Desarrollo avanzado de aplicaciones II

Tema Nº8:Integración MVC, JPA y Security.

Indicador de logro Nº8:

Consolida los conocimientos de MVC, Spring Data JPA y Security aplicando las buenas prácticas para el desarrollo de software.

**TEMA 01 Teoría de los**

Imagen que contiene Icono

Descripción generada automáticamente

**TEMA Nº8:**

**Integración MVC, JPA y Security.**

**Subtema 8.1:**

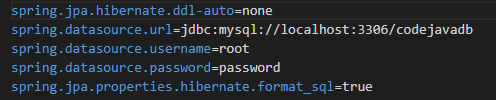
Registro de usuarios y encriptar passwords con el algoritmo BCrypt.

**Ejemplo:**

Creamos un nuevo proyecto en Spring Boot con las siguientes dependencias:

* Spring Data JPA
* MySQL JDBC
* Thymeleaf.
* Spring Boot Web.
* Spring Boot Web Tools.
* Spring Security.

Agregamos la cadena de conexión:



Crear una página html con el siguiente contenido:

Texto

Descripción generada automáticamente Texto

Descripción generada automáticamente

Crear la clase User y agregamos los siguientes atributos.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

Crear una clase Java con el nombre WebSercurityConfig y añadimos lo siguiente:

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

Crear una interface con el nombre de UserRepository y añadimos lo siguiente:

Texto

Descripción generada automáticamente

Crear una clase CustomUserDetails e implementamos la interface UserDetails

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Crear una clase CustomUserDetailsService y añadir lo siguiente:

Texto

Descripción generada automáticamente

Finalmente, en el controller añadir lo siguiente:

Texto

Descripción generada automáticamente

**Subtema 8.2:**

Thymeleaf sec:authorize="isAnonymous()".

En entornos Spring MVC, el módulo de integración Spring Security funciona como un reemplazo del taglib de seguridad Spring.

Usamos este dialecto en el ejemplo para imprimir las credenciales del usuario registrado y mostrar contenido diferente a roles diferentes.

El atributo sec: authorize muestra su contenido cuando la expresión del atributo se evalúa como verdadera.

**Subtema 8.3:**

Thymeleaf sec:authorize="isAuthenticated()".

**Ejemplos:**

Texto

Descripción generada automáticamente

**Subtema 8.4:**

Cerrar la sesión del usuario en la aplicación

**Ejemplos:**

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Actividad:**

a) CUESTIONARIO TÉCNICO

* ¿Qué dependencias necesita nuestro proyecto para añadir Spring Security?
* ¿Qué desventajas tiene trabajar con Spring Security?
* ¿Existe otra propiedad de seguridad para Thymeleaf ?
* ¿Para qué sirve sec:authorize de Thymeleaf ?
* Cree un nuevo proyecto con Spring Boot y realice un registro de usuarios utilizando Spring Security con base de datos MySQL.

b) CONCLUSIONES DE LA EXPERIENCIA

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_