

Centro de Ciencias Básicas

Ingeniería en Sistemas Computacionales

**Documentación del Proyecto Final: Buscaminas**

Pamela Carolina Rodríguez Armas

Abel Soto Valdez

Andrés David Vásquez Serna

1 “E”

Lógica de Programación

L.I. Clelia Ivette Ruiz Vértiz

**Fecha:**

06/Diciembre/2019

**ÍNDICE**

[1. Análisis](#_49cekip0e2xq) 3

[2. Diseño](#_sk6nmuegg0xb) 5

[3. Verificación del algoritmo](#_2klyvj6tbhje) 5

[4. Tabla de variables y constantes](#_ol4yfbxz86ou) 5

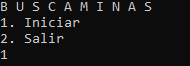
[5. Codificación](#_l899h33lnnr5) 8

[6. Conclusiones](#_hpg8i75bzuxs) 17

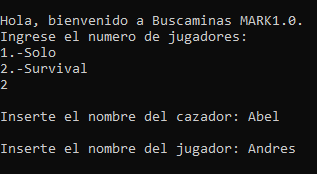
# 1. Análisis

Datos de Entrada y de Salida

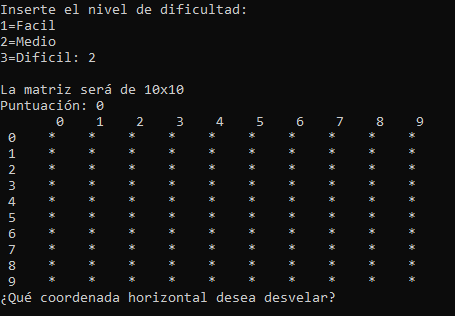
Para la realización del presente proyecto se contempla el manejo de una interfaz a partir de la cual el usuario ha de decirnos, a partir de la impresión de un menú, el si decide proseguir o cerrar la partida.



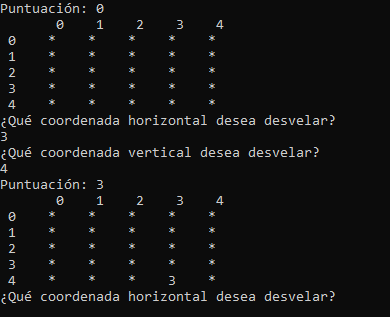
Posterior a ello, de la misma manera este ha de decidir la modalidad de su preferencia, que en cuyo caso, habrá de brindarnos el nombre de uno o más jugadores.



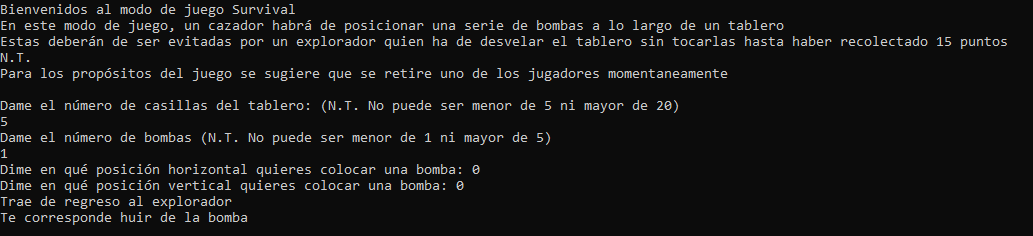
En lo que refiere a la modalidad individual, se le pregunta al usuario acerca de la dificultad de juego deseada, con lo cual se define el tamaño del tablero a imprimir.



Una vez efectuada tal selección, a razón de las coordenadas que nos brinde el usuario, se le habrán de desvelar las casillas correspondientes en la impresión ulterior.

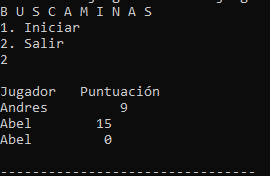


Con respecto de la modalidad en que se dispone la necesidad de dos jugadores, se les imprime las instrucciones concernientes a cómo efectuar el juego para que este resulte lo más dinámico posible, para posteriormente solicitarle a uno de los usuarios el que introduzca la cantidad de bombas y casillas a emplear.



Tras ello, la interfaz acaba por ser semejante a la del modo individual, con la diferencia de que la dificultad es definida por un usuario externo.

Finalmente, tras concluir sus respectivas partidas, al seleccionar la salida del programa, se imprimen los puntajes ordenados de mayor a menor.

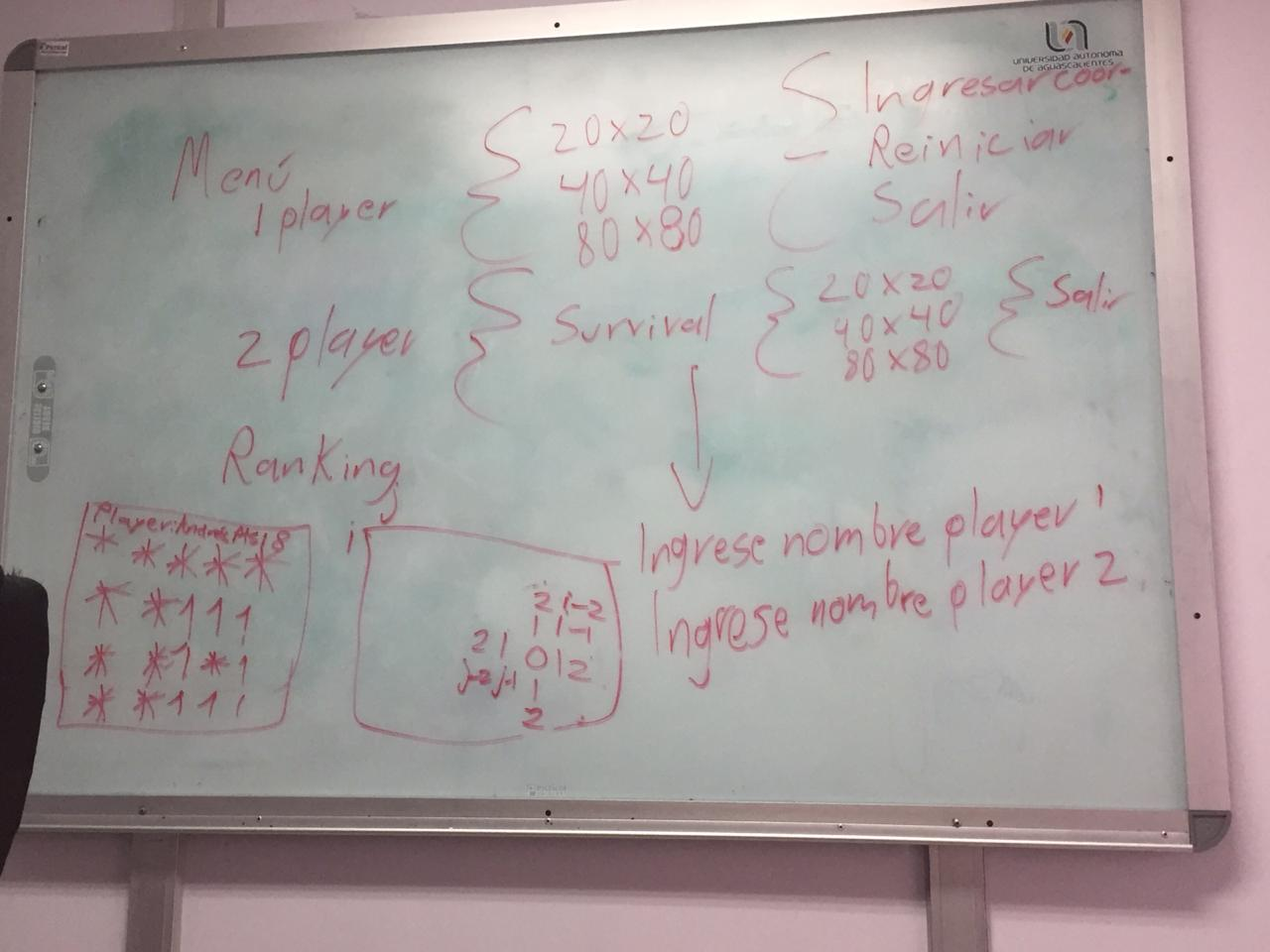


# 2. Diseño

Proceso y Construcción del Algoritmo

El proceso lógico o algorítmico a partir del cual se dio la elaboración del presente programa consiste a partir del manejo de matrices internas con las cuales se han de correlacionar los valores impresos en pantalla con respecto de las posiciones en el array dadas por el usuario.

El llenado de las matrices internas se da a partir de números aleatorios con los que se ha de definir el posicionamiento de las bombas, a partir de ello es que se dan los indicadores de posicionamiento, haciendo alusión a la cercanía de una bomba con un condicional.



# 3. Verificación del algoritmo

Es a partir de la traducción del algoritmo en lo que respecta de la sintaxis del lenguaje C, y la puesta en marcha del mismo en un entorno de desarrollo que nos permite correr el mismo que podemos corroborar el funcionamiento óptimo de la lógica algorítmica planteada por los integrantes del equipo.

# 4. Tabla de variables y constantes

A continuación se presentan las variables principales que llegaron a ser empleadas dentro del juego en relación a la lógica establecida por los integrantes del equipo.

|  |  |  |
| --- | --- | --- |
| **Tipo de Variable** | **Nombre de la variable** | **Uso** |
| **int** | numcas | Variable global con la que se determina el tamaño de la matriz |
| **int** | cont | Contador con el que se cuenta el número de partidas del usuario |
| **int** | numbom | Variable global en que se almacenan la cantidad de bombas en modo survival |
| **int** | mat[20][20] | Matriz interna del juego en que se almacenan los valores ocultos |
| **int** | punt | Variable en la que se suma la puntuación del jugador |
| **int** | controlador | Esta variable determina si el juego prosigue o concluye |
| **int** | dif | Variable que determina la dificultad del juego |
| **int** | nplayers | Varibale con que se determina la cantidad de jugadores |
| **char** | player1[30] | Variable en que se almacena el nombre de los jugador 1 |
| **char** | player2[30] | Variable en que se almacena el nombre de los jugador 2, en caso de haberlo |
| **int** | contpartju | Variable que cuenta cuántas partidas han sido empezadas |
| **struct** | Jugadores js[100] | Estructura en que se almacenan los nombres y puntuaciones de todos los jugadores |
| **char** | nombres[30] | Variable dentro del struct en que se almacenan los nombres de los jugadores pasados |
| **int** | punta | Variable dentro del struct en que se almacenan los puntajes anteriores |

# 5. Codificación

Programa en C

#include<stdlib.h>

#include<stdio.h>

#include<string.h>

#include<time.h>

#include<iostream>

#include<conio.h>

#include<windows.h>

using namespace std;

void gotoxy (int x, int y){

HANDLE hcon;

hcon=GetStdHandle(STD\_OUTPUT\_HANDLE);

COORD dwPos;

dwPos.X=x;

dwPos.Y=y;

SetConsoleCursorPosition(hcon,dwPos);

}

int dificultad ();

void minas();//Prototipo de función del llenado de las casillas

void juego(int mat[20][20]);//Prototipo de función de la interfaz de juego del modo de un sólo jugador

void puntaje();//función que muestra los puntajes de los jugadores

void minaSurvival();//Prototipo de función del llenado de las casillas

void juegoSurvival(int mat[20][20]);//Prototipo de función de la interfaz de juego del modo de un sólo jugador

int numcas;//Variable global con la que se determina el tamaño de la matriz

int cont=0;//Contador con el que cuenta el número de partidas del usuario;

int numbom;//Variable global en que se almacenan la cantidad de bombas en modo survival

int mat[20][20]; //matriz interna del juego

int punt;//Variable en la que se suma la puntuación del jugador

int controlador;//Esta variable determina si el juego prosigue o concluye

int dif;//Variable que determina la dificultad del juego

int nplayers;//Varibale con que se determina la cantidad de jugadores

char player1[30],player2[30];//Variables en que se almacena el nombre de los jugadores

int contpartju=0;//Variable que cuenta cuántas partidas han sido empezadas

struct jugadores{//Struct en que se almacenan los nombres y puntuaciones de todos los jugadores

char nombres[30];//Vector en que se almacena el nombre de los jugadores

int punta;//Vector en que se almacena el puntaje de los jugadores

};

struct jugadores js[100];

int main (){

int opc=1;

while (opc==1){

printf("B U S C A M I N A S\n1. Iniciar\n2. Salir \n");

scanf("%d",&opc);

if (opc==2){

if(contpartju>=1){

puntaje();//Si sale el jugador tras haber jugado una partida, se muestran los puntajes

}

exit(0);

}

printf("\nHola, bienvenido a Buscaminas MARK1.0.\n");

printf("Ingrese el numero de jugadores: \n1.-Solo\n2.-Survival \n");

scanf("%d",&nplayers);

fflush(stdin);

switch(nplayers){

case 1:{

printf("\nInserte su nombre por favor: ");

gets(player1);

strcpy(js[contpartju].nombres, player1);//Se guarda el nombre de los jugadores en el strcuct

switch(dificultad()){

case 1: {

printf("\nLa matriz ser%c de 5x5\n",160);

numcas=5;

cont=0;

punt=0;

controlador=0;

break;

}

case 2: {

printf("\nLa matriz ser%c de 10x10\n",160);

numcas=10;

cont=0;

punt=0;

controlador=0;

break;

}

case 3: {

printf("\nLa matriz ser%c de 20x20\n",160);

numcas=20;

cont=0;

punt=0;

controlador=0;

break;

}

}

minas();

while(controlador==0){

juego(mat);

}

printf("\nTermin%c el juego\n", 162);

}

break;

case 2:{

printf("\nInserte el nombre del cazador: ");

gets(player1);

printf("\nInserte el nombre del jugador: ");

gets(player2);

strcpy(js[contpartju].nombres, player1);//Se guarda el nombre de los jugadores en el struct

strcpy(js[contpartju+1].nombres, player2);

contpartju++;

cont=0;

punt=0;

controlador=0;

int chec=0;

do{

printf("\nBienvenidos al modo de juego Survival\n");

printf("En este modo de juego, un cazador habr%c de posicionar una serie de bombas a lo largo de un tablero\n", 160);

printf("Estas deber%cn de ser evitadas por un explorador quien ha de desvelar el tablero sin tocarlas hasta haber recolectado 15 puntos\n", 160);

printf("N.T.\nPara los prop%csitos del juego se sugiere que se retire uno de los jugadores momentaneamente", 162);

printf("\n\n");

printf("Dame el n%cmero de casillas del tablero: (N.T. No puede ser menor de 5 ni mayor de 20)\n", 163);

scanf("%d", &numcas);

printf("Dame el n%cmero de bombas (N.T. No puede ser menor de 1 ni mayor de 5)\n", 163);

scanf("%d", &numbom);

if(numcas>=5 && numcas<=20 && numbom>=1 && numbom<=5){

chec=1;

}

}while(chec==0);

minaSurvival();

printf("Trae de regreso al explorador\n");

printf("Te corresponde huir de la bomba\n\n");

while(controlador==0){

juegoSurvival(mat);

}

printf("\nTermin%c el juego\n\n", 162);

}

}//Fin del Switch

contpartju++;

}//Fin del While

system("pause");

return 0;

}

int dificultad (){

printf("\nInserte el nivel de dificultad: \n1=Facil\n2=Medio\n3=Dificil: ");

scanf("%d",&dif);

return dif;

}

void minaSurvival(){//Función para llenado de la matriz modo survival

int posh, posv; //Variables en que se almacena la posición horizontal y vertical de las bombas

//Vaciado de la matriz

for(int i=0;i<20;i++){

for(int j=0;j<20;j++){

mat[i][j]=3;

}

}

for(int i=0;i<numbom;i++){//El cazador coloca las bombas en cada una de las posiciones que él desee

printf("Dime en qu%c posici%cn horizontal quieres colocar una bomba: ", 130, 162);

scanf("%d", & posh);

printf("Dime en qu%c posici%cn vertical quieres colocar una bomba: ", 130, 162);

scanf("%d", & posv);

//Llenado de la matriz interna con bombas

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

for(int j=0;j<numcas;j++){

if(i==posv && j==posh){

mat[posv][posh]=0;//Si la coordenada de la matriz coincide con los números dados por el cazador, se inserta una bomba

}

}

}

}

/\*for(int i=0;i<numcas;i++){

for(int j=0;j<numcas;j++){

printf("%d", mat[i][j]);

}

printf("\n");

}\*/

}

void juegoSurvival(int mat[20][20]){

int cooh, coov, lin[numcas];

char busmat[numcas][numcas], gua;//gua almacena el valor de la matriz interna en char para mostrarlo en la interfaz

//Llenado del vector de las métricas que delimitan las coordenadas

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

lin[i]=i;

}

//Llenado de la matriz que con la que habrá de interactuar el usuario

if(cont==0){//Si es la primera vez que se imprime la matriz se llena, caso contrario se mantiene intacta

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

for(int j=0;j<numcas;j++){

busmat[i][j]='\*';

}

}

}

//Impresión de la interfaz de juego

printf("Puntuaci%cn: %d\n", 162, punt);//Puntuación del jugador

printf(" ");

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

printf("%5d", lin[i]);//impresión de las métricas horizontales

}

printf("\n");

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

if(lin[i]<10){

printf(" %d ", lin[i]);//impresión de las métricas verticales

}

if(lin[i]>=10){

printf(" %d ", lin[i]);//impresión de las métricas verticales

}

for(int j=0;j<numcas;j++){

if(i==coov && j==cooh){//Si la coordenada dada por el usario coincide con una coordenada de la matriz, mostrar el valor de la matriz interna

gua=mat[i][j]+'0';

busmat[i][j]=gua;

}else if(busmat[i][j]!='\*' && (busmat[i][j]=='1' || busmat[i][j]=='2' || busmat[i][j]=='3')){

busmat[i][j];

}else {

busmat[i][j]='\*';

}

printf(" %c ", busmat[i][j]);//impresión de la matriz

}

printf("\n");

}

//Se le solicita al usuario las coordernadas a desvelar de la matriz interna

printf("%cQu%c coordenada horizontal desea desvelar?\n", 168, 130);

scanf("%d", & cooh);

printf("%cQu%c coordenada vertical desea desvelar?\n", 168, 130);

scanf("%d", & coov);

if(cooh>numcas-1 || coov>numcas+1){

printf("Esa coordenada no est%c permitida \n", 160);

cooh=100;

coov=100;

}

punt+=mat[coov][cooh];

if(punt==15){

js[contpartju-1].punta=punt;

printf("Libraste las bombas\n");

printf("Gan%c el Explorador", 162);

controlador=1;

}

if(mat[coov][cooh]==0){//si el jugador se topa con una bomba, termina el juego

controlador=1;

printf("\nLa coordenada que acabas de señalar es una mina\n");

js[contpartju].punta=(punt\*2)+3;

printf("Gan%c el Cazador", 162);

}

cont++;

}

//Funciones para un Jugador

void minas(){//Función para el llenado de la matriz en modo de un sólo jugador

int numra1[5], numra2[5], lin[numcas];

srand(time(NULL));

//Declaración de números aleatorios

for(int i=0;i<5;i++){

numra1[i]=rand()%numcas-1;

numra2[i]=rand()%numcas-1;

/\* PRUEBAS DE IMPRESIÓN

printf("El número aleatorio 1 es: %d\n", numra1[i]);

printf("El número aleatorio 2 es: %d\n", numra2[i]);

\*/

}

//Vaciado de la matriz

for(int i=0;i<20;i++){

for(int j=0;j<20;j++){

mat[i][j]=3;

}

}

//Llenado de la matriz interna con bombas

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

for(int j=0;j<numcas;j++){

for(int x=0;x<5;x++){

if(i==numra1[x] || j==numra2[x]){

mat[numra1[x]][numra2[x]]=0;//Si la coordenada de la matriz coincide con los números aleatorios, se inserta una bomba

}

}

}

}

//Llenado de los localizadores

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

for(int j=0;j<numcas;j++){

if(mat[i][j-1]==0 || mat[i][j+1]==0){

mat[i][j]=1;//si se encuentra al lado izquierdo o derecho de una bomba imprime 1

} else if(mat[i-1][j]==0 || mat[i+1][j]==0){

mat[i][j]=1;//si se encuentra por debajo de una bomba o arriba de esta imprime 1

} else if(mat[i][j-2]==0 || mat[i][j+2]==0){

mat[i][j]=2;//si se encuentra al lado izquierdo o derecho de un 0 con una unidad extra 0 imprime 2

} else if(mat[i-2][j]==0 || mat[i+2][j]==0){

mat[i][j]=2;//si se encuentra por debajo de un 0 o arriba de este con una unidad extraimprime 2

}else if(mat[i][j]!=0 || j==0){

mat[i][j]=3;

}

}

}

}

void juego(int mat[20][20]){//Función de juego para la interfaz de un sólo jugador

int cooh, coov, lin[numcas];

char busmat[numcas][numcas], gua;

//Llenado del vector de las métricas que delimitan las coordenadas

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

lin[i]=i;

}

//Llenado de la matriz que con la que habrá de interactuar el usuario

if(cont==0){//Si es la primera vez que se imprime la matriz se llena, caso contrario se mantiene intacta

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

for(int j=0;j<numcas;j++){

busmat[i][j]='\*';

}

}

}

//Impresión de la interfaz de juego

printf("Puntuaci%cn: %d\n", 162, punt);//Puntuación del jugador

printf(" ");

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

printf("%5d", lin[i]);//impresión de las métricas horizontales

}

printf("\n");

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

if(lin[i]<10){

printf(" %d ", lin[i]);//impresión de las métricas verticales

}

if(lin[i]>=10){

printf(" %d ", lin[i]);//impresión de las métricas verticales

}

for(int j=0;j<numcas;j++){

if(i==coov && j==cooh){//Si la coordenada dada por el usario coincide con una coordenada de la matriz, mostrar el valor de la matriz interna

gua=mat[i][j]+'0';

busmat[i][j]=gua;

}else if(busmat[i][j]!='\*' && (busmat[i][j]=='1' || busmat[i][j]=='2' || busmat[i][j]=='3')){

busmat[i][j];

}else {

busmat[i][j]='\*';

}

printf(" %c ", busmat[i][j]);//impresión de la matriz

}

printf("\n");

}

//Se le solicita al usuario las coordernadas a desvelar de la matriz interna

printf("%cQu%c coordenada horizontal desea desvelar?\n", 168, 130);

scanf("%d", & cooh);

printf("%cQu%c coordenada vertical desea desvelar?\n", 168, 130);

scanf("%d", & coov);

if(cooh>numcas-1 || coov>numcas+1){

printf("Esa coordenada no est%c permitida \n", 160);

cooh=100;

coov=100;

}

punt+=mat[coov][cooh];

if(punt==15){

js[contpartju].punta=punt;

printf("Ganaste el juego");

controlador=1;

}

if(mat[coov][cooh]==0){//si el jugador se topa con una bomba, termina el juego

printf("\nLa coordenada que acabas de señalar es una mina\n");

controlador=1;

}

cont++;

}

void puntaje(){

int aux;

char auxnom[30];

for(int i=0;i<contpartju;i++){//Método de ordenación de la burbuja para colocar a los jugadores por puntaje

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

if(js[i].punta<js[i+1].punta){

aux=js[i].punta;

strcpy(auxnom, js[i].nombres);

js[i].punta=js[i+1].punta;

strcpy(js[i].nombres, js[i+1].nombres);

js[i+1].punta=aux;

strcpy(js[i+1].nombres, auxnom);

}

}

}

gotoxy(70, 0);

printf("\nJugador Puntuaci%cn\n", 162);

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

printf("%3s",js[i].nombres);

printf("%10d", js[i].punta);

printf("\n");

}

}

# 6. Conclusiones

Conclusiones Individuales

***Andrés:***

Trabajar en este programa me ha ayudado bastante en el aprendizaje de mi primer semestre de Lógica, me parece un buen inicio como antecedentes para mi carrera en Programación de Videojuegos. Al trabajar con gente que ya tiene conocimientos previos en esta área, dio un buen alcance a mis conocimientos.

***Pamela:***

Dentro de la realización de este proyecto se aplicaron diversos conocimientos nuevos sobre distintas funcionalidades que nos ofrece la programación tales como el manejo de arreglos, cadenas, funciones, etc. que me ayudaron entender la lógica de manera aplicada a este juego.

***Abel:***

El trabajar en el proyecto me ha resultado por demás grato ya que, en mi caso particular, me ha permitido corroborar mis cualidades como programador en lo que refiere a campos más vastos como lo es el manejo de lenguajes de programación más variados.

Conclusiones Generales

Es a partir de la realización del presente proyecto que podemos, no sólo corroborar, sino incluso aplicar a un campo como lo es el desarrollo de software los conocimientos adquiridos a lo largo del curso de lógica de programación, mismos que a su vez nos han de ser de utilidad para el estudio posterior de la materia, así como en nuestro desempeño en el campo laboral como ingenieros en sistemas, resultando pues el presente en una experiencia por demás fructífera.