

Objetivo: Desarrollar un programa en lenguaje de programación que gestione el inventario de un instituto con las funcionalidades indicadas.

Duración: 4 sprints (4 semanas)

Miembros del equipo: 3 personas

Herramientas:

- **Gestión de tareas:** Trello
- **Control de versiones:** GitHub
- **Documentación:** Documento PDF con las actas de reuniones e instrucciones para ejecutar el programa.

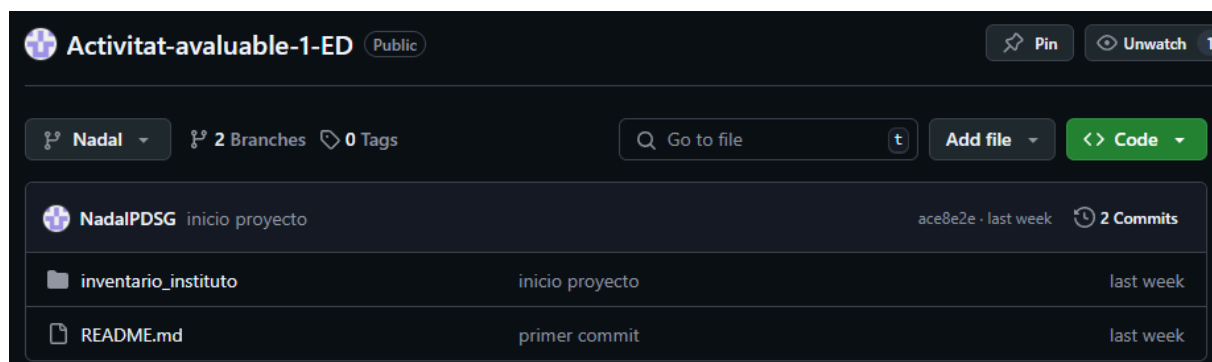
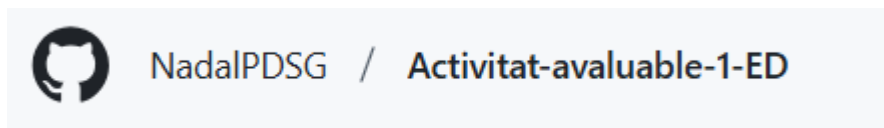
Para gestionar el proyecto, utilizaremos el sistema Kanban con las siguientes columnas:

- **Backlog:** Lista de todas las tareas pendientes
- **To Do:** Tareas planificadas para el sprint actual
- **In Progress:** Tareas que están en proceso
- **Code Review:** Tareas que necesitan ser revisadas (Pull Request en GitHub)
- **Done:** Tareas finalizadas





Sprint 1 (Semana 1): Planificación e inicio del desarrollo

- Creación del repositorio y configuración inicial (branches, README, etc.)



Nadal creara el repositorio:



Tendremos la rama main + nuestras ramas creadas:

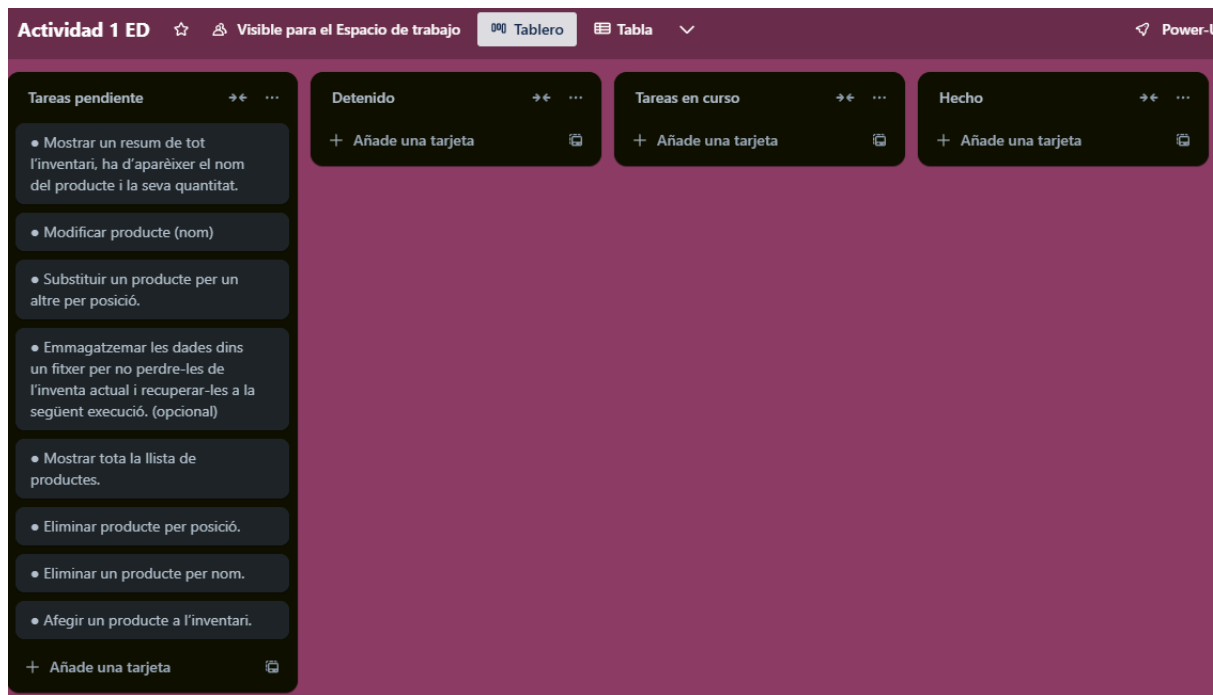
Branch	
main	
Nadal	
Miguel	
JoanMarc	

Añadiremos el documento mediante el README.MD y tendremos el código del programa:

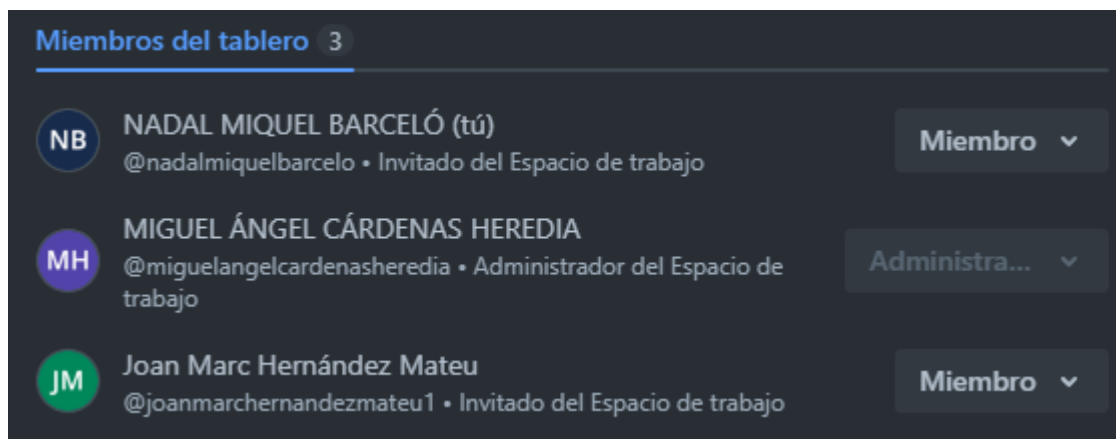
 inventario_instituto
 README.md

- Planificación del proyecto (Trello)

Una vez nos hemos registrado empezaremos nuestro proyecto, con las funcionalidades a implementar:



Miembros del grupo:



- Implementar funcionalidades básicas:

Para las funcionalidades cada uno tendrá una funcionalidad por hacer:

- Nadal: **Eliminar Producto Por Posición**
- Joan Marc: **Mostrar Toda La Lista De Los Productos**
- Miguel Ángel: **Eliminar Un Producto Por Nombre**

Sprint 2 (Semana 2): Desarrollo de funcionalidades principales

Objetivo del Sprint 2: En este sprint, el objetivo principal es implementar las funcionalidades clave del programa que permitirán gestionar el inventario de manera eficiente

Proceso para subir el código a nuestro repositorio:

Cada miembro ha seguido los siguientes pasos para subir su código al repositorio:

1. Clonar el repositorio (sólo la primera vez):

```
$ git clone https://github.com/usuario/repository.git
```

2. Crear una nueva rama para cada uno:

```
$ git checkout -b Miguel
```

3. Desarrollar la funcionalidad correspondiente y realizar commits regularmente:

```
$ git add.  
git commit -m "Funcion realizada correctamente"
```

4. Subir los cambios al repositorio:

```
$ git push origin Activitat-avaluable-1-ED
```

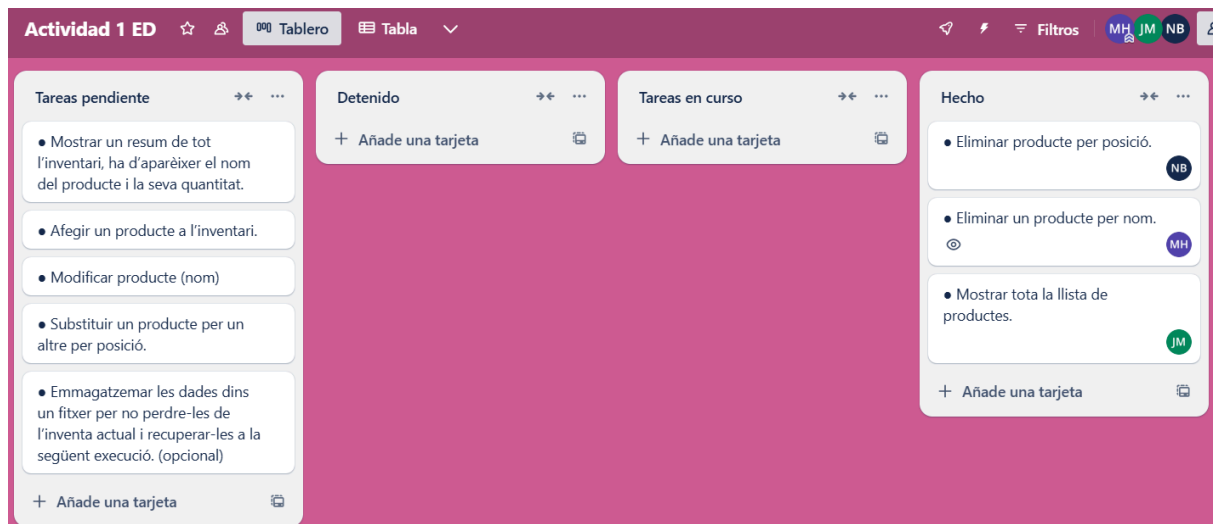
5. Una vez hecho haremos un pull request desde nuestro GitHub, incluyendo a los demás usuarios

6. Después de la fusión, cada miembro debe actualizar su repositorio local:

```
$ git checkout main  
git pull origin main
```

7. Repetir el proceso para cada nueva funcionalidad.

Las funcionalidades ya estarán implementadas, por lo que lo notificaremos en Trello:



Ahora nos enfocaremos en las demás funcionalidades.

Sprint 3 (Semana 3):

Objetivo del Sprint 3: En este sprint nos enfocaremos en implementar las funcionalidades adicionales que faltan para completar el programa de gestión del inventario. Además, comenzaremos a optimizar el código y nos prepararemos para la integración final de todas las funcionalidades.

Funcionalidades a implementar:

1. **Mostrar resumen del inventario, Añadir producto al inventario:**

2. **Modificar nombre de un producto:**

- **Responsable: Nadal**
-

Proceso de trabajo en equipo:

1. **Subida del código al repositorio:**

- Cada miembro deberá continuar utilizando las ramas individuales para implementar las funcionalidades asignadas.
- Después de desarrollar las funcionalidades, realizaremos los commits regularmente para mantener un historial claro de cambios.

```
Nadal@DESKTOP-BHQ13A6 MINGW64 /d/ACTIVITAT (Nadal)
$ git commit -m "Funcion modificar nombre de producto y algunos retoques en otras
s funciones."
[Nadal dddcc69] Funcion modificar nombre de producto y algunos retoques en otras
funciones.
1 file changed, 87 insertions(+), 33 deletions(-)
```

```
Nadal@DESKTOP-BHQ13A6 MINGW64 /d/ACTIVITAT (Nadal)
$ git push origin Nadal
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 16 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 1.18 KiB | 1.18 MiB/s, done.
Total 5 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/NadalPDSG/Activitat-avaluable-1-ED
5ecb159..dddcc69  Nadal -> Nadal
```

```
Nadal@DESKTOP-BHQ13A6 MINGW64 /d/ACTIVITAT (Nadal)
$ git commit -m "Funcion resumen inventario."
[Nadal 0ee3731] Funcion resumen inventario.
1 file changed, 30 insertions(+), 2 deletions(-)

Nadal@DESKTOP-BHQ13A6 MINGW64 /d/ACTIVITAT (Nadal)
$ git push origin Nadal
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 16 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 758 bytes | 758.00 KiB/s, done.
Total 5 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/NadalPDSG/Activitat-avaluable-1-ED
dddcc69..0ee3731  Nadal -> Nadal
```

- Una vez se haya completado la funcionalidad, se hará un pull request desde las ramas de desarrollo a la rama main. El equipo deberá revisar y aprobar las pull requests para asegurarse de que todo el código sea correcto y no haya errores.

2. Actualización de repositorios locales:

- Después de cada fusión de las pull requests, todos los miembros deberán actualizar sus repositorios locales para incorporar los últimos cambios realizados por el resto del equipo.

3. Pruebas y revisión de funcionalidades:

- Una vez implementadas las funcionalidades, cada miembro debe probar el programa de manera individual para verificar que cada una de las nuevas características funcione correctamente.
- Posteriormente, realizaremos una revisión grupal del código para detectar posibles errores y optimizar el funcionamiento del programa.

Sprint 4 (Semana 4) Corrección de errores y optimización

Una vez hemos terminado de añadir las funcionalidades, analizaremos el código para garantizar su máxima optimización.

También terminamos de redactar el documento y lo pasaremos a la preparación de un documento completo

Tareas a realizar:

1. Analisis del codigo:

Responsable: Nadal

Nos aseguramos de que el código está bien implementado para el mejor rendimiento posible.

2. Revisión final y corrección de errores:

Responsables: Miguel Ángel, Joan Marc

Haremos una revisión de cada una de las funcionalidades para asegurar nos de que funcione correctamente.

Probaremos el programa con distintos casos de uso para identificar posibles errores y hacer correcciones en caso necesario.

3. Documentación final

Responsables: Todos

Prepararemos la documentación final del proyecto:

- **Registro de reuniones:** Explicación de las reuniones y las decisiones que se han tomado en cada sprint y sus tareas realizadas.
- **Funcionalidades:** Explicación del código con imágenes.
- **Git:** Explicación del proceso para subir el código a nuestro repositorio

Actividad 1 ED

Visible para el Espacio de trabajo

Tablero

Tabla

Power

Tareas pendiente

• Emmagatzemar les dades dins un fitxer per no perdre-les de l'inventa actual i recuperar-les a la següent execució. (opcional)

+ Añade una tarjeta

Detenido

+ Añade una tarjeta

Tareas en curso

+ Añade una tarjeta

Hecho

• Eliminar producte per posició.

NB

• Eliminar un producte per nom.

MH

• Mostrar tota la llista de productes.

JM

• Substituir un producte per un altre per posició.

JM

• Afegir un producte a l'inventari.

JM

• Modificar producte (nom)

NB

• Mostrar un resum de tot l'inventari, ha d'aparèixer el nom del producte i la seva quantitat.

NB

+ Añade una tarjeta