



**INSTITUTO POLITÉCNICO NACIONAL
UNIDAD PROFESIONAL
INTERDISCIPLINARIA DE INGENIERÍA
Y CIENCIAS SOCIALES
Y ADMINISTRATIVAS.**



INGENIERÍA EN INFORMÁTICA.

Algoritmos computacionales

MANUAL TÉCNICO.

Integrantes:

Jiménez Flores Jhoan.

Ramírez Saldaña Paris.

Sánchez Valencia Leonardo

Profesor: José Luis López Goytia

Grupo: 3NM40

Índice

Introducción	1
Requerimientos de mantenimiento o actualización	2
Objetivo general del sistema.....	3
Objetivos específicos.	3
Contenido técnico	3
Diagrama General	4
Diagrama de cada clase	5
Diagrama completo.....	9
Modelo de datos físico.	11
Pruebas de la aplicación versus entidades.	13
Contacto con los desarrolladores.....	15

Introducción.

El proyecto consiste en un juego estilo batalla naval utilizando los conceptos aprendidos durante el curso de Fundamentos de Programación Orientada a Objetos, en el cual hicimos uso de la creación de diagramas de clases, la misma creación de clases, métodos, atributos, etc.

El presente manual hace referencia a las especificaciones técnicas relativas a su elaboración, si se desea consultar alguna funcionalidad propia del juego consulte el manual de usuario.

Requerimientos de mantenimiento o actualización.

Conocimiento en lenguaje C# y programación orientada a objetos; se usaron las librerías estándar.

Requisitos para darle mantenimiento con Visual Studio 2015:

➤ REQUISITOS DE HARDWARE.

- Procesador de 1,6 GHz o superior
- 1 GB de RAM (1,5 GB si se ejecuta en una máquina virtual)
- 10 GB de espacio disponible en el disco duro
- Unidad de disco duro de 5400 rpm
- Tarjeta de vídeo compatible con DirectX 9 con una resolución de pantalla de 1024 x 768 o superior

➤ REQUISITOS DEL SISTEMA.

- En Windows 8.1 y Windows Server 2012 R2, se necesita la actualización 2919355 (también disponible a través de Windows Update) para que Visual Studio 2015 se instale correctamente.

Objetivo general del sistema.

Crear un juego estilo batalla naval en tres dimensiones lógicas con el lenguaje de programación C# y aplicando los conceptos de la clase de Fundamentos de Programación Orientada a Objetos.

Objetivos específicos.

Mostrar una interfaz gráfica amigable con la cual usuario pueda interactuar seleccionando munición, aéreas atacadas, seleccionar el turno, poder alternar entre las diferentes capas y mostrar la simbología de las casillas y en una partida de un jugador contra la máquina, de igual forma se busca el funcionamiento correcto del juego.

Permitir el desarrollo de estrategias creativas por parte de los jugadores para vencer dificultades cada vez más complejas

Contenido técnico.

En el código se encontrarán capas con las diferentes clases que se crearon para poder hacer más limpio y organizado; para el mantenimiento y futuras modificaciones; no se creó ninguna librería extra o diferente, solo las estándar por lo que lo único que se requiere para un mantenimiento es lo mencionado con anterioridad.

Diagrama General

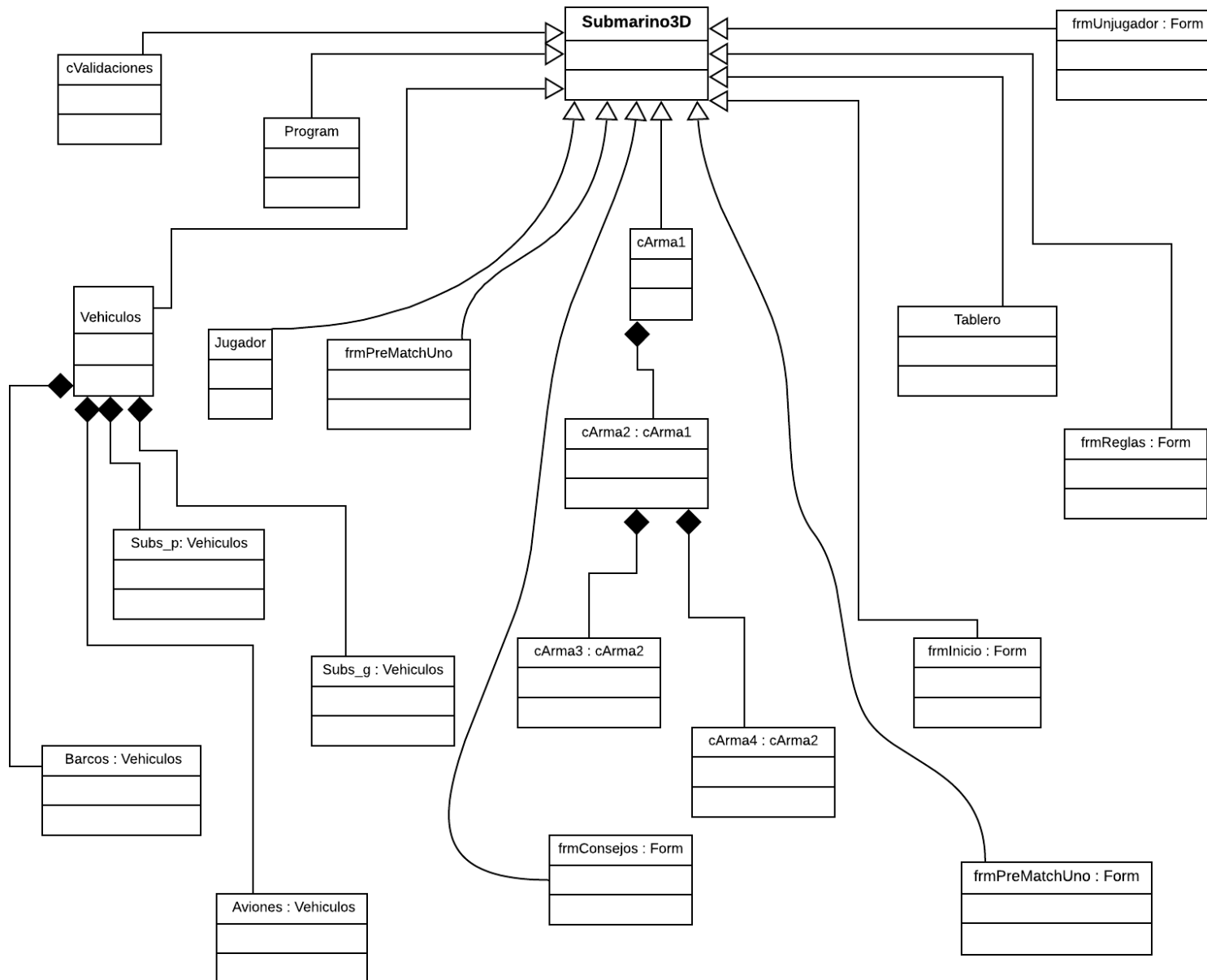
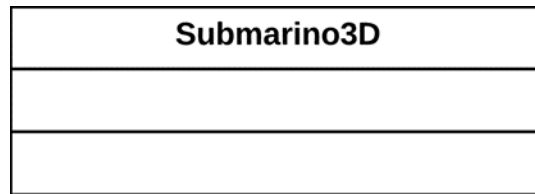
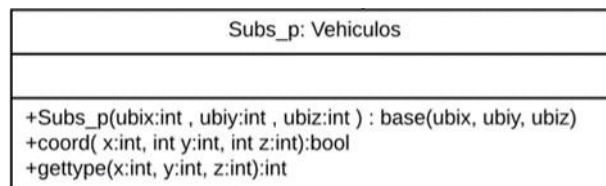
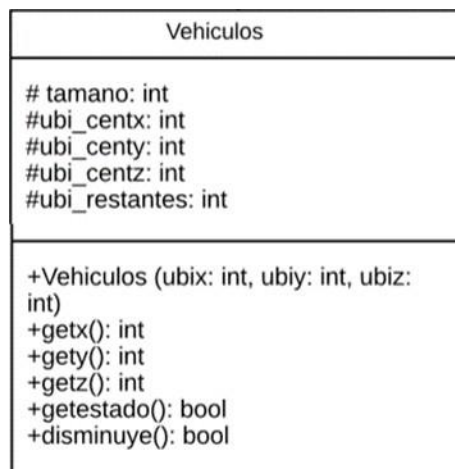
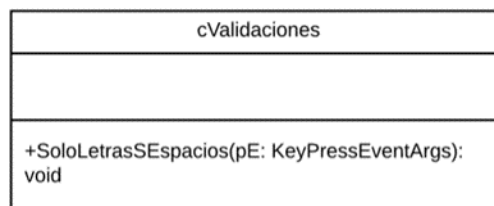
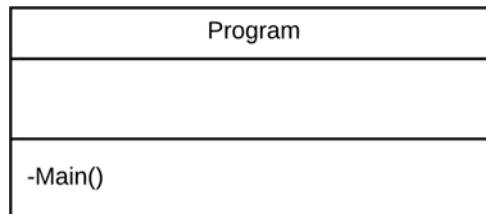


Diagrama de cada clase



(Clase principal)



Subs_g : Vehiculos
+Subs_g(ubix:int, ubiy:int, ubiz:int) : base(ubix, ubiy, ubiz) +coord(x:int,y:int,z:int):bool +gettype(x:int,y:int,z:int):int

Barcos : Vehiculos
+Barcos(ubix:int, ubiy:int, ubiz:int) : base(ubix, ubiy, ubiz) +coord(x:int, y:int, z:int):bool +gettype(x:int, y:int, z:int):int

Aviones : Vehiculos
+Aviones(ubix:int, ubiy:int, ubiz:int) : base(ubix, ubiy, ubiz) +coord(x:int, y:int, z:int):bool +gettype(x:int, y:int, z:int):int

Jugador
-Aviones, Barcos, Sub_p, Sub_g, lim:int
+Jugador(tam_x:int, tam_y:int, tam_z:int) +retira(tipo:string):void +getlimit():int +getAviones():int +getBarcos():int +getSub_p():int +getSub_g():int

frmPreMatchUno
+frmPreMatchUno() -btnInicio_Click(sender:object, e:EventArgs): void -btnAceptar_Click(sender:object, e:EventArgs): void -txtnomrbe_KeyPress(sender:object, e:EventArgs): void -frmPreMatchUno_FormClosed(sender:object, e:FormClosedEventArgs): void

cArma1
+disp(ref tablero: char[,], x:int, y:int, z:int):bool +phant(tablero: char[,], x:int, y:int, z:int):bool

cArma2 : cArma1
#municiones:int
+cArma2(mun:int) +getmun(): int +disparo(ref tablero:char[,], x:int, y:int, z:int, maxY:int):int +pantom(tablero:char[,], x:int, y:int, z:int, maxY:int):bool

cArma3 : cArma2
+cArma3(mun:int) : base(mun) +disparo(ref tablero:char[,], y:int, z:int, maxX:int):int +phantom(tablero:char[,], y:int, z:int, maxX:int):bool

cArma4 : cArma2
+ cArma4(mun:int) : base(mun) +disparo(ref tablero:char[,], x:int, y:int, z:int, maxX:int, maxY:int, maxZ:int, pi:int, pj:int, pk:int, li:int, lj:int, lk:int):int +phantom(tablero:char[,], x:int, y:int, z:int, maxX:int, maxY:int, maxZ:int):bool

frmInicio : Form
+frmInicio() -frmInicio_Load(sender:object, e:EventArgs): void -btnsalir_Click(sender:object, e:EventArgs): void -btn1player_Click(sender:object, e:EventArgs): void -btnReglas_Click(sender:object, e:EventArgs): void -frmInicio_FormClosed(sender:object, e:FormClosedEventArgs): void -btnConsejos_Click(sender:object, e:EventArgs): void

frmConsejos : Form
+frmConsejos() -frmConsejos_FormClosed(sender:object, e:FormClosedEventArgs) -rbtnMunicion_CheckedChanged(sender:object, e:EventArgs) -btnInicio_Click(sender:object, e:EventArgs) -rbtnNavios_CheckedChanged(sender:object, e:EventArgs) -rbtnOponentes_CheckedChanged(sender:object, e:EventArgs) -rbtnempezar_CheckedChanged(sender:object, e:EventArgs)

frmUnjugador : Form
<p>-pseudo:char[...] -tamtabx, tamtaby, tamtabz, turnos, dificultad:int -endgame: bool</p>
<p>+frmUnjugador(dif:int, name:string) +automata():void +showtab(y:int, aux:int):void -cboNivel_SelectedIndexChanged(sender:object, e:EventArgs):void -frmUnjugador_Load(sender:object, e:EventArgs):void -CargaTablero():void -rbtnA_CheckedChanged(sender:object, e:EventArgs):void -rbtnB_CheckedChanged(sender:object, e:EventArgs):void -btnInicio_Click(sender:object, e:EventArgs):void -btnQuedarse_Click(sender:object, e:EventArgs):void - btnlrse_Click(sender:object, e:EventArgs):void -cargatab(type:int, tablero:int, x:int, z:int):void -btnAtacar_Click(sender:object, EventArgs e):void -rbtnDisparoschange():void -rbtnDisparoA_CheckedChanged(sender:object, e:EventArgs):void -rbtnDisparoB_CheckedChanged(sender:object, e:EventArgs):void -rbtnDisparoC_CheckedChanged(sender:object, e:EventArgs):void -rbtnDisparoD_CheckedChanged(sender:object, e:EventArgs):void -btnMusica_Click(sender:object, e:EventArgs):void -btnNoConfirmo_Click(sender:object, EventArgs e):void -btnConfirmo_Click(sender:object, EventArgs e):void -retiraveh(tipo: string, objetivo: int):void -SonidoResutante(caso: int):void -ataque(objetivo: int):int</p>

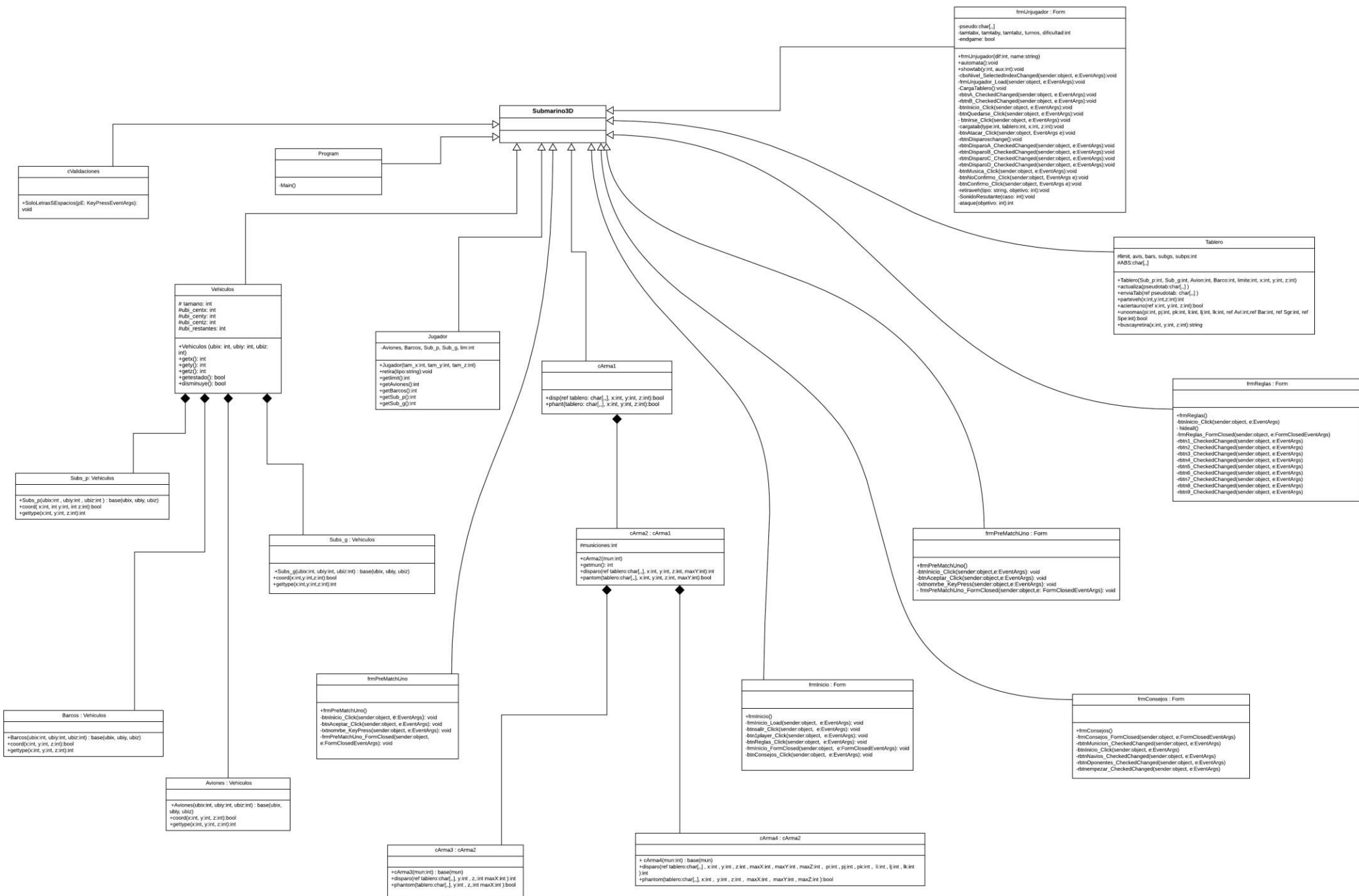
frmReglas : Form
<p>+frmReglas() -btnInicio_Click(sender:object, e:EventArgs) - hideall() -frmReglas_FormClosed(sender:object, e:FormClosedEventArgs) -rbtn1_CheckedChanged(sender:object, e:EventArgs) -rbtn2_CheckedChanged(sender:object, e:EventArgs) -rbtn3_CheckedChanged(sender:object, e:EventArgs) -rbtn4_CheckedChanged(sender:object, e:EventArgs) -rbtn5_CheckedChanged(sender:object, e:EventArgs) -rbtn6_CheckedChanged(sender:object, e:EventArgs) -rbtn7_CheckedChanged(sender:object, e:EventArgs) -rbtn8_CheckedChanged(sender:object, e:EventArgs) -rbtn9_CheckedChanged(sender:object, e:EventArgs)</p>

frmPreMatchUno : Form
<p>+frmPreMatchUno() -btnInicio_Click(sender:object,e:EventArgs): void -btnAceptar_Click(sender:object,e:EventArgs): void -txtnomrbe_KeyPress(sender:object,e:EventArgs): void - frmPreMatchUno_FormClosed(sender:object,e: FormClosedEventArgs): void</p>

Tablero
#limit, avis, bars, subgs, subps:int #ABS:char[,,]
+Tablero(Sub_p:int, Sub_g:int, Avion:int, Barco:int, limite:int, x:int, y:int, z:int) +actualiza(pseudotab:char[,,]) +enviaTab(ref pseudotab: char[,,]) +parteveh(x:int,y:int,z:int):int +aciertauno(ref x:int, y:int, z:int):bool +unoomas(pi:int, pj:int, pk:int, li:int, lj:int, lk:int, ref Avi:int,ref Bar:int, ref Sgr:int, ref Spe:int):bool +buscayretira(x:int, y:int, z:int):string

Diagrama completo

También se puede ver en una manera más clara en el siguiente archivo de [PDF](#) (sino funciona está anexo, se llama “Diagrama Completo”). En algunas partes del diagrama, en las clases que comienzan con “frm” se puede observar “:Form” que es una herencia a una clase de la librería estándar (no requiere instalar nada): “System.Windows.forms”.



Modelo de datos físico.

Se probó el rendimiento y si se encontraban errores en el desarrollo del juego conocidos como “bugs” tales como equipos con procesadores de Intel o AMD con un sistema operativo que es Windows 10 o equivalente. Las características y el entorno donde fue probado y ejecutado correctamente son:

-Sistema operativo de Windows.

-Entre algunos procesadores como requisitos mínimos son:

- Core duo con 1 gb de ram ddr2 a 1333 MHz en un Windows 10 Itsb miniOS



- Con 8gb de ram ddr4, Windows 10



- Con 8gb de ram, windos 10.



Procesador	AMD Ryzen 3 2200G with Radeon Vega Graphics 3.50 GHz
RAM instalada	4.00 GB (3.43 GB usable)
Tipo de sistema	Sistema operativo de 64 bits, procesador basado en x64
Edición	Windows 10 Pro
Versión	1903
Versión del sistema operativo	18362.900

-

Por lo tanto, tener en cuenta al utilizarlo de que se cumplen los requisitos anteriores al menos.

Pruebas de la aplicación versus entidades.

Se hicieron pruebas con distintos usuarios, los resultados son los siguientes:

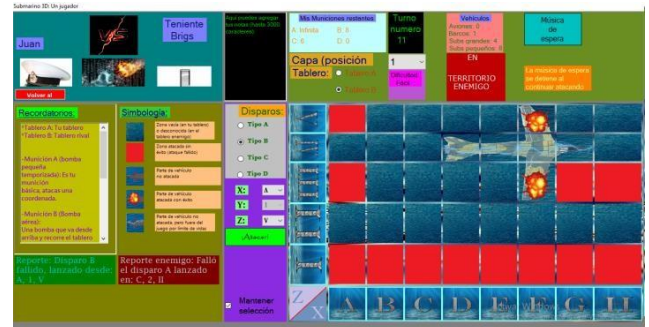
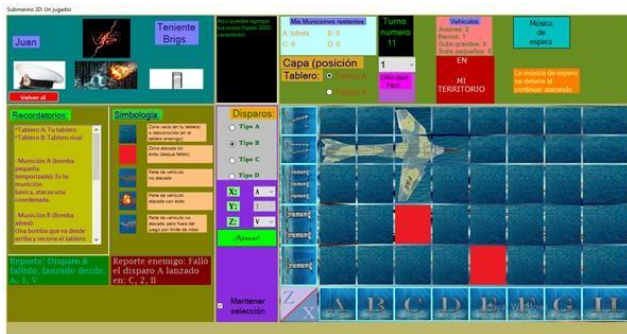
- ✓ Marco Vázquez (20 años) en la prueba no se encontró ningún bug.



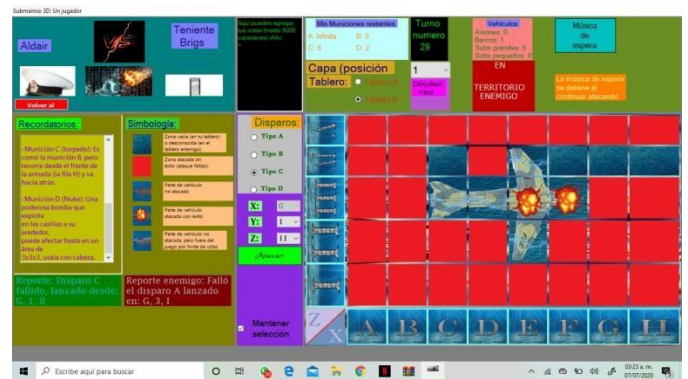
- ✓ Leonardo Sánchez (20 años) en la prueba no se encontró ningún bug.



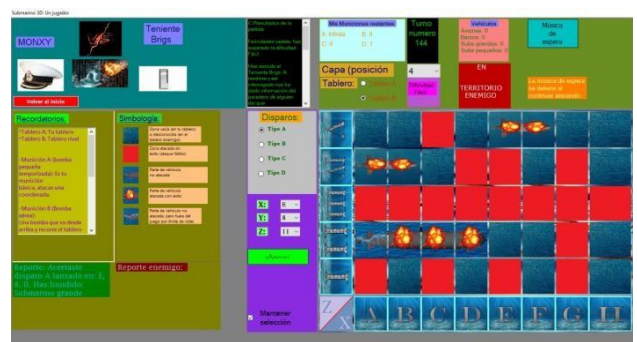
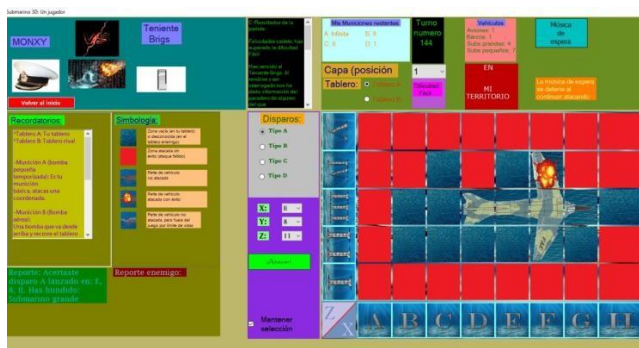
- ✓ Juan Díaz (23 años) no terminó la prueba al no entender el funcionamiento por saltarse las reglas.



- ✓ Aldair Balderas (20 años) no término la prueba al no entender el funcionamiento por saltarse las reglas.



- ✓ Montserrat Luna (19 años) en la prueba no se encontró ningún bug.



Contacto con los desarrolladores.

Nombres.	Teléfono	correo
Jiménez Flores Jhoan.	55 2711 2427	jimenezfloresjhoan@gmail.com
Ramírez Saldaña Paris.	55 1918 8765	paris.140815@gmail.com
Sánchez Valencia Leonardo	56 2209 5732	lion180100@gmail.com