Universidad de Sevilla  
Escuela Técnica Superior de Ingeniería Informática

**WIS Report**



Grado en Ingeniería Informática – Ingeniería del Software  
Diseño y Pruebas 2.

Curso 2024 – 2025

|  |  |  |
| --- | --- | --- |
| **Grupo de prácticas: C1.004** | | |
| **Autores por orden alfabético** | **Rol** | **Descripción del rol** |
| Cortés Carrasco, Claudio - 77929527F | Developer | Persona encargada de desarrollar el código. |
| Delgado Pallares, David - 29519510E | Tester | Persona encargada de realizar pruebas sobre el código. |
| Fernández Román, Santiago- 30276668C | Project Manager | Persona encargada de tomar decisiones de diseño y vigilar el correcto desarrollo |
| García León, Guillermo -  17486052A | Developer | Persona encargada de desarrollar el código. |
| Bernardos Ruiz, Alberto-  78137900T | Developer | Persona encargada de desarrollar el código. |

**Control de Versiones**

|  |  |  |
| --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** |
| 19/02/2025 | v1.0.0 | Desarrollo de la primera versión. |
|  |  |  |
|  |  |  |
|  |  |  |

**Índice de contenido**

[1. Resumen ejecutivo](#_wkdgh14d2jlq) 4

[2. Introducción](#_14h6bglc6s60) 5

[3. Contenidos](#_icebub40rzkm) 6

[2.1. Backend - Spring Boot](#_t8rupbemurf2) 6

[2.2. Frontend - React](#_gvfxpquuumal) 6

[4. Conclusiones](#_471n1tejp1g1) 7

[5. Bibliografía](#_syrhz6ozjbez) 8

## Resumen ejecutivo

Este documento presenta un resumen de la tecnología Web Information Systems (WIS), abarcando los conceptos clave aprendidos hasta el momento.

Asimismo, este documento ha sido elaborado para garantizar el cumplimiento del Requisito 21 del proyecto, asegurando que la información presentada esté alineada con los criterios establecidos. Se han estructurado los contenidos de manera clara y concisa, proporcionando una visión general efectiva de la tecnología WIS y su aplicación en el desarrollo de software.

## Introducción

El presente informe tiene como objetivo recopilar y estructurar los conocimientos adquiridos sobre Web Information Systems (WIS) en el marco de la asignatura Diseño y Pruebas 1 (DP1). A lo largo del curso, hemos trabajado en la implementación de un sistema basado en la arquitectura Spring Boot + React, explorando cómo estas tecnologías se combinan para el desarrollo de aplicaciones web modernas.

En este documento, se detallarán los conceptos clave abordados, desde la estructura del backend con Spring Boot, incluyendo controladores, servicios y persistencia de datos, hasta la construcción del frontend con React, destacando la gestión del estado, los componentes y la comunicación con el backend a través de APIs REST.

Este reporte servirá como una síntesis del aprendizaje adquirido y como referencia para el desarrollo futuro de aplicaciones basadas en WIS.

## Contenidos

La arquitectura de un Web Information System (WIS) basado en Spring Boot y React sigue un modelo de aplicación full-stack, en el cual el frontend y el backend están desacoplados pero se comunican mediante servicios REST.

#### 2.1. Backend - Spring Boot

Spring Boot es un framework de Java que facilita la creación de aplicaciones web robustas y escalables. Su estructura modular permite dividir el backend en varias capas:

* Controladores: Manejan las solicitudes HTTP y exponen los endpoints REST.
* Servicios: Contienen la lógica de negocio y coordinan la operación entre el controlador y el repositorio.
* Repositorios: Se encargan de la persistencia de datos, generalmente utilizando Spring Data JPA y bases de datos relacionales como MySQL o PostgreSQL.

Spring Boot también incluye herramientas para seguridad (Spring Security), gestión de dependencias (Spring Boot Starter) y documentación de APIs (Swagger).

#### 2.2. Frontend - React

React es una librería de JavaScript enfocada en la creación de interfaces de usuario dinámicas y eficientes. Los principales elementos en el desarrollo de un frontend con React incluyen:

* Componentes: Unidades reutilizables que representan partes de la interfaz de usuario.
* Estado y Propiedades (State & Props): Mecanismos para gestionar la información dentro de los componentes.
* React Router: Para la navegación entre distintas vistas de la aplicación.
* Consumo de APIs REST: Mediante fetch o Axios, React se comunica con el backend para obtener y enviar datos.

## Conclusiones

Nuestro conocimiento en la arquitectura es limitado y las tecnologías que usamos son nuevas para todos nosotros (Acme Framework). Por ello creemos que será necesaria formación en el futuro, así como apoyo entre compañeros de grupo para la resolución de dudas o errores.

## Bibliografía

Apuntes de la asignatura Diseño y Pruebas 1 (DP1) contenidos en la plataforma Enseñanza Virtual.