

---

COSTA RICAN INSTITUTE OF TECHNOLOGY  
COMPUTER ENGINEERING

---



BATTLESHIP

SECOND PROJECT INTRODUCTION TO PROGRAMMING

PROFESOR ING. MILTON VILLEGAS LEMUS

**BY: OSCAR ARTURO ACUÑA DURÁN - 2022049304**

**SARAY MARIANA ROJAS ROJAS - 2020076936**

COMPUTER ENGINEERING

JUNE 16, 2022

# 1. Introducción

Para el Desarrollo de este proyecto de software y hardware se baso en el juego de Battleship o Batalla Naval, el cual consiste en colocar cierta cantidad de naves o barcos en un tablero e intentar atinarles a las naves que coloco el oponente para así hundirlo.

En este caso, se plantea un juego del usuario versus la computadora, donde el usuario determina la cantidad de barcos que desea colocar en el tablero de 10x10 casillas tanto para el como para la computadora. Luego tiene el turno de disparar a cualquier casilla del oponente, y la computadora también le disparara, todo esto mediante el lenguaje de programación Python en su versión 3.10.4.

En primer lugar, se plantea un fundamento matemático para desarrollar el proyecto, basándose en las matrices de Python estudiadas en el curso; donde una matriz cuadrada de 10 elementos describe perfectamente el comportamiento de los tableros de juego que se busca. Usar la lógica de una matriz permite que cualquier espacio de la matriz pueda ser referenciado o editados mediante su ubicación, haciendo que la modificación de la matriz sea muy sencilla, aspecto buscado para garantizar la jugabilidad. Además, guardar los datos contenidos en una matriz para editarlos después, resulta bastante sencillo.

La implementación de la interfaz gráfica basada en la lógica descrita del juego se hace al hacer un desarrollo hibrido entre dos bibliotecas de Python: Pygame 2.1.2 y tkinter 8.6.12. La pantalla inicial y las entradas de nombre y barcos que suministra el usuario se desarrollan en tkinter 8.6.12 ya que esta permite el mejor desarrollo de interfaces graficas de usuario, mientras que la programación del juego como tal, es decir, los tableros, colocar los barcos y el disparar y ser disparado se desarrolla en la biblioteca Pygame 2.1.2 que como su nombre lo indica facilitan el desarrollo de juegos en Python, ya que poseen un mejor manejo de eventos.

Luego se hace la implementación del hardware se utiliza un código de Arduino (v.1.8.9) que mantiene la comunicación con el código de Python (v 3.10.4. ), que se utiliza para enviar los comandos que el controlador debe ejecutar. Sin embargo, al código de Arduino se le implementaron las instrucciones específicas a realizar, al ingresar entradas en la interfaz virtual programada en Python, para el control de las posiciones de las naves me-

dian­te 4 botones, cada uno en una dirección y otros dos encargados de rotar y de colocar la nave en la posición establecida. Además, al Arduino se le conectan cuatro diodos de tipo LED, que se encienden según las instrucciones del programa como lo son: disparo acertado, disparo fallido, fin del juego y entrada al salón de la fama. Para realizar la implementación en hardware, se necesito hacer previamente una simulación en la plataforma de modelado de circuitos Tinkercad.

## **2. Conclusiones**

### **Relacionadas al Arduino**

1. El software de programación de Arduino es muy útil a la hora de realizar prototipos de proyectos que involucren el control de componentes electrónicos externos como por ejemplo LEDs y botones.
2. El software de Arduino cuenta con gran variedad de ejemplos que pueden resultar muy útiles para entender el funcionamiento de un equipo o bien para realizar pruebas rápidas.
3. Arduino es una plataforma de desarrollo sumamente versátil, pues cuenta con controladores para gran variedad de equipos, que además pueden ser aumentados mediante la descarga de bibliotecas específicas.
4. Arduino se basa en el lenguaje de programación C++, sin embargo hay herramientas como firmata que permiten tener una integración entre el arduino y Python.
5. A la hora de integrar el arduino con Python es de mucha ayuda tener un modulo que revisa el correcto funcionamiento de los componentes del arduino.
6. El gran reconocimiento que tiene arduino implica que existen gran cantidad de apoyos y referencia en muchas de páginas web y obras literarias.
7. Arduino resulta ideal para el desarrollo de este proyecto pues, gracias a su versatilidad, este puede comunicarse con programas de Python, que facilita el desarrollo

de interfaces gráficas más significativas.

8. Mediante python se pueden realizar programas que se comuniquen directamente con Arduino lo cual involucra un aumento importante de los límites de capacidad para un programa escrito en python.
9. Con la biblioteca pyfirmata 2.2 se puede integrar el arduino con interfaces gráficas permitiendo realizar programación tangible.
10. Las variables globales son un recurso para lograr comunicación eficiente, y si se quiere bilateral, entre varias funciones.
11. Mediante variables globales es posible controlar el flujo de los threads de un programa.
12. Utilizar archivos de texto plano para guardar información permite que un programa tenga “memoria” de lo que sucedió anteriormente.
13. El uso incorrecto de los archivos es Python puede generar conflictos para editar, abrir o cerrar un documento.
14. tkinter 8.6.12 solo puede utilizar imágenes en formato .gif o .png, sin embargo, existen bibliotecas como PIL que permiten usar muchas otras extensiones.
15. La biblioteca time permite al desarrollador ejecutar comandos utilizando medidas de tiempo como referencia.
16. Time, puede ser utilizada con threading para ejecutar varios módulos a distintas velocidades, simultáneamente.
17. Cuando se abren archivos es importante volver a cerrarlos después de realizar los procesos de escritura o lectura para no tener problemas si se quiere editar desde otra función o si se quiere visualizar el archivo sin Python.

**Relacionadas con Git**

18. La herramienta Git permite tener un ambiente de trabajo mucho mas organizado cuando se esta en un equipo.
19. Los commits de Git permiten tener respaldos más seguros de las versiones anteriores de un proyecto.
20. Los commits de Git permiten tener un control más específico de los cambios que se realizaron día a día, y también del aporte individual de cada miembro del grupo.
21. GitHub permite un uso más sencillo, ordenado y gratuito de las funciones de Git.

#### **Relacionadas a hardware (componentes y electrónica)**

22. Las resistencias son componentes que juegan un papel importante en la protección de los componentes, pues estas permiten tener una regulación de la cantidad de corriente que pasa por un componente.
23. La Protoboard es el componente ideal para probar circuitos de manera preliminar. Muy útil para comprender y asegurar el funcionamiento de los distintos componentes que se van a utilizar en la elaboración de un circuito.
24. Cuando se trabaja con una protoboard, es muy importante comprender el funcionamiento interno de estas, puede pasar que se cometen errores en circuitería por imaginarse la protoboard distinta a como es en la realidad.
25. Cuando se tiene algún problema con el funcionamiento de algún circuito, es sumamente útil hacer uso de un multímetro y con este probar la funcionalidad de cada componente.

#### **Relacionadas a Organización Grupal e investigación previa.**

26. Al trabajar en grupo es fundamental definir reglas y acuerdos, en prevención a cualquier discusión o inconveniente.
27. La comunicación con el grupo de trabajo puede facilitar e incluso agilizar la dinámica de trabajo grupal.

28. “Telegram” es una gran herramienta de comunicación a distancia que permite la organización del proyecto sin necesidad de la cercanía física.
29. Contar con alguien a quien remitir dudas puede resultar de gran apoyo, teniendo en cuenta la gran
30. Hacer una lluvia de ideas y repartir correctamente el trabajo es fundamental si se quiere tener un resultado fructífero.

### **3. Recomendaciones**

1. Arduino es muy recomendable al realizar prototipos de proyectos, sin embargo, es importante saber utilizar el software. De lo contrario se debe invertir tiempo para aprender a utilizarlo.
2. Cuando se tenga duda acerca de alguna función específica los ejemplos de Arduino resultan especialmente útiles y permiten formular una idea del código que es necesario. De lo contrario se deberá utilizar tiempo extra para generar algunas de los módulos que ya existentes.
3. Cuando se vaya a utilizar un componente para ser programado mediante Arduino se recomienda valorar si este ya tiene una biblioteca incluida en el programa o es necesario realizar la descarga de esta. De lo contrario no se podrá utilizar el componente.
4. Para generar programas en Arduino estos deben estar escritos en una variante C++, es entonces que, si no se comprende el funcionamiento de este lenguaje, vale la pena explorar herramientas de integración con otros lenguajes como pyfirmata.
5. Para comprender Arduino o buscar información específica es recomendable buscar información en la web o en literatura. YouTube es especialmente útil dada su gran variedad de contenido y ejemplos donde se puede apreciar el funcionamiento claramente. Sin embargo, dada la gran cantidad de información se debe invertir tiempo

en encontrar lo que se busca específicamente.

6. Para la elaboración de interfaces gráficas simples se recomienda tkinter 8.6.12 es la herramienta estándar de Python. Resulta bastante útil y relativamente fácil de usar, sin embargo puede tener limitaciones importantes para proyectos de mayor calibre.
7. Para controlar threads las variables globales son especialmente útiles pues afectan a todos los módulos a la vez. Sin embargo el uso de estas puede fácilmente inducir a errores debido a la gran trascendencia que tienen.
8. Se recomienda utilizar archivos de texto plano para guardar datos que prevalecen después de la ejecución del programa. Estos son muy fáciles de usar, sin embargo presentan grandes limitaciones cuando se trabaja con grandes cantidades de información.
9. Es importante mantener el control sobre los archivos abiertos a través de Python, aunque esto sea tedioso, de no cerrarlos o abrirlos adecuadamente el archivo puede perderse, o deshabilitarse o no poder modificarse o abrirse.
10. Si se desea trabajar con imágenes en tkinter 8.6.12 se recomienda que sean de extensión .png, pues de lo contrario ha de utilizarse un software que las convierta o bien instalar alguna biblioteca para poder utilizarlas.
11. Se recomienda utilizar threads con la biblioteca de Time, para así solo detener ciertas partes del código mediante el comando `time.sleep()`, sin embargo muchas veces es complicado cerrar estos threads.
12. Se recomienda utilizar matrices para guardar un conjunto de datos ya que son fáciles de manipular y en cada elemento se pueden guardar diferentes tipos de datos, esto ya que permiten una fácil manipulación de la información, sin embargo es complicado hacer una implementación de código que guarda y luego lee los valores de estas matrices.

13. A la hora de crear interfaces, si se necesitan editar imágenes o crear fondos para darle una mejor estética al programa, se recomienda utilizar Photoshop ya que es un programa muy versátil y con muchas herramientas, pero hay que considerar la compra de licencias que hace que sea menos accesible.
14. Git de Windows en conjunto con GitHub son herramientas muy útiles para el intercambio y modificación de código entre varias personas ya que se pueden subir los cambios más fácilmente, sin embargo, puede generar problemas si más de una persona trabaja en el mismo archivo.
15. Es recomendable hacer un estudio de la herramienta git antes de comenzar a usarla, ya que de esta manera todo funcionara de maneja mucho mas orgánica, sin embargo esto puede consumir mucho tiempo.
16. Se recomienda un estudio previo de los componentes antes de comprarlos, de lo contrario podría terminarse con componentes que realmente no van a cumplir con la función deseada, sin embargo es pertinente considerar que esto puede consumir mucho tiempo.
17. Al adquirir alguna pieza de hardware programable es necesario asegurarse de contar con los medios necesarios para comunicarlo con el computador, esto permite que sea mucho mas efectivo el tiempo de trabajo, sin embargo, tener un inventario de todos los cables que se tienen puede ser tedioso.
18. Para todo momento en el que se quiera probar el funcionamiento de un circuito se recomienda utilizar una Protoboard para probar el prototipo del circuito, ya que esta es sencilla de usar, de lo contrario se arriesga el funcionamiento correcto del circuito.
19. Se recomienda mantener una bitácora limpia y ordenada, con las actualizaciones debidas al día, aunque estas consuman mucho tiempo, facilitan en gran medida la redacción de la documentación, de no llevarse al día puede ser problemático recordar lo sucedido para la documentación y por lo tanto puede quedar incompleta.

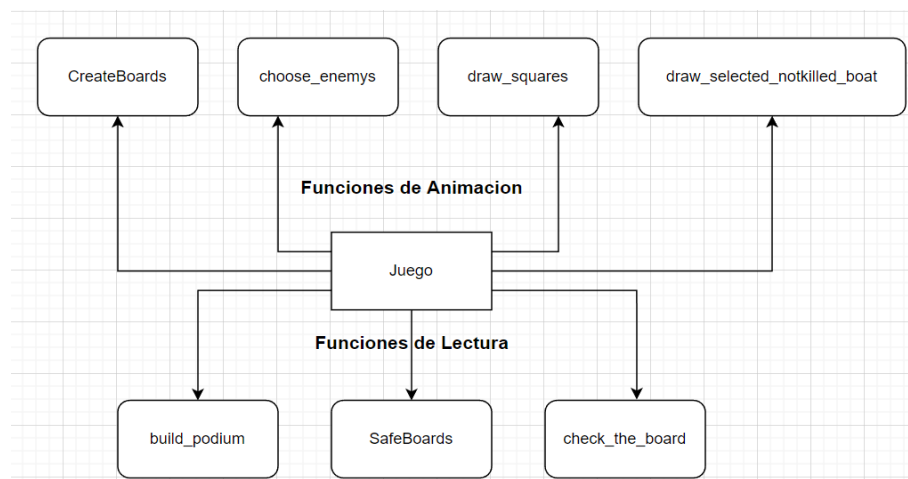


20. Se recomienda definir las reglas de grupo al principio del proyecto, aunque puede ser un poco incomodo esto evita disputas o malentendidos.
21. Se recomienda que tener a alguien con conocimientos de electrónica para así poder orientar al grupo, ya sea un profesor, un técnico o un compañero. Aunque esto consuma mucho tiempo del ayudante vale la pena ya que la realización de este proyecto sin conocimiento previo puede resultar problemática.
22. Cuando se trabaja en un equipo, vale la pena llevar cuentas claras de la inversión que ha hecho cada individuo, aunque esto sea incomodo, puede evitar muchos problemas a futuro.
23. Antes de comenzar a escribir código es importante preguntar al equipo si ya alguno esta trabajando en esa parte, aunque esto sea cansado, puede ayudar a ahorrar mucho tiempo con las colisiones de código.
24. Cuando se esta trabajando en equipo, es pertinente escribir la programación pensando en el ordenador que tienen todos, aunque esto pueda ser limitante para el que tiene un ordenador mas potente, es pertinente de hacer si se quiere tener una dinámica de trabajo orgánica.
25. Las simulaciones virtuales de las implementaciones tangibles de la programación pueden ser continuamente útiles, aunque estas no son tan entretenidas de usar, trascienden la barrera del espacio cuando alguien se esta trabajando a distancia.
26. Aunque se tenga estrés en el ambiente de trabajo, es pertinente siempre ser comprensivo ante las situaciones que le pase al otro, inclusive si afectan la calidad del resultado final.
27. Antes de defender una programación es pertinente revisar que todo este funcionando a la perfección, aunque esto consuma tiempo puede ahorrar muchos problemas.
28. En internet se encuentran muchas veces códigos muy parecidos a los que se quieren implementar, usar estos pueden opacar la creatividad del código propio, pero son sumamente útiles de usar continuamente.

29. Los distintos algoritmos de ordenamiento son útiles en distintas circunstancias, es pertinente comprender el funcionamiento de cada uno, pese a que esto consuma tiempo.
30. Leer la literatura de una herramienta antes de utilizarla es continuamente útil, aunque esto consuma mucho tiempo, esto puede ayudar a agilizar mucho el trabajo cuando se esta escribiendo código.

## 4. Análisis de Resultados

### 4.1. Diagrama de Módulos



### 4.2. Plan de Pruebas

```
window = tk.Tk()
```

```
window.title( Battleship )
```

```
window.minsize(700, 700)
```

```
window.resize(False, False)
```

```
Inicio = tk.Canvas(window, width = 700, height = 700)
```

```
Inicio.place(x = 0, y = 0)
```

Se crea una ventana de tkinter 8.6.12 que tiene la información about, ranking y le pregunta al usuario si quiere iniciar nueva partida o si quiere cargar una partida ya jugada.

```

def RunGame(TheUser, CBoats, Matrix, Bool):
    global ImPlaying
    if Bool == False:
        ImPlaying = True
    global User
    User = TheUser
    global CantBoats
    CantBoats = CBoats
    global WIN
    WIN = pygame.display.set mode((WIDTH, HEIGHT))
    main()

```

Se crea una función que esta encargada de abrir el Pygame 2.1.2 desde tkinter 8.6.12, para de esta manera poder comenzar a jugar.

```

def draw window():
    global board
    board = Board()
    pygame.display.set caption("Battleship")
    board.draw(WIN)
    board.draw enemy(WIN)
    board.make enemys(WIN)
    board.draw on enemyboard(WIN)
    pygame.display.update()

```

Se crea una función que esta encargada de dibujar todo en la ventana de Pygame 2.1.2. Esta es llamada en el loop que ejecuta al Pygame 2.1.2.

## 5. Bitácora

### **22 de mayo:**

Decidimos ser pareja de trabajo en el segundo proyecto de taller de programación.

### **25 de mayo:**

Nos sentamos a leer el proyecto en grupo, establecimos las reglas del grupo y un canal de comunicación donde pudiéramos atender cualquier duda o aporte al proyecto.

### **27 de mayo:**

Investigamos el funcionamiento del juego en la vida real y vimos videos de YouTube sobre programas que han desarrollado otras personas, para tener una visión más clara de hacia dónde debe dirigirse nuestro proyecto.

### **28 de mayo:**

Iniciamos con el planteamiento de la cuadrícula de juego como una matriz cuadrada de datos, siguiendo la lógica de que las posiciones donde hay un barco se convierten en 1, y donde no hay un barco son un 0. También se logra que a partir de la biblioteca Random, que los barcos se coloquen en posiciones al azar en su respectiva matriz.

Para este punto no hay interfaz gráfica, las matrices se imprimen en la terminal de Python.

### **31 de mayo:**

Se hace una primera aproximación a la interfaz gráfica usando Pygame 2.1.2. Lo sentimos un poco confuso, porque es algo nuevo, pero al ver un tutorial largo sobre aspectos importantes, se logra entender sobre los básicos de la biblioteca Pygame 2.1.2 de Python. A Mariana se le dificultó poder instalarlo en la terminal, pero se pudo finalmente.

Se crea el archivo Funciones, en este el día de hoy se crearon las funciones que se encargan de dibujar los tableros de juego, tanto del usuario así como la de la computadora, inicializándolos como matrices 10x10 llenas de ceros (0).

Se logran colocar los barcos en los tableros, con las teclas de movimiento a,s,d,w, y la tecla que lo coloca es la q. También en el tablero de la computadora, aunque no se vean, ya por 'debajo' están los barcos enemigos colocados.

### **2 de Junio:**

Recibimos un taller de Git, para poder desarrollar, guardar y seguir las versiones del proyecto de una manera más sincronizada y ordenada.

Se implementa junto con las teclas de movimiento y de ubicación a la tecla e, encargada de rotar la posición del arco de horizontal a vertical y viceversa a la hora que uno quiera ubicarlo en la gráfica. Se logran hacer los tres tamaños de barcos, de 4 casillas, de 3 casillas y de dos casillas, y se almacenan en una lista, los barcos se hacen a partir de imágenes que tienen el tamaño de las celdas que necesitan abarcar. Cuando se colocan, se dibujan unos círculos sobre donde estaba la imagen.

También, los barcos se acomodan según el número de casillas, empezando por el mas grande. La cantidad de barcos es definida en una lista, por e momento no la digita el usuario. Se debe de hacer dentro de la programación. Además el día de hoy se modificó un poco más con los colores del juego para hacerlo visualmente mas ameno.

#### **4 de Junio:**

Se plantea, con ayuda de Marcelo, que, pensando en la implementación del Arduino en secciones posteriores del proyecto, se debe de hacer un cursor el cual indique que posición es la que se quiere disparar. Ya que, con las teclas, como se pensó inicialmente, no seria buena idea pensando en la implementación en hardware.

Se hace este cursor, que es una pequeña imagen de un cuadrado verde que se puede mover por el tablero con ayuda de las teclas a,w,s,d y al igual que los barcos, se coloca con la tecla q.

#### **7 de junio:**

Con lo que se trabajo del cursor el día 4, se siguió para poder que cuando se coloque ahí el cursor, revise si ahí había un barco (un 1 en la matriz): si había un barco escriba un 3 en la matriz, y en la ventana grafique dibuje un círculo rojo, si, por el contrario, se coloca donde no había barcos, se dibuja un circulo azul y se escriba un 2. Así, con esta función se permite que haya disparos para del usuario a la computadora.

Luego, se programa el disparo random que tiene que hacer la computadora, donde selecciona una coordenada al azar y dispara ahí, con las mismas condiciones explicadas anteriormente, solo que para la matriz del usuario.

### **9 de junio:**

Para este punto tenemos bastante de la jugabilidad básica desarrollada. Entonces aquí se procede a hacer la ventana inicial. Se nos ocurre hacerla en tkinter 8.6.12 porque en Pygame 2.1.2 no existen los entrybox, que son necesarios para ingresar el usuario. Hay que destacar que, si se puede hacer una especie de entrybox con Pygame 2.1.2, donde el detecte la tecla que se presiona y la escriba en pantalla, sin embargo, este camino es mas engorroso que usar un hibrido entre Pygame 2.1.2 y tkinter 8.6.12.

Una vez finalizado el usuario planteamos que, para guardar el juego, habría que guardar el nombre de usuario, y la matriz de jugador y la matriz de la computadora en un archivo de texto y ya podría ser editado esta partida usando el nombre. Se hace otro entrybox para ingresar a una partida guardada.

### **10 de junio:**

Este día el profesor Jason impartió un taller de ayuda para el proyecto al cual asistimos y con las explicaciones del profesor pudimos ver que íbamos por buen camino en el proyecto. Este día fue posible también hacer el cronometro del juego, que es clave para poder hacer en secciones posteriores el salón de la fama.

También, algunos tutores presentes nos ayudaron con bugs que había, como el parpadeo de imágenes que se daba cuando se corría la programación.

También nos dan como recomendación que vayamos comentando el código desde ya para que no sea muy complicado cuando la entrega se acerque más, así como ir iniciando con secciones de la documentación.

### **13 de junio:**

Este día pudimos hacer las restricciones de movimiento de cada tipo de barco, para que no se salga de las matrices. También la restricción de rotación, cuando dependiendo de la ubicación del barco que no se pueda rotar.

Estas restricciones ayudan a que el juego no tenga errores que se pueden evitar. Sin embargo, esta sección consumió bastante tiempo ya que cada tipo de barco tiene una restricción diferente dependiendo de si está en orientación vertical u horizontal.

### **14 de junio:**

Este día Arturo empezó con la implementación del Hardware. Usando el Arduino y los videos que nos proporcionó Marcelo, así como la ayuda de él, fue posible conectar los botones en la Protoboard con la programación permitiendo que los barcos fueran colocados con estos y ya no con los botones del teclado.

Este día Mariana Inicio con la documentación, en secciones como introducción, conclusiones y recomendaciones.

### **15 de junio:**

Se termino la ventana de inicio, con la ventana de About y el desarrollo grafico que queríamos. Se agrego una bandera para impedir que el usuario hiciera de alguna forma trampa a la hora de dispararle a los barcos del enemigo. Y se hizo la validación y restricción para que los barcos no se superpusieran unos sobre otros. Además, finalmente se hizo la implementación los entrys para cada tipo de barco, donde se hace un entrybox para cada uno de los tipos donde el usuario puede ingresar la cantidad de que desea tener en el juego. También se siguió avanzando con la documentación del proyecto tomando en cuenta el feedback que nos dio el profe a cada uno de forma independiente para lograr el mejor desarrollo de esta sección.

### **16 de junio:**

Este día se dedicó a la implementación de los diodos LEDs que se encienden cuando el usuario acierta un tiro, falla un tiro, gana, y entra al salón de la fama, con la Protoboard y el Arduino esto es posible. En esta sección Marcelo nos ayudó muchísimo.

Por último, este día se finalizó con la programación del salón de la fama usando InsertSort y QuickSort como algoritmos de ordenamiento de datos. Además, se terminó la documentación del proyecto.

## **6. Organización Grupal**

Como grupo, y antes de empezar el trabajo, se establecen algunas reglas para velar por el buen desarrollo de todo el proyecto con la prevención de conflictos entre los miem-

bro de grupo. También, se establece que el grupo no trabajara mediante roles establecidos previamente, ya que a medida que se avance en el trabajo así como las disponibilidad de tiempo de cada integrante, van a ir dictando que aspectos debe cubrir cada uno.

Tomamos en primer lugar como meta final, tener la mejor nota posible, buscando siempre el 100 en el trabajo que desarrollamos, incluyendo la finalización al 100 del trabajo así como los puntos extra correspondientes a 30 puntos extra. Esta meta nos ayuda a trazar un norte a la hora de trabajar en el proyecto durante tantos días.

Luego se establecieron las siguientes reglas del grupo para que el avance del trabajo fuera ameno y de provecho para ambos.

**Tener comunicación clara y constante.** Esta primera regla recae en que permite conocer los avances y contratiempos que enfrenta constantemente el otro compañero. Para evitar distracciones y poder dar soluciones más rápidas, establece como canal de comunicación para los colaboradores Telegram.

**Priorización de Tareas.** Se establece que es de primera importancia trabajar en los aspectos que mas consumen tiempo de la programación de primero, para evitar contratiempos a la hora de realizar la entrega, dejando en segundo plano las partes mas triviales del trabajo.

**Solucionar los problemas en el momento** . Si se llegara a tener un conflicto entre los miembros del grupo, recordar que ambos somos amigos y compañeros, y que podemos resolver cualquiera de los problemas hablando, pero estos deben ser resueltos en el momento.

**Empatía** Se debe de recordar que ambas partes del grupo, somos estudiantes y que tenemos otras responsabilidades además de este trabajo, por esto, se solicita que la empatía sea un valor que sobresalga siempre en este grupo. Entender cuando el otro no puede trabajar por motivos de tiempo, de enfermedad o cualquier situación



fuera del control es de suma importancia.

**Tener un lenguaje positivo.** Aunque el trabajo vaya mal, nada se soluciona teniendo un vocabulario soez o teniendo una mala actitud respecto al trabajo. En realidad, esto solo hace que el trabajo no avance y el ambiente sea tenso.

**Escuchar las ideas del otro** Recordar la frase ‘dos cabezas piensan mejor que una’, esto porque la otra persona puede interpretar el problema diferente o ayudar con una solución que tal vez uno no tenga en mente aún.

**Pedir ayuda.** Hay que recordar que somos compañeros de grupo, no una competencia en ver quien entiende o sabe mas sobre el tema. Pedir ayuda esta bien, ya sea al otro compañero o buscar ayuda de tutores o Marcelo.

**La única pregunta tonta es la que no se hace.** Es mejor saber que todos los conceptos, avances, funciones y demás van quedando claro para ambos a medida que se avanza en el proyecto.

**Liderazgo en conjunto.** Con respecto al liderazgo, se determina que la mejor estrategia es la de un liderazgo compartido ya que como grupo coincidimos que a ninguno nos gusta realmente ese rol.

**Productividad** Finalmente, los resultados se deben de ver satisfactorios con respecto a su tarea, y deben ser entregados en el momento oportuno. Esto ayuda a que haya una motivación a seguir trabajando juntos en el proyecto, en el curso e inclusive en otros cursos a futuro.

## 7. Actividades y Fechas de Entrega.

Se divide el proyecto en 4 Stages o etapas de avance principales. Estas están divididas según los requerimientos del proyecto. Algunas stages son mas complicadas que otras.

**Stage One** Esta se encarque de la ventana inicial y se desarrolla mayoritariamente en Tkinter. Abarca, las siguientes tareas. Abarca la sección mas inicial del trabajo, y se propone como fecha de entrega el 4 de junio: se considera que se entrega a tiempo en su mayoría, porque se decide priorizar otras secciones que abarcan mas tiempo.

1. Hacer las imágenes de background, aboud, salón de la fama.
2. Hacer los tableros de 10x10 casillas. Un tablero para el usuario y otro tablero para la compu.
3. Hacer la pantalla inicial con los botoes y acceso a subpantallas y los entrybox de usuario y barcos.
4. Crear las imágenes para las naves tipo 1, 2 y 3, y el cursor.

**Stage Two** Esta stage esta encargada de todos los ships settings, esta tiene una fecha de entrega del 11 de junio, sin embargo no fue posible alcanzarla porque la finalización de algunas condiciones de restriccion, no permitió que se pudiera avanzar rápidamente. Esta stage consistió en las actividades que se proponen a continuación:

1. La pantalla de juego se abre después de llenar la pantalla de inicio.
2. Colocar las naves aleatorias en el tablero pero no hay que mostrarlas.
3. Las naves deben de ocupar cierta cantidad de casillas.
4. Se debe poder colocar las naves del jugador: dependiendo de la cantidad de barcos que se escoge.
5. Se debe de poder rotar horizontal o verticalmente la nave antes de colocarla.

**Stage Three** Corresponde a la jugabilidad o Playing del juego como tal. Se proponen las siguientes tareas, mismas que tuvieron como fecha de entrega el 13 de junio y esta fue cumplida con éxito.

1. Al tocar las casillas del enemigo (tablero compu): se marque rojo y haga un sonido de explosión si había una nave en esa casilla. Si se dispara y no hay nave, que se marque en azul y un sonido que simbolice que no le dio.
2. Debe de ser un turno el usuario y luego un turno la computadora, siempre debe ser así. No puede permitirse hacer trampa con los turnos.

**Stage Four** Esta etapa se denomino Fancy Playing, ya que a jugabilidad mas robusta estaba construida, en esta se añaden aspectos que la hacen mas user friendly así como que permiten una mejor interpretación del juego, con el desarrollo de las siguientes tareas con deadline el 14 de junio pero realizado el 13.

1. Agregar un timer a la partida.
2. Cuando gana alguno, se debe de mostrar las estadísticas del programa.

**Stage Five** Esta stage esta dedicada al manejo de Data en el programa, tiene como deadline el 15 de junio fecha que fue satisfecha correctamente. Actividades:

1. Actualizar el salón de la fama, y hacerlo con InsertSort y Quicksort.
2. Guardar la partida: guarda las posiciones de usuario y computador para abrirlas después.

**Stage Six** Se encarga de todo el hardware de la programación. Tiene como deadline el 16 de junio a media noche. También en esta se incluye la documentación terminada del proyecto.

## 8. Evaluación

Estudiante: Mariana Rojas Rojas			
Aspectos:	Siempre	A veces	Se le difitulta
Participa en la toma de acuerdos del grupo.	X		
Cumple con las tareas asignadas	X		
Participa en todas las actividades del grupo	X		
Apoya a los compañeros que lo necesitan	X		
Colaora en la presentacion final del proyecto	X		

Estudiante: Oscar Arturo Acuña			
Aspectos:	Siempre	A veces	Se le difitulta
Participa en la toma de acuerdos del grupo.	X		
Cumple con las tareas asignadas	X		
Participa en todas las actividades del grupo	X		
Apoya a los compañeros que lo necesitan	X		
Colaora en la presentacion final del proyecto	X		

## Referencias

- Back-End, T. R. (s.f). *Control arduino with python and pyfirmata (from raspberry pi)*. <https://roboticsbackend.com/control-arduino-with-python-and-pyfirmata-from-raspberry-pi/>.
- Tim, T. W. (2020, Septiembre). *Python checkers*. <https://github.com/techwithtim/Python-Checkers>.