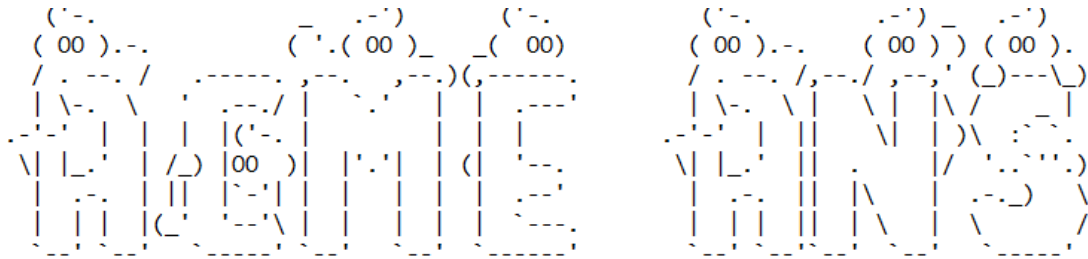


# REPORTE SOBRE WIS

## Acme-ANS-D04



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

Creado por el grupo C1.022, del G1

Participantes	
Nombres	Correos
Varo Vera, Juan	juavarver@alum.us.es

## Índice

Portada.....	1
Índice.....	2
Resumen ejecutivo.....	3
Tabla de revisiones.....	4
Introducción.....	5
Contenidos.....	6
Conclusiones.....	15
Referencias.....	16

## **Resumen ejecutivo**

Este es el proyecto del grupo C1.022 sobre Acme AirNav Solutions, S.A. (Acme ANS, S.A. abreviado), la cual es una compañía ficticia especializada en ayudar a aeropuertos a organizar y coordinar sus operaciones a partir de soluciones desarrolladas en software. La logística de los vuelos (la programación de los vuelos, la organización de reservas y de tripulación, etc.) se gestionan mediante el desarrollo de un WIS.

Con esto, para el correcto funcionamiento de la aplicación, se deberá escribir un reporte de testing donde se describan los valores de cobertura de código junto con los valores correspondientes al rendimiento de la aplicación, ambos previos y posteriores a realizar mutaciones en el código.

## Tabla de revisiones

Número	Fecha	Descripción
1	19/02/2025	Versión inicial del reporte.
2	25/05/2025	Revisión final

## Introducción

Los Sistemas de Información Web (WIS) son fundamentales para la gestión y distribución eficiente de la información a través de internet. Están estructurados en tres capas o niveles principales: presentación, aplicación y base de datos. Cada una de estas capas tiene un papel crítico en el procesamiento y visualización de datos, lo que permite ofrecer una experiencia de usuario fluida y accesible.

En este documento, se analiza la arquitectura de los WIS, incluyendo los componentes que constituyen cada capa y su interacción. El enfoque principal del proyecto es el diseño, implementación y prueba de la aplicación en la capa de aplicación, que se lleva a cabo en el servidor de aplicaciones Tomcat. Además, se detalla cómo la capa de presentación utiliza el navegador Firefox Developer en modo incógnito para realizar pruebas tanto informales como formales, y cómo la capa de base de datos utiliza MariaDB para almacenar y gestionar los datos.

Este reporte está organizado de la siguiente forma:

1. Resumen ejecutivo: Introducción breve sobre el reporte.
2. Tabla de revisiones: Historial de revisiones del documento.
3. Introducción: Contextualización de los WIS y la importancia de su arquitectura.
4. Contenidos: Descripción detallada de los tres niveles y sus componentes.
5. Conclusiones: Resumen de los hallazgos y la importancia de los WIS.
6. Bibliografía: Fuentes consultadas durante la investigación.

## **Contenidos**

### **Capa de presentación:**

Aquí es donde entra en juego el navegador web, que se encarga de mostrar la interfaz al usuario y permitir la interacción con la aplicación. Usamos tecnologías como HTML, CSS y JavaScript para diseñar esa interfaz. En este proyecto, estamos utilizando Firefox Developer en modo incógnito, tanto para pruebas informales como formales, lo que nos ayuda a mantener un entorno limpio y controlado para depurar y evaluar el rendimiento.

### **Capa de aplicación:**

Esta es la parte central del sistema. Utilizamos Tomcat como servidor de aplicaciones, el cual incluye componentes importantes como Coyote, que actúa como servidor HTTP para recibir las solicitudes, Catalina, que es el motor de servlets donde realmente se procesa la lógica del sistema y Jasper, que transforma las respuestas en HTML y otros formatos que entiende el navegador.

En esta capa es donde implementamos la lógica de negocio del proyecto: los servlets que hemos desarrollado definen cómo responde el sistema a las diferentes acciones del usuario. La mayor parte del trabajo se ha centrado aquí: en diseñar, implementar y probar esta lógica.

### **Capa de base de datos:**

Es donde se almacena toda la información que maneja el sistema. Usamos MariaDB como servidor de base de datos, que se encarga de gestionar las consultas, inserciones y actualizaciones de datos. Su papel es esencial para asegurar que la información esté siempre disponible y organizada correctamente.

El flujo de funcionamiento entre estas capas es claro: el usuario hace una solicitud desde el navegador; esta llega al servidor de aplicaciones (Tomcat), que la procesa usando la lógica definida y, si es necesario, accede a la base de datos. Finalmente, se genera una respuesta que vuelve al navegador y se muestra al usuario.

## **Conclusiones**

Los Sistemas de Información Web (WIS) juegan un papel crucial en la gestión moderna de datos, proporcionando una interfaz interactiva y accesible a través de internet. La arquitectura de tres capas, formada por las capas de presentación, aplicación y base de datos permite una gestión eficiente de la información, garantizando el rendimiento y la seguridad del sistema. El proyecto se centra en el desarrollo y prueba de la capa de aplicación, utilizando Tomcat como servidor de aplicaciones y MariaDB como base de datos, mientras que las pruebas se realizan en el navegador Firefox Developer en modo incógnito para asegurar la funcionalidad de la interfaz.

## **Bibliografía**

Intencionalmente en blanco