



UNIVERSIDAD DE SEVILLA

DISEÑO Y PRUEBAS 2

Architecture WIS Knowledge

Group: C1.031

Repository: <https://github.com/>

Student #1

Name: Cuenca Pérez, Pablo
Email: pabcueper@alum.us.es

Student #2

Name: Núñez Sánchez, Juan
Email: juanunsan2@alum.us.es

Student #3

Name: Peña Núñez, Felipe
Email: felpennun@alum.us.es

Student #4

Name: Muñoz Romero, Javier
Email: javmunrom@us.es

Student #5

Name: Paradas Borrego, Álvaro
Email: alvparbor@alum.us.es

Date: Seville February 14, 2024

Contenido

Resumen Ejecutivo	4
Registro de cambios	5
Introducción	6
Contenido	7
Conclusión	8
Bibliografía	9

Resumen Ejecutivo

Durante toda la etapa de aprendizaje en la ETSII hemos estudiado multitud de arquitecturas que explican cómo cada una de ellas solucionan diversos problemas recurrentes, dando mucha importancia a cómo estas facilitan la planificación y diseño del software siguiendo un patrón.

La arquitectura nos permite planificar a priori nuestro desarrollo y elegir el mejor conjunto de herramientas para llevar a cabo nuestros proyectos. Así conseguimos que nuestros proyectos mejoren en diferentes aspectos principalmente en escalabilidad, seguridad, rendimiento, mantenibilidad e integración con otros sistemas.

Registro de cambios

Versión	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
1.0	Primera versión	Juan Núñez Sánchez	14/02/2024
2.0	Última Versión	Juan Núñez Sánchez	15/02/2024

Introducción

La arquitectura de un sistema de información web permite anticipar problemas que suelen ocurrir a menudo antes de que se materialicen. Para ello se elige un estilo arquitectónico, es decir, es un conjunto de decisiones de diseño aplicables en un contexto específico, restringiendo a este contexto las decisiones que tomar y planteando un objetivo enfocado en las cualidades del sistema resultante.

Por tanto, debemos tener en cuenta las cualidades, componentes y conexiones de nuestro sistema y elegir un estilo arquitectónico que se adecue mejor a nuestro sistema.

Cada estilo tiene sus ventajas y desventajas, y cada uno se aplica en un contexto y problema específico.

Podemos destacar algunas de las arquitecturas más comunes: arquitectura de capas, arquitectura cliente-servidor, arquitectura orientada a servicios y el patrón arquitectónico MVC (modelo, vista, controlador).

Contenido

La arquitectura de software es un criterio fundamental en la creación de un sistema software. Determina la organización, estructura y relaciones entre los componentes de un sistema, generando unos cimientos consolidados para su diseño, implementación y evolución.

Las diferentes arquitecturas que existen pueden mejorar diferentes aspectos y características que se consideran muy importantes, aunque pueden provocar desventajas sobre otros aspectos. Algunas de estas características son:

- **Escalabilidad:** Es la capacidad de un sistema software para expandirse para ampliar sus funcionalidades y alcance.
- **Seguridad:** Es la capacidad de un sistema software para soportar ataques externos.
- **Rendimiento:** Mide la eficacia en la que un sistema software cumple las funcionalidades para lo que estaba pensado.
- **Mantenibilidad:** Es la capacidad de un sistema software para cambiar y realizar correcciones.
- **Integración:** Es la capacidad de un sistema software de conectarse o fusionarse con otros sistemas

Cada estilo arquitectónico tiene sus ventajas y desventajas, y cada uno se aplica en un contexto y problema específico. Algunas de las arquitecturas más comunes y el contexto donde resultan idóneos son:

- **Arquitectura de capas:** Aplicaciones donde haya que desarrollar y mantener partes del sistema independientemente. Organiza el sistema en capas o niveles, donde cada capa tiene una función específica y se comunica con las capas adyacentes.
- **Arquitectura cliente-servidor:** Una parte conocida como la parte del cliente, la interfaz de usuario; y una parte llamada parte del servidor, que contiene la lógica de negocio y almacenamiento.
- **Arquitectura orientada a servicios:** Consiste en la creación de servicios independientes y reutilizables que pueden ser utilizados por diferentes aplicaciones.
- **Patrón arquitectónico MVC:** Patrón arquitectónico que separa datos de una aplicación, interfaz de usuario y la lógica del negocio en tres componentes distintos.

Conclusión

La arquitectura de un sistema de información web nos permite tomar decisiones de diseño en una etapa temprana del desarrollo y solucionar problemas recurrentes. Para ello conseguimos mejorar características como, por ejemplo, la escalabilidad y la integración. No obstante, puede provocar algunas desventajas en otros aspectos como el rendimiento, mantenimiento, seguridad...

Bibliografía

<https://es.linkedin.com/pulse/arquitectura-de-software-fundamentos-características-y-ojeda-montoya>