SPRING BOOT + SECURITY + BOOTSTRAP

SEGURIDAD CON SPRING BOOT + THYMELEAF Y MYSQL. CAPÍTULO 1. PRESENTACIÓN

**Presentación del proyecto:**

En esta serie de artículos vamos a crear una aplicación monolítica similar a la de los tutoriales publicados en este mismo blog sobre autenticación con JWT en una api REST. En dichos tutoriales implementábamos un cliente con Angular / Ionic que consumía una api REST Spring Boot. En realidad eran aplicaciones separadas, cada una corría en un puerto distinto.

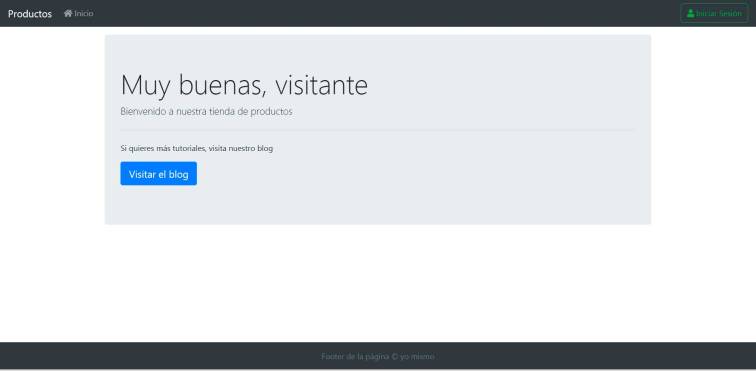
Pues bien, en el caso que nos ocupa vamos a crear una aplicación monolítica donde todo se encontrará en el lado del servidor. Para las vistas utilizaremos el motor de plantillas Thymeleaf, el cual incorpora también varias librerías para la seguridad.

Y en cuanto a la autenticación, al no tratarse de un servicio REST, utilizaremos cookies. Como podréis comprobar se trata de una aplicación web básica, pensada para sitios web pequeños en los que no se necesita una arquitectura distribuida.

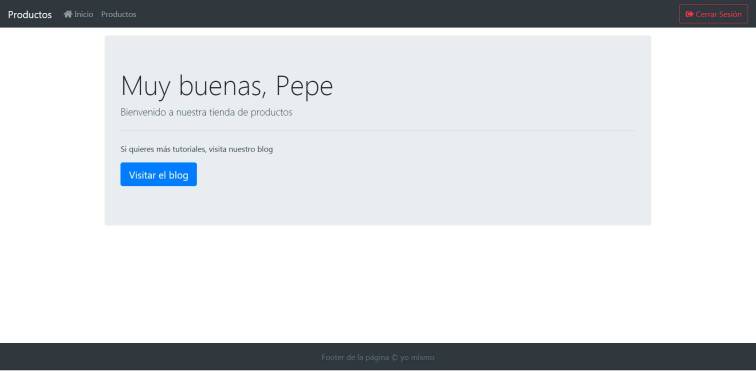
Por último, desplegaremos la aplicación en un servidor local Tomcat, para lo cual generaremos un archivo WAR, que es el que nos va a hacer falta para que el proyecto se pueda ejecutar en dicho servidor.

**Capturas:**

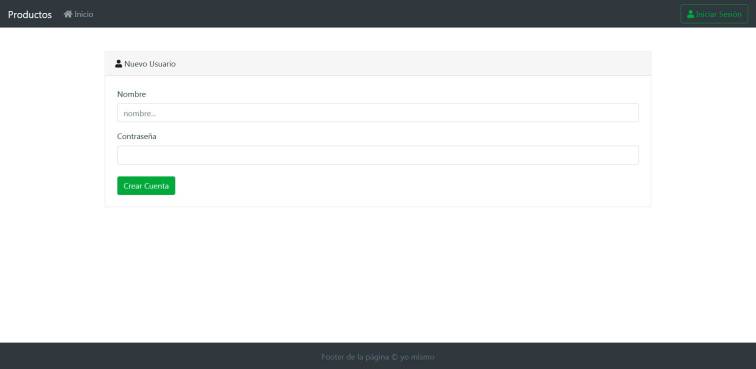
Sin extendernos más comenzamos a mostrar la aplicación. En la página de bienvenida se nos muestra el saludo según hayamos iniciado sesión o no:



Como se ve, aparece el mensaje de bienvenida junto al nombre de usuario si este ha iniciado sesión. El botón de la derecha del menú también depende de la sesión, en caso de estar logueado es el de cerrarla, en caso contrario, el de iniciarla.



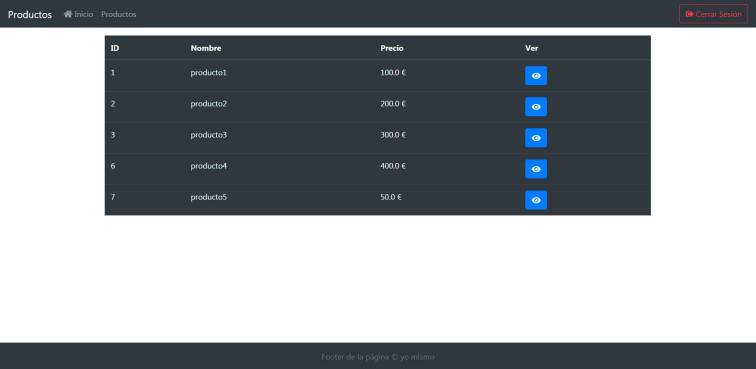
La pantalla del login tiene un enlace a la del registro si no se tiene una cuenta creada



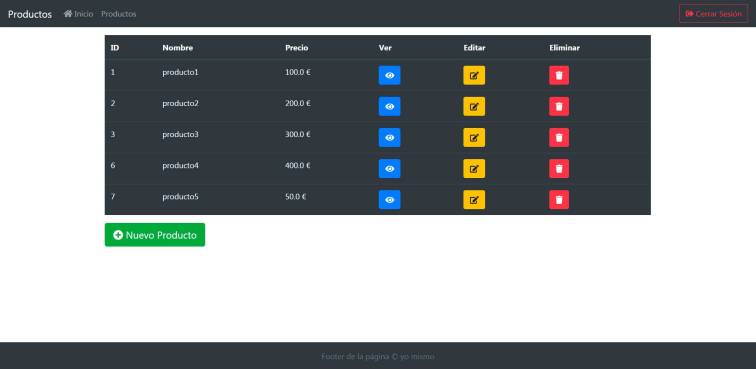
la cual es muy similar a la de login.

Una vez que se inicia sesión, el programa nos redirige a la página de inicio, donde se muestra el saludo y el nombre de usuario.

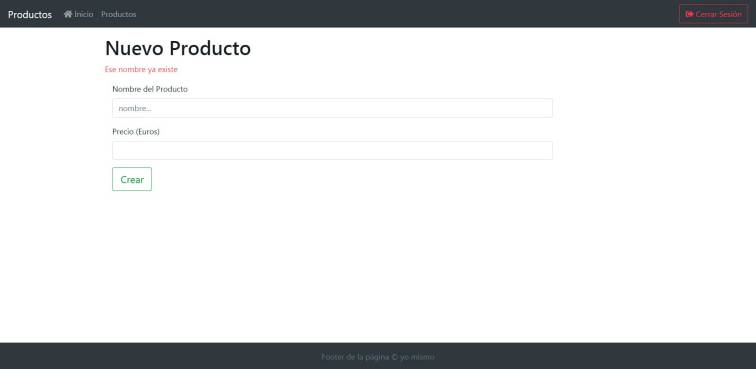
Al iniciar sesión aparece un elemento más en el menú, el de productos. Si hacemos click en él nos lleva a la página donde se muestra una lista con todos los productos.



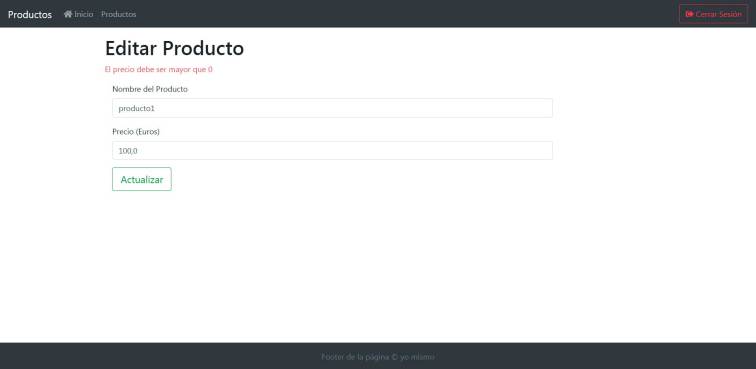
Como nos hemos logueado con un usuario genérico no muestra los botones de crear/editar/borrar; para ello hay que iniciar sesión como administrador, tal como hicimos en los tutoriales JWT.



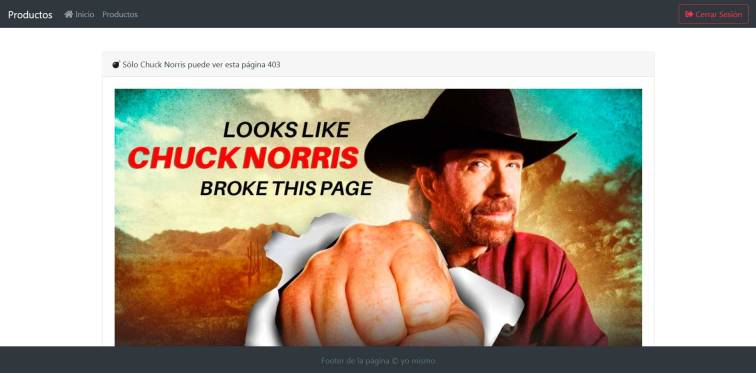
La pantalla de nuevo producto muestra los campos nombre y precio; en caso de poner un nombre que ya exista, dejar algún campo vacío o un precio menor que 1 mostrará error.



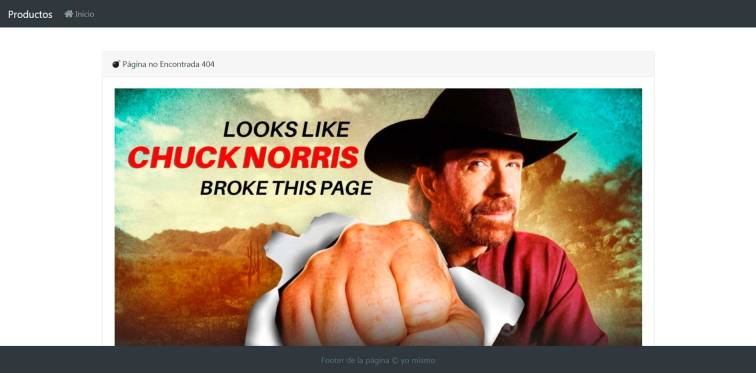
La vista de edición es similar, sólo que los campos contienen el valor del producto a editar…y lo mismo con los valores inválidos, se muestra un mensaje de error.



Y por último tendremos también una página de error cuando se intente acceder a una URL errónea o cuando un usuario que no tenga privilegios de administrador intente acceder a la vista de crear/editar.



La página de error 404 (página no encontrada) es similar a la de forbidden, sólo cambia el texto (la imagen es la misma).



Como vemos se trata de una aplicación muy sencilla pero más que suficiente para explicar como se implementa la seguridad con Spring Boot y Thymeleaf realizando el login hacia una base de datos MySQL.

**Software utilizado:**

* IDE IntelliJ
* XAMPP (para la base de datos MySQL)
* Java 1.8 jdk
* Spring Boot 2.1.8
* Servidor Apache Tomcat versión 8.5 (esta es la versión mínima requerida para realizar el tutorial)

Y damos por concluido este capítulo. En el siguiente crearemos el proyecto, la base de datos e importaremos las dependencias necesarias.