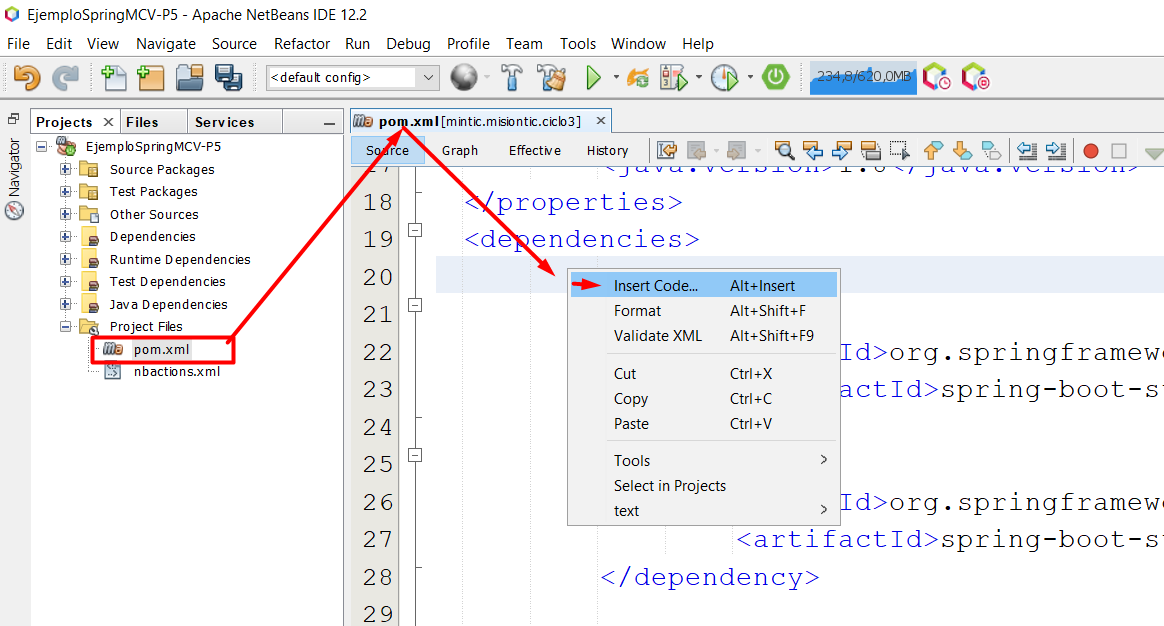
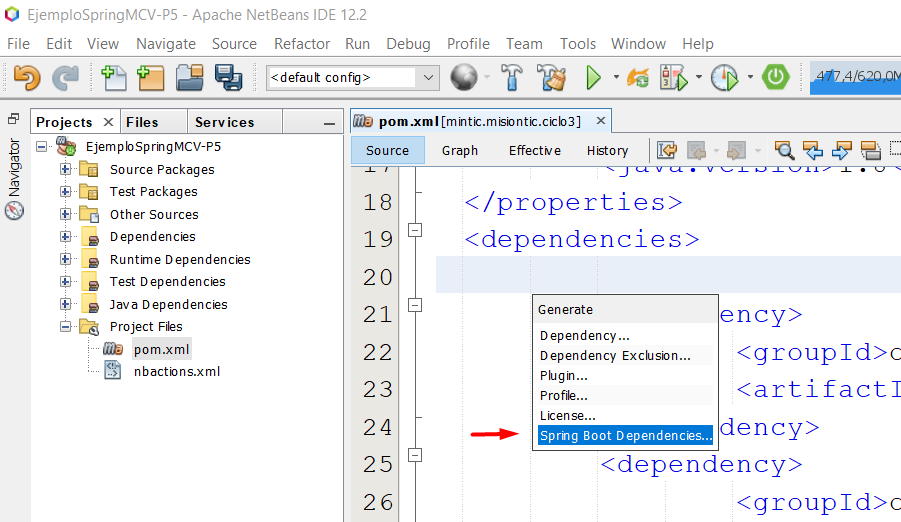
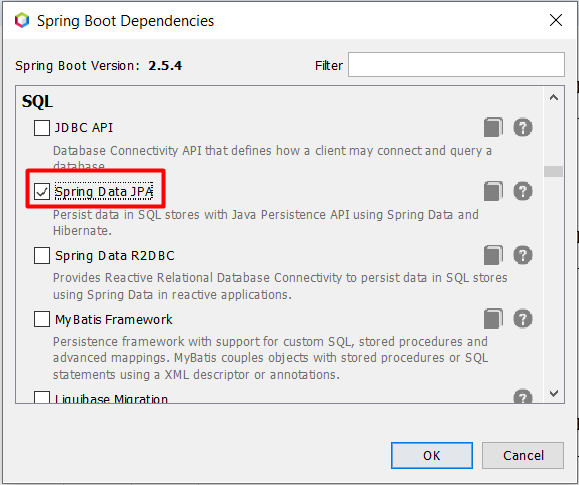
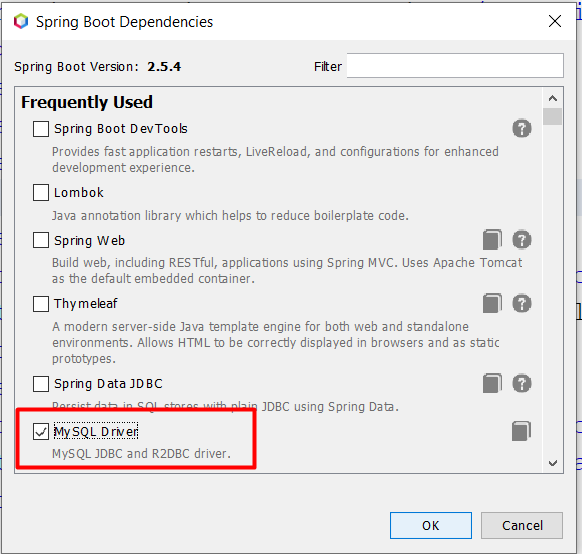
AGREGAR LAS DEPENDENCIAS NECESARIAS

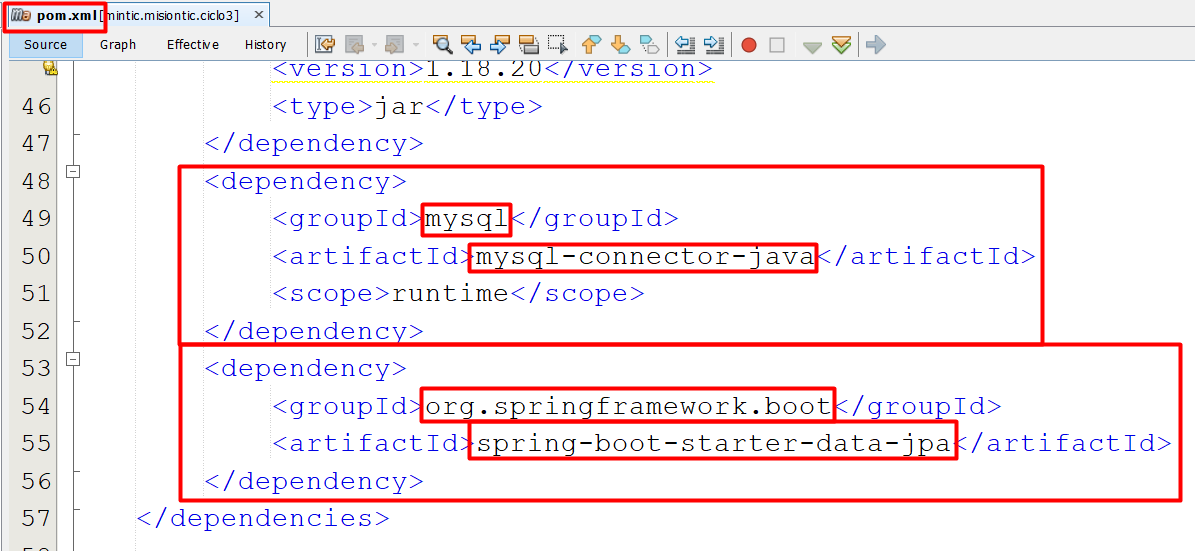




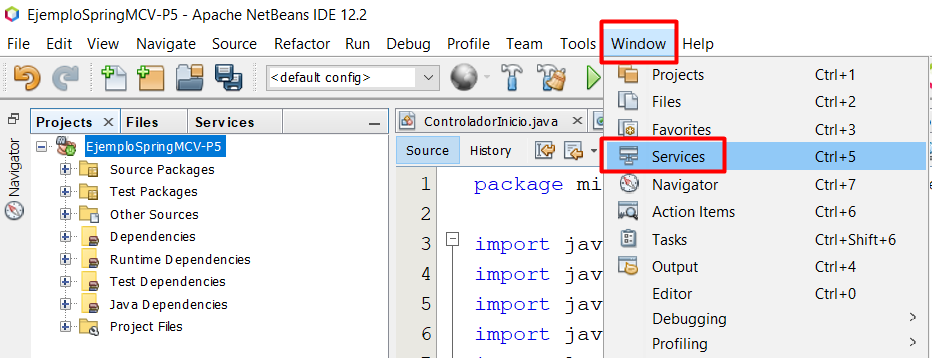
click derecho en el apartado de <dependencies> de nuestro archivo pom y buscamos el Spring Boot Dependencies..



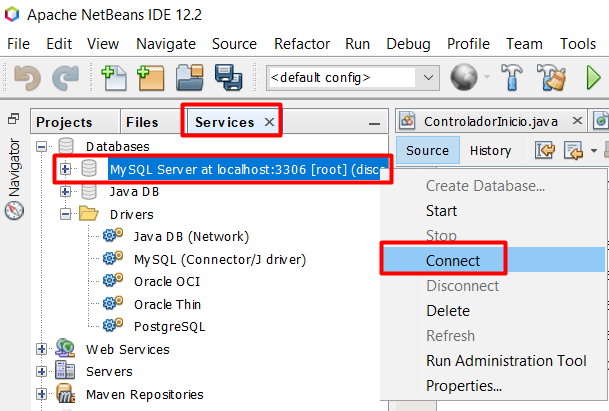
buscaremos el JPA y el MysqlDriver



confirmamos que aparezcan en nuestro archivo pom

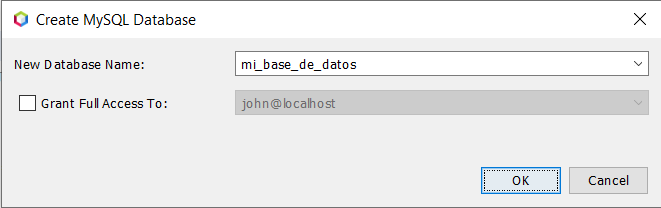
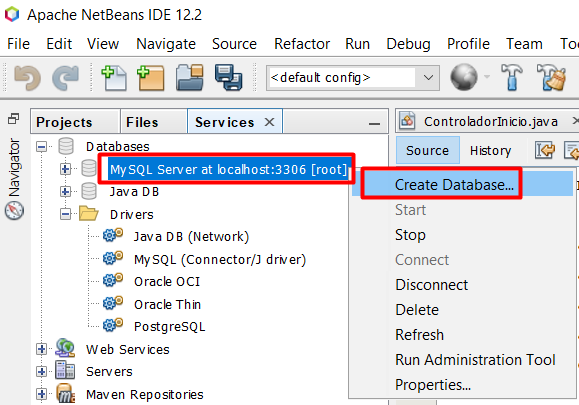


Nos dirigimos a la barra de navegación y buscamos la opcion window y click en Services

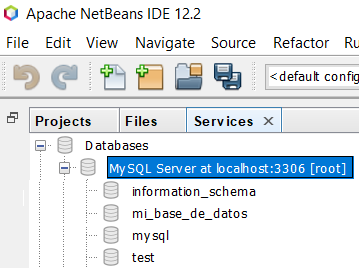


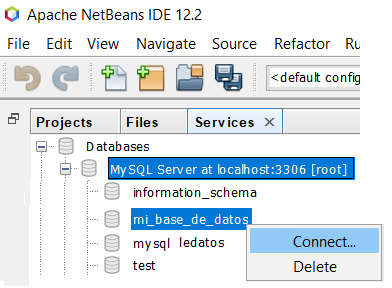
luego en Services buscamos en el apartado MySQL, click derecho y Connect

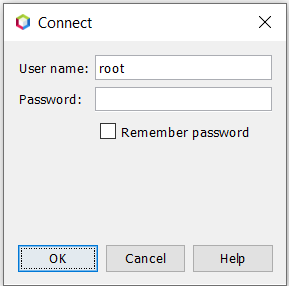
y caso de que nos pida usuario y contraseña se los pasamos, si nunca hemos configurado esto es posible que nos hagan falta los drivers de mysql en dado caso debemos dirigirnos a la página, descargarlos y cargarlos en Netbeans.



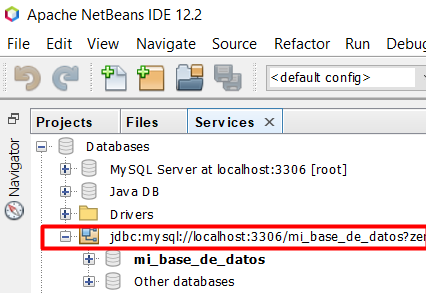
luego crearemos la base de datos con el nombre de nuestra preferencia, para este ejemplo usaremos el que se ve en la imagen.

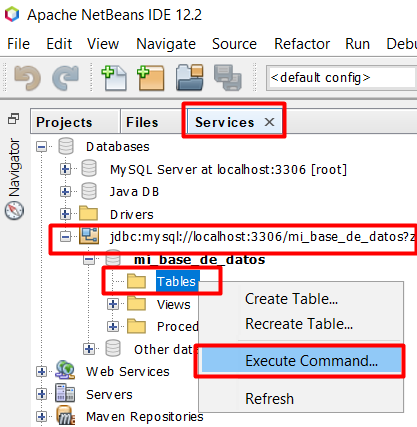




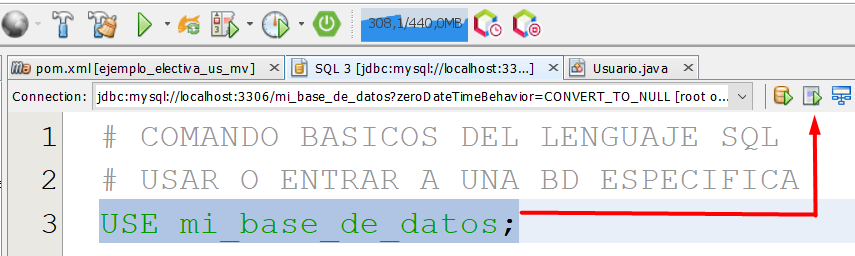


buscamos nuestra base de datos y clic derecho conectar, en caso de tener por defecto siempre será el usuario root y sin contraseña.

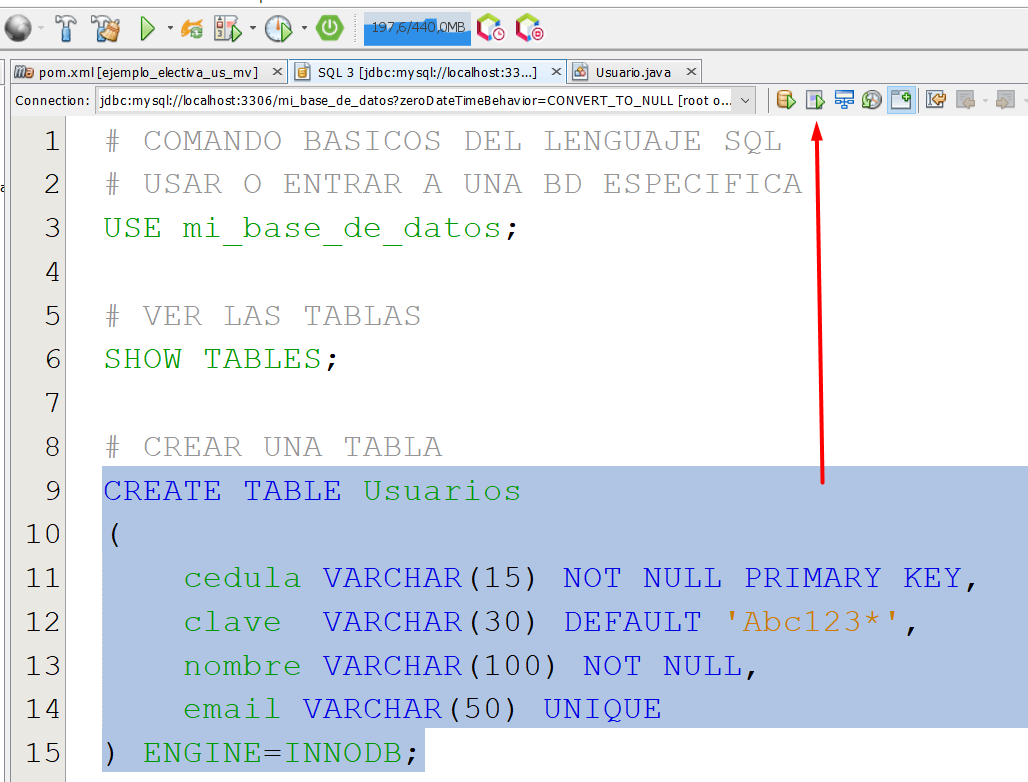




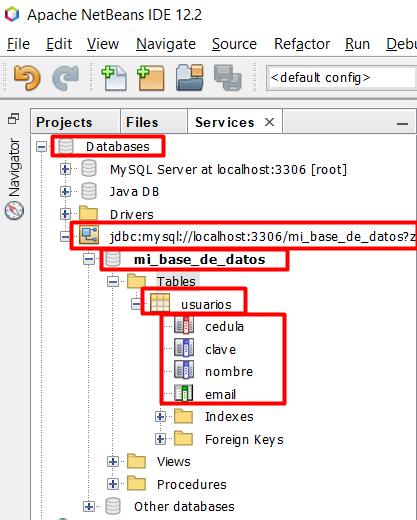
nos posicionamos sobre las carpetas de nuestra bd y execute command para abrir el editor de texto y poder ejecutar comandos sql.



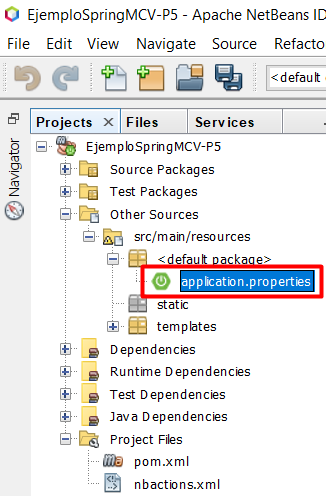
Use para especificar que vamos a usar la base de datos, seguido del nombre.



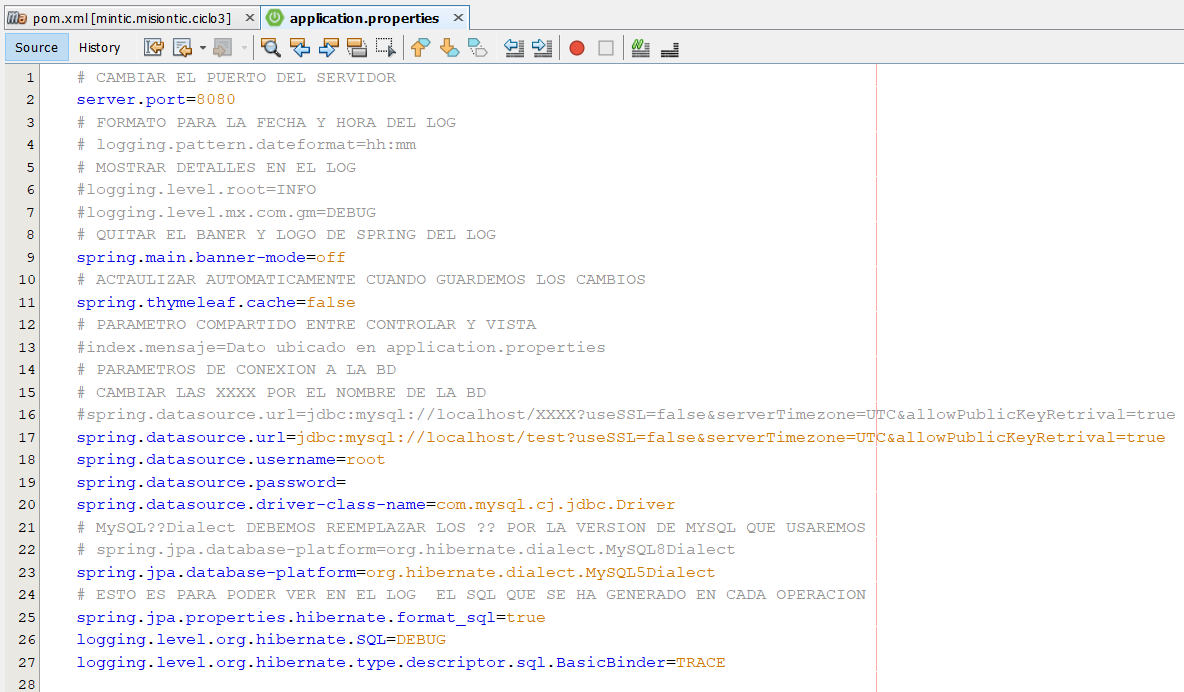
Creamos la tabla para la clase usuarios con los atributos correspondientes de la clase.



Veremos las tabla creada con sus atributos.



volvemos al apartado de project, en other sources y buscamos la application.properties.



pegaremos el código que se ve abajo, ahí se explica para qué sirve cada comando que estamos viendo

*# CAMBIAR EL PUERTO DEL SERVIDOR*

server.port=8080

*# FORMATO PARA LA FECHA Y HORA DEL LOG*

*# logging.pattern.dateformat=hh:mm*

*# MOSTRAR DETALLES EN EL LOG*

*#logging.level.root=INFO*

*#logging.level.mx.com.gm=DEBUG*

*# QUITAR EL BANNER Y LOGO DE SPRING DEL LOG*

spring.main.banner-mode=off

*# ACTUALIZAR AUTOMÁTICAMENTE CUANDO GUARDEMOS LOS CAMBIOS*

spring.thymeleaf.cache=false

*# PARÁMETRO COMPARTIDO ENTRE CONTROLAR Y VISTA*

*#index.mensaje=Dato ubicado en application.properties*

*# PARAMETROS DE CONEXION A LA BD*

*# CAMBIAR LAS XXXX POR EL NOMBRE DE LA BD*

*#spring.datasource.url=jdbc:mysql://localhost/XXXX?useSSL=false&serverTimezone=UTC&allowPublicKeyRetrival=true*

spring.datasource.url=jdbc:mysql://localhost/**mi\_base\_de\_datos**?useSSL=false&serverTimezone=UTC&allowPublicKeyRetrival=true

spring.datasource.username=root

spring.datasource.password=

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

*# MySQL??Dialect DEBEMOS REEMPLAZAR LOS ?? POR LA VERSIÓN DE MYSQL QUE USAREMOS*

*# spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect*

spring.jpa.database-platform=org.hibernate.dialect.MySQL5Dialect

*# ESTO ES PARA PODER VER EN EL LOG DEL SQL QUE SE HA GENERADO EN CADA OPERACIÓN*

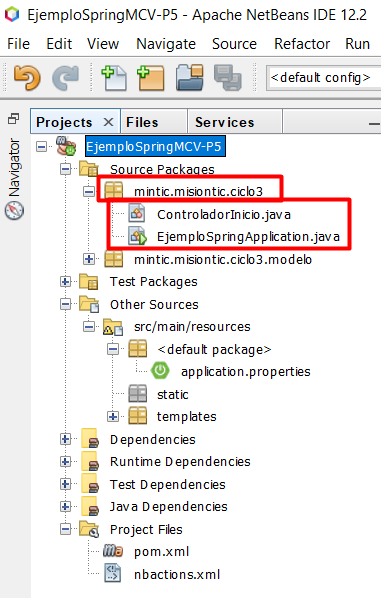
spring.jpa.properties.hibernate.format\_sql=true

logging.level.org.hibernate.SQL=DEBUG

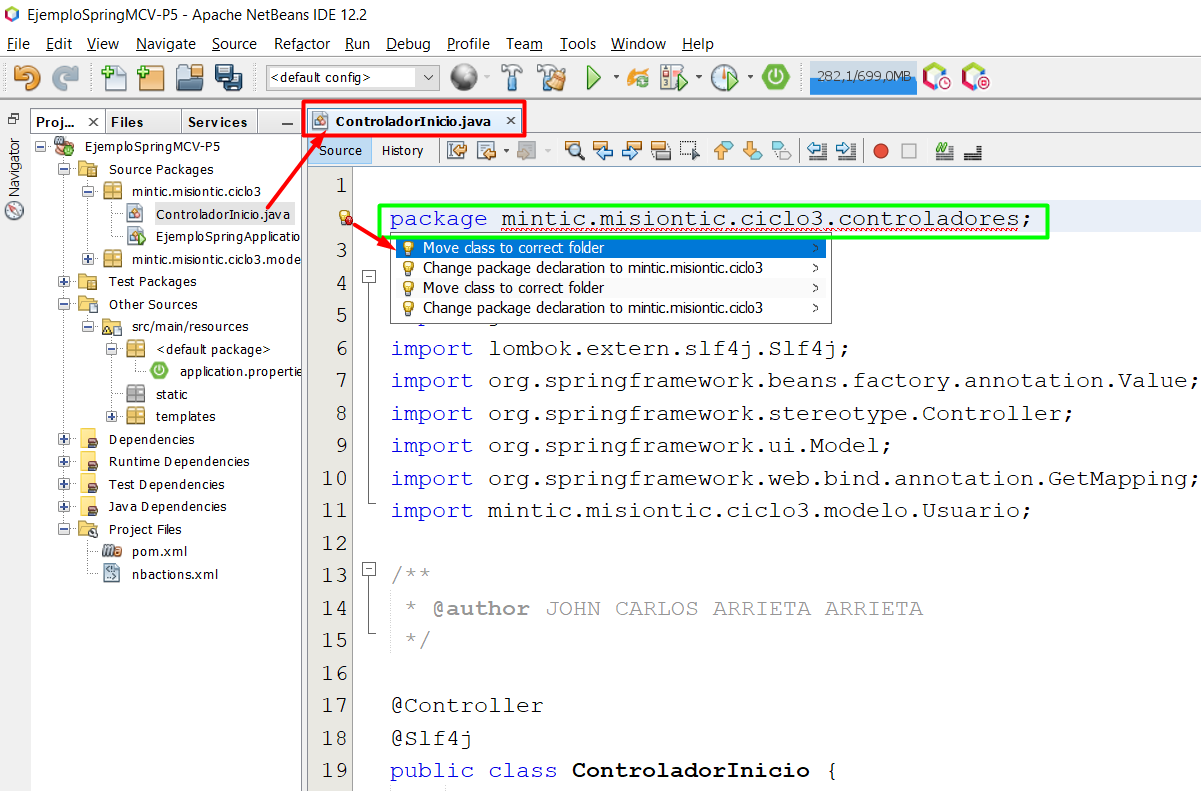
logging.level.org.hibernate.type.descriptor.sql.BasicBinder=TRACE

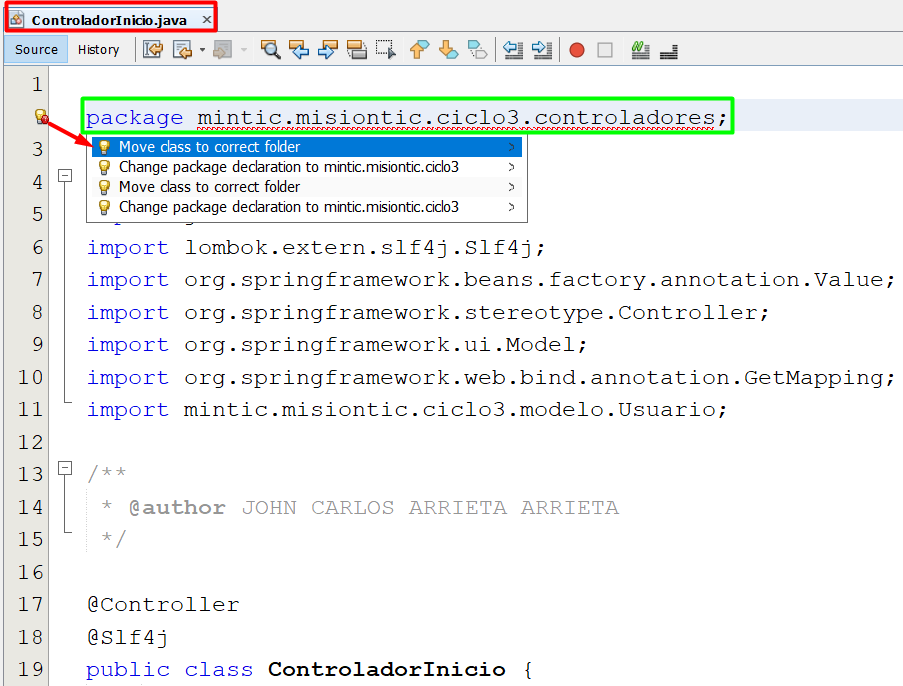
Debemos de tener cuidado en esta línea, en mi caso me presentaba error porque mis versiones son diferentes a la que vemos aquí, yo use:”org.hibernate.dialect.MySQL57Dialect” siempre podremos verificar nuestra version si nos vamos a las dependencias buscamos el hibernate, luego la ruta que vemos ahi, org.hibernate.dialect y ahi la version. por ejemplo en mi caso: como vemos es 57 y no 5 pero esto por las versiones como ya mencione anteriormente, (también sirve con 8).

spring.jpa.database-platform=org.hibernate.dialect.MySQL5Dialect

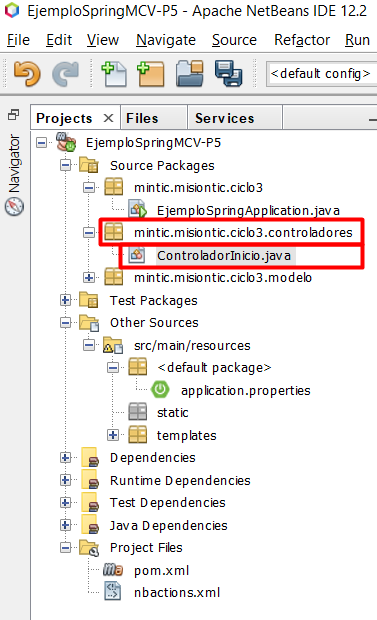


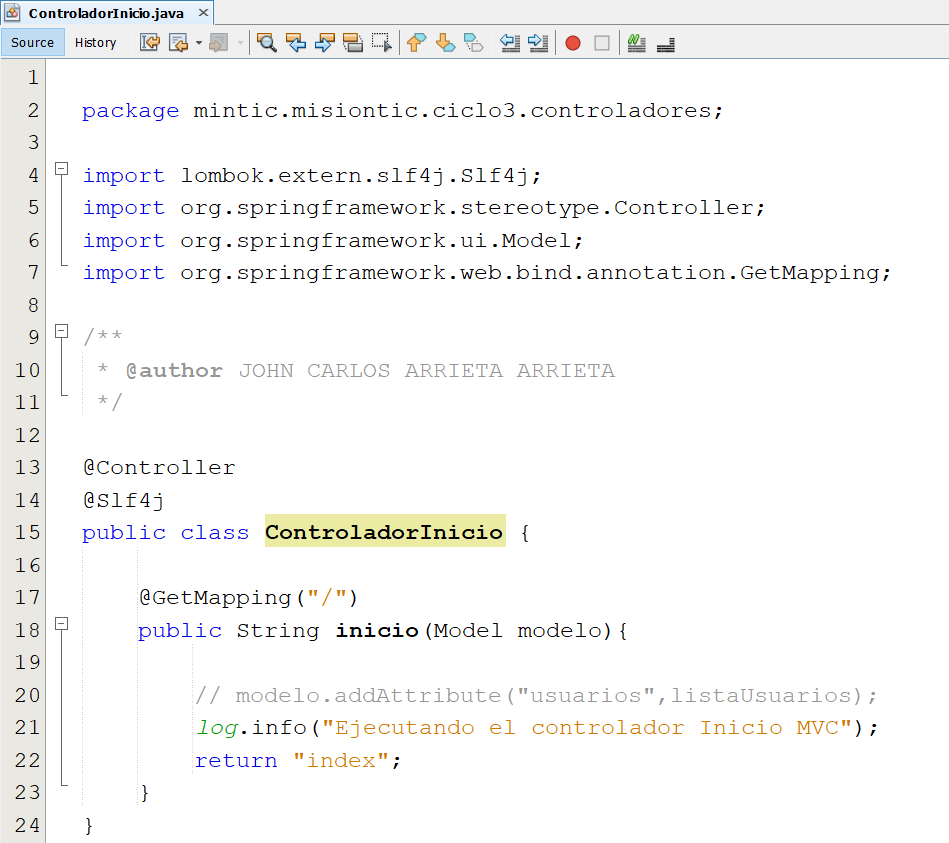
crearemos la un paquete llamado controladores donde moveremos el ControladorInicio,



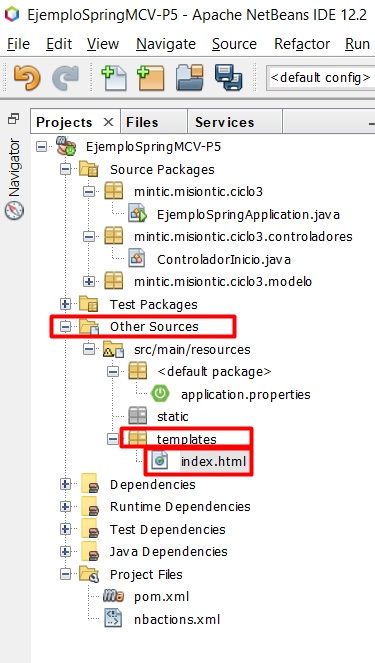


podemos modificar directamente la ruta del package en el codigo del controladorInicio.java y luego en el foco que nos indica error. clickear en Move to class to correct folder y asi se moverá y creará automáticamente este folder.

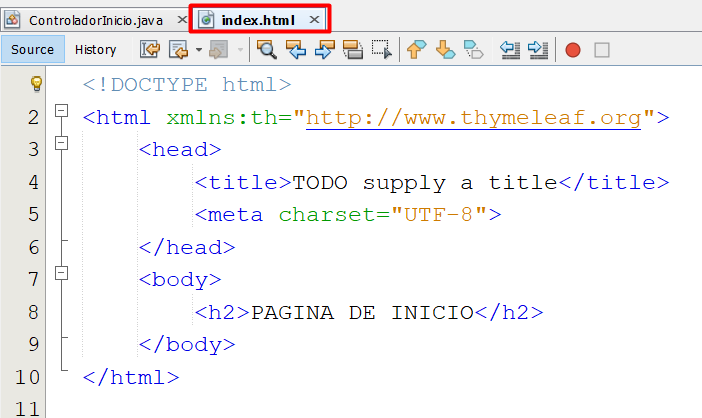


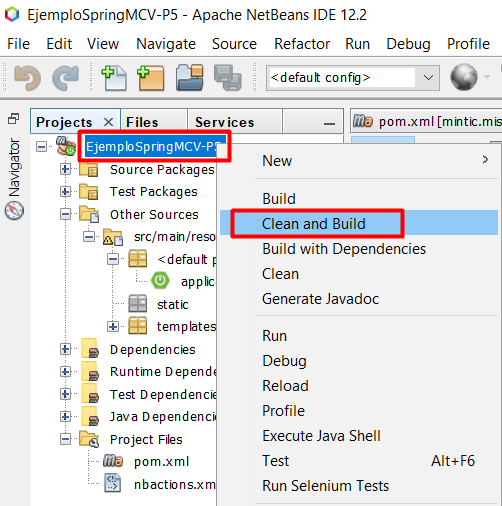


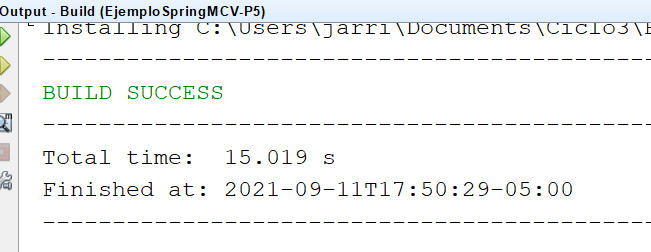
Vemos que ya no muestra este error.



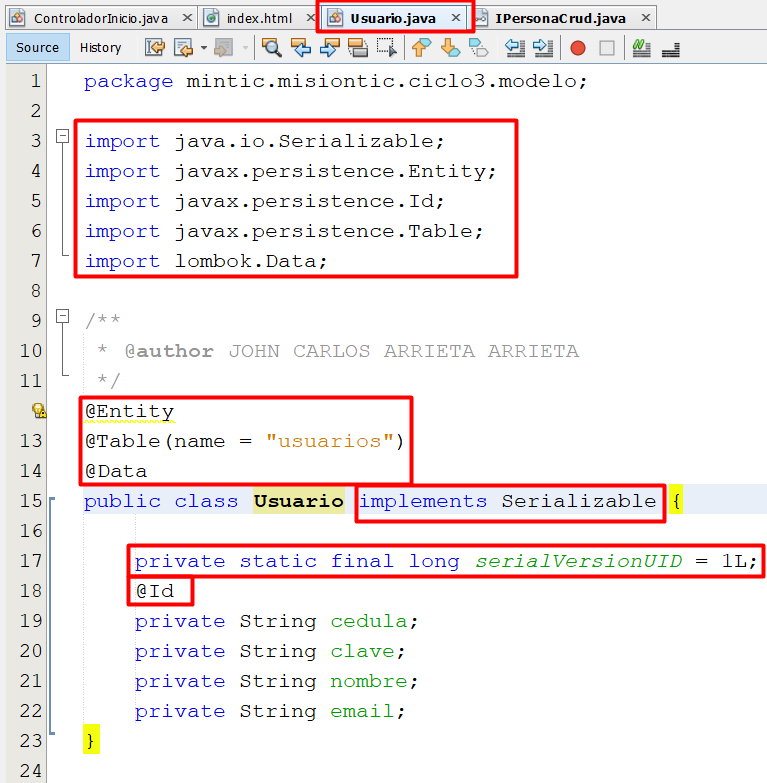
Realizaremos un Clean and build para verificar que no se nos haya dañado nada con las dependencias.



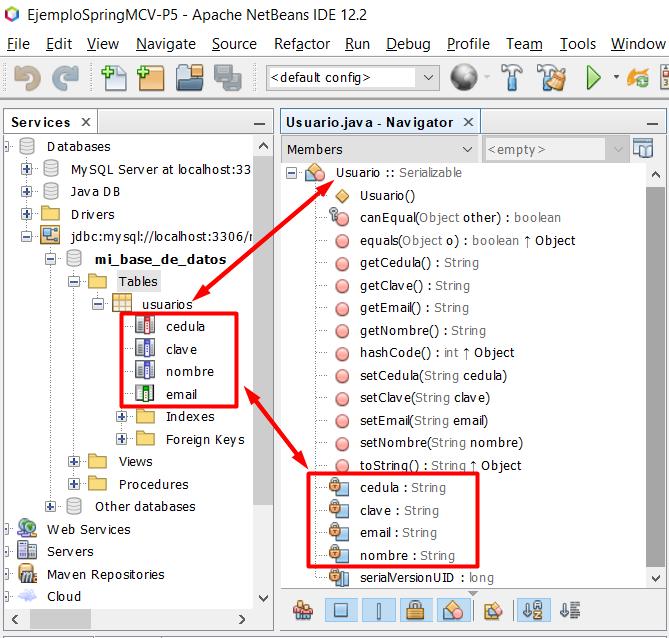




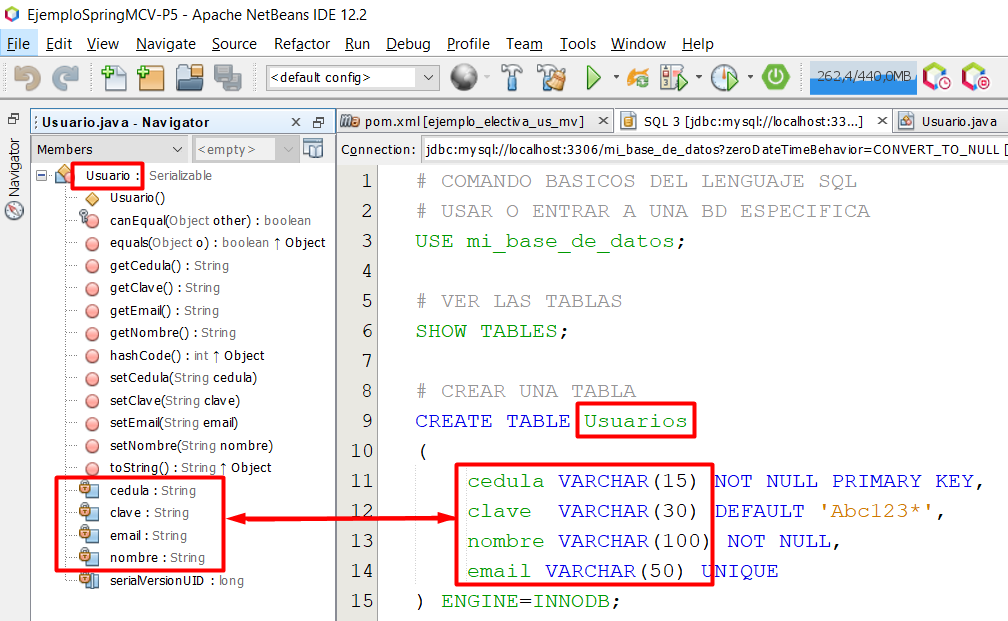
Ahora nos vamos a dirigir a nuestra clase Usuario



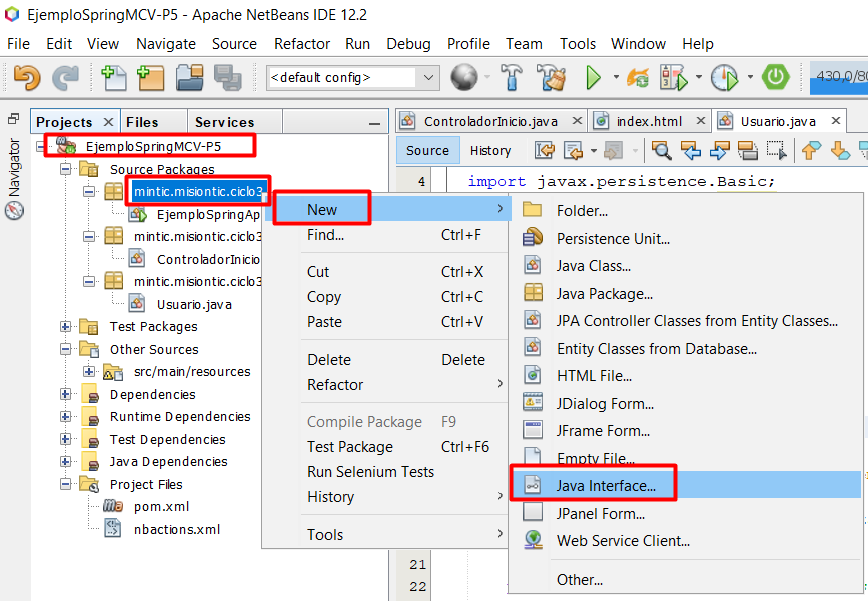
Reemplazamos algunas de las anotaciones que teníamos anteriormente por las que se ven en pantalla y agregaremos las librerías necesarias para estas desde el import.



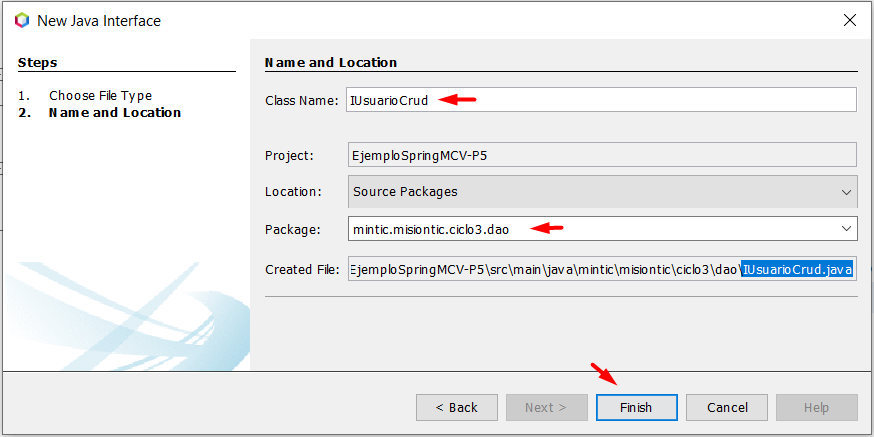
usaremos estos atributos para crear nuestros usuarios y guardarlos de la mismo forma en la base de datos.

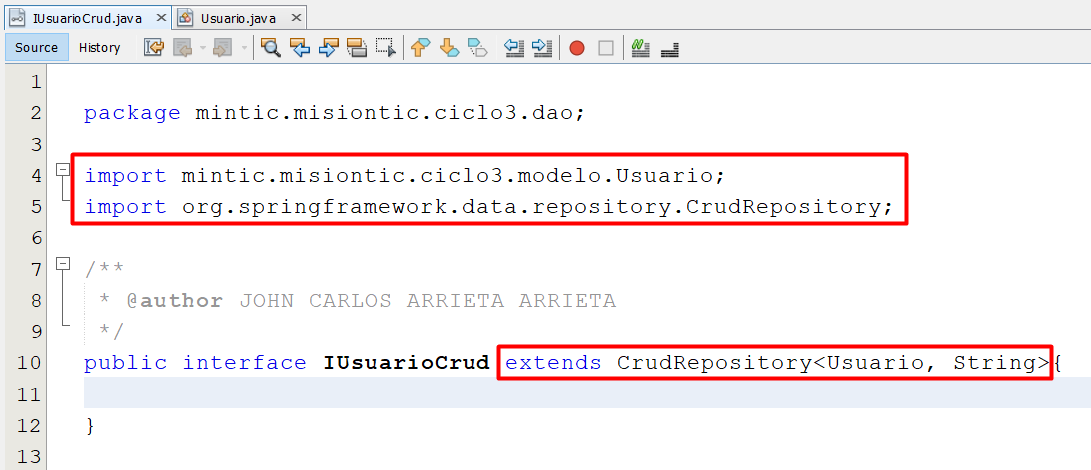


verificamos el tipo de dato que vamos a almacenar.



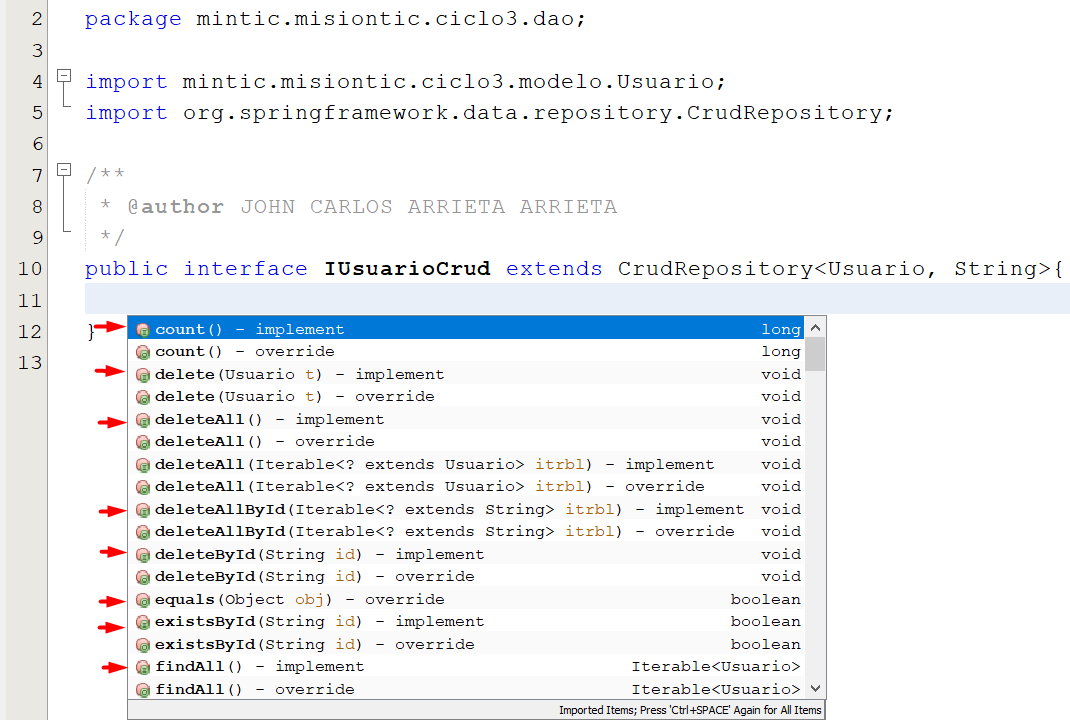
Nos dirigimos al source packages, click derecho y new java interface.



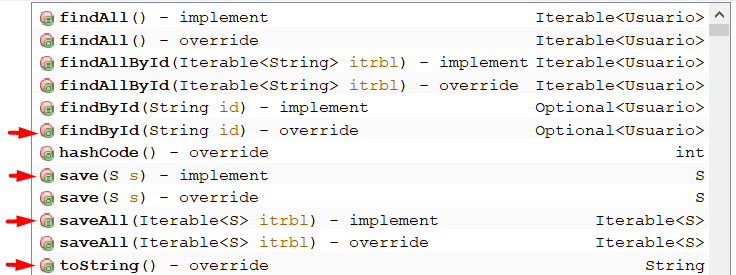


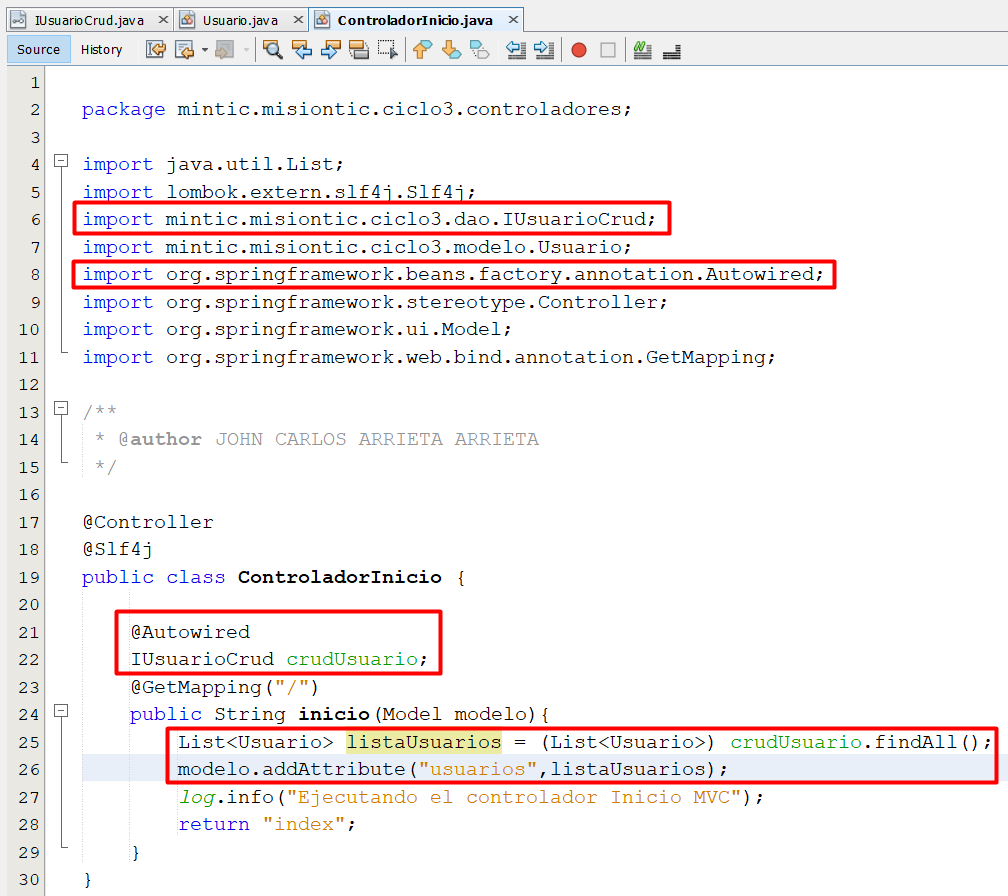
en este caso lo guardamos como IUsuarioCrud y en el paquete dentro de un paquete que crearemos llamado .dao

Importamos las librerías que vamos a usar

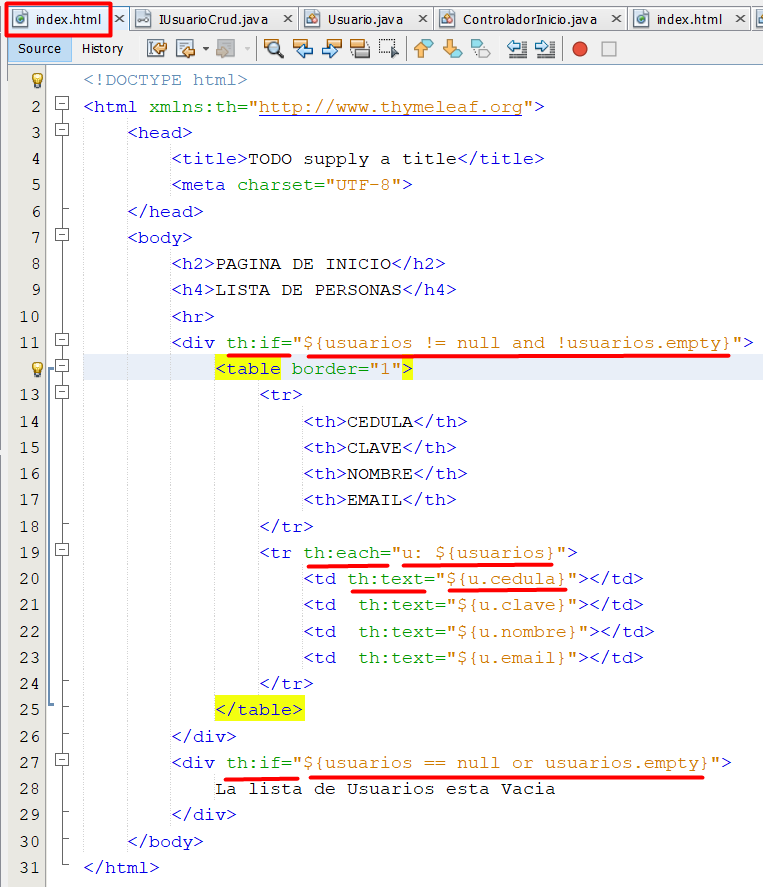


nos vamos al usuariocrud, click derecho>generar codigo> override methods y seleccionamos todos para agregarlos automáticamente.

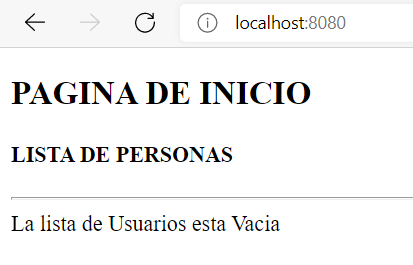




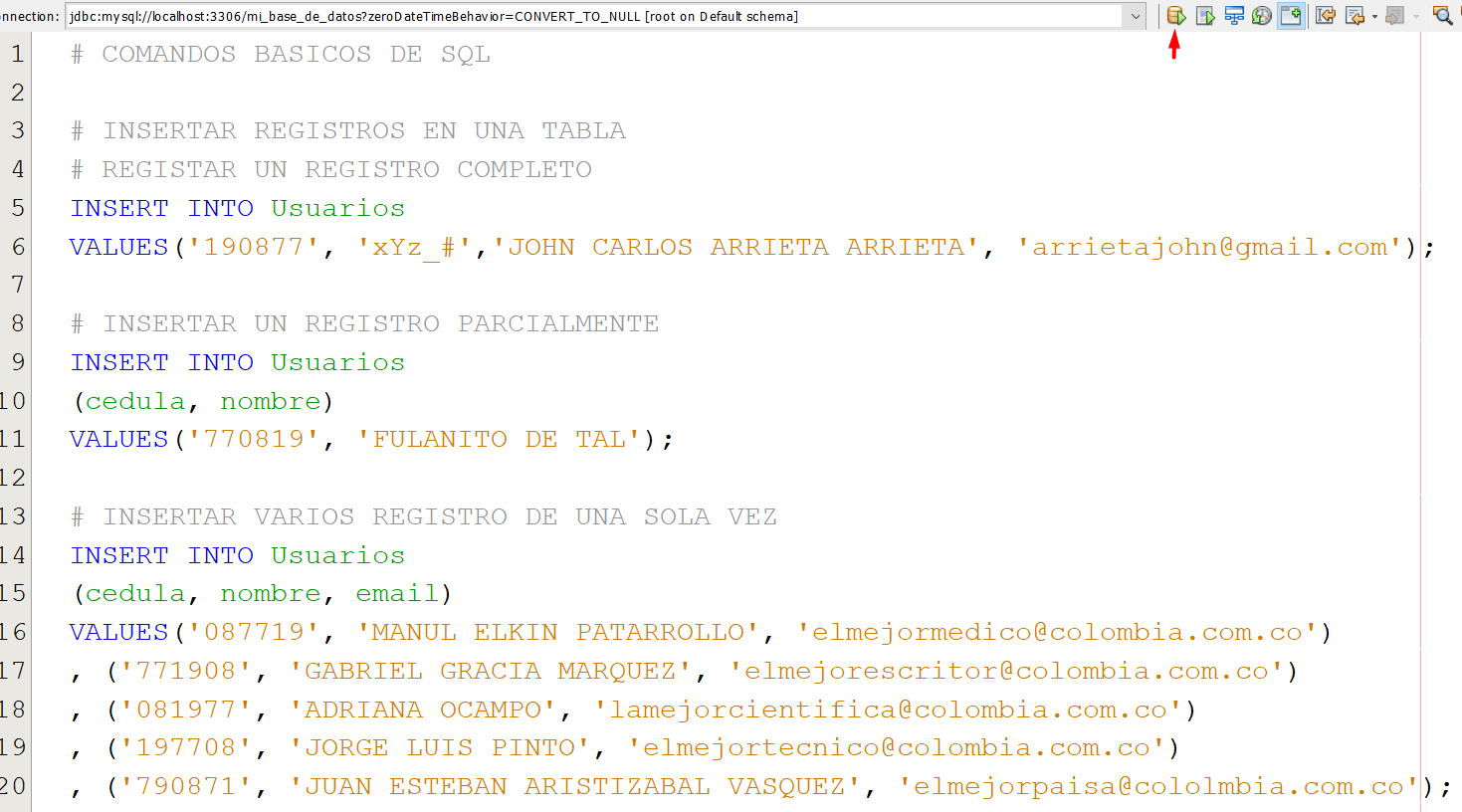
Nos movemos al controlador y agregamos las anotaciones que vemos en pantalla, y también importamos las librerías, asi como importamos el IUsuarioCrud que esta en el paquete .dao, modificamos el list<Usuario> para recibir el crudUsuario.



Para no tener errores eliminamos los u, u1,u2 y todos los usuarios que habíamos creado para que nos mostrara, ya no los necesitamos.



si todo sale bien nos debería mostrar que la lista de usuarios está vacía, ya que no tenemos ningún usuario en la base de datos.



insertamos por comandos de sql usuarios con el INSERT INTO.



y podremos ver estos usuarios una vez ejecutemos la página.