

Universidad Don Bosco.

Facultad de Ingeniería.

Escuela de Electrónica.



“ LAZARILLO ELECTRÓNICO ”

TESIS DE GRADUACIÓN PARA OPTAR AL TÍTULO DE:
Ingeniero en Electrónica.

PRESENTADO POR:

Marvin Noé Artiga Tejada.
Mario Wilfredo Beltrán Fuentes.
Carlos Jonathan López Cisneros.

MARZO DE 2004.

San Salvador, El Salvador C. A.

UNIVERSIDAD DON BOSCO.



RECTOR

ING. FEDERICO MIGUEL HUGUET

VICERECTOR ACADÉMICO

VÍCTOR BERMÚDEZ

SECRETARIO GENERAL

LIC. MARIO RAFAEL OLMOS

DECANO DE LA FACULTAD DE INGENIERÍA

ING. CARLOS BRAN.

UNIVERSIDAD DON BOSCO.
FACULTAD DE INGENIERIA



JURADO EVALUADOR DEL TRABAJO DE GRADUACIÓN.

ING. CARLOS HURTADO.

ING. JULIO RIVERA.

ING. WALTER SANCHEZ

ING. SERGIO MARTIN.

ING. ISRAEL CARDONA.

DEDICATORIA

Agradesco primeramente a nuestro Dios todo poderoso por acompañarme durante toda mi vida e iluminarme el camino al éxito.

*A mis padres **Maximiliano Artiga Bonilla** y **María Graciela de Artiga** a quienes amo mucho. Por darme todo el cariño y apoyo necesario para mi porvenir, ya que gracias a ellos he alcanzado otra meta más y espero no fallarles con mis esfuerzos.*

*A mis hermanos **Wilfredo, Max, Concepción, Sandra, Francisca, Yaneth, Rolando**, por apoyarme en los momentos mas difíciles de la etapa de mis estudios y ayudándome en tomar decisiones importantes en mi vida.*

A gradesco tambien a todas a aquellas persona que confian y creen en mis capacidades ya que gracias a los animos que ellos me brindan tengo deseos de seguir adelante a pesar de las dificultades.

Y por ultimo quiero agradecerle también al asesor por apoyarnos durante todo el tiempo que duro la tesis al igual que a todos los que fueron mis profesores durante toda mi educación.

Marvin Noé Artiga Tejada.

DEDICATORIA

A Dios todo poderoso, pues es quien brinda la sabiduría, nunca me ha desamparado y siempre me ha iluminado como un mentor espiritual. Sin su ayuda y amor no hubiera alcanzado este éxito profesional.

A mis padres **Medardo Beltrán** y **Maura Fuentes de Beltrán**, quienes siempre han creído en mí, fueron y son luz en mi camino. He coronado esta carrera gracias a su apoyo moral, su incansable esfuerzo y amor incondicional. Gracias totales, los amo.

A mi hermana **Flor de María Beltrán**, por su apoyo absoluto y desinteresado. Este triunfo no solo es mío y de mis padres, es también tuyo. Un beso a mis queridas sobrinas **Karen** y **Chelsea**.

A todos los catedráticos, instructores, familiares y demás personas que de una u otra manera me transmitieron sus conocimientos. Muchas gracias a todos.

Mario Wilfredo Beltrán Fuentes.

DEDICATORIA.

Quiero agradecer a DIOS TODO PODEROSO por darme los mejores padres del mundo:

Carlos alberto Lopez Beltran y Reyna Emperatriz Cisneros, ya que ellos han sido el pilar mas grande en mi vida y el mejor ejemplo de dedicacion, esfuerzo, comprensión y amor.

A mis hermanos: Olinda, Fredy, Cesar y Karina, asi como mi novia, por aguantar mi mal humor en fechas de evaluaciones y apoyarme siempre en la búsqueda de mis metas.

A mis familiares, que han apoyado tanto a mis padres como a mi para ayudarme a lograr este triunfo, asi como los que ya no estan con nosotros, pero que nos cuidan desde el cielo, mi abuela Eva Romero (Q.D.D.G.)

Y finalmente a todos mis amigos, amigas y personas que de alguna manera han estado presentes en mí caminar hacia el cumplimiento de esta meta.

Jonathan Cisneros.

ÍNDICE

Introducción.....	3
Objetivos.....	4
Alcances y Limitaciones.....	5
CAPITULO 1: MARCO TEORICO.....	7
• Orígenes.	7
• Definiciones.	7
• Importancia.	11
• Características de las tecnologías.	12
CAPITULO 2: DEFINICIÓN DEL SISTEMA.....	16
• Planteamiento del sistema.	16
• Planteamiento de la solución.	16
• Definición de las interfases físicas.	26
CAPITULO 3: METODOLOGÍA (INVESTIGACIÓN DE CAMPO).....	27
• Recolección de los datos.	27
• Corrección de los datos.	27
CAPITULO 4: DISEÑO E IMPLEMENTACION DE LA APLICACIÓN.....	29
• Definición.	29
• Características.	29
• Requerimientos mínimos de hardware y de software.....	33
• Descripción de los componentes.	33
• Costos de producción y venta.	35

CAPITULO 5: CONCLUSIONES Y RECOMENDACIONES.....	36
Referencias Bibliográficas.....	38
ANEXOS.....	39
• Anexo #1: Manual del programador.....	40
• Anexo #2: Especificaciones técnicas GPS.	101
• Anexo #3: Protocolo de comunicación GPS.	102
• Anexo #4: Mapa de la universidad Don Bosco.	110
• Anexo #5: Glosario.	111
• Anexo #6: Código fuente de la ayuda.	113
• Anexo #7: Manual del usuario.....	127

INTRODUCCIÓN.

El Lazarillo electrónico es la conclusión de un proyecto de tesis orientado a la ayuda de personas no videntes en El Salvador. A continuación se presenta, una pequeña reseña de cada uno de los capítulos que componen el documento.

En primer lugar se presenta el marco teórico de el Lazarillo electrónico, es decir las definiciones básicas (por ejemplo que es latitud, longitud, etc.). La importancia de un proyecto de este tipo y las características de cada una de las tecnologías a utilizar.

El capítulo dos se mostrará en líneas generales, como se concibe será el funcionamiento del sistema y de que partes constara (software y hardware). Se incluyen las diferentes subrutinas con sus respectivos flujogramas.

Enseguida, en el tercer capítulo se aborda el tema de la investigación de campo, de cómo se obtuvo una copia del mapa. Así mismo se da a conocer la forma como se obtiene un factor de corrección de los datos que se obtiene del GPS.

En el capítulo cuatro se presentan la descripción y características de cada componente de la aplicación así como los costos de producción y venta.

Se presentan las valoraciones finales de los estudiantes así como las sugerencias para futuras mejoras al sistema se mostraran en el capítulo cinco

El anexo contiene información de suma relevancia, por que contiene las especificaciones técnicas del GPS, código fuente de la aplicación, etc.

OBJETIVOS.

- Adquirir un GPS capaz de alcanzar niveles de precisión aceptables para la aplicación que se piensa implementar.
- Lograr una conexión entre el GPS con la computadora de tal forma que se obtenga una lectura de los parámetros requeridos (latitud y longitud).
- Generar una base de datos eficiente de manera que el acceso sea fácil y la obtención de datos rápida.
- Implementar un método que pueda guiar a la persona no vidente a través de rutas predefinidas, bajo previa solicitud del usuario.
- Conseguir que el fonema reproducido sea para el usuario inteligible y casi natural, para que no tenga problemas de comprensión a la hora de escuchar las indicaciones
- Realizar una consola de botones la cual sea funcional para el ciego y no presente problemas en su uso o colocación, ya que, en situaciones en que la consulta vía comando de voz fuese dificultosa, será la consola la solución.
- Lograr que el reconocimiento de la voz del usuario sea captado en la menor cantidad de intentos posibles, ya que el sistema será capaz de solicitarle nuevamente el comando si este no es reconocido o no esta definido.
- Lograr finalmente la realización de todos los objetivos anteriores para así obtener una aplicación que sea capaz de realizar todo lo propuesto.

ALCANCES.

- Se espera obtener un alto grado de orientación con respecto a la ubicación en la zona bajo estudio. En otras palabras evitar que usuario no se desvíe de una ruta preestablecida.
- El sistema será capaz de proporcionar al usuario la información solicitada en un formato audible, el cual se espera según el proceso descrito con anterioridad; que sea claro y natural.
- El resultado de la ubicación será en tiempo real; lo que facilitará la velocidad de desplazamiento del usuario.
- No es necesario que la persona tenga grandes conocimientos de computación, ya que el sistema; tendrá un sencillo y fácil uso.
- Que ayude no solo a no videntes, sino también a usuarios que desconozcan la zona bajo estudio

LIMITACIONES.

- El alcance del sistema se aplicará a una zona geográfica establecida, la cual será correspondiente al mapa cargado en el sistema (campus de la Universidad Don Bosco).
- La precisión se espera que este en un rango de tolerancia aceptable para la aplicación, esto dependerá directamente de la calidad del GPS que se obtenga para la aplicación. Además dicha precisión será requerida e lugares más críticos de ubicación o desplazamiento.

- La introducción de información se llevara a cabo por medio de introducir un reducido numero de comandos de voz (entre 5 y 8 comandos) y en circunstancias que estos no apliquen, se cuenta con una consola de botones los cuales proporcionarán el mismo funcionamiento.
- En el método utilizado para desplazar al usuario de un punto a otro, se tendrán muchas rutas, sin embargo el sistema tratará de darle la ruta que presente la menor distancia entre los puntos, según el entorno geográfico.
- Debido a la tecnología que utilizan los GPS, hay puntos en los cuales la señal de estos es demasiado imprecisa.
- Debido a los accidentes geográficos con que cuenta la zona de aplicación se tendrán áreas inaccesibles al usuario, desde el punto de vista que, el sistema asumirá que por su situación, el desplazamiento por estas no es necesario.

CAPITULO 1: MARCO TEORICO.

ORIGENES.

Anteriormente, se han implementado los GPS para aplicaciones de ubicación de sitios, edificios militares y civiles, así como de personas. También en el área de la cartografía es ampliamente utilizado para la elaboración de mapas. Las empresas marítimas usan cada día más esta tecnología de posicionamiento para sus barcos. A la vez, se han hecho muchas aplicaciones en el ramo de motores de voz y / o voces artificiales.

De tal forma, se han combinado los GPS, los teléfonos móviles y computadoras para desarrollar aplicaciones que van mas allá de la simple ubicación geográfica, haciendo mucho más complejas y útiles sus aplicaciones; dando así más soluciones a diversas necesidades.

DEFINICIONES.

El desarrollo de la aplicación se basará principalmente en un conjunto (GPS, software y computadora) que determine la latitud y longitud en donde se encuentra el usuario, la cual se traducirá a puntos de referencia correspondientes al sitio geográfico en que se encuentre.

- Localización geográfica de un punto

Se puede realizar detallando uno de estos dos parámetros:

1. Coordenadas geográficas en formatos de longitud y latitud.
2. Coordenadas (X,Y) UTM. Universal Transversa Mercator

Cada una de estas formas de localizar un punto sobre la superficie terrestre debe cumplir los siguientes requisitos:

- a) El punto a ubicar debe de ser único.
- b) Identificación del sistema de proyección empleado para localizar el punto.

- c) Que permita referenciar la coordenada Z del punto.
 - Latitud y longitud

a) *Latitud:*

Se denomina latitud geográfica (ω) de un punto P al ángulo formado por la vertical a la tierra que pasa por dicho punto con el plano paralelo al ecuador.

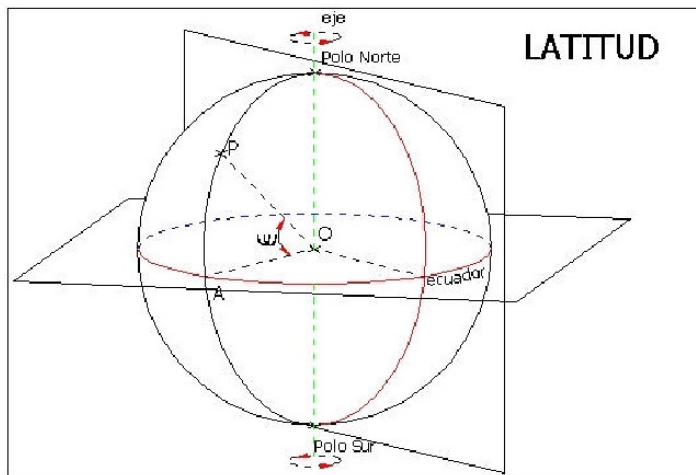


Figura Nº 1. Cálculo de la latitud de un punto.

La vertical se considera la unión del punto con el origen o centro de la tierra, obteniendo la latitud midiendo el ángulo (ω) sobre el meridiano que pasa por el punto P.

$$\omega = \angle OAP$$

El valor de la latitud máxima y mínima va desde los 0° hasta los 90° , ya sea hacia el sur o el norte. Los 90° de latitud coinciden con los polos (norte y sur).

b) *Longitud:*

Se define longitud (λ) de un punto como el valor del ángulo formado por el intercepto del plano meridiano que pasa por P y el meridiano origen (0° meridiano de Greenwich) ver figura N° 2. La longitud es gráficamente el ángulo formado por OAB:

$$\lambda = \angle OAB$$

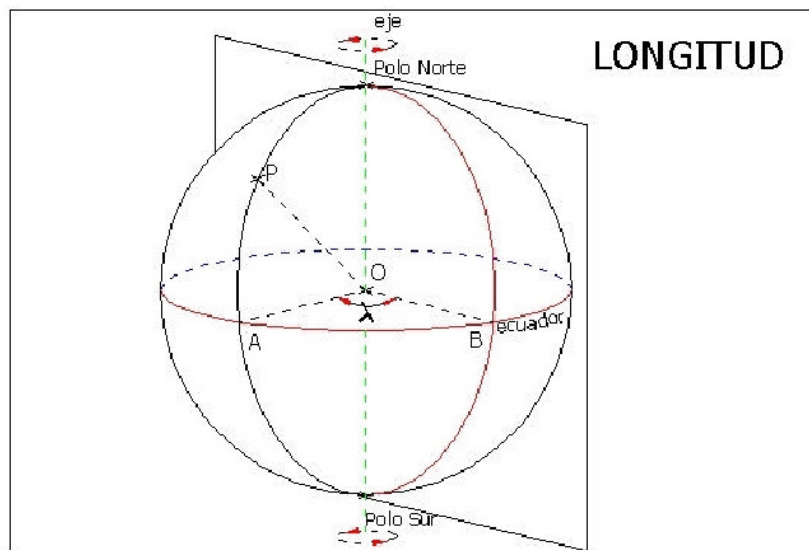


Figura N° 2. Cálculo de la longitud de un punto

La designación de la longitud lleva incluida la elección de la posición espacial del punto con respecto al meridiano origen, así se designa posición Oeste (W) cuando esta a la izquierda y Este (E), cuando esta situada a la derecha. El valor de longitud mínimo posible es 0° hasta un máximo de 180° (tanto Este como Oeste).

- Zona Geográfica Para la Aplicación del Sistema:

Además, debido a que el área de trabajo para la cual se implementará este sistema será el campus de la Universidad Don Bosco; se ha pensado en dividir la zona de aplicación en

puntos críticos de referencia, los cuales serán correspondientes a latitudes y longitudes respectivas y servirán para el trazado de rutas de desplazamientos predeterminadas.

Las rutas determinadas y puntos de referencias dentro de la universidad deberán estar adaptados al mapa geográfico; con el fin de dar un mayor realismo del entorno.

- Lenguajes de Programación:

Debido a la naturaleza del funcionamiento del sistema se tendrá una amplia cantidad de puntos y rutas determinadas, lo que implica una gran cantidad de información que debe ser debidamente almacenada; por ello, es necesario almacenar la información en una base de datos, que contenga campos numéricos y alfanuméricos.

Una herramienta que cuenta con todas estas características y es de uso relativamente sencillo es Microsoft Access, este es un programa para crear y manejar bases de datos.

Por medio de Microsoft Access, se puede administrar toda la información desde un único archivo de base de datos. Dentro del archivo, puede dividir los datos en contenedores de almacenamiento independientes denominados tablas; puede ver, agregar y actualizar datos de la tabla por medio de formularios en pantalla; puede asimismo buscar y recuperar sólo los datos que desee por medio de consultas. Esta base de datos servirá para que cuando el usuario solicite información sobre su posición, se le proporcione (esto implica la ubicación del usuario concerniente a un punto de referencia) que será análogo a la latitud y longitud de dicho punto.

Otro punto a tomar en cuenta, es que por ser un sistema interactivo; se contará con ciertos medios de comunicación entre el usuario y el sistema, ya que este no solo podrá obtener su ubicación, si no que además se ha pensado en que la persona cuente con comandos por medio de voz y de teclado, esto se logrará por medio de Visual Basic. Este es un lenguaje de programación que se disfruta debido a su entorno visual. A veces escribir un programa puede ser un trabajo tedioso; con este lenguaje se reduce el esfuerzo y se vuelve más agradable la

programación, haciendo que muchos aspectos de esta sean tan sencillos como arrastrar con el ratón objetos gráficos en la pantalla.

En realidad Visual Basic es el único lenguaje de programación actual que los programadores principiantes pueden aprender con facilidad.

Debido a todo lo anterior, este lenguaje nos permitirá manejar las consultas a la base de datos, además contiene librerías para la generación y reconocimiento de voz; respecto a este punto se piensa implementar una aplicación basada en dichas librerías para facilitar el avance en esta parte del proyecto, ya que si se diseñan completamente dichos motores voz tomaría mucho más tiempo modelarlos, así como también el uso de otras herramientas más complicadas como redes neuronales y otros tipos de tecnología existentes.

A su vez Visual Basic se utilizará para crear una pequeña aplicación que permita la interacción entre el software propio del GPS y la computadora para lograr así obtener la información correspondiente a latitud y longitud¹.

IMPORTANCIA.

La importancia del Lazarillo electrónico radica en que será una herramienta multimedia que se utilizará; para que una persona, especialmente una no vidente; conozca su posición y que además se desplace a través de un mapa en la dirección que él desee.

También para que las instituciones involucradas en ayudar a personas de este tipo, tomen interés en proyectos como éste; para que las personas con esta incapacidad tengan una mejor adaptación a la sociedad.

Además, un proyecto de esta naturaleza, marca la pauta para que futuras investigaciones logren, basadas en esta; mayores prestaciones multimedia, mejor precisión y ergonomía.

¹ En el anexo #3 se muestra mas información acerca del protocolo de comunicación NMEA

CARACTERÍSTICAS DE LAS TECNOLOGÍAS.

- Introducción a los GPS's

GPS es la abreviatura de NAVSTAR GPS. Este es el acrónimo en Inglés de NAVigation System with Time And Ranging Global Positioning System, (que en Español significa Sistema de Posicionamiento Global con Sistema de Navegación por Tiempo y Distancia).

Durante muchos siglos, se empleo al Sol y las estrellas para navegar. Asimismo, en tierra, los topógrafos y los exploradores utilizaban puntos conocidos hacia los cuales hacían referencia para sus mediciones o para encontrar su camino. Estos métodos cumplían su cometido dentro de ciertos límites, pues el Sol y las estrellas no pueden ser observados cuando el cielo está nublado. Además, aún efectuando las mediciones lo más precisas posibles, la posición no podía ser determinada en forma muy exacta.

Después de la Segunda Guerra Mundial, se hizo necesario que el Departamento de Defensa de los Estados Unidos de Norteamérica encontrara una solución al problema de determinar una posición absoluta y exacta.

Se llevaron a cabo muy diversos proyectos y experimentos, entre los que se cuentan los sistemas Transit, Timation, Loran, Decca etc. Todos ellos permitían determinar posiciones, pero continuaban siendo muy limitados en precisión y funcionalidad.

A principios de los años 70 se propuso un nuevo proyecto - el GPS. Este concepto prometía satisfacer todos los requerimientos del gobierno de los Estados Unidos, principalmente el poder determinar (en cualquier momento y bajo cualquier condición atmosférica), una posición precisa en cualquier punto de la superficie terrestre. El GPS es un sistema basado en satélites artificiales, dispuestos en una constelación de 24 de ellos, para brindar al usuario una posición precisa. En este punto es importante definir el término "precisión". Para un sujeto que se encuentre en el desierto, la precisión significa más o menos 15 m. Para un barco en aguas costeras, la precisión significa 5m. Para un topógrafo, la precisión significa 1cm o

menos. El GPS se puede emplear para obtener todos estos rangos de precisión, la diferencia radicaré en el tipo de receptor a emplear y en la técnica aplicada.

El GPS fue diseñado originalmente para emplearse con fines militares, en cualquier momento y sobre cualquier punto de la superficie terrestre. Poco tiempo después de presentarse las propuestas originales de este sistema, resultaba claro que el GPS también podía ser utilizado en aplicaciones civiles y no únicamente para obtener el posicionamiento personal (como era previsto para los fines militares). Las dos primeras aplicaciones principales de tipo civil fueron aquellas para navegación y topografía. Hoy en día, el rango de aplicaciones va desde la navegación de automóviles o la administración de una flotilla de camiones, la automatización de maquinaria de construcción, hasta un Lazarillo para personas no videntes.

- Motores de voz

Recientemente, los conversores texto-voz están alcanzando un nivel de calidad que posibilita su uso en aplicaciones tanto personales como dirigidas al gran público. Existen algunos sistemas de alta calidad de conversión texto-voz para idiomas como el inglés y el francés, mientras que la mayoría de los disponibles para el español son adaptaciones de los anteriores y no siempre proporcionan una calidad suficiente.

La tarea de los sistemas texto-a-voz (motores de voz), es convertir un texto entero en un discurso que represente justamente el sentido que conlleva el texto.

La generación debe hacerse de forma automática, sin mediar correcciones o ajustes por parte de un operador en ninguna de las etapas del proceso.

La meta de la conversión texto-voz es producir habla emulando, en lo posible, el modo en que un ser humano lee. No bastara que se pueda entender lo que el conversor dice (inteligibilidad), sino que además debe ser apreciado por oyentes humanos como semejante a un hablante humano (naturalidad).

- Historia del reconocimiento de voz

La historia del reconocimiento de voz empezó en el año 1870. Alexander Graham Bell quiso desarrollar un dispositivo que capaz de proporcionar la palabra visible para la gente que no escuchara. Bell no tuvo éxito creando este dispositivo, sin embargo, el esfuerzo de esta investigación condujo al desarrollo del teléfono. Mas tarde, en los años 30 Tihamer Nemes científico Húngaro quiso patentar el desarrollo de una maquina para la transcripción automática de la voz. La petición de Nemes fue negada y a este proyecto lo llamaron poco realista

Fue hasta 1950, 80 años después del intento de Bell, cuando se hizo el primer esfuerzo para crear la primera maquina de reconocimiento de voz. La investigación fue llevada a los laboratorios AT&T. El sistema tuvo que ser entrenad para reconocer el discurso de cada locutor individualmente, pero una vez especializada la maquina tenia una exactitud de un 99% de reconocimiento.

El primer sistema de reconocimiento de voz fue desarrollado en 1952 sobre una computadora analógica que reconocía dígitos del 0 al 9, este sistema era dependiente del locutor. Los experimentos dieron una exactitud de reconocimiento del 98%. Mas tarde, en esa misma época, se creo un sistema que reconocía consonantes y vocales (Dudley,58).

Durante los 60's, los investigadores que trabajaban en el área de reconocimiento de voz empezaron a comprender la complejidad del desarrollo de una verdadera aplicación dentro del reconocimiento de voz, y se comenzaron a realizar aplicaciones con vocabularios pequeños, dependientes del locutor y con palabras de flujo discreto. El flujo discreto, es la forma como hablan los locutores, es decir, con pequeñas pausas entre palabras y frases.

Para los 70's, se desarrollo el primer sistema de reconocimiento de voz comercial. Se mejoraron las aplicaciones de los sistemas dependientes del locutor que requerían una entrada discreta y tenia un vocabulario pequeño. También durante esa época hubo grandes avances

tecnológicos, ya que se cambio del enfoque basado en reconocimiento de patrones a métodos de modelado probabilísticos, como los modelos ocultos de Markov.

Para los 90's los costos de las aplicaciones de reconocimiento de voz continuaron decreciendo y los vocabularios extensos comenzaron a ser normales. También las aplicaciones independientes del locutor y de flujo continuo (lo contrario al flujo discreto, es decir, en el habla no hay pausas significantes) comenzaron a ser más comunes.

- Conversores de texto a voz

Recientemente, los conversores texto-voz están alcanzando un nivel de calidad que posibilita su uso en aplicaciones tanto personales como dirigidas al gran público.

Existen algunos sistemas de alta calidad de conversión texto-voz para idiomas como el inglés y el francés, mientras que la mayoría de los disponibles para el español son adaptaciones de los anteriores y no siempre proporcionan una calidad suficiente.

La tarea de los sistemas texto-a-voz (motores de voz), es convertir un texto entero en un discurso que represente justamente el sentido que conlleva el texto.

La generación debe hacerse de forma automática, sin mediar correcciones o ajustes por parte de un operador en ninguna de las etapas del proceso.

La meta de la conversión texto-voz es producir habla humana, emulando en lo posible, el modo en que un ser humano lee.

No bastara que se pueda entender lo que el conversor dice (inteligibilidad), sino que además debe ser apreciado por oyentes como si fuese un hablante humano (naturalidad).

Afortunadamente, Visual Basic posee una librería relacionada con la conversión de texto a fonemas de la voz humana.

Para este caso, no se utilizará el FreeSpeech 2000, si no que simplemente se utilizará la librería Microsoft Direct Text To Speech API.

CAPITULO 2: DEFINICIÓN DEL SISTEMA.

PLANTEAMIENTO DEL SISTEMA.

El desarrollo de la aplicación se basara principalmente en un conjunto (GPS , software y computadora) que determine la latitud y longitud en donde se encuentra el usuario, la cual se traducirá a direcciones correspondientes al sitio geográfico en que se encuentre; este sitio será correspondiente al mapa cargado en el sistema (campus de la Universidad Don Bosco).

Se definirá una base de datos, la cual servirá para que cuando el usuario quiera desplazarse de un punto A hacia un punto B; le indicara si va en la dirección correcta, esto se realizará mediante un monitoreo continuo de su posición con respecto al entorno.

Otro punto a tomar en cuenta, es que por ser un sistema interactivo; se contará con ciertos medios de comunicación entre el usuario y el sistema, ya que este no solo podrá obtener su ubicación, si no que además se ha pensado en que la persona cuente con comandos por medio de voz, para introducir un punto de destino y que el software sea capaz de decirle al usuario si se acerca o aleja del punto al cual quiere llegar.

PLANTEAMIENTO DE LA SOLUCIÓN.

Para que el sistema funcione adecuadamente se tiene que contar con tres archivos con extensiones bmp, txt, mdb; todos con el nombre de “sitio” los cuales le darán al sistema información de la zona.

- El archivo sitio.bmp contiene el croquis de la zona bajo estudio.
- Sitio.mdb archivo de base de datos que contiene la información correspondiente a las coordenadas.

- Sitio.txt contiene la información de las coordenadas superior derecha e inferior izquierda de toda la zona. Estos datos son utilizados a la hora de generar la base de datos y adaptar el mapa al sistema.

Estos tres archivos son de vital importancia para sincronizar todo el sistema.

Ya teniendo el sistema con estos requerimientos básicos La solución se divide en dos aplicaciones las cuales se plantean a continuación.

- Petición de la ubicación actual.

La persona hará la petición ya sea por medio de un comando de voz “donde estoy”, o por medio de una consola de botones, el software que se diseño tiene la capacidad de identificar este comando.

Lo descrito con anterioridad se muestra en el flujograma a continuación:

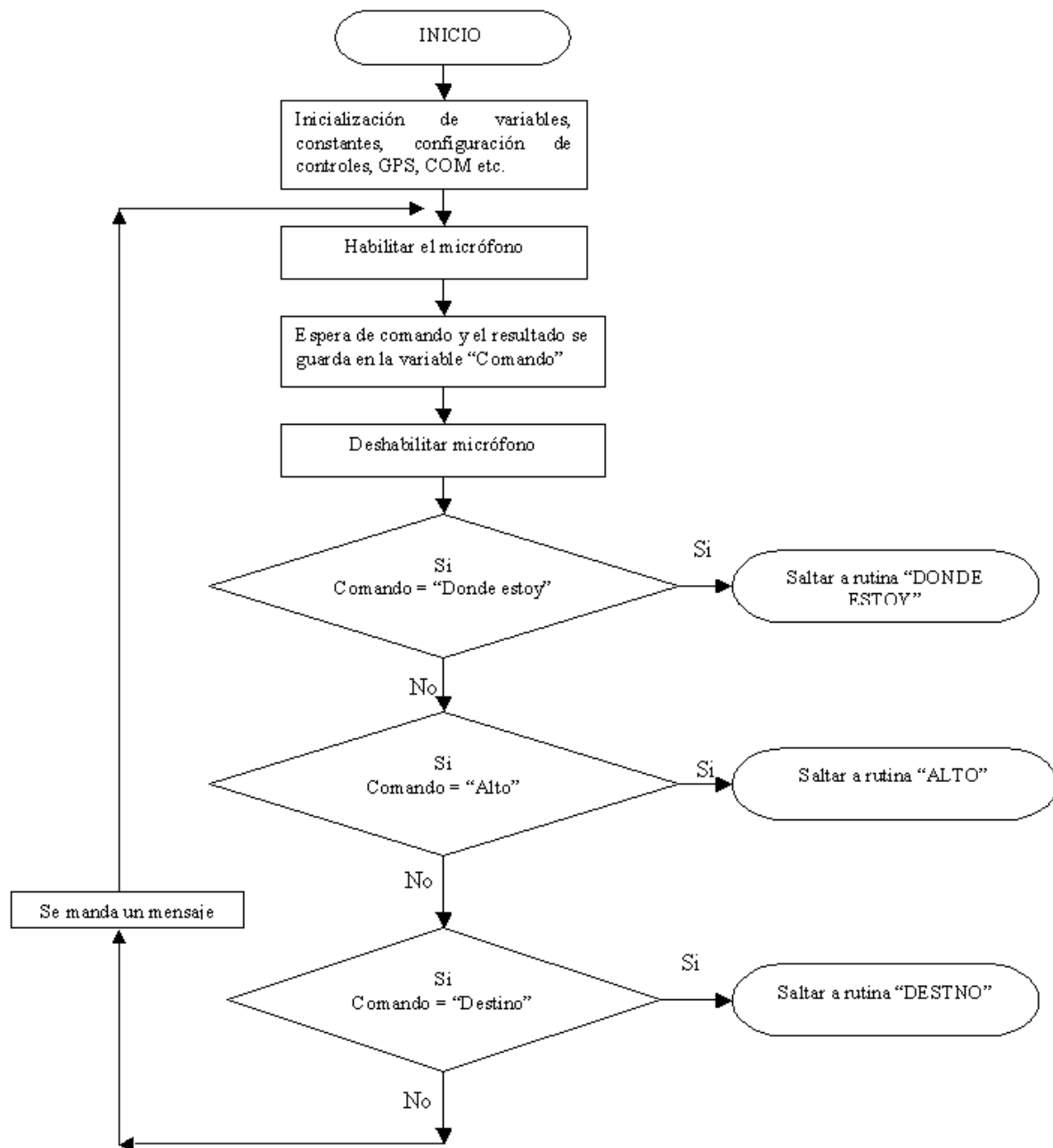


Figura # 3. Diagrama de flujo del programa principal.

Si el comando es "Donde estoy" le indicará que debe realizar una lectura desde el GPS Magellan SporTrak Handheld para después identificar los valores de latitud y longitud en los que se encuentra.

Después estos valores se procesaran en la base de datos, que poseen dos tablas: “lugares” y “caminos”. El motivo de incluir en la base de datos dos tablas, es pensando en la aplicación de elaborar un software, el cual defina la ruta conveniente para el traslado de un lugar a otro y pueda hacer la diferencia entre sitios validos por los cuales si se puede circular. Para obtener el valor análogo pero ya a nivel de un sitio de referencia; Ambas tablas poseen los mismos campos, los cuales se detallan en la siguiente figura.

descripción	Ltminima	Lgminima	Ltmaxima	Lgmaxima	Altitud
edificio a	13.7161484734	89.1534752150	13.7161668865	89.1534570091	0.0000000000
edificio a	13.7161484734	89.1534934210	13.7161668865	89.1534752150	0.0000000000
edificio a	13.7161484734	89.1535116269	13.7161668865	89.1534934210	0.0000000000
edificio a	13.7161484734	89.1535298328	13.7161668865	89.1535116269	0.0000000000
edificio a	13.7161484734	89.1535480387	13.7161668865	89.1535298328	0.0000000000
edificio a	13.7161484734	89.1535662446	13.7161668865	89.1535480387	0.0000000000
edificio a	13.7161484734	89.1535844506	13.7161668865	89.1535662446	0.0000000000
edificio a	13.7161484734	89.1536026565	13.7161668865	89.1535844506	0.0000000000
edificio a	13.7161484734	89.1536208624	13.7161668865	89.1536026565	0.0000000000
edificio a	13.7161300603	89.1534752150	13.7161484734	89.1534570091	0.0000000000
edificio a	13.7161300603	89.1534934210	13.7161484734	89.1534752150	0.0000000000
edificio a	13.7161300603	89.1535116269	13.7161484734	89.1534934210	0.0000000000
edificio a	13.7161300603	89.1535298328	13.7161484734	89.1535116269	0.0000000000
edificio a	13.7161300603	89.1535480387	13.7161484734	89.1535298328	0.0000000000
edificio a	13.7161300603	89.1535662446	13.7161484734	89.1535480387	0.0000000000

Figura # 4. Campos comprendidos en la base de datos.

Descripción: Este campo, tipo texto; contiene el nombre del lugar físico al cuál hará referencia el resultado de la búsqueda.

Ltminima: Es un campo de tipo numérico doble que contiene los valores de latitud minina, correspondiente a la horizontal inferior de cada lugar de referencia, dentro del campo “Descripción”.

Ltmaxima: Es un campo de tipo numérico doble que contiene los valores de latitud máxima, correspondiente a la horizontal superior de cada lugar de referencia, dentro del campo “Descripción”.

Lgminima: Es un campo de tipo numérico doble que contiene los valores de longitud mínima, correspondiente a la vertical izquierda de cada lugar de referencia, dentro del campo “Descripción”.

Lgmaxima: Es un campo de tipo numérico doble que contiene los valores de longitud máxima, correspondiente a la horizontal derecha de cada lugar de referencia, dentro del campo “Descripción”.

Altitud: Es un campo de tipo numérico doble que contiene los valores relativos de altitud de cada lugar de referencia, dentro del campo “Descripción”. El cual será indispensable para diferenciar si se va en ruta ascendente o descendente.

Los campos que contienen los valores de latitud y longitud correspondientes a las zonas, están en grados; la altura tiene valores relativos en metros.

Hay un criterio de búsqueda para encontrar un valor en las tablas de la base de datos la cual se fundamenta en las comparaciones simultáneas basadas en la siguiente regla:

$Ltminima \leq Latitud \leq Ltmaxima \text{ and } Lgminima \leq Longitud \leq Lgmaxima$

Donde:

Latitud: es el valor de latitud proporcionada por el GPS

Longitud: es el valor de longitud proporcionado por el GPS

Los otros parámetros son los nombres de los campos de la base de datos.

Continuando con la petición, este valor textual obtenido de la base de datos se transformara a un fonema de la voz humana por medio de una librería que posee Microsoft SAPI.

Todo lo anterior se resume en una rutina dentro del programa principal, el diagrama de flujo se detalla en la figura #5.

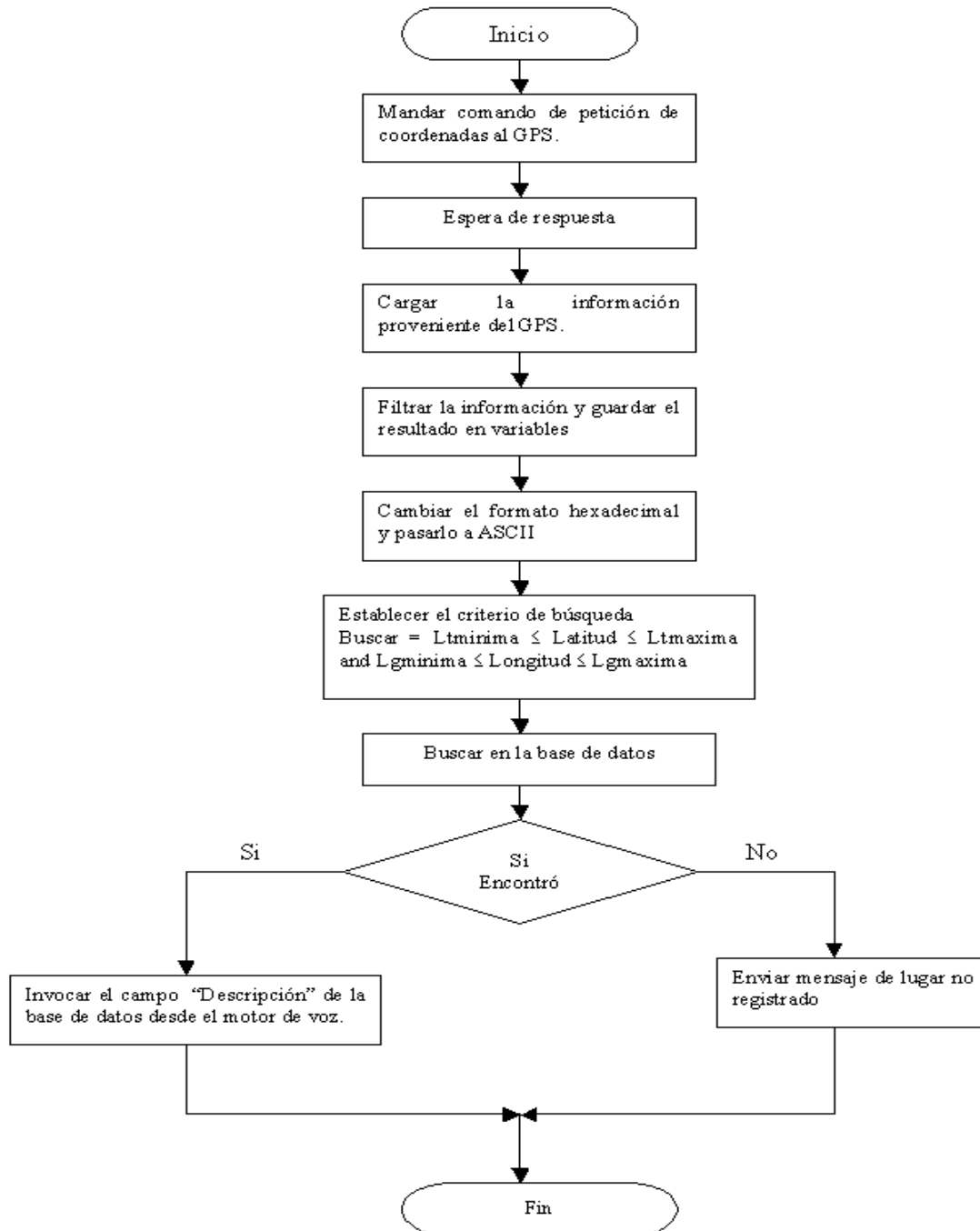


Figura # 5. Diagrama de flujo de la rutina "donde estoy".

- Desplazamiento de un lugar a otro a través de rutas predefinidas.

Para este modo de funcionamiento el usuario introducirá el comando de voz distintivo “destino”. Posteriormente indicara el lugar al que quiere llegar; entonces el sistema lo guiara a través de una ruta determinada. Para esto la rutina se dividirá en tres posibles estados:

a) *Subrutina destino.*

La cual buscara en la base de datos la posición del lugar que el usuario ha introducido a través del micrófono. Una vez identificado, llamara la subrutina de trazado de la ruta. El diagrama de flujo de esta subrutina se presenta en la figura #6

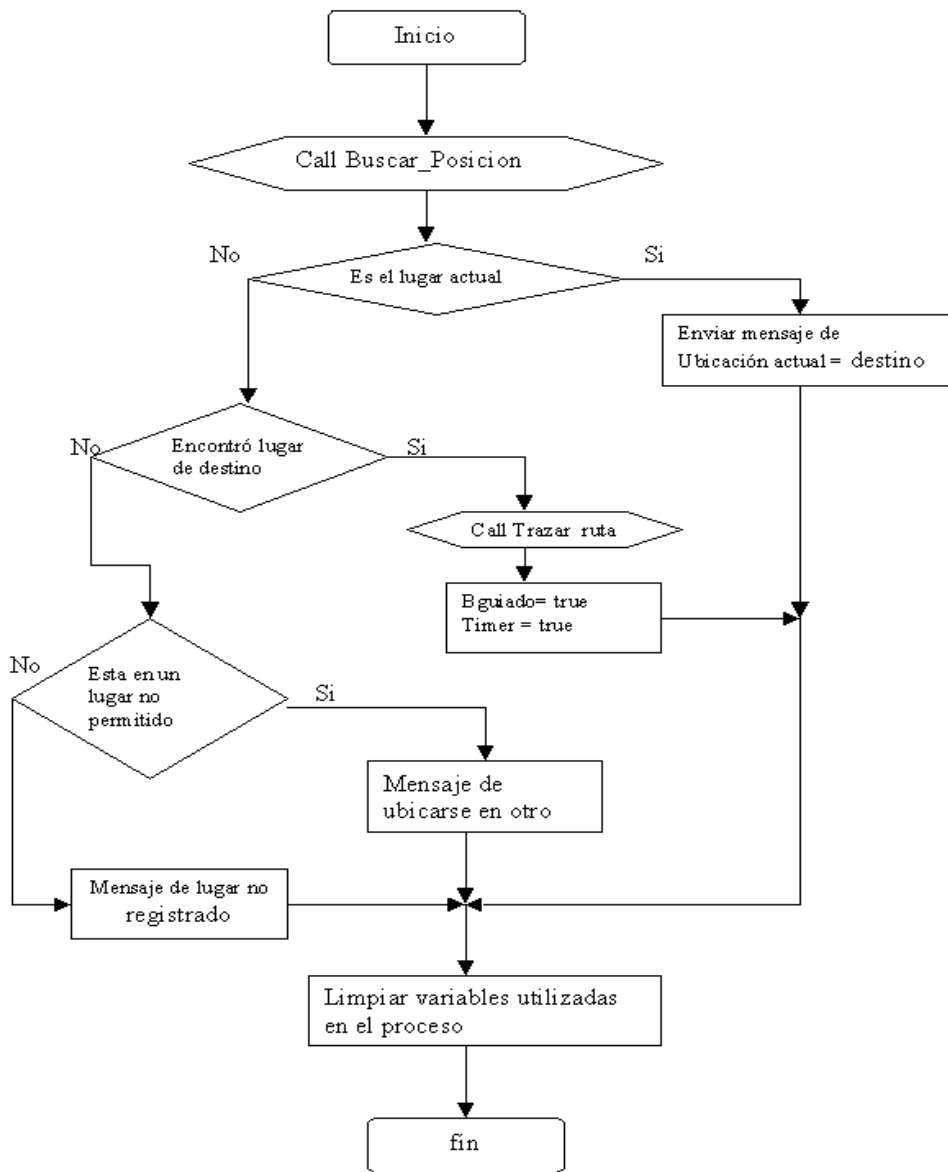


Figura #6. Diagrama de flujo de la subrutina destino.

b) *Subrutina trazado de la ruta.*

Esta subrutina dimensiona las matrices que se utilizan para se llevar el control de lugares de referencia y sus puntos medios. Para determinar que camino resulta mas corto se llama a la subrutina adyacente (diagrama de flujo en la siguiente figura #8) que genera una matriz que se

compara con la de la subrutina de trazado de ruta si la matriz arrojada por adyacente es el lugar de destino la subrutina llega a su fin. En líneas generales la subrutina adyacente, en base a las coordenadas actuales de posición en la que se encuentra hace la selección del lugar adyacente al cual desplazarse. Luego; se comienza a evaluar cada área tomando un punto de ella si esta en el área se toma la distancia hacia el punto de destino y se guarda en una matriz que lleva el control; esto se hace para los cuatro lugares adyacentes para luego hacer una selección de cual es el lugar mas cercano al destino y la que cumple la condición este es el lugar devuelto por la rutina. El diagrama de flujo de la subrutina se muestra en la figura #7.

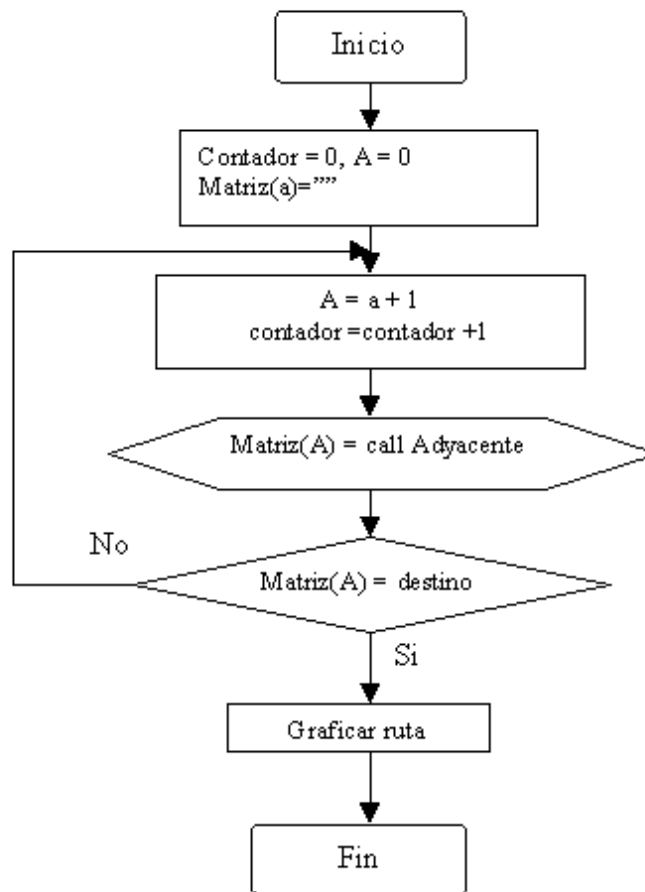
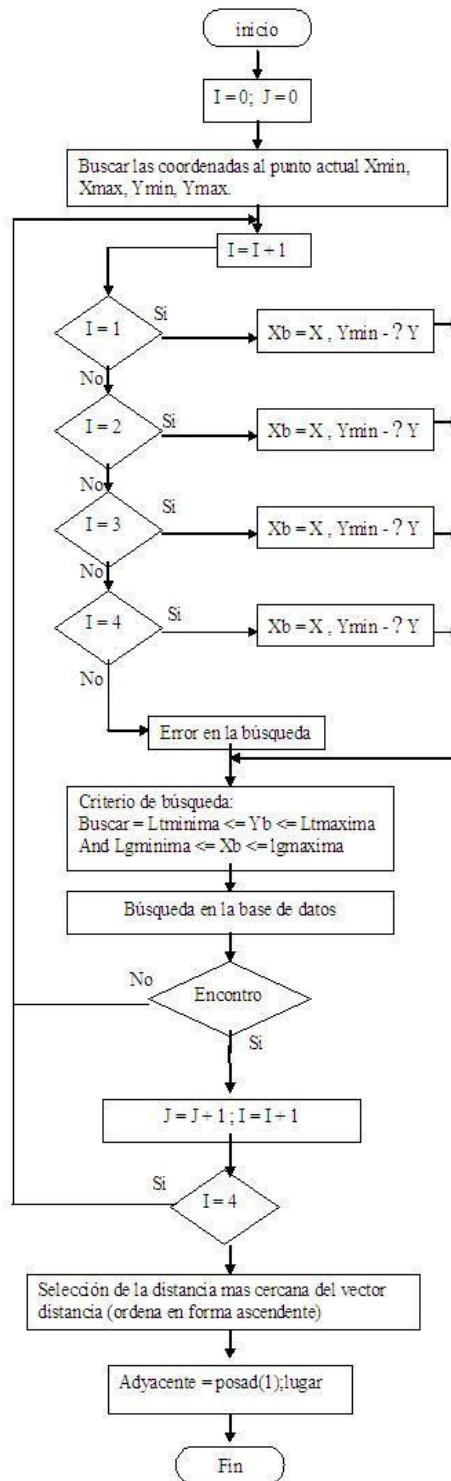


Figura #7. Diagrama de flujo de la subrutina trazado de ruta.

Figura #8. Diagrama de flujo de la subrutina adyacente.



c) *Subrutina de alto.*

Esta subrutina detiene la el proceso de guiado y a su vez llama la rutina donde estoy para que le indique al usuario la nueva ubicación en la que se encuentra. El diagrama de flujo se muestra a continuación, en la figura #9.

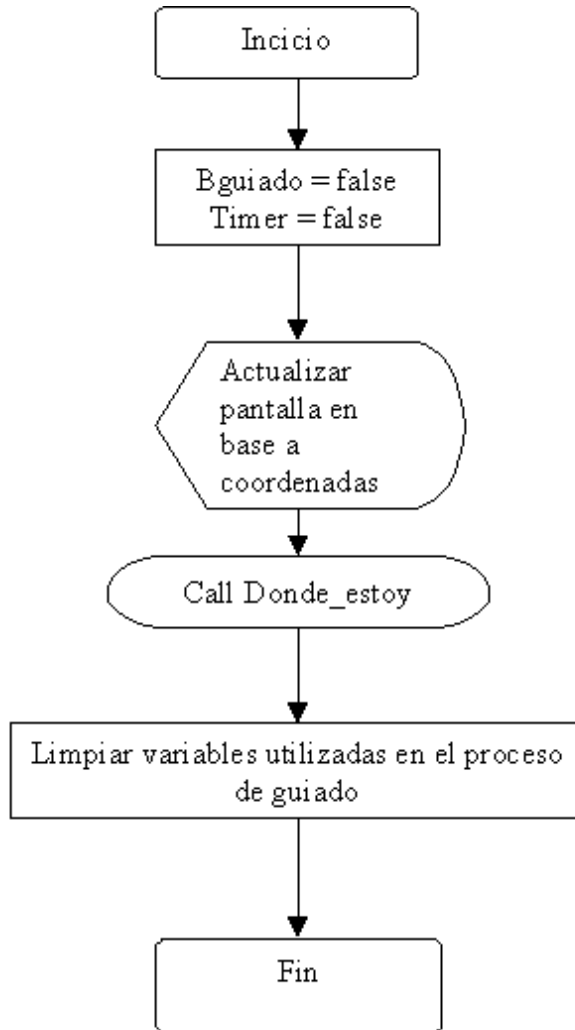


Figura #9. Diagrama de flujo de la subrutina alto.

DEFINICIÓN DE LAS INTERFACES FÍSICAS.

Según lo tratado y detallado en el sumario, se ve la importancia de utilizar dispositivos periféricos aparte de la computadora donde se encontrará almacenada el software y la información, es por ello que en este apartado se justificará el uso de dichos dispositivos.

Debido a que este sistema esta orientado a un cierto grupo de la sociedad (en su mayoría personas no videntes) se ve la necesidad de utilizar algún tipo de dispositivo transductor el cual sirva para transformar señales análogas a acústicas lo cual se logra con una bocina similar a las que poseen los audífonos.

Pensando en que la mayoría de usuarios solamente estén con la incapacidad visual, se piensa utilizar otro tipo de transductor el cual transforme las señales acústicas a señales análogas para que de esta forma el usuario se comuniqué o interactúe de una forma más directa con el sistema lo cual se logra con el uso de un micrófono.

Actualmente en el mercado se encuentran gran variedad de estos dispositivos, sin embargo, se puede obtener ambos en un solo elemento, con el fin de mejorar la comodidad del usuario se usará un “manos libres” (freehand) los cuales son dispositivos mucho mas ergonómicos utilizados actualmente en teléfonos celulares.

Finalmente el complemento de todos estos dispositivos será una computadora (portátil) la cual en estos días es de uso común para una gran variedad de aplicaciones; y el dispositivo por medio del cual se tendrán las coordenadas del usuario será un GPS, del cual se hizo referencia en otro apartado de este documento.

CAPITULO 3: METODOLOGÍA (INVESTIGACIÓN DE CAMPO).

RECOLECCIÓN DE LOS DATOS.

Priemero, se necesito contar con un mapa de la zona de aplicación la cual fue facilitado por la ciudadela Don Bosco ya que está es la zona de estudio. Dicho mapa estaba dividido en tres partes

Se hizo necesario tener el mapa en un formato digital por lo cual se creo en autocad a una escala aproximada 1:1000.

Otro de los factores importantes es tener las coordenadas geográficas del mapa ya que en base a estas es la que se trabajará; para obtenerlas se realizaron pruebas con el GPS en lugares apropiados para luego asociar los valores con la zona de aplicación.

CORRECCIÓN DE LOS DATOS.

A medida se realizaron pruebas con el prototipo de la aplicación fue necesario tomar ciertos consideraciones; como podemos mencionar que hay ciertos puntos donde el GPS proporciona lecturas no muy congruentes de acuerdo a la necesidad, por lo cual se saco un factor de corrección a la hora de realizar las lecturas.

$$F_c = \frac{\text{lectura deseada} - \text{lectura obtenida}}{\text{Lectura deseada}}$$

Esta formula se realizo en puntos críticos para así tomar un promedio y ocupar el factor.

CAPITULO 4: DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN.

DEFINICIÓN.

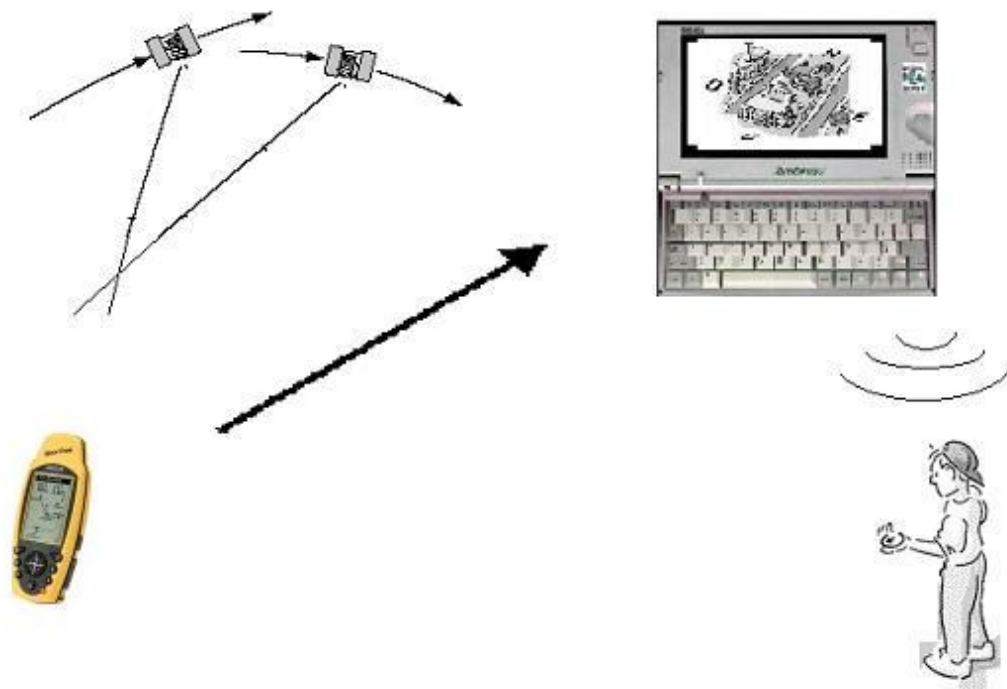


Figura N° 10. Definición de la aplicación.

CARACTERÍSTICAS DE LOS COMPONENTES.

- Funcionamiento del GPS

El sistema NAVSTAR-GPS se basa en la medida simultánea entre el receptor y al menos 4 satélites. El sistema ofrece las siguientes informaciones:

1. Posición del receptor
2. Referencia temporal muy precisa

La distancia entre el receptor y el satélite se obtienen mediante el retardo temporal entre el satélite que envía la señal hasta que el receptor la recibe.

Los satélites emiten dos portadoras a la misma frecuencia. Estas portadoras están moduladas en fase por diferentes códigos pseudoaleatorios. El receptor GPS calcula la correlación entre los códigos del receptor y los códigos enviados por los satélites, de esta forma:

1. Se pueden separar la información de los diferentes satélites
2. Y finalmente se obtiene el retardo temporal

Las trayectorias que estos satélites siguen los mueven en un área que abarca hasta 60° Norte y 60° Sur en latitudes. Esto significa que se puede recibir señales satelitales en cualquier lugar del mundo, en cualquier momento. A medida que se aproxima a los polos, se seguirá captando señal de GPS, simplemente no vendrán directamente de encima del receptor. Esto puede afectar la geometría satelital o la precisión, pero solo un poco.

- ¿Qué tipo de información transmite un satélite GPS?

La señal GPS contiene un código pseudo-rango, efemérides (los cuales dan a la ubicación de los satélites) y datos de almanaque. El código pseudo-rango identifica al satélite que está enviando la señal, en otras palabras transmite la identificación del satélite. Los datos de efemérides son constantemente transmitidos por cada satélite y contienen información como por ejemplo, el status del satélite (buen o mal funcionamiento), fecha actual y hora. Sin esta parte del mensaje, la unidad de GPS no sabrá la fecha ni la hora actual. Los datos de almanaque comunican a la unidad GPS donde se deberían encontrar ubicados cada satélite en todo momento del día. Cada satélite transmite datos comunicando su trayectoria y las de los demás satélites de la red.

Cada satélite transmite un mensaje que básicamente dice: "Soy el GPS N° X, mi posición actual es Y, esta información fue enviada en la hora de zona horaria Z". La unidad de GPS lee el mensaje, guarda las efemérides y la información de almanaque para el uso continuo. Esta información también puede ser utilizada para setear -o corregir- el reloj interno de la unidad.

Luego, para determinar la posición, el GPS comparará la hora en que una señal fue transmitida por un satélite, con la hora en que esa señal fue recibida por la unidad. La diferencia de tiempo comunica al GPS la distancia en que ese satélite se encuentra. Si a esto se suman mediciones de distancias con otros satélites, podemos triangular la posición. Es esto precisamente lo que hace una unidad de GPS. Con un mínimo de tres o más satélites, el GPS puede determinar la posición latitud / longitud.

Un factor que puede afectar la precisión del GPS es la geometría satelital. En términos más sencillos, con "geometría satelital" nos referimos a la ubicación de cada satélite con respecto a los demás satélites (desde la perspectiva de la unidad GPS). Este problema se puede presentar si todos los satélites que en un momento dado proveen de señal a una unidad se encuentran en la misma latitud.

- ¿Qué tan preciso puede ser un GPS?

Un GPS de uso civil standard entrega una precisión de entre 15 y 70 metros, dependiendo del número de satélites disponibles y la geometría de dichos satélites. Unidades más sofisticadas y costosas pueden llegar a entregar una precisión de centímetros utilizando más de una frecuencia. De todas maneras un GPS standard puede mejorar su precisión de 5 metros y a veces hasta menos de un metro mediante un proceso llamado GPS Diferencial (DGPS). El DGPS emplea un segundo receptor para computar correcciones a las mediciones del GPS. Estos servicios están disponibles dependiendo del país y pueden tener un costo extra.

- Reconocimiento y generación de voz (motores de voz)

Como se dijo anterior. En la actualidad, los modelos de interfaz más extendidos son JSAPI (Java Speech API), de Sun Microsystems; SRAPI (Speech Recognition API), de Novell y SAPI (Speech API), de la propia Microsoft.

Un API es un Interfaz de Programación de Aplicaciones, que nos facilita realizar programas de forma independiente de lo que halla a un nivel inferior y en este caso, se trata de unas librerías para hacer programas relacionados con varios aspectos del habla (Speech), ya sea síntesis de voz y / o reconocimiento de esta. Este SAPI ha sido desarrollado por Microsoft y su uso es muy amplio. SAPI está compuesto por varios API's básicos especializados en ciertas tareas:

Voice Command API Reference: Es el interfaz de alto nivel que permite incorporar controles y comandos de voz en las aplicaciones Windows.

Voice Dictation API Reference: Es el interfaz de alto nivel que permite incorporar dictado por voz en las aplicaciones Windows.

Voice Text API Reference: Es el SAPI de alto nivel para incorporar texto-hablado en aplicaciones Windows.

Voice Telephony API Reference: Es el interfaz SAPI de alto nivel para incorporar habla en las aplicaciones de telefonía de Windows.

Direct Speech Recognition API Reference: Es el interfaz de bajo nivel para incorporar reconocimiento de habla en aplicaciones Windows.

Direct Text To Speech API Reference: Es el interfaz de bajo nivel de la SAPI para incorporar texto-hablado en aplicaciones Windows.

Speech Tools Reference: Describe varias herramientas proporcionadas para aumentar la velocidad de desarrollo de la aplicación.

En nuestro caso, los API's que más nos interesan son el "Direct Text To Speech API", el "Direct Speech Recognition API" y el "Voice Text API".

La gran ventaja de SAPI es su compatibilidad con todas las tarjetas de sonido que funcionan en ambiente Windows. Por otra parte, tiene soporte para el software de reconocimiento de voz de IBM, Phillips, Lernout & Hauspie y Dragon Systems.

Para la tesis se utilizará el FreeSpeech 2000 de Phillips, el cual como si dijo anteriormente, utiliza el SAPI .FreeSpeech emplea la herramienta Microsoft Active Accessibility (MSAA) para algunas de las funciones de Comando & Control.²

REQUERIMIENTOS MÍNIMOS DE HARDWARE Y DE SOFTWARE.

Los requerimientos mínimos de hardware para la aplicación, son los siguientes

Procesador Pentium a 200 MHz.

Memoria RAM de 64 MB.

Tarjeta de video de 2MB

Tarjeta de sonido standard.

Los requerimientos de software son los siguientes:

Microsoft Windows 98.

Free Speech 2000.

Microsoft SAPI 4.0

DESCRIPCIÓN DE LOS COMPONENTES.

² <http://www.speech.philips.com/FreeSpeech 2000>.

Los componentes utilizados en la aplicación son los siguientes:

a) *Micrófono y audífonos.*

Son los encargados de ser la puerta de entrada para los comandos de voz del usuario y salida para las instrucciones que genere la aplicación.

b) *GPS.*

Es el dispositivo que recibe las coordenadas de latitud, longitud y altitud que provienen de los satélites.

c) *Computadora portátil.*

Es el dispositivo que se utiliza para albergar el software que necesita la aplicación y

d) *Visual Basic.*

Es el lenguaje de programación en el cual se desarrolla el programa principal

e) *Free speech 2000.*

Es el software que se utiliza para reconocer las instrucciones de petición que el usuario realizara por medio del micrófono.

f) *SAPI 4.*

Es el software que el programa se apoyara para generar las instrucciones que la persona escuchara a través del audífono.

g) *Access*

Es el tipo de software par crear bases de datos y la cual se utiliza para realizar consultas de acuerdo a criterios de búsqueda compatible con visual Basic.

Costos de producción y venta.

Cantidad	Descripción	Precio unitario	Precio total
1	GPS Magellan SportTrack HandHeld	\$212.00	\$212.00
1	Computadora portátil	\$500.00	\$500.00
1	Set de cables de comunicación de GPS	\$23.00	\$23.00
1	Manos libres	\$ 5.00	\$5.00
1	Espiga estereo	\$2.00	\$2.00
1	Mapa de la universidad	\$10.00	\$10.00
	Otros	\$50.00	\$50.00
	Total		\$802
	Precio de venta Total *		\$715.

* El precio final para el usuario variará debido a que este no tendrá la necesidad de adquirir mapas ni herramientas para la elaboración del sistema, y en otros casos este pudiera ya obtener una portátil lo que reduciría aun más el costo.

CAPITULO 5: CONCLUSIONES Y RECOMENDACIONES.

Desde un inicio se tubo una vision de lo que se pretendia realizar como Lazarillo electrónico. A medida que se fue desarrllando el proyecto se cumplieron objetivos y a su vez surgieron algunos incovenientes; varios de ellos tomados desde un inicio como limitantes. otros aparecieron en el camino a raiz de las observaciones realizadas en cada una de las defensas. A continuación se detallan lo que el grupo concluye luego de haber modelado e implemetado la aplicación:

- Se vio la necesidad de que el sistema fuera capaz de adaptarse a cualquier zona geografía. Es por ello que fue necesario que dentro del algoritmo del sistema; se creara un modulo que normalizará la creación de la base de datos, a un nivel de medida estándar para cualquier escala que posea el mapa.
- Después de analizar la forma en que realmente se produciria el fonema de la voz humana, el sistema creado tiene la capacidad de ajustarse a la recepción auditiva de cada usuario. Ademas para complementar esta herramienta; se agrego al sistema la facilidad de variar la velocidad y tonalidad de dicho fonema.
- Luego de experimentar con diferentes metodos de captura de datos provenientes del GPS via puerto serial, se implemento finalmente un metodo en el cual esta lectura, se realiza de forma instantánea. Esto sin tomar en cuenta el tiempo de respuesta del GPS como tal.
- La interacción del sistema con el usuario es de facil uso, esto debido a que la interface grafica cuenta con todas las herramientas que se utilizan en la aplicación. Tambien se considero necesario agregar una ayuda incorporada, la cual, contiene los pasos para ejecutar cualquier comando que el usuario desee implementar; es decir el sistema tiene un entorno amigable, como en un principio se se planteo.

- Debido a que el sistema contiene en su entorno muchas variables, se optó por desallorar una configuración por defecto; para que el sistema funcione con un tipo de voz, zona seleccionada, nombre de usuario, etc., preestablecido. Esto luego de analizar el funcionamiento de cualquier sistema de software en la actualidad.
- Finalmente para complementar la facilidad de la aplicación; se diseñó un archivo ejecutable del Lazarillo electrónico, el cual se instala de la misma forma que otro archivo ejecutable. Este, debido a la información que contiene; reduce los requerimientos de software en la máquina del usuario. Esto complementa la funcionalidad del sistema

El Lazarillo electrónico es un proyecto de tesis que no termina en este punto sino que con la ayuda de las siguientes recomendaciones se puede mejorar:

- Con un GPS de mayor precisión se lograría más eficiencia en la aplicación. Debido a que los valores de latitud y longitud serían más exactos; lo que ayudaría al sistema a determinar con menos margen de error la posición actual del usuario.
- El uso de una PC puede sustituirse por otro dispositivo como PALM que contenga un puerto serial. Para que la portabilidad y ergonomía del sistema, en conjunto, sea mayor. En todo caso, para una computadora portátil, ayudaría el hecho de omitir el uso de dispositivos periféricos que no están incluidos en los requerimientos mínimos del sistema.
- Investigar más acerca de algoritmos de trazado de rutas. Con el objeto de mejorar el prototipo utilizado en esta aplicación, ya que, si bien este funciona, sería bueno considerar la estructura de otros algoritmos que han sido diseñados para un fin similar.
- El uso de equipo multimedia dirigido a trabajar en entornos ruidosos con el objeto de que este no afecte la interacción del usuario y el sistema.

REFERENCIAS BIBLIOGRAFICAS.

- ❖ www.nautigalia.com/gps/
- ❖ www.aero.org/publications/GPSPRIMER/
- ❖ www.tel.uva.es/~jpozdom/telecomunicaciones/portadagps.html
- ❖ Manual de usuario GPS Magellan SportTrack HandHeld
- ❖ <http://www.speech.philips.com/FreeSpeech> 2000

ANEXOS.

Manual del Programador.



Manual del Programador Lazarillo Electrónico.

Sistema para determinar la posición del usuario dentro de una zona bajo estudio, basado en un sistema de posicionamiento global. (GPS)

FORMULARIOS

Nombre	Archivo	Descripción
FrmPrincipal	FrmPrincipal.frm	Formulario principal que muestra la aplicación, con una barra de tareas.
FrmVelocidad	FrmVelocidad.frm	Formulario en la cual se muestra las funciones de configuración de tipo de voz.
FrmMantenimientoG	FrmMantenimientoG.frm	Formulario que es utilizado para poder hacer modificaciones de la base de datos en forma grafica.
FrmMantenimientob	FrmMantenimientob	Formulario en la cual se muestra los datos que están almacenados en la base de datos.
FrmAcerca	FrmAcerca.frm	Formulario que muestra la información de la versión y datos de contacto de los creadores.
Form1	FrmNombrar.frm	Formulario que es utilizado pa agragar zonas a la base de datos en forma grafica.
Form2	FrmCrearbase.frm	Formulario que crea los datos que contienen cada lugar de la zona de aplicación.

MODULOS

Nombre	Archivo	Descripción
MdlCofiguración	MdlCofiguración.bas	Contine las rutinas de configuración del sistema.
Mdlcrearbase	Mdlcrearbase.bas	Contiene las rutinas que se utilizan para crear la base de datos asi como tambien para nombrar las zonas.
MdlGuiado	MdlGuiado.bas	Modulo que contiene las rutinas que son utilizadas a la hora de guiar al usuario cuando se encuentra en modo de guiado.

MdlRComandos	MdlRComandos.bas	Modulos que contiene las rutinas “donde estoy”, ”destino”, “alto” entre otras utilizadas para el mismo propósito.
mdlRuta	MdlRuta.bas	Modulo que contiene las rutinas de trazado de rutas y otros calculos matemáticos para realizar el mismo fin.
MdlVariables	MdlVariables.bas	Modulo donde estan definidas todas las variables publicas utilizadas en la aplicación.

PROCEDIMIENTOS

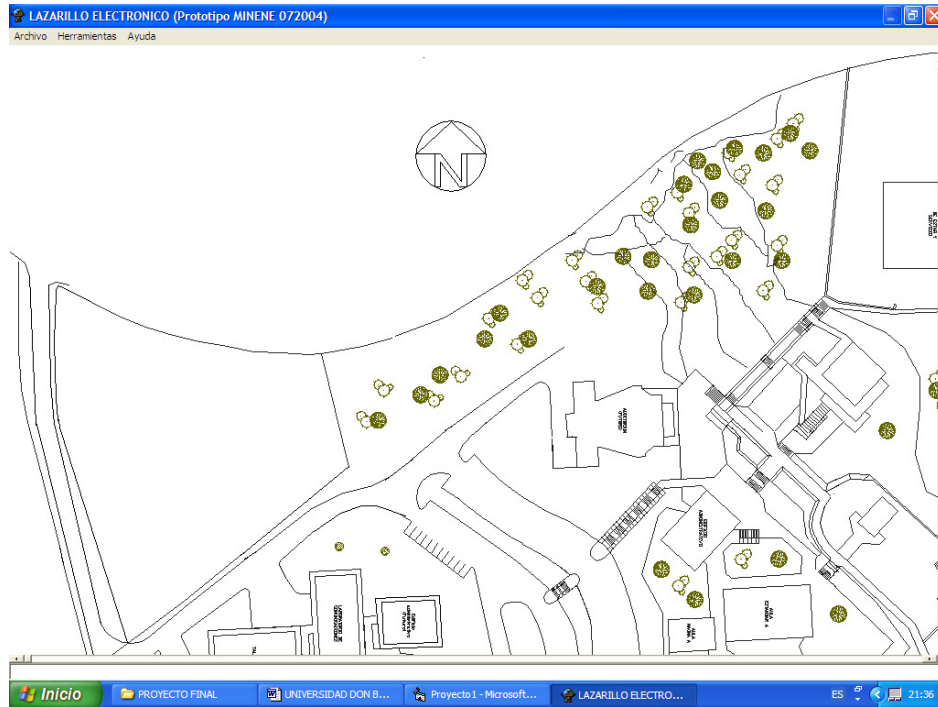
Nombre	Parámetros que recibe	Descripción de la funcion.
Configuracion_GPS	Ninguno	Configura el gps para establecer la comunicación.
NombrarArea	Valores de extremos de la zona que se quiere nombrar	Asigna un nombre pre establecido a un area asignada.
NombrarLineal	Punto que se quiere nombrar	Asigna un nombre al punto que se selecciona.
Guiado	Un entero que indica el lugar de referencia en la que se encuentra.	Dirige al usuario cuando se encuentra en modo de guiado.
Alto	Ninguno	Detiene el proceso cuando se encuentra en modo de guiado.
destino	Puntos de salida y puntos de destino	Hace la petición de un lugar de destino para calcular la ruta a seguir.
Get_GPS	Ninguno	Hace la petición al gps.
Ayuda_Audible	Ninguno	Rutina que pregunta que tipo de ayuda quiere recibir el usuario.
TrazarRuta	Parámetros de coordenadas tanto del destino como el lugar actual en el que se encuentra	Calcula la ruta posible a seguir.
Adyacente	Coordenadas del punto actual	Decide el lugar de referencia al cual desplazarse en base a distancia.

Graficar_ruta	Vector de lugares de referencia	Grafica la ruta el cual se tiene que seguir.
donde_estoy	Valores de latitud y longitud	Hace la petición al gps y busca el lugar de referencia en una base de datos.

FUNCIONES

Nombre	Parámetros que devuelve	Descripción de la función.
Lado	Los grados al cual desplazarse y la dirección.	Calcula la dirección al cual se tiene que orientar.
grados	Grados de desplazamiento	Calcula los grados para la orientación
Dist	Distancia entre el punto actual y el destino	Calcula la distancia entre dos puntos.
Buscar_Posición	Parámetros de destino(nombre del lugar de destino)	Se utiliza para buscar las coordenadas del lugar de destino.

Formulario principal



El código siguiente es utilizado para poder visualizar el formulario anterior

```
Dim Constx As Double
Dim Consty As Double
Dim Programa As Double
Dim maxscroll As Double
Dim Comando As String
Dim MoverX As Boolean
Dim MoverY As Boolean
Dim my As Double
Dim mx As Double
```

```
Private Sub Form_Load()
'recuperación de configuración
```

```
Call Cargar 'carga la los ultimos cambios modificados desde la ultima vez
'que se utilizo el sistema.
```

```
DirectSS1.Visible = False
Operadores
```

```
Call Configuracion_GPS
```

```
'comparación para determinar si hay registros
```

```
If Base = "" Or BMP = "" Or latmin = "" Or latmax = "" Or lonmin = "" Or lonmax = "" Then
    Seleccionar_Zona 'rutina que es utilizada para cargar la base de datos
                    'y el mapa correspondiente con su escala
```

End If

'configuración del control Data

Data1.DatabaseName = Base

Data1.RecordSource = "Caminos"

Data1.Refresh

'cargando el mapa correspondiente

'BMP trae la dirección correspondiente

Picture2.Picture = LoadPicture(BMP)

'configuración de controles del formulario principal

FrmPrincipal.ScaleMode = vbPixels

Picture1.ScaleMode = vbPixels

'procedimiento para que el cuadro de imagen del fondo tome

'el tamaño original del mapa.

Picture2.AutoSize = True

'determinando la escala, las variables tienen el valor

'correspondiente a las coordenadas del mapa

DlatY = Abs(Val(latmax) - Val(latmin)) / Val(DLat)

DlonX = Abs(Val(lonmin) - Val(lonmax)) / Val(DLon)

Picture2.Scale (Val(lonmin), Val(latmax))-(Val(lonmax), Val(latmin))

Picture1.BorderStyle = 0

Picture2.BorderStyle = 0

Call Actualizar_formulario 'procedimiento que actualiza el formulario

'-----

'bloque de configuración de inicialización de variables

'variables que se utilizan a la hora del guiado

Bguiado = False

Bguiado1 = False

Timer1.Enabled = False

'condiciones de inicio del motor de voz.

DirectSS1.CurrentMode = Val(Voz) 'tipo de voz

DirectSS1.Speed = Val(Velocidad) 'rapidez de habla

Indicador = 0

'bandera que se utiliza para control del foco

'del cuadro de texto de comandos

Bfoco = True

*'bandera que se pone en True a la hora de dictar el lugar
'de destino*

Bdestino = False

'procedimiento para abrir freespeech

Programa = Shell("C:\Archivos de programa\FreeSpeech 2000\BIN\free.exe",
vbNormalFocus)

End Sub

Private Sub Form_LostFocus()

If Bfoco Then

TxtComandos.SetFocus

End If

End Sub

Private Sub Form_Resize()

Picture1.Height = FrmPrincipal.Height

Picture1.Width = FrmPrincipal.Width

Call Actualizar_formulario *'llamada para actualizar formulario*

End Sub

Private Sub Form_Unload(Cancel As Integer)

'llamada a rutina que guarda la configuración actual

Guardar_Configuración

End Sub

Private Sub mnuArchivoCongifuracion_Click()

Data1.DatabaseName = Base

Data1.RecordSource = "Caminos"

Data1.Refresh

Picture2.Picture = LoadPicture(BMP)

FrmPrincipal.ScaleMode = vbPixels

Picture1.ScaleMode = vbPixels

Picture2.AutoSize = True

DlatY = Abs(Val(latmax) - Val(latmin)) / Val(DLat)

DlonX = Abs(Val(lonmin) - Val(lonmax)) / Val(DLon)

Picture2.Scale (Val(lonmin), Val(latmax))-(Val(lonmax), Val(latmin))

Picture1.BorderStyle = 0

Picture2.BorderStyle = 0

Call Actualizar_formulario

Voz = "3"

DirectSS1.CurrentMode = Val(Voz) *'tipo de voz*

Velocidad = "175"

DirectSS1.Speed = Val(Velocidad) *'rapidez de habla*

End Sub

```
Private Sub mnuarchivonombre_Click()  
' solicita el nombre de usuario  
Usuario = InputBox("favor digite el nombre de usuario")  
End Sub
```

```
Private Sub mnuarchivosalir_Click()  
' emite un sonido  
Beep
```

```
' llamada a rutina que guarda la configuración actual  
Call Guardar_Configuración  
End  
End Sub
```

```
Private Sub mnuherramientasajuste_Click()  
' quita el foco del cuadro de texto y muestra  
' el formulario de configuración del motor de voz  
Bfoco = False  
FrmVelocidad.Show  
End Sub
```

```
Private Sub mnuherramientasmantenimientocrearbase_Click()  
Bfoco = False  
Form1.Show  
End Sub
```

```
Private Sub mnuherramientasmantenimientoModo_Click()  
' muestra el formulario de mantenimiento grafico  
Bfoco = False  
FrmMantenimientoG.Show  
End Sub
```

```
Private Sub mnuherramientasmantenimientoVerbase_Click()  
' muestra el formulario con la base de datos  
Bfoco = False  
FrmMantenimientob.Show  
End Sub
```

```
Private Sub mnuherramientasseleccionar_Click()
```

```
Call Seleccionar_Zona ' rutina para seleccionar la zona de aplicación
```

```
' bloque que actualiza el control Data y formulario
```

```
Data1.DatabaseName = Base
Data1.Refresh
Picture2.Picture = LoadPicture(BMP)
FrmPrincipal.ScaleMode = vbPixels
Picture1.ScaleMode = vbPixels
Picture2.AutoSize = True
' escala
Picture2.Scale (Val(lonmin), Val(latmax))-(Val(lonmax), Val(latmin))
Operadores
' llamada a rutina que actualiza el formulario
Picture1.BorderStyle = 0
Picture2.BorderStyle = 0
Call Actualizar_formulario
End Sub

Private Sub Picture2_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If Button = 1 Then
        'Activar el microfono
        TxtComandos.SetFocus
    ElseIf Button = 2 Then
        'llamar a rutina donde estoy
        Call Alto
    End If
End Sub

End Sub

Private Sub Picture2_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    TxtComandos.SetFocus
End Sub

Private Sub Picture2_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If Button = 1 Then
        ' desactivar el micrófono
    End If
End Sub

Private Sub Picture2_Paint()
    If Bguiado Then
        Call Graficar_ruta
    End If
    If Bfoco Then
        TxtComandos.SetFocus
    End If
End Sub
```

End Sub

Private Sub Timer1_Timer()

'por el momento esto sirve para saber la posición

Dim G As String

Dim Lad As String

Call Get_GPS(Longitud, Latitud)

If Longitud <> 0 And Latitud <> 0 Then

Call Mover_Pantalla

If Bguiado1 Then

Call Orientar(G, Lad)

If Lad <> "adelante" And Lad <> "" Then

DirectSS1.Speak Usuario & " desplazarse " & G & " hacia " & Lad

By = True

Bx = True

Bguiado1 = False

Bguiado = True

ElseIf G = "" And Lad <> "" Then

Bguiado1 = False

Bguiado = True

End If

End If

'bandera Bguiado se activa ala hora de guiar

If Bguiado Then

Call indicar

End If

End If

TxtComandos.SetFocus

End Sub

Private Sub TxtComandos_KeyPress(KeyAscii As Integer)

'por el momento se hace así para reconocer la voz por medio

'del programa freespeech

'comparación si el carácter es enter

If KeyAscii = vbKeyReturn Then

KeyAscii = 0

TxtComandos.Text = LTrim(TxtComandos.Text)

'comparación si la bandera destino es verdadero

'que se pone en True cuando el comando de voz es destino

If Bdestino Then

Bdestino = False

Ldestino = TxtComandos.Text

'llamada a rutina que determina el destino

Call Destino

Else

Comando = UCase(TxtComandos.Text)

'identificación del comando de voz

Select Case Comando

Case Is = "DONDE ESTOY"

Call donde_estoy

Case Is = "ALTO"

Call Alto

Case Is = "DESTINO"

Bdestino = True

DirectSS1.Speak "favor diga lugar de destino"

Case Else

DirectSS1.Speak "Comando no válido, repita por favor"

End Select

TxtComandos.Text = ""

End If

End If

TxtComandos.SetFocus

End Sub

Sub Actualizar_formulario()

'rutina que actualiza el formulario a las nuevas condiciones

Picture1.Move 0, 0, ScaleWidth - VScroll1.Width, ScaleHeight - HScroll1.Height -

TxtComandos.Height

Picture2.Move 0, 0

HScroll1.Top = Picture1.Height

HScroll1.Left = 0

HScroll1.Width = Picture1.Width

TxtComandos.Top = Picture1.Height + HScroll1.Height

TxtComandos.Left = 0

TxtComandos.Width = Picture1.Width + VScroll1.Width

VScroll1.Top = 0

VScroll1.Left = Picture1.Width

VScroll1.Height = Picture1.Height

HScroll1.Max = Picture2.Width - Picture1.Width

VScroll1.Max = Picture2.Height - Picture1.Height

VScroll1.Visible = (Picture1.Height < Picture2.Height)

```
HScroll1.Visible = (Picture1.Width < Picture2.Width)

MoverY = (Picture1.Height < Picture2.Height)
MoverX = (Picture1.Width < Picture2.Width)

mx = (HScroll1.Max - 0) / (Val(lonmax) - Val(lonmin))
Constx = -Val(lonmin) * mx

my = (0 - VScroll1.Max) / (Val(latmax) - Val(latmin))
Consty = VScroll1.Max - my * Val(latmin)
End Sub
Sub Seleccionar_Zona()
' rutina que se utiliza para seleccionar la zona de aplicación
Dim Escala As String
Dim algo As String
Seleccionarbase:
DlgBase.ShowOpen
Algo = DlgBase.FileName
If algo <> "" Then
Base = CurDir + "\Sitio.mdb"
BMP = CurDir + "\Sitio.bmp"
Escala = CurDir + "\Sitio.txt"
Open Escala For Input As #1
Input #1, lonmin, latmin, lonmax, latmax, DLat, DLon
Close #1
Else
If Base = "" Then
GoTo seleccionarbase:
End If
End If
DlatY = Abs(Val(latmax) - Val(latmin)) / Val(DLat)
DlonX = Abs(Val(lonmin) - Val(lonmax)) / Val(DLon)
End Sub
Sub Guardar_Configuración()
'guarda la configuración actual
MSComm1.PortOpen = False
SaveSetting "Lazarillo", "Variables", "Usuario", Usuario
SaveSetting "Lazarillo", "Variables", "Base", Base
SaveSetting "Lazarillo", "Variables", "BMP", BMP
SaveSetting "Lazarillo", "Variables", "Longitudmin", lonmin
SaveSetting "Lazarillo", "Variables", "Latitudmin", latmin
SaveSetting "Lazarillo", "Variables", "Longitudmax", lonmax
SaveSetting "Lazarillo", "Variables", "Latitudmax", latmax
SaveSetting "Lazarillo", "Variables", "Dlat", DLat
SaveSetting "Lazarillo", "Variables", "Dlon", DLon
SaveSetting "Lazarillo", "Variables", "Velocidad", Velocidad
SaveSetting "Lazarillo", "Variables", "Voz", Voz
```

```
' cierra el programa freespeech
AppActivate Programa
SendKeys "%{F4}", True
SendKeys "~", True
End Sub
Sub Cargar()
'carga los registros con la ultima modificación que se le hizo
Usuario = GetSetting("Lazarillo", "Variables", "Usuario", "Marvin")
Base = GetSetting("Lazarillo", "Variables", "Base", "")
BMP = GetSetting("Lazarillo", "Variables", "BMP", "")
lonmin = GetSetting("Lazarillo", "Variables", "Longitudmin", "")
latmin = GetSetting("Lazarillo", "Variables", "Latitudmin", "")
lonmax = GetSetting("Lazarillo", "Variables", "Longitudmax", "")
latmax = GetSetting("Lazarillo", "Variables", "Latitudmax", "")
DLat = GetSetting("Lazarillo", "Variables", "Dlat", "")
DLon = GetSetting("Lazarillo", "Variables", "Dlon", "")
Velocidad = GetSetting("Lazarillo", "Variables", "Velocidad", "120")
Voz = GetSetting("Lazarillo", "Variables", "Voz", "3")

End Sub
Private Sub TxtComandos_LostFocus()
If Bfoco Then
TxtComandos.SetFocus
End If
End Sub
Sub Mover_Pantalla()
If MoverX Then
If MoverY Then
VScroll1.Value = (my * Latitud + Consty)
Picture2.Top = -VScroll1.Value
End If
HScroll1.Value = (mx * Longitud + Constx)
Picture2.Left = -HScroll1.Value
Else
If MoverY Then
VScroll1.Value = (my * Latitud + Consty)
Picture2.Top = -VScroll1.Value
End If
End If
End Sub
```

Este es formulario que muestra la base de datos

Mantenimiento de la base de datos						
Id	descripción	Ltminima	Lgminima	Ltmaxima	Lgmaxima	Altitud
	estamos probando	0.0001	0.000001	0.0000001	0.0000001	0.0000001
	estamos probando	13.71404938	89.15644278	13.7140677931	89.15642457408	0
	estamos probando	13.71404938	89.15642457408	13.7140677931	89.15640636816	0
	estamos probando	13.71404938	89.15638816224	13.7140677931	89.15638816224	0
	estamos probando	13.71404938	89.15636995632	13.7140677931	89.15636995632	0
	estamos probando	13.71404938	89.1563517504	13.7140677931	89.1563517504	0
	estamos probando	13.71404938	89.15633354448	13.7140677931	89.15633354448	0
	estamos probando	13.71404938	89.15631533856	13.7140677931	89.15629713264	0
	estamos probando	13.71404938	89.15629713264	13.7140677931	89.15627892672	0
	estamos probando	13.71404938	89.15627892672	13.7140677931	89.1562607208	0
	estamos probando	13.71404938	89.1562607208	13.7140677931	89.15624251488	0
	estamos probando	13.71404938	89.15624251488	13.7140677931	89.15622430896	0
	estamos probando	13.71404938	89.15622430896	13.7140677931	89.15620610304	0
	estamos probando	13.71404938	89.15620610304	13.7140677931	89.15618789712	0
	estamos probando	13.71404938	89.15618789712	13.7140677931	89.1561696912	0
	estamos probando	13.71404938	89.1561696912	13.7140677931	89.15615148528	0
	estamos probando	13.71404938	89.15615148528	13.7140677931	89.15613327936	0
	estamos probando	13.71404938	89.15613327936	13.7140677931	89.15611507344	0

Selección del tipo de zona a realizar los cambios.

Caminos
 Lugares

Código utilizado para visualizar el formulario anterior.

```

Private Sub Form_Activate()
FrmPrincipal.Enabled = False
'carga labase de datos y la tabla a mostrar
Data1.DatabaseName = Base
Data1.RecordSource = "Caminos"
Data1.Refresh
End Sub

```

```

Private Sub Form_Unload(Cancel As Integer)
'pone Bfoco en true para darle el foco al cuadro de texto
'del formulario principal
Bfoco = True
FrmPrincipal.Enabled = True
End Sub
Private Sub OptCaminos_Click()
'Data1.DatabaseName = Base
Data1.RecordSource = "Caminos"
Data1.Refresh
End Sub

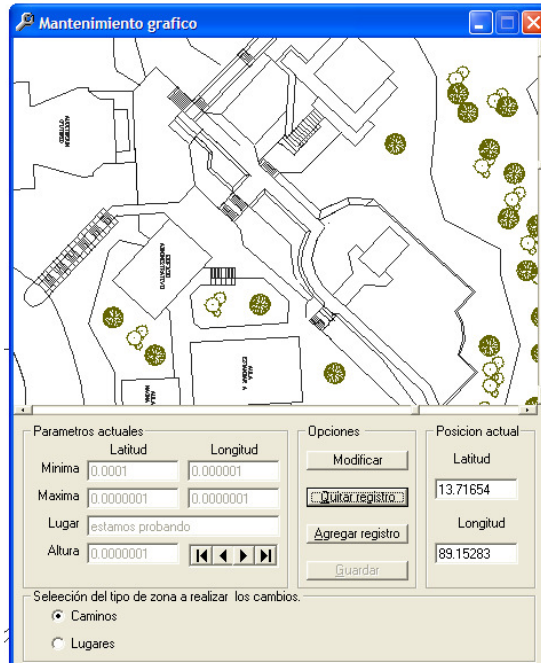
```

```

Private Sub OptLugares_Click()
'Data1.DatabaseName = Base
Data1.RecordSource = "Lugares"
Data1.Refresh
End Sub

```

Formulario que se muestra en modo grafico



Código utilizado para visualizar el formulario anterior.

```
Dim Constx As Double
Dim Consty As Double
Dim my As Double
Dim mx As Double
Dim MoverY As Boolean
Dim MoverX As Boolean
Private Sub CmdAgregar_Click()
    'agregar un nuevo registro
    Text1.Enabled = True
    Text2.Enabled = True
    Text3.Enabled = True
    Text4.Enabled = True
    Text5.Enabled = True
    Text6.Enabled = True
    Data1.Recordset.AddNew
    CmdAgregar.Enabled = False
    CmdGuardar.Enabled = True
    CmdQuitar.Enabled = False
    CmdModificar.Enabled = False
End Sub
```

```
Private Sub CmdGuardar_Click()
    'guarda registr nuevo
```

```
CmdAgregar.Enabled = True
CmdQuitar.Enabled = True
CmdGuardar.Enabled = False
CmdModificar.Enabled = True
Data1.Recordset.Update
FrmPrincipal.Data1.Refresh
Text1.Enabled = False
Text2.Enabled = False
Text3.Enabled = False
Text4.Enabled = False
Text5.Enabled = False
Text6.Enabled = False
End Sub
```

```
Private Sub CmdModificar_Click()
If CmdModificar.Caption = "Modificar" Then
Text1.Enabled = True
Text2.Enabled = True
Text3.Enabled = True
Text4.Enabled = True
Text5.Enabled = True
Text6.Enabled = True
CmdModificar.Caption = "Cerrar"
CmdQuitar.Enabled = False
CmdAgregar.Enabled = False
Else
Text1.Enabled = False
Text2.Enabled = False
Text3.Enabled = False
Text4.Enabled = False
Text5.Enabled = False
Text6.Enabled = False
CmdModificar.Caption = "Modificar"
CmdQuitar.Enabled = True
CmdAgregar.Enabled = True
End If
End Sub
```

```
Private Sub CmdQuitar_Click()
Dim Mensaje As Integer
'procedimiento para quitar un registro actual
Mensaje = MsgBox("Desea Quitar el registro actual", 1 + 35, _
Usuario)
If Mensaje = 6 Then
Data1.Recordset.Delete
End If
End Sub
```

```
Private Sub Form_Activate()  
CmdGuardar.Enabled = False  
Text1.Enabled = False  
Text2.Enabled = False  
Text3.Enabled = False  
Text4.Enabled = False  
Text5.Enabled = False  
Text6.Enabled = False  
'configurando el control data  
Data1.DatabaseName = Base  
Data1.RecordSource = "Caminos"  
Data1.Refresh
```

'enlazando los cuadros de textos con el control data

```
Text1.DataField = "Ltminima"  
Text2.DataField = "Ltmaxima"  
Text3.DataField = "Lgminima"  
Text4.DataField = "Lgmaxima"  
Text5.DataField = "descripción"  
Text6.DataField = "Altitud"
```

'cargando el mapa correspondiente

```
Picture2.Picture = LoadPicture(BMP)
```

'desactiva el formulario principal

```
FrmPrincipal.Enabled = False
```

'configuración del formulario de mantenimiento

```
FrmMantenimientoG.ScaleMode = vbPixels  
Picture1.ScaleMode = vbPixels  
Picture2.AutoSize = True
```

'la escala

```
Picture2.Scale (Val(lonmin), Val(latmax))-(Val(lonmax), Val(latmin))
```

'rutina que actualiza el formulario

```
Picture1.BorderStyle = 0  
Picture2.BorderStyle = 0  
Operadores
```

```
Actualizar_formulario
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

'activa el formulario principal

Bfoco = True

FrmPrincipal.Enabled = True

End Sub

Private Sub OptCaminos_Click()

Data1.RecordSource = "Caminos"

Data1.Refresh

End Sub

Private Sub OptLugares_Click()

Data1.RecordSource = "Lugares"

Data1.Refresh

End Sub

Private Sub Picture2_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)

Dim lat As Double

Dim lon As Double

lat = Y

lon = X

buscar = "Ltminima" & OperLt1 & lat & "and Ltmaxima" & OperLt2 & lat & "and Lgminima" & OperLg1 & lon & "and Lgmaxima" & OperLg2 & lon

With Data1.Recordset

.FindFirst buscar

If .NoMatch Then

FrmPrincipal.DirectSS1.Speak "Lugar no registrado"

Else

FrmPrincipal.DirectSS1.Speak !descripción

End If

End With

End Sub

Private Sub Picture2_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

TxtLatitud.Text = Y

TxtLongitud.Text = X

If MoverX Then

If MoverY Then

VScroll1.Value = (my * Y + Consty)

Picture2.Top = -VScroll1.Value

End If

HScroll1.Value = (mx * X + Constx)

Picture2.Left = -HScroll1.Value

Else

```
If MoverY Then
  VScroll1.Value = (my * Y + Consty)
  Picture2.Top = -VScroll1.Value

End If

End If

End Sub
Sub Actualizar_formulario()
Picture1.Move 0, 0, ScaleWidth - VScroll1.Width, ScaleHeight - HScroll1.Height -
Frame1.Height - 8 - Frame4.Height
Picture2.Move 0, 0

HScroll1.Top = Picture1.Height
HScroll1.Left = 0
HScroll1.Width = Picture1.Width
VScroll1.Top = 0
VScroll1.Left = Picture1.Width
VScroll1.Height = Picture1.Height

HScroll1.Max = Picture2.Width - Picture1.Width
VScroll1.Max = Picture2.Height - Picture1.Height

VScroll1.Visible = (Picture1.Height < Picture2.Height)
HScroll1.Visible = (Picture1.Width < Picture2.Width)

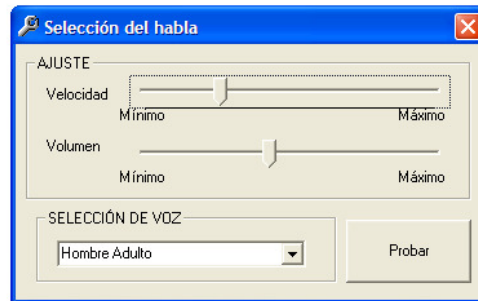
MoverY = (Picture1.Height < Picture2.Height)
MoverX = (Picture1.Width < Picture2.Width)

mx = (HScroll1.Max - 0) / (Val(lonmax) - Val(lonmin))
Constx = -Val(lonmin) * mx

my = (0 - VScroll1.Max) / (Val(latmax) - Val(latmin))
Consty = VScroll1.Max - my * Val(latmin)

End Sub
Private Sub HScroll1_Change()
Picture2.Left = -HScroll1.Value
End Sub
Private Sub VScroll1_Change()
Picture2.Top = -VScroll1.Value
End Sub
```

Este es la parte que muestra el form de velocidad



codigo utilizado

```
Private Sub Combo1_Click()  
Voz = CStr(Combo1.ListIndex + 1)  
End Sub
```

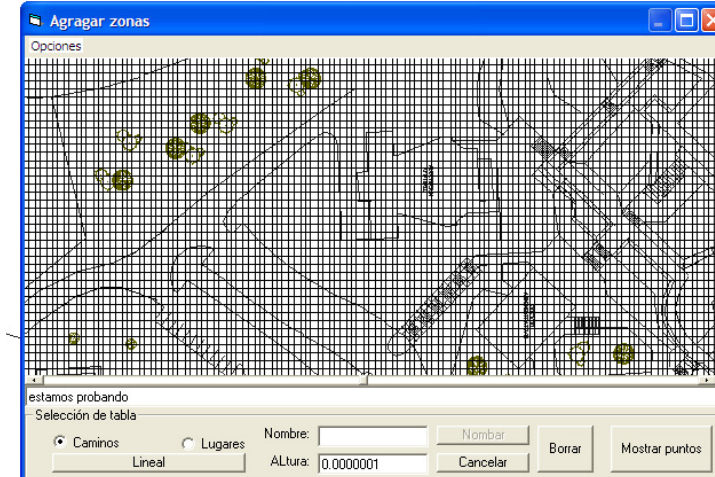
```
Private Sub Command1_Click()  
FrmPrincipal.DirectSS1.CurrentMode = Val(Voz)  
FrmPrincipal.DirectSS1.Speed = Val(Velocidad)  
FrmPrincipal.DirectSS1.Speak Usuario & "esta es una prueba"  
End Sub
```

```
Private Sub Form_Activate()  
FrmPrincipal.Enabled = False  
Combo1.Text = Combo1.List(Val(Voz) - 1)  
Slider1.Value = Val(Velocidad)  
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)  
Bfoco = True  
FrmPrincipal.Enabled = True  
FrmPrincipal.DirectSS1.CurrentMode = Val(Voz)  
FrmPrincipal.DirectSS1.Speed = Val(Velocidad)  
End Sub
```

```
Private Sub Slider1_Click()  
Velocidad = CStr(Slider1.Value)  
End Sub
```

Esta parte es del la que se utiliza para crear la base de datos



codigo utilizado

```
Dim bMoverX As Boolean
Dim bMoverY As Boolean
Dim bmy As Double
Dim bmx As Double
Dim bConstx As Double
Dim bConsty As Double
Dim bx1 As Double
Dim bx2 As Double
Dim by1 As Double
Dim by2 As Double
Dim blinea As Boolean
Dim bLatitud As Double
Dim bLongitud As Double
```

```
Private Sub CmdBorrar_Click()
Data1.Recordset.Delete
End Sub
```

```
Private Sub CmdCancelar_Click()
bN = 0
CmdNombrar.Caption = "Nombrar"
ReDim Preserve bPuntos(1, bN)
End Sub
```

```
Private Sub CmdGraficar_Click()
Text1.SetFocus
graficar
End Sub
```

```
Private Sub CmdNombrar_Click()
```

```
If CmdNombrar.Caption = "Aceptar" Then
If Command1.Caption = "Lineal" Then
    bNombre = Text2.Text
    baltura = Text3.Text
    NombrarLineal
    bN = 0
Else
    bNombre = Text2.Text
    baltura = Text3.Text
    NombrarArea
    bN = 0
End If
CmdNombrar.Caption = "Nombrar"
Else
CmdNombrar.Caption = "Aceptar"
End If
End Sub
```

```
Private Sub Command1_Click()
If Command1.Caption = "Lineal" Then
    Command1.Caption = "Area"
Else
Command1.Caption = "Lineal"
End If
End Sub
```

```
Private Sub Form_Activate()
FrmPrincipal.Enabled = False
End Sub
```

```
Sub Seleccionar_Zona()
Dim Escala As String
Dim algo As String
seleccionarbase:
DlgBase.ShowOpen
algo = DlgBase.FileName
If algo <> "" Then
bBase = CurDir + "\Sitioc.mdb"
bBMP = CurDir + "\Sitio.bmp"
Escala = CurDir + "\Sitio.txt"
Open Escala For Input As #1
Input #1, blonmin, blatmin, blonmax, blatmax, bDlat, bDlon
Close #1
Else
If bBase = "" Then
GoTo seleccionarbase:
End If
```


End Sub

```
Private Sub mnuopcionescrear_Click()  
Form2.Show  
End Sub
```

```
Private Sub mnuopcionesseleccionar_Click()  
Seleccionar_Zona  
Data1.DatabaseName = bBase  
Data1.Refresh  
Picture2.Picture = LoadPicture(bBMP)  
Form1.ScaleMode = vbPixels  
Picture1.ScaleMode = vbPixels  
Picture2.AutoSize = True  
' escala  
Picture2.Scale (Val(blonmin), Val(blatmax))-(Val(blonmax), Val(blatmin))  
operadores  
'llamada a rutina que actualiza el formulario  
Picture1.BorderStyle = 0  
Picture2.BorderStyle = 0  
Actualizar_formulario
```

End Sub

```
Private Sub Option1_Click()  
Data1.RecordSource = "Caminos"  
Data1.Refresh  
FrmPrincipal.Data1.RecordSource = "Caminos"  
FrmPrincipal.Data1.Refresh  
End Sub
```

```
Private Sub Option2_Click()  
Data1.RecordSource = "Lugares"  
Data1.Refresh  
FrmPrincipal.Data1.RecordSource = "Lugares"  
FrmPrincipal.Data1.Refresh  
End Sub
```

```
Private Sub Picture2_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As  
Single)
```

'por el momento esto sirve para saber la posición

```
bLatitud = Y  
bLongitud = X  
If bMoverX Then  
    If bMoverY Then  
        VScroll1.Value = (bmy * bLatitud + bConsty)
```

```
    Picture2.Top = -VScroll1.Value
End If
HScroll1.Value = (bmx * bLongitud + bConstx)
Picture2.Left = -HScroll1.Value
Else
If bMoverY Then
    VScroll1.Value = (bmy * bLatitud + bConsty)
    Picture2.Top = -VScroll1.Value
End If

End If

End Sub
Private Sub Picture2_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
Dim lat As Double
Dim lon As Double
Dim Yp As Double
Dim Xp As Double

If Button = 1 Then
lat = Y
lon = X

CmdNombrar.Enabled = True
buscar = "Ltminima" & bOperLt1 & lat & "and Ltmaxima" & bOperLt2 & lat & "and Lgminima" & bOperLg1 & lon & "and Lgmaxima" & bOperLg2 & lon
With Data1.Recordset
    .FindFirst buscar
    If .NoMatch Then

Else
If CmdNombrar.Caption = "Aceptar" Then
Picture2.AutoRedraw = True
    Yp = (!Ltminima + !Ltmaxima) / 2
    Xp = (!Lgminima + !Lgmaxima) / 2
    If bN > 0 Then
If Xp <> bPuntos(0, bN - 1) Or Yp <> bPuntos(1, bN - 1) Then
ReDim Preserve bPuntos(1, bN)
bPuntos(0, bN) = Xp
bPuntos(1, bN) = Yp
bN = bN + 1
Picture2.DrawWidth = 4
Form1.Picture2.PSet (Xp, Yp), color1
Else
```

```
Picture2.DrawWidth = 4
Form1.Picture2.PSet (Xp, Yp), vbWhite
bN = bN - 1
ReDim Preserve bPuntos(1, bN)
End If
Else
ReDim Preserve bPuntos(1, bN)
bPuntos(0, bN) = Xp
bPuntos(1, bN) = Yp
bN = bN + 1
Picture2.DrawWidth = 4
Form1.Picture2.PSet (Xp, Yp), vbBlue
End If
Picture2.AutoRedraw = False
End If
End If
End With

Else
DlgBase.ShowColor
color1 = DlgBase.Color
End If
Text1.SetFocus

End Sub
Private Sub Form_Resize()
Picture1.Height = Form1.Height
Picture1.Width = Form1.Width
Actualizar_formulario
End Sub
Sub Cargar()
bBase = GetSetting("Lazarillo2", "Crear", "bBase", "")
bBMP = GetSetting("Lazarillo2", "Crear", "bBMP", "")
blonmin = GetSetting("Lazarillo2", "Crear", "Longitudmin", "")
blatmin = GetSetting("Lazarillo2", "Crear", "Latitudmin", "")
blonmax = GetSetting("Lazarillo2", "Crear", "Longitudmax", "")
blatmax = GetSetting("Lazarillo2", "Crear", "Latitudmax", "")
bDlat = GetSetting("Lazarillo2", "Variables", "bDlat", "")
bDlon = GetSetting("Lazarillo2", "Variables", "bDlon", "")
End Sub
Sub Guardar()
SaveSetting "Lazarillo2", "Crear", "bBase", bBase
SaveSetting "Lazarillo2", "Crear", "bBMP", bBMP
SaveSetting "Lazarillo2", "Crear", "Longitudmin", blonmin
SaveSetting "Lazarillo2", "Crear", "Latitudmin", blatmin
```

```
SaveSetting "Lazarillo2", "Crear", "Longitudmax", blonmax
SaveSetting "Lazarillo2", "Crear", "Latitudmax", blatmax
SaveSetting "Lazarillo2", "Variables", "bDlat", bDlat
SaveSetting "Lazarillo2", "Variables", "bDlon", bDlon
End Sub
Sub Actualizar_formulario()
Picture1.Move 0, 0, ScaleWidth - VScroll1.Width, ScaleHeight - HScroll1.Height -
Text1.Height - Frame1.Height
Picture2.Move 0, 0

HScroll1.Top = Picture1.Height
HScroll1.Left = 0
HScroll1.Width = Picture1.Width

Text1.Top = Picture1.Height + HScroll1.Height
Text1.Left = 0
Text1.Width = Picture1.Width + VScroll1.Width

Frame1.Top = Picture1.Height + HScroll1.Height + Text1.Height
Frame1.Left = 0
Frame1.Width = Picture1.Width + VScroll1.Width + Text1.Width

VScroll1.Top = 0
VScroll1.Left = Picture1.Width
VScroll1.Height = Picture1.Height

HScroll1.Max = Picture2.Width - Picture1.Width
VScroll1.Max = Picture2.Height - Picture1.Height

VScroll1.Visible = (Picture1.Height < Picture2.Height)
HScroll1.Visible = (Picture1.Width < Picture2.Width)

bMoverY = (Picture1.Height < Picture2.Height)
bMoverX = (Picture1.Width < Picture2.Width)

bmx = (HScroll1.Max - 0) / (Val(blomax) - Val(blomin))
bConstx = -Val(blomin) * bmx

bmy = (0 - VScroll1.Max) / (Val(blomax) - Val(blomin))
bConsty = VScroll1.Max - bmy * Val(blomin)

End Sub
Sub operadores()
If Val(blamin) < Val(blomax) Then
bOperLt1 = "<="
bOperLt2 = ">="
by1 = Val(blamin)
```

```
by2 = Val(blattmax)
bOpy = "+"
Else
bOperLt1 = ">="
bOperLt2 = "<="
by1 = Val(blattmax)
by2 = Val(blattmin)
bOpy = "-"
End If
If Val(blonmin) < Val(blonmax) Then
bOperLg1 = "<="
bOperLg2 = ">="
bx1 = Val(blonmin)
bx2 = Val(blonmax)
bOpx = "+"
Else
bOperLg1 = ">="
bOperLg2 = "<="
bx1 = Val(blonmax)
bx2 = Val(blonmin)
bOpx = "+"
End If
End Sub
```

```
Private Sub Picture2_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
blinea = False
End Sub
```

```
Private Sub Picture2_Paint()
cuadrangular
End Sub
```

```
Sub cuadrangular()
```

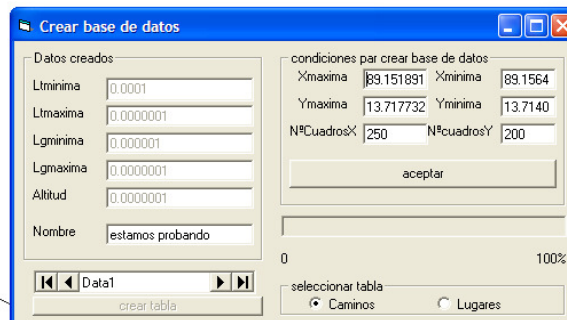
```
Picture2.DrawWidth = 1
For bi = bx1 To bx2 Step bDlonX
Picture2.Line (bi, by1)-(bi, by2)
Next
For bi = by1 To by2 Step bDlatY
Picture2.Line (bx1, bi)-(bx2, bi)
Next
```

```
End Sub
Sub graficar()
Dim JK As Double
Picture2.AutoRedraw = True
```

```
With Data1.Recordset
Picture2.DrawWidth = 5
Do
.MoveNext
If .EOF Then
Exit Do
End If
If !descripción <> "" Then
Yp = (!Ltminima + !Ltmaxima) / 2
Xp = (!Lgminima + !Lgmaxima) / 2

Form1.Picture2.PSet (Xp, Yp), color1
End If
Loop
End With
Picture2.AutoRedraw = False
Picture2.DrawWidth = 1
End Sub
```

Crea los valores de la base de datos



```
Dim EscalaY As Double
Dim EscalaX As Double
Dim DeltaX As Double
Dim DeltaY As Double
Dim CuadrosX As Double
Dim CuadrosY As Double
Dim ValorY1 As Double
Dim ValorY2 As Double
Dim ValorX As Double
Dim Nom As String
Dim base1 As String
Private Sub Command1_Click()
Dim bi As Integer
Dim bj As Integer
Dim contador As String
```

```
Command1.Enabled = False

contador = 0
For bj = 1 To CuadrosY
ValorY2 = ValorY1 + DeltaY
ValorX = Val(Text5.Text)
For bi = 1 To CuadrosX
Data1.Recordset.AddNew
TxtLtminima.Text = ValorY1
TxtLtmaxima.Text = ValorY2
TxtLgminima.Text = ValorX
ValorX = ValorX + DeltaX
TxtLgmaxima.Text = ValorX
Text7.Text = Nom
contador = contador + 1
If ProgressBar1.Object < ProgressBar1.Max Then
ProgressBar1.Object = contador
End If
'Txtdescripción.Text = contador
Data1.Recordset.Update
Next
ValorY1 = ValorY2
Next
Command2.Enabled = True
Label14.Caption = "basede datos creada"

End Sub

Private Sub Command2_Click()
Command1.Enabled = True
Command2.Enabled = False
ProgressBar1.Object = 0
EscalaX = Val(Text1.Text) - Val(Text5.Text)
EscalaY = Val(Text2.Text) - Val(Text6.Text)
CuadrosX = Val(Text3.Text)
CuadrosY = Val(Text4.Text)
DeltaX = EscalaX / CuadrosX
DeltaY = EscalaY / CuadrosY
ValorX = Val(Text5.Text)
ValorY1 = Val(Text6.Text)
ValorY2 = Val(Text6.Text)
Nom = Text7.Text
ProgressBar1.Min = 0
ProgressBar1.Max = Val(Text3.Text) * Val(Text4.Text)
End Sub
```

```
Private Sub Form_Activate()  
Form1.Enabled = False  
Command1.Enabled = False  
Label14.Caption = ""  
seleccionar  
Data1.DatabaseName = base1  
Data1.RecordSource = "Caminos"  
Data1.Refresh
```

```
Text1.Text = lonmax  
Text2.Text = latmax  
Text5.Text = lonmin  
Text6.Text = latmin  
Text3.Text = bDlon  
Text4.Text = bDlat  
End Sub
```

```
Sub seleccionar()  
Dim Escala As String  
Dim algo As String  
seleccionarbase:  
Cdlgcrear.ShowOpen  
algo = Cdlgcrear.FileName  
If algo <> "" Then  
base1 = CurDir + "\Sitioc.mdb"  
Else  
GoTo seleccionarbase:  
End If  
bDlatY = Abs(Val(latmax) - Val(latmin)) / Val(bDlat)  
bDlonX = Abs(Val(lonmin) - Val(lonmax)) / Val(bDlon)  
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)  
Form1.Enabled = True  
End Sub
```

```
Private Sub Option1_Click()  
Data1.RecordSource = "Caminos"  
Data1.Refresh  
End Sub
```

```
Private Sub Option2_Click()  
Data1.RecordSource = "Lugares"  
Data1.Refresh  
End Sub
```

módulos

variables globales

'variables ocupadas para almacenar las coordenadas del punto de destino

Public X2 As Double

Public Y2 As Double

'Matriz que se utiliza para almacenar los puntos de referencia definidos por la ruta

Public Ruta() As String

'guarda la distancia de los puntos de referencia

Public Puntos() As Double

'variables que almacenan las coordenadas máximas y mínimas de la escala del mapa

Public latmin As String

Public lonmin As String

Public latmax As String

Public lonmax As String

Public DLat As String

Public DLon As String

Public DlatY As Double

Public DlonX As Double

'variables utilizadas para determinar los tipos de operadores para determinar el criterio de búsqueda

Public OperLt1 As String

Public OperLt2 As String

Public OperLg1 As String

Public OperLg2 As String

Public Opx As String

Public Opy As String

Public Velocidad As String 'utilizada para el control de velocidad del habla

Public Voz As String

'variable que guarda la dirección donde se encuentra

'la base de datos en cuestión

Public Base As String

'variable que guarda la dirección donde se encuentra

'el mapa de bit en cuestión

Public BMP As String

*'bandera que se utiliza para indicar que se tiene que
'desplazar*

Public Bdesplazar As Boolean

*'bandera que se utiliza para indicar que
'desea saber la dirección*

Public Bsaberdirec As Boolean

Public Bguiado As Boolean

Public Bguiado1 As Boolean

'indica si esta en modo de guiado

Public Bx As Boolean

Public By As Boolean

Public Bcorregir As Boolean

Public BSalto As Boolean

Public BSisalto As Boolean

Public BSicorregir As Boolean

Public Bfoco As Boolean *'se utiliza para establecer el foco
'en el cuadro de texto del form principal*

'variable utilizada para establecer el criterio

Public buscar As String

*'variable que es utilizada para almacenamiento temporal
'del lugar actual*

'Public sitio As String

'variable que es utilizada para la hora del guiado

Public l As Integer

*'variables que se utilizan para almacenar el valor obtenido
'por el GPS.*

'por ahora se leen las coordenadas x, y de la posición del puntero

Public Latitud As Double

Public Longitud As Double

Public Altura As Double

'Public Altura As Double

Public Alturanterior As Double

*'bandera utilizada para indicar si el comando fue destino
'para luego saber que el siguiente reconocimiento
'de voz es el lugar de destino*

Public Bdestino As Boolean

Public Ldestino As String

'variable que almacena el nombre del usuario

Public Usuario As String

'se almacena el numero máximo de puntos de referencia por

'donde el usuario debe pasar

Public contador As Integer
Public Cuadrante As Integer

'variables para crear base de datos

Public bOperLt1 As String
Public bOperLt2 As String
Public bOperLg1 As String
Public bOperLg2 As String
Public blatmin As String
Public blonmin As String
Public blatmax As String
Public blonmax As String
Public bBase As String
Public bBMP As String
Public bDlat As String
Public bDlon As String
Public bPuntos() As Double
Public bN As Integer
Public bNombre As String
Public baltura As String
Public bOpx As String
Public bOpy As String
Public color1
Public bDlatY As Double
Public bDlonX As Double

modulo de configuración de puerto y GPS

con esta rutina se configura la comunicación con el gps

Sub Configuracion_GPS()

Dim Abierto As Boolean

Dim S

Dim i As Integer

Abierto = FrmPrincipal.MSComm1.PortOpen

If Abierto Then

FrmPrincipal.MSComm1.PortOpen = False

FrmPrincipal.MSComm1.CommPort = 1

FrmPrincipal.MSComm1.Settings = "4800,n,8,1"

FrmPrincipal.MSComm1.Handshaking = comXOnXoff

FrmPrincipal.MSComm1.PortOpen = True

Else

FrmPrincipal.MSComm1.CommPort = 1

FrmPrincipal.MSComm1.Settings = "4800,n,8,1"

FrmPrincipal.MSComm1.Handshaking = comXOnXoff

FrmPrincipal.MSComm1.PortOpen = True

End If

```
S = "$PMGNCMD,VERSION*28"  
FrmPrincipal.MSComm1.Output = S  
For i = 0 To 1000  
Next  
S = FrmPrincipal.MSComm1.Input  
S = "$PMGNCMD,HANDON*70"  
FrmPrincipal.MSComm1.Output = S  
For i = 0 To 1000  
Next  
S = "$PMGNCMD,TON*27"  
FrmPrincipal.MSComm1.Output = S  
For i = 0 To 1000  
Next  
S = FrmPrincipal.MSComm1.Input
```

End Sub

Modulo para crear base

```
Dim bi As Integer  
Dim bj As Integer  
Dim bk As Double  
Dim bl As Integer  
Dim bl1 As Integer  
Dim bXa As Double  
Dim bya As Double  
Dim bXc1 As Double  
Dim bXc2 As Double  
Dim byc As Double  
Dim Id As Integer  
Dim Repetidos() As Double  
Dim Brepetido As Boolean  
Sub NombrarArea()  
  
For bj = 0 To bN - 1  
  For bi = bj + 1 To bN - 1  
    If bPuntos(1, bj) = bPuntos(1, bi) Then  
      byc = bPuntos(1, bj)  
      If bPuntos(0, bj) < bPuntos(0, bi) Then  
        bXc1 = bPuntos(0, bj)  
        bXc2 = bPuntos(0, bi)  
      Else  
        bXc1 = bPuntos(0, bi)  
        bXc2 = bPuntos(0, bj)  
      End If  
    End If  
  End For  
End For
```

```
        ReDim Preserve Repetidos(1, bl)
        Repetidos(0, bl) = bPuntos(0, bi)
        Repetidos(1, bl) = byc
        bl = bl + 1
    Call Asignar_Nombre
    Exit For
Else
    If bi >= bN + 1 Then
        byc = bPuntos(1, bj)
        bXc1 = bPuntos(0, bj)
        bXc2 = bXc1
        Call Asignar_Nombre
        Exit For
    End If
End If
Next bi

Next bj

End Sub
Sub Asignar_Nombre()
Form1.Picture2.AutoRedraw = True
For bk = bXc1 To bXc2 Step bDlonX
    buscar = "Ltminima" & bOperLt1 & byc & "and Ltmaxima" & bOperLt2 & byc & "and
Lgminima" & bOperLg1 & bk & "and Lgmaxima" & bOperLg2 & bk
With Form1.Data1.Recordset
    .FindFirst buscar
    If .NoMatch Then
'oooo
    Else
        Form1.Text1.Text = bNombre
        Form1.Text3.Text = baltura
        Form1.Picture2.DrawWidth = 5

        FrmPrincipal.Data1.Recordset.AddNew
        FrmPrincipal.Data1.Recordset.Fields("Ltminima") = !Ltminima
        FrmPrincipal.Data1.Recordset.Fields("Lgminima") = !Lgminima
        FrmPrincipal.Data1.Recordset.Fields("Ltmaxima") = !Ltmaxima
        FrmPrincipal.Data1.Recordset.Fields("Lgmaxima") = !Lgmaxima
        FrmPrincipal.Data1.Recordset.Fields("Descripción") = bNombre
        FrmPrincipal.Data1.Recordset.Fields("Altitud") = baltura
        'FrmPrincipal.Data1.Recordset.Fields("Id") = !Id
        FrmPrincipal.Data1.Recordset.Update
```

```
FrmPrincipal.Data1.Refresh
```

```
Yp = (!Ltminima + !Ltmaxima) / 2  
Xp = (!Lgminima + !Lgmaxima) / 2  
Form1.Picture2.PSet (Xp, Yp), color1  
Form1.Picture2.DrawWidth = 1  
End If  
End With  
Next bk
```

```
Form1.Picture2.AutoRedraw = False  
End Sub
```

```
Sub NombrarLineal()  
Form1.Picture2.AutoRedraw = True  
With Form1.Data1.Recordset  
For bi = 0 To bN - 1 Step 1  
bXa = bPuntos(0, bi)  
bya = bPuntos(1, bi)  
buscar = "Ltminima" & bOperLt1 & bya & "and Ltmaxima" & bOperLt2 & bya & "and  
Lgminima" & bOperLg1 & bXa & "and Lgmaxima" & bOperLg2 & bXa
```

```
.FindFirst buscar  
If .NoMatch Then  
'oooo  
Else  
Form1.Text1.Text = bNombre  
Form1.Text3.Text = baltura  
Form1.Picture2.DrawWidth = 5
```

```
FrmPrincipal.Data1.Recordset.AddNew  
FrmPrincipal.Data1.Recordset.Fields("Ltminima") = !Ltminima  
FrmPrincipal.Data1.Recordset.Fields("Lgminima") = !Lgminima  
FrmPrincipal.Data1.Recordset.Fields("Ltmaxima") = !Ltmaxima  
FrmPrincipal.Data1.Recordset.Fields("Lgmaxima") = !Lgmaxima  
FrmPrincipal.Data1.Recordset.Fields("Descripción") = bNombre  
FrmPrincipal.Data1.Recordset.Fields("Altitud") = baltura  
FrmPrincipal.Data1.Recordset.Fields("Id") = !Id  
FrmPrincipal.Data1.Recordset.Update  
FrmPrincipal.Data1.Refresh
```

```
Yp = (!Ltminima + !Ltmaxima) / 2  
Xp = (!Lgminima + !Lgmaxima) / 2
```

```
Form1.Picture2.PSet (Xp, Yp), color1
Form1.Picture2.DrawWidth = 1
End If
```

```
Next
End With
Form1.Picture2.AutoRedraw = False
End Sub
```

Modulo de guiado

```
Dim indicador As Integer
Dim Lanterior As Integer
Dim Indicadorstree As Integer
Dim Xp As Double
Dim Yp As Double
Dim Bcruzar As Boolean
Sub Guiado(M As Integer, Avanzar As String)
'comparación si se va moviendo en dirección paralelo
'a latitudes o longitudes
If M > 0 And M < contador Then
'determina el cuadrante en el que se encuentra
Select Case Cuadrante
    Case Is = 1
    GoTo CUADRANTE1:
    Case Is = 2
    GoTo CUADRANTE2:
    Case Is = 3
    GoTo CUADRANTE3:
    Case Is = 4
    GoTo CUADRANTE4:
    Case Else
```

```
End Select
GoTo FIN:
```

CUADRANTE1:

```
'comparación para determinar si se mueve en longitud
'constante
If Puntos(0, M) = Puntos(0, M - 1) Then
    'determina si se mueve hacia la izquierda, derecha
    'o hacia adelante
    If Puntos(1, M) > Puntos(1, M - 1) Then
        If Puntos(0, M) < Puntos(0, M + 1) And _
            Puntos(1, M) = Puntos(1, M + 1) Then
            'moverse a la derecha
```

```
Avanzar = "derecha"

ElseIf Puntos(0, M) > Puntos(0, M + 1) And _
Puntos(1, M) = Puntos(1, M + 1) Then
  'moverse izquierda
  Avanzar = "izquierda"

Else
  'moverse adelante
  Avanzar = "Adelante"

End If
Else
  If Puntos(0, M) < Puntos(0, M + 1) And _
  Puntos(1, M) = Puntos(1, M + 1) Then
    'moverse izquierda
    Avanzar = "izquierda"

    ElseIf Puntos(0, M) > Puntos(0, M + 1) And _
    Puntos(1, M) = Puntos(1, M + 1) Then
      'moverse derecha
      Avanzar = "derecha"

    Else
      'moverse adelante
      Avanzar = "Adelante"

    End If
  End If
Else
  'determina si se mueve hacia la izquierda, derecha
  'o hacia adelante
  If Puntos(0, M) > Puntos(0, M - 1) Then
    If Puntos(1, M) < Puntos(1, M + 1) And _
    Puntos(0, M) = Puntos(0, M + 1) Then
      'moverse a la izquierda.....
      Avanzar = "izquierda"
    ElseIf Puntos(1, M) > Puntos(1, M + 1) And _
    Puntos(0, M) = Puntos(0, M + 1) Then
      'moverse derecha
      Avanzar = "derecha"
    Else
      'moverse adelante
      Avanzar = "Adelante"
```

```
End If
Else
  If Puntos(1, M) < Puntos(1, M + 1) And _
    Puntos(0, M) = Puntos(0, M + 1) Then
    'moverse derecha.....
    Avanzar = "derecha"

    ElseIf Puntos(1, M) > Puntos(1, M + 1) And _
      Puntos(0, M) = Puntos(0, M + 1) Then
      'moverse izquierda
      Avanzar = "izquierda"

    Else
      'moverse adelante
      Avanzar = "adelante"

    End If

  End If

End If

End If
GoTo FIN:

CUADRANTE2:
'comparación para determinar si se mueve en longitud
'constante
If Puntos(0, M) = Puntos(0, M - 1) Then
  'determina si se mueve hacia la izquierda, derecha
  'o hacia adelante
  If Puntos(1, M) > Puntos(1, M - 1) Then
    If Puntos(0, M) > Puntos(0, M + 1) And _
      Puntos(1, M) = Puntos(1, M + 1) Then
      'moverse a la derecha
      Avanzar = "derecha"

    ElseIf Puntos(0, M) < Puntos(0, M + 1) And _
      Puntos(1, M) = Puntos(1, M + 1) Then
      'moverse izquierda
      Avanzar = "izquierda"

    Else
      'moverse adelante
      Avanzar = "Adelante"

    End If

  End If
```

```
Else
  If Puntos(0, M) > Puntos(0, M + 1) And _
  Puntos(1, M) = Puntos(1, M + 1) Then
    'moverse izquierda
    Avanzar = "izquierda"

  ElseIf Puntos(0, M) < Puntos(0, M + 1) And _
  Puntos(1, M) = Puntos(1, M + 1) Then
    'moverse derecha
    Avanzar = "derecha"

  Else
    'moverse adelante
    Avanzar = "Adelante"

  End If
End If
Else

  If Puntos(0, M) > Puntos(0, M - 1) Then
    If Puntos(1, M) < Puntos(1, M + 1) And _
    Puntos(0, M) = Puntos(0, M + 1) Then
      'moverse a la derecha
      Avanzar = "derecha"

    ElseIf Puntos(1, M) > Puntos(1, M + 1) And _
    Puntos(0, M) = Puntos(0, M + 1) Then
      'moverse izquierda
      Avanzar = "izquierda"

    Else
      'moverse adelante
      Avanzar = "Adelante"

    End If
  Else
    If Puntos(1, M) < Puntos(1, M + 1) And _
    Puntos(0, M) = Puntos(0, M + 1) Then
      'moverse izquierda
      Avanzar = "izquierda"

    ElseIf Puntos(1, M) > Puntos(1, M + 1) And _
    Puntos(0, M) = Puntos(0, M + 1) Then
      'moverse derecha
      Avanzar = "derecha"
```

```
Else
    'moverse adelante
    Avanzar = "adelante"

End If

End If

End If
GoTo FIN:
CUADRANTE3:
GoTo FIN:
CUADRANTE4:
FIN:

End If
End Sub
Sub Orientar(grados As String, lado1 As String)
Dim stree As String
Dim Xc As Double
Dim Yc As Double
Dim P As Double
Dim Q As Double
Dim R As Double
Dim Z As Double
Dim teta
stree = Lugar(Longitud, Latitud, Xc, Yc)
If stree = "" Then
    FrmPrincipal.Data1.RecordSource = "Lugares"
    FrmPrincipal.Data1.Refresh
    stree = Lugar(Longitud, Latitud, Xc, Yc)
    If stree = "" Then
        FrmPrincipal.DirectSS1.Speak Usuario & "Usted está en una zona no registrada por el
sistema"
    Else
        GoTo saltar:
    End If
Else
saltar:
    If Xc = Puntos(0, 0) And Yc = Puntos(1, 0) Then
        grados = ""
        lado1 = ""
    Else
        P = Dist(Puntos(0, 0), Puntos(1, 0), Puntos(0, 1), Puntos(1, 1))
        Q = Dist(Puntos(0, 0), Puntos(1, 0), Xc, Yc)
        R = Dist(Xc, Yc, Puntos(0, 1), Puntos(1, 1))
```

```
Z = (P * P + Q * Q - R * R) / (2 * P * Q)
If Round(Z) = 1 Then
teta = 0
Elseif Round(Z) = -1 Then
teta = 180
Else
teta = (Atn(-Z / Sqr(-Z * Z + 1)) + 2 * Atn(1)) * 180 / (4 * Atn(1))
End If
Select Case Val(teta)
Case Is = 0
    grados = ""
    lado1 = "adelante"
    l = l + 1
    GoTo FIN1:
Case 45, 90, 135
    grados = "135 grados"
Case Is = 180
    grados = "180 grados"
Case Else
    grados = ""
End Select
lado1 = Lado(Xc, Yc)
FIN1:
End If
End If
FrmPrincipal.Data1.RecordSource = "Caminos"
FrmPrincipal.Data1.Refresh
End Sub
Function Lado(Xc As Double, Yc As Double) As String
Select Case Cuadrante
Case Is = 1
GoTo CUADRANTE22:
Case Is = 2
GoTo CUADRANTE11:
Case Else
GoTo FIN2:
End Select
CUADRANTE11:

If Xc = Puntos(0, 0) Then
If Yc > Puntos(1, 0) Then
If Xc > Puntos(0, 1) Then
Lado = "derecha"
Else
Lado = "izquierda"
End If
Else
```

```
    If Xc > Puntos(0, 1) Then
        Lado = "izquierda"
    Else
        Lado = "derecha"
    End If
End If
```

```
Else
    If Xc > Puntos(0, 0) Then
        If Yc > Puntos(1, 1) Then
            Lado = "Izquierda"
        Else
            Lado = "Derecha"
        End If
    Else
        If Yc > Puntos(1, 1) Then
            Lado = "derecha"
        Else
            Lado = "izquierda"
        End If
    End If
End If
```

```
GoTo FIN2:
CUADRANTE22:
```

```
If Xc = Puntos(0, 0) Then
    If Yc > Puntos(1, 0) Then
        If Xc > Puntos(0, 1) Then
            Lado = "izquierda"
        Else
            Lado = "derecha"
        End If
    Else
        If Xc > Puntos(0, 1) Then
            Lado = "derecha"
        Else
            Lado = "izquierda" 'ssss
        End If
    End If
```

```
Else
    If Xc > Puntos(0, 0) Then
        If Yc > Puntos(1, 1) Then
            Lado = "derecha"
        Else
            Lado = "izquierda"
        End If
```

```
Else
  If Yc > Puntos(1, 1) Then
    Lado = "izquierda"
  Else
    Lado = "derecha"
  End If
End If
End If

FIN2:
End Function
Sub indicar()
Dim stree As String
Dim stree1 As String
Dim i As Integer
'contador trae el valor del numero de lugares de referencia por
'las cuales se tiene que pasar

If l <= contador Then
  'funcion que devuelve el nombre del lugar de
'referencia correspondiente a latitud y longitud
'puntos medio y altura
  stree = Lugar(Longitud, Latitud, Xp, Yp, Altura)
  'si no encuentra cambia de tabla
  If stree = "" Then
    FrmPrincipal.Data1.RecordSource = "Lugares"
    FrmPrincipal.Data1.Refresh
    stree = Lugar(Longitud, Latitud, Xp, Yp, Altura)
  End If
  If l > 0 Then
    Call Cruzar(l)
  End If
  'comparación si esta en el rango o alrededor del lugar
  'de referencia
  If ((Xp = Puntos(0, l) Or Bx) And (Yp = Puntos(1, l) Or By)) And UCase(Ruta(l)) =
  UCase(stree) Then

    If l > 0 Then
      'comparación con el objetivo de no repetir constante
      'el lugar por donde se encuentra
      If Ruta(l) <> Ruta(l - 1) Or indicador = 6 Then
        'determina si hay que subir o bajar gradas
        If Alturanterior < Altura Then
          FrmPrincipal.DirectSS1.Speak "subir" & Ruta(l)
          Alturanterior = Altura
        ElseIf Alturanterior > Altura Then
```

```

    FrmPrincipal.DirectSS1.Speak "bajar" & Ruta(l)
    Alturanterior = Altura
Else
    FrmPrincipal.DirectSS1.Speak "usted está en " & Ruta(l)
    Alturanterior = Altura
End If

    indicador = 0
Else
    'controla las veces de repetición del lugar
    'de referencia por donde se encuentra
    indicador = indicador + 1
End If

End If

'bloque utilizado para realizar la corrección del lugar
'de referencia por donde se debe pasar
Call Guiado(l, stree)
Call Guiado(l + 1, stree1)
If (UCase(stree) = "IZQUIERDA" Or UCase(stree) = "DERECHA") _
And (UCase(stree1) = "IZQUIERDA" Or UCase(stree1) = "DERECHA") Then
    l = l + 2
    If Indicadorstree = 5 Then
    FrmPrincipal.DirectSS1.Speak "desplazarse hacia adelante"
    Indicadorstree = 0
    End If
    Indicadorstree = Indicadorstree + 1

ElseIf (UCase(stree) = "IZQUIERDA" Or UCase(stree) = "DERECHA") _
And (UCase(stree1) <> "IZQUIERDA" Or UCase(stree1) <> "DERECHA") Then
    If l <> Lanterior Then
    FrmPrincipal.DirectSS1.Speak "desplazarse hacia" & stree
    End If
    Lanterior = l
    Call Corregir(l)
Else
    If Indicadorstree = 5 Then
    FrmPrincipal.DirectSS1.Speak "desplazarse hacia" & stree
    Indicadorstree = 0
    End If
    l = l + 1
    Indicadorstree = Indicadorstree + 1
End If

FrmPrincipal.TxtComandos.SetFocus

```

```
Else
  If l > 1 Then
    'comparación que determina si se encuentra en el lugar
    'de referencia anterior o si se a salido de la ruta
    If (((Xp = Puntos(0, l - 1) Or Bx) And (Yp = Puntos(1, l - 1) Or By)) And
    (UCase(Ruta(l - 1)) = UCase(stree))) Then
      'oooooooo
      ElseIf (((Xp = Puntos(0, l - 2) Or Bx) And (Yp = Puntos(1, l - 2) Or By)) And
      (UCase(Ruta(l - 2)) = UCase(stree))) Then
        'debido a que si el GPS da una lectura de atrás
        l = l - 1
        ElseIf (((Xp = Puntos(0, l + 1) Or Bx) And (Yp = Puntos(1, l + 1) Or By)) And
        (UCase(Ruta(l + 1)) = UCase(stree))) Then
          'debido a que si el GPS da una lectura adelante
          l = l + 2
        Else
          FrmPrincipal.DirectSS1.Speak Usuario & "se dirige en una dirección no
          establecida"
          FrmPrincipal.DirectSS1.Speak "favor repita el lugar de destino"
          Bguiado = False
          FrmPrincipal.Picture2.Refresh
          l = 0
          X2 = 0
          Y2 = 0
          FrmPrincipal.TxtComandos.SetFocus
          FrmPrincipal.Timer1.Enabled = False
        End If
      End If
    End If
  Else
    Bguiado = False
    FrmPrincipal.Picture2.Refresh
    FrmPrincipal.TxtComandos.Text = ""
    l = 0
    FrmPrincipal.Timer1.Enabled = False
    FrmPrincipal.TxtComandos.SetFocus
  End If
  FrmPrincipal.Data1.RecordSource = "Caminos"
  FrmPrincipal.Data1.Refresh
End Sub
Sub Cruzar(M As Integer)
If Puntos(0, M) = Puntos(0, M - 1) Then
  By = False
  Bx = True
Else
```

```
By = True
Bx = False
End If
End Sub
Sub Corregir(M As Integer)
Dim i As Integer
Dim Encontro As Boolean
Encontro = False
For i = 0 To contador
    If Xp = Puntos(0, i) And Yp = Puntos(1, i) Then
        Encontro = True
        Exit For
    End If
Next i
If Encontro Then
    M = i + 1
Else
    M = M
End If
End Sub
```

Modulo de comandos

```
Sub donde_estoy()
Dim lugar1 As String
Dim Xp As Double
Dim Yp As Double
'en este bloque se convoca la información del GPS
'para luego compararla con la información de la base de datos y devolver el mensaje
en forma de habla
'-----
'establecer el criterio de búsqueda
'los valores de latitud y longitud tienen que traer
'los leídos por el GPS

'lee los valores de latitud y longitud del archivo ya creado
Call Get_GPS(Longitud, Latitud)

If Longitud <> 0 And Latitud <> 0 Then

lugar1 = Lugar(Longitud, Latitud, Xp, Yp)
If lugar1 <> "" Then
    FrmPrincipal.DirectSS1.Speak Usuario & "Usted está en" & lugar1
    Call FrmPrincipal.Mover_Pantalla
```

```
FrmPrincipal.Picture2.PSet (Xp, Yp), vbBlack

Else
  FrmPrincipal.Data1.RecordSource = "Lugares"
  FrmPrincipal.Data1.Refresh
  lugar1 = Lugar(Longitud, Latitud, Xp, Yp)
  If lugar1 <> "" Then
    FrmPrincipal.DirectSS1.Speak Usuario & "Usted está en" & lugar1
    Call FrmPrincipal.Mover_Pantalla
    FrmPrincipal.Picture2.PSet (Xp, Yp), vbBlack
  Else
    FrmPrincipal.DirectSS1.Speak Usuario & "Usted está en una zona no registrada por el
sistema"
  End If
  FrmPrincipal.Data1.RecordSource = "Caminos"
  FrmPrincipal.Data1.Refresh
End If
Else
  FrmPrincipal.DirectSS1.Speak "petición fallida"
End If
End Sub
Sub Alto()
'deshabilita el procedimiento de guiado
  Bguiado = False
  FrmPrincipal.Timer1.Enabled = False
  FrmPrincipal.Picture2.Refresh
  Call donde_estoy
  l = 0
  X2 = 0
  Y2 = 0
End Sub
Sub Destino()
  Dim sitio As String
  Dim Xi As Double
  Dim Yi As Double

  FrmPrincipal.TxtComandos.Text = ""
  sitio = Buscar_Posición(Xi, Yi, Alturanterior)
  If sitio <> "" Then
    Call Coordinadas(Ldestino, Xi, Yi, X2, Y2)
    If Xi <> X2 Or Yi <> Y2 Then
      If X2 <> 0 And Y2 <> 0 Then
        TrazarRuta Xi, Yi, X2, Y2
        Bguiado1 = True
        FrmPrincipal.DirectSS1.Speak Usuario & "favor desplazarse hacia adelante"
        FrmPrincipal.Timer1.Enabled = True
      Else

```

```
        FrmPrincipal.DirectSS1.Speak "Lugar destino no encontrado"  
    End If  
    Else  
        FrmPrincipal.DirectSS1.Speak Usuario & "usted esta en" & sitio  
    End If  
    Else  
        FrmPrincipal.DirectSS1.Speak "favor ubicarse en otro sitio"  
    End If  
    X2 = 0  
    Y2 = 0  
    l = 0  
End Sub
```

```
Sub operadores()  
Dim g1 As Integer  
Dim g2 As Integer  
Dim G As Integer  
If Val(latmin) < Val(latmax) Then  
    OperLt1 = "<="   
    OperLt2 = ">="   
    y1min = Val(latmin)  
    y2max = Val(latmax)  
    Opy = "+"  
    g1 = 2  
Else  
    OperLt1 = ">="   
    OperLt2 = "<="   
    y1min = Val(latmax)  
    y2max = Val(latmin)  
    Opy = "-"  
    g1 = 4  
End If  
If Val(lonmin) < Val(lonmax) Then  
    OperLg1 = "<="   
    OperLg2 = ">="   
    x1min = Val(lonmin)  
    x2max = Val(lonmax)  
    Opx = "+"  
    g2 = 1  
Else  
    OperLg1 = ">="   
    OperLg2 = "<="   
    x1min = Val(lonmax)  
    x2max = Val(lonmin)  
    Opx = "-"  
    g2 = 5  
End If
```

```
G = g1 + g2
Select Case G
Case Is = 3
    Cuadrante = 1
Case Is = 7
    Cuadrante = 2
Case Is = 9
    Cuadrante = 3
Case Is = 5
    Cuadrante = 4
Case Else
    Cuadrante = 1
End Select
End Sub
Function Buscar_Posición(Xn As Double, Yn As Double, Optional H As Double) As String
Dim sitio As String
Dim Xp As Double
Dim Yp As Double
Dim Xpn As Double
Dim Ypn As Double
Dim D As Double
Dim Da As Double
Dim pd As Boolean
pd = False
sitio = Lugar(Longitud, Latitud, Xp, Yp, H)
If sitio <> "" Then
    Xn = Xp
    Yn = Yp
    Buscar_Posición = sitio
Else
    FrmPrincipal.Data1.RecordSource = "Lugares"
    FrmPrincipal.Data1.Refresh
    sitio = Lugar(Longitud, Latitud, Xp, Yp, H)
    If sitio <> "" Then
        '-----
        FrmPrincipal.Data1.RecordSource = "Caminos"
        FrmPrincipal.Data1.Refresh
        buscar = "Descripción like" & sitio & "" _
            & "and ((Ltminima" & OperLt1 & Yp & "and Ltmaxima" & OperLt2 & Yp & ")" _
            & "or (Lgminima" & OperLg1 & Xp & "and Lgmaxima" & OperLg2 & Xp & ")")"

        With FrmPrincipal.Data1.Recordset
        Buscar_Posición = ""
        Do
        .FindNext buscar
        If .NoMatch Then
            Exit Do
        End With
    End If
End If
```

```

Else
  If pd Then
    Ypn = (!Ltminima + !Ltmaxima) / 2
    Xpn = (!Lgminima + !Lgmaxima) / 2
    D = Dist(Xp, Yp, Xpn, Ypn)
    If D < Da Then
      H = !Altitud
      Buscar_Posición = sitio
      Da = D
      Xn = Xpn
      Yn = Ypn
    End If
  Else
    Ypn = (!Ltminima + !Ltmaxima) / 2
    Xpn = (!Lgminima + !Lgmaxima) / 2
    D = Dist(Xp, Yp, Xpn, Ypn)
    Da = D
    Buscar_Posición = sitio
    H = !Altitud
    Xn = Xpn
    Yn = Ypn
    pd = True
  End If
End If
Loop
End With
'-----
Else
  FrmPrincipal.DirectSS1.Speak Usuario & "Usted está en una zona no registrada por el
sistema"
  End If
  FrmPrincipal.Data1.RecordSource = "Caminos"
  FrmPrincipal.Data1.Refresh
End If

End Function
Sub Get_GPS(LGx As Double, LTy As Double)
Dim Dato As String
Dim Laty As String
Dim Lonx As String
Dim Repetir As Integer
Repetir = 0
LA:
FrmPrincipal.MSComm1.Output = "$PMGNCMD,WAYPOINT*78" & Chr$(10) & Chr$(13)
For i = 0 To 1000
Next
Dato = FrmPrincipal.MSComm1.Input

```

```
If Mid(Dato, 1, 1) <> "$" Or Len(Dato) < 27 Then
  Repetir = Repetir + 1
  If Repetir < 30 Then
    GoTo LA:
  Else
    Repetir = 0
    Dato = ""
  End If
End If
```

```
Lonx = Mid(Dato, 19, 7)
```

```
Laty = Mid(Dato, 8, 7)
```

```
LGx = grados(Lonx)
```

```
LTy = grados(Laty)
```

```
End Sub
```

```
Function grados(G As String) As Double
```

```
Dim HG As String
```

```
Dim LG As String
```

```
'parte alta del valor
```

```
HG = Mid(G, 1, 2)
```

```
'parte baja
```

```
LG = Mid(G, 3, 5)
```

```
grados = Val(HG) + Val(LG) / 60
```

```
End Function
```

Modulo para trazar ruta

```
Dim Distancia() As Double
```

```
Dim PosAdy() As String
```

```
Dim Pmedios() As Double
```

```
Dim Ymax As Double
```

```
Dim Ymin As Double
```

```
Dim Xmax As Double
```

```
Dim Xmin As Double
```

```
Dim PXanterior As Double
```

```
Dim PYanterior As Double
```

```
Dim PYactual As Double
```

```
Dim PXactual As Double
```

```
Dim a As Integer
```

```
Dim yas As Boolean
```

```
'funcion que de vuelve en la matriz Ruta() los nombres de  
'los puntos de referencia por donde se trazo la ruta.
```

```
'X,Y,X2,Y2 son los puntos medios de partida y llegada respectivamente  
'para trazar la ruta
```

Sub TrazarRuta(X As Double, Y As Double, X2, Y2)

Dim Trazada As Boolean

Trazada = False

a = 0

*'por el momento para efectos de prueba se esta utilizando
'la variable "contador" con el motivo de llevar el conteo*

'de los lugares de referencia que se establecen en la ruta

contador = 0

'se guarda la distancia entre el punto de partida y destino

ReDim Preserve Distancia(0)

Distancia(0) = Dist(X, Y, X2, Y2)

ReDim Preserve PosAdy(0)

PosAdy(0) = Lugar(X, Y, X, Y)

ReDim Preserve Pmedios(1, 0)

Pmedios(0, 0) = X

Pmedios(1, 0) = Y

ReDim Preserve Ruta(a)

Ruta(a) = PosAdy(0)

ReDim Preserve Puntos(1, 0)

Puntos(0, 0) = X

Puntos(1, 0) = Y

PXactual = X

PYactual = Y

PXanterior = X

PYanterior = Y

yas = False

Do

'variable que lleva el control de lugares de referencia

'encontrados

a = a + 1

contador = contador + 1

ReDim Preserve Ruta(a)

Ruta(a) = Adyacente(X, Y, X, Y) *'rutina que devuelve el nombre
'del lugar adyacente y el punto*

'medio de su área

'almacena el punto medio del lugar de referencia

'encontrado

ReDim Preserve Puntos(1, a)

Puntos(0, a) = X

Puntos(1, a) = Y

If a > 2 Then

```

'compara si el lugar de referencia ya a sido encontrado
If Puntos(0, a) = Puntos(0, a - 2) And Puntos(1, a) = Puntos(1, a - 2) Then
  'borra el lugar de referencia si esta repetido
  a = a - 2
  contador = contador - 2
  ReDim Preserve Ruta(a)
  ReDim Preserve Puntos(1, a)
End If
End If

```

```

'determina si ya se llego al destino
If (X = X2 And Y = Y2) Or UCase(Ruta(a)) = UCase(Ldestino) Then
  Trazada = True
End If

```

```

Loop Until Trazada = True
Call Graficar_ruta 'grafica la ruta establecida
End Sub

```

```

'funcion que saca la distancia entre dos puntos
Function Dist(X As Double, Y As Double, X2, Y2) As Double
Dist = Sqr((X2 - X) ^ 2 + (Y2 - Y) ^ 2)
End Function

```

```

'funcion que devuelve el lugar según la coordenada y los puntos
'mínimos, máximos del rango o área
'X,Y son puntos dentro del rango
Function Lugar(X As Double, Y As Double, Xp As Double, Yp As Double, Optional H As
Double) As String
'establece el criterio de búsqueda
buscar = "Ltminima" & OperLt1 & Y & "and Ltmaxima" & OperLt2 & Y & "and Lgminima"
& OperLg1 & X & "and Lgmaxima" & OperLg2 & X
With FrmPrincipal.Data1.Recordset
  'busca en la base se datos
  .FindFirst buscar
  'determina si encontró
  If .NoMatch Then
    Lugar = ""
  Else
    'asigna los valores de los campos a las variables
    Lugar = !descripción
    Ymax = !Ltmaxima
    Ymin = !Ltminima
    Xmax = !Lgmaxima
    Xmin = !Lgminima
    H = !Altitud
    'calcula el punto medio

```

```

    Yp = (!Ltminima + !Ltmaxima) / 2
    Xp = (!Lgminima + !Lgmaxima) / 2
End If

```

```

End With
End Function

```

```

' este procedimiento busca en la base de datos las coordenadas
' correspondientes al lugar que se le indica por medio de la variable
' "Lugar"

```

```

Sub Coordinadas(Lugar As String, Xp1 As Double, Yp1 As Double, Xp As Double, Yp As
Double)

```

```

    Dim D As Double

```

```

    Dim Da As Double

```

```

    Dim pd As Boolean

```

```

    Dim Xpn As Double

```

```

    Dim Ypn As Double

```

```

    pd = False

```

```

    With FrmPrincipal.Data1.Recordset

```

```

        Xp = 0

```

```

        Yp = 0

```

```

        .FindFirst "Descripción like" & Lugar & ""

```

```

        If .NoMatch Then

```

```

            GoTo SALIR:

```

```

        Else

```

```

            Ypn = (!Ltminima + !Ltmaxima) / 2

```

```

            Xpn = (!Lgminima + !Lgmaxima) / 2

```

```

            D = Dist(Xp1, Yp1, Xpn, Ypn)

```

```

            Da = D

```

```

            Xp = Xpn

```

```

            Yp = Ypn

```

```

        End If

```

```

        'busca el lugar mas cercano

```

```

        Do

```

```

            .FindNext "Descripción like" & Lugar & ""

```

```

            If .NoMatch Then

```

```

                Exit Do

```

```

            Else

```

```

                Ypn = (!Ltminima + !Ltmaxima) / 2

```

```

                Xpn = (!Lgminima + !Lgmaxima) / 2

```

```

                D = Dist(Xp1, Yp1, Xpn, Ypn)

```

```

                If D < Da Then

```

```

                    Da = D

```

```

                    Xp = Xpn

```

```

                    Yp = Ypn

```

```

                End If

```

```

            End If

```

```
    Loop
End With
SALIR:
End Sub
'funcion que evalúa los lugares adyacentes validos al punto actual
Function Adyacente(X As Double, Y As Double, Xp As Double, Yp As Double) As String
Dim sitio As String
Dim Nady As Integer
Dim i As Integer
Dim j As Integer
Dim k As Integer
Dim j1 As Integer
Dim Xa As Double
Dim ya As Double
Dim Xb As Double
Dim Yb As Double
Dim Temp1 As Double
Dim Temp2 As String
Dim Temp3X As Double
Dim Temp3Y As Double
Dim distan As Double
sitio = Lugar(X, Y, Temp3X, Temp3Y) 'con esta funcion se conoce ala vez las
coordenadas
        'máximas y mínimas del área

If sitio = "" Then
'mandar algún mensaje de posición actual no valida
Else
j = 0
For i = 1 To 4 Step 1
'este bloque se va a utilizar para establecer el lugar adyacente

'determina los valores de evaluación del lugar de referencia
adyacente al lugar actual
Select Case i
    Case Is = 1
        Xa = X: ya = Ymin - Val(Opy & Str(DlatY)) / 2
    Case Is = 2
        Xa = X: ya = Ymax + Val(Opy & Str(DlatY)) / 2
    Case Is = 3
        Xa = Xmin - Val(Opx & Str(DlonX)) / 2: ya = Y
    Case Is = 4
        Xa = Xmax + Val(Opx & Str(DlonX)) / 2: ya = Y
    Case Else
        Dim Mensaje As Integer
```

```

Mensaje = MsgBox("error en buscar lugar adyacente", 1 + 35, _
Usuario)
End Select

```

'criterio de búsqueda

```

buscar = "Ltminima" & OperLt1 & ya & "and Ltmaxima" & OperLt2 & ya & "and
Lgminima" & OperLg1 & Xa & "and Lgmaxima" & OperLg2 & Xa

```

```

With FrmPrincipal.Data1.Recordset

```

```

.FindFirst buscar

```

```

If .NoMatch Then

```

```

'hooooooooooooooooooooo

```

```

Else

```

```

j = j + 1

```

```

'guarda el punto medio del lugar de referencia encontrada

```

```

Yb = (!Ltminima + !Ltmaxima) / 2

```

```

Xb = (!Lgminima + !Lgmaxima) / 2

```

```

ReDim Preserve Pmedios(1, j)

```

```

Pmedios(0, j) = Xb

```

```

Pmedios(1, j) = Yb

```

```

ReDim Preserve Distancia(j)

```

```

'calcula la distancia y la almacena temporalmente

```

```

Distancia(j) = Dist(Xb, Yb, X2, Y2)

```

```

ReDim Preserve PosAdy(j)

```

```

'almacena temporalmente la descripción del lugar de
referencia

```

```

PosAdy(j) = !descripción

```

```

End If

```

```

End With

```

```

'siguiente evaluación

```

```

NET

```

*'bloque utilizado para ordenar la matriz de lugares adyacente
en orden ascendente*

```

j1 = j

```

```

For j = 1 To j1

```

```

For i = 1 To j1

```

```

If Distancia(j) > Distancia(i) And j < i Then

```

```

Temp1 = Distancia(i)

```

```

Distancia(i) = Distancia(j)

```

```

Distancia(j) = Temp1

```

```

Temp2 = PosAdy(i)

```

```
PosAdy(i) = PosAdy(j)
PosAdy(j) = Temp2
```

```
Temp3X = Pmedios(0, i)
Temp3Y = Pmedios(1, i)
Pmedios(0, i) = Pmedios(0, j)
Pmedios(1, i) = Pmedios(1, j)
Pmedios(0, j) = Temp3X
Pmedios(1, j) = Temp3Y
```

```
End If
```

```
Next i
NET j
```

'lazo que determina cuantos lugares adyacentes hay

```
Nady = 0
```

```
For i = 1 To j1 Step 1
```

```
  If PosAdy(i) <> "" Then
```

```
    Nady = Nady + 1
```

```
  End If
```

```
NET
```

'se utiliza para evaluar el lugar adyacente

'que mejor cumple las condiciones para que al final

'de el mejor lugar

'determina si el lugar adyacente es de donde se viene

'si es cierto buscar los siguientes lugares, por lo

'contrario establecer el lugar

```
If (Pmedios(0, 1) = PXanterior And _
```

```
  Pmedios(1, 1) = PYanterior) And _
```

```
  Nady > 1 Then
```

'evalúa lugares adyacentes que no son repetidos

```
For i = 2 To Nady
```

'si no es repetido determinar el lugar por lo

'contrario evaluar el siguiente

```
If (Pmedios(0, i) <> Puntos(0, a - i) Or _
```

```
  Pmedios(1, i) <> Puntos(1, a - i)) Then
```

```
  Adyacente = PosAdy(i)
```

```
  PXanterior = PXactual
```

```
  PYanterior = PYactual
```

```
  PXactual = Pmedios(0, i)
```

```
  PYactual = Pmedios(1, i)
```

```
  Xp = Pmedios(0, i)
```

```
  Yp = Pmedios(1, i)
```

```
        yas = True
        Exit For
    End If
Next i
If yas Then
    yas = False
Else
    Adyacente = PosAdy(Nady)
    PXanterior = PXactual
    PYanterior = PYactual
    PXactual = Pmedios(0, Nady)
    PYactual = Pmedios(1, Nady)
    Xp = Pmedios(0, Nady)
    Yp = Pmedios(1, Nady)
    yas = False
End If

Else
    Adyacente = PosAdy(1)
    PXanterior = PXactual
    PYanterior = PYactual
    PXactual = Pmedios(0, 1)
    PYactual = Pmedios(1, 1)
    Xp = Pmedios(0, 1)
    Yp = Pmedios(1, 1)
End If

End If
End Function
Sub Graficar_ruta()
    Dim X As Double
    Dim Y As Double
    Dim i As Integer
    For i = 0 To contador Step 1
        X = Puntos(0, i)
        Y = Puntos(1, i)
        FrmPrincipal.Picture2.PSet (X, Y), vbBlue
    Next
End Sub
```

ESPECIFICACIONES TÉCNICAS GPS.

Technical data

- **Receiver:** 12 channels
- **Differential ready (DGPS):** Yes
- **Acquisition time (cold):** 2 minutes
- **Acquisition time (warm):** 1 minute
- **Update rate:** 1 per second, continuous
- **Position accuracy:** Less than 7 meters, 95% 2D RMS
- **Velocity accuracy:** 10 meters RMS; velocity of 0.1 knots RMS
- **Dynamics:** 6 Gs
- **Water-resistance rating:** IEC-529 IPX7
- **Operating temperature range:** 5.25 to 158.25 F
- **Interfaces:** 76 Datums plus 1 user-defined RTCM 104 (in), NMEA 0183 V1.5 and 2.1 (out)
- **Antenna:** Quadifiler Helix
- **Battery life:** 14 hours
- **Waypoints:** 500
- **Track log:** 2000 tracklog points
- **Track back:** Yes
- **Route log:** 20 routes
- **Points of interest:** Downloadable from MapSource CD-ROMs
- **Compass screen:** Yes
- **Moving-map display:** Information unavailable
- **Map storage:** Yes
- **Data storage:** 23 MB for map storage; 9 MB basemap
- **Sun/moon rise/set/phase:** Yes
- **Additional features:** Satellite status display, northfinder with sun/moon positions, sunrise/sunset calculator, best fish and game calculator
- **Display size:** 2.3 by 1.4 inches
- **Display resolution:** 160 x 104 pixels
- **Display color:** 4-level gray
- **Backlight:** Yes
- **Battery requirements:** 2 AA
- **Power cable:** Optional
- **Carrying case:** Optional
- **Vehicle mounting bracket:** Optional
- **PC cable:** Optional
- **Software:** Optional map software available
- **Data card:** No
- **Width:** 2.2 inches
- **Depth:** 1.3 inches
- **Height:** 5.6 inches
- **Weight (with batteries):** 6 ounces
- **Warranty, labor:** 1 year
- **Warranty, parts:** 1 year



Data Transmission Protocol Specification for Magellan Products

Revision 1.0
August 16, 1999
P/N 21-00091-000

GENERAL

- Purpose

This document describes the protocol used by Magellan's consumer GPS units to transfer waypoint data to and from an external device (presumably a PC). This protocol is an extension of the *National Marine Electronics Association NMEA 0183 Standard for Interfacing Marine Electronic Devices* Version 2.10 dated October 15, 1995.

- Intended Audience

This document does not deal with the hardware issues necessary to implement this protocol. Electrical issues involved in connecting a GPS unit to a PC, or other device, are not covered. If necessary which pins are used for data transmission to and from a Magellan GPS unit can be obtained from Magellan technical support. The intended user of this document is assumed to have some level of familiarity with serial data transfer as used by PCs and other devices. The user is assumed to be familiar with the concepts of bytes, bit order, general messaging concepts, acknowledgement based protocols and the purpose of checksums. The intended use of this document is to provide sufficient information to a developer so that data can be properly formatted for communications between Magellan products and an external Personal Computer or similar device. While not specified in this document, it is assumed that a user

interface will be provided on the Personal Computer to handle all necessary control functions to accomplish the task at hand.

Data Transmisión

- Format

All data formats in this document are assumed to be based on an 8 bit byte as the fundamental unit of transfer. All byte (0 to 255) may be used in a transmitted message, unless otherwise indicated all bytes are encoded as per ASCII standards. Although any arbitrary byte value could be used in a message that is part of this protocol, an effort has been made to stick to the limitations described in the NMEA standards. The primary affect of this is to limit most characters to ASCII characters with values of 20 (hex) to 60 (hex). This comprises numbers, uppercase letters and most of the punctuation characters. Some early Magellan units supported the use of an “Icon” within a text string. These graphics characters are defined as binary values above the NMEA limit of decimal 125. These characters were transmitted through an *escape character* mechanism. This consisted of using three bytes to represent the icon’s code, an ASCII ESC character followed by two characters that represented the icon’s code. For example, an icon whose code is 03 would be represented by the three character string <ESC>03.

- Baud Rates

This protocol can be implemented at any baud rate. For reasons dealing with the internal priority structure of the software inside most Magellan consumer GPS units, in normal mode this protocol will work best at 4800 baud. If the unit is put into transfer mode with the TON command (see below) higher baud rates are possible.

Message Sequences

The data communications messages as described in the following sections may be sent in any order. The Magellan unit will process each command as it is received, even if it means interrupting the command that is currently in process. Since a user may cause the Magellan unit to start data transmission at any time, the PC to which it is connected must expect any message to be received at any time. In normal mode there is no acknowledgement returned by the unit when it processes a message. This means that the PC cannot determine whether a message has been received and/or understood. In order to receive acknowledgement based handshaking a *hand shaking* mode has been implemented as part of this protocol. It is recommended that this mode be used when implementing this protocol. Otherwise, processor load and limited buffer size in the GPS unit could cause commands and data to be lost.

In order to activate this mode the first message that should be sent to the Magellan unit is the command **HANDON** (see below). After the next message is sent, the sending program should wait until either the Checksum response or the Unable response is received from the Magellan unit before transmitting again.

NMEA Messages

Some of the messages in this protocol are duplicates of those described in the NMEA 0183 specification. Whatever the origin of their format all the messages described in this document are treated the same and operate as described.

Magellan's consumer GPS units also have the capability of outputting streams of standard NMEA 0183 messages. These are used to convey information regarding position, velocity, time, navigation data, GPS satellite status, etc. Under certain conditions (for example, the user has activated these commands, the GPS unit is receiving GPS signals, the unit is not in *transfer mode*) these messages will be interspersed with the messages described in this document. In such cases, responsibility for handling these interruptions of the protocol reside with the external software.

MESSAGES SUPPORTED

- Format

In general, the Messages defined by this specification follow the NMEA Message structure in that they consist of a header, one or more fields, followed by a hex checksum, and ended by a Carriage Return Line Feed delimiter as follows:

$$\text{\$PMGNxxx, <fields>*hh<CR><LF>}$$

The Header portion of the message, in conformance with NMEA standards, consists of the dollar sign and the letter “P” indicating that this is a private message. Magellan’s registered private message identifier of MGN is next, followed by the command name. In the example above, the lower case “xxx” is replaced with the command identifier in upper case. A comma terminates the Header field. The data part of the message consists of one or more fields as described in this document. Commas separate each field in the data. Each field, and the data values that they can contain, are described under the various commands as shown in this document. The tail of the message consists of a two character checksum (hexadecimal notation) followed by a carriage return then a line feed. The checksum consists of the byte-wise exclusive OR of all bytes in the message. This includes the leading dollar sign and is terminated by the asterisk immediately before the checksum (which is included in the checksum).

- General Rules for Position and Time

Unless otherwise indicated all position information is referenced to the WGS-84 datum. All altitude information is referenced to the geoid (e.g. it is height above mean sea level). All date and time information is in UTC.

MESSAGE LIST

The messages that are contained in this protocol are listed in the following table..

Magellan Message	To Unit	From Unit	Description
PMGNCMD ¹	X		Command Messages.
PMGNCSM	X	X	Checksum of message that was just received.
PMGNRTE	X	X	Route Information.
PMGNTRK		X	Track information
PMGNVER		X	Hardware and software version numbers.
PMGNWPL	X	X	Description of a single waypoint
PMGNACC	X	X	Aircraft Characteristics
PMGNALM	X	X	Almanac Information.
PMGNPLN	X	X	Aircraft Flight Plans
PMGNLST	X	X	Aircraft Check Lists
PMGNMPH		X	Map header
PMGNMPU	X		Begin map upload
PMGNPIH		X	POI header
PMGNPIU	X		Begin POI upload
PMGNEML	X	X	Email messages
PMGNADD	X	X	Email addresses

Protocol Messages

Due to different information needs in different GPS units it is possible that not all empty fields at the end of the message will be transmitted. Missing, or empty, fields are to be set to the default state for each field. When receiving additional data beyond that which is defined in this document, the extra data is to be ignored.

Fields described in this document as <reserved> must be transmitted empty as shown, as they are reserved for implementation of features that have not yet been released by Magellan. When they are released, Magellan will revise this document to reflect those fields. Persons implementing this protocol that decide to use these fields without the concurrence of Magellan risk having their implementation break when Magellan begins to utilize these fields.

- PMGNCMD

This message is used to command the GPS unit to do something. This could result in data messages being sent from the unit, the unit going into a certain mode, etc. The commands are

sent as ASCII text, in upper case. Where the Command requires more than one field, the command and its fields will be sent consecutively within the same message. In no case may a command be given in one message, with the second field sent in a different message. Fields that are not needed need not be sent.

\$PMGNCMD, c---c, p--p*hh<CR><LF>

Note that the PC program can send all of these commands. The unit, however, will only send the **END** and **UNABLE** commands.

COMMAND	REPLY	DESCRIPTION
ALMANAC	PMGNALM	Send Almanac Data
DELETE		Empties the data buffer specified in the next field: WAYPOINTS – deletes all user waypoints ROUTES – deletes all user routes TRACK – deletes all points in the track buffer FIX – deletes all saved fixes ALL – deletes all of the above AIRCRAFT – deletes all aircraft characteristics
END		End of data when multiple NMEA records have been sent that does not include a record count identifier.
EPHEMERIS	GSV	Send Ephemeris data
FIX	GGA	Begin sending fix information until a STOP command is received
HANDOFF		Turns off <i>handshaking</i> mode
HANDON		Turns on <i>handshaking</i> mode
ROUTE	PMGNRTE	Send Route messages
STOP		Discontinue sending stream data at completion of current block.
TRACK	PMGNTRK	Send Track messages
UNABLE		Sent by the unit when a previous command cannot be complied with.
VERSION	PMGNVER	Send a string giving product and software version information.
WAYPOINT	WPL or PMGNWPL ²	Send NMEA Way Point Messages
CHECKLIST	PMGNLST	Send aircraft check list data.
FLIGHTPLAN	PMGNPLN	Send aircraft flight plan data.
AIRCRAFT	PMGNACC	Send aircraft characteristic data.
TON		Turn on <i>transfer</i> mode (for support of higher baud rates)
TOFF		Turn off <i>transfer</i> mode
MPHDR	PMGNMPH	Send map header information
MPUPLOAD		Put unit into map upload mode
PIHDR	PMGNPIH	Send POI header information
PIUPLOAD		Put unit into POI upload mode
MESSAGES	PMGNMSG	Send email messages
ADDRESSES	PMGNADR	Send addresses from address book

Commands

EXAMPLES

- General

This section has examples of the message exchange in a number of different scenarios. The purpose of these is to illustrate one possible sequence of messages between a PC and a Magellan unit

- Initial sequence

The following exchange will verify that a GPS unit is attached to the PC, place it in *handshaking* and *transfer* modes. The first phase of this exchange should be a repeated transmission of the request for a PMGNVER message from the unit. Once this message is received the PC will know what unit is attached and that the connection is operating correctly.

The following assumes that the GPS unit starts in its default state, which has *handshaking* mode turned off.

<u>Transmit To GPS Unit</u>	<u>Received From GPS Unit</u>
SPMGNCMD,VERSION*28	SPMGNVER,015,4.01,GPS3000XL*05
SPMGNCMD,HANDON*70	
SPMGNCMD,TON*27	SPMGNCM,27*60

Sample Initial Handshake Sequence

WAYPOINTS

- Retrieving All Waypoints From the GPS Unit

One of the most common tasks is to retrieve all of the waypoints stored in the unit for display, and/or editing on a PC. This task is accomplished by sending and receiving a sequence of messages. The PC starts the sequence by sending a command message to the unit. The unit then responds with one or more PMGNWPL messages. After all messages have been sent, the

unit will send the END command to indicate that all data has been sent. This sequence is shown below which assumes that there are only two waypoints stored in the unit.

<u>Sent To GPS Unit</u>	<u>Received From GPS Unit</u>
SPMGNCMD,WAYPOINT*71	SPGMNCSM,71*63
SPGMNCSM,hh*hh	SPMGNWPL,lll.l,a,yyyyy.yy,a,c--c*hh
SPGMNCSM,hh*hh	SPMGNWPL,lll.l,a,yyyyy.yy,a,c--c*hh
SPGMNCSM,3D*12	SPMGNCMD,END*3D

Retrieving All Way Points

MAPA DE LA UNIVERSIDAD DON BOSCO.



ESCALA 1: 1000.

GLOSARIO.

Comando (definición): Palabra o frase hablada con el propósito de obtener cierto resultado. En FreeSpeech los comandos son usados para dictar y editar textos, navegar por programas y mucho más.

Control (definición): Mediante el uso de su voz puede moverse por ventanas, cuadros de diálogo, documentos, etc. en el entorno Windows.

Dictado (definición): Hablar por un micrófono para convertir el habla en texto.

Entrenamiento (definición): Durante el Entrenamiento Usted lee textos en voz alta, mientras FreeSpeech aprende sus características vocales. FreeSpeech genera un archivo de referencia vocal para mejorar el porcentaje de reconocimiento. Se recomienda hacer este Entrenamiento.

FreeSpeech (definición): Aplicación de reconocimiento del habla. Brinda habla-en-texto (dictado) y la función de Comando & Control para varios usuarios. Es posible dictar en diferentes idiomas.

Modo (definición): Forma de trabajar con FreeSpeech. El programa tiene cuatro modos: el Modo de dictado para dictar textos, el Modo de deletreo para deletrear palabras, el Modo de comando para usar las funciones de Comando & Control, y el Modo de descanso para que FreeSpeech esté en standby.

Modo de comando (definición): Modo de FreeSpeech que le permite navegar por el sistema operativo, los programas, cuadros de diálogo, documentos y también formatear y editar textos.

Reconocimiento continuó del habla (definición): Reconocimiento del habla natural, es decir sin pausas entre las palabras. FreeSpeech transforma las palabras habladas en texto mientras usted habla de manera natural.

Reconocimiento del habla (definición): Sistema computarizado mediante el cual el programa transforma el texto hablado en texto escrito o ejecuta los comandos de voz.

Tarjeta de sonido (definición): Tarjeta de adaptador del hardware de su computadora que permite la entrada y salida de señales de audio, por ejemplo, por medio de un micrófono y altavoces.

CODIGO FUENTE DE LA AYUDA.

#

**Ayuda sobre las diferentes aplicaciones y comandos del proyecto
LAZARILLO ELECTRONICO**

El usuario puede obtener información acerca de los siguientes temas :

Debido a que es orientado para usuarios no videntes, este utiliza varios comandos de voz los cuales realizan diferentes funciones.

Por la naturaleza del proyecto este será capaz de funcionar bajo diferentes lugares lo que lo vuelve un sistema abierto y más funcional.

Para adaptarse al oído de cada usuario este sistema tendrá la posibilidad de modificar la velocidad y el timbre del emulador de voz.

principal

#

#K\$

Seleccionar Zona

Esta opción permite al usuario cambiar la zona en la que se esta desplazando.

La forma de cambiar el mapa es la siguiente:

- 1.Clic en menu *archivo*.
- 2.Luego escoger *seleccionar zona*.

Buscar en la ventana del explorador el archivo con extensión .MDB que contiene la información de las coordenadas de la nueva zona.

zona

^K cambio de zona

^S zona

K
S

S K

Cambio de tipo de voz

Esta opción permite al usuario cambiar el timbre de voz por una masculina o femenina.

La forma de cambiar la voz es la siguiente:

- 1.Clic en menú *Herramientas*.
- 2.Luego escoger *Selección del Habla*
- 3.En la ventana emergente seleccionar el tipo de voz a utilizar
4. Si se desea tener una idea de cómo será el tipo de voz seleccionado, hacer clic en el botón *Probar*.
5. Finalmente cerrar la ventana.

voz

S voz

K tipo de voz

S
K

K S

Personalizar Nombre

Esta opción permite al usuario ser llamado por su nombre al momento que el sistema le transmita algún tipo de información.

La forma de personalizar el nombre en el mensaje es la siguiente:

- 1.Clic en menu *Archivo*.
- 2.Luego escoger *Nombre de Usuario*.
3. Digitar en la casilla en blanco el nombre
- 4.Clic en *Cerrar* para finalizar

nombre

^K personalizar nombre

^S nombre

K
S

#KS

Ajuste de velocidad del habla.

Esta opción permite al usuario mediante una barra de desplazamiento disminuir o aumentar con la cual será reproducido el fonema de la voz humana.

La forma de personalizar la velocidad de reproducción es la siguiente:

1. Clic en menú *Herramientas*.
2. Luego escoger *Selección del habla*.
3. En la ventana emergente desplazar el botón hacia la derecha para aumentar o izquierda para disminuir la velocidad.
4. Clic en *Cerrar* para finalizar

velocidad

^K velocidad del habla

^S habla

K
S

SK

Desplazarse a un lugar de destino específico.

Esta comando de voz permite al usuario colocar al sistema en modo guiado. La forma de utilizar este comando es decir la frase destino.

Luego de ejecutar el comando, el sistema solicitara el lugar al cual se pretende llegar, para calcular la ruta mas conveniente posible.

A su vez el sistema ira indicando al usuario hacia que dirección avanzar, así como los lugares por donde se vaya desplazando.

destino

\$ destino

K modo guiado

\$
K

#KS

Posición Actual.

Esta opción permite al usuario saber la posición actual donde se encuentra.

La forma de utilizar este comando es decir la frase donde estoy

Después de medio segundo el sistema reproduce un fonema con el cual indica el lugar donde se encuentra.

donde estoy

^K posición actual

^S actual

K
S

#KS

Detener subrutina de Destino.

Esta comando de voz permite al usuario detener en cualquier momento la ejecución de la subrutina “destino” a su vez, corre la rutina “donde estoy” para indicarle al usuario la posición actual donde esta, después de haber detenido el proceso.

La forma de utilizar este comando es decir la frase alto

alto

^ detener destino

\$ alto

^
\$

#KS

Cerrar la aplicación.

Esta opción permite al usuario cerrar por completo el entorno grafico del Lazarillo Electrónico.

Otra forma de efectuar la misma operación es la siguiente:

1.Clic en menu *Archivo*.

2.Luego escoger *Salir*.

Sino presionando la combinación de teclas *Ctrl + S*

salir

^K cerrar aplicación

^S cerrar

K
S

#KS

Reestablecer Valores Predeterminados.

Puede restablecer la configuración de la aplicación a valores predeterminados, incluidos los valores de tono y velocidad de la voz del emulador. Otra forma de efectuar la misma operación es la siguiente:

- 1.Clic en menu *Archivo*.
- 2.Luego escoger *Configuración Inicial*.

default

^K reestablecer valores

^S predeterminados

K
S

#KS

Crear Base de Datos.

Esta opción permite al usuario agregar o modificar la información contenida en las tablas, además posee una interfase grafica la cual hace esta operación bastante sencilla e interactiva .

La forma de agregar lugares o caminos a la base de datos es de la siguiente manera:

1. Clic en menú *Herramientas*.

2. Luego escoger *Mantenimiento*.

3. En la ventana emergente seleccionar la opción *Crear Base de Datos* posteriormente surgirá una ventana en la cual habrá que seleccionar lo siguiente:

- Si el nuevo dato se almacenara en la tabla de lugares o en la de caminos
- Especificar si será una área o si la selección será de forma lineal.
- Asignar un nombre al área introducida, después presionar el botón nombrar, para comenzar a puntear la zona

4. Finalmente clic en *Aceptar* para que el nuevo dato sea almacenado .

crearbase
κ crear base
§ base

#KS

Ver Base de Datos.

Esta opción permite al usuario acceder a la información contenida en las tablas de caminos y lugares con el objeto de comprobar cierta información o modificar el valor de uno de sus contenidos .

La forma de acceder y visualizar dichas tablas es de la siguiente manera:

- 1.Clic en menú *Herramientas*.
2. Luego escoger *Mantenimiento*.
3. En la ventana emergente seleccionar la opción *Ver Base de Datos* posteriormente surgirá un archivo de Microsoft Acces el cual contiene toda la información de los campos contenidos.

verbase

κ ver base

\$ base

#

κ

\$

#KS

Entrar en modo Gráfico.

Esta opción permite al usuario visualizar la información contenida en las tablas, con la variante que lo hace en un ambiente totalmente gráfico es decir, se puede desplazar por toda la zona bajo estudio y saber los datos de sus coordenadas .

La forma de acceder a este tipo de entorno es de la siguiente manera:

1. Clic en menú *Herramientas*.
2. Luego escoger *Mantenimiento*.
3. En la ventana emergente seleccionar la opción *Modo Gráfico* posteriormente surgirá una ventana en la cual se tendrá información como latitud, longitud y ubicación dentro del mapa del punto correspondiente.

modografico

κ modo grafico

\$ grafico

#

κ

\$

#Pronunciar el comando de voz en forma clara y legible

estoy

#Asegurarse de tener encendido los parlantes, para escuchar el resultado de su selección

probar

#Estos suelen ser palabras cortas que ejecutan una acción específica.

comandos

#Lo que implica seleccionar y cargar el mapa de la zona en que me encuentro.

lugares

#Esto es por si el usuario esta acostumbrado a una fluidez de palabras específica.

ajuste

#Tener la opción de escoger entre una voz femenina o masculina.

timbre

#Configuración con la que se cargaran todas las opciones luego de instalar la aplicación.

predeterminados

#

#

#

#

#

#

predeterminados

Manual del Usuario.



Manual del Usuario Lazarillo Electrónico.

Sistema para determinar la posición del usuario dentro de una zona bajo estudio, basado en un sistema de posicionamiento global. (GPS)

INDICE.

FORMULARIOS.....	42
MODULOS.....	42
PROCEDIMIENTOS.....	43
FUNCIONES.....	44
<u>Ayuda sobre las diferentes aplicaciones y comandos del proyecto LAZARILLO</u>	
ELECTRONICO.....	113
.....	113
.....	114
Seleccionar Zona.....	114
Cambio de tipo de voz.....	115
Personalizar Nombre.....	116
Ajuste de velocidad del habla.....	117
Desplazarse a un lugar de destino específico.....	118
Posición Actual.....	119
Detener subrutina de Destino.....	120
Cerrar la aplicación.....	121
Reestablecer Valores Predeterminados.....	122
Crear Base de Datos.....	123
Ver Base de Datos.....	124
Entrar en modo Gráfico.....	125
Instalación del Lazarillo Electrónico.....	4
Objetivos.....	4
Proceso de Instalación.....	5
Funcionamiento básico de la aplicación.....	7
Objetivos.....	7
Cambiar el nombre del usuario.....	7
Seleccionar zona bajo estudio.....	8
Configuración inicial.....	9
Cerrar la aplicación.....	9
Crear base de datos.....	9
.....	10
Ver base de datos.....	10
Utilizar modo gráfico.....	11
Seleccionar el tipo de habla.....	13

Ayuda.....	14
Versión de la aplicación.....	14

Instalación del Lazarillo Electrónico.

Objetivos.

Al terminar este modulo usted será capaz de realizar las siguientes acciones:

- ✓ Identificar los requisitos necesarios para la instalación.
- ✓ Instalar correctamente el Lazarillo Electrónico

Requisitos necesarios para la instalación del Lazarillo Electrónico.

- HARDWARE

Mínimo		Recomendado	
Procesador	Pentium II 233 Hz	Procesador	Pentium III 500 Hz
RAM	64 MB	RAM	128 MB
VIDEO	8 MB	VIDEO	16 MB
CD ROM	24 X	CD ROM	32 X
Color	24 bits	Color	32 bits
Disco Duro	4 GB	Disco Duro	10 GB

Tabla 1. Requisitos de Hardware para la instalación.

- SOFTWARE

- ✓ Se requiere como sistema operativo a Windows 98 o superior.
- ✓ La resolución debe establecerse en 1024 x 768 píxeles.

Proceso de Instalación.

Con un doble clic ejecute el instalador del *Lazarillo Electrónico*

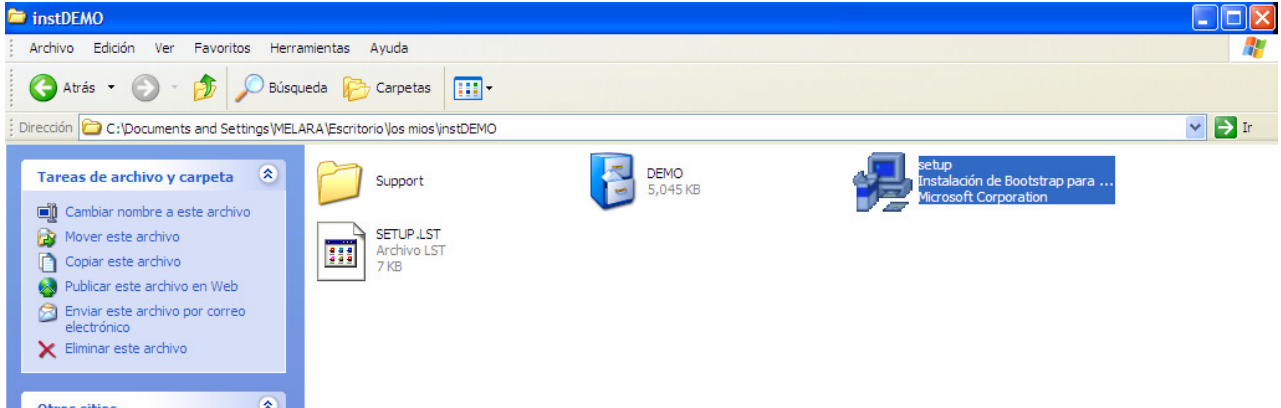


Figura 1. Proceso de instalación del Lazarillo Electrónico.

Cuando aparezca el ayudante del instalador, presione el botón

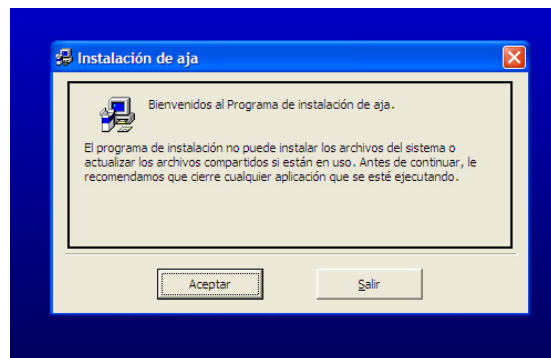


Figura 2. Ayudante del instalador.

Seleccione la ruta donde se instalara el programa por defecto.

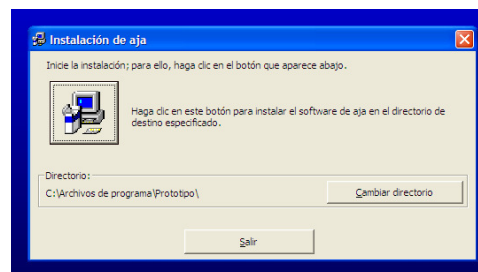


Figura 3. Ruta donde quedara almacenado el programa.

Presione el botón  para continuar.

El Lazarillo esta listo para ser instalado. Presione continuar.

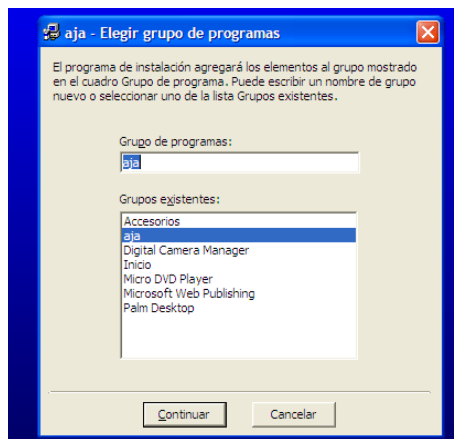


Figura 4. Programas relacionados con el archivo.

Después de esto se inicia el proceso de instalación.

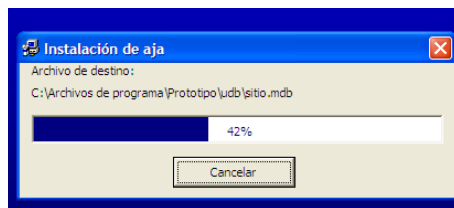
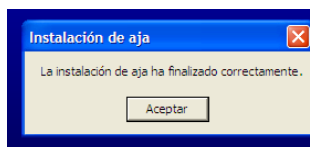


Figura 5. Proceso de instalación.

Finalmente presione  para finalizar.



Funcionamiento básico de la aplicación.

Objetivos.

Al terminar este modulo usted será capaz de realizar las siguientes acciones:

- ✓ Cambiar el nombre del usuario.
- ✓ Seleccionar zona bajo estudio.
- ✓ Configuración inicial.
- ✓ Cerrar la aplicación.
- ✓ Crear base de datos.
- ✓ Ver base de datos.
- ✓ Utilizar modo gráfico.
- ✓ Seleccionar el tipo de habla
- ✓ Ayuda
- ✓ Versión de la aplicación.

Cambiar el nombre del usuario.

Esta opción permite al usuario ser llamado por su nombre al momento que el sistema le transmita algún tipo de información.

La forma de personalizar el nombre en el mensaje es la siguiente:

- 1.Clic en menu Archivo.
- 2.Luego escoger Nombre de Usuario.
3. Digitar en la casilla en blanco el nombre
- 4.Clic en Cerrar para finalizar

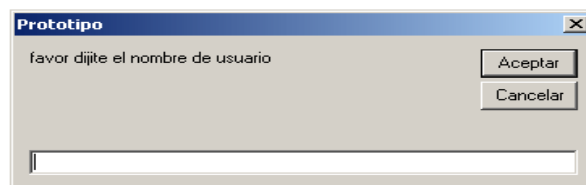


Figura 6. Ventana para cambiar nombre del ususario.

Seleccionar zona bajo estudio.

Esta opción permite al usuario cambiar la zona en la que se está desplazando.

La forma de cambiar el mapa es la siguiente:

- 1.Clic en menu archivo.
- 2.Luego escoger seleccionar zona.

Buscar en la ventana del explorador el archivo con extensión.MDB que contiene la información de las coordenadas de la nueva zona.

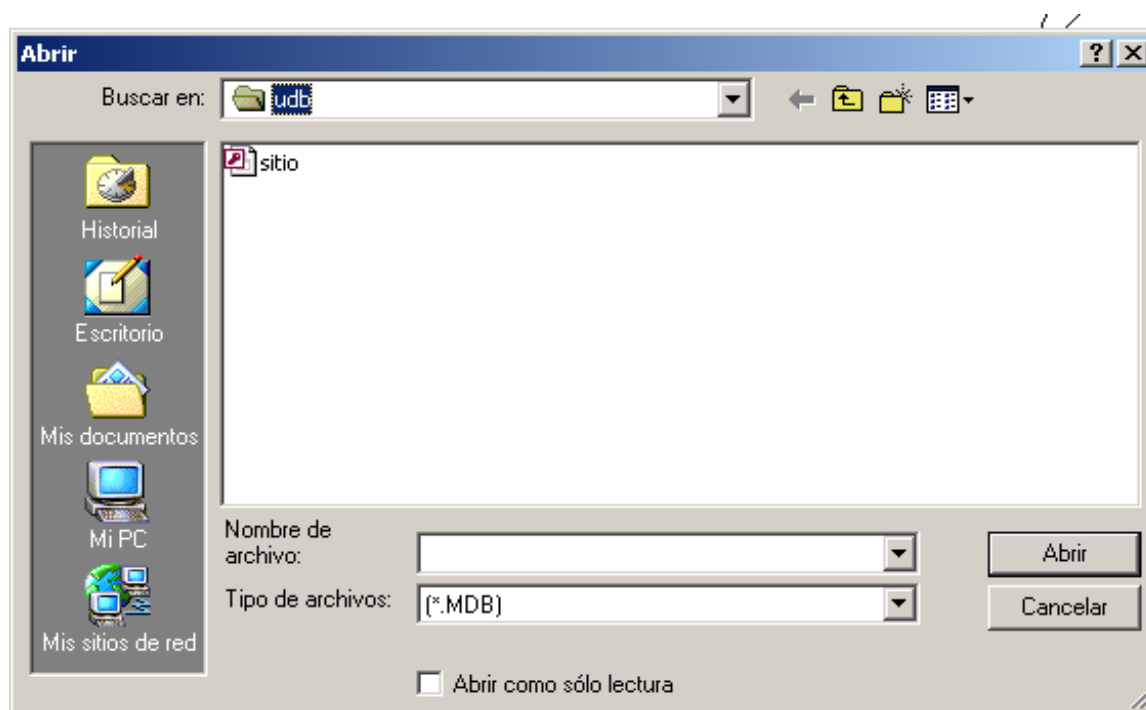


Figura 7. Ventana para seleccionar zona.

Configuración inicial.

Esta opción permite usuario cargar la aplicación con los valores predeterminados del sistema, la forma de ejecutar esta herramienta es la siguiente:

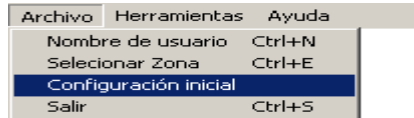


Figura 8. Menu para seleccionar la configuración inicial.

Cerrar la aplicación.

Esta opción permite cerrar de manera más fácil y rápida la aplicación.

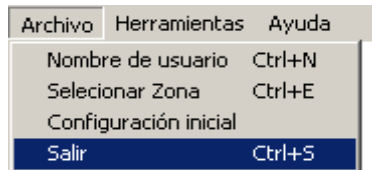


Figura 9. Menu para seleccionar salir del programa.

Crear base de datos.

Esta opción permite al usuario agregar o modificar la información contenida en las tablas, además posee una interfase grafica la cual hace esta operación bastante sencilla e interactiva.

La forma de agregar lugares o caminos a la base de datos es de la siguiente manera:

- 1.Clic en menú Herramientas.
2. Luego escoger Mantenimiento.
3. En la ventana emergente seleccionar la opción Crear Base de Datos posteriormente surgirá una ventana en la cual habrá que seleccionar lo siguiente:

- Si el nuevo dato se almacenara en la tabla de lugares o en la de caminos
 - Especificar si será una área o si la selección será de forma lineal.
 - Asignar un nombre al área introducida, después presionar el botón nombrar, para comenzar a puntear la zona
4. Finalmente clic en Aceptar para que el nuevo dato sea almacenado

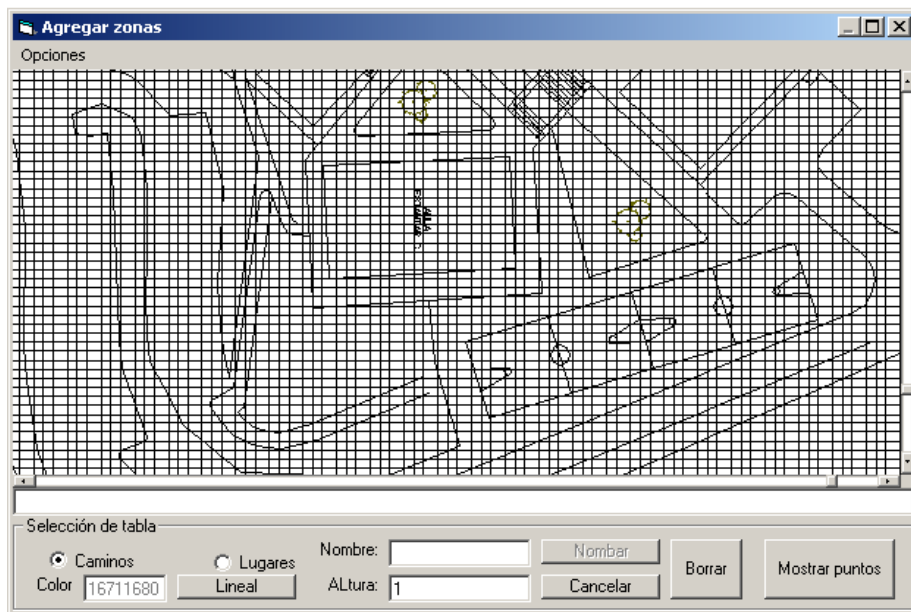


Figura 10. Ventana para crear la base de datos.

Ver base de datos.

Esta opción permite al usuario acceder a la información contenida en las tablas de caminos y lugares con el objeto de comprobar cierta información o modificar el valor de uno de sus contenidos .

La forma de acceder y visualizar dichas tablas es de la siguiente manera:

1. Clic en menú Herramientas.
2. Luego escoger Mantenimiento.
3. En la ventana emergente seleccionar la opción Ver Base de Datos posteriormente surgirá un archivo de Microsoft Acces el cual contiene toda la información de los campos contenidos.

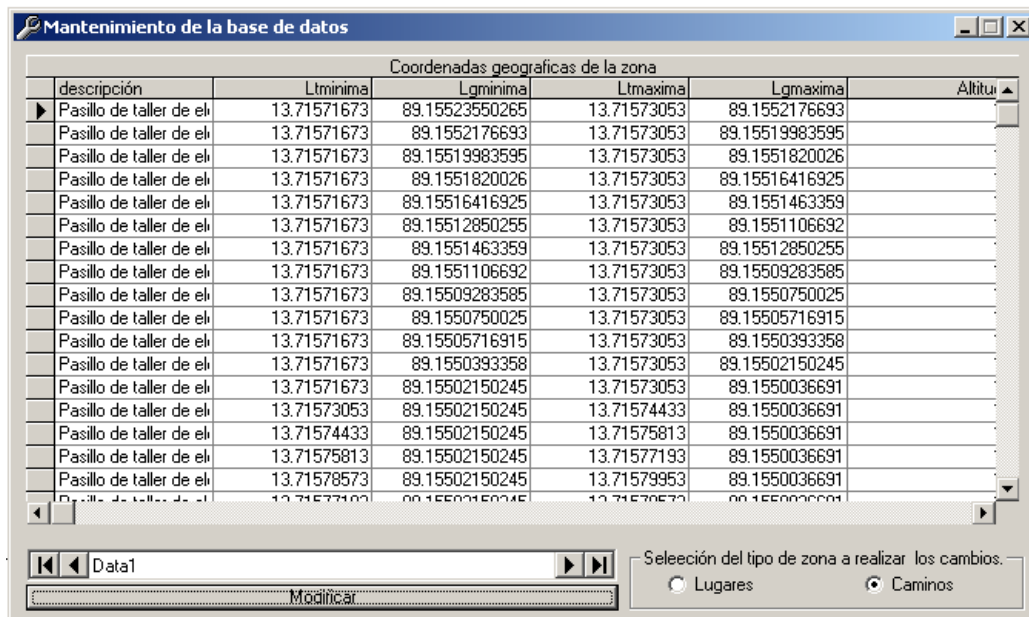


Figura 11. Ventana para ver la base de datos.

Utilizar modo gráfico.

Esta opción permite al usuario visualizar la información contenida en las tablas, con la variante que lo hace en un ambiente totalmente grafico es decir, se puede desplazar por toda la zona bajo estudio y saber los datos de sus coordenadas .

La forma de acceder a este tipo de entorno es de la siguiente manera:

1. Clic en menú Herramientas.
2. Luego escoger Mantenimiento.

3. En la ventana emergente seleccionar la opción Modo Grafico posteriormente surgirá una ventana en la cual se tendrá información como latitud, longitud y ubicación dentro del mapa del punto correspondiente.

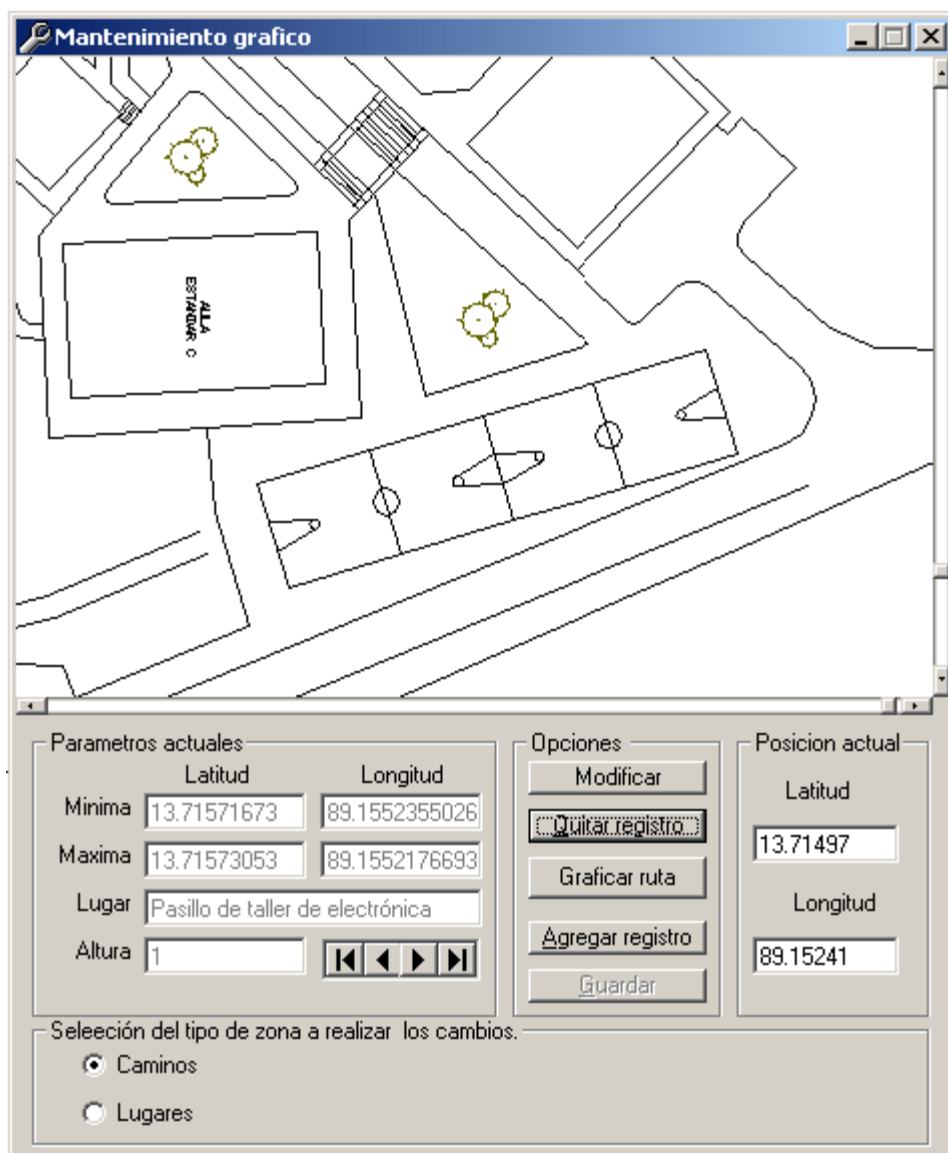


Figura 12. Ventana para ver el modo gráfico.

Seleccionar el tipo de habla

Esta opción permite al usuario cambiar el timbre de voz; además, mediante una barra de desplazamiento disminuir o aumentar con la cual será reproducido el fonema de la voz humana.

La forma de cambiar esto es la siguiente:

1. Clic en menú Herramientas.
2. Luego escoger Selección del Habla
3. En la ventana emergente seleccionar el tipo de voz y desplazar el botón hacia la derecha para aumentar o izquierda para disminuir la velocidad a utilizar
4. Si se desea tener una idea de cómo será el tipo de voz seleccionado, hacer clic en el botón Probar.
5. Finalmente cerrar la ventana.

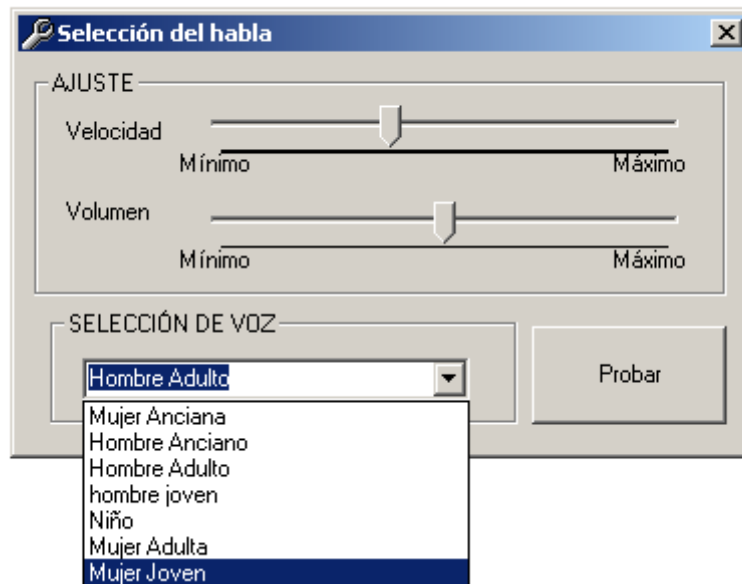


Figura 13. Ventana para seleccionar el tipo del habla.

Ayuda

Presenta el contenido de las funciones del Lazarillo Electrónico

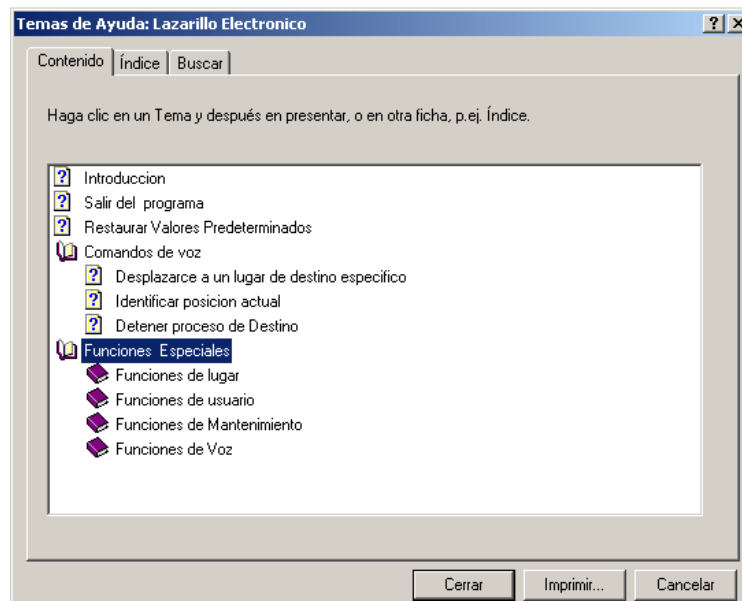


Figura 14. Ventana para visualizar la ayuda del sistema.

Versión de la aplicación.

Muestra el nombre y versión de la aplicación, además algunos datos acerca de los creadores de la aplicación así como sus respectivos contactos.



Figura 15. Ventana para visualizar la versión y autores del sistema.