**Descripción del problema:**

Con el propósito de fortalecer el modelo de gobierno digital en las entidades del orden nacional y prestar un servicio a los usuarios para lograr una Colombia más conectada en el 2015 el ministerio de tecnologías de la información y las comunicaciones (MinTic) han implementado a lo largo de todo el país unas zonas de wifi gratis, así pudiendo abastecer a más de cuatro mil usuarios. Esto ha traído nuevos problemas para el ministerio de tecnologías, pues necesitan saber cuántas zonas wifi hay en cada municipio, además necesita que los usuarios puedan acceder a la aplicación para buscar una zona en específico mostrando de forma automática todas las opciones a las que pueden acceder, así pudiendo ayudar tanto a turistas como a ciudadanos a encontrar dichas zonas y aprovecharlas.

Para solucionar dicho problema también le han pedido utilizar el lenguaje c# con Windows Forms, además se le proporcionara una base de datos en la cual están todas las zonas disponibles. Para que sea más fácil hacerle mantenimiento al código por otras personas se le pide un diagrama de clases y hacer un informe aplicando el método de la ingeniería. Por otra parte, se necesita que se pueda mostrar el grafico con las zonas por municipio y poder, tanto ver como eliminar, los puntos del mapa.

Identificación de necesidades:

* Se necesita poder leer un archivo y convertirlo en datos útiles para el programa.
* Se necesita mostrar un grafico con las zonas por municipio.
* Se requiere poder mostrar las opciones que pueden tomar los turistas para encontrar una zona en específico.
* Se debe poder mostrar en un mapa la ubicación exacta de la zona.
* Se debe poder eliminar todos los puntos del mapa.

Definición del problema:

Se requiere realizar un programa en el cual se pueda leer una base de datos para poder tratarla y así permitir a los usuarios saber cuantas zonas wifi hay en cada municipio, buscar una zona en especifico y añadir o eliminar puntos al mapa.

**Recopilación de información:**

Gmaps: es una librería de C# que se basa en el lenguaje Google maps, y nos permite crear un mapa, utilizando Gmap.Net, una biblioteca multiplataforma, la cual reduce el código haciendo que la aplicación del mapa sea mucho más sencilla, además de esto gmap tiene otras funciones como, por ejemplo: crear mapas con marcadores, rutas, geolocalización, perfiles longitudinales, etc.

Latitud: es la distancia entre un punto de la superficie de la tierra hasta el paralelo del ecuador, se mide en grados, puede variar entre -90 a 90, sirve para localizar un lugar en dirección Norte o Sur

Longitud es la distancia entre el meridiano de Greenwich y el meridiano de referencia que se toma como 0, se mide de -180 hasta 180, sirve para localizar un lugar en dirección Este u Oeste.

Ideas para la creación de la aplicación:

1. Crear una aplicación que todo el tratamiento lo haga en el from1, sin utilizar clases, pero utilizando los métodos auxiliares necesarios para cumplir con todas las necesidades requeridas. Además, se puede dejar que el programa trabaje sin acceso a internet y haciendo que los usuarios descarguen una versión del gmaps.
2. Crear una aplicación que guarde todos los datos leídos en una clases, permitiendo facilitar su acceso, su manipulación y reduciendo la carga que puede tener from1.

Ideas para acceder a la ubicación:

1. Crear una clase en la que las zonas wifi tengan sus propias coordenadas ordenadas para poder acceder a ellas de forma rápida.
2. Tener todas las coordenadas ya agregadas y simplemente mostrar la que pide el usuario.

Alternativa 1 Creación de la aplicación:

* Puede ser una opción válida, pero va a ser muy difícil de hacerle mantenimiento pues va a estar saturada de métodos para que la aplicación funcione. Por otra parte seria una forma “rápida” y “sencilla” de poder solucionar el problema, aunque no estaría al nivel de un programa hecho por ingenieros.

Alternativa 2 acceder a la ubicación:

* Puede ser muy buena idea, pero para tratar una base de datos pequeña. En este caso no nos va a ser útil, pues al tener tan alto volumen de datos usar este método significaría sacrificar mucho espacio de almacenamiento y, por ende, optimización de la aplicación.

Evaluación y selección de la solución:

Criterio 1: Se le darán 3 puntos a la solución que permita un mejor tratamiento de los datos.

Criterio 2: Se le darán 2 puntos a la solución que permita tener un mejor mantenimiento.

Tabla 1.

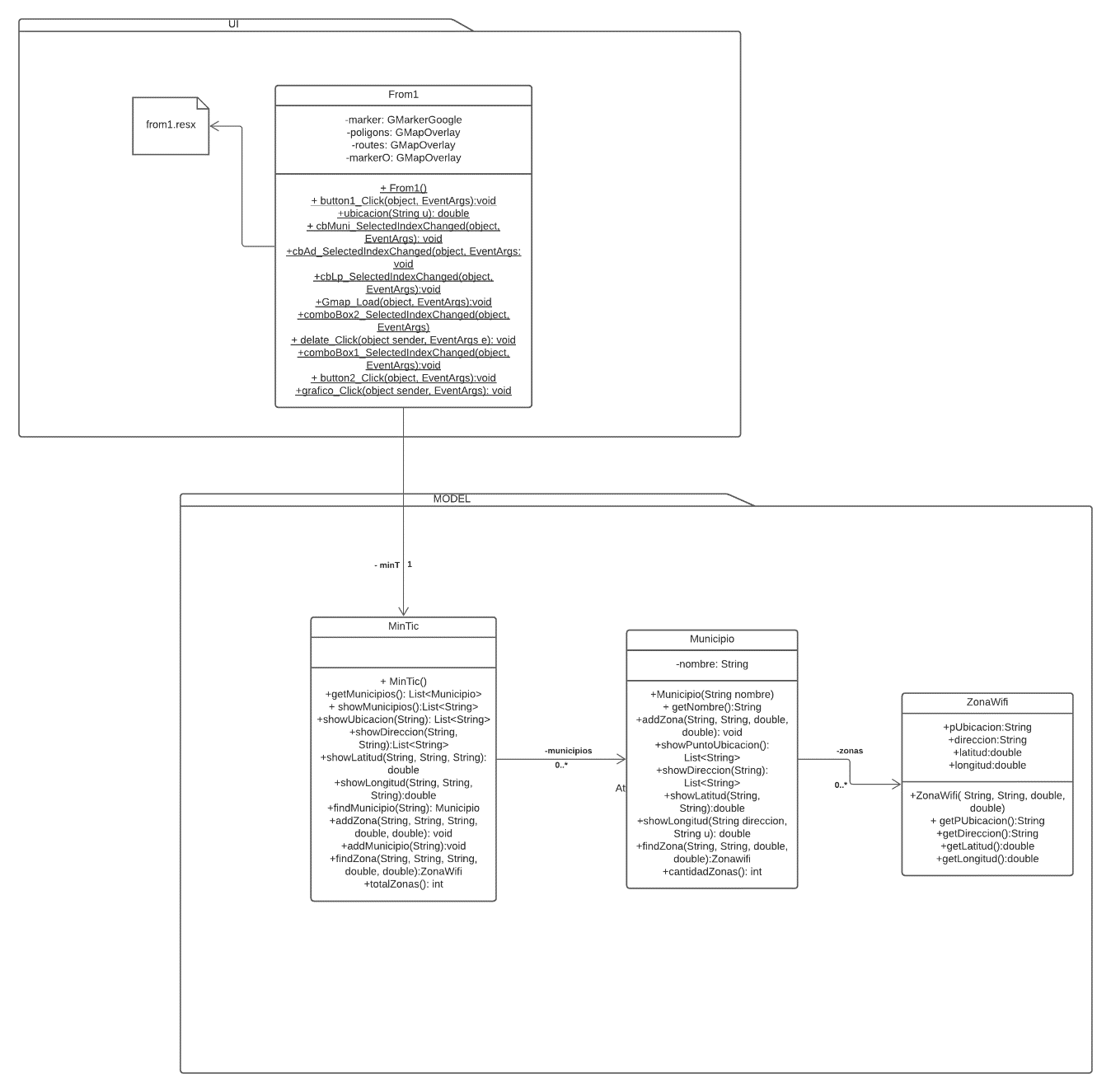
|  |  |  |
| --- | --- | --- |
|  | Diseño de la aplicación |  |
|  | Criterio 1 | Criterio 2 |
| Alternativa 1 | 0 | 0 |
| Alternativa 2 | 3 | 2 |

Tabla 2.

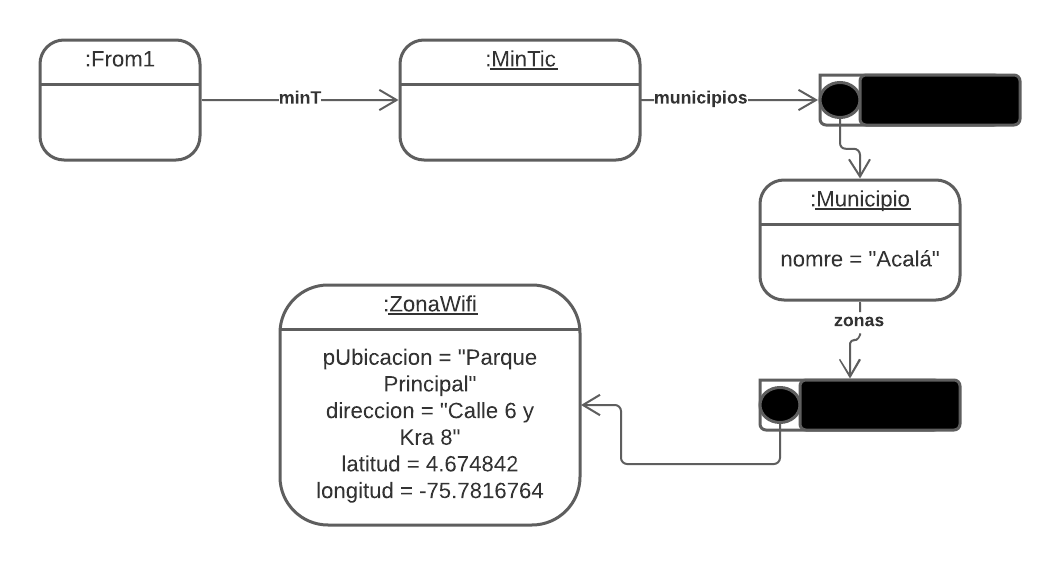
|  |  |  |
| --- | --- | --- |
|  | Acceder a la ubicación |  |
|  | Criterio 1 | Criterio 2 |
| Alternativa 1 | 3 | 2 |
| Alternativa 2 | 0 | 2 |

Como podemos ver en la Tabla 1 y en la Tabla 2, lo mejor que podemos hacer es escoger la segunda alternativa del diseño de la aplicación y la primer de acceder a la ubicación.

**Diagrama de clase:**

Gráfico 1. 

**Diagrama de objetos:**

Gráfico 2.

**Síntesis:**

La estrategia elegida fue la mejor opción, pues logro acomodar los datos de forma ordenada y permitir un fácil acceso a lo que la aplicación requería. Además, se logró cumplir con todos los requerimientos que se necesitaban y si había una mejor opción, no debe de diferenciarse mucho de lo que desarrollamos. Algo que si se pudo mejorar es la parte visual y algunos errores, pero por cuestión de tiempo no se alcanzaron a pulir.

**Bibliografía**

* Zonas Wifi. Recuperado de olombiatic.mintic.gov.co/679/w3-propertyvalue-36408.html#:~:text=Ficha%20técnica%20%2F%20Ayuda,con%20la%20oportunidad%20de%20reconexión.