

DP2 2024

Acme Software Factory

Repositorio: <https://github.com/DP2-2024-C1-029/Acme-Software-Factory.git>

Miembros:

- David Godoy Fernández (davgodfer@alum.us.es)
- Ismael Gata Dorado (ismgatdor@alum.us.es)
- Jaime Varas Cáceres (jaivarcac@alum.us.es)
- José María Portela Huerta (josporhue@alum.us.es)
- Juan José Gómez Borrallo (juagombor@alum.us.es)

Tutor: José González Enríquez
16/02/2024

GRUPO C1.029
Versión 1.0

Índice

Historial de versiones.....	3
Resumen ejecutivo.....	4
Introducción.....	4
Contenido.....	4
Conclusiones	5
Bibliografía	5

Historial de versiones

Fecha	Versión	Descripción	Entrega
16/02/2024	V1.0	Descripción de los conocimientos de la arquitectura WIS	D01

Resumen ejecutivo

Explicamos la experiencia que tenemos con la arquitectura WIS el modelo de capas, componentes e interacciones entre ellos, es decir, las capas de presentación, aplicación y base de datos, los componentes, navegador para la presentación, servidor y estructura de los componentes para la lógica y la base de datos H2 para la persistencia de datos, además de la interacción entre ellos.

Introducción

En este documento recogemos un breve resumen de lo que conocemos de la arquitectura WIS hasta ahora.

Contenido

Acerca de la arquitectura WIS el equipo tienes algunas nociones y hemos trabajado sobre ella en la asignatura Diseño y Pruebas I.

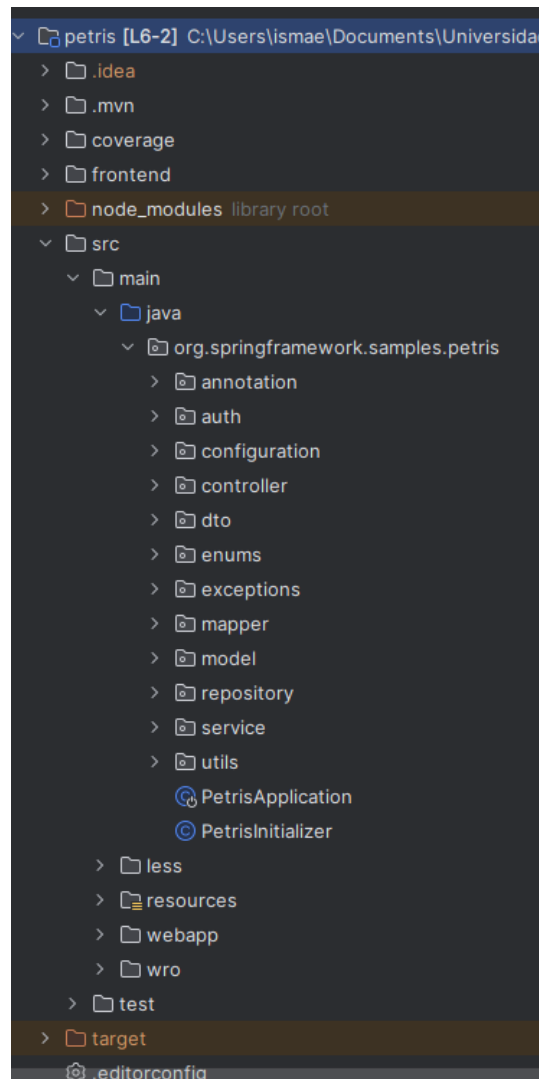
En dicha asignatura tratamos con las diferentes partes de la arquitectura lo que es las capas, componentes e interacción entre ellos.

A nivel de capas teníamos un proyecto muy parecido al que se muestra en la teoría de esta asignatura en donde teníamos un Tomcat embebido por Spring el cual se arrancaba solo al arrancar la aplicación, pero en este caso usábamos una base de datos h2, mas simple y liviana para lo que requiere la asignatura y como capa de presentación usábamos React.

Como componentes teníamos una estructura parecida, el navegador (cualquiera) para la capa de presentación, Tomcat como servidor y una estructura en forma de controlador, servicio y repositorio para la capa de aplicación y la base de datos h2 para la capa de datos.

Y finalmente la interacción entre ellos lo teníamos estructurado de forma que definíamos en un paquete de repository todos los repositorios de la aplicación, en un paquete service todos los servicios, en un paquete controller todos los controladores (peticiones RestApi), en un paquete model las entidades (representación en clases de las tablas de base de datos) y un conjunto de mas paquetes de utilidades, mapeos etc.

Adjunto una captura de ejemplo:



El flujo que tendrían entre ellos es básicamente el mismo descrito en las diapositivas del temario. El cliente (React) llamaría por medio de RestApi lo cual lo procesaría Tomcat y acabaría entrando por el controlador correspondiente de la petición, este llamaría al servicio donde aplicaría toda la lógica de negocio y a su vez este llamaría al repositorio donde a través de JPA e Hibernate harían las modificaciones pertinentes en la base de datos H2.

Conclusiones

En conclusión, podemos confirmar que tenemos una nociones del tipo de Arquitectura y que a lo largo de la asignatura nos resultara familiar lo cual nos ayudara a entender mejor la asignatura y el proyecto.

Bibliografía

Intencionadamente en blanco.