Informe Proyecto 2 Parte 1

Batalla Naval

Estudiantes:

Landau Nicolas

Puhl Juan Manuel

Materia:

Sistemas Distribuidos

# Problema resuelto

Este proyecto consiste en diseñar e implementar el juego de la Batalla naval para 2 jugadores ubicados en diferentes maquinas. Cuando un jugador está pensando que elemento elegir el otro jugador no puede realizar ninguna actividad en el juego. Cada jugador debe poder visualizar las 2 grillas en la pantalla al mismo tiempo.

Consideraciones:

1. Cualquier jugador puede realizar la invitación para comenzar el juego.
2. Antes de iniciar el juego se deben poner de acuerdo los componentes de la otra naval.

# Capa de interfaz

## Librería utilizada

Para la implementación de la capa de interfaz se utilizó la librería GTK3, junto con glade para poder agregar cosas fácil y rápidamente.

Para instalar todo, los pasos son:

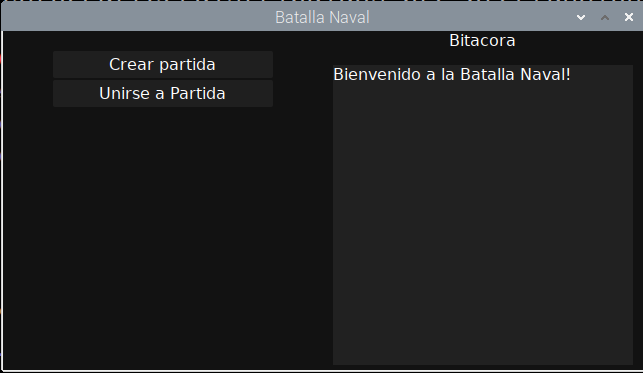
* sudo apt-get install glade

Si no se pudo instalar

* sudo apt-get update --fix-missing

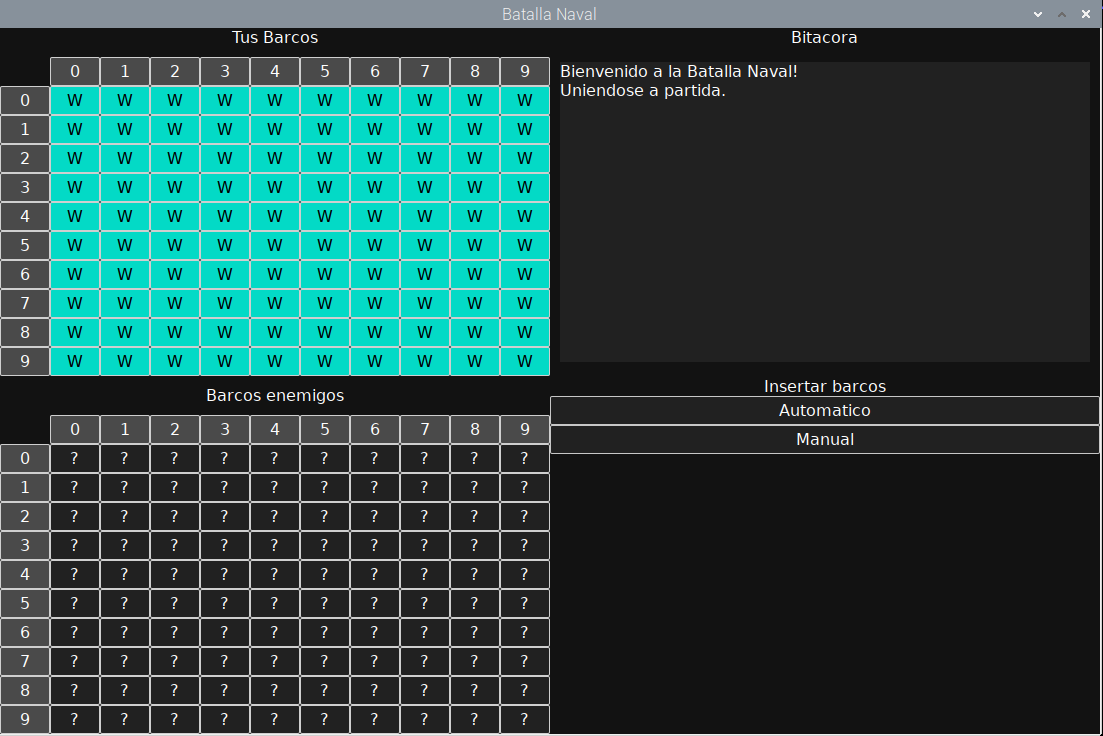
Una vez finalizado se debe instalar nuevamente el glade.

## Creación de partida



Inicialmente, tendremos dos botones, crear partida y unirse a la partida. Para crear se necesita ingresar el puerto, y para unirse se necesita la ip y el puerto. Si no se ingresa nada se establecerán los valores por defecto (IP: 127.0.0.1, PORT: 14550). Cuando ambos participantes hayan ingresado a la partida, se actualizará la pantalla.

## Posicionamiento de barcos



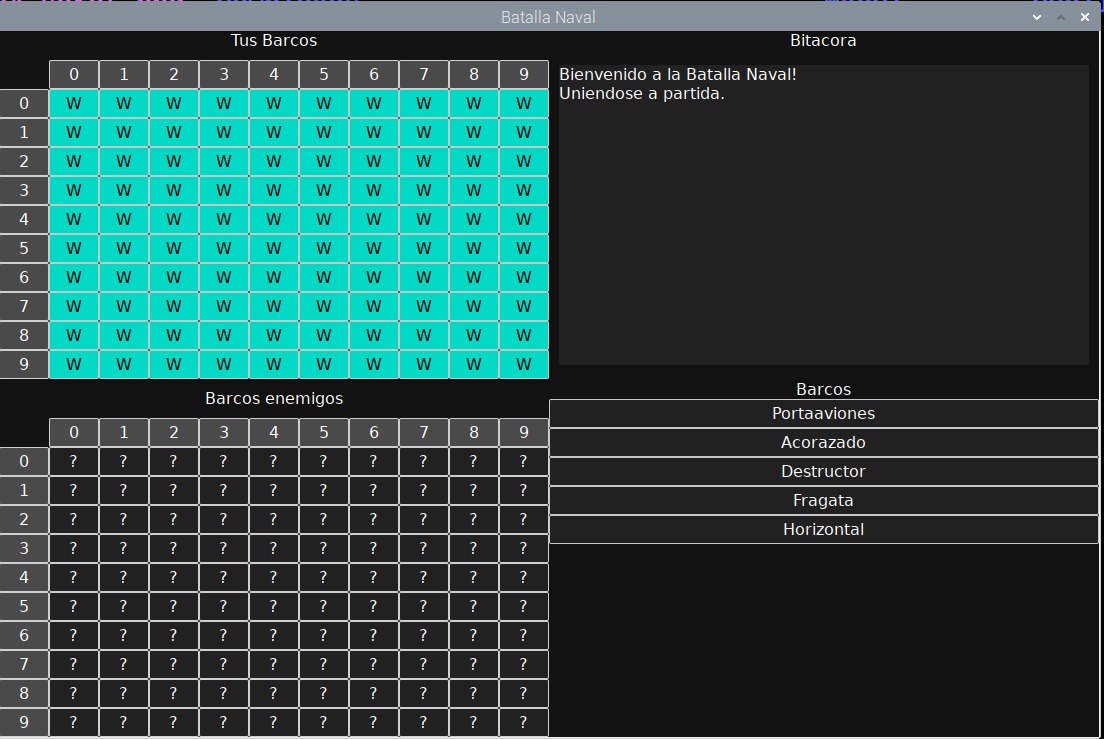
La pantalla se compone de 2 tablas, una de posición (arriba) y una de ataque (abajo). Además, a la derecha se encuentra una bitácora que lleva registro de todos los ataques y los resultados.

Para iniciar la partida, los jugadores deberán elegir si sus naves se insertan automáticamente o de forma manual. Si se elige manual, entonces aparecerá un panel con los barcos a insertar. Se debe hacer click en uno y luego en la tabla de posición, en la ubicación que se desee (Tener en cuenta que la celda indicada será la primera del barco, tanto vertical como horizontalmente). Además, se puede elegir la orientación del barco.

Cuando se ubican los barcos, se mostrará en la tabla de ubicación cada uno, cambiando el color de la celda correspondiente.

Tanto para la tabla de posición como de ataque utilizamos el mismo código de color:

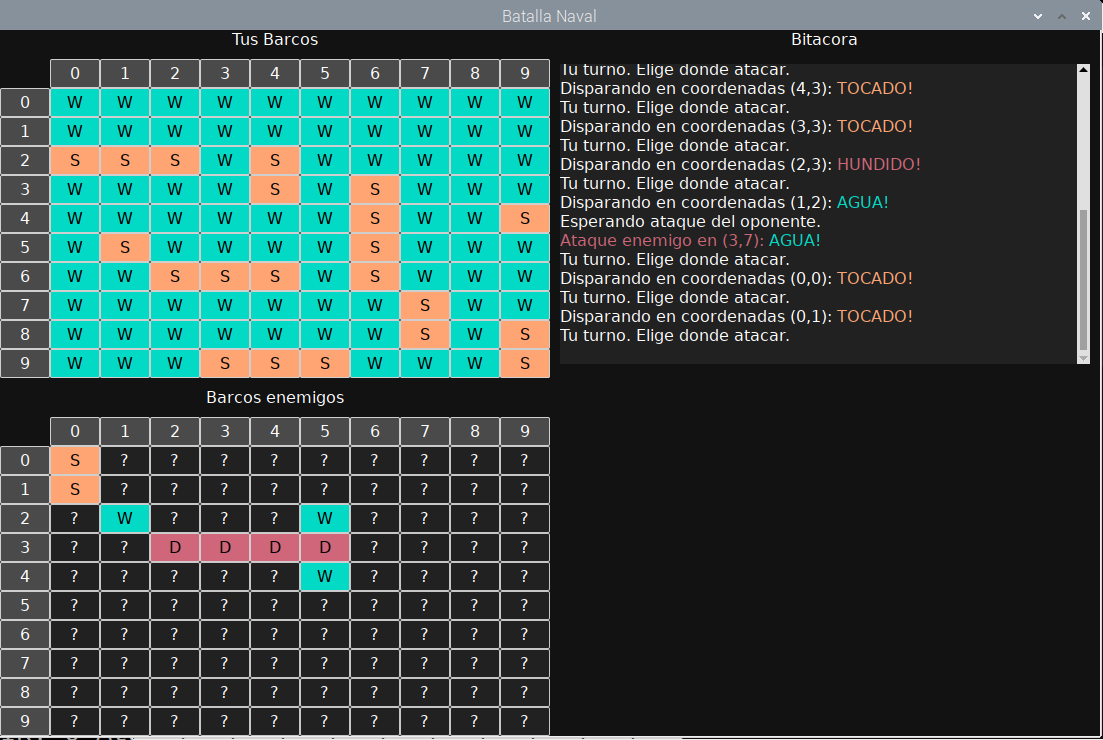
* Naranja: indica que en esa posición hay un fragmento de barco sin dañar.
* Rojo: Indica que en esa posición hay un fragmento de barco hundido. (Cuando el barco entero se hundió)
* Celeste: Indica la presencia de agua en esa coordenada.



## Ataque

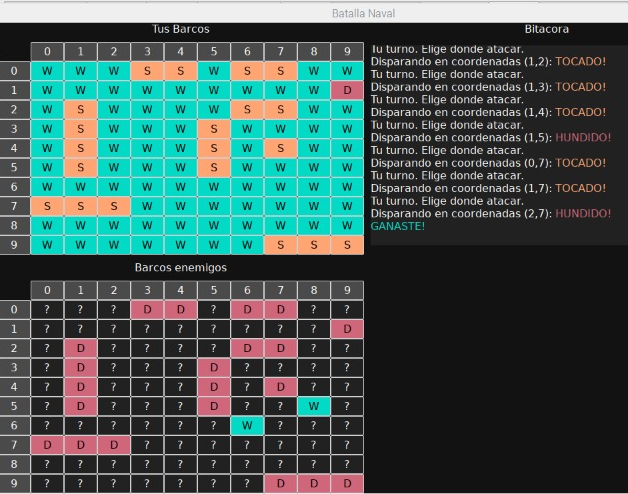
Una vez que ambos jugadores hayan ingresado sus barcos, la partida comenzara, siendo el primer turno de ataque para aquel jugador que haya creado la partida.

Para atacar hay que hacer click en la posición deseada en la tabla de ataque, la bitácora se actualizara con las coordenadas, y luego de unos momentos, el resultado del ataque aparece en pantalla, pintando del color correspondiente la coordenada.

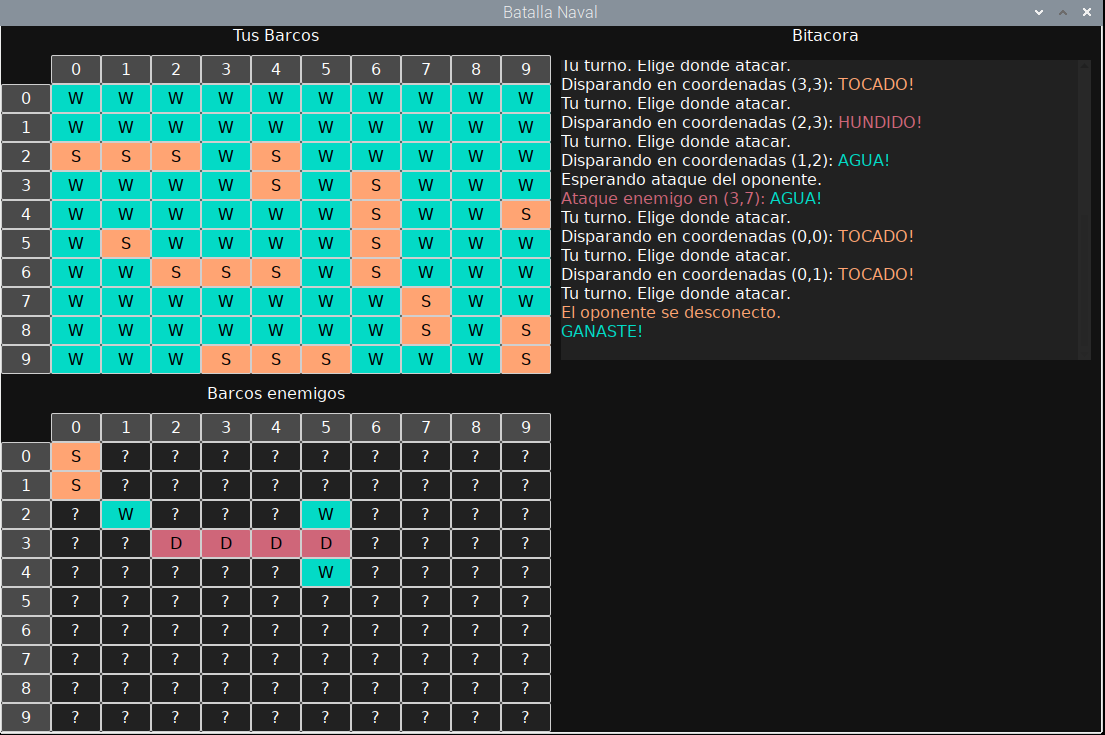


Si se falla el ataque el turno pasa al oponente, el cual puede atacar y el procedimiento es el mismo.

Cuando se ha hundido la flota entera del oponente aparece un mensaje que indica que has ganado el juego, en caso contrario, si el oponente hundió toda tu flota, ¡aparece el mensaje “PERDISTE!”.



Para cerrar el juego solo hay que presionar el botón x en la esquina superior derecha de la ventana. Si en algún momento del juego se presiona, se cierra la conexión y el otro jugador gana por abandono, indicándole además que el oponente se desconectó.



# Diseño e implementación

## Modelo

Se opto por el modelo peer-to-peer, donde cada jugador solo tiene información completa de su propio tablero y para obtener el resultado de un disparo necesita preguntarle al otro jugador.

## Módulos

La solución está compuesta por 6 módulos

* gui.c es el módulo central, en este está definido main(). En este módulo se realiza toda la entrada y salida y se progresa el juego según su estado.
* battleship.c contiene la lógica del juego en sí. Aquí se definen las estructuras utilizadas en el juego como el gamestate, el barco o lo tableros. Además, las funciones para colocar correctamente barcos, evaluar disparos y evaluar el estado del juego.
* connection.c es el encargado de todo lo relacionado a la conexión con el oponente. Aquí se encuentran funciones para crear o unirse a una partida, para enviar mensajes relacionados a los disparos o para desconectarse.
* init.c es el encargado de realizar la inicialización de las estructuras del juego.
* utils.c es un pequeño modulo que aloja algunas funciones de utilidad general para los otros módulos, como conversión entre int y char.
* gui.glade es donde se define la interfaz gráfica.

Una visión más detallada de los módulos, funciones, estructuras, miembros y gráficos de dependencias se puede obtener a partir de la documentación entregada junto a este informe. También, claro está, puede verse en el código fuente mismo.

## Comunicación

Para la comunicación entre computadoras se utilizamos sockets. Si se opta por crear una partida, se crea un socket y queda a la escucha de un oponente. Si se intenta unir a una partida, se crea un socket e intenta conectarse a la dirección y puerto ingresados por el usuario. Una vez establecida la conexión, no hay diferencia entre el que creo la partida y el que se unió. No hay noción de cliente ni de servidor, ambos jugadores son iguales.

Debido a que el juego avanza completamente por turnos, no fue necesario crear un protocolo de comunicación, ya que siempre que espero un mensaje, se exactamente qué tipo de mensaje me va a llegar.

## Implementacion

Los archivos fuente han sido enviados a la catedra junto a este documento. Además, un repositorio con los mismos está disponible en github.com/nlandau91/SD\_Batalla\_Naval

## Instalación

Estos pasos están pensados para distribuciones basadas en Debian

Primero, es necesario instalar glade:

* $ sudo apt-get update -y
* $ sudo apt-get install -y glade

En caso de error:

* $ sudo apt-get update --fix-missing

Y volver a intentar los comandos anteriores

Una vez instalado glade, utilizar el script de make provisto:

* $ make

Y ahora solo queda ejecutar el programa

* $ ./ejecutable