

Procesos de Ingeniería del Software

Grado en Ingeniería Informática - UCLM

BATALLA NAVAL

Juego de HUNDIR LA FLOTA

Curso 2022/2023



AUTOR: JOSE DANIEL MARTÍNEZ CASTILLO

ÍNDICE

Introducción	4
Sprint 1: Puesta en marcha del proyecto	5
Tareas de formación.....	5
Guía del desarrollador web	5
Tutoriales de HTML, JavaScript, JQuery y NodeJS.....	5
Reunión con el cliente	5
Inicio de sesión	6
Poder crear partidas y unirse con un nick.....	6
Revisión del Sprint 1	7
Documentación del Sprint 1	7
Sprint 2: Funciones de salir y abandonar partida	7
Reunión con el cliente	7
Salir del juego	7
Abandonar partida	8
Implementación del despliegue continuo.....	8
Revisión del Sprint 2	8
Documentación del Sprint 2	8
Sprint 3: Definir la lógica GUI de la Batalla Naval.....	8
Reunión con el cliente	8
Diagramas.....	9
Colocar barcos.....	10
Mostrar tablero propio y tablero rival	10
Disparar	11
Resultado final de la partida	12
Revisión del Sprint 3	13
Documentación del Sprint 3	13
Sprint 4: Persistencia.....	13
Reunión con el cliente	13
Revisión del Sprint 4.....	14
Documentación del Sprint 4.....	14
Sprint 5: Barcos con orientación vertical	14
Reunión con el cliente	14
Pruebas unitarias.....	16
Revisión del Sprint 5.....	16
Documentación del Sprint 5.....	16

Anexos.....	16
Enlace de Cloud Run.....	16

Introducción

Este es un proyecto que hemos realizado en la asignatura de Procesos de Ingeniería del Software del grado de Ingeniería Informática de la Universidad de Castilla-La Mancha. La idea fue desarrollar el antiguo y popular juego de “Hundir la Flota” para poder jugar online. Esta idea fue propuesta por José Antonio Gallud Lázaro, el profesor de la asignatura, para entender de forma más fácil el desarrollo de software, junto con su arquitectura y las partes de backend y frontend realizadas para llevar a cabo este proyecto. También sirvió para entender algún lenguaje de programación, como HTML y JavaScript, y con esto algún entorno de ejecución, como Node.js, ya que el proyecto se orientaba a eventos asíncronos.

Para la realización del proyecto hemos optado por seguir un desarrollo con metodologías ágiles ya que no estaban del todo claros los requisitos y estos podrían ir cambiando. La metodología elegida al final será similar a Scrum dividiendo el proyecto en Sprint que no tenían un tiempo fijo de finalización, simplemente se iba revisando el proyecto continuamente hasta comprobar que se habían completado todas las funcionalidades previstas para ese Sprint y entonces cerrarlo para empezar con el siguiente.

Este proyecto consiste en desarrollar un repositorio que contiene la arquitectura base de una solución software estilo SaaS (Software as a Service) de una aplicación genérica que implementa la siguiente funcionalidad:

- Los usuarios inician sesión con un nick (no hay clave) o con su cuenta de Google. El nick es único. Solo podrá volverse a utilizar un nick ya utilizado cuando ese usuario se salga del juego.
- Los usuarios pueden crear partidas (sin nombre). Las partidas son de 2 usuarios. El sistema asigna un código a la partida.
- Los usuarios pueden unirse a partidas disponibles (las que tienen sólo un jugador).
- Los usuarios pueden abandonar la partida. Si lo hacen la partida finalizará.
- Los usuarios pueden salir del sistema (cerrar sesión). Si lo hacen en medio de una partida, esta finalizará.
- Una vez que la partida esté llena, los dos usuarios podrán colocar todos los barcos. Cuando todos los barcos de los dos jugadores estén colocados se iniciará la partida.
- Una vez que la partida se haya iniciado los jugadores podrán disparar sobre el tablero del rival (siguiendo unos turnos) a ciegas intentando encontrar las celdas en las que están colocados los barcos del oponente. El resultado de los disparos debe ser mostrado al oponente.
- Si se dispara sobre agua se cambiará de turno.
- Si se dispara sobre un barco el jugador podrá volver a disparar. Si dispara sobre todas las casillas sobre las que está colocado un barco este se hundirá. Si un jugador hunde todos los barcos del rival ganará la partida y esta finalizará.

Sprint 1: Puesta en marcha del proyecto

Tareas de formación

El equipo tuvo que invertir un tiempo al principio del sprint en realizar unas tareas de formación necesarias para obtener conocimiento y así un correcto desarrollo del proyecto, ya que este no tenía los conocimientos necesarios de las tecnologías y los lenguajes que se iban a utilizar en el desarrollo.

Guía del desarrollador web

El equipo necesitaba aprender sobre más sobre el desarrollo web y sobre las tecnologías utilizadas. Para ello vieron la diferencia entre Backend (parte visual de un sitio Web o UI de una app), Frontend (APIs de servidor, datos, tareas, etc.), Full Stack (combina las tecnologías Front y Back) y los lenguajes utilizados en cada una de las partes.

Tutoriales de HTML, JavaScript, JQuery y NodeJS

Después de introducir al equipo con la guía del desarrollador web, este realizó unos tutoriales para entender los lenguajes necesarios a través de la página W3Schools, que es una herramienta web para aprender tecnologías, además cuenta con cuestionarios para asegurar que se han entendido los tutoriales a la perfección.

El primero realizado fue el de HTML, en el cual el equipo vio que es un lenguaje que sirve para gestionar el código que se utiliza para estructurar y desplegar una página web.

El segundo tutorial realizado fue el de JavaScript, en el que vio que este es un lenguaje de programación que se utiliza para hacer páginas web interactivas.

En tercer lugar, el equipo vio el tutorial relacionado con JQuery ya que esta es una biblioteca de JavaScript que nos permite simplificar operaciones y podría serle útil en un futuro.

Por último, el equipo realizó el tutorial de NodeJS ya que este proyecto se orienta a eventos asíncronos y este entorno era el más útil para ello.

Reunión con el cliente

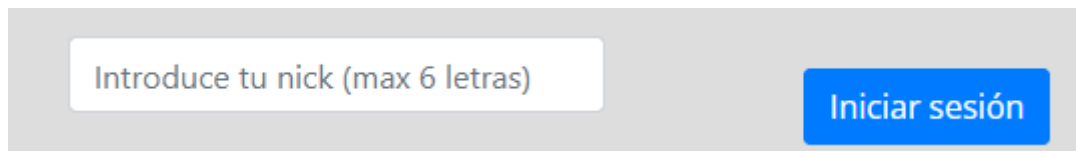
Después de la formación el equipo empezó con el desarrollo del proyecto creando el marco del cliente/navegador el cual giraría entorno al index.js que es el primer fichero que un navegador intentará leer al acceder a tu hosting. Para empezar con el proyecto primero tuvieron que reunirse con el cliente, el cual les especificó que quería una aplicación SaaS y un juego multijugador (para dos jugadores). Después de la reunión sacaron las siguientes historias de usuario:

- Iniciar sesión con un nick.
- Poder crear partidas y unirse con un nick.

Después de recoger estas especificaciones el equipo empezó a desarrollar el proyecto.

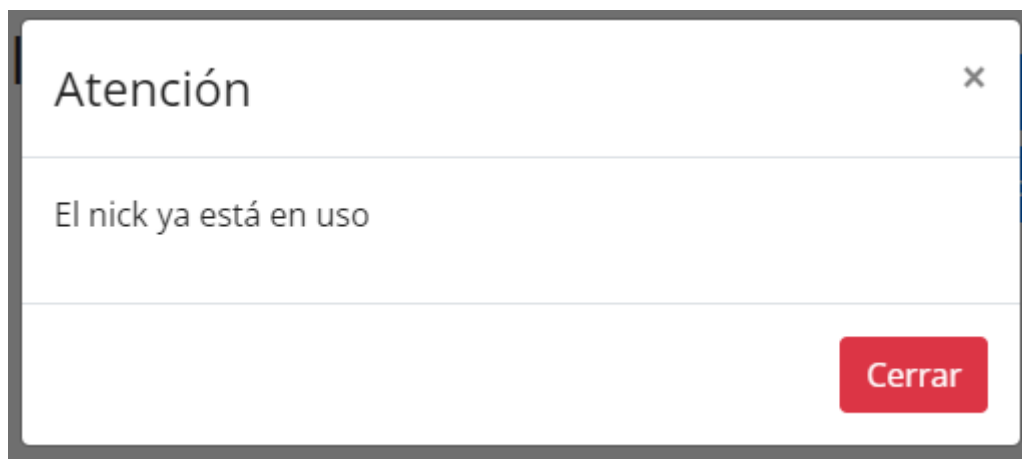
Inicio de sesión

El equipo empezó el proyecto desarrollando un inicio de sesión con el que no se pudiera repetir nick ya que a la hora de crear partida saldría el nick del usuario que la ha creado y en caso de repetirse el nick y que los dos crearan una partida no se podría saber que partida ha creado cada usuario. En la página se vería de la siguiente manera:



A login form with a light gray background. It features a white text input field with the placeholder text "Introduce tu nick (max 6 letras)". To the right of the input field is a blue button with the text "Iniciar sesión" in white.

Si dos usuarios intentan iniciar sesión con el mismo nick saldrá un mensaje de error en la pantalla y no se moverá a la pantalla de inicio en la que se podrán crear o unirse a partidas.

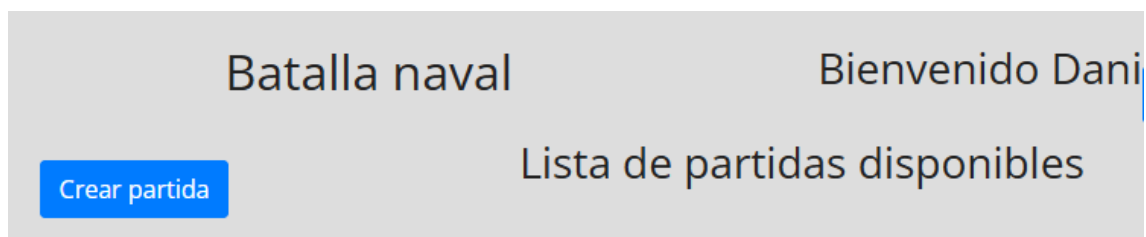


An error message dialog box with a white background and a gray border. The title bar says "Atención" with a close button (X) on the right. The main text says "El nick ya está en uso". At the bottom right is a red button with the text "Cerrar".

Poder crear partidas y unirse con un nick

Tras haber iniciado sesión, lo siguiente sería que el usuario pudiera crear partidas y que estas partidas sean visibles para los demás usuarios y de esta forma pudieran unirse para poder jugar juntos.

Una vez que el usuario inicia sesión se le mostrará la pantalla de inicio en la que se mostrará el botón para crear una partida, junto con una lista con las partidas disponibles.



A main menu screen with a light gray background. At the top left is the text "Batalla naval". At the top right is the text "Bienvenido Dani". Below "Batalla naval" is a blue button with the text "Crear partida". To the right of the button is the text "Lista de partidas disponibles".

En este caso no nos muestra ninguna partida disponible porque aún no se ha creado ninguna, pero si el usuario crea una partida irá a la fase de inicio de la partida en la que se le mostrará el código de la partida.

Batalla naval **Bienvenido Dani** Código de la partida: 1674344722574

A los demás usuarios se les mostrará la partida en la “Lista de partidas disponibles” para que puedan unirse solo con un clic.

Lista de partidas disponibles

Nick propietario: Dani

Revisión del Sprint 1

Tras completar el desarrollo de las historias de usuario mencionadas anteriormente, que mostraban lo acordado con el cliente hasta el momento, se probó que todo funcionaba correctamente y al obtener un resultado positivo se cerró el Sprint 1.

Documentación del Sprint 1

Después de la finalización del Sprint 1 realizaremos la documentación, que es lo realizado en este documento hasta este momento.

Sprint 2: Funciones de salir y abandonar partida

Reunión con el cliente

En la siguiente reunión con el cliente se sacaron las siguientes historias de usuario:

- Salir del juego.
- Abandonar partida.

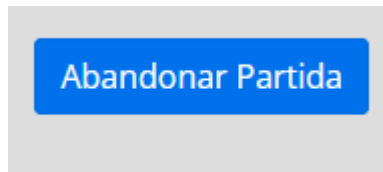
Salir del juego

Para poder salir del juego el equipo ha añadido un botón que estará en todas las pantallas del juego y de esta forma poder salir en cualquier momento, aunque esté en medio de una partida.

Salir

Abandonar partida

Para poder abandonar la partida hemos añadido un botón que se mostrará en la pantalla una vez que hayas creado la partida o te hayas unido a ella. Si seleccionas este botón no tendrás que volver a iniciar sesión, solamente el usuario se saldrá de la partida y nos llevará a la pantalla de inicio.



Implementación del despliegue continuo

Tras completar funcionalidades el equipo se dio cuenta de que necesitaba una implementación con despliegue continuo para no seguir mostrándole los cambios al cliente en "local", ya que de esta forma se realiza más rápidamente. Para realizar este despliegue primero realizó un curso de Cloud Run para aprender a utilizar la aplicación correctamente, ya que fue la que finalmente utilizaron y con la que obtuvieron el enlace con el que, solo con abrirlo, entraremos en el juego. Esta aplicación cogerá los datos de un repositorio de GitHub (solo con actualizar este repositorio se actualizará el contenido del enlace de Cloud Run).

Revisión del Sprint 2

Tras completar el desarrollo de las historias de usuario mencionadas anteriormente, que mostraban lo acordado con el cliente en la segunda reunión, se probó que todo funcionaba correctamente y al obtener un resultado positivo se cerró el Sprint 2.

Documentación del Sprint 2

Después de la finalización del Sprint 2 realizaremos la documentación, que es lo realizado en este documento hasta este momento.

Sprint 3: Definir la lógica GUI de la Batalla Naval

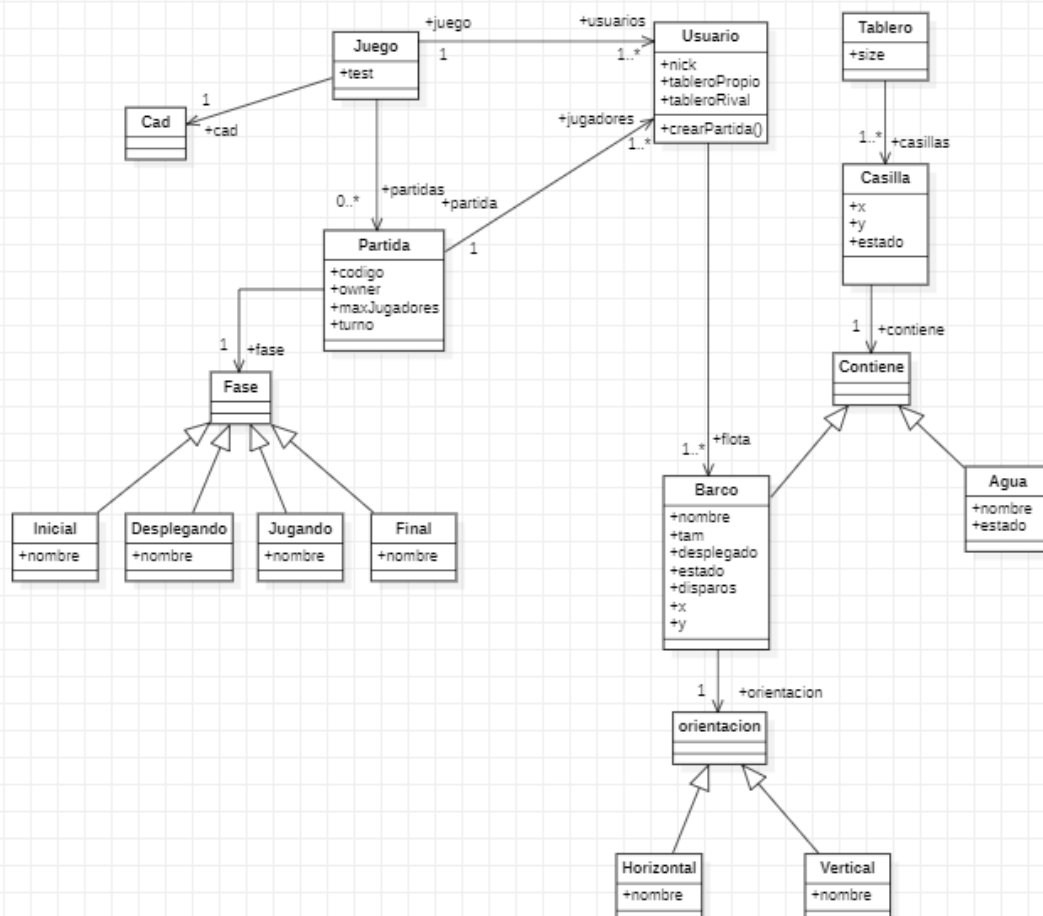
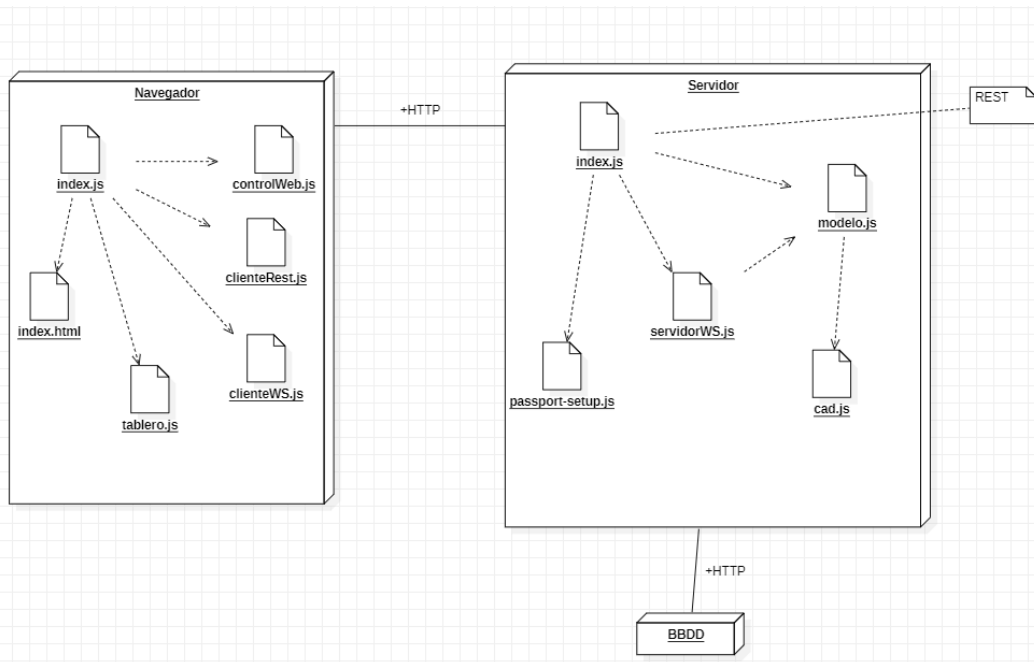
Reunión con el cliente

En la siguiente reunión con el cliente se sacaron las siguientes historias de usuario:

- Colocar barcos.
- Mostrar tablero propio y tablero rival.
- Disparar.
- Resultado final de la partida.

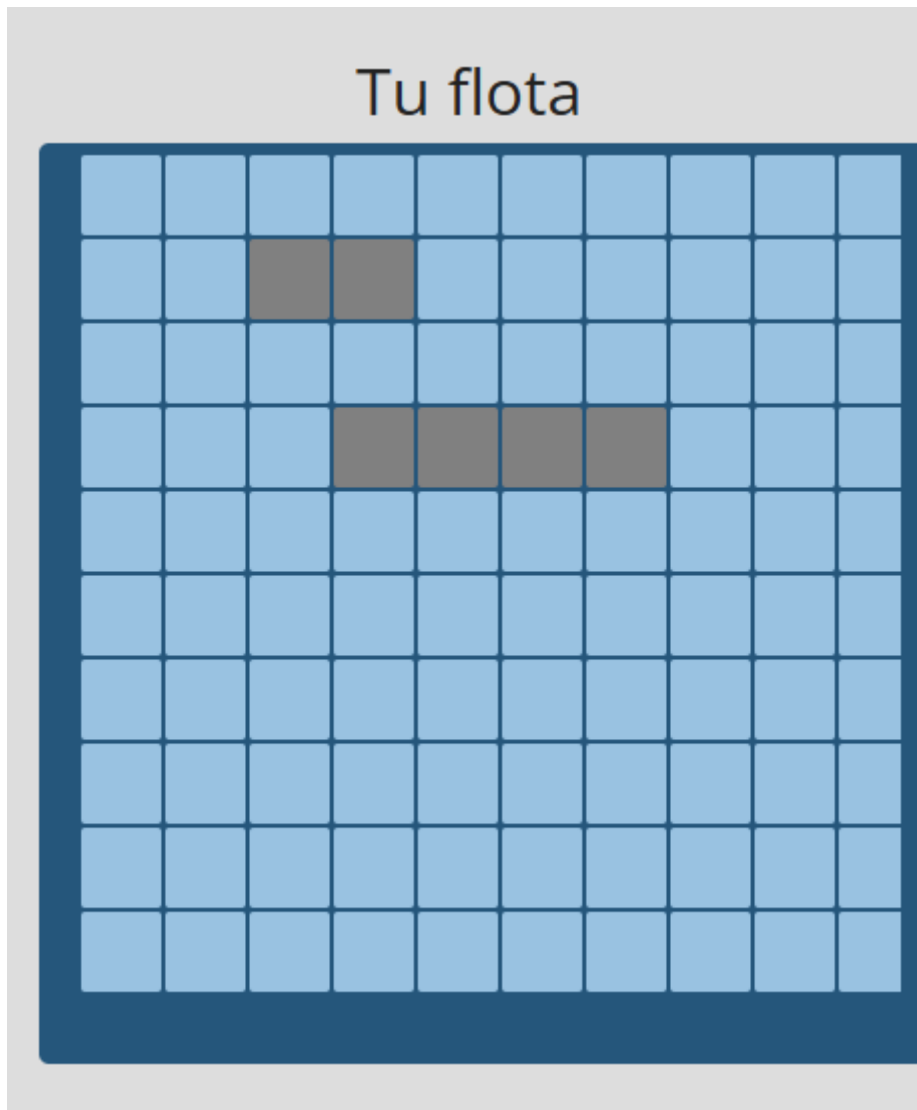
Diagramas

Antes de empezar a desarrollar estas historias de usuario tendremos que definir la lógica final del juego y retocar los últimos aspectos del diagrama de clases para poder realizar el juego correctamente. Estos serían los resultados:



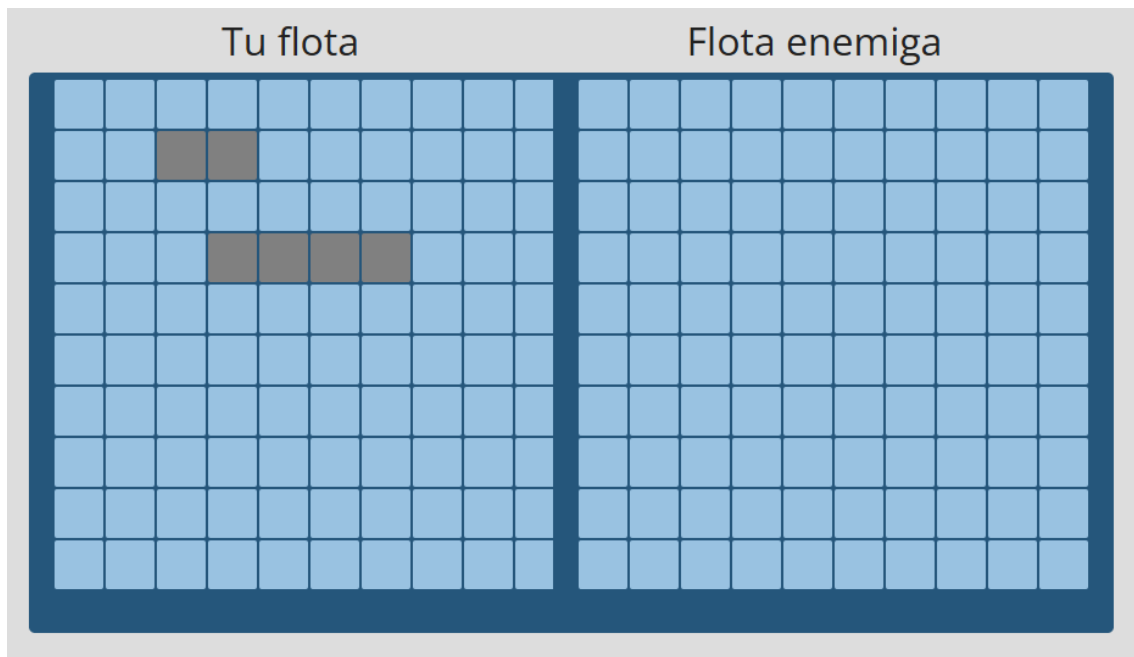
Colocar barcos

Primero el equipo desarrolló la función de colocar los barcos. Para ello se le muestra su tablero y los barcos. Para colocar cada barco tienen que seleccionarlo y después seleccionar la casilla del tablero a partir de la cual queremos que se coloque el barco. Este proceso deberemos seguirlo con cada barco hasta estar todos colocados. Como el cliente no nos había mencionado en la reunión nada sobre el tamaño del tablero, ni sobre la orientación y el tamaño de los barcos el equipo decidió establecer estos valores por su cuenta para poder probar el juego. El tamaño del tablero de 10x10, dos barcos de tamaño 2 y 4 casillas y ambos de orientación horizontal.



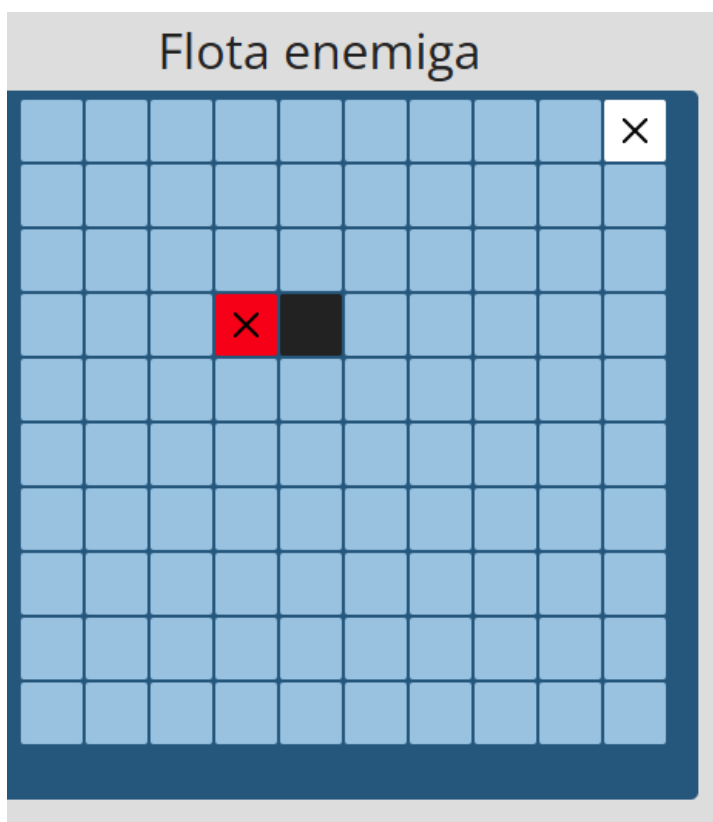
Mostrar tablero propio y tablero rival

Después de implementar las funciones necesarias para poder colocar los barcos y de esta forma tener el tablero propio listo, desarrollamos que se pudiesen ver los dos tableros, el propio y el del rival, donde verá los barcos colocados en su tablero.

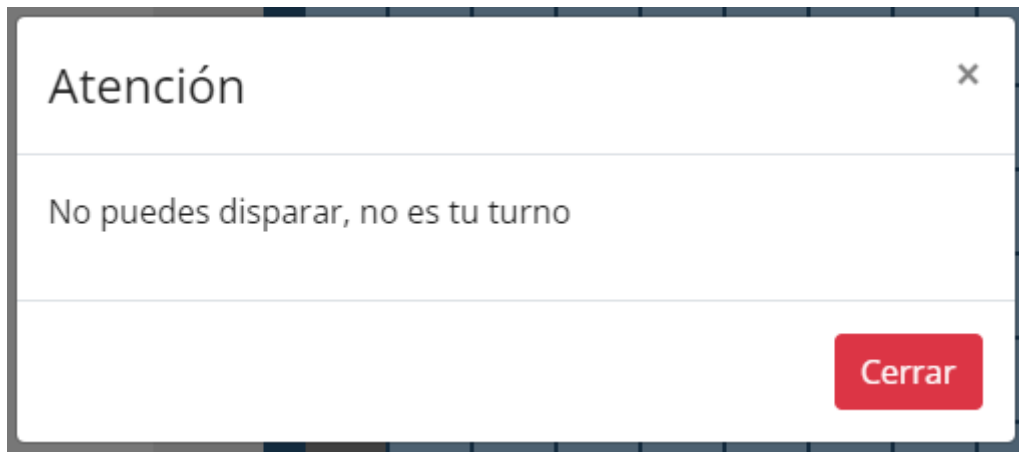


Disparar

Una vez que los dos jugadores tienen colocados todos los barcos el equipo desarrolló las funciones necesarias para poder disparar. Para ello se creó una función de asignar turno para saber de quién es el turno de disparar en cada momento. Cuando dispare el jugador que le toca deberá mostrar en el tablero rival las casillas disparadas siendo un cuadrado blanco si ha sido agua, uno rojo si un barco ha sido tocado y por último uno negro si el barco ha sido hundido.

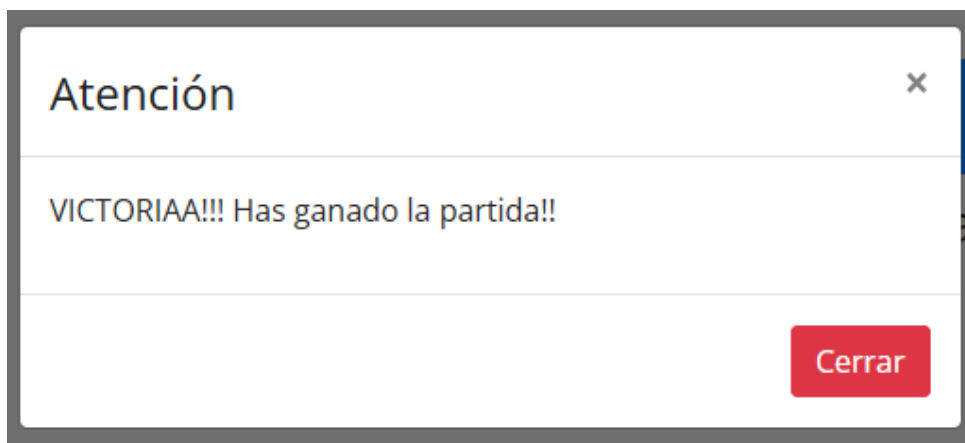


Si intenta disparar un jugador que no es su turno saldrá una pestaña de error avisando al usuario de que no es su turno.

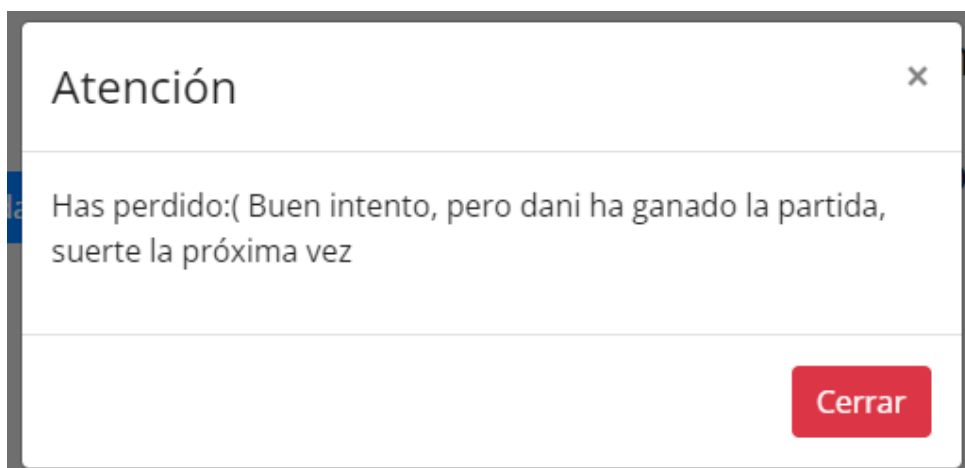


Resultado final de la partida

Una vez resueltos todas las funciones funcionadas anteriormente el equipo desarrolló las funciones necesarias para finalizar la partida y que los jugadores supiesen quién había ganado. Dependiendo de si has ganado o has perdido el sistema muestra un aviso con un mensaje distinto. Si has ganado muestra el siguiente mensaje:



Si has perdido el sistema te mostrará el siguiente mensaje:



Revisión del Sprint 3

Tras completar el desarrollo de las historias de usuario mencionadas anteriormente, que mostraban lo acordado con el cliente en la tercera reunión, se probó que todo funcionaba correctamente y al obtener un resultado positivo se cerró el Sprint 3.

Documentación del Sprint 3

Después de la finalización del Sprint 3 realizaremos la documentación, que es lo realizado en este documento hasta este momento.

Sprint 4: Persistencia

Reunión con el cliente

Después de esta reunión y el cliente ver el juego comentó que quería tener una base de datos donde guardar los usuarios que habían iniciado sesión y las partidas que se han jugado, junto con el resultado final. Una vez terminada la reunión con el cliente el equipo decidió utilizar MongoDB, que es una base de datos NoSQL orientada a documentos que ofrece una gran escalabilidad y flexibilidad. Una de las principales razones por las que decidió elegir esta base de datos es porque no se basa en tablas y columnas, sino, que los datos se almacenan como colecciones y documentos. Como se ha mencionado anteriormente en MongoDB se mostrarán las funcionalidades principales, que serán las siguientes:

- Inicio de sesión:

```
_id: ObjectId('63cd24c26aae1805541ec2cd')  
operacion: "inicioSesion"  
usuario: "Dani"  
fecha: "Sun Jan 22 2023 12:57:54 GMT+0100 (hora estándar de Europa central)"
```

- Crear partida:

```
_id: ObjectId('63cd24dc6aae1805541ec2cf')  
operacion: "crearPartida"  
propietario: "Dani"  
codigo: 1674388700362  
fecha: "Sun Jan 22 2023 12:58:20 GMT+0100 (hora estándar de Europa central)"
```

- Unirse a una partida:

```
_id: ObjectId('63cd24ea6aae1805541ec2d2')  
operacion: "unirsePartida"  
usuario: "Pepe"  
codigoPartida: "1674388711373"  
fecha: "Sun Jan 22 2023 12:58:34 GMT+0100 (hora estándar de Europa central)"
```

- Resultado del final de la partida:

```
_id: ObjectId('63cd25336aae1805541ec2d6')
operacion: "finalizarPartida"
perdedor: "Dani"
ganador: "Pepe"
codigo: 1674388728037
fecha: "Sun Jan 22 2023 12:59:47 GMT+0100 (hora estándar de Europa central)"
```

- Abandonar partida:

```
_id: ObjectId('63cd24e46aae1805541ec2d0')
operacion: "abandonarPartida"
usuario: "Dani"
codigo: 1674388700362
fecha: "Sun Jan 22 2023 12:58:28 GMT+0100 (hora estándar de Europa central)"
```

- Salir del juego:

```
_id: ObjectId('63cd253f6aae1805541ec2d9')
operacion: "finSesion"
usuario: "Dani"
fecha: "Sun Jan 22 2023 12:59:59 GMT+0100 (hora estándar de Europa central)"
```

Revisión del Sprint 4

Tras completar el desarrollo de la base de datos mencionada por el cliente en la cuarta reunión, en la que se muestran las funciones principales de la aplicación, se probó que todo funcionaba correctamente y al obtener un resultado positivo se cerró el Sprint 4.

Documentación del Sprint 4

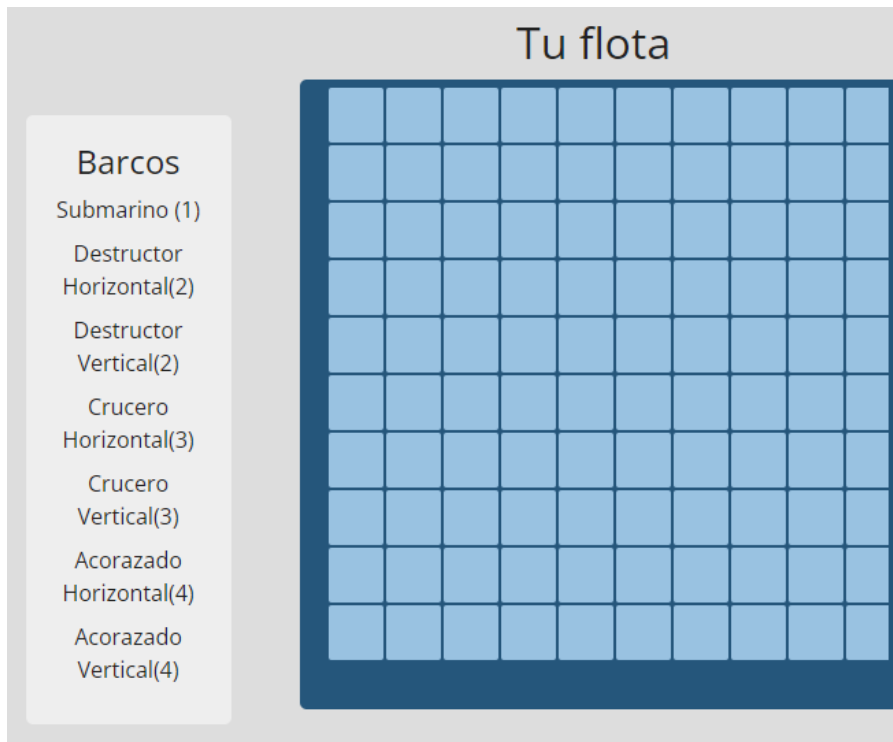
Después de la finalización del Sprint 4 realizaremos la documentación, que es lo realizado en este documento hasta este momento.

Sprint 5: Barcos con orientación vertical

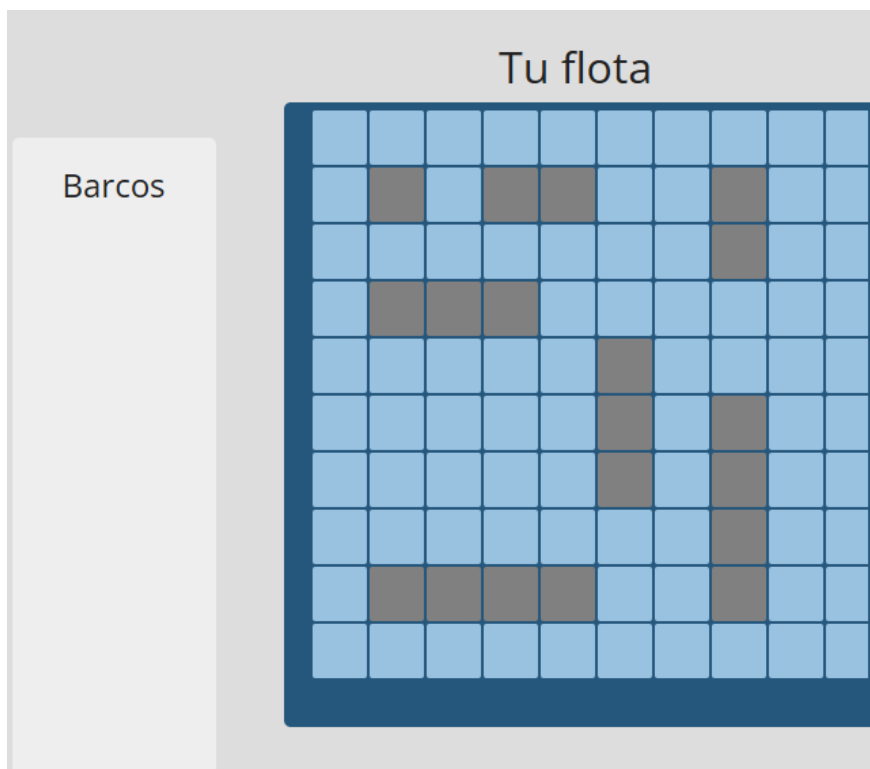
Reunión con el cliente

En la última reunión el cliente comentó que estaba de acuerdo con el tamaño que el equipo eligió para los tableros, pero que quería que hubiera más barcos, en concreto siete, siendo el primero de 4 tamaños distintos, desde 1 hasta 4 casillas concretamente, siendo el de 1 casilla único y dos del resto de tamaños, teniendo cada uno su pareja, uno horizontal y el otro vertical ya que así el juego sería más complicado y de esta forma más entretenido. Después de tener esta reunión el equipo especificó las siguientes historias de usuario:

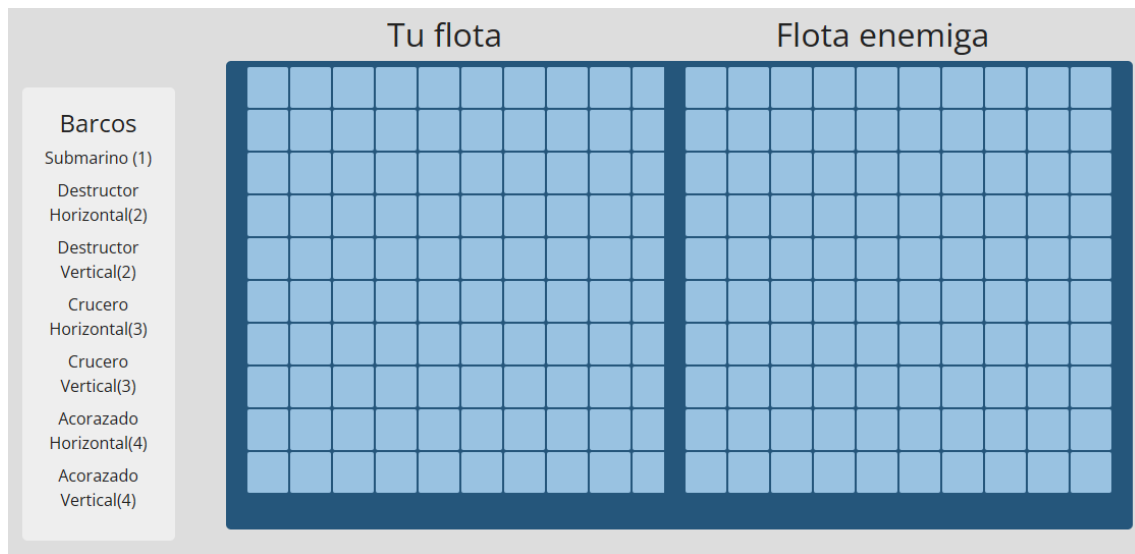
- Siete barcos de tamaños entre 1 y 4 casillas.
- Dos orientaciones posibles para los barcos (horizontal y vertical).



Una vez que los barcos estén colocados la lista de barcos estará vacía y todos aparecerán en el tablero de nuestra flota en la posición que los hayamos colocado.



En el resultado final de este Sprint se vería la lista de barcos, junto con el tablero propio y el tablero rival (su funcionamiento seguirá siendo el mismo que especificamos en el Sprint 3). Esto se vería de la siguiente manera de la siguiente manera:



Pruebas unitarias

Una vez realizado todo el funcionamiento que el cliente solicitó el equipo comenzó con las pruebas unitarias donde se comprobó que todo funcionaba correctamente. En estas pruebas se comprobaron todas las funciones del juego y el resultado fue el siguiente:

```
Finished in 0.115 seconds  
19 tests, 307 assertions, 0 failures, 0 skipped
```

Revisión del Sprint 5

Tras completar el desarrollo de las historias de usuario mencionadas anteriormente, que mostraban lo acordado con el cliente en la quinta reunión, y comprobar que las pruebas unitarias pasaban sin ningún error, se probó que todo funcionaba correctamente y al obtener un resultado positivo se cerró el Sprint 5.

Documentación del Sprint 5

Después de la finalización del Sprint 5 realizaremos la documentación, que es lo realizado en este documento hasta este momento.

Anexos

Enlace de Cloud Run

<https://proyecto22-3ru5xmyj3a-no.a.run.app>