Universidad Nacional Experimental de Guayana

Vicerrectorado Académico

Coordinación General de Pregrado

Estado Bolívar

Puerto Ordaz

**Mini-Proyecto I: Buscaminas en C**

Docente: Alumno:

Ing. William Mercado Sebastián Landaeta

Cédula:

28.240.979

Carrera:

Ingeniería Informática

Materia:

Técnicas de

Programación II

Sección:

1

23/01/2022

**Índice**

Introducción………………………………………………………………………. 3

¿Qué es el Buscaminas y cómo se juega?................................................... 4

Características fundamentales del código…………………………………….. 5

Partes del código………………………………………………………………. 5-9

Capturas del juego en ejecución……………………………………………. 9-10

Conclusiones y testimonios personales del autor…………………………… 11

Bibliografías…………………………………………………………………….. 12

**Introducción**

El presente trabajo es la documentación del código fuente de un Buscaminas en C. Como es bien sabido el en mundo del software, el código fuente de un programa, es un conjunto de instrucciones (comúnmente llamadas líneas de código), las cuales traduce un compilador o interprete, con el fin de que el procesador lea esas instrucciones y las ejecute en el orden que son declaradas.

Los códigos fuente están escritos en un lenguaje de programación entendible para los humanos, aunque a veces, por la complejidad del programa, leer el código sin más no es lo suficientemente comprensible, y es necesaria la explicación del creador (o los creadores) para conocer el funcionamiento de cada elemento presente en el software. Es por ello, que los lenguajes de programación implementan la posibilidad de escribir “notas” dentro del código fuente, las cuales son líneas que no leerá el compilador, sino que cumplen la función de explicar partes del programa o plasmar información que desea reflejar el programador.

Sin embargo, también es una buena costumbre, escribir documentos de texto los cuales tengan la información necesaria para comprender el funcionamiento de un programa o sistema informático. Así, si entra un nuevo desarrollador a un proyecto, o un programador desea hacerle modificaciones a un programa open source (código abierto en español), podrá leer la documentación para estar al corriente del comportamiento del software.

Habiendo leído lo anterior, se puede apreciar la importancia de la documentación en el desarrollo de proyectos, y como se mencionó al principio, este trabajo tiene como propósito explicar las partes que componen al proyecto “Buscaminas en C”, desarrollado por Sebastián Landaeta, inspirado en el código fuente de Miguel Rosas. Dicho trabajo se dividirá en distintos tópicos, los cuales están reflejados en el índice del trabajo.

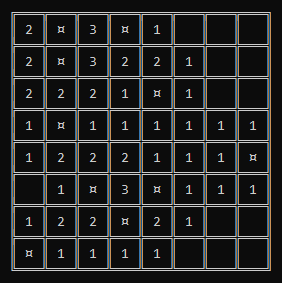
Sin nada más que agregar, de parte del autor, espero que este trabajo resulte tanto interesante, como educativo, y dé una imagen clara del funcionamiento del programa, el cual es una recreación de un juego mítico creado en 1989 por Robert Donner y Curt Johnson.

**¿Qué es el Buscaminas y cómo se juega?**

Buscaminas (o Minesweeper como se conoce en inglés), es un videojuego de un solo jugador cuyo objetivo es despejar todas las minas que se encuentran en un tablero de dimensiones nxn. Para ello, el jugador cuenta con un número limitado de banderas (igual al número de minas que hay en el tablero), las cuales le permiten marcar los lugares en donde piensa que hay minas.

Al comenzar una partida, todas las casillas del tablero estarán cerradas, por lo que el jugador deberá abrir algunas casillas para comenzar a jugar. Si el jugador abre una casilla, pueden ocurrir tres cosas: La primera es que no haya nada, la segunda es que haya un número, y la tercera es que haya una mina.

Si en la casilla abierta no hay nada, es porque no tiene mina, y además no hay minas a su alrededor. Si en la casilla hay un número, significa que la misma no tiene mina, pero que a su alrededor hay por lo menos una mina. Para eso sirve el número, para indicar cuantas minas hay alrededor de una casilla. Es importante aclarar que cada casilla puede tener hasta 8 minas a su alrededor, por ende, el número más grande que puede salir en una casilla es 8, y de ser ese el caso, el jugador puede marcar con confianza todas las casillas que haya alrededor de ese número, usando sus banderas (aunque la probabilidad de que eso pase es bastante baja). Finalmente, si la casilla tiene una mina, el jugador pierde inmediatamente, y se muestra en pantalla todas las minas que había en el tablero.



**Características fundamentales del código**

* *Lenguaje de programación*: C.
* *Entorno de desarrollo*: Dev-C++.
* *Variables globales*: Ninguna.
* *Librerías*: stdio.h (standard input-output header)

stdlib.h (standard library header).

* *Funciones declaradas*: “configuración\_de\_juego()”,

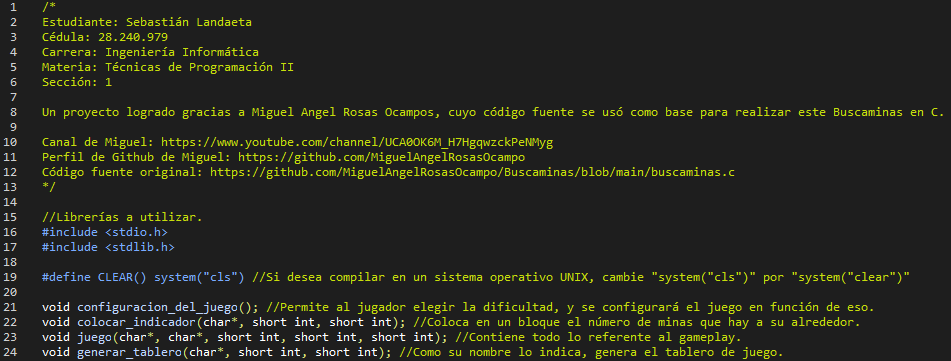
“colocar\_indicador(char\*, short int, short int)”,

“juego(char\*, char\*, short int, short int, short int)”

“generar\_tablero(char\*, short int, short int)”.

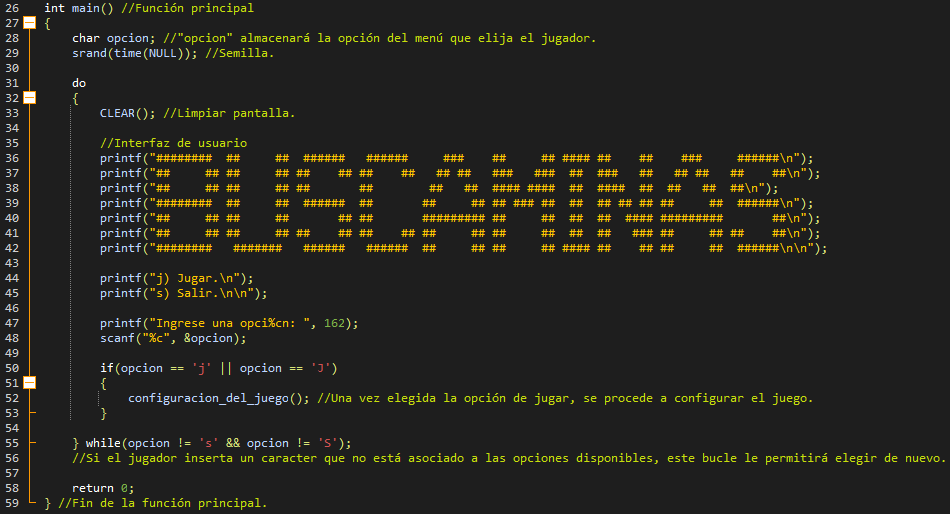
* *Macros declaradas*: “CLEAR()”. Sustituye a “system(“cls”)”.
* *Líneas de código*: 423.
* *Caracteres en total*: 18379.

**Partes del código**

Lo primero que se puede ver al abrir el código fuente, son los datos del estudiante que realizó el proyecto, seguido de un agradecimiento a Miguel Rosas e información referente a su persona. Más abajo, se pueden ver las librerías utilizadas, la macro para limpiar pantalla, y finalmente los prototipos de las funciones que componen al programa (sin contar a la función main).

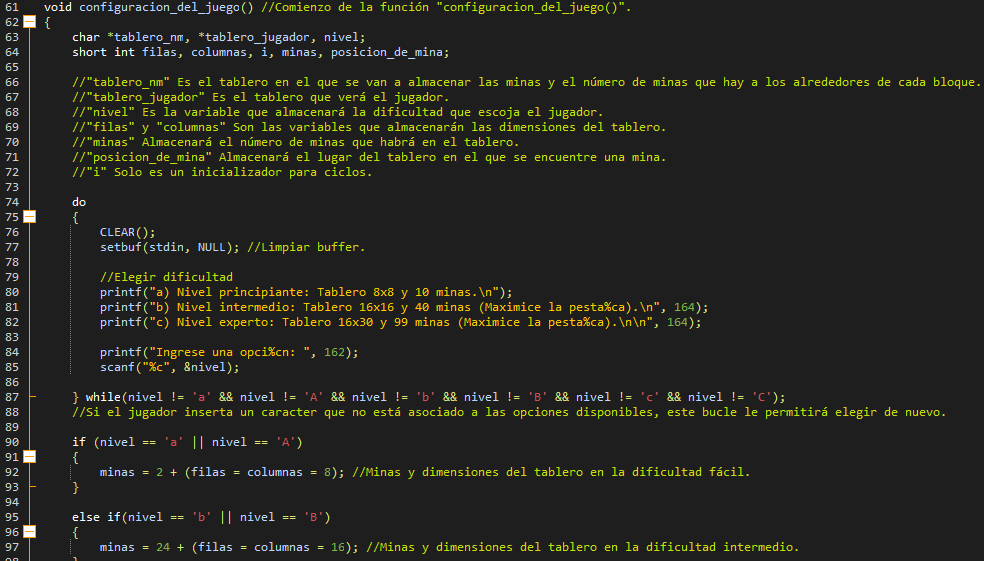
Es importante leer y analizar las notas que hay en todo el código fuente, ya que las mismas indican información relevante para entender el funcionamiento del programa. Ahí se explican hasta las variables usadas en cada función y lo que almacena cada una. En caso de haber duda, consultarla con dichas notas.

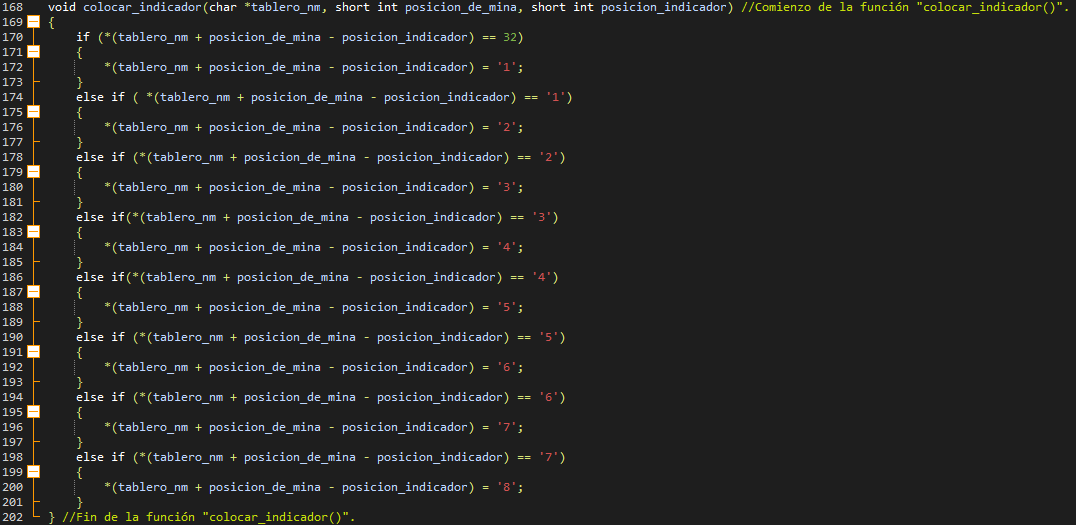
* *Función principal*: En la función principal se encuentra la interfaz, la cual le muestra al usuario el título del juego y las opciones disponibles, las cuales son jugar o salirse del programa. Si el usuario eligió jugar, se ejecutará la función “configuración\_de\_juego()”. Por el contrario, si eligió salir, el programa se cerrará inmediatamente.



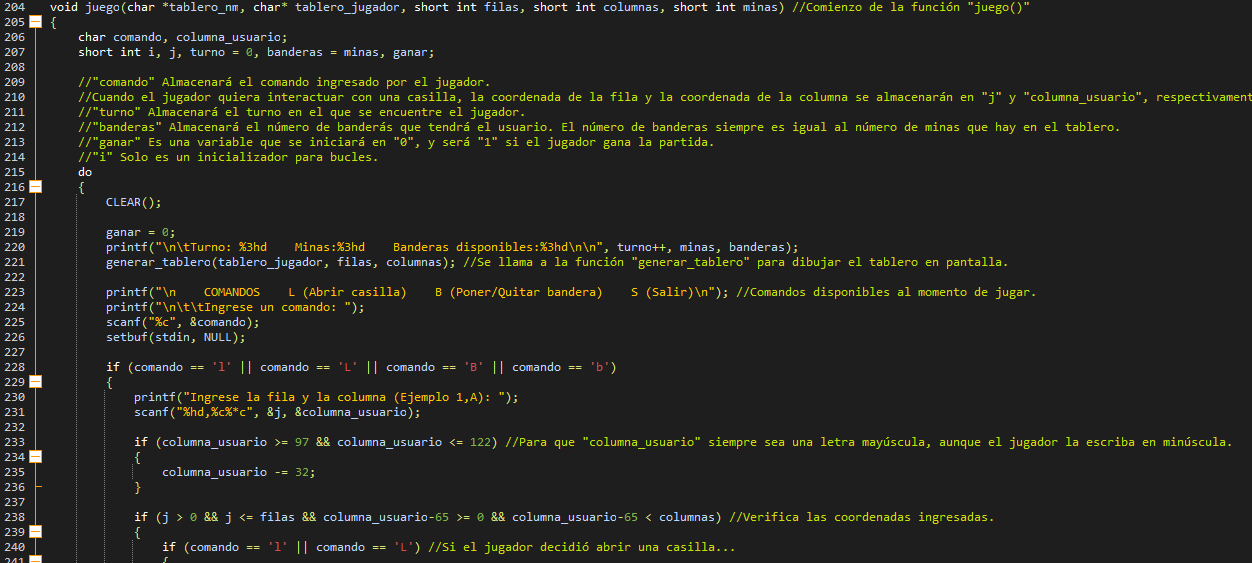
* *Función “configuración\_de\_juego()”*: Esta función le permite al usuario elegir la dificultad, y en base a ella, se configurará todo lo necesario para poder iniciar la partida. La dificultad radica en el tamaño del tablero y el número de minas presentes en el mismo. En la dificultad principiante, el tablero será un 8x8 y solo tendrá 10 minas. En la dificultad intermedio, el tablero será un 16x16 y tendrá 40 minas. Y en la dificultad experto, el tablero será un 16x30 y tendrá 99 minas.

A parte de permitirle al usuario elegir la dificultad, esta función también se encarga de crear dos tableros, uno el cual verá el usuario al momento de jugar, y otro con las minas, los espacios vacíos y los números. Para lograr esto último, la función llama a otra función, la cual tiene como nombre “colocar\_indicador()”.



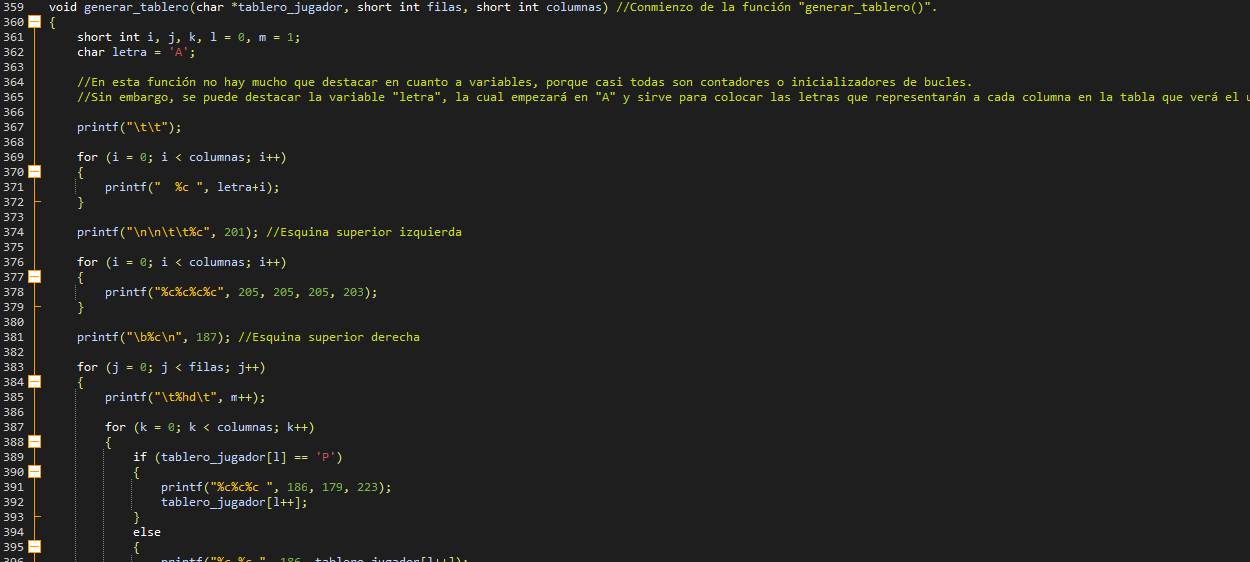
* *Función “colocar\_indicador()”*: Se dedica única y exclusivamente a colocar en las casillas sin mina, el número de minas que hay a sus alrededores. Esta función no tiene variables locales, ya que solo depende de los argumentos que se le pasan desde “configuración\_de\_juego()”.

Cuando ya se crean los vectores dinámicos y las variables con los valores necesarios para comenzar la partida, se procede en enviar todo eso por valor a la función más importante del código, la cual se llama “juego()”.

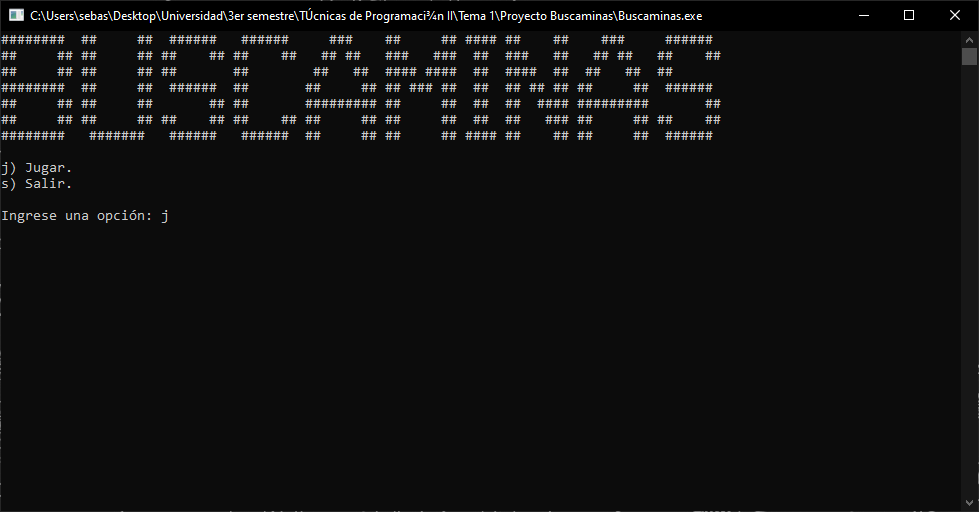
* *Función “juego()”*: Esta es la función que más argumentos recibe, y es la encargada de integrar todo y permitirle al jugador iniciar una partida. En esta implementación del Buscaminas, todo funciona a través de comandos, desde poner o quitar una bandera, hasta seleccionar y abrir una casilla. Por ende, son necesarias variables que almacenen valores como el número de banderas disponibles, el turno en el que va el usuario, la coordenada de la fila, la coordenada de la columna, o el comando ingresado. Todas esas variables están presentes en esta parte del código, y se opera con ellas a través de estructuras condicionales y ciclos. También está presente la condición de victoria, y la condición de derrota.

Aunque falta responder una pregunta. ¿Cómo se dibuja el tablero en pantalla? Bueno, los tableros se dibujan por medio de una función llamada “generar\_tablero()”, la cual es llamada constantemente por la función “juego()”, debido a que se dibuja un nuevo tablero cuando el usuario interactúa con el entorno.

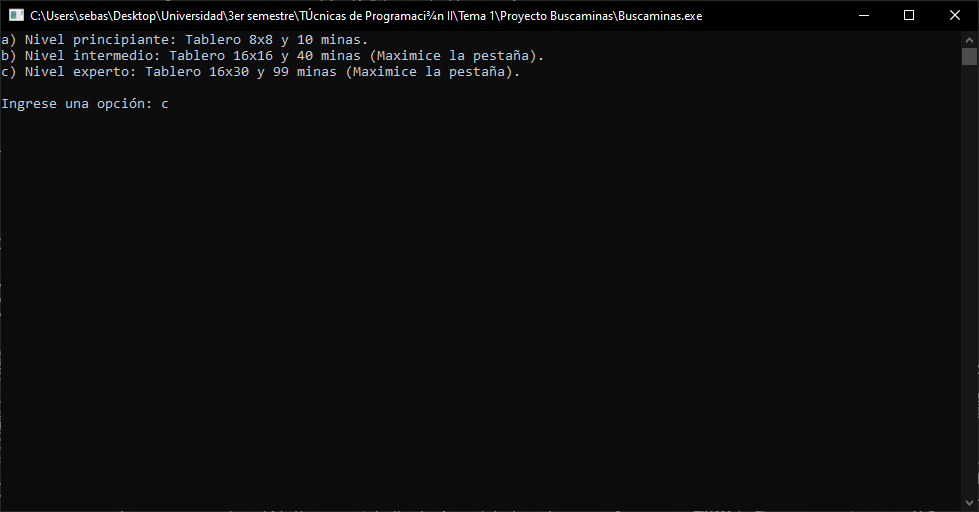
* *Función “generar\_tablero()”*: Como su nombre lo indica, esta función genera el tablero gráfico que verá el jugador en partida. Como en C no hay librerías gráficas, todo el tablero es creado usando caracteres ASCII. La función recibe como argumentos el vector dinámico que contiene los elementos visuales del tablero, la variable con el número de filas, y la variable con el número de columnas, ya que estas últimas son necesarias para saber las dimensiones del tablero e imprimir la cantidad exacta de elementos.



**Capturas del juego en ejecución**

****

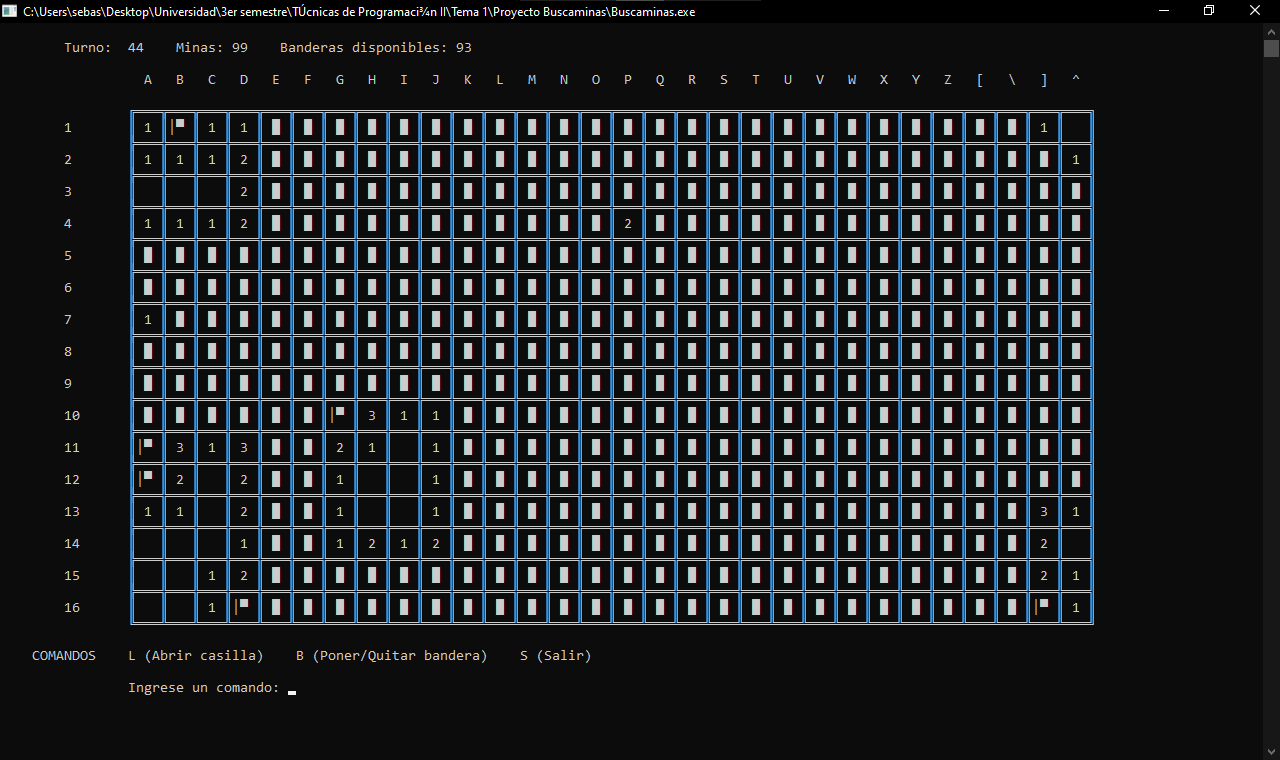
*Interfaz de usuario.*

**

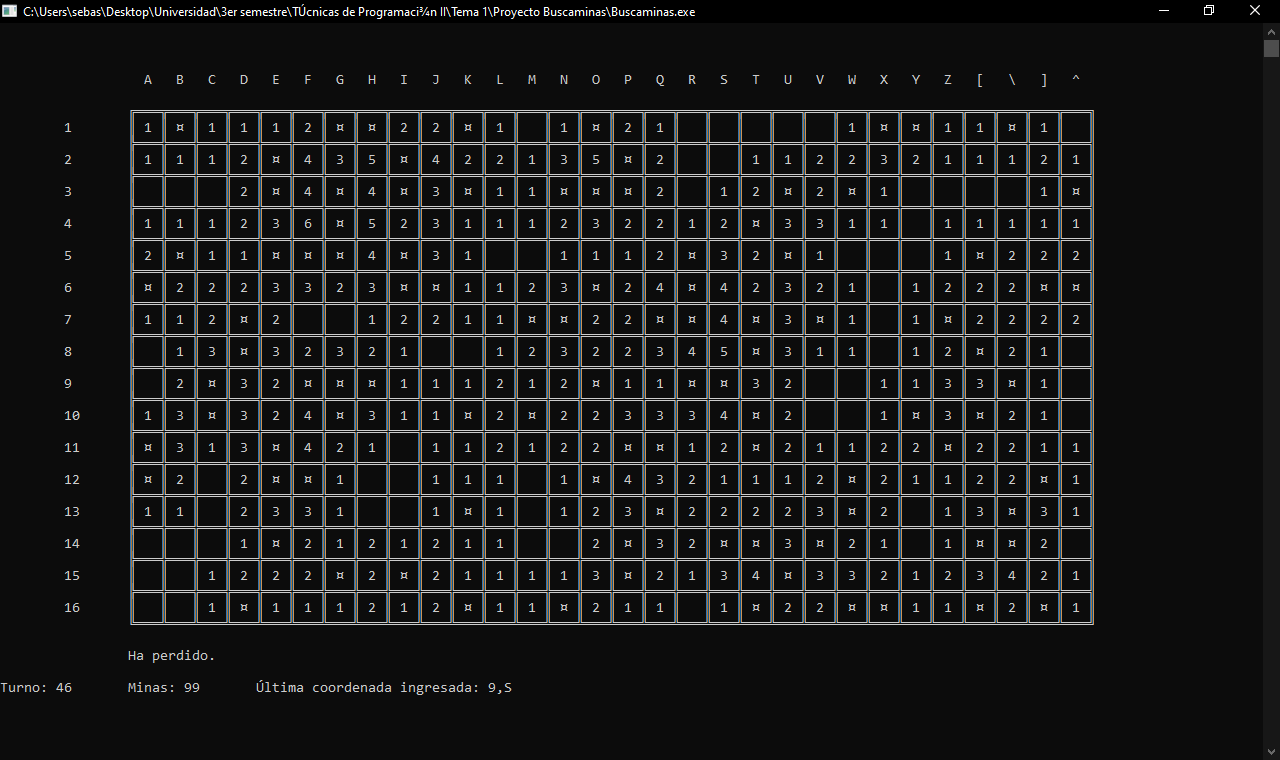
*Elegir dificultad.*

**

*Gameplay.*

**

*Casillas con números, espacios vacíos y banderas.*

**

*Tablero con todas las casillas abiertas.*

**Conclusiones y testimonios personales del autor**

Este proyecto fue más complicado de lo que pensé, ya no solo por la dificultad que conlleva hacer un buscaminas en C, sino también por documentar cada detalle presente el en código fuente, tener que analizar el código de Miguel Rosas de forma rigurosa, corregir errores presentes en su código sin estropear el funcionamiento del programa, eliminar variables y elementos que estaban siendo inutilizados, entre muchas otras cosas. Este fue un trabajo de varios días, el cual consumió bastante tiempo, tanto de codificación, como de investigación y lectura.

Sin embargo, estoy bastante contento con el resultado. A decir verdad, no pensé que quedaría tan bien, y verlo funcionar sin problemas, me llena de satisfacción, más teniendo en cuenta que antes de conseguir el código de Miguel, tuve que probar y leer otros códigos fuente, que, a pesar de ser funcionales, no cumplían con los requisitos necesarios para ser considerados óptimos. En este proyecto, reafirmé lo ciertas que son dos frases típicas en el mundo de la programación, una es: “*No reinventes la rueda*”, y la otra es: “*El programador pasa más tiempo leyendo código, que escribiendo código*”.

Adjunto a este documento, estará presente el código fuente original de Miguel, y el código modificado por el autor. Cabe mencionar, que tanto mi código como el de Miguel son open source, por lo cual se pueden modificar o usar con total libertad. Sin nada más que decir, espero que este trabajo haya del agrado del lector.

**Bibliografías**

* Definición de Código fuente. 7 de febrero de 2021. *ConceptoDefinición*.

Recuperado de: https://conceptodefinicion.de/codigo-fuente/

* Todo lo que sabías de Buscaminas está mal. 27 de diciembre de 2019. *Canalrcn.com*. SuperTrending.

Recuperado de: <https://www.canalrcn.com/supertrending/trucos-y-guias/articulo-nota/todo-lo-que-sabias-de-buscaminas-esta-mal-431>

* Cómo se juega al Buscaminas: Aprende de una vez por todas. 21 de enero de 2022. *Alfa beta*. Raquel Morales.

Recuperado de: <https://alfabetajuega.com/guia/como-se-juega-buscaminas>

* Miguel Angel Rosas Ocampos[Miguel Angel Rosas Ocampos]. (12 de agosto de 2020). *Lenguaje C – Buscaminas*. Youtube.

Recuperado de: <https://www.youtube.com/watch?v=tVEyvIKV7uo&t=65s>

* Miguel Angel Rosas Ocampos[MiguelAngelRosasOcampo]. (12 de octubre de 2020). *Buscaminas / Buscaminas.c.* Github.

Recuperado de : <https://github.com/MiguelAngelRosasOcampo/Buscaminas/blob/main/buscaminas.c>