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

28/02/2022

Tabla de contenidos

Historial de versiones

Fecha	Versión	Descripción de los cambios
28/02/2022	1.0.0	Creación y finalización del documento.

Introducción

En este documento realizado por Andrea Meca Sánchez explicaré mis conocimientos previos sobre las arquitecturas software.
En él se tratarán los patrones arquitectónicos que he utilizado a lo largo de mi desarrollo académico previo a la realización de esta asignatura.

La estructura del documento consta de la actual introducción al documento, el contenido donde se detallarán los distintos patrones, la conclusión y la bibliografía.

Contenido

A continuación explicamos brevemente los patrones arquitectónicos vistos. Estos patrones son una forma de expresar una estructura de organización base o esquema para un software.

Modelo-Vista-Controlador

Este patrón, también conocido como patrón MVC, divide una aplicación interactiva en 3 partes, como

- **Modelo** — contiene la funcionalidad y los datos básicos
- **Vista** : muestra la información al usuario (se puede definir más de una vista)
- **Controlador** : maneja la entrada del usuario

Esto se hace para separar las representaciones internas de información de las formas en que se presenta y acepta la información del usuario. Desacopla los componentes y permite la reutilización eficiente del código.

Arquitectura en capas

Los patrones de arquitectura en capas son patrones de n niveles donde los componentes están organizados en capas horizontales. Este es el método tradicional para diseñar la

mayoría de los programas informáticos y está destinado a ser auto-independiente. Esto significa que todos los componentes están interconectados pero no dependen unos de otros

Se tienen cuatro capas.

- **Capa de presentación:** Se trata de la interfaz que ve el usuario.
- **Capa de negocio:** Donde se procesa las reglas de negocio
- **Capa de persistencia:** Donde se almacenan u obtienen datos.
- **Capa de base de datos:** Donde se tienen los proveedores de bases de datos.

Arquitectura de microservicios.

La arquitectura de microservicios es un método de desarrollo de aplicaciones software que funciona como un conjunto de pequeños servicios que se ejecutan de manera independiente y autónoma, proporcionando una funcionalidad de negocio completa. En ella, cada microservicio es un código que puede estar en un lenguaje de programación diferente, y que desempeña una función específica. Los microservicios se comunican entre sí a través de APIs, y cuentan con sistemas de almacenamiento propios, lo que evita la sobrecarga y caída de la aplicación.

Arquitectura Cliente-Servidor

Este patrón consiste en dos partes; un servidor y múltiples clientes. El componente del servidor proporcionará servicios a múltiples componentes del cliente. Los clientes solicitan servicios del servidor y el servidor proporciona servicios relevantes a esos clientes. Además, el servidor sigue escuchando las solicitudes de los clientes.

Conclusión

En conclusión podemos decir que se tienen conocimientos sobre varios tipos de patrones arquitectónicos de software.

Bibliografía

1. Apuntes de la asignatura de Diseño y Pruebas 1 (ev.us.es)
2. Patrones de diseño comunes (Los 10 patrones comunes de arquitectura de software | by Wilber Ccori huaman | Medium)

3. Más tipos de patrones de diseño (5 patrones comunes de arquitectura software)