

# Laboratorio 4: Arquitectura y Organización de Computadores

**Profesor:** Viktor Tapia

**Ayudante de cátedra:** Mauricio Cortés y Muryel Constanzo

**Ayudante de Tarea:** Vicente Alvear y Luciano Yevenes

16 de Marzo 2024

## 1 Reglas Generales

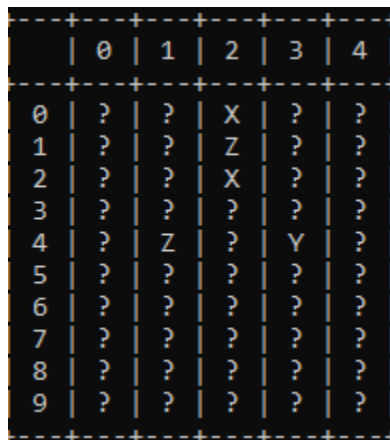
Para esta tarea se utilizará python para generar una simulación del juego de mesa *Battleship* que utilice elementos de conversión de bases numéricas que cumpla con los requerimientos estipulados en la sección 2. Deberá incluir un README con la identificación de los estudiantes que desarrollaron la tarea, además de cualquier supuesto utilizado.

## 2 BITtleship

ARMando y Tramposin juegan todas las semanas una partida de *Battleship* los viernes a la hora de almuerzo. Sin embargo, Tramposin esta muy frustrado por que tiene una racha de 4 partidas perdidas , y le ha preparado un truco a su rival la próxima vez que se reunen. ARMando se da cuenta de ahora las naves de Tramposin solo pueden ser accedidas si se usan coordenadas en bases numéricas distintas a la decimal, y no tiene como ganar. Ya que usted es muy bueno en python y en los contenidos de **INF 245 ARQUITECTURA DE COMPUTADORES** ARMando le ha pedido ayuda en crear un programa que pueda dismantelar las malvadas artimañas de Tramposin.

### 2.1 Lógica del juego

El juego Bittleship va a consistir de una simplificación del juego *Battleship* donde las 'naves' del otro jugador ya son visibles pero solo pueden ser atacadas introduciendo las coordenadas en la base numérica pedida.



|   | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | ? | ? | X | ? | ? |
| 1 | ? | ? | Z | ? | ? |
| 2 | ? | ? | X | ? | ? |
| 3 | ? | ? | ? | ? | ? |
| 4 | ? | Z | ? | Y | ? |
| 5 | ? | ? | ? | ? | ? |
| 6 | ? | ? | ? | ? | ? |
| 7 | ? | ? | ? | ? | ? |
| 8 | ? | ? | ? | ? | ? |
| 9 | ? | ? | ? | ? | ? |

Figure 1: Ejemplo de tablero Bittleship

Como se puede apreciar, hay tres tipos de naves:

- Las naves con letra 'X' deben ser accedidas con base 2 / binario.
- Las naves con letra 'Y' deben ser accedidas con base 8 / octal.
- Las naves con letra 'Z' deben ser accedidas con base 16 / Hexadecimal.

El juego no se termina hasta que todas las naves hayan sido eliminadas. La partida de Bittleship en resumen sería la siguiente:

1. Se inicializa el tablero con el tamaño y la cantidad de naves como parametro.
2. En cada turno se visualiza el mapa y tiene la opción de introducir coordenadas en una de las tres bases numéricas (con input 1,2 y 3 por ejemplo)
3. Se introduce la coordenada x(fila en la matriz) en la base señalada.
4. Se introduce la coordenada y(columna en la matriz) en la base señalada.
5. las coordenadas se convierten a decimal con la función que usted implementó
6. Se accede a la celda de la matriz con las coordenadas en decimal.
7. Se revisa que el tipo de nave (de haber nave en esas coordenadas) sea congruente con el tipo seleccionado al inicio del turno, si esto se cumple, la nave es eliminada del mapa. Se puede usar un caracter a elección para reemplazar las naves hundidas pero '?' o '0' esta bien.
8. Repetir hasta eliminar todas las naves.

Como ejemplo, para las X de la figura uno, se deberían introducir (0,10) y (10,10) respectivamente. Para el resto, el tablero es muy pequeño para que tenga sentido explicar.

### 3 README

Debe contener como mínimo:

- Nombre, Rol y Paralelo de los integrantes.
- Especificación de los algoritmos y desarrollo realizado.
- Supuestos utilizados

### 4 Consideraciones

- Se deberá trabajar de a pares. Se deberá entregar en Aula a mas tardar el día 31 de marzo. de 2023 a las 23:59 horas. Se descontarán 5 puntos por cada hora o fracción de atraso. Las copias serán evaluadas con nota 0 en el promedio de las tareas.
- La tarea debe realizarse usando el lenguaje python. Se asume que ha tenido vivencias con él, o que aprende con rapidez.
- Pueden crear todas las funciones auxiliares que deseen, siempre y cuando estén debidamente comentadas.

- La entrega considera un único archivo de nombre **t1.py** junto con el README. Los archivos deberán ser comprimidos y enviados en un archivo .tar.gz o en .zip (esto queda a su conveniencia) en el formato **LAB1.ROL1.ROL2**.
- Si no se entrega README, o si su programa no funciona, la nota es 0 hasta la corrección.
- Una vez entregadas las notas de la tarea existirá un plazo de 5 días para apelar. Transcurrido este plazo las notas no podrán ser modificadas.
- Se va a usar un tamaño de mapa de hasta 40x80, que es lo que permite la terminal sin desconfiguraciones.
- Se recomienda usar el código entregado para facilitar la visualización, cabe mencionar que usa la librería *Prettytable*.
- Usted debe implementar la conversión en ambos sentidos. **ESTA EXPLICITAMENTE PROHIBIDO CUALQUIER LIBRERÍA O FUNCIÓN QUE DERROTE EL PROPÓSITO DE LA TAREA (ES DECIR, QUE HAGAN LAS CONVERSIONES DE BASE), SI NO SE CUMPLE ESTO TENDRÁ NOTA 0**