

Documento de finalización del Sprint 2

Nivel de acabado al que se presentan:

Nuestro grupo se presenta al nivel máximo de acabado (10).

Justificación de cómo se han ido consiguiendo cada uno de los requisitos del nivel señalado y de qué ha hecho cada pareja

En este Sprint, hemos corregido todos los fallos del anterior Sprint. Además, hemos implementado todos los modelos y corregido algunos fallos en los diagramas. También realizamos la división de los diagramas (diseño y dominio). Implementamos todas las reglas de negocio, todos los test unitarios para los modelos, realizamos todos los métodos de los servicios necesarios para las historias de usuario y finalmente llevamos a cabo el proceso de login (con su token) para poder usarlo con React. Por último, implementamos una clase configuración para poder almacenar datos específicos, que sean fáciles de cambiar (SalarioMinimo o NumMaxNotificaciones).

Pareja 1: Tomás Galera Barrera y Alejandro Santisteban Corchos: diagramas UML, corrección de fallos, planificación, modelos, reglas de negocio, test unitarios para los modelos, métodos de los servicios para las historias de usuarios, clase configuración, actualización del documento

Pareja 2: Carlos Atienza Carretero, Lucas Pérez Ruíz y Borja Rondán Domínguez: corrección de fallos, planificación, modelos, reglas de negocio, test unitarios para los modelos, métodos de los servicios para las historias de usuarios, proceso de login (con token) para usarlo con React, actualización del documento

Análisis retrospectivo del Sprint 2

En este segundo Sprint ya teníamos más conocimientos sobre el tema y ya éramos más conscientes de a lo que nos enfrentábamos. Por ello, sabíamos mejor como abordar el proyecto.

En los primeros días nos dedicamos a corregir los fallos del Sprint anterior, además planificamos y discutimos acerca de cómo íbamos a organizarnos en cuanto a la división del trabajo, orden de hacer las cosas...

Conforme pasaron los días e íbamos implementando nuevas cosas, nos íbamos dando cuenta de pequeños errores, atributos faltantes o necesarios, atributos que sobraban, relaciones a cambiar en los UML...

Gracias a los tests descubrimos unos cuantos fallos nuevos relacionado con todo lo anterior.

En cuanto a las relaciones de las entidades, fue algo sencillo al principio, pero que más tarde nos dimos cuenta de que algunas sobraban (tema de bidireccionalidad) o directamente al cambiar cosas en los UML también nos cambiaban estas.

La mayor dificultad fueron las relaciones con herencias, como estas clases no eran entidades, no eran persistentes en BD y las soluciones “a simple vista” requerían una pérdida de información al eliminar tablas o directamente eran complejas e ineficientes.

También nos dimos cuenta de que, a la hora de pasar objetos, solo se podía pasar uno, por lo que creamos varios tipos Tuple (con distinto número de atributos) para ello.

Probablemente, lo que más nos ha hecho dar vueltas y vueltas en este Sprint (tras mucho Googlear y preguntar al tutor) es el test de una CustomAnnotation. Al intentar hacer un Autowired en la clase del ConstraintValidator daba nulo el valor y no era posible realizar la validación, es por ello por lo que optamos finalmente por implementarlo como una excepción tal y como nos sugirió el tutor.

Finalmente, por recomendación del tutor decidimos crear “La Pingupedia”, un documento en el que cada miembro del grupo podía escribir los fallos que se hubiese encontrado durante la realización de sus tareas y explicar tanto el fallo como la solución realizada.

Como ya mencionamos en el Sprint anterior, al querer abordar un proyecto tan grande, todos los miembros del grupo estamos muy implicados con el trabajo, es por ello por lo que prácticamente, cuando un miembro del grupo estaba trabajando, los otros también lo estaban simultáneamente.

Ambas parejas han realizado un buen trabajo en equipo, y la comunicación entre las distintas parejas ha sido fluida y muy eficaz a la hora de resolver problemas.

Prácticamente ya el backend queda casi completo, faltando crear los métodos en los controladores y “conectarlo” con React para empezar a montar las vistas.

Esfuerzos (en horas) de cada pareja:

Para este Sprint, hemos usado parcialmente Clockify. Aproximadamente los miembros del grupo trabajaron simultáneamente (debido al desarrollo de técnicas como pair programming) por lo que cada uno de ellos estuvo en torno a 36 horas.

Pareja 1: Tiempo estimado = 72 horas

Pareja 2: Tiempo estimado = 108 horas

Summary report

11/09/2020 - 11/29/2020



Total: 36:54:01 Billable: 36:54:01 Amount: 0.00 USD

