

# Escuela Técnica Superior de Ingeniería Informática

Diseño y Pruebas II



Report on prior knowledge of WIS architecture

C1.03.06 - Entregable 1

**Campos Mejías, Gonzalo (GM)**, goncammej@alum.us.es

**Gallardo Pelayo, Alejandro (AG)**, alegalpel@alum.us.es

**López-Benjumea Novella, Alberto Miguel (ALB)**, a.lopezbenjumea@gmail.com

**Medina Durán, Alejandro (AM)**, alepridur@alum.us.es

**Vargas Castro, Francisco José (FJV)**, fravarcas1@alum.us.es

**Repositorio:** <https://github.com/alejandromd/acme-L3-D01>

17 FEBRERO - 2023

# Historial de Revisiones

Versión	Fecha	Autor(es)	Descripción
1.1	13.02.2023	GM	Creación del documento
1.2	14.02.2023	GM	Versión incial completa de acuerdo a los requisitos
2.0	15.02.2023	GM	Revisión y formato formal del report
3.0	16.02.2023	GM	Versión final del documento

## **Resumen ejecutivo**

El presente informe tiene como finalidad exponer el conocimiento previo que el grupo posee sobre la arquitectura WIS (Web Information System). Dicho informe se elabora con el propósito de documentar y rememorar los conceptos adquiridos con anterioridad a esta asignatura, los cuales pueden resultar de utilidad tanto en un contexto laboral como en el marco del proyecto actual.

# Tabla de Contenidos

I.	Introducción . . . . .	1
II.	Conocimiento previo sobre la arquitectura WIS . . . . .	2
III.	Conclusión . . . . .	4
IV.	Referencias . . . . .	5

# **I. Introducción**

Este informe tiene como objetivo presentar una síntesis del conocimiento que hemos adquirido acerca de la arquitectura WIS. Se abordará tanto el conocimiento teórico como la aplicación práctica que hemos hecho de la misma mediante el uso de diferentes herramientas.

En términos generales, la arquitectura WIS se define como un modelo de diseño que permite la creación de sistemas de información web para el desarrollo de diversas aplicaciones en línea. La arquitectura se divide en varias capas con funciones específicas, desde la capa de presentación hasta la capa de acceso a datos.

La experiencia adquirida nos ha permitido comprender los elementos fundamentales y su aplicación en un contexto real, lo que nos proporciona una sólida base de conocimientos para abordar el proyecto y realizar un trabajo de alta calidad en el futuro.

## II. Conocimiento previo sobre la arquitectura WIS

Nuestra experiencia en el desarrollo de sistemas de información web se ha desarrollado mayormente durante la asignatura DP1, en la que hemos adquirido conocimientos tanto teóricos como prácticos. A pesar de haber trabajado con estos sistemas en ocasiones anteriores, nuestra dedicación se ha centrado en la elicitación de requisitos o en la tecnología utilizada en cada caso, sin profundizar en su arquitectura.

En particular, hemos trabajado con Spring Boot, un framework diseñado para acelerar el proceso de desarrollo, implementando el patrón de diseño Modelo-Vista-Controlador (MVC) y basándose en la utilización de microservicios, los cuales se centran en funcionalidades específicas.

Con el fin de comprender a fondo la arquitectura de Spring Boot y su funcionamiento, hemos estudiado en profundidad las diferentes capas que la componen:

- La Capa de Presentación se encarga de recopilar la información proporcionada por el usuario y enviarla al servidor, procesarla, generar la presentación y mostrarla al usuario.
- La Capa de Servicios, por su parte, recibe la entrada de datos de la Capa de Presentación, interactúa con la Capa de Acceso de Datos para realizar las operaciones requeridas y envía los resultados procesados de vuelta a la Capa de Presentación.
- Finalmente, la Capa de Acceso de Datos se encarga de almacenar, recuperar y mantener la integridad de los datos.

Es importante destacar que para crear la Capa de Acceso de Datos solo se necesita una clase repositorio que implemente las operaciones de CRUD (Create, Read, Update, Delete).

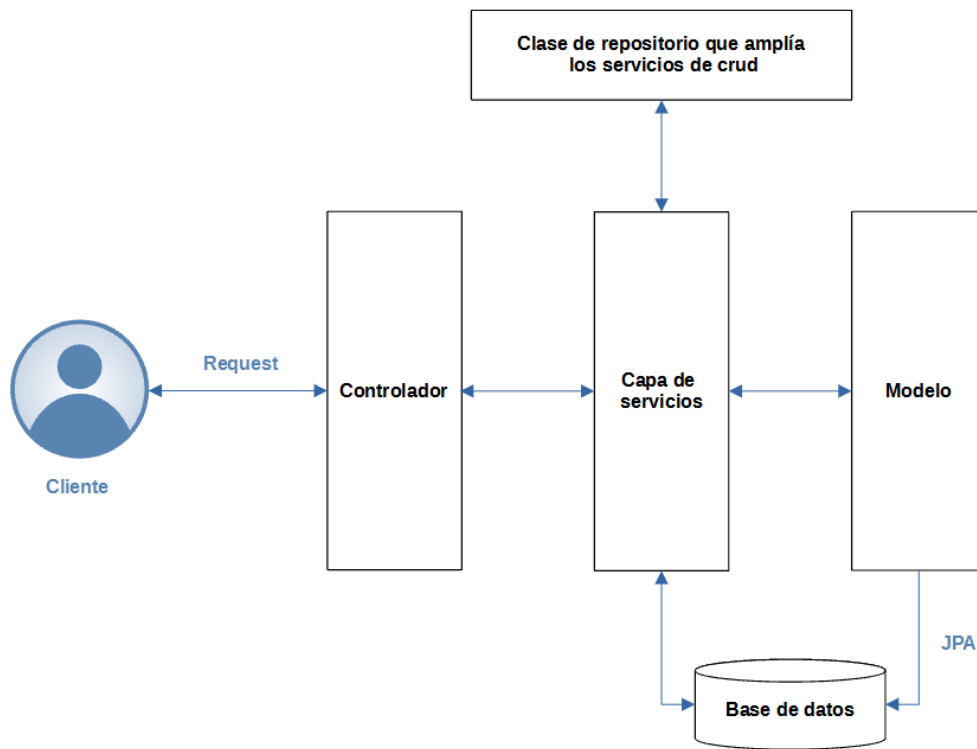


Figura 1: Flujo de operación Spring Boot

Cuando el cliente realiza una petición al servidor web, esta es dirigida al controlador mapeado con la ruta correspondiente, que se comunica con la Capa de Servicios en caso de ser necesario. La lógica de negocio se realiza en la Capa de Servicios, que trabaja con los datos de la base de datos que se mapea mediante JPA con las entidades. Finalmente, una página JSP se envía como respuesta al cliente si no ha ocurrido ningún error.

Es fundamental mencionar que JPA es una colección de clases y métodos que actúa como un puente para preservar los datos entre objetos Java (entidades) y bases de datos relacionales, mientras que JSP es una colección de herramientas que permite la creación de páginas web dinámicas para interactuar con el usuario y modificar los contenidos que se presentan.

### III. Conclusión

En conclusión, nuestra experiencia en el desarrollo de sistemas de información web se ha fortalecido gracias a la asignatura DP1, en la cual hemos adquirido conocimientos teóricos y prácticos sobre el funcionamiento de Spring Boot, un framework que implementa el patrón de diseño MVC y se basa en la utilización de microservicios. Hemos profundizado en la arquitectura de este framework, estudiando detalladamente cada una de las capas que lo componen: la Capa de Presentación, la Capa de Servicios y la Capa de Acceso de Datos. Asimismo, hemos aprendido que JPA es una herramienta que permite preservar los datos entre objetos Java y bases de datos relacionales, mientras que JSP es una herramienta que permite la creación de páginas web dinámicas. Todo esto nos permite estar mejor preparados para enfrentar futuros retos en el desarrollo de sistemas de información web.



## IV. Referencias

Intentionally blank