**Universidad ICESI**

**Tarea Integradora #2 – Algoritmos y programación II**

**Requerimientos Funcionales.**

**RF1.** El programa debe permitir identificar las casillas de la cuadricula a través de una nomenclatura en la cual la fila esta dad por un numero entero (1, 2, 3, …) y la columna esta dada por una letra mayúscula (desde A hasta Z). Las entradas están dadas de la siguiente manera (A1, B2, A2, B2, …). El resultado es la ubicación de la celda dentro de la matriz.

**RF2.** El programa debe permitir al usuario la posibilidad de disparar un rayo laser de manera horizontal o vertical desde cualquier borde de la matriz con la excepción de que debe indicar la dirección solo cuando se vaya a dispara el rayo desde alguna de las esquinas de la matriz. Las entradas están dadas por la siguiente nomenclatura: 1C, 2D, 4G, … (para celdas diferentes a las esquinas) y 1AH, 1AV, … (para celdas que son bordes de la matriz). El resultado el lanzar el rayo y retornar la celda por la cual sale el rayo.

**RF3.** El programa debe permitir al usuario ingresas los parámetros necesarios para inicializar el juego, esto deben ser un nick\_name, un N y un M que significan las dimensiones de la matriz que desea generar y un K que representa la cantidad de espejos que desea colocar en el juego. Las estadas están dadas por: el nick\_name, n, m, k. El resultado debe ser una partida con una matriz nxm con k espejos dentro de ella.

**RF4.** El programa debe permitir al usuario digitar la ubicación de un espejo dentro de la matriz cuando este considere conocer su ubicación exacta. Las entradas están dadas por la siguiente nomenclatura: L (de Locate) seguido de la celda correspondiente 3C (fila 3 y columna C) seguido de L o R (left o right) que significa la inclinación del espejo. El resultado será mostrar el espejo en la celda que el usuario ingresó siempre y cuando haya acertado en su ubicación, de lo contrario, aparecerá una X indicando que no hay espejo en esa celda.

**RF5.** El programa debe permitir generara aleatoriamente la ubicación y la inclinación de los espejos dentro de la matriz. No tiene entradas. El resultado será la matriz con sus respectivos espejos dentro de ella.

**RF6.** El programa debe permitir señalar con S la celda por donde entra el rayo y con una E la celda por donde sale el rayo. No tiene entradas. El resultado será la matriz señalando con S la entrada del rayo y con E la salida del rayo.

**RF7.** El programa debe permitir mostrar la cantidad de espejos que el usuario debe encontrar y mostrar los restantes a medida que vaya encontrando los espejos uno a uno. No tiene entradas. El resultado será el numero de espejos restante por encontrar.

**RF8.** El programa debe permitir calcular el puntaje obtenido por cada usuario. No tiene entradas. No tiene resultado o salida.

**RF9.** El programa debe permitir almacenar los puntajes de cada jugador en un árbol binario de búsqueda. No tiene entradas. No tiene resultado o salida.

**RF9.** El programa debe permitir recorrer el árbol binario de búsqueda InOrder e imprimir los resultados en dicho orden incluyendo los datos de juego (nick, n, m, k). No tiene entradas. El resultado será la lista de puntajes de los usuarios.