

# REPORTE

# ARQUITECTURA WIS

**GRUPO C1.02.03**



## **Autores:**

- **Alejandro Campano Galán** – [alecamgal1@alum.us.es](mailto:alecamgal1@alum.us.es)
- **Juan Jesús Campos Garrido** – [juacamgar2@alum.us.es](mailto:juacamgar2@alum.us.es)
- **David Cortabarra Romero** – [davcorrom@alum.us.es](mailto:davcorrom@alum.us.es)
- **Alejandro García Sánchez-Hermosilla** – [alegarsan11@alum.us.es](mailto:alegarsan11@alum.us.es)
- **Pablo Mera Gómez** – [pabmergom@alum.us.es](mailto:pabmergom@alum.us.es)

**Repositorio:** <https://github.com/Alejandrocg024/Acme-L3-D01.git>

**Fecha del documento:** 15/02/2023

**Tabla de contenidos.**

Resumen general..... 3

Tabla de revisión..... 4

Introducción. .... 5

Contenido. .... 6

Conclusiones..... 7

Bibliografía..... 8

## **Resumen general.**

Tras cursar distintas asignaturas sobre el desarrollo de sistemas de información software, hemos aprendido cómo funcionan y a implementarlos. Más concretamente, nos hemos centrado en Sistemas de Información basado en Web (WIS). Estos sistemas son aquellas aplicaciones de software que podemos utilizar accediendo a un servidor web a través de Internet. Algunas de sus ventajas es que son aplicaciones muy prácticas debido al uso del navegador web como cliente, con independencia del sistema operativo, lo que también nos permite la ventaja de poder acceder al sistema desde un navegador en cualquier momento y lugar. Además, son fáciles de usar y ahorran grandes costes.

**Tabla de revisión.**

Fecha	Versión	Descripción
13/02/2023	0.1	Creación de una primera versión
14/02/2023	0.2	Creación de todo el documento y redacción del informe
15/02/2023	1.0	Corrección de errores. Versión final

## **Introducción.**

En este documento, hablaremos sobre los conceptos que hemos ido aprendiendo sobre los sistemas de información basados en Web en las distintas asignaturas que hemos ido cursando. A lo largo de estos años, hemos diseñado distintos sistemas en varias asignaturas, pero con el proyecto que más hemos aprendido, porque más partes de este hemos implementado, ha sido el sistema realizado para la asignatura *Diseño y Pruebas I* en el que hemos tenido que realizar la funcionalidad de un juego de mesa con distintos módulos. Gracias a este proyecto, hemos adquirido la mayor parte de nuestro conocimiento y experiencia sobre estos sistemas WIS.

Entonces, hablaremos de como se componen estos sistemas, destacado como se estructuran gracias al patrón de diseño MVC y al diseño arquitectónico en capas que detallaremos y, finalmente, destacaremos varios patrones de diseño que hemos encontrado interesantes y útiles para diseñar un WIS.

## Contenido.

Basándonos en los conocimientos aprendidos en las asignaturas anteriores, diremos que sabemos las bases de una arquitectura WIS. En primer lugar, hablaremos sobre como se componen y, sabemos que se basa en una arquitectura en capas que consta en dividir la aplicación en la capa de presentación, de aplicación y de persistencia.

Cabe destacar también, que nuestro sistema hace uso del patrón de diseño MVC (Modelo-Vista-Controlador) con el que estructuramos nuestra aplicación en tres componentes distintos como son el modelo que es la representación de la información con todos los datos y la lógica necesaria para trabajar con este modelo. La vista que es la representación del modelo de forma visual para que el usuario pueda trabajar con esta mediante una interfaz de usuario y, finalmente, el controlador, que responde a los eventos en la interfaz de usuario e invoca cambios en el modelo y, probablemente, en la vista.

También es importante entender algunos patrones de diseño que se usan como son los repositorios que proporcionan métodos para recuperar y guardar objetos de dominio en la base de datos y que solemos tener uno por entidad. Otro elemento importante que usamos es la capa de Servicio con la que exponemos toda la funcionalidad del dominio como API y que solemos organizarlas por entidades

Pasando a detallar las capas, en primer lugar, hablaremos de la capa de presentación en la que tendremos un solo componente que es el navegador con el que renderizaremos la interfaz de usuario. En esta capa procesaremos renderizaremos las Vistas, que nosotros tratábamos como JSP, y trataremos las interacciones del mismo mediante el controlador con HTTP request que las trataremos con nuestro framework (Spring) al que le daremos total control de la aplicación con la inversión de control.

En segundo lugar, hablamos de la capa de aplicación. En esta tendremos toda la lógica de negocio de nuestra aplicación, básicamente toda la aplicación. Esta aplicación implementa todas las características de nuestro proyecto y hace uso del Modelo que contiene el Controlador, Servicio y Repositorio para realizar todas las operaciones implementadas.

En tercer lugar, hablamos de la capa de la base de datos, a la que accedemos mediante los repositorios y que tiene un solo componente que es nuestra base de datos con la que realizaremos las distintas llamadas para trabajar con los datos.

Finalmente, destacaremos algunos patrones de diseño más que hemos usado anteriormente para desarrollar sistemas WIS y, algunos de estos pueden ser el patrón “Proxy” con el creamos sustitutos para cualquier objeto que se solicite, a partir de un servicio y el controlador puede acceder a éste, creando una simple instancia de la clase. También, hemos usado el patron de diseño de “pagination” con el que organizamos la información de forma muy clara cuando cargamos muchos datos y los mostramos paginados. Finalmente, destacar el patrón de diseño “Template View”, que nos hemos referido anteriormente, que nos sirve para realizar todos los JSP que usamos para la vista de la aplicación y, posteriormente, convertiremos a HTML.

## **Conclusiones.**

En conclusión, diremos que esta arquitectura de sistema de información es muy usada hoy en día por la gran cantidad de ventajas que tienen sobre otro tipo de sistemas. También, podemos afirmar que, poco a poco, comenzamos a entender mejor cómo funcionan y como desarrollarlos correctamente. Además, destacamos que, cuando entiendes bien como funcionan todos sus patrones, es muy intuitivo de programar y se hace muy ameno poder localizar donde se ha producido un error o que tenemos que cambiar para que funcione algo correctamente, debido a la gran separación de responsabilidades de la que disponen estos sistemas.

## **Bibliografía.**

Intencionalmente en blanco