

# Proyecto final PC

Steven Baquero

Diana Gutiérrez

Sebastián Gaitán

Pontificia Universidad Javeriana

Departamento De Ingeniería De Sistemas

Programación de computadores

24 de Noviembre, 2017

## 1 Introduccón

El siguiente trabajo presenta una verificación del correcto uso de TAD's tipo lista y plantillas, mediante el análisis, diseño e implementación de algoritmos para la resolución de problemas.

## 2 Declaración del problema

El turismo es un fenómeno económico y social, que contribuye al bienestar de las comunidades en países en desarrollo. Este sector económico ha presentado un rápido crecimiento en los últimos años, incluso superando los niveles de las exportaciones de petróleo y otros productos tradicionales.

El sistema debe permitir agregar, consultar, modificar y eliminar información de ciudades y sitios. Además, deberá permitir a un usuario turista revisar la información de ciudades y sitios, avanzando y retrocediendo para ver los diferentes sitios de una ciudad seleccionada. Será posible agregar una calificación y un comentario a un sitio.

## 3 Procedimientos

### 3.1 Importar librerías

para poder desarrollar el análisis correspondiente es necesario importar unas librerías ofrecidas por c++ es decir: "iostream.h", pero para un uso adecuado y organizado de archivos importaremos nuestras propias librerías, denominadas "listaT.h", "Ciudad.h", "Sitio.h", estas, incluídas en el main().

### 3.2 Declaración de estructuras

#### 3.2.1 Archivo ListaT.h

Dentro del archivo ListaT.h declaramos 2 estructuras sobre una plantilla llamadas Nodo y Lista.

La primera cuyos elementos son datos de tipo T(tipo de dato declarado en la plantilla), un dato(va a ser una estructura) y un nodo llamado siguiente que apunta a otra estructura.

---

Listing 1: Implementación de la estructura nodo.

---

```
1  template <class T>
2  struct Nodo{
3      T dato;
4      Nodo<T> *siguiente;
5  };
```

---

La segunda estructura, llamada Lista, cuyo elemento es un nodo apuntando a la cabeza de la lista. Además se tienen en nuestra segunda estructura algunas funciones creadoras, analizadoras y modificadoras cuyo funcionamiento se explicarán a continuación y se mostrarán solo las que permitirán el desarrollo del problema.

---

Listing 2: Implementación de la estructura Lista.

---

```
1  template <class T>
2  struct Lista {
3      Nodo<T> *cabeza;
4
5      void listaVacia();
6      bool esVacia();
7      void insertar(T dato, Nodo<T> *pos);
8      void insertarCabeza(T dato);
9      void insertarCola(T dato);
```

```

10     Nodo<T>* localizar(T dato);
11     void eliminar(T dato);
12     Nodo<T>* anterior(Nodo<T> *pos);
13     Nodo<T>* cabezaL();
14     void limpiar();
15     int longitud();
16     void imprimir();
17 };
18
19 #include "listaT.cpp"
20
21 #endif

```

---

La tercera estructura, llamada Ciudad, ubicada en Ciudad.h. Funciones creadoras, analizadoras y modificadoras cuyo funcionamiento se explicarán a continuación y se mostrarán solo las que permitirán el desarrollo del problema.

---

Listing 3: Implementación de la estructura Ciudad.

---

```

1
2 struct Ciudad{
3 private:
4     string nombre;
5     string pais;
6     Lista<Sitio> listaSitios;
7
8 public:
9     Ciudad();
10    Ciudad(string nombre, string pais, Lista<Sitio> lista);
11
12    void setNombre(string nombre);
13    void setPais(string pais);
14    string getNombre();
15    string getPais();
16    Lista<Sitio> getSitios();
17
18    void agregarSitio(Sitio s);
19    bool buscarSitio(Sitio s);
20    void imprimir();
21    bool operator == (Ciudad &a);
22
23 };

```

---

Análogamente, la cuarta estructura, llamada Sitio, ubicada en Sitio.h. Funciones creadoras, analizadoras y modificadoras cuyo funcionamiento se explicarán a continuación y se mostrarán solo las que permitirán el desarrollo del problema.

Listing 4: Implementación de la estructura Sitio.

---

```
1 struct Sitio{
2 private:
3     string nombre;
4     string tipo;
5     string descripcion;
6     int calificacion;
7
8 public:
9     Sitio();
10    Sitio(string n, string t, string d);
11
12    void setNombre(string nombre);
13    void setTipo(string tipo);
14    void setDescripcion(string descrip);
15    void setCalificacion(int n);
16    string getNombre();
17    string getTipo();
18    int getCalificacion();
19    void imprimir();
20    bool operator == (Sitio &a);
21 };
```

---

### 3.3 Creando funciones

#### 3.3.1 Lista vacia()-ListaT

Esta función, que es vacía, nos permite inicializar nuestra lista colocando nuestra cabeza en nulo.

#### 3.3.2 esVacia()-Lista

Esta función, que es de booleana, nos permite ver si nuestra lista está vacía o no.

#### 3.3.3 insertarCola(Tdato)-listaT

Esta función, que es de vacía y tiene como parámetros de entrada una estructura de sitio, nos permitirá añadir un sitio a los nodos de la lista de sitios siendo el ingreso al final de la misma.

#### 3.3.4 localizar(T dato)-ListaT

Esta función, que retorna un nodo y tiene como parámetro de entrada un dato(como nombre de sitio, ciudad o país), nos permitirá retornar el nodo donde está ubicado el dato(puede ser un sitio, ciudad más adelante).

### **3.3.5 longitud()-ListaT**

Esta función, nos retorna un entero que nos permitirá saber longitud de una lista.

### **3.3.6 setNombre(string nombre)-Ciudad**

Este método, del objeto Ciudad, asigna un nombre a una ciudad

### **3.3.7 setPais(string pais)-Ciudad**

Este método, del objeto Ciudad, asigna un nombre a un país

### **3.3.8 getCiudad()-Ciudad**

Este método, del objeto Ciudad, retorna un nombre de una ciudad.

### **3.3.9 getPais()-Ciudad**

Análogamente, este método, del objeto Ciudad, retorna un nombre de un país.

### **3.3.10 agregarSitio(Sitio s)-Ciudad**

Este método, del objeto Ciudad, asigna un sitio a la lista de sitios perteneciente a la ciudad.

### **3.3.11 BuscarSitio(Sitio s)-Ciudad**

Este método, del objeto Ciudad, busca un sitio de la lista de sitios perteneciente a la ciudad.

### **3.3.12 imprimir()-Ciudad**

Este método, del objeto Ciudad, muestra por pantalla los datos pertenecientes a esta ciudad, incluyendo la lista de sitios asociada a la misma.

### **3.3.13 setNombre(string nombre)-Sitio**

Este método, del objeto Sitio, asigna un nombre a una sitio.

### **3.3.14 setTipo(string tipo) -Sitio**

Este método, del objeto Sitio, asigna un tipo a una sitio.

### **3.3.15 setDescripcion(string descrip) -Sitio**

Este método, del objeto Sitio, asigna una descripción a una sitio.

#### **3.3.16 setCalificacion(int n)-Sitio**

Este método, del objeto Sitio, asigna una calificación a una sitio.

#### **3.3.17 getNombre()-Sitio**

Este método, del objeto Sitio, retorna un nombre de una ciudad.

#### **3.3.18 getTipo()-Sitio**

Este método, del objeto Sitio, retorna un el tipo de una ciudad.

#### **3.3.19 getDescripcion()-Sitio**

Este método, del objeto Sitio, retorna una descripción de una ciudad.

#### **3.3.20 imprimir()-Sitio**

Este método, del objeto Sitio, muestra por pantalla los datos pertenecientes a esta ciudad, incluyendo la lista de sitios asociada a la misma.