

EIE

Escuela de
Ingeniería Eléctrica



**UNIVERSIDAD DE
COSTA RICA**

UNIVERSIDAD DE COSTA RICA

FACULTAD DE INGENIERIA

ESCUELA DE INGENIERIA ELÉCTRICA

PROGRAMACIÓN BAJO PLATAFORMAS ABIERTAS

**LABORATORIO 4:
TRIVIA**

**ESTUDIANTE:
JESÚS ZÚÑIGA MÉNDEZ**

**PROFESOR:
RICARDO ROMÁN BRENES; M. SC.**

I CICLO 2019

Índice

1. Introducción	2
2. Resolución	3
3. Conclusión	13
Referencias	13

Índice de figuras

1. Introducción

Para este laboratorio el objetivo fue crear un programa tipo juego de trivia con el fin de aprender a utilizar punteros, a continuación se adjunta el código escrito en C con el que se soluciono el problema, en el código se encuentra la documentación necesaria para comprender que lógica siguen las funciones presentes en el programa.

```
//Universidad de Costa Rica
//Jesus Zuñiga Mendez
//Carne: B59084
//laboratorio 4: Juego de Trivia
```

3


```

//ciclo que controla que el usuario llene los datos de forma correcta
//se ejecuta siempre que los usuarios sean mayores a 4 y sale de la funcion si
//digita un uno
int Jugadores = 5;
while (Jugadores > 4)
{
    printf(ANSI_COLOR_RESET "Digite el numero de jugadores (maximo 4)\n");
    scanf("%d",&Jugadores);
    //comprobamos que el usuario digite un numero correcto entre 2 y 4
    if (Jugadores > 4)
    {
        printf("\n");
        printf(ANSI_COLOR_RED "EL MAXIMO DE JUGADORES
        ES 4 (0 PARA SALIR)\n"ANSI_COLOR_RESET);
        printf("\n");
    }
    if (Jugadores == 1)
    {
        printf("\n");
        printf(ANSI_COLOR_RED "Este Juego empieza
        a ser divertido cuando juegas con alguien\n
        Ve y busca un amigo (0 PARA SALIR)\n"ANSI_COLOR_RESET);
        Jugadores = 5;
        printf("\n");
    }
}

for(int i = 0; i < Jugadores; i++) {
    system("clear");
    Titulo();
    if (i==0){
        printf(ANSI_COLOR_BLUE "Digite el
        Nombre de jugador 1\n" ANSI_COLOR_RESET);
        scanf("%s", arreglo[i]);
        printf("\n");
    }
    if (i==1){
        printf(ANSI_COLOR_YELLOW "Digite el
        Nombre de jugador 2\n" ANSI_COLOR_RESET);
        scanf("%s", arreglo[i]);
        printf("\n");
    }
    if (i==2){
        printf(ANSI_COLOR_GREEN "Digite el
        Nombre de jugador 3\n" ANSI_COLOR_RESET);
        scanf("%s", arreglo[i]);
        printf("\n");
    }
    if (i==3){
        printf(ANSI_COLOR_MAGENTA "Digite el
        Nombre de jugador 4\n" ANSI_COLOR_RESET);
        scanf("%s", arreglo[i]);
        printf("\n");
    }
}

```

```

    }
    return Jugadores;
}

//*****
//*****
//_____

//funcion que aumenta en uno los puntos de un jugador
// SumarPunto(PPuntos[],0);
void SumarPunto(int arreglo[], int posicion)
{
    //recibimos un puntero como parametro para modificarlo
    int a = arreglo[posicion];
    a++;
    arreglo[posicion] = a;
}

//*****
//*****
//_____

//funcion que extrae una pregunta nueva
//PreguntaNueva(PArregloJugadores, NumeroPregunta);;
int PreguntaNueva(char** Pcontrol, char* arreglo[], int pregunta)
{
    //abrimos el archivo
    FILE* archivo;
    archivo = fopen(Pcontrol[1], "r");
    //declaracion de variables
    int Correcta = 0;
    char saltarLinea[128];
    char saltarLineaVacía[128];
    char caracter;
    //ciclo que se recorre hasta llegar a la pregunta pedida, el ciclo lo que hace es
    saltar bloques de
    //lineas hasta llegar a la pedida
    for (int i=1; i<pregunta; i++){
        fscanf(archivo, "%[^\\n]*c", saltarLinea);
        fscanf(archivo, "%[^\\n]*c", saltarLinea);
        fscanf(archivo, "%[^\\n]*c", saltarLinea);
        fscanf(archivo, "%[^\\n]*c", saltarLinea);
        fscanf(archivo, "%[^\\n]*c", saltarLinea);
        fscanf(archivo, "%[^\\n]*c", saltarLinea);
        caracter = fgetc(archivo);

        //el siguiente ciclo permite ignorar la linea 8, en caso de que tenga
        algun tipo de dato innecesario
        //evita que el programa se caiga
        char salto = '\\n';
        int contador = 1;
        while (contador > 0){

```

```

        if (caracter == salto){
            contador=0;
        }
        else{
            caracter = fgetc(archivo);
            contador=1;
        }
    }
}
fscanf(archivo, "%[^\\n]*c", arreglo[0]);
fscanf(archivo, "%[^\\n]*c", arreglo[1]);
fscanf(archivo, "%[^\\n]*c", arreglo[2]);
fscanf(archivo, "%[^\\n]*c", arreglo[3]);
fscanf(archivo, "%[^\\n]*c", arreglo[4]);
fscanf(archivo, "%d", &Correcta);
fclose(archivo);
return Correcta;
}

//*****
//*****
//_____

//funcion que devuelve un numero random que sera el jugador inicial
//codigo tomado de https://www.tutorialspoint.com/c_standard_library/
c_function_rand.htm
int JugadorRandom(int limite)
{
    int resultado;
    time_t t;
    /* Intializes random number generator */
    srand((unsigned) time(&t));
    resultado = (rand() % limite);
    return resultado;
}

//*****
//*****
//_____

//funcion que controla el formato del archivo, devuelve el numero de lineas
int ComprobarArchivo(char** Pcontrol)
{
    //se abre el archivo
    FILE* archivo;
    archivo = fopen(Pcontrol[1], "r");
    //numeroLinea sera el numero de linea
    int numeroLinea = 0;
    //caracter que se va a leer
    char caracter;
    //contador que controla el ciclo
    int contador = 0;
    //sera 1 si hay dos lineas vacias seguidas
    int error = 0;

```



```

//controla el caracter que se ah leído, 0 si es salto de linea, 1 si es caracter
int esCaracter = 0;
//sirve para comprobar el orden de lo leído
int caracterAnterior = 0;
//variable para controlar el
char salto = '\n';
//se ejecuta hasta llegar al final del archivo
while (contador < 1) {
    caracterAnterior = esCaracter;
    caracter = fgetc(archivo);
    //verificamos que lo leído sea un salto de linea o un caracter
    //si es salto de linea evaluemos si esta en un lugar correcto, sino pasamos
    if (caracter == salto){
        //si lo leído anteriormente es un caracter no ha problema ya que no hay
        //doble salto de linea
        if (caracterAnterior == 1){
            esCaracter = 0;
            numeroLinea++;
        }
        else{
            //leemos el siguiente caracter para verificar que no se trate de
            //otro salto de linea
            caracterAnterior = esCaracter;
            caracter = fgetc(archivo);
            //error si no hay al menos una pregunta
            if (numeroLinea < 6){
                error = 100;
            }
            //error si hay doble salto de linea
            if (caracter == salto){
                error = 200;
                esCaracter=0;
            }else{
                esCaracter=1;
                numeroLinea++;
            }
            //error si el salto de linea se encuentra entre preguntas
            float modulo = (numeroLinea % 7);
            if (modulo != 0){
                error = 300;
            }
        }
    }
    else{
        esCaracter=1;
    }
    if((caracterAnterior == 0) && (caracter == -1)){
        error = 500;
    }
    if(caracter == -1)
    {
        contador = 1;
        //printf("caracterAnterior: %i, caracter: %i\n",caracterAnterior,caracter);
    }
}

```

```

//se ajustan las lineas para que sean multiplo de 7
numeroLinea++;
numeroLinea++;
int modulo = (numeroLinea % 7);
int salir = 1;
if (modulo !=0){
    error = 400;
}
if (error != 0){
    numeroLinea = 0;
    printf(ANSI_COLOR_RED "Error: %d\n"ANSI_COLOR_RESET,error);
}

fclose(archivo);
return numeroLinea;
}

//*****
//*****
//_____

//funciones que declaran el ganador de la partida
void DeclararGanador(char* arregloJugadores[], int arregloPuntos[])
{
    printf(ANSI_COLOR_RESET);
    //int puntajes[4] = {10,11,10,10};
    int puntajes[4] = {arregloPuntos[0],arregloPuntos[1],
arregloPuntos[2],arregloPuntos[3]};
    int empate = 0;
    int puntajeEmpate = 0;
    int mayor = puntajes[0];
    int puntajeA = 0;
    int puntajeB = 0;
    int imprimirGanador = 0;
    //sacamos el mayor puntaje de los cuatro y definimos si hay empates
    for (int i = 0; i < 4; i++){
        puntajeA = puntajes[i];
        for (int j = 0; j < 4; j++){
            puntajeB = puntajes[j];
            if ((puntajeA > puntajeB) && (puntajeA > mayor)){
                mayor = puntajeA;
            }
            if ((puntajeA == puntajeB) && (i != j) && (puntajeA != 0)){
                puntajeEmpate = puntajeA;
                empate++;
            }
        }
    }

    //se imprime el ganador
    if (mayor > puntajeEmpate){
        system("clear");
    }
}

```

```

        Ganador();
        for (int i=0; i < 4;i++){
            puntajeA=puntajes[i];
            if (puntajeA == mayor){
                printf("El Ganador es: %s\nPuntos finales: %d\nFelicidades\n",
                    arregloJugadores[i],mayor);
            }
        }
    }
    //se imprime los empates
    if (mayor == puntajeEmpate){
        system("clear");
        Empate();
        printf("Hay un Empate con %d puntos:\n",mayor);
        for (int i=0; i < 4;i++){
            puntajeA=puntajes[i];
            if (puntajeA == mayor){
                printf("Jugador %d : %s\n" , (i+1) ,arregloJugadores[i]);
            }
        }
    }
}

//*****
//*****
//_____

```

```

// argc: la cantidad de argumentos enviados por la CLI
// argv: los valores de los argument os enviados por la CLI
int main(int argc, char** argv) {
    //se imprime la bienvenida
    system("clear");
    Bienvenida();
    //comprobamos el archivo que venga en formato correcto y
    obtenemos el numero de lineas
    int lineas = ComprobarArchivo(argv);
    int resultado = 0;
    //este if comprueba que el archivo de las preguntas este con formato
    correcto sino imprimimos
    un error.
    if (lineas == 0){
        printf(ANSI_COLOR_RED "ERROR EN EL ARCHIVO DE PREGUNTAS COMPRUEBE QUE
        ESTÉ EN EL FORMATO CORRECTO\nLeer el archivo README
        para mayor informacion\n" ANSI_COLOR_RESET);
    }else if (lineas <= 28){
        //se imprime la alerta de archivo corto
        printf(ANSI_COLOR_RED "El Archivo solo contiene 4 preguntas,
        se recomiendan que tenga más preguntas\n(0 para Salir y cargar otro
        archivo o cualquier tecla para continuar)\n" ANSI_COLOR_RESET);
        char respuesta[128];
        scanf("%s", respuesta);
        //strcmp compara strings, en este caso se usa para saber que

```

```

        digito el ususario
        resultado = strcmp(respuesta,"0");
    }else{
        resultado = 1;
    }
    if (resultado != 0){
        //se declaran los arreglos y variables que contendran los nombres y
        puntos de los jugadores
        char jugadorUno[128] = {" "};
        char jugadorDos[128] = {" "};
        char jugadorTres[128] = {" "};
        char jugadorCuatro[128] = {" "};
        int puntosJugadorUno = 0;
        int puntosJugadorDos = 0;
        int puntosJugadorTres = 0;
        int puntosJugadorCuatro = 0;
        int numeroJugadores = 0;
        //punteros que apuntaran a cada uno de los nombres y puntos de los jugadores.
        int Ppuntos[4] = {puntosJugadorUno,puntosJugadorDos,puntosJugadorTres,
        puntosJugadorCuatro};
        char* PJugadores[4] = {jugadorUno,jugadorDos,jugadorTres,jugadorCuatro};
        //se declara el areglo y variables que contendra las preguntas y la respuesta
        char pregunta[128];
        char opcionA[128];
        char opcionB[128];
        char opcionC[128];
        char opcionD[128];
        int correcta = 0;
        //se declara el puntuero que apuntara a las pregunta
        char* Ppregunta[5] = {pregunta,opcionA,opcionB,opcionC,opcionD};

        //parte del codigo donde todo se integra para realizar el juego
        //Llenamos la lista de jugadores
        numeroJugadores = LeerNombres(PJugadores);
        //calculamos el numero de preguntas que traera el archivo
        int cantidadPreguntas = (lineas / 7);
        int contadorPregunta = 1;
        //definimos el jugador inicial
        int jugador = JugadorRandom(numeroJugadores);
        //mientras existan preguntas se ejecuta el ciclo
        //se le resta uno para que empiece en 0
        numeroJugadores = numeroJugadores -1;
        while (contadorPregunta <= cantidadPreguntas){
            system("clear");
            Titulo();
            //formato de color para la impresion de las preguntas al jugador
            if (jugador == 0){
                printf(ANSI_COLOR_BLUE);
            }
            if (jugador == 1){
                printf(ANSI_COLOR_YELLOW);
            }
            if (jugador == 2){
                printf(ANSI_COLOR_GREEN);
            }

```

```

    }
    if (jugador == 3){
        printf(ANSI_COLOR_MAGENTA);
    }
    //lineas tiene la cantidad de lineas del archivo
    //numeroJugadores tiene la cantidad de jugadores
    //jugadorInicial tenfra el jugador que comenzara la partida
    correcta = PreguntaNueva(argv, Ppregunta, contadorPregunta);
    printf("Turno de: %s",PJugadores[jugador]);
    printf("\n");
    printf("Pregunta numero %d\n",contadorPregunta);
    printf("* %s\n",Ppregunta[0]);
    printf("1: %s\n",Ppregunta[1]);
    printf("2: %s\n",Ppregunta[2]);
    printf("3: %s\n",Ppregunta[3]);
    printf("4: %s\n",Ppregunta[4]);
    //printf("%d\n",correcta);
    int respuesta = 0;
    scanf("%d",&respuesta);
    if (respuesta == correcta){
        SumarPunto(Ppuntos, jugador);
    }
    //continuamos con el siguiente jugador
    jugador++;
    if (jugador > numeroJugadores){
        jugador = 0;
    }
    contadorPregunta++;
}
DeclararGanador(PJugadores,Ppuntos);
}
}

```

3. Conclusión

Se concluye que el uso de punteros en C es fundamental para el buen manejo de los datos, aunque al principio el uso de punteros puede resultar algo abstracto después de usarlo algún tiempo se puede aclarar mejor el uso de los mismos y se logra entender la importancia que estos tienen