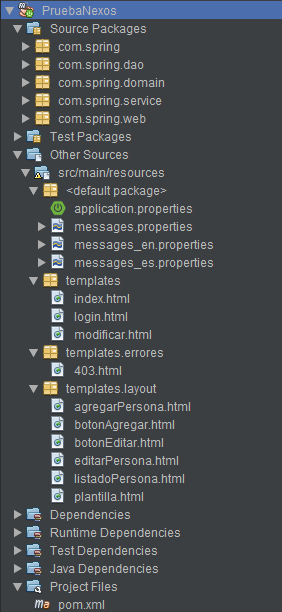
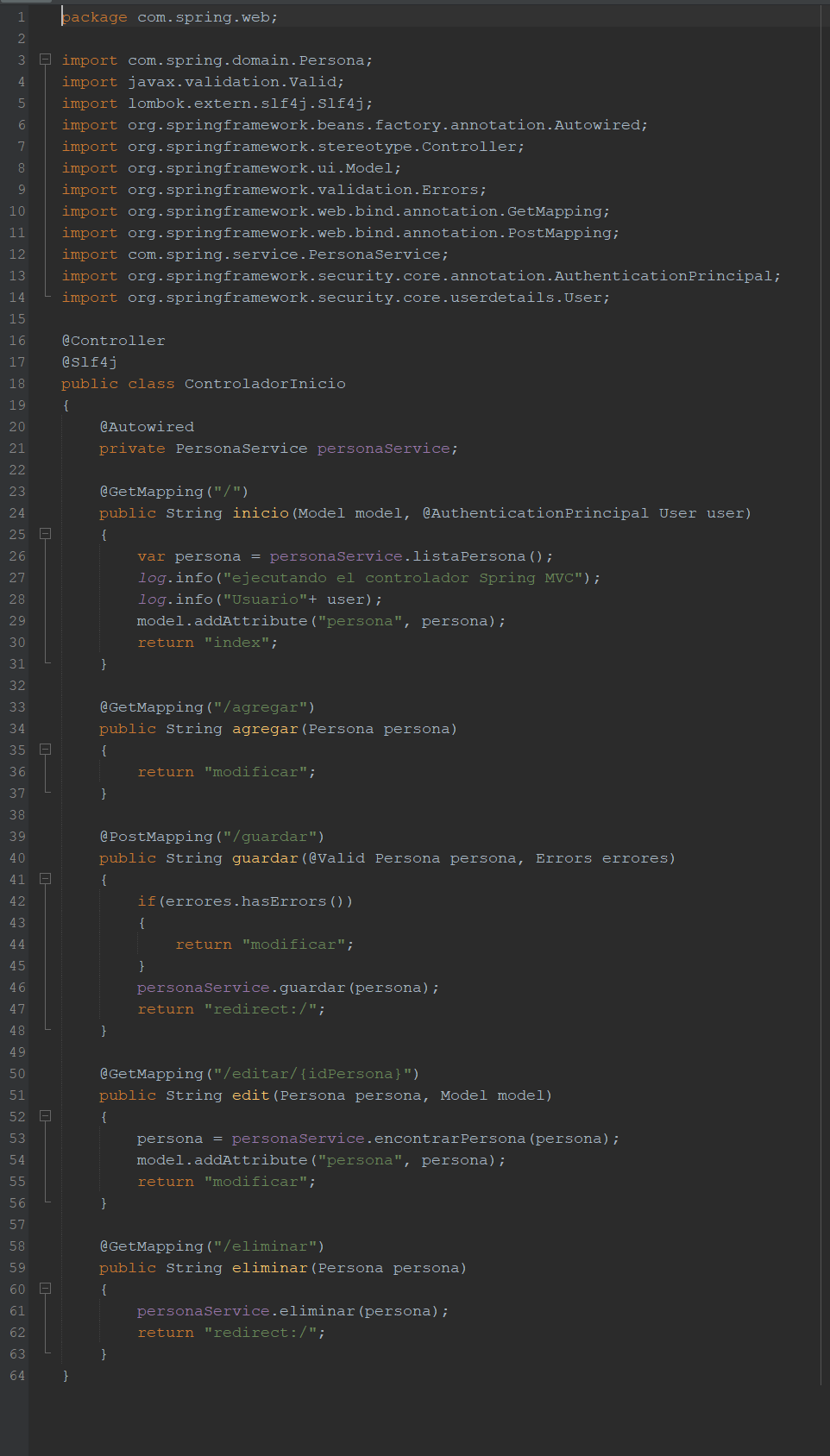
**DOCUMENTO PRUEBA NEXOS SOFTWARE**

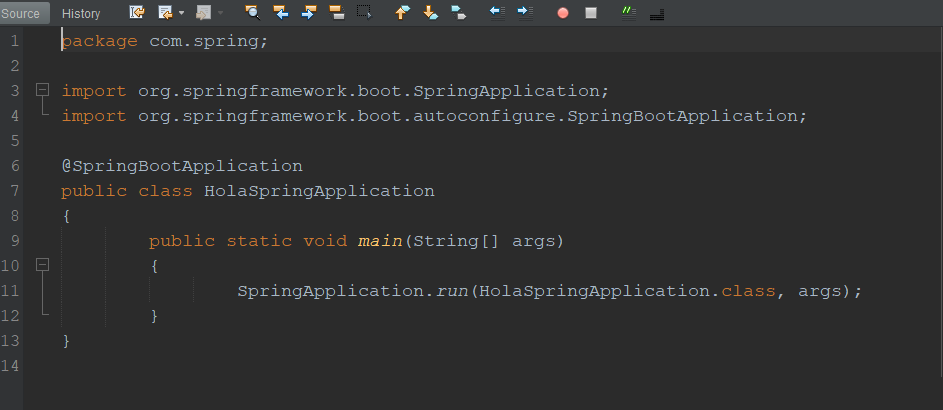
Mi nombre es Jaime Andrés Arboleda Garzón fue seleccionado por ustedes para realizar la prueba para la vacante desarrollador, por medio de este documento me permito evidenciar los procedimientos de la prueba q fueron realizadas en (Apache NetBeans , framework SpringBoot, MySQL) debido a mis conocimientos realice la prueba con estas herramientas.

De antemano agradezco la oportunidad que me están brindando

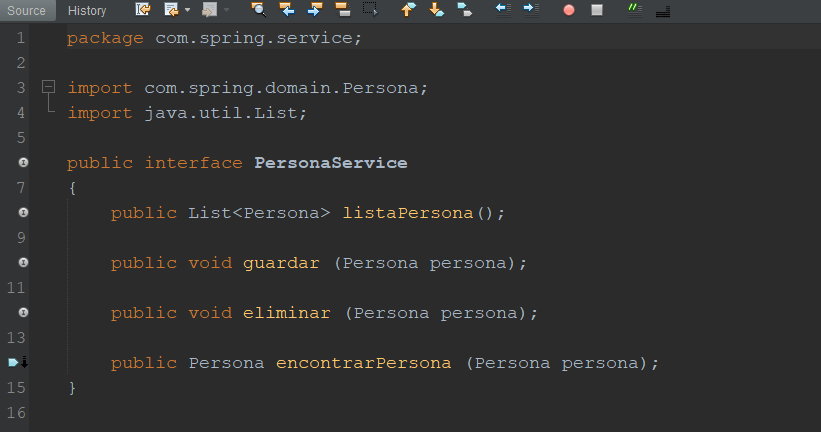
1. Descargar un plugin para Apache NetBeans el cual nos va a permitir también  
   integrar este proyecto de Springboot y crearlo desde nuestro IDE de Apache NetBeans así que abrimos nuestro navegador y descargamos el plugin Spring Boot NetBeans para que nuestra aplicación se pueda ejecutar y poder evidenciar el código desarrollado
2. Como siguiente el nombre del proyecto se va a llamar PRUEBA Nexos, aquí usaremos THYMELEAF este va a ser la tecnología de despliegue que vamos a utilizar para nuestras aplicaciones Web con Spring



1. Como evidenciamos aquí usamos la notación GETMAPPING para procesar la información get, de igual manera tenemos el POSTMAPPING ya que esta petición va a recibir información de tipo post, la ruta que en este caso es localhost:9090 que es donde podemos evidenciar nuestra aplicación en la web
2. Esta es nuestra clase principal la cual es la que contiene el main para que nuestra aplicación funcione y se ejecute en nuestro navegador



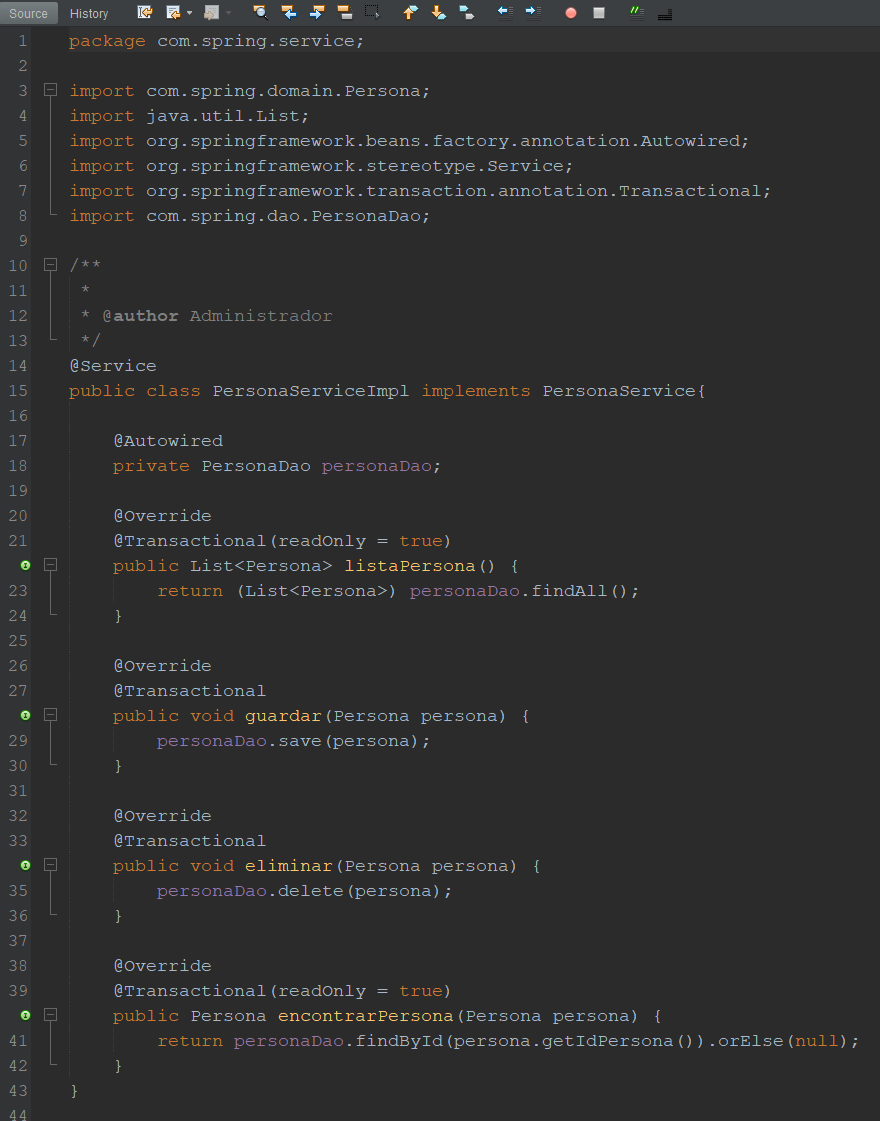
1. PERSONA SERVICE la interfaz contiene los métodos que vamos a implementar en nuestra clase llamada personaServiceImplementation



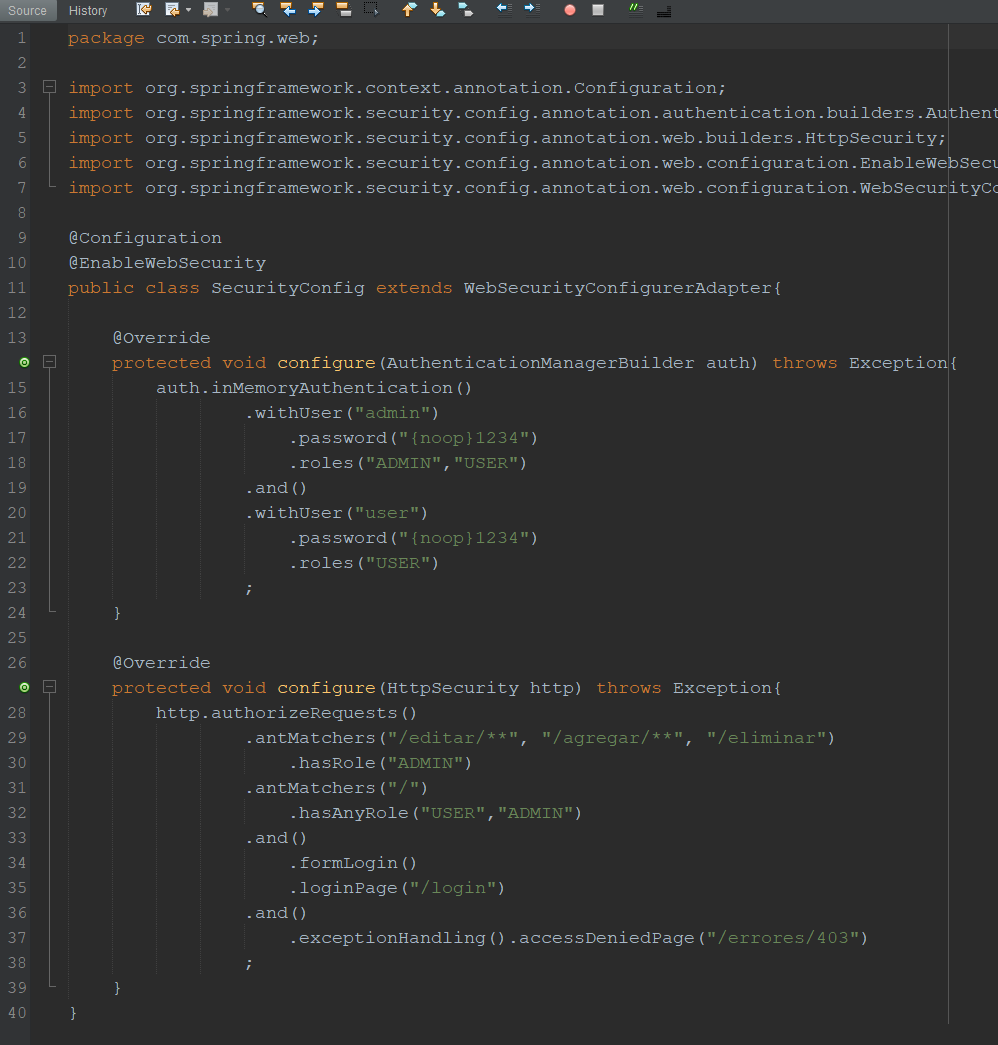
Obviamente podemos agregar más métodos según necesitemos, pero de momento lo que vamos a hacer es completar nuestro código para poder crear nuestra aplicación de tipo CRUD Créate Read Update Delete.

Para poder ejecutar todas esas acciones desde la aplicación web así que ya tenemos nuestra interfase y ahora vamos a agregar nuestra implementación la clase se va a llamar personaService y MPL de Implementación

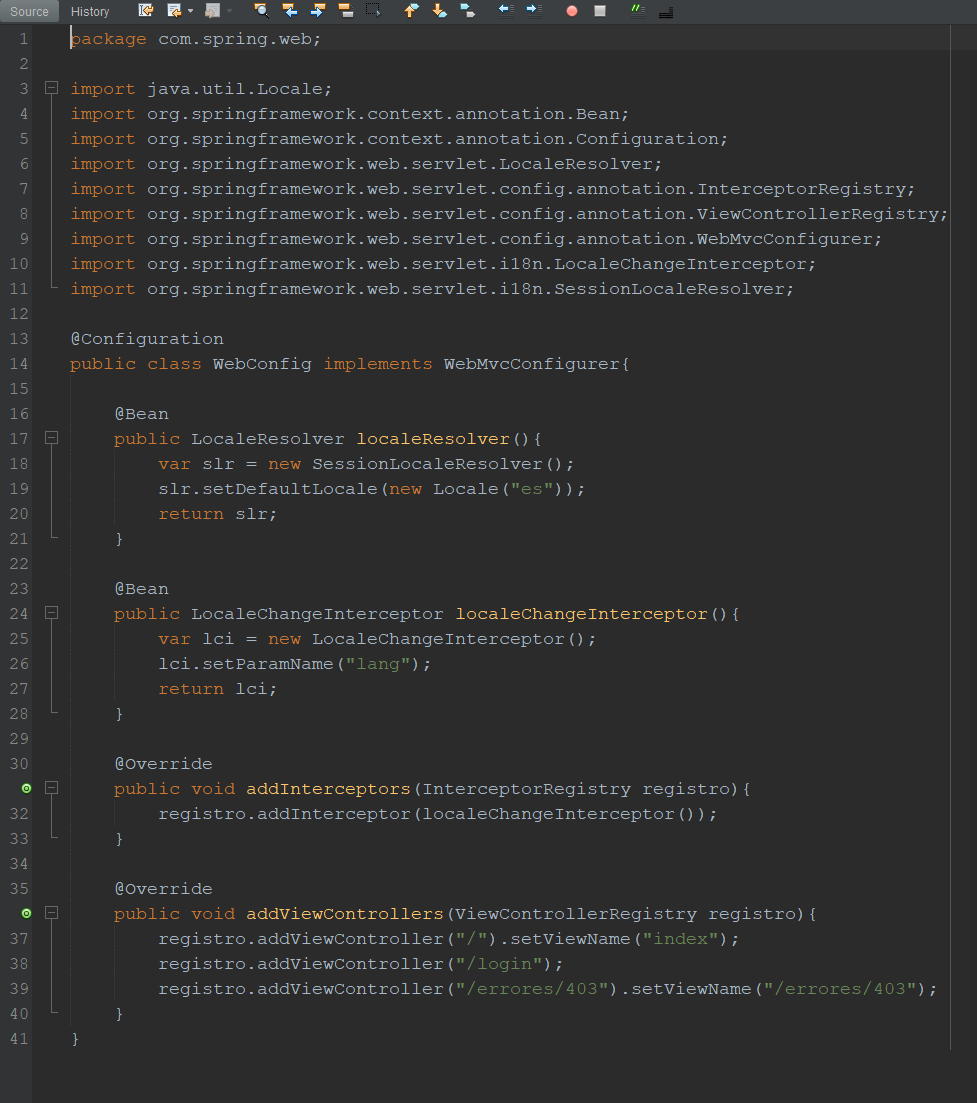
PERSONA SERVICE IMPLEMENTACION



1. SECURITY CONFIG lo que se hizo en esta clase básicamente es configurar los usuarios para poder acceder a nuestra aplicación

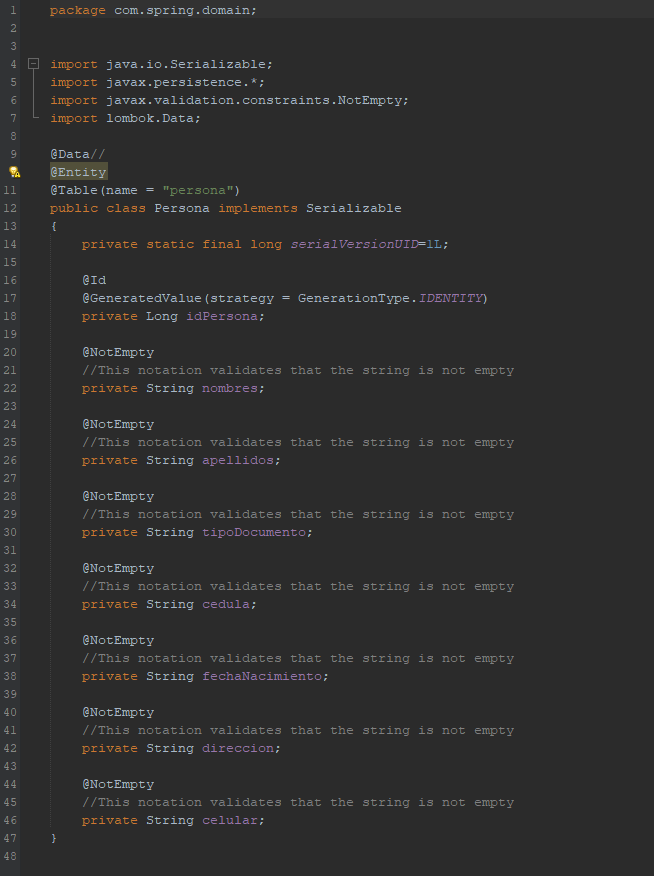


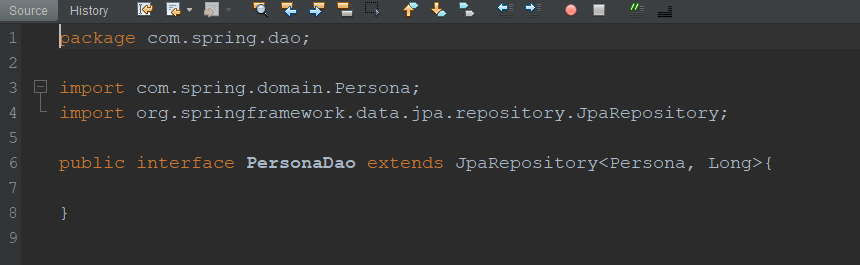
WEB CONFIG



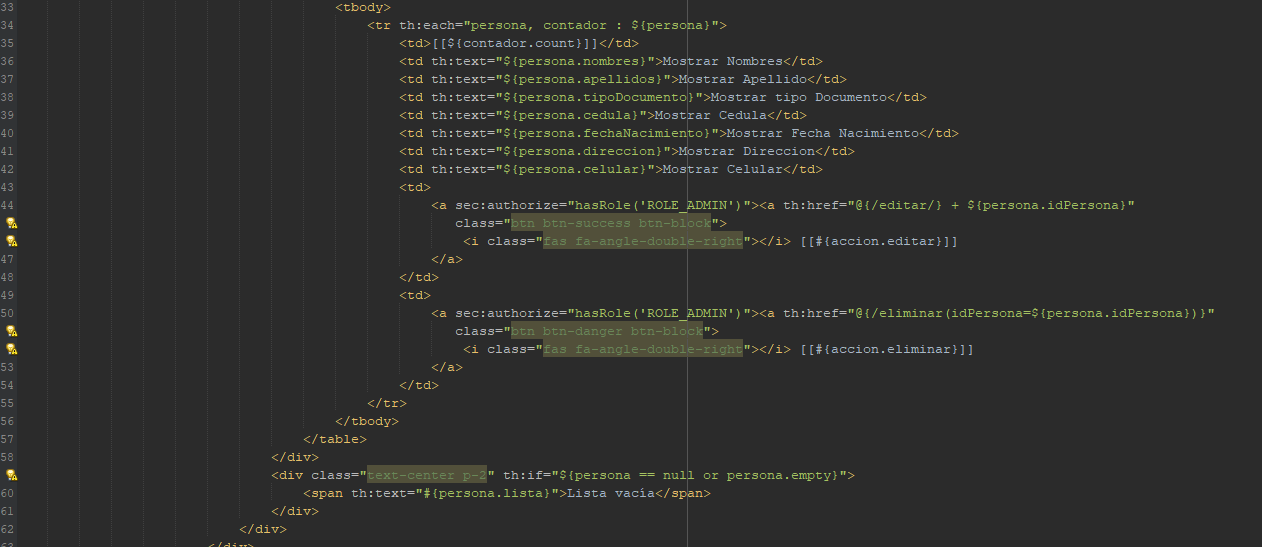
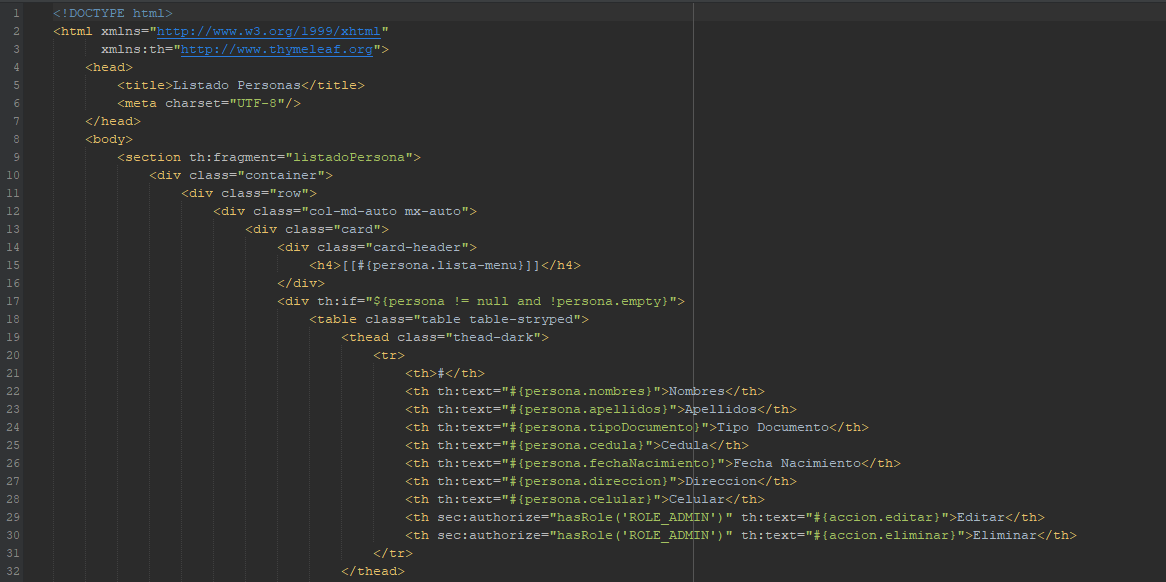
CLASE PERSONA

1. Aquí vemos la notación @DATA esta notación nos permite usar los getters y setters de nuestra aplicación sin necesidad de especificarlos en nuestro código, de igual manera la notación @Table ya que en la base de datos podemos tener nuestras tablas en minúsculas y en java las tenemos en mayúsculas entonces se usa esta notación para que no tengamos ningún inconveniente

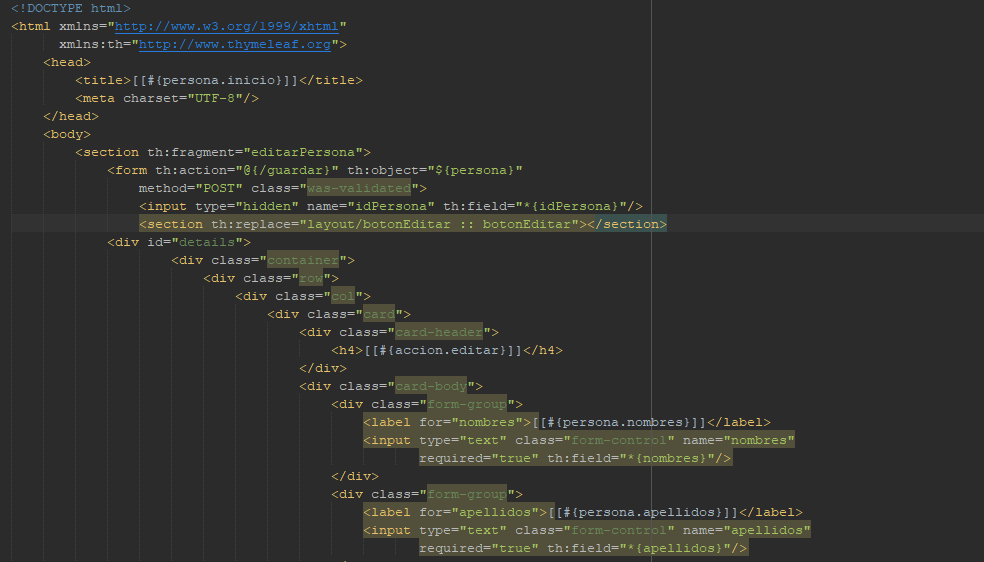


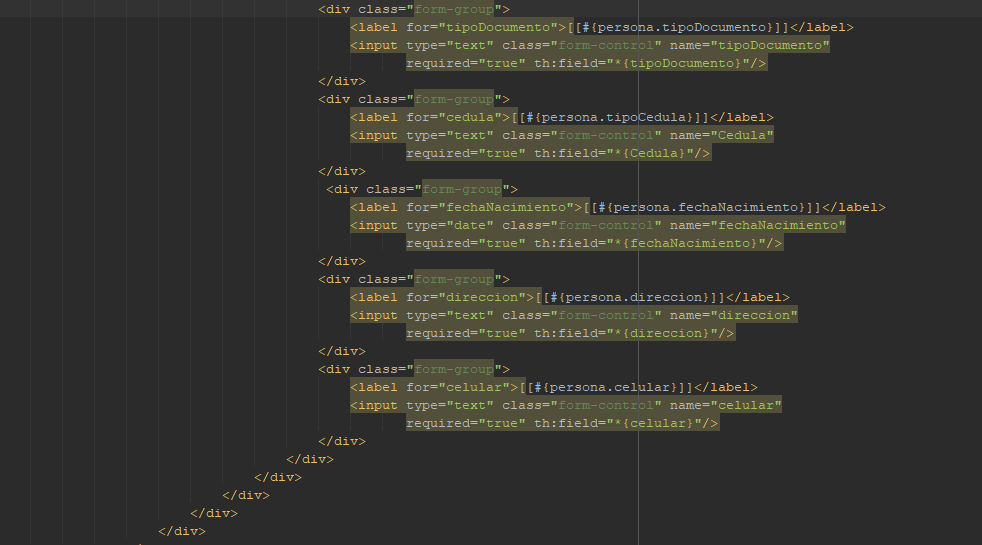


1. En esta sección se crea la lista o tabla de personas que se tienen en el aplicativo, también le damos una acción que dependiendo el usuario que ingrese (admin o user) se va a evidenciar los permisos que tiene cada uno de ellos

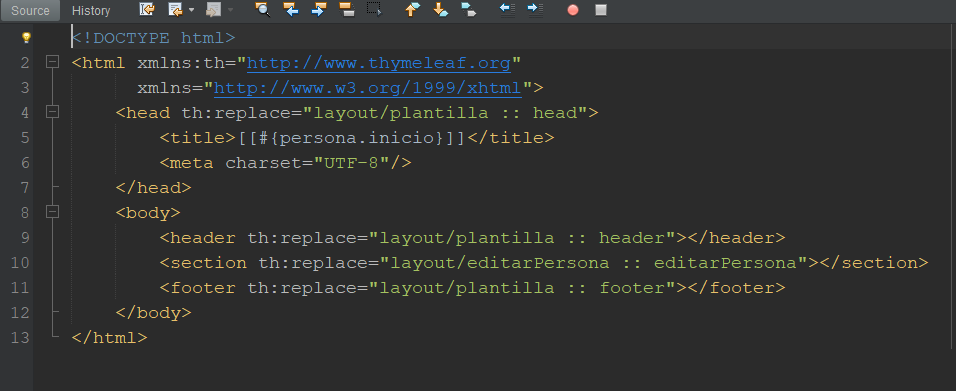


EDITAR PERSONA

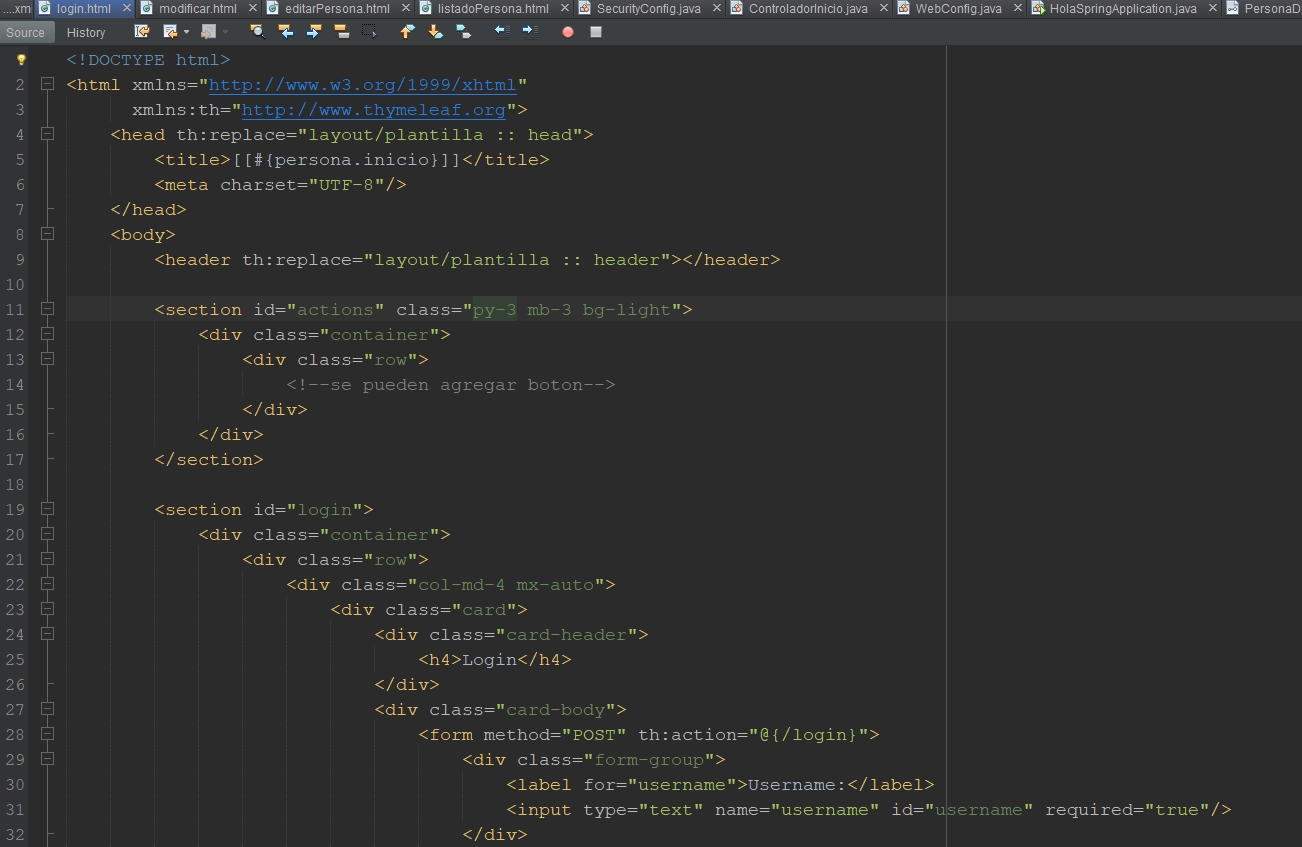


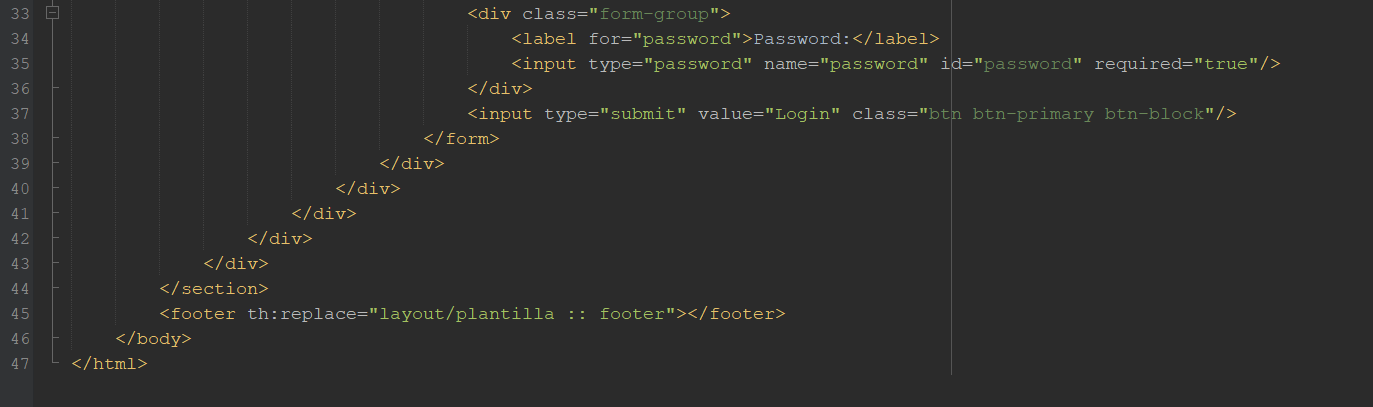


MODIFICAR PERSONA

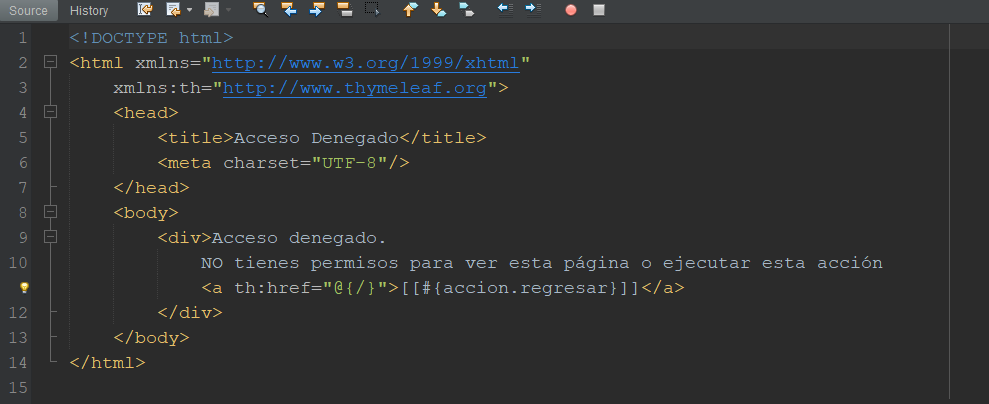


1. Cree un login para el usuario de (admin y user) ya que los dos tienen diferentes permisos para validar la información que se encuentra en nuestra aplicación

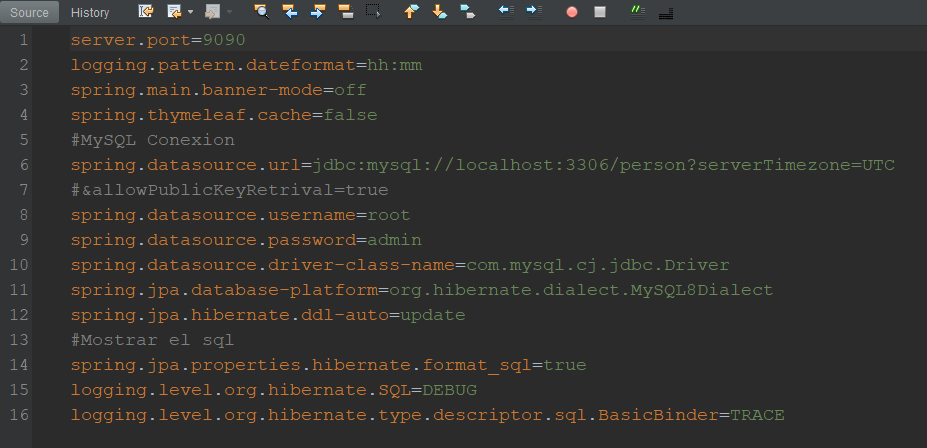




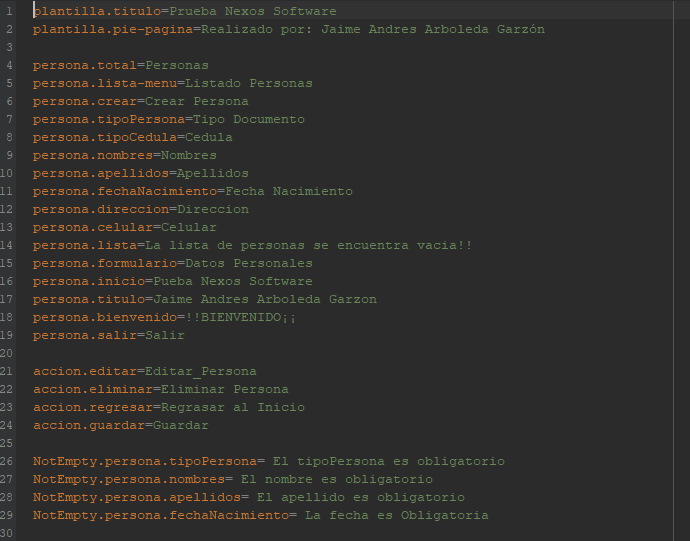
403 ERROR



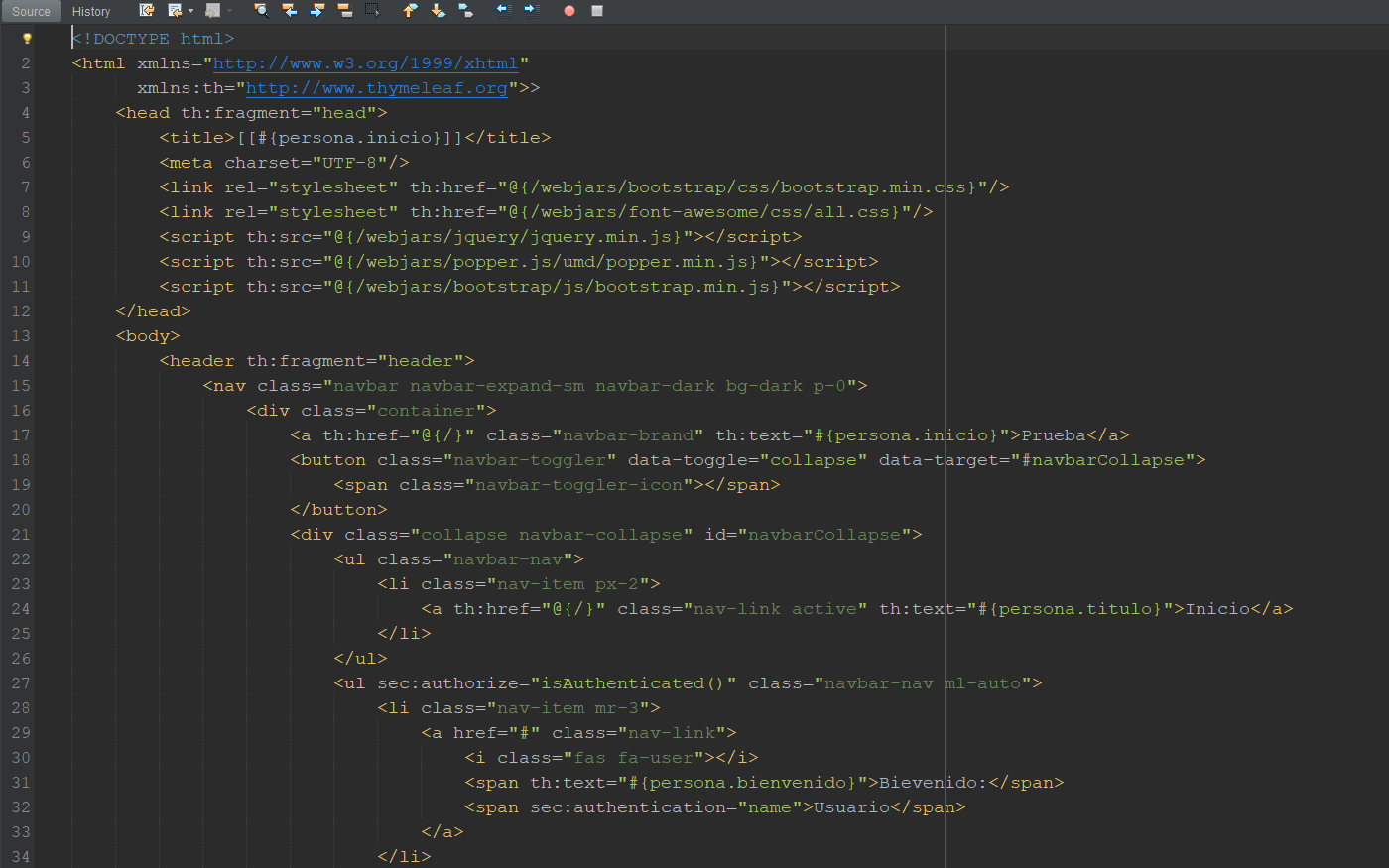
1. APPLICATION PROPERTIES Aquí es donde se va a configurar la conexión a la base de datos de igual manera también el driver Esa es la clase del driver de MySQL que se va a utilizar para crear la conexión a la base de datos, estamos involucrando hibérnate ya que lo que sucede cuando estamos utilizando JPA Spring por default la implementación que va a utilizar de JPA es hibérnate, si quisiéramos utilizar otra implementación como Eclipse Link Open JPA u otro tipo de implementación entonces se tiene que especificar en el archivo XML y además realizar las configuraciones necesarias para que se pueda conectar correctamente utilizando otro tipo de implementación pero en este caso la implementación por default es hibérnate

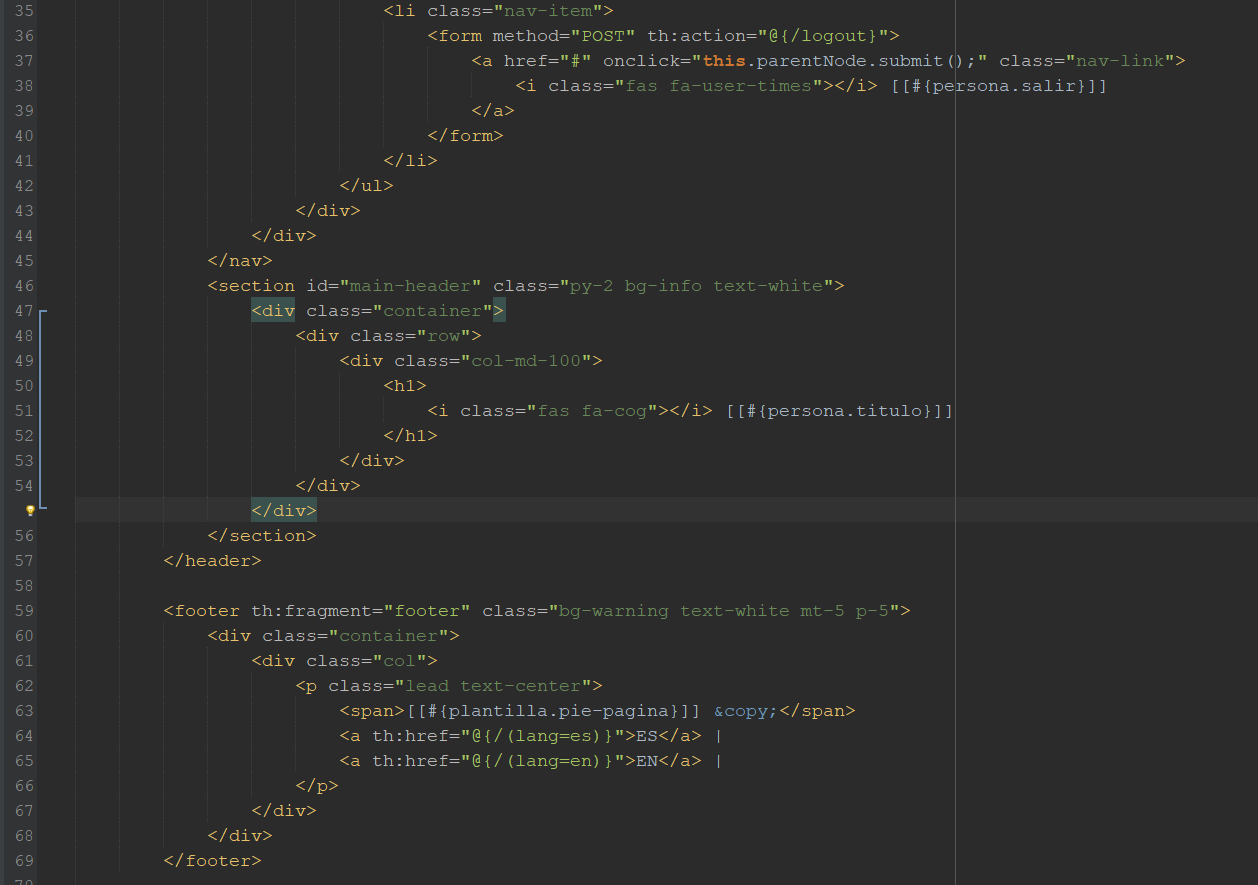


1. MESSAGES PROPERTIES aquí usamos diferente tipio de mensajes para que se vean en nuestra aplicación solo usando nuestro nombre personalizado los mensajes se verán en la aplicación en español o en ingles dependiendo nuestro caso

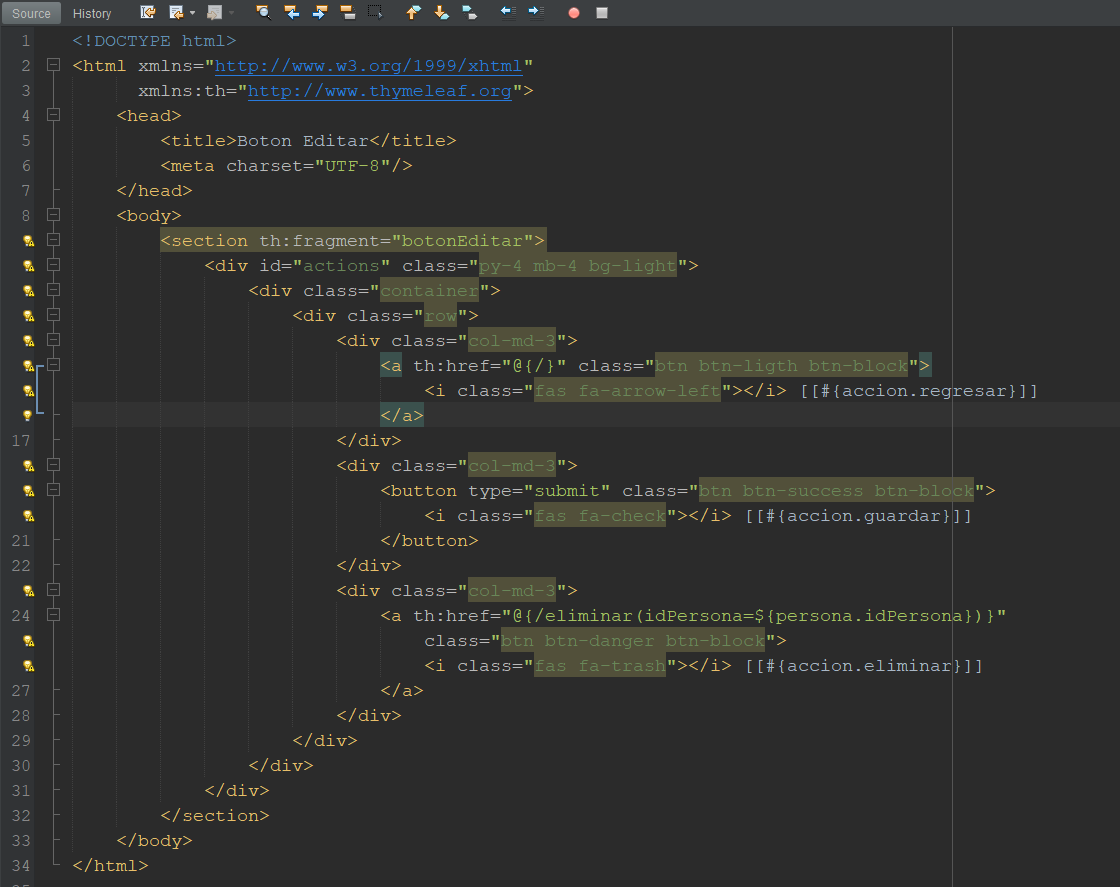


PLANTILLA

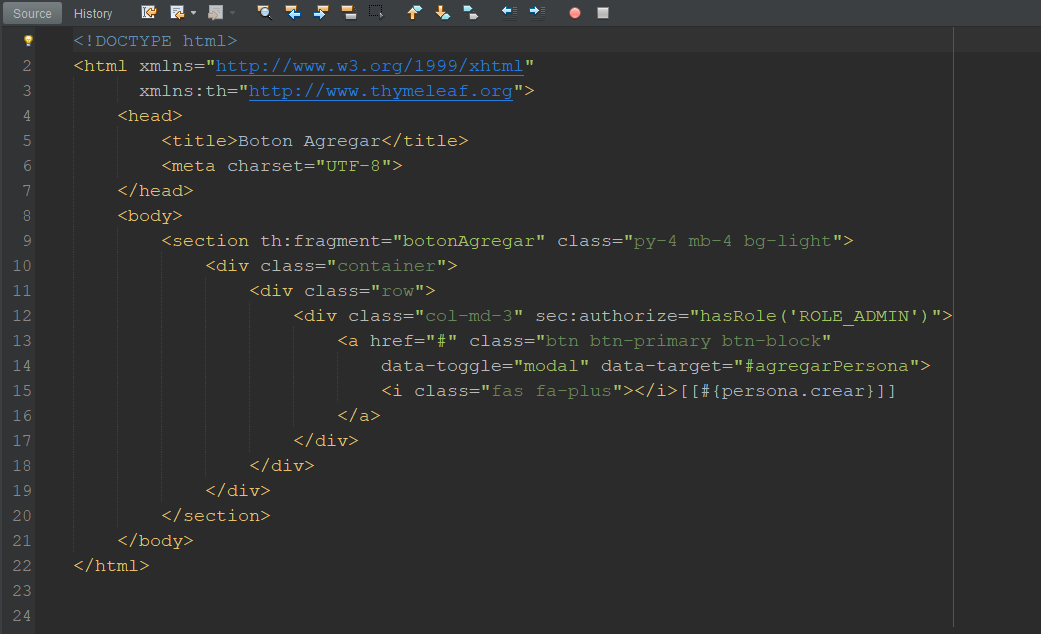




BOTONES EDITAR



BOTON AGREGAR



Agregar persona

