

**Universidad Mariano Gálvez De Guatemala
(UMG)**

Catedra: Programación III

Catedrático: José Luis Xiloj



Tema:

Intercomunicación de la red vial de la república de Guatemala a nivel de departamentos

Wilmer Antonio Sinay 1590 – 15 – 10299

Ana Raquel Donis Portillo 1590 – 22 – 11779

Jorge Leonardo Antuche Solis 1590 – 16 – 5676

Ana Rocío Del Cid López 1590 – 22 – 14313

José Miguel Ángel Sánchez López 1590 – 19 – 22031

ALGORITMOS

Agregar Departamento Al Principio De Ruta:

1. Crear un nuevo nodo con los parámetros pasados
2. Validar si la lista está vacía
 - Si está vacía:
 - Asignar el nuevo nodo a primero y ultimo
 - Si no está vacía:
3. Establecer el siguiente del nuevo nodo como el anterior nodo primero
4. Establecer el anterior del anterior nodo primero como el nuevo nodo
5. Asignar el nuevo nodo como el nuevo primer nodo
6. (Opcional) Actualizar algún puntero de rutas recorridas si se usa para recorridos

Agrega departamento al final de la ruta:

1. Crear un nuevo nodo pasando los parámetros recibidos
2. Validar si la lista está vacía
 - Si está vacía:
 - Asignar el nuevo nodo a primero y ultimo
 - Si no está vacía:
3. Establecer el puntero anterior del nuevo nodo como el nodo actual ultimo
4. Establecer el puntero siguiente del nodo actual ultimo como el nuevo nodo
5. Asignar el nuevo nodo como el nuevo nodo ultimo
6. (Opcional) Actualizar puntero de ruta actual

Agregar departamento después de otro departamento:

1. Validar que la lista no esté vacía. De lo contrario, retornar error.
2. Crear un nuevo nodo pasando los parámetros correspondientes.
3. Inicializar un puntero auxiliar apuntando al primer nodo de la lista.
4. Inicializar otro puntero auxiliar para almacenar el nodo actual durante el recorrido.
5. Recorrer la lista con el primer puntero auxiliar:
 - 5.1. Mover el puntero al nodo siguiente en cada iteración.
 - 5.2. Comparar el segundo puntero auxiliar con el nodo de referencia.
 - 5.2.1. Si coinciden, salir del bucle.
 - 5.2.2. Si no, asignar el nodo actual al segundo puntero.
6. Al salir del bucle, el primer puntero apunta al nodo después de referencia.
7. Enlazar el nuevo nodo después del nodo de referencia:
 - 7.1. Establecer siguiente del nuevo nodo como siguiente del nodo de referencia.
 - 7.2. Establecer siguiente del nodo de referencia como el nuevo nodo.
 - 7.3. Enlazar punteros anterior de forma similar.
8. Retornar operación exitosa.

agregar departamento antes de otro departamento:

1. Validar que la lista no esté vacía. De lo contrario, retornar error.
2. Crear un nuevo nodo pasando los parámetros correspondientes (nombre, distancia, etc).
3. Inicializar un puntero auxiliar apuntando al primer nodo de la lista.
4. Inicializar otro puntero auxiliar para almacenar el nodo anterior durante el recorrido.
5. Recorrer la lista con el primer puntero auxiliar hasta encontrar el nodo de referencia:
 - 5.1. Mover el puntero al nodo siguiente en cada iteración.
 - 5.2. Comparar el puntero con el nodo de referencia.
6. Al salir del bucle, el primer puntero auxiliar apunta al nodo de referencia.
7. El segundo puntero auxiliar apunta al nodo anterior al de referencia.
8. Enlazar el nuevo nodo entre los nodos anterior y de referencia:

- 8.1. Enlazar siguiente del nuevo nodo con el nodo de referencia.
- 8.2. Enlazar siguiente del nodo anterior con el nuevo nodo.
- 8.3. Enlazar anterior del nuevo nodo con el nodo anterior.
- 8.4. Enlazar anterior del nodo de referencia con el nuevo nodo.
9. Retornar operación exitosa

Algoritmo Para la Búsqueda:

1. Validar si la lista está vacía. Si lo está retornar un mensaje.
2. Inicializar un puntero que recorra la lista apuntando al primer nodo
3. Iniciar un bucle que recorra la lista hasta encontrar el último nodo:
 - 3.1 Mover el puntero al nodo siguiente en cada iteración
 - 3.2 Comparar el campo de búsqueda del nodo actual con el valor buscado
 - 3.2.1 Si coinciden, retornar el nodo encontrado
 - 3.2.2 Si no coinciden, seguir iterando
4. Una vez recorrida toda la lista sin encontrar coincidencias, retornar NULL

Mostrar los departamentos:

1. Validar si la lista está vacía, de ser así retornar mensaje
2. Inicializar puntero de recorrido apuntando al último nodo
3. Iniciar bucle mientras puntero sea distinto de null:
 - 3.1 Tomar los campos de información del nodo actual
 - 3.2 Concatenar una línea al resultado con los campos separados
 - 3.3 Mover el puntero al nodo anterior
4. Fuera del bucle:
 - Retornar el resultado generado

Eliminar el primer departamento de ruta:

1. Validar si la lista está vacía, retornar mensaje si lo está
2. Validar si sólo hay un nodo
 - Si es así, establecer primero y último en null
 - Si no sigue el siguiente paso
3. De lo contrario, hay más de un nodo:
 - 3.1 Establecer el nodo siguiente al primero como el nuevo primero
 - 3.2 Establecer el puntero anterior del nuevo primero en null
4. Liberar la referencia al nodo eliminado
5. Retornar un mensaje de operación exitosa

Eliminar el ultimo departamento de ruta:

1. Validar si la lista está vacía, retornar mensaje si lo está
2. Validar si sólo hay un nodo
 - Si es así, establecer primero y último en null
 - Si no, seguir al paso siguiente
3. De lo contrario, hay más de un nodo:
 - 3.1 Establecer el nodo anterior al último como el nuevo último
 - 3.2 Establecer el puntero siguiente del nuevo último en null
4. Liberar la referencia al nodo eliminado
5. Retornar mensaje de operación exitosa

Eliminar un departamento específico:

1. Validar lista vacía, retornar mensaje
2. Inicializar puntero para recorrer la lista
3. Recorrer lista buscando nodo a eliminar:
 - 3.1 Comparar nodo actual con el de referencia
 - 3.2 Si coinciden:
 - 3.2.1 Validar si es primer o último nodo
 - 3.2.2 Si es primero, llamar método de eliminar primero
 - 3.2.3 Si no, enlazar punteros anterior y siguiente de nodos adyacentes
 - 3.2.4 Liberar memoria del nodo
 - 3.3 Si no coinciden, seguir recorriendo
4. Si no se encontró, retornar error

Calcular la distancia entre 2 departamentos:

1. Validar si la lista está vacía, retornar mensaje si lo está
2. Inicializar puntero para recorrer la lista desde el primer nodo
3. Inicializar variables para almacenar distancias
4. Recorrer lista:
 - 4.1 Comparar nombre actual con el de inicio
 - 4.2 Si coinciden, almacenar distancia en variable inicio
 - 4.3 Comparar nombre actual con el de fin
 - 4.4 Si coinciden, almacenar distancia en variable fin
 - 4.5 Mover puntero al siguiente nodo
5. Calcular distancia final restando las almacenadas
6. Generar mensaje con el resultado

Mostrar los departamentos que son visitados por viaje:

1. Validar lista vacía, retornar mensaje
2. Inicializar puntero en primer nodo
3. Inicializar bandera en false
4. Recorrer lista:
 - 4.1 Comparar nodo actual con inicio de viaje
 - 4.2 Si coinciden, establecer bandera en true
 - 4.3 Comparar nodo actual con fin de viaje
 - 4.4 Si coinciden, establecer bandera en false
 - 4.5 Si bandera está activa:
 - 4.5.1 Agregar nombre de nodo al resultado
 - 4.6 Mover puntero al siguiente nodo
5. Retornar resultado generado

Calcular la distancia entre departamento y la ciudad:

1. Validar si la lista está vacía, retornar mensaje
2. Inicializar puntero en primer nodo
3. Recorrer lista:
 - 3.1 Comparar nombre actual con el buscado
 - 3.2 Si hay coincidencia:
 - Retornar el nodo encontrado
 - Salir del bucle
 - 3.3 De lo contrario, mover puntero al siguiente nodo
4. Al salir del bucle sin encontrar:
 - Retornar null

CODIGO DE LA SOLUCION

```
@page "/counter"
```

```
@rendermode InteractiveServer
```

```
<PageTitle>Rutas</PageTitle>
```

```
<h1>Administracion de rutas</h1>
```

```
<hr />
```

```
<p>Seleccione una de las rutas que desea administrar</p>
```

```
<select @onchange="SetearLaRuta">
```

```
    <option value="1">Sur</option>
```

```
    <option value="2">Oeste</option>
```

```
    <option value="3">Norte</option>
```

```
    <option value="4">Este</option>
```

```
</select>
```

```
<br>
```

```
<br>
```

```
<p>En la siguiente lista se muestra la ruta seleccionada</p>
```

```
<textarea role="status" value="@miVariableParaMostrarResultadosDeLaRutaSeleccionada"
style="width:100%;" rows="10"></textarea>
```

```
<hr />
```

```
<p>Distancia entre la ciudad capital y un departamento especifico</p>
```

```
<select name="cars" @onchange="SetearLaDepartamentoDelInicio">
```

```
    <option value=""></option>
```



```

<option value="Alta Verapaz">Alta Verapaz</option>
<option value="Baja Verapaz">Baja Verapaz</option>
<option value="Chimaltenango">Chimaltenango</option>
<option value="Chiquimula">Chiquimula</option>
<option value="Guatemala">Guatemala</option>
<option value="El Progreso">El Progreso</option>
<option value="Escuintla">Escuintla</option>
<option value="Huehuetenango">Huehuetenango</option>
<option value="Izabal">Izabal</option>
<option value="Jalapa">Jalapa</option>
<option value="Jutiapa">Jutiapa</option>
<option value="Petén">Petén</option>
<option value="Quetzaltenango">Quetzaltenango</option>
<option value="Quiché">Quiché</option>
<option value="Retalhuleu">Retalhuleu</option>
<option value="Sacatepequez">Sacatepequez</option>
<option value="San Marcos">San Marcos</option>
<option value="Santa Rosa">Santa Rosa</option>
<option value="Solola">Solola</option>
<option value="Suchitepequez">Suchitepequez</option>
<option value="Totonicapán">Totonicapán</option>
<option value="Zacapa">Zacapa</option>
</select>
<br><br>

<button class="btn btn-primary"
@onclick="CalcularDistanciaEntreDepartamentoyCiudad">Calcular la distancia entre el
departamento y la ciudad</button>

<br><br>

<p role="status">@miVariableParaMostrarResultadosDistanciaEntreDepartamentoyCapital</p>

```

<hr />

<p>Departamentos visitados previo a llegar a un departamento en particular, en esta opcion si es importante seleccionar la ruta</p>

<p>Punto de partida</p>

<select name="cars" @onchange="SetearLaDepartamentoDelInicio">

<option value=""></option>

<option value="Alta Verapaz">Alta Verapaz</option>

<option value="Baja Verapaz">Baja Verapaz</option>

<option value="Chimaltenango">Chimaltenango</option>

<option value="Chiquimula">Chiquimula</option>

<option value="Guatemala">Guatemala</option>

<option value="El Progreso">El Progreso</option>

<option value="Escuintla">Escuintla</option>

<option value="Huehuetenango">Huehuetenango</option>

<option value="Izabal">Izabal</option>

<option value="Jalapa">Jalapa</option>

<option value="Jutiapa">Jutiapa</option>

<option value="Peten">Peten</option>

<option value="Quetzaltenango">Quetzaltenango</option>

<option value="Quiche">Quiche</option>

<option value="Retalhuleu">Retalhuleu</option>

<option value="Sacatepequez">Sacatepequez</option>

<option value="San Marcos">San Marcos</option>

<option value="Santa Rosa">Santa Rosa</option>

<option value="Solola">Solola</option>

<option value="Suchitepequez">Suchitepequez</option>

<option value="Totonicapan">Totonicapan</option>

<option value="Zacapa">Zacapa</option>

</select>

<p>Destino del viaje</p>

<select name="cars" @onchange="SetearLaDepartamentoDeFinalizacion">

<option value=""></option>

<option value="Alta Verapaz">Alta Verapaz</option>

<option value="Baja Verapaz">Baja Verapaz</option>

<option value="Chimaltenango">Chimaltenango</option>

<option value="Chiquimula">Chiquimula</option>

<option value="Guatemala">Guatemala</option>

<option value="El Progreso">El Progreso</option>

<option value="Escuintla">Escuintla</option>

<option value="Huehuetenango">Huehuetenango</option>

<option value="Izabal">Izabal</option>

<option value="Jalapa">Jalapa</option>

<option value="Jutiapa">Jutiapa</option>

<option value="Petén">Petén</option>

<option value="Quetzaltenango">Quetzaltenango</option>

<option value="Quiché">Quiché</option>

<option value="Retalhuleu">Retalhuleu</option>

<option value="Sacatepequez">Sacatepequez</option>

<option value="San Marcos">San Marcos</option>

<option value="Santa Rosa">Santa Rosa</option>

<option value="Solola">Solola</option>

<option value="Suchitepequez">Suchitepequez</option>

<option value="Totonicapán">Totonicapán</option>

<option value="Zacapa">Zacapa</option>

</select>

<button class="btn btn-primary" @onclick="LugaresVisitadosPorViaje">Lugares visitados por viaje</button>

<p role="status">@miVariableParaMostrarResultadosDeLugaresVisitadosPorUnViaje</p>

<hr>

<p>Distancia que hay que recorrer de un departamento a otro, en esta opcion no importa seleccionar la ruta.</p>

<p>Punto de partida</p>

<select name="cars" id="DepartamentoDelMapa" @onchange="SetearLaDepartamentoDelInicio">

<option value=""></option>

<option value="Alta Verapaz">Alta Verapaz</option>

<option value="Baja Verapaz">Baja Verapaz</option>

<option value="Chimaltenango">Chimaltenango</option>

<option value="Chiquimula">Chiquimula</option>

<option value="Guatemala">Guatemala</option>

<option value="El Progreso">El Progreso</option>

<option value="Escuintla">Escuintla</option>

<option value="Huehuetenango">Huehuetenango</option>

<option value="Izabal">Izabal</option>

<option value="Jalapa">Jalapa</option>

<option value="Jutiapa">Jutiapa</option>

<option value="Petén">Petén</option>

<option value="Quetzaltenango">Quetzaltenango</option>

<option value="Quiché">Quiché</option>

```
<option value="Retalhuleu">Retalhuleu</option>
<option value="Sacatepequez">Sacatepequez</option>
<option value="San Marcos">San Marcos</option>
<option value="Santa Rosa">Santa Rosa</option>
<option value="Solola">Solola</option>
<option value="Suchitepequez">Suchitepequez</option>
<option value="Totonicapan">Totonicapan</option>
<option value="Zacapa">Zacapa</option>
</select>
<br><br>
```

<p>Destino del viaje</p>

```
<select name="cars" id="DepartamentoDelMapa2"
@onchange="SetearLaDepartamentoDeFinalizacion">
  <option value=""></option>
  <option value="Alta Verapaz">Alta Verapaz</option>
  <option value="Baja Verapaz">Baja Verapaz</option>
  <option value="Chimaltenango">Chimaltenango</option>
  <option value="Chiquimula">Chiquimula</option>
  <option value="Guatemala">Guatemala</option>
  <option value="El Progreso">El Progreso</option>
  <option value="Escuintla">Escuintla</option>
  <option value="Huehuetenango">Huehuetenango</option>
  <option value="Izabal">Izabal</option>
  <option value="Jalapa">Jalapa</option>
  <option value="Jutiapa">Jutiapa</option>
  <option value="Peten">Peten</option>
  <option value="Quetzaltenango">Quetzaltenango</option>
  <option value="Quiche">Quiche</option>
```

```
<option value="Retalhuleu">Retalhuleu</option>
<option value="Sacatepequez">Sacatepequez</option>
<option value="San Marcos">San Marcos</option>
<option value="Santa Rosa">Santa Rosa</option>
<option value="Solola">Solola</option>
<option value="Suchitepequez">Suchitepequez</option>
<option value="Totonacapan">Totonacapan</option>
<option value="Zacapa">Zacapa</option>
</select>

<br>

<br>

<button class="btn btn-primary" @onclick="DistanciaEntreDepartamentos">Distancia entre
departamentos</button>

<br>

<br>

<p role="status">@miVariableParaMostrarResultadosDeDistanciaEntreDepartamentos</p>

<hr>

<p>Top 10 lugares mas cercanos a la capital.</p>

<button class="btn btn-primary" @onclick="TopLugaresMasCercanosALaCapital">Lugares mas
cercanos a la capital</button>

<br>

<br>

<button class="btn btn-primary" @onclick="TopLugaresMasLejanosALaCapital">Lugares mas
lejanos a la capital</button>

<br>

<br>
```

```
<textarea role="status" value="@miVariableParaMostrarResultados" style="width:100%;"
rows="10"></textarea>
```

```
<hr>
```

```
<br>
```

```
<br>
```

```
<button class="btn btn-primary" @onclick="RecorrerElArbolNacional">Mostrar arbol
nacional</button>
```

```
<button class="btn btn-primary" @onclick="LimpiarResultados">Limpiar resultados</button>
```

```
<br>
```

```
<br>
```

```
<!--
```

```
<p>Segmento para agregar o eliminar datos</p>
```

```
<select name="cars" id="DepartamentoDelMapa"
@onchange="SetearLaDepartamentoDeAdministracion">
```

```
  <option value="Alta Verapaz">Alta Verapaz</option>
```

```
  <option value="Baja Verapaz">Baja Verapaz</option>
```

```
  <option value="Chimaltenango">Chimaltenango</option>
```

```
  <option value="Chiquimula">Chiquimula</option>
```

```
  <option value="Guatemala">Guatemala</option>
```

```
  <option value="El Progreso">El Progreso</option>
```

```
  <option value="Escuintla">Escuintla</option>
```

```
  <option value="Huehuetenango">Huehuetenango</option>
```

```
  <option value="Izabal">Izabal</option>
```

```
  <option value="Jalapa">Jalapa</option>
```

```
  <option value="Jutiapa">Jutiapa</option>
```

```
<option value="Peten">Peten</option>
<option value="Quetzaltenango">Quetzaltenango</option>
<option value="Quiche">Quiche</option>
<option value="Retalhuleu">Retalhuleu</option>
<option value="Sacatepequez">Sacatepequez</option>
<option value="San Marcos">San Marcos</option>
<option value="Santa Rosa">Santa Rosa</option>
<option value="Solola">Solola</option>
<option value="Suchitepequez">Suchitepequez</option>
<option value="Totonicapan">Totonicapan</option>
<option value="Zacapa">Zacapa</option>
</select>

<br>

<br>

Nombre del departamento
<input type="text" threshold="Nombre del departamento"
@bind="nombreDelDepartamento"/><br>

Distancia entre la ciudad capital
<input type="text" threshold="Distancia entre la ciudad capital" @bind="distanciaDesdeCiudad"
/><br>

Cantidad poblacional
<input type="text" threshold="Cantidad poblacional" @bind="cantidadPoblacional" /><br>

Cantidad de municipios
<input type="text" threshold="Cantidad de municipios" @bind="cantidadDeMunicipios" /><br>

<button class="btn btn-primary"
@onclick="GuardarDespuesDeUnDepartamentoDeLaRuta">Guardar despues de un
departamento</button>

<button class="btn btn-primary" @onclick="GuardarAntesDeUnDepartamentoDeLaRuta">Guardar
antes de un departamento</button>
```


<button class="btn btn-primary" @onclick="GuardarAlFinalDeLaRuta">Guardar al final de la ruta</button>

<button class="btn btn-primary" @onclick="GuardarAlInicioDeLaRuta">Guardar al inicio de la ruta</button>

<button class="btn btn-primary" @onclick="EliminarDepartamentoEspecificoEnUnaRuta">Eliminar departamento especifico en una ruta</button>

<button class="btn btn-primary" @onclick="EliminarElUltimoDestinoDeLaRuta">Eliminar el ultimo destino de la ruta</button>

<button class="btn btn-primary" @onclick="EliminarElPrimerDestinoDeLaRuta">Eliminar el primer destino de la ruta</button>

-->

```

@code {

    //Inicio de la programacion orientada a objetos

    //A continuacion debemos programar las class necesarias para poder implementar listas,
    //la cual seran utilizadas como 4 rutas

    class Nodo

    {

        //Atributos de cada nodo, cada nodo es un departamento

        public string nombreDeDepartamentoParaLista;

        public int DistanciaQueExisteEntreCiudadCapital;

        public int CantidadPoblacional;

        public int CantidadDeMunicipios;

        public string LugaresTuristicosMasVisitados;


        public Nodo siguiente;

        public Nodo anterior;

        public Nodo siguienteSecundaria;

        public Nodo anteriorSecundaria;


        public Nodo()

        { }


        //si en la lista no existen elementos se llama a este metodo

        public Nodo(string ElParametroNombreDepartamento, int ElParametroDistancia, int
ParametroPoblacion, int ElParametroMunicipios, string parametroLugaresTuristicosMasVisitados)

        {

            nombreDeDepartamentoParaLista = ElParametroNombreDepartamento;

            DistanciaQueExisteEntreCiudadCapital = ElParametroDistancia;

            CantidadPoblacional = ParametroPoblacion;

```

```

    CantidadDeMunicipios = ElParametroMunicipios;

    LugaresTuristicosMasVisitados = parametroLugaresTuristicosMasVisitados;

    siguiente = null;

    anterior = null;

    siguienteSecundaria = null;

    anteriorSecundaria = null;

}

```

```

//si en la lista existen elementos se llama a este metodo

```

```

    public Nodo(string ElParametroNombreDepartamento, int ElElParametroDistancia, int
ParametroPoblacion, int ElParametroMunicipios, Nodo liga_siguiente, Nodo liga_anterior, string
parametroLugaresTuristicosMasVisitados)

    {

        nombreDeDepartamentoParaLista = ElParametroNombreDepartamento;

        DistanciaQueExisteEntreCiudadCapital = ElElParametroDistancia;

        CantidadPoblacional = ParametroPoblacion;

        CantidadDeMunicipios = ElParametroMunicipios;

        LugaresTuristicosMasVisitados = parametroLugaresTuristicosMasVisitados;

        siguiente = liga_siguiente;

        anterior = liga_anterior;

    }

}

```

```

//class para crear listas doblemente ligadas

```

```

class ModeloDeRutasDoblementeLigada

```

```

{

    Nodo primero, ultimo, NodoParaRecorridos, NodoParaRecorridos2;

    public string resultado = "";


```

```

//metodo constructor

```

```

public ModeloDeRutasDoblementeLigada()
{
    //El primero y ultimo nodo se inician en null
    primero = ultimo = null;
}

//Metodo para retornar el primer nodo de la lista
public Nodo devolverPrimerNodoDeLista()
{
    return primero;
}

//verificar si existen elementos
public bool estaVaciaLaLista()
{
    return primero == null;
}

//limpiar resultado
public void limpiarResultado()
{
    resultado = "";
}

//agregar departamento al inicio de la ruta
public void agregarDepartamentoAlPrincipioDeRuta(string ElParametroNombreDepartamento,
int ElElParametroDistancia, int ParametroPoblacion, int ElParametroMunicipios, string
parametroLugaresTuristicosMasVisitados)
{
    if (estaVaciaLaLista())

```

```

        {
            Nodo nuevoNodo = new Nodo(ElParametroNombreDepartamento,
            ElElParametroDistancia, ParametroPoblacion, ElParametroMunicipios,
            parametroLugaresTuristicosMasVisitados);

            primero = ultimo = nuevoNodo;
        }
    else
    {
        primero = new Nodo(ElParametroNombreDepartamento, ElElParametroDistancia,
        ParametroPoblacion, ElParametroMunicipios, primero, null,
        parametroLugaresTuristicosMasVisitados);

        primero.siguiente.anterior = primero;
    }
}

//agrega departamento al final de la ruta

public void agregarDepartamentoAlFinalDeRuta(string ElParametroNombreDepartamento, int
ElElParametroDistancia, int ParametroPoblacion, int ElParametroMunicipios, string
parametroLugaresTuristicosMasVisitados)
{
    if (estaVaciaLaLista())
    {
        Nodo nuevoNodo = new Nodo(ElParametroNombreDepartamento,
        ElElParametroDistancia, ParametroPoblacion, ElParametroMunicipios,
        parametroLugaresTuristicosMasVisitados);

        primero = ultimo = nuevoNodo;
    }
    else
    {
        ultimo = new Nodo(ElParametroNombreDepartamento, ElElParametroDistancia,
        ParametroPoblacion, ElParametroMunicipios, null, ultimo,
        parametroLugaresTuristicosMasVisitados);
    }
}

```

```
        ultimo.anterior.siguiente = ultimo;
    }
}
```

```
//agregar departamento antes de otro departamento
```

```
public void agregarDepartamentoAntesDeOtro(string ElParametroNombreDepartamento, int
ElElParametroDistancia, int ParametroPoblacion, int ElParametroMunicipios, Nodo refe, string
parametroLugaresTuristicosMasVisitados)
```

```
{
    if (estaVaciaLaLista())
    {
        resultado = "La lista esta vacia.";
    }
    else
    {
        NodoParaRecorridos = primero;
        while (NodoParaRecorridos != refe)
        {
            NodoParaRecorridos2 = NodoParaRecorridos;
            NodoParaRecorridos = NodoParaRecorridos.siguiente;
        }
        Nodo nuevo = new Nodo(ElParametroNombreDepartamento, ElElParametroDistancia,
ParametroPoblacion, ElParametroMunicipios, parametroLugaresTuristicosMasVisitados);
        nuevo.siguiente = NodoParaRecorridos;
        NodoParaRecorridos2.siguiente = nuevo;
        nuevo.anterior = NodoParaRecorridos2;
        NodoParaRecorridos.anterior = nuevo;
    }
}
```

```

//agregar departamento despues de otro departamento

public void agregarDepartamentoDespuesDeOtro(string ElParametroNombreDepartamento,
int ElElParametroDistancia, int ParametroPoblacion, int ElParametroMunicipios, Nodo refe, string
parametroLugaresTuristicosMasVisitados)

{
    if (estaVaciaLaLista())
    {
        resultado = "La lista esta vacia.";
    }
    else
    {
        NodoParaRecorridos = primero;

        while (NodoParaRecorridos2 != refe)
        {
            NodoParaRecorridos2 = NodoParaRecorridos;
            NodoParaRecorridos = NodoParaRecorridos.siguiente;
        }

        Nodo nuevo = new Nodo(ElParametroNombreDepartamento, ElElParametroDistancia,
ParametroPoblacion, ElParametroMunicipios, parametroLugaresTuristicosMasVisitados);

        nuevo.siguiente = NodoParaRecorridos;
        NodoParaRecorridos2.siguiente = nuevo;
        nuevo.anterior = NodoParaRecorridos2;
        NodoParaRecorridos.anterior = nuevo;
    }
}

//para las busquedas

public Nodo buscar(string busqueda)

{

```

```
if (estaVaciaLaLista())
{
    resultado = "La lista esta vacia";
}
else
{
    NodoParaRecorridos = primero;
    while (NodoParaRecorridos.siguiente != null)
    {
        if (NodoParaRecorridos.nombreDeDepartamentoParaLista == busqueda)
        {
            return NodoParaRecorridos;
            break;
        }
        NodoParaRecorridos = NodoParaRecorridos.siguiente;
    }
}
return NodoParaRecorridos;
}
```

```
//para mostrar los departamentos
public void generarReporteDeRuta()
{
```

```
    if (estaVaciaLaLista())
    {
        resultado = "La lista esta vacia.";
    }
    else
```



```

    {
        NodoParaRecorridos = ultimo;

        while (NodoParaRecorridos != null)
        {
            resultado += "_____ " +
String.Format(Environment.NewLine);

            resultado += "Departamento: " +
NodoParaRecorridos.nombreDeDepartamentoParaLista + String.Format(Environment.NewLine);

            resultado += "Distancia a la capital: " +
NodoParaRecorridos.DistanciaQueExisteEntreCiudadCapital +
String.Format(Environment.NewLine);

            resultado += "Poblacion: " + NodoParaRecorridos.CantidadPoblacional +
String.Format(Environment.NewLine);

            resultado += "Municipios: " + NodoParaRecorridos.CantidadDeMunicipios +
String.Format(Environment.NewLine);

            resultado += "Lugares turisticos mas visitados: " +
NodoParaRecorridos.LugaresTuristicosMasVisitados + String.Format(Environment.NewLine);

            NodoParaRecorridos = NodoParaRecorridos.anterior;
        }
    }
}

//eliminar el primer departamento de ruta
public void eliminarInicioDeRuta()
{
    if (estaVaciaLaLista())
    {
        resultado = "La lista esta vacia.";
    }
    else
    {

```

```

        if (primero == ultimo)
        {
            primero = ultimo = null;
            resultado = "Se elimino exitosamente.";
        }
        else
        {
            primero = primero.siguiente;
            primero.anterior = null;
            resultado = "Se elimino exitosamente.";
        }
    }
}

```

//eliminar el ultimo departamento de ruta

```

public void eliminarFinalDeRuta()
{
    if (estaVaciaLaLista())
    {
        resultado = "La lista esta vacia.";
    }
    else
    {
        if (primero == ultimo)
        {
            primero = ultimo = null;
            resultado = "Se elimino exitosamente.";
        }
        else

```

```

    {
        ultimo = ultimo.anterior;
        ultimo.siguiete = null;
        resultado = "Se elimino exitosamente.";
    }
}
}

```

//eliminar un departamento especifico

```

public void eliminar(Nodo refe)
{
    if (estaVaciaLaLista())
    {
        resultado = "La lista esta vacia";
    }
    else
    {
        NodoParaRecorridos = primero;
        if (primero.siguiete != null)
        {
            while (NodoParaRecorridos != null)
            {
                if (NodoParaRecorridos.nombreDeDepartamentoParaLista ==
refe.nombreDeDepartamentoParaLista)
                {
                    if (NodoParaRecorridos == primero)
                    {
                        eliminarInicioDeRuta();
                    }
                }
            }
        }
    }
}

```

```

        else
        {
            NodoParaRecorridos2.siguiente = NodoParaRecorridos.siguiente;
            NodoParaRecorridos.siguiente.anterior = NodoParaRecorridos.anterior;
            resultado = "Se elimino exitosamente.";
        }
    }

    NodoParaRecorridos2 = NodoParaRecorridos;
    NodoParaRecorridos = NodoParaRecorridos.siguiente;
}
}
else
{
    resultado = "El departamento no existe, o el dato que desea eliminar es el ultimo o el
    primero en la lista, elija la opcion de eliminar el ultimo o el primero en la lista para que la
    eliminacion funcione correctamente.";
}
}
}

```

//para calcular la distancia entre 2 departamentos

```
public void distanciaEntreDepartamento(string inicio, string fin)
```

```

{
    int distanciaDelPrimerDepartamento = 0, distanciaDelSegundoDepartamento = 0;
    if (estaVaciaLaLista())
    {
        resultado = "La lista esta vacia";
    }
    else
    {

```

```

        NodoParaRecorridos = primero;

        while (NodoParaRecorridos != null)
        {
            if (NodoParaRecorridos.nombreDeDepartamentoParaLista == inicio) {
                distanciaDelPrimerDepartamento = NodoParaRecorridos.DistanciaQueExisteEntreCiudadCapital; }

            if (NodoParaRecorridos.nombreDeDepartamentoParaLista == fin) {
                distanciaDelSegundoDepartamento = NodoParaRecorridos.DistanciaQueExisteEntreCiudadCapital;
            }

            NodoParaRecorridos = NodoParaRecorridos.siguiente;
        }

        int resultadoFinal = distanciaDelSegundoDepartamento -
        distanciaDelPrimerDepartamento;

        resultado = "Distancia en km:" + resultadoFinal;
    }
}

```

//para mostrar los departamentos que son visitados por viaje

```

public void departamentosRecorridosDeUnViaje(string inicioDeViaje, string finalDeViaje)
{

    bool bandera = false;

    if (estaVaciaLaLista())
    {
        resultado = "La lista esta vacia";
    }
    else
    {
        NodoParaRecorridos = primero;

        while (NodoParaRecorridos != null)

```

```

        {
            if (NodoParaRecorridos.nombreDeDepartamentoParaLista == inicioDeViaje) { bandera =
true; }

            if (NodoParaRecorridos.nombreDeDepartamentoParaLista == finalDeViaje) { bandera =
false; }

            if (bandera == true)
            { resultado += " -> " + NodoParaRecorridos.nombreDeDepartamentoParaLista; }

            NodoParaRecorridos = NodoParaRecorridos.siguiente;
        }

    }
}

```

//para calcular la distancia entre departamento y la ciudad o nodo origen

```

public Nodo distanciaEntreDepartamentoYLaCiudad(string busqueda)
{
    if (estaVaciaLaLista())
    {
        resultado = "La lista esta vacia";
    }
    else
    {
        NodoParaRecorridos = primero;
        while (NodoParaRecorridos.siguiente != null)
        {
            if (NodoParaRecorridos.nombreDeDepartamentoParaLista == busqueda)
            {
                return NodoParaRecorridos;
                break;
            }
        }
    }
}

```

```

        NodoParaRecorridos = NodoParaRecorridos.siguiente;
    }
}
return NodoParaRecorridos;
}
}
/*finaliza la clase de las listas para las rutas*/

```

//A continuacion necesitamos programar las class para implementar arboles binarios

```

class NodoDelMunicipioDelArbol
{
    //Atributos
    public string NombreDelNodoMunicipioDelArbol;
    public int DistanciasALaCabeceraDelArbol;
    public string lugaresTuristicosMasVisitados;
    public NodoDelMunicipioDelArbol ligalZquierda;
    public NodoDelMunicipioDelArbol ligaDerecha;
} //fin de la class NodoDelMunicipioDelArbol

```

//La siguiente clase permite manipular los datos de los arboles

```

class ArbolDelDepartamento
{
    public NodoDelMunicipioDelArbol raizDelArbol;
    public string resultado = "";
    int contadorTop10 = 1;

    //Metodo constructor
    public ArbolDelDepartamento()
    {

```

```

        //inicia la raiz en null

        raizDelArbol = null;
    }

    //limpiar resultado

    public void limpiarResultado()
    {
        resultado = "";
    }

    //para insertar nodo

    public NodoDelMunicipioDelArbol InsertarNodo(string parametroNombreDeLaCabecera, int
    ElElParametroDistanciaEntreLaCabecera, NodoDelMunicipioDelArbol NodoDelMunicipioDelArbol,
    string parametrolugaresTuristicosMasVisitados)
    {
        if (NodoDelMunicipioDelArbol == null)
        {
            NodoDelMunicipioDelArbol nodoRaiz = new NodoDelMunicipioDelArbol();

            nodoRaiz.NombreDelNodoMunicipioDelArbol = parametroNombreDeLaCabecera;

            nodoRaiz.DistanciasALaCabeceraDelArbol = ElElParametroDistanciaEntreLaCabecera;

            nodoRaiz.lugaresTuristicosMasVisitados = parametrolugaresTuristicosMasVisitados;

            raizDelArbol = nodoRaiz;

            return nodoRaiz;
        }
        else
        {
            if (ElElParametroDistanciaEntreLaCabecera <
            NodoDelMunicipioDelArbol.DistanciasALaCabeceraDelArbol)
            {

```



```

        NodoDelMunicipioDelArbol.ligalZquierda =
InsertarNodo(parametroNombreDeLaCabecera, ElElParametroDistanciaEntreLaCabecera,
NodoDelMunicipioDelArbol.ligalZquierda, parametroLugaresTuristicosMasVisitados);

    }

    if (ElElParametroDistanciaEntreLaCabecera >
NodoDelMunicipioDelArbol.DistanciasALaCabeceraDelArbol)

    {

        NodoDelMunicipioDelArbol.ligaDerecha =
InsertarNodo(parametroNombreDeLaCabecera, ElElParametroDistanciaEntreLaCabecera,
NodoDelMunicipioDelArbol.ligaDerecha, parametroLugaresTuristicosMasVisitados);

    }

    return NodoDelMunicipioDelArbol;

}

} //Fin para insertar nodo

//para mostrar los municipios

public void MostrarMunicipiosDelDepartamento(NodoDelMunicipioDelArbol
NodoDelMunicipioDelArbol)

{

    if (NodoDelMunicipioDelArbol.ligalZquierda != null)

    {

        MostrarMunicipiosDelDepartamento(NodoDelMunicipioDelArbol.ligalZquierda);

    }

    if (NodoDelMunicipioDelArbol.DistanciasALaCabeceraDelArbol > 0)

    {

        resultado += raizDelArbol.NombreDelNodoMunicipioDelArbol;

        resultado += "_____ " +
String.Format(Environment.NewLine) + "Municipio: " +
NodoDelMunicipioDelArbol.NombreDelNodoMunicipioDelArbol +
String.Format(Environment.NewLine) + "Distancia a la cabecera en km: " +

```

```
NodoDelMunicipioDelArbol.DistanciasALaCabeceraDelArbol +  
String.Format(Environment.NewLine);
```

```
}
```

```
if (NodoDelMunicipioDelArbol.ligaDerecha != null)
```

```
{
```

```
    MostrarMunicipiosDelDepartamento(NodoDelMunicipioDelArbol.ligaDerecha);
```

```
}
```

```
//Fin para mostrar los municipios
```

```
//para mostrar el arbol nacional
```

```
public void MostrarArbolNacional(NodoDelMunicipioDelArbol NodoDelMunicipioDelArbol)
```

```
{
```

```
    if (NodoDelMunicipioDelArbol.ligalzquierda != null)
```

```
    {
```

```
        MostrarArbolNacional(NodoDelMunicipioDelArbol.ligalzquierda);
```

```
    }
```

```
    if (NodoDelMunicipioDelArbol.DistanciasALaCabeceraDelArbol > 0)
```

```
    {
```

```
        resultado +=
```

```
        "_____" +  
        String.Format(Environment.NewLine) + "Municipio: " +  
        NodoDelMunicipioDelArbol.NombreDelNodoMunicipioDelArbol +  
        String.Format(Environment.NewLine) + "Distancia a la cabecera en km: " +  
        NodoDelMunicipioDelArbol.DistanciasALaCabeceraDelArbol +  
        String.Format(Environment.NewLine);
```

```
}
```

```
if (NodoDelMunicipioDelArbol.ligaDerecha != null)
```

```
{
```

```
    MostrarArbolNacional(NodoDelMunicipioDelArbol.ligaDerecha);
```

```
}
```

```
}//Fin para mostrar los municipios
```

```
//para devolver la distancia entre la cabecera departamental y municipio
```

```
public NodoDelMunicipioDelArbol MostrarDistanciaDeUnMunicipioDesdeLaCabecera(String  
nombreDelNodoABuscar, NodoDelMunicipioDelArbol NodoDelMunicipioDelArbol)
```

```
{
```

```
    if (NodoDelMunicipioDelArbol.ligaIzquierda != null)
```

```
    {
```

```
        if (NodoDelMunicipioDelArbol.NombreDelNodoMunicipioDelArbol ==  
nombreDelNodoABuscar)
```

```
        {
```

```
            //MessageBox.Show(NodoDelMunicipioDelArbol.NombreDelNodoMunicipioDelArbol);
```

```
            return NodoDelMunicipioDelArbol;
```

```
        }
```

```
        NodoDelMunicipioDelArbol =  
MostrarDistanciaDeUnMunicipioDesdeLaCabecera(nombreDelNodoABuscar,  
NodoDelMunicipioDelArbol.ligaIzquierda);
```

```
    }
```

```
if (NodoDelMunicipioDelArbol.ligaDerecha != null)
```

```
{
```

```

        if (NodoDelMunicipioDelArbol.NombreDelNodoMunicipioDelArbol ==
nombreDelNodoABuscar)
        {
            //MessageBox.Show(NodoDelMunicipioDelArbol.NombreDelNodoMunicipioDelArbol);

            return NodoDelMunicipioDelArbol;
        }

        NodoDelMunicipioDelArbol =
MostrarDistanciaDeUnMunicipioDesdeLaCabecera(nombreDelNodoABuscar,
NodoDelMunicipioDelArbol.ligaDerecha);
    }

```

```

return NodoDelMunicipioDelArbol;

```

```

} //para devolver la distancia entre la cabecera departamental y municipio

```

```

//Variables para calcular las distancias entre departamentos

```

```

int primeraDistancia = 0, segundaDistancia = 0, calculoDeDistancia = 0;

```

```

bool bandera = false;

```

```

//para reiniciar las variables de calculos de distancias

```

```

public void reiniciarVariablesDeCalculo() {

```

```

    primeraDistancia = 0;

```

```

    segundaDistancia = 0;

```

```

    calculoDeDistancia = 0;

```

```

    bandera = false;

```

```

    resultado = "";

```

```

    contadorTop10 = 1;

```

```

}

```

```

//Top 10 lugares mas cercanos a la capital

```

```

    public void LugaresMasCercanosALaCapital(NodoDelMunicipioDelArbol
NodoDelMunicipioDelArbol)
    {

        if (NodoDelMunicipioDelArbol.ligal Izquierda != null)
        {
            LugaresMasCercanosALaCapital(NodoDelMunicipioDelArbol.ligal Izquierda);
        }

        if (contadorTop10 <= 10)
        {
            resultado +=
"_____ \n" + contadorTop10
+ String.Format(Environment.NewLine) + "Municipio: " +
NodoDelMunicipioDelArbol.NombreDelNodoMunicipioDelArbol +
String.Format(Environment.NewLine) + "Distancia en km: " +
NodoDelMunicipioDelArbol.DistanciasALaCabeceraDelArbol +
String.Format(Environment.NewLine);
        }

        contadorTop10++;

        if (NodoDelMunicipioDelArbol.liga Derecha != null)
        {
            LugaresMasCercanosALaCapital(NodoDelMunicipioDelArbol.liga Derecha);
        }

    } //fin top 10 lugares mas cercanos a la capital

    //Top 10 lugares mas lejanos a la capital

    public void LugaresMasLejanosALaCapitalCorredor(NodoDelMunicipioDelArbol
NodoDelMunicipioDelArbol)
    {

        if (NodoDelMunicipioDelArbol.ligal Izquierda != null)

```

```

    {
        LugaresMasLejanosALaCapitalCorredor(NodoDelMunicipioDelArbol.ligaIzquierda);
    }

    resultado += NodoDelMunicipioDelArbol.NombreDelNodoMunicipioDelArbol + ",";

    if (NodoDelMunicipioDelArbol.ligaDerecha != null)
    {
        LugaresMasLejanosALaCapitalCorredor(NodoDelMunicipioDelArbol.ligaDerecha);
    }

}

//fin top 10 lugares mas lejanos a la capital

public void mostrarLosMasLejanos()
{
    string resultadosMasLejanos = resultado;
    resultado = "";
    for (int i = resultadosMasLejanos.Split(",").Length - 2; i >= 0; i--)
    {
        if (contadorTop10 <= 10)
        {
            resultado +=
"_____ \n" + contadorTop10
+ "\n" + resultadosMasLejanos.Split(",")[i] + "\n";
        }
        contadorTop10++;
    }
}

//para devolver la distancia entre departamentos sin importar la ruta

```

```

    public void MostrarDistanciaEntreDepartamentosSinImportarRuta(String
nombreDelNodoABuscar, String nombreDelNodoABuscar2, NodoDelMunicipioDelArbol
NodoDelMunicipioDelArbol)

    {

        if (NodoDelMunicipioDelArbol.NombreDelNodoMunicipioDelArbol ==
nombreDelNodoABuscar && bandera == false)

        {

            bandera = true;

            primeraDistancia = NodoDelMunicipioDelArbol.DistanciasALaCabeceraDelArbol;

        }

        if (NodoDelMunicipioDelArbol.NombreDelNodoMunicipioDelArbol ==
nombreDelNodoABuscar2)

        {

            segundaDistancia = NodoDelMunicipioDelArbol.DistanciasALaCabeceraDelArbol;

            calculoDeDistancia = primeraDistancia - segundaDistancia;

            if (calculoDeDistancia < 0)

            {

                calculoDeDistancia = calculoDeDistancia * -1;

            }

            resultado = Convert.ToString(calculoDeDistancia);

        }

        if (NodoDelMunicipioDelArbol.ligalIzquierda != null)

        {

            MostrarDistanciaEntreDepartamentosSinImportarRuta(nombreDelNodoABuscar,
nombreDelNodoABuscar2, NodoDelMunicipioDelArbol.ligalIzquierda);

        }

        if (NodoDelMunicipioDelArbol.ligaDerecha != null)

        {

```

```
        MostrarDistanciaEntreDepartamentosSinImportarRuta(nombreDelNodoABuscar,
nombreDelNodoABuscar2, NodoDelMunicipioDelArbol.ligaDerecha);
    }
}
```

```
}//fin para devolver la distancia entre departamentos sin importar la ruta
```

```
string nombreDelDepartamentoOrigen = "";
```

```
//para devolver el departamento al que pertenece el municipio
```

```
public void ConocerLaCabeceraDeUnMunicipio(String nombreDelNodoABuscar,
NodoDelMunicipioDelArbol NodoDelMunicipioDelArbol)
```

```
{
```

```
    if (NodoDelMunicipioDelArbol.DistanciasALaCabeceraDelArbol == 0)
```

```
    {
```

```
        nombreDelDepartamentoOrigen =
NodoDelMunicipioDelArbol.NombreDelNodoMunicipioDelArbol;
```

```
    }
```

```
    if (NodoDelMunicipioDelArbol.NombreDelNodoMunicipioDelArbol ==
nombreDelNodoABuscar)
```

```
    {
```

```
        resultado = "";
```

```
        resultado += "_____ " +
String.Format(Environment.NewLine);
```

```
        resultado += "Se ha calculado que pertenece al departamento de " +
nombreDelDepartamentoOrigen + String.Format(Environment.NewLine) + "Municipio: " +
NodoDelMunicipioDelArbol.NombreDelNodoMunicipioDelArbol +
String.Format(Environment.NewLine) + "Distancia a la cabecera en km: " +
NodoDelMunicipioDelArbol.DistanciasALaCabeceraDelArbol +
String.Format(Environment.NewLine);
```



```

    }

    if (NodoDelMunicipioDelArbol.ligalZquierda != null)
    {
        ConocerLaCabeceraDeUnMunicipio(nombreDelNodoABuscar,
        NodoDelMunicipioDelArbol.ligalZquierda);
    }

    if (NodoDelMunicipioDelArbol.ligaDerecha != null)
    {
        ConocerLaCabeceraDeUnMunicipio(nombreDelNodoABuscar,
        NodoDelMunicipioDelArbol.ligaDerecha);
    }

    } //fin para devolver el departamento al que pertenece el municipio

} //Fin de la class ArbolDelDepartamento
//Fin de la programacion orientada a objetos

public class DatosDeDepartamentos
{
    public int IdDepartamento { get; set; }

    public string NombreDelDepartamento { get; set; }

    public int DistanciaQueExisteEntreCiudadCapital { get; set; }

    public int CantidadPoblacional { get; set; }

    public int CantidadDeMunicipios { get; set; }
}

```

```

public string LugaresTuristicosMasVisitados { get; set; }

}

/*Creamos 4 instancias que simulan las 4 rutas*/
ModeloDeRutasDoblementeLigada LaPrimeraRuta = new ModeloDeRutasDoblementeLigada();
ModeloDeRutasDoblementeLigada LaSegundaRuta = new
ModeloDeRutasDoblementeLigada();
ModeloDeRutasDoblementeLigada LaTerceraRutaRuta = new
ModeloDeRutasDoblementeLigada();
ModeloDeRutasDoblementeLigada LaCuartaRuta = new ModeloDeRutasDoblementeLigada();

/*El siguiente segmento de codigo hace que los nodos raiz sean accesibles en cualquier parte
en el programa*/
/*Cada uno representa un departamento*/
NodoDelMunicipioDelArbol LaRaizDelDepartamentoDeAltaVerapaz;
NodoDelMunicipioDelArbol LaRaizDelDepartamentoDeBajaVerapaz;
NodoDelMunicipioDelArbol LaRaizDelDepartamentoDeChimaltenango;
NodoDelMunicipioDelArbol LaRaizDelDepartamentoDeChiquimula;
NodoDelMunicipioDelArbol LaRaizDelDepartamentoDeGuatemala;
NodoDelMunicipioDelArbol LaRaizDelDepartamentoDeElProgreso;
NodoDelMunicipioDelArbol LaRaizDelDepartamentoDeEscuintla;
NodoDelMunicipioDelArbol LaRaizDelDepartamentoDeHuehuetenango;
NodoDelMunicipioDelArbol LaRaizDelDepartamentoDeIzabal;
NodoDelMunicipioDelArbol LaRaizDelDepartamentoDeJalapa;
NodoDelMunicipioDelArbol LaRaizDelDepartamentoDeJutiapa;
NodoDelMunicipioDelArbol LaRaizDelDepartamentoDePeten;
NodoDelMunicipioDelArbol LaRaizDelDepartamentoDeQuetzaltenango;

```

NodoDelMunicipioDelArbol LaRaizDelDepartamentoDeQuiche;
NodoDelMunicipioDelArbol LaRaizDelDepartamentoDeRetalhuleu;
NodoDelMunicipioDelArbol LaRaizDelDepartamentoDeSacatepequez;
NodoDelMunicipioDelArbol LaRaizDelDepartamentoDeSanMarcos;
NodoDelMunicipioDelArbol LaRaizDelDepartamentoDeSantaRosa;
NodoDelMunicipioDelArbol LaRaizDelDepartamentoDeSolola;
NodoDelMunicipioDelArbol LaRaizDelDepartamentoDeSuchitepequez;
NodoDelMunicipioDelArbol LaRaizDelDepartamentoDeTotonicapan;
NodoDelMunicipioDelArbol LaRaizDelDepartamentoDeZacapa;
NodoDelMunicipioDelArbol LaRaizDelArbolNacional;

/*A continuacion necesitamos crear los arboles necesarios para cada departamento*/

/*Cada instancia representa el arbol interno de cada departamento permitiendo organizar los municipios de cada uno de ellos*/

/*Ademas permite crear una ruta hacia otro departamento*/

ArbolDelDepartamento ObjetoInstanciadoDeAltaVerapaz = new ArbolDelDepartamento();
ArbolDelDepartamento ObjetoInstanciadoDeBajaVerapaz = new ArbolDelDepartamento();
ArbolDelDepartamento ObjetoInstanciadoDeChimaltenango = new ArbolDelDepartamento();
ArbolDelDepartamento ObjetoInstanciadoDeChiquimula = new ArbolDelDepartamento();
ArbolDelDepartamento ObjetoInstanciadoDeGuatemala = new ArbolDelDepartamento();
ArbolDelDepartamento ObjetoInstanciadoDeElProgreso = new ArbolDelDepartamento();
ArbolDelDepartamento ObjetoInstanciadoDeEscuintla = new ArbolDelDepartamento();
ArbolDelDepartamento ObjetoInstanciadoDeHuehuetenango = new ArbolDelDepartamento();
ArbolDelDepartamento ObjetoInstanciadoDeIzabal = new ArbolDelDepartamento();
ArbolDelDepartamento ObjetoInstanciadoDeJalapa = new ArbolDelDepartamento();
ArbolDelDepartamento ObjetoInstanciadoDeJutiapa = new ArbolDelDepartamento();
ArbolDelDepartamento ObjetoInstanciadoDePeten = new ArbolDelDepartamento();
ArbolDelDepartamento ObjetoInstanciadoDeQuetzaltenango = new ArbolDelDepartamento();
ArbolDelDepartamento ObjetoInstanciadoDeQuiche = new ArbolDelDepartamento();

```

ArbolDelDepartamento ObjetoInstanciadoDeRetalhuleu = new ArbolDelDepartamento();
ArbolDelDepartamento ObjetoInstanciadoDeSacatepequez = new ArbolDelDepartamento();
ArbolDelDepartamento ObjetoInstanciadoDeSanMarcos = new ArbolDelDepartamento();
ArbolDelDepartamento ObjetoInstanciadoDeSantaRosa = new ArbolDelDepartamento();
ArbolDelDepartamento ObjetoInstanciadoDeSolola = new ArbolDelDepartamento();
ArbolDelDepartamento ObjetoInstanciadoDeSuchitepequez = new ArbolDelDepartamento();
ArbolDelDepartamento ObjetoInstanciadoDeTotonacapan = new ArbolDelDepartamento();
ArbolDelDepartamento ObjetoInstanciadoDeZacapa = new ArbolDelDepartamento();

//Creamos la instancia para el nodo raiz que conecta a todos los departamentos, es la ciudad
capital

ArbolDelDepartamento ObjetoInstanciaArbolNacional = new ArbolDelDepartamento();

/*Las siguiente variables permite almacenar temporalmente datos que se utilizar para calcular
distancias o parametros de consultas*/

int estaVariableEsParaCalcularDistancias;

string nombreDelDepartamentoParaTransaccion;
string nombreDepartamentoDelInicioParaUnViaje;
string nombreDepartamentoDelFinalParaUnViaje;
string nombreDeDepartamentoParaBuscar;
string ElNombreDelMunicipio;

Nodo ElNodoOrigenCiudad;

DatosDeDepartamentos[] departamentosExtraidos;

public HttpClient httpClient = new HttpClient();

protected async Task ExtraerDatosDepartamentosDeLaApi()

{

```

```
departamentosExtraidos = await  
httpClient.GetFromJsonAsync<DatosDeDepartamentos[]>("https://localhost:7104/api/Departamen  
tos");
```

```
/*Es importante crear el nodo de partida*/
```

```
/*El nodo origen cuenta con 4 rutas*/
```

```
/*Creamos la instancia del nodo central*/
```

```
ElNodoOrigenCiudad = new Nodo(departamentosExtraidos[0].NombreDelDepartamento,  
departamentosExtraidos[0].DistanciaQueExisteEntreCiudadCapital,  
departamentosExtraidos[0].CantidadPoblacional,  
departamentosExtraidos[0].CantidadDeMunicipios,  
departamentosExtraidos[0].LugaresTuristicosMasVisitados);
```

```
ConfigurarRutasIniciales();
```

```
DesplegarLaRutaCompleta();
```

```
}
```

```
protected override void OnInitialized() // = On Page Load
```

```
{
```

```
LaRutaSeleccionada = "1";
```

```
DepartamentoDelInicioParaTransaccion = "";
```

```
DepartamentoDeFinalizacionParaTransaccion = "";
```

```
distanciaDesdeCiudad = "";
```

```
ExtraerDatosDepartamentosDeLaApi();
```

```
}
```

```

private void ConfigurarRutasIniciales()
{

    //En este segmento es donde se configuran las rutas iniciales

    //En este momento debemos insertar el primer nodo del arbol, donde cada arbol pertenece a
    un departamento especifico

    //creamos el nodo raiz para cada arbol para su respectivo departamento, este representa la
    cabecera de dicho departamento

    LaRaizDelDepartamentoDeAltaVerapaz =
    ObjetoInstanciadoDeAltaVerapaz.InsertarNodo(departamentosExtraidos[1].NombreDelDepartame
    nto, 0, null, "");

    LaRaizDelDepartamentoDeBajaVerapaz =
    ObjetoInstanciadoDeBajaVerapaz.InsertarNodo(departamentosExtraidos[2].NombreDelDepartame
    nto, 0, null, "");

    LaRaizDelDepartamentoDeChimaltenango =
    ObjetoInstanciadoDeChimaltenango.InsertarNodo(departamentosExtraidos[3].NombreDelDeparta
    mento, 0, null, "");

    LaRaizDelDepartamentoDeChiquimula =
    ObjetoInstanciadoDeChiquimula.InsertarNodo(departamentosExtraidos[4].NombreDelDepartame
    nto, 0, null, "");

    LaRaizDelDepartamentoDeGuatemala =
    ObjetoInstanciadoDeGuatemala.InsertarNodo(departamentosExtraidos[0].NombreDelDepartamen
    to, 0, null, "");

    LaRaizDelDepartamentoDeElProgreso =
    ObjetoInstanciadoDeElProgreso.InsertarNodo(departamentosExtraidos[5].NombreDelDepartamen
    to, 0, null, "");

    LaRaizDelDepartamentoDeEscuintla =
    ObjetoInstanciadoDeEscuintla.InsertarNodo(departamentosExtraidos[6].NombreDelDepartamento
    , 0, null, "");

    LaRaizDelDepartamentoDeHuehuetenango =
    ObjetoInstanciadoDeHuehuetenango.InsertarNodo(departamentosExtraidos[7].NombreDelDepart
    amento, 0, null, "");

    LaRaizDelDepartamentoDelzabal =
    ObjetoInstanciadoDelzabal.InsertarNodo(departamentosExtraidos[8].NombreDelDepartamento, 0,
    null, "");

```

```
LaRaizDelDepartamentoDeJalapa =  
ObjetoInstanciadoDeJalapa.InsertarNodo(departamentosExtraidos[9].NombreDelDepartamento,  
0, null, "");
```

```
LaRaizDelDepartamentoDeJutiapa =  
ObjetoInstanciadoDeJutiapa.InsertarNodo(departamentosExtraidos[10].NombreDelDepartamento,  
0, null, "");
```

```
LaRaizDelDepartamentoDePeten =  
ObjetoInstanciadoDePeten.InsertarNodo(departamentosExtraidos[11].NombreDelDepartamento,  
0, null, "");
```

```
LaRaizDelDepartamentoDeQuetzaltenango =  
ObjetoInstanciadoDeQuetzaltenango.InsertarNodo(departamentosExtraidos[12].NombreDelDepartamento,  
0, null, "");
```

```
LaRaizDelDepartamentoDeQuiche =  
ObjetoInstanciadoDeQuiche.InsertarNodo(departamentosExtraidos[13].NombreDelDepartamento,  
0, null, "");
```

```
LaRaizDelDepartamentoDeRetalhuleu =  
ObjetoInstanciadoDeRetalhuleu.InsertarNodo(departamentosExtraidos[14].NombreDelDepartamento,  
0, null, "");
```

```
LaRaizDelDepartamentoDeSacatepequez =  
ObjetoInstanciadoDeSacatepequez.InsertarNodo(departamentosExtraidos[15].NombreDelDepartamento,  
0, null, "");
```

```
LaRaizDelDepartamentoDeSanMarcos =  
ObjetoInstanciadoDeSanMarcos.InsertarNodo(departamentosExtraidos[16].NombreDelDepartamento,  
0, null, "");
```

```
LaRaizDelDepartamentoDeSantaRosa =  
ObjetoInstanciadoDeSantaRosa.InsertarNodo(departamentosExtraidos[17].NombreDelDepartamento,  
0, null, "");
```

```
LaRaizDelDepartamentoDeSolola =  
ObjetoInstanciadoDeSolola.InsertarNodo(departamentosExtraidos[18].NombreDelDepartamento,  
0, null, "");
```

```
LaRaizDelDepartamentoDeSuchitepequez =  
ObjetoInstanciadoDeSuchitepequez.InsertarNodo(departamentosExtraidos[19].NombreDelDepartamento,  
0, null, "");
```

```
LaRaizDelDepartamentoDeTotonicapan =  
ObjetoInstanciadoDeTotonicapan.InsertarNodo(departamentosExtraidos[20].NombreDelDepartamento,  
0, null, "");
```

```
LaRaizDelDepartamentoDeZacapa =  
ObjetoInstanciadoDeZacapa.InsertarNodo(departamentosExtraidos[21].NombreDelDepartamento,  
0, null, "");
```

```
//Luego necesitamos configurar el arbol nacional esto quiere decir que se creara un arbol  
donde se guardaran los departamento de guatemala
```

```
//Esto permitir insertar a la izquierda o a la derecha respecto a su distancio con la ciudad  
capital
```

```
//Configuramos el arbol nacional de todos los departamentos el nodo raiz es la ciudad de  
guatemala
```

```
LaRaizDelArbolNacional =  
ObjetoInstanciaArbolNacional.InsertarNodo(departamentosExtraidos[0].NombreDelDepartamento  
, 0, null, "");
```

```
ObjetoInstanciaArbolNacional.InsertarNodo(departamentosExtraidos[1].NombreDelDepartamento  
, departamentosExtraidos[1].DistanciaQueExisteEntreCiudadCapital, LaRaizDelArbolNacional, "");
```

```
ObjetoInstanciaArbolNacional.InsertarNodo(departamentosExtraidos[2].NombreDelDepartamento  
, departamentosExtraidos[2].DistanciaQueExisteEntreCiudadCapital, LaRaizDelArbolNacional, "");
```

```
ObjetoInstanciaArbolNacional.InsertarNodo(departamentosExtraidos[3].NombreDelDepartamento  
, departamentosExtraidos[3].DistanciaQueExisteEntreCiudadCapital, LaRaizDelArbolNacional, "");
```

```
ObjetoInstanciaArbolNacional.InsertarNodo(departamentosExtraidos[4].NombreDelDepartamento  
, departamentosExtraidos[4].DistanciaQueExisteEntreCiudadCapital, LaRaizDelArbolNacional, "");
```

```
ObjetoInstanciaArbolNacional.InsertarNodo(departamentosExtraidos[5].NombreDelDepartamento  
, departamentosExtraidos[5].DistanciaQueExisteEntreCiudadCapital, LaRaizDelArbolNacional, "");
```

```
ObjetoInstanciaArbolNacional.InsertarNodo(departamentosExtraidos[6].NombreDelDepartamento  
, departamentosExtraidos[6].DistanciaQueExisteEntreCiudadCapital, LaRaizDelArbolNacional, "");
```

```
ObjetoInstanciaArbolNacional.InsertarNodo(departamentosExtraidos[7].NombreDelDepartamento  
, departamentosExtraidos[7].DistanciaQueExisteEntreCiudadCapital, LaRaizDelArbolNacional, "");
```

```
ObjetoInstanciaArbolNacional.InsertarNodo(departamentosExtraidos[8].NombreDelDepartamento  
, departamentosExtraidos[8].DistanciaQueExisteEntreCiudadCapital, LaRaizDelArbolNacional, "");
```


ObjetoInstanciaArbolNacional.InsertarNodo(departamentosExtraidos[9].NombreDelDepartamento, departamentosExtraidos[9].DistanciaQueExisteEntreCiudadCapital, LaRaizDelArbolNacional, "");

ObjetoInstanciaArbolNacional.InsertarNodo(departamentosExtraidos[10].NombreDelDepartamento, departamentosExtraidos[10].DistanciaQueExisteEntreCiudadCapital, LaRaizDelArbolNacional, "");

ObjetoInstanciaArbolNacional.InsertarNodo(departamentosExtraidos[11].NombreDelDepartamento, departamentosExtraidos[11].DistanciaQueExisteEntreCiudadCapital, LaRaizDelArbolNacional, "");

ObjetoInstanciaArbolNacional.InsertarNodo(departamentosExtraidos[12].NombreDelDepartamento, departamentosExtraidos[12].DistanciaQueExisteEntreCiudadCapital, LaRaizDelArbolNacional, "");

ObjetoInstanciaArbolNacional.InsertarNodo(departamentosExtraidos[13].NombreDelDepartamento, departamentosExtraidos[13].DistanciaQueExisteEntreCiudadCapital, LaRaizDelArbolNacional, "");

ObjetoInstanciaArbolNacional.InsertarNodo(departamentosExtraidos[14].NombreDelDepartamento, departamentosExtraidos[14].DistanciaQueExisteEntreCiudadCapital, LaRaizDelArbolNacional, "");

ObjetoInstanciaArbolNacional.InsertarNodo(departamentosExtraidos[15].NombreDelDepartamento, departamentosExtraidos[15].DistanciaQueExisteEntreCiudadCapital, LaRaizDelArbolNacional, "");

ObjetoInstanciaArbolNacional.InsertarNodo(departamentosExtraidos[16].NombreDelDepartamento, departamentosExtraidos[16].DistanciaQueExisteEntreCiudadCapital, LaRaizDelArbolNacional, "");

ObjetoInstanciaArbolNacional.InsertarNodo(departamentosExtraidos[17].NombreDelDepartamento, departamentosExtraidos[17].DistanciaQueExisteEntreCiudadCapital, LaRaizDelArbolNacional, "");

ObjetoInstanciaArbolNacional.InsertarNodo(departamentosExtraidos[18].NombreDelDepartamento, departamentosExtraidos[18].DistanciaQueExisteEntreCiudadCapital, LaRaizDelArbolNacional, "");

```
ObjetoInstanciaArbolNacional.InsertarNodo(departamentosExtraidos[19].NombreDelDepartamento, departamentosExtraidos[19].DistanciaQueExisteEntreCiudadCapital, LaRaizDelArbolNacional, "");
```

```
ObjetoInstanciaArbolNacional.InsertarNodo(departamentosExtraidos[20].NombreDelDepartamento, departamentosExtraidos[20].DistanciaQueExisteEntreCiudadCapital, LaRaizDelArbolNacional, "");
```

```
ObjetoInstanciaArbolNacional.InsertarNodo(departamentosExtraidos[21].NombreDelDepartamento, departamentosExtraidos[21].DistanciaQueExisteEntreCiudadCapital, LaRaizDelArbolNacional, "");
```

```
//configurando la primera ruta
```

```
LaPrimeraRuta.agregarDepartamentoAlPrincipioDeRuta("Escuintla", 1890, 14000, 13,"");
```

```
LaPrimeraRuta.agregarDepartamentoAlPrincipioDeRuta("Suchitepequez", 2177, 2100, 32,"");
```

```
LaPrimeraRuta.agregarDepartamentoAlPrincipioDeRuta("Retalhuleu", 5000, 11000, 38,"");
```

```
LaPrimeraRuta.agregarDepartamentoAlPrincipioDeRuta("San Marcos", 7000, 13000, 40,"");
```

```
//configurando la segunda ruta
```

```
LaSegundaRuta.agregarDepartamentoAlPrincipioDeRuta("Chimaltenango", 2890, 24000, 23,"");
```

```
LaSegundaRuta.agregarDepartamentoAlPrincipioDeRuta("Solola", 3147, 3140, 12,"");
```

```
LaSegundaRuta.agregarDepartamentoAlPrincipioDeRuta("Totonicapan", 3147, 3140, 12, "");
```

```
LaSegundaRuta.agregarDepartamentoAlPrincipioDeRuta("Huehuetenango", 3147, 3140, 12, "");
```

```
LaSegundaRuta.agregarDepartamentoAlPrincipioDeRuta("Quetzaltenango", 3147, 3140, 12, "");
```

```
//En la segunda ruta despues de solo existen 3 destinos las cuales se configuran a continuacion
```

```
ObjetoInstanciadoDeSolola.InsertarNodo("Totonicapan", 780, LaRaizDelDepartamentoDeSolola, ""); //Distancia a la cabecera 780
```

```
ObjetoInstanciadoDeSolola.InsertarNodo("Huehuetenango", 870, LaRaizDelDepartamentoDeSolola, ""); //Distancia a la cabecera 870
```

```
ObjetoInstanciadoDeSolola.InsertarNodo("Quetzaltenango", 540,  
LaRaizDelDepartamentoDeSolola, ""); //Distancia a la cabecera 540
```

```
//configurando la tercera ruta
```

```
LaTerceraRutaRuta.agregarDepartamentoAlPrincipioDeRuta("El Progreso", 5890, 34000,  
15, "");
```

```
//En la tercer ruta depues de el progreso existen 2 caminos las cuales se configuran a  
continuacion
```

```
ObjetoInstanciadoDeSolola.InsertarNodo("Baja Verapaz", 990,  
LaRaizDelDepartamentoDeSolola, ""); //Distancia a la cabecera 990
```

```
ObjetoInstanciadoDeSolola.InsertarNodo("Alta Verapaz", 810,  
LaRaizDelDepartamentoDeSolola, ""); //Distancia a la cabecera 810
```

```
ObjetoInstanciadoDeSolola.InsertarNodo("Zacapa", 980, LaRaizDelDepartamentoDeSolola,  
""); //Distancia a la cabecera 980
```

```
ObjetoInstanciadoDeSolola.InsertarNodo("Izabal", 890, LaRaizDelDepartamentoDeSolola,  
""); //Distancia a la cabecera 890
```

```
ObjetoInstanciadoDeSolola.InsertarNodo("Peten", 580, LaRaizDelDepartamentoDeSolola,  
""); //Distancia a la cabecera 580
```

```
//configurando la cuarta ruta
```

```
//Aqui se usa la misma distancia utilizada en el arbol nacional con la diferencia que ahora se  
crean rutas
```

```
LaCuartaRuta.agregarDepartamentoAlPrincipioDeRuta("Jalapa", 4590, 74000, 13, "");
```

```
LaCuartaRuta.agregarDepartamentoAlPrincipioDeRuta("Jutiapa", 3887, 8100, 32, "");
```

```
LaCuartaRuta.agregarDepartamentoAlPrincipioDeRuta("Santa Rosa", 3000, 45600, 38, "");
```

```
LaCuartaRuta.agregarDepartamentoAlPrincipioDeRuta("Chiquimula", 3000, 45600, 38, "");
```

```
//Jalapa tiene un destino hacia chiquimula
```

```
ObjetoInstanciadoDeJalapa.InsertarNodo("Chiquimula", 892,  
LaRaizDelDepartamentoDeSolola, ""); //Distancia a la cabecera 892
```

```
}
```

```
private string nombreDelDepartamento { get; set; }
```

```
private string distanciaDesdeCiudad { get; set; }
```

```
private string cantidadPoblacional { get; set; }
```

```
private string cantidadDeMunicipios { get; set; }
```

```
private string DepartamentoDelInicioParaTransaccion { get; set; }
```

```
private string DepartamentoDeFinalizacionParaTransaccion { get; set; }
```

```
private string DepartamentoDeAdministracion { get; set; }
```

```
private string miVariableParaMostrarResultadosDeLaRutaSeleccionada = "",  
miVariableParaMostrarResultados = "", LaRutaSeleccionada = "1", VariableDeLogDeErrores = "",  
miVariableParaMostrarResultadosDistanciaEntreDepartamentoyCapital = "",  
miVariableParaMostrarResultadosDeLugaresVisitadosPorUnViaje="",  
miVariableParaMostrarResultadosDeDistanciaEntreDepartamentos="";
```

```
private void SetearLaRuta(ChangeEventArgs e)
```

```
{
```

```
    LaRutaSeleccionada = (string)e.Value;
```

```
    DesplegarLaRutaCompleta();
```

```
}
```

```
private void SetearLaDepartamentoDeAdministracion(ChangeEventArgs e)
```

```
{
```

```
    DepartamentoDeAdministracion = (string)e.Value;
```

```
}
```

```

private void SetearLaDepartamentoDelInicio(ChangeEventArgs e)
{
    DepartamentoDelInicioParaTransaccion = (string)e.Value;
}

private void SetearLaDepartamentoDeFinalizacion(ChangeEventArgs e)
{
    DepartamentoDeFinalizacionParaTransaccion = (string)e.Value;
}

/*Funciones para administrar las turas que permiten insertar departamentos a la ruta*/
private void GuardarAlInicioDeLaRuta()
{
    if (LaRutaSeleccionada == "")
    {
        VariableDeLogDeErrores = "No ha seleccionado una ruta.";
    }
    else if (LaRutaSeleccionada == "1")
    {
        LaPrimeraRuta.agregarDepartamentoAlPrincipioDeRuta(nombreDelDepartamento,
        Convert.ToInt32(distanciaDesdeCiudad), Convert.ToInt32(cantidadPoblacional),
        Convert.ToInt32(cantidadDeMunicipios), "");

        ElNodoOrigenCiudad.siguiete = LaPrimeraRuta.devolverPrimerNodoDeLista();

        VariableDeLogDeErrores="Se guardo en la ruta 1";
    }

    else if (LaRutaSeleccionada == "2")
    {

```

```

        LaSegundaRuta.agregarDepartamentoAlPrincipioDeRuta(nombreDelDepartamento,
Convert.ToInt32(distanciaDesdeCiudad), Convert.ToInt32(cantidadPoblacional),
Convert.ToInt32(cantidadDeMunicipios),"");

        ElNodoOrigenCiudad.anterior = LaSegundaRuta.devolverPrimerNodoDeLista();

        VariableDeLogDeErrores="Se guardo en la ruta 2";

    }

    else if (LaRutaSeleccionada == "3")
    {

        LaTerceraRutaRuta.agregarDepartamentoAlPrincipioDeRuta(nombreDelDepartamento,
Convert.ToInt32(distanciaDesdeCiudad), Convert.ToInt32(cantidadPoblacional),
Convert.ToInt32(cantidadDeMunicipios),"");

        ElNodoOrigenCiudad.siguienteSecundaria =
LaTerceraRutaRuta.devolverPrimerNodoDeLista();

        VariableDeLogDeErrores="Se guardo en la ruta 3";

    }

    else if (LaRutaSeleccionada == "4")
    {

        LaCuartaRuta.agregarDepartamentoAlPrincipioDeRuta(nombreDelDepartamento,
Convert.ToInt32(distanciaDesdeCiudad), Convert.ToInt32(cantidadPoblacional),
Convert.ToInt32(cantidadDeMunicipios),"");

        ElNodoOrigenCiudad.anteriorSecundaria = LaCuartaRuta.devolverPrimerNodoDeLista();

        VariableDeLogDeErrores="Se guardo en la ruta 4";

    }

    else if (LaRutaSeleccionada.Length < 1)
    {

        VariableDeLogDeErrores="No ha seleccionado la ruta.";

    }

    MostrarResultados();

}

```

```

private void GuardarAlFinalDeLaRuta()
{
    if (LaRutaSeleccionada == "")
    {
        VariableDeLogDeErrores = "No ha seleccionado una ruta.";
    }
    else if (LaRutaSeleccionada == "1")
    {
        LaPrimeraRuta.agregarDepartamentoAlFinalDeRuta(nombreDelDepartamento,
        Convert.ToInt32(distanciaDesdeCiudad), Convert.ToInt32(cantidadPoblacional),
        Convert.ToInt32(cantidadDeMunicipios), "");

        ElNodoOrigenCiudad.siguiente = LaPrimeraRuta.devolverPrimerNodoDeLista();
        VariableDeLogDeErrores = "Se guardo en la ruta 1";
    }

    else if (LaRutaSeleccionada == "2")
    {
        LaSegundaRuta.agregarDepartamentoAlFinalDeRuta(nombreDelDepartamento,
        Convert.ToInt32(distanciaDesdeCiudad), Convert.ToInt32(cantidadPoblacional),
        Convert.ToInt32(cantidadDeMunicipios), "");

        ElNodoOrigenCiudad.anterior = LaSegundaRuta.devolverPrimerNodoDeLista();
        VariableDeLogDeErrores = "Se guardo en la ruta 2";
    }

    else if (LaRutaSeleccionada == "3")
    {
        LaTerceraRutaRuta.agregarDepartamentoAlFinalDeRuta(nombreDelDepartamento,
        Convert.ToInt32(distanciaDesdeCiudad), Convert.ToInt32(cantidadPoblacional),
        Convert.ToInt32(cantidadDeMunicipios), "");

        ElNodoOrigenCiudad.siguienteSecundaria =
        LaTerceraRutaRuta.devolverPrimerNodoDeLista();
    }
}

```

```

        VariableDeLogDeErrores = "Se guardo en la ruta 3";
    }

    else if (LaRutaSeleccionada == "4")
    {
        LaCuartaRuta.agregarDepartamentoAlFinalDeRuta(nombreDelDepartamento,
        Convert.ToInt32(distanciaDesdeCiudad), Convert.ToInt32(cantidadPoblacional),
        Convert.ToInt32(cantidadDeMunicipios), "");

        ElNodoOrigenCiudad.anteriorSecundaria = LaCuartaRuta.devolverPrimerNodoDeLista();

        VariableDeLogDeErrores = "Se guardo en la ruta 4";
    }

    else if (LaRutaSeleccionada.Length < 1)
    {
        VariableDeLogDeErrores = "No ha seleccionado la ruta.";
    }

    MostrarResultados();
}

private void GuardarAntesDeUnDepartamentoDeLaRuta()
{
    if (LaRutaSeleccionada == "")
    {
        VariableDeLogDeErrores = "No ha seleccionado una ruta.";
    }

    else if (LaRutaSeleccionada == "1")
    {
        LaPrimeraRuta.agregarDepartamentoAntesDeOtro(nombreDelDepartamento,
        Convert.ToInt32(distanciaDesdeCiudad), Convert.ToInt32(cantidadPoblacional),
        Convert.ToInt32(cantidadDeMunicipios), LaPrimeraRuta.buscar(nombreDelDepartamento), "");

        ElNodoOrigenCiudad.siguiete = LaPrimeraRuta.devolverPrimerNodoDeLista();

        VariableDeLogDeErrores = "Se guardo en la ruta 1";
    }
}

```



```

    }

    else if (LaRutaSeleccionada == "2")
    {
        LaSegundaRuta.agregarDepartamentoAntesDeOtro(nombreDelDepartamento,
        Convert.ToInt32(distanciaDesdeCiudad), Convert.ToInt32(cantidadPoblacional),
        Convert.ToInt32(cantidadDeMunicipios), LaSegundaRuta.buscar(nombreDelDepartamento), "");

        ElNodoOrigenCiudad.anterior = LaSegundaRuta.devolverPrimerNodoDeLista();

        VariableDeLogDeErrores = "Se guardo en la ruta 2";
    }

    else if (LaRutaSeleccionada == "3")
    {
        LaTerceraRutaRuta.agregarDepartamentoAntesDeOtro(nombreDelDepartamento,
        Convert.ToInt32(distanciaDesdeCiudad), Convert.ToInt32(cantidadPoblacional),
        Convert.ToInt32(cantidadDeMunicipios), LaTerceraRutaRuta.buscar(nombreDelDepartamento), "");

        ElNodoOrigenCiudad.siguienteSecundaria =
        LaTerceraRutaRuta.devolverPrimerNodoDeLista();

        VariableDeLogDeErrores = "Se guardo en la ruta 3";
    }

    else if (LaRutaSeleccionada == "4")
    {
        LaCuartaRuta.agregarDepartamentoAntesDeOtro(nombreDelDepartamento,
        Convert.ToInt32(distanciaDesdeCiudad), Convert.ToInt32(cantidadPoblacional),
        Convert.ToInt32(cantidadDeMunicipios), LaCuartaRuta.buscar(nombreDelDepartamento), "");

        ElNodoOrigenCiudad.anteriorSecundaria = LaCuartaRuta.devolverPrimerNodoDeLista();

        VariableDeLogDeErrores = "Se guardo en la ruta 4";
    }

    else if (LaRutaSeleccionada.Length < 1)
    {
        VariableDeLogDeErrores = "No ha seleccionado la ruta.";
    }

    MostrarResultados();

```

```

}

private void GuardarDespuesDeUnDepartamentoDeLaRuta()
{
    if (LaRutaSeleccionada == "")
    {
        VariableDeLogDeErrores = "No ha seleccionado una ruta.";
    }

    else if (LaRutaSeleccionada == "1")
    {
        LaPrimeraRuta.agregarDepartamentoDespuesDeOtro(nombreDelDepartamento.ToString(),
        Convert.ToInt32(distanciaDesdeCiudad), Convert.ToInt32(cantidadPoblacional),
        Convert.ToInt32(cantidadDeMunicipios),
        LaPrimeraRuta.buscar(nombreDelDepartamento.ToString()), "");

        ElNodoOrigenCiudad.siguiente = LaPrimeraRuta.devolverPrimerNodoDeLista();

        VariableDeLogDeErrores = "Se guardo en la ruta 1";
    }

    else if (LaRutaSeleccionada == "2")
    {
        LaPrimeraRuta.agregarDepartamentoDespuesDeOtro(nombreDelDepartamento.ToString(),
        Convert.ToInt32(distanciaDesdeCiudad), Convert.ToInt32(cantidadPoblacional),
        Convert.ToInt32(cantidadDeMunicipios),
        LaPrimeraRuta.buscar(nombreDelDepartamento.ToString()), "");

        ElNodoOrigenCiudad.anterior = LaSegundaRuta.devolverPrimerNodoDeLista();

        VariableDeLogDeErrores = "Se guardo en la ruta 2";
    }

    else if (LaRutaSeleccionada == "3")
    {
        LaPrimeraRuta.agregarDepartamentoDespuesDeOtro(nombreDelDepartamento.ToString(),
        Convert.ToInt32(distanciaDesdeCiudad), Convert.ToInt32(cantidadPoblacional),
        Convert.ToInt32(cantidadDeMunicipios),
        LaPrimeraRuta.buscar(nombreDelDepartamento.ToString()), "");
    }
}

```

```

        ElNodoOrigenCiudad.siguienteSecundaria =
LaTerceraRutaRuta.devolverPrimerNodoDeLista();

        VariableDeLogDeErrores = "Se guardo en la ruta 3";

    }

    else if (LaRutaSeleccionada == "4")

    {

        LaPrimeraRuta.agregarDepartamentoDespuesDeOtro(nombreDelDepartamento.ToString(),
Convert.ToInt32(distanciaDesdeCiudad), Convert.ToInt32(cantidadPoblacional),
Convert.ToInt32(cantidadDeMunicipios),
LaPrimeraRuta.buscar(nombreDelDepartamento.ToString()), "");

        ElNodoOrigenCiudad.anteriorSecundaria = LaCuartaRuta.devolverPrimerNodoDeLista();

        VariableDeLogDeErrores = "Se guardo en la ruta 4";

    }

    else if (LaRutaSeleccionada.Length < 1)

    {

        VariableDeLogDeErrores = "No ha seleccionado la ruta.";

    }

    MostrarResultados();

}

/*Fin de las funciones para administrar las turas que permiten insertar departamentos a la
ruta*/

```

```

/*Funciones para la eliminacion de departamentos de una ruta*/

```

```

private void EliminarElPrimerDestinoDeLaRuta()

{

    try

    {

        if (LaRutaSeleccionada == "")

        {

            VariableDeLogDeErrores = "No ha seleccionado una ruta.";

        }

    }

}

```

```

else if (LaRutaSeleccionada == "1")
{
    LaPrimeraRuta.eliminarInicioDeRuta();
    ElNodoOrigenCiudad.siguiete = LaPrimeraRuta.devolverPrimerNodoDeLista();
}
else if (LaRutaSeleccionada == "2")
{
    LaSegundaRuta.eliminarInicioDeRuta();
    ElNodoOrigenCiudad.anterior = LaSegundaRuta.devolverPrimerNodoDeLista();
}
else if (LaRutaSeleccionada == "3")
{
    LaTerceraRutaRuta.eliminarInicioDeRuta();
    ElNodoOrigenCiudad.siguieteSecundaria =
LaTerceraRutaRuta.devolverPrimerNodoDeLista();
}
else if (LaRutaSeleccionada == "4")
{
    LaCuartaRuta.eliminarInicioDeRuta();
    ElNodoOrigenCiudad.anteriorSecundaria = LaCuartaRuta.devolverPrimerNodoDeLista();
}
MostrarResultados();
}
catch
{
    VariableDeLogDeErrores = "La distancia que existe entre el departamento y el municipio es
de: " + estaVariableEsParaCalcularDistancias;
}
}

```

```

private void EliminarElUltimoDestinoDeLaRuta()
{
    try
    {
        if (LaRutaSeleccionada == "")
        {
            VariableDeLogDeErrores = "No ha seleccionado una ruta.";
        }
        else if (LaRutaSeleccionada == "1")
        {
            LaPrimeraRuta.eliminarFinalDeRuta();
            ElNodoOrigenCiudad.siguiente = LaPrimeraRuta.devolverPrimerNodoDeLista();
        }
        else if (LaRutaSeleccionada == "2")
        {
            LaSegundaRuta.eliminarFinalDeRuta();
            ElNodoOrigenCiudad.anterior = LaSegundaRuta.devolverPrimerNodoDeLista();
        }
        else if (LaRutaSeleccionada == "3")
        {
            LaTerceraRutaRuta.eliminarFinalDeRuta();
            ElNodoOrigenCiudad.siguienteSecundaria =
LaTerceraRutaRuta.devolverPrimerNodoDeLista();
        }
        else if (LaRutaSeleccionada == "4")
        {
            LaCuartaRuta.eliminarFinalDeRuta();
            ElNodoOrigenCiudad.anteriorSecundaria = LaCuartaRuta.devolverPrimerNodoDeLista();
        }
    }
}

```

```

        MostrarResultados();
    }
    catch
    {
        VariableDeLogDeErrores = "La distancia que existe entre el departamento y el municipio es de: " + estaVariableEsParaCalcularDistancias;
    }
}

```

```

private void EliminarDepartamentoEspecificoEnUnaRuta()
{
    try
    {
        if (LaRutaSeleccionada == "")
        {
            VariableDeLogDeErrores = "No ha seleccionado una ruta.";
        }
        else if (LaRutaSeleccionada == "1")
        {
            LaPrimeraRuta.eliminar(LaPrimeraRuta.buscar(DepartamentoDeAdministracion));
            ElNodoOrigenCiudad.siguiente = LaPrimeraRuta.devolverPrimerNodoDeLista();
        }

        else if (LaRutaSeleccionada == "2")
        {
            LaSegundaRuta.eliminar(LaSegundaRuta.buscar(DepartamentoDeAdministracion));
            ElNodoOrigenCiudad.anterior = LaSegundaRuta.devolverPrimerNodoDeLista();
        }

        else if (LaRutaSeleccionada == "3")
    }
}

```

```

        {
            LaTerceraRutaRuta.eliminar(LaTerceraRutaRuta.buscar(DepartamentoDeAdministracion));

            ElNodoOrigenCiudad.siguienteSecundaria =
LaTerceraRutaRuta.devolverPrimerNodoDeLista();
        }
        else if (LaRutaSeleccionada == "4")
        {
            LaCuartaRuta.eliminar(LaCuartaRuta.buscar(DepartamentoDeAdministracion));

            ElNodoOrigenCiudad.anteriorSecundaria = LaCuartaRuta.devolverPrimerNodoDeLista();
        }
        MostrarResultados();

    }
    catch
    {
        VariableDeLogDeErrores = "No es valida la eliminacion.";
    }
}

/*Fin de las funciones para la eliminacion de departamentos de una ruta*/

/*Funciones para generar reportes*/
private void DesplegarLaRutaCompleta()
{
    if (LaRutaSeleccionada == "")
    {
        VariableDeLogDeErrores = "No ha seleccionado una ruta.";
    }
    else if (LaRutaSeleccionada == "1")

```

```

{
    LaPrimeraRuta.generarReporteDeRuta();

    ElNodoOrigenCiudad.siguiente = LaPrimeraRuta.devolverPrimerNodoDeLista();

    miVariableParaMostrarResultadosDeLaRutaSeleccionada = LaPrimeraRuta.resultado;

    LaPrimeraRuta.limpiarResultado();
}

else if (LaRutaSeleccionada == "2")
{
    LaSegundaRuta.generarReporteDeRuta();

    ElNodoOrigenCiudad.anterior = LaSegundaRuta.devolverPrimerNodoDeLista();

    miVariableParaMostrarResultadosDeLaRutaSeleccionada = LaSegundaRuta.resultado;

    LaSegundaRuta.limpiarResultado();
}

else if (LaRutaSeleccionada == "3")
{
    LaTerceraRutaRuta.generarReporteDeRuta();

    ElNodoOrigenCiudad.siguienteSecundaria =
LaTerceraRutaRuta.devolverPrimerNodoDeLista();

    miVariableParaMostrarResultadosDeLaRutaSeleccionada = LaTerceraRutaRuta.resultado;

    LaTerceraRutaRuta.limpiarResultado();
}

else if (LaRutaSeleccionada == "4")
{
    LaCuartaRuta.generarReporteDeRuta();

    ElNodoOrigenCiudad.anteriorSecundaria = LaCuartaRuta.devolverPrimerNodoDeLista();

    miVariableParaMostrarResultadosDeLaRutaSeleccionada = LaCuartaRuta.resultado;

    LaCuartaRuta.limpiarResultado();
}
}

```



```

private void CalcularDistanciaEntreDepartamentoyCiudad()
{

    ObjetoInstanciaArbolNacional.reiniciarVariablesDeCalculo();

    ObjetoInstanciaArbolNacional.MostrarArbolNacional(LaRaizDelArbolNacional);

    ObjetoInstanciaArbolNacional.MostrarDistanciaEntreDepartamentosSinImportarRuta("Guatemala"
, DepartamentoDeInicioParaTransaccion, LaRaizDelArbolNacional);

    miVariableParaMostrarResultadosDistanciaEntreDepartamentoyCapital = "Resultado de
distancia: " + ObjetoInstanciaArbolNacional.resultado;

}

private void DistanciaEntreDepartamentos()
{

    if (DepartamentoDeInicioParaTransaccion != "" &&
DepartamentoDeFinalizacionParaTransaccion != "")

    {

        ObjetoInstanciaArbolNacional.reiniciarVariablesDeCalculo();

        ObjetoInstanciaArbolNacional.MostrarArbolNacional(LaRaizDelArbolNacional);

        ObjetoInstanciaArbolNacional.MostrarDistanciaEntreDepartamentosSinImportarRuta(Departamen
toDeInicioParaTransaccion, DepartamentoDeFinalizacionParaTransaccion, LaRaizDelArbolNacional);

        miVariableParaMostrarResultadosDeDistanciaEntreDepartamentos = "Resultado de
distancia: " + ObjetoInstanciaArbolNacional.resultado;

    }

}

private void LugaresVisitadosPorViaje()

```

```

{
    try
    {
        string nombreDepartamentoDelInicioParaUnViaje =
DepartamentoDeFinalizacionParaTransaccion;

        string nombreDepartamentoDelFinalParaUnViaje = DepartamentoDelInicioParaTransaccion;
        if (LaRutaSeleccionada == "")
        {
            VariableDeLogDeErrores = "No ha seleccionado una ruta.";
        }
        else if (LaRutaSeleccionada == "1")
        {

LaPrimeraRuta.departamentosRecorridosDeUnViaje(nombreDepartamentoDelInicioParaUnViaje,
nombreDepartamentoDelFinalParaUnViaje);

            miVariableParaMostrarResultadosDeLugaresVisitadosPorUnViaje =
LaPrimeraRuta.resultado;

            LaPrimeraRuta.limpiarResultado();
        }
        else if (LaRutaSeleccionada == "2")
        {

LaSegundaRuta.departamentosRecorridosDeUnViaje(nombreDepartamentoDelInicioParaUnViaje,
nombreDepartamentoDelFinalParaUnViaje);

            miVariableParaMostrarResultadosDeLugaresVisitadosPorUnViaje =
LaSegundaRuta.resultado;

            LaSegundaRuta.limpiarResultado();
        }
        else if (LaRutaSeleccionada == "3")
        {

```

```
LaTerceraRutaRuta.departamentosRecorridosDeUnViaje(nombreDepartamentoDelInicioParaUnViaje, nombreDepartamentoDelFinalParaUnViaje);
```

```
        miVariableParaMostrarResultadosDeLugaresVisitadosPorUnViaje =  
LaTerceraRutaRuta.resultado;
```

```
        LaTerceraRutaRuta.limpiarResultado();
```

```
    }
```

```
    else if (LaRutaSeleccionada == "4")
```

```
    {
```

```
LaCuartaRuta.departamentosRecorridosDeUnViaje(nombreDepartamentoDelInicioParaUnViaje,  
nombreDepartamentoDelFinalParaUnViaje);
```

```
        miVariableParaMostrarResultadosDeLugaresVisitadosPorUnViaje =  
LaCuartaRuta.resultado;
```

```
        LaCuartaRuta.limpiarResultado();
```

```
    }
```

```
}
```

```
catch
```

```
{
```

```
    VariableDeLogDeErrores = "Termine de configurar la transaccion.";
```

```
}
```

```
}
```

```
/*Fin funciones para generar reportes*/
```

```
private void TopLugaresMasCercanosALaCapital()
```

```
{
```

```
    ObjetoInstanciaArbolNacional.reiniciarVariablesDeCalculo();
```

```
    ObjetoInstanciaArbolNacional.LugaresMasCercanosALaCapital(LaRaizDelArbolNacional);
```

```
    MostrarResultados();
```

```
}
```

```
private void TopLugaresMasLejanosALaCapital()
```

```
{
```

```
ObjetoInstanciaArbolNacional.reiniciarVariablesDeCalculo();

ObjetoInstanciaArbolNacional.LugaresMasLejanosALaCapitalCorredor(LaRaizDelArbolNacional);

    ObjetoInstanciaArbolNacional.mostrarLosMasLejanos();

    MostrarResultados();
}

private void RecorrerElArbolNacional()
{
    ObjetoInstanciaArbolNacional.MostrarArbolNacional(LaRaizDelArbolNacional);

    MostrarResultados();
}

private void MostrarResultados()
{
    miVariableParaMostrarResultados = ObjetoInstanciaArbolNacional.resultado;

    ObjetoInstanciaArbolNacional.limpiarResultado();
}

private void LimpiarResultados()
{
    miVariableParaMostrarResultados = "";
}

}
```