

Universidad ORT Uruguay
Facultad de Ingeniería

Ingeniería de software ágil 2

Documentación de la entrega 1

<https://github.com/IngSoft-ISA2-2023-2/obligatorio-grupo11-bandera-caffarena-gonzalez>

Sebastián Bandera-220417
Martín Caffarena-230351
Nicolas Gonzalez-231843



Entregado como requisito de la materia Ingeniería de software ágil 2

Índice

Índice	2
Kanban	3
Definición del proceso de ingeniería en el contexto de KANBAN	3
Designación de roles	4
Tablero	4
Explicación del tablero y su vínculo con el proceso de ingeniería	5
Creación y posterior mantenimiento del repositorio	5
Análisis de deuda técnica (bugs)	6
Informe de avance de la etapa	6
Registro de las actividades realizadas (fecha, horas, integrante).	7
Dificultades encontradas y formas de solución.	8
Lecciones aprendidas y mejoras en el proceso	9
Video de retrospectiva	10
Tablero DAKI	10

Kanban

Se definió como proceso de ingeniería para llevar a cabo la primera iteración del proyecto la metodología ágil llamada Kanban.

Kanban indica que las tareas que se van desarrollando o hay que desarrollar posteriormente deben ser visibles para todo el equipo de trabajo por lo que se creó el tablero Kanban, el cual se describió en su respectiva sección. A lo largo del proyecto son agregadas nuevas tareas en este tablero accesible para todo el equipo y estas son realizadas por el equipo a medida que se avanza en el proyecto.

Las nuevas tarjetas son creadas en “Backlog” y cuando estas pasan a ser parte de la iteración se colocan en “To Do” para indicar que deben ser realizadas. Cuando alguien las toma para trabajar, pasan a la columna “In Progress”, para finalmente terminar en la columna “Done” al finalizar.

Definición del proceso de ingeniería en el contexto de KANBAN

Las etapas del proceso en esta iteración son planificación, análisis de código y test.

La etapa de planificación se realizará por el SM con apoyo del resto de integrantes y generará las tarjetas en el tablero Kanban, columna “backlog”, luego de revisar los requisitos para la iteración.

La etapa de análisis de código la realizará el rol de desarrollador que tiene entendimiento sobre el código y genera el registro de las issues en GitHub, las cuales a su vez generan una tarjeta en el Backlog. Este análisis se realiza con una herramienta para optimizar los tiempos de trabajo

La etapa de Test la realiza el tester ejecutando la aplicación y comparando el comportamiento contra lo especificado en la letra original. También produce issues y tarjetas asociadas en el Backlog. También se usa test exploratorio para buscar errores generales como permitir cantidades negativas, precios negativos.

Lo anterior se puede resumir en la siguiente tabla:

Actividad	Rol	Artefacto	Método
Planificación	SM	Tarjetas en tablero	Lectura requisitos/letra
Análisis de código	Desarrollador	Issues, tarjetas	Uso de herramientas de análisis
Test	Tester	Issues, tarjetas	Test exploratorio, Comparación con requerimientos

Designación de roles

Para esta primera entrega se decidió la utilización de 3 roles; Product Owner (PO), Simil Scrum Master (SM); Desarrollador; Tester. De momento se tomó la decisión de solo utilizar dichos roles pero tal vez luego el equipo vea posibilidades de mejoras y decida tanto agregar nuevos, como modificar los ya existentes.

Se decidió a su vez que el “PO” sea Martín ya que este conocía previamente el producto, y se designó a Sebastián como “SM”. Sin embargo estos roles se irán cambiando de estudiante para que todos puedan generar una experiencia en cada uno de ellos.

Todos los estudiantes del grupo también serán de forma paralela desarrolladores y testers, siempre tener en cuenta que esto es solo algo realizado para este trabajo, ya que si se tratara de un proyecto verdadero cada uno tendrá un único rol, pero sería necesario más miembros en el equipo.

Tablero

El tablero Kanban fue implementado dentro de Github, donde se tiene el repositorio del proyecto, usando la herramienta de Github projects. Si bien hay otras herramientas para poder implementar y utilizar el tablero Kanban se decidió utilizar Github projects y así mantener todos los elementos relacionados al avance del proyecto dentro del mismo repositorio con mayor facilidad. A su vez al utilizar esta herramienta tenemos mayor conectividad con el proyecto, lo que nos permite linkear las tarjetas del tablero a los issues correspondientes y así aportar mayor detalle a las tareas.

El tablero se compone por un product backlog, en esta comenzaran todas las tarjetas del tablero y este es el conjunto completo de tarjetas del tablero las cuales luego se irán implementando, las columnas “To Do”, en esta se encontraran todas las tarjetas que se deben realizar en la iteración y estan todavia sin ser comenzadas, “In Progress”, en esta se encuentran todas las tarjetas correspondientes a las tareas que se están realizando en el momento, y “Done”, donde se ubican aquellas tareas las cuales se consideran finalizadas.

Cada tarjeta tendrá una asociación con los usuarios que la trabajaron o la trabajarán. Para los casos donde una tarjeta necesita ser trabajada por todo el equipo, por ejemplo “Restauración de BD y conexión con backend”, además de asignar a la tarjeta a todo el equipo, se coloca en la descripción un checkbox por cada integrante. De este modo cuando un integrante completa esta tarea, marca su check y coloca una descripción breve de cómo la resolvió, de modo que sirva como guía para el resto de integrantes de cómo se puede completar. Además ayuda al que la resolvió primero a ver otras posibles vías de resolución si se dieron.

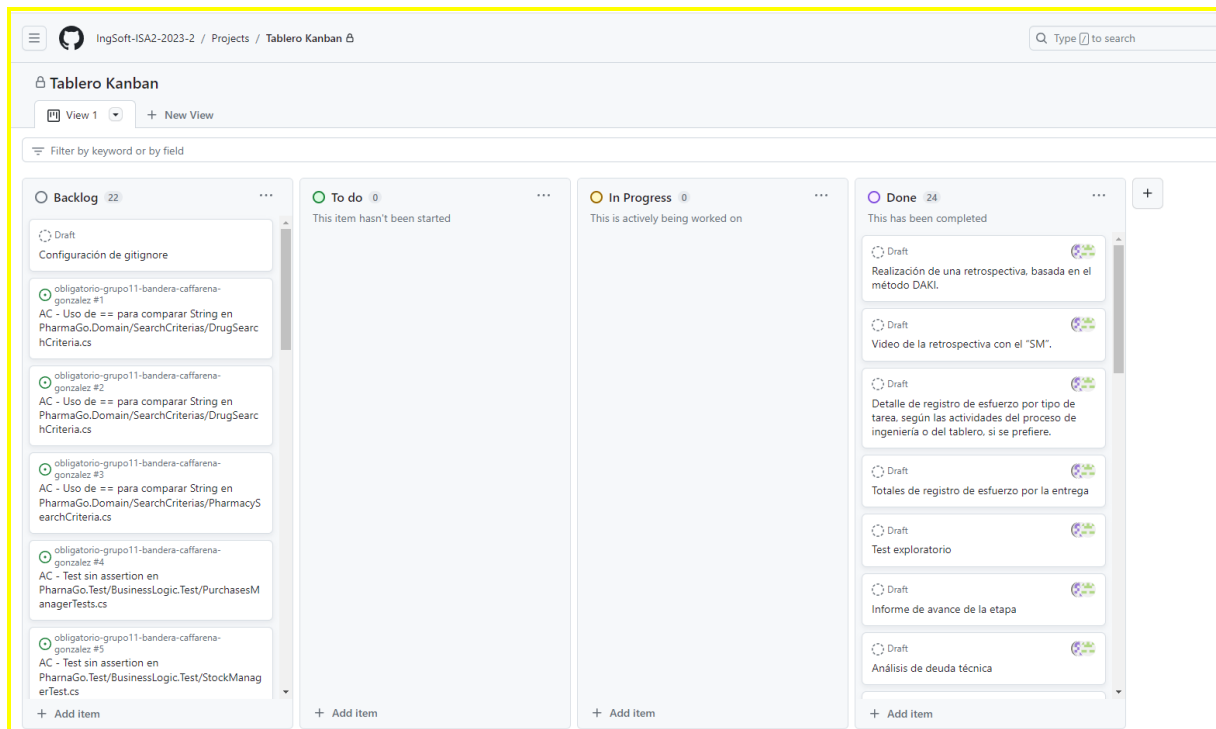


Foto final del tablero Entrega 1

Explicación del tablero y su vínculo con el proceso de ingeniería

El tablero en esta iteración tiene 4 columnas, Backlog, To Do, In Progress y Done.

La etapa de planificación genera las tarjetas en Backlog, y quedan a la espera que esta misma planificación las apruebe para su avance a "To Do". En caso de avanzar a "To Do", la tarjeta pasará a la actividad "Análisis de código" o "Test" según corresponda. Para finalmente alojarse en "Done" una vez finalice.

Creación y posterior mantenimiento del repositorio

El repositorio tiene una carpeta "Código", en la cual se encuentra todo el código correspondiente a la aplicación separado por dos subcarpetas, una carpeta llamada "Frontend" la cual se compone por el proyecto de Angular y otra carpeta llamada "Backend" que contiene la solución .NET y las pruebas unitarias.

Otra carpeta "Base de Datos" en la cual se encuentra una base de datos de prueba y otra vacía.

Por último una carpeta "Documentación" donde se encuentra la documentación de cada etapa en conjunto con la documentación general de la aplicación. En la subcarpeta "Original" se encuentra la documentación entregada por los autores originales del obligatorio. Así como la letra del obligatorio original.

Definimos un formato para los commits el cual se basa en poner como título del commit que acción se realizó (ADD, DELETE, FIX, UPDATE) seguido de una pequeña

descripción de la acción en sí (Esto puede ser sobre que archivo fue acción, que se intentó con la acción, entre otros.).

En resumen el repositorio en GitHub contiene:

- Proyecto del Backend
 - Se irá actualizando a medida que se solucionen los issues
- Proyecto del Frontend
 - Se irá actualizando a medida que se solucionen los issues
- Base de datos para restaurar
 - Se irá actualizando a medida que se solucionen los issues
 - Pueden cambiar el DDL de la base
- Documentación original
- Letra original
- Documentación de las entregas
 - Se irá actualizando a medida que se realicen entregas

Cada integrante necesita realizar los siguientes pasos:

1. Clonar el repositorio
2. Restaurar la base de datos
3. Abrir el proyecto backend C#
4. Abrir el proyecto frontend con Angular. Puede requerir npm install
5. Levantar los dos proyectos y revisar que no existan problemas de conectividad/puertos.

Análisis de deuda técnica (bugs)

Para esta primera entrega el equipo realizó un análisis técnico de las diferentes partes del proyecto, de esta forma tener una idea sobre la deuda técnica del mismo. Es decir que el equipo pueda saber a grandes rasgos los bugs/issues que deberán ser resueltos por el equipo de desarrolladores.

La herramienta utilizada fue SonarQube que además de analizar el código fuente de ambas aplicaciones, permitirá comparar el estado actual del programa con el estado futuro luego de las correcciones e incorporaciones nuevas de funcionalidades.

Informe de avance de la etapa

Para esta etapa se comenzó con el proyecto, determinando metodologías de trabajo, tablero a utilizar, roles y su designación, a su vez se realizó el análisis de deuda técnica.

Resumen de issues generados

- Issues totales: 21
- Issues de tipo análisis de código: 16
 - Gravedad media: 4
 - Gravedad leve: 12
- Issues de tipo bug: 5
 - Severidad mayor: 3
 - Severidad menor: 2

Resumen de tarjetas generadas

- 46 tarjetas en total
- 21 corresponden a issues
- 22 tarjetas en el backlog
- 24 tarjetas finalizadas para la entrega 1.

Registro de las actividades realizadas (fecha, horas, integrante).

Para el análisis de deuda técnica se realizaron dos actividades principales, el análisis de código y el test, mediante el uso de herramientas de análisis y test exploratorios respectivamente.

***Se utilizó como período de tiempo mínimo, 30 minutos = 0,5 horas**

Fecha	Concepto	Horas Bandera	Horas Caffarena	Horas Gonzalez	Total esfuerzo	Total día
9/09/2023	Instalación de herramientas faltantes	0,5			0,5	
9/09/2023	Restauración de BD y conexión con backend	1,0			1,0	
9/09/2023	Backend levantado con conexión a la bd	1,0			1,0	
9/09/2023	Armado y comunicación de grupo GitHub y Teams	0,5	0,5	0,5	1,5	4,0
11/09/2023	Creación tablero inicial	0,5	0,5	0,5	1,5	
11/09/2023	Inicio Documento Entrega 1	0,5	0,5	0,5	1,5	
11/09/2023	Designación de roles	0,5	0,5	0,5	1,5	
11/09/2023	Creación de estructura de carpetas	0,5	0,5	0,5	1,5	
11/09/2023	Definición de estándar para creación de commits	0,5	0,5	0,5	1,5	7,5
15/09/2023	Reunión	2,0	2,0	2,0	6,0	
15/09/2023	Agregado al repositorio el código inicial (Front y back) + base foto final (Entrega 2)	0,0	1,0	0,0	1,0	
15/09/2023	Explicación del tablero y su vínculo con el proceso de ingeniería	0,0	0,0	0,5	0,5	
15/09/2023	Tarjetas en Board Git	1,5	0,0	0,0	1,5	
15/09/2023	Explicación designación de roles	0,0	0,5		0,5	
15/09/2023	Incluir en la Documentación: Creación y posterior mantenimiento	0,0	0,0	0,5	0,5	

	del repositorio: elementos que contiene y cómo los van a versionar					
15/09/2023	Investigar herramientas para análisis de código	1,0	0,0	0,0	1,0	11,0
16/09/2023	Análisis de deuda técnica	3,5	0,0	0,0	3,5	
16/09/2023	Compilacion y ejecucion del proyecto en conjunto con la BD	0,0	1,0	1,0	2,0	5,5
17/09/2023	Análisis de deuda técnica	1,0	0,0	0,0	1,0	
17/09/2023	Ajustes documentación	3,0	0,0	0,0	3,0	
18/09/2023	Dificultades encontradas y formas de solucionarlo	0,0	0,5	0,0	0,5	
18/09/2023	Lecciones aprendidas y mejoras del proceso	0,0	0,0	0,5	0,5	
18/09/2023	Reunión final	1,0	1,0	1,0	3,0	
18/09/2023	Retro	0,5	0,5	0,5	1,5	5,5
		0,0	0,0	0,0		33,5

Total: 33,5 horas personas.

Dificultades encontradas y formas de solución.

Uso de nuevas herramientas y uso tablero

Una de las primeras complicaciones que se encontró el equipo fue el uso del tablero, ya que una parte del equipo no tenía experiencia con el uso de tableros de trabajo en GitHub. La manera en la que se sobrepasó dicha dificultad, fue gracias a Sebastian quien si tenía se encargó de comunicarle y enseñarle al resto del equipo el funcionamiento del mismo.

Por otro lado el equipo decidió utilizar la herramienta SonarQube para el análisis de código, la cual no había sido utilizada antes por ningún miembro del equipo y al principio llevó a algunas complicaciones y a un análisis más lento pero con la práctica de uso del mismo esto fue ralentizando el equipo cada vez menos.

Compilación del proyecto y conexión a la base de datos

Otra gran dificultad que se encontró el equipo fue la compilación del proyecto y la conexión a la base de datos, ya que si bien dos miembros del equipo ya contaban con experiencia en el proyecto, tuvieron dificultades para ejecutar el mismo debido a diferencias con el uso de la tecnología conocida por estos. A su vez otro miembro del equipo tuvo dificultades con la conexión a la base de datos, pero esto fue solucionado con una más profunda investigación de la herramienta y fue documentada la solución en su respectivo issue/incidencia.

Lecciones aprendidas y mejoras en el proceso

En esta sección detallaremos algunos aprendizajes que el equipo considero de gran valor a lo largo de esta iteración.

Manejo del tablero

Para esta primera iteración se creó el tablero kanban y a su vez se empezó a utilizar el mismo, con esto todos los integrantes del grupo se familiarizaron con el mismo y pudieron ver en la práctica como se maneja este y también nos ayudó a tener un orden de trabajo con mayor transparencia sabiendo siempre qué tareas estaban pendientes y en que se estaba trabajando en todo momento. Por lo tanto rescatamos como aprendizaje dentro de kanban el manejo del tablero, tanto en la creación y mantenimiento del mismo, como también el utilizarlo y mantener un orden a la hora de trabajar.

Registro de esfuerzo

Para poder registrar el esfuerzo tanto de cada uno como general del equipo en esta primera etapa decidimos utilizar una planilla de excel, donde cada integrante luego de desarrollar una tarea debía establecer el esfuerzo que esta tuvo y así poder tener estos datos actualizados en todo momento. Si bien hay herramientas para llevar un registro de esto el equipo decidió utilizar la manera más fácil, rápida y útil y nos pareció que con excel se podría realizar perfectamente. Por lo tanto el equipo teniendo la organización previamente mencionada aprendió a tener un registro del esfuerzo de cada tarea y poder así tener una noción de las horas/persona de cada integrante y grupales de las tareas.

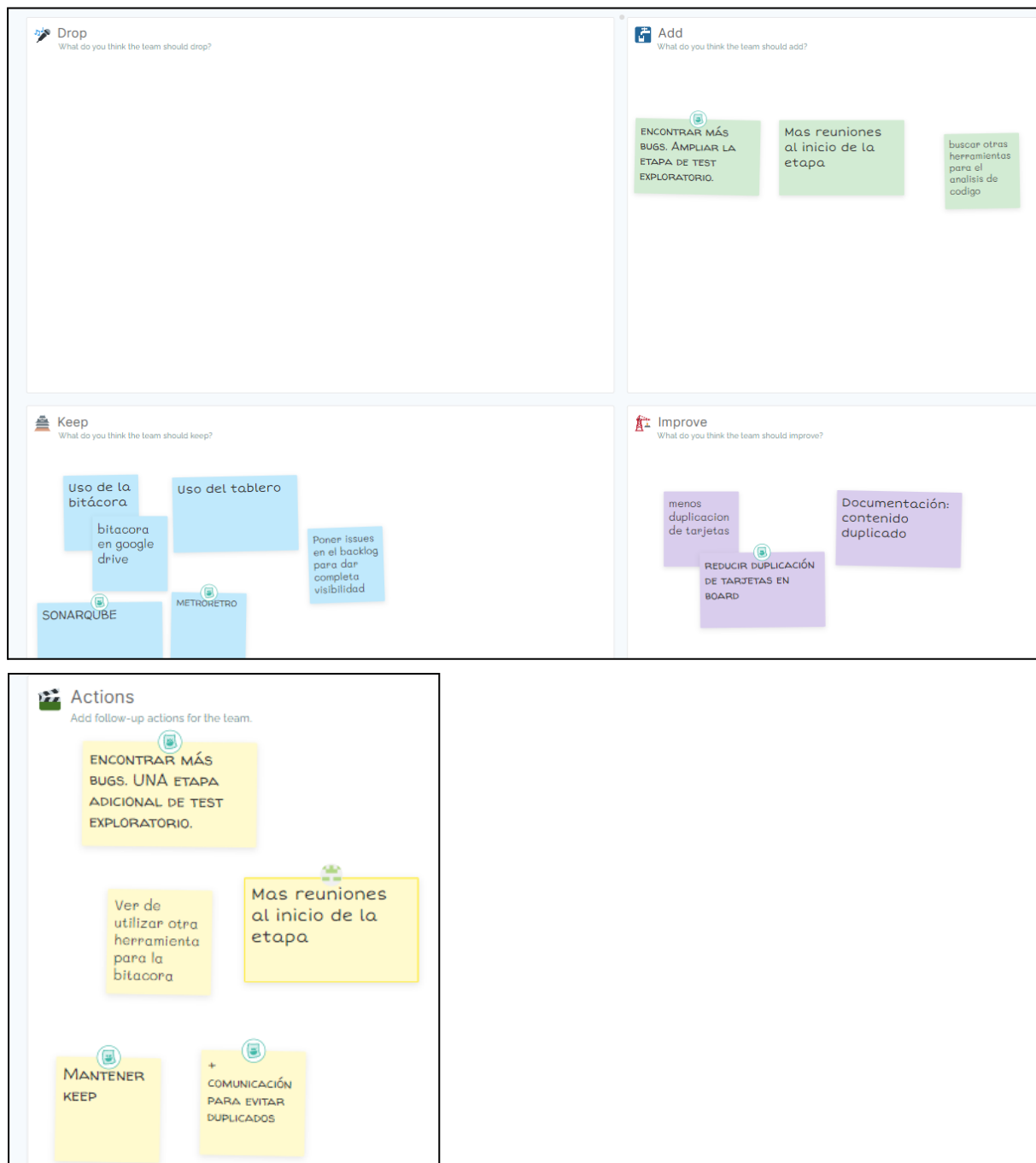
Video de retrospectiva

El vídeo de la retrospectiva se puede encontrar en el repositorio en GitHub, rama main, directorio “Documentación/Entrega 1 - ISA2”.

El link directo es:

<https://github.com/IngSoft-ISA2-2023-2/obligatorio-grupo11-bandera-caffarena-gonzalez/blob/1f6108e681ac20b0da7a9439b14f3f161527363f/Documentaci%C3%B3n/Entrega%201%20-%20ISA2/Retro1.mkv>

Tablero DAKI



Para realizar la retrospectiva decidimos realizar la técnica de DAKI (Drop, Add, Keep, Improve) la cual consiste en establecer 4 sectores, el primero "DROP" refiere a que dejar de hacer, "ADD" que herramienta o proceso se puede agregar, "KEEP" que se realizó de manera exitosa y por lo tanto se debería seguir haciendo y "IMPROVE" que de lo realizado se debería seguir haciendo pero de mejor manera.

En conclusión a rasgos generales el equipo consideró que se trabajó con un flujo de trabajo bastante organizado y se hizo un buen uso de las herramientas, esto se vio reflejado en la retrospectiva viendo que no se colocaron aspectos en "DROP". También consideramos que algunas herramientas podrían tener una alternativa la cual sea una mejor opción, pero de igual manera se destacó el uso de las mismas y como se trabajó con ellas. Como aspectos a mejorar habría que tener una mejor comunicación, ya que se ve como tanto en las tarjetas del tablero como en la documentación se vio duplicación de información. Por último a raíz de los distintos aspectos vistos en DAKI surgieron algunas acciones que consideramos serían importantes para la siguiente etapa, estas son tener mayor comunicación dentro del equipo, capaz poder encontrar algún bug más realizando más tests exploratorios, tener más reuniones al inicio para poder organizar bien el trabajo al inicio y ver si sería una mejora encontrar alguna otra herramienta para hacer uso de la bitácora.