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	Creación del documento y primera versión del mismo

Resumen ejecutivo

¿Qué estilos arquitectónicos conozco?
Los dos principales estilos arquitectónicos que conozco son el de capas y el de microservicios.
¿Qué patrones arquitectónicos conozco?
Los que conozco principalmente son:

- Model and view controller
- Front controller pattern
- Single page application

Introducción

En este reporte, realizado de forma individual por mí, hablaré acerca de mis conocimientos previos a diseño y pruebas 2 acerca de las pruebas de un sistema de información web,

centrándome especialmente en los conocimientos adquiridos en otras asignaturas durante la carrera.

Seguiré la siguiente estructura:

- Conceptos relevantes.
- Estilos arquitectónicos.
- Patrones arquitectónicos.

Contenido

Conceptos relevantes

- Cruft: Diferencia entre el código actual y el ideal. A mayor sea más tiempo se ha de invertir luego cuando sea necesario añadir nuevas funcionalidades.
- Patrones de diseño: Soluciones generales a problemas recurrentes en el diseño.
- Librerías: Aportan código reutilizable, generalmente de terceros.
- Framework: Conjunto integrado de artefactos software que implementan una arquitectura reutilizable. Es necesario implementar sobre ellos la funcionalidad específica del sistema.
- Diferencia entre estilo y patrón arquitectónicos: El patrón es una solución general a un problema recurrente mientras que el estilo afecta desde un principio tanto a la estructura del proyecto entero como a la forma en que se organiza el mismo.

Estilos arquitectónicos

Los estilos que conozco me fueron introducidos principalmente en las asignaturas AISS, ISSI2 y DP1. De estos estilos los que mejor recuerdo y además he empleado en la carrera son los dos siguientes:

Capas

Consiste en dividir la funcionalidad del sistema en capas. En cada una se agrupan componentes que ofrezcan una funcionalidad común y dependen de la funcionalidad ofrecida por la capa que está justo debajo. Cada capa debe ofrecer una interfaz pública estable. Normalmente suele haber 3 o más capas. En algunos casos puede variar la forma de interacción entre capas.

- Ventajas:
  - Alta cohesión y bajo acoplamiento.
  - Promueve la separación de responsabilidades.
  - Capas independientes con desarrollo en paralelo.

- Desventajas:
    - A veces es compleja la separación de capas.
    - Disminuye el rendimiento de la aplicación.

Microservicios

En este estilo arquitectónico el sistema es un conjunto de microservicios que se comunican mediante mecanismos ligeros. Cada servicio se desarrolla con la tecnología más apropiada para realizar esa tarea. Suele haber una única interfaz de usuario.

- Ventajas:
    - Modularidad, cohesión, separación de responsabilidades y ocultación de la información.
    - Desarrollo, despliegue y escalado de microservicios por separado.
- Desventajas:
    - Comunicación lenta y riesgos de fallo de comunicación.
    - Consistencia entre distintos almacenes de datos.
    - Complejidad mayor por gestionar muchos servicios.

Patrones arquitectónicos

Model View Controller(MVC)

Consiste en dividir la aplicación en 3 componentes:

- Modelo: Representación de la información, en este componente se incluyen los datos y la lógica de negocio.
- Vista: Aquí se encuentra la interfaz de usuario.
- Controlador: Este componente se encarga de responder a eventos de la interfaz de usuario, provocando cambios en el modelo en caso de que sea necesario y en la vista.
- Ventajas:
    - Soporte para múltiples vistas
    - Favorece la alta cohesión, el bajo acoplamiento y la separación de responsabilidades.
- Desventajas:
    - Mayor complejidad.

Front controller pattern

Consiste en que haya una clase/función que gestione todas las peticiones y las redirija al gestionador apropiado. En java se crean como servlets. Dispatcher Servlet sería el encargado de invocar al método al que le corresponda responder a esa petición.

- Ventajas:
    - Un controlador centralizado permite reforzar políticas que afecte a toda la aplicación como la seguridad.
    - Suele venir implementado en los framework.
    - Mayor funcionalidad.
    - Puede aportar funcionalidad extra.

- Desventajas:
    - Puede afectar al rendimiento y la escalabilidad.

Single Page Application(SPA)

Consiste en desarrollar la aplicación de forma que el cliente al acceder solo carga una página html y ésta no se recarga. Al interactuar con la página el código se reescribe automáticamente.

- Ventajas:
    - Tiempo de respuesta rápido y fluido.
    - Sencillo de construir y de hacer debug.
    - Transición suave al desarrollo móvil.
    - Buena integración con capas y microservicios.

- Desventajas:
    - Tiempo de espera inicial largo.
    - Sigues necesitando esperar a los datos.
    - Requieres frameworks javascript específicos.

Conclusión

En la parte de contenido he expuesto mis principales conocimientos en lo referente a la arquitectura de los sistemas de información web, aunque sí es cierto que conozco más patrones y estilos arquitectónicos en mucha menor profundidad, ya que o no los he usado personalmente o hace más tiempo desde que los estudié. También hay otros que he considerado de menor relevancia y no los he introducido en este documento para no hacerlo excesivamente largo. Como he comentado, en cuanto a las arquitecturas de un sistema de información web no parto desde cero, sino que tengo una base de conocimientos basada en lo aprendido hasta ahora en la carrera, principalmente en diseño y pruebas 1 que además es una asignatura que tengo más reciente que el resto al haberla impartido este mismo año.

Bibliografía

Para desarrollar este informe me he basado en los conocimientos que he adquirido en asignaturas previas y en los apuntes de DP1.