



INFORME DE ARQUITECTURA WIS

Grupo 1-C1.020 | <https://github.com/PDJ6975/Acme-ANS>



Nombre	Correo Corporativo
Antonio Rodríguez Calderón	antrodcal@alum.us.es
Adrián Ramírez Gil	adrramgil@alum.us.es
Jianwu Hu	jiahu3@alum.us.es
Pablo Castrillón Mora	pabcasmor1@alum.us.es
Pablo Olivencia Moreno	pabolimor@alum.us.es

17 DE FEBRERO DE 2025
DISEÑO Y PRUEBAS II

Tabla de Contenido

1. Resumen Ejecutivo	2
2. Tabla de Revisión	2
3. Introducción	2
4. Conocimiento previo sobre la arquitectura de un WIS	3
4.1 Definición de un Web Information System (WIS)	3
4.2 Componentes principales de un WIS	3
4.2.1 Interfaz de usuario (Frontend)	3
4.2.2 Servidor web y lógica de negocio (Backend)	4
4.2.3 Base de datos.....	4
4.2.4 Comunicación entre componentes	4
4.3 Tipos de arquitecturas de un WIS.....	5
4.3.1 Arquitectura Cliente-Servidor.....	5
4.3.2 Arquitectura Monolítica	5
4.3.3 Arquitectura basada en microservicios.....	5
5. Conclusión	6
6. Bibliografía.....	6

1. Resumen Ejecutivo

Este informe documenta el conocimiento previo del equipo sobre la arquitectura de un **Web Information System (WIS)** antes de cursar esta asignatura.

Antes de cursar la asignatura, entendemos que un WIS gestiona información en línea mediante una arquitectura cliente-servidor. Conocemos tecnologías como **HTML, CSS, JavaScript, SQL...**, así como el uso de **REST APIs**. Sin embargo, desconocemos aspectos avanzados como **escalabilidad, balanceo de carga y seguridad**.

El informe incluye nuestra perspectiva inicial sobre WIS y ejemplos de sistemas conocidos.

2. Tabla de Revisión

Versión	Fecha	Descripción de los cambios
1.0	17/02/2025	Creación inicial del documento

3. Introducción

Nuestro conocimiento sobre la arquitectura de un **Web Information System (WIS)** se basa en lo aprendido en materias previas. Entendemos que un WIS es un sistema diseñado para gestionar y presentar información a través de internet, compuesto por **una base de datos, un servidor web y una interfaz de usuario**. Sin embargo, nuestra comprensión está limitada a modelos básicos de funcionamiento sin un conocimiento profundo sobre temas avanzados como **escalabilidad, seguridad, balanceo de carga y optimización del rendimiento**.

A lo largo de la carrera, hemos trabajado con tecnologías relacionadas con el desarrollo web, incluyendo **HTML, CSS y JavaScript** para la interfaz, **SQL para la gestión de bases de datos**, y lenguajes **Java y JavaScript en el backend**. También estamos familiarizados con el modelo **cliente-servidor**, el uso de **REST APIs** y la

conexión entre el frontend y el backend. No obstante, aspectos más avanzados no forman parte de nuestro conocimiento práctico o teórico.

Este documento nos permite reflexionar sobre nuestra base de conocimientos antes de la asignatura y sirve como punto de partida para comprender la evolución de nuestra comprensión sobre los Web Information Systems.

4. Conocimiento previo sobre la arquitectura de un WIS

4.1 Definición de un Web Information System (WIS)

Antes de esta asignatura, entendemos que un **Web Information System (WIS)** es un sistema diseñado para almacenar, procesar y presentar información a través de internet. Su propósito principal es proporcionar acceso estructurado a la información mediante una interfaz web, permitiendo a los usuarios interactuar con datos en tiempo real.

Desde nuestra perspectiva, un WIS es una evolución de los sistemas de gestión de información tradicionales, integrando tecnologías web para facilitar el acceso desde cualquier dispositivo con conexión a internet.

4.2 Componentes principales de un WIS

Sabemos que un WIS se compone de varios elementos fundamentales que permiten su correcto funcionamiento:

4.2.1 Interfaz de usuario (Frontend)

El frontend es la parte visible del sistema con la que interactúan los usuarios. Se encarga de mostrar la información de manera accesible y funcional. Estamos familiarizados con tecnologías como:

- **HTML, CSS y JavaScript**, para la estructura, el diseño y la interacción dinámica.
- **Frameworks como Bootstrap**, que facilitan el diseño responsivo.
- **Librerías como React o Vue.js**, que permiten una mejor gestión de interfaces dinámicas.

4.2.2 Servidor web y lógica de negocio (Backend)

El backend se encarga de procesar las solicitudes del usuario y gestionar la lógica de negocio. Antes del curso, entendemos que este componente incluye:

- **Lenguajes como PHP, Python y JavaScript (Node.js)** para el procesamiento de datos.
- **Frameworks como Express.js o Django**, que facilitan la construcción de APIs y la gestión del backend.
- **Conexión con bases de datos**, permitiendo almacenar y recuperar información.

4.2.3 Base de datos

Sabemos que un WIS requiere una base de datos para almacenar y gestionar la información. Conocemos:

- **Bases de datos relacionales (SQL)** como MySQL y PostgreSQL.
- **Bases de datos NoSQL**, como MongoDB, utilizadas para datos estructurados más flexibles.

4.2.4 Comunicación entre componentes

Antes de este curso, entendemos que los sistemas WIS utilizan distintos métodos de comunicación:

- **Peticiones HTTP (GET, POST, PUT, DELETE)** para la comunicación entre frontend y backend.
- **REST APIs**, que estructuran las interacciones entre servicios.
- **WebSockets**, utilizados para actualizaciones en tiempo real.

4.3 Tipos de arquitecturas de un WIS

Sabemos que existen distintas formas de estructurar un WIS dependiendo de sus necesidades y escalabilidad.

4.3.1 Arquitectura Cliente-Servidor

El modelo cliente-servidor es el más común y el que mejor comprendemos. Funciona mediante la interacción entre un **cliente (navegador web)** que envía solicitudes y un **servidor** que procesa y responde con la información solicitada.

4.3.2 Arquitectura Monolítica

Entendemos que en una arquitectura monolítica, toda la lógica del sistema está en una única aplicación. Este modelo es más sencillo de desarrollar, pero puede volverse difícil de escalar con el tiempo.

4.3.3 Arquitectura basada en microservicios

Sabemos que los microservicios permiten dividir una aplicación en múltiples servicios pequeños e independientes. Aunque no tenemos experiencia práctica con esta arquitectura, conocemos sus ventajas en términos de escalabilidad y mantenimiento.

5. Conclusión

Antes de cursar esta asignatura, nuestro conocimiento sobre la arquitectura de un **Web Information System (WIS)** se limita a los conceptos fundamentales aprendidos en asignaturas previas. Entendemos la importancia de los componentes básicos, como la base de datos, el servidor web y la interfaz de usuario, y estamos familiarizados con tecnologías y lenguajes utilizados en el desarrollo web.

Sin embargo, nuestro conocimiento es limitado en aspectos más avanzados como **escalabilidad, balanceo de carga, seguridad y arquitecturas distribuidas**. Este informe nos ayuda a reflexionar sobre nuestra base de conocimientos y servirá como referencia para evaluar nuestro progreso a lo largo del curso.

6. Bibliografía

Intencionalmente en blanco.