INSTITUTO TECNOLÓGICO DE ESTUDIOS SUPERIORES DE OCCIDENTE

**Proyecto de programación estructurada**

**“Minijuegos”**

|  |  |
| --- | --- |
| Materia: | Programación Estructurada |
| Participantes: | Luis David Bucio Macias.  Pablo Gerardo Zarate Félix |
| Profesor: | Francisco Cervantes |
| Fecha: | 09/12/19 |

Documento del Proyecto Final

Documento de definición

# Integrantes

LUIS DAVID BUCIO MACÍAS

PABLO GERARDO ZARATE FELIX.

# Resumen del proyecto

Para este proyecto buscamos realizar un programa donde haya dos juegos, uno de batalla naval y otro de buscaminas, en el cual buscaremos incorporar nuestros conocimientos del curso, así como cosas que hayamos investigado por nuestra cuenta para mejorar el juego.

Esperamos que realizar este proyecto reafirme nuestro conocimiento y nos ponga en contacto con otro tipo de herramientas que no alcanzaríamos a ver durante el curso. Como equipo tenemos los conocimientos y la organización necesaria para sacar adelante este proyecto.

# Contexto y terminologías del proyecto

1. Objetivo del juego de Batalla Naval:

Hundir los 9 barcos de tu oponente antes que él acabe con los tuyos.

Cada jugador tiene 2 tableros compuesto por 10 filas y 10 columnas.

Hundir todos los barcos del enemigo antes de que él lo haga.

Cada jugador tiene 9 barcos que ocuparan determinadas casillas en el tablero.

• 1 portaaviones: ocupa 4 casillas

• 3 submarinos/ Acorazados: ocupan 3 casillas. Son aleatorios, es decir, puedes tener 3 submarinos y 0 acorazados, 2 submarinos y 1 acorazado...etc.

• 3 destructores: ocupan 2 casillas.

• 2 fragatas: ocupan 1 casilla.

Terminología y movimientos:

• Agua: cuando disparas sobre una casilla donde no está colocado ningún barco enemigo, disparas al agua.

• Tocado: Cuando disparas en una casilla en la que está ubicado un barco enemigo que ocupa 2 o más casillas y destruyes sólo una parte del barco, le has tocado

• Hundido: Cuando disparas correctamente a todas las casillas de un barco

Reglas. -

• Número de jugadores: 2

• Una vez posicionas tus barcos y comienzas la partida, no podrás volver a cambiarlos de posición.

• Podrás disparar en cualquier casilla del tablero, salvo en las que ya has disparado.

• No puedes deshacer disparos ni propios ni de tus oponentes.

• Es un juego por turnos: haces tu disparo, si es "agua" el turno pasa a tu oponente; si "tocas" y/o hundes un barco enemigo, vuelves a disparar.

• La partida acaba cuando un jugador ha hundido la flota completa del enemigo.

2.- Objetivos juego de Buscaminas

El objetivo del juego es despejar un campo de minas sin detonar ninguna.

Reglas. -

* El juego consiste en despejar todas las casillas de una pantalla que no oculten una mina.
* Algunas casillas tienen un número, el cual indica la cantidad de minas que hay en las casillas circundantes. Así, si una casilla tiene el número 3, significa que de las ocho casillas que hay alrededor (si no es en una esquina o borde) hay 3 con minas y 5 sin minas. Si se descubre una casilla sin número indica que ninguna de las casillas vecinas tiene mina y éstas se descubren automáticamente.
* Si se descubre una casilla con una mina se pierde la partida.
* Se puede poner una marca en las casillas que el jugador piensa que hay minas para ayudar a descubrir las que están cerca.

Referencias:

https://es.wikipedia.org/wiki/Buscaminas

https://www.retos.com/juego/reglas?id=6

A nivel de código tendrá muchas similitudes con juegos que utilizan coordenadas par localización de objetivos o trampas por ejemplo como Memorama, ajedrez, etc.

# Alcance del Proyecto

## Lo que sí se incluye

Batalla Naval:

* Juego funcional de batalla naval.
* *Mapa en base a coordenadas.*
* *3 tamaños de mapa*
* *3 dificultades*
* *Contador de intentos*
* *Jugar contra la maquina*
* *Jugar contra otro jugador*
* *Barcos de 1x1*
* *ASCII ART*

*Buscaminas*

* *Juego funcional de Buscaminas*
* *Mapa en base a coordenadas*
* *3 tamaños de mapa*
* *3 dificultades*
* *Ganas al revelar todas las casillas que no tengan bomba*
* *Pierdes al revelar una casilla con bomba*
* *ASCII ART*

## Lo que no se incluye

Batalla Naval:

* Longitud de barcos (varias casillas con un solo barco)
* Que la maquina te ataque

Buscaminas:

* La opción de poner una flecha como en el juego original donde creas que haya una mina
* Números en las casillas cercanas a una bomba como el juego original

## Riesgos posibles y medidas de prevención

Creemos que lo más complicado será desarrollar el sistema de Servidor Cliente, ya que es un tema bastante avanzado y que no vimos en el curso por lo que tendremos que investigar todo por nuestra cuenta.

También creemos que se nos puede complicar el sistema de posicionamiento de los barcos con base en coordenadas.

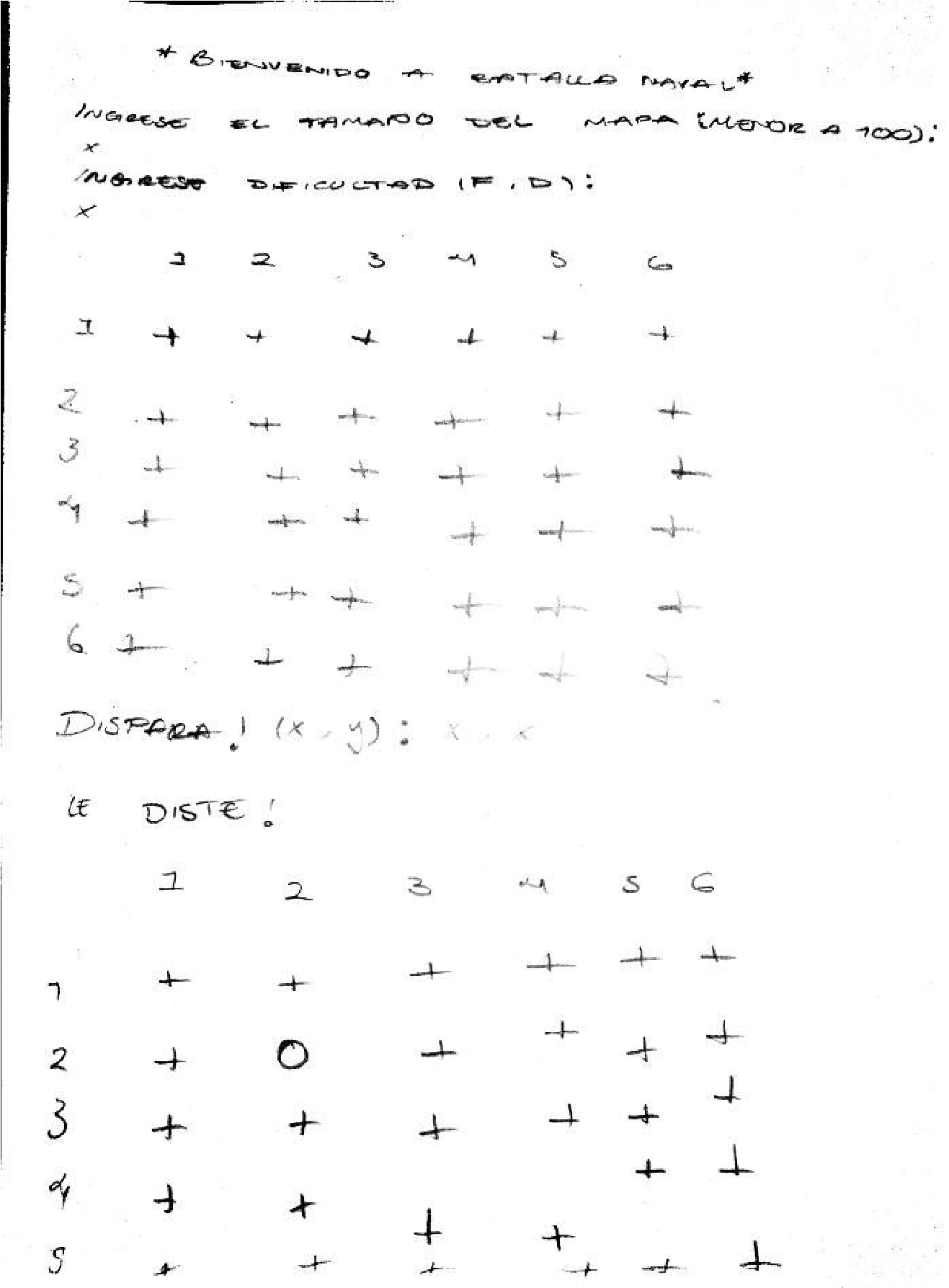
Las prevenciones que tomaremos será investigar bien los temas, mediante páginas de internet o videos, así como ordenarnos para que luego no nos falte el tiempo.

**6.- Hipótesis de solución y borradores de interfaz**

**Análisis del problema:** Al finalizar nuestro programa esperamos haber creado 2 juegos, uno de batalla naval y otro de buscaminas, en el cual el usuario pueda ingresar la dificultad del juego, la cual afectará la cantidad de barcos o minas en el espacio y además el usuario podrá ingresar el tamaño del espacio. Así mismo también que el juego de batalla naval pueda jugarse entre dos personas y no solo contra la máquina.

**Solución propuesta:** La solución de este problema la pensamos hacer separando cada cosa del juego mediante funciones y llamarlas en la función main para que todo sea mas ordenado y más fácil al momento de juntar todo y crear el juego, así como tomar las medidas adecuadas para que no nos falte tiempo. Aparte para el problema del cliente, servidor la solución será dedicar bastante tiempo en leer y entender la documentación de las librerías externas que vamos a ocupar

**Borradores de interfaz:**



**Diagramas de flujo**

# Herramientas y recursos necesarios

* Librerias stdio.h, stdlib.h, time.h, winsock2.h y la libreria externa ws2\_32

● Ciclos:

○ For para llenar nuestros arreglos

○ For Para recorrer nuestras matrices

* Condicionales
* Funciones

|  |  |  |  |
| --- | --- | --- | --- |
| **Herramienta o Recurso** | **Descripción y utilidad en el proyecto** | **Links a tutoriales y documentación** | **¿Qué harán para aprender a usarlo?** |
| **Librería stdio.h y stdlib.h** | **Básicamente es la librería principal que se suele usar en C para las funciones como printf y scanf** | [**https://es.wikipedia.org/wiki/Stdlib.h**](https://es.wikipedia.org/wiki/Stdlib.h)  **https://es.wikipedia.org/wiki/Stdio.h** | **Creemos que ya sabemos usar esta librería por todo lo que hemos visto en el curso** |
| **Librería time.h** |  | [**https://es.wikipedia.org/wiki/Time.h**](https://es.wikipedia.org/wiki/Time.h)  [**https://www.tutorialspoint.com/c\_standard\_library/time\_h.htm**](https://www.tutorialspoint.com/c_standard_library/time_h.htm)  [**https://www.geeksforgeeks.org/time-h-header-file-in-c-with-examples/**](https://www.geeksforgeeks.org/time-h-header-file-in-c-with-examples/) | **Viendo videos de YouTube de como se usa en diferentes casos y leyendo documentación de esta librería en internet.** |
| **Winsock2.h** |  | [**https://gist.github.com/piscisaureus/906386**](https://gist.github.com/piscisaureus/906386)  **https://docs.microsoft.com/es-mx/windows/win32/api/winsock2/nf-winsock2-wsasetblockinghook** | **En esta librería no hay muchos videos por lo que nos tocara leer mucha documentación de esta.** |
| **ws2\_32** |  |  |  |
| **Funciones** | **Haremos funciones para cada cosa que haga el programa para tener más ordenado el juego y el programa mismo.** | [**https://es.wikibooks.org/wiki/Programación\_en\_C/Uso\_de\_funciones**](https://es.wikibooks.org/wiki/Programación_en_C/Uso_de_funciones)  [**https://www.mheducation.es/bcv/guide/capitulo/8448148681.pdf**](https://www.mheducation.es/bcv/guide/capitulo/8448148681.pdf)  **http://cursos.iteso.mx/pluginfile.php/1342871/mod\_resource/content/2/Sesión%2016.pdf** | **Creemos que esta ya sabemos usarla bastante bien, por lo visto en este curso y en anteriores, pero de todos modos veremos algunos videos para casos específicos** |

# Resultados

## Contenido del paquete entregable (.zip)

En el zip estarán las siguientes cosas:

El archivo .c del proyecto donde vendrá el código que hicimos.

El .exe del proyecto el cual servirá para abrir el programa e iniciarlo para poder jugar cualquiera de los dos juegos.

El ws2\_32.dll que es la librería externa del winsock2.h

## Instrucciones de compilación y uso

Para compilar el programa se necesita abrir el .c en algún ID y usar un compilador.

Se utilizo una librería externa llamada ws2\_32, este se debe agregar en las opciones del compilador y estando ahí en las librerías del compilador agregar en la zona de Librerias -l lo siguiente “Ws2\_32” sin las comillas.

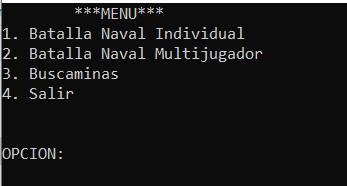
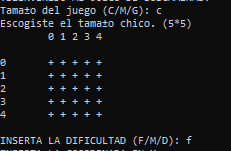
Al iniciar el programa (el .exe) aparecerá un menú donde el usuario podrá escoger entre 4 opción, jugar batalla naval individual, jugar batalla naval multijugador, jugar buscaminas o salir. El usuario deberá ingresar el numero alado de la opción para ingresar a cada opción. Una vez que el usuario ingrese el numero mediante el teclado el programa lo mandara a la opción ingresada. En el caso de los 3 juegos la mecánica es similar, se le pedirán coordenadas al usuario (x, y) y este deberá ingresarla mediante el teclado.

Hay un error que conocemos si el usuario ingresa una coordenada más alta a la del tablero mostrada se contara como fallo y se le restara las oportunidades.

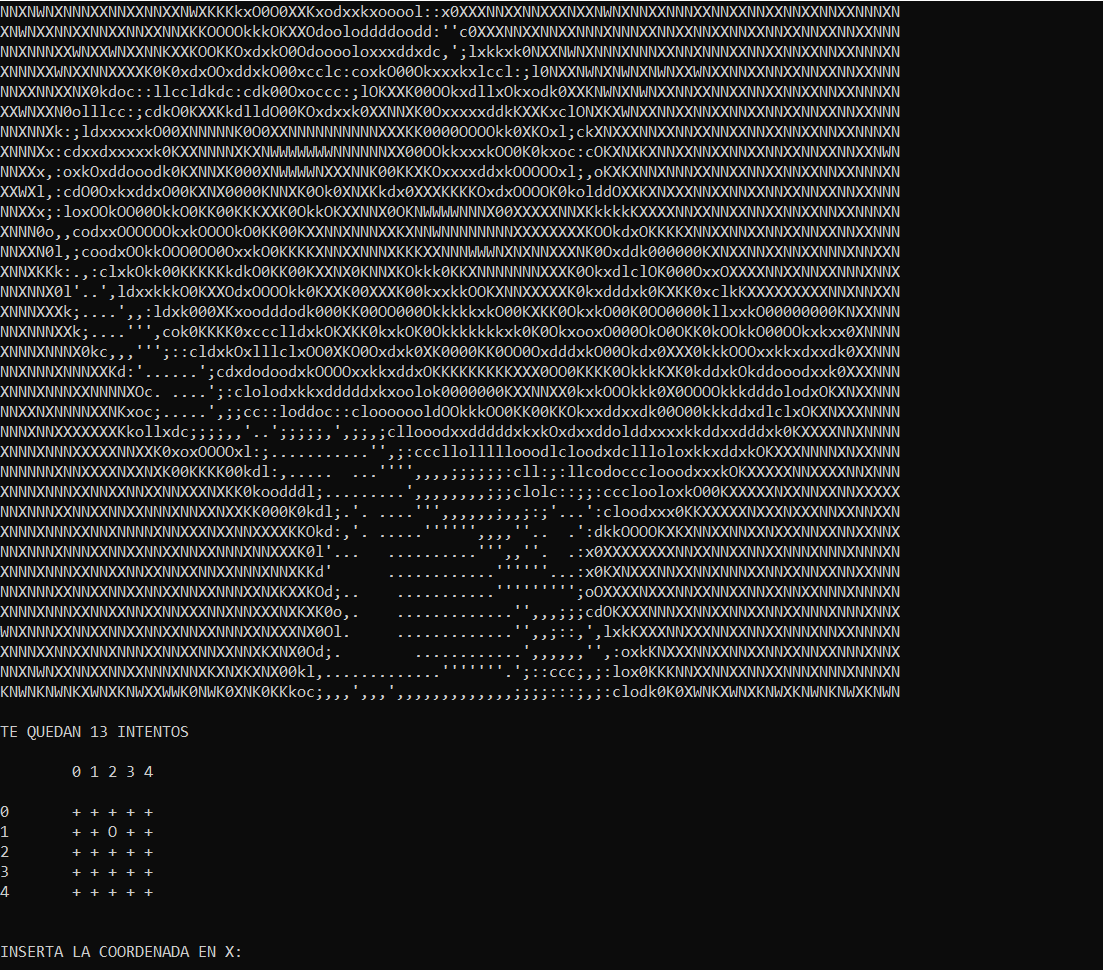
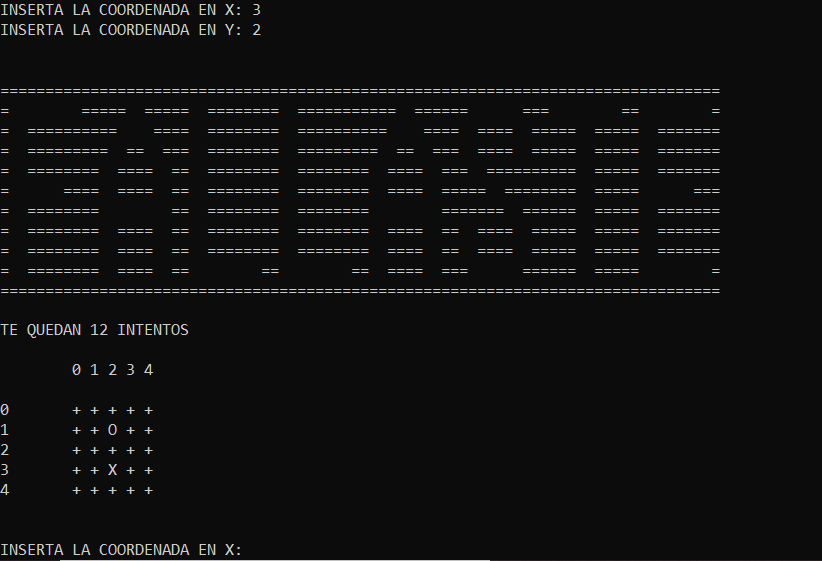
Para la parte del modo multijugador

## Ejemplos de Uso

La opción de escoger el tamaño de juego y la dificultad

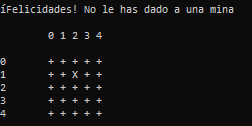
El menú donde se le da la opción al usuario.

Cuando pierde el juego Y te regresa al menú principal.

Cuando le das a un barco en el juego de Batalla Naval

Cuando fallas el disparo en Batalla Naval

Cuando le das a una mina en buscaminas

Cuando no le das a una mina

Pruebas Realizadas

1. La primera prueba fue probar que cuando se te acaben los intentos en batalla naval contra la maquina pierdas y te regrese al menú principal. La prueba fue exitosa
2. La segunda prueba fue la condición de victoria de batalla naval, mediante un if se prueba que cuando le des a todas las casillas donde hay un barco te muestre un mensaje que ganaste y te regrese al menú principal. La prueba fue exitosa
3. La tercera prueba fue en buscaminas que cuando le des a una mina de regrese que perdiste y te mande al menú principal. La prueba fue exitosa
4. La cuarta prueba fue la condición de victoria en buscaminas, mediante un if que resta las casillas totales menos las minas que hay y si se cumple con un contador te muestra que ganaste y te lleve al menú principal. La prueba fue exitosa.
5. La quinta prueba es la batalla naval multijugador

# Conclusiones

Cada miembro del equipo deberá escribir su experiencia durante el proyecto. Intenten plasmar pensamientos, aprendizajes y reflexiones significativas; esta reflexión es más para ustedes que para el profesor: **pensar en cómo aprendieron les hará aprender mejor la próxima vez**.