

**PROYECTO FINAL**  
**Fundamentos de Programación**  
**2015-2**

**Profesor: Carmen Vaca, PhD**

**Introducción**

En el proyecto final usted deberá escribir un programa que haga la simulación de una batalla de barcos.

Su programa permitirá a un jugador disparar en un tablero donde se han colocado barcos aleatoriamente. Los barcos pueden estar en posición vertical u horizontal.

El programa mostrará las siguientes opciones:

- Jugar
- Consultar los tipos de barco y su forma

**Objetivos**

Diseñar e implementar un programa en Python:

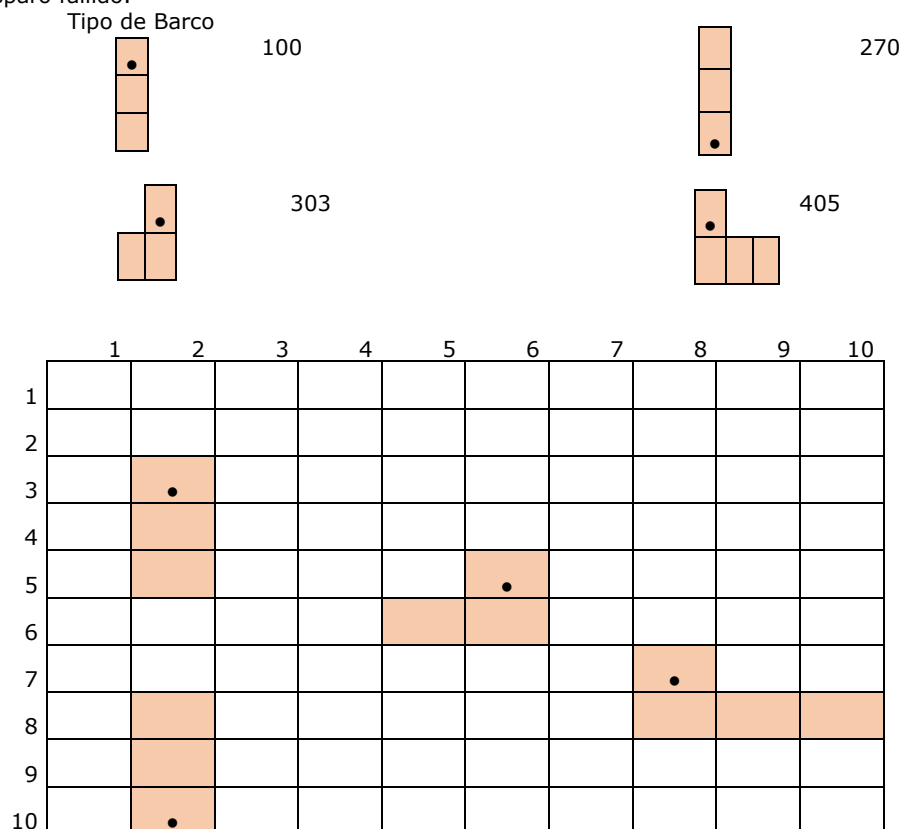
- Desarrollando y usando librerías de funciones relacionadas (al menos una librería )
- Almacenando información en archivos de texto con una determinada estructura

**Los barcos**

El jugador tiene un panel con cuatro barcos. La información de cada barco se almacenará en 1 diccionario. En el diccionario usted almacenará:

- El tipo del barco ( hay 4 posibles tipos de barco: 100,270,303,405)
- El **número** de barco (es un número secuencial que se genera desde 1 para cada barco que se crea)
- Las coordenadas ocupadas por el barco en el tablero.
- Un campo que indica el estado del barco: intacto, golpeado o hundido.
- El país al que pertenece el barco
- Las posiciones que han sido golpeadas.

El tablero inicial luciría así pero se muestran solo una X donde hay un disparo acertado y un guión donde hay un disparo fallido.



Las posiciones vacías se representan con 0, y las casillas donde hay un barco contienen el **número** del barco.

**El juego**

La batalla de barcos se juega usando un tablero de 10x10 que contiene 4 barcos.

Al inicio del juego:

1. Se colocan los barcos horizontalmente de manera aleatoria en el tablero. Los barcos deben ser colocados dentro del tablero, no se puede colocar un barco con una parte fuera del área de juego.
2. Su programa:
  - a. Crea el mar y lee la información del archivo capitales.txt y el archivo fondoMar.txt a diccionarios que serán usados luego.
  - b. Llama a la función **colocarFondoMar**. La **función colocarFondoMar** coloca contenido en casillas del mar (fondo de mar) seleccionadas aleatoriamente. El contenido será tomado de un archivo. Cada fondo de mar tiene puntos asociados y un HINT para adivinar qué es. Esto está en el archivo **nombreDelArchivo.txt**.

Ejemplo:

ATACAR 1,2

Hay algo en el fondo del mar en esa casilla.

Hint : "**HINT**", Adivine: **Coral**

Adivinaste ganas **tantos** puntos.

- c. **Función atacar:**

Cuando ataca gana puntos como se muestra:

- 2 puntos por cada parte de un barco que se hunda

- Puntos correspondientes a Fondo de Mar si hubiera algo allí y el jugador adivina lo que es

5 puntos si en el ataque HUNDE un barco

- d. Cada barco pertenece a un país, cuando el barco es hundido se pide la capital del país, si la adivina gana 3 puntos, si no la adivina pierde 7 puntos.

Los países y sus capitales se leerán del archivo capitales.txt que se adjunta.

- e. Tarea:

Jugar, Atacar, Adivinar las capitales,

Mostrar tablero al iniciar

- Se ingresa la palabra **ATACAR!**.
- Luego se ingresan las coordenadas en donde se quiere atacar

Ejemplo del comando ATACAR,1,2

Si al hacer el disparo "acierta" aparece una X en la casilla en la que disparó si no acierta aparece otro caracter.

Si ha tocado una parte de un barco aparece debajo la palabra **HIT!**. Si con ese disparo hundió el barco aparece la palabra **HUNDIDO!**. Cada vez que ataca debe verificar si encontró algo en el fondo del mar, le muestra el Hint al usuario y si el usuario adivina, gana los puntos correspondientes a ese objeto.

El jugador no puede hundir dos veces el mismo barco ni obtener un HIT dos veces en la misma posición. Para ello, los disparos acertados se representan con -1 en la matriz.

3. El tablero se visualizará después de cada jugada junto con los puntos ganados y el número de barcos hundidos.
4. El punto 4 se repite hasta que el jugador hunda todos los barcos, entonces se escribe en un archivo los puntos acumulados por el jugado, los objetos encontrados en el fondo del mar.. Si el jugador decide retirarse, su programa le permite hacerlo ingresado el comando FIN. En este caso el puntaje también debe almacenarse.
5. Al final del juego sale el tablero con la distribución inicial y final de los barcos.

### Archivo

En un archivo (puntajes.txt) se deben guardar los puntajes de los jugadores con algunos campos por línea:

Alias: String

Puntaje: int

Fecha de último juego

### Variables globales

Recuerde que su programa NO debe usar variables globales.