

Universidad de Sevilla
Escuela Técnica Superior de Ingeniería Informática
Conocimientos previos arquitectura WIS



Grado en Ingeniería Informática – Ingeniería del Software

Diseño y Pruebas 2

Curso 2022 – 2023

Fecha	Versión
17 / 02 / 2023	V1.1

Grupo: C1.02.05	
Repositorio: https://github.com/alepervaz/Acme-L3-D01.git	
Miembros	Correo
Pérez Vázquez, Alejandro	alepervaz@alum.us.es
Rosso Ramírez, Francisco de Asís	frarosram@alum.us.es
Santiago Félix, Alejandro	alesanfel@alum.us.es
Santiago Sánchez, Sergio	sersansan2@alum.us.es
Vico Martín, María	marvicmar@alum.us.es

Tutor: Soria Morillo, Luís Miguel

Tabla de Contenidos

Tabla de Revisión	3
Resumen	4
Introducción	5
Arquitectura WIS	6
Conociendo la arquitectura	6
Nuestra experiencia	7
Ventajas	7
Desventajas	7
Conclusiones	8
Bibliografía	9

Tabla de Versiones

Nº de revisión	Fecha	Descripción	Sprint
1.0	15 / 02 / 2023	Se detallan las características principales de WIS y nuestra experiencia con la arquitectura.	1

Resumen

Con el objetivo de mostrar el contraste entre los conocimientos previos a la asignatura y los posteriores a la misma, se va a detallar todo aquello que el grupo conocía acerca de arquitecturas WIS previamente.

Introducción

Durante el desarrollo de un proyecto podemos observar como conforme aumentamos las líneas de código escritas aumenta la complejidad del mismo. A largo plazo esto puede suponer un problema para incorporar nuevas funcionalidades, solucionar fallos o aplicar mejoras. La arquitectura WIS nos permite separar la lógica de la aplicación en capas y facilitar el desarrollo.

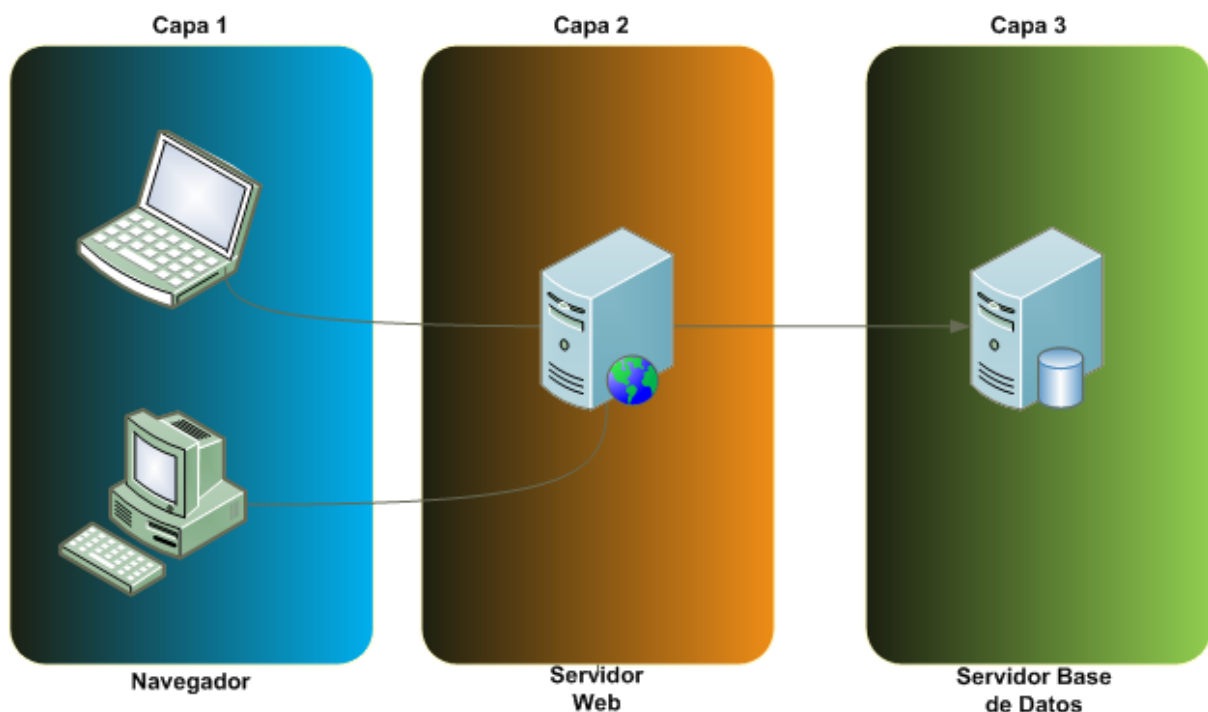
A la par que detallaremos la información adquirida sobre las arquitecturas WIS, detallaremos brevemente nuestra experiencia con el framework Spring, ya que ha sido el único caso en el que hemos desarrollado dicha arquitectura en todos sus niveles: presentación, lógico y datos.

Arquitectura WIS

Conociendo la arquitectura

La arquitectura WIS nos permite dividir la lógica de la aplicación en tres capas fundamentales: presentación, lógica y datos. El objetivo es aumentar la escalabilidad del sistema y su mantenimiento.

- Capa de presentación: Representa todo aquello que ve el usuario del sistema, es decir, aquella persona o sistema inclusive que use la aplicación desarrollada, por ejemplo: el JSON generado por una api formaría parte de esta capa. Trata de dar forma a los datos manipulados por la capa lógica para hacerla accesible a otros sistemas.
- Capa lógica: Emplea la información presente en la capa de datos en una serie de procesos con el objetivo de llegar a algún fin concreto. Un ejemplo podría ser obtener la distancia entre dos puntos dados.
- Capa de datos: Se encarga de almacenar la información de la aplicación, es decir, la persistencia de datos. Un elemento muy importante de esta capa es la base de datos, al igual que las interfaces que permiten acceder a la información que contiene.



[Enlace imagen capas](#)

Nuestra experiencia



[Enlace spring](#)

En nuestro caso, para llevar a cabo la arquitectura WIS empleamos el framework Spring, el cual permite montar sistemas web. Para la capa de presentación contamos con los archivos JSP, luego para la lógica de negocio utilizamos código java y por último para la capa de datos hacemos uso de hibernate. Este último es un ORM, es decir, un sistema que nos permite mediante manipular una base de datos relacional con código java en lugar de sql nativo, por ejemplo.

Ventajas

Trabajando con esta arquitectura hemos podido detectar los siguientes puntos a favor de su implementación:

1. Permite diferenciar claramente las funcionalidades de las distintas capas.
2. Aisla posibles errores.
3. Aumenta la escalabilidad, gracias a su bajo acoplamiento.

Desventajas

Al igual que toda clase de tecnología, esta también tiene algún contra, entre ellos destacamos:

1. Posibles pérdidas de rendimiento, no debemos transferir información de la capa de la de datos a la de presentación directamente, por lo que puede suponer una lacra.
2. Aumento de la complejidad inicial del estructuramiento del proyecto.

Conclusiones

Aunque supone un mayor esfuerzo inicial, la arquitectura WIS permite evitar o atenuar posibles problemas futuros en el desarrollo de un sistema. A su vez para aquellas aplicaciones que puedan optar a un crecimiento exponencial o simplemente acelerado puede suponer un factor diferencial en el coste de incremento o mejora.

Bibliografía

Intencionalmente vacío.