

Universidad ORT Uruguay
Facultad de Ingeniería

Informe de avance de etapa: Entrega 3
Ingeniería de Software Ágil 2

**Entregado como requisito para la obtención del título de Ingeniero en
Sistemas**

Sofia Montero - 250618

Romina Pardo - 253445

Andres Wilchinski - 193694

Tutor: Carina Fontan y Alvaro Ortas

2023

https://github.com/IngSoft-ISA2-2023-2/250618_193694_253445

Índice

Índice	1
Información sobre la entrega	4
Objetivos del Proyecto	4
Entregas y Evaluación	4
Registro de actividades	4
Definiciones Iniciales	5
Definición de Kanban	5
Roles asignados	5
Trunk Based	6
Explicación del tablero y su vínculo con el proceso de ingeniería	6
Uso de tableros por entrega	7
Explicación del pipeline y su vínculo con el proceso de ingeniería	7
Proceso de ingeniería basado en BDD	8
Nuevos escenarios	8
User Story 1: Alta de Producto	9
Escenarios genéricos	9
Escenario 1: Alta de producto con datos correctos	9
Escenario 2: Alta de producto con código incorrecto	10
Escenario 3: Alta de producto con nombre incorrecto	10
Escenario 4: Alta de producto con descripción incorrecta	10
Escenario 5: Alta de producto con precio incorrecto	10
Escenario 6: Alta de producto con código repetido	10
Escenarios específicos	11
Escenario 1: Alta de producto con datos correctos	11
Escenario 2: Alta de producto con código incorrecto	11
Escenario 3: Alta de producto con nombre incorrecto	11
Escenario 4: Alta de producto con descripción incorrecta	11
Escenario 5: Alta de producto con precio incorrecto	11
Escenario 6: Alta de producto con código repetido	12
SpecFlow	12
User Story 2: Baja de Producto	15
Escenario genérico	15
Escenario 1: Baja de producto correcta	15
Escenario 2: Baja de productos incorrecta	15
Escenarios específicos	15
Escenario 1: Baja de producto correcta	15

Escenario 2: Baja de productos incorrecta	16
SpecFlow	16
User Story 3: Modificación de productos	16
Escenario genérico	17
Escenario 1: Modificación correcta	17
Escenario 2: Modificación con código de largo invalido	17
Escenario 3: Modificación con código incorrecto	17
Escenario 4: Modificación con nombre incorrecto	17
Escenario 5: Modificación con descripción incorrecta	18
Escenario 6: Modificación dejando algún campo vacío	18
Escenario específico	18
Escenario 1: Modificación correcta	18
Escenario 2: Modificación con código incorrecto	19
Escenario 3: Modificación con código incorrecto	19
Escenario 4: Modificación con nombre incorrecto	19
Escenario 5: Modificación con descripción incorrecta	20
Escenario 6: Modificación dejando algún campo vacío	20
SpecFlow	21
User Story 4: Comprar productos	24
Escenario genérico	24
Escenario 1: Compra correcta	24
Escenario 2: Compra incorrecta, producto no existe en dicha farmacia o no existe en general	24
Escenario 3: Compra incorrecta, cantidad no válida	24
Casos de pruebas	24
Alta Producto:	24
Baja Producto:	25
Editar Producto:	25
Estructura y Mantenimiento del repositorio	26
Creación del repositorio	26
Mantenimiento del repositorio	26
Uso de Documentos de Google	26
Uso de Hojas de Cálculos de Google	26
Análisis del trabajo	27
Registro de las daily's	27
11/10/2023: Llamada por Teams	27
14/10/2023: Llamada por Google Meets	27
17/10/2023: Mensaje de texto	27
20/10/2023: Llamada por Google Meets	27

22/10/2023: Llamada por Google Meets	28
Registro de la Review	28
Registro de retrospectiva	28
Drop	29
Add	29
Keep	30
Improve	31
Registro de avance	33
Registro del trabajo y Esfuerzo	33
Métricas	34
Resultados obtenidos en el período	36
Dificultades encontradas y formas de solución	36
Lecciones aprendidas y mejoras en el proceso	36
Con respecto a la entrega anterior	37

Información sobre la entrega

Objetivos del Proyecto

- Aplicar un marco de gestión ágil.
- Analizar la deuda técnica.
- Implementar un repositorio y procedimientos de versionado.
- Crear un pipeline con eventos y acciones.
- Integrar prácticas de QA en el pipeline y gestionar el feedback.
- Generar escenarios de testing desde la perspectiva del usuario.
- Automatizar el testing funcional o de caja negra.
- Reflexionar sobre DevOps

Entregas y Evaluación

El proyecto se divide en cinco entregas. Cada una se evaluará en función de su cumplimiento y calidad. Además, se llevarán a cabo retrospectivas después de cada entrega para reflexionar sobre el proceso y realizar mejoras continuas.

Informe Final

Al final del proyecto, se presentará un informe académico que documentará todas las actividades realizadas, lecciones aprendidas y conclusiones obtenidas a lo largo del proyecto.

Registro de actividades

En cada entrega se aplicarán ceremonias de gestión ágil que el equipo deberá incorporar y adaptar.

Se espera que cada integrante dedique un mínimo de 5 horas semanales a las actividades de ingeniería del proyecto (sin incluir gestión y retrospectivas). Antes de cada entrega, el equipo realizará una retrospectiva y confeccionará un informe de avance, que debe incluir:

- Registro de las actividades realizadas (fecha, horas, integrante).
- Resultados obtenidos en el período.
- Dificultades encontradas y formas de solución.
- Lecciones aprendidas y mejoras en el proceso.

Se debe realizar un informe académico que detalle las actividades realizadas y justifique las decisiones tomadas para cumplir con estos objetivos. Es esencial fundamentar todas las decisiones tomadas durante el proyecto.

Definiciones Iniciales

Definición de Kanban

Para este proyecto se decidió utilizar la metodología Kanban y su artefacto Kanban board. El mismo está formado por 2 ceremonias obligatorias, el stand up diario y la retrospectiva. En esta metodología no hay roles de product owner o scrum master como si lo hay en scrum pero como también la metodología se tiene que adaptar a las necesidades del equipo y no al revés decidimos agregar dichos roles ya que nos favorecía. El stand up diario se realiza todos los días y dura aproximadamente 15 minutos, en esta instancia los desarrolladores del equipo y cada uno da un breve resumen de en qué trabajaron el día anterior, en que van a trabajar hoy y también dicen si tienen alguna dificultad. Como no trabajábamos todos los días en este proyecto decidimos realizar stand ups cada 2 o 3 días. Otra ceremonia es la retrospectiva, que se realiza con el objetivo de que el equipo mejore. Aquí se aplica un sistema DAKI y además de ir el equipo de desarrollador, se agrega el scrum master, el cual conduce dicha ceremonia. Además de estas 2 ceremonias decidimos agregar la review, en esta le mostraremos al product owner como resolvimos los bugs o cuales son las nuevas funcionalidades agregadas al proyecto. Como mencionado anteriormente, el artefacto más importante de la metodología es la Kanban board, en este se puede visualizar fácilmente qué tareas o user stories fueron ya realizadas, se están haciendo o todavía no se han comenzado a hacer. Hay varios tipos de tableros, dependiendo de las necesidades y especificaciones que necesitemos según el tipo de tareas necesarias para cada entrega.

Kanban se basa en 6 principios bien claros.

- Visualizar el flujo de trabajo: esto es posible gracias al kanban board
- Limitar el WIP (work in progress): permite evitar los cuellos de botella y defectos rápidamente.
- Gestionar y medir el flujo: medir para ver si se mejora y en base a eso tomar medidas acordes.
- Implementar ciclos de feedback: para este principio las retrospectivas son clave.
- Explicar políticas y procedimientos.
- Mejora continua mediante la colaboración.

Roles asignados

Roles Posibles: PO, SM, DESA, TES

Todos los integrantes del equipo deben poseer los roles de DESA y TES. Por otro lado, en esta segunda instancia, se ve pertinente que sean asignados los siguientes roles: Desarrollador, Tester, Product Owner y Scrum Master. Esta decisión fue tomada teniendo en cuenta la letra de la entrega y los artefactos recomendados para la misma. En el caso de Scrum Master, lo consideramos necesario debido a que una de las ceremonias, la retrospectiva, es liderada por el. En cambio, la necesidad del Product Owner se debe a que otro artefacto recomendado es el video revisión de bugs con el product owner. Debido a que cada uno de nosotros arreglo un bug decidimos nombrar a dos product owners para que el bug que cada

uno arreglo sea revisado por otro integrante. A su vez, decidimos que el rol de Scrum Master rotará, por lo tanto este fue asignado a alguien distinto en comparación a la primera entrega.

- **Sofia Montero:** Desarrollador, Tester, Product Owner, Scrum Master
- **Romina Pardo:** Desarrollador, Tester
- **Andres Wilchinski:** Desarrollador, Tester

Trunk Based

Debido a que en esta entrega comenzamos a codificar, creímos necesario definir la forma de trabajo en git. Para el proyecto, hemos decidido trabajar con trunk based development, una práctica de gestión del control de versiones en la que los desarrolladores incorporan pequeñas actualizaciones frecuentes a la rama principal. Al usar trunk based se optimiza para la productividad del equipo ya que todos trabajan en una misma área y los merge constantes hacen que los problemas de merge sean más concretos y simples de resolver. Debido al alcance del proyecto y de cada entrega los commits se realizan todos los días en los que el código es trabajado, no diariamente. Aunque trabajar con esta práctica puede resultar más complicado ya que todos los desarrolladores estamos acostumbrados a usar Git Flow consideramos que el esfuerzo extra vale la pena debido a los beneficios que trae.

Explicación del tablero y su vínculo con el proceso de ingeniería

Para la creación del tablero hemos realizado algunas mejoras y ajustes de la primera versión del tablero. Esta segunda versión consta de las siguientes columnas: Sprint Backlog, Requirements Definition, Test Cases Implementation, Refactoring, Integration Testing, Done. Para esta versión se mantiene la clasificación de tarea o user story.

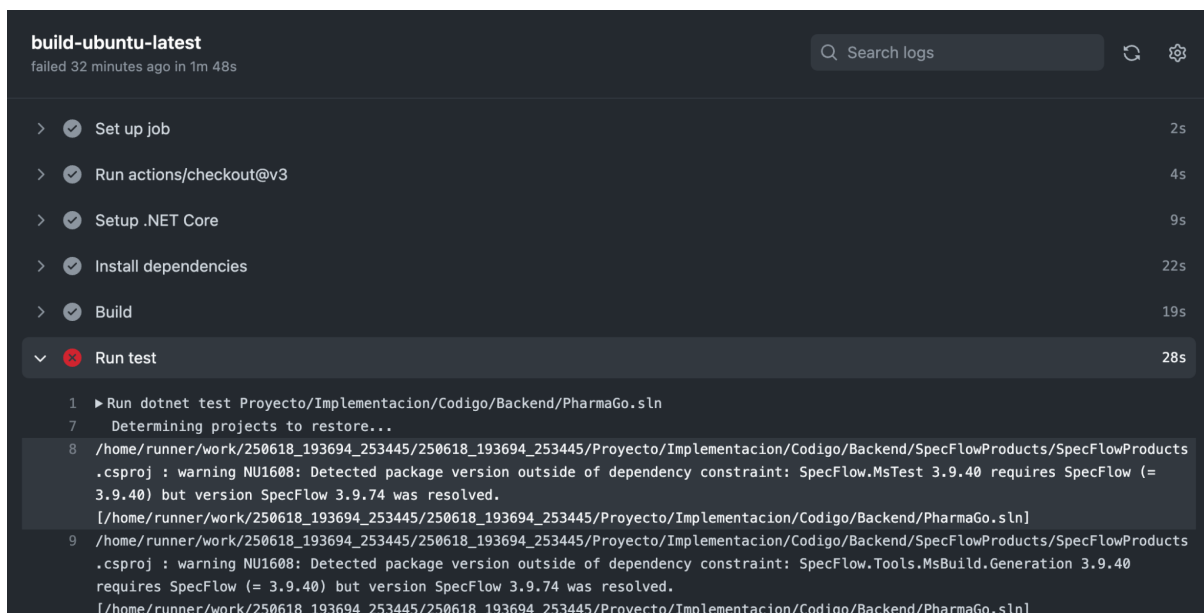
El flujo de trabajo en este nuevo tablero es el siguiente: inicialmente, todas las tareas se agregan a la columna Sprint Backlog que actúa de la misma manera que la columna ToDo en la versión pasada. Cuando se comienza a trabajar en una tarea o User Story se pasa a la columna "Requirements Definition" que para las tareas actúa como In Progress pero para las User Stories es donde se definen los requerimientos. Luego, cuando comienza la implementación de los test las User Stories se pasan a la columna Test Cases Implementation. Cuando este paso es terminado y se comienza a programar el frontend y backend se pasa a la columna App Implementation. Una vez que la implementación o la tarea es terminada se pasa a la columna Refactoring. Luego de esta la user story pasa a la columna integration testing, donde se testea la funcionalidad de punta a punta y como se integra con el sistema desde el front end para darla por terminada. Finalmente, al completarse, se coloca en la columna Done. El registro de fechas en las diferentes etapas del proceso permite un seguimiento preciso del progreso de cada tarea y facilita la gestión del proyecto.

Uso de tableros por entrega

Se creará un tablero por entrega utilizando la herramienta git project. La división de tablero por entrega nos permitirá poseer un control registrado del estado del proyecto en cada entrega. Podrá acceder al tablero de está [aquí](#).

Explicación del pipeline y su vínculo con el proceso de ingeniería

Al agregar los test de specflow la pipeline nos dejó de correr correctamente la pipeline de github actions ya que ahora dan error en los test. Esto antes no pasaba y cuando solo corremos los unit test estos pasan todos. Creemos que el problema se da porque la action de correr los test también trata de correr los test de specflow y por esta razón cae.



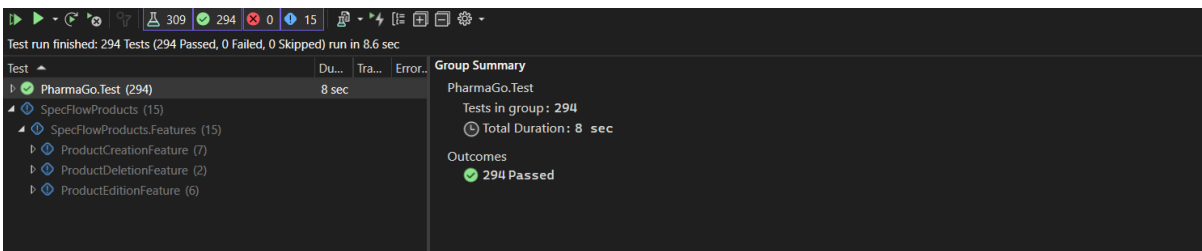
The screenshot shows a GitHub Actions workflow run for the job 'build-ubuntu-latest'. The workflow failed 32 minutes ago in 1m 48s. The steps listed are: Set up job (2s), Run actions/checkout@v3 (4s), Setup .NET Core (9s), Install dependencies (22s), Build (19s), and Run test (28s). The 'Run test' step is expanded, showing the following logs:

```
1 ▶ Run dotnet test Proyecto/Implementacion/Codigo/Backend/PharmaGo.sln
7 Determining projects to restore...
8 /home/runner/work/250618_193694_253445/250618_193694_253445/Proyecto/Implementacion/Codigo/Backend/SpecFlowProducts/SpecFlowProducts
.csproj : warning NU1608: Detected package version outside of dependency constraint: SpecFlow.MsTest 3.9.40 requires SpecFlow (=
3.9.40) but version SpecFlow 3.9.74 was resolved.
[/home/runner/work/250618_193694_253445/250618_193694_253445/Proyecto/Implementacion/Codigo/Backend/PharmaGo.sln]
9 /home/runner/work/250618_193694_253445/250618_193694_253445/Proyecto/Implementacion/Codigo/Backend/SpecFlowProducts/SpecFlowProducts
.csproj : warning NU1608: Detected package version outside of dependency constraint: SpecFlow.Tools.MsBuild.Generation 3.9.40
requires SpecFlow (= 3.9.40) but version SpecFlow 3.9.74 was resolved.
[/home/runner/work/250618_193694_253445/250618_193694_253445/Proyecto/Implementacion/Codigo/Backend/PharmaGo.sln]
```

Aunque no entendemos del todo al 100% porque esto pasa ya que dice que corrió correctamente los 294 unit test que tenemos.


```
Run test 28s
1126 Warning:
/home/runner/work/250618_193694_253445/250618_193694_253445/Proyecto/Implementacion/Codigo/Backend/PharnaGo.Test/DataAccess.Test/PurchasesRepositoryTests.cs(28,36): warning CS0169: The field 'PurchasesRepositoryTests.pharmacyRepository' is never used
[/home/runner/work/250618_193694_253445/250618_193694_253445/Proyecto/Implementacion/Codigo/Backend/PharnaGo.Test/PharnaGo.Test.csproj]
1127 PharmaGo.Test ->
/home/runner/work/250618_193694_253445/250618_193694_253445/Proyecto/Implementacion/Codigo/Backend/PharnaGo.Test/bin/Debug/net6.0/PharnaGo.Test.dll
1128 Test run for
/home/runner/work/250618_193694_253445/250618_193694_253445/Proyecto/Implementacion/Codigo/Backend/PharnaGo.Test/bin/Debug/net6.0/PharnaGo.Test.dll (.NETCoreApp,Version=v6.0)
1129 Microsoft (R) Test Execution Command Line Tool Version 17.7.1 (x64)
1130 Copyright (c) Microsoft Corporation. All rights reserved.
1131
1132 Starting test execution, please wait...
1133 A total of 1 test files matched the specified pattern.
1134
1135 Passed! - Failed: 0, Passed: 294, Skipped: 0, Total: 294, Duration: 3 s - PharmaGo.Test.dll (net6.0)
1136 Error: Process completed with exit code 1.

Post Setup .NET Core 0s
Post Run actions/checkout@v3 0s
```



Proceso de ingeniería basado en BDD

El Behavior-Driven Development (BDD) se enfoca en el comportamiento del software desde la perspectiva del usuario, fomentando una colaboración estrecha entre desarrolladores, probadores y expertos en el dominio. Utiliza "historias de usuario" para describir el comportamiento deseado de la aplicación en un lenguaje comprensible, las cuales se convierten en pruebas automatizadas para verificar los criterios de aceptación, garantizando que el software se desarrolle teniendo en cuenta las necesidades del cliente.

La Tercera Versión del Proceso de Ingeniería combina BDD con Kanban, lo que se traduce en un enfoque eficiente para la gestión del desarrollo de software. Las historias de usuario de BDD se representan como tarjetas en un tablero Kanban, permitiendo un flujo de trabajo transparente y una respuesta ágil a cambios en los requisitos del cliente.

También esto nos permite mantener una documentación actualizada del sistema.

Nuevos escenarios

Se utilizará este esquema genérico:

User Story [Número de la User Story]: [Nombre de la Historia de Usuario]

Como [rol del usuario]

Quiero [realizar una acción]

Para [lograr un objetivo]

Escenario Genérico: [Nombre del Escenario]

Escenario [Número del Escenario]: [Descripción del Escenario]

Dado [condición inicial]

Cuando [acción realizada por el usuario]

Y [acciones adicionales, si es necesario]

Entonces [resultado esperado]

Escenarios Específicos: [Nombre del Escenario]

Escenario [Número del Escenario]: [Descripción del Escenario]

Dado [condición inicial específica]

Cuando [acción realizada por el usuario]

Y [acciones adicionales, si es necesario]

Entonces [resultado esperado específico]

User Story 1: Alta de Producto

Como empleado de una farmacia

Quiero realizar el alta de un producto

Para que quede registrado en la farmacia

Escenarios genéricos

Escenario 1: Alta de producto con datos correctos

Dado un código con 5 dígitos, no repetido

Y un nombre del producto alfanumérico de menos de 30 caracteres, no vacío

Y una descripción alfanumérica de menos de 70 caracteres, no vacío

Y un precio decimal

Cuando ingreso los datos y presiono el botón crear

Entonces debería crearse el producto en la farmacia

Escenario 2: Alta de producto con código incorrecto

Dado un código de más de 5 dígitos

Y un nombre, descripción y precio correcto

Cuando ingreso los datos y presiono el botón crear

Entonces debería mostrarme un mensaje de error indicando que el código es incorrecto

Escenario 3: Alta de producto con nombre incorrecto

Dado un nombre mayor de 30 caracteres

Y un código, descripción y precio correcto

Cuando ingreso los datos y presiono el botón crear

Entonces debería mostrarme un mensaje de error indicando que el nombre es incorrecto

Escenario 4: Alta de producto con descripción incorrecta

Dado una descripción mayor de 70 caracteres

Y un código, nombre y precio correcto

Cuando ingreso los datos y presiono el botón crear

Entonces debería mostrarme un mensaje de error indicando que la descripción es incorrecta

Escenario 5: Alta de producto con precio incorrecto

Dado un precio no decimal

Y un código, descripción y nombre correcto

Cuando ingreso los datos y presiono el botón crear

Entonces debería mostrarme un mensaje de error indicando que el precio es incorrecto

Escenario 6: Alta de producto con código repetido

Dado el código de un producto ya ingresado al sistema

Y un nombre, descripción y precio correcto

Cuando ingreso los datos y presiono el botón crear

Entonces debería mostrarme un mensaje de error indicando que el código es incorrecto

Escenarios específicos

Escenario 1: Alta de producto con datos correctos

Dado el código 12345

Y el nombre Shampoo Sedal 200 ml

Y la descripción Dale vida a tu pelo con el nuevo shampoo Sedal

Y el precio 80,00

Cuando ingreso los datos y presiono el botón crear

Entonces debería crearse el producto en la farmacia

Escenario 2: Alta de producto con código incorrecto

Dado el código 123456

Y el nombre shampoo Sedal, descripción Dale vida a tu pelo con el nuevo shampoo Sedal y precio 80,00

Cuando ingreso los datos y presiono el botón crear

Entonces debería mostrarme un mensaje de error: ErrorAltaProducto: Codigo incorrecto

Escenario 3: Alta de producto con nombre incorrecto

Dado el nombre Shampoo Sedal de 200 mililitros

Y un código, descripción y precio correcto

Cuando ingreso los datos y presiono el botón crear

Entonces debería mostrarme un mensaje de error: ErrorAltaProducto: Nombre incorrecto

Escenario 4: Alta de producto con descripción incorrecta

Dada la descripción Dale vida a tu pelo con el nuevo Shampoo Sedal de la línea pelo hidratado

Y un código, nombre y precio correcto

Cuando ingreso los datos y presiono el botón crear

Entonces debería mostrarme un un mensaje de error: ErrorAltaProducto: Descripción incorrecta

Escenario 5: Alta de producto con precio incorrecto

Dado un precio PRECIO

Y un código, descripción y nombre correcto

Cuando ingreso los datos y presiono el botón crear

Entonces debería mostrarme un mensaje de error: ErrorAltaProducto: Precio incorrecto

Escenario 6: Alta de producto con código repetido

Dado el código de un producto del sistema, 12345

Y el nombre shampoo Sedal, descripción Dale vida a tu pelo con el nuevo shampoo Sedal y precio 80,00

Cuando ingreso los datos y presiono el botón crear

Entonces debería mostrarme un mensaje de error: ErrorAltaProducto: Código repetido

SpecFlow

Feature: ProductCreation

In order to register products in my pharmacy

As a pharmacy employee

I want to be able to create them

@tag1

Scenario: Create product with correct data

Given the code "12347"

And the name "Shampoo Sedal 200 ml"

And the description "Dale vida a tu pelo con el nuevo shampoo Sedal"

And the price 80,00

And the pharmacy "farmacia1234"

When press the create button

Then the product should be registered correctly with code 200

@tag2

Scenario: Create product with wrong code length

Given the code "123478"

And the name "Shampoo Sedal 200 ml"

And the description "Dale vida a tu pelo con el nuevo shampoo Sedal"

And the price 80,00

And the pharmacy "farmacia1234"

When press the create button

Then the product should not be registered with code 400

@tag3

Scenario: Create product with wrong code, no digits

Given the code "1234A"

And the name "Shampoo Sedal 200 ml"

And the description "Dale vida a tu pelo con el nuevo shampoo Sedal"

And the price 80,00

And the pharmacy "farmacia1234"

When press the create button

Then the product should not be registered with code 400

@tag4

Scenario: Create product with wrong name, no alphanumeric

Given the code "12345"

And the name "Shampoo Sedal 200 ml%%\$"

And the description "Dale vida a tu pelo con el nuevo shampoo Sedal"

And the price 80,00

And the pharmacy "farmacia1234"

When press the create button

Then the product should not be registered with code 400

@tag5

Scenario: Create product with wrong name, length longer than 30 characters

Given the code "12345"

And the name "123456789 123456789 123456789 0"

And the description "Dale vida a tu pelo con el nuevo shampoo Sedal"

And the price 80,00

And the pharmacy "famacia1234"

When press the create button

Then the product should not be registered with code 400

@tag7

Scenario: Create product with wrong description, non alphanumeric

Given the code "12345"

And the name "Shampoo Sedal 200 ml"

And the description "Dale vida a tu pelo con el nuevo shampoo Sedal &"

And the price 80,00

And the pharmacy "famacia1234"

When press the create button

Then the product should not be registered with code 400

@tag8

Scenario: Create product with wrong description, length longer than 70 characters

Given the code "12345"

And the name "Shampoo Sedal 200 ml"

```
And the description "123456789 123456789 123456789 123456789 123456789 123456789
123456789 1"
```

And the price 80,00

And the pharmacy "farmacia1234"

When press the create button

Then the product should not be registered with code 400

User Story 2: Baja de Producto

Como empleado de una farmacia

Quiero realizar la baja de un producto

Para que deje de quedar registrado en la farmacia

Escenario genérico

Escenario 1: Baja de producto correcta

Dado un producto ingresado en el sistema perteneciente a la farmacia del empleado

Cuando selecciono el producto

Y presiono el botón eliminar

Entonces el producto queda eliminado de los productos de la farmacia

Escenario 2: Baja de productos incorrecta

Dado un producto ingresado en el sistema no perteneciente a la farmacia del empleado

Cuando selecciono el producto

Y presiono el botón eliminar

Entonces debería mostrarme un mensaje de error indicando que no se pudo eliminar el producto

Escenarios específicos

Escenario 1: Baja de producto correcta

Dado el producto Shampoo Sedal ingresado en el sistema perteneciente a la farmacia 1234

Cuando selecciono el producto

Y presiono el botón eliminar

Entonces el producto queda eliminado de los productos de la farmacia 1234

Escenario 2: Baja de productos incorrecta

Dado el producto ingresado en el sistema perteneciente a la farmacia 1234

Cuando selecciono el producto

Y presiono el botón eliminar

Entonces debería mostrarme un mensaje de error indicando que no se pudo eliminar el producto

SpecFlow

Feature: ProductDeletion

In order to stop showing certain products in my pharmacy

As a pharmacy employee

I want to be able to delete them

@tag1

Scenario: Delete product correctly

Given the product registered in my pharmacy with id 1

When press the delete button for that product

Then the product should be deleted correctly

@tag2

Scenario: Delete product incorrectly

Given the product not registered in my pharmacy with id 2

When press the delete button for that product

Then the product should not be deleted

User Story 3: Modificación de productos

Como empleado de una farmacia

Quiero poder modificar los atributos de un producto ya creado

Para que no se deba eliminar y recrear el producto cada vez que se desea cambiar alguna propiedad

Escenario genérico

Escenario 1: Modificación correcta

Dado un producto ingresado al sistema de su farmacia

Cuando presione el botón "Modificar"

Y agregue mis modificaciones a sus atributos

Cuando apriete el botón "Guardar"

Entonces el producto se modificó correctamente con código 200

Escenario 2: Modificación con código de largo invalido

Dado un producto ingresado en el sistema de su farmacia

Cuando presione el botón "Modificar"

Y agregue mis modificaciones a sus atributos

Y cambie el código a uno con más o menos de 5 dígitos

Y presione el botón "Guardar"

Entonces no me debe modificar el producto con código 400

Escenario 3: Modificación con código incorrecto

Dado un producto ingresado en el sistema de su farmacia

Cuando presione el botón "Modificar"

Y agregue mis modificaciones a sus atributos

Y cambie el código a uno ya existente en otro producto de la farmacia

Y presione el botón "Guardar"

Entonces no me debe modificar el producto con código 500

Escenario 4: Modificación con nombre incorrecto

Dado un producto ingresado en el sistema de su farmacia

Cuando presione el botón "Modificar"

Y agregue mis modificaciones a sus atributos

Y cambie el nombre a uno con más de 30 caracteres

Y presione el botón "Guardar"

Entonces no me debe modificar el producto con código 400

Escenario 5: Modificación con descripción incorrecta

Dado un producto ingresado en el sistema de su farmacia

Cuando presione el botón "Modificar"

Y agregue mis modificaciones a sus atributos

Y cambie la descripción a una con más de 70 caracteres

Y presione el botón "Guardar"

Entonces no me debe modificar el producto con código 400

Escenario 6: Modificación dejando algún campo vacío

Dado un producto ingresado en el sistema de su farmacia

Cuando presione el botón "Modificar"

Y agregue mis modificaciones a sus atributos

Y deje un campo vacío

Y presione el botón "Guardar"

Entonces no me debe modificar el producto con código 400

Escenario específico

Escenario 1: Modificación correcta

Dado el producto con id 1

Y código "12345"

Y nombre "Roll-On Antitranspirante"

Y descripción "crema para el cuidado de las axilas"

Y precio de 70.00

Cuando presiono el botón "Modificar" para ese producto

Y modifique el nombre a "Roll-On Antitranspirante"

Y modifique la descripción a "Con crema hidratante para el cuidado de las axilas"

Y guarde los cambios

Entonces el producto debe ser editado correctamente con el código 200

Escenario 2: Modificación con código incorrecto

Dado el producto con id 1

Y código "12345"

Y nombre "Roll-On Antitranspirante"

Y descripción "Con crema hidratante para el cuidado de las axilas"

Y precio de 70.00

Cuando presiono el botón "Modificar" para ese producto

Y modifique el código a "123"

Y guarde los cambios

Entonces el producto debe ser editado correctamente con el código 400

Escenario 3: Modificación con código incorrecto

Dado el producto con id 1

Y código "12345"

Y nombre "Roll-On Antitranspirante"

Y descripción "Con crema hidratante para el cuidado de las axilas"

Y precio de 70.00

Cuando presiono el botón "Modificar" para ese producto

Y modifique el código a "45678"

Y guarde los cambios

Entonces el producto debe ser editado correctamente con el código 500

Escenario 4: Modificación con nombre incorrecto

Dado el producto con id 1

Y código "12345"

Y nombre "Roll-On Antitranspirante"

Y descripción "Con crema hidratante para el cuidado de las axilas"

Y precio de 70.00

Cuando presiono el botón "Modificar" para ese producto

Y modifique el nombre a "El mejor Roll-On Antitranspirante"

Y modifique la descripción a "Con crema hidratante para el cuidado de las axilas"

Y guarde los cambios

Entonces el producto debe ser editado correctamente con el código 400

Escenario 5: Modificación con descripción incorrecta

Dado el producto con id 1

Y código "12345"

Y nombre "Roll-On Antitranspirante"

Y descripción "Con crema hidratante para el cuidado de las axilas"

Y precio de 70.00

Cuando presiono el botón "Modificar" para ese producto

Y modifique el nombre a "Roll-On Antitranspirante"

Y modifique la descripción a "Roll-On antitranspirante original Dove con crema hidratante para el cuidado de las axilas"

Y guarde los cambios

Entonces el producto debe ser editado correctamente con el código 400

Escenario 6: Modificación dejando algún campo vacío

Dado el producto con id 1

Y código "12345"

Y nombre "Roll-On Antitranspirante"

Y descripción "Con crema hidratante para el cuidado de las axilas"

Y precio de 70.00

Cuando presiono el botón "Modificar" para ese producto

Y modifique el nombre a vacío

Y modifique la descripción a "Con crema hidratante para el cuidado de las axilas"

Y guarde los cambios

Entonces el producto debe ser editado correctamente con el código 400

SpecFlow

Feature: ProductEdition

In order to update product information in my pharmacy

As a pharmacy employee

I want to be able to modify them

@tag1

Scenario: Successful Modification

Given the product with id 1

And code "12345"

And name "Antiperspirant Roll"

And description "cream for underarm care"

And price as 70.00

When I press the "Modify" button for that product, I should be able to modify the name to "Antitranspirante RollOn" and the description to "With moisturizing cream for underarm care"

And I save the changes

Then the product should be edited correctly with code 200

@tag2

Scenario: Modification with Incorrect Code

Given the product with id 1

And code "12345"

And name "Antitranspirante RollOn"

And description "With moisturizing cream for underarm care"

And price as 70.00

When I press the "Modify" button for that product, I should be able to modify the code to "123"

And I save the changes

Then the product should be edited correctly with code 400

@tag3

Scenario: Modification with Incorrect Code (Different Product)

Given the product with id 1

And code "12345"

And name "Antitranspirante RollOn"

And description "With moisturizing cream for underarm care"

And price as 70.00

When I press the "Modify" button for that product, I should be able to modify the code to "45678"

And I save the changes

Then the product should be edited correctly with code 500

@tag4

Scenario: Modification with Incorrect Name

Given the product with id 1

And code "12345"

And name "Antitranspirante RollOn"

And description "With moisturizing cream for underarm care"

And price as 70.00

When I press the "Modify" button for that product, I should be able to modify the name to "The best Antitranspirante RollOn" and the description to "With moisturizing cream for underarm care"

And I save the changes

Then the product should be edited correctly with code 400

@tag5

Scenario: Modification with Incorrect Description

Given the product with id 1

And code "12345"

And name "Antitranspirante RollOn"

And description "With moisturizing cream for underarm care"

And price as 70.00

When I press the "Modify" button for that product, I should be able to modify the name to "Antitranspirante RollOn" and the description to "Dove Original Roll On antiperspirant with moisturizing cream for underarm care"

And I save the changes

Then the product should be edited correctly with code 400

@tag6

Scenario: Modification Leaving a Field Blank

Given the product with id 1

And code "12345"

And name "Antitranspirante RollOn"

And description "With moisturizing cream for underarm care"

And price as 70.00

When I press the "Modify" button for that product, I should be able to modify the name to "" and the description to "With moisturizing cream for underarm care"

And I save the changes

Then the product should be edited correctly with code 400

User Story 4: Comprar productos

Como usuario anónimo

Quiero poder agregar a mi compras distintos productos

Para poder conseguir dichos productos

Escenario genérico

Escenario 1: Compra correcta

Dado un producto de una farmacia y una cantidad positiva mayor a 0 del mismo

Cuando selecciono añadir al carrito

Entonces se agrega a la compra dicho producto con ese stock al carrito

Escenario 2: Compra incorrecta, producto no existe en dicha farmacia o no existe en general

Dado un el nombre o el id de producto que no existe y una cantidad cualquiera

Cuando selecciono añadir al carrito

Entonces no se puede añadir al carrito dicho producto ya que no existe, y aparece un mensaje "The selected product does not exists"

Escenario 3: Compra incorrecta, cantidad no válida

Dado un producto de una farmacia y una cantidad negativa o decimal del mismo

Cuando selecciono añadir al carrito

Entonces no se puede añadir al carrito dicho producto ya que la cantidad solicitada es inválida. "The amount selected for this product is invalid"

Casos de pruebas

Alta Producto:

Partiendo de una base de datos que puede estar vacía o con elementos. Realizamos 7 test en specflow testeando los escenarios descritos arriba. En el primer test CreateProductWithCorrectData, como es el caso feliz donde el producto se crea correctamente en la base de datos tuvimos que agregar que luego se borre para que el test sea repetible y así siga cumpliendo con la sigla first. En la imagen de abajo se puede ver una foto de los test de specflow para alta producto. Specflow nos permite testear una de user stories, en este caso crear producto

ProductCreationFeature (7)	8,2 s
CreateProductWithCorrectData	2,3 s F...
CreateProductWithWrongCodeLength	1,9 s F...
CreateProductWithWrongCodeNoDigits	923 ms F...
CreateProductWithWrongDescriptionLengthLongerThan70Characters	709 ms F...
CreateProductWithWrongDescriptionNonAlphanumeric	766 ms F...
CreateProductWithWrongNameLengthLongerThan30Characters	826 ms F...
CreateProductWithWrongNameNoAlphanumeric	861 ms F...

Baja Producto:

Para testear el baja producto con specflow testeamos 2 casos porque nos encontramos un problema al hacerlo. DeleteProductCorrectly, testea el camino feliz de la user story. Antes de eliminarlo en este caso agregamos un producto a la base de datos. esto lo hacemos para que el test sea repetible. En el caso triste esto no es necesario ya que no se borra el dato.

ProductDeletionFeature (2)	116 ms
DeleteProductCorrectly	108 ms
DeleteProductIncorrectly	8 ms

-> En ambos casos nos da el mismo error: The given key 'ResponseStatusCode' was not present in the dictionary. (0,0s), presentado a aulas como consulta a ser resuelta en la próxima entrega, de igual manera la funcionalidad fue probada en el front y la evidencia queda en la review con el Product Owner

Editar Producto:

Partiendo de estos datos en la base de datos:

	Id	Code	Name	Price	Description	Deleted	PharmacyId
	1	12345	Antitranspirante RollOn	70,00	With moisturizing cream for underarm care	False	1
	2	45678	Perfume	100,50	Floral	False	1
	3	56743	Product	40,00	rty	False	1

ProductEditionFeature (6)	7,2 s
ModificationLeavingAFieldBlank	2 s F...
ModificationWithIncorrectCode	1,6 s F...
ModificationWithIncorrectCodeDifferentProduct	779 ms F...
ModificationWithIncorrectDescription	842 ms F...
ModificationWithIncorrectName	1,9 s F...
SuccessfulModification	82 ms F...

Estructura y Mantenimiento del repositorio

Creación del repositorio

El repositorio fue creado en la plataforma github. Esto nos permite tener un control de versiones sobre el proyecto, lo cual significa poseer control de las modificaciones realizadas sobre el proyecto.

Para hacer esto, primero se creó un equipo en la plataforma, donde se encontraran presentes cada uno de los mails de los integrantes. Después se agregó a la organización de la materia un nuevo repositorio agregándolo como perteneciente al equipo creado con anterioridad.

Después de eso, se agregó el contenido del proyecto a analizar y modificar al repositorio. Este fue encontrado en la página de aulas de la materia. Después de haberlo descargado, se usó git bash para incorporarlo a la rama “main” del repositorio.

Cuando el proyecto DevOps estuvo incorporado, se decidió agregar una descripción resumida de los objetivos de este obligatorio a su vez del mecanismo de evaluación.

Mantenimiento del repositorio

Para realizar el mantenimiento del repositorio se decidió utilizar las herramientas de Google Hojas de Cálculos y Documentos. A su vez, todos los archivos creados con estas plataformas, serán guardados en una carpeta compartida de Drive. Podrá acceder a esta carpeta [aquí](#).

Uso de Documentos de Google

Esta plataforma nos permitirá crear documentos de informes en formato pdf. Estos informes serán los que poseerán toda la información de análisis de cada entrega. Se creará un documento para cada entrega y luego se resumen todos para la entrega final.

Uso de Hojas de Cálculos de Google

Esta plataforma nos permitirá crear libros de trabajo donde registramos el esfuerzo con tareas y subtareas por entrega. Esto significa que se creará un libro de trabajo para cada entrega.

Análisis del trabajo

Registro de las daily's

11/10/2023: Llamada por Teams

P1: ¿Alguien está trabajando en algo que no está en el tablero?

- No

P2: Como equipo , ¿qué estamos buscando finalizar?

- En esta conversación se dialogó sobre la letra del obligatorio. Planteando preguntas e iniciando una estructuración de lo que hacer. Además de distribuir tareas.

P3: ¿Existen cuellos de botella u otros impedimentos al flujo de trabajo?

- No

14/10/2023: Llamada por Google Meets

P1: ¿Alguien está trabajando en algo que no está en el tablero?

- No

P2: Como equipo , ¿qué estamos buscando finalizar?

- En esta reunión se buscó iniciar la parte de specFlow

P3: ¿Existen cuellos de botella u otros impedimentos al flujo de trabajo?

- Se dio un error sobre el specFlow de creación de productos.

17/10/2023: Mensaje de texto

P1: ¿Alguien está trabajando en algo que no está en el tablero?

- No

P2: Como equipo , ¿qué estamos buscando finalizar?

- Solucionar el error de specFlow

P3: ¿Existen cuellos de botella u otros impedimentos al flujo de trabajo?

- Si, aunque se logró solucionar el problema anterior, surgió otro.

20/10/2023: Llamada por Google Meets

P1: ¿Alguien está trabajando en algo que no está en el tablero?

- No

P2: Como equipo , ¿qué estamos buscando finalizar?

- Solucionar el error de specFlow

P3: ¿Existen cuellos de botella u otros impedimentos al flujo de trabajo?

- No, se logró solucionar

22/10/2023: Llamada por Google Meets

P1: ¿Alguien está trabajando en algo que no está en el tablero?

- No

P2: Como equipo , ¿qué estamos buscando finalizar?

- Se busco finalizar el backend como el frontend de las user story.

P3: ¿Existen cuellos de botella u otros impedimentos al flujo de trabajo?

- Errores de spec flow y complicaciones con la implementación. Se pudieron solucionar al finalizar la reunión.

Registro de la Review

Se realizó un único video verificando todas las funcionalidades, más una extensión donde se corrige un error que saltó en la review inicial.

- [Inicial](#)
- [Continuación](#)

Registro de retrospectiva

Este registro de retrospectiva se basa en el método DAKI y se utiliza para documentar los aprendizajes y resultados de la retrospectiva.

También se creó un tablero en retrometro para acompañar la retrospectiva.

Formato de la retrospectiva:

- 5 minutos iniciales para responder todas las secciones. Esto se hace de forma anónima.
- Abrir las notas para que todos puedan ver lo que se escribió.
- Dialogar sobre cada sección y discutir para llegar a una respuesta conjunta.
- Luego, para finalizar, se eligiran las ideas más importantes de las planteadas para cumplir en la próxima entrega.

Podrá ver el tablero [aquí](#) y el video de la retrospectiva [aquí](#).

Drop

En esta sección, identificamos las cosas que hicimos mal en esta entrega y que deberíamos dejar de hacer en el futuro.

Pregunta: ¿Qué cosas hicimos mal en esta entrega que deberíamos dejar de hacer en el futuro?



La idea más importante destacada fue: “El tiempo nos jugó en contra, complicaciones con errores en specflow”

Add

Aquí, enumeramos las acciones o prácticas que no realizamos en esta entrega, pero que deberíamos comenzar a hacer en el futuro.

Pregunta: ¿Qué acciones o prácticas no realizamos en esta entrega, pero deberíamos comenzar a hacer en el futuro?

Vision - Add

What should we add to get to our objective?

Dejar los ultimos dos dias solo para revision dentro de lo posible

Seguir así

Intentar empezar antes las entregas para no lidiar con errores a ultima hora



La idea más importante destacada fue: “Intentar empezar antes las entregas para tener tiempo de revisión”

Keep

En esta sección, destacamos las cosas que hicimos bien en esta entrega y que deberíamos seguir haciendo en el futuro.

Pregunta: ¿Qué cosas hicimos bien en esta entrega que deberíamos seguir haciendo en el futuro?

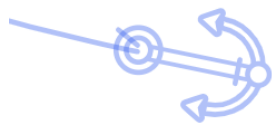


La idea más importante destacada fue: “Buena organización y comunicación”

Improve

Finalmente, identificamos las cosas que hicimos relativamente bien en esta entrega, pero que podríamos mejorar.

Pregunta: ¿Qué cosas hicimos relativamente bien en esta entrega, pero podríamos mejorar?



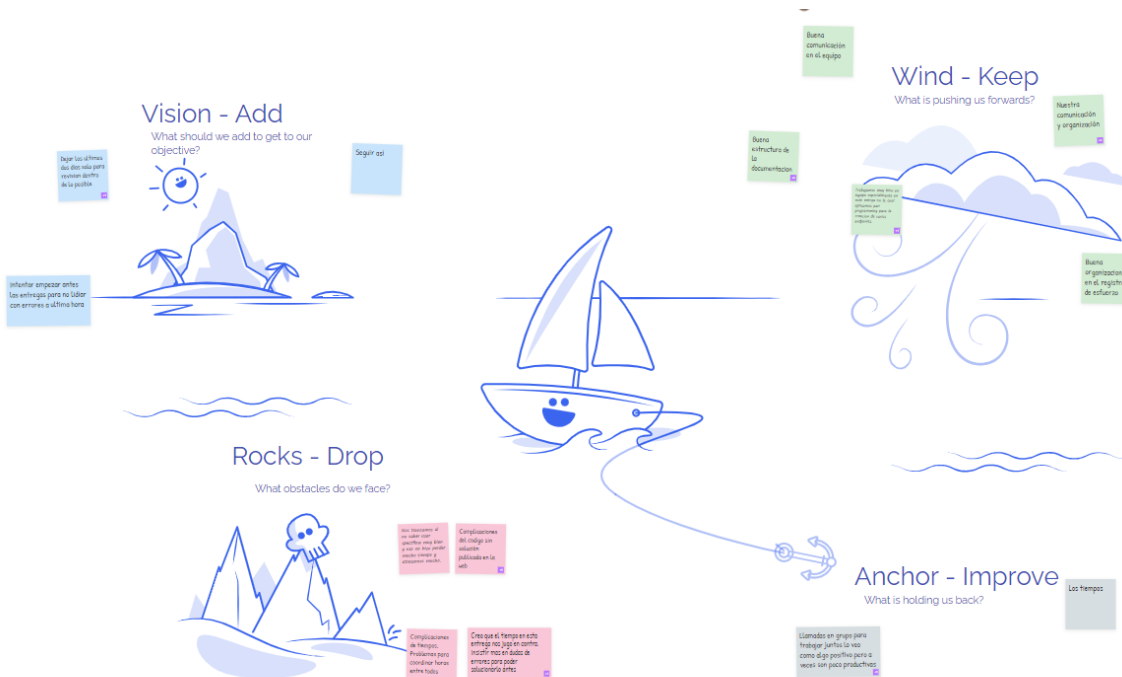
Anchor - Improve

What is holding us back?

Los tiempos

Llamadas en grupo para trabajar juntos lo veo como algo positivo pero a veces son poco productivas

La idea más importante destacada fue: “Llamadas largas en grupo mejor organizadas”



Registro de avance

Registro del trabajo y Esfuerzo

Nro sub tarea	Tipo	Descripción	Horas/Esfuerzo persona	Responsable	Sprint Backlog
1	Tarea	Creación de tercera versión del tablero	0:30:00	Romina	9/10/2023
2	Tarea	Ingreso de tareas al tablero de KANBAN	1:00:00	Todos	9/10/2023
3	Tarea	Actualización del proceso de ingeniería	0:30:00	Romina	11/10/2023
4	Tarea	Configuración del pipeline (github action)	0:30:00	Andy	11/10/2023
5	Tarea	Guía de explicación del tablero	0:30:00	Sofia	11/10/2023
6	User Story	User Story 1 - Backend y Frontend	5:00:00	Todos*	11/10/2023
7	User Story	User Story 2 - Backend y Frontend	3:30:00	Sofia	11/10/2023
8	User Story	User Story 3 - Backend y Frontend	10:00:00	Romina	11/10/2023
9	User Story	User Story 4 - Backend y Frontend		Andy	11/10/2023
			21:30:00		

Requirement Definition	Test Case Implementation	APP Implementation	Refactoring	Integration Testing	Done	Cycle Time (dias)	Lead Time (dias)
9/10/2023	9/10/2023	9/10/2023	9/10/2023	9/10/2023	9/10/2023	0.00	0.00
11/10/2023	20/10/2023	20/10/2023	20/10/2023	20/10/2023	23/10/2023	12.00	14.00
12/10/2023	12/10/2023	12/10/2023	12/10/2023	12/10/2023	12/10/2023	0.00	1.00
23/10/2023	23/10/2023	23/10/2023	23/10/2023	23/10/2023	23/10/2023	0.00	12.00
20/10/2023	20/10/2023	20/10/2023	20/10/2023	20/10/2023	20/10/2023	0.00	9.00
14/10/2023	14/10/2023	22/10/2023	22/10/2023	22/10/2023	22/10/23	8.00	11.00
14/10/2023	14/10/2023	23/10/2023	23/10/2023	23/10/2023	23/10/2023	9.00	12.00
14/10/2023	14/10/2023	22/10/2023	22/10/2023	22/10/2023	22/10/2023	8.00	11.00
14/10/2023	14/10/2023	23/10/2023	23/10/2023	23/10/2023	23/10/2023	9.00	12.00

Para esta entrega contamos con 9 tareas y el total del registro de esfuerzo fue de 21 horas con 30 minutos.

Podrá ver el seguimiento realizado [aquí](#).

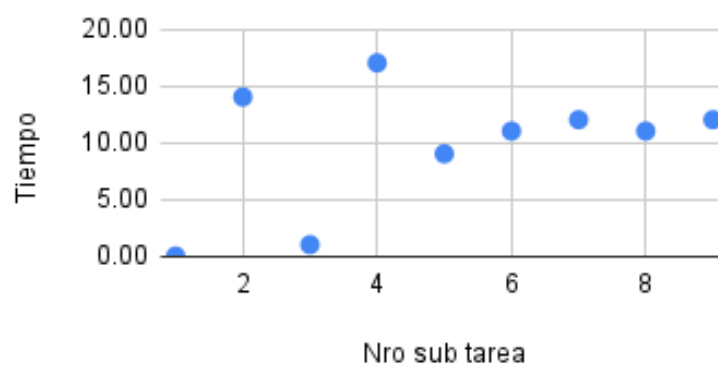
Métricas

Kanban system lead time histogram



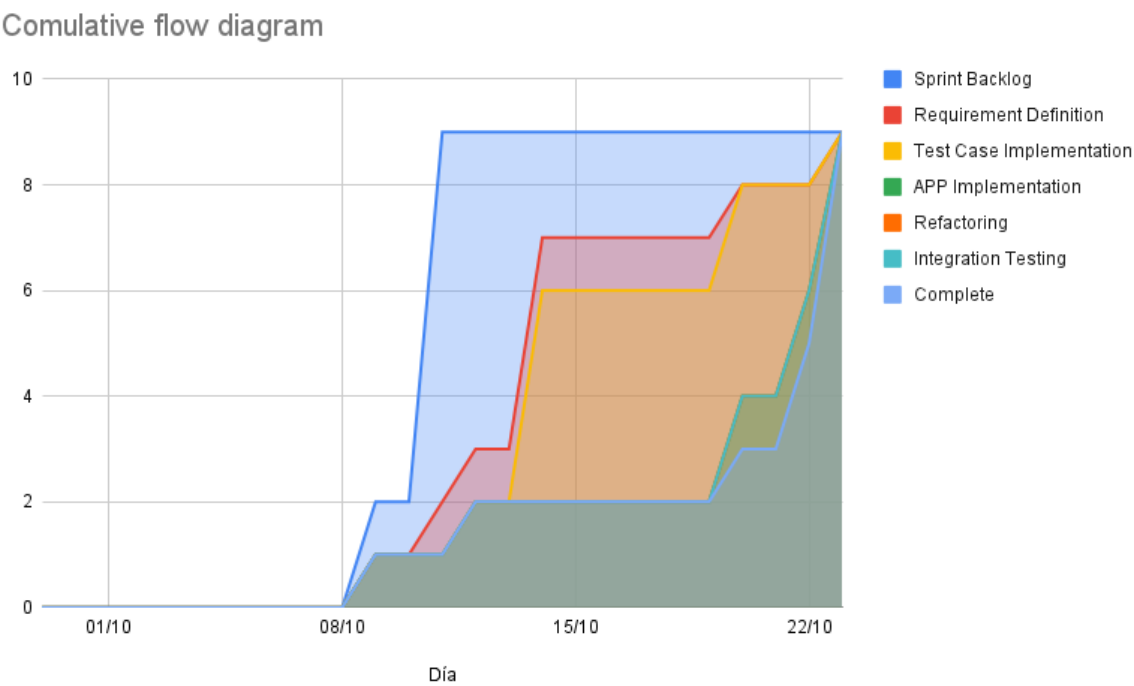
Esta gráfica mide cuántas tareas se terminaron por día (azul) y el porcentaje del proyecto que teníamos terminado hasta ese momento (rojo). Se puede ver que logramos completar la totalidad del trabajo que nos propusimos a tiempo. También podemos ver que los días que más trabajamos, en realidad qué más tareas terminamos fueron los días 22/10 y el 23/10.

Kanban system lead time run Chart



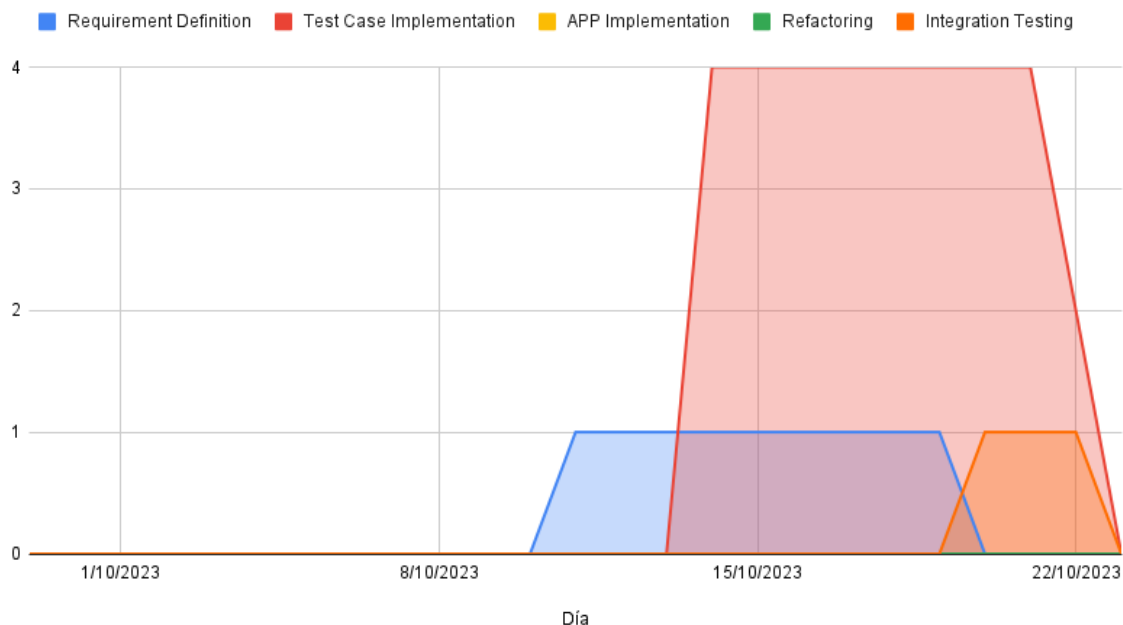
En esta gráfica se puede observar el tiempo que llevó cada tarea/user story desde que se definió hasta estar completa (pasar de Sprint Backlog a Done). Se puede ver cómo en promedio las tareas demoran 9 días estar prontas. También se pueden ver algunos outliers por ejemplo la tarea 7 que nos tomó 12 días tenerla pronta. Esto no quiere decir que dicha tarea nos tomó 12 días realizarla sino que puede ser que dicha tarea estuvo 3 días en el Sprint Backlog sin que nadie la toque y luego demoró otros 9 días en

terminarse. Para saber cuánto nos tomó realizar dicha tarea uno debe ver el cycle time (tiempo in progress, o sea desde design hasta Done) y no el lead time.



Se ve la acumulación de cuantas tareas llegaron a cada columna. Si se desea saber cuántas tareas estaban en tal columna tal día, esto se puede hacer calculando la diferencia vertical en ese día entre una gráfica y su sucesora. Ejemplo si deseamos saber cuántas cards (user stories o tareas) estaban en la columna “Test Case Implementation” el día 15/10 simplemente le restamos al valor de la curva amarilla (Test Case Implementation), 6, el valor de la curva celeste (Complete), 2. Realizando la cuenta podemos ver que el día 15/10 había 4 cards en la columna Test Case Implementation, igual a lo que nos dice la gráfica siguiente (WIP) para ese día. También se puede observar que no hay línea naranja (Refactoring) ni verde (App Implementation) esto se debe a que está debajo de la línea verde (Integration Testing) y por esta razón no es visible. Esto sucede porque ninguna tarea quedó en Refactoring, ni en App Implementation por un día o más. Esto se puede ver mucho más claramente en la gráfica a continuación.

WIP



En esta gráfica se podrá observar las tareas y/o user stories que se encontraban en cada columna en cada día. Se puede observar que en las columnas Refactoring y App Implementation al finalizar el día las tareas que ahí se encontraban se habían movido a Integration Testing. Es por esta razón que ningún día se terminó con tareas ni en Refactoring o App Implementation, por lo cual se puede ver que la curva verde y amarilla están siempre en 0.

Resultados obtenidos en el período

El objetivo de esta entrega, y sus resultados, fue agregar 4 nuevas funcionalidades al sistema utilizando SpecFlow. A su vez, se implementaron las métricas las cuales nos dan una representación de lo realizado.

Dificultades encontradas y formas de solución

Las dificultades encontradas en esta fue la creación de las pruebas de specFlow, ya que se encontraron errores los cuales costaron un poco resolverlos. Su solución fue realizada por ayuda de los profesores.

Lecciones aprendidas y mejoras en el proceso

Aprendimos a utilizar specFlow, a su vez de la implementación del frontend y backend a través de APIs. Que no habíamos trabajado en la entrega pasada.

Con respecto a la entrega anterior

Con respecto a la entrega anterior, en esta entrega hemos observado un avance significativo en términos de funcionalidad y conocimiento adquirido. Mientras que en la entrega previa no teníamos experiencia con specFlow ni con la implementación de frontend y backend a través de APIs, en esta entrega logramos agregar tres nuevas funcionalidades al sistema utilizando specFlow y trabajar en la implementación tanto del frontend como del backend con APIs.

Además, hemos enfrentado dificultades durante este proceso, particularmente en la creación de las pruebas de specFlow, encontrando errores que nos tomaron tiempo resolver.

Como dijimos en la retrospectiva pasada, que uno de los improves que queríamos realizar era la mejora del esquema de commits, quiero aclarar que esta mejora la logramos implementar en esta entrega, ya que definimos un estilo para que los 3 commitiemos de manera más uniforme, así facilitando leer el historial de cambios en un futuro si se quiere.

merge	andresWilchinski committed 5 hours ago	8ba09f2	<>
Merge pull request #26 from IngSoft-ISA2-2023-2/feature/backend/delet...	sofimontero committed 7 hours ago	18b0276	<>
feat: add front end	andresWilchinski committed 20 hours ago	f492f03	<>
feat: backend buy product	andresWilchinski committed 20 hours ago	70d8574	<>
fix: error db update	RomiPardo committed yesterday	78e2695	<>
feat: backend delete product	sofimontero committed yesterday	246af9d	<>

Commits on Oct 22, 2023

Merge pull request #25 from IngSoft-ISA2-2023-2/feature/frontend/edit...	RomiPardo committed yesterday	9e2d3c2	<>
feat: frontedn edit product	RomiPardo committed yesterday	1c42d59	<>
Merge branch 'main' into feature/frontend/editProduct	RomiPardo committed yesterday	5735413	<>
Merge pull request #24 from IngSoft-ISA2-2023-2/feature/frontend/dele...	RomiPardo committed yesterday	94cd876	<>
feat: frontend delete product	RomiPardo committed yesterday	7858751	<>
feat: first version front update product	RomiPardo committed yesterday	b4b54ec	<>
Merge pull request #23 from IngSoft-ISA2-2023-2/feature/frontend/crea...	RomiPardo committed yesterday	ba05254	<>
fix: create product angular	RomiPardo committed yesterday	4988069	<>

Al igual que en la retro pasada dijimos de mantener la buena organización y en esta retro también mencionamos mantener nuestra buena organización y comunicación, lo cual quiere decir que vamos en buen camino. También en la retro pasada mencionamos como mejora tratar de acordarnos de registrar los esfuerzos cuando terminamos una tarea, en esta entrega logramos realizarlo de manera efectiva.