

# DOCUMENTACIÓN TAREA 4

**-DÍA 1:** En el día de hoy he comenzado la tarea 4 la cual consiste en crear una base de datos que se alimente de los diferentes bloques y luego en el jsp poder mostrar los ficheros desde la base de datos.

Mi primera idea ha sido crear una base de datos dinámica, es decir, que se creen y se rellenen las tablas en función del fichero que se este leyendo pero David no me ha recomendado esto debido a que decía que me iba a llevar mucho tiempo y no iba a sacar demasiado de esto. Con lo cual he descargado “XAMPP” para acceder a “php my admin” y realizar la base de datos en esta plataforma. He realizado algunas pruebas y he decidido crear un script usando LDD y ejecutarlo en “php my admin”.

**-DÍA 2:** Con la base de datos creada el próximo objetivo ha sido relacionarla con mi proyecto en net beans importando la librería de MySQL y con esto realizado he realizado algunas pruebas para comprobar que la conexión se ha establecido correctamente. También he estado planteando como iba a cargar la base de datos una sola vez, ya que si la cargaba más veces se iban a repetir continuamente las mismas líneas y he llegado a la conclusión de que voy a poner un botón que sea específicamente para cargar la base de datos y realizar todos los “inserts” y justo al hacer click en este botón lo primero que haga sea un “truncate” de todas las tablas que haya dentro de la base de datos. Para realizar esto he acoplado la aplicación 2.2 tal cual esta solo que en la función de escritura en vez de generar los “.XML” va a realizar los “inserts”, aunque el acoplar esta tarea no ha sido facil ya que las funciones requiere que esten en un “package” pero en “.jsp” no los hay, con lo cual mi decisión ha sido crear una librería con las funciones e importarlas desde la aplicación “.jsp” y así he conseguido que funcione correctamente. Esto ha desembocado en un nuevo error y es que “glash fish” me decía que no tenía suficiente memoria y me he tenido que meter en el menú controlador de “glass fish” para poder ampliarlo, entonces funcionó perfectamente.

**-DÍA 3:** Lo próximo que he realizado ha sido adaptar el código de la practica 2.2 de tal manera que en lugar de generar los archivos “.XML” genere inserciones. Para comenzar he creado una tabla en la base de datos de prueba y un fichero de texto de prueba pero con la misma estructura que el lote que usaremos realmente. A la hora de generar la estructura de la consulta también he establecido las excepciones de si es el final de la sentencia que no ponga una coma tras el elemento y si no es al final que si que la ponga. El siguiente paso será detectar si la columna de la base de datos es INT, DECIMAL O VARCHAR para poner o no las comillas simples.

**-DÍA 4:** Hoy he encontrado una función llamada ReadMetaData con la cual puedo obtener los metadatos de las tablas de la base de datos y de esta misma, con lo cual puedo obtener su estructura y de que tipo es cada columna de una tabla dentro de la

base de datos. He añadido esa excepción, además he hecho pruebas con más ficheros en más tablas arreglando pequeños errores de estructura en las tablas y campos de MySQL, y todo me ha funcionado correctamente menos en una tabla en concreto la cual se llama "IDCL". Me da problemas debido a que es la única que acaba con dos tabulaciones entonces el contador de campos en ese punto en concreto no las contaba y el tamaño del "array" no era el correcto y además los valores que almacenaba eran "null". Esto me ha impedido recorrer el fichero usando un común bucle "for" y he acabado usando un bucle "for each" pero me ha dado otro error debido a problemas de espacio en la base de datos.

Tras comentarle este problema a David el me aconsejó que le pusiera algunos valores en ese campo de momento ya que arreglar el error nos iba a llevar mucho tiempo y no quería que perdiésemos tiempo en ese punto ya que conocemos el error y como solucionarlo pero requeriría mucho cambio a nivel de código.

**-DÍA 5:** En el día de hoy he comenzado a intentar leer las diferentes tablas que hay dentro de la base de datos, para ello he usado la librería "metadata" la cual es una ampliación del "ResultSet" y se utiliza para obtener los datos más estructurales de una tabla o de una base de datos. Con esto he obtenido los nombres de las tablas que existen dentro de la base de datos .

Además le usado también para obtener el numero de registros de cada tabla y el numero de columnas de estas. Estos datos me han servido a la hora de mostrar por pantalla el contenido de las tablas cuando seleccionas alguna del desplegable.

**-DÍA 6:** Hoy me faltaban los últimos pasos, y han sido calcular el numero de registros totales (el cual e averiguado haciendo un sumatorio en otra variable de los registros insertados por bloques.) y los tiempos totales de inserción y de la consulta del bloque seleccionado.

Para obtener ambos tiempos he usado la función "System.currentTimeMillis()" la cual te dice la hora actual del sistema en milisegundos, usandola de tal manera que antes del bucle de visualización obtenga la hora y junto cuando acabe el bucle también y se realiza una resta y te dice el tiempo que ha tardado en realizarse. Para hacer el tiempo de las inserciones he tenido un pequeño problema ya que las inserciones las hago en una función dentro del "index.jsp" y la visualización la hago dentro de "visualizar.jsp". He planteado el crear una clase estática y también crear una nueva función, pero he decidido ir por el camino fácil y lo que he hecho ha sido pasar los valores por la URL y solo mostrar por pantalla el tiempo de las inserciones la primera vez que se carga "visualizar.jsp" ya que para mostrarlo siempre tendría que estar pasando el dato siempre por URL y me parece redundante. Es decir, el dato ese solo se mostrara si detecta que se han hecho inserciones, el resto de veces no.

Por último he tenido que plantearme el hecho de crear un índice ya que de la forma que yo tengo montada la base de datos no veo demasiado eficiente crear un índice sin relaciones en las tablas, con lo cual tendría que crear un índice para cada tabla, y por si fuera poco no todas las columnas de de todas las tablas son iguales, es decir, una tabla puede tener una columna importe y otra puede no tenerla.