

Deliverable-05

WIS Architecture lessons learned Report Workgroup: E7.04 Date: 02/06/2022

Miembros:

Baños González, Alejandro (alebangon@alum.us.es)

Flores Rodríguez, Román (romflorod@alum.us.es)

Grosso Gómez de Terreros, Javier (javgrogom@alum.us.es)

Gutiérrez Ceballos, Pablo (pabgutceb@alum.us.es)

Ibáñez Montero, Julia (julibamon@alum.us.es)

Roldán Cadena, Jesús (jesroldcad@alum.us.es)

Repositorio de Github: <https://github.com/jesroldcad/Acme-Toolkits>

Índice

1. Resumen ejecutivo	1
2. Tabla de control de cambios	2
3. Introducción	2
4. Arquitecturas de Sistemas de Información Web	4
5. Arquitectura aprendida en Diseño y Pruebas II	4
6. Conclusiones	5
7. Referencias bibliográficas	6

1. Resumen ejecutivo

Este documento detalla los conocimientos aprendidos en la asignatura Diseño y Pruebas II en relación con la arquitectura y los patrones de diseño software, así como una breve descripción de los mismos.

Aunque estamos más familiarizados con el patrón Modelo Vista Controlador, consideramos que la nueva arquitectura estudiada nos ha permitido profundizar en nuestros conocimientos sobre las arquitecturas de software y, sobre todo, sobre la estructura de software en capas. Hemos aprendido la utilidad y la importancia de un software desacoplado y con un buen reparto de tareas entre los distintos componentes.

En la arquitectura estudiada y utilizada para el desarrollo del proyecto, hemos utilizado un sistema parecido al patrón MVC, aunque con notables diferencias, ya que cuenta con entidades, repositorios, servicios, controladores y vistas.

En conclusión, creemos que hemos sido capaces de adaptarnos a la nueva arquitectura sin mucha dificultad y enriqueciendo nuestros conocimientos acerca del modelo por capas en el desarrollo de un software aprendiendo a hacer uso de repositorios y servicios (más allá de los conocimientos que nos ha podido proporcionar el uso del framework y lo impartido en la asignatura).

2. Tabla de control de cambios

Número de revisión	Descripción	Fecha
1	Creación del documento acorde a la plantilla. Completados todos los apartados del documento.	31/05/2022
2	Revisión general del documento. Se han realizado pequeñas correcciones de forma y contenido.	01/06/2022

3. Introducción

En este documento se va a reflejar lo aprendido en la asignatura en relación con las arquitecturas de Sistemas de Información Web, así como una breve descripción de las mismas. Para ello, hemos organizado este documento en distintas secciones descritas a continuación.

Sección 4. Arquitecturas de Sistemas de Información Web. Antes de hablar de la arquitectura aprendida en la asignatura, se hará una breve descripción de qué son estas arquitecturas software, así como qué es un patrón arquitectónico y qué ventajas aporta.

Sección 5. Arquitectura aprendida en Diseño y Pruebas II. En esta sección se hará una breve descripción de lo aprendido en la asignatura respecto de las Arquitecturas de Sistemas de Información Web.

Sección 6. Conclusiones. Se reflejarán las conclusiones que hemos compartido todo el equipo acerca de los conocimientos aprendidos tras la asignatura Diseño y Pruebas II.

Sección 7. Referencias bibliográficas. En esta sección reflejaremos las fuentes que hemos utilizado para recopilar alguna información necesaria para el correcto desarrollo de este informe.

Finalmente, el documento posee la siguiente estructura: portada, índice, resumen ejecutivo, tabla de control de cambios, introducción, contenidos (divididos en las secciones *Arquitecturas de Sistemas de Información Web* y *Arquitectura aprendida en Diseño y Pruebas II*) y conclusiones.

4. Arquitecturas de Sistemas de Información Web

Una arquitectura de software, también denominada arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan un marco definido y claro para interactuar con el código fuente del software. Define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos.

Los patrones arquitectónicos, también denominados patrones de arquitectura o arquetipos, ofrecen soluciones a problemas de arquitectura de software en ingeniería de software. Dan una descripción de los elementos y el tipo de relación que tienen junto con un conjunto de restricciones sobre cómo pueden ser usados.

5. Arquitectura aprendida en Diseño y Pruebas II

En la asignatura Diseño y Pruebas II nos hemos familiarizado, a parte de con el framework estudiado, con un nuevo modelo de arquitectura de software que se diferencia en algunos aspectos con los estudiados anteriormente.

Para el desarrollo y testing de la aplicación, hemos hecho uso de Tomcat como servidor y Chrome como navegador (Mozilla Firefox para el formal testing). Gracias al servidor se realiza el procesamiento de las peticiones del navegador, y se envían las respuestas en el formato válido mediante la comunicación con los servlets. Desde el navegador se han enviado peticiones en formato HTTP y renderizado las respuestas en formato HTML, CSS y JS.

Para la base de datos hemos utilizado MariaDB, un sistema de gestión de bases de datos derivado de MySQL.

Para la estructura interna de la aplicación, hemos utilizado una arquitectura parecida al patrón MVC, aunque con notables diferencias. Esta arquitectura por capas es más desacoplada que el MVC ya que implica mayor separación de las partes que componen el sistema software, contando con entidades, repositorios, servicios, controladores y vistas. Para implementar y hacer uso de esta arquitectura se ha hecho uso de JPA, Java y JSP.

En esta arquitectura, la entidad es la encargada de construir los objetos, sus atributos y las relaciones con otras entidades. El repositorio, cuya función es obtener los datos de la base de datos para que sean procesados, separa la lógica que recupera los datos y los asigna a un modelo de entidad de la lógica de negocios que actúa sobre el modelo. El servicio es el encargado de procesar los datos provenientes de los repositorios, centrándose en esta capa la lógica de negocio de

la aplicación. El controlador es la capa intermedia entre los servicios y las vistas, facilitando el procesamiento y traducción de información entre ellos.

El controlador actúa entre los servicios y las vistas, ofreciendo procesamiento y traducción de información entre ellos. Finalmente, la vista es la interfaz de usuario que compone la información que se envía al cliente y los mecanismos de interacción con éste.

6. Conclusiones

Tras la redacción de este documento y puesta en común del punto de vista de los distintos miembros del equipo, hemos sacado varias conclusiones relacionadas con lo aprendido sobre la arquitectura del software y cómo nos hemos enfrentado al proyecto que hemos desarrollado.

Gracias a la asignatura Diseño y Pruebas II hemos profundizado nuestros conocimientos sobre la arquitectura de software en capas. Dejando de lado la complejidad propia de la asignatura, no hemos tenido inconvenientes en comprender y adaptarnos al uso de la arquitectura de software propuesta, ya que, aunque incluyendo más capas, teníamos conocimientos previos y experiencia con el patrón Modelo-Vista-Controlador.

En comparación a las arquitecturas de software conocidas previamente por el equipo, consideramos que este nuevo modelo de arquitectura resulta más completo, además de fácil de entender, ya que proporciona más facilidades a la hora de estructurar el código, permitiendo un mejor reparto y separación de tareas entre las distintas capas de la aplicación.

De manera consensuada, podemos asegurar que nos encontramos satisfechos con el trabajo realizado y el cómo nos hemos adaptado al reto propuesto por esta asignatura.

7. Referencias bibliográficas

-Sección 4:

Wikipedia (Patrones de arquitectura):

https://es.wikipedia.org/wiki/Patrones_de_arquitectura