3CM5

Instituto Politécnico Nacional

ESCOM

Redes de Computadoras

Práctica 2

Buscaminas

Alumno:

Naranjo Miranda Javier Said

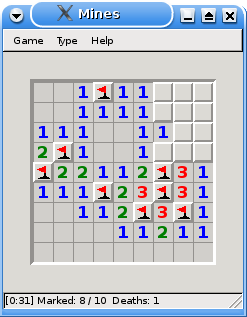
Flores García Sergio

Grupo: 3CM5

**Introducción**

**Introducción**

El juego Buscaminas es un videojuego para un jugador inventado por Robert Donner en 1989. El objetivo del juego es despejar un campo de minas sin detonar ninguna mina.



El juego consiste en despejar todas las casillas de una pantalla que no oculten una mina.

Algunas casillas tienen un número, este número indica las minas que suman todas las casillas circundantes. Así, si una casilla tiene el número 3, significa que de las ocho casillas que hay alrededor (si no es en una esquina o borde) hay 3 con minas y 5 sin minas. Si se descubre una casilla sin número indica que ninguna de las casillas vecinas tiene mina y estas se descubren automáticamente.

Si se descubre una casilla con una mina se pierde la partida.

Se puede poner una marca en las casillas que el jugador piensa que hay minas para ayudar a descubrir la que están cerca.

El juego también posee un sistema de récords para cada uno de Los niveles en el que se indica el menor tiempo en terminar el juego. Los niveles son (para las nuevas versiones):

• Nivel principiante: tablero de 9 × 9 casillas y 10 minas.

• Nivel intermedio: tablero de 16 × 16 casillas y 40 minas.

• Nivel experto: tablero de 16 × 30 casillas y 99 minas.

El juego se desarrollará por medio de Sockets de Flujo

**Código**

Cliente.java

package buscaminas;

import java.io.\*;

import java.net.\*;

import java.util.ArrayList;

import java.util.Calendar;

import java.util.InputMismatchException;

import java.util.List;

import java.util.Scanner;

/\*\*

\*

\* @author saidn

\*/

public class Cliente {

private static final String HOST = "127.0.0.1";

private static final int PUERTO = 5000;

public static final byte NIVEL\_PRINCIPANTE = 9;

public static final byte NIVEL\_INTERMEDIO = 16;

public static final byte NIVEL\_EXPERTO = 30;

public static final byte MINAS\_NIVEL\_PRINCIPANTE = 3;

public static final byte MINAS\_NIVEL\_INTERMEDIO = 40;

public static final byte MINAS\_NIVEL\_EXPERTO = 99;

public static List<Casillas> tablero = new ArrayList<>();

public static byte filas, columnas, intentos, minasDificultad, numeroMinas, f, c;

public static boolean Ganador = true;

private static int hora;

private static int min;

private static int seg;

private static int tiempoI;

private static int tiempoF;

public static void main(String[] args) {

try {

intentos = 0;

Scanner sc = new Scanner(System.in);

Socket cl = new Socket(HOST, PUERTO);

DataOutputStream dos = new DataOutputStream(cl.getOutputStream());

System.out.println("Conexion establecida... Inicio del juego");

System.out.println("Elije una dificultad(1.-Sencillo//2.-Intermedio//3.-Dificil)");

byte dificultad = sc.nextByte();

dos.writeByte(dificultad);

if (dificultad == 2) {

columnas = filas = NIVEL\_INTERMEDIO;

minasDificultad = MINAS\_NIVEL\_INTERMEDIO;

} else if (dificultad == 3) {

filas = NIVEL\_INTERMEDIO;

columnas = NIVEL\_EXPERTO;

minasDificultad = MINAS\_NIVEL\_EXPERTO;

} else {

columnas = filas = NIVEL\_PRINCIPANTE;

minasDificultad = MINAS\_NIVEL\_PRINCIPANTE;

}

Casillas matrizTablero[][] = new Casillas[filas][columnas];

ObjectInputStream ois = new ObjectInputStream(cl.getInputStream());

tablero.clear();

tablero = (List<Casillas>) ois.readObject();

dos.writeUTF("OK");

int z = 0;

for (int i = 0; i < filas; i++) {

for (int j = 0; j < columnas; j++, z++) {

matrizTablero[i][j] = tablero.get(z);

}

}

//Tomar tiempo inicial

Calendar calendario=Calendar.getInstance();

min=calendario.get(Calendar.MINUTE);

seg=calendario.get(Calendar.SECOND);

hora = calendario.get(Calendar.HOUR\_OF\_DAY);

tiempoI=(hora\*60)+min+(seg/60);

System.out.println("tiempo: "+tiempoI);

imprimir(matrizTablero);

while (Ganador) {

System.out.println("Elije una opcion: 1.-Casilla//2.-Bandera");

if (sc.nextByte() == 1) {

System.out.println("Posicion del tiro");

System.out.println("Fila: ");

f = sc.nextByte();

System.out.println("Columna: ");

c = sc.nextByte();

Ganador = seleccionaMina(matrizTablero, f, c, false);

} else {

System.out.println("Posicion de la bandera: ");

System.out.println("Fila: ");

f = sc.nextByte();

System.out.println("Columna: ");

c = sc.nextByte();

Ganador = seleccionaMina(matrizTablero, f, c, true);

}

imprimir(matrizTablero);

if (Ganador == false) {

System.out.println("Perdiste");

imprimirTodo(matrizTablero);

dos.writeUTF("LOST");

break;

}

if (intentos == minasDificultad) {

System.out.println("Ganaste");

dos.writeUTF("WIN");

Calendar c = Calendar.getInstance();

min=c.get(Calendar.MINUTE);

seg=c.get(Calendar.SECOND);

hora = c.get(Calendar.HOUR\_OF\_DAY);

tiempoF= (hora\*60)+min+(seg/60);

float reg = Math.round(((tiempoF-tiempoI)\*100)/100);

System.out.println("tiempo: "+tiempoF);

dos.flush();

dos.writeUTF(Float.toString(reg));

dos.flush();

break;

}

System.out.println("intentos : "+intentos);

}

dos.close();

ois.close();

cl.close();

} catch (IOException | ClassNotFoundException e) {

e.toString();

}

}

public static void imprimirTodo(Casillas[][] matrizTablero) {

for (int i=0;i<columnas-1;i++){

System.out.print("C"+i+" ");

}

System.out.println("");

for (int i = 0; i < filas - 1; i++) {

for (int j = 0; j < columnas - 1; j++) {

Casillas casilla = matrizTablero[i][j];

if (casilla.getIsMina() == false) {

System.out.print(" " + casilla.getMinasAlrededor() + " ");

} else {

System.out.print(" \* ");

}

}

System.out.println("...F"+i);

}

System.out.println("");

}

public static void imprimir(Casillas[][] matrizTablero) {

int i = 0;

int j = 0;

for(int k=0;k<columnas-1;k++){

System.out.print("C"+k+" ");

}

System.out.println("");

for (i = 0; i < filas - 1; i++) {

for (j = 0; j < columnas - 1; j++) {

Casillas casilla = matrizTablero[i][j];

if (casilla.getIsShown() == (byte) 0) {

System.out.print(" - ");

} else if (casilla.getIsShown() == (byte) 1) {

System.out.print(" " + casilla.getMinasAlrededor() + " ");

} else {

System.out.print("<1>");

}

}

System.out.println("...F"+i);

}

System.out.println("");

}

public static boolean seleccionaMina(Casillas[][] matrizTablero, byte fila, byte columna, boolean isBandera) {

try{

Casillas seleccion = matrizTablero[fila][columna];

if (!isBandera) {

//es mina

if (seleccion.getIsMina()) {

return false;

}

//destapar

seleccion.setIsShown((byte) 1);

//Si la mina es cero avisa a sus hermanos ceros

if (seleccion.getMinasAlrededor() == 0) {

avisarCeros(seleccion, matrizTablero);

}

} else {

//Poner bandera

seleccion.setIsShown((byte) 2);

if(seleccion.getIsMina()){

intentos++;

}

}

return true;

}catch(IndexOutOfBoundsException | InputMismatchException e){

System.out.println("Coordenadas incorrectas");

return true;

}

}

public static void avisarCeros(Casillas casillaActual, Casillas[][] tablero) {

byte i\_aux = casillaActual.getFila();

byte j\_aux = casillaActual.getColumna();

//Se puso una mina, así que avisamos a sus vecinos

if (casillaActual.getTipo() == 1) {

if (casillaActual.getFila() == 0) {

if (casillaActual.getColumna() == 0) { //superior izquierda

destaparCero(tablero[0][1], tablero);

destaparCero(tablero[1][0], tablero);

destaparCero(tablero[1][1], tablero);

} else if (casillaActual.getColumna() == (columnas - 1)) { //superior derecha

destaparCero(tablero[0][(j\_aux) - 2], tablero);

destaparCero(tablero[1][(j\_aux) - 1], tablero);

destaparCero(tablero[1][(j\_aux) - 2], tablero);

}

} else { //casillaActual.getFila()==(filas-1)

if (casillaActual.getColumna() == 0) { //inferior izquierda

destaparCero(tablero[(i\_aux) - 1][1], tablero);

destaparCero(tablero[(i\_aux) - 2][0], tablero);

destaparCero(tablero[(i\_aux) - 2][1], tablero);

} else {//inferior derecha

destaparCero(tablero[(i\_aux) - 1][(j\_aux) - 2], tablero);

destaparCero(tablero[(i\_aux) - 1][(j\_aux) - 2], tablero);

destaparCero(tablero[(i\_aux) - 2][(j\_aux) - 1], tablero);

}

}

} else if (casillaActual.getTipo() == 2) {

if (i\_aux == 0) {//superiores

destaparCero(tablero[0][(j\_aux) - 1], tablero);

destaparCero(tablero[0][(j\_aux) + 1], tablero);

destaparCero(tablero[1][(j\_aux) - 1], tablero);

destaparCero(tablero[1][(j\_aux) + 1], tablero);

destaparCero(tablero[1][j\_aux], tablero);

} else if (i\_aux == (filas - 1)) {//inferiores

destaparCero(tablero[(filas - 1)][(j\_aux) - 1], tablero);

destaparCero(tablero[(filas - 1)][(j\_aux) + 1], tablero);

destaparCero(tablero[(filas - 2)][(j\_aux) - 1], tablero);

destaparCero(tablero[(filas - 2)][(j\_aux) + 1], tablero);

destaparCero(tablero[(filas - 2)][(j\_aux)], tablero);

} else if (j\_aux == 0) {//izquierda

destaparCero(tablero[(i\_aux) - 1][0], tablero);

destaparCero(tablero[(i\_aux) + 1][0], tablero);

destaparCero(tablero[(i\_aux) - 1][1], tablero);

destaparCero(tablero[(i\_aux) + 1][1], tablero);

destaparCero(tablero[i\_aux][1], tablero);

} else if (j\_aux == (columnas - 1)) {//derecha

destaparCero(tablero[(i\_aux) - 1][j\_aux], tablero);

destaparCero(tablero[(i\_aux) + 1][j\_aux], tablero);

destaparCero(tablero[(i\_aux) - 1][(j\_aux) - 1], tablero);

destaparCero(tablero[(i\_aux) + 1][(j\_aux) - 1], tablero);

destaparCero(tablero[i\_aux][(j\_aux) - 1], tablero);

}

} else {

destaparCero(tablero[i\_aux][(j\_aux) - 1], tablero);

destaparCero(tablero[i\_aux][(j\_aux) + 1], tablero);

destaparCero(tablero[(i\_aux) - 1][(j\_aux) - 1], tablero);

destaparCero(tablero[(i\_aux) - 1][j\_aux], tablero);

destaparCero(tablero[(i\_aux) - 1][(j\_aux) + 1], tablero);

destaparCero(tablero[(i\_aux) + 1][(j\_aux) - 1], tablero);

destaparCero(tablero[(i\_aux) + 1][j\_aux], tablero);

destaparCero(tablero[(i\_aux) + 1][(j\_aux) + 1], tablero);

}

}

public static void destaparCero(Casillas casilla, Casillas conteo[][]) {

if(casilla.getIsShown()!=0){

return;

}

if (casilla.getMinasAlrededor() == 0) {

casilla.setIsShown((byte) 1);

avisarCeros(casilla, conteo);

}

else{

if(!casilla.getIsMina()){

casilla.setIsShown((byte)1);

//avisarCeros(casilla,conteo);

}

}

}

}

Servidor.java

package buscaminas;

import java.net.\*;

import java.io.\*;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

/\*\*

\*

\* @author saidn

\*/

public class Servidor {

public static final int PUERTO=5000;

public static final byte NIVEL\_PRINCIPANTE = 9;

public static final byte NIVEL\_INTERMEDIO = 16;

public static final byte NIVEL\_EXPERTO = 30;

public static final byte MINAS\_NIVEL\_PRINCIPANTE = 3;

public static final byte MINAS\_NIVEL\_INTERMEDIO = 40;

public static final byte MINAS\_NIVEL\_EXPERTO = 99;

/\*public static final byte tipoEsquina = 1;

public static final byte tipoBorde = 2;

public static final byte tipoInterna = 3;\*/

public static final List<Casillas> tablero = new ArrayList<>();

public static byte columnas = 0;

public static byte filas = 0;

public static byte numeroMinas = 0;

public static void main(String[] args) {

try{

byte dificultad;

ServerSocket s;

s = new ServerSocket(PUERTO);

s.setReuseAddress(true);

System.out.println("Servidor iniciado...............");

for(;;){

System.out.println("Esperando jugador...");

Socket cl = s.accept();

//Configuracion juego

DataInputStream dis = new DataInputStream(cl.getInputStream());

DataOutputStream dos = new DataOutputStream(new FileOutputStream("Registro.txt"));

dificultad = dis.readByte();

if(dificultad==2){

columnas = filas = NIVEL\_INTERMEDIO;

numeroMinas = MINAS\_NIVEL\_INTERMEDIO;

}else if(dificultad==3){

filas = NIVEL\_INTERMEDIO;

columnas = NIVEL\_EXPERTO;

numeroMinas = MINAS\_NIVEL\_EXPERTO;

}else{

columnas = filas = NIVEL\_PRINCIPANTE;

numeroMinas = MINAS\_NIVEL\_PRINCIPANTE;

}

//Creación de un tablero para el usuario

tablero.clear();

crearTablero();

//Envío de una tablero

ObjectOutputStream oos = new ObjectOutputStream(cl.getOutputStream());

oos.writeObject(tablero);

oos.flush();

//Inicio temporizador

String tempo = dis.readUTF();

if(tempo.equals("OK")){

Date date = new Date();

DateFormat inicio = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");

System.out.println(inicio.format(date));

String marcador = dis.readUTF();

if(marcador.equals("WIN")){

String reg = dis.readUTF();

dos.writeUTF(reg);

System.out.println("Ganó");

}else{

System.out.println("Perdió");

}

date = new Date();

DateFormat fin = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");

System.out.println(fin.format(date));

}

dis.close();

oos.close();

cl.close();

System.out.println("\n");

}

}catch(Exception e){

e.toString();

}

}

private static void crearTablero(){

Casillas[][]conteo = new Casillas[filas][columnas];

for(byte i = 0; i < filas;i++){

for(byte j = 0; j < columnas; j++){

if(i == 0 || i==(filas-1)){//Tipo esquina

if(j == 0 || j==(columnas-1)){ //Tipo esquina

Casillas aux = new Casillas(i,j,(byte)1);

tablero.add(aux);

conteo[i][j] = aux;

}else{ //Tipo borde

Casillas aux = new Casillas(i,j,(byte)2);

tablero.add(aux);

conteo[i][j] = aux;

}

}else{

if(j==0||j==(columnas-1)){ //Tipo Borde

Casillas aux = new Casillas(i,j,(byte)2);

tablero.add(aux);

conteo[i][j] = aux;

}else{ // Tipo interna

Casillas aux = new Casillas(i,j,(byte)3);

tablero.add(aux);

conteo[i][j] = aux;

}

}

} //for

}//for

//LLenado de minas

byte minasPuestas = 0;

double ponerMina;

for(int z = 0; z<tablero.size(); z=z+2 ){

Casillas casillaActual = tablero.get(z);

if(minasPuestas < numeroMinas){ //aun faltan minas

if(z == tablero.size()-1){ //faltan pero recorri el tableo

z = 0;

}

if(!casillaActual.getIsMina()){

//Estoy en una casilla sin mina

ponerMina = Math.random();

if(ponerMina<0.3){ //Random

casillaActual.setIsMina(true);

minasPuestas++;

byte i\_aux = casillaActual.getFila();

byte j\_aux = casillaActual.getColumna();

//Se puso una mina, así que avisamos a sus vecinos

if(casillaActual.getTipo() == 1){

if(casillaActual.getFila()==0){

if(casillaActual.getColumna()==0){ //superior izquierda

conteo[0][1].hasMinaVecina();

conteo[1][0].hasMinaVecina();

conteo[1][1].hasMinaVecina();

}else if(casillaActual.getColumna()==(columnas-1)){ //superior derecha

conteo[0][(j\_aux)-2].hasMinaVecina();

conteo[1][(j\_aux)-1].hasMinaVecina();

conteo[1][(j\_aux)-2].hasMinaVecina();

}

}else{

if(casillaActual.getColumna()==0){ //inferior izquierda

conteo[(i\_aux)-1][1].hasMinaVecina();

conteo[(i\_aux)-2][0].hasMinaVecina();

conteo[(i\_aux)-2][1].hasMinaVecina();

}else{//inferior derecha

conteo[(i\_aux)-1][(j\_aux)-2].hasMinaVecina();

conteo[(i\_aux)-1][(j\_aux)-2].hasMinaVecina();

conteo[(i\_aux)-2][(j\_aux)-1].hasMinaVecina();

}

}

}else if(casillaActual.getTipo() == 2){

if(i\_aux==0){//superiores

conteo[0][(j\_aux)-1].hasMinaVecina();

conteo[0][(j\_aux)+1].hasMinaVecina();

conteo[1][(j\_aux)-1].hasMinaVecina();

conteo[1][(j\_aux)+1].hasMinaVecina();

conteo[1][j\_aux].hasMinaVecina();

}else if(i\_aux==(filas-1)){//inferiores

conteo[(filas-1)][(j\_aux)-1].hasMinaVecina();

conteo[(filas-1)][(j\_aux)+1].hasMinaVecina();

conteo[(filas-2)][(j\_aux)-1].hasMinaVecina();

conteo[(filas-2)][(j\_aux)+1].hasMinaVecina();

conteo[(filas-2)][(j\_aux)].hasMinaVecina();

}else if(j\_aux==0){//izquierda

conteo[(i\_aux)-1][0].hasMinaVecina();

conteo[(i\_aux)+1][0].hasMinaVecina();

conteo[(i\_aux)-1][1].hasMinaVecina();

conteo[(i\_aux)+1][1].hasMinaVecina();

conteo[i\_aux][1].hasMinaVecina();

}else if(j\_aux==(columnas-1)){//derecha

conteo[(i\_aux)-1][j\_aux].hasMinaVecina();

conteo[(i\_aux)+1][j\_aux].hasMinaVecina();

conteo[(i\_aux)-1][(j\_aux)-1].hasMinaVecina();

conteo[(i\_aux)+1][(j\_aux)-1].hasMinaVecina();

conteo[i\_aux][(j\_aux)-1].hasMinaVecina();

}

}else{

conteo[i\_aux][(j\_aux)-1].hasMinaVecina();

conteo[i\_aux][(j\_aux)+1].hasMinaVecina();

conteo[(i\_aux)-1][(j\_aux)-1].hasMinaVecina();

conteo[(i\_aux)-1][j\_aux].hasMinaVecina();

conteo[(i\_aux)-1][(j\_aux)+1].hasMinaVecina();

conteo[(i\_aux)+1][(j\_aux)-1].hasMinaVecina();

conteo[(i\_aux)+1][j\_aux].hasMinaVecina();

conteo[(i\_aux)+1][(j\_aux)+1].hasMinaVecina();

}

}//mina puesta

}

}

}//for minas

//mostrando tablero

for(int i=0;i<columnas-1;i++){

System.out.print("C"+i+" ");

}

System.out.println("");

for(int i=0; i<filas-1;i++){

for(int j = 0; j < columnas-1; j++){

Casillas casilla = conteo[i][j];

if(casilla.getIsMina()==false){

System.out.print(" "+casilla.getMinasAlrededor()+" ");

}else{

System.out.print(" \* ");

}

}

System.out.println(".."+i);

}

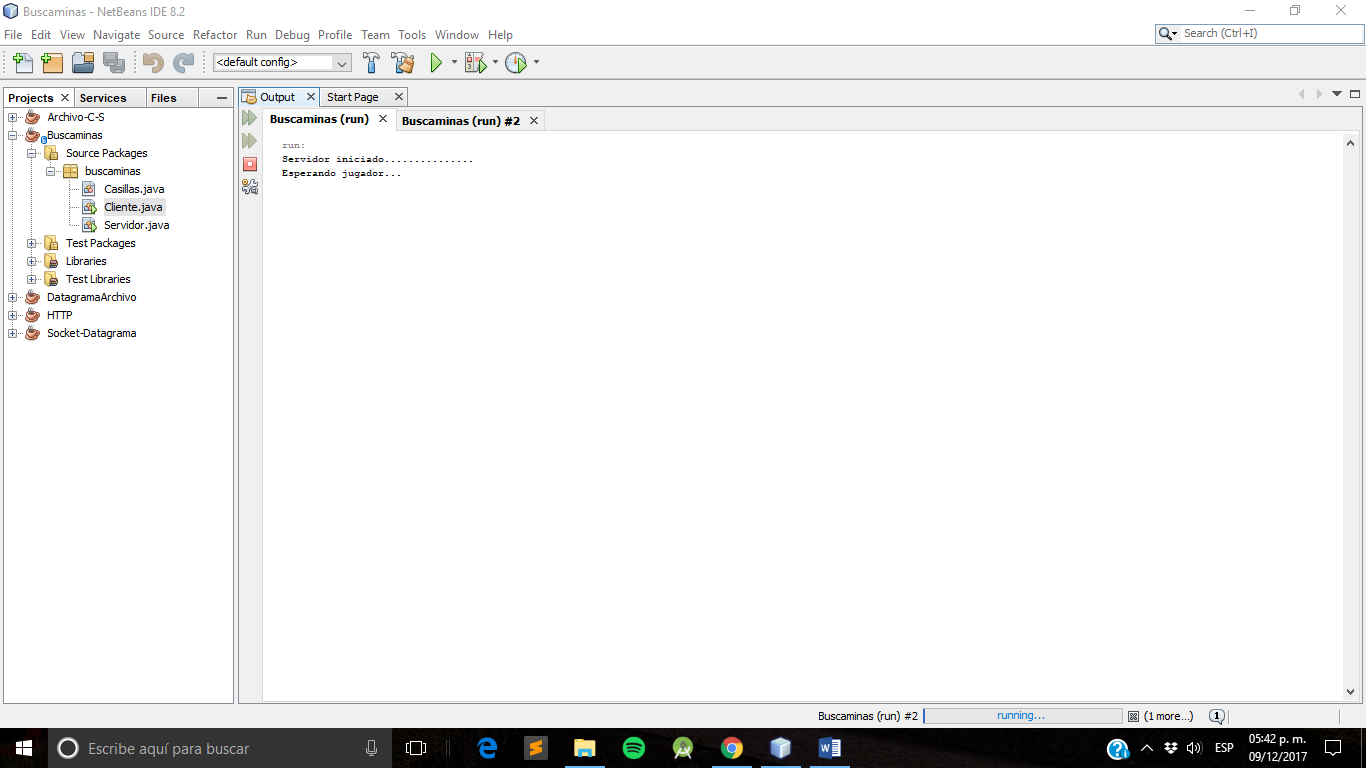
System.out.println("");

}

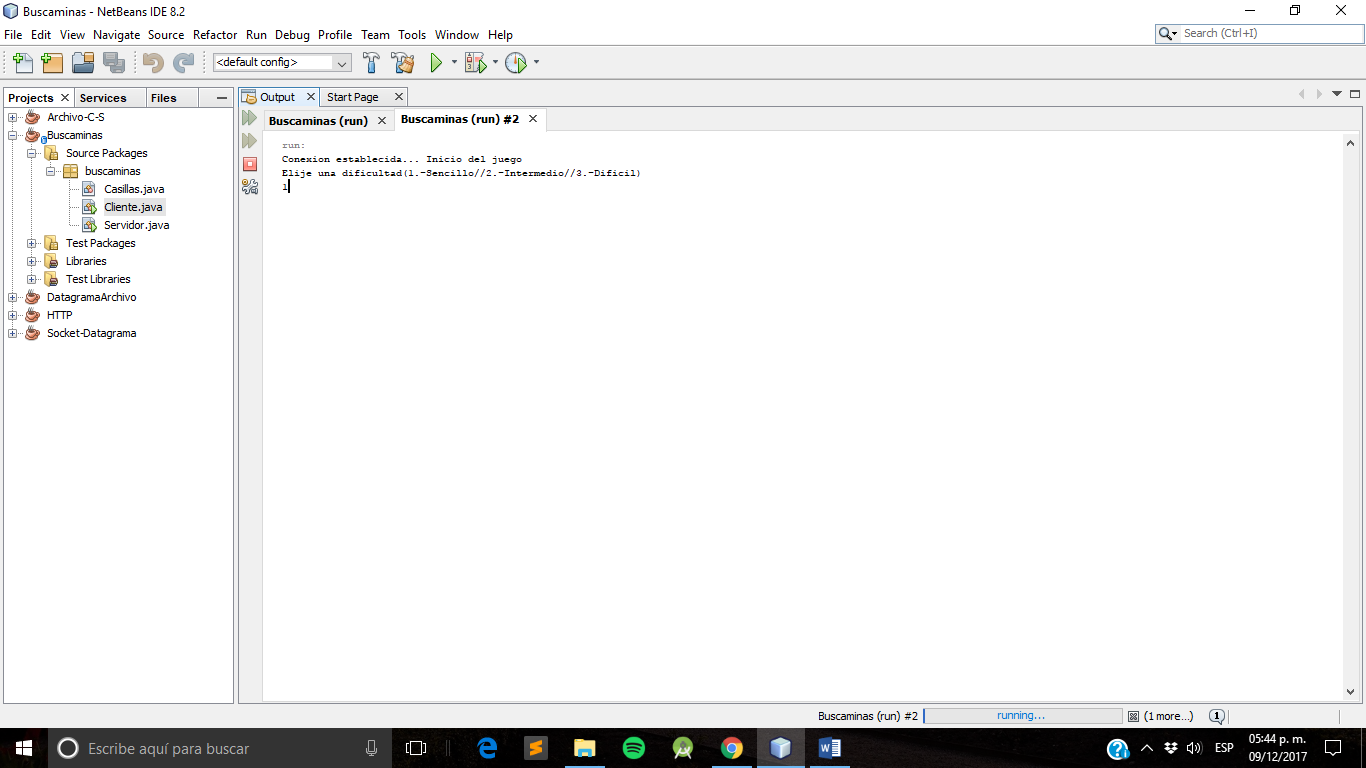
}

**Desarrollo**

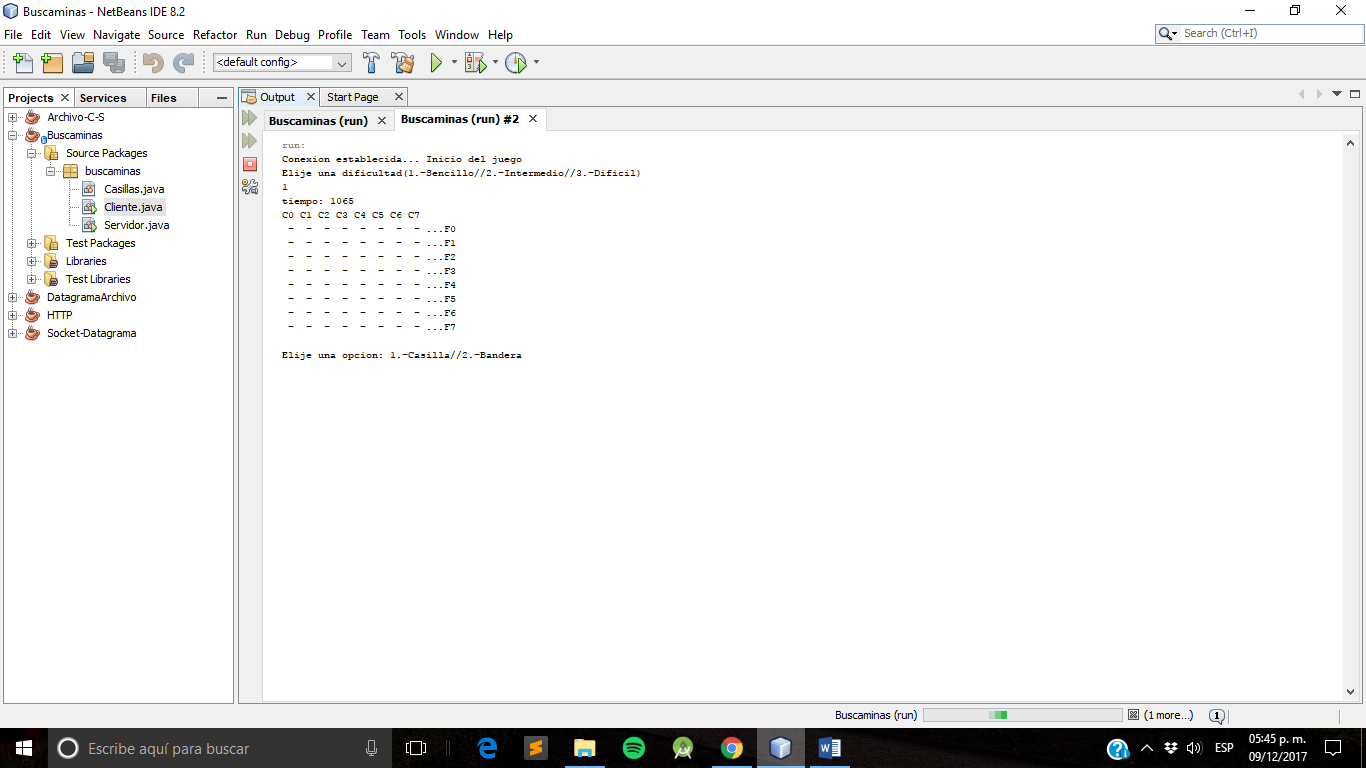
Para el desarrollo de la practica primero se desarrolló el servidor, el cual espera la conexión del cliente para enviarle el tablero sobre el cual se jugara.



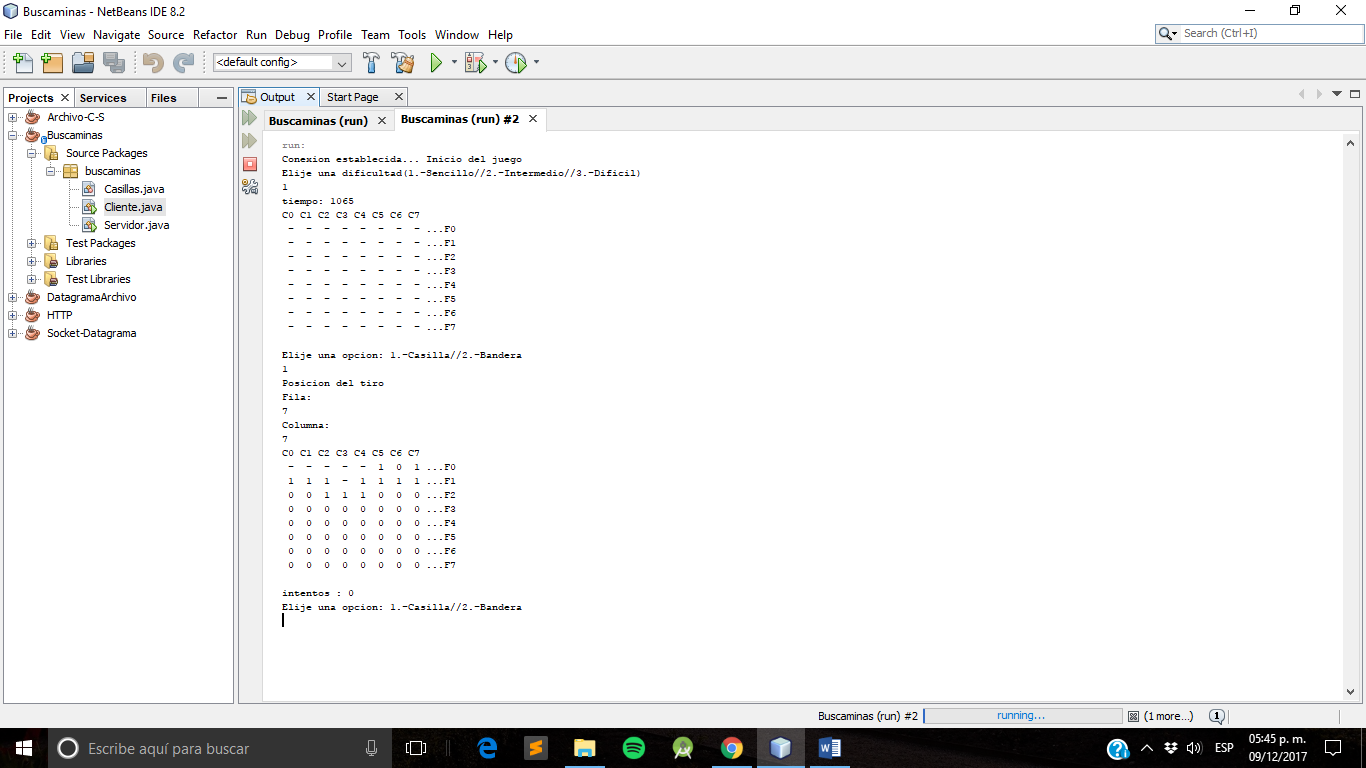
Posteriormente se corre el cliente y nos pide que ingresemos la dificultad del juego para pedir el tipo de tablero a jugar. En este caso solicitaremos un juego con un tablero sencillo.



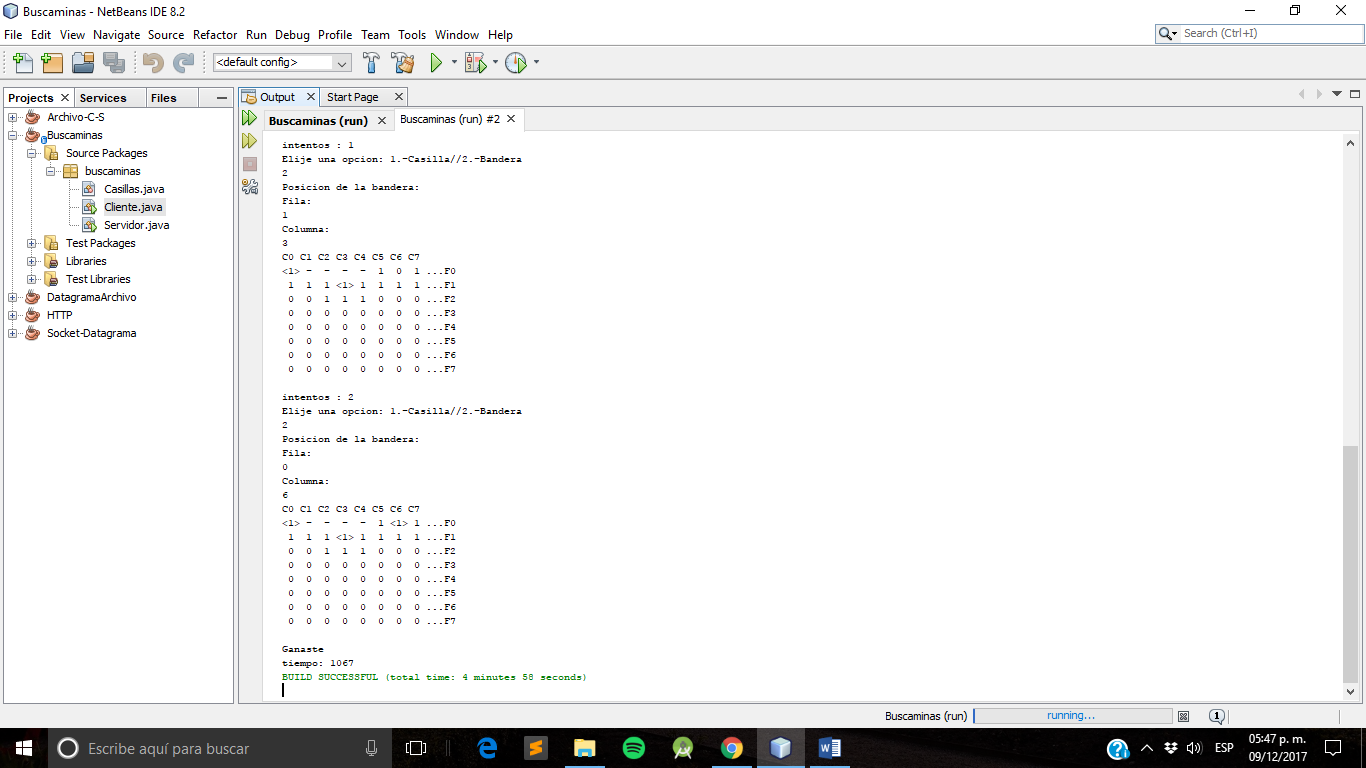
El tablero que obtenemos es el siguiente



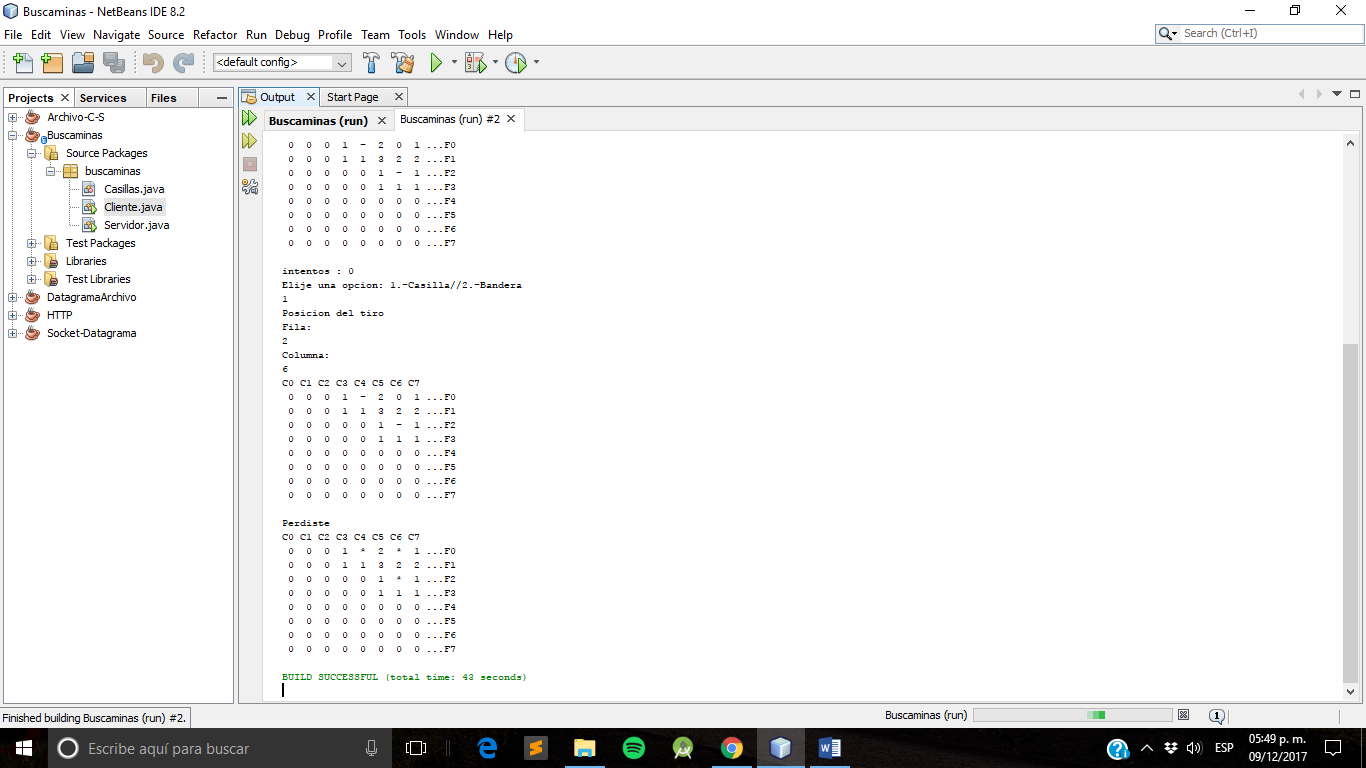
Despues elegimos si pondremos una bandera o si elegiremos desbloquear una casilla, tras el tiro obtendremos lo siguiente:



El juego terminara cuando todas las minas tengan una bandera.

Así al final nos indica el tiempo que se tardó en ejecución y que eres el ganador

En el caso de que se perdiera nos indicaría tal como se muestra en la imagen siguiente.



**Conclusión**

Con la ayuda de esta práctica pudimos observar el funcionamiento del envió de objetos por medio de sockets de flujo.

La forma en la cual se hace el envió de un objeto tablero es muy interesante debido a que nos hace crear una conclusión de que podemos enviar cualquier tipo de cosas por medio de sockets de flujo y ya no solo un archivo sino también los atributos que este tiene.

La práctica fue muy interesante porque nos hizo que nos diéramos cuenta de la importancia de los sockets al momento de hacer envíos.