

República Dominicana
Pontificia Universidad Católica Madre y Maestra
Facultad de ingenierías
Escuela de computación y telecomunicaciones
Proyecto final



Presentado por:

Lenin E. Gutiérrez S.

Matrícula:

2019-0010

Materia:

ISC-417

Profesor:

Carlos Camacho

Santiago de los caballeros

29 de agosto del 2024

Introducción

Este proyecto final surge como un esfuerzo integral para desarrollar una aplicación web que gestione usuarios y formularios utilizando Javalin como framework principal, junto con MongoDB como base de datos. La elección de estas tecnologías se fundamenta en su capacidad para manejar de manera eficiente peticiones y respuestas HTTP, así como para facilitar la persistencia de datos en una base de datos NoSQL.

El proyecto no solo se centra en la funcionalidad básica de la aplicación, sino que también incorpora buenas prácticas de programación, incluyendo la utilización de un ORM (Object-Relational Mapping) para la interacción con MongoDB y la implementación de un sistema robusto de gestión de usuarios y formularios. Se ha buscado asegurar la modularidad del código y la escalabilidad de la aplicación, todo dentro de un entorno de desarrollo ágil y con plazos ajustados.

En este informe se detallará el proceso completo de desarrollo, desde la configuración inicial del entorno de trabajo, pasando por la creación y mapeo de entidades en Java, hasta la integración del frontend con el backend. Además, se abordarán los desafíos técnicos encontrados y las soluciones implementadas, incluyendo la gestión de la conexión con MongoDB, la configuración de rutas en Javalin, y la adaptación de plantillas HTML para una experiencia de usuario óptima tanto en dispositivos móviles como en escritorio.

Desarrollo del tema

1. Configuración Inicial y Elección de Tecnologías

El proyecto comenzó con la selección cuidadosa de las tecnologías que se utilizarían a lo largo de su desarrollo. La decisión de emplear Javalin como el framework principal se basó en su ligereza y flexibilidad, ideales para construir aplicaciones web modernas sin la complejidad adicional de frameworks más pesados. Además, Javalin ofrece una integración sencilla con diversas bibliotecas, lo que facilitó la conexión con MongoDB y la manipulación de datos en un entorno NoSQL.

2. Mapeo de Entidades y Configuración de la Conexión a MongoDB

Uno de los primeros desafíos fue configurar correctamente la conexión a MongoDB desde una aplicación Java. Se implementó la clase `MongoClientConnection`, encargada de establecer y mantener la conexión con la base de datos. Este proceso involucró la configuración de la cadena de conexión, la selección del servidor API y la implementación de un patrón Singleton para asegurar que solo hubiera una instancia de conexión activa en cualquier momento dado. Este enfoque evitó problemas de sobrecarga del servidor y aseguró una gestión eficiente de los recursos.

En cuanto al mapeo de entidades, se desarrollaron tres clases principales: `Usuario` y `Formulario`. Inicialmente, `Usuario` incluía un campo `ObjectId`, que se utilizaba como identificador único en la base de datos. Sin embargo, este diseño se revisó y modificó para utilizar el campo `usuario` como identificador principal, simplificando las operaciones de búsqueda y edición.

La clase `Formulario` también fue objeto de múltiples iteraciones, especialmente con la adición de los campos `latitud` y `longitud`, que permitían almacenar la ubicación geográfica donde se completaba el formulario. Este enfoque innovador aprovechó las capacidades de HTML5 para capturar la ubicación del usuario, una funcionalidad que requería una integración cuidadosa entre el frontend y el backend.

3. Implementación de Servicios y CRUD Completo

Con las entidades mapeadas y la conexión a MongoDB establecida, el siguiente paso fue implementar los servicios que gestionarían las operaciones CRUD (Crear, Leer, Actualizar, Borrar) para Usuario y Formulario. Se diseñaron métodos específicos para cada una de estas operaciones, asegurando que la aplicación pudiera manejar todas las necesidades de gestión de datos de manera eficiente.

Crear y Leer: La creación y lectura de usuarios y formularios se implementó utilizando métodos en `UsuarioService` y `FormularioService`. Estos métodos interactuaban directamente con MongoDB, transformando los documentos de la base de datos en objetos Java y viceversa. Se prestó especial atención a la validación de los datos ingresados, asegurando que solo se almacenaran entradas válidas y completas en la base de datos.

Actualizar: La actualización de registros presentó un desafío particular, dado que requería una combinación de operaciones de lectura y escritura. Se implementaron métodos para recuperar los registros existentes, realizar modificaciones y guardarlos nuevamente en la base de datos, asegurando que se mantenía la integridad de los datos en todo momento.

Eliminar: La eliminación de registros fue implementada con cautela, asegurando que cualquier eliminación se realizara de manera segura y que los datos asociados (como los formularios vinculados a un usuario) se manejaran adecuadamente para evitar inconsistencias en la base de datos.

4. Desarrollo del Frontend y Conexión con el Backend

El desarrollo del frontend implicó la selección y adaptación de una plantilla HTML preexistente. Se eligió una plantilla que, aunque no perfecta, proporcionaba una base estética sobre la cual construir. Esta plantilla se integró cuidadosamente con el backend utilizando Thymeleaf, un motor de plantillas que permitió la renderización dinámica de las páginas HTML con los datos provenientes de la base de datos.

Se desarrollaron varias vistas clave, incluyendo:

Listado de Usuarios: Una página que mostraba todos los usuarios registrados, con opciones para editar o eliminar cada registro. Esta vista fue diseñada para ser responsiva, asegurando una buena usabilidad tanto en dispositivos móviles como en pantallas de escritorio.

Formulario de Usuarios y Formularios: Estas vistas permitían la creación y edición de usuarios y formularios, respectivamente. Se implementó lógica en el frontend para manejar los modos de visualización y edición, asegurando que los usuarios solo pudieran modificar los campos cuando fuera apropiado.

La conexión entre el frontend y el backend se manejó a través de rutas definidas en Javalin. Estas rutas gestionaban las solicitudes HTTP y facilitaban la interacción entre el navegador del usuario y la base de datos, permitiendo operaciones fluidas y seguras.

5. Integración de Geolocalización y Otros Desafíos

Una de las características más destacadas del proyecto fue la integración de la funcionalidad de geolocalización en los formularios. Utilizando las capacidades de HTML5, se capturaba la latitud y longitud del usuario al completar un formulario. Esta información se almacenaba junto con los demás datos del formulario en MongoDB, permitiendo un análisis geográfico de los datos recopilados.

Este proceso no estuvo exento de desafíos. Se encontraron y resolvieron varios problemas relacionados con la captura y almacenamiento de datos de ubicación, incluyendo errores de validación de datos y problemas de compatibilidad entre las diferentes versiones de navegadores y dispositivos.

6. Pruebas y Validación

La última fase del proyecto consistió en realizar pruebas exhaustivas para garantizar la funcionalidad completa del sistema. Esto incluyó la validación de datos en el frontend y backend, pruebas de integración para asegurar que todas las partes del sistema funcionaran juntas sin problemas, y la resolución de errores y excepciones que surgieron durante las pruebas.

Se abordaron problemas específicos como la conversión de identificadores, errores de null pointers, y dificultades en la renderización de plantillas, asegurando que el sistema final fuera robusto y confiable.

Conclusión

Este proyecto demostró ser un ejercicio integral en el desarrollo de aplicaciones web modernas. Desde la configuración inicial del entorno de trabajo hasta la implementación de funcionalidades avanzadas como la geolocalización, cada paso requirió una cuidadosa planificación y ejecución.

La experiencia adquirida a lo largo de este proceso ha consolidado una base sólida de conocimientos en programación web, gestión de bases de datos NoSQL, y desarrollo de interfaces de usuario responsivas.

Este proyecto es un testimonio de la capacidad para superar desafíos técnicos y entregar una solución completa que integra las mejores prácticas de desarrollo de software, asegurando tanto la funcionalidad como la escalabilidad del sistema.