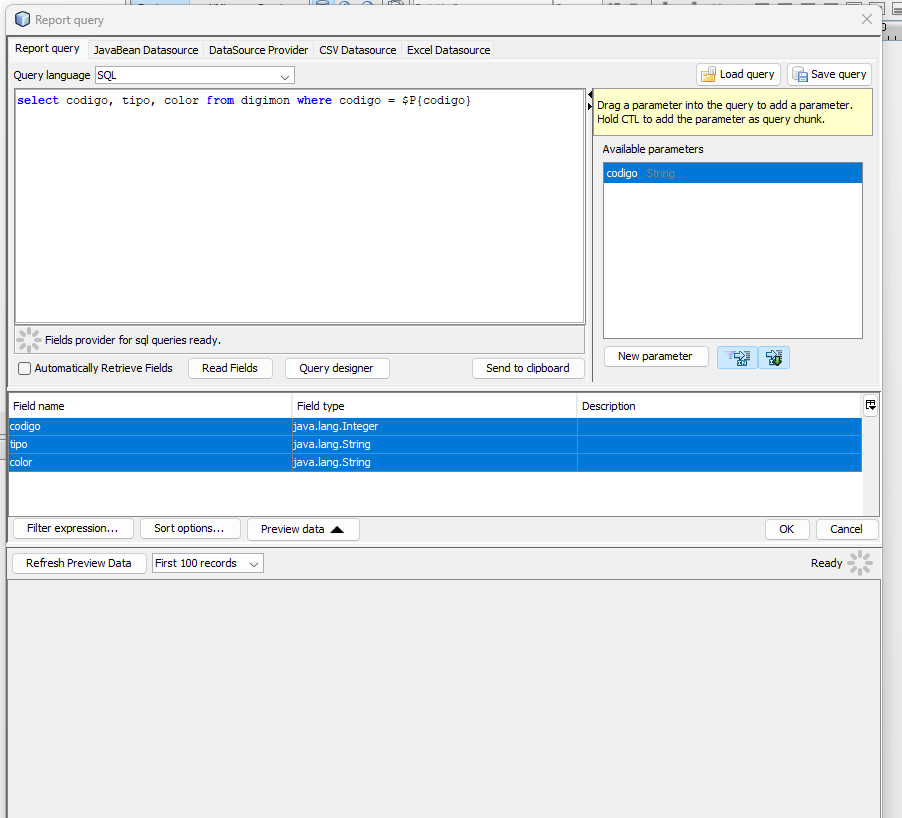
Esto lo dejamos como está y hacemos clic en Finish:

Y con esto habríamos creado nuestro subinforme:



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:

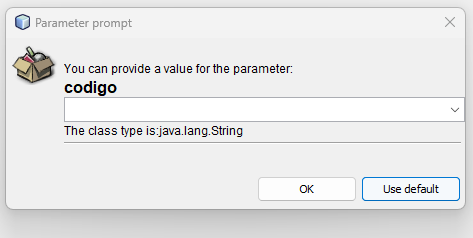




también tendremos que seleccionar el codigo para poder filtrar por el, ahora simplemente arrastramos la información y borramos el resto de bandas del reporte:

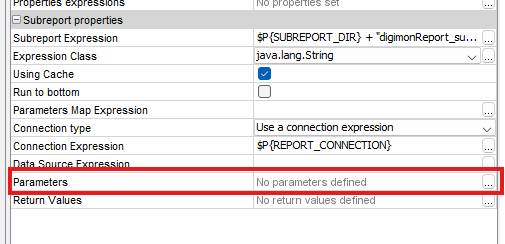


Ahora si hacemos una preview del informe podremos ver que nos solicita un parámetro y si le pasamos el código de algún digimon nos aparecerá su color y su tipo:

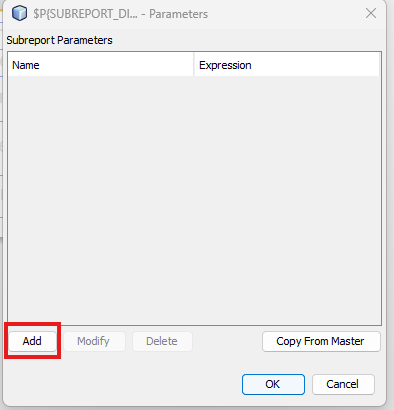




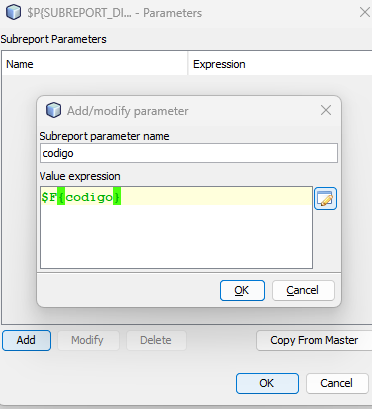
Ahora volvemos al informe principal y tenemos que establecer que para cada subinforme le pasamos el codigo de cada digimon del informe principal, para esto nos vamos a la parte que dice parameters y hacemos clic en los 3 puntos para configurarlo:



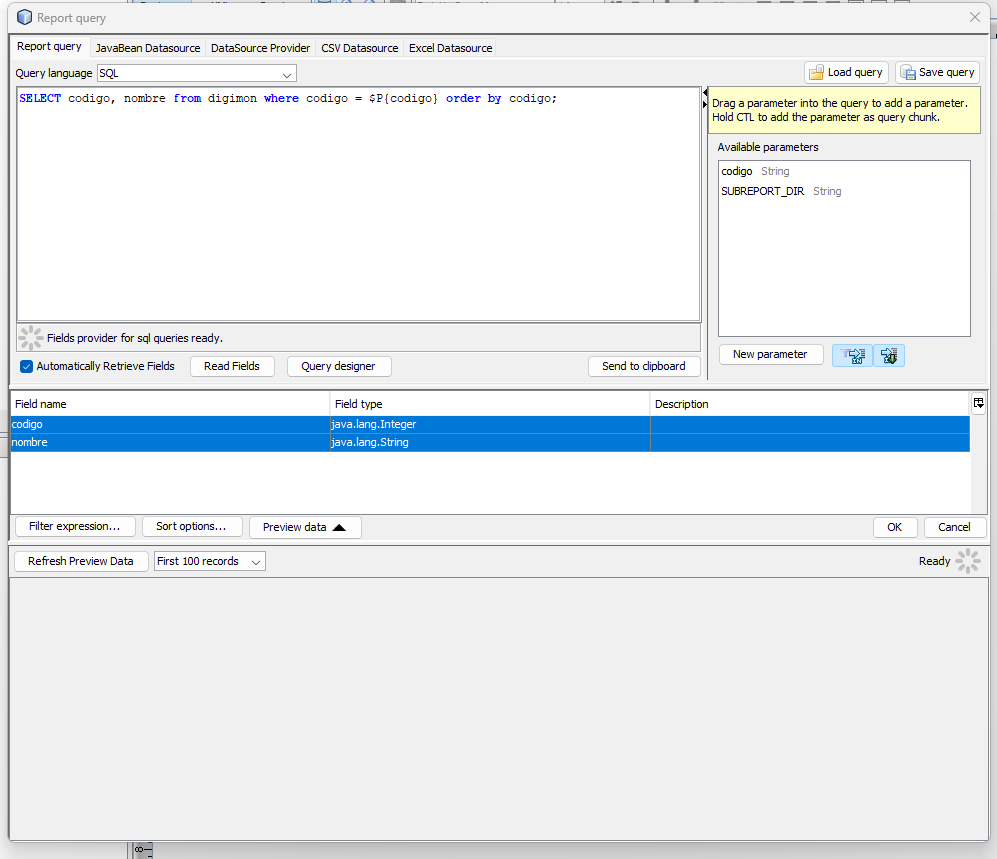
Aquí hacemos clic en add para añadir el parámetro:



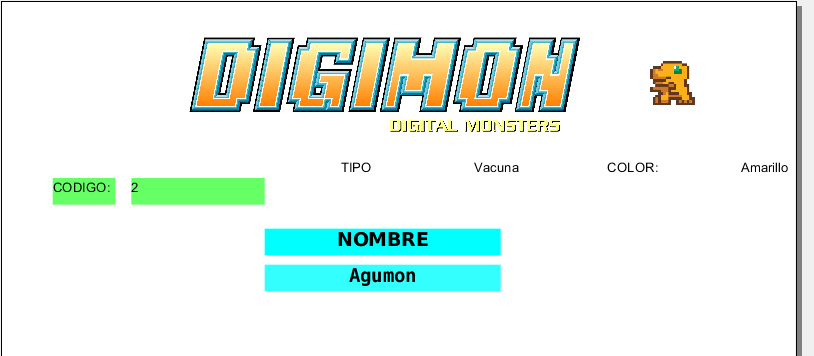
Y añadimos el parámetro:



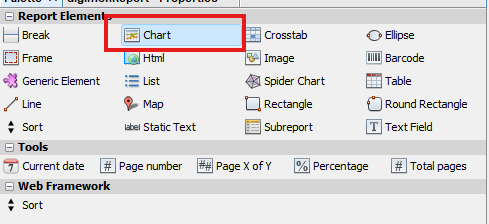
Ahora vamos a cambiar la consulta de nuestro informe principal para añadir un filtrado de digimon por codigo también para que nos aparezca de cada digimon su informe con su subinforme, simplemente modificamos la consulta y añadimos el parámetro codigo.



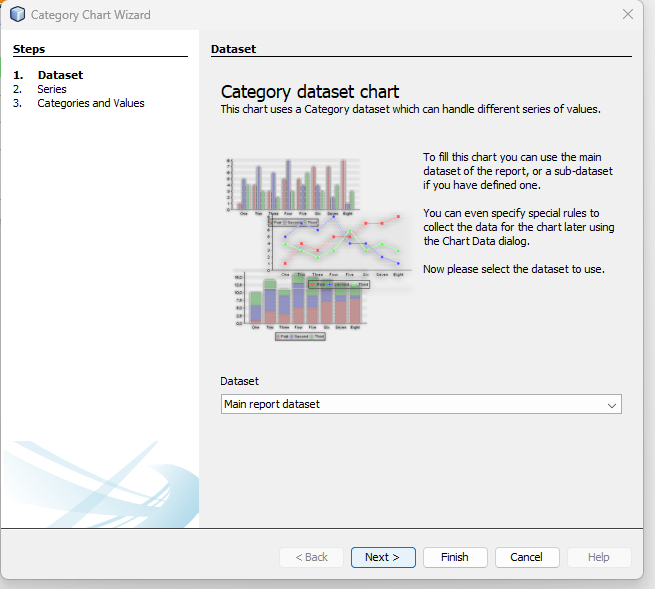
Finalmente así sería nuestro informe, que sería como una hoja de detalle de cada digimon:



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:

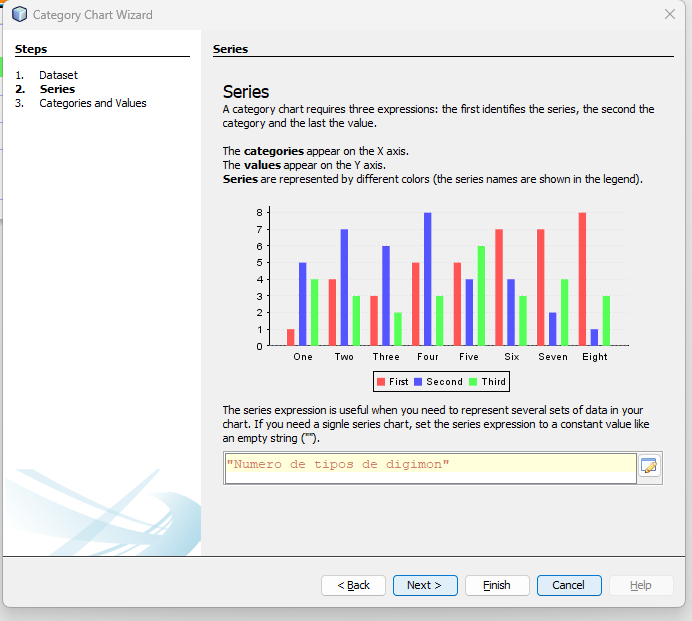


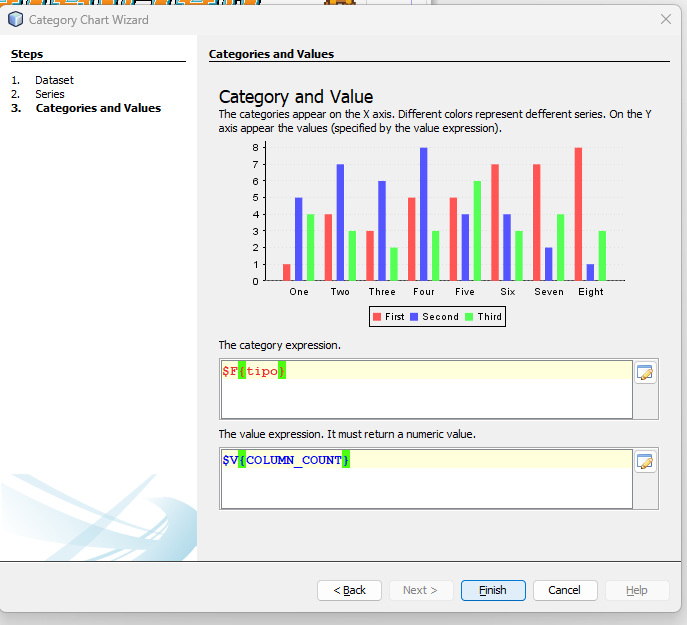
Al arrastrar el gráfico nos recibirá esta pestaña:



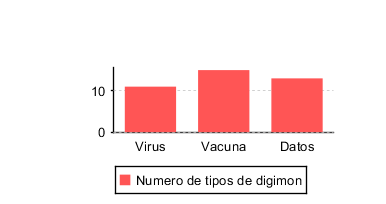
Hacemos clic en next:

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

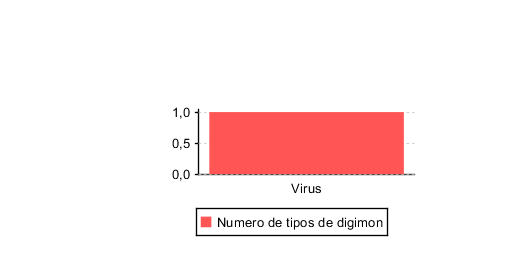


(Ahora tipo sale en rojo porque debemos añadirlo a la consulta del informe principal)

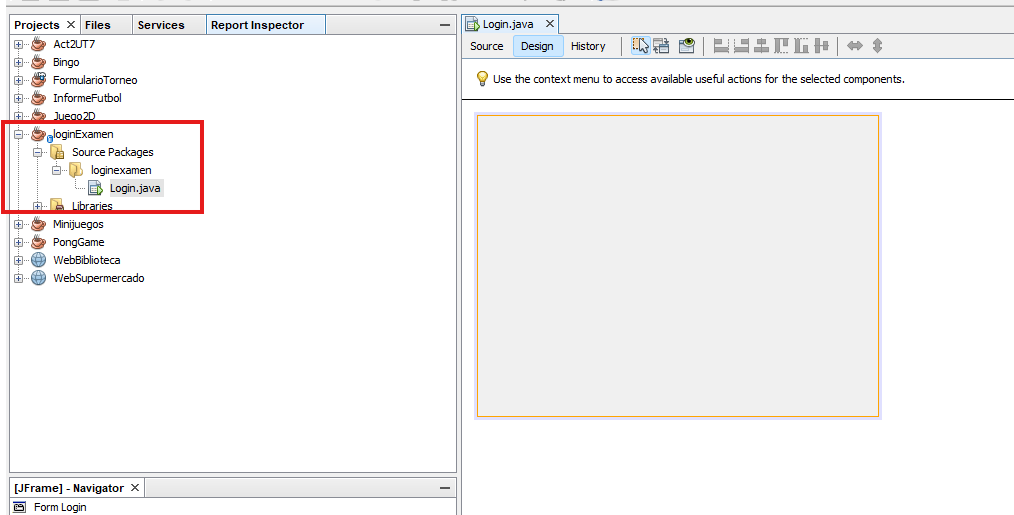
Una vez hagamos clic en finish podremos ir a la preview y consultar nuestro gráfico:



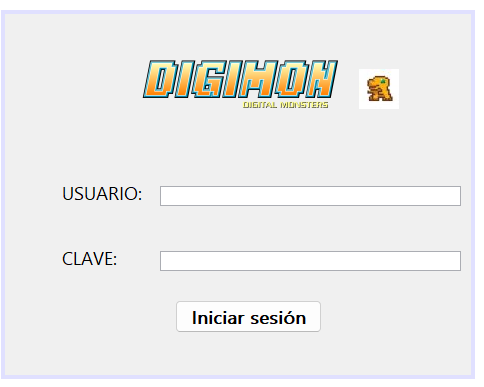
Y si establecemos la consulta para un solo digimon pues nos aparecerá únicamente el tipo de ese digimon:



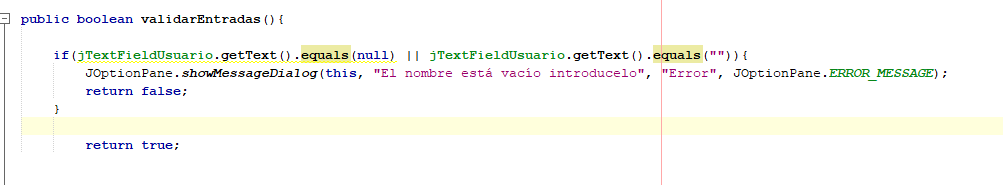
3ºA continuación se nos solicita realizar una pantalla de login para el inicio de sesión de los usuarios del nuevo juego de Digimon, para esto lo primero que haremos será crear un nuevo proyecto en NetBeans y un nuevo jFrame:

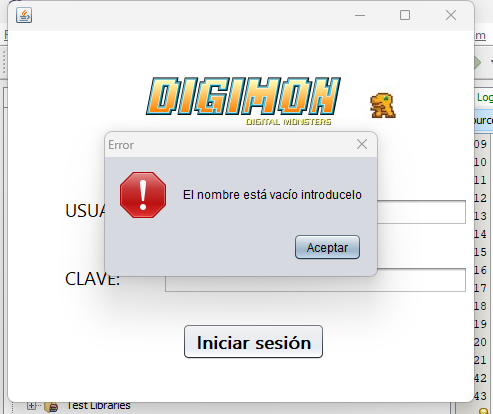


Ahora vamos a diseñar la pantalla de login:

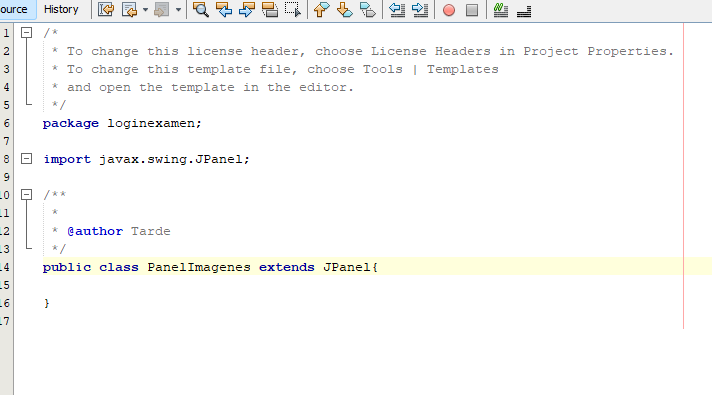


Ahora vamos a introducir la función de inicio de sesión, para ello el campo de usuario no podrá estar vacío.

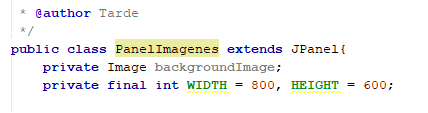




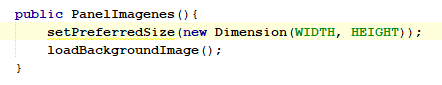
Ahora vamos a configurar la pantalla que muestre las imágenes del juego, para ello vamos a crear un nuevo Jpanel donde estableceremos las imágenes del juego, para esto vamos a crear una nueva clase de java de nombre PanelImagenes en la cual vamos a extender jPanel:



En esta clase vamos a incluir lo siguiente:

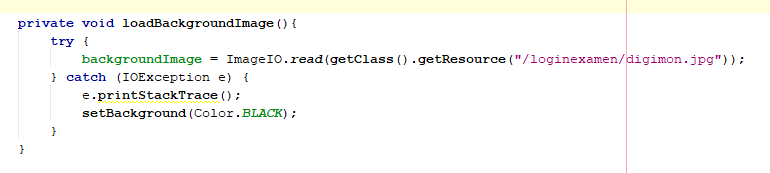


Una imágen para representar de nombre backgroundImage y dos variables enteras para el tamaño de la nueva pestaña del Jpanel, ahora vamos a implementar un constructor con los siguientes métodos:

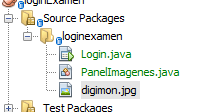


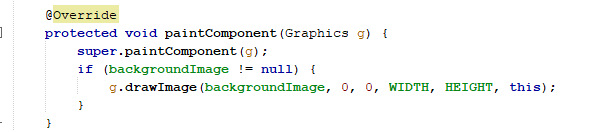
con setPreferredSize establecemos el tamaño que hemos indicado en las variables y con loadBackgroundImage cargaremos la imagen que queremos representar:

Ahora creamos el método loadBackgroundImage:

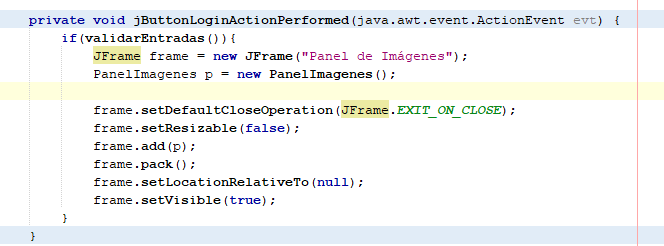


Deberemos tener un recurso de nombre digimon en nuestro paquete para que funcione:

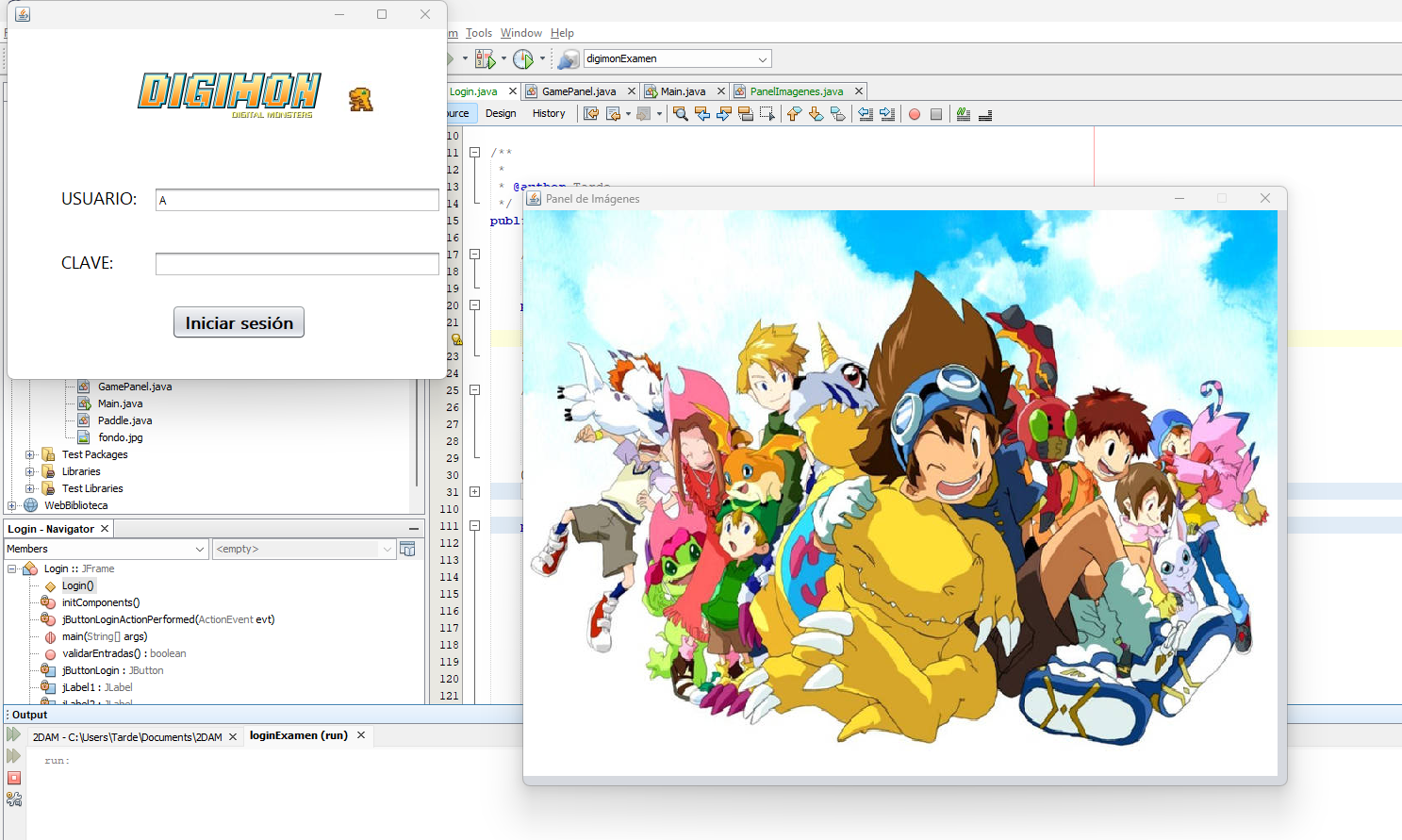


Finalmente nos queda introducir el método paint component para pintar automáticamente la imagen del fondo en el jpanel:

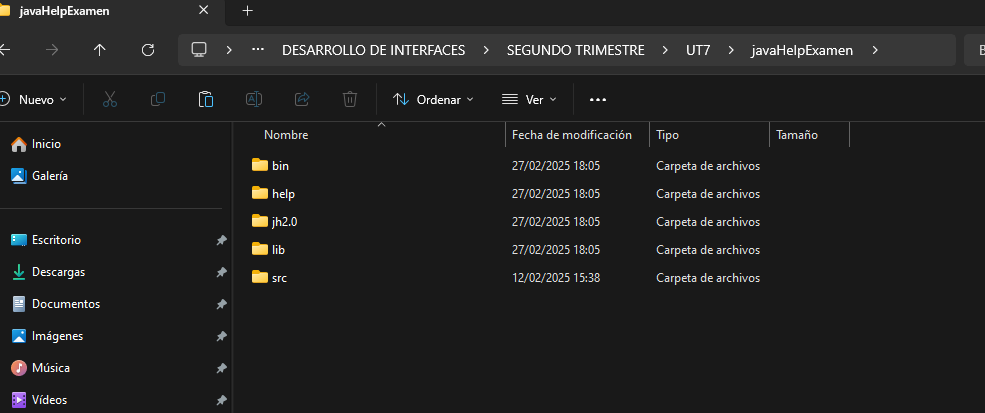
Ahora volvemos a nuestro login y debemos de declarar una nueva instancia de PanelImagen al validar la entrada y un nuevo Jframe para representar el jPanel:



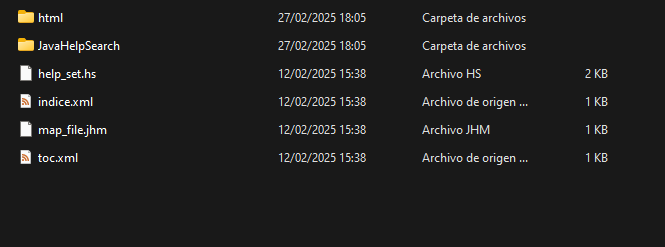
Con estos métodos le damos un método de cerrado para que se cierre la aplicación al cerrar el panel, no permitimos que se le pueda cambiar el tamaño, lo mostramos en el centro y la hacemos visible, y se vería así:



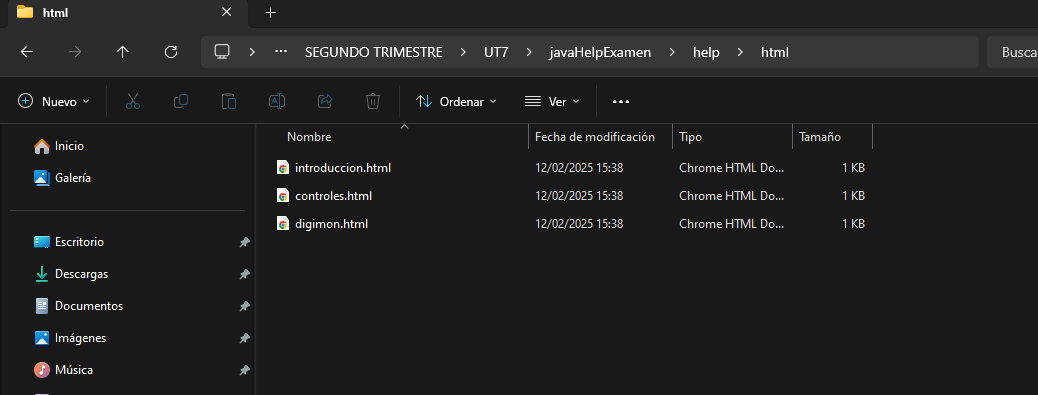
Ahora vamos a introducir JavaHelp en el proyecto, para esto lo primero que debemos de hacer es en obtener las herramientas de JavaHelp de cualquier otro proyecto y modificar su contenido para adaptarlo a este proyecto:



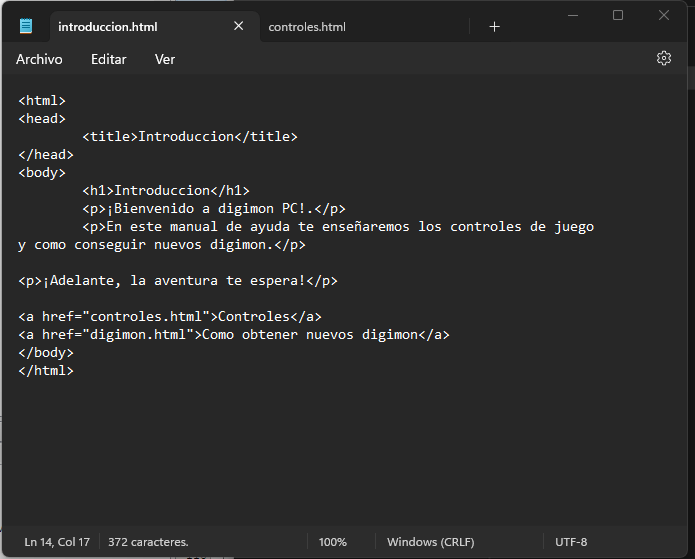
Ahora en la carpeta help vamos a modificar las distintas páginas que tendrá nuestro manual de ayuda, que se encuentran en html:



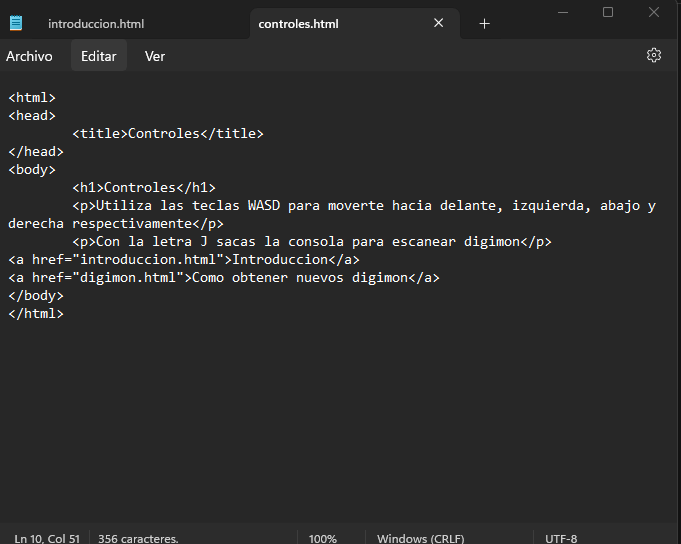
En este caso yo voy a realizar un manual de ayuda JavaHelp con tres páginas:



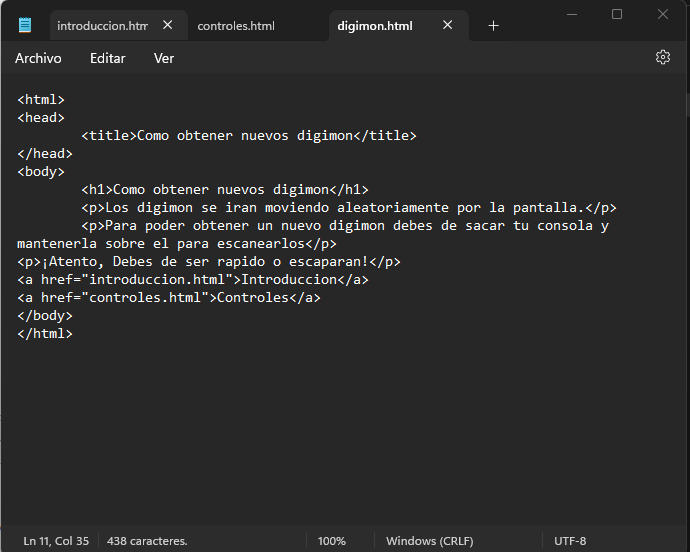
En la página de introducción tendremos una breve información sobre el manual de ayuda

y enlaces a las otras páginas:

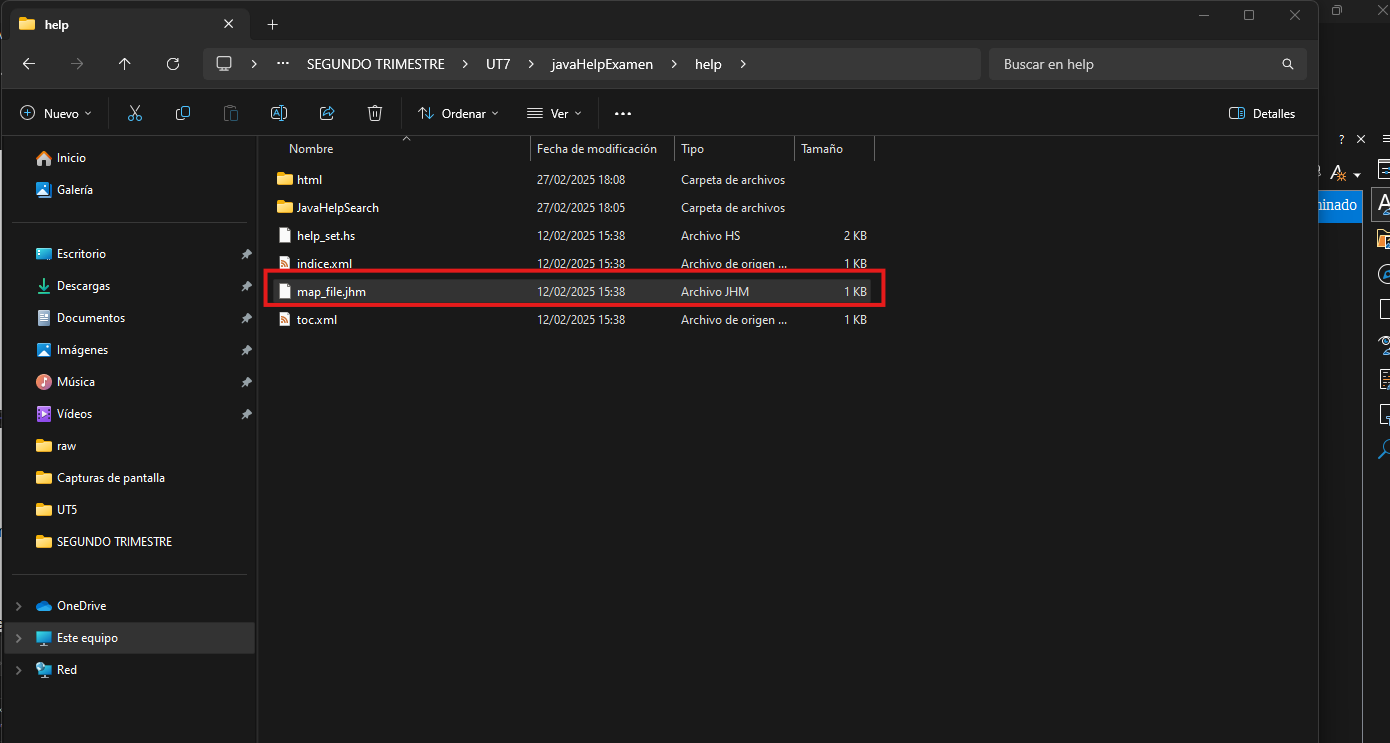
En controles tenemos información básica sobre los controles del juego:

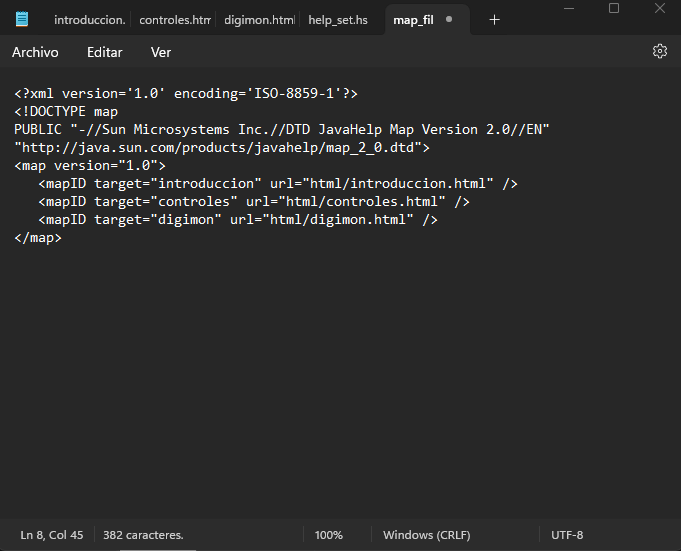


Finalmente en el apartado de digimon tenemos una breve guía de como obtener nuevos digimon:

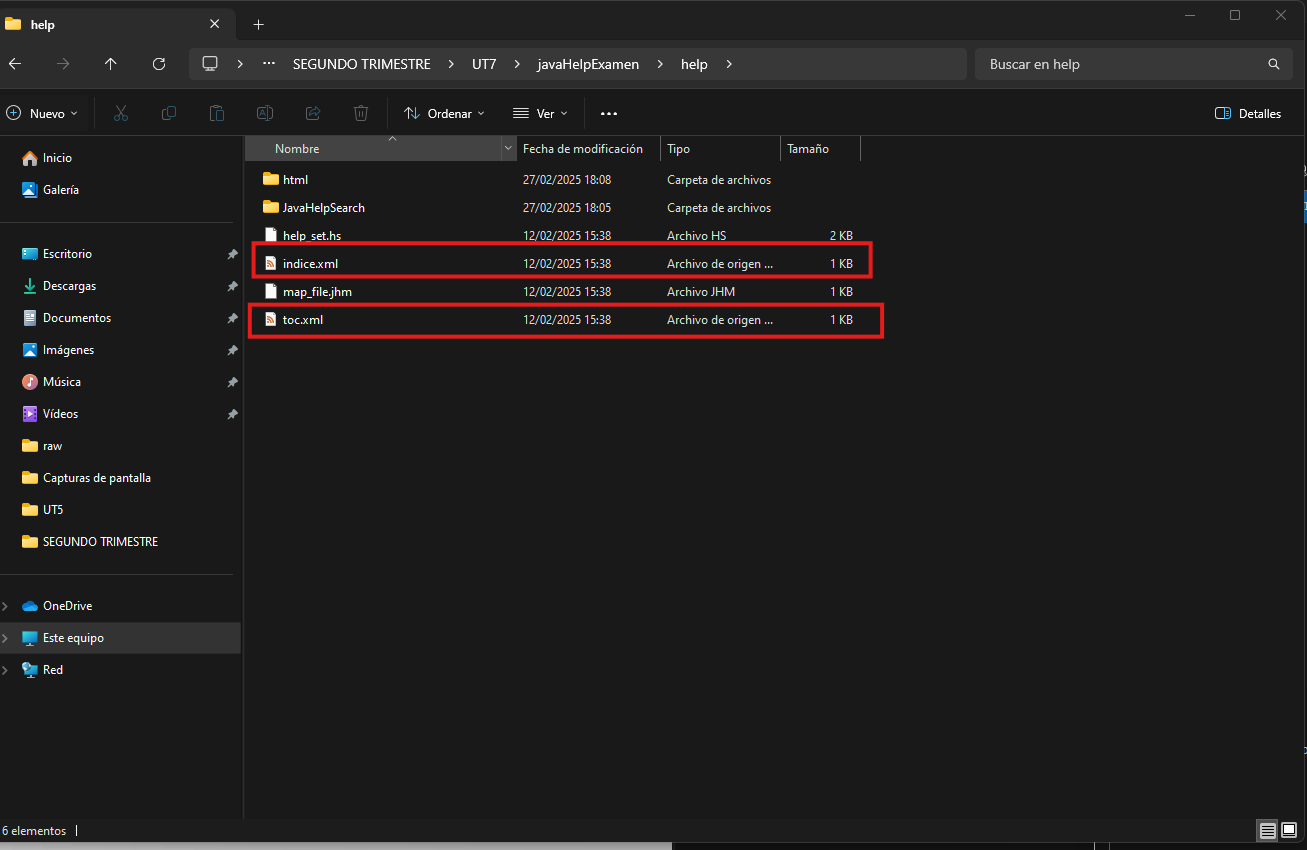


Una vez terminado de editar las diferentes páginas debemos de configurar el mapa, para ello modificaremos el archivo:

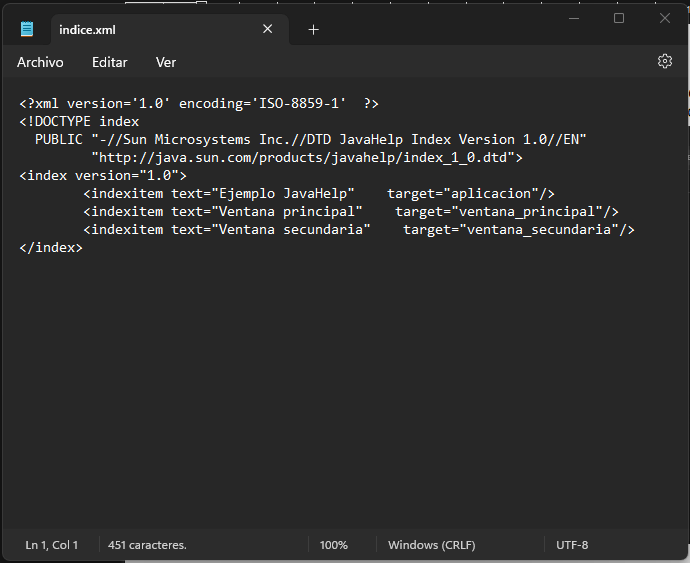




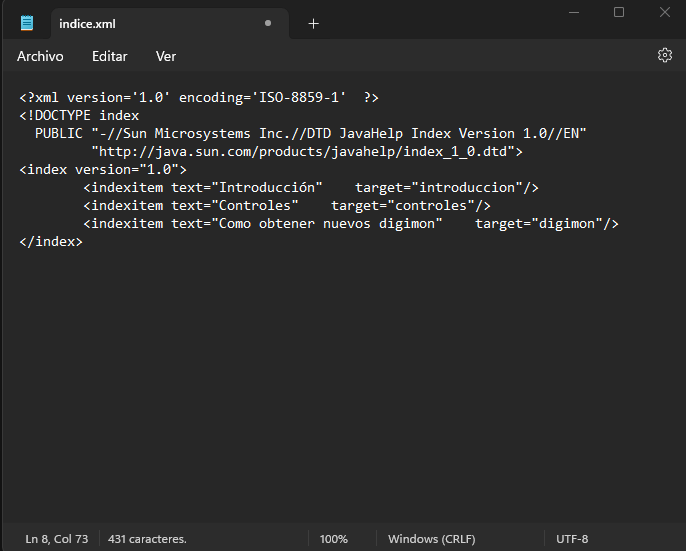
Una vez configurado el mapa, configuraremos el índice y la tabla de contenidos para que se representen correctamente, para ello vamos a abrir y editar los archivos:



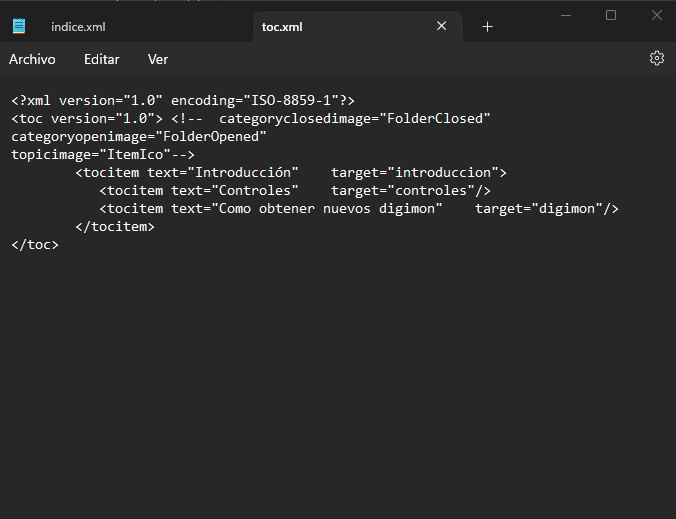
Ahora vemos por ejemplo en indice tenemos distintos elementos de nombre indexitem que hacen referencia a “aplicación”, “principal” y “secundaria”:



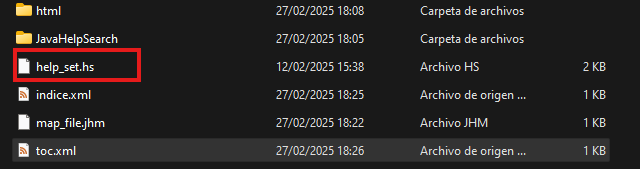
Estos eran los nombres que tenían antes nuestras pestañas en el mapa, entonces debemos de cambiar esos nombres por los que hemos establecido en el mapa, yo he puesto el mismo nombre que las páginas pero no tiene porque ser así.

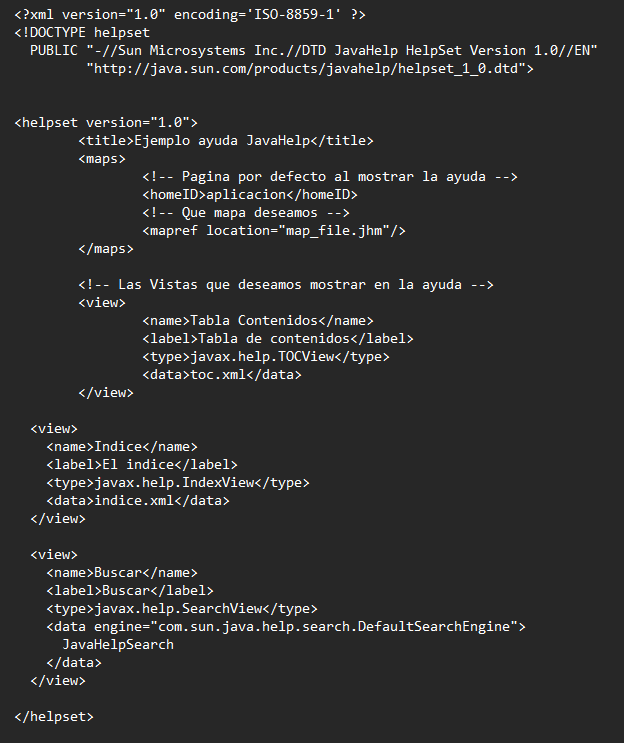


Y ahora hacemos lo mismo con el toc que veremos que es prácticamente lo mismo, solo que en vez de ser indexitems son tocitems:

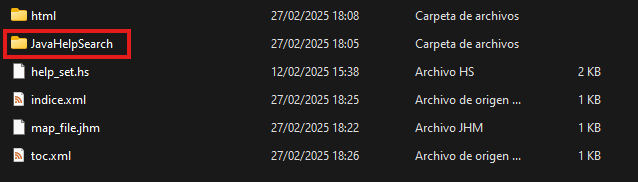


A continuación vamos a editar el archivo:



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

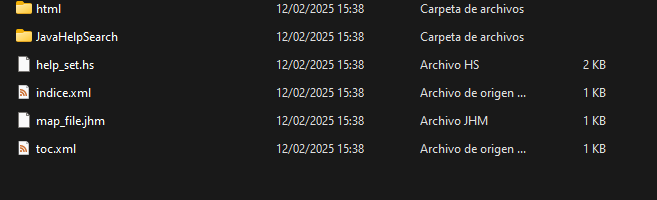
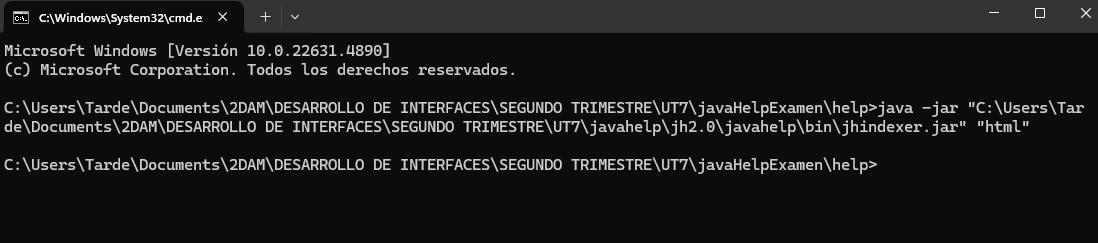
Finalmente vamos a establecer el JavaHelpSearch para que el usuario pueda filtrar por palabras clave, para ello lo primero que debemos hacer es borrar esta carpeta:



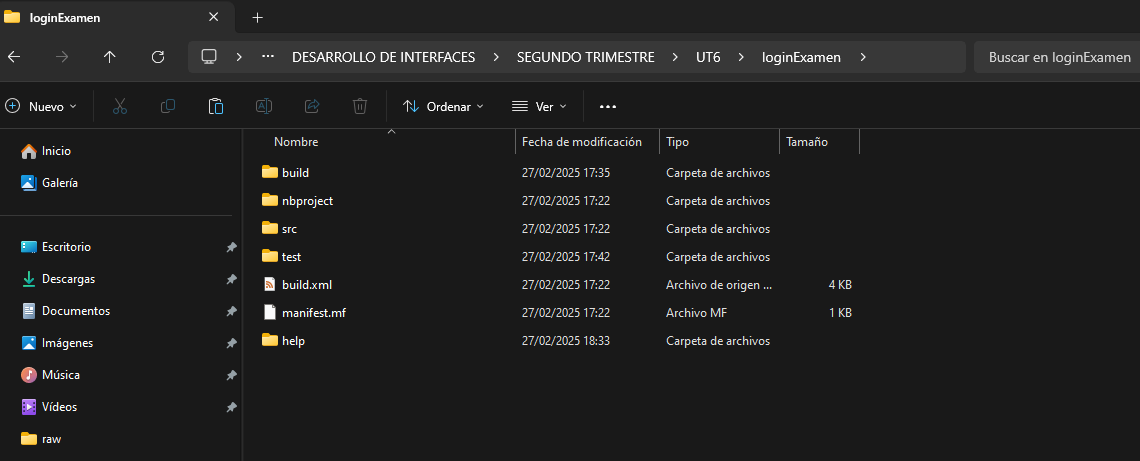
Una vez borrada abrimos una nueva consola de comandos en esta carpeta (lo podemos hacer escribiendo cmd en la ruta):

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:

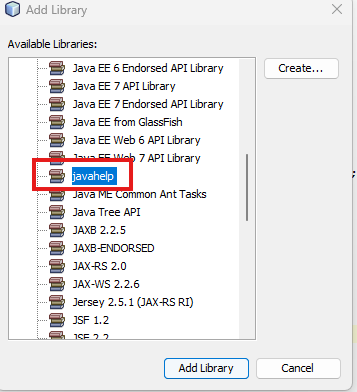


Ahora vamos a implementarlo en nuestro proyecto, para ello lo primero que tenemos que hacer es copiar la carpeta de help a la ruta del proyecto base:

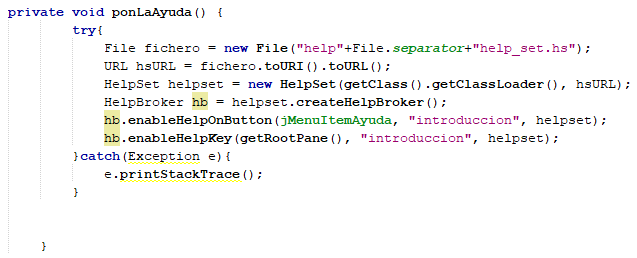


Ahora vamos de nuevo al proyecto para añadir el menú de ayuda y añadimos una menú bar y un menú item y le cambiamos el nombre para que nos quede así:

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

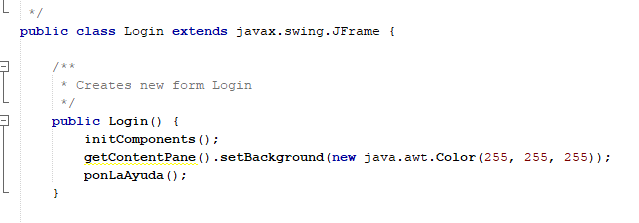


Una vez añadidas las librerías utilizaremos el siguiente método para vincular la ayuda de JavaHelp al botón del menú que hemos declarado:

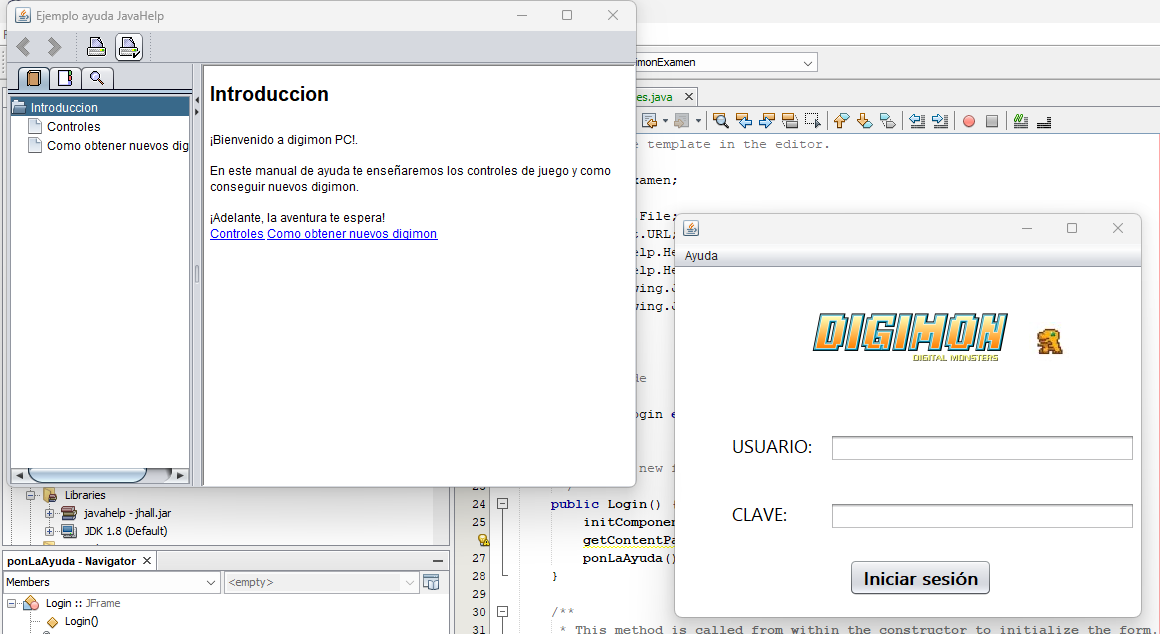


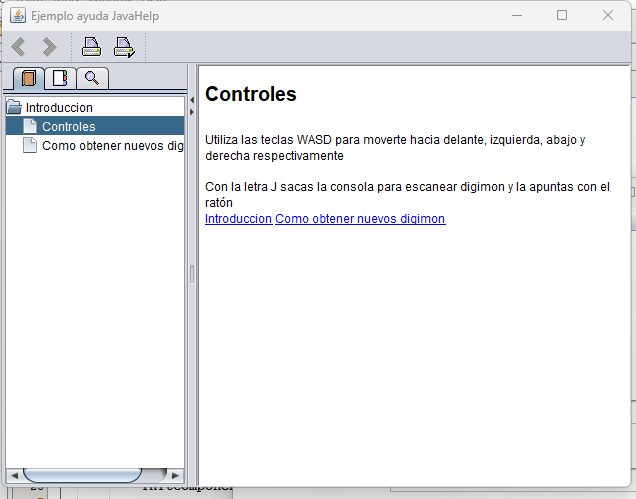
Así vinculamos la JavaHelp con los botones que hemos creado.

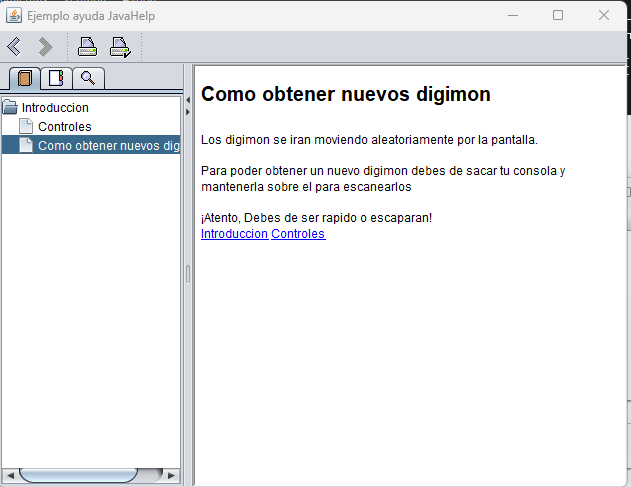
Ahora simplemente llamamos a este método aquí:

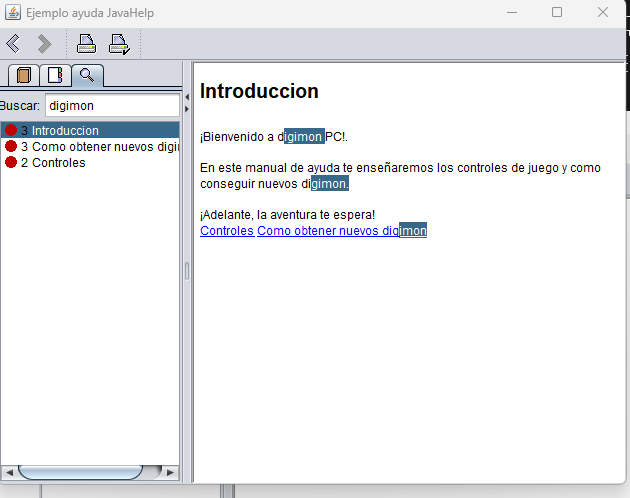
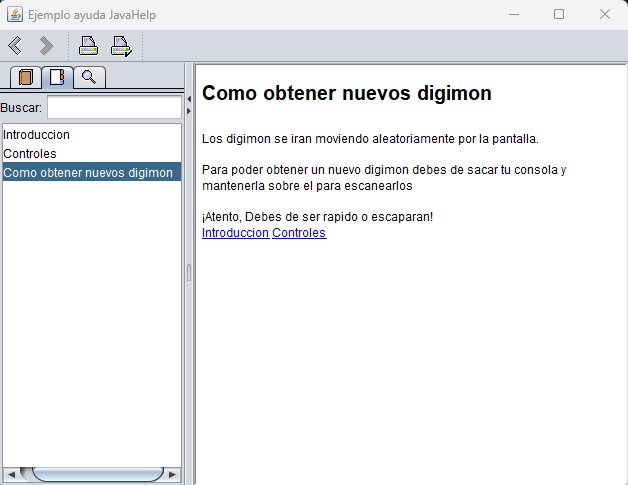


Y comprobemos que funciona:

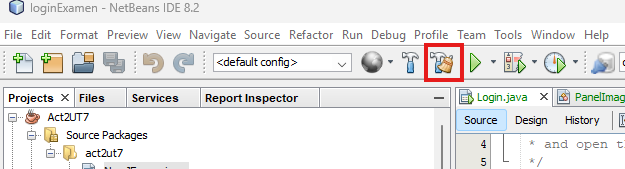








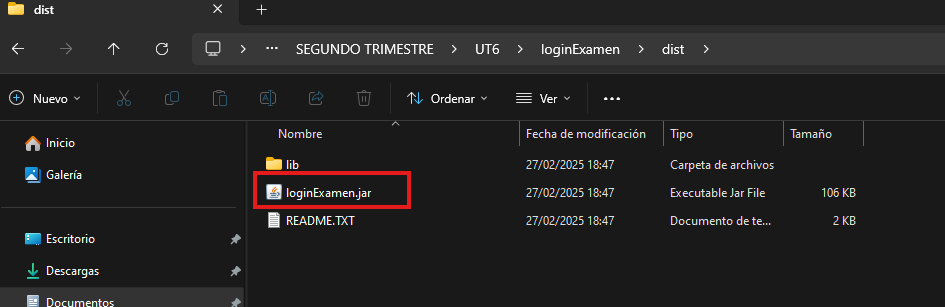
4º Vamos a construir el jar de nuestro proyecto, para esto simplemente vamos a hacer clic en clean and build:



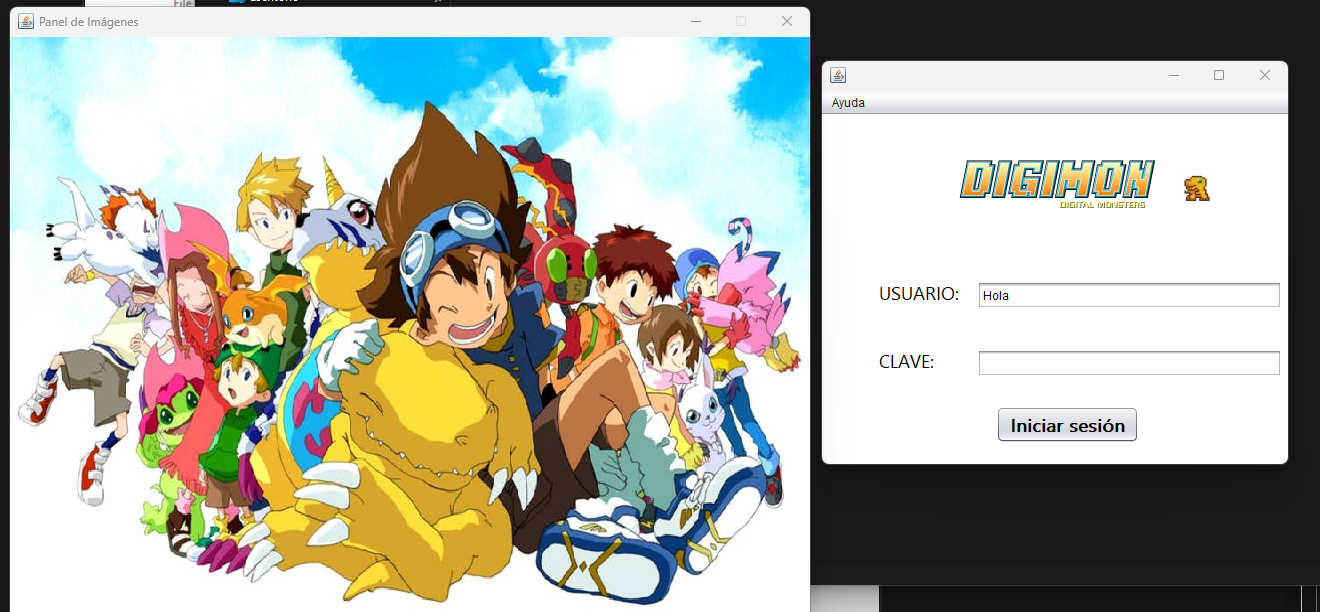
Aquí podemos ver que nuestro jar se ha generado sin problemas en la siguiente dirección:



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.