

## MÉTODO DE LA INGENIERÍA

### FASE 1: IDENTIFICACIÓN DEL PROBLEMA.

La empresa Metro Cali S.A fue creada mediante el acuerdo N° 016 de 1998 del Concejo Municipal de Santiago de Cali, y se constituyó el 25 de febrero de 1999 mediante escritura pública No. 0580 como una sociedad por acciones en la que participen el Municipio de Santiago de Cali y otras entidades descentralizadas del orden Municipal.

Metro Cali es una entidad descentralizada, industrial y comercial del Estado del orden municipal, cuyo objeto es desarrollar el Sistema Integrado de Transporte Masivo SITM del Municipio Santiago de Cali. Nació para construir e implementar el sistema integrado de transporte de Santiago de Cali.

Metro Cali S.A. asumió un reto doble: la construcción y la operación del sistema MIO. Se encarga del diseño, construcción y puesta en marcha del Sistema Integrado de Transporte Masivo (SITM) de pasajeros para Santiago de Cali a partir de las definiciones técnicas, legales y financieras que imparta la banca de inversión.

Para Metro Cali es muy importante innovar y mantener una alta calidad en los servicios que presta. Por esa razón, la empresa desea que sus clientes (pasajeros en Santiago de Cali) tengan un fácil acceso a la información de los buses que están operando en ese instante, para que así puedan planear mejor los viajes que realizan.

#### Identificación de necesidades:

- Mostrar las posiciones de cada ruta de Metro Cali en tiempo real
- Mostrar las paradas de Metro Cali en toda la ciudad de Santiago de Cali
- Filtrar la información de las rutas de Metro Cali a deseo del usuario
- Mostrar las paradas de cada ruta en específico a través de todo el mapa de la ciudad de Santiago de Cali

### FASE 2: RECOPIACIÓN DE INFORMACIÓN NECESARIA.

Los datos fueron recopilados del sistema de información de la empresa Metro Cali S.A.

Al investigar en la web se obtuvo información de aplicaciones móviles que están actualmente en el mercado y que satisfacen las necesidades identificadas anteriormente. A continuación, se listan cada una junto con una descripción obtenida de Google Play, que es la plataforma en la que están publicadas estas aplicaciones.

Víctor Mora  
Juan Camilo Castillo  
Johan Camilo Cortés

### **Mio\_App:**

Es la aplicación oficial de Metro Cali S.A. Permite encontrar toda la información para planear el viaje, encontrar las rutas en tiempo real y visualizarlas en un mapa, consultar puntos de recarga cercanos, consultar el saldo de la tarjeta MIO, e incluso se puede interponer una pregunta, queja, reclamo, solicitud o algún otro comentario directamente a la empresa por medio de esta aplicación. También es posible trazar una ruta que va desde un origen hasta un destino por medio de búsqueda inteligente que se autocompleta con la tecnología de Moovit.

### **Rutas del Mio:**

Es una aplicación móvil desarrollada por la empresa de desarrollo móvil y web MoviliXa S.A.S, en la que se puede acceder a la información de cada una de las rutas del MIO, incluyendo estaciones, buses, horarios, puntos de recargar y mapas. Esta aplicación aprovecha el GPS del celular del usuario y la API de mapas de Google Maps.

### **Enrútate MIO:**

Es una aplicación móvil desarrollada por BSS Cali S.A.S. en la cual se puede conocer

- Las rutas de cada estación o parada.
- Recorrido de paradas que tiene de cada una de las rutas que se encuentren dentro del sistema.
- Podrán ver los sitios cercanos a un punto que se escoja en el mapa, gracias al api de Foursquare.
- Podrán ver noticias relacionados con el sistema MIO, Tweets Metrocali, Secretaría de Tránsito (@movilidadcali).
- Podrás conocer los puntos de recarga de tu tarjeta, (mapa o listado).
- Podrás conocer las formas de integración dentro del sistema.
- También tendrás la posibilidad de buscar estaciones o paradas, por el nombre o una dirección.

Las herramientas y tecnologías que se requieren para implementar una aplicación similar a las anteriores serían las siguientes:

### **Lenguaje de programación:**

En términos generales, un lenguaje de programación es una herramienta que permite desarrollar software o programas para computadora. Los lenguajes de programación son empleados para diseñar e implementar programas encargados de definir y administrar el comportamiento de los dispositivos físicos y lógicos de una computadora. Los anterior se logra mediante la creación e implementación de algoritmos de precisión que se utilizan como una forma de comunicación humana con la computadora.

A grandes rasgos, un lenguaje de programación se conforma de una serie de símbolos y reglas de sintaxis y semántica que definen la estructura principal del lenguaje y le dan un significado a sus elementos y expresiones (Universidad Nacional Autónoma de México, s.f.).

Víctor Mora  
Juan Camilo Castillo  
Johan Camilo Cortés

### **GMaps.NET:**

Es un control .NET de código abierto, gratis, cross-platform, que usa enrutamiento, geocodificación, direcciones y mapas de Google, Yahoo!, Bing, OpenStreetMap, ArcGIS, Pergo, SigPac, Yendux, Mapy.cz, Maps.lt, iKarte.lv, NearMap, HereMap, CloudMade, WikiMapia, MapQuest para visualizarlos en aplicaciones C# de tipo Windows Forms y Mobile.

### Referencias

BSS Cali S.A.S. (s.f.). *Enrútate MIO: BSS Cali S.A.S.* Obtenido de Google Play:

[https://play.google.com/store/apps/details?id=com.enrutatemio&hl=es\\_CO](https://play.google.com/store/apps/details?id=com.enrutatemio&hl=es_CO)

Metrocali S.A. (s.f.). *Mio\_App: App oficial de Metrocali S.A.* Obtenido de Google Play:

[https://play.google.com/store/apps/details?id=co.gov.metrocali.mioapp&hl=es\\_CO](https://play.google.com/store/apps/details?id=co.gov.metrocali.mioapp&hl=es_CO)

Microsoft. (2019). *GMap.Net.Window 1.9.3: NuGet*. Obtenido de NuGet:

<https://www.nuget.org/packages/GMap.NET.Windows/>

MoviliXa SAS. (s.f.). *Rutas del Mio: MoviliXa SAS*. Obtenido de Google Play:

[https://play.google.com/store/apps/details?id=com.movilixa.rutasmio&hl=es\\_CO](https://play.google.com/store/apps/details?id=com.movilixa.rutasmio&hl=es_CO)

Universidad Nacional Autónoma de México. (s.f.). *Lenguajes de Programación - Definición*. Obtenido de Lenguajes de Programación:

[https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/1023/mod\\_resource/content/1/contenido/index.html](https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/1023/mod_resource/content/1/contenido/index.html)

### **FASE 3: BÚSQUEDA DE SOLUCIONES CREATIVAS (Usando Lista de atributos).**

1. *Manipulación de datos por medio de arreglos de tamaño fijo.*
2. *Manipulación de datos por medio de arreglos de tamaño indefinido (ArrayList).*
3. *Manipulación de datos por medios de Grafos.*
4. *Manipulación de datos por medio de Hashtables.*
5. *Manipulación de datos por medio de Listas Enlazadas.*
6. *Manipulación de datos por medio de Arboles Binarios de Búsqueda (ABB)*

### **FASE 4: TRANSICIÓN DE LA FORMULACIÓN DE IDEAS A LOS DISEÑOS PRELIMINARES.**

**Alternativa 1:** Usar arreglos de tamaño fijo podría funcionar para almacenar toda la base de datos, pero se debe considerar que a la hora de hacer consultas sobre ese mismo arreglo sería demasiado costoso computacionalmente, así mismo, al mover o modelar algunas características de la base de datos sería poco útil usar este tipo de estructura de datos

**Alternativa 2:** El uso de un arreglo de tamaño variable conllevaría los mismos problemas que usar un arreglo de tamaño fijo sin contar que al momento de agregar valores de la base de datos al ArrayList se gastaría muchos recursos computacionales para poder cumplir con dicha necesidad

Víctor Mora  
Juan Camilo Castillo  
Johan Camilo Cortés

**Alternativa 3:** El uso de Grafos podría facilitar la manera en que se llega a cada uno valores de la base de datos. De igual manera, usar Grafos permite modelar la conexión entre valores de la base de datos, ya que entre cada uno de ellos hay caminos que los conecta.

**Alternativa 4:** Usar una Tabla Hash y una función asociada a la base de datos permitiría que cada valor se ubicara en un espacio específico lo que posibilita su consulta y acceso a los valores que dicho objeto ahí alojado tenga.

**Alternativa 5:** El uso de las listas enlazadas podría reducir la manera en la que se guardan y modifican las posiciones de los valores de la base de datos, sumando esto la consulta de esta lista sería diferente, pero al momento de agregar o eliminar un valor conllevaría a mucho gasto computacional.

**Alternativa 6:** Con el ABB pasa algo muy parecido que con la lista enlazada ya que, al momento de agregar, eliminar se necesita muchos recursos computacionales para poder hacerlos en una Base de Datos tan grande.

#### FASE 5: EVALUACIÓN Y SELECCIÓN DE LA MEJOR SOLUCIÓN.

**FACILIDAD DE IMPLEMENTACIÓN:** evalúa de 0 a 5, siendo 0 una calificación dada a aquella solución que sea bastante difícil de implementar con las herramientas y conocimientos que tiene el grupo actualmente, y siendo 5 una calificación a una implementación bastante sencilla y alcanzable con las herramientas y conocimientos del grupo

**EFICIENCIA EN LA SOLUCIÓN:** evalúa de 0 a 5, siendo 0 una calificación dada a aquella solución cuya capacidad de solucionar el problema sea casi nulo o muy poco preciso en comparación a lo esperado, y siendo 5 la calificación dada a aquella solución cuyo resultado sea capaz de satisfacer los requerimientos establecidos

#### Alternativa 3

Criterio	Facilidad de implementación	Eficiencia en la solución	total
Visualización	2	3	5
Algoritmo de búsqueda	3	4	7
Agregar Dato	4	5	9
TOTAL			21

#### Alternativa 4

Criterio	Facilidad de implementación	Eficiencia en la solución	total
----------	-----------------------------	---------------------------	-------

Víctor Mora  
 Juan Camilo Castillo  
 Johan Camilo Cortés

Visualización	3	3	6
Algoritmo de búsqueda	4	5	9
Agregar Dato	3	5	8
TOTAL			23

### Complejidad temporal de los algoritmos

Nombre: saveCoordinates			
Línea	Instrucción	Costo	Ejecuciones
1	<b>try</b> {	C1	1
2	string line = "";	C2	1
3	StreamReader sr = new StreamReader("data/BDTEXTO.txt");	C3	1
4	<b>while</b> ((line = sr.ReadLine()) != null) {	C4	n + 1
5	if (line.Split(',')[3].Equals(app.getTestBus().getBusId()))	C5	n
6	double newLat = app.adjustCoordinates(line.Split(',')[0]);	C6	n
7	double newLon = app.adjustCoordinates(line.Split(',')[1]);	C7	n
8	app.getTestBus().getCoordinates().Add (newKeyValuePair<double, double>(newLat, newLon));	C8	n
9	<b>catch</b> (Exception e) {	C9	0
10	MessageBox.Show(e.Message);	C10	0

Siendo n el número de líneas no vacías del documento de texto plano.

$$T(n) = 1 + 1 + 1 + n + 1 + n + n + n + n + 0 + 0 = 5n + 4$$

$$O(T(n)) = O(5n + 4) = O(n)$$

Víctor Mora  
 Juan Camilo Castillo  
 Johan Camilo Cortés

Nombre: ReadFile			
Línea	Instrucción	Costo	Ejecuciones
1	<b>try</b> {	C1	1
2	int i = 0;	C2	1
3	string line = "";	C3	1
4	StreamReader sr = new StreamReader("data/stops.csv");	C4	1
5	<b>while</b> ((line = sr.ReadLine()) != null) {	C5	n + 1
6	int type = 0;	C6	n
7	String[] individual = line.Split(',');	C7	n
8	double lat = double.Parse(individual[7], CultureInfo.InvariantCulture);	C8	n
9	double lon = double.Parse(individual[6], CultureInfo.InvariantCulture);	C9	n
10	<b>if</b> (individual[3].Contains("con")    individual[3].Contains("entre")) {	C10	n
11	type = 1;	C11	n
12	<b>else</b> { type = 2; }	C12	0
13	Stop st = new Stop(individual[0], type, individual[2], individual[3], lon, lat);	C13	n
14	<b>if</b> (type==1) {	C14	n
15	app.getStreetStop().Add(st);	C15	n
16	<b>else</b> { app.getStationStop().Add(st); }	C16	0
17	i++; }	C17	n
18	app.saveElements();	C18	1
19	<b>catch</b> (Exception e) {	C19	0
20	MessageBox.Show(e.Message);	C20	0

Siendo n el número de líneas no vacías del documento de texto plano.

$$T(n) = 1 + 1 + 1 + 1 + n + 1 + n + n + n + n + n + n + 0 + n + n + n + 0 + n + 1 + 0 + 0 = 11n + 6$$

$$O(T(n)) = O(11n + 6) = O(n)$$

Nombre: StopMap_Load			
Línea	Instrucción	Costo	Ejecuciones
1	stopMap.DragButton = MouseButton.Left;	C1	1
2	stopMap.MapProvider = GMapProviders.GoogleMap;	C2	1
3	double latitud = 3.4372201;	C3	1
4	double longitud = -76.5224991;	C4	1
5	stopMap.Position = <b>new</b> PointLatLng(latitud, longitud);	C5	1
6	stopMap.MinZoom = 1;	C6	1
7	stopMap.MaxZoom = 100;	C7	1
8	stopMap.Zoom = 13;	C8	1

$$T(n) = 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 = 8$$

$$O(T(n)) = O(8) = O(1)$$

Nombre: drawStops			
Línea	Instrucción	Costo	Ejecuciones
1	stopMap.Overlays.Clear();	C1	1
2	GMapOverlay markers = new GMapOverlay("markers");	C2	1
3	foreach (var aux in GetApp().getStreetStop()) {	C3	n + 1
4	streetsMarker(aux.Declati, aux.Declong); }	C4	n
5	stopMap.Overlays.Add(markers);	C5	1
6	stopMap.Zoom = stopMap.Zoom + 1;	C6	1
7	stopMap.Zoom = stopMap.Zoom - 1;	C7	1

Siendo n el número de paradas en el mapa.

$$T(n) = 1 + 1 + n + 1 + n + 1 + 1 + 1 = 2n + 6$$

$$O(T(n)) = O(2n + 6) = O(n)$$

Nombre: streetsMarker			
Línea	Instrucción	Costo	Ejecuciones
1	PointLatLng point = new PointLatLng(lat, lng);	C1	1
2	GMapMarker theMarker = new GMarkerGoogle(point, GMarkerGoogleType.red_dot);	C2	1
3	GMapOverlay markers = new GMapOverlay("markers");	C3	1
4	markers.Markers.Add(theMarker);	C4	1
5	stopMap.Overlays.Add(markers);	C5	1

$$T(n) = 1 + 1 + 1 + 1 + 1 = 5$$

$$O(T(n)) = O(5) = O(1)$$

Nombre: drawStations			
Línea	Instrucción	Costo	Ejecuciones
1	stopMap.Overlays.Clear();	C1	1
2	foreach (var aux in GetApp().getStationStop())	C2	n + 1
3	stationsMarker(aux.Declati, aux.Declong);	C3	n
4	stopMap.Zoom = stopMap.Zoom + 1;	C4	1
5	stopMap.Zoom = stopMap.Zoom - 1;	C5	1

Siendo n el número de estaciones en el mapa.

$$T(n) = 1 + n + 1 + n + 1 + 1 = 2n + 4$$

$$O(T(n)) = O(2n + 4) = O(n)$$

Nombre: StationsMarkers			
Línea	Instrucción	Costo	Ejecuciones
1	PointLatLng point = new PointLatLng(lat, lng);	C1	1
2	GMapMarker theMarker = new GMarkerGoogle(point, GMarkerGoogleType.blue_dot);	C2	1
3	GMapOverlay markers = new GMapOverlay("markers");	C3	1
4	markers.Markers.Add(theMarker);	C4	1
5	stopMap.Overlays.Add(markers);	C5	1

$$T(n) = 1 + 1 + 1 + 1 + 1 = 5$$

$$O(T(n)) = O(5) = O(1)$$

Nombre: drawZones			
Línea	Instrucción	Costo	Ejecuciones
1	cbCentro.Visible = true;	C1	1
2	cbMenga.Visible = true;	C2	1
3	cbCalima.Visible = true;	C3	1
4	cbAguablanca.Visible = true;	C4	1
5	cbCiudadCordoba.Visible = true;	C5	1
6	cbCañaveralejo.Visible = true;	C6	1
7	cbValleDeLili.Visible = true;	C7	1
8	cbPrado.Visible = true;	C8	1
9	cbGuadalupe.Visible = true;	C9	1
10	lbOpt.Location = new Point(12,271);	C10	1
11	optionComBox.Location = new Point(40,300);	C11	1
12	btnSatelite.Location = new Point(25, 501);	C12	1
13	btnRelieve.Location = new Point(106, 501);	C13	1
14	btnNormal.Location = new Point(64, 530);	C14	1
15	lbZoom.Location = new Point(37, 556);	C15	1
16	tbZoom.Location = new Point(40, 572);	C16	1

$$T(n) = 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 = 16$$

$$O(T(n)) = O(16) = O(1)$$



Víctor Mora  
 Juan Camilo Castillo  
 Johan Camilo Cortés

Nombre : drawZone1			
Línea	Instrucción	Costo	Ejecuciones
1	zona0 = new GMapOverlay("Zona 0");	C1	1
2	List<PointLatLng> puntosZ0 = new List<PointLatLng>();	C2	1
3	puntosZ0.Add(new PointLatLng(3.458952, -76.513447));	C3	1
4	puntosZ0.Add(new PointLatLng(3.463029, -76.521014));	C4	1
5	puntosZ0.Add(new PointLatLng(3.460478, -76.527052));	C5	1
6	puntosZ0.Add(new PointLatLng(3.450851, -76.537823));	C6	1
7	puntosZ0.Add(new PointLatLng(3.451531, -76.541646));	C7	1
8	puntosZ0.Add(new PointLatLng(3.455851, -76.542208));	C8	1
9	puntosZ0.Add(new PointLatLng(3.458972, -76.553556));	C9	1
10	puntosZ0.Add(new PointLatLng(3.458552, -76.582726));	C10	1
11	puntosZ0.Add(new PointLatLng(3.448433, -76.562660));	C11	1
12	puntosZ0.Add(new PointLatLng(3.452630, -76.549432));	C12	1
13	puntosZ0.Add(new PointLatLng(3.441510, -76.536748));	C13	1
14	puntosZ0.Add(new PointLatLng(3.428423, -76.533186));	C14	1
15	puntosZ0.Add(new PointLatLng(3.434846, -76.523287));	C15	1
16	puntosZ0.Add(new PointLatLng(3.443270, -76.518067));	C16	1
17	GMapPolygon poligonoZ0 = new GMapPolygon(puntosZ0, "CENTRO");	C17	1
18	zona0.Polygons.Add(poligonoZ0);	C18	1
19	stopMap.Overlays.Add(zona0);	C19	1
20	stopMap.Zoom = stopMap.Zoom + 1;	C20	1
21	stopMap.Zoom = stopMap.Zoom - 1;	C21	1

$$T(n) = 1 + 1 = 21$$

$$O(T(n)) = O(21) = O(1)$$

Víctor Mora  
Juan Camilo Castillo  
Johan Camilo Cortés

Nombre: drawZone2			
Línea	Instrucción	Costo	Ejecuciones
1	zona2 = new GMapOverlay("Zona 2");	C1	1
2	List<PointLatLng> puntosZ2 = new List<PointLatLng>();	C2	1
3	puntosZ2.Add(new PointLatLng(3.463029, -76.521014));	C3	1
4	puntosZ2.Add(new PointLatLng(3.460478, -76.527052));	C4	1
5	puntosZ2.Add(new PointLatLng(3.450851, -76.537823));	C5	1
6	puntosZ2.Add(new PointLatLng(3.451531, -76.541646));	C6	1
7	puntosZ2.Add(new PointLatLng(3.455851, -76.542208));	C7	1
8	puntosZ2.Add(new PointLatLng(3.454776, -76.540491));	C8	1
9	puntosZ2.Add(new PointLatLng(3.472167, -76.531307));	C9	1
10	puntosZ2.Add(new PointLatLng(3.475791, -76.532884));	C10	1
11	puntosZ2.Add(new PointLatLng(3.478440, -76.529107));	C11	1
12	puntosZ2.Add(new PointLatLng(3.481190, -76.529311));	C12	1
13	puntosZ2.Add(new PointLatLng(3.488169, -76.537783));	C13	1
14	puntosZ2.Add(new PointLatLng(3.489748, -76.538549));	C14	1
15	puntosZ2.Add(new PointLatLng(3.494435, -76.535436));	C15	1
16	puntosZ2.Add(new PointLatLng(3.489850, -76.528852));	C16	1
17	puntosZ2.Add(new PointLatLng(3.496918, -76.526449));	C17	1
18	puntosZ2.Add(new PointLatLng(3.498546, -76.522458));	C18	1
19	puntosZ2.Add(new PointLatLng(3.506085, -76.521685));	C19	1
20	puntosZ2.Add(new PointLatLng(3.516365, -76.519454));	C20	1
21	puntosZ2.Add(new PointLatLng(3.519449, -76.533659));	C21	1
22	puntosZ2.Add(new PointLatLng(3.527416, -76.526020));	C22	1
23	puntosZ2.Add(new PointLatLng(3.520434, -76.519239));	C23	1
24	puntosZ2.Add(new PointLatLng(3.521805, -76.517651));	C24	1
25	puntosZ2.Add(new PointLatLng(3.525615, -76.506765));	C25	1
26	puntosZ2.Add(new PointLatLng(3.523958, -76.495361));	C26	1
27	puntosZ2.Add(new PointLatLng(3.505948, -76.500053));	C27	1
28	puntosZ2.Add(new PointLatLng(3.505516, -76.492691));	C28	1
29	puntosZ2.Add(new PointLatLng(3.499343, -76.492160));	C29	1
30	GMapPolygon poligonoZ2 = new GMapPolygon(puntosZ2, "MENA");	C30	1
31	zona2.Polygons.Add(poligonoZ2);	C31	1
32	stopMap.Overlays.Add(zona2);	C32	1
33	stopMap.Zoom = stopMap.Zoom + 1;	C33	1
34	stopMap.Zoom = stopMap.Zoom - 1;	C34	1

[illegible]

$$O(T(n)) = O(34) = O(1)$$

Víctor Mora  
 Juan Camilo Castillo  
 Johan Camilo Cortés

Nombre: drawZone3			
Línea	Instrucción	Costo	Ejecuciones
1	zona3 = new GMapOverlay("Zona 3");	C1	1
2	List<PointLatLng> puntosZ3 = new List<PointLatLng>();	C2	1
3	puntosZ3.Add(new PointLatLng(3.499343, -76.492160));	C3	1
4	puntosZ3.Add(new PointLatLng(3.497010, -76.485022));	C4	1
5	puntosZ3.Add(new PointLatLng(3.482987, -76.482092));	C5	1
6	puntosZ3.Add(new PointLatLng(3.480321, -76.479688));	C6	1
7	puntosZ3.Add(new PointLatLng(3.477062, -76.479344));	C7	1
8	puntosZ3.Add(new PointLatLng(3.474059, -76.476589));	C8	1
9	puntosZ3.Add(new PointLatLng(3.468334, -76.475987));	C9	1
10	puntosZ3.Add(new PointLatLng(3.458952, -76.513447));	C10	1
11	puntosZ3.Add(new PointLatLng(3.463029, -76.521014));	C11	1
12	GMapPolygon poligonoZ3 = new GMapPolygon(puntosZ3, "CAUMA");	C12	1
13	zona3.Polygons.Add(poligonoZ3);	C13	1
14	stopMap.Overlays.Add(zona3);	C14	1
15	stopMap.Zoom = stopMap.Zoom + 1;	C15	1
16	stopMap.Zoom = stopMap.Zoom - 1;	C16	1

$$T(n) = 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 = 16$$

$$O(T(n)) = O(16) = O(1)$$

Nombre: drawZone4			
Línea	Instrucción	Costo	Ejecuciones
1	zona4 = new GMapOverlay("Zona4");	C1	1
2	List<PointLatLng> puntosZ4 = new List<PointLatLng>();	C2	1
3	puntosZ4.Add(new PointLatLng(3.443270, -76.518067));	C3	1
4	puntosZ4.Add(new PointLatLng(3.458952, -76.513447));	C4	1
5	puntosZ4.Add(new PointLatLng(3.468334, -76.475987));	C5	1
6	puntosZ4.Add(new PointLatLng(3.465410, -76.474958));	C6	1
7	puntosZ4.Add(new PointLatLng(3.460690, -76.478052));	C7	1
8	puntosZ4.Add(new PointLatLng(3.440685, -76.472893));	C8	1
9	puntosZ4.Add(new PointLatLng(3.441297, -76.466824));	C9	1
10	puntosZ4.Add(new PointLatLng(3.436129, -76.463077));	C10	1
11	puntosZ4.Add(new PointLatLng(3.399183, -76.466336));	C11	1
12	puntosZ4.Add(new PointLatLng(3.400404, -76.476322));	C12	1
13	puntosZ4.Add(new PointLatLng(3.413793, -76.482086));	C13	1
14	puntosZ4.Add(new PointLatLng(3.418084, -76.486381));	C14	1
15	puntosZ4.Add(new PointLatLng(3.428568, -76.483722));	C15	1
16	puntosZ4.Add(new PointLatLng(3.446679, -76.484064));	C16	1
17	puntosZ4.Add(new PointLatLng(3.443159, -76.493519));	C17	1
18	GMapPolygon poligonoZ4 = new GMapPolygon(puntosZ4, "AGUABLANCA");	C18	1
19	zona4.Polygons.Add(poligonoZ4);	C19	1
20	stopMap.Overlays.Add(zona4);	C20	1
21	stopMap.Zoom = stopMap.Zoom + 1;	C21	1
22	stopMap.Zoom = stopMap.Zoom - 1;	C22	1

$$T(n) = 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 = 22$$

$$O(T(n)) = O(22) = O(1)$$

Víctor Mora  
Juan Camilo Castillo  
Johan Camilo Cortés

Nombre: drawZone5			
Línea	Instrucción	Costo	Ejecuciones
1	zona5 = new GMapOverlay("Zonas 5");	C1	1
2	List<PointLatLng> puntosZ5 = new List<PointLatLng>();	C2	1
3	puntosZ5.Add(new PointLatLng(3.443827, -76.491165));	C3	1
4	puntosZ5.Add(new PointLatLng(3.428302, -76.498642));	C4	1
5	puntosZ5.Add(new PointLatLng(3.407200, -76.521070));	C5	1
6	puntosZ5.Add(new PointLatLng(3.404841, -76.518878));	C6	1
7	puntosZ5.Add(new PointLatLng(3.402954, -76.520984));	C7	1
8	puntosZ5.Add(new PointLatLng(3.403125, -76.522445));	C8	1
9	puntosZ5.Add(new PointLatLng(3.400466, -76.523132));	C9	1
10	puntosZ5.Add(new PointLatLng(3.394890, -76.523605));	C10	1
11	puntosZ5.Add(new PointLatLng(3.394333, -76.519308));	C11	1
12	puntosZ5.Add(new PointLatLng(3.390859, -76.519738));	C12	1
13	puntosZ5.Add(new PointLatLng(3.390215, -76.516300));	C13	1
14	puntosZ5.Add(new PointLatLng(3.391288, -76.516000));	C14	1
15	puntosZ5.Add(new PointLatLng(3.393732, -76.513164));	C15	1
16	puntosZ5.Add(new PointLatLng(3.388929, -76.509598));	C16	1
17	puntosZ5.Add(new PointLatLng(3.392801, -76.504657));	C17	1
18	puntosZ5.Add(new PointLatLng(3.402838, -76.499415));	C18	1
19	puntosZ5.Add(new PointLatLng(3.414504, -76.483088));	C19	1
20	puntosZ5.Add(new PointLatLng(3.418084, -76.486381));	C20	1
21	puntosZ5.Add(new PointLatLng(3.428568, -76.483722));	C21	1
22	puntosZ5.Add(new PointLatLng(3.446679, -76.484064));	C22	1
23	GMapPolygon poligonoZ5 = new GMapPolygon(puntosZ5, "CIUDAD CORDOBA");	C23	1
24	zona5.Polygons.Add(poligonoZ5);	C24	1
25	stopMap.Overlays.Add(zona5);	C25	1
26	stopMap.Zoom = stopMap.Zoom + 1;	C26	1
27	stopMap.Zoom = stopMap.Zoom - 1;	C27	1

[illegible]

$$O(T(n)) = O(27) = O(1)$$

Víctor Mora  
 Juan Camilo Castillo  
 Johan Camilo Cortés

Nombre: drawZone6			
Línea	Instrucción	Costo	Ejecuciones
1	zona6 = new GMapOverlay("Zona 6");	C1	1
2	List<PointLatLng> puntosZ6 = new List<PointLatLng>();	C2	1
3	puntosZ6.Add(new PointLatLng(3.419485, -76.508024));	C3	1
4	puntosZ6.Add(new PointLatLng(3.419999, -76.508604));	C4	1
5	puntosZ6.Add(new PointLatLng(3.419432, -76.521462));	C5	1
6	puntosZ6.Add(new PointLatLng(3.434846, -76.523287));	C6	1
7	puntosZ6.Add(new PointLatLng(3.428423, -76.533186));	C7	1
8	puntosZ6.Add(new PointLatLng(3.398191, -76.538025));	C8	1
9	puntosZ6.Add(new PointLatLng(3.398277, -76.542537));	C9	1
10	puntosZ6.Add(new PointLatLng(3.386310, -76.544481));	C10	1
11	puntosZ6.Add(new PointLatLng(3.387827, -76.532647));	C11	1
12	puntosZ6.Add(new PointLatLng(3.392956, -76.532493));	C12	1
13	puntosZ6.Add(new PointLatLng(3.392462, -76.524631));	C13	1
14	puntosZ6.Add(new PointLatLng(3.387518, -76.524073));	C14	1
15	puntosZ6.Add(new PointLatLng(3.389252, -76.519946));	C15	1
16	puntosZ6.Add(new PointLatLng(3.389971, -76.520956));	C16	1
17	puntosZ6.Add(new PointLatLng(3.389209, -76.522588));	C17	1
18	puntosZ6.Add(new PointLatLng(3.392104, -76.523738));	C18	1
19	puntosZ6.Add(new PointLatLng(3.394890, -76.523605));	C19	1
20	puntosZ6.Add(new PointLatLng(3.400466, -76.523132));	C20	1
21	puntosZ6.Add(new PointLatLng(3.403125, -76.522445));	C21	1
22	puntosZ6.Add(new PointLatLng(3.402954, -76.520984));	C22	1
23	puntosZ6.Add(new PointLatLng(3.404841, -76.518878));	C23	1
24	puntosZ6.Add(new PointLatLng(3.407200, -76.521070));	C24	1
25	GMapPolygon poligonoZ6 = new GMapPolygon(puntosZ6, "GUADALUPE");	C25	1
26	zona6.Polygons.Add(poligonoZ6);	C26	1
27	stopMap.Overlays.Add(zona6);	C27	1
28	stopMap.Zoom = stopMap.Zoom + 1;	C28	1
29	stopMap.Zoom = stopMap.Zoom - 1;	C29	1

$$T(n) = 1 + 1 = 29$$

$$O(T(n)) = O(29) = O(1)$$



Nombre: drawZone7			
Línea	Instrucción	Costo	Ejecuciones
1	zona7 = new GMapOverlay("Zona 7");	C1	1
2	List<PointLatLng> puntosZ7 = new List<PointLatLng>();	C2	1
3	puntosZ7.Add(new PointLatLng(3.428423, -76.533186));	C3	1
4	puntosZ7.Add(new PointLatLng(3.441510, -76.536748));	C4	1
5	puntosZ7.Add(new PointLatLng(3.452630, -76.549432));	C5	1
6	puntosZ7.Add(new PointLatLng(3.448433, -76.562660));	C6	1
7	puntosZ7.Add(new PointLatLng(3.435942, -76.557605));	C7	1
8	puntosZ7.Add(new PointLatLng(3.427738, -76.571950));	C8	1
9	puntosZ7.Add(new PointLatLng(3.417393, -76.567787));	C9	1
10	puntosZ7.Add(new PointLatLng(3.411738, -76.574267));	C10	1
11	puntosZ7.Add(new PointLatLng(3.407067, -76.578130));	C11	1
12	puntosZ7.Add(new PointLatLng(3.402269, -76.571864));	C12	1
13	puntosZ7.Add(new PointLatLng(3.391773, -76.572207));	C13	1
14	puntosZ7.Add(new PointLatLng(3.384706, -76.560319));	C14	1
15	puntosZ7.Add(new PointLatLng(3.386310, -76.544481));	C15	1
16	puntosZ7.Add(new PointLatLng(3.398277, -76.542537));	C16	1
17	puntosZ7.Add(new PointLatLng(3.398191, -76.538025));	C17	1
18	GMapPolygon poligonoZ7 = new GMapPolygon(puntosZ7, "CAÑAVERALEJO");	C18	1
19	zona7.Polygons.Add(poligonoZ7);	C19	1
20	stopMap.Overlays.Add(zona7);	C20	1
21	stopMap.Zoom = stopMap.Zoom + 1;	C21	1
22	stopMap.Zoom = stopMap.Zoom - 1;	C22	1

$$T(n) = 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 = 22$$

$$O(T(n)) = O(22) = O(1)$$

Nombre: drawZone8			
Línea	Instrucción	Costo	Ejecuciones
1	zona8 = new GMapOverlay("Zona 8");	C1	1
2	List<PointLatLng> puntosZ8 = new List<PointLatLng>();	C2	1
3	puntosZ8.Add(new PointLatLng(3.443827, -76.491165));	C3	1
4	puntosZ8.Add(new PointLatLng(3.428302, -76.498642));	C4	1
5	puntosZ8.Add(new PointLatLng(3.419485, -76.508024));	C5	1
6	puntosZ8.Add(new PointLatLng(3.419999, -76.508604));	C6	1
7	puntosZ8.Add(new PointLatLng(3.419432, -76.521462));	C7	1
8	puntosZ8.Add(new PointLatLng(3.434846, -76.523287));	C8	1
9	puntosZ8.Add(new PointLatLng(3.443270, -76.518067));	C9	1
10	puntosZ8.Add(new PointLatLng(3.443159, -76.493519));	C10	1
11	GMapPolygon poligonoZ8 = new GMapPolygon(puntosZ8, "PRADO");	C11	1
12	zona8.Polygons.Add(poligonoZ8);	C12	1
13	stopMap.Overlays.Add(zona8);	C13	1
14	stopMap.Zoom = stopMap.Zoom + 1;	C14	1
15	stopMap.Zoom = stopMap.Zoom - 1;	C15	1

$$T(n) = 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 = 15$$

$$O(T(n)) = O(15) = O(1)$$

Nombre: saveElements			
Línea	Instrucción	Costo	Ejecuciones
1	try {	C1	1
2	BinaryFormatter format = new BinaryFormatter();	C2	1
3	FileStream stream = new FileStream("data/StreetStopsList.txt", FileMode.Create);	C3	1
4	format.Serialize(stream, streetStopList);	C4	1
5	FileStream stream2 = new FileStream("data/StationStopsList.txt", FileMode.Create);	C5	1
6	format.Serialize(stream2, stationStopList);	C6	1
7	stream.Close();	C7	1
8	stream2.Close();	C8	1
9	catch(Exception e)	C9	1
10	Console.WriteLine(e.Message);	C10	1

$$T(n) = 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 = 10$$

$$O(T(n)) = O(10) = O(1)$$

Nombre: loadElements			
Línea	Instrucción	Costo	Ejecuciones
1	BinaryFormatter format = new BinaryFormatter();	C1	1
2	FileStream stream = new FileStream("data/StreetStopsList.txt", FileMode.Open);	C2	1
3	FileStream stream2 = new FileStream("data/StationStopsList.txt", FileMode.Open);	C3	1
4	streetStopList = (List<Stop>) format.Deserialize(stream);	C4	1
5	stationStopList = (List<Stop>) format.Deserialize(stream2);	C5	1
6	stream.Close();	C6	1
7	stream2.Close();	C7	1

$$T(n) = 1 + 1 + 1 + 1 + 1 + 1 + 1 = 7$$

$$O(T(n)) = O(7) = O(1)$$