



**Facultad de**

Ciencia, Tecnología y Ambiente

INGENIERÍA EN SISTEMAS DE INFORMACIÓN

Metodología de la Programación

**Docente:**

Ing. José Alejandro Duran García

**PRÁCTICA DE FAMILIARIZACIÓN**

**Autores:**

1. Maura Sophia Quintanilla Quant
2. Lorena Judith Flores Vega
3. Stephany Daiana Flores Baltodano

Managua, Nicaragua - 2023

Contenido

[**1.**](#_gjdgxs) **FASE DE ANÁLISIS** 1

[1.1.](#_30j0zll) PLANTEAMIENTO DEL PROBLEMA 1

[1.2.](#_1fob9te) DIAGRAMA DE ESTRUCTURA 1

[1.3.](#_3znysh7) RESULTADOS DE LA FASE DE ANÁLISIS 1

[**2.**](#_2et92p0) **FASE DE DISEÑO** 2

[1.1.](#_tyjcwt) ALGORITMOS 2

[1.1.1.](#_3dy6vkm) Pseudocódigo 2

[1.1.2.](#_1t3h5sf) Diagrama de Flujo de Datos 3

[**3.**](#_4d34og8) **FASE DE CODIFICACIÓN Y DEPURACIÓN** 3

[3.1.](#_2s8eyo1) Estructuras de datos del proyecto 3

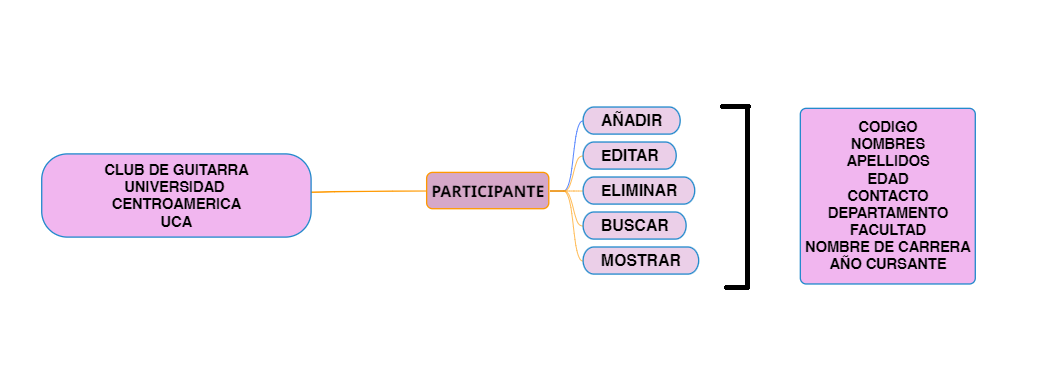
[3.2.](#_17dp8vu) Código fuente y salidas 3

# **FASE DE ANÁLISIS**

## **PLANTEAMIENTO DEL PROBLEMA**

El problema o necesidad es la falta de una herramienta para registrar la inscripción de estudiantes interesados en formar un club de guitarra en la Universidad Centroamericana (UCA). Esto afecta a los estudiantes que desean practicar, compartir conocimientos y disfrutar de la música en un entorno comunitario centrado en la guitarra. El impacto asociado incluye limitaciones en el desarrollo de habilidades musicales, pérdida de oportunidades de aprendizaje y colaboración, así como la falta de visibilidad y reconocimiento por parte de la universidad y otros estudiantes interesados en la guitarra.

## **DIAGRAMA DE ESTRUCTURA**



## RESULTADOS DE LA FASE DE ANÁLISIS

| **DATOS DE ENTRADA y SALIDA** | | | |
| --- | --- | --- | --- |
|  |  |  |  |
| **Identificador** | **Tipo de Entrada / Salida** | **Tipo de Dato** | **Valor inicial** |
| Identificador | Tipo de Entrada | Tipo de Dato |  |
| nombreCompleto | Desconocido | Cadena |  |
| departamento | Desconocido | Cadena |  |
| carreraUniversitaria | Desconocido | Entero |  |
| añoCursando | Desconocido | Entero |  |
| telefono | Desconocido | Entero |  |
| celular | Desconocido | Entero |  |
| id | Desconocido | Entero |  |
| nombreFacultad | Desconocido | Cadena |  |

| **MÉTODOS** | | |
| --- | --- | --- |
|  |  |  |

|  | | |
| --- | --- | --- |
| Nombre | Descripción | Fórmulas |
| agregarNombre | necesitaremos que ingrese su id,nombre completo, departamento, año que cursa, correo Electronico,nuero de telefono y celular, la facultar y su carrera | resultado=numero1+numero2 |
| editarEstudiante |  |  |
| eliminarEstudiante | Se necesitara Id y reordenar el arreglo |  |
| mostrarEstudiante | Muestra los datos del estudiantes especifico por medio del ID |  |
| mostrarEstudiantes | Muestro los estudiantes |  |
| buscarEstudiantes | Busco el estudiante por el id retorno la posicion donde se encuentra |  |

| **SALIDA 1** |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
| **Datos** |  | **Formato** | |
| **nombreDelEstudiante** |  | **Nombre del estudiante:Ludwing Ivan** | |
| **ApellidoDelEstudiante** |  | **Apellido delEstudiante: Flores Gonzalez** | |
| **idDelEstudante** |  | **Ide del Estudiante: 000012345** | |
| **numeroDeTelefono** |  | **numero de telefono: 0000-0000** | |
| **emailDelEstudiante** |  | **E-mail: ludwing.flores1234@est.uca.edu.ni** | |
| **facultadDeLaUni** |  | **Facultad: Facultad de ciencia, tecnologia y ambiente** | |
| **CarreraDeLaUni** |  | **Carrera: Ingieneria en sistema deinformacion** | |
| **añoCursando** |  | **Año cursando 2 año** |  |

# **FASE DE DISEÑO**

## ALGORITMOS

## Pseudocódigo



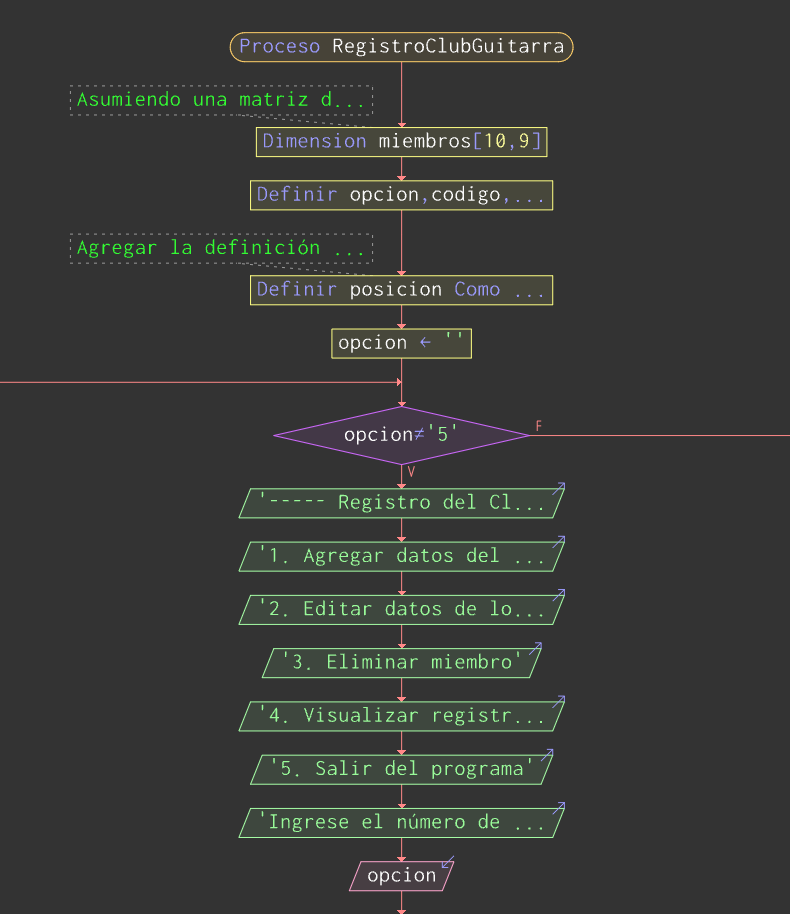


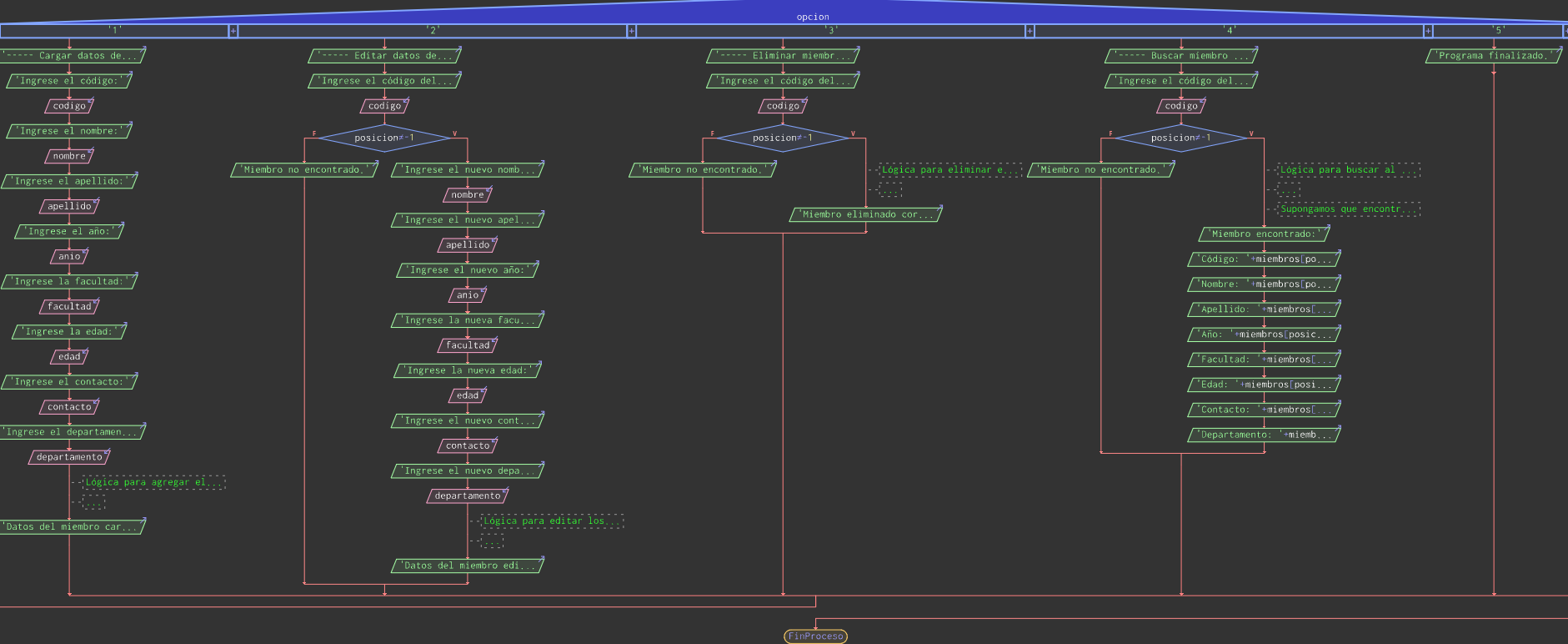






## Diagrama de Flujo de Datos





# **FASE DE CODIFICACIÓN Y DEPURACIÓN**

## Estructuras de datos del proyecto



## Código fuente y salidas

#include <iostream>

#include "cursor.h"

#include <stdio.h>

#include <string.h>

using namespace std;

#define MAX 70

// ESTRUCTURAS DE DATOS CARRERA

typedef struct

{

char facultad[MAX];

char nombre[40];

char año[9];

} carrera;

// ESTRUCTURAS DE DATOS PARTICIPANTE

typedef struct

{

char cod[10];

char nombre[35];

char apellido[30];

char edad[7];

char contacto[8];

char departamento[MAX];

carrera car;

} participante;

participante parti[MAX];

int UltReg = 0;

// FUNCIONES CRUD

// CREATE

void addStudent(participante par);

// READ

void showStudent(int pos);

void showStudents();

int isStudent(char cod[]);

void startStudent(int pos);

participante getStudent(int pos);

// UPDATE

void updateStudent(participante par, int pos);

// DELETE

void deleteStudent(int pos);

// FICHEROS

FILE \*registroParticipante;

void saveStudents();

void readStudent();

int calcUltReg(FILE \*archivo);

void saveStudents()

{

registroParticipante = fopen("datosParticipante.bin", "wb");

fwrite(parti, sizeof(participante), UltReg, registroParticipante);

fclose(registroParticipante);

}

void readStudent()

{

registroParticipante = fopen("datosParticipante.bin", "rb");

if (registroParticipante == NULL)

{

return;

}

UltReg = calcUltReg(registroParticipante);

fread(parti, sizeof(participante), MAX, registroParticipante);

fclose(registroParticipante);

}

int calcUltReg(FILE \*archivo)

{

int size, num;

fseek(archivo, 0, SEEK\_END);

size = ftell(archivo);

rewind(archivo);

num = size / sizeof(participante);

return num;

}

void addStudent(participante par)

{

parti[UltReg] = par;

UltReg++;

}

void showStudent(int pos)

{

gotoxy(10, 2);

cout << "------------ Datos estudiante ------------" << endl;

gotoxy(10, 3);

cout << "Codigo: " << parti[pos].cod << endl;

gotoxy(10, 4);

cout << "Nombres: " << parti[pos].nombre << endl;

gotoxy(10, 5);

cout << "Apellidos: " << parti[pos].apellido << endl;

gotoxy(10, 6);

cout << "Edad: " << parti[pos].edad << endl;

gotoxy(10, 7);

cout << "Numero de contacto: " << parti[pos].contacto << endl;

gotoxy(10, 8);

cout << "Departamento: " << parti[pos].departamento << endl;

gotoxy(10,9);

cout << "------------ Datos carrera ------------" << endl;

gotoxy(10,10);

cout << "Facultad: " << parti[pos].car.facultad << endl;

gotoxy(10,11);

cout << "Carrera: " << parti[pos].car.nombre << endl;

gotoxy(10,12);

cout << "Ano cursante: " << parti[pos].car.año << endl;

}

int isStudent(char cod[9])

{

int pos = -1;

for (int i = 0; i < UltReg; i++)

{

if (strcmp(cod, parti[i].cod) == 0)

{

pos = i;

break;

}

}

return pos;

}

participante getStudent(int pos)

{

return parti[pos];

}

void updateStudent(participante par, int pos)

{

parti[pos] = par;

}

void deleteStudent(int pos)

{

if (pos == UltReg)

{

cout << "No tenemos un registro de este participante." << endl;

return;

}

for (int i = 0; i < UltReg; i++)

{

parti[i] = parti[i + 1];

}

UltReg--;

startStudent(UltReg);

}

void showStudents()

{

system("cls");

if (UltReg == 0)

{

cout << "No existe registro de participantes" << endl;

return;

}

for (int i = 0; i < UltReg; i++)

{

showStudent(i);

}

cout << "Ultimo registro de estudiante... " << endl;

}

void startStudent(int pos)

{

strcpy(parti[pos].cod, "");

strcpy(parti[pos].nombre, "");

strcpy(parti[pos].apellido, "");

strcpy(parti[pos].edad, "");

strcpy(parti[pos].contacto, "");

strcpy(parti[pos].car.facultad, "");

strcpy(parti[pos].car.nombre, "");

strcpy(parti[pos].car.año, "");

}

int menu()

{

int op;

cout << "-----------------------------------------" << endl;

cout << "' UNIVERSIDAD CENTROAMERICANA '" << endl;

cout << "' UCA '" << endl;

cout << "' CLUB DE GUITARRA 2023 '" << endl;

cout << "'---------------------------------------'" << endl;

cout << "Cantidad de estudiantes inscritos: " << UltReg << endl;

cout << "1. Inscribir nuevo participante" << endl;

cout << "2. Editar registro de participante" << endl;

cout << "3. Eliminar participante" << endl;

cout << "4. Buscar datos de inscripcion de participante" << endl;

cout << "5. Mostrar todas los participantes" << endl;

cout << "6. Salir del programa." << endl;

cin >> op;

return op;

}

void start()

{

int op, pos, answ;

char cod[9];

participante par;

readStudent();

do

{

system("cls||clear");

op = menu();

switch (op)

{

case 1:

system("cls || clear");

gotoxy(10,4);

cout << "---------------DATOS PERSONALES---------------" << endl;

gotoxy(10, 5);

cout << "Id del estudiante:";

gotoxy(10, 6);

cout << "Ingrese los nombres del estudiante: ";

gotoxy(10, 7);

cout << "Ingrese los apellidos del estudiante: ";

gotoxy(10, 8);

cout << "Ingrese la edad del estudiante: ";

gotoxy(10, 9);

cout << "Ingrese el numero de contacto que proporciono el estudiante: ";

gotoxy(10, 10);

cout << "Ingrese el departamento de residencia del estudiante: ";

gotoxy(10,11);

cout << "---------------DATOS DE CARRERA---------------" << endl;

gotoxy(10, 12);

cout << "Ingrese la facultad de la Universidad: " << endl;

gotoxy(10, 13);

cout << "Ingrese el nombre de la carrera: " << endl;

gotoxy(10, 14);

cout << "Ingrese el ano que esta cursando este participante: " << endl;

gotoxy(29, 5);

scanf(" %[^\n]", par.cod);

gotoxy(46, 6);

scanf(" %[^\n]", par.nombre);

gotoxy(48, 7);

scanf(" %[^\n]", par.apellido);

gotoxy(42, 8);

scanf(" %[^\n]", par.edad);

gotoxy(71, 9);

scanf(" %[^\n]", par.contacto);

gotoxy(64, 10);

scanf(" %[^\n]", par.departamento);

gotoxy(49, 12);

scanf(" %[^\n]", par.car.facultad);

gotoxy(43, 13);

scanf(" %[^\n]", par.car.nombre);

gotoxy(62, 14);

cin >> par.car.año;

addStudent(par);

cout << "Participante anadido con exito." << endl;

system("pause");

break;

case 2:

system("cls||clear");

gotoxy(10, 1);

cout << "Escribe el codigo del participante a buscar: " << endl;

gotoxy(55, 1);

scanf(" %[^\n]", cod);

pos = isStudent(cod);

showStudent(pos);

gotoxy(2, 10);

cout << "------------ Datos a editar ------------" << endl;

gotoxy(10, 11);

cout << "Id del estudiante:";

gotoxy(10, 12);

cout << "Ingrese los nombres del estudiante: ";

gotoxy(10, 13);

cout << "Ingrese los apellidos del estudiante: ";

gotoxy(10, 14);

cout << "Ingrese la edad del estudiante: ";

gotoxy(10, 15);

cout << "Ingrese el numero de contacto que proporciono el estudiante: ";

gotoxy(10, 16);

cout << "Ingrese el departamento de residencia del estudiante: ";

gotoxy(10, 18);

cout << "Ingrese la facultad de la Universidad: " << endl;

gotoxy(10, 19);

cout << "Ingrese el nombre de la carrera: " << endl;

gotoxy(10, 20);

cout << "Ingrese el ano que esta cursando este participante: " << endl;

gotoxy(29, 11);

scanf(" %[^\n]", par.cod);

gotoxy(46, 12);

scanf(" %[^\n]", par.nombre);

gotoxy(48, 13);

scanf(" %[^\n]", par.apellido);

gotoxy(42, 14);

scanf(" %[^\n]", par.edad);

gotoxy(71, 15);

scanf(" %[^\n]", par.contacto);

gotoxy(64, 16);

scanf(" %[^\n]", par.departamento);

gotoxy(49, 18);

scanf(" %[^\n]", par.car.facultad);

gotoxy(43, 19);

scanf(" %[^\n]", par.car.nombre);

gotoxy(62, 20);

cin >> par.car.año;

updateStudent(par, pos);

gotoxy(10, 22);

cout << "Registro de inscripcion actualizado... " << endl;

system("pause");

break;

case 3:

system("cls||clear");

if (UltReg == 0)

{

cout << "No hay registros que eliminar..." << endl;

}

cout << "Escribe el Id del participante: ";

cin >> cod;

pos = isStudent(cod);

par = getStudent(pos);

cout << "Estas seguro de eliminar la inscripcion del siguiente estudiante: " << par.nombre << " " << par.apellido << "?\n";

cout << "Escribe 1 para SI o 2 para NO: ";

cin >> answ;

if (answ == 1)

{

deleteStudent(pos);

cout << "Registro Eliminado... \n";

}

else

{

cout << "Operacion cancelada.... \n";

}

system("pause");

break;

case 4:

system("cls||clear");

cout << "Escribe el codigo del estudiante a buscar: ";

scanf(" %[^\n]", cod);

pos = isStudent(cod);

showStudent(pos);

system("pause");

break;

case 5:

system("cls||clear");

showStudents();

system("pause");

break;

case 6:

break;

default:

system("cls||clear");

cout << "Opcion invalida" << endl;

system("pause");

break;

}

} while (op != 6);

saveStudents();

}