



Gestión de Proyectos **SISTEMAS DE INFORMACIÓN**

Carlos Nicolás Andicoberry López, Alejandro Gómez Amaro, Noelia Gómez Huertas , Natalia Garrido Rey y Paula López Villalba

Usuario: p32goama@uco.com



Contraseña: informatica32.

ÍNDICE

EJERCICIO 1. Creación del tablero en Trello	pág. 3
EJERCICIO 2. Diseño del proyecto	pág. 11
Fases del proyecto	pág. 13
EJERCICIO 3. Estimación Scrum	pág. 18
Power-Ups Trello	pág. 19
EJERCICIO 4. Desarrollo de una aplicación en Taiga	pág. 21
Desarrollo y creación en Taiga	pág. 22
Daily Scrum	pág. 30
EJERCICIO 5. Planificación del proyecto	pág. 34
Comparativa Trello vs. Taiga	pág. 39

EJERCICIO 1. Creación del tablero en Trello

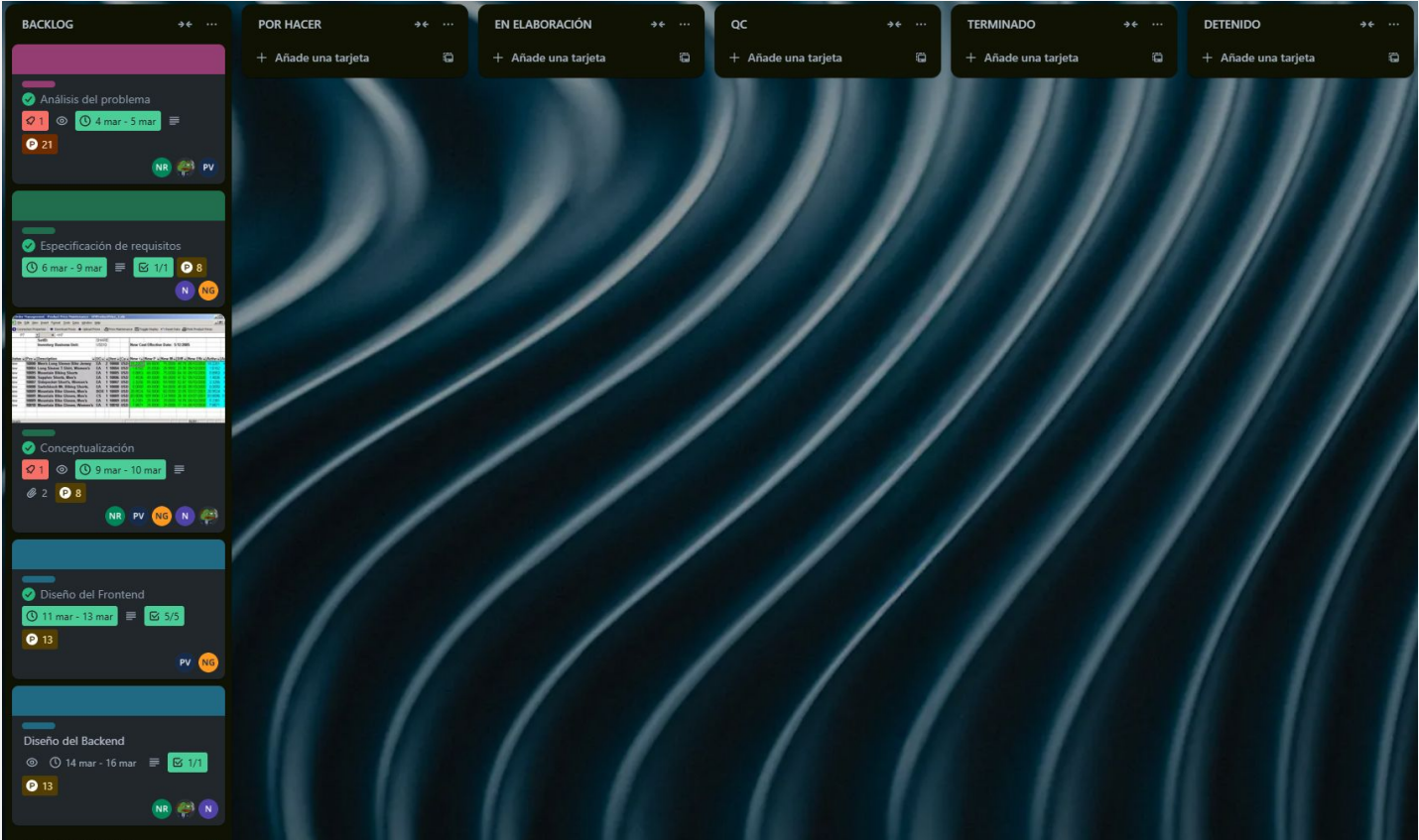
Crea un tablero usando Trello para tu empresa, con el nombre 'Trabajo de SIF' que tenga asignado el grupo de trabajadores 'Empleados' en el cual haya 6 listas de trabajos separadas. Las listas serán nombradas como:

- **Backlog:** Aquí es donde se mantiene el Backlog de tareas para trabajar.
- **Por hacer:** Cuando se planifica el Sprint, saco tareas del Backlog a esta lista. Este es el Sprint real en el que se está trabajando.
- **En elaboración:** Cuando una tarea ya está empezada, se mueve aquí.
- **QC:** Verificación de calidad (Quality check). A medida que las tareas se completan, se mueven a "QC". Al final de la semana, se revisa esta lista para asegurarnos de que todo cumple con la calidad esperada.
- **Terminado:** Después de la verificación de calidad, ponemos en este listado las tareas ya terminadas y verificadas.
- **Detenido:** Cuando algo impide completar una tarea (puede que primero tenga que comprar algo y se necesita que el jefe lo apruebe), se mueva Detenido (blocked), junto con un comentario sobre qué cosa lo está bloqueando. En la lista backlog o tareas por hacer añada las siguientes tareas que conforman el sprint: Análisis del problema, especificación de requisitos, conceptualización, diseño del Frontend, diseño del Backend.

NOTA: Se deberá usar la configuración de las diferentes etiquetas, para diferenciar y personalizar éstas dependiendo de la finalidad de cada una.

EJERCICIO 1

Creamos un tablero al que nos unimos todos los miembros del equipo. En el creamos las 6 listas mencionadas anteriormente y comenzamos a desarrollar el backlog.



EJERCICIO 1

Descripción de las tareas iniciales del backlog:

Análisis del problema:

Descripción del problema: Una tienda necesita tener un control preciso de su inventario para evitar desabastecimientos o excesos de productos, lo que afecta la eficiencia y las ganancias. El proceso manual de gestión de inventario es propenso a errores y consume mucho tiempo. Además, es crucial tener una forma de consultar rápidamente la disponibilidad de productos y recibir alertas cuando el stock esté bajo.

Objetivo del sistema: El objetivo principal es crear un sistema digital que permita gestionar el inventario de forma rápida y sencilla, que permita:

1. Añadir, editar y eliminar productos.
2. Ver el inventario en tiempo real.
3. Recibir alertas cuando el stock sea bajo.
4. Generar informes básicos de productos.

EJERCICIO 1

Especificación de requisitos:

Requisitos Funcionales:

1. Añadir productos: El usuario debe ser capaz de ingresar productos al inventario con su nombre, cantidad y precio.
2. Editar productos: El usuario podrá modificar la cantidad o precio de los productos existentes.
3. Eliminar productos: El sistema debe permitir la eliminación de productos del inventario.
4. Mostrar inventario: Debe haber una vista donde se muestren todos los productos con su nombre, cantidad y precio.
5. Alerta de bajo stock: El sistema notificará cuando un producto tenga una cantidad inferior a un umbral establecido (por ejemplo, menos de 5 unidades).
6. Búsqueda y filtrado: El sistema debe permitir buscar productos por nombre o filtrar por cantidad (por ejemplo, "mostrar productos con stock bajo").
7. Generar reportes: Se podrá exportar el inventario actual en formato CSV o PDF.

Requisitos No Funcionales:

8. Usabilidad: El sistema debe ser fácil de usar y entender para personas sin experiencia técnica.
9. Escalabilidad: El sistema debe permitir añadir más productos a medida que la tienda crezca.
10. Rendimiento: El sistema debe cargar y procesar información rápidamente, incluso con grandes cantidades de productos.
11. Seguridad: El sistema debe tener un control básico de acceso para que solo usuarios autorizados puedan modificar el inventario.

EJERCICIO 1

Conceptualización:

Usuarios del sistema:

- Administrador: Tiene acceso completo para añadir, editar, eliminar productos y generar reportes.
- Empleado: Puede ver el inventario y agregar productos, pero no modificar ni eliminar nada.

Flujo básico de uso:

1. El administrador inicia sesión en el sistema.
2. Accede a la sección de Inventario.
3. Puede añadir productos, modificarlos o eliminarlos según sea necesario.
4. El sistema muestra el inventario actualizado.
5. Si un producto tiene bajo stock, aparece una alerta.
6. El administrador puede generar un reporte en formato CSV o PDF.

Interfaz básica:

- Una barra de navegación para acceder a las diferentes funcionalidades: Agregar Producto, Ver Inventario, Reportes.
- Un formulario para añadir nuevos productos con campos como nombre, cantidad y precio.
- Una tabla que muestre todos los productos con la opción de Editar o Eliminar.
- Un área de alertas para los productos con bajo stock.

Order Management - Product Price Maintenance - OProductPrice_Maint

File Edit View Insert Format Tools Data Window Help

Connection Properties Download Prices Upload Prices Price Maintenance Toggle Display Reset Data Print Product Prices

P7

SetID: Inventory Business Unit

SHARE US\$10

New Cost Effective Date: 5/12/2005

Status	Prod	Description	UO	Item	Cu	New t	New P	New M	Diff	New Eff	Active	Act	
New	10000	Men's Long Sleeve Bike Jersey	EA	2	10000	USD	15.2201	65.0000	75.0000	49.78	08/10/2000	15.2201	65
New	10004	Long Sleeve T-Shirt, Women's	EA	1	10004	USD	1.6162	25.0000	29.9900	23.38	05/10/2000	1.6162	25
New	10005	Mountain Biking Shorts	EA	1	10005	USD	0.8953	65.0000	75.0000	64.10	05/10/2000	0.8953	65
New	10006	Supplex Shorts, Men's	EA	1	10006	USD	1.4836	49.0000	55.0000	47.52	05/10/2000	1.4836	49
New	10007	Sidepocket Short's, Women's	EA	1	10007	USD	2.3256	55.0000	59.9900	52.67	05/10/2000	2.3256	55
New	10008	Switchback Mt. Biking Shorts	EA	1	10008	USD	0.0000	49.0000	64.0000	49.00	05/10/2000	0.0000	49
New	10009	Mountain Bike Gloves, Men's	BOX	1	10009	USD	20.9524	54.0000	60.0000	33.05	03/27/2001	20.9524	54
New	10010	Mountain Bike Gloves, Men's	CS	1	10010	USD	83.8096	109.9900	124.9900	19.18	03/27/2001	83.8096	109
New	10009	Mountain Bike Gloves, Men's	EA	1	10009	USD	5.2381	25.0000	29.0000	26.76	05/10/2000	5.2381	25
New	10010	Mountain Bike Gloves, Women's	EA	1	10010	USD	7.8571	25.0000	29.0000	17.14	05/10/2000	7.8571	25

Ready

NUM

EJERCICIO 1

Diseño del Frontend:

¿Qué es el frontend? El frontend es la parte del sistema con la que los usuarios interactúan directamente. Incluye todo lo que el usuario ve y con lo que interactúa en la interfaz gráfica, como botones, formularios y tablas.

Diseño del Frontend:

1. Página de inicio:
 - Barra de navegación con botones para acceder a "Inventario", "Añadir Producto", "Reportes", etc.
 - Un dashboard que muestre el número total de productos y alertas de productos con bajo stock.
2. Formulario para añadir productos:
 - Un formulario con campos de entrada para nombre, cantidad y precio del producto.
 - Un botón para guardar el producto.
3. Vista de inventario:
 - Una tabla que muestre todos los productos en el inventario.
 - Cada fila tiene botones para Editar y Eliminar el producto.
 - Opcionalmente, un filtro para ver solo productos con bajo stock.
4. Alertas:
 - Los productos con stock bajo se mostrarán en color rojo o con un icono de alerta junto a su nombre en la tabla.
5. Estilo visual:
 - Utilizar un diseño limpio y profesional, con colores como azul o verde para transmitir confianza.
 - Usar frameworks como Bootstrap o Tailwind CSS para hacer que el diseño sea responsivo y se vea bien en diferentes dispositivos.

EJERCICIO 1

Diseño del Backend:

¿Qué es el backend? El backend es la parte del sistema que se encarga de la lógica de negocio, la gestión de datos y la interacción con la base de datos. El backend maneja la manipulación de la información y se comunica con el frontend para mostrar los datos al usuario.

Diseño del Backend:

1. Base de datos:
 - Tablas necesarias: Una tabla productos con las columnas: id, nombre, cantidad, precio, fecha_agregado.
 - Base de datos simple: Puedes usar SQLite para algo rápido o MySQL si buscas algo más robusto.
2. Lógica de negocio:
 - Añadir producto: El sistema recibirá los datos del formulario del frontend, validará que sean correctos (por ejemplo, que la cantidad sea un número positivo) y los almacenará en la base de datos.
 - Editar producto: Similar a la adición, pero aquí el sistema buscará el producto por su id, actualizará los datos y los guardará de nuevo.
 - Eliminar producto: El sistema buscará el producto por su id y lo eliminará de la base de datos.
 - Alertas de stock bajo: El backend debe verificar el stock de cada producto y enviar las alertas correspondientes.

EJERCICIO 1

3. API RESTful:

- Si deseas que el backend sea escalable y flexible, podrías implementar una API RESTful usando Flask(Python) o Express (Node.js).
- Rutas básicas:
 - GET /productos: Obtiene todos los productos.
 - POST /productos: Añade un nuevo producto.
 - PUT /productos/id: Edita un producto existente.
 - DELETE /productos/id: Elimina un producto.

4. Autenticación y seguridad:

- Si deseas control de acceso, implementa una simple autenticación mediante JWT (JSON Web Tokens) para verificar que solo los administradores puedan modificar el inventario.

EJERCICIO 2. Diseño del proyecto

Ponte de acuerdo con los miembros de tu equipo para crear un proyecto personalizado sobre una idea original, donde utilicéis la mayor cantidad de características de un tablero de Trello.

Entre todas las herramientas de las que disponemos en Trello, las que hemos considerado más interesantes son:

- **Color de Tarea:** A cada tarea se le puede asignar un color para concretar la prioridad o la dificultad de la misma de manera visual.
- **Etiquetas:** Al igual que el color de tarea, se pueden asignar etiquetas de colores para clasificar las tareas.
- **Fecha:** Se puede asignar una fecha de inicio y fin a cada tarea para tener una estimación cronológica.
- **Imágenes:** Se puede añadir imágenes a las tareas para poder abstraer información de manera más visual.
- **Asignación de tareas:** A todas las tareas se le pueden asignar los miembros del equipo de forma que podéis dividir el trabajo sin necesidad de aplicaciones externas a Trello.
- **Descripción:** Dentro de cada tarea se puede adjuntar una descripción donde poner notas informativas, ayudas, que se busca en esa tarea o la tarea desarrollada.
- **Puntos de Complejidad:** Utilizados para medir la velocidad del equipo y la dificultad de una tarea, con la extensión o “power-up” llamada Corrello.

EJERCICIO 2

A la hora de asignar las etiquetas a las distintas tareas del proyecto nos hemos basado el siguiente código de colores y categorías:

- El **color verde** es de documentación, todo lo relacionado con la recogida de información y de datos en bruto para poder aplicar a la idea de proyecto, como por ejemplo final los requisitos que vamos a implementar.
- El **color azul** es de tecnologías y herramientas de programación, representa todo aquello es es la parte técnica del trabajo como el diseño del frontend y del backend.
- El **color rosa** es de planificación y organización, directamente proporcional a la definición y montaje de cara al producto, como puede ser el diseño final del proyecto o la verificación de sus funcionalidades.

✕

Etiquetas

+

Modo apto para daltónicos

Mostrar nombres de etiqueta en la parte delantera de la tarjeta

Documentación

Tecnologías y Herramientas de...

Planificación y organizació...

✓

✓

Especificación de requisitos

🕒 6 – 9 mar

☑️ 1/1

NG

N

✓

Diseño del Frontend

🕒 11 – 13 mar

☑️ 5/5

PV

NG

✓


Diseño del Backend

👁️ 🕒 14 – 16 mar

☑️ 1/1

N

NR



✓

Diseño final del proyecto

🕒 13 – 16 mar

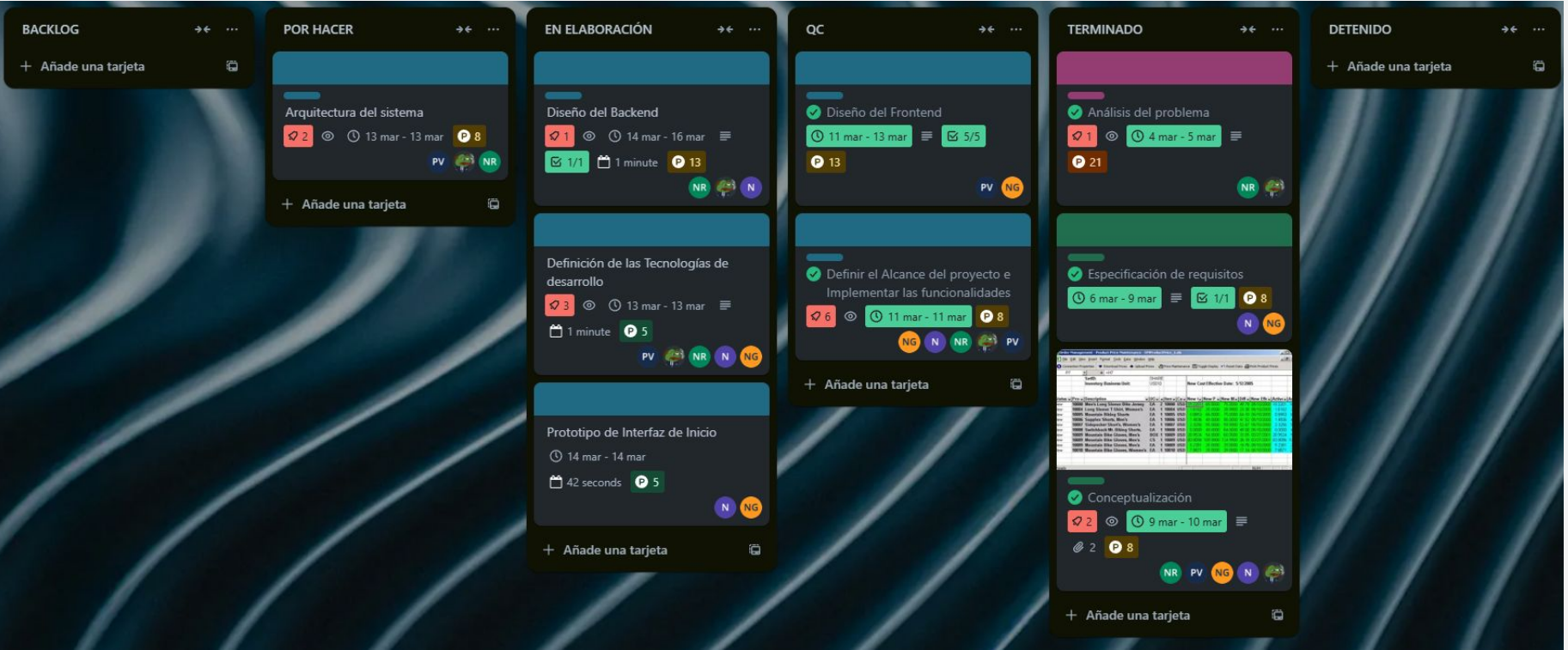
EJERCICIO 2

A partir de estas herramientas hemos desarrollado el trabajo el cual hemos dividido en fases. Al final de cada fase hemos hecho una captura del proyecto para describir cada una de ellas.



FASE 1: En esta etapa, se define el alcance del proyecto, detallando las funcionalidades principales, como la gestión de inventario. Se documentan los requisitos funcionales y no funcionales, y se seleccionan las tecnologías adecuadas para el desarrollo (Frontend: HTML, CSS, JS; Backend: Python/Flask o Node.js; Base de datos: SQLite o MySQL).

EJERCICIO 2

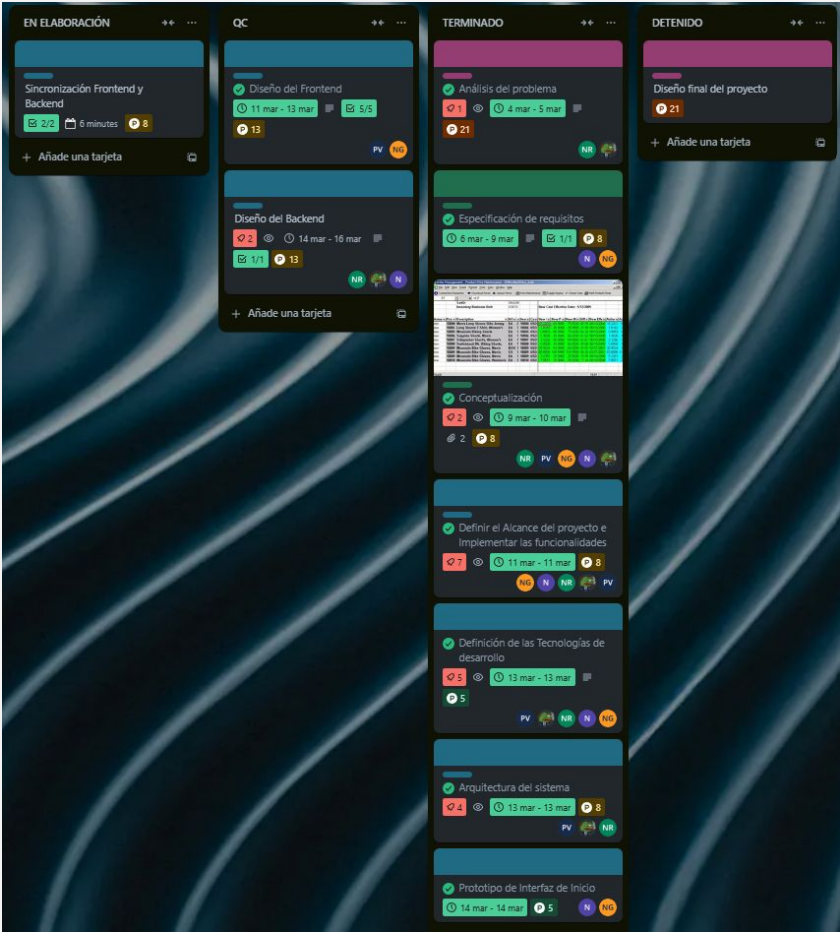


FASE 2: Se diseña la interfaz de usuario mediante wireframes y prototipos, definiendo la apariencia de las pantallas principales. También se diseña la base de datos, estableciendo la estructura de la tabla de productos. Finalmente, se planifica la arquitectura del sistema, organizando las rutas del backend.

EJERCICIO 2

FASE 3: Desarrollo del Frontend (creando la estructura base con HTML. Se desarrollan las vistas clave, como la tabla de inventario y el formulario para añadir productos. Además, se implementan alertas visuales para indicar productos con bajo stock) y Backend (Se inicia el proyecto backend, configurando el entorno y la base de datos. Se desarrollan las rutas para gestionar productos, incluyendo la lógica para agregar, editar, eliminar y mostrar el inventario.)

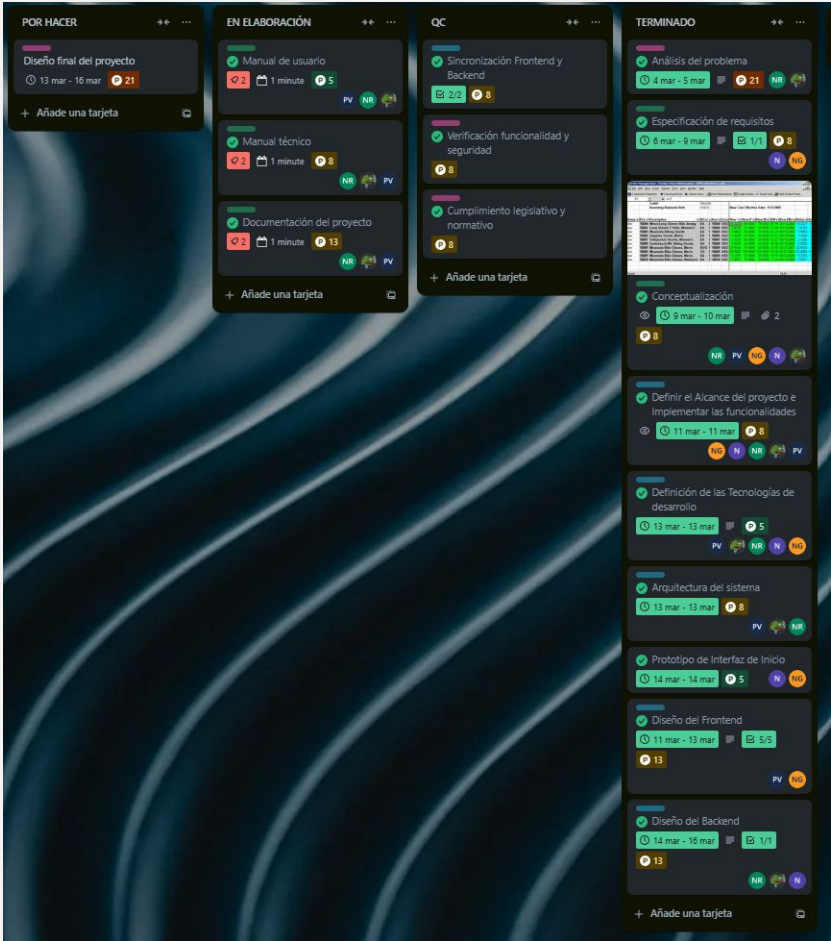
Conectamos el frontend y el backend Se establece la comunicación entre el frontend y el backend utilizando Fetch API o AJAX. Se implementa la funcionalidad para que las solicitudes del frontend actualicen la base de datos y la interfaz refleje los cambios en tiempo real.



EJERCICIO 2

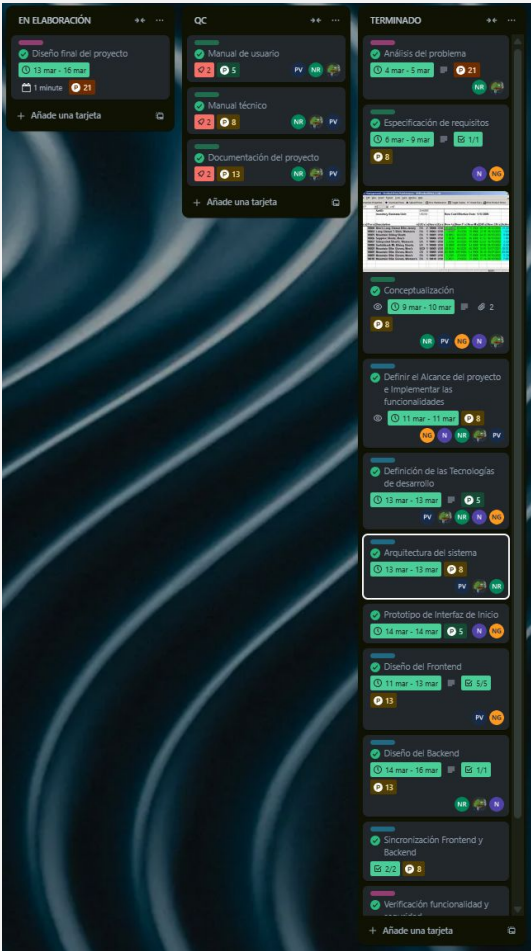
FASE 4: Se realizan pruebas de funcionalidad y seguridad para verificar el correcto funcionamiento del sistema. Se identifican posibles errores y se optimiza la experiencia de usuario mejorando la interfaz y usabilidad.

Se redacta la documentación del proyecto, detallando sus funcionalidades y cómo ejecutarlo. También se genera un informe de pruebas con los resultados obtenidos y posibles mejoras.



EJERCICIO 2

FASE 5: Diseño final del proyecto. En esta etapa se diseña lo que es la interfaz final con todo lo estudiado y comprobado anteriormente. Es la fase más sencilla y compleja a la vez, es la simple porque es aplicar todo lo anterior y es importante porque es el producto final del proyecto.



EJERCICIO 3. Estimación Scrum

Usando el tablero del ejercicio 2, haga una estimación usando la metodología Scrum de lo que va a tardar en las tareas haciendo una estimación de la complejidad en puntos, el consumo final de puntos con Scrum, así como la velocidad del equipo en el sprint.

OPCIONAL: *Investiga otros add-ons para incluir metodología Scrum en Trello, como por ejemplo Screenful y cita algunas diferencias funcionales respecto al tablero Scrum que trae por defecto Trello.*

Para llevar a cabo una correcta planificación del proyecto, se ha realizado una estimación de tiempo para cada una de las tareas. Esta estimación se basa en factores como la complejidad de cada actividad, la cantidad de trabajo involucrado y la disponibilidad del equipo.

Las tareas han sido divididas en diferentes categorías:

- **En elaboración:** Aquí se encuentran tareas en proceso, como el "Diseño final del proyecto", con un rango de fechas estimado entre el 13 y el 16 de marzo.
- **QC (Quality Control):** Incluye revisiones finales como el "Manual de usuario" o la "Documentación del proyecto". Estas tareas son cruciales para garantizar la calidad antes de darlas por finalizadas.
- **Terminado:** Refleja actividades que ya han sido completadas, como el "Análisis del problema" (4-5 de marzo) o la "Especificación de requisitos" (6-9 de marzo). También se encuentran tareas más recientes como la "Definición de Tecnologías de desarrollo" y la "Arquitectura del sistema" (ambas finalizadas el 13 de marzo).

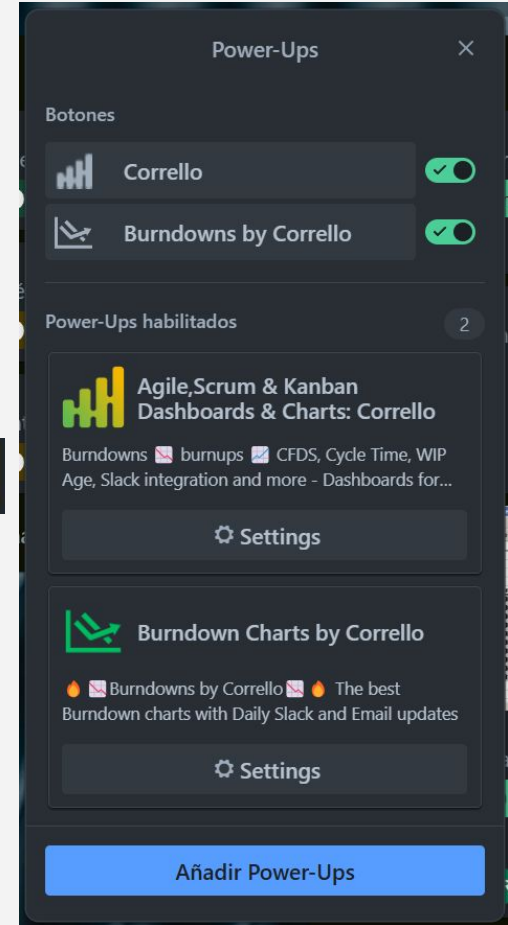
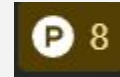
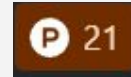
Las fechas asignadas a cada tarea permiten llevar un control preciso del avance y evitar retrasos. Además, la estimación de tiempo se ha ajustado conforme al progreso del equipo, asegurando así que cada fase se complete dentro del plazo esperado.

EJERCICIO 3

Hemos usado dos “Power-Ups”, nombre que usa Trello para llamar a sus extensiones. Estas dos extensiones se complementan ya que nos permiten asignar puntos de complejidad, y además de ello nos genera una gráfica y estadísticas sobre la evolución del proyecto usando métodos ágiles Scrum.

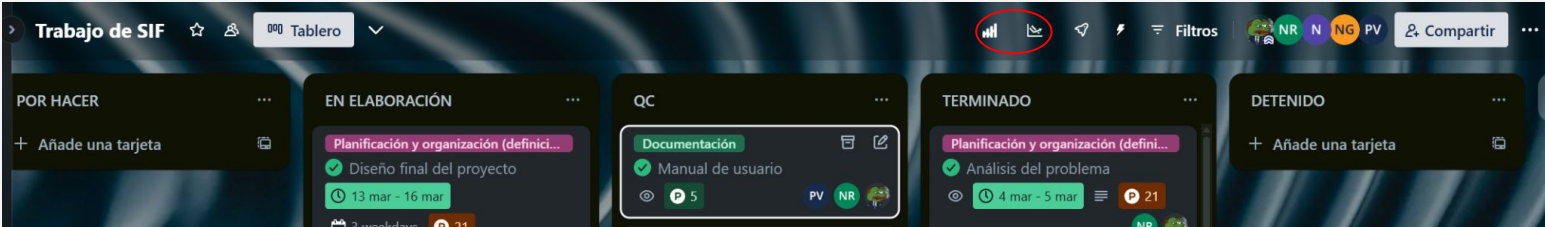
Nos centraremos más en la puntuación de las tareas, como se puede ver en capturas anteriores, hemos dividido las tareas en “4 bloques” dependiendo de su importancia y tiempo necesario para ejecutarlas como son:

- **5 puntos:** Tareas simples o poco necesarias para elaborar el resto del proyecto. Ej: Prototipo de Interfaz de Inicio.
- **8 puntos:** Tareas con un poco más de desarrollo y con algo de importancia. Ej: Verificación funcionalidad y seguridad.
- **13 puntos:** Tareas que necesiten una planificación sobre su desarrollo e importantes. Ej: Diseño del Frontend
- **21 puntos:** Principales tareas del proyecto. Ej: Diseño final del proyecto.

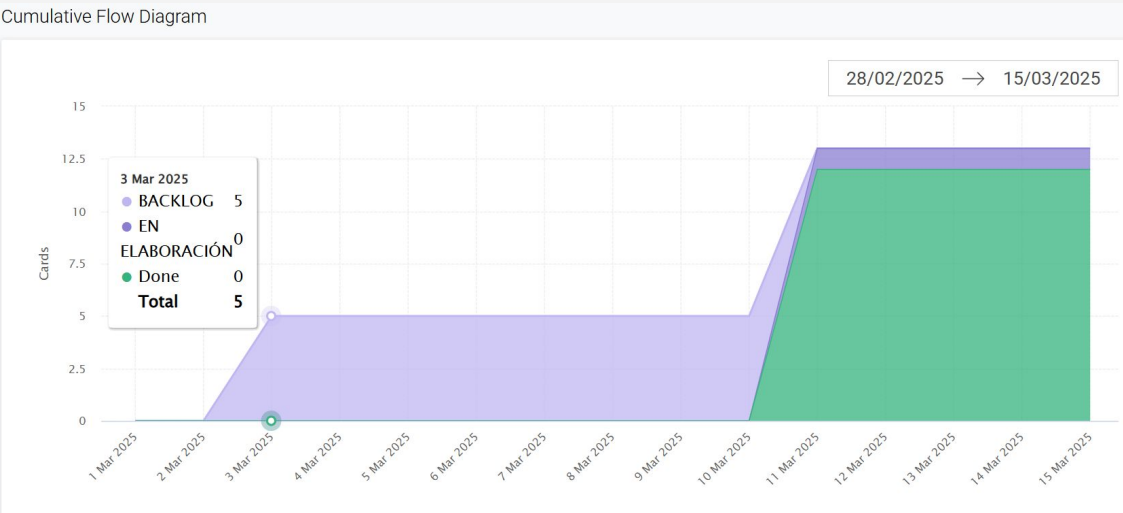


EJERCICIO 3

Podemos acceder a las extensiones desde el menú superior del tablero de Trello.



Al acceder podemos ver las gráficas generadas con respecto a los puntos de complejidad las cuales debido a que la implementación que hemos hecho ha sido secuencial no se han generado como deberían y tenemos y crecimiento inminente de un día a otro (Imagen generada por la extensión Corrello).



EJERCICIO 4. Desarrollo de una aplicación en Taiga

Crea tu primer proyecto utilizando Scrum. Planifica el desarrollo de una aplicación de transporte a través de Taiga.

Crea un Sprint nuevo con el nombre 'Implementación de servicios esenciales', con una duración de siete días, el cual tendrá dos historias de usuario: 1) Diseño de servicios y 2) Métodos de pago en la aplicación, esta Última contará con dos tareas (a libre elección). Crea otro Sprint con el nombre 'Diseñar panel de ajustes'. Éste tendrá dos historias: 1) Personalización de la aplicación y 2) Personalización de informes.

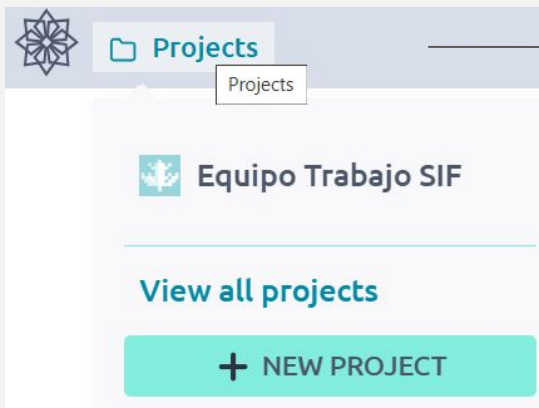
Una vez realizado, asigna una tarea a cada miembro del grupo de trabajo.

Al final de cada día, se realizará un 'Daily Scrum', donde se ratificará la finalización de las tareas de ese día, y se comprobarán las tareas del día siguiente. Se valorará especialmente la actualización diaria del proyecto.

A lo largo del proyecto, se deberá usar el backlog para hacer un estudio diario de la diferencia de la estimación que se ha tenido para Éste, y la duración final que ha tenido. Además, se deberán incluir al menos, dos opciones 'locaina'.

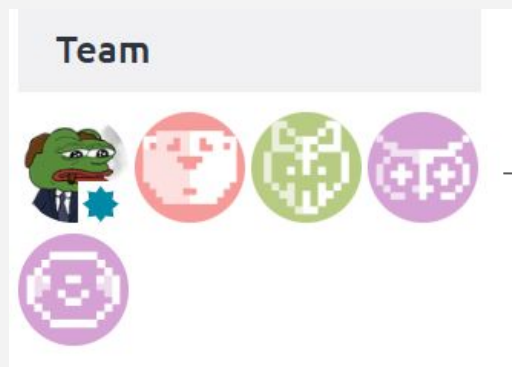
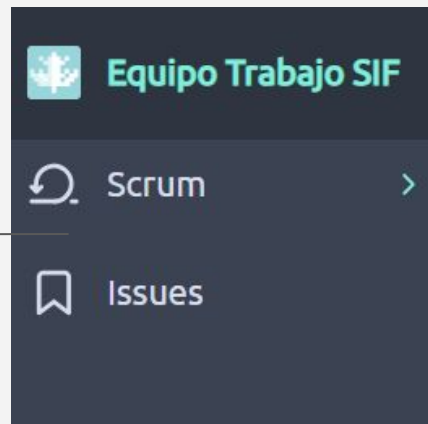
EJERCICIO 4

Diseño y creación en Taiga



Interfaz para la creación de un proyecto / selección de un proyecto ya creado

Pestaña para realizar el proyecto siguiendo la metodología ágil (SCRUM)



Esquema del equipo mediante sus avatares. Para añadirlos podemos hacerlo mediante un link o directamente introduciendo su usuario/correo

EJERCICIO 4

Creación de las historias de usuario

Título
Ej: “Diseño de servicios”

Etiquetas
Ej: “Documentación”
Se le puede poner títulos y colores personalizados

Adjuntos
Ej: “Finanzas.xlsx”

completed points 4 open tasks 0 closed tasks **New task** incaine uses

Add tag+

IN PROGRESS

READY

Please add descriptive text to help others better understand this task

0 Attachments

+

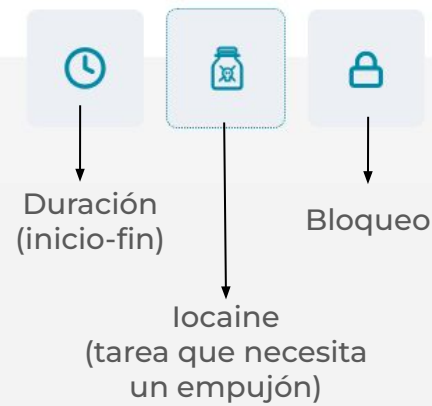
Drop attachments here!

CREATE

New

Assign or Assign to me

Asignación
Ej: “Natalia Garrido Rey”



EJERCICIO 4

Asignación de puntos



USER STORY

^

#1 Diseño de servicios

46 pts

+ +≡

NEW

^

#2 Métodos de pago en la aplicación

13 pts

+ +≡

NEW

^

Storyless tasks

+ +≡

Creación de una tarea de esa historia

completed points 4 open tasks 0 closed tasks

New task

Subject

New

Add tag+

Assign or Assign to me

Please add descriptive text to help others better understand this task

0 Attachments

+

Drop attachments here!

CREATE

EJERCICIO 4

Creación del primer sprint: Implementación de servicios esenciales

Implementación de servicios esenciales04 Mar 2025 to 11 Mar 2025

0%

59 total points

0 completed points

4 open tasks

0 closed tasks

0 localne doses

Filters

subject or referenc

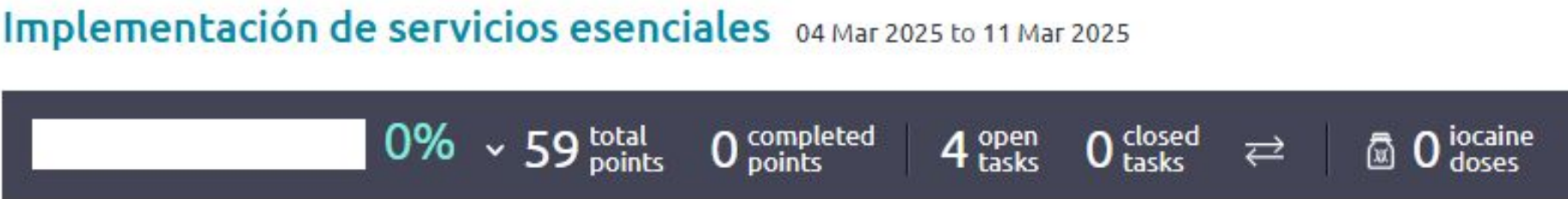
ZOOM:

Detailed

USER STORY	NEW	IN PROGRESS	READY FOR TEST	CLOSED	NEEDS INFO
<div>#1 Diseño de servicios</div> <div>46 pts</div> <div>NEW</div>		<div>#14 Identificación de los servicios que debe ofrecer la aplicación</div> <div></div> <div>#13 Reunión con el equipo o clientes para definir los requisitos.</div> <div></div>			
<div>#2 Métodos de pago en la aplicación</div> <div>13 pts</div> <div>NEW</div>	<div>#8 Crear validaciones para la seguridad del pago.</div> <div></div> <div>#9 Integrar servicios de pago (Ejemplo: PayPal).</div> <div></div>				
<div>Storyless tasks</div>					

EJERCICIO 4

La interfaz muestra detalladamente en la parte superior la duración de dicho Sprint.
De forma esquemática y visual muestra el estado de progreso del proyecto y las “dosis de iocaina”.



A continuación organiza cada tarea, basándose en la pertenencia a cada historia de usuario, agrupa las tareas. Además sigue un criterio según el estado de cada tarea, añadiéndolas en las columnas de: “New”, “In progress”, “Ready for test”, “Closed”, “Needs Info”

USER STORY	NEW	IN PROGRESS	READY FOR TEST	CLOSED	NEEDS INFO
------------	-----	-------------	----------------	--------	------------

EJERCICIO 4

Creación del segundo sprint: Diseñar panel de ajustes

Diseñar panel de ajustes 11 Mar 2025 to 18 Mar 2025

70%

43 total points

30 completed points

1 open tasks

1 closed tasks

1 iocaine doses

Filters

subject or reference

USER STORY	NEW	IN PROGRESS	READY FOR TEST	CLOSED	NEEDS INFO
<div>#3 Personalización de la aplicación</div> <div>30 pts</div> <div>NEW</div>				<div>#10 Crear un sistema de preferencias de usuario.</div>	
<div>#4 Personalización de informes</div> <div>13 pts</div> <div>NEW</div>					<div>#11 Crear un sistema que permita a los usuarios generar informes personalizados.</div>
<div>Storyless tasks</div>					

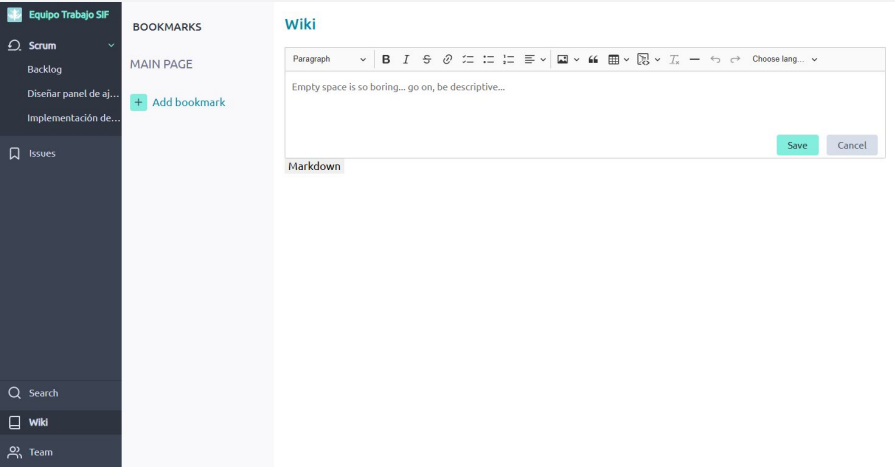
EJERCICIO 4

La interfaz es igual que la del primer sprint, sin embargo, encontramos una diferencia:



Básicamente, si una tarea te parece especialmente difícil o desafiante, puedes marcarla con "iocane" para que el equipo lo sepa. La idea es simbólica: al igual que el personaje de *The Princess Bride* que se vuelve inmune al veneno iocane al exponerse a pequeñas dosis, tú también puedes volverte más fuerte al enfrentarse a tareas complicadas poco a poco. Es una forma ingeniosa de animarte a superar retos y comunicar cuándo algo requiere un esfuerzo extra.

Hemos querido explorar más la aplicación Taiga de lo que el enunciado pedía y hemos creado una **Wiki** del proyecto.



Esto nos ofrece una interfaz para diseñar una presentación clara y organizada con información del proyecto. Podemos crear más páginas y enlazarlas dentro del Wiki para documentar cada aspecto del desarrollo.

EJERCICIO 4

Wiki

🚩 Desarrollo del proyecto
Bienvenidos al proyecto **Desarrollo del Stock Online**. Aquí documentaremos todo lo necesario para el desarrollo del proyecto.

🎯 Objetivo
Crear una herramienta software para que la empresa que la use, sea del sector que sea, pueda personalizarla y aumentar la productividad tanto en la relación con el cliente como con el proveedor.

📄 Características Principales

- ✓ Registro de cuentas con los roles: proveedor, empresa y cliente.
- ✓ Posibilidad de añadir, eliminar y modificar las características de los productos que se venden/compran.
- ✓ Sincronización entre la interfaz de productos del cliente-empresa-proveedor.

🛠️ Tecnologías Utilizadas

Tecnología	Descripción
React Native	Desarrollo de la app móvil
Node.js	Backend API
PostgreSQL	Base de datos

🗺️ Roadmap del Proyecto

- Semana 1:** Diseño y definición de requerimientos
- Semana 2:** Desarrollo del backend y frontend
- Semana 3:** Desarrollo de la interfaz
- Semana 4:** Pruebas y lanzamiento 🚀

📞 Contacto
Para dudas o sugerencias, contacta con **@alejandrogomezamaro** (CEO del proyecto) en Taiga.

4 Attachments			🗖️	☰	+
📎 Requisitos_Proyecto.pdf	Especificación de los requisitos	2.2 KB			
📎 Manual_Usuario.pdf	Manual de Uso para el Usuario	1.6 KB			
📎 Manual_Tecnico.pdf	Manual Técnico de la App	1.7 KB			
📎 Normativa_Estandarizacion.pdf	Cumplimiento de las normas y estándares del Software	2.3 KB			

En la descripción hemos puesto lo relevante de proyecto para tener una idea definida de lo que debemos de hacer y cómo.

También tenemos la opción de agregar documentos a tener en cuenta durante el desarrollo del software (los que hemos añadido son borradores sin información real)

EJERCICIO 4

Hemos decidido realizar dos reuniones semanales debido a que hay pocas tareas y no creemos que sea necesario una actualización diaria. La primera reunión se realiza el martes durante la hora de las clases prácticas y la segunda los viernes de cara al fin de semana. Las reuniones constan de tres partes:

1. Avances respecto a la última reunión.
2. Que voy a realizar hasta la próxima reunión.
3. Dificultades o necesidades que hemos tenido desde la última reunión

DAILY SCRUM 4 Marzo

Como esta fue la reunión inicial planificamos el proyecto en general y nos marcamos unas pautas que seguir. Al final de la reunión concluimos con las siguientes directrices:

- Tanto **Natalia** como **Alejandro** deberían dejar el análisis del problema finalizado en los primeros días.
- **Nicolás** y **Noelia** comenzarían a trabajar en la especificación de requisitos.
- **Paula** por su parte comenzaría a documentarse sobre el proyecto y que todo el desarrollo que estaban realizando los compañeros fue acorde a las directrices dadas.

Quedamos en reunirnos tres días después para ver cómo se estaba desarrollando el trabajo. Además, hemos decidido que el CEO del equipo será Alejandro.

EJERCICIO 4

DAILY SCRUM 7 Marzo

En esta reunión ya comenzamos a usar el modelo de 3 puntos en el que comenzamos con:

Avances:

- **Natalia y Alejandro** con ayuda de Paula y Noelia finalizaron el análisis del problema.
- **Noelia y Nicolás**, como se dijo comenzaron a especificar los requisitos pero avisaron al equipo de que necesitarían ayuda para finalizar la tarea en el plazo acordado.
- **Paula** por su parte nos expuso toda la documentación recaudada.

Tareas:

- **Noelia y Nicolás**, ahora junto a **Natalia** deberían finalizar la especificación de requisitos lo antes posible.
- **Paula y Alejandro** deberían comenzar la conceptualización. Una vez estuvieran finalizados los requisitos, todo el equipo debería acabar teniendo en cuenta dichos requisitos y ciñéndose a las directrices acordadas.

Dificultades:

- A la hora de especificar los requisitos, el equipo tuvo dificultades para definirlos correctamente. Para solucionar el problema se optó por ampliar el equipo de trabajo para de esta forma tener una tercera visión del enunciado del problema que pudiese ayudar a tomar decisiones al respecto.

EJERCICIO 4

DAILY SCRUM 11 Marzo

Seguimos con el proyecto una semana más para entrar en las últimas partes de este.

Avances:

- En una reunión de **todo el grupo** como equipo, se definió el alcance del proyecto y se implementan funcionalidades clave
- Se establecieron las tecnologías de desarrollo a utilizar, hecho por **todos** al tratarse de una decisión conjunta.
- Se avanzó en el diseño del frontend, cubriendo los requisitos planteados, en el que cooperaron **Noelia y Paula**

Tareas:

- Definir el Alcance del proyecto e Implementar las funcionalidades y definición de las Tecnologías de desarrollo, fue una tarea conjunta de **todo el equipo**.
- Arquitectura del sistema, dando pie al diseño del frontend hecho por **Paula, Natalia y Alejandro**.
- Diseño del Frontend, colaboraron en esta parte **Noelia y Paula**.

Dificultades:

- Asegurar que el alcance del proyecto fuera realista dentro del plazo definido.
- Evaluación y selección de tecnologías adecuadas para el desarrollo.
- Coordinación entre los miembros del equipo para estandarizar el diseño del frontend.

EJERCICIO 4

DAILY SCRUM 14 Marzo

Hablamos para planificar el diseño final verificando todos los pasos anteriores.

Avances:

- Se completó el prototipo de la interfaz de inicio, trabajo de **Nicolás y Noelia**.
- Se finalizó el diseño del backend y se comenzó la sincronización con el frontend, de lo que se encargaron **Natalia, Paula y Alejandro**.
- Se avanzó en la documentación del proyecto con manuales de usuario y técnico, todo ello parte del trabajo de **Natalia, Paula y Alejandro**.

Tareas:

- Sincronización frontend y backend, fue realizado por **todos**, uniendo las ideas que se habían tenido por separado de los diseños de frontend y backend.
- Manual técnico, de usuario fue cosa de **Natalia, Paula y Alejandro**.
- Verificación funcionalidad y seguridad, y cumplimiento legislativo y normativo. Todo ello junto la documentación del proyecto.

Dificultades:

- Integración del backend con el frontend sin generar conflictos en la funcionalidad.
- Documentar de manera clara y estructurada para facilitar futuras mejoras y mantenimiento.
- Validar que el diseño del backend cumpliera con los requisitos funcionales y de seguridad.

EJERCICIO 5. Planificación del proyecto

Diseñar la planificación del proyecto que habéis hecho en los puntos 2 y 3, pero en este caso utilizando Taiga Scrum. Valorar las diferencias que habéis encontrado entre ambas aproximaciones con Trello y Taiga y sacad conclusiones de cuál de los dos utilizarías en un proyecto real.

El proyecto se ha dividido en 2 sprints ambos con una duración de 7 días:

SPRINT 1: Implementación de servicios esenciales.

▼

Implementación de servicios esenciales

0 cerrados

04 de marzo de 2025

11 de marzo de 2025

59 en total

#1

Diseño de servicios

46

#2

Métodos de pago en la aplicación

13

TABLERO DE TAREAS DE SPRINT

SPRINT 2: Diseñar panel de ajustes.

▼

Diseñar panel de ajustes

0 cerrados

11 de marzo de 2025

18 de marzo de 2025

43 en total

#3

Personalización de la aplicación

30

#4

Personalización de informes

13

TABLERO DE TAREAS DE SPRINT

EJERCICIO 5

En el primer sprint, nuestro equipo trabajó en la implementación de servicios esenciales, abordando dos historias de usuario clave:

Diseño de servicios (46 puntos)

- Se identificaron los servicios esenciales que debe ofrecer la aplicación.
- Se realizaron reuniones con el equipo y los clientes para establecer los requisitos.
- Se generó una primera estructura del sistema para asegurar su eficiencia.

Métodos de pago en la aplicación (13 puntos)

- Se trabajó en la integración de servicios de pago, asegurando su compatibilidad con la plataforma.
- Se ha añadido la opción de pagar con servicios como PayPal para que los usuarios puedan hacer pagos de forma segura y fácil.

Aunque no se cerraron estas tareas hasta más adelante, se logró un gran progreso en la estructura del sistema.

EJERCICIO 5

En el segundo sprint, nuestro equipo ha trabajado en el diseño del panel de ajustes, abordando dos historias de usuario principales:

Personalización de la aplicación (30 puntos)

- Se ha creado el Manual de usuario y el Manual técnico para que el equipo tenga instrucciones claras sobre cómo usar y configurar las nuevas funciones.
- Se creó la versión final del diseño del proyecto, definiendo cómo se verá y funcionará el panel de ajustes.
- Se diseñó un sistema que permite a los usuarios modificar sus preferencias dentro de la aplicación.







Personalización de informes (13 puntos)

- Se desarrolló un sistema que permite a los usuarios generar informes personalizados según sus necesidades.








EJERCICIO 5

En la parte izquierda del panel de equipo podemos ver los roles asignados a cada miembro del equipo, los roles son:

- **DISEÑADOR:** Encargado de diseñar la arquitectura del software. Este rol lo tiene asignado Natalia Garrido Rey.

	natalia garrido rey Design						2
---	--------------------------------------	---	---	--	---	---	---

- **PRODUCT OWNER:** Creador y dueño del proyecto. Este rol lo tiene asignado Alejandro Gómez Amaro.

	Alejandro Gómez Amaro  Product Owner						2
---	---	---	---	--	---	---	---

- **UX:** Se encarga de realizar las tareas que corresponden a la interacción de los usuarios con el entorno de la aplicación. Este rol lo tiene asignado Carlos Nicolás Andicoberry López.

	Nicolas UX						1
---	----------------------	---	---	--	---	---	---

EJERCICIO 5

- **FRONT:** Encargado de realizar el frontend de la aplicación. Este rol lo tiene asignado Noelia Gómez Huertas.



Noelia Gómez


Front

× Leave this project



1

- **BACK:** Encargado de realizar el backend de la aplicación. Este rol lo tiene asignado Paula López Villalba.



Paula López Villalba

Back



1

EJERCICIO 5

Comparativa Trello vs. Taiga

Trello

Ventajas:

- Simplicidad: Trello es una herramienta intuitiva y fácil de usar para proyectos pequeños.
- Visualización Kanban: Trello tiene un tablero visual de tableros Kanban donde se pueden mover tarjetas (tareas) de una columna a otra fácilmente (por ejemplo: "Por hacer", "En proceso", "Hecho").
- Colaboración: Permite asignar tareas, agregar comentarios y adjuntar archivos.

Desventajas:

- Falta de soporte específico para Scrum: Aunque se pueden crear tableros en Trello para organizar el proyecto Scrum, no tiene funciones específicas como Backlogs, Sprints o historias de usuario de forma estructurada.
- Limitada la gestión de tareas: Para proyectos con muchas tareas y equipos grandes, puede volverse desorganizado rápidamente.
-

Taiga

Ventajas:

- Gestión de Sprints: Taiga permite crear y gestionar Sprints de manera sencilla, creando tareas y llevando un seguimiento de lo que se completa y lo que no.

Desventajas:

- Menos visual que Trello: Aunque Taiga tiene tableros visuales, no tiene la misma simplicidad que Trello.

En resumen:

Taiga es más adecuado para proyectos más complejos y equipos familiarizados con Scrum y la metodología ágil, mientras que Trello puede ser útil para proyectos más sencillos. Taiga ofrece un control más detallado y organizado del proyecto, que dependiendo de nuestras necesidades puede ser más interesante que Trello. En conclusión para realizar un proyecto real escogeríamos Trello por su simplicidad ya que muestra una visualización del trabajo más clara.