

# Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática



Grado en Ingeniería Informática del Software

**Diseño y Pruebas II**

Curso 2021/2022

## WIS ARCHITECTURE REPORT

**Repositorio:** <https://github.com/Ginpasfer/Acme-Recipes>

| Grupo de Prácticas            | S07  |
|-------------------------------|--|
| Estudiantes                   | Rol  |
| Pastor Fernández, Ginés       | Project Manager<br>Developer<br>Operator<br>Tester |
| Giráldez Álvarez, Pablo       | Developer<br>Analyst<br>Tester                     |
| Rijo Hernández, Badayco       | Developer<br>Tester                                |
| Solís Miranda, Antonio Manuel | Developer<br>Tester                                |
| Paradas Borrego, Álvaro       | Developer<br>Tester                                |

## Índice

|   |   |
|---|---|
| 1. Resumen ejecutivo .....                      | 5 |
| 2. Tabla de revisiones .....                    | 5 |
| 3. Introducción.....                            | 5 |
| 4. Contenido .....                              | 5 |
| 4.1. MVC (Modelo Vista Controlador) .....       | 5 |
| 4.1.1. Modelo .....                             | 5 |
| 4.1.2. Vista .....                              | 5 |
| 4.1.3. Controlador.....                         | 6 |
| 4.2. Estilos arquitectónicos.....               | 6 |
| 4.2.1. Capas.....                               | 6 |
| 4.2.2. Microservicios.....                      | 6 |
| 4.2.3. Aplicación de una sola página (SPA)..... | 6 |
| 5. Conclusión.....                              | 6 |
| 6. Bibliografía .....                           | 6 |

## 1. Resumen ejecutivo

Este documento es el WIS Report. En él se indican los conocimientos acerca de las diferentes arquitecturas de un sistema de información web que se han adquirido a lo largo de la carrera por parte de los miembros del grupo. En este documento se encuentra una serie de puntos donde se detallan los elementos correspondientes a una arquitectura WIS.

## 2. Tabla de revisiones

| Versión | Fecha      | Autor                  | Descripción de cambios   |
|---------|------------|------------------------|--------------------------|
| 1.0     | 22/07/2022 | Badayco Rijo Hernández | - Creación del documento |

## 3. Introducción

En el presente documento se detallan los conocimientos adquiridos a lo largo de la carrera de ingeniería del software por parte de los miembros de grupo. Cada uno de los integrantes del grupo han cursado asignaturas como **Diseño y Pruebas I (DP1)**, **Arquitectura e Integración de Sistemas Software (A/SS)**, **Proceso Software y Gestión (PSG I y II)**, **Introducción a la Ingeniería del Software y los Sistemas de Información (I/SSI I y II)** ..., entre otras, lo que ha permitido obtener unos conocimientos sólidos para trabajar en proyectos como en el actual *Acme Recipes*.

Cabe destacar que en la asignatura de Introducción a la Ingeniería del Software y los Sistemas de Información se ha enseñado todo el sistema que posteriormente aplicaremos en proyecto que se está trabajando actualmente (Navegador, Servidor de aplicación y base de datos).

Gracias a ello el grupo cuenta con la experiencia y el conocimiento sobre la base de datos, así como con el entorno de desarrollo de Eclipse. El grupo también tiene el conocimiento con todo lo relacionado con la propia aplicación en sí, ya que anteriormente los miembros del grupo tuvieron que desarrollar un sistema con controlador, servicio y repositorio en la asignatura de DP1.

El término WIS se corresponde con el término de Sistema de Información Web, el cual está dividido en cuatro componentes básicos:

- Navegador: Se encarga de enviar peticiones HTTP al servidor y en la que se forma la respuesta de este. Para el proyecto se utilizará Firefox.
- Servidor de la aplicación: Su labor es recibir las peticiones del navegador, procesarlas y enviar la respuesta correspondiente.
- Aplicación: Esta posee un controlador, un repositorio y un servicio.
- Servidor de base de datos: Se utilizará el servidor de MariaDB, que se encargará de almacenar los datos correspondientes durante el desarrollo del proyecto.

## 4. Contenido

### 4.1. MVC (Modelo Vista Controlador)

El patrón MVC (*Modelo Vista Controlador*) es un patrón arquitectónico que está estructurado en tres componentes:

#### 4.1.1. Modelo

Este representa la información, incluye los datos y la lógica. Dentro de este podemos destacar:

Entidades: Sus datos se conservan en la base de datos y tienen su propia identidad.

Repositorios: Proporcionan métodos para recuperar y guardar objetos en la base de datos. Suele haber un repositorio por entidad.

Servicios: Exponen la funcionalidad del dominio.

#### 4.1.2. Vista

Es la representación del modelo donde el usuario puede interactuar con ella. Suele realizarse mediante una interfaz de usuario. Se suele utilizar JSP para la interfaz

#### 4.1.3. Controlador

Invoca cambios en el modelo y en la vista y responde a eventos en la interfaz (estas suelen ser solicitudes HTTP). Estos son responsables de la validación de la entrada proporcionada por las vistas, llamar a los servicios para realizar una acción, devolver el nombre de la vista que se debe encargar, etc.

Cabe destacar que en otras asignaturas se ha utilizado este patrón junto con el framework de Spring.

### 4.2. Estilos arquitectónicos

#### 4.2.1. Capas

La funcionalidad del sistema está organizada en capas, cada capa agrupa componentes que ofrecen funcionalidad común y se basan en la funcionalidad que les ofrece la capa de abajo.

Si la interfaz pública que proporciona una capa permanece sin cambios, se puede cambiar una capa por otra equivalente y en el peor de los casos solo afectaría a la capa adyacente. Los componentes están asignados a una sola capa.

Este es uno de los estilos más utilizados para el desarrollo software.

Como cualquier estilo arquitectónico tiene una serie de ventajas e inconvenientes. Algunas de sus ventajas son la alta cohesión y el bajo acoplamiento, se promueven la separación de responsabilidades, etc. Una de las desventajas a destacar es que es difícil proporcionar una separación limpia entre capas.

#### 4.2.2. Microservicios

En este estilo el sistema se desarrolla como un conjunto de pequeños servicios que ejecutan sus propios procesos y se comunican a menudo a través de mensajes HTTP (API RESTful)

Cada servicio se desarrolla con una tecnología adecuada en cada caso. Por lo general hay una interfaz de usuario única.

Permitir abordar los principios claves del diseño y tener interfaces bien definidas son algunas de sus ventajas. Algunos de los inconvenientes son: entorno distribuido, comunicación lenta y riesgo de fallo en la comunicación.

#### 4.2.3. Aplicación de una sola página (SPA)

Una SPA es una aplicación web que solo tiene una página, es decir, una sola página HTML que se carga en el navegador y no se recarga durante su uso. Cuando los usuarios interactúan con la página, la aplicación reescribe la página actual en vez de crear una nueva página del servidor.

Las ventajas de esta son: Tiempo de respuesta rápido y fluido, simple de construir y depurar, etc. Por el contrario, sus desventajas son: Largo tiempo de espera inicial, requiere marcos de JavaScript específicos e implicaciones SEO.

## 5. Conclusión

Para concluir este documento se puede observar que todos los integrantes cuentan con los conocimientos adecuados respecto a la arquitectura WIS, ya que estos han sido adquiridos en algunas de las asignaturas de la carrera de *Ingeniería del Software*, previamente a la asignatura de **Diseño y Pruebas II**. Estos conocimientos se pondrán en práctica en el proyecto a desarrollar denominado *Acme-Recipes*.

## 6. Bibliografía

Intencionalmente en blanco.