



APELLIDOS	NOMBRE		GRUPO	CALIFICACIÓN
ASIGNATURA Técnicas de Programación	FECHA 27/03/2023	DNI	<b>GIM 02</b>	

Hoja 1/13

La duración de la prueba es de dos horas:

- 20 minutos para la evaluación de teoría (2 puntos).
- 80 minutos para la evaluación de práctica (8 puntos).

Antes de comenzar:

- **Los dispositivos electrónicos deben estar guardados en la mochila y apagados o sin sonido.** En caso contrario debe indicarse al profesor los motivos por los que debe tenerse el teléfono con sonido antes de comenzar el examen.
- **Se permite el uso de calculadora, aunque su utilización no es necesaria para la realización del examen.**
- **No se permite la utilización de apuntes, teoría, la escritura en papel, el acceso a recursos externos, hablar, prestando especial atención a la lectura del enunciado del examen en voz alta, o cualquier otra actividad que no sea la escritura de las respuestas en la plantilla de respuestas facilitada en Word.**
- **El uso de cualquier dispositivo electrónico, el plagio o copia conllevará el suspenso del examen y de la evaluación ordinaria, acudiendo directamente a la evaluación extraordinaria según el artículo 9 del reglamento sobre pruebas de evaluación y su revisión:**

Artículo 9º. De la utilización de métodos ilícitos para la superación de las pruebas de evaluación.  
"Cualquier evidencia de plagio, de copia del examen de un compañero, o cualquier intento de obtener de forma fraudulenta las respuestas a las preguntas de una prueba de evaluación, será penalizada con el suspenso en la convocatoria ordinaria y extraordinaria y podrá suponer la apertura de expediente y la aplicación de las correspondientes sanciones, pudiendo llegar a sustanciarse en la expulsión de la Universidad del alumno o alumnos implicados, conforme a lo establecido en los arts. 29 y siguientes del Reglamento del Alumnado de esta Universidad."

- **Contestar en esta hoja, escribiendo en ella el <ApellidosEstudiante>\_<NombreEstudiante> y entregar todas las hojas.**
- **No se podrá abandonar la silla durante la realización del examen.**
- **Cualquier sospecha sobre el incumplimiento de las normas aquí descritas puede concluir en la revisión oral del examen, en la realización de un nuevo examen oral o en el suspenso de la asignatura y posible apertura de expediente académico si corresponde, según la gravedad de las acciones.**

Resultados de aprendizaje que se evalúan en este examen:

- Entender los conocimientos básicos de algorítmica y complejidad computacional.
- Ser capaz de implementar algoritmos básicos.



APELLIDOS	NOMBRE		GRUPO	CALIFICACIÓN
ASIGNATURA Técnicas de Programación	FECHA 27/03/2023	DNI	<b>GIM 02</b>	

Hoja 2/13

## Ejercicio 1 (0.5 puntos) Tiempo estimado: 5 minutos.

Responda a las cuestiones:

La calificación máxima de esta parte tipo test es de 1 punto. Tenga en cuenta que:

- Para cada pregunta, sólo existe una respuesta correcta.
- Cada respuesta correcta sumará 0.10 puntos.
- Cada respuesta incorrecta restará 0.10 puntos.
- Las preguntas no contestadas ni suman ni restan puntuación.

- 1) Los elementos que definen a un objeto son:
  - a) El tipo de visibilidad que tiene asignado: privado o público.
  - b) La identidad, Los atributos que representan su estado y los métodos que representan su comportamiento.
- 2) El hecho de que un método pueda recibir un número variable de atributos se denomina:
  - a) Herencia
  - b) Polimorfismo
- 3) Imagine que la clase C hereda de la clase B. Y la clase B hereda de la clase A. Cuando se cree un objeto de la clase C, ¿cuál será la secuencia en que se ejecutarán los constructores?
  - a) Primero el constructor de C, luego constructor de B y, finalmente, el constructor de A
  - b) Primero el constructor de A, luego constructor de B y, finalmente, el constructor de C
- 4) Para referenciar los atributos de un objeto en un método de la misma clase
  - a) Debo utilizar la palabra reservada self como prefijo
  - b) Los atributos deben ser públicos, en caso contrario no se pueden referenciar desde métodos que no sean el constructor.étodos que no sean el constructor.étodos que no sean el constructor.
- 5) En el paradigma de la programación orientado a objetos:
  - a) Se recomienda que las acciones estén lo más acopladas posibles al código para obtener el mayor rendimiento posible.
  - b) Se favorece la semántica y la reutilización de código en detrimento del rendimiento.



APELLIDOS	NOMBRE		GRUPO	CALIFICACIÓN
ASIGNATURA Técnicas de Programación	FECHA 27/03/2023	DNI	<b>GIM 02</b>	

Hoja 3/13

## Ejercicio 2 (0,5 punto): 5 minutos.

¿Cuáles son las características de la programación orientada a objetos?



APELLIDOS	NOMBRE		GRUPO	CALIFICACIÓN
ASIGNATURA Técnicas de Programación	FECHA 27/03/2023	DNI	<b>GIM 02</b>	

Hoja 4/13

### Ejercicio 3 (0.5 puntos): 5 minutos.

¿Qué es la visibilidad de una variable o un método? ¿Qué tipos hay? ¿Cuándo debe utilizarse cada tipo?



APELLIDOS	NOMBRE		GRUPO	CALIFICACIÓN
ASIGNATURA Técnicas de Programación	FECHA 27/03/2023	DNI	<b>GIM 02</b>	

Hoja 5/13

## Ejercicio 4 (3 puntos) Tiempo estimado: 60 minutos.

El juego hundir la flota, conocido en el mundo anglosajón por su nombre en inglés battleship, es un juego tradicional de estrategia y algo de suerte, que involucra a dos participantes.

En este ejercicio, vamos a hacer una versión programada en Java para que dos jugadores puedan jugar a este juego.

La aplicación va a comenzar permitiendo que cada jugador posicione sus barcos en un único tablero. Cada jugador va a ocupar, según sus preferencias, una misma cantidad de casillas, horizontal y/o verticalmente, en base a las casillas sobre las cuales están representadas sus naves.

Cada jugador va a posicionar como máximo 3 barcos. Así, por ejemplo, cinco casillas consecutivas conforman un portaaviones; tres, una fragata; y una casilla aislada, una canoa.

**No es necesario implementar el docString correspondiente a las funciones y métodos desarrollados, aunque se recomienda hacer el diagrama de flujo de los métodos** en papel de forma previa a su resolución.

En base a estas especificaciones se solicita que:

a) Programe la clase **enumarada CardinalPoints**, que contendrá los puntos cardinales: NORTH, EAST, SOUTH y WEST, por este orden.

b) Programe la clase Ship.

- Incluya los atributos de esta clase atributos y establezca la visibilidad adecuada (público, privado, protegido). Se aconseja utilizar un punto de partida, el tamaño y el enumerado llamado CardinalPoint para poner la dirección en la que se sitúa el barco a partir de un punto de partida **●** utilizar un punto inicial y un punto final y comprobar en el constructor que ambos puntos forman una recta. Se necesitarán más atributos para llevar la cuenta de los toques del barco.
- Programe un constructor que reciba los datos necesarios para crear un barco.
- Programe los métodos setters y getters para la clase en función de lo que necesite.
- Programe el método is\_sunk(self) de la clase Ship. Método para saber si el barco ya está hundido.
- Programe el método get\_shot(self, shot\_point). Implementación recibir disparo. El disparo afectará al barco si se encuentra entre el punto de partida y el punto final. No se tendrá en cuenta si el disparo ha sido previamente hecho en el mismo punto o no. No es necesario hacer las comprobaciones para ver si el disparo está fuera del tablero de juego.
- Pruebe los objetos de la clase Usuario con los test cases.

Para ayudar en el desarrollo de este ejercicio, se le hace entrega de un UML parcialmente completo de la posible implementación de este juego, así como de la clase Point, la cual representa un punto. Se facilitan también los archivos vacíos dónde deberían estar implementadas las clases que se piden y que tienen que ser completadas por el alumno. En dichas clases, están ya añadidos los casos de tests de cada una de ellas.

c) Programe una clase User.

- Incluya los atributos de esta clase y establezca la visibilidad adecuada (público, privado, protegido).

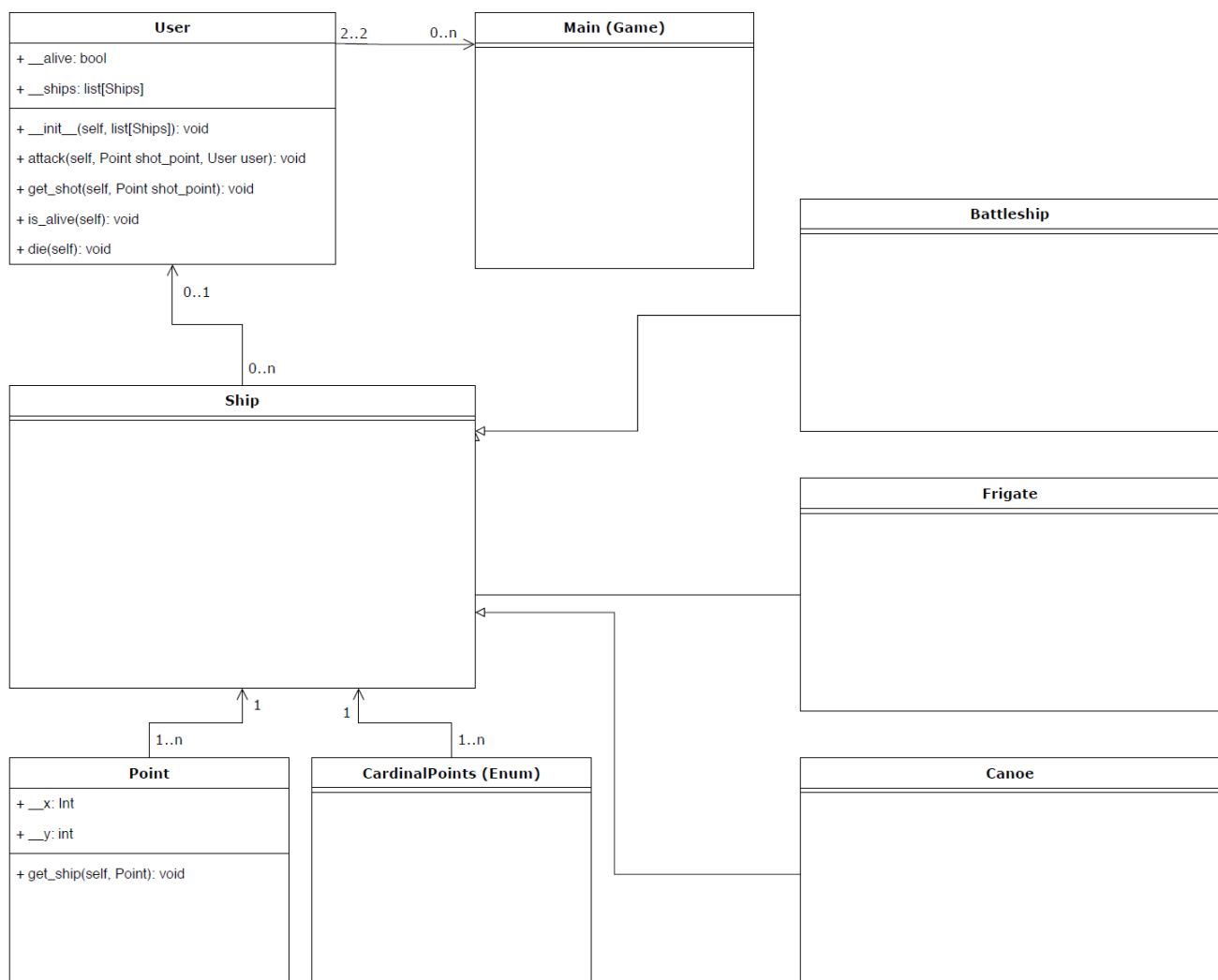
Entregar esta hoja con el nombre y apellidos del alumno, DNI y grupo.



APELLIDOS	NOMBRE		GRUPO	CALIFICACIÓN
ASIGNATURA Técnicas de Programación	FECHA 27/03/2023	DNI	<b>GIM 02</b>	

Hoja 6/13

- Programe un constructor que reciba como parámetro los barcos de inicio del usuario (al menos un barco). El método debe verificar el tipo y valor del parámetro y lanzar la excepción correspondiente cuando no se cumplan los requisitos.
- Programe los métodos setters y getters para la clase en función de lo que necesite. Si no necesita algún o ningún getter y/o setter, argumente por qué en un comentario del módulo.
- Programe el método `attack(self, Point shot_point, User user)`. Este método se utiliza para realizar un disparo al usuario `user` en el punto `shot_point`. El método devolverá `True` si cualquiera de los barcos que no han sido hundidos ha sido acertado y falso en caso contrario. Se debe hacer el control de excepciones para este método.
- Programe el método `get_shot(self, shot_point)` de la clase `Usuario`. Este método se utiliza para recibir un disparo en el punto `shot_point`.
- Programe el método `is_alive(self)` de la clase `Usuario`. Este método `get` se utiliza para obtener el atributo `is_alive`. Este atributo sirve para saber si el usuario aún está vivo.
- Programe el método `die(self)` de la clase `Usuario`. Este método `set` se utiliza para establecer el atributo `__alive` en `False`. De esta forma, es posible matar al usuario.
- Pruebe los objetos de la clase `Usuario` con los test cases que se le han pasado.





APELLIDOS	NOMBRE		GRUPO	CALIFICACIÓN
ASIGNATURA Técnicas de Programación	FECHA 27/03/2023	DNI	<b>GIM 02</b>	

Hoja 7/13

Nota:

- La diferencia en el Battleship, la frigate y la canoe con el Ship es la forma de ser atacado y el tamaño fijo de cada uno de ellos.
- Los barcos y los usuarios no guardan las posiciones



APELLIDOS	NOMBRE		GRUPO	CALIFICACIÓN
ASIGNATURA Técnicas de Programación	FECHA 27/03/2023	DNI	<b>GIM 02</b>	

Hoja 8/13

## Ejercicio 5 (1 punto) Tiempo estimado: 20 minutos.

Se propone una extensión del juego hundir la flota en la que, en vez de jugar con un solo tipo de barco, se pueda jugar con tres tipos de barcos diferentes. Estos tres tipos diferentes de barcos van a ser acorazados (Battleship), fragatas (Frigates), y canoas (Canoes).

En base a estas especificaciones se solicita que:

**Programe una clase Battleship.** Este barco es grande y fuerte con un tamaño fijo (es decir, su tamaño va a ser igual a 5 posiciones del tablero). Además, este barco especial tiene contenedores aislados, por lo tanto, necesita que toquen todas sus posiciones aisladas para hundirlo y requiere de la creación de los atributos correspondientes para controlar esta casuística. La implementación al ser atacado deberá ser modificada.

**Programe una clase Frigate.** Este barco es mediano con un tamaño fijo (es decir, su tamaño va a ser igual a 3 posiciones del tablero).

**Programe una clase Canoe.** Este barco es pequeño con un tamaño fijo (es decir, su tamaño va a ser igual a 1 posiciones del tablero).

Reutilice tanto código como sea posible del ejercicio 4 para hacer esta extensión del juego.

No es necesario implementar el docString correspondiente a las funciones y métodos desarrollados, aunque se recomienda hacerlo para facilitar la comprensión por parte del estudiante.





APELLIDOS	NOMBRE		GRUPO	CALIFICACIÓN
ASIGNATURA Técnicas de Programación	FECHA 27/03/2023	DNI	<b>GIM 02</b>	

Hoja 9/13

## Ejercicio 6 (1,5 puntos) Tiempo estimado: 20 minutos.

En este ejercicio se pide que se implemente el módulo principal del juego hundir la flota. Este módulo va a implementar la partida que van a jugar los jugadores.

En este módulo main, lo primero que se hace es obtener la configuración deseada de los barcos por parte de cada usuario sobre el tablero. Hay que tener en cuenta, que cada usuario solamente va a tener como máximo tres barcos en el tablero al iniciar la partida. La configuración de cada usuario va a ser introducida por consola

Seguidamente, una vez que se tienen las configuraciones, el juego puede comenzar entre los jugadores. En cada turno cada jugador va a atacar al otro jugador en una posición aleatoria. Cuando uno de los dos usuarios tenga ya todos sus barcos hundidos al acabar el turno, se acaba el juego y se indica quien es el ganador. En el caso de que los dos acaben sin barcos en el mismo turno, el juego indicará un empate. En cada turno los dos usuarios atacan con independencia de si sus barcos están hundidos.

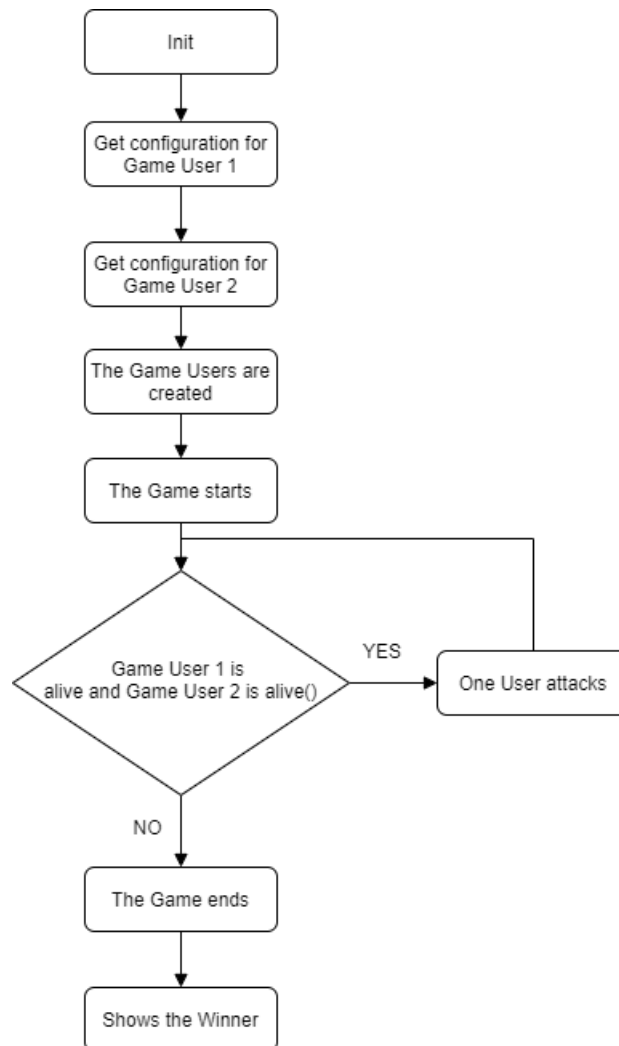
Recuerde utilizar las clases implementadas tanto en el ejercicio 4 como en el ejercicio 5 para implementar todos los componentes de este juego.

El diagrama de flujo de este módulo principal le ha sido proporcionado a modo de guía.



APELLIDOS	NOMBRE		GRUPO	CALIFICACIÓN
ASIGNATURA Técnicas de Programación	FECHA 27/03/2023	DNI	<b>GIM 02</b>	

Hoja 10/13



- La acción One User attacks es realmente el ataque secuencial de User1 a user2 y de user2 a user1 (o viceversa).



APELLIDOS	NOMBRE		GRUPO	CALIFICACIÓN
ASIGNATURA Técnicas de Programación	FECHA 27/03/2023	DNI	<b>GIM 02</b>	

Hoja 11/13

## Ejercicio 7 (1,5 Puntos) Tiempo estimado: 20 minutos

Escribir un algoritmo que permita utilizar tres tablas hash para guardar los datos de barcos, que contemple las siguientes actividades:

- En la primera tabla hash la función hash debe ser sobre el tipo de barco, en la segunda tabla la función hash deberá utilizar el número del barco como clave, mientras que en la tercera el campo clave de la función hash será por el nombre del barco.
- El tamaño de la primera tabla debe ser lo suficientemente grande como para que pueda almacenar todos los distintos tipos de barcos, debe manejar las colisiones con alguna función de sondeo.
- El tamaño de cada una de las segundas tablas debe ser 15.
- El algoritmo debe permitir cargar tipos de barcos en la primera tabla y crear su respectiva segunda tabla, en el caso de que no exista.
- Si el barco es de más de un tipo deberá cargarlo en cada una de las tablas que indiquen estos tipos.
- Deberá permitir cargar barcos de los cuales se dispone de su número, nombre, tipo y nivel.



APELLIDOS	NOMBRE		GRUPO	CALIFICACIÓN
ASIGNATURA Técnicas de Programación	FECHA 27/03/2023	DNI	<b>GIM 02</b>	

Hoja 12/13

## Ejercicio 8 (1,5 Puntos) Tiempo estimado: 20 minutos

Implementar un grafo no dirigido que permita cargar puertos y las aristas que conecten dichos puertos, que permita resolver las siguientes tareas:

- cada arista debe tener la distancia que separa dichos puertos;
- realizar un barrido en profundidad desde el primer puerto en el grafo;
- determinar el camino más corto desde puerto Madero al puerto de Rodas;
- determinar el puerto con mayor número de aristas y eliminarlo.



APELLIDOS	NOMBRE		GRUPO	CALIFICACIÓN
ASIGNATURA Técnicas de Programación	FECHA 27/03/2023	DNI	<b>GIM 02</b>	

Hoja 13/13