

# Documentación MADs Equipo 03

## Release 1.4.0

<b>Sprint Backlog</b> .....	2
Historia de Usuario 1:.....	2
Historia de Usuario 2:.....	4
Historia de Usuario 3:.....	5
Historia de Usuario 4:.....	7
Historia de Usuario 5:.....	9
Historia de Usuario 6:.....	10
Historia de Usuario 7:.....	12
<b>Funcionalidades implementadas</b> .....	14
Poder ver el perfil del usuario y editarlo .....	14
Administrador equipo .....	15
Destacar Tareas.....	16
Estado de las tareas.....	17
Comentarios en las tareas .....	18
Calendario de Tareas.....	19
<b>Puesta en producción</b> .....	20
<b>Informe sobre la evolución del desarrollo</b> .....	22
<b>Informe sobre sesiones de pair programming</b> .....	26
Primera sesión: Validaciones de Editar Perfil - Luis y Noel.....	26
Segunda sesión: Drag and Drop (Kanban) - Noel y Jose .....	27
<b>Resultado de la retrospectiva</b> .....	28

# Sprint Backlog

El Sprint Backlog se creó durante la reunión de planificación del Sprint en la primera semana del trabajo, que marcó el inicio del Sprint.

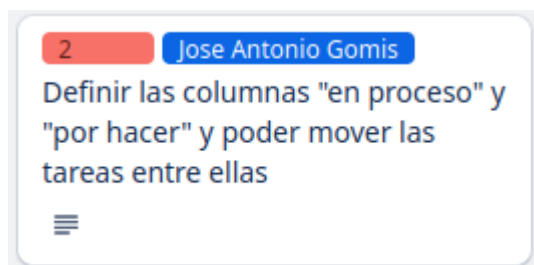
Durante esta reunión revisamos el [tablero Trello](#) con ejemplos de funcionalidades de cursos pasados, del que comunicándonos en equipo ordenamos de mayor a menor importancia y utilidad (a la hora de definir un producto coherente, fácil de usar y que puede representar un *mínimo producto viable*) las funcionalidades entre las columnas de Backlog(1) y Backlog(2) creadas en el Trello, ordenadas de mayor a menor importancia (arriba a la izquierda la más importante y abajo a la derecha la menos).

Una vez ordenadas las funcionalidades en el Backlog, procedimos a comentar y debatir la dificultad de cada una para asignarle 1 o 2 puntos mediante etiquetas en el Trello.

Hecho ya esto, asignamos a cada uno 4 puntos de las funcionalidades más relevantes y que daban más sentido a la aplicación final que íbamos a desarrollar.

Estas son las historias de usuario que hemos desarrollado:

## Historia de Usuario 1:



## Definir las columnas "en proceso" y "por hacer" y poder mover las tareas entre ellas

### Descripción

Como usuario registrado en el sistema, quiero tener una página con un listado de las tareas que tengo y ver en que estado están estas y poder mover de forma cómoda las tareas entre columnas las cuales muestran el estado de la tarea.

## Detalles

- **Sección de Visualización por Columnas:** La interfaz debe presentar claramente tres columnas identificadas como "En Proceso", "Por Hacer" y "Terminadas". Cada columna debe mostrar las tareas correspondientes al estado respectivo.
- **Creación y Edición de Tareas:** Donde se crean las tareas, éstas automáticamente se crearán con el estado, "por hacer". En la edición de tareas también se podrá cambiar el estado de estas.
- **Arrastrar y Soltar:** Los usuarios deben poder seleccionar una tarea haciendo clic en ella y luego arrastrarla hacia otra columna. Al soltar la tarea en una columna diferente, su estado debe actualizarse automáticamente al estado de esa columna.
- **Indicadores Visuales:** Se deben incorporar indicadores visuales para resaltar el estado actual de cada tarea, como colores diferentes o iconos específicos para "En Proceso", "Por Hacer" y "Terminadas".

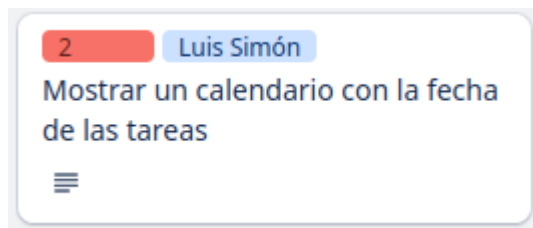
## COS (Condiciones de Aceptación)

1. La opción para ver este listado estará en el menú o un submenú para ver las distintas opciones de listado de tareas.
2. Una vez dentro deben aparecer las tareas en sus respectivas columnas según su estado
3. Se debe de poder mover tareas entre columnas seleccionándolas y arrastrándolas a la columna que nos interese.
4. Si movemos una tarea a otra columna, el estado de esta deberá de cambiar en la base de datos.

## Borrador del aspecto de la interfaz de usuario resultante



## Historia de Usuario 2:



# Mostrar un calendario con la fecha de las tareas

## Descripción

Como usuario de la página, deseo contar con una funcionalidad que me permita visualizar las fechas asociadas a las tareas de manera clara y organizada. Para lograr esto, se propone la implementación de un botón en el apartado tareas que muestre un calendario dedicado. Este calendario deberá mostrar de manera intuitiva las fechas de las tareas del usuario existentes en la plataforma.

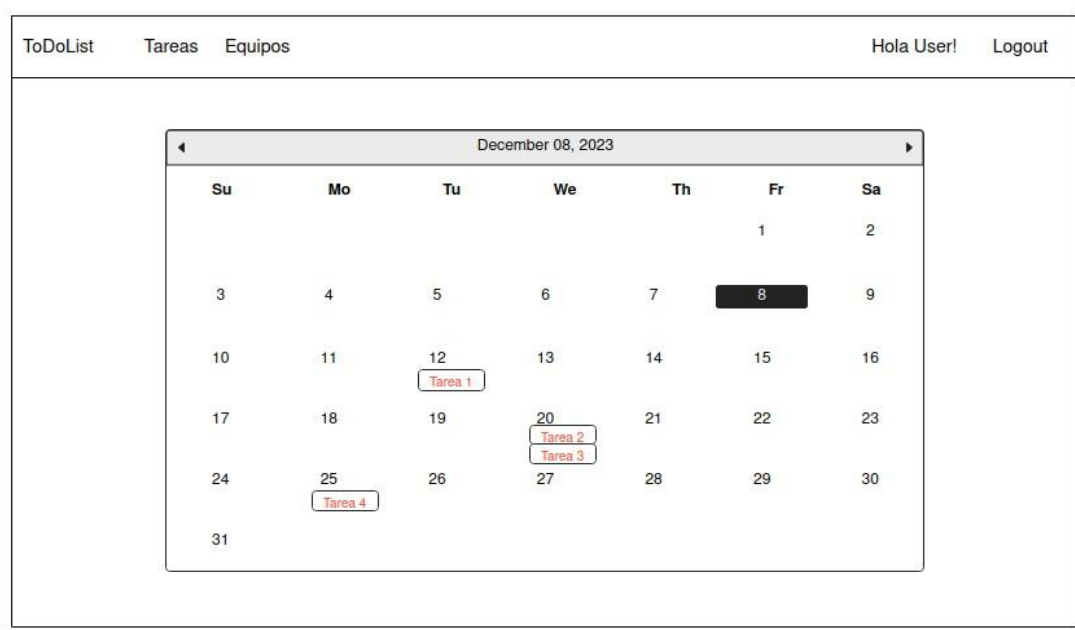
## Detalles

- El botón para mostrar el calendario debe ser fácilmente identificable y ubicado de manera lógica en la interfaz de usuario.
- El calendario debe mostrar todas las fechas relevantes de las tareas.
- Se debe proporcionar un sistema de navegación eficiente en el calendario, permitiendo al usuario cambiar entre días, semanas o meses con facilidad.
- La información mostrada en el calendario debe estar sincronizada en tiempo real con las actualizaciones de las fechas de las tareas en la plataforma.

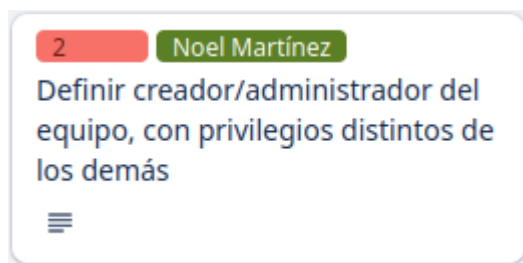
## COS (Condiciones de Aceptación):

1. El botón para mostrar el calendario debe ser implementado y funcional en todas las plataformas admitidas por la aplicación.
2. El calendario debe ser visualmente atractivo y fácil de entender.
3. Todas las fechas mostradas en el calendario deben ser precisas.
4. La navegación en el calendario debe ser intuitiva y permitir al usuario desplazarse fácilmente entre diferentes períodos de tiempo.

## Borrador del aspecto de la interfaz de usuario resultante:



### Historia de Usuario 3:



## Definir creador/administrador del equipo, con privilegios distintos de los demás

### Descripción

Como usuario administrador del sistema, se busca la capacidad de asignar a un creador/administrador del equipo con privilegios especiales, con el objetivo de facilitar una gestión eficiente de tareas y permisos dentro del equipo.

Un usuario podrá ser administrador de un solo equipo y un equipo solo podrá tener un administrador.

## Detalles

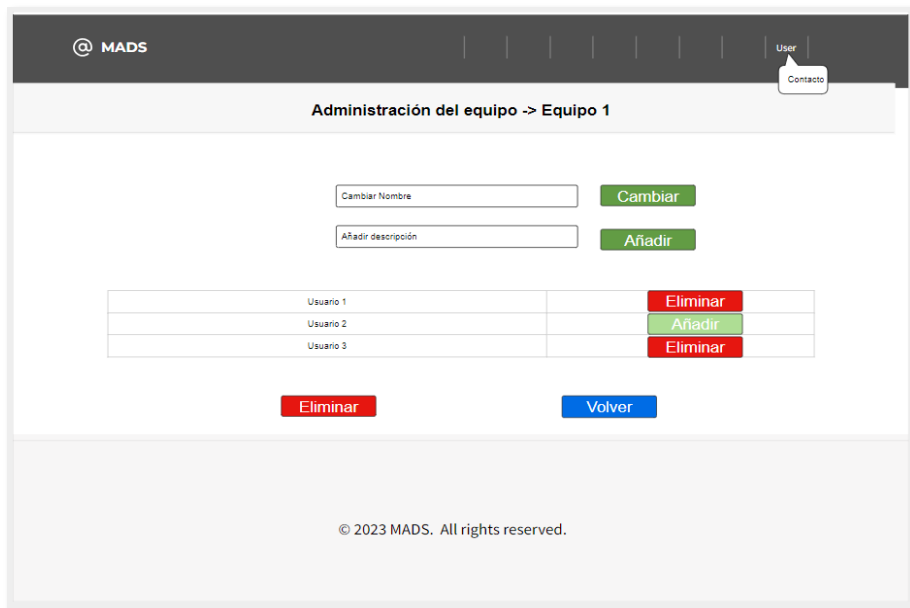
- Para asegurar la efectividad de esta función, se debe incorporar una opción clara en la interfaz de administración que permita la designación de un miembro del equipo como creador/administrador.
- Este individuo designado deberá contar con privilegios exclusivos, tales como la habilidad para agregar o eliminar miembros del equipo, y la capacidad de modificar la configuración del equipo.
- El creador/administrador designado tendrá la capacidad de realizar acciones que otros miembros del equipo no pueden llevar a cabo, garantizando así un control completo sobre la gestión del equipo. Este incluye la modificación de la configuración del equipo, lo que implica un conjunto completo de acciones administrativas.

## Condiciones de Aceptación (COS)

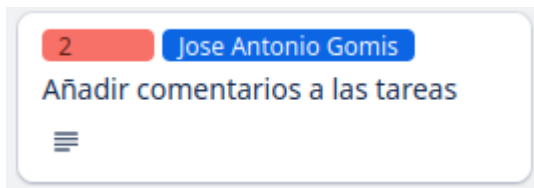
- La función de designar al creador/administrador debe ser fácilmente accesible desde la interfaz de administración del equipo, asegurando una experiencia intuitiva para los usuarios administradores.
- El creador/administrador designado deberá recibir un conjunto claro y específico de privilegios que le permitan gestionar eficientemente el equipo, asegurando la transparencia en sus capacidades y responsabilidades.
- Los cambios realizados por el creador/administrador, como la adición o eliminación de miembros, deben reflejarse de manera precisa y oportuna en la configuración del equipo, manteniendo la coherencia en la información del equipo.

## Borrador del aspecto de la interfaz de usuario resultante

The mockup shows a web interface for team management. At the top is a dark header with the 'MADS' logo on the left and a 'User' profile dropdown on the right. Below the header, the page title is 'Administración del equipo -> Equipo 1'. The main content area contains three input fields with corresponding action buttons: 'Cambiar Nombre' with a green 'Cambiar' button, 'Añadir descripción' with a green 'Añadir' button, and a dropdown menu 'Selecciona el usuario administrador del equipo' with a green 'Asignar' button. At the bottom of the form are two buttons: a red 'Eliminar' button and a blue 'Volver' button. The footer of the page states '© 2023 MADS. All rights reserved.'



## Historia de Usuario 4:



## Añadir a las tareas comentarios

### Descripción

Como usuario registrado en el sistema, quiero poder comentar sobre las tareas que tengo y poder ver los comentarios de las tareas.

### Detalles

- **Sección de de detalles de la tarea:**

Al hacer clic en una tarea en particular, los usuarios serán dirigidos a una vista detallada que proporcionará información más exhaustiva. En esta vista detallada, se mostrarán no solo los detalles esenciales de la tarea, sino también un historial completo de comentarios asociados con la tarea..

- **Comentarios:**

Cada comentario estará claramente etiquetado con el nombre del usuario que lo escribió y la fecha y hora en que se realizó. Esto permitirá un seguimiento transparente de las interacciones relacionadas con la tarea a lo largo del tiempo.

## COS (Condiciones de Aceptación)

1. La opción para ver los detalles de la tarea será pulsando en el título de esta el cual contendrá un enlace y nos llevará ahí.
2. Una vez dentro deben aparecer los detalles de la tarea con sus respectivos comentarios
3. Se debe poder comentar en nuestras tareas apareciendo un recuadro con un botón para enviar los comentarios.
4. Si hacemos un comentario, este se deberá guardar en la bbdd para así poder verlo en el listado de comentarios de cada tarea.

## Borrador del aspecto de la interfaz de usuario resultante

moqups Equipos Tareas Cerrar Sesión

### Tarea: Bajar al perro

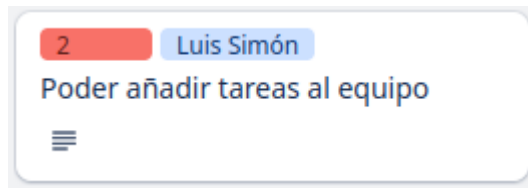
NombreTarea	ID	Estado
FechaLimite		Destacada

Nombre Usuario Fecha comentario

Texto comentario



## Historia de Usuario 5:



# Agregar la posibilidad de añadir tareas a un equipo por parte de un usuario administrador del equipo

## Descripción

Como usuario administrador de equipos en la plataforma, deseo tener la capacidad de asignar tareas a estos. La implementación de esta funcionalidad facilitará la colaboración y la distribución eficiente de responsabilidades dentro del equipo.

## Detalles

- Debe introducirse una opción clara en la interfaz de usuario que permita al usuario administrador del equipo asignar tareas directamente al equipo al que pertenecen.
- Al asignar una tarea a un equipo, esta debe ser visible para todos los miembros del equipo en su lista de tareas pendientes.
- Se debe proporcionar la opción de asignar una tarea a un equipo desde la vista detallada de la tarea o desde una lista específica de tareas.

## COS (Condiciones de Aceptación):

1. La opción para asignar tareas a un equipo debe estar disponible y funcional en todas las plataformas compatibles con la aplicación.
2. Al asignar una tarea a un equipo, esta debe ser visible inmediatamente en la lista de tareas del equipo para todos sus miembros.
3. La interfaz de usuario debe ser intuitiva y fácil de usar, facilitando la asignación de tareas a equipos sin confusiones.

## Borrador del aspecto de la interfaz de usuario resultante:

ToDoList

Administración Equipo

Nuevo Nombre Equipo

Nueva Descripción

Integrante 1	Eliminar
Integrante 2	Eliminar

Tarea2

▼

Agregar Tarea

Tarea 1	Eliminar
---------	----------

## Historia de Usuario 6:

1

Noel Martínez

Poder destacar tareas en el listado

☰

## Poder destacar tareas en el listado

### Descripción

Como usuario, quiero tener la capacidad de destacar ciertas tareas en el listado para resaltar su importancia o prioridad, facilitando así la gestión y organización de mis actividades.

### Detalles

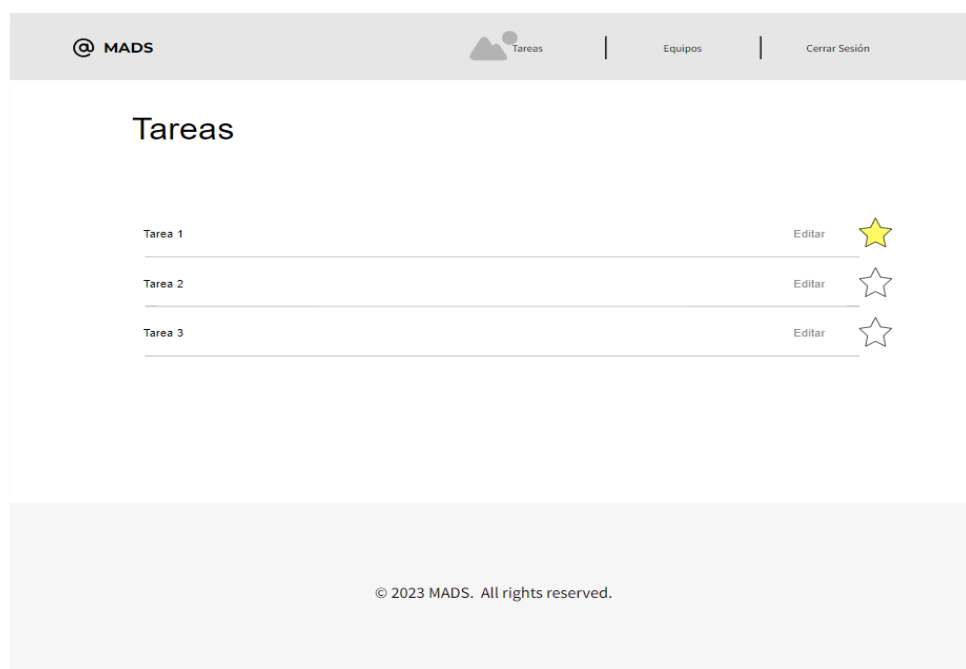
- **Interfaz de Usuario:** Deberá existir una opción clara y fácilmente accesible en la interfaz de usuario principal que permita destacar una tarea. Esto podría lograrse mediante un icono específico junto a cada tarea en el listado.

- **Destacado Visual:** Las tareas destacadas deben ser visualmente distintas en el listado. Pueden resaltarse mediante colores llamativos, iconos especiales o algún otro elemento visual que las diferencie claramente del resto de las tareas no destacadas.
- **Quitar el Destacado:** Además de poder destacar tareas, se debe proporcionar una opción sencilla para quitar el destacado de una tarea. Esto podría realizarse mediante un botón de "Quitar Destacado" visible al lado de cada tarea destacada.
- **Persistencia de Estado:** El estado de destacado de las tareas debe persistir incluso después de cerrar y volver a abrir la aplicación. Esto garantizará que las tareas prioritarias sigan siendo visibles y reconocibles en futuras sesiones de trabajo.

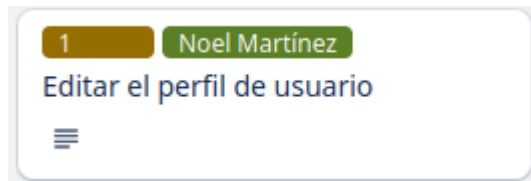
## Condiciones de Aceptación (COS)

- La opción para destacar una tarea debe ser fácilmente accesible desde la interfaz de usuario principal, sin requerir múltiples pasos o búsquedas.
- Una vez que una tarea está destacada, debe mostrarse visualmente de manera clara en el listado para que el usuario pueda identificarla fácilmente. El método de resaltado debe ser distintivo y fácilmente reconocible.
- Debe haber una opción sencilla y directa para quitar el destacado de una tarea si es necesario, sin necesidad de navegar por menús complicados.
- Las tareas destacadas deben mantener su estado incluso después de cerrar y volver a abrir la aplicación. La persistencia del estado de destacado garantiza la consistencia en la gestión de tareas prioritarias a lo largo del tiempo.

## Borrador del aspecto de la interfaz de usuario resultante



## Historia de Usuario 7:



## Editar el perfil del usuario

### Descripción

Como usuario registrado en el sistema, la funcionalidad de "Editar el perfil de usuario" me proporciona la capacidad de actualizar de manera eficiente mi información personal. Esta herramienta es esencial, pues garantiza que mis datos estén siempre al día, reflejando con precisión mi identidad en la plataforma.

### Detalles

- **Sección de Edición en la Interfaz de Usuario:**
  - Deberá existir una sección claramente identificada en la interfaz de usuario que facilite la edición del perfil.
  - La navegación hacia esta sección debe ser intuitiva, y su acceso debe ser directo desde la interfaz principal del usuario.
- **Campos Editables:**
  - Los campos editables deben abarcar información básica, como nombre, apellido y dirección de correo electrónico.
  - Deberá haber una opción para cambiar la contraseña, proporcionando así una capa adicional de seguridad y control para los usuarios.
- **Interfaz Intuitiva y Fácil de Usar:**
  - La interfaz de edición del perfil debe ser diseñada de manera intuitiva y fácil de usar, permitiendo a los usuarios realizar cambios de manera rápida y sin complicaciones.

### Condiciones de Aceptación (COS)

- **Accesibilidad desde la Interfaz Principal:** La opción para editar el perfil debe ser fácilmente accesible desde la interfaz principal del usuario, asegurando que la funcionalidad esté al alcance con unos pocos clics.
- **Modificación Efectiva de la Información del Usuario:** Los campos editables deben permitir la modificación de la información del usuario de manera efectiva, asegurando que los cambios se guarden correctamente.
- **Cambio de Contraseña Seguro:** La opción para cambiar la contraseña debe funcionar correctamente y seguir las políticas de seguridad establecidas para proteger la información del usuario.

- **Mensajes de Error Claros:** Se deben mostrar mensajes de error claros y descriptivos si el usuario intenta realizar cambios no válidos o si surgen problemas durante la edición del perfil.
- **Actualización Inmediata del Perfil:** Los cambios realizados en el perfil deben reflejarse de inmediato en la aplicación, asegurando que la información del usuario esté siempre actualizada.

## Borrador del aspecto de la interfaz de usuario resultante

The mockup shows a web interface for updating a user profile. At the top is a dark navigation bar with the MADS logo on the left and a menu on the right containing 'User' and 'Contacto'. Below the navigation bar is a light gray header section titled 'Perfil de User'. The main content area is white and contains a form with the following fields: 'Nombre' (with a placeholder 'Nombre'), 'PASSWORD' (with a placeholder 'Password'), 'CONFIRM' (with a placeholder 'Confirm Password'), 'EMAIL' (with a placeholder 'Email'), and 'Fecha' (with a placeholder 'Fecha'). Each field is preceded by its label and followed by a text input box. Below the form is a green button labeled 'Actualizar'. At the bottom of the page is a light gray footer section containing the text '© 2023 MADS. All rights reserved.'

# Funcionalidades implementadas

## Poder ver el perfil del usuario y editarlo



The screenshot shows a web interface for a user profile. At the top right, there is a dropdown menu with the name 'noel' and two options: 'Cuenta' and 'Cerrar sesión'. The main heading is 'Perfil de noel'. Below it is a table with the following data:

Id	Correo Electrónico	Nombre	Contraseña	Fecha de nacimiento
1	noelmartinezpomares@gmail.com	noel	noel	2000-12-01

Below the table is a blue button labeled 'Actualizar Perfil'.

Siendo un usuario logueado en la página se podrá acceder al perfil, mediante el desplegable situado en el menú superior en el lado derecho, donde aparece el nombre del usuario, en la opción “Cuenta”.

Una vez en esta página puedes observar los datos personales y mediante el botón “Actualizar Perfil”, podrás acceder a la ventana de editar el perfil, donde se pueden modificar los datos personales, y tiene este diseño:



The screenshot shows a form titled 'Actualizar Perfil'. It contains four input fields with labels above them: 'Correo electrónico' (with a placeholder 'Email'), 'Nombre' (with a placeholder 'Nombre'), 'Contraseña' (with a placeholder 'Contraseña'), and 'Fecha de nacimiento' (with a placeholder 'dd-MM-yyyy'). At the bottom of the form are two buttons: a blue 'Actualizar Perfil' button and a grey 'Volver al Perfil' button.

## Administrador equipo

### Administración del equipo: Equipo 4

Cambiar Nombre

Añadir Descripción

Asignar Administrador

Eliminar Equipo

Al panel de administración de un equipo que es accesible sólo por parte del administrador de la página, se le ha añadido la opción de seleccionar de entre los integrantes del equipo un administrador, que podrá acceder mediante el botón de Modificar:

Usuarios y Tareas

Unirse

Abandonar

Modificar

Administrar

A esta página:

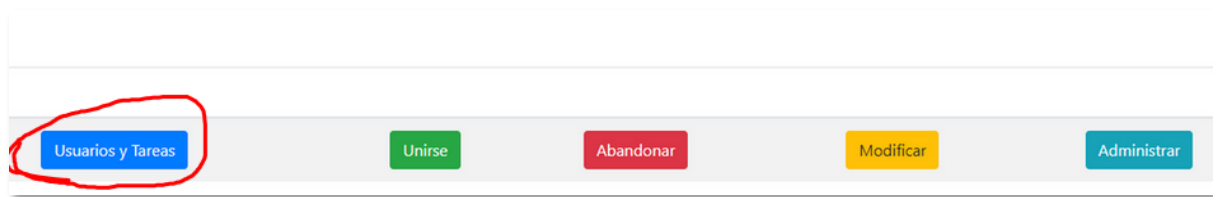
ID	Nombre	Acción
1	noel	<button>Eliminar</button>

Selecciona una tarea a agregar + Añadir

ID Tarea	Nombre Tarea	Fecha Límite	Acción
1	comer	2023-12-29	<button>Eliminar</button>

Donde el administrador del equipo podrá eliminar integrantes y agregar o quitar tareas asociadas al equipo, las tareas que puede agregar son las propias del administrador, asique para que aparezcan en el desplegable para poder agregarlas previamente deben ser creadas por este.

Para visualizar los integrantes y las tareas asociadas al equipo, se accede desde este botón:



Y nos llevará esta página donde podremos ver esos datos:

Usuarios y Tareas del equipo: Equipo2

Id	Correo Electrónico	Nombre	Fecha de Nacimiento
1	comer@comer.com	comer	
2	luis@luis.com	luis	

ID Tarea	Nombre tarea	Fecha Limite
1	Por Programar	2024-01-08

[Volver al equipo](#)

## Destacar Tareas

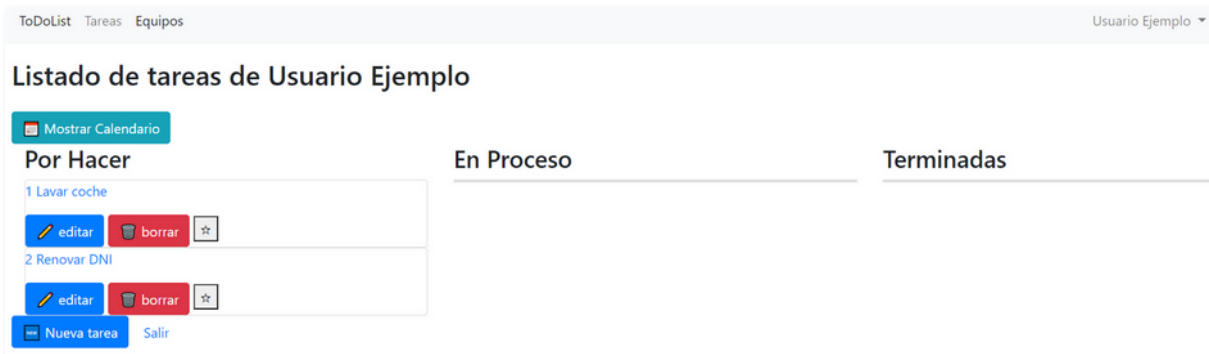
Dentro del listado de tareas podemos destacar tareas haciendo click sobre el símbolo de la estrella, para que aparezcan más arriba, es decir que se fijen al principio de la columna



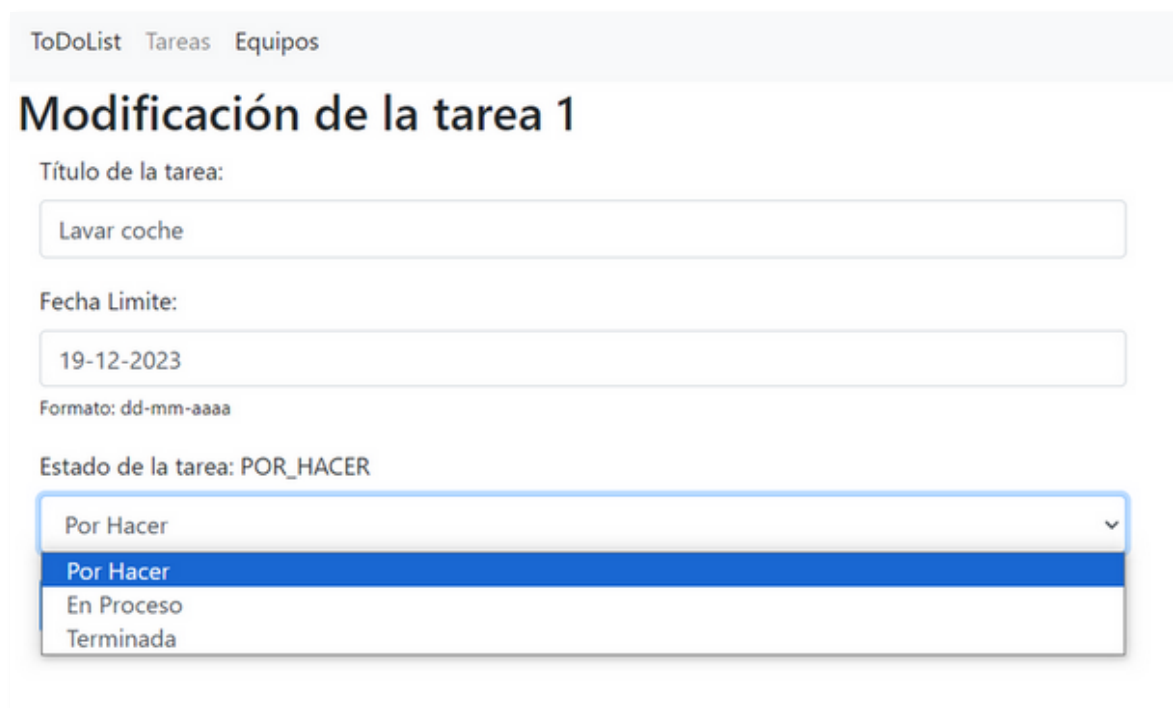


## Estado de las tareas

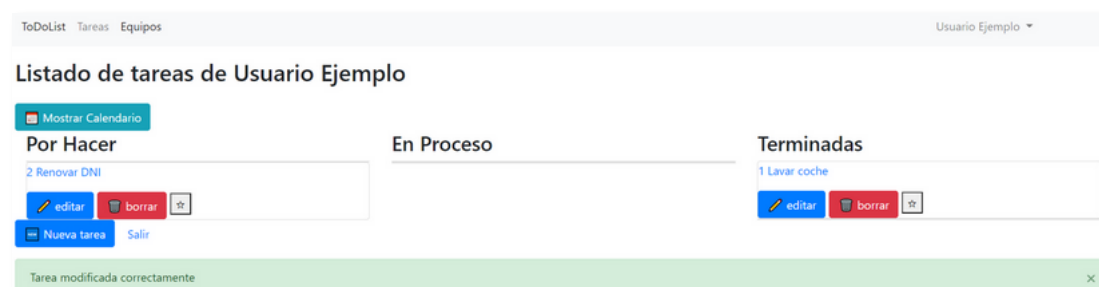
El listado de tareas en la actualización a la versión 1.4.0 ha cambiado, ahora aparecen tres columnas que definen el estado de las tareas:



Para cambiar el estado de una tarea podemos o bien arrastrar la tarea entre las diferentes columnas manteniendo clicado el mouse y soltándola en la columna deseada, o bien clicamos en el botón de editar tarea y accederemos a esta página:



Donde podremos modificar el estado de la tarea, y al volver al listado aparecerá en la columna correspondiente:



## Comentarios en las tareas

Dentro del listado de tareas, si clicamos sobre el nombre de cualquiera de nuestras tareas, nos llevará a esta página:

[ToDoList](#) [Tareas](#) [Equipos](#)

### Tarea: Lavar coche

Fecha Límite: 2023-12-19

Estado: EN\_PROCESO

#### Comentarios:

Escribe un comentario

Agregar Comentario

Donde podremos agregar comentarios a esta

Tras insertar comentarios, se puede ver los detalles del comentario como el usuario que lo ha escrito, su fecha y el texto del comentario. También está la opción de borrarlo gracias al botón de “borrar” incorporado.

[ToDoList](#) [Tareas](#) [Equipos](#) Usuario Ejemplo ▾

### Tarea: Lavar coche


Fecha Límite: 2023-12-19

Estado: EN\_PROCESO

#### Comentarios:


Usuario: Usuario Ejemplo, Fecha: 2023-12-19

Comentario: llegando a la gasolinera

 borrar


Usuario: Usuario Ejemplo, Fecha: 2023-12-19

Comentario: metiendo monedas en el lavadero

 borrar

Usuario: Usuario Ejemplo, Fecha: 2023-12-19

Comentario: saliendo de la gasolinera

 borrar

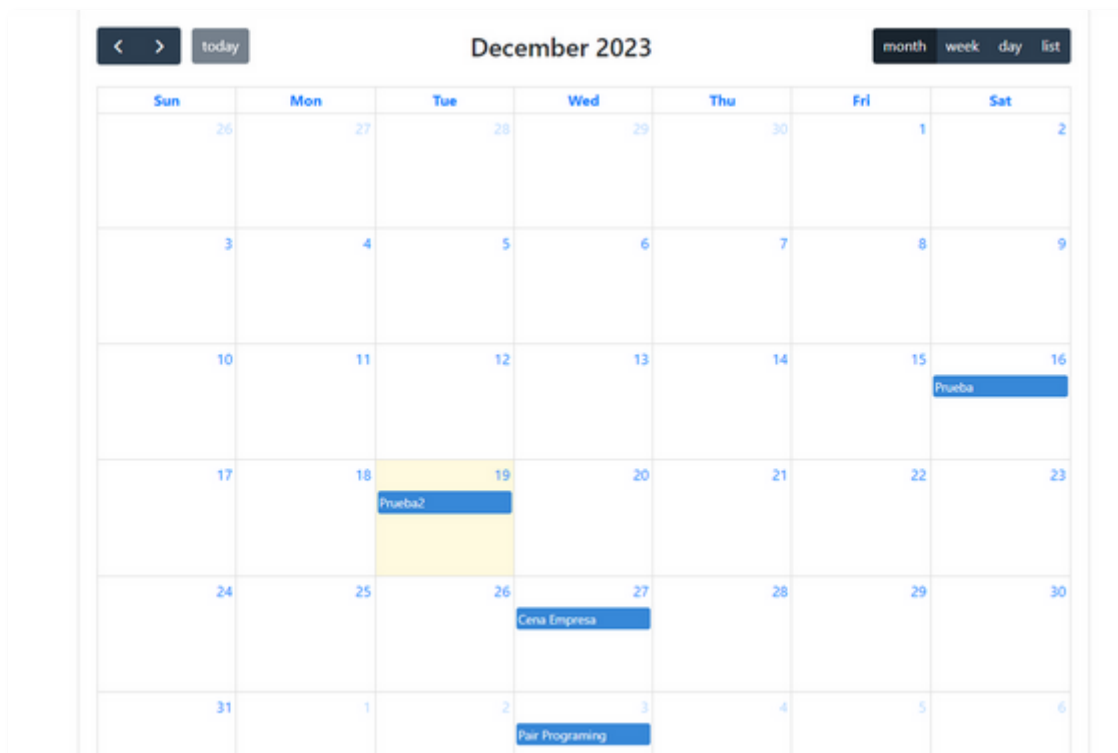
Escribe un comentario

## Calendario de Tareas

Dentro del listado se ha implementado un nuevo botón:



Que nos llevará a una nueva página que contiene un calendario de tareas, donde se mostrarán las tareas que contengan el nuevo atributo de las tareas llamado fecha límite:



Dentro de este nos podremos desplazar por mes, semana, día o un listado de tareas del mes seleccionado, también se puede regresar a la fecha actual dándole al botón “today”, si se ha desplazado a una fecha lejana en el calendario.

# Puesta en producción

Para la puesta en marcha de la aplicación, como cada uno iba haciendo sus tareas, cuando uno terminaba una de estas, después lo que hacía era añadir en el script de migración que teníamos de entre versiones los cambios que había que hacer en la base de datos para que todo pudiese funcionar, quedando como resultado de esto este script:

```
ALTER TABLE public.tareas
ADD COLUMN destacada BOOLEAN;

ALTER TABLE tareas
  ADD COLUMN estado VARCHAR(255);

Alter TABLE tareas
  add column fecha_limite DATE;

ALTER TABLE public.usuarios
  ADD COLUMN fk_admin_equipo_id bigint;

-- Añadir la restricción de clave foránea (foreign key) para la
relación
ALTER TABLE public.usuarios
  ADD CONSTRAINT fk_admin_equipo
    FOREIGN KEY (fk_admin_equipo_id)
      REFERENCES public.equipos(id);

ALTER TABLE public.equipos
  ADD COLUMN fk_admin_usuario_id bigint;

-- Añadir la restricción de clave foránea (foreign key) para la
relación inversa
ALTER TABLE public.equipos
  ADD CONSTRAINT fk_admin_usuario
    FOREIGN KEY (fk_admin_usuario_id)
      REFERENCES public.usuarios(id);

CREATE TABLE public.comentarios (
  id bigint PRIMARY KEY ,
  Texto VARCHAR(255) NOT NULL,
  fecha DATE,
  tarea_id bigint NOT NULL,
  usuario_id bigint NOT NULL
);

ALTER TABLE public.comentarios OWNER TO mads;

--
-- Name: equipos_id_seq; Type: SEQUENCE; Schema: public; Owner: mads
--
```

```

CREATE SEQUENCE public.comentarios_id_seq
  START WITH 1
  INCREMENT BY 1
  NO MINVALUE
  NO MAXVALUE
  CACHE 1;

ALTER TABLE public.comentarios_id_seq OWNER TO mads;

--
-- Name: equipos_id_seq; Type: SEQUENCE OWNED BY; Schema: public;
-- Owner: mads
--

ALTER SEQUENCE public.comentarios_id_seq OWNED BY public.equipos.id;

Alter table public.comentarios
  add CONSTRAINT fk_comentario_tarea FOREIGN KEY (tarea_id)
REFERENCES public.tareas(id);

Alter table public.comentarios
  add CONSTRAINT fk_comentario_usuario FOREIGN KEY (usuario_id)
REFERENCES public.usuarios(id);

ALTER TABLE ONLY public.comentarios ALTER COLUMN id SET DEFAULT
nextval('public.comentarios_id_seq'::regclass);

ALTER TABLE public.tareas
  ADD COLUMN equipo_Asignado_id bigint;

-- Añadir la restricción de clave foránea (foreign key) para la
relación inversa
ALTER TABLE public.tareas
  ADD CONSTRAINT fk_tarea_quipo
  FOREIGN KEY (equipo_Asignado_id)
  REFERENCES public.tareas(id);

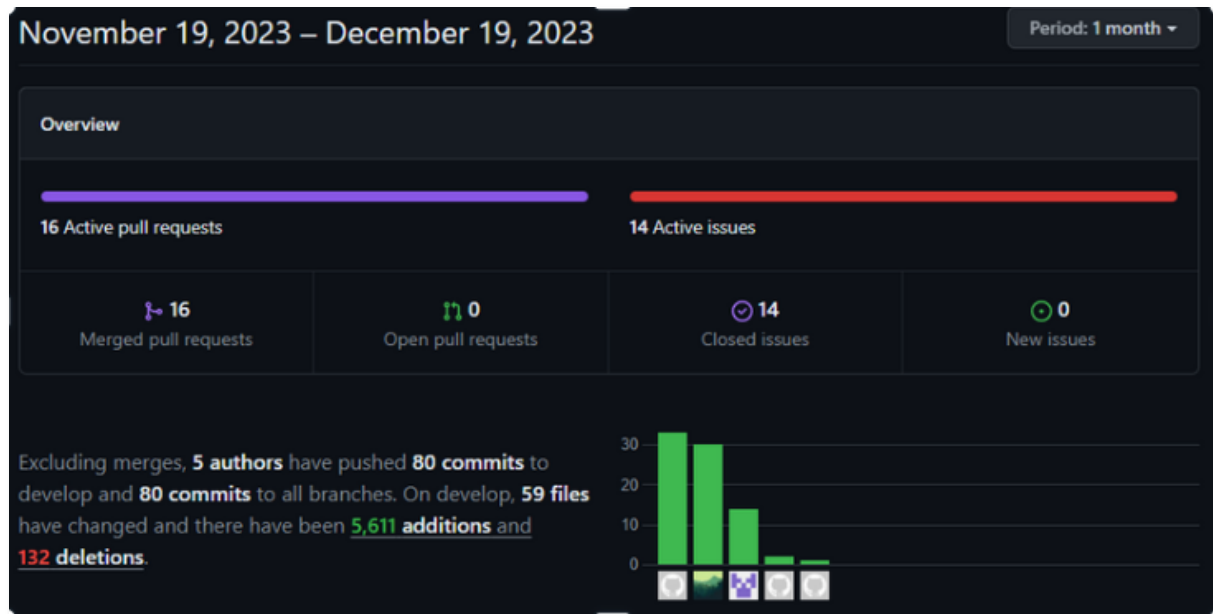
```

Se han tenido que hacer muchos cambios en la base de datos y algunos han sido hechos de forma manual por malentendidos con la forma de hacer la migración de forma rápida, de modo que la secuencia creada para la automatización de las ids de la tabla de comentarios ha sido creada por uno de nuestro miembros en base a otras secuencias que habían en anteriores scripts de releases. Por lo que en la versión final puesta en producción lo nuevo que tenemos en la aplicación es esto: Comentarios en tareas, Estado de tareas, Admin del equipo, Tareas del equipo Calendario de Tareas, Destacar Tareas y Editar perfil del usuario.

# Informe sobre la evolución del desarrollo

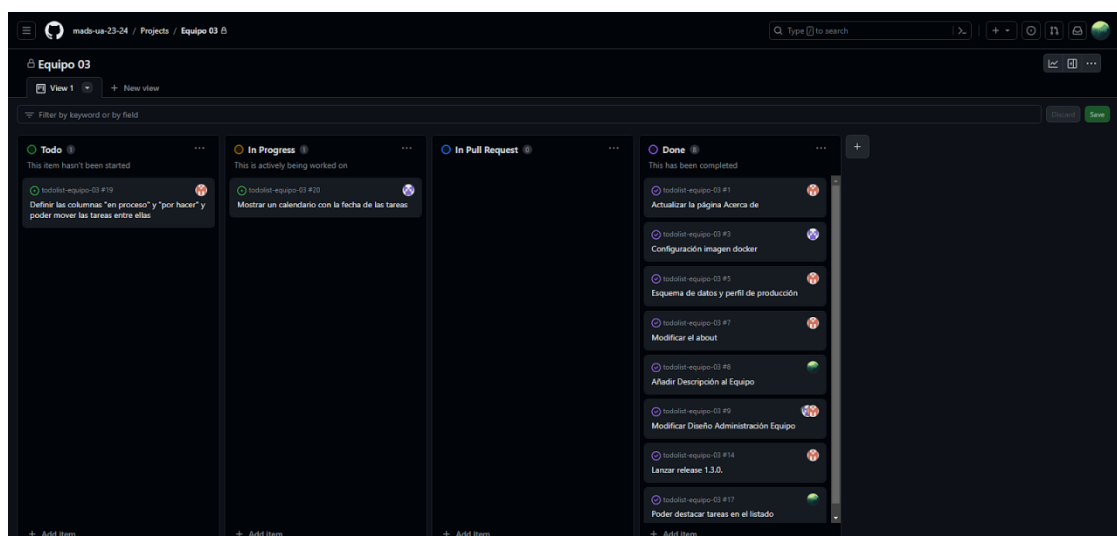
Para el desarrollo de esta aplicación hemos seguido dos metodologías de trabajo, una es el TDD la cual se trata en primero hacer el test de la característica que queremos desarrollar y luego escribir el código necesario para que este test pueda ser pasado, la otra metodología es el desarrollo iterativo que consiste en ver los requerimientos, escribir el código en base a estos y luego probarlo con tests.

Sobre la evolución del desarrollo podemos ver esta captura de pantalla de los commits hechos:

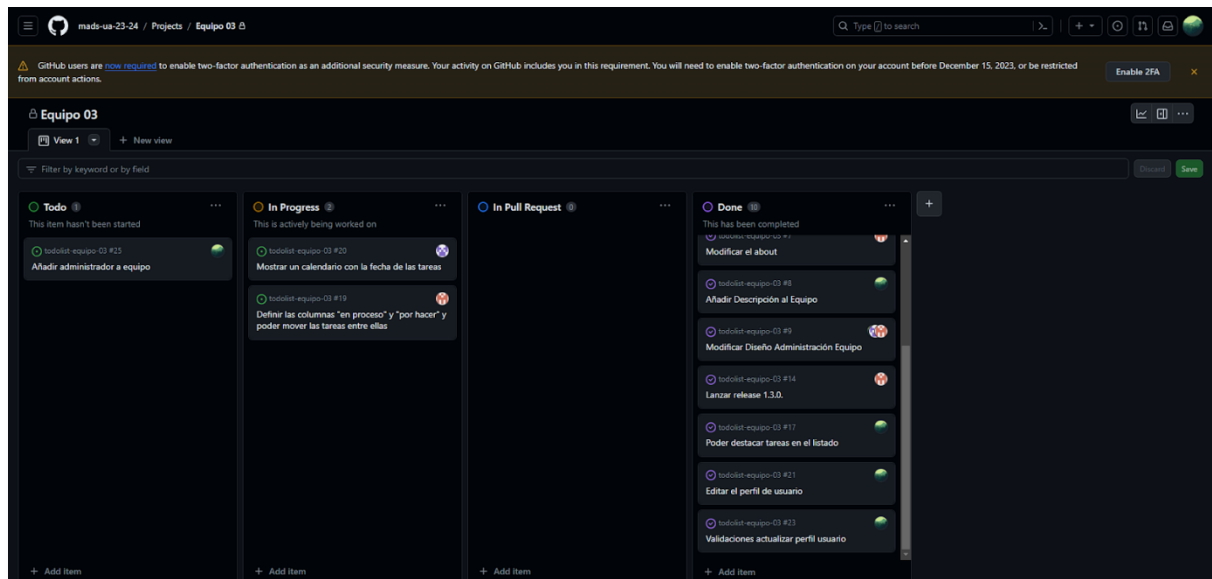


En esta captura aparte de ver el número de commits hechos por cada uno de nosotros, vemos también los pull requests que ha habido junto a sus issues también, más líneas de código añadidas y eliminadas.

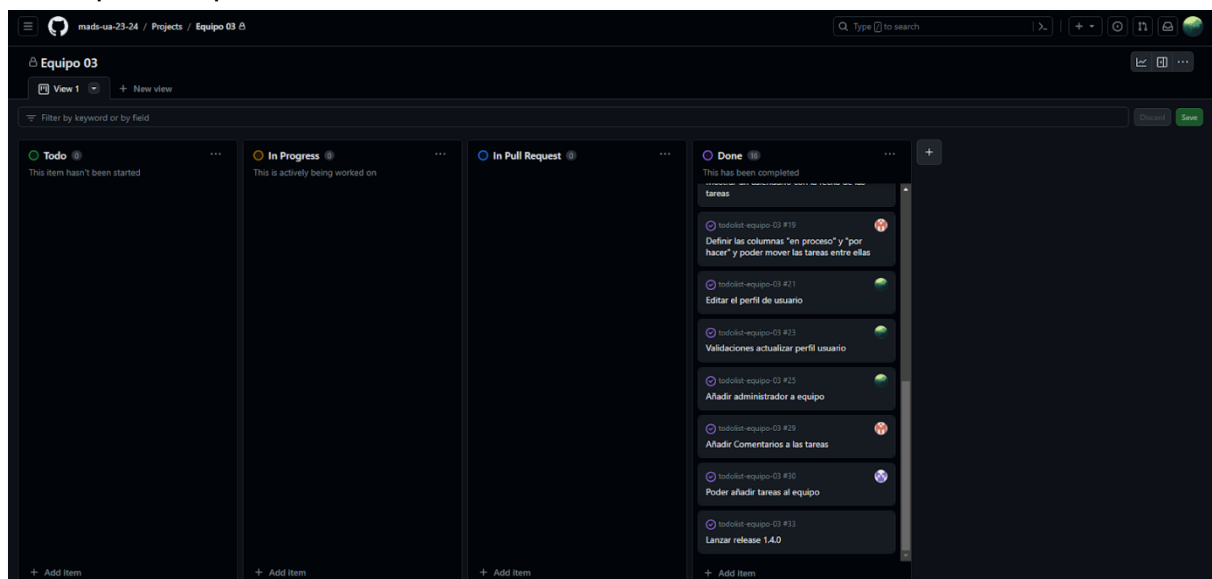
Luego en cuanto a desarrollo que ha habido podemos observar estas capturas de github:



Esta captura es del primer día que empezamos el desarrollo, el cual data del 10-12-2023, podemos ver que había 2 tareas nuevas en el tablero las cuales una estaba para empezar y otra en progreso.

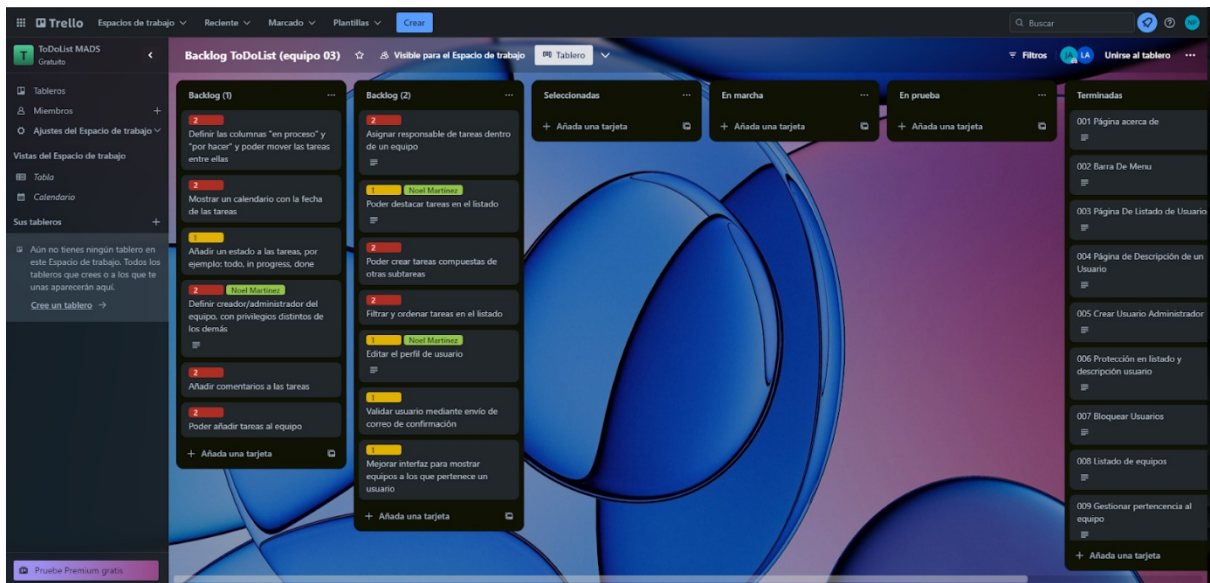


3 días después, vemos como una de las tareas pasaba a en progreso y había una nueva para empezar.

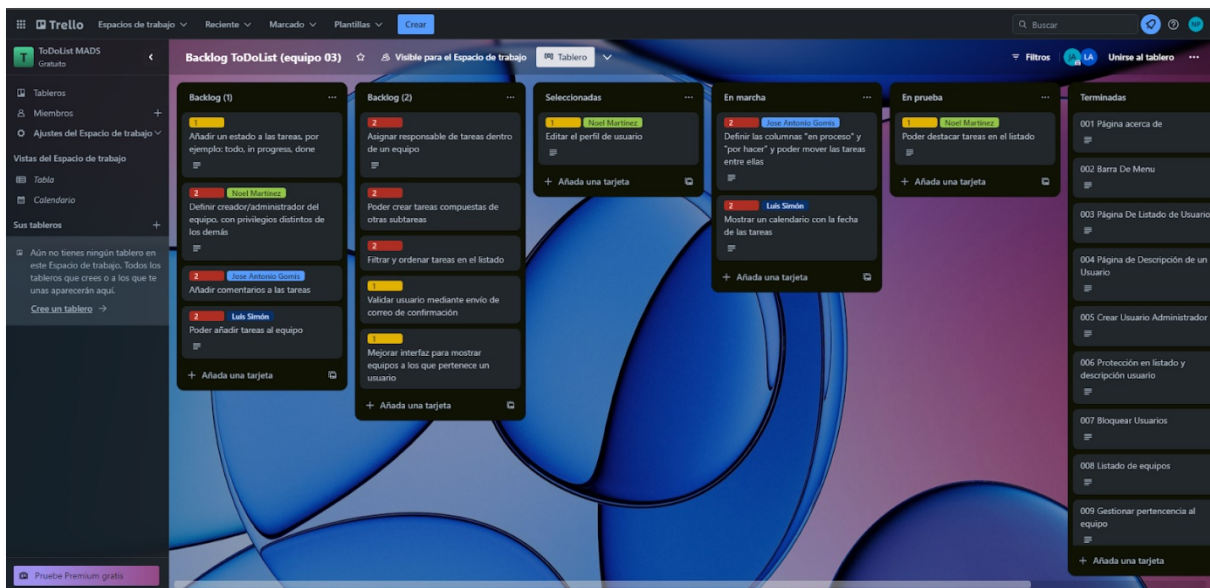


Y ya el día 19 teníamos todo el código terminado.

Luego en cuanto al tablero en trello tenemos esto:

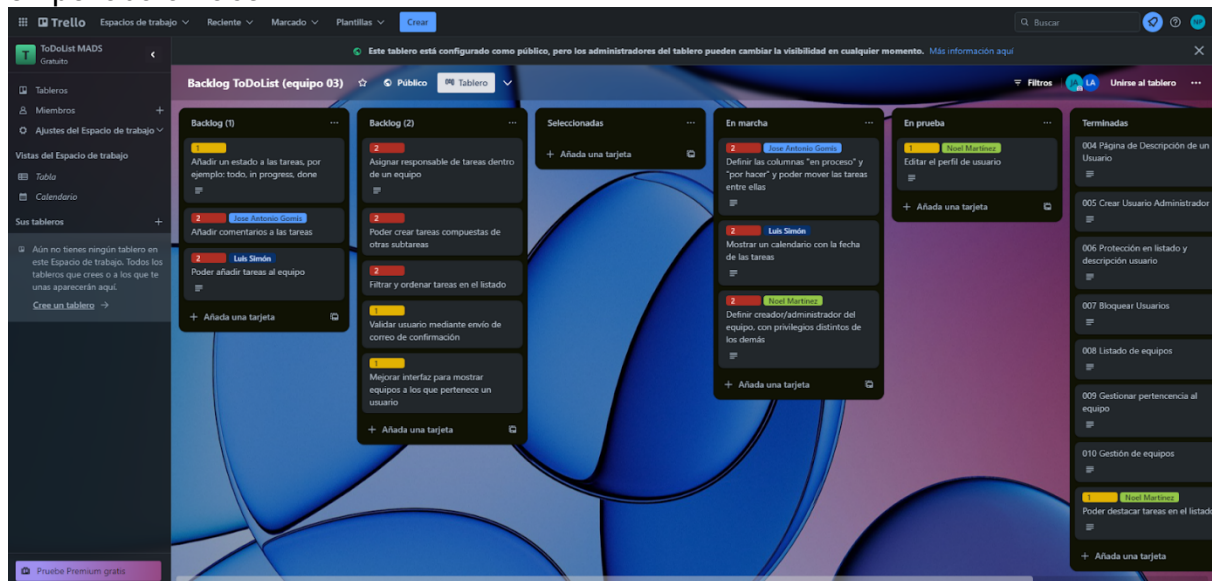


Esto es del 4-12-2023, día en el cual nos juntamos para hacer la brainstorming y poder así sacar funcionalidades para añadir a la aplicación, a lo cual escogimos cada uno las que nos resultaban más atractivas por hacer y que podrían ser más útiles para el usuario.

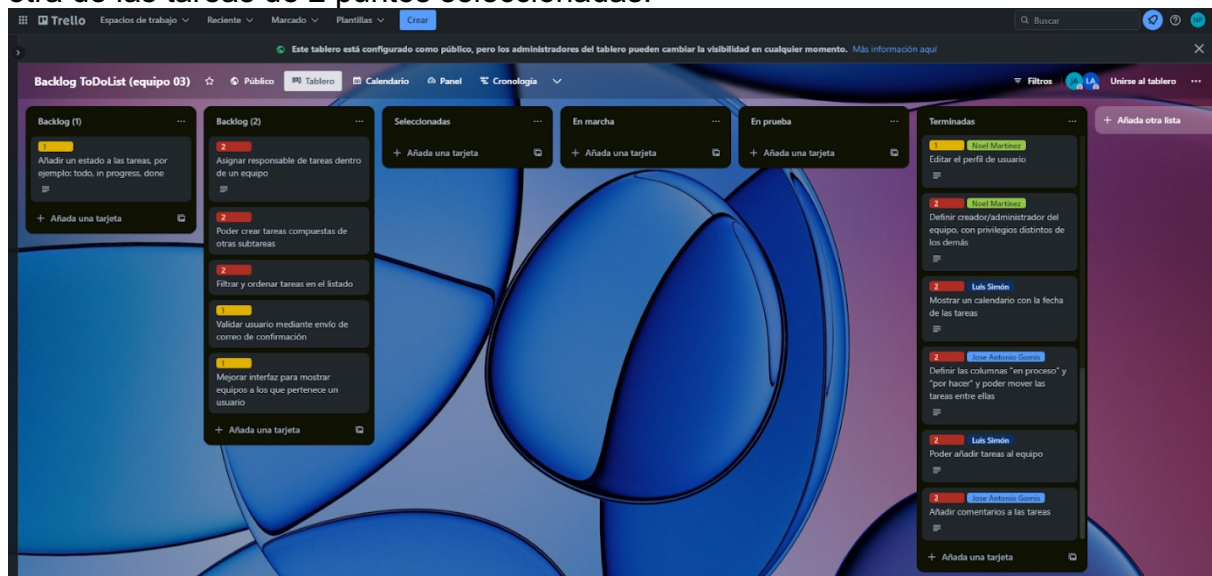




Luego el día 10-12-2023, teníamos unas cuantas tareas en marcha las cuales al ser de 2 puntos llevo su tiempo hacerlas, mientras que había dos tareas de un punto que una estaba prácticamente terminada y la otra estaba ya seleccionada para ser empezada a hacer.



El día 13 seguíamos con las tareas de 2 puntos de antes en marcha mientras que algunas tareas de 1 punto ya habían sido terminadas y había sido puesta en marcha otra de las tareas de 2 puntos seleccionadas.



Ya para el día 19 tuvimos todo terminado.

# Informe sobre sesiones de pair programming

En cuanto a la metodología ágil del Pair Programming, se llevaron a cabo 2 sesiones de una hora y veinte minutos cada una, separadas en intervalos de 20 minutos en los que se turnaron las posiciones de la pareja. Es decir, primero uno de la pareja programó mientras que la otra persona observó y ayudó a la persona que estaba programando; luego de 20 minutos, la persona que estaba programando pasó a ayudar en la programación y viceversa.

## Primera sesión: Validaciones de Editar Perfil - Luis y Noel

Esta primera sesión fue muy productiva, ya que había muchos problemas en la actualización del perfil del usuario. Esto se debía a que cada usuario tenía una fecha de nacimiento, y daba error al intentar actualizar los datos. El error ocurría antes de llegar al controlador del botón para actualizar; es decir, el error se producía al intentar asignar el atributo Date al objeto registroData (el utilizado para la actualización del perfil).

El problema se pudo solventar correctamente, dando por finalizada la sesión de Pair Programming de manera exitosa. Se logró solucionar creando desde cero el formulario para introducir los nuevos datos del usuario, ya que el error estaba en la propia estructura del elemento form (formulario). Se llegó a la conclusión que, de la manera en que estaba estructurado, el atributo Date no se asignaba correctamente al objeto registroData.

**Luis:** La sesión transcurrió de manera increíble, destacando por la eficiencia en la implementación de gran parte de la funcionalidad en un período corto. La solución de errores fue ágil, gracias a una comunicación fluida. Mi experiencia fue positiva, manteniendo la concentración que facilitó el desarrollo.

**Noel:** Al trabajar en pareja para una misma funcionalidad, desapareció cualquier posibilidad de estancamiento. Los errores fueron escasos, y la distribución de roles entre el programador y el observador permitió prevenir problemas y organizar los pasos de manera efectiva. Desde mi perspectiva, esta forma de agilizar el trabajo fortalece la interacción activa con los compañeros de equipo.

## Segunda sesión: Drag and Drop (Kanban) - Noel y Jose

La segunda sesión de Pair Programming también fue de gran ayuda, ya que este problema era más complejo que el solucionado en la sesión anterior. No se trataba de solucionar un problema en sí, pero el código era más difícil de crear y entender.

Se necesitaba un listado de tareas organizado en tareas divididas por estados: Por Hacer, En Proceso, Terminadas. Al comenzar la sesión, las tareas ya tenían asignados de forma correcta estos estados. El problema estaba en el sistema de columnas y drag and drop, ya que no había columnas y solamente había distintas filas por cada estado. Además, el drag and drop no funcionaba completamente bien, ya que se podía mover una tarea del estado Por Hacer al estado En Proceso, pero no se podía hacer a la viceversa.

Al final de la sesión, no logramos realizar todo lo que teníamos en mente, pero sí conseguimos que el sistema de drag and drop funcionara correctamente y que el script reconociera de manera correcta en qué fila iba a caer la tarea al soltarla, asignando correctamente el estado que le correspondía. Más tarde, Jose logró modificar el sistema de filas y convertirlo en un sistema de columnas.

**Noel:** Nuestra segunda sesión de Pair Programming me resultó positiva. La atención al código escrito durante la sesión demostró ser de gran utilidad. Además, el tiempo necesario para pensar e implementar funcionalidades complejas disminuyó significativamente gracias a la cooperación constante.

**Jose:** La tarea fue bastante complicada, ya que implicaba agregar nueva información a la base de datos y realizar cambios en páginas y métodos existentes. Sin embargo, el Pair Programming facilitó el proceso de desarrollo. Las ideas para superar complicaciones surgieron rápidamente, y en todo momento teníamos claro cuál sería el siguiente paso a realizar.

# Resultado de la retrospectiva

Todas las funciones previstas para este sprint se han implementado con éxito. La coordinación del equipo ha sido efectiva a través de comunicación grupal en WhatsApp, y la planificación y seguimiento de tareas se han gestionado mediante Trello y las funciones de GitHub.

Para determinar las funcionalidades y asignar niveles de complejidad, llevamos a cabo una sesión en Discord. Durante esta reunión, uno de nosotros compartió su pantalla, mostrando el tablero de Trello con las historias de usuario en el backlog. A través del razonamiento y opiniones, logramos aclarar la asignación de niveles de dificultad a cada historia de usuario. Más tarde, al seleccionar las historias de usuario para su implementación, no surgió ningún contratiempo; cada uno eligió las suyas sin problemas.

En los últimos días de la implementación, surgieron diversos problemas relacionados con los merges, especialmente debido a la introducción de nuevos elementos/atributos en la base de datos, generando algunos conflictos. Afortunadamente, todos estos desafíos se resolvieron de manera efectiva.

Tras la presentación y varias conversaciones en equipo, hemos creado una lista de ajustes y adiciones que consideramos necesarias para el próximo sprint:

- Refinar el diseño visual de las plantillas HTML.
- Facilitar la edición de tareas y la adición de comentarios, ya que la forma de acceder a la página para agregar comentarios es un poco confusa.
- Mejorar el listado de equipos para restringir la visualización de usuarios y tareas únicamente a aquellos que pertenecen al equipo correspondiente. Evitar que usuarios que no pertenecen al equipo puedan acceder a información de otros equipos.
- Dar al administrador del equipo la capacidad de crear tareas directamente desde la página de administración del equipo, quitando la posibilidad de agregar tareas personales. Las tareas deberían ser exclusivamente destinadas al equipo.

Docker: <https://hub.docker.com/r/noelmartinez/mads-todolist-equipo03/tags>

GitHub: <https://github.com/mads-ua-23-24/todolist-equipo-03>

Trello: <https://trello.com/b/4GHoXOQW/backlog-todolist-equipo-03>