Actividad 1.2 / 3.1
Desarrollo de una aplicación Web con Spring Boot
Segunda parte – Proyecto inicial y estructura de paquetes – Página inicial

CONTENIDO

1.- Objetivos

Construir la aplicación mínima y la estructura de paquetes necesaria para comenzar con el desarrollo de la aplicación. Crear una página inicial para el sitio web. Ir tomando decisiones de maquetación.

2.- Requisitos

IntelliJ Idea Ultimate. Si no se dispone de Ultimate, se puede adquirir gratis con una cuenta educativa, que se obtienen con el correo @educa.madrid.org, a través del programa de licencias para estudiantes y profesores.
Alternativamente, se puede usar otro entorno de desarrollo, como Spring Tool Suite o NetBeans, pero las clases se impartirán con IntelliJ, y si se hacen referencias a entornos de desarrollo, serán respecto a IntelliJ.
Todos los paquetes, clases, métodos, variables, etc. Deberán escribirse en inglés. Por ejemplo, se usará mejor “data” que “datos”, mejor “found” que “encontrado” y mejor “client” que “cliente”. Lo mismo se aplicará a clases CSS personalizadas, si es que se crea alguna.

3.- Tareas a realizar

3.1.- Creación del proyecto

Crear un proyecto Spring Boot utilizando el asistente de IntelliJ o Spring Initializr.
Configurar el proyecto para:

- Nombre del proyecto: shop-alumno1-alumno2. Ejemplo: shop-juan-maria.
- Lenguaje: Java
- Tipo de proyecto: Maven
- Grupo: es.iesclaradelrey.da2d1e2425
- Artefacto: el mismo valor que el nombre del proyecto
- Nombre del paquete: es.iesclaradelrey.da2d1e2425.shopalumno1alumno2
- JDK: Amazon Corretto 21 (minor versión y patch son indiferentes)
- Versión de Java: 21
- Empaquetado: Jar
- Dependencias:
    - Spring Web
    - Lombok
    - Thymeleaf

Arrancar el proyecto, y comprobar que funciona, abriendo un navegador y accediendo al servidor. Aunque falle, devolviendo un 500, tiene que responder

3.2.- Renombrado de clase

Renombrar la clase de la aplicación, la anotada con @SpringBootApplication, para que su nombre sea “App”. Hacerlo de forma que cualquier referencia a ella se cambie.
Volver a probar la aplicación y verificar que está funcionando.

3.3.- Cambio de puerto de escucha de Tomcat

Cambiar el puerto en el que escucha el servidor Tomcat embebido en la aplicación, para que, en lugar de escuchar en el 8080, que es el puerto por defecto, use el 8000. Comprobar que no se producen errores.

3.4.- Creación de paquetes para distintas clases del proyecto

Crear los siguientes paquetes dentro de la aplicación:

- entities – Clases que representarán los datos de la aplicación. Son las clases que representan los datos que se persistirán en los repositorios en memoria o en base de datos. Estas clases se utilizarán a lo largo de toda la aplicación, y, entre otras cosas, podrán utilizarse en el paso de datos entre controladores y vistas.
- repositories – Clases que se encargarán de la persistencia, en memoria o en base de datos, de los objetos del paquete "entities".
- services – Clases que contendrán la lógica y los procesos del negocio. Utilizarán beans del paquete "repositories" para realizar la persistencia de datos, cuando sea necesario.
- controllers – Clases que se encargarán de procesar las peticiones recibidas del cliente. Utilizarán beans del paquete "services" para realizar la lógica de negocio, y persistir entidades cuando sea necesario.
- models – Clases que se usarán en Model, ModelAndView o ModelMap para pasar datos de controlador a vista, o que servirán para recibir datos de formularios, pero que no representan directamente datos persistentes. Aunque en muchos casos se usarán clases del paquete "entities" para el paso de datos entre el controlador y la vista, en algunos casos los modelos no serán datos que encajen exactamente en la estructura de la base de datos.
- config – Clases para realizar configuración de cualquier tipo: seguridad, conversiones, mapeos, etc.

3.5.- Creación de la página inicial

Crear la página inicial de la aplicación. Debe ser una página que responda a la URL http://localhost:8000, termine o no en "/".

Esta URL se debe atender en un controlador "HomeController", y debe usar una vista "index.html".

De momento, esta página sólo mostrará contenido estático, por lo que no será necesario que el controlador envíe datos a la vista.

3.6.- Selección de tema y maquetación inicial

El tema de la tienda es libre. Cada grupo puede elegir qué va a vender en la tienda. Discos, videojuegos, herramientas, productos de alimentación, comics, maquetas y réplicas a escala, etc.

Sólo hay que tener en cuenta algunas cosas:

- Los productos se organizarán en categorías. Sobre esto:
  - Inicialmente, la organización será simple:
    - No habrá categorías anidadas
    - Un producto pertenecerá solo a una categoría.
  - Posteriormente, se agregará una jerarquía de categorías y subcategorías.
  - Y más adelante puede que un producto pertenezca a varias categorías.

- Los productos tendrán cierto grado de complejidad. Esto quiere decir que no bastará con dos o tres atributos por producto. Al final del proyecto, se espera tener productos con, por ejemplo:
    - Identificadores, como código de producto interno y código universal (EAN, por ejemplo)
    - Nombre del producto.
    - Descripción del producto. Inicialmente será texto, pero posteriormente se ampliará la aplicación para que admita HTML.
    - Imágenes
    - Precio
    - Fabricante
    - Etc.

  Cada tienda o tipo de producto vendido puede necesitar de atributos específicos. Es parte del trabajo el analizar y diseñar las distintas clases.
- Aunque las primeras etapas se acometerán sólo con categorías y productos, habrá otras entidades, como pedidos, comentarios, valoraciones, usuarios, etc., que se irán incorporando al modelo de datos a medida que la aplicación crezca.
- También, a medida que avance la actividad, se podrán ir ampliando los atributos de las distintas clases.
- La página tendrá que ser multi idioma en lo que respecta al UI. Respecto a los datos, no se hará base de datos multi idioma.

Modificar la plantilla index.html para comenzar a dar una imagen corporativa.

Sugerencia: en Internet hay sitios web que ofrecen modelos de sitios e-commerce (y de otras temáticas) con maquetación completa, elaborada y profesional. Dado que el objetivo de este bloque es el desarrollo en servidor, se pueden usar estos sitios para conseguir una imagen corporativa atractiva sin tener que hacer el esfuerzo de maquetación CSS. En el caso de que se opte por esta opción, no puede haber dos equipos que usen la misma plantilla. A medida que se elija plantilla, se tendrán que notificar al profesor, y si otro equipo elige esa plantilla más tarde, no podrá hacerlo.

Respecto a la valoración de la maquetación, en la rúbrica habrá un criterio de calificación asociado, con poco peso, pero estará reflejado. Se valorará que la aplicación sea más o menos atractiva, y que responda bien en distintos navegadores, incluidos móviles. No se penalizará el uso de modelos de página obtenidos en Internet.

Todos los recursos utilizados para maquetación pueden:

- Ubicarse en el directorio o subdirectorios de resources/static, y enlazarlos a rutas relativas a la raíz del sitio.
- Enlazarse por CDN, si se trata de scripts o CSS de frameworks como Bootstrap o JQuery, que dispongan de CDN