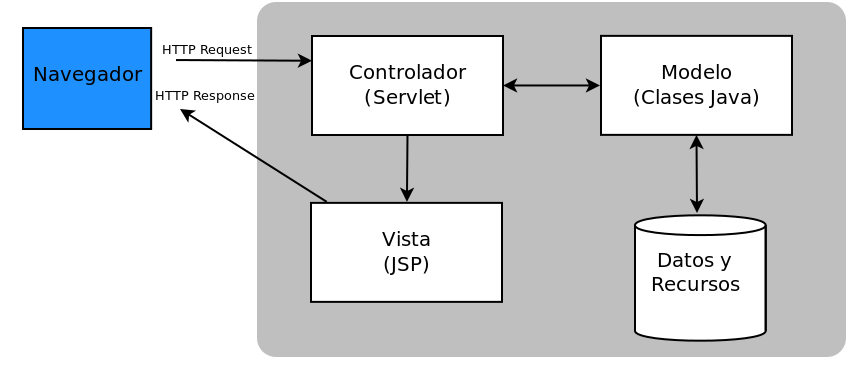
Manual Técnico

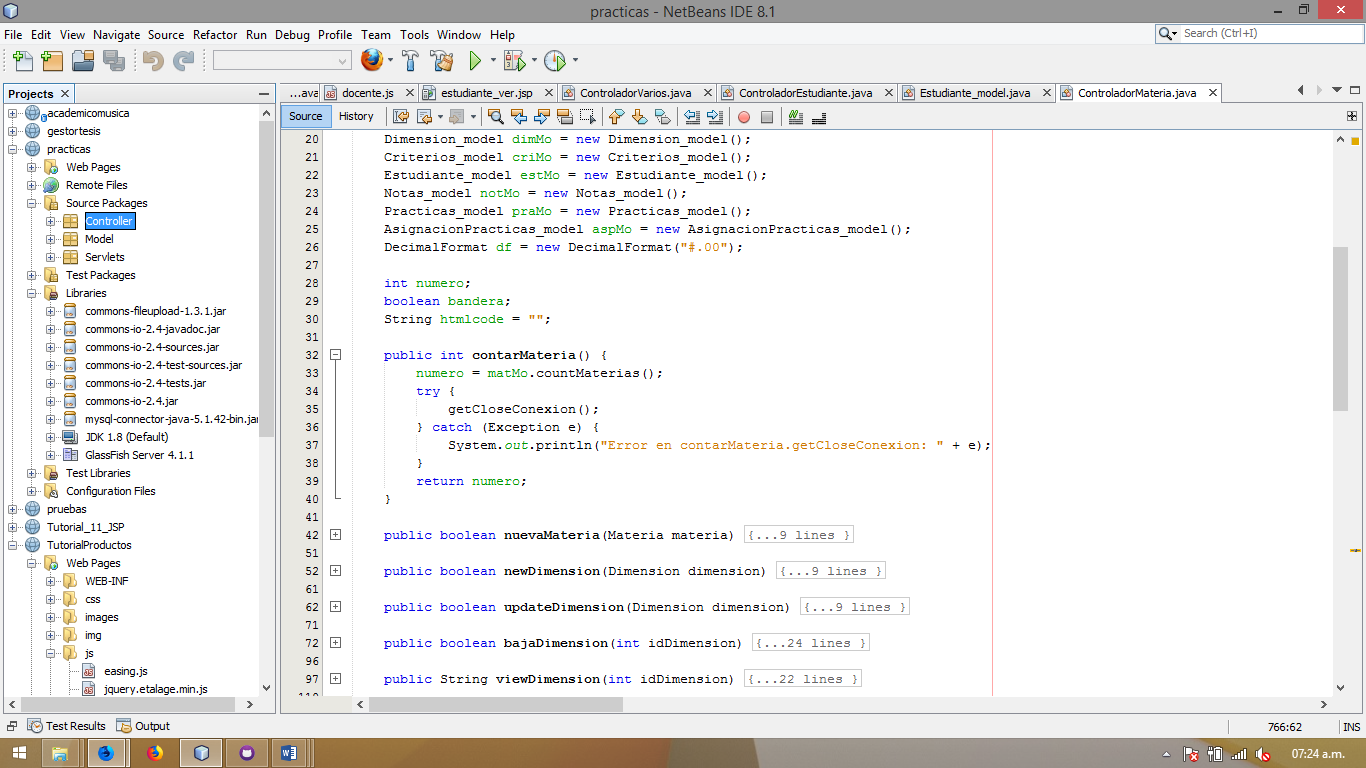
El sistema está desarrollado en JAVA WEB o también conocido como JSP (Java Servlet Page), este lenguaje de programación web implementa desde sus raíces el patrón de diseño MVC debido a que existe:

1. Una Vista que son las páginas .jsp
2. Un Controlador que son los Servlets
3. Y un modelo que son las clases de java

Como podemos observar en el ejemplo cada una de las partes cumple con el requerimiento del patrón de diseño MVC



Pero a su vez estoy implementando unas clases controladoras más para aumentar su eficiencia al momento de mostrar en la vista los datos y las llame clases Controladoras, siendo asi el resultado final de mi proyecto el siguiente.



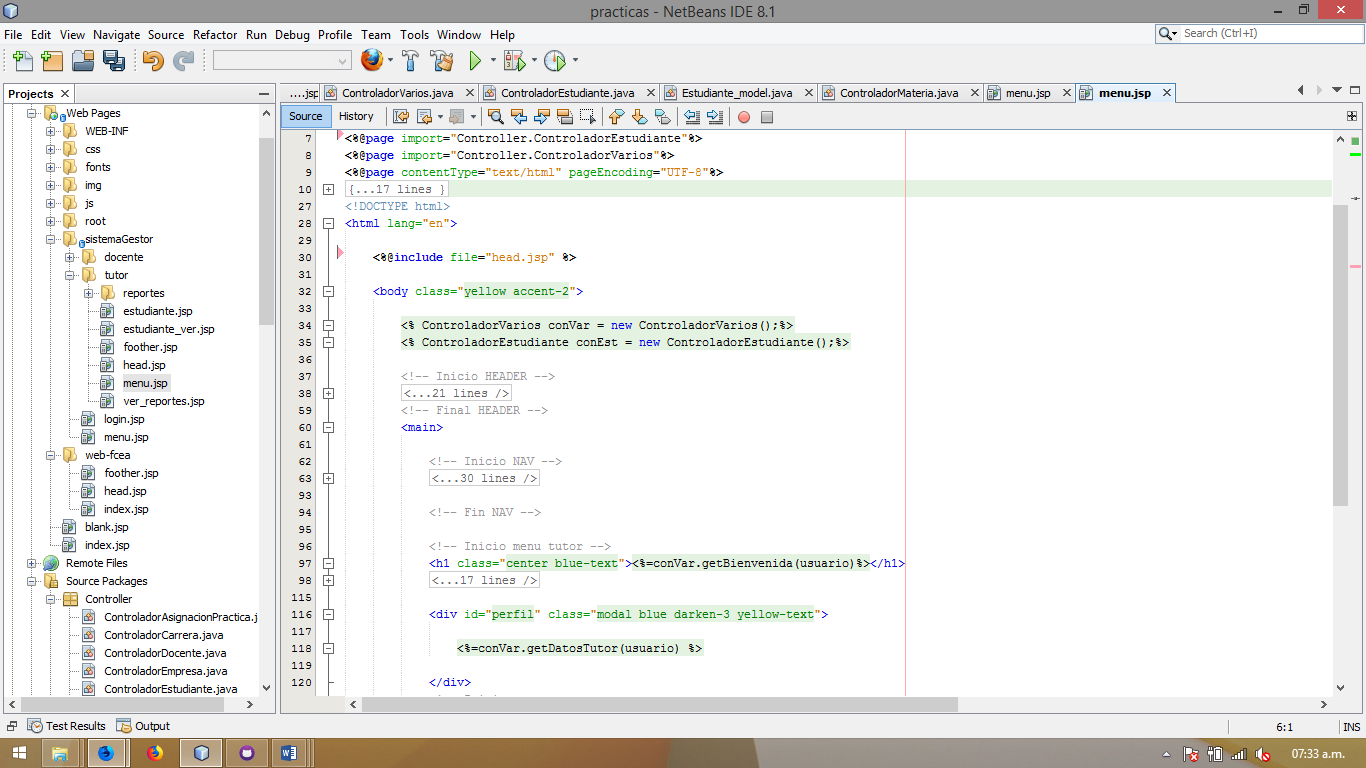
Las Web Pages 🡪 ahí están alojadas todas las páginas .jsp del proyecto (las vistas)

Los Controller 🡪 ahí están todas las clases de java que utilizo como controladores ayudantes

Los Model 🡪 es ahí donde guardo todos los modelos y las consultas a la base de datos

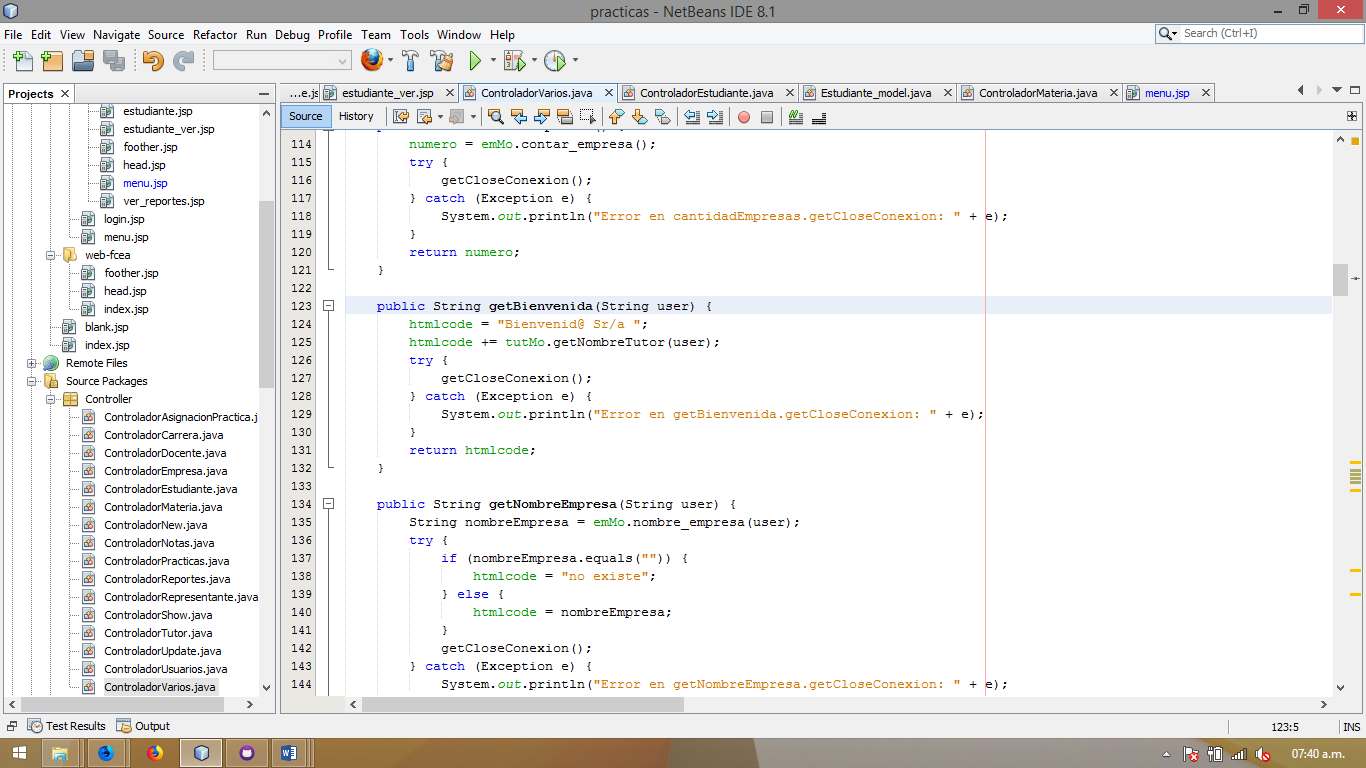
Los Servlets 🡪 y por ultimo ahí están los Servlets que son los controladores básicos de JSP

Web Pages

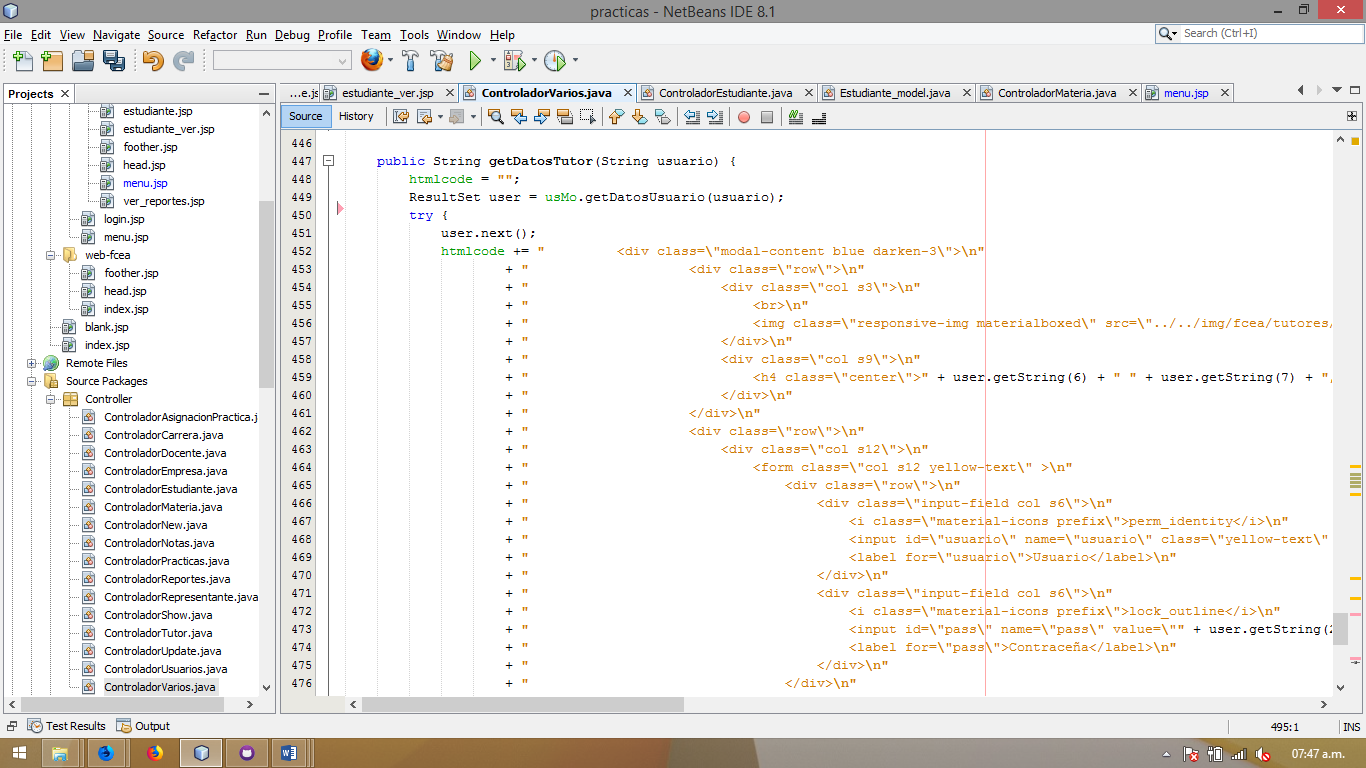


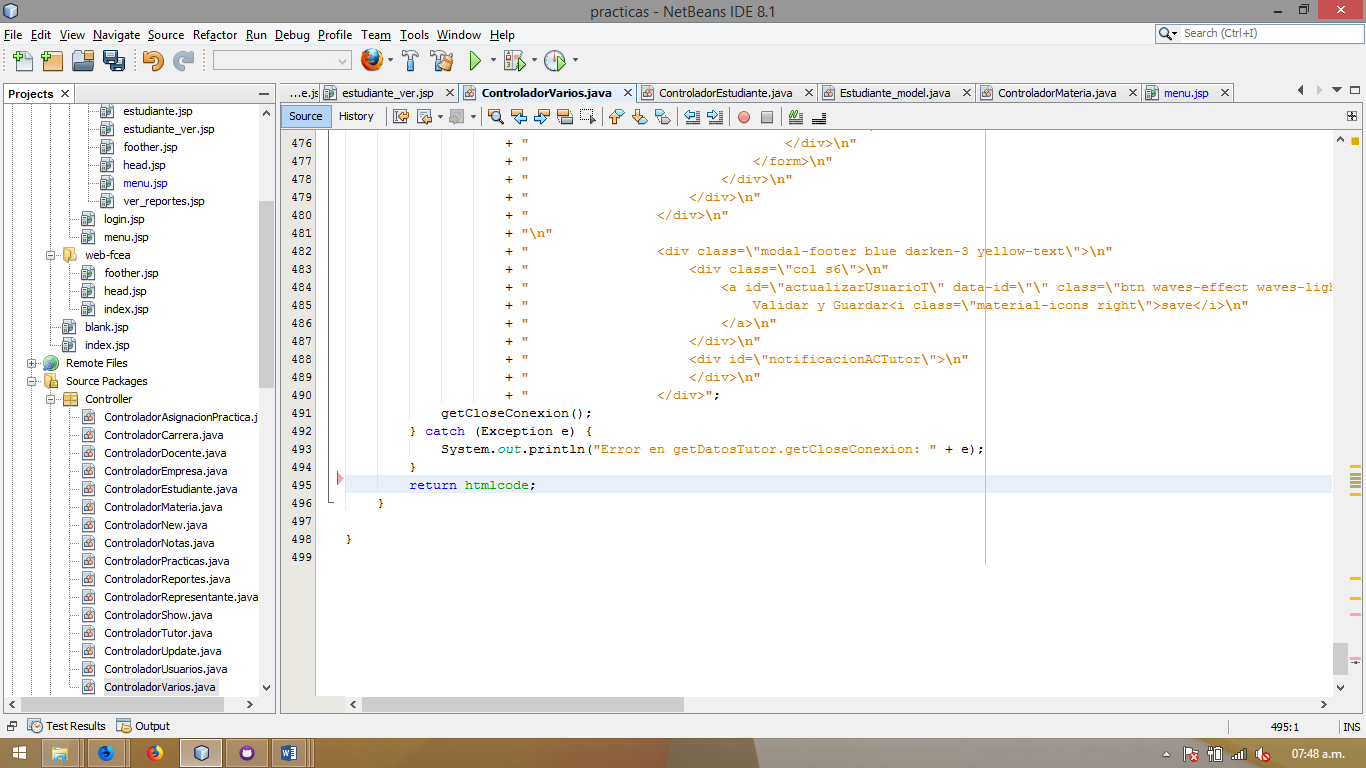
Están constituidos de la siguiente manera:

1. Las Primeras líneas que se observan son las importaciones externas que estoy realizando en este caso lo de los controladores necesarios para generar correctamente
2. Luego importo una vista que es la del head.jsp donde declaro todos mis estilos (en su contraparte al final importo el foother.jsp que es donde están guardados los js necesarios)
3. Luego se declara normalmente como un código HTML lo que necesite el body, main, etc.
4. Y ahí es cuando uso clases controladoras para mostrar datos tanto específicos como un conjunto de HTML. Donde se observa el <%=conVar.getBienvenida(usuario)%> se solicita un dato más específico como lo es solo el nombre del usuario.



1. Tambien existen las clases que devuelven mas HTML como lo es la siguiente que se encuentra ahí <%=conVar.getDatosTutor(usuario) %>, esta devuelve lo siguiente:





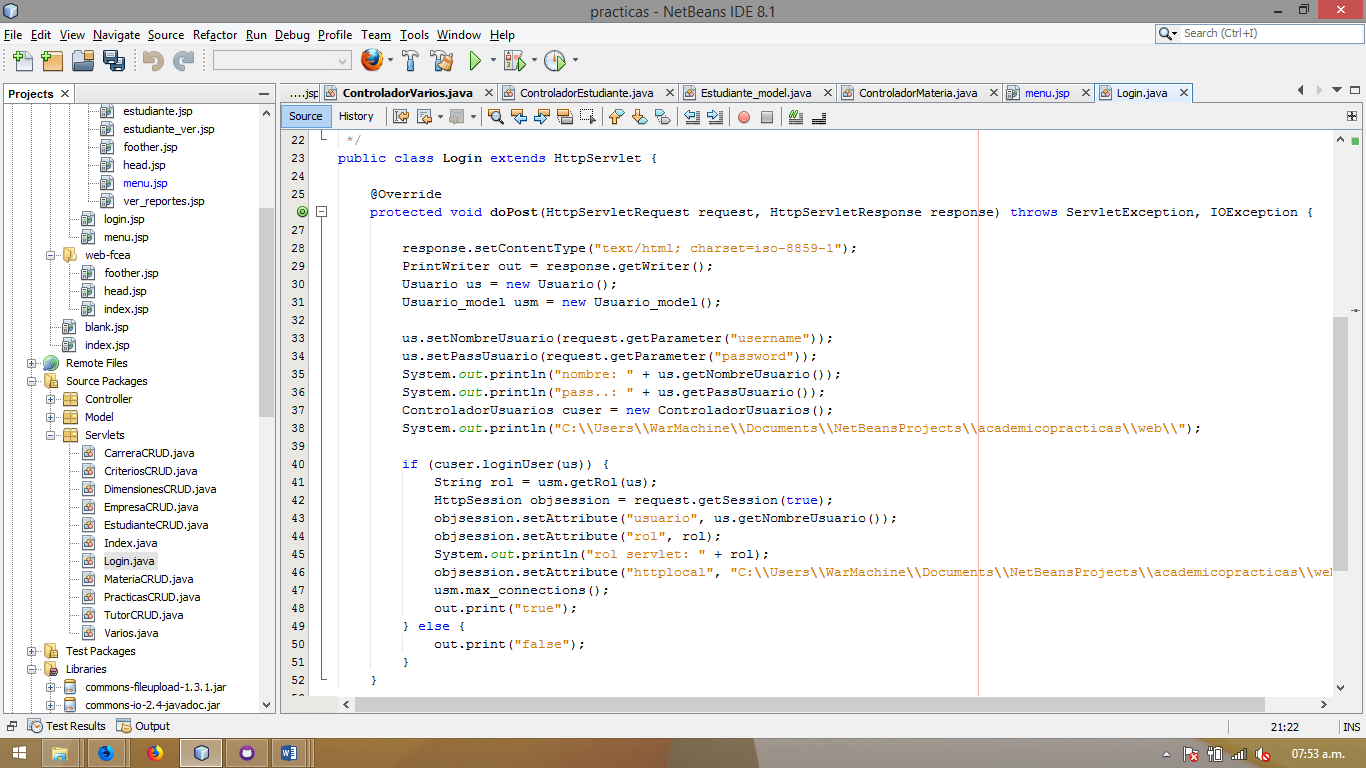
Como se puede observar hay más código HTML y consulta a la base de datos

Los Controller

Como ya fue mostrado existen 2 tipos de controlador uno es de tipo clases y ya vimos que hay controladores que realizan una función de mostrar solo un valor específico y hay otros que muestran de manera más extendida con código HTML.

Los Servlets

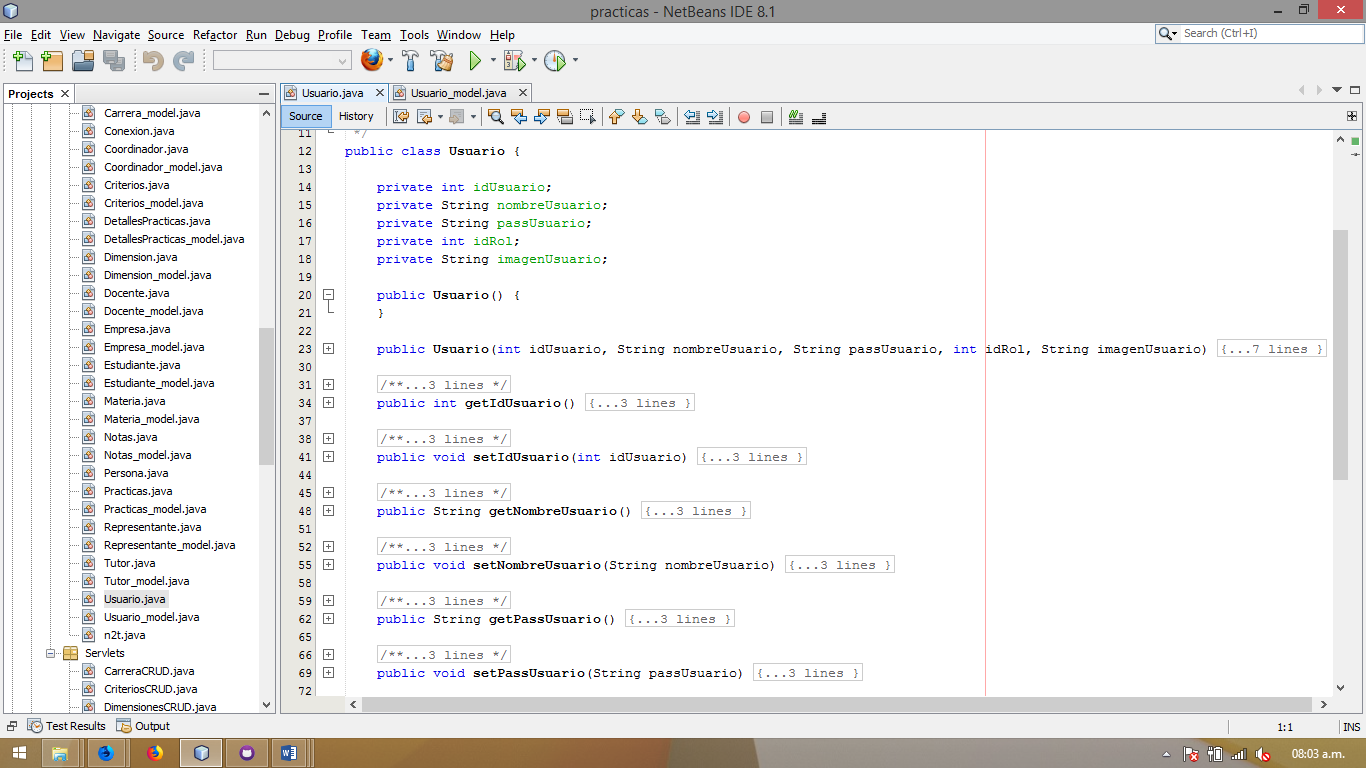
Los Servlets son los controladores por defecto que tiene JSP y trabajan de la siguiente manera:



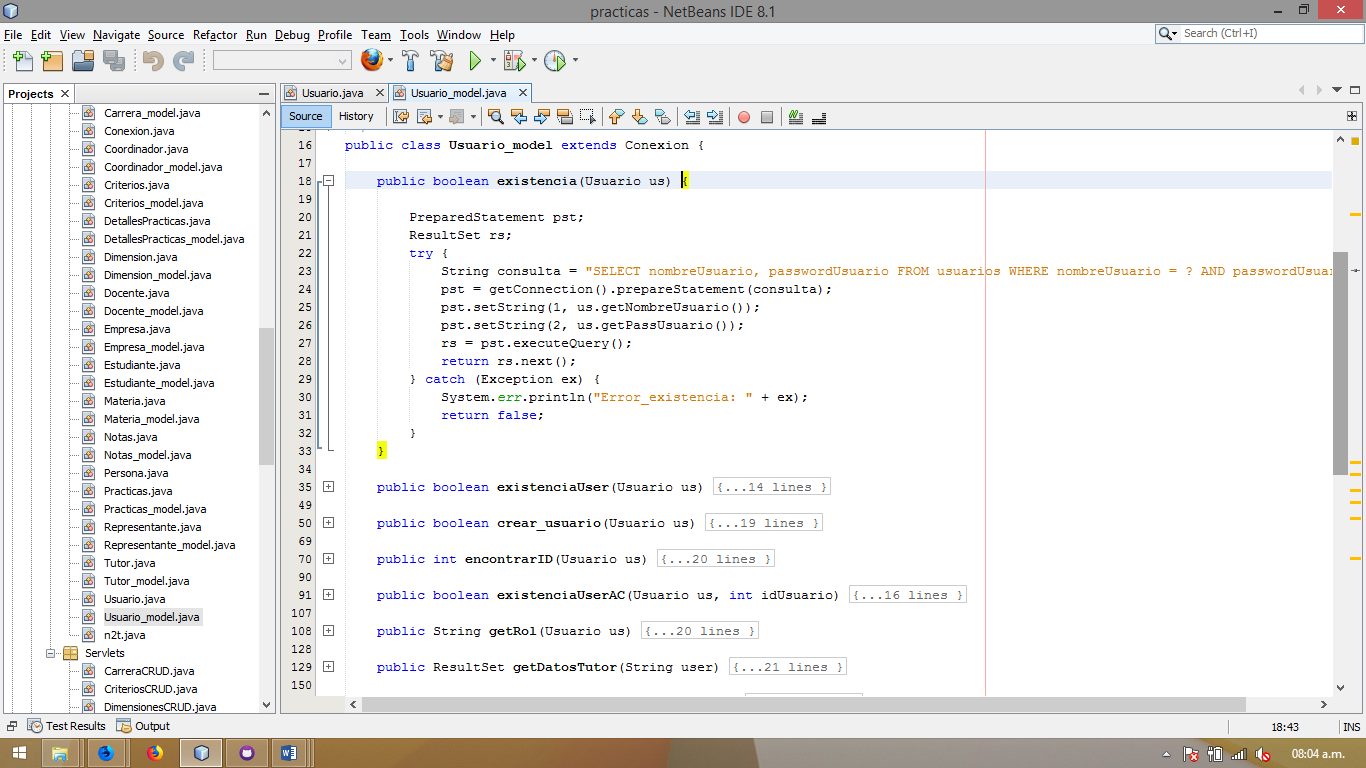
1. Reciben un conjunto de datos por el método POST puesto en el formulario previamente definido en las vistas, en el Servlet hay una función predefinida que es el doPost en el que se reciben los parámetros enviados del formulario, tiene 2 atributos uno es el request que es donde llegan todos los parámetros y el otro es el response que es donde se imprimen las respuestas del servidor.
2. El request.getParameter(“”); son la manera por la que elegimos del montón cuales serán la variables obtenidas del POST.
3. Una vez obtenidos todos los parámetros consulto mediante mi clase de Controlador el cual envía la solicitud al modelo y realiza la consulta a la base de datos lo que se necesita saber.
4. Y cuando el modelo responde realizo la validación necesaria en este caso para el login y devuelve un valor true o false al JS que esta realizando la solicitud.

El Modelo

Dentro del modelo hay 2 tipos en el que se estructuran los objetos y en el que se declaran las consultas que realizare a la base de datos.



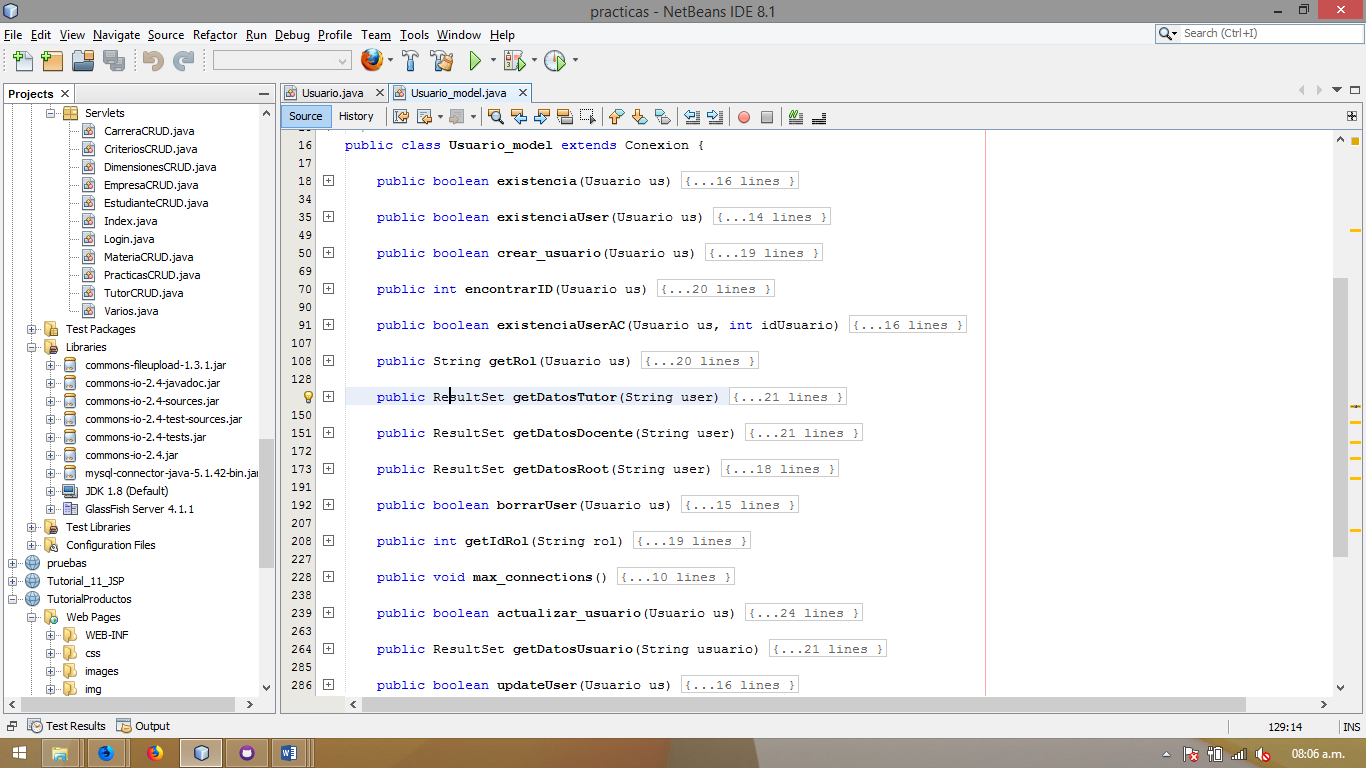
1. Ahí podemos observar las sus atributos y las clases tanto como las geters y seters como las constructoras que tendrá esa clase.



1. Y luego tenemos la clase donde se realizan las consultas a la base de datos de la siguiente manera.

Las Librerías

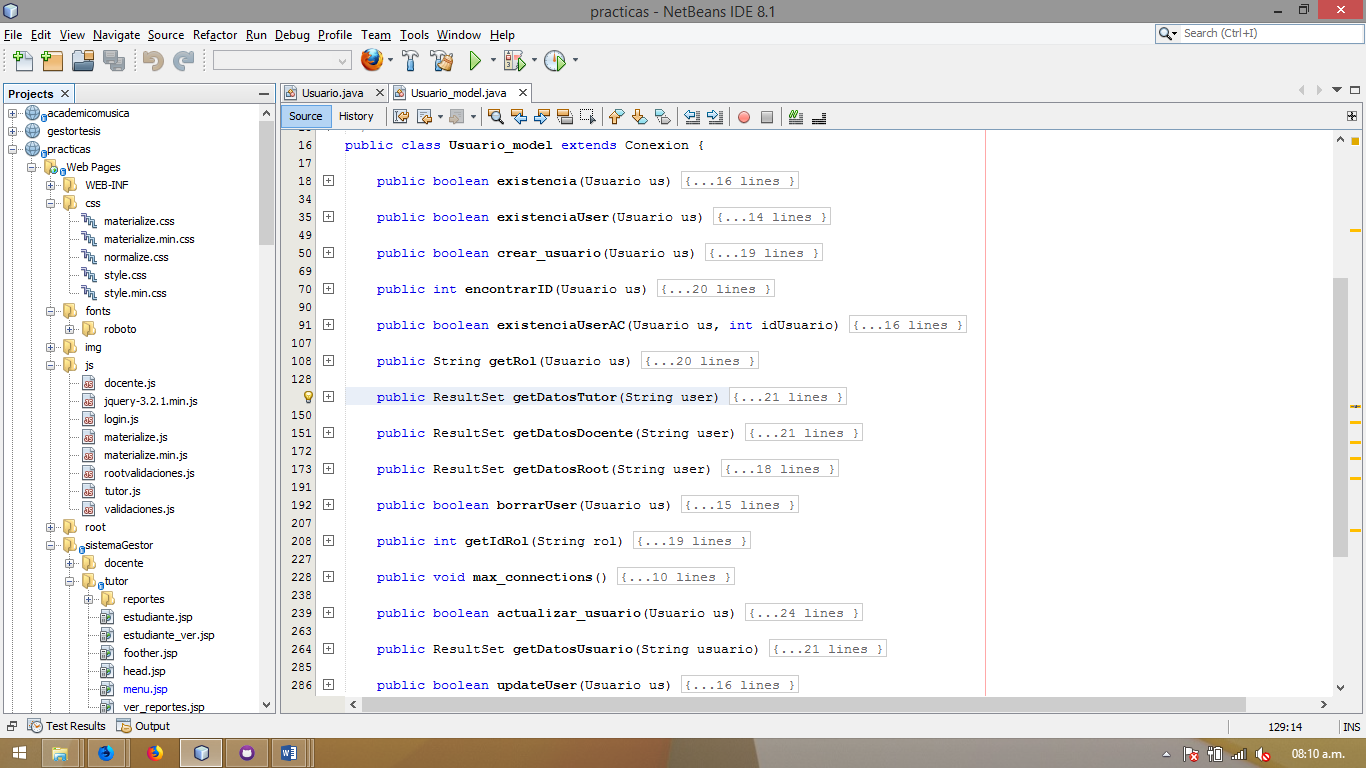
Las librerías que utilizo son las siguientes:



1. Las COMMONS sirven para realizar la subida de imágenes o archivos a la base de datos.
2. La de MYSQL-CONNECTOR-JAVA-5.1.42-BIN.JAR sirve para realizar la conexión a la base de datos en este caso sirve para MySql.
3. Y luego están las librerías de fábrica de java y la de Glassfish (que es el servidor local que estoy utilizando).

Paquetes Extras

Algunos paquetes extras que estoy utilizando son los sigtes:



CSS:

En las hojas de estilo estoy desarrollando con Materialize que me pareció muy bueno en sus diseños del CSS, al igual que sus fonts por defecto roboto.

JS:

Los JS son los que trae por defecto Materialize más algunos que yo cree para realizar las validaciones de mis formularios. Al igual que también el Jquery que está en su última versión.