

Laboratorio 3 - Fundamentos de Programación

Fecha de entrega: 28 de Noviembre, 23:55 h (vía aula virtual).

Modalidad: Trabajo en grupo de **máximo** dos personas.

1. Objetivos

El objetivo de este laboratorio es evaluar su capacidad para:

- Llevar a cabo un programa completo en el lenguaje de programación C, utilizando todos los elementos básicos que provee el lenguaje, y sus tipos de datos básicos.
- Resolver un problema, considerando que se cuenta con el procedimiento en lenguaje natural, que se debe implementar para resolverlos.
- Trabajar con tipos de datos estructurados simples.

2. Introducción

El objetivo del juego *buscaminas* consiste en identificar en un tablero/mapa, dividido en celdas, inicialmente desconocido, cuáles de dichas celdas poseen bombas. En cada paso se despeja una celda del mapa basado en la información desplegada por las celdas libres.

En el *buscaminas* se considera un mapa de tamaño $n \times n$ inicialmente desconocido. Se sabe que en algunas de las $n \times n$ celdas pueden estar escondidas una cierta cantidad de bombas (m bombas). La idea es que, en cada paso, se elija una celda para revelar su contenido: Si la celda posee una bomba, el juego termina inmediatamente y el usuario pierde la partida, pero si la casilla no posee bomba se despliega el valor de la celda. El valor de una celda libre de bomba corresponde a la cantidad de bombas adyacentes a la casilla en cuestión.

La figura 1 muestra una celda de ejemplo (en gris) y sus 8 celdas adyacentes. La casilla posee el valor **3** lo que indica que de entre sus celdas adyacentes, 3 de ellas poseen una bomba.

2	2	B
B	3	1
B	1	0

Figura 1: Ejemplo de celda con tres bombas adyacentes.

La figura 2 muestra una celda de ejemplo (en gris otra vez) y sus 8 casillas adyacentes. La casilla posee el valor **8** lo que indica que sus ocho casillas adyacentes poseen bombas.

B	B	B
B	8	B
B	B	B

Figura 2: Ejemplo de celda con ocho bombas adyacentes.

3. Tarea

Construya un programa que permita jugar al *buscaminas* utilizando la interfaz que se muestra en la Figura 3.

```

XXXX  X  X  XXXX  XXXX  XXX  XX XX X X  X  XXX  XXXX
X  X  X  X  XX  XX  X  X X X X X XX  X X  X XX
XXXX  X  X  XXXX  XX  XXXXX X  X X X X X XXXXX XXXX
X  X  X  X  XX  XX  X  X X  X X X XX X  X  XX
XXXX  XXXX  XXXX  XXXX  X  X X  X X X  X X  X XXXX

Ingrese dificultad de la partida (F (facil), M (medio), D (dificil)): D

```

Figura 3: Interfaz principal

Dependiendo la dificultad seleccionada y el tamaño del tablero se debe generar un mapa aleatorio en cada partida. Un *buscaminas* fácil debe poseer aproximadamente 10 % de bombas respecto al total de casillas. Un *buscaminas* medio debe poseer aproximadamente 15 % de bombas y un *buscaminas* difícil debe poseer aproximadamente 20 % de bombas.

Para generar un tablero de $n \times n$ con m bombas utilice la función `construir_tablero` provista en el archivo "libreria.h". Las minas no serán mostradas al usuario en el período de juego tal como muestra la Figura 4.

```

      a  b  c  d  e  f
1  |  |  |  |  |  |  |
   -----
2  |  |  |  |  |  |  |
   -----
3  |  |  |  |  |  |  |
   -----
4  |  |  |  |  |  |  |
   -----
5  |  |  |  |  |  |  |
   -----
6  |  |  |  |  |  |  |
   -----

Ingrese M para marcar, A para abrir una casilla:

```

Figura 4: Partida nueva

Después de elegir Marcar una casilla (M) o Abrirla (A) para mirar su contenido, el usuario a continuación debe indicar la coordenada de la casilla separada por coma, como se muestra en la Figura 5.

```
Ingrese M para marcar, A para abrir una casilla: M
Ingrese coordenada de casilla (ej: 1,a): 2,c

  a  b  c  d  e  f
1 |  |  |  |  |  |  |
  -----
2 |  |  | M |  |  |  |
  -----
3 |  |  |  |  |  |  |
  -----
4 |  |  |  |  |  |  |
  -----
5 |  |  |  |  |  |  |
  -----
6 |  |  |  |  |  |  |
  -----

Ingrese M para marcar, A para abrir una casilla:
```

Figura 5: Interfaz Marcar casilla

En caso de abrir una casilla existen tres posibilidades:

- La casilla no tiene nada. En cuyo caso el tablero queda igual.
- La casilla tiene una mina. En cuyo caso el jugador pierde automáticamente y se debe mostrar el tablero completo con la cantidad de aciertos logrados como se muestra en la figura 6.

```
Ingrese M para marcar, A para abrir una casilla: A
Ingrese coordenada de casilla (ej: 1,a): 4,c

  a  b  c  d  e  f
1 |  |  |  |  |  |  |
  -----
2 | * |  |  | * |  | * |
  -----
3 |  |  |  | 2 |  |  |
  -----
4 |  |  | * |  |  |  |
  -----
5 |  | * |  |  |  |  |
  -----
6 |  |  |  |  | * |  |
  -----

Perdiste!!!
Acertaste con 0 minas.
```

Figura 6: Interfaz apertura de una celda con mina/falla en la partida.

- La casilla tiene un número que indica cuántas minas hay a su alrededor. La Figura 7 muestra la interfaz al Abrir una casilla que contiene un número. Usted debe implementar una función que permita calcular este número considerando las casillas adyacentes.

```
Ingrese M para marcar, A para abrir una casilla: A
Ingrese coordenada de casilla (ej: 1,a): 3,d

  a  b  c  d  e  f
1 |  |  |  |  |  |  |
  -----
2 |  |  | M |  |  |  |
  -----
3 |  |  |  | 2 |  |  |
  -----
4 |  |  |  |  |  |  |
  -----
5 |  |  |  |  |  |  |
  -----
6 |  |  |  |  |  |  |
  -----

Ingrese M para marcar, A para abrir una casilla:
```

Figura 7: Interfaz de apertura de una celda con número.

- El usuario gana la partida cuando todas las minas han sido satisfactoriamente marcadas, en cuyo caso se le debe mostrar un mensaje de felicitaciones por la victoria como se muestra en la Figura 8.

```
Ingrese M para marcar, A para abrir una casilla: M
Ingrese coordenada de casilla (ej: 1,a): 4,c

  a  b  c  d  e  f
1 |  |  |  | 1 |  |  |
  -----
2 | M | 1 |  | M | 2 | M |
  -----
3 | 1 |  | 2 | 2 |  | 1 |
  -----
4 |  | 2 | M | 1 |  |  |
  -----
5 | 1 | M | 2 | 1 | 1 | 1 |
  -----
6 |  | 1 |  | 1 | M | 1 |
  -----

Ganaste!!!
Marcaste correctamente las 6 minas.
```

Figura 8: Interfaz apertura de una celda con mina/falla en la partida.

NOTA: La función `mostrar_tablero` toma como segundo argumento un entero que:

- Si es 1, muestra todo el contenido del tablero, incluyendo minas.
- Si es 2, muestra el tablero sólo con las marcas (M) que el usuario haya ingresado.

3.1. Sobre el código

1. El tablero es cuadrado y su tamaño depende de una constante definida en el archivo "libreria.h" este tamaño no debe ser menor a 6 y no mayor a 26.
2. La cantidad de minas a disponer en el tablero vendrá dada por el tamaño del tablero y la dificultad seleccionada.
3. Su programa debe ser capaz de mostrar al usuario solo el contenido de las casillas que él ha abierto durante la partida, omitiendo otras casillas no exploradas.
4. Su programa debe ser capaz de mostrar el contenido de todas las casillas minadas una vez que el usuario pierde, borrando las casillas que contengan M y que no sean aciertos de minas marcadas.
5. Debe usar identificadores representativos para sus variables y constantes.
6. Debe comentar cada una de las funciones que posea su programa, indicando para ellas:
 - Parámetros de entrada.
 - Salida.
 - Descripción.
7. Su código debe estar correctamente indentado (uso de sangrías para cada sub-bloque de instrucciones).

3.2. Sobre las validaciones e interfaz con el usuario

1. Su programa debe ser capaz de reconocer si se ha ingresado una opción válida o no.
2. En cada nueva partida la posición de las minas debe ser totalmente distinta, para esto use las funciones dispuestas en la librería provista.

3.3. Sobre el manual de usuario

La interfaz del programa solicitado está totalmente establecida por lo que no se pedirá Manual de Usuario. Esto quiere decir que usted debe cumplir al pie de la letra con la interfaz declarada en este enunciado, sino tendrá descuentos en su nota.

3.4. Archivos disponibles

Para facilitarles el trabajo de la interfaz, se pondrá a su disposición junto a este enunciado el archivo "libreria.h". Este archivo contiene las funciones para “dibujar” las diferentes interfaces con el usuario y las de despliegue y construcción de tableros. Usted solo deberá usarlas en su programa. En caso de que le resulte necesario modificar alguna de las funciones provistas, favor comentarlo en el mismo código de la función. Las funciones que posee este archivo son:

- Titulo();
- LimpiaPantalla();

- `crear_minas(char tablero[tam][tam]);`
- `construir_tablero(char tablero[tam][tam]);`
- `mostrar_tablero(char tablero[tam][tam]);`

4. Reglas del Laboratorio

- Debe implementar este laboratorio en lenguaje C (no C++).
- Se deben entregar todos los códigos fuentes en un archivo comprimido denominado **lab3-apellido1-apellido2.zip**
- Al comienzo de cada código debe ir comentado los nombres de los integrantes.
- El archivo debe ser entregado en la tarea **Laboratorio 3** que estará disponible en el aula virtual hasta el **jueves 28 de noviembre** a las **23:55 h.**
- Puede trabajar con el IDE, entorno y compilador de C, que más le acomode. No obstante lo anterior, ~~su código debe ser compilable y ejecutable sin problemas bajo ambiente Linux.~~
- Frente a cualquier caso de copia, se evaluará a los alumnos involucrados con nota 1.0.