

POST-MORTEM: CICLO 3
Desarrollo de SW en equipo
Perapple

Integrantes:

Daniel Althviz - 201313031
Esteban Dalel - 201227078
Nicolas Galvis - 201313973
Harold Gonzalez - 201213646
Camilo Mendoza - 201218124
Maria Camila Remolina - 201217379

Profesor:

Jaime Chavarriaga

Universidad de los Andes
Bogotá, Colombia

Tabla de Contenidos

1. Evaluación de avance del proyecto

- 1.1. Producto Producido**
- 1.2. Esfuerzo Invertido**
- 1.3. Proceso de realización**

2. Evaluación de Roles

- 2.1. Líder de Desarrollo (Daniel Althviz)**
- 2.2. Líder (Esteban Dalel)**
- 2.3. Líder de Soporte (Nicolas Galvis)**
- 2.4. Líder de Interfaz (Harold Gonzalez)**
- 2.5. Líder de Calidad (Camilo Mendoza)**
- 2.6. Líder de Planeación (Maria Camila Remolina)**

3. Reporte de Ingeniero

- 3.1. Daniel Althviz**
- 3.2. Esteban Dalel**
- 3.3. Nicolas Galvis**
- 3.4. Harold Gonzalez**
- 3.5. Camilo Mendoza**
- 3.6. Maria Camila Remolina**

4. Retrospectiva Estrella

1. Evaluación de avance del proyecto (qué se hizo, qué debió haberse hecho - generación de metas):

1.1. Producto Producido

El producto final de este ciclo incluyó la mayoría de los requerimientos que se pidieron. El desarrollo de una página funcional fue posible gracias al uso de las herramientas de programación de JavaScript + AngularJS, Bootstrap y HTML. El manejo de servicios, controladores, mocks y mods está en su fase final y con su estructura definida. Luego de la implementación de los servicios REST por medio de JAXRS, y la comunicación de la vista con los mismos, se decidió la creación de pruebas unitarias, las cuales fueron realizadas en este ciclo junto con la lógica y la persistencia del sitio.

1.2. Esfuerzo Invertido

Como veníamos haciendo en en los ciclos anteriores, se manejaron las tareas mediante la herramienta Teamwork. Nuestra estimación de los tiempos requeridos por las tareas fue más cercana a lo que en realidad nos gastamos haciendo el trabajo. Para clases complejas y con mucho código la estimación se hace más difícil que para clases con menos métodos y con menor complejidad.

1.3. Proceso de realización

Se intentó mantener las reuniones bisemanales, sin embargo el semestre y responsabilidades externas al proyecto de los integrantes, hicieron cada vez menos fácil encontrar momentos en común. Los nuevos aspectos que se incluyeron en la aplicación (capa lógica, capa de persistencia y pruebas) requirieron entender primero su funcionamiento para luego implementarlos. Las pruebas fueron la parte más desafiante, aunque se logró avanzar considerablemente en este campo.

2. Evaluación de Roles:

2.1. Líder de Desarrollo (Daniel Althviz)

Se buscó, junto al grupo, establecer la distribución de la implementación de la lógica de la aplicación además de la implementación de las pruebas asociadas a dicha lógica. Partiendo de este objetivo, se realizó una reunión en la que se estableció quien se encargaría de que implementación teniendo en cuenta los módulos sobre los que

hemos estado trabajando. De igual forma se buscó realizar una revisión de cambios posibles a lo ya implementado, encontrando que la estructura de los elementos JSON entre el recurso REST y el front end debía de ser modificado. Sin embargo dicho cambio está siendo implementado. De igual forma, se estableció un intercambio de ideas para determinar las convenciones que se seguirían en cuanto a las peticiones http y la representación de los elementos (minimum, basic y full) de tal manera que por convención se pudiera determinar las traducciones adecuadas de la información para lograr evitar pérdida de información al persistir los cambios o modificaciones de los datos en la base de datos.

Así pues, continuando con la dinámica de los ciclos anteriores, dentro del grupo de trabajo se estableció que cada persona siguiera asignada a un módulo particular (Viajero, Itinerario o Ciudad, intentando que dicha distribución fuera similar a la del ciclo anterior), para que trabajara únicamente sobre esta unidad del proyecto, más aún dicha modularización no incluye únicamente la fachada de la aplicación, sino también los servicios REST antes mencionados y las pruebas e implementación de la lógica, buscando de esta forma evitar conflictos entre modificaciones, y todo esto enmarcado en las convenciones que se lograron establecer en la reunión de planeación para la distribución del trabajo y las convenciones de representaciones a utilizar.

Sin embargo, debido a dificultades propias de la implementación de lógica y sus pruebas, y la realización de las modificaciones pertinentes a lo que ya se encontraba implementado, no nos fue posible realmente realizar la implementación de nuevos requerimientos dentro de la aplicación.

Por otro lado, se buscó el asesoramiento o consulta del profesor en errores surgidos de la implementación de las pruebas para lograr determinar si la falla se encontraba en la implementación de las pruebas mismas o efectivamente en la lógica desarrollada.

2.2. Líder (Esteban Dalel)

En el ciclo 3 las reuniones fueron más informales. Cada integrante del grupo asume sus responsabilidades de manera adecuada. En trabajo de ingeniero, el uso de Teamwork como herramienta de organización mejoró mucho el flujo de trabajo y cada uno cumplió bastante bien sus responsabilidades.

En este caso, el grupo logró adelantarse a lo requerido por el cliente, por ende, se permitió un poco más de esparcimiento en grupo. Como tal, el grupo ya no es solo de trabajo. Ya no es necesario la mediación porque todos conocen a los demás y se asume que funciona.

2.3. Líder de Soporte (Nicolas Galvis)

Como líder de Soporte en el ciclo 3 no se presentó ningún inconveniente que pudiese afectar el desarrollo del proyecto. No obstante el nuevo manejo de plataformas como Jenkins es un aspecto importante para tener en cuenta si alguno de los miembros del grupo no logra entenderlo.

La principal tarea a realizar para el líder de soporte es verificar que todas las herramientas necesarias para el programa sirvan y si esto no sucede este líder debe de notificar a la persona que pueda ayudar a solucionarlo.

2.4. Líder de Interfaz (Harold Gonzalez)

En el ciclo 3 no hubo modificaciones en la interfaz gráfica, puesto que se trabajó sobre las pruebas unitarias y de integración. Así entonces como líder de diseño esta vez no cumplí función alguna, pero trabajé en el desarrollo de las pruebas.

Las tareas para el líder de interfaz en este ciclo tuvieron menor grado de importancia puesto que en la primera parte se intentó dejar definido un diseño permanente para todo curso del proyecto.

2.5. Líder de Calidad (Camilo Mendoza)

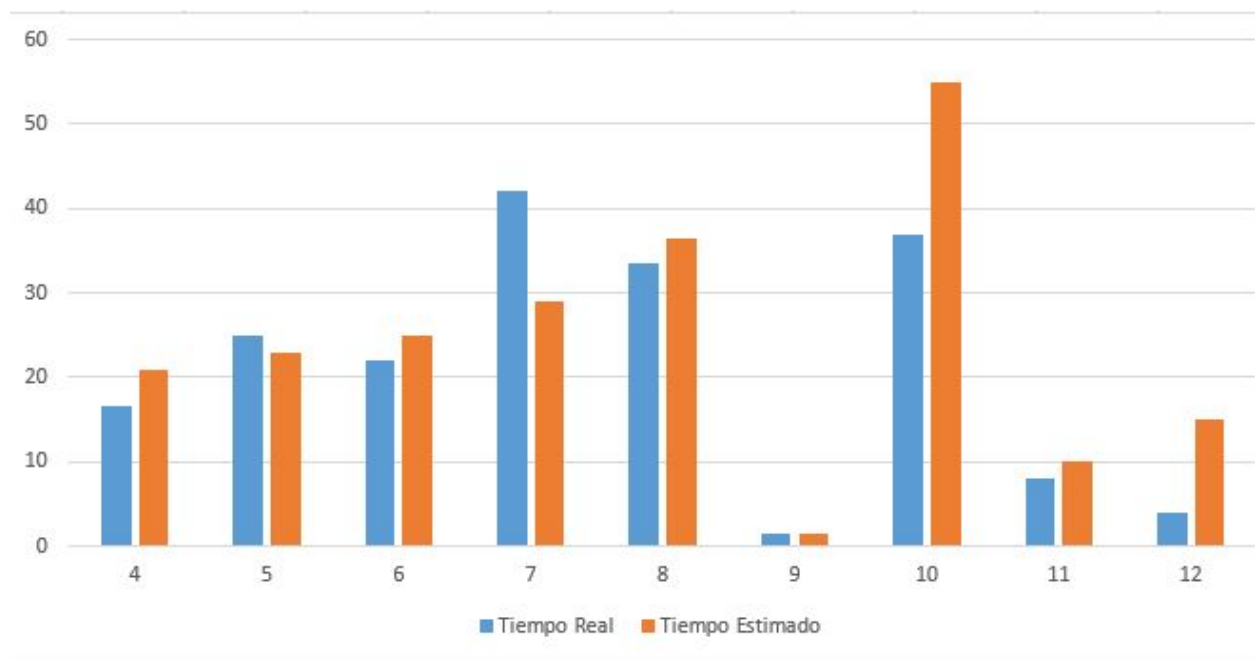
En el ciclo 3 seguimos con las reuniones semanales en las que acordamos las tareas y el procedimiento a seguir para lograr los objetivos de esta etapa y obtener un producto de calidad. Para este ciclo se acordó tener completamente claro los aspectos técnicos de las herramientas que se utilizarían en el desarrollo y la ejecución del proyecto. Este ciclo vio la inclusión de la diferenciación por capas, las pruebas y la integración continua. Esto hizo más complicado el proyecto, pero con la división por módulos y las diferentes reuniones que hicimos se nos hizo algo más fácil la implementación.

El Líder de Calidad requiere un constante seguimiento del trabajo de todo el equipo, por lo que la próxima persona que desempeñará este rol tiene que estar inspeccionando que los demás integrantes realicen sus tareas asignadas con los índices de calidad acordados, usando las herramientas de integración continua como Jenkins y SonarQube.

2.6. Líder de Planeación (Maria Camila Remolina)

Para la planeación semanal se sigue utilizando la herramienta teamwork. Cada vez se hace mucho más énfasis en registrar los tiempos de cada tarea con la aproximación más real posible. Se crearon las tareas de registrar tiempos esperados y reales, por recomendación del grupo, entonces es aún más difícil olvidar hacerlo. Ya todas las tareas están completamente desglosadas para que cada integrante sepa puntualmente su tarea. Esto ha ayudado mucho a ser más eficientes.

Se pueden ver a continuación los tiempos reales y esperados de cada semana. Cabe aclarar que la semana 9 corresponde a la semana de receso. También, que las semanas 10, 11 y 12 aún tienen tareas pendientes por terminar. Los tiempos reales ahí reflejados son sólo de tareas terminadas.



Por último, ya se cerraron todos los issues de GitHub y las únicas tareas presentes son las de teamwork. Aunque hay algunas tareas en pausa esto se debe a que se quiere probar completamente el producto existente antes de agregar nuevas funcionalidades.

3. Reporte de los ingenieros

3.1. Daniel Althviz

En cuanto al trabajo individual realizado, la diferencia de tiempo entre el estimado y el real fue en promedio baja. Sin embargo la distribución inicial de trabajo que se me fue asignado tuvo que ser redistribuida en los miembros del equipo debido a carga académica (falta de tiempo). Así pues, buscando no retrasar la implementación de los cambios a realizar en el ciclo dicha redistribución se buscó hacer en los primeros días del ciclo. Además, debido a que la realización de pruebas fue algo nuevo, en principio no era posible determinar fácilmente si los errores arrojados por las mismas se encontraban en la implementación de las pruebas mismas o efectivamente en los elementos que estaban siendo probados.

3.2. Esteban Dalel

Las pruebas mostraron no ser un reto con la entidad asignada, además de tener ayuda de los demás integrantes y el EjemploBook. Por diversos factores, GitHub se ha convertido más en una carga que en un apoyo, incluso con el sistema de aviso previo de en qué se está trabajando implementado por el grupo en Telegram. A pesar de esto, y con ayuda de Teamwork, se ha logrado un buen trabajo a final de ciclo, incluso con los problemas de tiempo de los integrantes del grupo. Cabe resaltar que no se ha hecho trabajo de legado, y el Back End no está conectado adecuadamente con el Front End.

3.3. Nicolas Galvis

El desarrollo de las pruebas unitarias fue una nueva experiencia de creación en este proyecto. El saber como deben de funcionar las cosas y poder guardarlas como en el ejemplo de persistencia hace que uno pueda entender más el funcionamiento del proyecto y por lo tanto poder corregirlo desde la raíz. A pesar de estar ausente una semana y media hábil por motivos académicos, el apoyo del grupo fue fundamental para ponerme en contexto y apoyarme en todas las dudas con respecto a mi trabajo.

3.4. Harold Gonzalez

En este ciclo trabajé sobre las entidades (entities) y la persistencia de la parte lógica de la aplicación, así como en las pruebas unitarias de la misma. Las pruebas en mi opinión son un trabajo tedioso, pero al fin de cuentas uno comprende la verdadera importancia de su realización, pues gracias a estas se puede verificar el correcto funcionamiento de

la aplicación de manera casi automática. Además las pruebas en el ámbito laboral juegan un papel demasiado importante pues son el sustento del trabajo realizado.

3.5. Camilo Mendoza

En esta etapa del proyecto se incluyeron nuevos aspectos como la diferenciación por capas (persistencia, lógica, web, etc), las pruebas (lógica, persistencia, integración) y el uso de herramientas para la integración continua (Jenkins, SonarQube). El manejo de las capas fue relativamente sencillo de manejar, y con el tiempo se logró manejar de manera ideal. Las pruebas fueron un reto pues fue difícil saber de dónde salían los problemas que se mostraban y si eran errores de código o de dependencias. Por último, Jenkins fue una herramienta que nos permitió controlar el estado de nuestra herramienta en el master branch, y a SonarQube es necesario dedicarle más tiempo para disminuir nuestra deuda técnica y resolver los problemas que nos muestra la herramienta.

3.6. Maria Camila Remolina

En este ciclo aprendí sobre la conexión a la base de datos y las pruebas. Al igual de medidores de calidad del código. Los primeros dos temas me parecieron un poco técnicos, pero fue muy agradable saber cómo funcionan. Me gustó el estilo de pruebas que manejamos, sin embargo sería también bueno explorar alguna otra forma. Por ejemplo, no hacer las pruebas secuenciales en integración.

Siento que sobre el tema de medidores de calidad aún tengo dudas y quiero ahondar un poco más en ellos para el siguiente ciclo. En particular, en saber usar correctamente Jenkins y SonarQube. Ésto para saber, de forma cualitativa, que deberíamos planear para resolver con prioridad. La razón de ésto es que siempre hacemos las tareas que como grupo nos parecen más urgentes, pero es una percepción algo subjetiva y aunque no errónea, sería correcto respaldarlo con datos estadísticos.

4. Retrospectiva de estrella



- **Integraciones deportivas:** Momentos en la semana en los que el grupo se reúne a jugar algún deporte (ej: fútbol o squash) para distensionar las cargas académicas de la semana de trabajo e integrarnos como equipo. De nuevo, el objetivo es no ser un grupo, ser un equipo.
- **Resolver issues de SonarQube:** Dedicar tareas para resolver los issues mostrados por la herramienta SonarQube y reducir la deuda técnica del proyecto
- **Utilizar Jenkins para integración continua:** Empezar a utilizar más la herramienta Jenkins para poder tener el seguimiento del estado de nuestro proyecto, incluyendo sus pruebas.
- **Comunicar cada push:** Usar el medio de comunicación del grupo para avisar cada vez que se realice un push y tener en cuenta hacer pull antes de trabajar. Igualmente la distribución de trabajos garantiza no tener que hacer merges.

- **Registrar tiempos estimados y reales en Teamwork:** Usar más y mejor la herramienta Teamwork, registrando los tiempos estimados y reales con frecuencia.
- **Reuniones presenciales:** Hemos probado que ahorramos más tiempo haciendo que las reuniones sean presenciales y con todo el equipo presente. Ya cambiamos el horario para hacer esto posible.
- **Discusiones grupales sobre el trabajo a realizar:** Se realizan discusiones sobre cómo abordar el trabajo que sigue y así cada integrante puede expresar sus ideas y plantear formas de resolver los requerimientos.
- **Dejar que se venzan las tareas sin completarlas:** Debemos respetar los tiempos acordados en la herramienta Teamwork y así lograr un desarrollo más fluido del proyecto.
- **Empezar tareas tarde:** Hay que empezar las tareas lo más temprano posible para poder comunicar con el grupo, corregir errores y realizar pruebas a tiempo.