



# Metodología de la Programación

*Grado en Ingeniería Informática*

Curso 2024/2025

Gutiérrez Loaiza, Lucía

García Barea, Adrián

Ruiz del Pino, Alejandro

Olivero del Pino, Antonio

*Proyecto Modularidad*

**HUNDIR LA FLOTA**

# Índice general

Índice general	1
1. Introducción . . . . .	2
2. Documentación de usuario . . . . .	3
2.1. Descripción funcional . . . . .	3
2.2. Tecnología . . . . .	4
2.3. Manual de instalación . . . . .	4
2.4. Acceso al sistema . . . . .	4
2.5. Manual de referencia . . . . .	5
2.6. Guía del operador . . . . .	6
3. Documentación de sistema . . . . .	7
3.1. Especificación del sistema . . . . .	7
3.2. Módulos . . . . .	7
3.3. Plan de prueba . . . . .	9

# 1. Introducción

El objetivo de nuestro proyecto ha sido desarrollar una adaptación del juego de mesa ***Hundir la flota***, de forma que un usuario pueda jugar al juego contra la máquina o si lo desea contra otro usuario desde el mismo equipo.

Si el usuario elige jugar contra la máquina deberá seleccionar la opción de uno de los jugadores como automática, y la opción del otro jugador sera manual, este jugador será el usuario. Si por el contrario, dos personas desean jugar desde el mismo equipo, ambos jugadores deberán seleccionar la opción manual.

Además, el usuario podrá establecer el tamaño del tablero, definir los barcos que se podrán utilizar y posteriormente colocar sobre el tablero, incluyendo el nombre del barco, el tamaño y la cantidad de unidades que hay de ese tipo de barco.

En este documento, explicaremos la documentación externa del proyecto, la cuál incluye la documentación del usuario y la documentación del sistema, asi como análisis, diagramas de flujos y/o pseudocódigos. A continuación describiremos el contenido de la documentación del usuario y la documentación interna:

- *Documentación del usuario*: la cuál va dirigida al usuario, contiene una descripción de como funciona el programa desde el punto de vista del usuario, incluyendo el manual de instalación.
- *Documentación del sistema*: en la cuál se explicarán los aspectos relacionados con la implementación del programa, incluyendo los aspectos sobre análisis, diseño, pruebas del software.

## 2. Documentación de usuario

Esta sección corresponde con la documentación del usuario en la cuál se detallarán aspectos del programa pero no relacionados con la implementación, es decir, se explicará como funciona el juego desde el punto de vista del usuario además del proceso de instalación.

A continuación, explicaremos brevemente en qué consiste el juego *Hundir la flota*. Se trata de un juego de dos jugadores, en el que ambos jugadores primero deben colocar todos los barcos de la flota en un tablero. Después, cuando ambos jugadores hayan colocado todos los barcos, por turnos deben ir diciendo coordenadas e intentar hundir todos los barcos del otro jugador. Ganará quien antes haya hundido todos los barcos del jugador contrario. En el programa que hemos implementado, es posible que uno de los jugadores sea la máquina, es decir, se colocarán los barcos de manera automática y los disparos serán automáticos, es decir, el usuario estará jugando contra la máquina.

### 2.1. Descripción funcional

- El sistema es una implementación del famoso juego de mesa *Hundir la flota*, esta implementación cuenta con dos opciones de juego, la opción manual y la automática. En la opción automática es la máquina quien se encarga de colocar los barcos al inicial la partida y de elegir la coordenada en la que se producirá el disparo. Sin embargo, en la opción manual, es el propio usuario quien se encarga de colocar los barcos y elegir la coordenada donde se producirá el disparo.
- En esta implementación del juego, se permite al usuario definir muchas características del juego, como por ejemplo el tamaño del tablero, la cantidad total de barcos y la cantidad de unidades por cada tipo de barco. Además, de definir los tipos de barco, indicando el tamaño y el nombre de cada tipo. También el usuario puede introducir su nombre.
- El sistema controla que las coordenadas en las que se desea colocar un barco sean correctas, es decir, que sobre esa coordenada no se encuentre un barco, no supere las dimensiones del tablero, y además, que contiguo a esa coordenada no se encuentre ningún barco. Ya que, no puede haber dos barcos contiguos, tiene que haber minimo una casilla en blanco entre dos coordenadas ocupadas por un barco.

## 2.2. Tecnología

En primer lugar, para el desarrollo de este programa hemos usado diferentes herramientas que nos permiten trabajar con el lenguaje de programación C. En este caso, las herramientas utilizadas han sido **Codeblocks**, **Embarcadero Dev C++**, y **VSCode**. Además, hemos utilizado las librerías `stdio.h`, `string.h`, `stdlib.h`, `time.h` y `math.h`.

En segundo lugar, para la coordinación de los diferentes módulos hemos utilizado la herramienta **GitHub**, la cuál nos ha permitido subir los diferentes archivos de cada integrante del equipo la última versión actualizada de nuestro archivo, permitiendo al resto de integrantes avanzar.

## 2.3. Manual de instalación

Para instalar el sistema es necesario tener todos los archivos. incluyendo los archivos de origen. Estos archivos están disponibles en el repositorio de GitHub. Para que sea posible ejecutar el programa basta con incluir todos los archivos del repositorio.

## 2.4. Acceso al sistema

Después de ejecutar al usuario, se le aparecerá al usuario un menú en el que podrá seleccionar una opción en función de lo que decida hacer. Las opciones son las siguientes:

- *Configuración*: en esta opción se le permitirá al usuario modificar aspectos importantes del juego, como por ejemplo, las dimensiones del tablero, características de los diferentes tipos de barcos, etc. Además de indicar el tipo de disparo de cada jugador, de esta forma eligiendo si jugar contra la máquina o contra otra persona.
- *Jugar*: en esta opción el usuario podrá jugar la partida, primero se le pedirá que coloque los barcos de la flota correctamente, de tal forma que no haya dos barcos seguidos. Para colocar los barcos al usuario se le pedirá que indique la id correspondiente al tipo de barco que desea colocar. Posteriormente el usuario introducirá la primera coordenada del barco y la dirección en la que desea colocarlo. Después de colocar los barcos, el usuario podrá disparar al tablero del otro jugador, de esta manera, empezará la partida. La partida acabará cuando uno de los dos jugadores haya hundido toda la flota del jugador oponente.
- *Salir*: al seleccionar esta opción el usuario saldrá del juego.

## 2.5. Manual de referencia

En primer lugar, el usuario puede personalizar el juego, es decir, definir el tamaño del tablero, definir los tipos de barcos así como las unidades de cada tipo de barco que serán utilizados en la flota. Además, el sistema controla que el tamaño de los barcos definidos por el usuario no supere las dimensiones del tablero. Por otro lado, el usuario puede decidir el tipo de disparo, siendo las siguientes opciones:

- *Manual*: es el usuario quien decide donde colocar los barcos y en que coordenada lanza el disparo
- *Automático*: es la máquina quien se encarga de colocar los barcos y de decidir la coordenada donde se lanzará el disparo. Esta opción es utilizada por el usuario, cuando quiere jugar una partida contra la máquina.

En segundo lugar, mientras el usuario está colocando los barcos, se pueden producir varios errores. El primer error es que el usuario introduzca una id no válida de un barco que desea colocar. Si se produce este error, el sistema le volverá a preguntar al usuario por una id válida. Una vez el introducida una id válida se puede producir el segundo error, el cuál es que el usuario introduzca una coordenada o dirección incorrecta. En este caso el sistema le mostrará porque esa coordenada o dirección es incorrecta, y volverá a preguntar al usuario por una id. Además cada vez que el usuario introduzca una id válida se comprobarán si quedan disponibles barcos de ese tipo por colocar. Se dejará de preguntar al usuario por una id cuando se hayan colocado los barcos o cuando ya no sea posible colocar más barcos porque no hay espacio suficiente. También, se muestra la última versión del tablero, antes y después de colocar un barco.

En tercer lugar, el usuario puede decidir en que coordenada realizar el disparo mediante la terminal, es decir, con las teclas "w,s,a,d" el usuario se podrá mover por el tablero y cuando pulse la tecla **enter**, se producirá el disparo en la coordenada indicada por el usuario. A continuación se le mostrará al usuario el resultado del disparo:

- 0: en la coordenada indicada hay *agua*.
- X: en la coordenada indicada hay un barco, es decir, se ha tocado un barco

- H: la coordenada indicada era la última coordenada de un barco que faltaba por tocar, por lo que el barco está hundido.

Por último, el usuario puede solicitar un resumen de la partida, en el cuál se le mostrará una tabla que indicará los disparos realizados, las casillas que son agua, vacías, tocadas, hundidas. Además de los barcos hundidos, los barcos restantes, y el ganador de la partida.

## **2.6. Guía del operador**

Los errores que se pueden producir están relacionados con valores que tiene que introducir el usuario, ya sea una id de un barco, una coordenada incorrecta o algunos datos incorrectos en la configuración del sistema. En estos casos, el sistema es quien se encarga de preguntarle al usuario otra vez por la respuesta.

### 3. Documentación de sistema

Documentación referida a los aspectos del análisis, diseño, implementación y prueba del *software*, así como la implantación del mismo. En general, debe hacer referencia al sistema, ya que está orientada a los programadores que van a realizar el mantenimiento. Debe estar muy bien estructurada, desde lo más general a los más interno.

#### 3.1. Especificación del sistema

Esta sección debe describir el análisis y la especificación de requisitos. Cómo se descompone el problema en distintos subproblemas y los módulos asociados a cada uno de ellos, acompañados de su especificación. También debe incluir el plan de desarrollo del *software*.

#### 3.2. Módulos

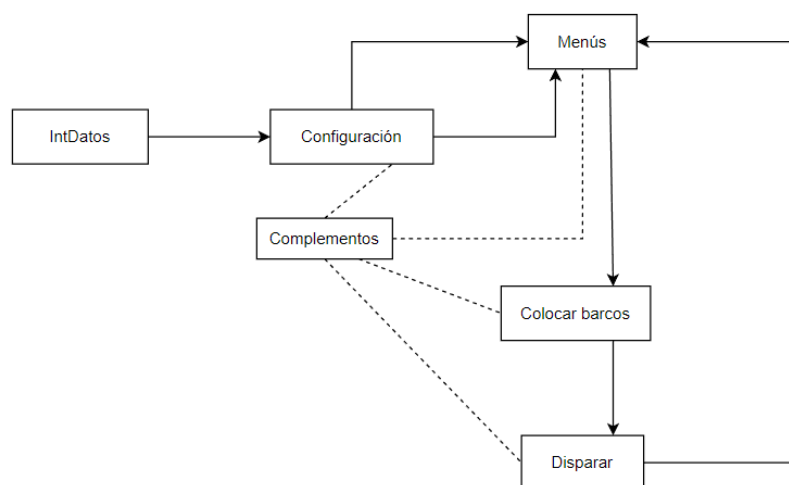


Figura 1: Ornigrama de los módulos

**Menús:** Incluye los archivos MenuP.c y MenuP.h en el que se encuentran todos los menús necesarios para el desarrollom del programa, es decir, la interfaz inicial del programa. Este módulo es el que se encarga de llamar



a las funciones de los demás módulos, de esta manera, se produce la unión de todos los módulos. Además, incluye las funciones de cargar los datos del fichero a los registros correspondientes y viceversa, es decir, incluye las funciones de guardar, las cuales se encargan de guardar en los ficheros los datos almacenados en los registros.

**Complementos:** Módulo que contine funciones básicas y/o recurrentes que se usan a lo largo del programa. Las funciones de este módulo no están relacionadas con un módulo concreto, ya que son funciones que han usado todos lo módulos del proyecto, como por ejemplo la funcion `clear()`, que se encarga de limpiar la terminal. Este módulo incluye los archivos `Complementos.c` y `Complementos.h`.

**Disparar:** Incluye los ficheros `disparar.c` y `disparar.h`. En este módulo es donde se encuentran las funciones de disparo, en las cuáles se encuentra la implementación de la funcionalidad de disparo automático. Tras recopilar toda la información necesaria, como por ejemplo los tableros correspondientes a la flota de cada usuario, mostrará al usuario un tablero por el que podra moverse y disparar mediante un cursor. En caso de estar en automático, se verá lo mismo pero no se requerira ninguna interacción por parte del usuario.

**Colocar barcos:** Incluye los ficheros `colocar_barcos.c` y `colocar_barcos.h`. Este módulo se encarga de la funcionalidad de colocar los barcos en el tablero1 correspondiente a la flota de cada jugador. Para ello, se muestra el tablero actual y los barcos que componen la flota, a continuación el usuario introduce una coordenada y una dirección. Si es posible se colocará el barco en el fichero, si por el contrario, no es posible, se le vuelve a pedir al usuario que introduzca una id de un barco.

Sin embargo si el tipo de disparo es automático, la máquina se encarga de colocar los barcos, y cada vez que esta coloque un nuevo barco en el tablero se mostrará el tablero actual.

**Configuración:** Este módulo incluye los ficheros `configuracion.c` y `configuracion.h`. En el módulo de configuracion es donde se encuentran las funciones que muestran los contenidos de los registros, permiten el borrado de los ficheros y cuenta con un menu desplegable que permite la modificación de los datos de configuración. Cuando el usurario acceda a alguna función del módulo, deberá introducir números dependiendo de lo que quiera hacer para llevar a cabo la tarea deseada.

**IntDatos:** Cuenta con los ficheros IntDatos.c e IntDatos.h. Estos ficheros se encuentran las funciones que se encargan de insertar o modificar un dato, ya sea nombre, número de barcos, dimensiones del tablero, etc. Se muestra al jugador cuando una de sus funciones es convocada por otra función. Le pedirá al jugador desde caracteres hasta números los cuales se almacenaran en el registro correspondiente.

### 3.3. Plan de prueba

En esta sección describiremos las pruebas de caja negra y caja blanca correspondientes a cada módulo.

#### Prueba de los módulos

A continuación, mostraremos las diferentes pruebas realizadas para cada módulo, incluyendo las pruebas de caja negra y caja blanca.

- **Prueba de caja blanca:**
  - *Colocar barcos:* Se puede ver en la figura. 2.
  - *Disparar:* Se puede ver en la figura3.
  - *IntDatos:* Se puede ver en la figura 4. Además está especificado en la carpeta zip de la documentación.
  - *Complementos:* Se puede ver en la figura 5.
- **Prueba de caja negra:** Se encuentra en el documento Caja Negra.tex

```

int id_valida(char id_b, bar_vect* v_barcos){
    int i, sol;
    sol=0; 1
    for(i=0;i<v_barcos->num_tipo_bar;i++){ 2
        if(id_b==v_barcos->bar[i].id_barco){ 3
            sol=1; 4
        } 5
    } 6
    return sol; 7
}

```

$V(G)$ =Número de regiones  
 $V(G)=3$

$V(G)=NA-NN+2$   
 $V(G)=7-6+2=3$

$V(G)=NNP+1$   
 $V(G)=2+1$

RUTAS  
 1º) 1-2-6  
 2º) 1-2-3-4-5-2-6  
 3º) 1-2-3-5-2-6

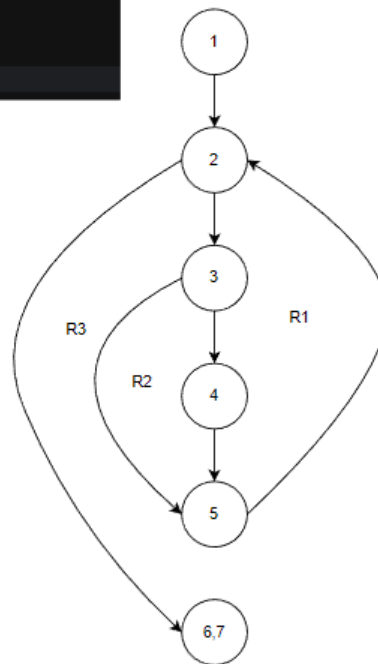


Figura 2: Prueba de ruta básica

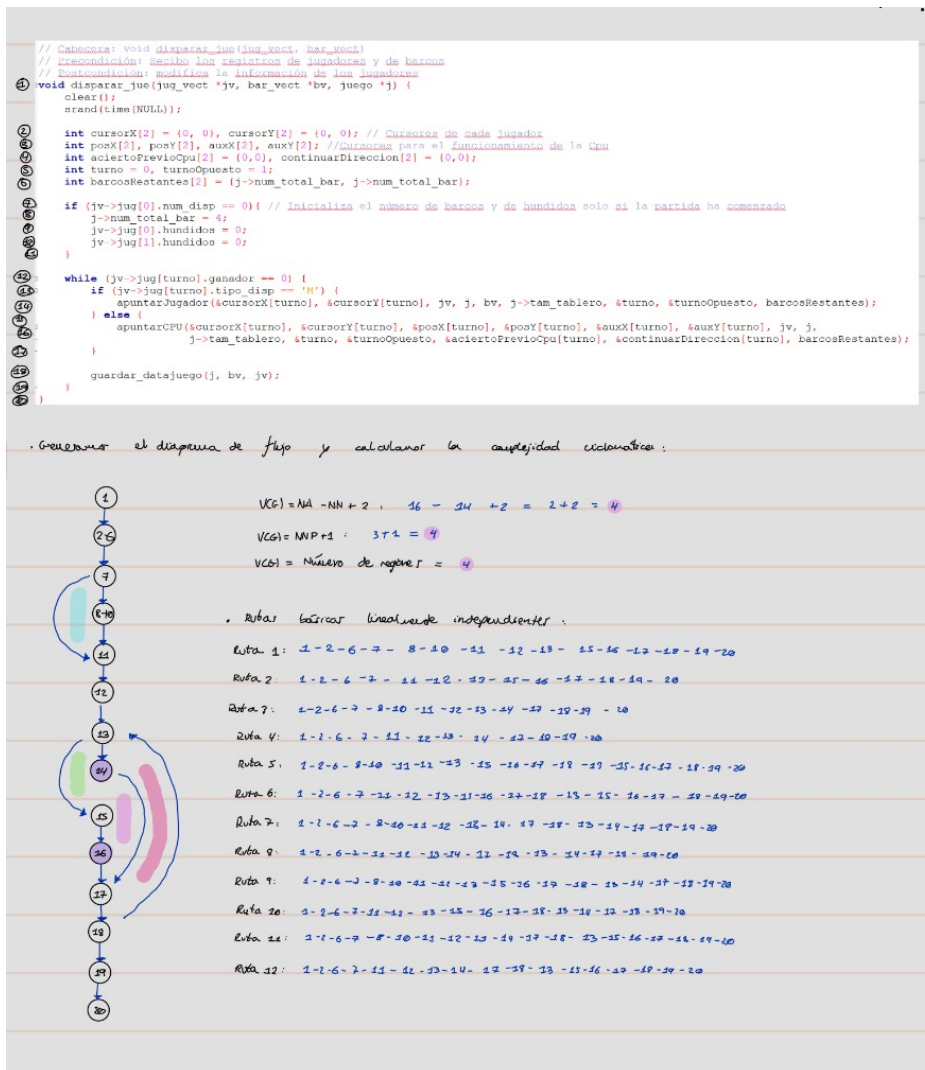


Figura 3: Prueba de ruta básica

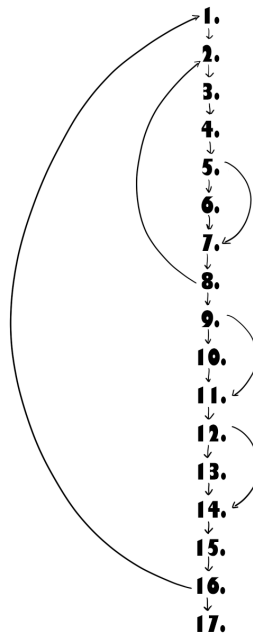


Figura 4: Prueba de ruta básica

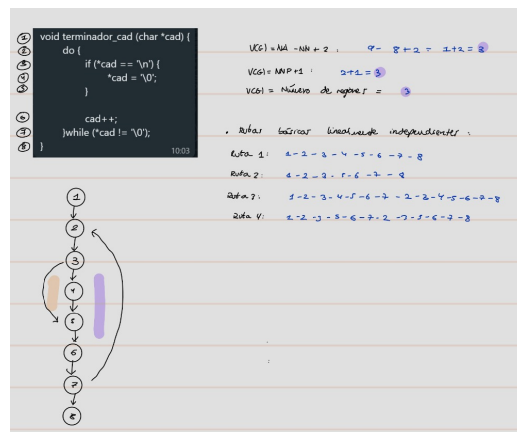


Figura 5: Prueba de ruta básica