

SEXTO LABORATORIO

CURSO: GESTIÓN DE SERVICIOS DE TICS [TEL137-H791]

SEMESTRE: 2025-2

Tema: Desarrollo de Aplicaciones Web con Spring Boot, CRUD y Dockerización Individual.

- Duración: 1 hora y 50 minutos.
- El laboratorio se realizará de manera individual.
- El entregable será el enlace del repositorio remoto del proyecto en github.
- Está terminantemente prohibido el uso de cualquier tipo de herramienta de inteligencia artificial para el desarrollo del laboratorio.

Recomendación: Leer todo el documento antes de empezar con el desarrollo del laboratorio.

Los estudiantes desarrollarán una aplicación web en Spring Boot para gestionar entradas de un blog personal. La aplicación deberá permitir operaciones CRUD (Crear, Leer, Actualizar, Eliminar) completas sobre las entradas del blog, utilizando una base de datos MySQL. La interfaz de usuario se construirá con Thymeleaf. La evaluación se centrará significativamente en la correcta creación de un Dockerfile para la aplicación y su capacidad para conectarse a una instancia de MySQL ejecutándose en un contenedor Docker separado.

Requisitos Funcionales:

1. Entidad BlogPost (Entrada de Blog):
 - Se debe definir una entidad BlogPost con los siguientes atributos mínimos:
 - ID (autogenerado, numérico, Long)
 - Título (texto, obligatorio, String)
 - Contenido (texto largo, obligatorio, String, usar @Lob o TEXT en DB)
 - Autor (texto, opcional, String)
 - Fecha de Publicación (fecha, autogestionada o manual, LocalDate o LocalDateTime)
 - Categoría (texto, opcional, String)
2. Listar Entradas del Blog (Read):
 - Una página principal (/posts o /) que muestre todas las entradas del blog en un formato resumido (ej. título, autor, fecha, y un extracto o enlace para leer más).
 - La tabla/listado debe mostrar al menos: Título, Autor, Fecha de Publicación.
 - Cada entrada listada debe tener opciones para "Ver Completo", "Editar" y "Eliminar".

- Debe haber un botón/enlace para "Crear Nueva Entrada".
- 3. Crear nueva entrada:
 - Un formulario (/posts/nuevo) para ingresar los datos de una nueva entrada de blog (Título, Contenido, Autor, Categoría). La fecha de publicación puede ser la actual por defecto.
 - Se deben aplicar validaciones básicas (ej: título y contenido no vacíos).
 - Al guardar, la entrada se persiste en la base de datos y se redirige al listado de entradas.
- 4. Ver entrada completa:
 - Al hacer clic en "Ver Completo" (o en el título de la entrada), se mostrará una página (/posts/{id}) con todos los atributos de la entrada seleccionada, mostrando el contenido completo.
- 5. Editar Entrada:
 - Un formulario (/posts/editar/{id}) pre-cargado con los datos de la entrada seleccionada.
 - Permitir la modificación de Título, Contenido, Autor, Categoría.
 - Al guardar los cambios, se actualiza la entrada en la base de datos y se redirige al listado o a la vista completa de la entrada.
- 6. Eliminar entrada:
 - Al hacer clic en "Eliminar" (ej. desde un POST a /posts/eliminar/{id}), se debe solicitar confirmación al usuario (puede ser un simple confirm() de JavaScript o una página de confirmación).
 - Tras la confirmación, la entrada se elimina de la base de datos y se redirige al listado de entradas.

Requisitos Técnicos y de Dockerización:

1. Spring Boot:
 - Utilizar Spring Boot para la estructura del proyecto.
 - Spring Web para controladores RESTful/MVC.
 - Spring Data JPA para la persistencia con la base de datos MySQL.
 - Spring Boot Starter Validation para validaciones de formulario.
 - Spring Boot DevTools (opcional, para desarrollo).
2. Base de datos:
 - Utilizar MySQL como sistema gestor de base de datos.
 - La aplicación Spring Boot debe estar configurada para conectarse a una instancia de MySQL. Para la evaluación, esta instancia de MySQL deberá estar ejecutándose en un contenedor Docker.
3. Thymeleaf:
 - Utilizar Thymeleaf para renderizar las vistas HTML.
 - Implementar fragmentos de Thymeleaf para partes comunes (ej. cabecera, pie de página, navegación) si se considera apropiado.

4. Dockerización:

- Dockerfile para la Aplicación Spring Boot:
 - Crear un Dockerfile en la raíz del proyecto para empaquetar la aplicación Spring Boot en una imagen Docker.
 - Debe utilizar una imagen base adecuada (e.g., openjdk:17-slim o similar).
 - Debe copiar el archivo JAR compilado de la aplicación (generado por mvn package o gradle build).
 - Debe exponer el puerto en el que corre la aplicación Spring Boot (e.g., 8080).
 - La aplicación debe poder iniciarse con `java -jar app.jar` (o el nombre de su JAR).

5. Conexión a Contenedor MySQL:

- La aplicación Spring Boot, cuando se ejecute dentro de su contenedor Docker, debe poder conectarse a un contenedor MySQL que se ejecutará de forma separada.
- La configuración de la base de datos en `application.properties` (o `application.yml`) debe permitir especificar el host, puerto, nombre de la base de datos, usuario y contraseña. Se recomienda enfáticamente el uso de variables de entorno para configurar la URL de la base de datos, usuario y contraseña al ejecutar el contenedor de la aplicación.
- Por ejemplo, la URL de conexión podría ser `jdbc:mysql://${DB_HOST:mysql-db}:${DB_PORT:3306}/${DB_NAME}` y estas variables (`DB_HOST`, `DB_PORT`, `DB_NAME`, `DB_USER`, `DB_PASSWORD`) se pasarían al ejecutar el contenedor de la aplicación.

Vistas sugeridas:

1. List Blog Posts

Mi Blog Personal

Entradas del Blog

Crear Nueva Entrada

No hay entradas en el blog todavía. ¡Crea una!

Título de la Entrada

Fecha

Autor: Autor

Categoría: Categoría

Extracto del contenido...

Ver Completo

Editar

Eliminar

© 2025 Mi Blog Personal - Laboratorio GTICS

2. Create/Edit Blog Post Form

Mi Blog Personal

Formulario de Entrada

Título *

Contenido *

Autor

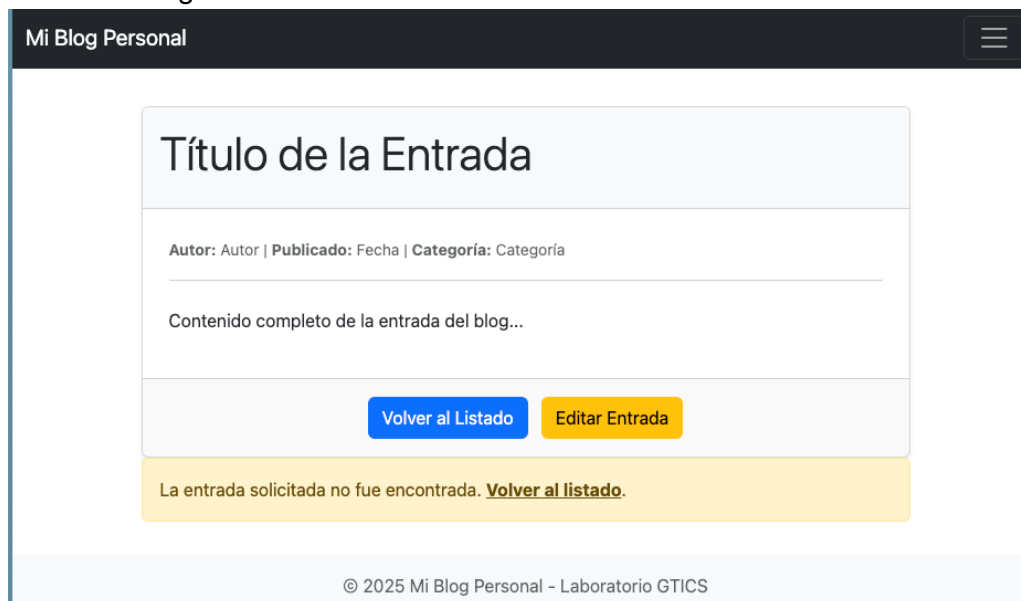
Categoría

Guardar Entrada

Cancelar

© 2025 Mi Blog Personal - Laboratorio GTICS

3. View Full Blog Post



Entrega:

- Repositorio GitHub: Incluye el enlace a tu repositorio con todos los archivos necesarios.
- Su repositorio debe ser nombrado de la siguiente manera: LAB8_GTICS_20251_XXXXXX donde 'XXXXXX' es el código de alumno, por ejemplo: LAB8_GTICS_20251_20160877
- Su repositorio en github debe contar con 2 ramas, 'main' y 'develop' su desarrollo lo debe hacer en la rama develop y al final el laboratorio debe realizar un 'pull request' desde develop hacia main tal como se explicó en el LAB 1.

Tips:

- Realiza commits frecuentes para mantener un buen control de versiones.
- Primero, asegúrate de que tu aplicación Spring Boot funciona localmente con una instancia de MySQL (instalada localmente o en Docker).
- Luego, enfócate en crear el Dockerfile para tu aplicación.
- Finalmente, prueba la configuración completa con ambos contenedores (app y MySQL) comunicándose a través de Docker, preferiblemente usando una red Docker definida por el usuario.