

# Programación

CICLO: 2024

NOMBRE: Jose Samuel

SECCION: "A"  
Aguilar

APELLIDOS: Jimenez

SEMESTRE: Tercer Semestre

CARNÉ: 5090-22-4196

CARRERA: Ingeniería en Sistemas

CURSO: PROGRAMCION



UNIVERSIDAD  
MARIANO GÁLVEZ

## PRUEBA No. 1

FECHA DE ENTREGA:     /     /

FIRMA: \_\_\_\_\_

## Introducción:

Este documento da un vistazo en el programa realiza que dos formas y vistas de cómo hacerlo dando que uno fue intento y el otro fue más elaborado.

Muy fácil o difícil es la experiencia, pero al fin a cabo se va a ver en el documento simple

Buenas soy: Jose Samuel Jimenez Aguilar

Es un programa que intente cambiar con los dos mundos un intento y el otro diseñado y modificado:

DICEÑO INTENTO:

```
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  class Personal {
7  private:
8      string nombre;
9      int edad;
10     string ocupacion;
11 public:
12     Personal(string n, int e, string o) {
13         nombre = n;
14         edad = e;
15         ocupacion = o;
16     }
17
18     void mostrarInfo() {
19         cout << "Nombre: " << nombre << ", Edad: " << edad << ", Ocupacion: " << ocupacion << endl;
20     }
21 };
22
23 int main() {
24     string n, o; // Declarar las variables para almacenar el nombre y la ocupación
25     int e; // Declarar la variable para almacenar la edad
26
27     cout << "Ingrese el nombre: ";
28     cin >> n;
29     cout << "Ingrese la edad: ";
30     cin >> e;
```

resulta que las modificaciones fueron que se puso int en edad pues son datos que tienen número.

Void es como el cin pues agarra a nivel general lo de la clase.

Además se nota que persona.mostrarInfo da la muestra de la información que agregué.

```
31     cout << "Ingrese la ocupacion: ";
32     cin >> o;
33
34     Personal persona(n, e, o); // Crear un objeto de la clase Personal
35
36     // Llamar al método mostrarInfo() para mostrar la información de la persona
37     persona.mostrarInfo();
38
39     return 0;
40 }
```

```
Ingrese el nombre: samuel
Ingrese la edad: 19
Ingrese la ocupacion: estudiante
Nombre: samuel, Edad: 19, Ocupacion: estudiante
```

```
-----
Process exited after 9.65 seconds with return value 0
Presione una tecla para continuar . . . |
```

MAS ESTENSO:

```
1  #include <iostream>
2  #include <fstream>
3  #include <vector>
4  #include <string>
5
6  using namespace std;
7
8  // Definición de la clase Persona
9  class persona {
10 private:
11     string nombre;
12     int edad;
13     string ocupacion;
14 public:
15     // Constructor
16     persona(string n, int e, string o) {
17
18         // Métodos para establecer y obtener los valores de los atributos
19         void setnombre(string n){ nombre = n; }
20         void setedad(int e) { edad = e; }
21         void setocupacion(string o) { ocupacion = o; }
22         string getnombre() const {
23             return nombre;
24         }
25         string getedad() const {
26             return edad;
27         }
28         string getocupacion() const {
29             return ocupacion;
30         }
31     };
32
33     // Función para almacenar Los datos de Las personas en un archivo de texto
34     void guardarpersonas(const vector<persona>& personas) {
35         ofstream archivo("personas.txt");
36         if (!archivo.is_open()) {
37             cout << "Error al abrir el archivo." << endl;
38             return;
39         }
40         for (const auto& persona : personas) {
41             archivo << "Nombre: " << persona.getnombre() << ", Edad: " << persona.getedad() << ", Ocupacion: " << persona.getocupacion() << endl;
42         }
43         archivo.close();
44         cout << "Datos guardados en el archivo personas.txt" << endl;
45     }
46
47     // Función para Leer Los datos de Las personas desde un archivo de texto y mostrarlos en pantalla
48     void mostrarpersonas() {
49         ifstream archivo("personas.txt");
50         if (!archivo.is_open()) {
51             cout << "Error al abrir el archivo." << endl;
52             return;
53         }
54         string linea;
55         cout << "Datos de las personas almacenadas:" << endl;
56         while (getline(archivo, linea)) {
57             cout << linea << endl;
58         }
59         archivo.close();
60     }
61 }
```

se usa el void como cin adeams de combinar especificacio con la parte de set con la espcificacion de nombre, edad y ocupacion con sus respectivas bolsas es decir sus donde ponen la informacion

```
31 }
32 };
33
34 // Función para almacenar Los datos de Las personas en un archivo de texto
35 void guardarpersonas(const vector<persona>& personas) {
36     ofstream archivo("personas.txt");
37     if (!archivo.is_open()) {
38         cout << "Error al abrir el archivo." << endl;
39         return;
40     }
41     for (const auto& persona : personas) {
42         archivo << "Nombre: " << persona.getnombre() << ", Edad: " << persona.getedad() << ", Ocupacion: " << persona.getocupacion() << endl;
43     }
44     archivo.close();
45     cout << "Datos guardados en el archivo personas.txt" << endl;
46 }
47
48 // Función para Leer Los datos de Las personas desde un archivo de texto y mostrarlos en pantalla
49 void mostrarpersonas() {
50     ifstream archivo("personas.txt");
51     if (!archivo.is_open()) {
52         cout << "Error al abrir el archivo." << endl;
53         return;
54     }
55     string linea;
56     cout << "Datos de las personas almacenadas:" << endl;
57     while (getline(archivo, linea)) {
58         cout << linea << endl;
59     }
60     archivo.close();
61 }
```

se un especificacion de que pasa si no cumple con los datos y se agraga un bucle si quiere repetir o

agregar informacion to puesto en es decir un archivo y enseña en la parte mostrarpersonas

```
61 }
62
63 int main() {
64     vector<persona> personas;
65
66     cout << "Ingrese los datos de las personas (nombre, edad y ocupacion), o escriba 'fin' para terminar:" << endl;
67     while (true) {
68         string nombre, ocupacion, edad;
69         cout << "Nombre: ";
70         getline(cin, nombre);
71         if (nombre == "fin") {
72             break;
73         }
74         cout << "Edad: ";
75         cin >> edad;
76         cin.ignore(); // Limpiar el buffer del salto de línea
77         cout << "Ocupacion: ";
78         getline(cin, ocupacion);
79
80         personas.push_back(persona(nombre, edad, ocupacion));
81     }
82
83     // Guardar los datos de las personas en un archivo
84     guardarpersonas(personas);
85
86     // Mostrar los datos de las personas almacenadas en el archivo
87     mostrarpersonas();
88
89     return 0;
90 }
```

## Conclusion.

Visto de otro modo se puede dar que el conocimiento es mejorable pues el del intento aunque no es tan detallado es facil y lo mas importante si funciona mientras el otro si es extenso ademas de mas detallado pero eso puede jugar en contra de la idea y la factividad del trabajo.

E-grafia

Chat gpt (“datos en combersacion privada”)