

Universidad Mariano Gálvez de Guatemala

Facultad de Ingeniería Matemática y Ciencias Físicas

Campus Villa Nueva, Guatemala

**Ingeniería en Sistemas de información y Ciencias de la
computación**

Ingeniero: Carlos Arias

Curso: Programacion 1

Código de Curso: 012

Código de Carrera: 5090



Examen Parcial 2

Byron Ignacio Salazar Orellana

Sección: A

No. Carné: 23-5096

Fecha: 17/04/2024

Introducción

La programación con orientación a objetos es de suma importancia para la estructuración optima y buena reutilización de código, además del mantenimiento del mismo. Así mismo, el trabajar con archivos de texto es muy importante para no perder los datos dentro del uso de un programa, en este caso, gracias a la librería `fstream` podremos manipular los archivos de texto con mayor facilidad, como crear y también introducir datos dentro del mismo.

Para el primer programa se nos solicitó realizar un archivo de texto en donde podamos ingresar los datos de un empleado. Con la biblioteca fstream podremos manipular dichos archivos de texto, podremos ingresar los datos que deseemos en el mismo además de visualizarlos luego dentro del bloc de notas.

```
1  #include<iostream>
2  #include<stdlib.h>
3  #include<fstream>
4  using namespace std;
5
6  void escribir();
7
8  int main(){
9      escribir();
10
11
12      return 0;
13  }
14
15  void escribir(){
16      ofstream archivo_parcial;
17      string nombre, apellido;
18      int edad;
19      double salario;
20
21      getline(cin, Empleados.txt);
22      archivo_parcial.open(getline(std::Empleados.txt), ios::out);
```

Lo primero es abrir el archivo y si este no existe, lo creamos, gracias a la función ofstream, directamente con el nombre Empleados.txt, luego de ello, creamos las variables para que estas luego sean los campos que se van a llenar dentro del archivo de texto. Como podemos notar nuestra función principal manda a llamar la función que nos ayuda a generar los archivos de texto.

Con getline, nos permite introducir el nombre del archivo, ya que lee la línea que contiene el nombre del mismo y se lo asigna en el archivo txt.

```

if(archivo_parcial.fail()){
    cout<<"no se pudo abrir el archivo ";
    exit(1);
}

cout<<"\n Escriba el nombre del empleado ";
getline(cin,nombre);

cout<<"\n Escriba el apellido del empleado ";
getline(cin,apellido);

cout<<"\n Escriba la edad del empleado ";
getline(cin,edad);

cout<<"\n Escriba el salario del empleado ";
getline(cin,salario);

archivo_parcial<<nombre;
archivo_parcial<<apellido;
archivo_parcial<<edad;
archivo_parcial<<salario;

cout<<"\n Texto generado! abra su archivo Empleados.txt ";

archivo_parcial.close();

```

Luego como podremos notar sale un mensaje por si en dado caso el archivo no existe o contiene algún error, también el usuario puede ingresar los datos y gracias al getline este lee los datos ingresados y con archivo_parcial se transcriben al archivo de texto. Por ultimo sale un mensaje que le indica al usuario que sus datos fueron ingresados a empleados.txt.

```
1  #include <iostream>
2  #include <string>
3  #include<fstream>
4
5  using namespace std;
6
7  class empleados{
8      private:
9          string nombre;
10         string apellido;
11         int edad;
12         double salario;
13     public:
14         empleados(string, string, int, double);
15         void nombre();
16         void apellido();
17         void edad();
18         void salario();
19 }
```

Para el segundo inciso se nos pide realizar una clase empleados en donde tengamos atributos como nombre, edad, apellido y salario, designado para cada empleado.

```

    },
    empleados::empleados(string _nombre, string _apellido, int _edad, double _salario){
        nombre = _nombre;
        apellido = _apellido;
        edad = _edad;
        salario = _salario;
    }

    int main(){
        empleados();
        string nombre, apellido;
        int edad;
        double salario;

        cout<<"Ingrese el nombre"<<endl;
        cin>> nombre;
        cout<<"Ingrese el apellido"<<endl;
        cin>> apellido;
        cout<<"Ingrese la edad"<<endl;
        cin>> edad;
        cout<<"Ingrese el salario"<<endl;
        cin>> salario;
        empleados = empleados()
    }

```

Luego el conductor de empleados contiene todas las variables que están igualados a los atributos, es quien les designa un valor a las mismas.

El main manda a llamar a la clase empleados y le permite al usuario ingresar los datos.

```
cout<<"Ingrese el nombre"<<endl;
cin>> nombre;
cout<<"Ingrese el apellido"<<endl;
cin>> apellido;
cout<<"Ingrese la edad"<<endl;
cin>> edad;
cout<<"Ingrese el salario"<<endl;
cin>> salario;
empleados = empleado1(nombre, apellido, edad, salario)

empleado1.nombre;
empleado1.apellido;
empleado1.edad;
empleado1.salario;
```

Luego de que el usuario asigna los datos, el objeto empleado ejecuta cada una de las acciones, que en este caso serían sus datos personales antes mencionados.