

# Anexo I - PLAN DE PROYECTO

## SOFTWARE

Trabajo de Fin de Grado

Ingeniería Informática



**VNiVERSiDAD**  
**DE SALAMANCA**

Septiembre de 2023

Autor

Luis Prada Rodrigo

Tutores

Roberto Therón Sánchez

Luis Martín Liras

Nicolás García Martín

## Índice

1. Introducción .....	2
1. Planificación temporal del proyecto .....	2
1.1 Calendario del proyecto .....	2
1.2 Diagramas de Gantt.....	3

## 1. Introducción

En este anexo se recoge la planificación temporal del proyecto.

Entre los contenidos del documento se encuentran los diagramas de Gantt por Sprints, creados mediante la herramienta YouTrack, y un calendario del proyecto para una representación más visual de la duración del proyecto.

una explicación de la forma en la que se va a aplicar la metodología Scrum debido a su relevancia, el calendario de trabajo, el product backlog con la estimación de tiempos del proyecto y, por último, los diagramas de Gantt donde se representa la duración de forma gráfica.

## 2. Planificación temporal del proyecto

### 2.1 Calendario del proyecto

Se ha elaborado un calendario para el desarrollo del proyecto, presentado en la Figura 1. Este contiene los 11 Sprints, los cuales tienen una duración de 2 semanas (10 días laborables).

Dado que este proyecto se realiza como estudiante, no se tiene un horario fijo que se vaya a poder seguir. Se partirá del supuesto de que se trabajarán unas 3 horas de media por día laborable, aunque estas 15 horas semanales realmente puedan ser todas en el fin de semana en vez de 3 por día, por la inestabilidad de mis horarios.

Como se puede observar, el proyecto tendría una duración total de 100 días laborables, comenzando el 1 de marzo y terminando el 1 de agosto.

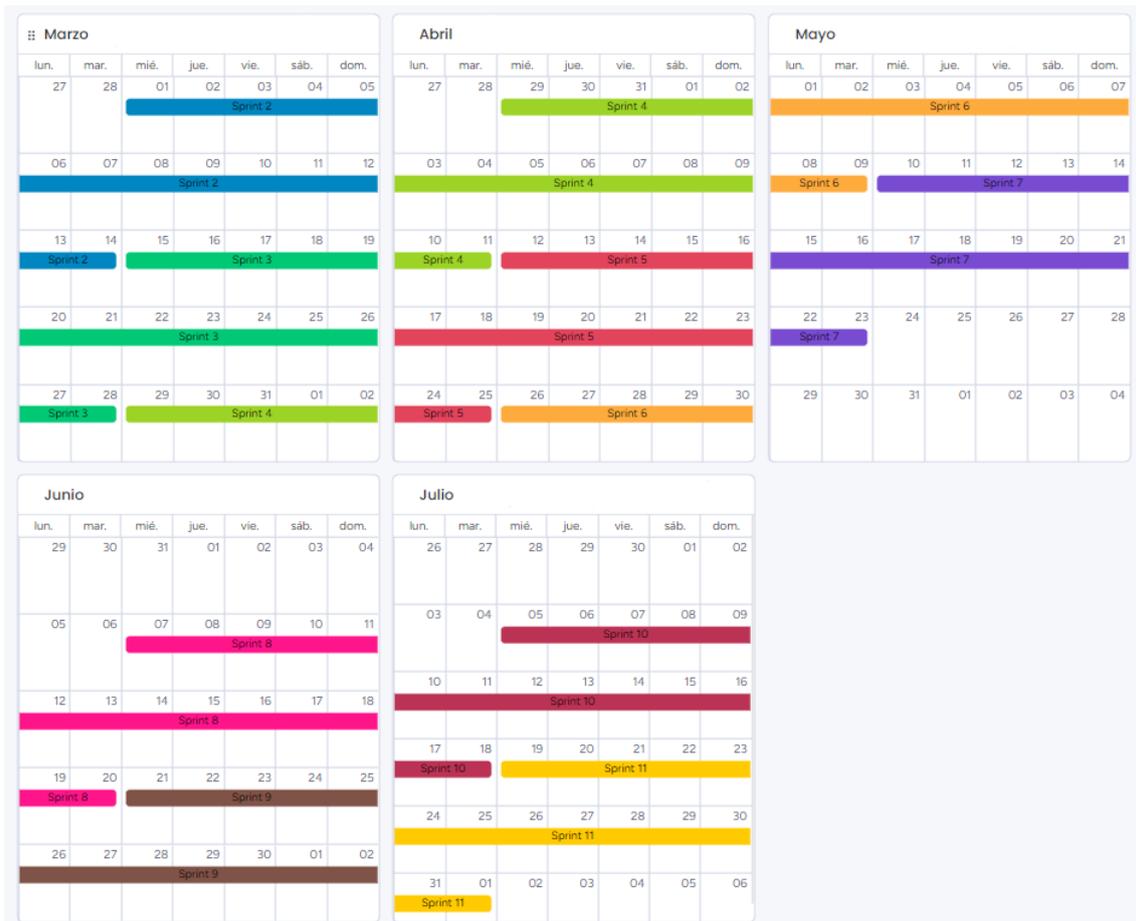


Figura 1. Calendario del proyecto

## 2.2 Diagramas de Gantt

Como se ha utilizado la metodología ágil SCRUM para el desarrollo del proyecto, la planificación temporal se hará por Sprints y con un Product Backlog de historias de usuario (presentado en el Anexo 2) que podrá variar en cada iteración. Al final de cada Sprint se tendrán reuniones con el resto del equipo (mis tutores), y en ellas se priorizarán las tareas del próximo Sprint.

La estimación de tiempo se hace teniendo en cuenta una media de 3 horas trabajadas por día laborable. Por esto parece que las tareas requieren demasiado tiempo, pero realmente 1 semana equivaldría a 15 horas reales de trabajo.

Solo se pondrán las tareas planificadas para cada Sprint en vez del Product Backlog entero en cada Sprint, para facilidad visual.

Se podrá observar que no hay ninguna tarea desarrollada en paralelo con otra, a pesar de que muchas tareas no dependan de otras. Esto se debe a que solo hay un desarrollador.

Las tareas para desarrollar las historias de usuario se crearán en cada Sprint basándose en lo que sea conveniente en ese momento y en errores encontrados.

Se comenzará la planificación temporal a partir del Sprint 2, pues el primer Sprint será dedicado a aprender C++, Cmake y configurar el IDE.

## Sprint 2

- Como desarrollador quiero un servidor REST para que los usuarios puedan acceder a él.
  - Buscar una biblioteca que me facilite crear el servicio REST en C++
  - Implementar un servidor inicial de prueba.



Figura 2. Diagrama de Gantt del Sprint #2

## Sprint 3

- Como desarrollador quiero un servidor REST para que los usuarios puedan acceder a él.
  - Poder colocar imágenes sin que de error de archivo inexistente en los HTML.
- Como desarrollador quiero que haya distintos tipos de soldados para aportar variedad al juego.
  - Crear clase Soldier.
  - Crear clases que hereden de Soldier (Attacker, Tank, Healer y Helicopter).

- Como usuario quiero poder ver el juego en una interfaz gráfica en tiempo real para ver cómo procede la partida.
  - Crear una interfaz inicial, solo de texto que muestre los soldados y su información.
- Como desarrollador quiero que haya distintos tipos de casillas y que estas afecten en el juego para que la programación del bot no sea muy simple.

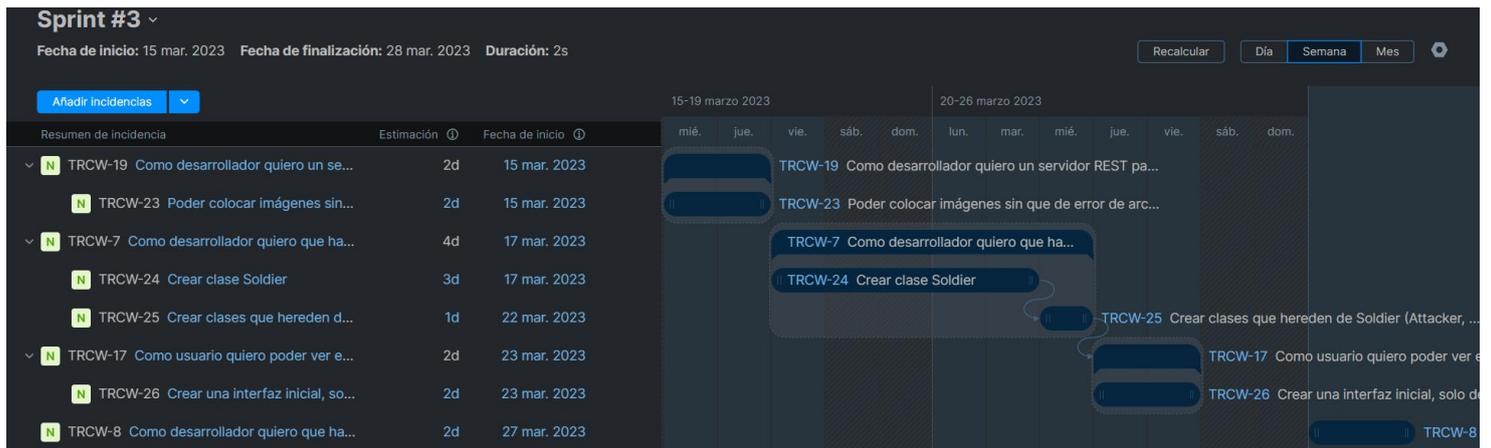


Figura 3. Diagrama de Gantt del Sprint #3

## Sprint 4

- Como desarrollador quiero que los soldados puedan nadar o volar para que actúen de forma distinta con los diferentes tipos de casillas.
- Como desarrollador quiero que los soldados tengan acciones para moverse, atacar, curar y crear soldados.
  - Implementar acción move()
- Como desarrollador quiero poder guardar mapas en ficheros para poder elegir un mapa ya creado al crear una partida.
- Como usuario quiero poder ver el juego en una interfaz gráfica en tiempo real para ver como procede la partida.

- Hacer la interfaz visual con el uso de un canvas.

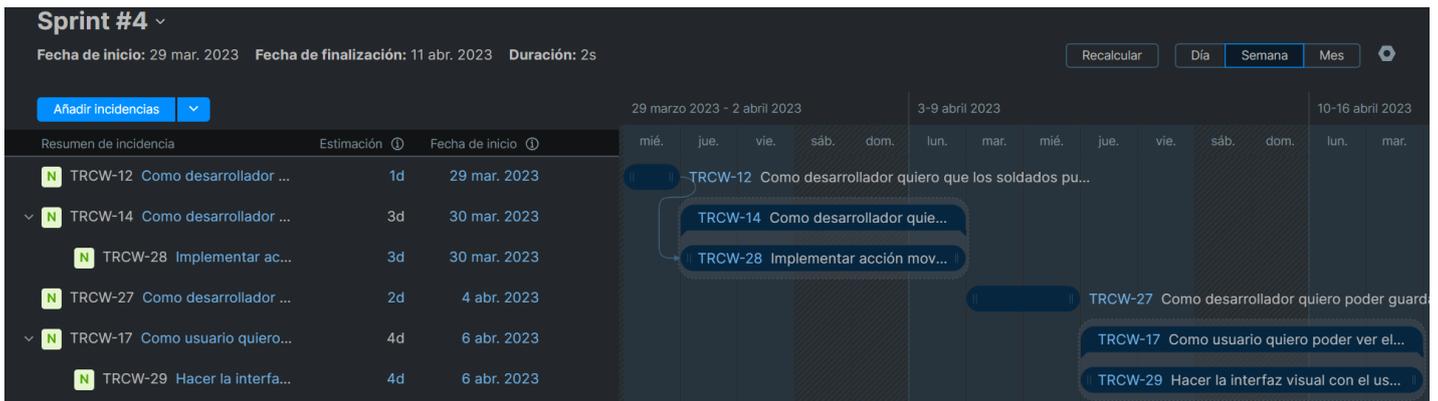


Figura 4. Diagrama de Gantt del Sprint #4

## Sprint 5

- Como desarrollador quiero que cada equipo tenga una base y que al destruirla se gane la partida para que el juego tenga un objetivo.
- Como desarrollador quiero que los soldados tengan acciones para moverse, atacar, curar y crear soldados.
  - Implementar acción attack()
  - Implementar acción create()
- Como usuario quiero poder ver el juego en una interfaz gráfica en tiempo real para ver como procede la partida.
  - Mejorar la interfaz visual, cambiando el canvas por un grid con imágenes para los soldados y casillas.

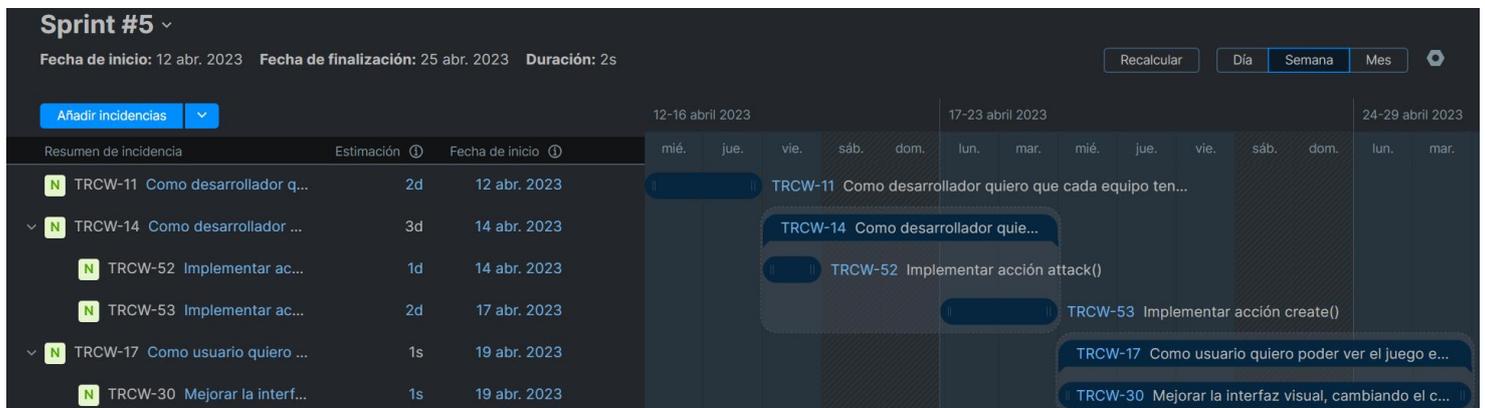


Figura 5. Diagrama de Gantt del Sprint #5

## Sprint 6

- Como desarrollador quiero que los usuarios puedan escribir el código de los bots para que ese código controle a su equipo en una partida.
  - Desarrollar la comunicación entre el bot y el servidor con endpoints del servicio REST.
  - Desarrollar un bot de prueba.
- Como usuario quiero poder subir el código del bot a la página para poder usarlo en partidas.
  - Desarrollar un endpoint en el que poder crear una partida a partir de los códigos proporcionados como bots.



Figura 6. Diagrama de Gantt del Sprint #6

## Sprint 7

- Como desarrollador quiero que los usuarios puedan escribir el código de los bots para que ese código controle a su equipo en una partida.
  - Cambiar la comunicación bot - Servidor de REST a sockets TCP para reducir la carga.

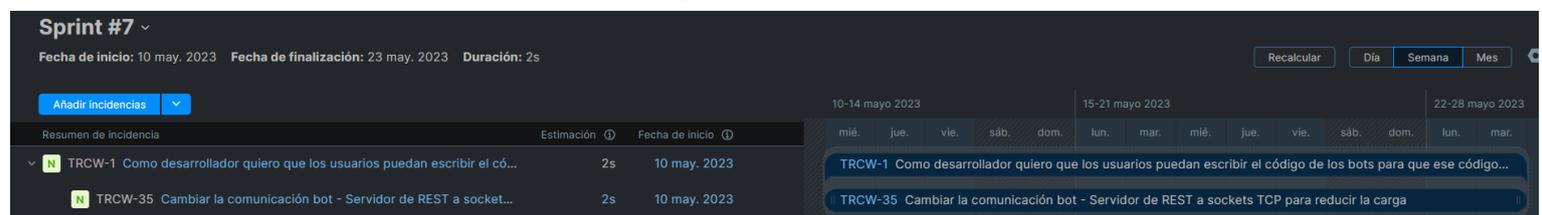


Figura 7. Diagrama de Gantt del Sprint #7

## Sprint 8

- Como desarrollador quiero que los soldados puedan ver la información de las casillas en su rango de alcance para que los usuarios puedan programar un bot que reaccione a su entorno.
- Como usuario quiero que exista una biblioteca con funciones y clases para que el código relacionado con el juego sea simple de implementar.
  - Desarrollar la biblioteca y sus funciones.
  - Desarrollar bot de ejemplo.
- Como desarrollador quiero que el juego se desarrolle por turnos para evitar que las decisiones tomadas por el bot sean erróneas si la posición del contrario cambia.
  - Desarrollar control de los turnos de cada equipo y evitar bloqueo de turnos por parte de cualquier equipo.
- Como usuario quiero que pueda crear un usuario para que almacene mi código.
  - Desarrollar landing page.
  - Desarrollar página de registro.
  - Desarrollar página de iniciar sesión.
  - Desarrollar tabla de SQLite para guardar usuarios y endpoint con la lógica por parte del servidor.

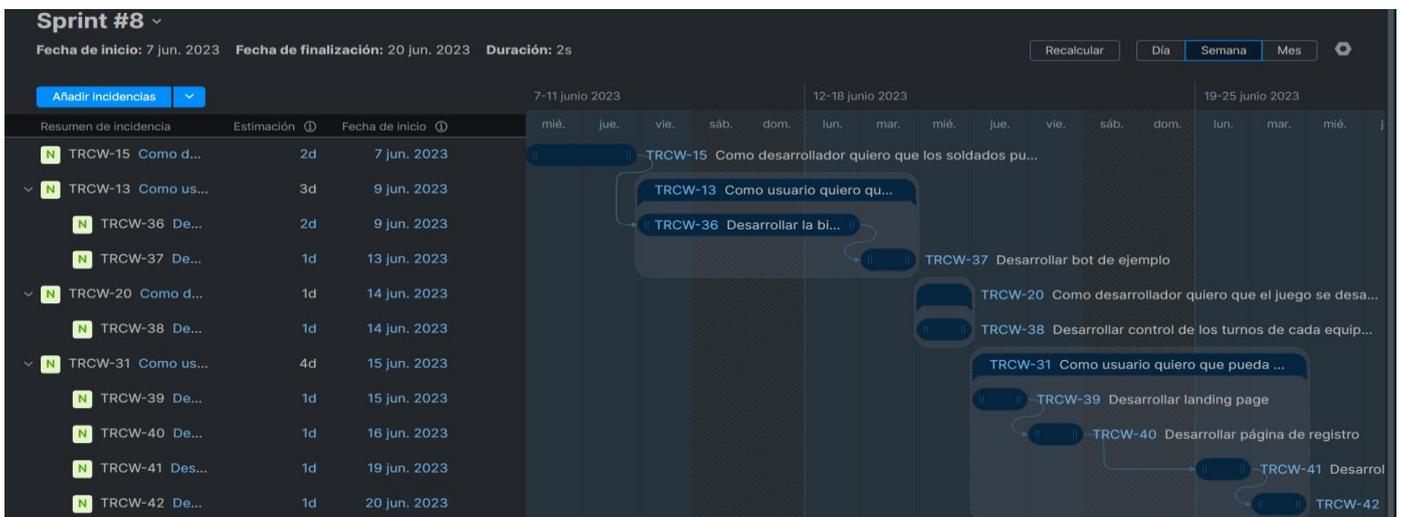


Figura 8. Diagrama de Gantt del Sprint #8

En este Sprint cabe mencionar que hay bastantes más tareas que en los anteriores, porque al acabar las clases he podido incrementar el tiempo empleado en el desarrollo del proyecto de 2-3h al día a 4h. Esto solo se aplica durante el mes de junio.

## Sprint 9

- Como desarrollador quiero que los soldados tengan acciones para moverse, atacar, curar y crear soldados.
  - Desarrollar acción de curar.
- Como usuario quiero que exista una biblioteca con funciones y clases para que el código relacionado con el juego sea simple de implementar.
  - Crear las clases de Soldier y Tile en vez de usar arrays.
- Como usuario quiero poder ver el juego en una interfaz gráfica en tiempo real para ver cómo procede la partida.
  - Añadir barras de vida a los soldados de cada equipo de un color distinto para diferenciarlos.

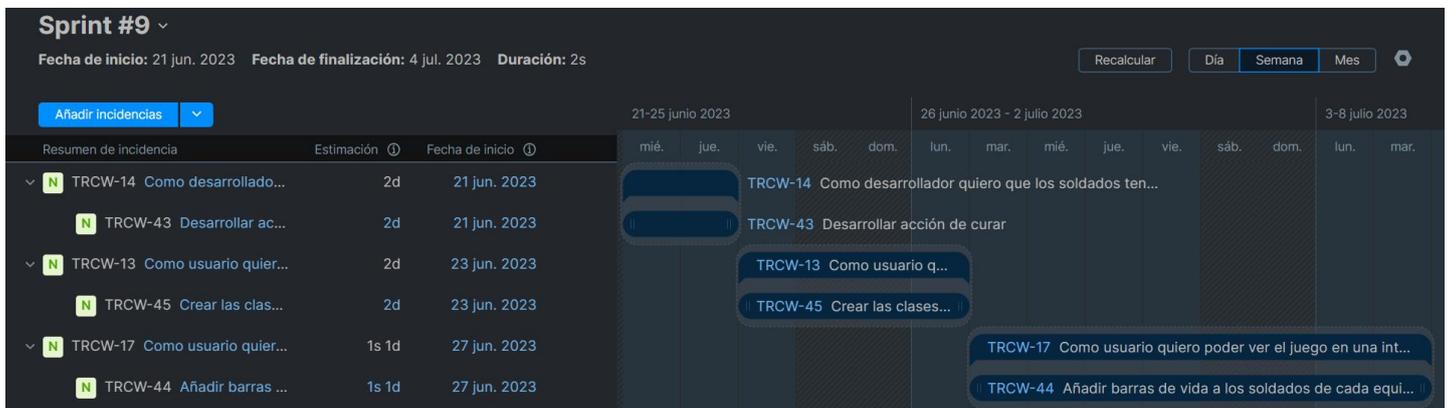


Figura 9. Diagrama de Gantt del Sprint #9

## Sprint 10

- Como usuario quiero poder subir el código del bot a la página para poder usarlo en partidas.
  - Desarrollar página principal del usuario.

- Desarrollar página para subir, ver o eliminar el código de un usuario.
- Como usuario quiero crear partidas entre los usuarios, código y mapa que yo elija para poder hacer torneos.
  - Desarrollar página para crear partida en función de los usuarios, sus códigos y los mapas disponibles.

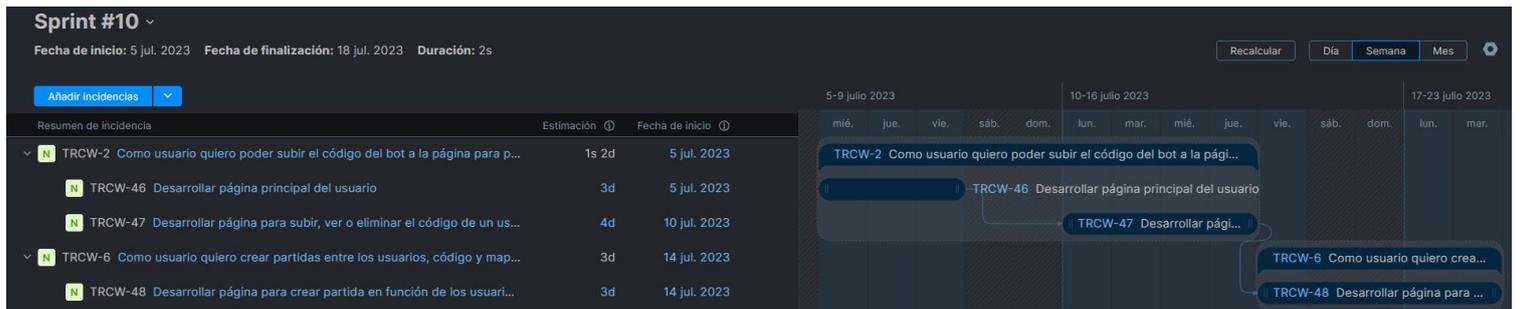


Figura 10. Diagrama de Gantt del Sprint #10

## Sprint 11

- Como usuario quiero poder acceder a un tutorial para aprender sobre el juego y lo necesario para crear el bot, subirlo y crear una partida.
- Como usuario quiero poder ver el juego en una interfaz gráfica en tiempo real para ver cómo procede la partida.
  - Añadir información para saber qué equipo es cada jugador.
- Como usuario quiero que exista una biblioteca con funciones y clases para que el código relacionado con el juego sea simple de implementar.
  - Cambiar los atributos de las clases Tile y Soldier para que no se puedan cambiar sus valores.
- Como usuario quiero poder probar mi bot en cualquier momento para comprobar si está bien escrito y si funciona como debería.
  - Crear un usuario que tenga varios bots ya creados para que los usuarios puedan poner a prueba su código.

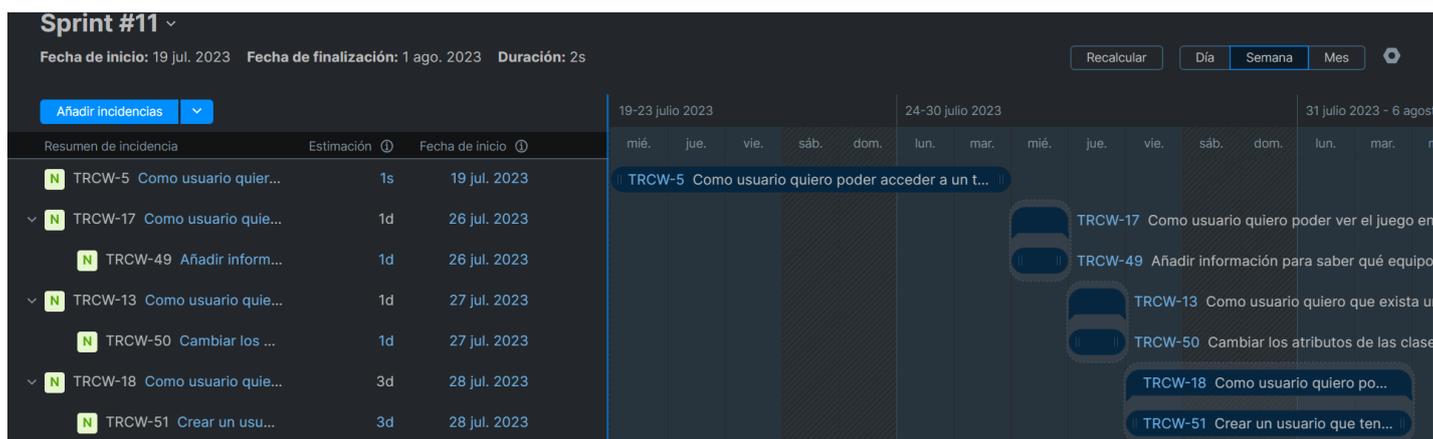


Figura 11. Diagrama de Gantt del Sprint #11

Al haber utilizado Scrum como metodología, aunque las historias de usuario relacionadas con los requisitos estén planteadas desde el principio del proyecto, estas pueden variar. Los diagramas de Gantt se hacen al inicio de cada Sprint para priorizar historias de usuario y organizar el trabajo. Al finalizar el Sprint si no se han completado los objetivos propuestos al inicio de este, también se podría ver en el diagrama de Gantt que se va con cierto retraso.