

Universidad Mariano Gálvez de Guatemala

Campus Villa Nueva, Guatemala

Ingeniería en Sistemas

Ing. Carlos Arias

Curso: Programación I

Parcial 2

Sección: “A”

Carné: 5090-23-1407

Nombre: Angelyn Judith Díaz Zeceña

Contenido

Ejercicio 1.

```
parcial2.cpp  +  X
PARCIAL2PROGRAANGELYNDIAZ  (Ámbito global)

1  //Angelyn Judith Diaz Zeceña
2  #include <iostream> //Aqui podemos ver que se utilizan la biblioteca para entrada y salida de datos
3  #include <fstream> //Aqui hay una biblioteca que sirve para el manejo de archivos adecuados
4  #include <string>
5  #include <sstream>
6
7  using namespace std;
8
9  struct Empleado {
10     string nombre;
11     string apellido;
12     int edad;
13     double salario;
14 };
15
16 // Aqui se muestra la información de un empleado
17 void mostrarEmpleado(const Empleado& empleado) {
18     cout << "Nombre: " << empleado.nombre << endl;
19     cout << "Apellido: " << empleado.apellido << endl;
20     cout << "Edad: " << empleado.edad << endl;
21     cout << "Salario: " << empleado.salario << endl;
22 }
23
24 // Aqui se busca el empleado por nombre completo en el archivo
25 bool buscarEmpleado(const string& nombreCompleto) {
26     ifstream archivo("empleados.txt");
27     if (!archivo.is_open()) { // Aqui verifica si el archivo está abierto
28         cout << "Error al abrir el archivo empleados.txt" << endl;
29         return false;
30     }
}
```

En esta captura de pantalla, podemos ver que se utilizó una estructura de empleado, y mostrando la información del empleado gracias al cout, mostrando al usuario las características del empleado y utilizando ciclos y funciones verificando si el archivo ha sido abierto correctamente y en un caso el archivo no se puede abrir, se cierra automáticamente.

```
parcial2.cpp  X
PARCIAL2PROGRAANGELYNDIAZ  (Ámbito global)

31
32 string linea;
33 while (getline(archivo, linea)) {
34     stringstream ss(linea);
35     Empleado empleado;
36     ss >> empleado.nombre >> empleado.apellido >> empleado.edad >> empleado.salario;
37     string nombreCompletoEmpleado = empleado.nombre + " " + empleado.apellido;
38     if (nombreCompletoEmpleado == nombreCompleto) {
39         mostrarEmpleado(empleado);
40         archivo.close();
41         return true;
42     }
43 }
44 //Aquí muestra al usuario que el archivo no se encontró
45 cout << "Empleado no encontrado" << endl;
46 archivo.close();
47 return false;
48 }
49
50 int main() {
51     cout << "-----Hola personita, bienvenido al sistema de consulta de empleados-----" << endl; //Aquí da la bienvenida al usuario.
52
53     string nombreCompleto;
54     do {
55         cout << "Ingrese el nombre completo del empleado a buscar (Nombre Apellido), o 'salir' para terminar: ";
56         getline(cin, nombreCompleto);
57
58         if (nombreCompleto != "salir") {
59             buscarEmpleado(nombreCompleto);
60         }
61     } while (nombreCompleto != "salir");
62
63     return 0;
64 }
65
66 //significa que en esta parte el programa ha sido ejecutado correctamente.
```

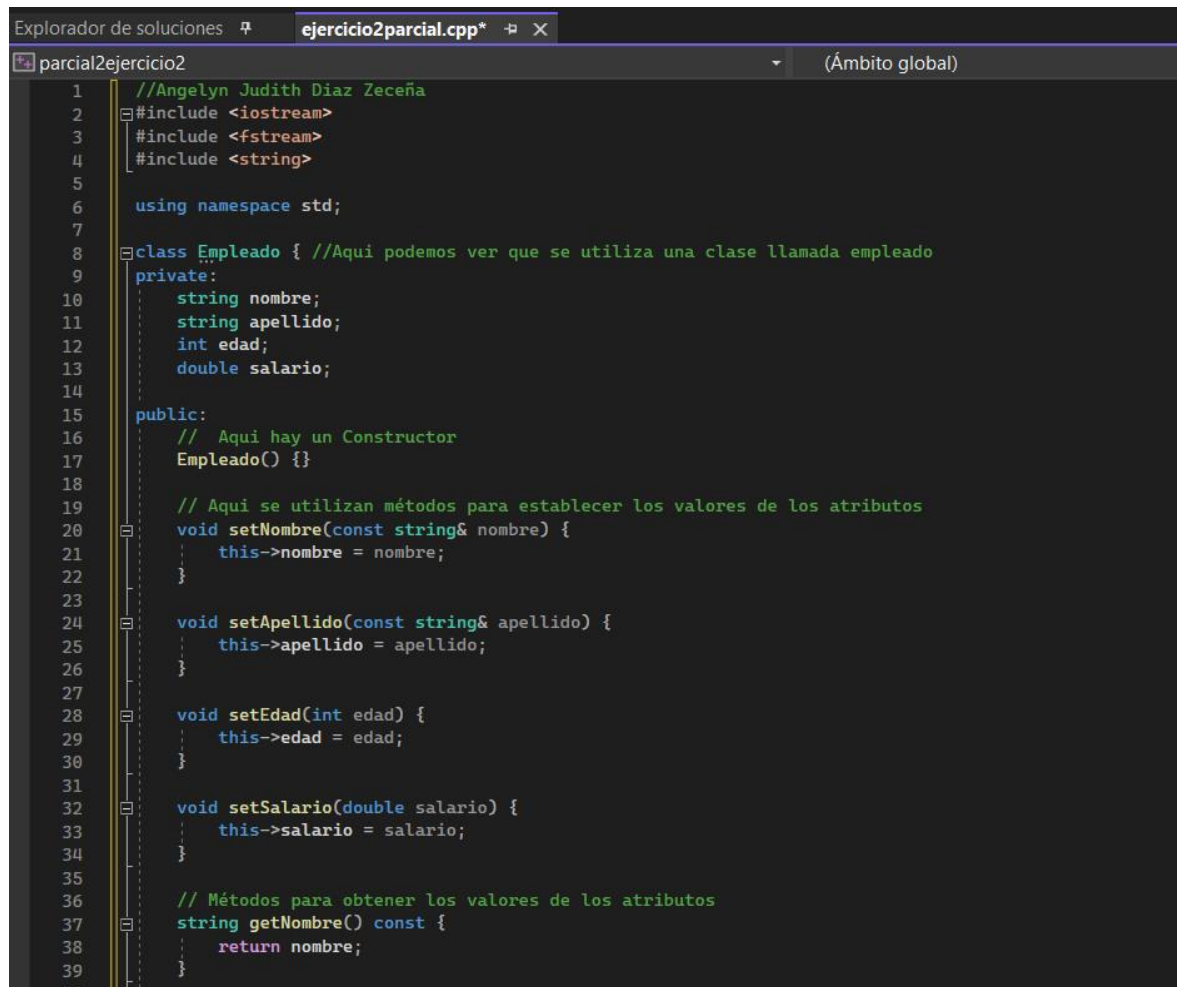
En esta parte del código, podemos ver también que ya muestra de forma más ordenada dentro del archivo, como es que está estructurada la línea del archivo, gracias al getline, y al string, y que se pueda guardar de manera correcta dentro del archivo, que es donde se puede observar que se utilizan ciclos if y while que para que, al momento de terminar el archivo, se cierre y se ejecute de manera correcta.

```
Consola de depuración de Mi  X  +  v
-----Hola personita, bienvenido al sistema de consulta de empleados-----
Ingrese el nombre completo del empleado a buscar e ingrese 'salir' para terminar: Juan lopez
Error al abrir el archivo empleados.txt
Ingrese el nombre completo del empleado a buscar e ingrese 'salir' para terminar: salir

C:\Users\Angy\OneDrive\Documentos\TERCER SEMESTRE\PARCIAL2PROGRAANGELYNDIAZ\x64\Debug\PARCIAL2PROGRAANGELYNDIAZ.exe (proceso 14520) se cerró con el código 0.
Para cerrar automáticamente la consola cuando se detiene la depuración, habilite Herramientas ->Opciones ->Depuración ->
Cerrar la consola automáticamente al detenerse la depuración.
Presione cualquier tecla para cerrar esta ventana. . .
```

Como podemos ver en esta captura de pantalla, se le pide al usuario que ingrese los datos del empleado, si es que existiera. Pero dado que no existe tal empleado, no se puede abrir el archivo. Y para salir del programa, se necesita colocar salir, para automáticamente, se salga.

Ejercicio 2.



```
Explorador de soluciones  ejercicio2parcial.cpp*  (Ámbito global)
parcial2ejercicio2
1 //Angelyn Judith Diaz Zeceña
2 #include <iostream>
3 #include <fstream>
4 #include <string>
5
6 using namespace std;
7
8 class Empleado { //Aqui podemos ver que se utiliza una clase llamada empleado
9 private:
10     string nombre;
11     string apellido;
12     int edad;
13     double salario;
14
15 public:
16     // Aqui hay un Constructor
17     Empleado() {}
18
19     // Aqui se utilizan métodos para establecer los valores de los atributos
20     void setNombre(const string& nombre) {
21         this->nombre = nombre;
22     }
23
24     void setApellido(const string& apellido) {
25         this->apellido = apellido;
26     }
27
28     void setEdad(int edad) {
29         this->edad = edad;
30     }
31
32     void setSalario(double salario) {
33         this->salario = salario;
34     }
35
36     // Métodos para obtener los valores de los atributos
37     string getNombre() const {
38         return nombre;
39     }
40 }
```

En esta parte del código, se muestra en las bibliotecas y se empiezan a definir métodos y atributos de las clases. Juntamente con un void y un string.

```
Explorador de soluciones  ejercicio2parcial.cpp*  (Ámbito global)
parcial2ejercicio2
40
41     string getApellido() const {
42         return apellido;
43     }
44
45     int getEdad() const {
46         return edad;
47     }
48
49     double getSalario() const {
50         return salario;
51     }
52
53     void imprimirDatos() const {
54         cout << "Nombre: " << nombre << endl;
55         cout << "Apellido: " << apellido << endl;
56         cout << "Edad: " << edad << endl;
57         cout << "Salario: " << salario << endl;
58     }
59 };
60
61 int main() {
62     cout << "-----Hola personita, bienvenido al sistema de registro de empleados-----" << endl;
63
64     int numEmpleados;
65     cout << "Ingrese el número de empleados: ";
66     cin >> numEmpleados;
67
68
69     cin.ignore();
70
71     Empleado empleados[numEmpleados];
72
73     // Aui se Solicita y almacena los datos de cada empleado
74     for (int i = 0; i < numEmpleados; ++i) {
75         Empleado empleado;
76         string nombre, apellido;
77         int edad;
78         double salario;
79
80         cout << "Ingrese el nombre del empleado " << i + 1 << ": ";
81         getline(cin, nombre);
82         empleado.setNombre(nombre);
83     }
```

aquí se muestra la otra parte, con un ciclo y ayuda de los cout para pedirle al usuario ingresar los datos y gracias también a la ayuda de un getline.

```
Explorador de soluciones  ejercicio2parcial.cpp  (Ámbito global)
parcial2ejercicio2
82     empleado.setNombre(nombre);
83
84     cout << "Ingrese el apellido del empleado " << i + 1 << ": ";
85     getline(cin, apellido);
86     empleado.setApellido(apellido);
87
88     cout << "Ingrese la edad del empleado " << i + 1 << ": ";
89     cin >> edad;
90     empleado.setEdad(edad);
91
92     cout << "Ingrese el salario del empleado " << i + 1 << ": ";
93     cin >> salario;
94     empleado.setSalario(salario);
95
96     // aqui se almacenan el empleado en el arreglo
97     empleados[i] = empleado;
98
99     cin.ignore();
100 }
101
102 // Aquis se escriben los datos de los empleados en un archivo
103 ofstream archivo("empleados.txt");
104 if (archivo.is_open()) {
105     for (int i = 0; i < numEmpleados; ++i) {
106         archivo << empleados[i].getNombre() << " "
107             << empleados[i].getApellido() << " "
108             << empleados[i].getEdad() << " "
109             << empleados[i].getSalario() << endl;
110     }
111     archivo.close();
112     cout << "Los datos de los empleados se han guardado en el archivo empleados.txt" << endl;
113 }
114 else {
115     cout << "Error al abrir el archivo empleados.txt" << endl;
116 }
117
118 return 0;
119 }
120 //aquí finaliza el código
```

Aquí se muestra la parte final del código, que es donde se le pide al usuario ingresar los datos, los registra y gracias a is open y el archivo close, se verifica el archivo y se cierra correctamente gracias a la ayuda de un if y un else.

```
input
----Hola personita, bienvenido al sistema de registro de empleados----
Ingrese el número de empleados: 2
Ingrese el nombre del empleado 1: mercy lopez
Ingrese el apellido del empleado 1: barrego
Ingrese la edad del empleado 1: 12
Ingrese el salario del empleado 1: 400
Ingrese el nombre del empleado 2: angy
Ingrese el apellido del empleado 2: diaz
Ingrese la edad del empleado 2: 23
Ingrese el salario del empleado 2: 20000
Los datos de los empleados se han guardado en el archivo empleados.txt

...Program finished with exit code 0
Press ENTER to exit console.
```

Como podemos ver en esta captura de pantalla, se pide al usuario ingresar la cantidad de empleados y dependiendo de la cantidad, sus nombres, apellidos, salario y años. Y se guardan automáticamente en el archivo txt.

Referencias

<https://github.com/Ashe122/PARCIAL2-EJERCICIO1.git>