

## **Anteproyecto de Trabajo Fin de Carrera**

### **Desarrollo de una aplicación web para la visualización de contenidos multimedia.**



Universidad  
de Alcalá

David Bachiller Vela

Grado en Ingeniería de Computadores  
Escuela Politécnica  
Universidad de Alcalá  
`david.bachiller@edu.uah.es`

## 1 Introducción

La aplicación es una API REST construida con Spring Boot y Java. Utiliza Maven como herramienta de gestión de proyectos. La API se centra en la gestión de usuarios, películas y comentarios y proporciona varias operaciones CRUD (Crear, Leer, Actualizar, Eliminar) a través de diferentes endpoints HTTP.

También contempla la existencia de un usuario administrador, que puede realizar estas operaciones CRUD con los usuarios, las películas y los comentarios. El resto de los usuarios solo podrán visualizar y comentar el contenido.

El flujo de información sigue el patrón Controlador -> Servicio -> Repositorio -> Entidades. Cuando se realiza una solicitud, por ejemplo, para eliminar un usuario, la solicitud llega primero al controlador. El controlador luego llama al servicio correspondiente. El servicio utiliza el repositorio para realizar la operación de eliminación en la base de datos. Finalmente, el resultado se devuelve al cliente a través del controlador.

Se ha elegido Spring Boot y no otras tecnologías como puede ser Jakarta EE debido a que lo que hace Spring Boot es gestionar las librerías que vamos a necesitar de forma automática, simplemente diciéndole que tipo de proyecto a grandes rasgos vamos a crear y el y se encarga automáticamente de según sea la aplicación usar unas librerías u otras, también nos ayuda teniendo un servidor apache tomcat embebido y solamente con un comando podemos desplegar nuestro proyecto y correrlo en el servidor tomcat. Lo que nos permite centrarnos en el desarrollo de la aplicación y hace que crear una aplicación sea mucho más rápido.

Las dependencias que mencionas son componentes clave en un proyecto Spring Boot. Las principales dependencias que se van a usar en el proyecto son:

**Spring Web:** Nos proporciona funcionalidades para construir aplicaciones web, incluyendo RESTful, utilizando Spring MVC. También ayuda en la manipulación de solicitudes web.

**Spring Data JPA:** Simplifica la implementación de la capa de persistencia de datos. Nos proporciona un marco para implementar repositorios que reducen la cantidad de código de boilerplate.

**MySQL Driver:** Es el conector JDBC que nos va a permitir interactuar con la base de datos MySQL.

**Spring Boot Dev Tools:** Nos proporciona herramientas de desarrollo útiles, como el reinicio automático de la aplicación cada vez que se cambia un archivo.

**Spring Security:** Es un marco de seguridad que proporciona autenticación, autorización y protección contra ataques comunes.

**Spring Session:** Proporciona una API y una implementación para administrar la sesión del usuario y del administrador.

**Thymeleaf:** Es un motor de plantillas Java para el desarrollo web en el lado del servidor. Nos va a facilitar crear vistas dinámicas utilizando HTML estándar.

Spring Framework Documentation: <https://docs.spring.io/spring-framework/docs/current/reference/html/>

Spring Data JPA Documentation: <https://docs.spring.io/spring-data/jpa/docs/current/reference/html/>

MySQL Connector/J Developer Guide: <https://dev.mysql.com/doc/connector-j/8.0/en/>

Spring Boot Developer Tools Documentation: <https://docs.spring.io/spring-boot/docs/current/reference/html/using.html#using.devtools>

Spring Security Reference: <https://docs.spring.io/spring-security/site/docs/current/reference/html5/>

Spring Session Reference: <https://docs.spring.io/spring-session/docs/current/reference/html5/>

Thymeleaf Documentation: <https://www.thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html>

## 2 Objetivos

### Objetivo General:

El propósito general de este proyecto es desarrollar una aplicación web basada en Java y Spring Boot que permita la gestión de usuarios, de películas y de comentarios de los propios usuarios. Esta aplicación proporcionará una interfaz RESTful para realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar).

### Objetivos Específicos:

Implementar un controlador REST (UserController) que maneje las solicitudes HTTP y proporcione respuestas adecuadas. Este controlador incluirá métodos para crear, leer, actualizar y eliminar usuarios, así como para leer todos los usuarios. Lo mismo con las películas.

Desarrollar un servicio (UserService) que contenga la lógica de negocio y se comunique con la capa de repositorio para realizar operaciones CRUD en la base de datos.

Crear un repositorio (UserRepository) que interactúe con la base de datos utilizando JPA (Java Persistence API). Este repositorio será responsable de realizar las operaciones de base de datos.

Definir una entidad (User) que represente la tabla de usuarios en la base de datos. Esta entidad será utilizada por el repositorio para mapear las operaciones de base de datos.

Integrar y configurar las dependencias necesarias para el proyecto, incluyendo Spring Web, Spring Data JPA, MySQL Driver, Spring Boot Dev Tools, Spring Security,

Spring Session y Thymeleaf. Estas dependencias proporcionarán las funcionalidades necesarias para construir la aplicación.

## 4 Metodología

La metodología que se seguirá para este proyecto es la metodología ágil Scrum, que es una de las metodologías de desarrollo de software más utilizadas. Scrum es especialmente adecuada para proyectos con requisitos rápidamente cambiantes, como es común en el desarrollo de software.

**Planificación del Sprint:** Al inicio de cada sprint, haré una planificación del trabajo a realizar durante el sprint.

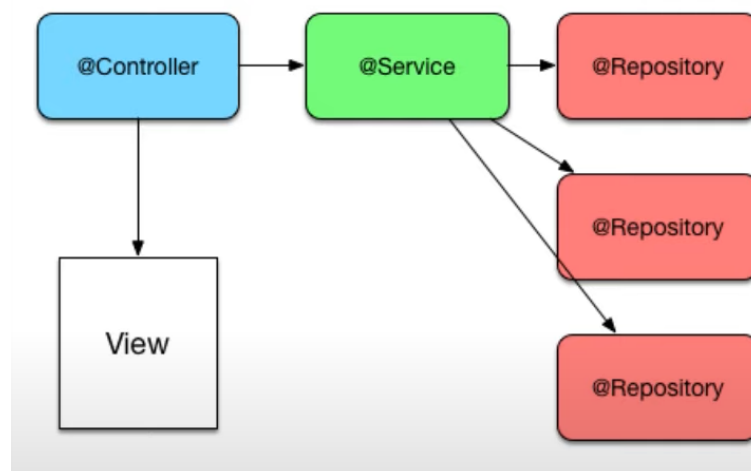
**Sprints:** Cada sprint tendrá una duración semanal, en la que se trabajará en la tarea seleccionada, también incluyendo la fase de investigación.

**Reuniones de Scrum:** En este caso es un trabajo individual con un tutor, por lo que cada dos semanas se le enseñaran los avances y las nuevas ideas propuestas, a la vez este mismo podrá ver mediante commits de github el proceso diario.

**Revisión del Sprint:** Al final de cada sprint, se hará una revisión del trabajo realizado durante el sprint y demostrar el producto incrementado.

**Retrospectiva del Sprint:** Se llevará un control diario en la memoria de lo que no funcionó bien y cómo mejorar en el próximo sprint.

En cuanto a la metodología que se va a seguir en el proyecto:  
Controlador -> Servicio -> Repositorio -> Entidades.



Ya que se trata de una Api Rest, la metodología siempre va a ser similar tanto con los usuarios como con las películas:

Va a ver un cliente que mediante la vista solicite o realice acciones SCRUM de los usuarios y de las películas a la Api Rest, esta mediante el proceso de Controlador -> Servicio -> Repositorio -> Entidades, buscara o almacenara dependiendo de lo que los clientes quieran hacer esos datos en la Base de datos MySQL.



## 5 Recursos

Para llevar a cabo este proyecto, se necesitarán los siguientes recursos.

### Recursos de Software:

**Java:** Es el lenguaje de programación principal utilizado en este proyecto. Se necesita una versión reciente de Java, preferiblemente Java 11 o superior.

**Spring Boot:** Es el marco de trabajo principal para el desarrollo de la aplicación. Proporciona una forma rápida y fácil de configurar y arrancar una aplicación web basada en Java.

**Maven:** Es una herramienta de gestión de proyectos y comprensión de software. Se utiliza para manejar las dependencias del proyecto.

**IntelliJ IDEA:** Es el entorno de desarrollo integrado (IDE) utilizado para escribir y probar el código.

**MySQL:** Es el sistema de gestión de bases de datos que se utilizará para almacenar los datos de la aplicación.

Git y GitHub: Se utilizarán para el control de versiones y la colaboración en el código.

**Recursos de Hardware:**

Computadora: Se necesita una computadora con suficiente memoria y capacidad de procesamiento para ejecutar el IDE, el servidor de la aplicación y la base de datos.

**Estándares y Documentos:**

Documentación de Java: Proporciona información detallada sobre el lenguaje de programación Java y sus bibliotecas estándar.

<https://docs.oracle.com/en/java/>

Documentación de Spring Boot: Proporciona guías y referencias para el uso de Spring Boot.

<https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>

Documentación de Maven: Proporciona información sobre cómo utilizar Maven para la gestión de proyectos.

<https://maven.apache.org/guides/index.html>

El resto de información se buscara en YouTube ya que es la forma de ver como se implementa toda esta documentación en casos reales.

**Acceso a Servicios o Instalaciones:**

No se requiere acceso a servicios o instalaciones específicas de la Universidad u otras instituciones para este proyecto. Sin embargo, se necesitará acceso a Internet para descargar dependencias y para el control de versiones con Git y GitHub.

## 6 Bibliografía

Para el desarrollo de este proyecto, se han consultado diversas fuentes de información, tanto libros como documentación en línea y blogs. Diez obras que se han utilizado como referencia:

Walls, C. (2016). Spring Boot in Action. Manning Publications Co.

Long, J., & White, K. (2017). Cloud Native Java: Designing Resilient Systems with Spring Boot, Spring Cloud, and Cloud Foundry. O'Reilly Media, Inc.

Jovanovic, A. (2018). Spring 5 Design Patterns. Packt Publishing Ltd.

Harrop, R., & Fisher, J. (2005). Pro Spring. Apress.

Kurniawan, B. (2020). Servlet, JSP and Spring MVC: A Tutorial. Brainy Software Inc.

Spring Boot Documentation: <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>

Java Documentation: <https://docs.oracle.com/en/java/>

Maven Documentation: <https://maven.apache.org/guides/index.html>

Baeldung Blog: <https://www.baeldung.com/>

Stack Overflow: <https://stackoverflow.com/>

## 7 Planificación

Para la planificación inicial del proyecto, se propone el siguiente desglose de tareas y su correspondiente duración. Este es un punto de partida y puede requerir ajustes a medida que avanza el proyecto.

Configuración del entorno de desarrollo (1 semana): Incluye la instalación y configuración de Java, Spring Boot, Maven, IntelliJ IDEA y MySQL. También incluye la configuración de Git y GitHub para el control de versiones y la colaboración.

Diseño de la aplicación (1 semana): Incluye la definición de las entidades, servicios, controladores y repositorios que se utilizarán en la aplicación.

Implementación del controlador y los servicios (2 semanas): Incluye la codificación de los controladores y servicios, así como la implementación de las operaciones CRUD.

Implementación del repositorio y la base de datos (2 semanas): Incluye la configuración de la base de datos MySQL y la implementación del repositorio.

Integración y configuración de las dependencias (1 semana): Incluye la integración de las dependencias necesarias para el proyecto y su configuración.

Pruebas (2 semanas): Incluye la escritura y ejecución de pruebas para asegurar que la aplicación funciona como se espera.

Depuración y resolución de problemas (1 semana): Incluye la identificación y resolución de cualquier problema o error que surja durante el desarrollo.

Documentación (1 semana): Incluye la escritura de la documentación del código y del uso de la aplicación.

Revisión final y entrega (1 semana): Incluye la revisión final del código y la documentación, así como la preparación para la entrega del proyecto.