

UNIVERSIDAD FRANCISCO DE VITORIA

ESCUELA POLITÉCNICA SUPERIOR



Desarrollo e integración de software

Práctica 2

grupoJavalovers_P2

**Guillermo Blasco Pérez
Óscar Martínez Urango
Ignacio Pomares Ricord
Cristian Sevilla Díaz**

Índice de contenidos

1. Introducción	2
2. Git y división del trabajo	3
Análisis de la práctica	3
División del trabajo	3
Git	4
3. Programa	7
Enunciado	7
XML y DTD	7
Pestañas	7
4. Conclusión final	9
5. Bibliografía	10

1. Introducción

El motivo de esta práctica es enseñarnos, a nosotros como estudiantes del grado de ingeniería informática, a trabajar mediante el uso de Git y las herramientas Github mientras realizamos una aplicación de Vaadin sobre SpringBoot que se comunica con la API (backend) hecha con SpringBoot a través de Docker. Dicha práctica ha sido realizada por 4 miembros de la clase de 3ºC. Para ello explicaremos la distribución y el proceso de elaboración del trabajo a lo largo de este periodo de tiempo en el que hemos tenido que realizar la práctica.

A continuación se mostrará la elaboración del resultado del trabajo en Git, como el uso correcto de las herramientas de Github, junto a una breve memoria de los resultados obtenidos en cada una de las pestañas, tanto el front como el back; una explicación del uso del xml y el dtd y los resultados obtenidos en las pruebas unitarias. Con ello se demuestra que el objetivo principal de la práctica se ha cumplido, pudiendo demostrar los conocimientos de la asignatura explicados en clase hasta el momento de hacer esta práctica.

2. Git y división del trabajo

Análisis de la práctica

La primera semana vimos y comentamos como considerábamos cada uno de los integrantes del grupo la elaboración de esta práctica antes de comenzar el desarrollo de la misma. Después de la puesta en común todos teníamos una idea clara de cómo íbamos a afrontar la elaboración a nivel global de esta entrega.

División del trabajo

Expusimos los puntos fuertes de cada uno para realizar un reparto equitativo a las fortalezas y disposición de cada miembro, además seleccionamos un “líder”, que ayudaría a que haya un control en la evolución del trabajo. Al ver que eran 4 opciones decidimos escoger cada miembro una de ellas en base a lo anterior mencionado:

- Ignacio Pomares Ricord → Docker, documentación y pestaña 1 front.
- Óscar Martínez Urango → XML, DTD y test unitarios.
- Guillermo Blasco Pérez (líder) → Parte 2 y 3.
- Cristian Sevilla Díaz → Pestaña 2 front y back.

Además de esto, Guillermo se prestó voluntario para realizar la creación de los repositorios ya que iba a comenzar su parte antes que el resto de compañeros del grupo. Después de terminar alguno el trabajo que le había sido otorgado, quedaba a disposición de alguna necesidad o problema que surgiera a lo largo de la práctica.

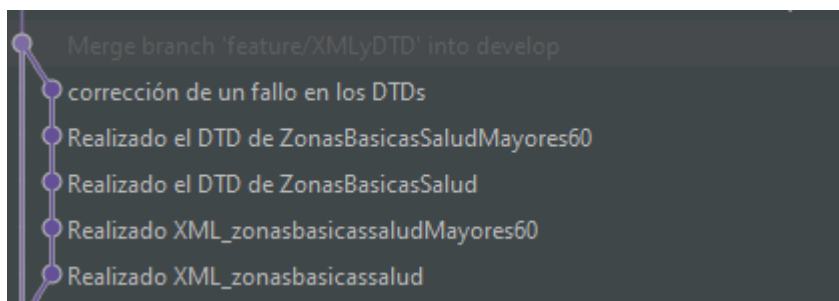
También, según se iban alcanzando objetivos, el líder iba organizando al resto de miembros en base a las comprobaciones de los errores para seleccionar quién iba a solventarlos de la mejor manera.

Git

Teniendo ya creado el grupo JavaLovers de la práctica anterior se clonaron ambos repositorios añadiendo los elementos necesarios para poder empezar con la práctica. A continuación nos posicionamos en develop usando git flow, una metodología que genera una estructura para las ramas y nos facilita el uso y movimiento entre estas. Debido a que ambos repositorios difieren desde este punto los explicaremos por separado.

- FrontEnd:

Al empezar se creó una feature para realizar tanto el XML como el DTD de ZonasBasicasSalud y de Zonas BasicasSalud_Mayores60. Una vez completadas las tareas y realizados los commits se cerró la feature integrándose en develop.

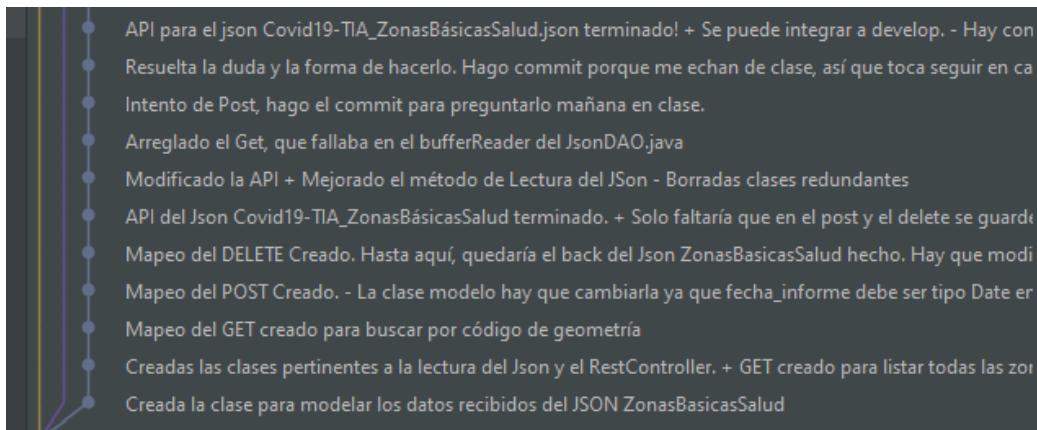


Se realizó una nueva feature para crear la pestaña 1 y obtener los requisitos requeridos en la práctica. En ella se realizó el grid, un form para añadir elementos al grid y la edición del grid. Una vez completadas las tareas y realizados los commits se cerró la feature integrándose en develop.

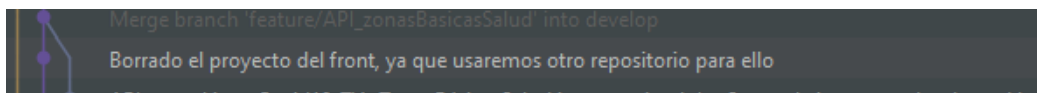
Se realizó una nueva feature para crear la pestaña 2 y obtener los requisitos requeridos en la práctica. En ella se realizó el grid y la edición de sus elementos a partir de un form. Una vez completadas las tareas y realizados los commits se cerró la feature integrándose en develop.

- Backend:

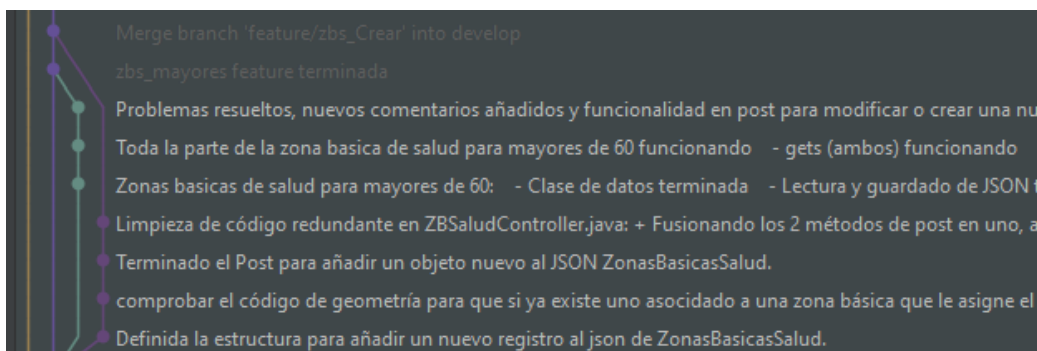
Al empezar se creó una feature para la modelización de los datos del JSON y su lectura, el mapeo del GET, POST y DELETE y la creación de API. Una vez completadas las tareas y realizados los commits se cerró la feature integrándose en develop.



Tuvimos que realizar un cambio en develop para eliminar la parte del front del repositorio porque decidimos moverla a uno nuevo.

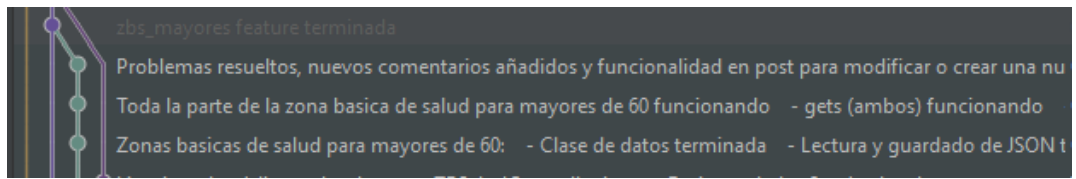


Se realizó una nueva feature para crear la parte del back correspondiente a la pestaña 1 utilizando ZonasBasicasSalud. Una vez completadas las tareas y realizados los commits se cerró la feature integrándose en develop.

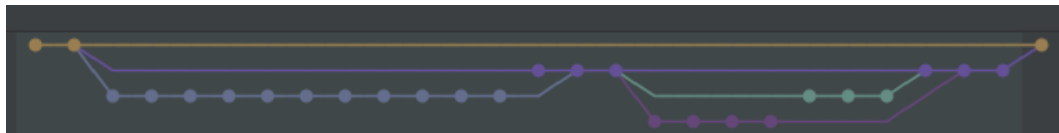


Se realizó una nueva feature para crear la parte del back correspondiente a la pestaña 2 utilizando ZonasBasicasSalud_Mayores60. Una vez

completadas las tareas y realizados los commits se cerró la feature integrándose en develop.



Al finalizar, el árbol del repositorio del backend fue:



3. Programa

Enunciado

El objetivo de la presente práctica es demostrar el conocimiento adquirido en la asignatura. Todos los temas vistos hasta el momento son necesarios para la realización de esta práctica: Control de versiones, Gestión de dependencias, TDD, pruebas unitarias, API's, Arquitecturas y Docker, etc.

La práctica está dividida en cuatro partes:

1. Definición y creación de los ficheros de datos: XML y DTD a partir de los ficheros JSON facilitados por los profesores.
2. Desarrollo de una API REST (backend), que será consumida desde la aplicación front creada en el siguiente apartado.
 - a. Unit testing de la parte back de la aplicación.
3. Desarrollo de la aplicación en Java usando Vaadin (Frontend)
4. Contenerización de las aplicaciones y comunicación entre ellas

XML y DTD

Realizamos la creación del XML y lo formateamos con el DTD. Este ejercicio nos sirve para saber cómo se utilizan y su relación con los archivos JSON.

Pestañas

Una vez ejecutada la aplicación se muestra una página con dos pestañas para navegar.

- Pestaña 1

Muestra un grid que contiene los datos de ZonaBasicaSalud. En grid tienes la opción de actualizar la lista para que vuelvan a aparecer los datos, y la opción de añadir un nuevo elemento o modificar un elemento existente de

la lista. Esto está soportado por un back en el que se encuentran los datos que utiliza el grid.

- Pestaña 2

Muestra un grid que contiene los datos de ZonasBasicasSalud_Mayores60. En grid tienes la opción de actualizar la lista para que vuelvan a aparecer los datos, y la opción de añadir un nuevo elemento a la lista. Esto está soportado por un back en el que se encuentran los datos que utiliza el grid.

Docker

Una vez instalado el docker desktop y estando ejecutado, creamos el ejecutable del proyecto y creamos el dockerfile dentro de IntelliJ. Una vez creado hacemos el build desde la terminal y ejecutamos la imagen para comprobar que funciona. Se realizaría de la misma manera en ambos repositorios.

4. Conclusión final

Consideramos que esta práctica es introducirnos a los conceptos que no habíamos llegado a utilizar hasta ahora, así como asentar las herramientas y programas utilizados en la práctica anterior. Mantenemos nuestra opinión con respecto a la primera práctica sobre grandeza de Git a la hora de realizar trabajos en los que participan varias personas, por su comodidad y la visibilidad que aporta a los participantes. Hemos podido comprobar lo útil que es Vaadin a la hora de realizar grids y formularios y conectarlos con el back. También hemos descubierto la importancia de los test unitarios para las pruebas de software, ya que es algo que habíamos estudiado de forma teórica, pero no habíamos aplicado como tal hasta ahora.

También ha sucedido algún problema con el que nos hemos topado que ha hecho que le tuviéramos que emplear más tiempo del que en un primer momento teníamos planeado, pero supimos solucionarlo completamente.

Creemos que con esta práctica hemos conseguido consolidar los conocimientos de los nuevos elementos de la asignatura (Vaadin, API, DTD, XML y test unitarios) y dominar los que obtuvimos en la práctica anterior (Git, Java y el entorno de IntelliJ).

5. Bibliografía

- *BibGuru*. (s/f). Bibguru.com. Recuperado el 16 de enero de 2023, de <https://app.bibguru.com/>
- *Casos de prueba: JUnit*. (s/f). Jtech.ua.es. Recuperado el 16 de enero de 2023, de <http://www.jtech.ua.es/j2ee/publico/lja-2012-13/sesion04-apuntes.html>
- *Declaración de atributos en una DTD*. (s/f). Abrirllave.com. Recuperado el 16 de enero de 2023, de <https://www.abrirllave.com/dtd/atributos.php>
- *Generating a DTD for the XML file*. (s/f). Oracle.com. Recuperado el 16 de enero de 2023, de <https://docs.oracle.com/cd/E19509-01/820-4380/agoip/index.html>
- Jimmy. (s/f). *Xml dtd*. Codebeautify.org. Recuperado el 16 de enero de 2023, de <https://codebeautify.org/xmlviewer/cb7b0fa1>
- *JSON to XML converter*. (s/f). Convertjson.com. Recuperado el 16 de enero de 2023, de <https://www.convertjson.com/json-to-xml.htm>
- *Vaadin 14 docs*. (s/f). Vaadin.com. Recuperado el 16 de enero de 2023, de <https://vaadin.com/docs/v14/>
- Jorge Cisneros y Nacho Serrano, curso 2022/2023, Documentación-ejercicios dados en clase y subidos en Canvas.