Grupo: C1.064

19/02/2025

Repositorio: <https://github.com/paugonpae67/Acme-ANS-D01>

Integrantes: Paula Rosa González Páez ([paugonpae@alum.us.es](mailto:paugonpae@alum.us.es)), Lidia Ning Fernández Casillas ([lidfercas@alum.us.es](mailto:lidfercas@alum.us.es)), María Auxiliadora Quintana Fernández ([marquifer@alum.us.es](mailto:marquifer@alum.us.es)), Fernando Partal García ([ferpargar@alum.us.es](mailto:ferpargar@alum.us.es)), Iratxe Parra Moreno ([iraparmor@alum.us.es](mailto:iraparmor@alum.us.es)).

Architecture of a wis report

ACME ANS D01

### Tabla de contenidos

Contenido

[0](#_Toc190959369)

[Tabla de contenidos 1](#_Toc190959370)

[Resumen Ejecutivo 2](#_Toc190959371)

[Tabla de Revisiones 3](#_Toc190959372)

[Introducción 4](#_Toc190959373)

[Contenidos 5](#_Toc190959374)

[Definición de WIS 5](#_Toc190959375)

[Componentes y Arquitectura 5](#_Toc190959376)

[Tecnologías y Herramientas Relacionadas 7](#_Toc190959377)

[Conclusión 8](#_Toc190959386)

[Bibliografía 9](#_Toc190959387)

### resumen ejecutivo

En este documento se explican los conocimientos previos a cursar la asignatura Diseño y Pruebas 2 sobre los Sistemas de Información Web (WIS). En dicho documento se presentan qué se entiende hasta el momento sobre los fundamentos de los WIS, definiendo su finalidad como herramienta de gestionar, procesar y distribuir información a través de Internet.

Se expone el entendimiento sobre la estructura en capas de un WIS, que incluye el cliente (front-end) y el servidor (back-end) así como la finalidad de la Base de Datos. Describiendo cómo los usuarios interactúan con la aplicación usando diversas tecnologías mediante la creación de interfaces dinámicas.

Por otro lado, también se abordan los componentes del back-end, dentro del cual tenemos más práctica aplicando la arquitectura Modelo-Vista-Controlador (MVC). Explicando cómo es la relación entre los distintos componentes y el proceso que se sigue desde que se recibe la petición de obtención de información hasta que se resuelve.

También cubre las tecnologías relevantes para el desarrollo de WIS usadas. En conjunto, se busca ofrecer una visión sobre nuestros conocimientos de los WIS, su arquitectura y componentes.

### tabla de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| **VERSIÓN** | **FECHA** | **MIEMBRO DEL EQUIPO** | **DESCRIPCIÓN** |
| 1.0 | 19/02/2025 | Iratxe Parra Moreno | Redacción del documento. |

### introducción

En el presente documento se aborda el conocimiento previo que se tiene sobre los Sistemas de Información Web (WIS) antes de cursar la asignatura de Diseño y Pruebas 2. A lo largo del desarrollo de este informe, se exploran los fundamentos de los WIS haciendo énfasis en su estructura que incluye componentes clave como el cliente y el servidor.

El documento inicia con una definición clara de lo que se entiende por un WIS, destacando como sus capas interaccionan para ofrecer el contenido dinámico a los usuarios. A continuación, se explica que se conoce del front-end para posteriormente explicar el back-end desglosando sus componentes, donde se destaca la experiencia que se tiene en el uso de la arquitectura Modelo-Vista-Controlador (MVC), mostrando el flujo de información desde que se recibe una solicitud hasta que se responde adecuadamente.

Posteriormente, se detallan los diversos modelos de arquitectura de los cueles se tienen entendimiento proporcionando una breve definición de estos. Seguidamente se especifican las tecnologías y herramientas que han ido estudiando y practicando, las cuales comprendeos fundamentales para el desarrollo de un WIS. Esto incluye, entre otras, el uso de tecnologías front-end como React.js, junto con el back-end utilizando Node.js, y las bases de datos MySQL y MariaDB.

Finalmente, la estructura del documento se organiza en varias secciones que incluyen la definición de WIS, los componentes y arquitectura, y las tecnologías y herramientas relacionadas. Esta organización permitirá ofrecer una visión clara de los conocimientos previos adquiridos sobre los Sistemas de Información Web.

### contenidos

### DEFINICIÓN DE WIS

Antes de comenzar esta asignatura, se tenía una noción general sobre los Sistemas de Información Web (WIS) como sistemas que permiten gestionar, procesar y distribuir información a través de internet. Dichos sistemas están relacionados con aplicaciones web que, a su vez, se relacionan con bases de datos con el objetivo de ofrecer un contenido dinámico a los usuarios que usen los WIS.

Partimos de la base de que un WIS es entendido como un sistema compuesto por múltiples capas las cuales tienen la finalidad de facilitar la interacción entre usuarios y la información que estos requieren.

### Componentes y arquitectura

Desde una perspectiva previa, se tenía un conocimiento general de los principales componentes de un WIS gracias a los estudiado en otras asignaturas con anterioridad. Dichos componentes son:

* **Cliente (Front-end):** Se sabe que el usuario interactúa con la aplicación a través de un navegador web, el cual interpreta código HTML, CSS y JavaScript. En particular, se tenía experiencia con React.js para la creación de interfaces. Se sabía que el front-end obtenía datos desde el back-end mediante llamadas a APIs REST utilizando fetch, enviando peticiones HTTP (GET, POST, PUT, DELETE). Además, se tiene conocimiento de los eventos en JavaScript, manejo de formularios y validación de datos antes de enviarlos al servidor sabiendo gestionar y desarrollar todo esto.
* **Servidor Web (Back-end):** Se sabe que el back-end es el encargado de recibir las peticiones del cliente, procesarlas y responder con los datos necesarios. Se comprende la arquitectura Modelo-Vista-Controlador (MVC) además de tener experiencia con Node.js donde:
* Controladores: Se encargan de recibir las peticiones HTTP, validarlas y delegar la lógica de negocio a los servicios correspondientes.
* Servicios: Contienen la lógica de negocio y organizan la comunicación con la base de datos llamando a los repositorios para obtenerla información necesaria.
* Repositorios: Se comunican directamente con la Base de Datos y obtienen la información necesaria.
* Modelos (Entidades): Representaban la estructura de los datos y permitían la interacción con la base de datos.
* Autenticación: Se usa JWT (JSON Web Tokens) con el fin de manejar la autenticación y autorización de usuarios.
* Excepciones: Se conoce el desarrollo de diversos tipos de excepciones personalizadas como las de validación o persistencia.
* **Base de Datos:** Se sabe trabajar con MySQL y MariaDB para el almacenaje de información estructurada. Donde se ha hecho uso de conceptos como las relaciones entre tablas, el uso de índices para optimizar consultas y transacciones para mantener la integridad de los datos en operaciones críticas.

También se tiene práctica en el ámbito del Testing, sabiendo asegurar el correcto funcionamiento tanto de back-end como de front-en mediante el desarrollo de pruebas unitarias y pruebas automatizadas (usando herramientas como Junit).

En lo referente a la arquitectura se conocen los siguientes modelos:

* **Arquitectura Cliente-Servidor:** Es un modelo donde el cliente (como un navegador o una app) solicita servicios y el servidor procesa y responde. Se usa en páginas web, bases de datos y aplicaciones en red.
* **Arquitectura en Tres Capas:** Divide la aplicación en tres capas independientes:

1. . Capa de Presentación (interfaz de usuario).
2. . Capa de Lógica de Negocio (procesamiento de datos).
3. . Capa de Datos (base de datos).

Separa responsabilidades y facilita el mantenimiento.

* **Arquitectura Modelo-Vista-Controlador (MVC):** Organiza el software en tres componentes:

1. . Modelo: Maneja los datos y la lógica del negocio.
2. . Vista: Representa la interfaz de usuario.
3. . Controlador: Gestiona la comunicación entre el modelo y la vista.

Este modelo es del que más conocimiento tenemos y más a fondo hemos estudiado llevándolo a la práctica mediante el uso del frameworks Spring MVC.

* **Arquitectura de Microservicios:** Divide una aplicación en servicios pequeños e independientes, cada uno con una función específica y comunicándose a través de APIs. Facilita el escalado y mantenimiento de grandes sistemas, esto ayuda aun bajo acoplamiento y una alta cohesión.
* **APIs y RESTful Services:**
* Las APIs (Interfaces de Programación de Aplicaciones) permiten la comunicación entre sistemas.
* Los RESTful Services siguen principios REST, usando HTTP para operaciones como GET, POST, PUT y DELETE, siendo muy utilizados en servicios web y aplicaciones móviles.

## **tecnologías y herramientas relacionadas**

### Se ha trabajado y por tanto se tiene conocimiento sobre diversas herramientas utilizadas en el desarrollo de WIS:

### Front-end: React.js, HTML, CSS, JavaScript.

### Back-end: Node.js, Spring Boot, Java.

### Bases de Datos: MySQL, MariaDB.

### Autenticación y Seguridad: JWT

### Herramientas de DevOps: GitHub

### Documentación: Markdown

### Frameworks: Spring MVC.

### conclusión

En conclusión, este documento refleja los conocimientos previos adquiridos antes de cursar la asignatura Diseño y Pruebas 2 en relación con los Sistemas de Información Web (WIS). Se establecen las bases conceptuales sobre su funcionamiento, estructura en capas y su papel en la gestión de información en la web. Asimismo, se detallan los componentes fundamentales, desde el front-end y back-end hasta la base de datos, resaltando la experiencia con diversas tecnologías utilizadas en dicho ámbito.

Este documento proporciona una visión clara de los conocimientos previos en arquitectura y desarrollo de WIS, sirviendo como punto de partida para profundizar en los aspectos avanzados que se abordarán en la asignatura.

### bibliografía

Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Sevilla (LSI) (2025) – *Diseño y Pruebas 1, Ingeniería de Requisitos, Introducción al Software y Sistemas de Información 1 y 2* 8