

Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática

Sprint 1: Documento de información de un WIS



Grado en Ingeniería Informática – Ingeniería del Software

Diseño y pruebas 2

Curso 2021 – 2022

Fecha	Versión
2/3/2022	1.0

Grupo de prácticas	E7.08	
Repositorio	https://github.com/diecrequi/Acme-Toolkit.git	
Autores		
Nombre	correo	Año de nacimiento
Cadenas Morales, Alfonso	alfcadmor@alum.us.es	30/05/1999
Crespo Quinta, Diego	diecrequi@alum.us.es	26/05/1998
Fraile Rodriguez, Alejandro	alefrarod@alum.us.es	11/10/1999
López Quirós, Juan	jualopqui@alum.us.es	17/03/2001
Oliva Rodriguez, Pedro	pedolirod@alum.us.es	25/06/2000
Quijano Herrera, Antonio	antquiher1@alum.us.es	08/02/2000

Índice

1. Resumen	2
2. Tabla de versión	3
3. Introducción	3
4. Contenido	4
5. Conclusión	4
6. Bibliografía	5

1. Resumen

En este documento se pone por escrito todos nuestros conocimientos sobre la arquitectura de un WIS.

2. Tabla de versión

Fecha	Versión	Descripción
28/02/2022	V1.0	Creación del documento, recapitulación de información de un WIS para la elaboración del contenido
02/03/2022	V1.1	Incluido un resumen, introducción, tabla de versión y conclusión.

3. Introducción

En este documento se hará un estudio de los conceptos aprendidos sobre la arquitectura de las WIS en la asignatura de Diseño y Pruebas I.

Para ello se ha preguntado a los distintos miembros del grupo sobre la información que recordaban de las WIS, posteriormente contrastada con las transparencias de clase del cuatrimestre pasado.

También comentamos los consejos y las exigencias que nos decía nuestro tutor en las distintas revisiones.

Para la elaboración de este documento, se ha optado por un enfoque realizado comentado de manera más genérica al comienzo, para terminar hablando de manera más avanzada en el tema.

4. Contenido

Entendemos como arquitectura de un WIS al “esqueleto” que nos muestra las interacciones entre componentes de la aplicación, interfaces de usuario y bases de datos. También la tratamos como la disposición que define lógicamente la conexión entre servidor y cliente para una mejor experiencia.

De esta manera, separamos un WIS en 3 capas diferentes: capa de web(controladores y jsps), capa de dominio(servicios y entidades) y capa de recursos(repositorio)

La forma en la que trabajamos con un WIS es la siguiente: a través de los jsps, los usuarios envían los parámetros en la url y son los controladores los que operan con ellos. Usando las etiquetas `@PathVariable` and `@RequestParam` podemos acceder a la información en la url, los métodos que recibirán esos parámetros no deben estar nunca en los controladores pues estos pertenecen a la capa de web y para seguir una buena práctica la capa de web solo debe mostrar, enviar y recibir la información.

Si buscamos operar con esos datos, los controladores deben enviarlos a los servicios o llamar al método creador de la entidad correspondiente. Ahí podremos operar con ellos, transformarlos o incluso hacer una consulta al repositorio utilizando los parámetros de entrada nuevamente.

De esta misma manera, podemos parametrizar los códigos de error de nuestra aplicación para mostrar al usuario concretamente cuál ha sido el problema y la mejor manera de solucionarlo.

5. Conclusión

Gracias al estudio realizado en este documento, se ha llevado de manera autodidáctica un análisis sobre la arquitectura de las WIS. Al poner en común los conocimientos básicos que recordábamos del anterior cuatrimestre y al refrescarlos con las transparencias de Diseño y Pruebas I nos encontramos en una mejor situación para atender el contenido de esta asignatura.

6. Bibliografía

-Transparencias de Diseño y Pruebas I