DP2 2021-2022

WIS Architecture

Acme Toolkits

Repositorio:
https://github.com/RingoML/Acme-Toolkits

Miembros:

- Caro Albarrán, Francisco Andrés (fracaralb@alum.us.es)
- Gallego Huerta, Alberto(albgalhue@alum.us.es)
- Martín Luque, José Manuel (josmarluq@alum.us.es)
- Reyes Madrid, Francisco (frareymad@alum.us.es)
- Sillero Manchón, Jorge (jorsilman@alum.us.es)

GRUPO E3.06

27 de febrero de 2022

Índice

Resumen ejecutivo 2
Tabla de revisiones 2
Introducción 2
Contenido 3
Conclusiones 3
Bibliografía 3

Resumen ejecutivo

Tras reunirnos todo el grupo, acordamos que gran parte del conocimiento que poseíamos acerca de la arquitectura WIS era gracias a las asignaturas de AISS, ISSI y DP1; las cuáles fueron nuestra primera toma de contacto con este tipo de arquitectura.

Tabla de revisiones

FECHA	VERSIÓN	DESCRIPCIÓN
2022-02-27	1	Versión inicial

Introducción

En este documento vamos a exponer que es lo que sabemos acerca de la arquitectura WIS hasta el momento. A continuación, vamos a exponer nuestros conocimientos sobre dicha arquitectura, la cual podemos sintetizar en tres partes bastantes diferenciadas: el navegador, la aplicación y la base de datos.

Este documento se divide en las siguientes partes: Índice, Resumen ejecutivo, Tabla de revisiones, Introducción, Contenido, Conclusiones y Bibliografía.

En Contenido se exponen nuestros conocimientos sobre la arquitectura WIS.

Contenido

Nuestros conocimientos previos nos hacen diferenciar entre tres elementos bastante diferenciados dentro de esta arquitectura. Estos son el navegador, encargado de enviar las peticiones HTTP al servidor y de renderizar la respuesta; la aplicación, que con su controlador, servicio, repositorio y vistas es capaz de producir una respuesta que el renderizador se encargará de transformar en el documento HTML; y por último la base de datos, que mediante las distintas tablas y las operaciones CRUD es la encargada de almacenar toda la información necesaria para el correcto funcionamiento de nuestro sistema.

El navegador es el encargado mediante HTML, JS y CSS de la interfaz de nuestro sistema. Este envía peticiones GET/POST que son recibidas por el servidor HTTP y posteriormente renderizadas en documento HTML para la construcción del documento HTML.

La aplicación, encargada de recibir las distintas peticiones del navegador. El controlador, junto con el servicio, el repositorio y algunas vistas y entidades son los responsables de procesar dichas peticiones y devolver una respuesta la cual usará el renderizador para elaborar el documento HTML.

Por último, la base de datos recibe distintos tipos de peticiones que solicitan acceso a los datos que están en ella. Mediante las peticiones *select, delete, create* o *update* la aplicación recibe datos en su repositorio de la base de datos. Estas peticiones son procesadas con éxito por la base de datos gracias a las tablas, claves o índices. Esta devuelve como respuesta una colección de datos o el número de filas que cumplen una condición dada, entre otras.

Conclusiones

El uso de una arquitectura apropiada y correcta, así como entender a la perfección su funcionamiento, es un punto clave del desarrollo. Ya que el uso de una arquitectura incorrecta puede ser una de las principales causas del fracaso de un proyecto.

Bibliografía

The starter project (Theory) - DP2.