

Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática

Sprint 5

Planning Report



Diseño y Pruebas II

Curso 2021 – 2022

Grupo de prácticas	E3.03
Autores	Email
Toledo Vega, Jorge	jortolhur@alum.us.es
Márquez López, José Antonio	josmarlop16@alum.us.es
Rodríguez García, Luis	luirodgar3@alum.us.es
Mira Otero, Antonio	antmirote@alum.us.es
Cáceres Gómez, José	joscacgom@alum.us.es
Oñate Parra, Julián	julonapar1@alum.us.es

Repositorio: <https://github.com/jvegax/Acme-Toolkits>

¿Qué hemos aprendido sobre la arquitectura WIS?.

Dentro de los conocimientos que poseíamos sobre la arquitectura de un sistema de información, hemos aprendido que hay aspectos importantes de los cuales no éramos conscientes, como son las actualizaciones del propio framework de manera constante, el cual hacía que el desarrollo de la aplicación fuera de una manera más “continua”, debido a que esas actualizaciones cambiaban parte del propio funcionamiento del framework y permitían en algunos casos, facilitar la implementación de los requisitos que nos pedía el cliente, y en otro, modificar lo ya hecho para que funcionara de manera más eficiente.

Otro aspecto a destacar es la importancia de la valoración de los requisitos que nos pedía el cliente, el cual debíamos tomar decisiones de diseño acordes con lo que nos solicitaba, pero también teniendo en cuenta lo que nos aporta el propio framework y nuestros conocimientos como programadores del propio sistema de información.

También cabe mencionar que hemos podido trabajar con archivos csv que nos permitían de forma rápida y eficiente, cargar elementos en la base de datos, estos acordes con los escenarios que planteamos para los propios test y poder probar las funcionalidades implementadas.

Por último, también encontramos similitudes entre el framework usado y Spring, ya que ambos usan modelos MVC, por lo que nos fué relativamente sencillo adaptarnos a trabajar con este framework. El framework de la asignatura está formado por un conjunto de componentes comunicados entre sí y compuestos por:

- Un controlador por paquete (Relacionado con una entidad)
- Al menos un repositorio
- Y tantos servicios como operaciones se requieran, es decir, p.e.: un list, un show y un create.

Todo ello suele estar relacionado en el controlador del paquete, en el cual se llama a todos esos servicios, dependiendo de la acción realizada por el cliente.

También hay que tener en cuenta que hay que crear al menos una vista por cada paquete, para poder mostrar toda la información sobre las entidades y las operaciones sobre ellas.

Con todo ello, al final hay que crear al menos un archivo .csv en el que se inicializan los primeros ejemplos de cada entidad, para poder probar el correcto funcionamiento de las vistas y de los servicios creados.

En general, el funcionamiento y la organización al principio fué un poco caótica, pero una vez tomamos de ejemplo Acme-Jobs, todo se hizo mucho más sencillo y práctico.