

GA8-220501096-AA1-EV01

Desarrollar software a partir de la integración de sus módulos componentes



Presentado por:

Michelle Rojas Bolaños

Presentado a:

Edward Javier García Corredor

Instructor área Técnica

Servicio nacional de aprendizaje – SENA

Tecnología en análisis y desarrollo de software

Ficha 2834865

Mayo 2025

Introducción

El presente documento tiene como finalidad evidenciar el desarrollo y la integración de los módulos componentes de una aplicación web desarrollada en java, utilizando el framework spring boot. El proyecto, denominado TareAppSpring, fue construido aplicando el patrón de arquitectura mvc, implementando funcionalidades de autenticación de usuario y gestión de tareas, con vistas construidas mediante el motor de plantillas thymeleaf y persistencia de datos con jpa.

Objetivos

Objetivo general:

Desarrollar una aplicación web funcional basada en spring boot, que integre módulos componentes para el registro de usuarios, autenticación y gestión de tareas.

Objetivos específicos:

- Implementar el patrón mvc para garantizar la separación de responsabilidades.
- Diseñar y codificar los módulos de usuario y tareas.
- Integrar la seguridad básica mediante sesiones http.
- Permitir operaciones crud desde una interfaz web construida con thymeleaf.

Tecnologías utilizadas

- | | |
|-------------------|-------------------|
| • Java 17 | • Jpa (hibernate) |
| • Spring boot 3.x | • Maven |
| • Thymeleaf | • Bootstrap 5 |
| • Mysql | • Git / github |

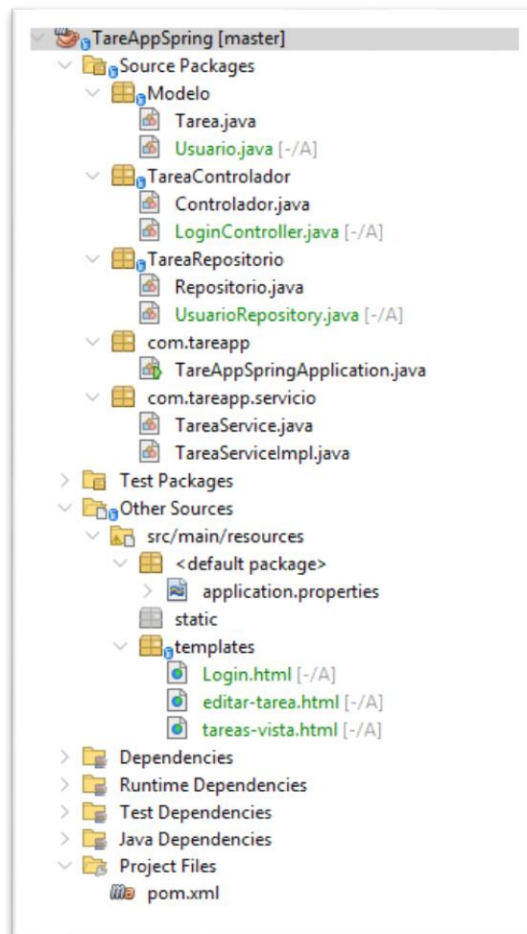
Descripción del sistema

Tareappspring es una aplicación web que permite a los usuarios autenticados gestionar tareas personales. Cuenta con las siguientes funcionalidades principales:

- Iniciar sesión mediante usuario y contraseña registrados en base de datos.
- Visualizar la lista de tareas personales.
- Agregar nuevas tareas.
- Editar tareas existentes.
- Eliminar tareas.
- Cerrar sesión de forma segura.

Estructura del proyecto

- Modelo: clases usuario y tarea anotadas como entidades jpa.
- Repositorio: interfaces que extienden JpaRepository para acceso a datos.
- Controlador: clase logincontroller que gestiona rutas y vistas.
- Vista: archivos .html con thymeleaf (login.html, tareas-vista.html, editar-tarea.html).



Diagramas del sistema

Diagrama de clases:

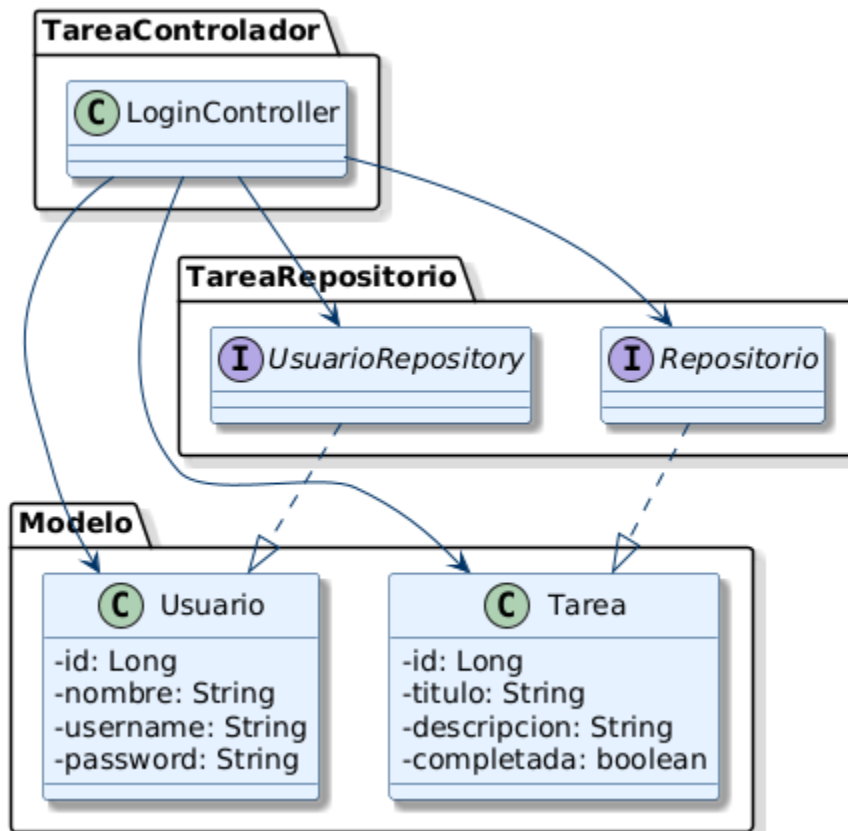


Diagrama de paquetes:

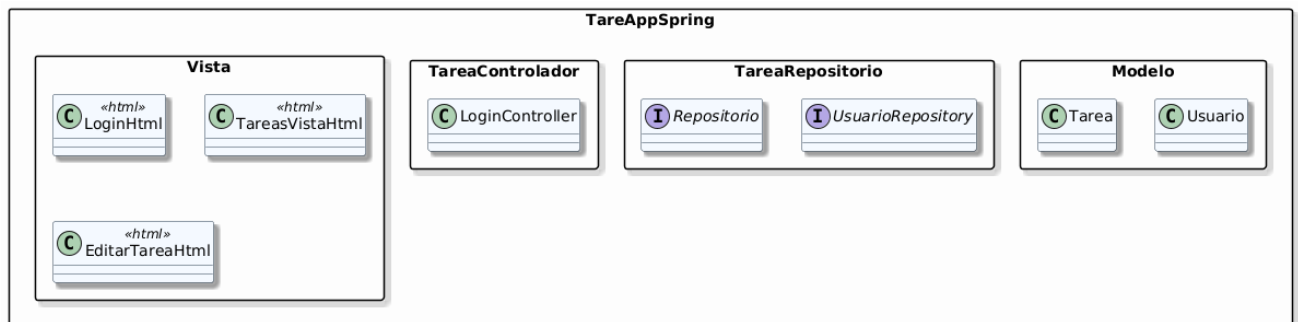
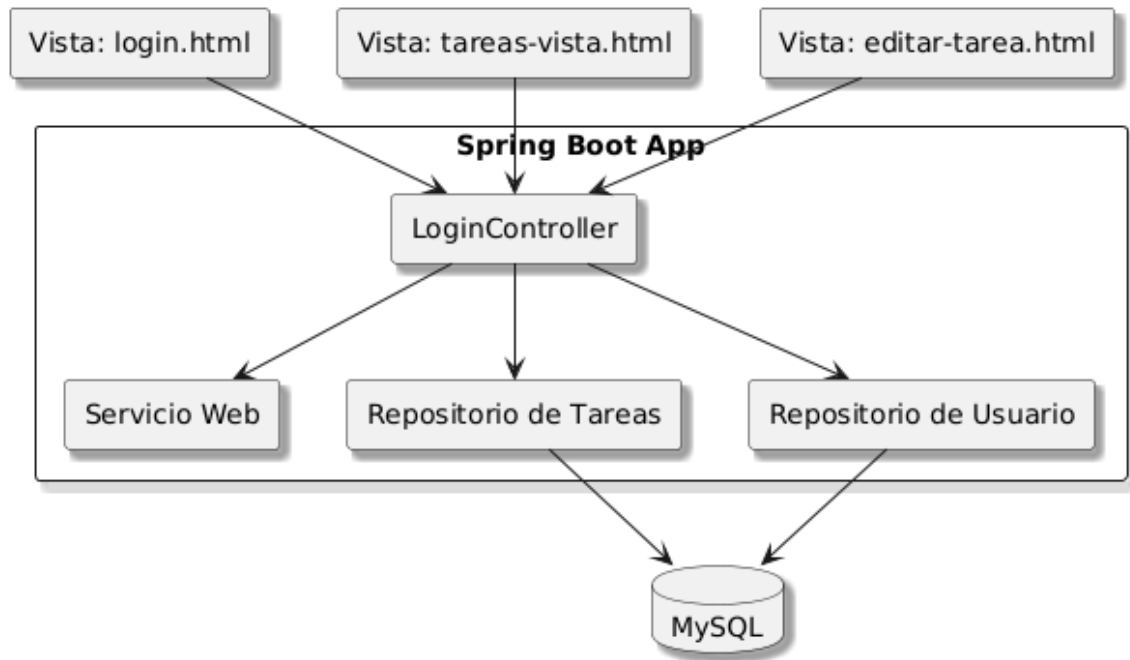
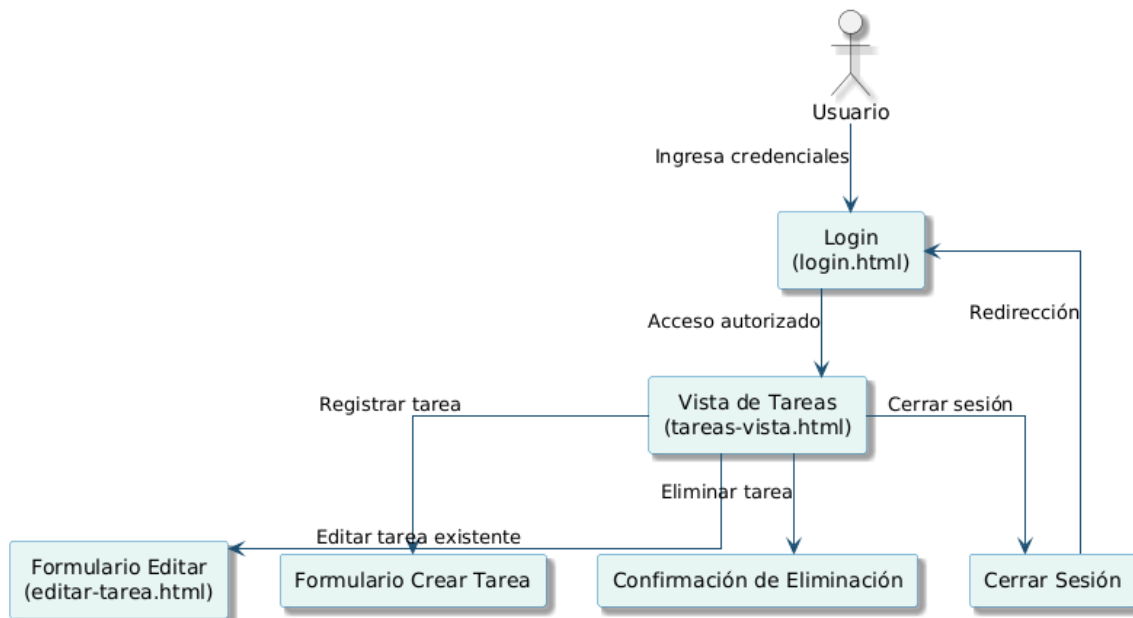


Diagrama de componentes:



Mapa de navegacion:



Casos de uso y funcionalidades

Caso de Uso 1: Iniciar sesión

Elemento	Descripción
Nombre	Iniciar sesión
Actor principal	Usuario registrado
Precondición	El usuario debe estar registrado en la base de datos
Flujo principal	1. El usuario accede al formulario de login.2. Ingresa credenciales.3. El sistema valida los datos.4. Si son correctos, redirige al módulo de tareas.
Postcondición	El usuario queda autenticado en sesión y accede al panel principal.

Caso de Uso 2: Consultar tareas

Elemento	Descripción
Nombre	Consultar lista de tareas
Actor principal	Usuario autenticado
Precondición	El usuario debe haber iniciado sesión
Flujo principal	1. El usuario accede al panel principal.2. El sistema recupera las tareas.3. Se muestran en tabla con sus respectivos datos.
Postcondición	El usuario visualiza todas las tareas almacenadas.

Caso de Uso 3: Registrar nueva tarea

Elemento	Descripción
Nombre	Registrar nueva tarea
Actor principal	Usuario autenticado
Precondición	El usuario debe haber iniciado sesión
Flujo principal	1. El usuario completa el formulario de tarea.2. Envía el formulario.3. El sistema guarda los datos.4. Se actualiza la lista.
Postcondición	La nueva tarea queda registrada correctamente.

Caso de Uso 4: Editar tarea

Elemento	Descripción
Nombre	Editar tarea existente
Actor principal	Usuario autenticado
Precondición	Debe existir al menos una tarea registrada
Flujo principal	1. El usuario selecciona una tarea para editar.2. El sistema carga los datos.3. El usuario actualiza la información.4. El sistema guarda los cambios.
Postcondición	Los cambios se reflejan correctamente en la lista.

Caso de Uso 5: Eliminar tarea

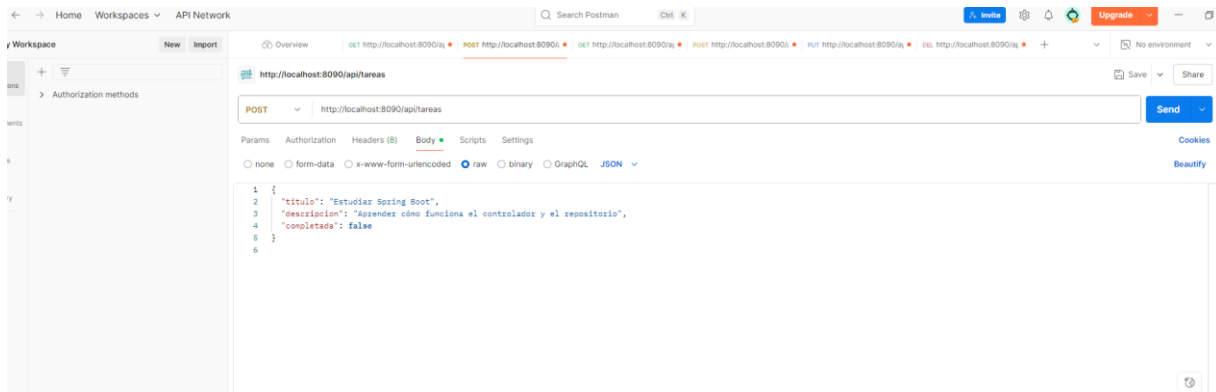
Elemento	Descripción
Nombre	Eliminar tarea existente
Actor principal	Usuario autenticado
Precondición	Debe haber tareas previamente registradas
Flujo principal	1. El usuario hace clic en “Eliminar”.2. El sistema solicita confirmación.3. La tarea es eliminada de la base de datos.
Postcondición	La tarea ya no aparece en la vista ni en la base de datos.

Historias de usuario

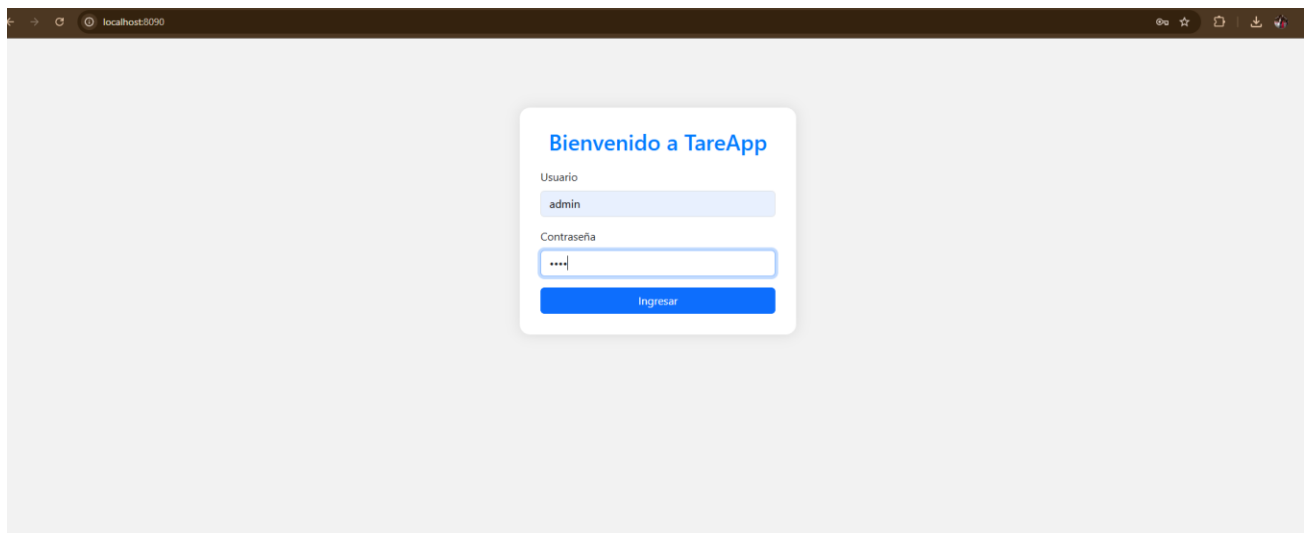
ID	Historia de Usuario
HU1	Como usuario registrado, quiero iniciar sesión en el sistema, para acceder a mis tareas.
HU2	Como usuario autenticado, quiero visualizar mis tareas, para saber cuáles tengo pendientes o completadas.
HU3	Como usuario autenticado, quiero agregar nuevas tareas, para mantener organizadas mis actividades.
HU4	Como usuario, quiero editar una tarea existente, para actualizar su información si cambia.
HU5	Como usuario, quiero eliminar una tarea, para mantener mi lista limpia y actualizada.

Pruebas y resultados

- Todas las funcionalidades fueron probadas desde el navegador y también con postman.



- Capturas de pantalla:
 - Login correcto



- Vista de tareas cargadas desde base de datos

Cerrar sesión

Tareas del Usuario

Bienvenido, ¡has iniciado sesión correctamente!

Agregar nueva tarea

Título

Descripción

Pendiente

Agregar

ID	Título	Descripción	¿Completada?	Acciones
1	Estudiar Spring Boot	Aprender cómo funciona el controlador, repositorio y el servicio	Sí	<div>Editar</div> <div>Eliminar</div>
3	Estudiar Spring Boot	Aprender cómo funciona el controlador y el repositorio	Sí	<div>Editar</div> <div>Eliminar</div>
4	Estudiar Spring Boot	Agregar nuevas tareas desde el formulario visual	No	<div>Editar</div> <div>Eliminar</div>

- Agregar tarea

Cerrar sesión

Tareas del Usuario

Bienvenido, ¡has iniciado sesión correctamente!

Agregar nueva tarea

Título

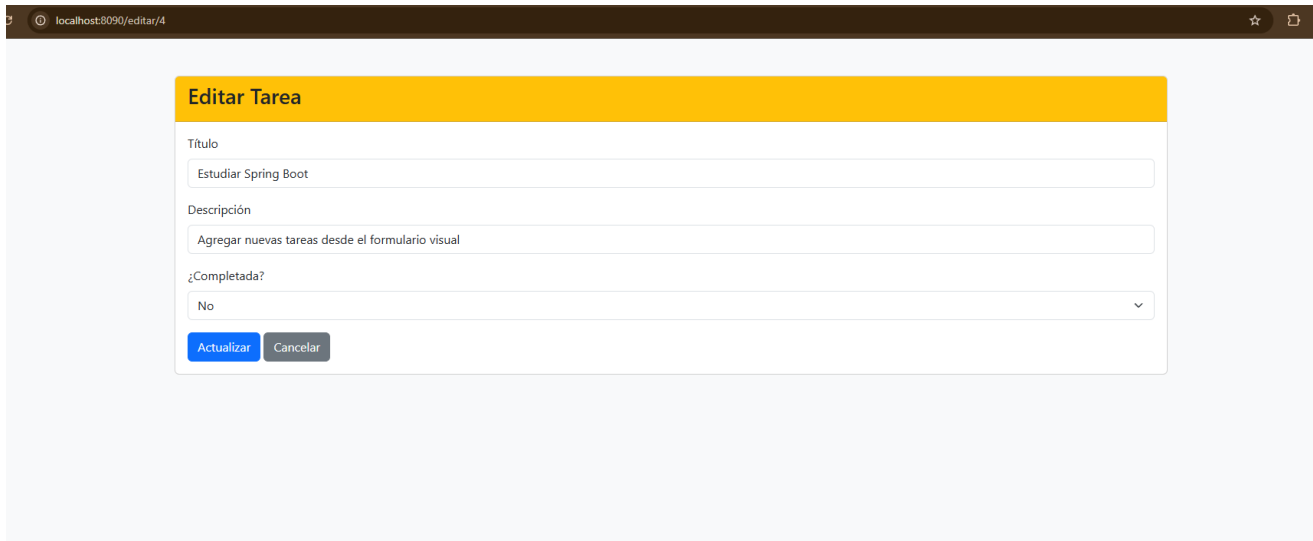
Descripción

Pendiente

Agregar

ID	Título	Descripción	¿Completada?	Acciones
1	Estudiar Spring Boot	Aprender cómo funciona el controlador, repositorio y el servicio	Sí	<div>Editar</div> <div>Eliminar</div>
3	Estudiar Spring Boot	Aprender cómo funciona el controlador y el repositorio	Sí	<div>Editar</div> <div>Eliminar</div>
4	Estudiar Spring Boot	Agregar nuevas tareas desde el formulario visual	No	<div>Editar</div> <div>Eliminar</div>
5	Estudiar Spring Boot	Agregar capturas de pantalla	Sí	<div>Editar</div> <div>Eliminar</div>

- Editar tarea



Editar Tarea

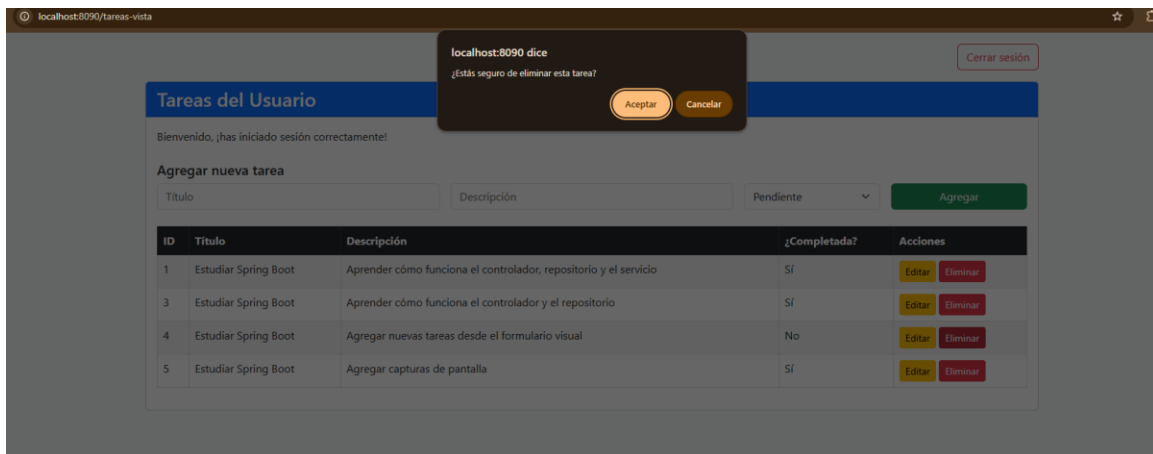
Título
Estudiar Spring Boot

Descripción
Agregar nuevas tareas desde el formulario visual

¿Completada?
No

Actualizar Cancelar

- Eliminar tarea



localhost:8090 dice
¿Estás seguro de eliminar esta tarea?

Aceptar Cancelar

Tareas del Usuario

Bienvenido, ¡has iniciado sesión correctamente!

Agregar nueva tarea

Título Descripción Pendiente Agregar

ID	Título	Descripción	¿Completada?	Acciones
1	Estudiar Spring Boot	Aprender cómo funciona el controlador, repositorio y el servicio	Sí	Editar Eliminar
3	Estudiar Spring Boot	Aprender cómo funciona el controlador y el repositorio	Sí	Editar Eliminar
4	Estudiar Spring Boot	Agregar nuevas tareas desde el formulario visual	No	Editar Eliminar
5	Estudiar Spring Boot	Agregar capturas de pantalla	Sí	Editar Eliminar

Cerrar sesión

Ambiente de desarrollo

El desarrollo de la aplicación TareAppSpring se llevó a cabo en un entorno configurado con herramientas modernas y ampliamente utilizadas en el desarrollo de

software basado en el lenguaje Java y la arquitectura MVC. A continuación, se detallan los componentes del ambiente de desarrollo:

Elemento	Descripción
Lenguaje de programación	Java 17 (compatible con las versiones requeridas por Spring Boot 3.x)
Framework principal	Spring Boot (versión 3.x) para facilitar la configuración, el empaquetado y la ejecución
Motor de plantillas	Thymeleaf, utilizado para generar vistas dinámicas y amigables al usuario
ORM / Persistencia	JPA (Hibernate como proveedor), para mapear clases Java a tablas de la base de datos
Base de datos	MySQL, configurada con credenciales locales para la conexión y prueba de persistencia
IDE de desarrollo	Spring Tool Suite (STS) o IntelliJ IDEA, compatibles con Maven y Spring
Gestor de dependencias	Maven, utilizado para gestionar las bibliotecas necesarias desde el archivo <code>pom.xml</code>
Servidor embebido	Apache Tomcat (integrado en Spring Boot), desplegado en el puerto 8090
Control de versiones	Git, con repositorio alojado en GitHub para facilitar la trazabilidad y el trabajo modular
Pruebas funcionales	Postman (para pruebas de API) y navegador web (para validar el comportamiento visual de la app)
Sistema operativo	Windows 10 de 64 bits (ambiente local de desarrollo y pruebas)

Este entorno permitió desarrollar una solución modular, escalable y fácil de mantener, cumpliendo con los estándares de desarrollo orientado a objetos y las buenas prácticas sugeridas por el SENA para la evidencia GA8-220501096-AA1-EV01.

Control de versiones

El proyecto TareAppSpring fue gestionado mediante control de versiones utilizando Git como herramienta principal, y GitHub como repositorio remoto para respaldo, trazabilidad y colaboración.

A continuación, se detallan los elementos clave de su implementación:

Elemento	Detalle
Sistema de control	Git
Plataforma remota	GitHub
Repositorio del proyecto	https://github.com/Verityxnova/TareAppSpring
Tipo de repositorio	Público (visible para evaluación y pruebas)
Estructura organizada	Proyecto dividido en paquetes por función: Modelo, Controlador, Repositorio, Vista
Ramas utilizadas	Rama principal (main) con actualizaciones directas y controladas
Commits significativos	Cada avance importante fue registrado con un mensaje descriptivo (ej. “Implementación de LoginController”, “CRUD de tareas completo”, “Integración de Thymeleaf”)
Ventajas obtenidas	Trazabilidad de cambios, respaldo en la nube, entrega remota rápida y segura

Conclusiones

El desarrollo de tareappspring ha permitido aplicar de forma práctica los conocimientos adquiridos en arquitectura de software, programación en java y el uso de frameworks modernos. La correcta integración de los componentes, el uso de control de versiones y la aplicación de buenas prácticas ha resultado en una solución funcional, escalable y fácil de mantener.

Referencias

Thymeleaf. (s.f.). *Thymeleaf: Modern server-side Java template engine for web and standalone environments*. <https://www.thymeleaf.org/>

Spring. (s.f.). *Spring Initializr*. <https://start.spring.io>

Oracle. (s.f.). *Jakarta Persistence API (JPA) - Documentación oficial*.
<https://jakarta.ee/specifications/persistence/>

Bootstrap. (s.f.). *Bootstrap: Framework CSS para desarrollo responsive*.
<https://getbootstrap.com>