DP2 2021-2022

D05 – Temas Avanzados

Informe de aprendizaje sobre la arquitectura de un Sistema de Información Web

URL Github: https://github.com/migueegomez7/Acme-Toolkits

Miembros:

- Miguel Ángel Gómez Gómez (miggomgom1@alum.us.es)
- Dámaris Gómez Serrano (damgomser@alum.us.es)
- Mariano Martín Avecilla (marmarave@alum.us.es)
- Iván Moreno Granado (ivamorgra@alum.us.es)
- Miguel Ángel Rivas Rosado (migrivros@alum.us.es)
- Rafael Sanabria Espárrago (rafasana9@gmail.com)

GRUPO E3.04

Versión 1.0.1

02-06-2022

Tabla de contenidos

Historial de versiones

Fecha	Versión	Descripción de los cambios	Sprint
30/05/2022	V1.0.0	• Creación del documento • Introducción • Resumen ejecutivo	5
02/06/2022	V1.0.1	• Contenido • Conclusiones	5

Introducción

En este informe sobre la arquitectura de un sistema de información web se describe a rasgos generales el aprendizaje que se ha llevado a cabo en la asignatura de “Diseño y Pruebas 2”.

Al principio, se comentan las características generales de la arquitectura.

Posteriormente, se describe cada uno de los elementos principales de dicha arquitectura y, finalmente, se indican una serie de ventajas y desventajas.

Resumen ejecutivo

El objetivo de este documento es dar a conocer los conocimientos obtenidos en el curso de esta asignatura sobre alguna de las posibles arquitecturas existentes para un WIS.

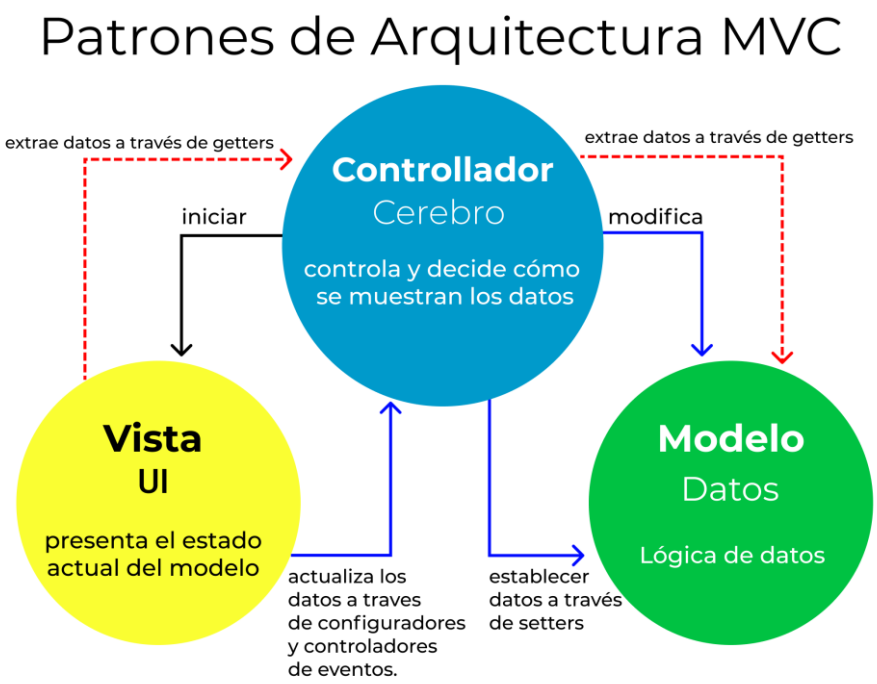
Para ello se han resumido la arquitectura en capas y, en concreto, el patrón MVC.

Contenido

Arquitectura WIS conocida

Arquitectura en capas

- Es una de las arquitecturas más utilizadas, debido a su simplicidad.
- Se trata de dividir la aplicación en capas, con la intención de que cada una de ellas tenga un rol definido, como podría ser, una capa de presentación (Interfaz de Usuario), una capa de reglas de negocio (servicios)…
- Sin embargo, esta arquitectura no define exactamente cuantas capas debe de tener la aplicación, sino en la separación de dicha aplicación en capas.
- Lo más utilizado es una división en 3 capas (presentación, negocio y base de datos).
- Cada capa debe de ser un componente independiente, de tal forma que se puedan desplegar por separado.
- Respetar el orden de las capas es muy importante.



Este modelo aplicado a SpringFramework se traduce en:

- **Entidades:** son las clases básicas que reciben los datos que serán introducidos en la base de datos.
- **Repositorios:** son las clases mediante las cuales se accede a la base de datos desde los servicios.
- **Servicios:** son las clases que implementan diferentes métodos con los datos obtenidos del repositorio, como creado, listado o borrado.
- **Controlador:** Son las clases que conectan al usuario con la aplicación. Implementan los endpoints de la aplicación, utilizando los métodos de los servicios y dirigiéndolos a unas vistas.
- **Vistas:** son las pantallas con las que interactúa un usuario y son creadas por un controlador.

Ventajas

- **Separación de responsabilidades:** cada capa tiene una sola responsabilidad.
- **Fácil de desarrollar.**
- **Fácil de probar:** permite probar por separada cada capa.
- **Fácil de mantener:** Debido a que cada capa hace una tarea muy específica, es fácil detectar el origen de un bug para corregirlo.
- **Seguridad.**

Desventajas

- **Performance:** el hecho de tener que comunicarnos de capa en capa puede hacer que haya un rendimiento más lento en nuestra aplicación.
- **Escalabilidad:** Las aplicaciones que implementan este patrón por lo general tienda a ser monolíticas, lo que hace que san difíciles de escalar.
- **Complejidad de despliegue:** existe una dependencia en el despliegue por lo que un pequeño cambio puede requerir el despliegue completo de la aplicación.
- **Tolerancia a los fallos:** Si una capa falla, todas las capas superiores comienzan a fallar en cascada.

¿Qué hemos aprendido?

Hemos aprendido a preparar un buen espacio de trabajo, importando librerías y plugins que más tarde iban a ser necesarios, y hemos aprendido a adaptarnos al funcionamiento de estas librerías externas, como el framework.

Hemos aprendido a trabajar con muchas tecnologías diferentes en las que no somos expertos, entre ellas, se encuentran nuevos frameworks de test, como geckodriver, o diferentes herramientas, como Dbeaver para la gestión de la base de datos.

Hemos aprendido a acotar los comportamientos inesperados del sistema de información para facilitar el trabajo de mentoría/tutoría de un experto cuando sea necesario

Hemos aprendido de la importancia de realizar test funcionales que permitan asegurar el correcto funcionamiento de la aplicación, usando el modo “marioneta” que implementa el framework.

Hemos aprendido la importancia de una correcta estimación del esfuerzo en las tareas para la consecución exitosa del proyecto.

Hemos aprendido la importancia de entender el funcionamiento de cada componente del sistema informático de manera individual y la separación de responsabilidades para poder comprender a vista de pájaro el funcionamiento coordinado y completo del sistema.

Conclusiones

En general, en el documento se detalla el aprendizaje obtenido por los miembros del equipo de trabajo sobre la arquitectura de un Sistema de Información Web en la asignatura de Diseño y Pruebas II.

Al ser la arquitectura en capas una de las arquitecturas más demandadas hoy día en la industria, se ha visto en esta asignatura y en otras anteriores. En cuanto al conocimiento, hemos podido ampliarlo mediante un punto de vista tecnológico completamente diferente al implementado anteriormente en otras ocasiones. Hemos aprendido otra forma de implementar los principales elementos de esta arquitectura, como son los controladores, las vistas y los servicios. En cuanto a la forma de implementar los repositorios no hemos notado gran diferencia.

Por otra parte, hemos aprendido la importancia de la adaptación a un Framework predefinido. El principal objetivo es que éste sea nuestro “amigo” y no nuestro “enemigo”.

Además de esto, hemos aprendido la importancia de una buena gestión de la configuración de nuestro proyecto, teniendo en cuenta desde el código fuente hasta documentos, pasando por archivos ejecutables y librerías.

La arquitectura descrita en este documento y el aprendizaje posterior sobre ella ha sido consensuada por todos los miembros del grupo mediante una reunión de grupo.

Bibliografía

Intencionadamente en blanco.