

# ESCUELA COLOMBIANA DE INGENIERÍA

## PROGRAMACIÓN ORIENTADA A OBJETOS

### HERENCIA - EXCEPCIONES

#### S07- S09 : 2019-02

## Memoria

Presente el mapa de memoria correspondiente a:

- En el juego se tienen dos flotas “LA GRAN FLOTA BLANCA” y “LA GRAN ARMADA DE CASTILLA”
- “LA GRAN ARMADA DE CASTILLA” tienen un barco (número 900); un porta avión (identificado con 333) y un avión (identificado con 'chkgood'). El avión corresponde al porta avión 333 y actualmente está en el aire.

## Algunos métodos. Diseñando y codificando

Implementen los siguientes métodos de la clase `Flota` (en el orden definido por su profesor)

Method Summary	
<b>void</b>	<b>alNorte()</b> Mueve la flota una posición al norte
<code>ArrayList&lt;Maquina&gt;</code>	<b>seranDestruídas</b> (int longitud, int latitud) Consulta las máquinas que pueden afectarse por una explosión
<b>int</b>	<b>maquinasDebiles()</b> Consulta el número de máquinas débiles Un barco es débil si tienen menos de cinco marinos; un avión, si no tiene piloto principal; y un portaaviones si es un barco débil o alguno de sus aviones en aire es débil.
<b>public boolean</b>	<b>esBuenAtaque</b> (int longitud, int latitud) Verifica si una ubicación de ataque es adecuada (destruye elementos enemigos sin ocasionar bajas propias) <b>Parameters:</b> longitud - longitud de la explosion latitud - latitud de la explosion

## Nueva máquina. Diseñando y codificando.

Las flotas ahora van a tener cápsulas submarinas, sin tripulantes, dirigidas por una máquina nodriza a quién le pide instrucciones. Las cápsulas pueden ser dirigidas por un barco o por otra cápsula, alcanzan profundidades superiores a (8.000 mts) que las hace inmunes a ataques y NUNCA son débiles. Si una cápsula pierde a su nodriza se auto-destruye.

Presente el diseño estructural, analice si debe modificar el diseño de los métodos anteriores y realice el diseño y la codificación completa de esta nueva clase.

## Nuevo comportamiento. Diseñando y codificando

Los marinos, aviones y cápsulas están preparados para auto-destruirse. Las cápsulas se auto-destruyen si su nodriza es destruida. Todos deben poder informar la causa por la que tomaron la decisión. Las flotas conocen los elementos que se han auto destruido.

Presente el diseño estructura. Escriba el código completo del nuevo elemento.

## Otros métodos. Diseñando y codificando.

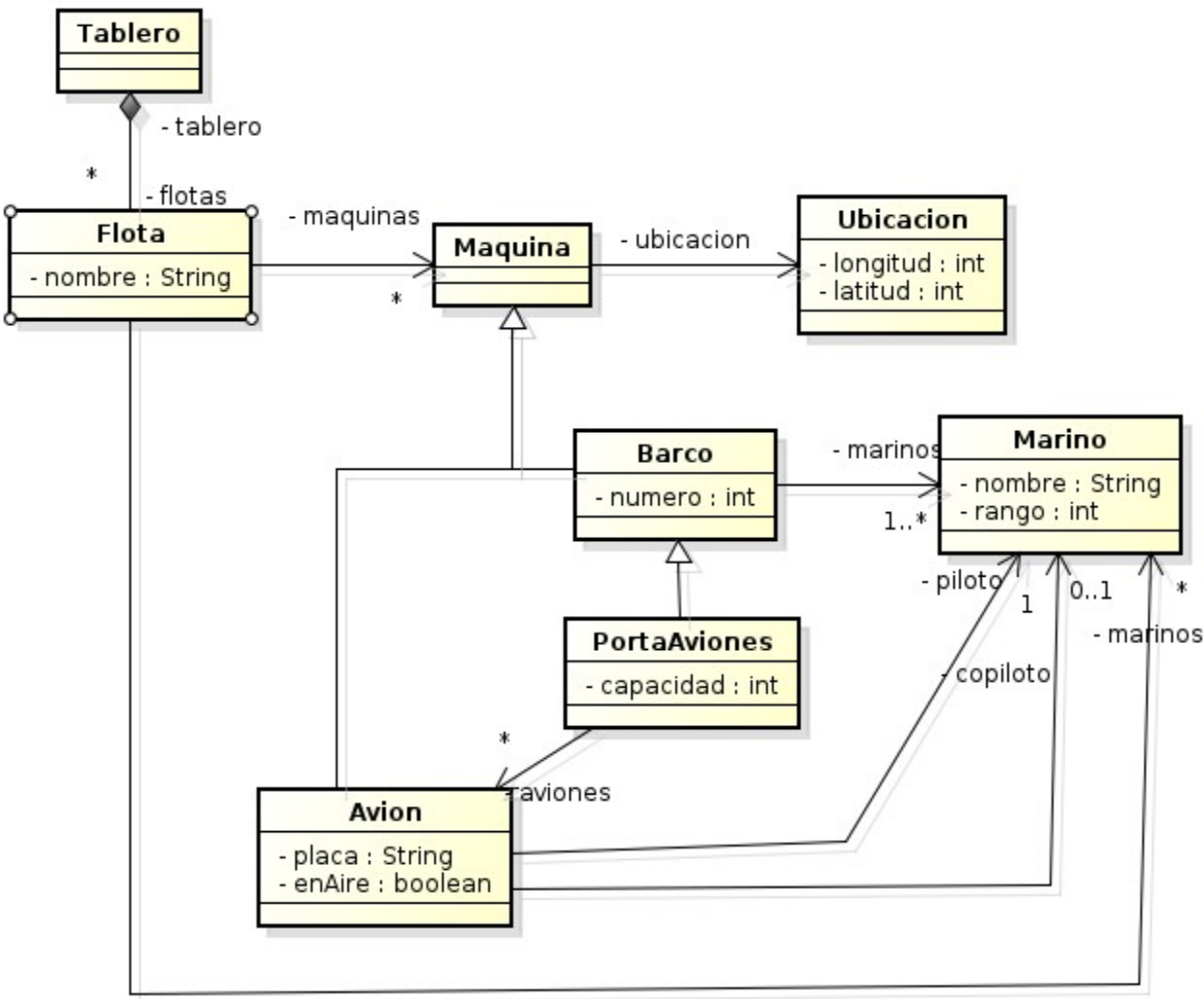
Implementen los métodos de la clase `Flota` y `Tablero` (en el orden definido por su profesor)

Method Summary Flota	
<b>void</b>	<b>alNorte()</b> throws <code>BatallaNavalExcepcion</code> Mueve las máquinas de una flota una posición al norte <b>Throws:</b> <code>BatallaNavalExcepcion</code> - si alguna de las máquinas no se puede mover al norte. El mundo tablero es cuadrado. De (-100,-100) a (100,100)
<code>ArrayList&lt;Marino&gt;</code>	<b>pilotos()</b> throws <code>BatallaNavalExcepcion</code> Consulta los pilotos de la flotas <b>Returns:</b> Los pilotos asignados a los aviones de las flotas

**Throws:** BatallaNavalExcepcion - si un piloto no es marino de la flota, un piloto de un avión asignado a un portavion no es piloto del portaavión o un piloto está asignado a más de un avión

**Method Summary Tablero**

int	<b>alNorte()</b> throws BatallaNavalExcepcion Consulta el número de flotas que lograron un movimiento completo
ArrayList<Flota>	<b>infiltrados()</b> throws BatallaNavalExcepcion Consulta las flotas que tienen pilotos infiltrados <b>Returns:</b> Las flotas con pilotos infiltrados <b>Throws:</b> BatallaNavalExcepcion - si alguna flota no tiene marinos asignados



- Presente el diagrama de secuencia, si es necesario.
- Escriba el código del método. No olvide indicar la clase en la que está escribiendo y la documentación (si no está documentado).
- No construya los básicos (get, set, is)