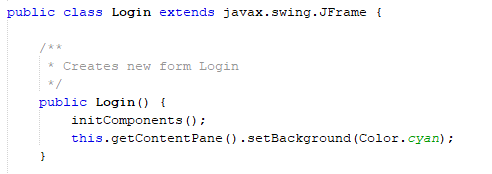
MANUAL TÉCNICO LOGIN

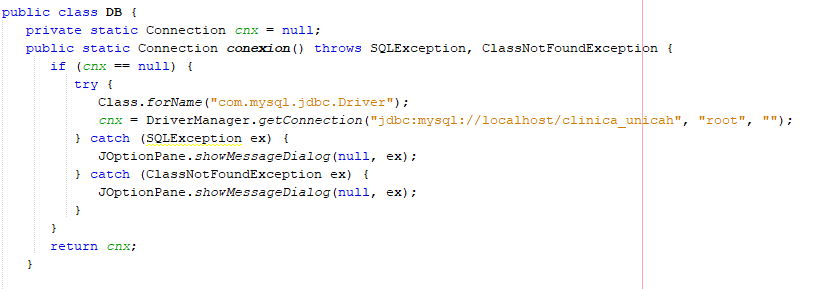


Esta es la primera parte del módulo login dónde se describirá cómo funciona el mismo. En la captura de arriba se puede ver que se define una clase con nombre login y se establece el color de fondo a un color cyan.



En el manual de usuario se puede ver que tenemos un botón de ingresar. Al darle clic pasa toda esta secuencia de código. Usamos un try catch al usar el login con la base de datos (mySqL) porque cuando se llama a la base puede llegar a ocurrir una excepción SQL si llega haber algún problema de la base de datos.

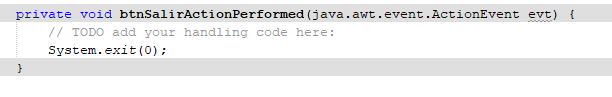
Primero, revisamos si los campos del usuario y contraseña están vacíos, si ocurre se mostrará un mensaje que diga “Campo Vacío Revise”. Si todo está bien, pasa al siguiente if donde se llama a la base de datos para revisar si existe el usuario y si su contraseña es correctas. La decisión te tiene si es igual a uno porque ese se consideró como un valor ideal para la decisión de poder entrar al sistema. Si se devuelve un uno, entonces se mostrará “Bienvenido” se cerrará la pantalla del login y se abrirá el módulo principal del sistema. Si la decisión devuelve un cero saltará al último else donde dirá “Usuario o Contraseña Incorrecto”.



Cuando se llama a DB.Login lo primero que esa función hará es ir a la función de conectar a la base. La imagen de arriba muestra cómo se conecta JDBC a MySql.



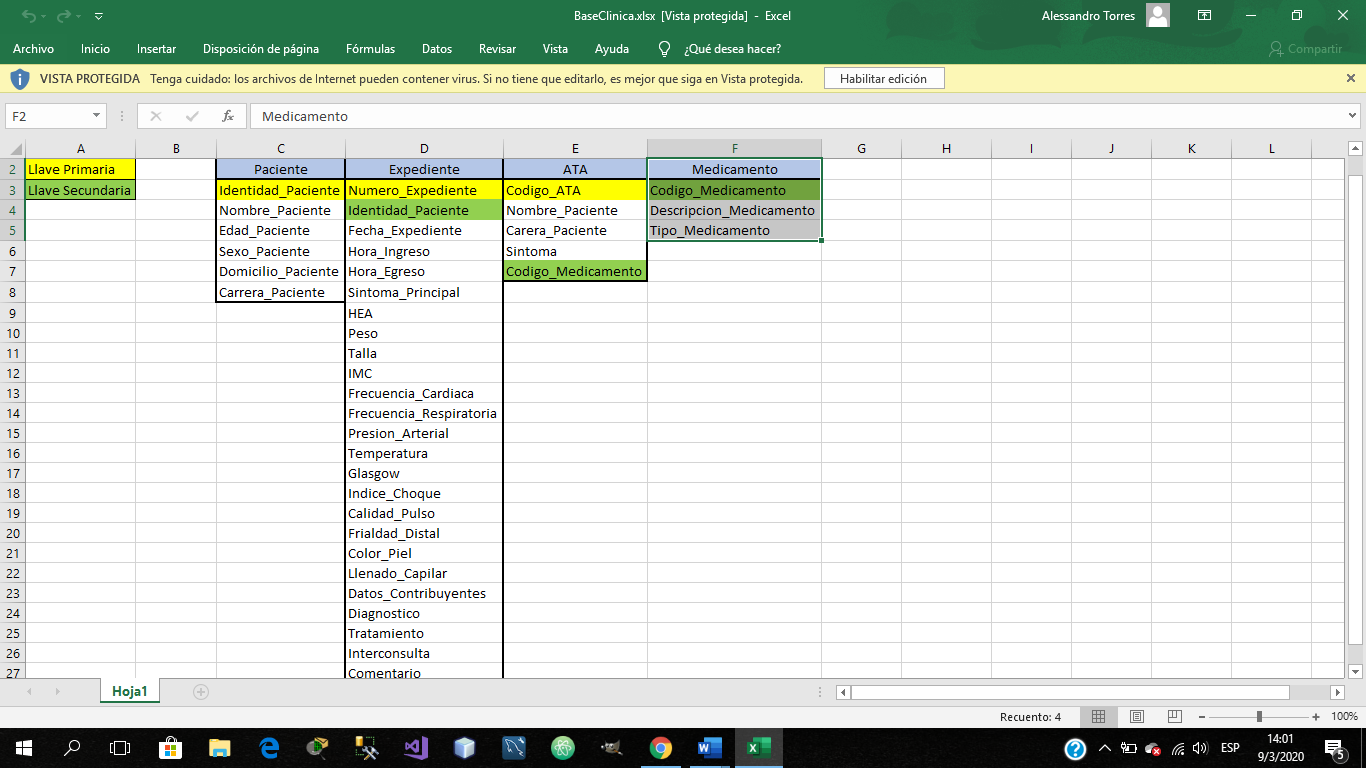
Esta es la función DB.Login donde anteriormente se dijo que primero va a conectar la base de datos y devolverá un cero o un uno. Esta función ocupa de parámetros uno para el usuario y otro para la contraseña. Para poder ejecutar una consulta ocupamos utilizar statement, resultset y una propiedad del resultset que es next. Statement dice que ejecutaremos la consulta. Resultset ejecuta esa consulta. Se utiliza el next para poder traer datos desde la base. Si todo funciona perfectamente, tenemos una variable “res” que es la que devuelve esta función. Al antes de retornar se cierra el resultset, statement y la conexión a la base de datos.



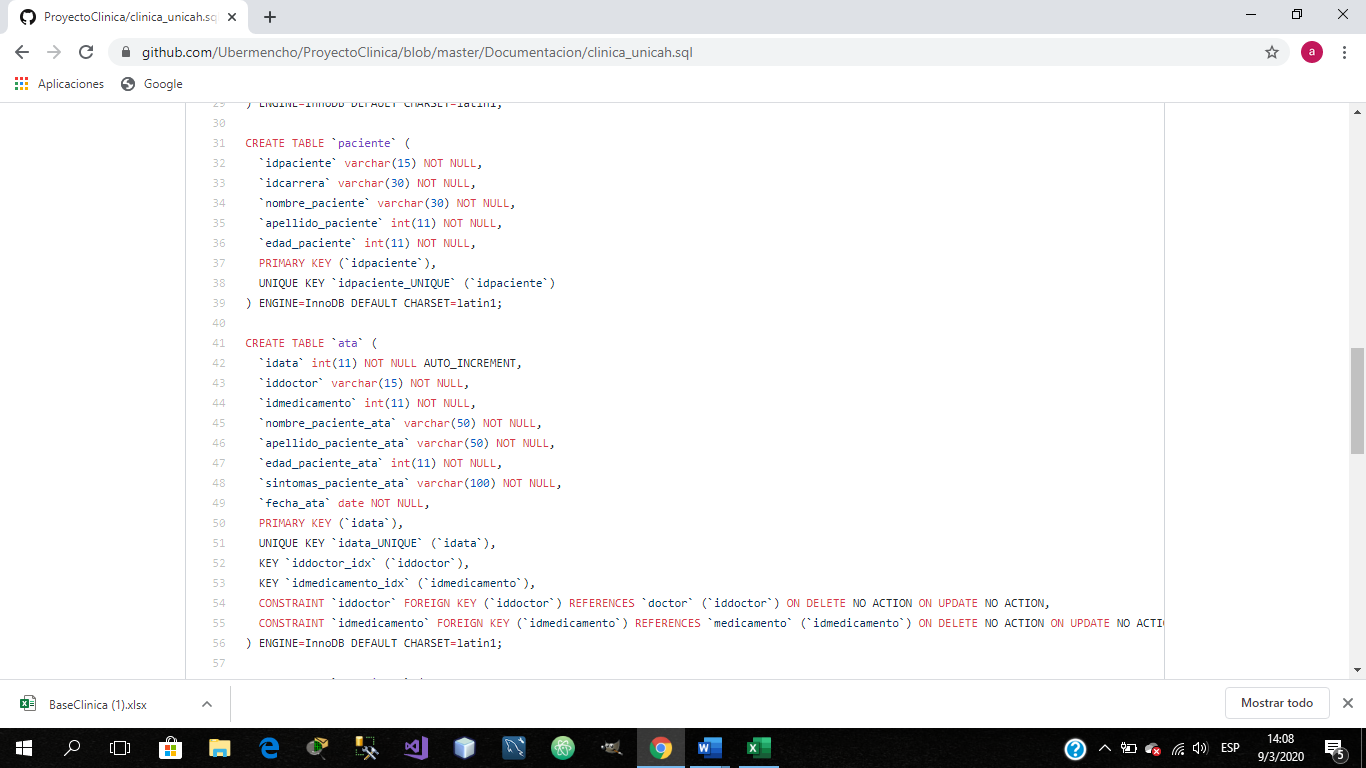
En el formulario de Login también hay un botón salir dónde se cierra el programa con un system.exit (0).

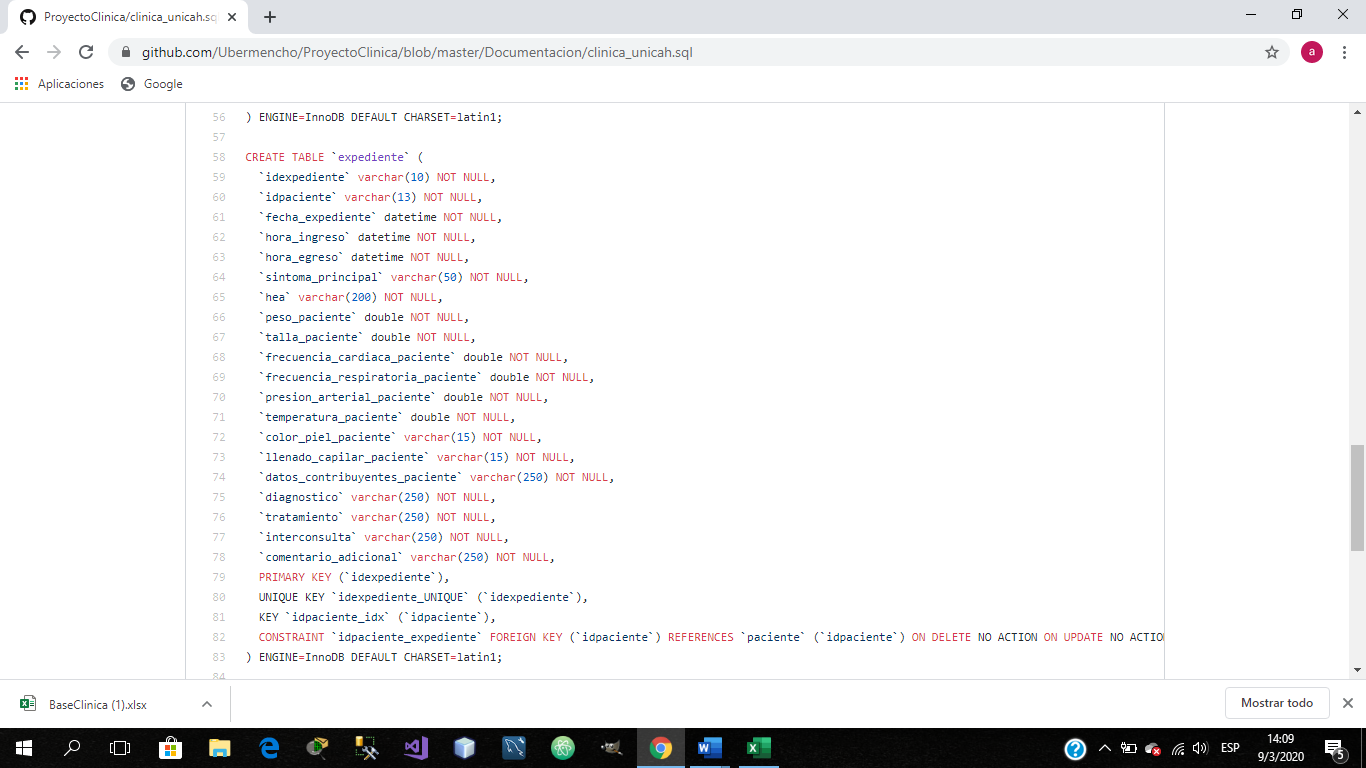
MANUAL TECNICO DE INGRESO ATA

El ATA es la recolección de datos del paciente, para que con esta información el medico pueda recetar o elaborar una prescripción medica y con esto poder curar o rebajar los síntomas del paciente. Este se elabora para beneficio de la universidad, a modo que todo se vuelva un poco mas tecnológico, y evitar mucho papeleo, que puede llegar a hacer tedioso. La base de datos, esta constituida de la siguiente manera:



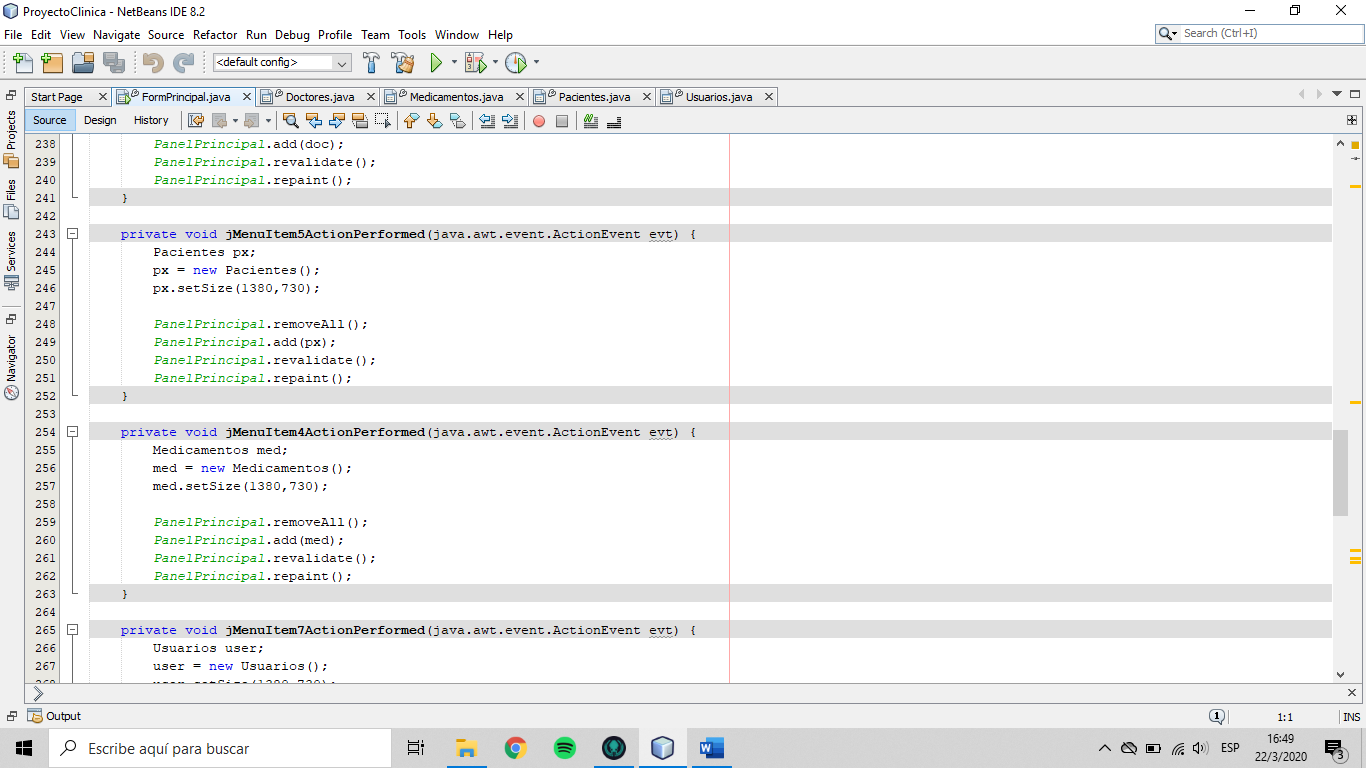
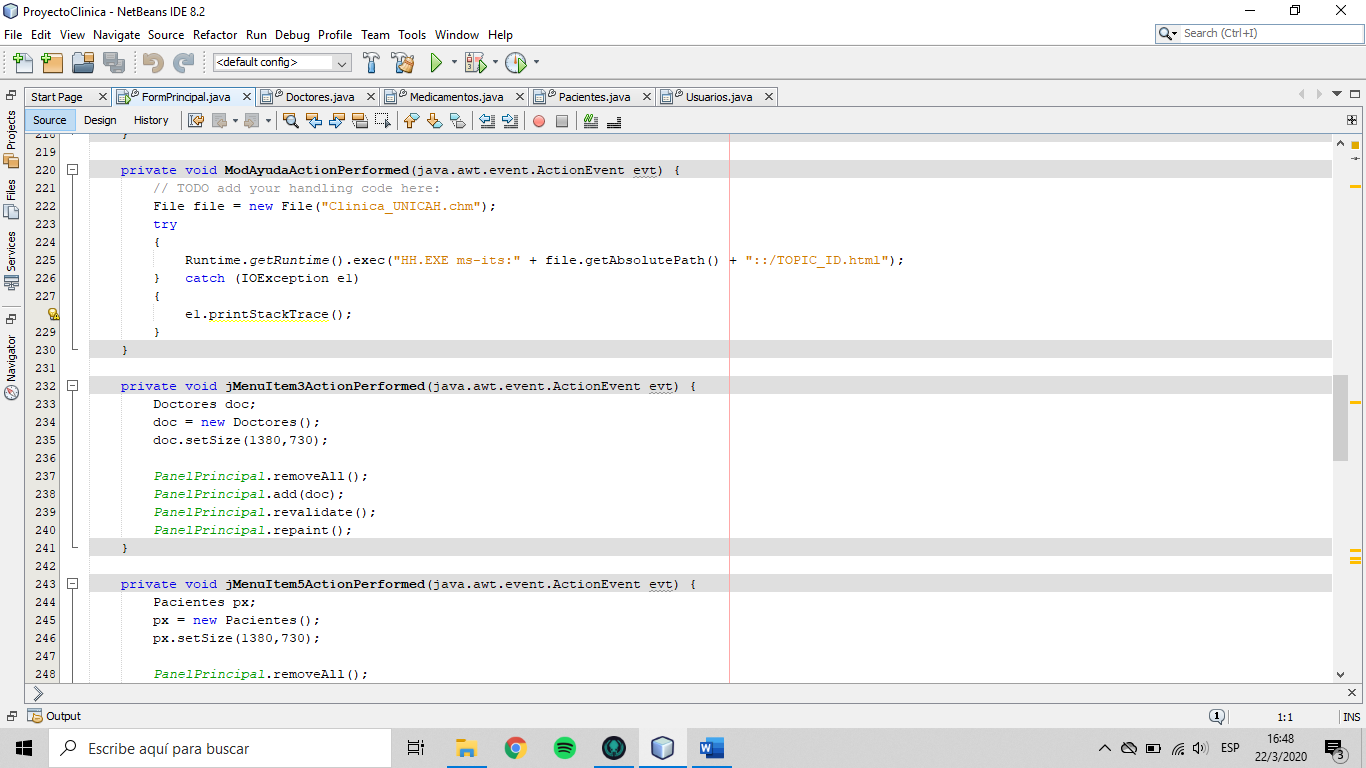
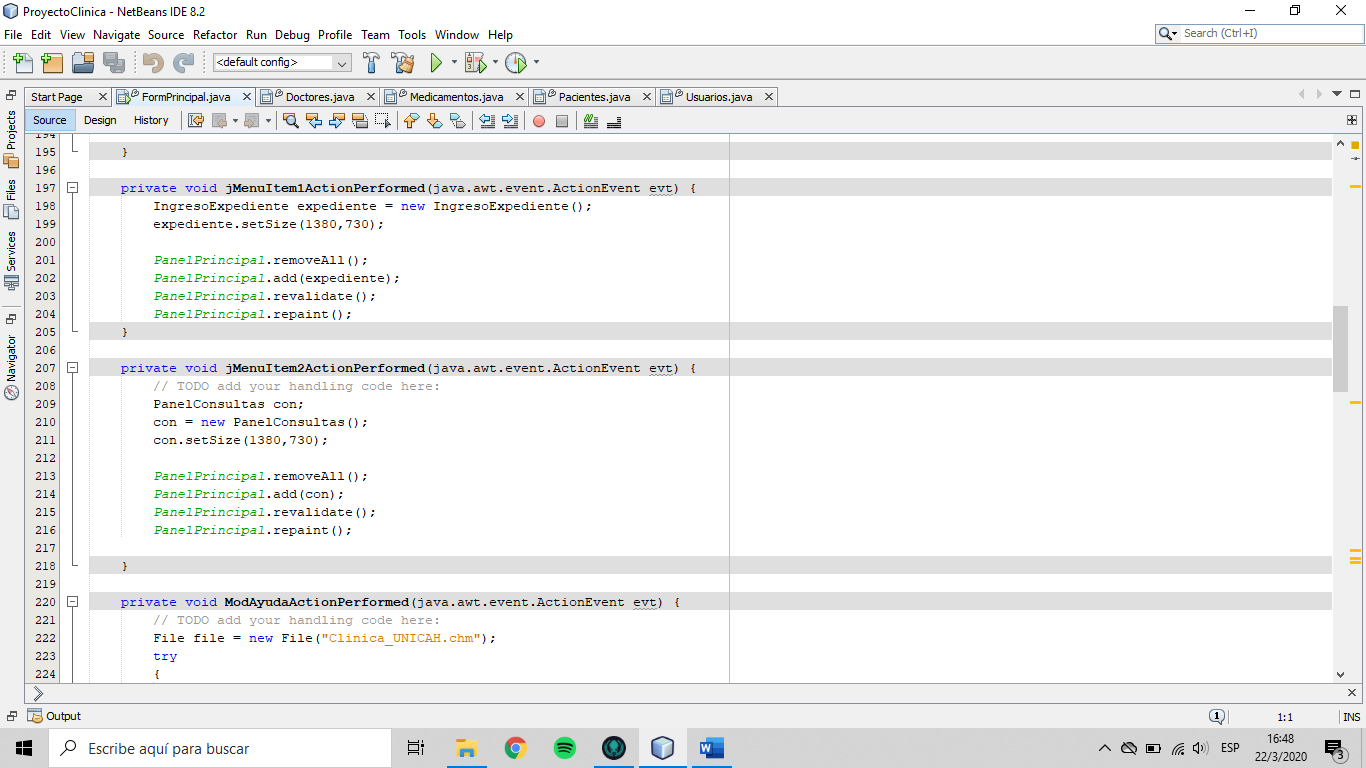
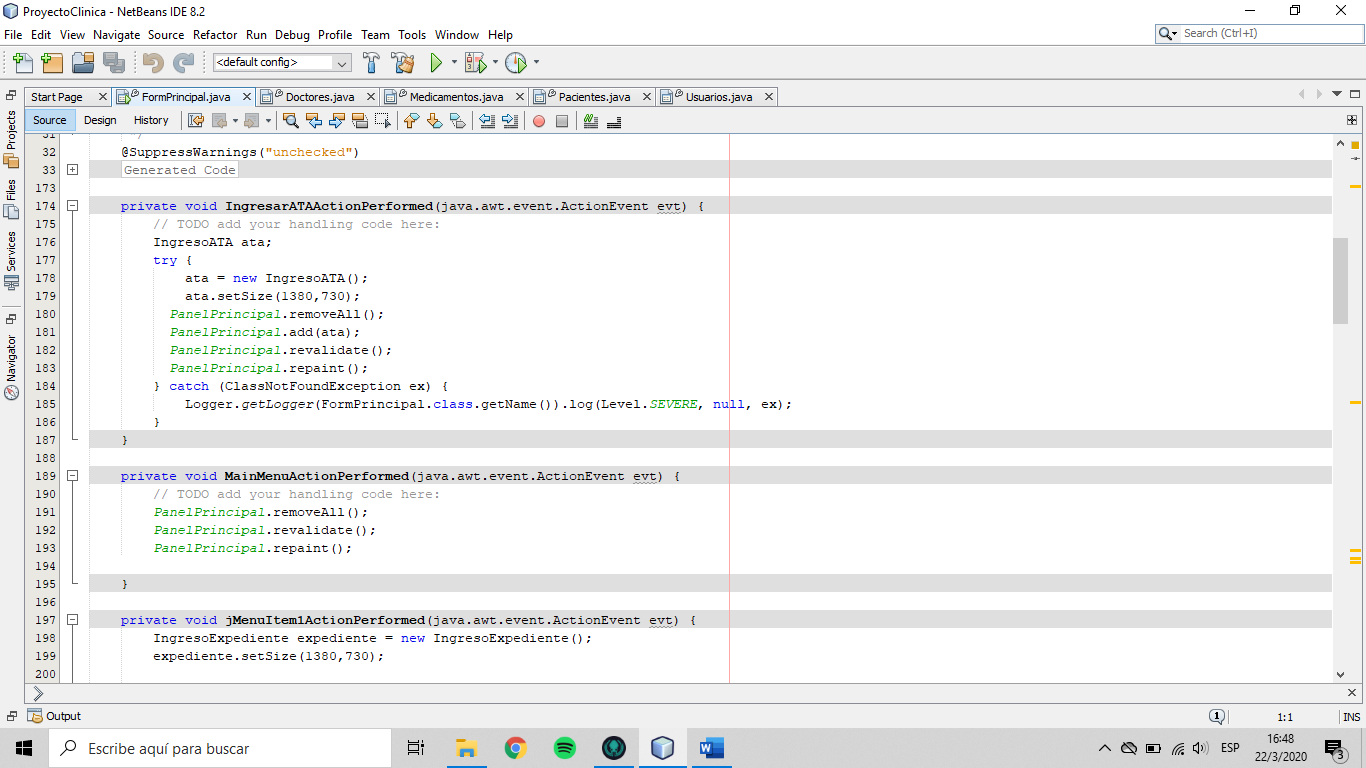
El ATA esta compuesta en su mayor parte por la tabla paciente y la tabla ATA. Se compone de ambas porque en su particularidad el ATA solo es el registro del paciente, tomar sus diagnosticos actuales, y elaborar una prescripción medica que ayude a alivianar o erradicar completamente los síntomas de el paciente. Los campos de la base de datos en sql están constituidos de la siguiente manera:

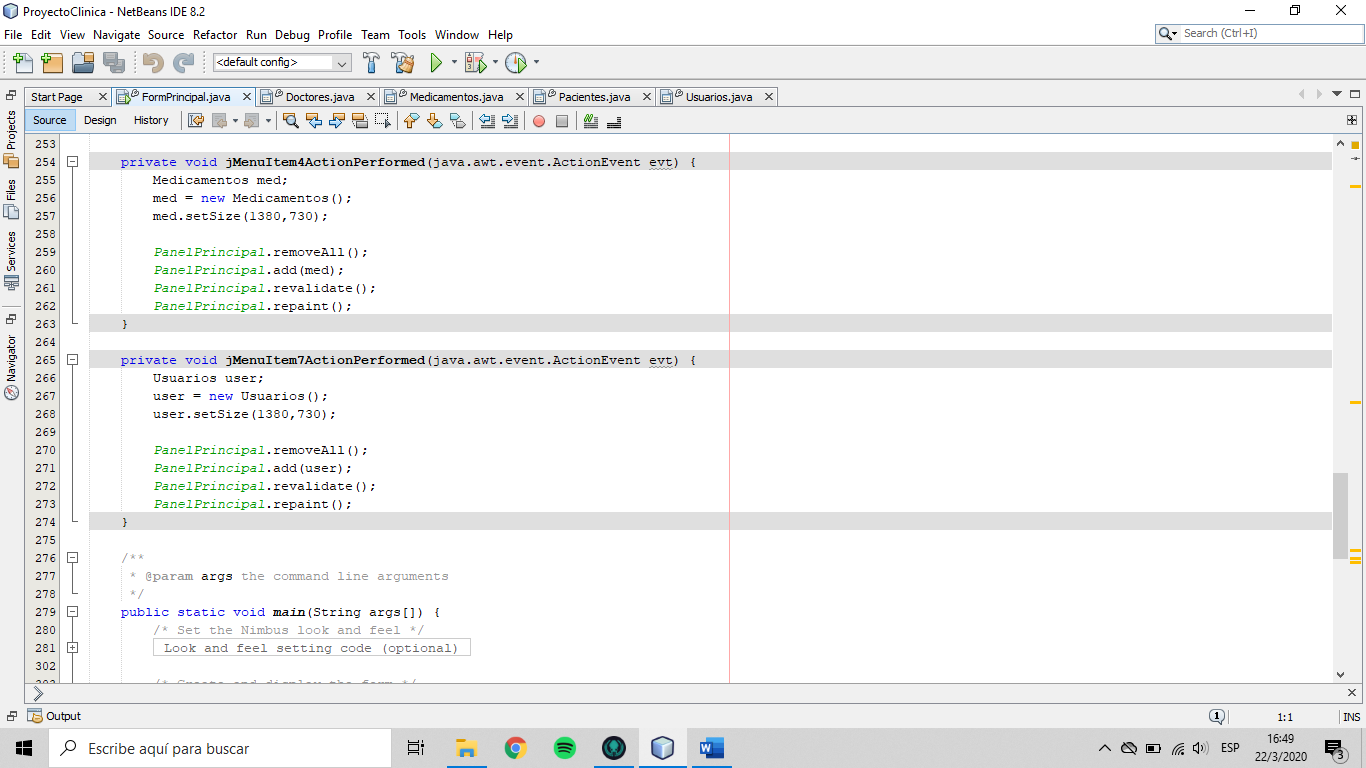




***Mantenimiento***

**FormPrincipal**

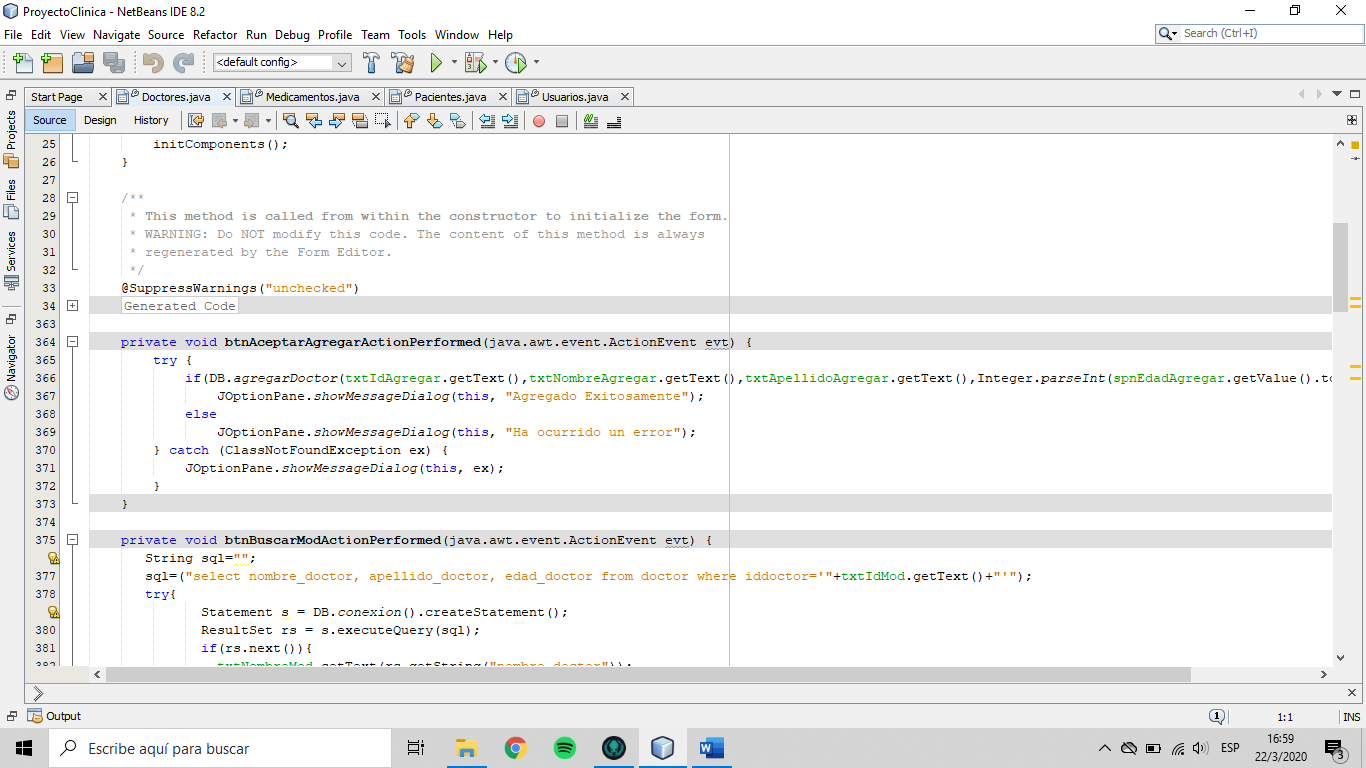




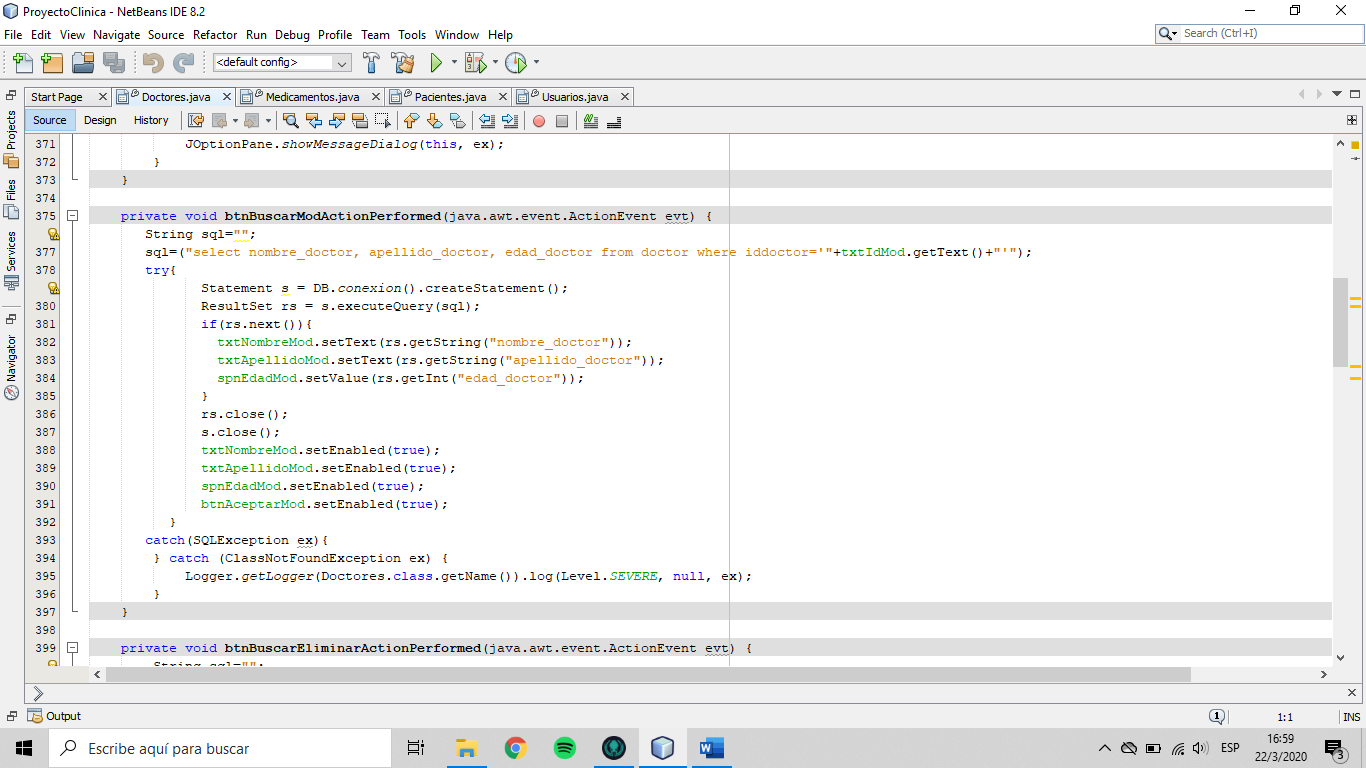
Básicamente toda esta sección del código se utiliza para llamar a los formularios y que estos se abran con las mismas especificaciones asignadas al momento de seleccionar un mantenimiento.

**Doctores**

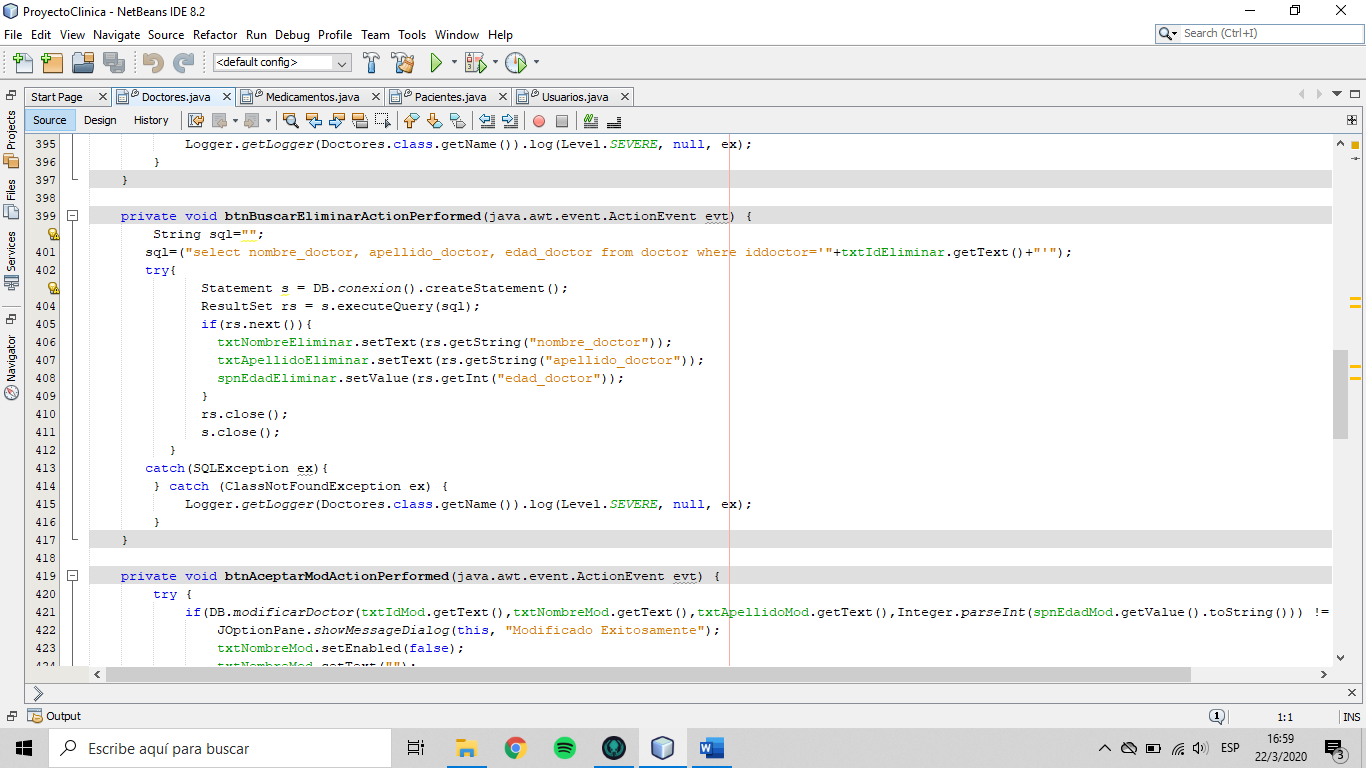
En la primera línea después del try siempre se manda a llamar los campos que se necesitan de la base de datos.



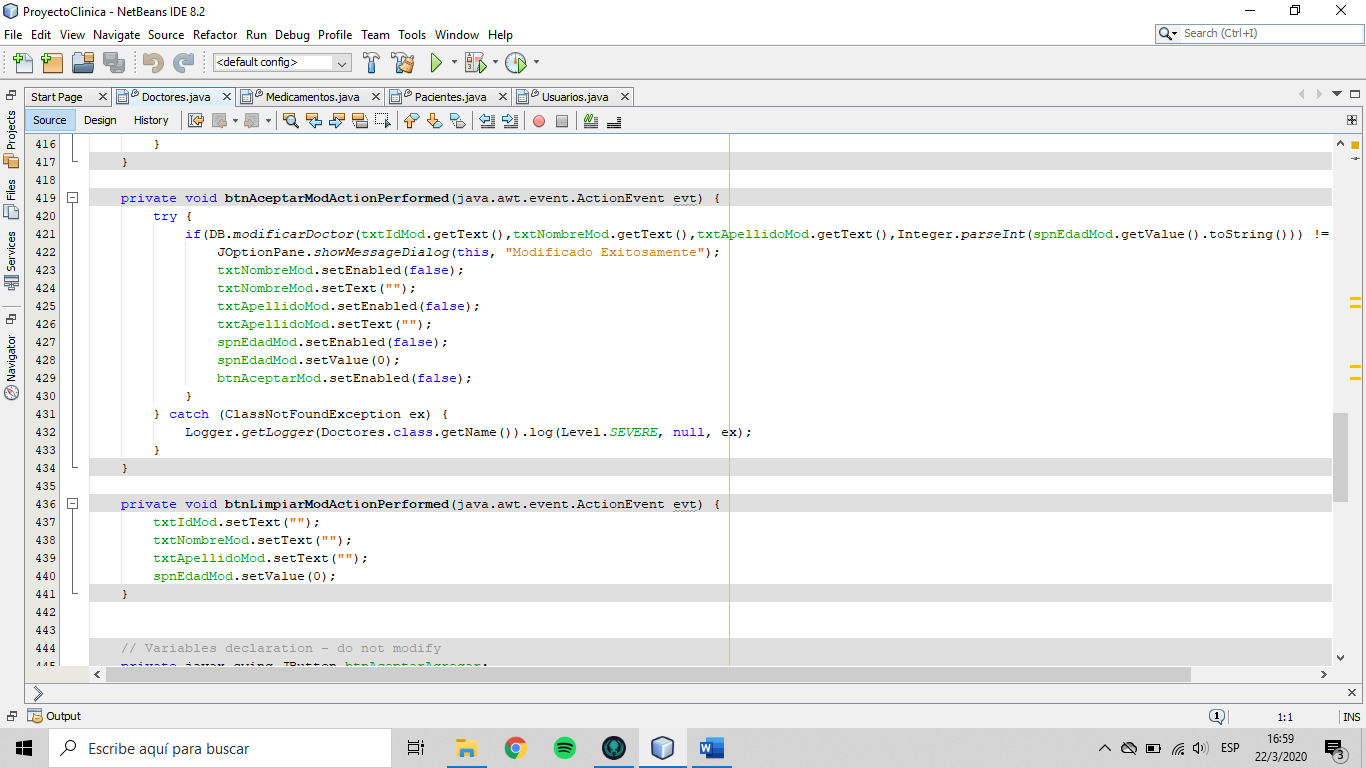
Esta parte del código lo que hace es que guarda en la base de datos los datos ingresados.



En el btnBuscar busca en la base de por medio del numero de identidad y muestra los datos que corresponden y luego guarda las modificaciones realizadas para dichos datos.

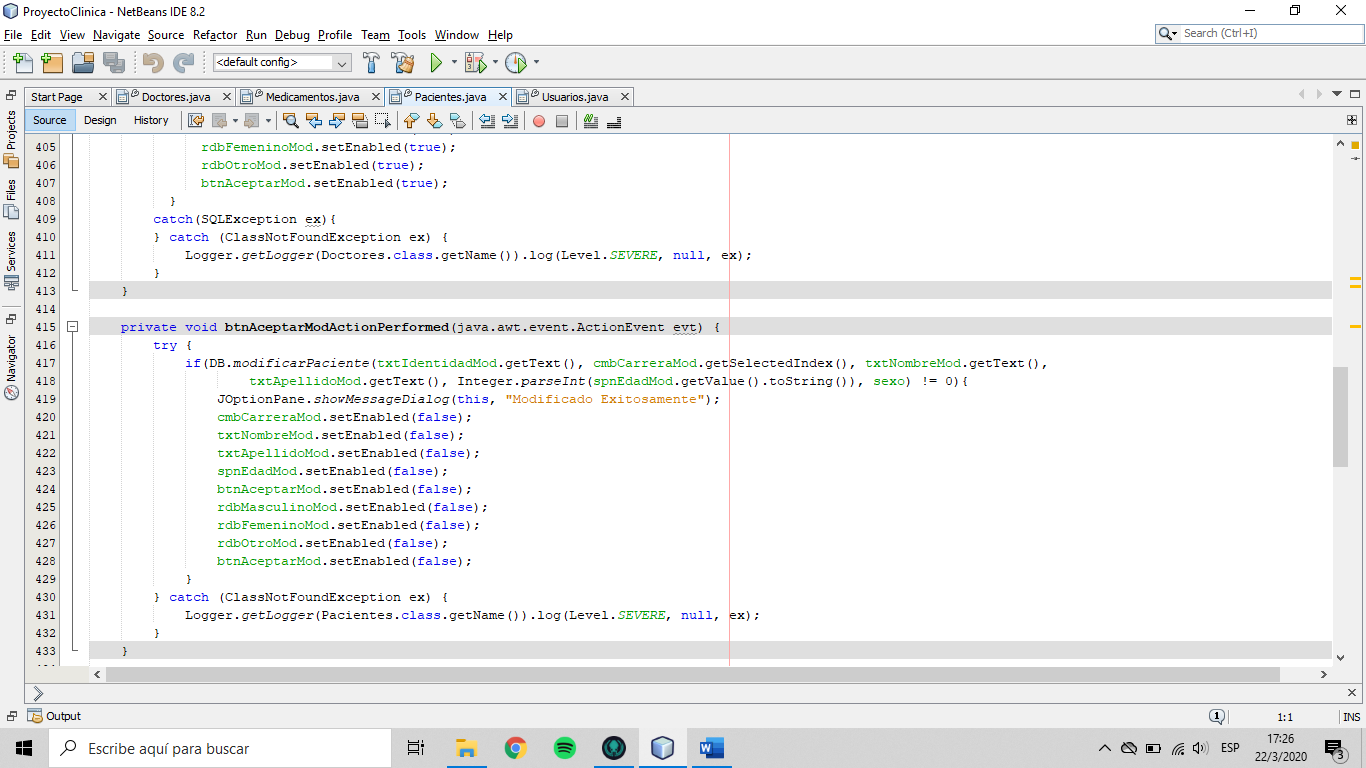
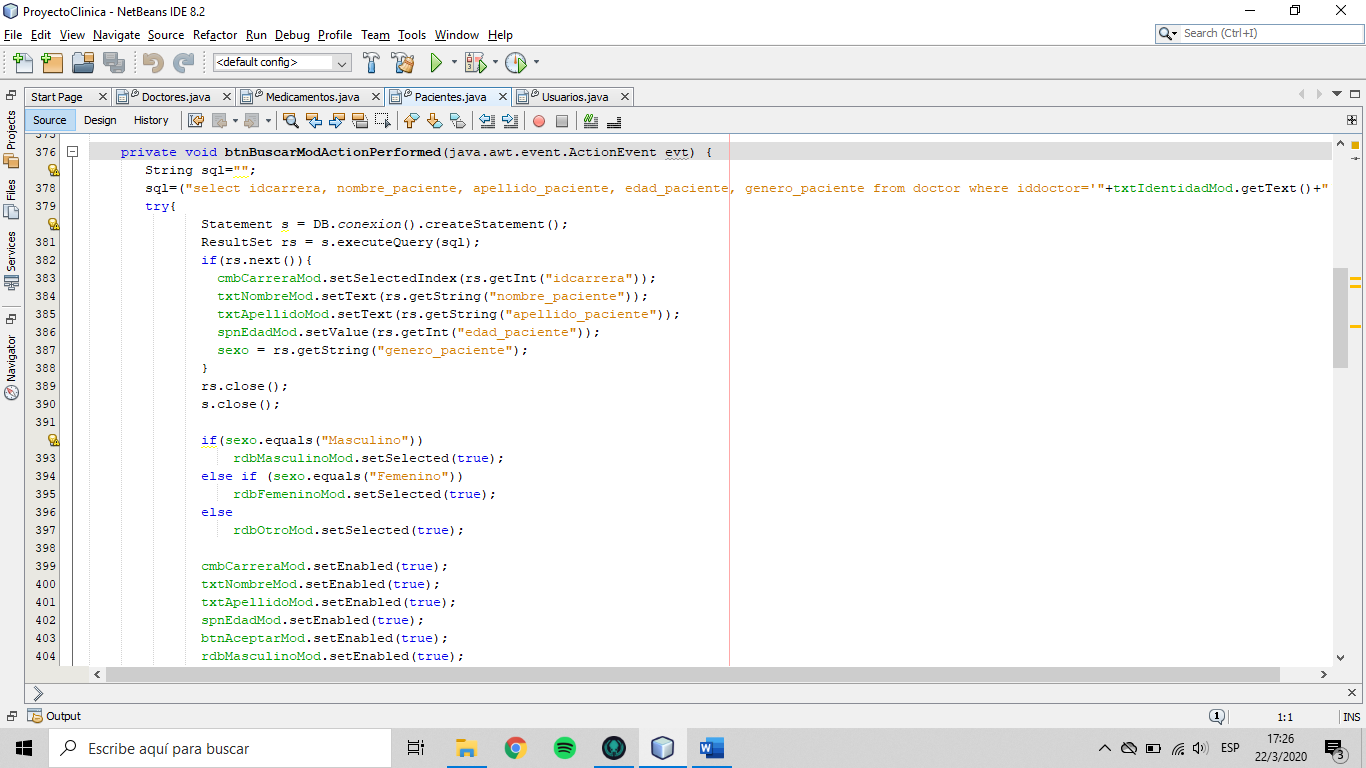


Acá se busca de igual forma mediante el numero de identidad para posteriormente eliminar los datos del sistema.



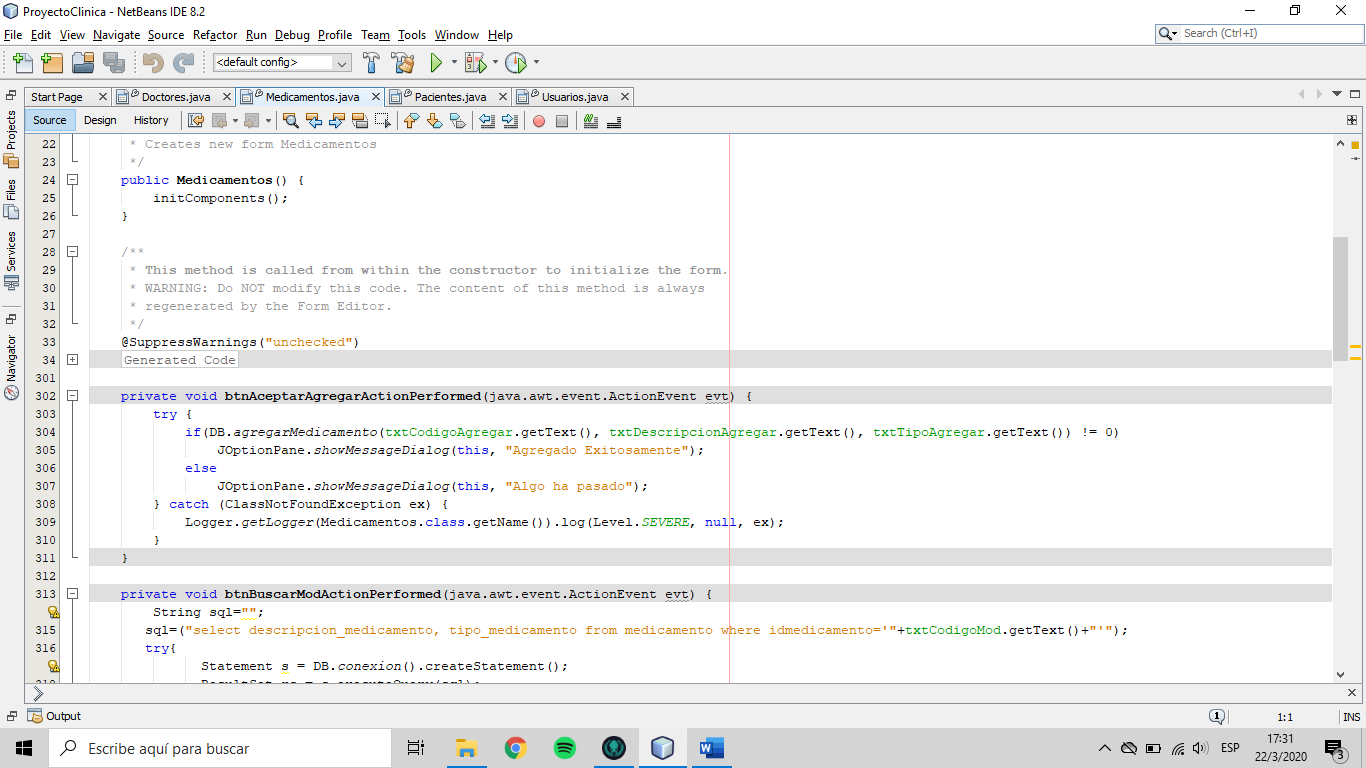
El btnAceptarMod guarda los datos modificados y el btnlimpiar deja en blanco los campos que contiene la pantalla.

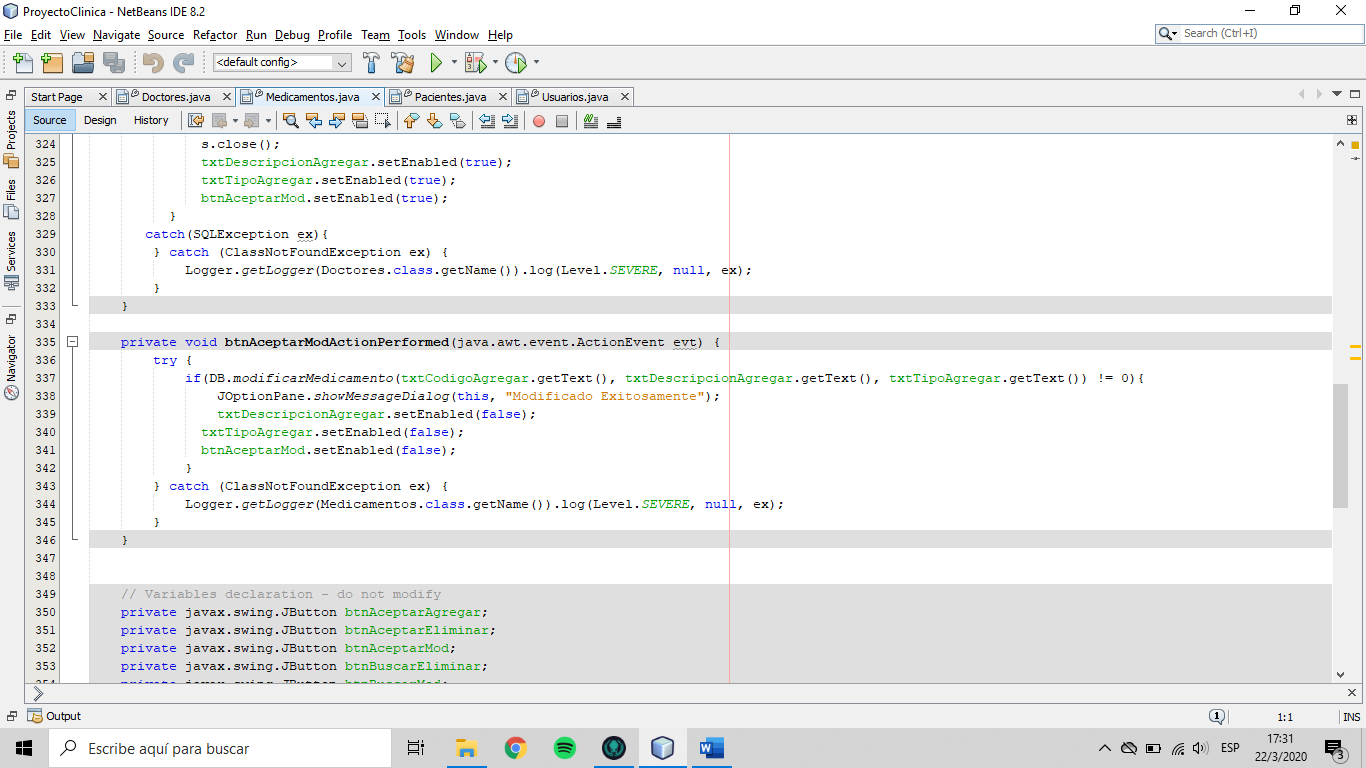
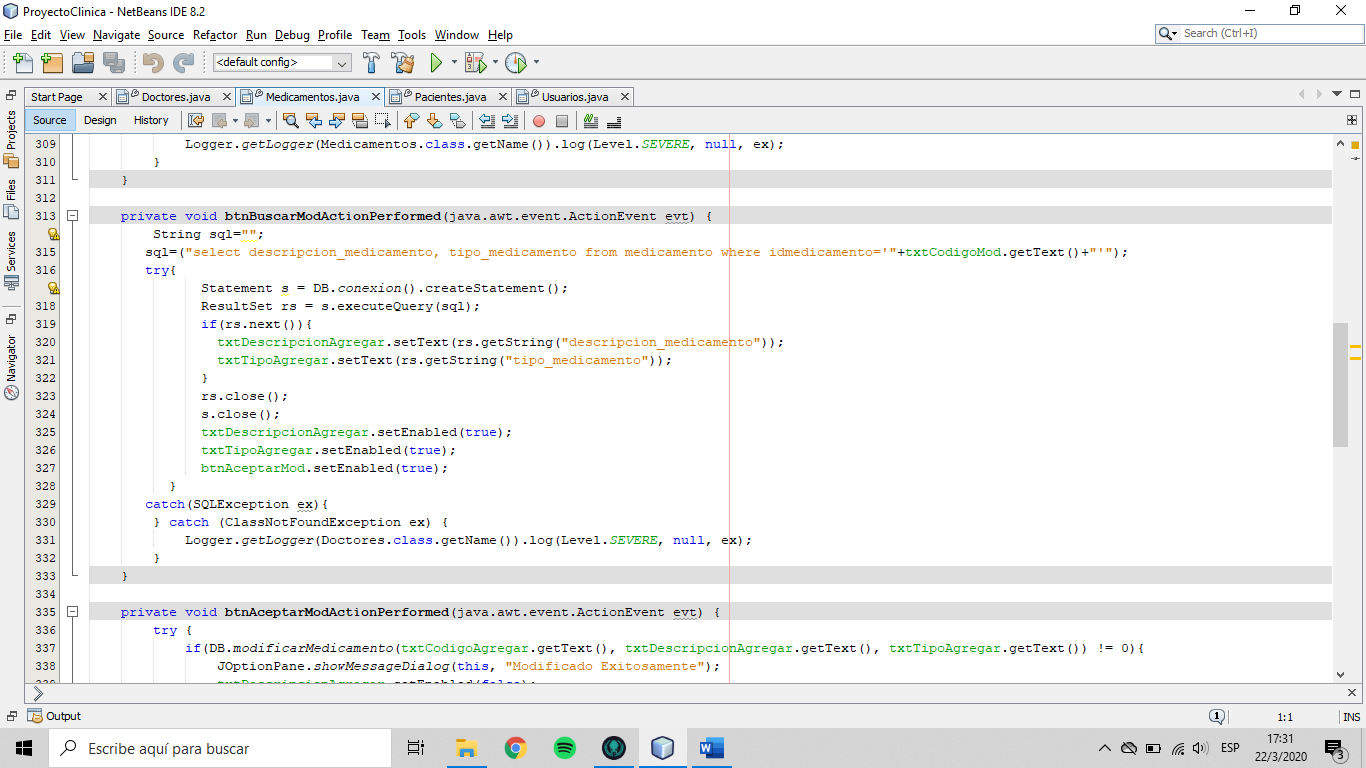
**Pacientes**



Básicamente el proceso se repite en la codificación de las demás pantallas del mantenimiento, únicamente que se cambian los datos que se desean usar desde la base de datos para que funcionen en el sistema.

**Medicamentos**





**Usuarios**

