

Universidad ORT Uruguay
Facultad de Ingeniería

Ingeniería de software ágil 2

Documentación de la entrega 3

<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

Designación de roles	3
Proceso de ingeniería	3
Tablero KANBAN	4
Informe de avance de la etapa	5
Dificultades encontradas y formas de solución.	5
Desarrollar usando SpecFlow y BDD	5
Trabajar con proyecto desconocido	5
Problemas entre backEnd y FrontEnd	6
Nuevos aprendizajes	6
Uso de SpecFlow y BDD	6
Cerrar Issues en GitHub	6
Registro de las actividades realizadas (fecha, horas, integrante)	7
Evidencia de ejecución de casos de prueba	8
Video de la revisión de las nuevas funcionalidades con el PO	11
Video de retrospectiva	11
Tablero DAKI	11

Designación de roles

De igual forma que en la etapa anterior comenzamos designando los distintos roles dentro del equipo, rotando así las responsabilidades de cada uno y permitiéndole a todos los miembros del equipo aprender desde distintas perspectivas. Para esta etapa se determinó que Nico sea el (SM), y Martin el (PO). Recordar que a su vez todos los miembros del equipo son Desarrolladores y Testers. Sin embargo en esta ocasión ya que hubo tanto trabajo de desarrollo por razones de organización y facilidades para ciertos aspectos del proyecto de cada miembro se decidió hacer algunas especificaciones dentro del rol desarrolladores. Se distinguió entre desarrollador backend y desarrollador frontend. En esta ocasión se designó a Nicolas Gonzalez como único desarrollador frontEnd y a Martin y Sebastian como desarrolladores de backend. Decidimos hacer diferencia entre los desarrolladores de esta manera por facilidad en el manejo de las herramientas (Nicolás tiene más experiencia con el desarrollo FrontEnd y conoce mejor la herramienta) y así optimizar los tiempos. Sin embargo tanto Nicolás como Martín y Sebastián tuvieron contacto con ambas herramientas gracias a pequeños cambios sobre el final de la etapa.

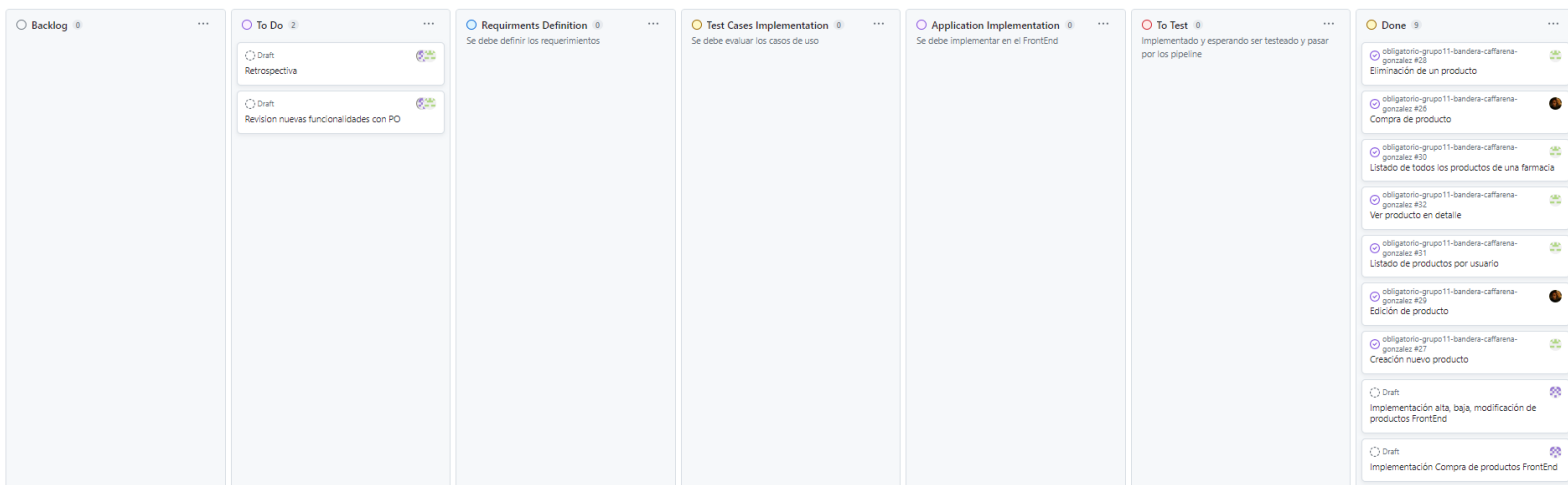
Proceso de ingeniería

Para la implementación de las nuevas funcionalidades se utilizó la metodología de BDD, la cual el desarrollo de las mismas son guiados por los comportamientos y para hacer uso de la misma se utilizó la herramienta de SpecFlow, en la cual se definió la narrativa de las historias de usuario, en nuestro caso cada funcionalidad tenía su propia historia de usuario porque nos pareció que quedaba con mayor claridad si se manejaba de esta manera, y también los escenarios de las mismas. Dentro de BDD identificamos la fase RED, que al igual que en TDD corresponde a la creación de los casos de prueba, en este caso la narrativa con sus escenarios, pero sin implementar la misma, luego la fase GREEN donde se implementa el código para que pasen todos los escenarios dentro de las pruebas de la historia de usuario correspondiente y por último la fase de REFACTOR, la cual no siempre está presente pero corresponde a hacer cambios sobre el código y las pruebas para mejorar estos.

Para el manejo del repositorio utilizamos el método de trunk-based, definimos que al final de cada momento/día donde los desarrolladores hubiesen realizado cambios sobre el proyecto y además estos estén funcionales, se trata de no “pushear” a main cualquier cambio que no se encuentre estable para así no perjudicar al resto de desarrolladores, los mismos pushean todo lo realizado al repositorio y describen los cambios. A pesar de que siempre se trató de mantener este flujo de trabajo en algunas instancias hubo que “pushear” algunos cambios antes de terminar de trabajar, porque se estaba trabajando en paralelo entre los desarrolladores y en algunas circunstancias se precisaba tener un cambio en el cual estaba trabajando otro desarrollador.

La pipeline utilizada se mantuvo debido a que ya es capaz de ejecutar todas las pruebas incluidas las de SpecFlow. Además está optimizada para sólo ejecutar la pipeline de Front o Back según corresponda revisando los archivos modificados del commit.

Tablero KANBAN



Este tablero se encuentra en Github en el apartado de “Projects” con el nombre de “Tablero Kanban Entrega 3”. En resumen se decidió tener 7 columnas considerando que se trabajaría con BDD. Como en las entregas anteriores se mantuvo el backlog, en esta columna se encuentra inicialmente todo el trabajo del proyecto y se van derivando las tarjetas a la columna de “To Do”, dependiendo si la misma se debe realizar en esta etapa o no. Luego definimos la columna de “Requirements Definition”, en esta columna se encuentran los issues los cuales se están definiendo sus requerimientos, es decir se están generando las narrativas de los mismos. Luego tenemos la columna de “TestCases Implementation”, se encuentran aquellas tarjetas de las historias de usuario las cuales están siendo escritos los escenarios. Tanto los requerimientos como los escenarios se vieron implementados en la herramienta de SpecFlow mencionada anteriormente. Estas dos últimas columnas corresponden a la fase RED de BDD. Luego tenemos la columna de “Application Implementation”, en la cual luego de haber definido los requerimientos y además los escenarios se encuentra en fase RED el issue esperando ser implementado en la aplicación pasando a fase GREEN, solo contemplamos el backend luego de la fase RED porque el frontend debería ya estar implementado y se generaron tarjetas aparte para realizar el mismo. Por último mantuvimos de las entregas anteriores la columna “To Test” donde ya implementado el issue se testea funcionalmente y se espera ver si el mismo pasa las pruebas del pipeline y la columna “Done”, donde se encuentran los issue que se consideran terminados, con terminados nos referimos a que el issue está implementado y cumple con los escenarios y criterios de aceptación previamente definidos tanto del issue mismo como de sus subtareas.

Informe de avance de la etapa

Para esta etapa se comenzó primero que nada con la designación de roles como ya se había repetido en etapas anteriores, de esta manera ya establecer una estructura clara dentro del equipo desde un inicio. Como ya mencionamos en esta ocasión se distinguió entre desarrolladores frontend y Backend.

Luego se hizo una definición de todas las tareas/issues que debían ser realizadas para esta entrega en el nuevo tablero del cual hablaremos más adelante. A su vez se asignó a cada miembro las tareas de las cual cada uno era responsable, teniendo en cuenta sus roles, como desarrollador frontend o backend, etc. Aclarar que en este momento fue cuando se definieron los diferentes escenarios los cuales debían de cumplir cada funcionalidad, esto con el fin de que luego el desarrollo de cada funcionalidad con el uso de SpecFlow y BDD sea más sencillo, claro y estructurado, cosa que vinimos que es muy importante al estar trabajando con BDD.

En un inicio el plan del equipo era primero realizar la parte del desarrollo de frontEnd para luego pasar al backend. Sin embargo por temas de tiempo y habilidades de cada integrante se decidió por dividir entre el equipo las tareas y comenzar con la parte de backend por más que algunas partes del front fueron terminadas más adelante, una vez que el backend ya estaba terminado. Aclara a su vez que a finales del proceso de desarrollo el equipo realizó varias pruebas exhaustivas de las nuevas funcionalidades para corroborar su correcta funcionalidad y a su vez se realizaron pruebas generales de la aplicación, incluso de partes anteriores, ya que el equipo quería asegurarse de que los nuevos cambios no reperquiesen negativamente en el resto de la aplicación. Por esta razón una vez terminado este proceso se debieron de realizar algunos cambios menores referenciados como BugFix para solucionar/mejorar algunos problemas que se encontraron en esta etapa de algunas funcionalidades, tanto nuevas como anteriores funcionalidades.

Algo que se puede notar si hacemos un test de cobertura del código, vemos que este bajo ligeramente, esto se debe a que se agregaron archivos al proyecto los cuales son tenidos en cuenta para dicha cobertura pero no entraba dentro de esta entrega el realizar ningún test sobre estos, como por ejemplo la parte de la WebApi de Productos, por esta razón vemos esta baja en los niveles de cobertura de código del proyecto.

Dificultades encontradas y formas de solución.

Desarrollar usando SpecFlow y BDD

Ya dando los primeros pasos dentro de esta etapa nos encontramos con nuestra primera dificultad, es uso de SpecFlow y BDD para el desarrollo de las nuevas funcionalidades. Si bien el equipo ya sabía sobre estos temas por lo visto en clase, no teníamos mucha experiencia con la aplicación de las mismas en el desarrollo de un proyecto, por lo que tuvimos un inicio lento hasta que el equipo logró un mayor entendimiento de las herramientas y se acostumbro a trabajar de esta manera

Trabajar con proyecto desconocido

Por otro lado otra dificultad que se encontró el equipo a lo largo de la etapa fue el estar trabajando /desarrollando con un proyecto el cual no fue desarrollado por los miembros del equipo. Si bien ya habíamos realizado algún pequeño cambio en el código y este había sido analizado en previas etapas, era la primera vez que el equipo debía de hacer cambios de

gran magnitud a nivel de estructura y código. Por esta razón cada que se deseaba implementar una nueva funcionalidad era necesario una breve investigación previa para entender mejor cómo cada cambio repercutirá en las partes ya implementadas y cual sería la mejor manera de lograr lo deseado.

Problemas entre backEnd y FrontEnd

Finalmente otro problema que se encontró el equipo, si bien no fue muy grande, vale la pena destacar que en una ocasión se dio que se commiteo un código/cambio en el backend que si bien corría las pruebas rompió otra parte del front end, la visualización de productos para ser exactos. Lo que llevó a que ambos desarrolladores se pongan en contacto para solucionar dicho error, esto se dio debido a que por tiempos se estaba desarrollando la parte de front y back de una misma funcionalidad al mismo tiempo. Por lo que para las siguientes funcionalidades se tuvo extremo cuidado y se adoptaron medidas como el no trabajar dos desarrolladores en una misma funcionalidad a la vez para evitar casos como este.

Nuevos aprendizajes

Uso de SpecFlow y BDD

Algo que debemos de destacar como nuevo aprendizaje fue el uso de la herramienta SpecFlow y la metodología de trabajo de BDD. Como ya mencionamos, si bien el equipo poseía algunos conocimientos sobre estos temas por lo visto en clase, tras haber realizado esta etapa podemos confirmar que el entendimiento de parte del equipo de estos dos aspectos mejoró de gran forma. De forma que el equipo ya se siente mucho más cómodo con el uso de esta herramienta y forma de trabajo y en caso de ser necesario en futuros proyectos los miembros del equipo ya no contarán con grandes dificultades para el uso de las mismas.

Cerrar Issues en GitHub

Por otro lado el equipo también se dio cuenta que, en la herramienta de GitHub en la visualización del tablero del proyecto donde manejamos todas las tareas que se debían de hacer para cada entrega, se pueden “cerrar” los issues, hasta ahora el equipo solo los pasaba a la columna de DONE. Sin embargo descubrimos para esta entrega que la herramienta ofrece una opción para cerrar los issues y que estos queden como completados.

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

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

Fecha	Concepto	Horas Bandera	Horas Caffarena	Horas Gonzalez	Total esfuerzo	Total día
20/10/2023	Reunión	2,0	2,0	2,0	6,0	
20/10/2023	Creación nuevo producto	0,0	5,0	0,0	5,0	
20/10/2023	Creacion y modificacion nuevo producto FrontEnd	0,0	0,0	3,5	3,5	14,5
21/10/2023	Escenarios comprar producto	0,5	0,0	0,0	0,5	
21/10/2023	BDD compra productos	3,0	0,0	0,0	3,0	
21/10/2023	Eliminación producto	0,0	3,0	0,0	3,0	
21/10/2023	Cambios de dominio	0,0	1,5	0,0	1,5	8,0
22/10/2023	Listados de producto	0,0	5,5	0,0	5,5	
22/10/2023	Eliminación producto y compra con productos FrontEnd	0,0	0,0	5,5	5,5	
22/10/2023	Edición de producto	2,0	0,0	0,0	2,0	
22/10/2023	Ajustes de integración	1,0	0,0	0,0	1,0	14,0
23/10/2023	Pruebas de funcionalidades	0,0	1,0	1,0	2,0	
23/10/2023	Detalles de funcionalidades	0,0	1,0	1,0	2,0	
23/10/2023	Reunión para desarrollo final de la documentación	1,5	1,5	1,5	4,5	
23/10/2023	Grabación revisión de nuevas funcionalidades con PO	0,5	0,5	0,5	1,5	
23/10/2023	Grabación Retro	0,5	0,5	0,5	1,5	11,5
	Total Entrega 3	0,0	0,0	0,0	0,0	48

Total: 48 horas/persona de trabajo.

Evidencia de ejecución de casos de prueba

Se puede visualizar la LivingDoc completa en “Documentación/Entrega 3 - ISA2/BDD - LivingDoc/LivingDoc.html”

Feature: newProduct

+

—

Test results

▼ SpecFlowProject

1 Passed 0 Failed 0 Others

▼ Features

1 Passed 0 Failed 0 Others

▼ newProduct

Correcta creación de un nuevo producto

Incorrecta creación de un nuevo producto por codigo largo

Incorrecta creación de un nuevo producto por codigo corto

Incorrecta creación de un nuevo producto por codigo repetido

Incorrecta creación de un nuevo producto nombre largo mayor a 30 caracteres

Incorrecta creación de un nuevo producto descripcion largo mayor a 70 caracteres

Incorrecta creación de un nuevo producto precio menor a 0

@mytag

✓ **Scenario: Correcta creación de un nuevo producto** 2ms

✓ **Given** que se ingresa un código numérico de 12345

✓ **And** se ingresa un nombre de producto "Shampoo"

✓ **And** se ingresa una descripción "Dale vida a tu pelo!"

✓ **And** se coloca un precio de 100

✓ **When** presiono el botón de creación

✓ **Then** se da de alta el producto correctamente

@mytag

✓ **Scenario: Incorrecta creación de un nuevo producto por codigo largo** 1ms

✓ **Given** que en el escenario_II se ingresa un código numérico de 1234567

✓ **And** en el escenario_II ingresa un nombre de producto "Shampoo"

✓ **And** en el escenario_II ingresa una descripción "Dale vida a tu pelo!"

✓ **And** en el escenario_II coloca un precio de 100

✓ **When** en el escenario_II presiono el botón de creación

SpecFlowProject (7)	94 ms		Pruebas en grupo: 7
SpecFlowProject.Features (7)	94 ms		⌚ Duración total: 94 ms
NewProductFeature (7)	94 ms		Salidas
Correcta creación de un nuevo producto	4 ms	Cate...	✓ 7 Correcta
Incorrecta creación de un nuevo producto descripcion largo mayor a 70 caracteres	5 ms	Cate...	
Incorrecta creación de un nuevo producto nombre largo mayor a 30 caracteres	2 ms	Cate...	
Incorrecta creación de un nuevo producto por codigo corto	2 ms	Cate...	
Incorrecta creación de un nuevo producto por codigo largo	2 ms	Cate...	
Incorrecta creación de un nuevo producto por codigo repetido	5 ms	Cate...	
Incorrecta creación de un nuevo producto precio menor a 0	74 ms	Cate...	

8

Feature: BuyProduct

Living Documentation Analytics

Filter by Keyword

Filter by Scenario Result

Test results

SpecFlowProject

Features

BuyProduct

Correcta compra de un producto único

Correcta compra de un producto con varias unidades

Correcta compra de dos productos distintos de la misma farmacia

Correcta compra de dos productos distintos de distinta farmacia

Correcta compra de productos combinado con medicamento

Intento de compra de producto fallido por monto negativo

deleteProduct

getAllProducts

getProductById

Feature: BuyProduct

@mytag

Scenario: Correcta compra de un producto único 2ms

Given que en el escenario_I, selecciono un producto con una unidad

When en el escenario_I, presiono el botón de añadir al carrito y finalizo la compra

Then en el escenario_I, se compra el producto correctamente

@mytag

Scenario: Correcta compra de un producto con varias unidades 7ms

Given que en el escenario_II, selecciono un producto con dos unidades

Feature: deleteProduct

Living Documentation Analytics

Filter by Keyword

Filter by Scenario Result

Test results

SpecFlowProject

Features

BuyProduct

deleteProduct

Correcta eliminacion de un nuevo producto

Incorrecta eliminacion de un nuevo producto

getAllProducts

getProductById

getProductByUser

newProduct

updateProduct

Feature: deleteProduct

@mytag

Scenario: Correcta eliminacion de un nuevo producto 4ms

Given que se ingresa un producto de id 12345

And que se ingresa un id de un product 12345 para eliminar

When presiono el botón de eliminar

Then se elimina el producto correctamente

@mytag

Scenario: Incorrecta eliminacion de un nuevo producto 57ms

Feature: getAllProducts

Living Documentation Analytics

Filter by Keyword

Filter by Scenario Result

Test results

SpecFlowProject

Features

BuyProduct

deleteProduct

getAllProducts

Correcta obtencion de los productos

Incorrecta obtencion de los productos de una farmacia

getProductById

getProductByUser

newProduct

updateProduct

Feature: getAllProducts

@mytag

Scenario: Correcta obtencion de los productos 55ms

Given Una farmacia de id 12345

And existe un producto en la base

When cargo la lista de productos

Then se cargan los productos correctamente

@mytag

Scenario: Incorrecta obtencion de los productos de una farmacia 1ms

Given un id de farmacia 12345789 que no existe

Feature: getProductById

Living Documentation Analytics

Filter by Keyword Filter by Scenario Result X

+ - Test results

- SpecFlowProject
 - 7 Passed 0 Failed 0 Others
 - Features
 - 7 Passed 0 Failed 0 Others
 - BuyProduct
 - deleteProduct
 - getAllProducts
 - getProductById
 - Correcta obtencion de los productos
 - Incorrecta obtencion del producto por Id
 - getProductByUser
 - newProduct
 - updateProduct

Feature: getAllProducts

@mytag

Scenario: Correcta obtencion de los productos 55ms

- Given Una farmacia de id 12345
- And existe un producto en la base
- When cargo la lista de productos
- Then se cargan los productos correctamente

@mytag

Scenario: Incorrecta obtencion de los productos de una farmacia 1ms

- Given un id de farmacia 12345789 que no existe

Feature: getProductByUser

Living Documentation Analytics

Filter by Keyword Filter by Scenario Result X

+ - Test results

- SpecFlowProject
 - 7 Passed 0 Failed 0 Others
 - Features
 - 7 Passed 0 Failed 0 Others
 - BuyProduct
 - deleteProduct
 - getAllProducts
 - getProductById
 - getProductByUser
 - Correcta obtencion de los productos por farmacia
 - newProduct
 - updateProduct

Feature: getProductByUser

@mytag

Scenario: Correcta obtencion de los productos por farmacia 55ms

- Given un usuario
- When hago la peticion para ver los productos de su farmacia
- Then obtengo los productos

Feature: updateProduct

Living Documentation Analytics

Filter by Keyword Filter by Scenario Result X

+ - Test results

- getAllProducts
- getProductById
- getProductByUser
- newProduct
- updateProduct
 - Correcta modificación de un producto
 - Incorrecta modificación de un producto por codigo largo
 - Incorrecta modificación de un producto por codigo corto
 - Incorrecta modificación de un producto que no existe
 - Incorrecta modificación de un producto nombre largo mayor a 30 caracteres
 - Incorrecta modificación de un producto descripcion largo mayor a 70 caracteres
 - Incorrecta modificación de un producto precio menor a 0

Feature: updateProduct

@mytag

Scenario: Correcta modificación de un producto 3ms

- Given que existe un producto con código numérico 12345
- And se ingresa en el escenario_UI un código numérico de 12345
- And se ingresa en el escenario_UI un nombre de producto "Shampoo"
- And se ingresa en el escenario_UI una descripción "Dale vida a tu pelo!"
- And se coloca en el escenario_UI un precio de 100
- When presiono el botón de modificación
- Then se modifica la información del producto correctamente

Generated by SpecFlow+LivingDoc - Give us feedback!

Video de la revisión de las nuevas funcionalidades con el PO

El vídeo de la revisión con el PO, se puede encontrar en el repositorio en GitHub, rama main, directorio “Documentación/Entrega 3 - ISA2”. Contiene la revisión de las nuevas funcionalidades de la aplicación

El link directo es:

<https://github.com/IngSoft-ISA2-2023-2/obligatorio-grupo11-bandera-caffarena-gonzalez/blob/main/Documentaci%C3%B3n/Entrega%203-20-ISA2/RevisionPO3.mkv>

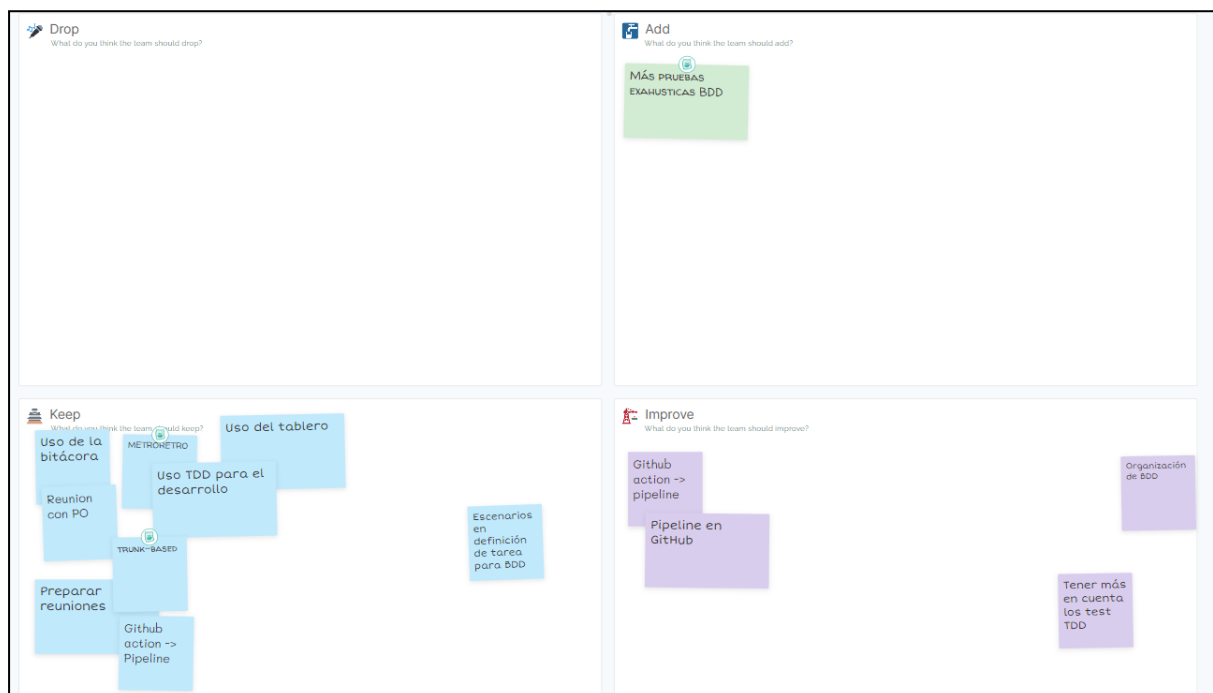
Video de retrospectiva

El vídeo de la retrospectiva se puede encontrar en el repositorio en GitHub, rama main, directorio “Documentación/Entrega 3 - ISA2”.

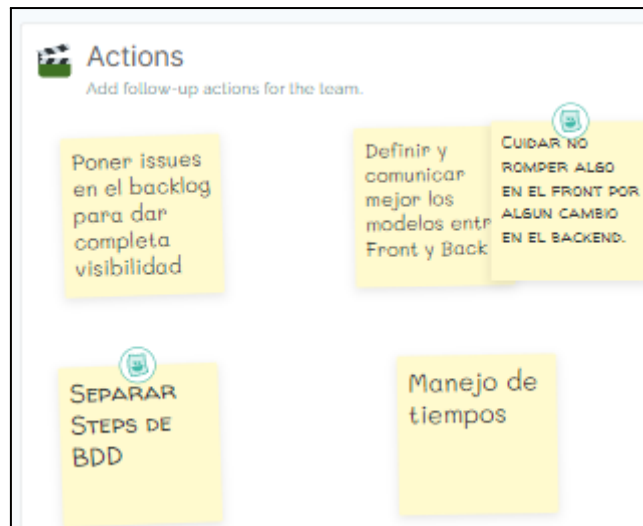
El link directo es:

<https://github.com/IngSoft-ISA2-2023-2/obligatorio-grupo11-bandera-caffarena-gonzalez/blob/main/Documentaci%C3%B3n/Entrega%203-20-ISA2/Retro3.mkv>

Tablero DAKI



DAKI - Imagen 1/2



DAKI - Imagen 2/2

En conclusión de la retrospectiva podemos entender que nuevamente se realizó un buen trabajo en esta etapa a pesar de los nuevos desafíos que plantearon la utilización de BDD y la restricción de realizar primero en FrontEnd antes que el BackEnd. A nivel de proceso no estamos identificando elementos para DROP porque no hay alguna actividad que no haya aportado valor a la entrega.

En cuanto a las herramientas, se han utilizado prácticamente las mismas que en la entrega anterior, lo que nos lleva a mantener un Keep y Add similar. Donde se realizó mayor foco, es en los aspectos de mejora, los cuales se ven reflejados en la pregunta "Improve" y es la fuente de la mayoría de las acciones a tomar para la siguiente entrega.