

DP II - 2021-2022

LESSONS LEARNT ABOUT WIS ARCHITECTURE

https://github.com/ddiazlop/Acme_Recipes_G10

Miembros:

- Carolina Carrasco Díaz
- Diego Jesús Díaz López
- Julio Navarro Rodríguez
- Salvador Parejo Ramos
- María Isabel Pedraz Rodríguez
- Alvaro Sevilla Cabrera

Tutor: RAFAEL CORCHUELO GIL

GRUPO - G10

1 ÍNDICE

Resumen Ejecutivo	1
Tabla de Versionado	1
Introducción	2
Análisis general de la arquitectura	2
Lógica de la arquitectura	2
Lecciones aprendidas	3
Conclusión	6
Bibliografía	6

2 RESUMEN EJECUTIVO

Para la realización de este documento se ha revisado la documentación disponible en la enseñanza virtual en relación a las prácticas como a la teoría, además de distintas fuentes consultadas. Se pretende abordar todos los conocimientos adquiridos en cuanto a la arquitectura WIS y el Framework aportado en la asignatura, así como las lecciones aprendidas a lo largo de esta asignatura en cuanto a la gestión del proyecto mediante las diferentes herramientas empleadas.

3 TABLA DE VERSIONADO

Versión	Fecha	Descripción
1.0	05/09/2022	Inicio y realización del documento
1.1	06/09/2022	Añadidas algunos puntos extra al documento (concepto SUT y lecciones aprendidas)
1.2	07/09/2022	Adjuntados en la Bibliografía los enlaces y referencias de los artículos utilizados para la realización y mejor adecuación de este documento
2.0	08/08/2022	Revisión y cierre del documento

4 INTRODUCCIÓN

En este documento haremos un análisis general sobre la arquitectura de un WIS así como lo que hemos aprendido en la asignatura sobre ello.

También se indicará la lógica de la arquitectura, el concepto de SUT, lo aprendido a lo largo del uso del Framework durante la realización de este proyecto. Por último, también se indicarán las buenas prácticas seguidas a lo largo de la ejecución del proyecto.

5 ANÁLISIS GENERAL DE LA ARQUITECTURA

A continuación, paso a definir generalmente la arquitectura implementada.

Esta arquitectura que hemos utilizado, es una arquitectura basada en capas y que consta de 4 capas: capa interfaz de usuario, capa de servidores webs, capa de aplicación y capa de datos.

La primera o la que consta de más abstracción es la capa que se muestra por el navegador es decir la capa de interfaz de usuario y se encarga de interpretar las interacciones de los usuarios y convertirlas en peticiones HTTP.

Después pasamos a la segunda capa que sería una capa intermedia que consta de servidores webs que se encarga de renderizar, redireccionar a las vistas correctas e interpretar las peticiones y protocolos HTTP.

También está la capa de aplicación que es donde se lleva a cabo toda la lógica de negocio y está compuesta por los controladores, el modelo con las entidades, servicios, repositorios y vistas.

Por último encontramos la capa datos que es la que se encarga de hacer que los datos persistan en nuestra base de datos así como de gestionar las queries de datos. Esta está compuesta por nuestra base de datos y su servidor.

6 LÓGICA DE LA ARQUITECTURA

La lógica de esta arquitectura se basa en la comunicación entre capas mediante diferentes tipos de peticiones.

Generalmente comienza cuando un usuario realiza una petición desde el navegador la cual es interpretada por el navegador, convertida en una petición HTTP y enviada al servidor HTTP, el cual lo envía al servlet. El Servlet determina cuál de los controladores es el indicado para procesar la petición indicada y se la envía.

Los controladores siempre siguen el mismo proceso para resolver una petición aunque este varía algo dependiendo del método que haya solicitado la petición, ya sea un POST o un GET, o el comando que se requiera.

Lo primero que se lleva a cabo en un controlador es mandar una petición a la base de datos para abrir una transacción y que así se puedan llevar a cabo modificaciones, esto es gestionado por el trx manager de la base datos.

Dentro de esta transacción se llama al servicio encargado de realizar la operación que requiera la petición sobre la base datos, el trx manager permite gestionar varias operaciones de forma concurrente ya que cada una se realiza dentro de su proceso individual, después de terminar la modificación de datos estos son persistidos en la base de datos y se redirige la vista requerida por la petición.

7 LECCIONES APRENDIDAS

Todos los miembros del grupo hemos sido capaces de entender y crear todos los tipos de componentes de los que consta el WIS, ya que todos hemos participado en la creación de algunas entidades y posteriormente hemos ido haciendo todas las funcionalidades de todos los tipos de roles sobre estas entidades.

Por lo que todo el equipo ha participado en la creación de: controladores, servicios, repositorios y vistas sobre varias entidades.

Aprender a gestionar un WIS durante la ejecución de este proyecto ha facilitado la interoperabilidad entre los compañeros, aportando una mayor facilidad y flexibilidad a la hora de trabajar en equipo, proporcionando una estabilidad al proyecto desarrollado.

→ APRENDIZAJE Y CONOCIMIENTOS ADQUIRIDOS

Gracias a la ejecución de este proyecto se han puesto en valor las capacidades de realizar un trabajo en conjunto. Como se ha comentado en el documento del D01, era la primera vez que este grupo ha trabajado en conjunto, por lo que se identificaron al principio de la conformación del equipo en el acuerdo formal del grupo para esclarecer y dejar reflejado los acuerdos a los que se ha llegado.

Se ha aprendido el uso de herramientas nuevas, proporcionadas por el profesorado en el Workspace, tales como UMLet, que ha permitido el diseño y modificación del UML solicitado en la entrega, que ha facilitado el diseño y la implementación de la aplicación.

También se ha aprendido a utilizar la herramienta SonarLint, mediante la cual se analiza el código y con la que se ha permitido identificar los “bad-smell” o los malos olores que ha identificado en la aplicación.

A la hora de realizar los análisis de los reportes, se ha aprendido también el uso de las funciones de excel, ya que a lo largo de la carrera no se ha utilizado esta herramienta, y permitirá poder utilizarlas a lo largo de nuestra vida profesional.

La herramienta GitHub ha permitido gestionar el código en un repositorio al que todos los integrantes del equipo teníamos acceso, permitiendo resolver las tareas de manera independiente.

→ A CERCA DEL ACME FRAMEWORK UTILIZADO

ACME FRAMEWORK, ha sido utilizado a lo largo de esta asignatura. Y el cual nos ha permitido adquirir mejor elaboración y control del código elaborado, así como facilitarnos una futura reutilización del mismo en caso de necesitarlo más adelante.

Cabe destacar que el principal objetivo a la hora de utilizar un framework es la capacidad de ofrecer un desarrollo ágil de aplicaciones mediante la aportación de librerías y/o funcionalidades ya desarrolladas, lo cual nos evita el tener que implementar funcionalidades que son de uso común en muchas aplicaciones.

A continuación se enumeran algunas de las ventajas que se han encontrado una vez trabajado con este Acme-Framework. Las cuales, estas ventajas, nos han servido y facilitado durante el desarrollo de todo este proyecto:

- ✓ Mejor estructura y organización del código predeterminado
- ✓ Aporta una mayor agilidad y rapidez en el desarrollo del mismo
- ✓ Permite una futura reutilización del código, evitando además, la duplicidad del mismo.
- ✓ El uso de patrones aporta una adquisición de buenas prácticas de desarrollo.
- ✓ Minimiza los errores y a su vez proporciona una mayor facilidad para solucionarlos
- ✓ Facilidad en cuanto al mantenimiento.
- ✓ Esto también se ve reflejado en cuanto al coste en el desarrollo
- ✓ Y por supuesto, facilita la colaboración con el equipo de desarrolladores del proyecto.

Por último, cabe destacar también, como se ha comentado anteriormente, que gracias a la utilización de Github se ha podido trabajar sin interferir en el trabajo del resto de los integrantes. La realización de modificaciones empleando esta herramienta han sido bastante sencillas y no se han reportado grandes complicaciones a la hora de la realización y desarrollo de las propias tareas. Así como en cuanto a las traducciones, también se han realizado de manera muy sencilla gracias a la utilización de este framework.

→ BUENAS PRÁCTICAS ADQUIRIDAS

En cuanto a las buenas prácticas adquiridas podemos destacar:

- Definir el alcance del proyecto correctamente y previo a su desarrollo..
- Organización del sistema de forma ordenada y óptima, facilitando la interoperabilidad entre los compañeros.
- Legibilidad del código, ya que se ha establecido una estructura, previo al desarrollo del proyecto.
- Realización de feedback.
- Colaboración del equipo.

En general, el framework utilizado presenta una estructura muy útil, la cual se ha seguido a lo largo y en todo el transcurso del proyecto presentado.

Además, por último cabe destacar que el uso de este framework nos ha permitido, a la hora de hacer “merge”, que las ramas no presentaran incompatibilidades y que los elementos generados y/o modificados fueran fácilmente identificables, gracias a la ruta en la que se encontraban y a la distribución seguida.

9 CONCLUSIÓN

Podemos concluir este documento con que se han cumplido y superado los conocimientos de la asignatura así como reiterar que gracias a ellas, se ha obtenido y adquirido una experiencia tanto, hacia las herramientas empleadas como en la gestión de código.

Así como destacar además de la gestión e implementación del framework y de las distintas herramientas empleadas para la realización de este proyecto, cabe mencionar y dar valor e importancia a la realización y clarificación del diseño de la aplicación en el modelo UML, permitiendo organizar de manera acorde a los compañeros una estructura a seguir, de manera clara y concisa. Lo que ha permitido en todo momento, como ya se ha mencionado, organizar el código de una manera óptima, facilitando en todo momento su implementación futura.

10 BIBLIOGRAFÍA

[1]: Garzás, Javier. (2015). ¿Qué estoy probando y cuáles son mis dependencias en testing?: <https://www.javiergarzas.com/2015/09/que-estoy-probando-y-cuales-son-mis-dependencias-en-testing.html>

[2]: Ortega, Dinarle - Guevara, María - Benavides, John (2020). Un framework de programación web: <https://www.redalyc.org/pdf/784/78457627004.pdf>

[3]: Gordillo, E. (s. f.). Buenas prácticas en arquitectura de la información. Inventtatte. <https://inventtatte.com/10-buenas-practicas-de-la-arquitectura-de-la-informacion/>