DP2 2021-2022

Knowledge of WIS' architecture

Repositorio:

https://github.com/mpadillatabuenca/Acme-Toolkit.git

Miembros:

- José Manuel Bejarano Pozo (josbezpoz@alum.us.es)
- Mario Espinosa Rodríguez (maresprod5@alum.us.es)
- Andrea Meca Sánchez (andmecsan@alum.us.es)
- Manuel Padilla Tabuenca (maresprod5@alum.us.es)
- Ezequiel Pérez Sosa (ezepersos@alum.us.es)
- Javier Terroba Orozco (javteroro@alum.us.es)

GRUPO G1-E2-05

Versión 1.0.0

25/02/2022

Tabla de contenidos

Historial de versiones

Fecha	Versión	Descripción de los cambios
25/02/2022	1.0.0	Creación y finalización del documento.

Introducción

En este documento, realizado por Manuel Padilla Tabuenca hablaré de forma individual sobre los conocimientos previos a la asignatura de Diseño y Pruebas II en cuanto a arquitectura de los sistemas de información.

La estructura de los contenidos será de la forma:

- **Diseño de arquitecturas:** Principales diferencias entre patrón y arquitectura, diferentes estilos de diseño.
- **Patrones de diseño:** Patrones comúnmente utilizados en el desarrollo de software.
- **Antipatrones de diseño:** Fallos que se realizan de forma inconsciente y generan malos olores a la hora de desarrollar software.

Diseño de arquitecturas

En cuanto al diseño de sistemas de información web y arquitecturas de desarrollo software, los conocimientos adquiridos a lo largo de la carrera de Ingeniería Informática del Software, se centran en la definición, tipos de arquitecturas y las diferencias entre arquitecturas y patrones.

Definición de arquitectura software:

Se puede definir el término de arquitectura software como el conjunto de decisiones principales que se toman a lo largo del desarrollo de aplicaciones web, estas decisiones podrían ser, entre otras: qué framework utilizar, atributos de calidad a priorizar, así como el estilo arquitectónico principal.
Cabe destacar que para la toma de estas decisiones, se utilizan herramientas como principalmente diagramas UML.

Principales tipos de arquitectura software:

Como he nombrado en el apartado anterior, los tipos, o estilos de arquitectura software sobre los que más conocimiento poseo son: capas, microservicios y tuberías y filtros.

Diferencias entre arquitectura software y patrón de diseño:

La principal diferencia entre la arquitectura software y los patrones de diseño son su magnitud, mientras que los patrones de diseño se centran en solucionar problemas que vayan surgiendo a lo largo del desarrollo, la arquitectura software engloba a todo el proyecto, y define el esqueleto principal a seguir.

Podríamos poner como ejemplo el de una casa de dos plantas: la arquitectura sería decidir construir dicha casa de dos plantas, el patrón decidiría la mejor solución entre escaleras o ascensor.

Patrones de diseño

En cuanto a patrones de diseño, los conocimientos se centran en su propia definición y diferencias con el concepto de estilo o arquitectura software (explicados anteriormente), principales patrones de diseño y buenas prácticas.

Principales patrones de diseño:

Los principales patrones de diseño sobre los que he programado, y he recibido información son: modelo-vista-controlador, constructor, prototipo, estados y cadenas de responsabilidades.

Buenas prácticas:

En cuanto a buenas prácticas, entre otras: evitar números mágicos, comentarios largos, trozos de código sin utilizar, tener una buena política de ramas y commits.

Antipatrones de diseño

En cuanto a antipatrones de diseño, mientras que los patrones son herramientas utilizadas para conseguir código más limpio y un desarrollo más fructífero del software, los antipatrones son prácticas en las que las personas pueden caer al programar.

Estos antipatrones generan deuda técnica, que se acaba pagando en horas extras de desarrollo, realizando pruebas de más, eliminación de código ilegible, refactorización, etc…

Conclusión

Las conclusiones obtenidas gracias a la realización de de este documento son principalmente, el pequeño resumen de las destrezas adquiridas a lo largo de mi carrera, pienso que en cuanto a la arquitectura de sistemas web poseo un buen nivel de conocimiento, sin embargo, me gustaría ahondar en otros temas como buenos patrones de diseño, o cómo trabajar en grupo con una herramienta como GitHub.

Espero que asignaturas de este segundo cuatrimestre como Diseño y Pruebas 2, o Proceso y Gestión Software 2 me ayuden a adquirir estos conocimientos nombrados.

Bibliografía

1. Apuntes de la asignatura de Diseño y Pruebas 1 (ev.us.es)
2. Lista de patrones de diseño (https://sourcemaking.com/design_patterns)