UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

“Sistema de Rastreabilidad GPS con Interfase WEB y SMS para Dispositivos Móviles”

T E S I S

QUE PARA OBTENER EL TITULO DE
INGENIERO EN COMPUTACION

PRESENTA:

JOSE ALBERTINO MARTINEZ OSORIO

DIRECTOR DE TESIS

M.I. JORGE VALERIANO ASSEM

MEXICO D.F.

AGRADECIMIENTOS

A Dios

Por poner los medios, los tiempos y las personas correctas en el transcurso de la carrera y así poder cumplir un anhelo más en mi caminar.

Te agradezco que me hayas forzado a concluir este paso tan importante al enseñarme que eres un Dios de proyectos finalizados, no de proyectos a medias o mediocres

A mis Padres, Salvador Martínez y Maria Osorio de Martínez

Porque con esfuerzo, lagrimas y sobre todo con el corazón y el sacrificio, me dieron la oportunidad de seguir estudiando.

Especialmente a ti mama que por las mañanas me despertabas y me dabas el ánimo de seguir en esta empresa y que por las noches estabas al pendiente de mi descanso y sobre todo por tus oraciones.

Gracias porque ahora conozco el verdadero valor del sacrificio el cual devolveré en mis hijos

A mis hermanos, Micael, Noe, Ramiro, Erika, Jazmin.

Porque gracias a ellos encontraba el animo cuando el estudio se tornaba difícil.

A mi Esposa Elizabeth e Hijos Brigitte y Emmanuel

Los cuales gracias a su amor, me insistían en que terminara y sobre todo me hicieron ver la gran responsabilidad que tengo al mostrarles que los proyectos tienen como principal objetivo su conclusión

A mi director de tesis Jorge Valeriano Assem , Jurado y maestros de la carrera

En la vida he aprendido que existe un principio que llamo “de multiplicación” y gracias a ustedes, parte de sus conocimientos y habilidades son multiplicados en cada uno de nosotros y son enseñanzas que permanecen toda la vida.

A todos ustedes Dios los llene de bendición conforme a los deseos de su corazón

I N D I C E

INTRODUCCION

Hoy en día el sistema GPS ha cobrado gran importancia en el desarrollo tecnológico. En sus inicios este sistema tenía la ardua tarea de brindar el servicio de localización y posicionamiento global con fines militares y hoy en día es utilizado en innumerables aplicaciones extremas de ingeniería las cuales son muy distintas al propósito original tales como:

- Aportar datos acerca de cómo el utilizar las desviaciones de la señal al cruzar la atmósfera para obtener datos acerca de las condiciones climatológicas

- En la navegación marítima se está estudiando la posibilidad de evaluar las diferencias de la reflexión de la señal en la superficie del océano para conocer la dirección y fuerza del viento y altitud de las olas.

- La utilización de la señal GPS (su contenido temporal) para sincronizar las transmisiones en Internet, esto debido a que los satélites poseen relojes atómicos.

Todas estas aplicaciones son posibles ya que el sistema GPS está disponible las 24 horas del día, con cobertura en prácticamente todo el planeta tierra, gracias a que sus satélites siguen rutas programadas para garantizar la señal. Pero lo más importante de todas, es que las señales GPS (código C/A) están al alcance de todos, gratuitamente sin necesidad de pagar licencia o rentas por su uso (el código denominado P(Y) es de uso militar y restringido a usuarios autorizados).

Adicional a esto, los satélites cuentan con relojes atómicos precisos, y se podría pensar que para que nuestro receptor de GPS mantuviera la sincronía, sería necesaria la inversión en receptores que tuvieran integrado también un reloj atómico, el cual elevaría el costo y lo haría inaccesible. Pero mediante técnicas de cálculo que consiste en una lectura satelital adicional, se puede lograr la sincronía aun nuestros relojes simples y mucho menos precios. Esto significa que en teoría, cada receptor de GPS es en esencia un reloj atómico por su precisión

Por lo tanto; sí solo tenemos que invertir en un receptor de GPS, ¿porque las aplicaciones actuales en el mercado son tan caras?. Y la prueba de ello es que actualmente existen en el mercado múltiples aplicaciones de diversas compañías que nos ofrecen localización global, pero dichas aplicaciones son costosas o son adquiridas a través de planes de pago forzosos que lo hacen inaccesibles para la mayoría de la población.

Por este motivo, se plantea esta solución con el fin de abatir dichos costos de adquisición y/o renta, y contribuir a la exploración de esta tecnología y sentar las bases para aplicaciones mucho más complejas, y todo esto gracias a que hoy en día la evolución de los dispositivos móviles nos permiten obtenerlos con receptor GPS integrado y un bajo costo.

Asimismo aprovechando la funcionalidad de telefonía, se está dotando al sistema con envíos de mensajes SMS a través del dispositivo, el cual enviará vía mensaje de texto la información con la posición global actual, el cual puede ser de mucha utilidad cuando

nos encontremos en casos de extrema emergencia o simplemente para compartir nuestra ubicación.

Pero esta aplicación estaría incompleta sin no contara con un mecanismo para poder visualizar gráficamente la ubicación del dato enviado, por lo tanto, dotaremos al sistema de un portal WEB (el cual será accesible para cualquier persona que tenga conexión a Internet), con el objetivo de que el usuario pueda monitorear y analizar los registros que el dispositivo móvil genera y envía. Y esto lo lograremos generando las instrucciones necesarias para realizar la interfase con Google Maps, la cual es gratuita.

Aunado a lo anterior, y echando mano nuevamente de aplicaciones gratuitas, generaremos la solución necesaria para identificar gráficamente la ruta en Google Earth, dentro del cual identificaremos los sitios o ubicaciones registradas por el dispositivo móvil, de un día en particular y desde un punto en específico.

Cabe aclarar, que Google Earth, Google Maps y Google Latitud, son aplicaciones con derechos reservados de sus respectivos autores los cuales pueden descargarse de manera gratuita de sus sitios y que solo son utilizadas por este sistema como interfase grafica

CAPITULO I. MARCO TEORICO

I.I. Historia del GPS

La tecnología, en cuanto a las ciencias de la tierra, supuso un gran avance desde que en 1957, fueron enviados los primeros satélites artificiales a bordo del Sputnik-1. De manera notable, se avanzo en el estudio de su forma y dimensión con la Geodesia, así como el estudio de fenómenos físicos inherentes a su forma y dimensión como la Geofísica.

El Sistema GPS (Sistema de Posicionamiento Global) fue creado por el Departamento de Defensa de los Estados Unidos para constituir un sistema de navegación preciso con fines militares que sustituyera al antiguo sistema utilizado, que no era otro que las mediciones Doppler sobre la constelación Transit. Para ello, aprovecharon las condiciones de la propagación de las ondas de radio de la banda L en el espacio, así como la posibilidad de modular las ondas para que en ellas se pueda incluir la información necesaria que permita posicionar un objeto en el sistema de referencia apropiado. Este proyecto se hizo realidad entre los meses de febrero y diciembre de 1978, cuando se lanzaron los cuatro primeros satélites de la constelación NAVSTAR, que hacían posible el sistema que resolvería la incógnita de nuestra posición en la Tierra.

Para entender las generalidades del sistema GPS y sus características más importantes, debemos dividir el sistema en tres sectores fundamentales y dependientes entre sí, el sector espacial, el sector de control y el sector de usuarios.

I.I.I. El sector espacial NAVSTAR.

Este sector lo forman los satélites de la constelación NAVSTAR (Navegación por satélite en tiempo y distancia). La constelación está formada por seis planos orbitales, y en cada uno de ellos existe una órbita elíptica casi circular donde se alojan los satélites regularmente distribuidos. Los planos tienen una inclinación de 55º respecto al plano del ecuador, y se nombran como A, B, C, D, E y F. Cada órbita contiene al menos cuatro satélites, aunque pueden contener más. Los satélites se sitúan a una distancia de 20,200 Km. respecto del geocentro, y completan una órbita en doce horas sidéreas. Estos satélites son puestos en funcionamiento por el Comando de las Fuerzas Aéreas Espaciales de U.S.A (AFSPC), la imagen 1.1., nos da una idea.

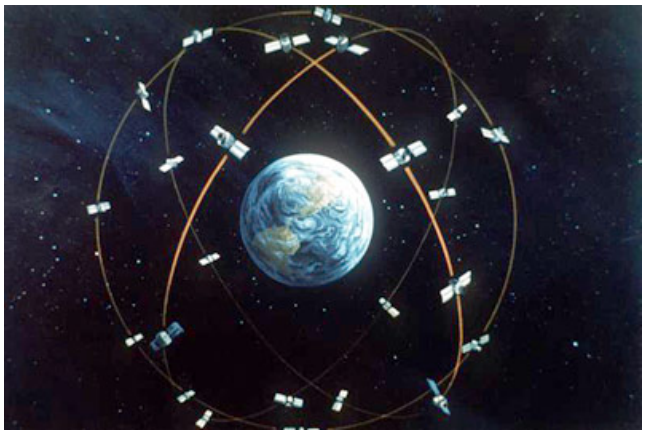


Imagen 1.1. Interpretación del Sector Espacial NAVSTAR

Con estos fundamentos, se garantiza la presencia de al menos cuatro satélites sobre el horizonte en todos los lugares de la superficie de la Tierra.

Los satélites de la constelación NAVSTAR son identificados de diversos modos:

- Por su número NAVSTAR (SVN).
- Por su código de ruido pseudoaleatorio (PRN). En los códigos de transmisión existen características de ruido pseudoaleatorio traducidas en bits que identifican a cada satélite de la constelación
- Por su número orbital. Un ejemplo sería el satélite 3D, que corresponde al satélite número tres del plano orbital D.

En enero de 1998, se disponía de un número de veintisiete satélites operativos, pertenecientes a los bloques IIA y IIR. Se disponen:

- Cinco en los planos A, E y F.
- Cuatro en los planos B, C y D.

Todos disponen de osciladores atómicos de cesio, salvo los SVN 24, 27 y 31 que lo tienen de rubidio. En el caso de los primeros la precisión es de $10^{-13}\,s$, mientras que los de rubidio es de $10^{-12}\,s$. La frecuencia fundamental de emisión de estos osciladores es de 10,23 MHz.

El tiempo utilizado por el sistema GPS es un tiempo universal coordinado denominado UTC (USNO) que define el Observatorio Naval de los Estados Unidos mediante relojes atómicos de hidrógeno. La unidad del tiempo GPS es el segundo atómico internacional y tiene su origen coincidente con el UTC a las cero horas del 6 de enero de 1980.

Las coordenadas, tanto de los satélites como de los usuarios que se posicionan con el sistema GPS, están referidas al sistema de referencia WGS84 (Sistema Geodésico Mundial de 1984). Estas coordenadas pueden ser cartesianas en el espacio respecto al centro de masas de la Tierra (X, Y, Z) o geodésicas (φ,λ,h). El sistema tiene las siguientes características:

- Origen en el Centro de Masas de la Tierra.
- El eje Z es paralelo al polo medio.
- El eje X es la intersección del meridiano de Greenwich y el plano del ecuador.
- El eje Y es perpendicular a los ejes Z y X, y coincidente con ellos en el Centro de Masas terrestre.

I.I.II. El sector de Control NAVSTAR.

Este sector tiene como misión el seguimiento continuo de todos los satélites de la constelación NAVSTAR para los siguientes fines:

- Establecer la órbita de cada satélite, así como determinar el estado de sus osciladores.
- Hallados los parámetros anteriores, emitirlos a los satélites para que éstos puedan difundirlos a los usuarios.

De este modo, el usuario recibe la información de las efemérides de posición de los satélites y el error que se está produciendo en su reloj, todo ello incluido en el mensaje de navegación.

Las Estaciones de Control de la constelación son fundamentalmente:

- Colorado Springs (U.S.A.). Central de cálculo y operaciones.
- Ascensión (Atlántico Sur).
- Hawai (Pacífico Oriental).
- Kwajalein (Pacífico Occidental).
- Diego García (Indico).

Existen además otras estaciones de seguimiento (láser, radar y ópticas), cuyo fin es la obtención de efemérides que no estén afectadas por la disponibilidad selectiva, denominadas precisas, y que están al alcance del usuario a través de organismos científicos como el IGS (International Geodinamic Service) o el NGS (National Geodetic Survey). Con ellas, tenemos la seguridad de posicionarnos en el sistema WGS84 con los errores típicos del sistema.

I.I.III. El sector de Usuarios NAVSTAR.

Este sector lo compone el instrumental que deben utilizar los usuarios para la recepción, lectura, tratamiento y configuración de las señales, con el fin de alcanzar los objetivos de su trabajo. Los elementos son el equipo de observación y el software de cálculo, que puede ser objeto de uso tras la campaña de observación, o bien realizable en tiempo real, donde se obtienen los resultados en sitio.

Equipo de observación. Lo componen la antena, el sensor y la unidad de control o controlador.

- La antena de recepción tiene la misión de recibir las radiaciones electromagnéticas que emiten los satélites y transformarlas en impulsos eléctricos, los cuales conservan la información modulada en las portadoras. Se denomina centro radioeléctrico de la antena al punto que se posiciona en nuestra observación. Dado que éste no suele coincidir con el centro físico, es conveniente orientar todas las antenas de una misma observación en la misma dirección con el fin de que el error se elimine.

- El sensor recibe los impulsos de la antena receptora, y reconstruye e interpreta los componentes de la señal, es decir, las portadoras, los códigos y el mensaje de navegación. En definitiva, lo que hace es demodular la señal original.

  El proceso es el siguiente, el sensor correlaciona los códigos, es decir, lo compara con una réplica que él mismo genera, y de este modo halla el tiempo que ha tardado en llegar la señal al receptor, obteniendo la distancia al satélite multiplicando esa diferencia de tiempos por el valor de la velocidad de propagación de las ondas en el vacío (aproximadamente unos 300,000 Km/s). Como estas distancias están afectadas de errores, se las denomina seudodistancias. El controlador realiza las tareas de controlar el sensor, gestionar la observación y almacenar los datos.

El Sistema de Posicionamiento Global NAVSTAR no es el único Sistema de Posicionamiento existente. El Sistema Ruso GLONASS es también operativo, y a pesar de que actualmente la constelación no está completada, proporciona a los usuarios civiles unas precisiones en el posicionamiento absoluto típicamente mejores que las que proporciona el Sistema GPS, debido a la aplicación de la degradación intencionada de la información denominada Disponibilidad Selectiva (SA)

En el año 1993, oficialmente el Gobierno Ruso colocó el programa GLONASS en manos de Fuerzas Espaciales Militares Rusas (RSF). Este organismo es el responsable del desarrollo de satélites GLONASS, de su mantenimiento y puesta en órbita, y certificación a los usuarios. Este organismo opera en colaboración con el CSIC (Coordinational Scientific Information Center), el cual publica la información sobre GLONASS.

Durante los 80s, la información acerca de GLONASS era escasa. No se sabía mucho de las órbitas de los satélites ni de las señales usadas para transmisión de las señales de navegación. Pero actualmente, gracias a estudios e investigaciones sobre este sistema, se dispone ya de gran cantidad de información acerca GLONASS. Los Rusos, a través del RSF y del CSIC publican el documento ICD (Interface Control Document). Este documento es similar en estructura al Segmento Espacial del sistema NAVSTAR GPS, donde se describe el sistema, sus componentes, estructura de la señal y el mensaje de navegación para uso civil.

La constelación ha experimentado un gran progreso desde los años 1994 y 1995. Los planes de GLONASS son ofrecer dos niveles de servicio:

- El Channel of Standard Accuracy (CSA), similar al Standar Positioning Service (SPS) del Sistema GPS, disponible para uso civil.
- El Channel of High Accuracy (CHA), similar al Precise Positioning Service (PPS) del Sistema GPS, disponible solo para usuarios autorizados.

La Organización Internacional de Aviación Civil (ICAO) aceptó formalmente en Julio 1996, el uso de GLONASS/CSA para uso en aviación civil, como ya se hizo en 1994 con el GPS/SPS.

El Sistema GLONASS, al igual que el Sistema GPS, está formado por tres sectores fundamentales: el Sector de Control, el Sector Espacial y el Sector Usuario.

Las efemérides GLONASS están referidas al Datum Geodésico Parametry Zemli 1990 o PZ-90, o en su traducción Parámetros de la Tierra 1990 o PE-90. Este sistema reemplazó al SGS-85, usado por GLONASS hasta 1993. El sistema PZ-90 es un sistema de referencia terrestre con coordenadas definidas de la misma forma que el Sistema de Referencia Internacional Terrestre (ITRF).

Los Sistemas NAVSTAR-GPS y GLONASS son sistemas autónomos, es decir, cada uno tiene su propio sistema de referencia y su propio sistema o escala de tiempo. Usan diferentes sistemas de referencia para expresar las posiciones de sus satélites, y por lo tanto, para determinar las posiciones de los usuarios.

Para poder utilizar los dos Sistemas de Posicionamiento por Satélite, NAVSTAR-GPS y GLONASS, es decir, recibir señales de los satélites de la constelación NAVSTAR-GPS y de la constelación GLONASS, es necesario establecer la relación entre los sistemas de tiempo y sistemas de referencia utilizados en los dos sistemas. En resumen, el Sistema GPS utiliza el sistema de referencia WGS-84, mientras que el Sistema GLONASS utiliza el PZ-90.

I.II. Factores Generadores de Error

Antes de analizar la exactitud y el grado de errores que podemos encontrar en nuestro receptor GPS, introduciremos un concepto el cual es vital en el cálculo de la posición y es el de trilateracion erróneamente llamado triangulación

I.II.I. Trilateracion

Para que un receptor obtenga la posición, se requiere la detección o vista de por lo menos 4 satélites, lo cual da lugar a la trilateracion 3D. Con éstas cuatro señales el receptor es capaz de determinar la longitud (x), la latitud(y), la altitud (y) y el tiempo (t) con lo que en un entorno ideal sin errores, el receptor GPS se situará en la intersección de las 4 esferas definidas por las señales de los satélites. El uso de únicamente 3 satélites deriva en el posicionamiento en 2D (exento de z, altitud).

Es de vital importancia el llamado cuarto satélite puesto que éste es que determinará el valor de t. Esto es debido a que se determina la distancia del receptor al satélite en base a la velocidad de la luz más una constante de rectificación de errores y la sincronización de todos los satélites mediante el reloj atómico.

Por lo tanto el trabajo de un receptor GPS es localizar a cuatro o más de estos satélites, calcular la distancia a cada uno, y utilizar esta información para deducir su propia ubicación. Esta operación se basa en un principio matemático simple llamado trilateración. Trilateración en el espacio tridimensional puede ser un poco complicado, así que comenzaremos con una explicación de trilateración simple de dos dimensiones y lo extrapolamos a tres dimensiones

Imaginémonos que estamos perdidos en cualquier parte de México, y al preguntarle a una persona A por la ubicación, nos dice que la desconoce pero nos comenta que estamos a 171.6 Kilómetros de Monclova Coahuila. Por lo tanto si observamos la ilustración, podemos ver que el circulo de la imagen 1.2. representa las posibles posiciones de nuestra ubicación, con centro en Monclova Coahuila y radio de 171.6 KM.

Imagen 1.2. Radio del Punto A.
(Fuente: Mapa obtenido de Google Earth)

Volvemos a preguntarle a otra persona B, y esta nos comenta de la misma manera que desconoce la ubicación pero que estamos a 250.5 Km de Ciudad Victoria Tamaulipas. Con estas dos informaciones, podemos nuevamente trazar otro circulo con las posibles ubicaciones como lo muestra la imagen 1.3., y la zona de intersección, nos dejan solo dos puntos posibles de nuestra ubicación (1,2), ya que en estos puntos, cumplimos con los dos criterios, es decir, nos encontramos a 171.6 KM de Monclova Coahuila y a 250.5 Km de Ciudad Victoria Tamaulipas.

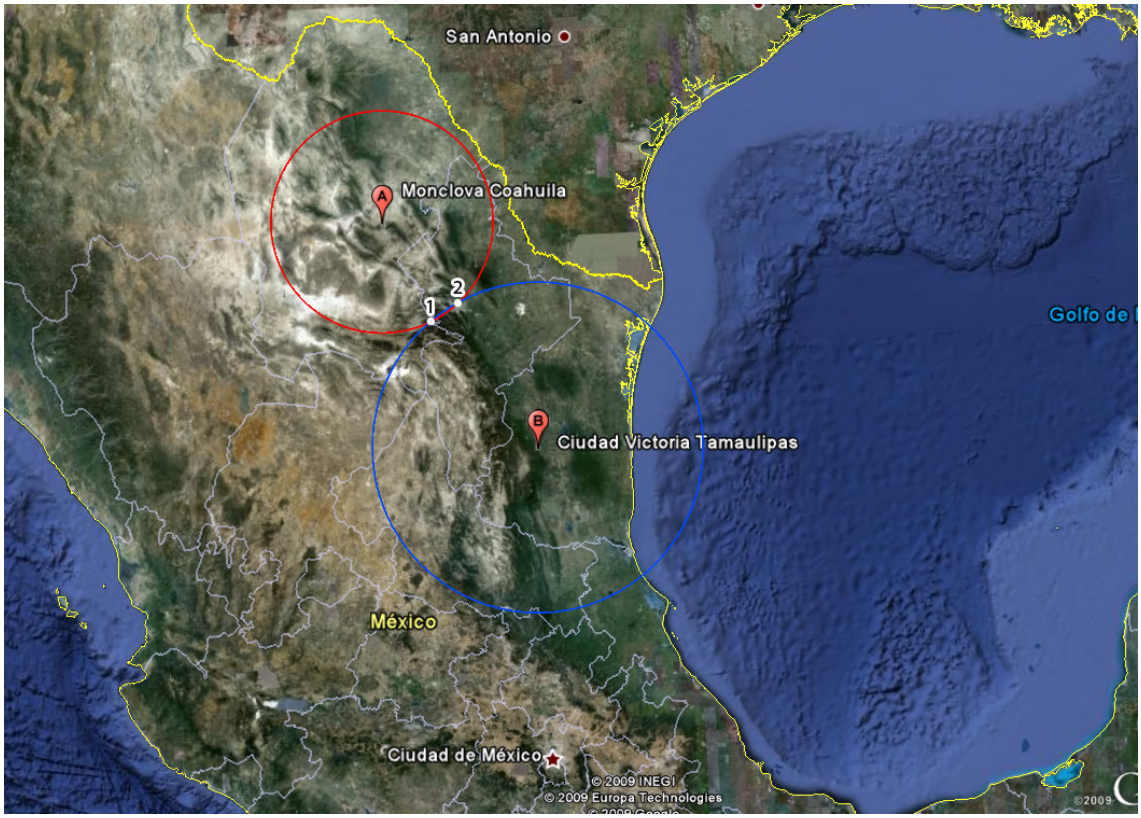


Imagen 1.3. Radio del Punto B.
(Fuente: Mapa obtenido de Google Earth)

Si una Persona C nos dice que estamos a 481.71 KM de la ciudad Victoria de Durango, eliminaremos uno de los puntos (1) en cual es una posición errónea y nos ubicaremos en la intersección de los tres círculos, como lo muestra la imagen 1.4. Por lo tanto ya sabemos la ubicación exacta de donde nos encontramos y es la Ciudad de San Nicolás de los Garza en Monterrey.

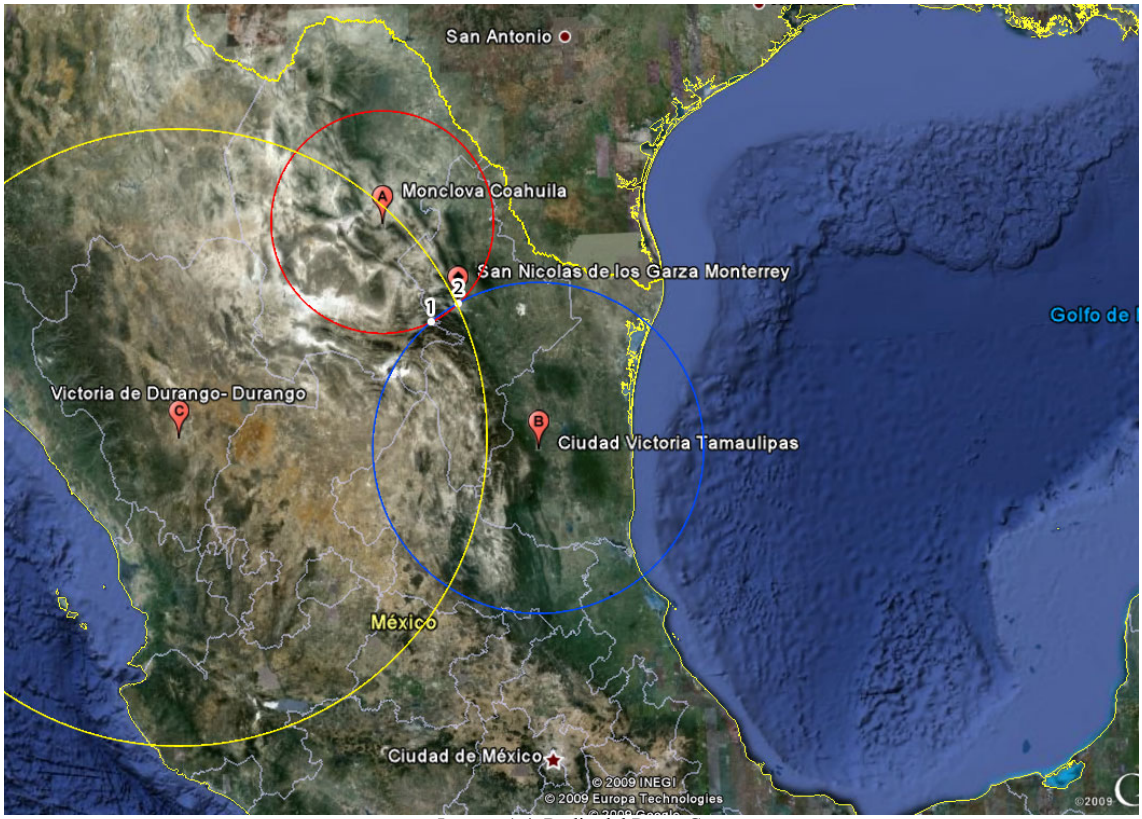


Imagen 1.4. Radio del Punto C.
(Fuente: Mapa obtenido de Google Earth)

Este mismo concepto funciona en el espacio tridimensional, pero en vez de ser círculos, estamos hablando de esferas, y se necesitarían de por lo menos 4 satélites de referencia porque tendremos una variable más que es la Z, la que nos daría el volumen.

I.II.II. Midiendo las distancias a los satélites

Sabemos ahora que nuestra posición se calcula a partir de la medición de la distancia hasta por lo menos tres satélites. Pero, ¿cómo podemos medir la distancia hacia algo que está flotando en algún lugar en el espacio?. Lo hacemos midiendo el tiempo que tarda una señal emitida por el satélite hasta nuestro receptor de GPS, utilizando la siguiente formula:

Velocidad x Tiempo = Distancia

En el caso del GPS estamos midiendo una señal de radio, que sabemos que viaja a la velocidad de la luz, alrededor de 300,000 Km. por segundo (velocidad teórica en el vació).

Por lo tanto nos queda solamente el problema de medir el tiempo de viaje de la señal. El problema de la medición de este tiempo en particular es complicado. Los tiempos son extremadamente cortos. Si el satélite estuviera justo sobre nuestras cabezas, a unos 20.000 Km. de altura, el tiempo total de viaje de la señal hacia nosotros sería aproximadamente de 0.06 segundos. Entonces necesitaríamos relojes muy precisos. Pero, aún admitiendo que tenemos relojes con la suficiente precisión, ¿cómo medimos el tiempo de viaje de la señal?

Supongamos que nuestro GPS, por un lado, y el satélite, por otro, generan una señal auditiva en el mismo instante exacto. Supongamos también que nosotros, parados al lado de nuestro receptor de GPS, podamos oír ambas señales (Obviamente es imposible "oír" esas señales porque el sonido no se propaga en el vacío).

Oiríamos dos versiones de la señal. Una de ellas inmediatamente, la generada por nuestro receptor GPS y la otra con cierto atraso, la proveniente del satélite, porque tuvo que recorrer alrededor de 20.000 Km. para llegar hasta nosotros. Podemos decir que ambas señales no están sincronizadas.

Si quisiéramos saber cual es la magnitud de la demora de la señal proveniente del satélite podemos retardar la emisión de la señal de nuestro GPS hasta lograr la perfecta sincronización con la señal que viene del satélite.

El tiempo de retardo necesario para sincronizar ambas señales es igual al tiempo de viaje de la señal proveniente del satélite. Supongamos que sea de 0.06 segundos. Conociendo este tiempo, lo multiplicamos por la velocidad de la luz y ya obtenemos la distancia hasta el satélite.

Tiempo de retardo (0.06 seg.) x Vel. de la luz (300.000 Km./seg.) = Dist. (18.000 Km.)

Así es, básicamente, como funciona el GPS. La señal emitida por nuestro GPS y por el satélite es algo llamado "Código Seudo Aleatorio" (Pseudo Random Code). La palabra "Aleatorio" significa algo generado por el azar. Este Código Pseudo Aleatorio es una parte fundamental del GPS. Físicamente solo se trata de una secuencia o código digital muy complicado. O sea una señal que contiene una sucesión muy complicada de pulsos "1" y "0". La señal es tan complicada que casi parece un ruido eléctrico generado por el azar. De allí su denominación de "Pseudo-Aleatorio".

Hay varias y muy buenas razones para tal complejidad. La complejidad del código ayuda a asegurarnos que el receptor de GPS no se sintonice accidentalmente con alguna otra señal. Siendo el modelo tan complejo es altamente improbable que una señal cualquiera pueda tener exactamente la misma secuencia. Dado que cada uno de los satélites tiene su propio y único Código Pseudo Aleatorio, esta complejidad también garantiza que el receptor no se confunda accidentalmente de satélite. De esa manera, también es posible que todos los satélites trasmitan en la misma frecuencia sin interferirse mutuamente. Esto también complica a cualquiera que intente interferir el sistema desde el exterior al mismo. El Código Pseudo Aleatorio le da la posibilidad al Departamento de Defensa de EEUU de controlar el acceso al sistema GPS.

Pero hay otra razón para la complejidad del Código Pseudo Aleatorio, una razón que es crucial para conseguir un sistema GPS económico. El código permite el uso de la "teoría

de la información" para amplificar las señales de GPS. Por esa razón las débiles señales emitidas por los satélites pueden ser captadas por los receptores de GPS sin el uso de grandes antenas. Cuando comenzamos a explicar el mecanismo de emisión de las señales por el GPS y el satélite, asumimos que ambos comenzaban la emisión de la señal exactamente al mismo tiempo. ¿Pero cómo podemos asegurarnos que todo esté perfectamente sincronizado?

Si la medición del tiempo de viaje de una señal de radio es clave para el GPS, los relojes que empleamos deben ser exactísimos, dado que si miden con un desvío de un milésimo de segundo, a la velocidad de la luz, ello se traduce en un error de 300 km. Por el lado de los satélites, el timing es casi perfecto porque llevan a bordo relojes atómicos de increíble precisión. ¿Pero que pasa con nuestros receptores GPS, aquí en la tierra?. Recordemos que ambos, el satélite y el receptor GPS, deben ser capaces de sincronizar sus Códigos Pseudo Aleatorios para que el sistema funcione.

Si nuestros receptores GPS tuvieran que alojar relojes atómicos (Cuyo costo está por encima de los 50 a 100.000 US) la tecnología resultaría demasiado costosa y nadie podría acceder a ellos. Por suerte los diseñadores del sistema GPS encontraron una brillante solución que nos permite resolver el problema con relojes mucho menos precisos en nuestros GPS. Esta solución es uno de los elementos clave del sistema GPS y como beneficio adicional, significa que cada receptor de GPS es en esencia un reloj atómico por su precisión.

El secreto para obtener un timing tan perfecto es efectuar una medición satelital adicional. Resulta que si tres mediciones perfectas pueden posicionar un punto en un espacio tridimensional, cuatro mediciones imperfectas pueden lograr lo mismo. Esta idea es fundamental para el funcionamiento del sistema GPS, pero su explicación detallada excede los alcances de la presente exposición. En un intento por explicar esta medición diremos lo siguiente:

Una medición adicional remedia el desfase del timing. Si todo fuera perfecto (es decir que los relojes de nuestros receptores GPS lo fueran), entonces todos los rangos o distancias a los satélites se intersectarían en un único punto (que indica nuestra posición). Pero con relojes imperfectos, una cuarta medición efectuada como control cruzado, NO intersectará con los tres primeros.

De esa manera la computadora de nuestro GPS detectará la discrepancia y atribuirá la diferencia a una sincronización imperfecta con la hora universal. Dado que cualquier discrepancia con la hora universal afectará a las cuatro mediciones, el receptor buscará un factor de corrección único que siendo aplicado a sus mediciones de tiempo hará que los rangos coincidan en un solo punto.

Dicha corrección permitirá al reloj del receptor ajustarse nuevamente a la hora universal y de esa manera tenemos un reloj atómico en la palma de nuestra mano.

Una vez que el receptor de GPS aplica dicha corrección al resto de sus mediciones, obtenemos un posicionamiento preciso. Una consecuencia de este principio es que cualquier GPS básico debe ser capaz de sintonizar al menos cuatro satélites de manera simultánea. En la práctica, como lo hemos visto, casi todos los GPS en venta actualmente, acceden a más de 6, y hasta a 12, satélites simultáneamente.

Ahora bien, con el Código Pseudo Aleatorio como un pulso confiable para asegurar la medición correcta del tiempo de la señal y la medición adicional como elemento de sincronización con la hora universal, tenemos todo lo necesario para medir nuestra distancia a un satélite en el espacio. Pero, para que la Trilateracion funcione necesitamos conocer no sólo la distancia sino que debemos conocer dónde están los satélites con toda exactitud.

Un satélite a gran altura se mantiene estable, la altura de 20,000 Km. es en realidad un gran beneficio para este caso, porque algo que está a esa altura está bien despejado de la atmósfera. Eso significa que orbitará de manera regular y predecible mediante ecuaciones matemáticas sencillas. La Fuerza Aérea de los EEUU colocó cada satélite de GPS en una órbita muy precisa, de acuerdo al Plan Maestro de GPS. En tierra, todos los receptores de GPS tienen un almanaque programado en sus circuitos que les informan donde está cada satélite en el espacio, en cada momento.

Las órbitas básicas son muy exactas pero con el fin de mantenerlas así, los satélites de GPS son monitoreados de manera constante por el Departamento de Defensa. Ellos utilizan radares muy precisos para controlar constantemente la exacta altura, posición y velocidad de cada satélite. Los errores que ellos controlan son los llamados errores de efemérides, o sea evolución orbital de los satélites. Estos errores se generan por influencias gravitacionales del sol y de la luna y por la presión de la radiación solar sobre los satélites.

Una vez que el Departamento de Defensa ha medido la posición exacta de un satélite, vuelven a enviar dicha información al propio satélite. De esa manera el satélite incluye su nueva posición corregida en la información que transmite a través de sus señales a los GPS. Esto significa que la señal que recibe un receptor de GPS no es solamente un Código Pseudo Aleatorio con fines de timing. También contiene un mensaje de navegación con información sobre la órbita exacta del satélite

Con un timing perfecto y la posición exacta del satélite podríamos pensar que estamos en condiciones de efectuar cálculos perfectos de posicionamiento. Sin embargo debemos resolver otros problemas.

Al igual que cualquier observación de topografía clásica, una observación GPS o GLONASS está sometida a varias fuentes de error que se pueden minimizar o modelar según los equipos y metodología de observación que utilicemos. Un receptor determina las distancias que hay entre su antena y las antenas de los satélites desde los cuales está recibiendo su señal. Basándose en estas distancias y en el conocimiento de las posiciones de los satélites, el receptor puede calcular su posición. Sin embargo, diversos errores afectan a la medida de la distancia y por consiguiente se propagan al cálculo de la posición del receptor.

Las medidas de código y las medidas de fase se ven afectadas por errores sistemáticos y por ruido aleatorio. La precisión en posicionamiento absoluto que un usuario puede alcanzar con un receptor depende principalmente de cómo sus sistemas de hardware y software puedan tener en cuenta los diversos errores que afectan a la medición. Estos errores pueden ser clasificados en tres grupos: los errores relativos al satélite, los errores relativos a la propagación de la señal en el medio, y los errores relativos al receptor.

Elemento	Fuente
Satélite	Errores o variaciones en los parámetros orbitales Errores en el Oscilador
Propagación de la Señal	Refracciones Ionosféricas y troposféricas Disponibilidad Selectiva Multipath Ondas reflejadas Perdidas de ciclos
Receptor	Error en las coordenadas del punto de referencia Errores en el Oscilador Variación y desfase en el centro de antena Manipulación e interpretación en el equipo

Tabla 1.1. Clasificación de errores

Algunos de estos errores sistemáticos de la tabla 1.1. son modelados e incluso eliminados utilizando combinaciones apropiadas de los observables a partir de una o dos frecuencias, o trabajando en modo diferencial, utilizando dos receptores.

En la medida de la calidad y bondad de una observación van a influir o contribuir dos términos: el URE y el DOP. El URE (User Range Error) es el error cometido en la medida de la seudodistancia por el usuario. Este error contempla los errores al predecir las efemérides, inestabilidades en el vehículo espacial, relojes de los satélites, efectos ionosféricos y troposféricos, efecto multipath, ruido de la señal y para GPS, la Disponibilidad Selectiva (SA). Todos estos errores en su conjunto se recogen en el valor URE . El DOP o Dilución de la Precisión es la contribución puramente geométrica al error en el posicionamiento de un punto. Es un valor adimensional que da una idea de la solidez de la figura formada por el receptor y los satélites que tiene a la vista.

I.II.III. Errores relativos al Satélite.

I.II.III.I. Error del reloj del satélite.

Este error es el desfase que tiene el reloj del satélite respeto al Tiempo GPS o respecto al Tiempo GLONASS. Los satélites llevan relojes atómicos con osciladores de cesio o de rubidio, sin embargo ningún reloj, incluso el atómico es perfecto.

Los errores en los osciladores de los satélites pueden eliminarse mediante las correcciones enviadas en el mensaje de navegación que recibe el receptor, y que son calculadas y actualizadas por las estaciones de seguimiento. Para cada reloj de satélite se determina su desfase para una época inicial, y los coeficientes de la marcha o deriva del estado del reloj. Estos parámetros se graban en el correspondiente satélite y se incluyen en el mensaje de navegación que manda el satélite. Pero aunque el receptor aplique las correcciones para el error del reloj del satélite, sigue permaneciendo un pequeño error residual estimado en unos 10 nanosegundos o menos, y que es debido a la imposibilidad de predecir exactamente la marcha del estado del reloj del satélite.

I.II.III.II. Error en los parámetros orbitales.

Para calcular su posición, el receptor debe conocer las posiciones de los satélites. Las estaciones de seguimiento registran datos de seudodistancia y medidas de fase que mandan a la Estación de Control principal, donde con un sofisticado software se predicen las futuras posiciones orbitales de los satélites, es decir sus efemérides. Éstas son transmitidas en el mensaje de navegación del satélite. Pero las efemérides transmitidas por los satélites tendrán asociado un error a causa de que es imposible predecir exactamente sus posiciones. El efecto del error de las efemérides transmitidas en la medida de la seudodistancia se obtiene proyectando el vector error de la posición del satélite sobre el vector que une el satélite y el receptor. Los errores en los parámetros orbitales se pueden eliminar trabajando con las efemérides precisas de los días de observación, donde aparecen las verdaderas posiciones de los satélites.

Para líneas base cortas, trabajando en modo diferencial con dos receptores, respecto a los mismos satélites de observación, podemos eliminar todos los errores relativos a los satélites, ya que afectan de igual forma a ambos receptores. Para líneas base largas, el error del reloj del satélite se elimina igual, ya que es independiente de la línea base e igual en ambos puntos, pero los errores en los parámetros orbitales no se eliminan del todo, porque los errores que provocan en la seudodistancia a un satélite en un punto no son los mismos que los que se producen en el otro punto para el mismo satélite e instante. El error depende de la orientación del vector error de la posición del satélite respecto de los vectores satélite-receptor para cada uno de los puntos.

I.II.IV. Errores relativos a la propagación de la señal.

La velocidad de propagación de la señal es crítica para cualquier sistema de medida de distancias. Esta velocidad multiplicada por el intervalo de tiempo en que se propagó la señal nos da una medida de la distancia. Si una onda electromagnética se propaga por el vacío, su velocidad de propagación, sea cual sea su frecuencia es la velocidad de la luz (c). Sin embargo, en el caso de observaciones GPS o GLONASS, las señales deben atravesar las capas de la atmósfera hasta llegar al receptor posicionado sobre la superficie de la tierra. Las señales interaccionan con partículas cargadas, que provocan un cambio en la velocidad y dirección de propagación, es decir, las señales son refractadas. Cuando la señal viaja por un medio que no es el vacío, ésta sufre un retardo debido a que la velocidad de propagación es menor, y a que la trayectoria aumenta su longitud al curvarse por refracción, si el medio no es isótropo.

I.II.IV.I. Refracción Ionosférica.

La Ionosfera es aquella región de la atmósfera comprendida entre 100 y 1000 Km de altitud, donde las radiaciones solares y otras radiaciones ionizan una porción de las moléculas gaseosas liberando electrones, que interfieren en la propagación de ondas de radio. La Ionosfera es un medio disperso para ondas de radio, por lo tanto su índice de refracción es función de la frecuencia de la onda. También es función de la densidad de electrones, y en menor grado, de la intensidad del campo magnético de la tierra.

Este error es negativo para la medida de fase (se produce un avance de la portadora y se miden distancias más pequeñas), y positivo para las seudodistancias (se produce un retardo y se miden distancias más largas), pero tienen el mismo valor absoluto.

El error es proporcional a la densidad de electrones (TEC-Total Electron Content) a lo largo del camino seguido por la señal, y está en función del cuadrado de la longitud de la onda (inversamente proporcional al cuadrado de la frecuencia de la portadora). Este error varía espacial y temporalmente, es decir, para cada punto según su latitud y longitud, y momento de la observación. Se pueden utilizar modelos ionosféricos, como el de Klobuchar (1986) que establecen la distribución del TEC, pero estas concentraciones de electrones son irregulares y poco predecibles, por lo que cualquier modelo ionosférico es sólo una aproximación. El TEC es función del cambio constante en la ionización solar, de la actividad magnética, de los ciclos de las manchas solares, hora del día, lugar de observación, y dirección del camino de la señal.

Debido a la dificultad de encontrar un modelo satisfactorio, se emplea un método más eficiente para eliminar la refracción ionosférica que es la utilización de dos señales con diferentes frecuencias. Como el retardo depende de la longitud de la onda, será distinto para cada frecuencia y podremos observar un retardo diferencial entre ambas, que será mayor cuanto mayor sea el retardo ionosférico sufrido, siendo por tanto este deducible.

También se pueden utilizar combinaciones de las observables que por su naturaleza estén libres del efecto ionosférico. Tal es el caso de la combinación de fases llamada "combinación libre de efecto ionosférico"

La eliminación de la refracción ionosférica es la mayor ventaja de la combinación lineal libre de efecto ionosférico, pero el término libre de efecto ionosférico no es del todo correcto, ya que para su obtención hay que considerar algunas aproximaciones.

Si sólo se registran medidas en una sola frecuencia, tanto en seudodistancias como en medida de fase, entonces se tiene que emplear un procedimiento alternativo para eliminar el efecto ionosférico. Normalmente se usan modelos empíricos para corregir el efecto, en los que se modela el TEC en función del tiempo, lugar de observación y dirección de la señal. En el mensaje de navegación se incluyen unos parámetros para tal modelo. Usando este modelo se pueden llegar a reducir en un 50% los efectos de la Ionosfera.

Actualmente, estamos saliendo de un mínimo en la actividad de las manchas solares (11 años de ciclo), por lo que las condiciones ionosféricas son ahora más idóneas. Pero dentro de unos 4 años, estaremos cerca del máximo, y entonces los efectos de la Ionosfera en las señales serán mucho peores.

El retardo ionosférico depende del ángulo de elevación del satélite, siendo menor en el cenit, y mayor cuando disminuye el ángulo de elevación. En observaciones nocturnas, los niveles de TEC son menores que durante el día, lo que implica un menor error en la seudodistancia.

Pero después de la aplicación del modelo empírico transmitido puede quedar algún error ionosférico residual que afectará principalmente a la componente altimétrica del punto y a la estimación del error del reloj del receptor. Este error contribuye poco a la posición planimétrica cuando la concentración de electrones encima del receptor es uniforme.

I.II.IV.II. Refracción Troposférica.

La Troposfera es la última zona o capa de la atmósfera (hasta unos 80 Km, pero sólo en los últimos 40 se producen retardos significativos), donde se produce retardo y donde las temperaturas decrecen con el incremento de altura. El espesor de la Troposfera no es el mismo en todas las zonas. La presencia de átomos y moléculas neutros en la Troposfera afecta a las señales de propagación electromagnética. El índice de refracción para un área parcial es función de su temperatura, de la presión de los gases secos y del vapor de agua. Esta atmósfera neutra es un medio no disperso con respecto a las ondas de radio de frecuencias superiores a 15 GHz, por lo tanto, la propagación es independiente de la frecuencia. Consecuentemente, no es necesario distinguir entre medidas de código y fase sobre las portadoras L1 y L2. La desventaja está en que no es posible eliminar la refracción troposférica con medidas en las dos frecuencias.

La integral puede ser evaluada conociendo el índice de refracción, o puede ser aproximada por funciones analíticas. Pero lo más normal es utilizar aproximaciones basadas en modelos atmosféricos simplificados. Algunos de estos modelos son: el modelo de Hopfield (1969), modelo de Saastamoinen (1972), modelo de Hopfield modificado, Goad y Goodman (1974), Black (1978), Robinson (1986), etc. En la mayoría de los casos, se considera por separado la componente seca y la componente húmeda :

El efecto del retardo ionosférico y el troposférico debido al vapor de agua sobre las emisiones de la banda radioeléctrica es menor cuanto mayor sea la frecuencia, o cuanto menor sea la longitud de la onda como la imagen 1.5. La refracción ionosférica y troposférica puede ser eliminada trabajando en modo diferencial, pero esto es sólo cierto para líneas base pequeñas, donde las medidas de distancias satélite-receptor se ven afectadas de igual forma por la refracción. De otro modo, ya vimos que la refracción ionosférica puede ser eliminada utilizando una adecuada combinación de datos en doble frecuencia.

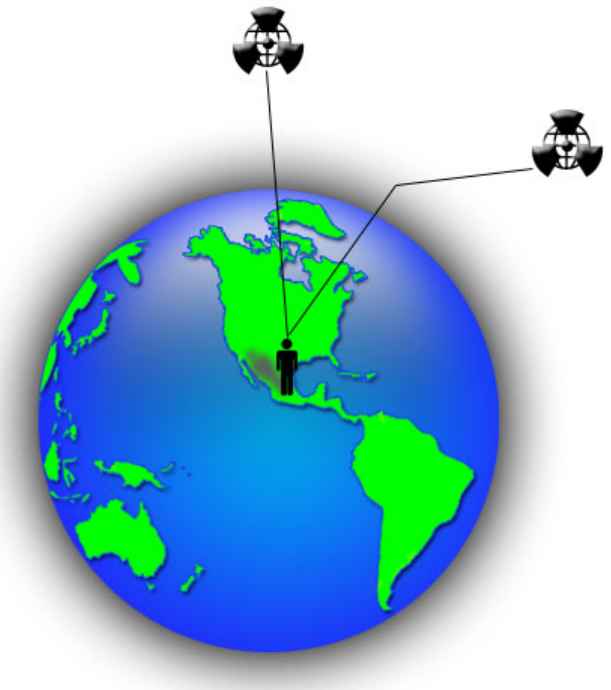


Imagen 1.5. Refracción Troposférica

I.II.IV.III. Disponibilidad Selectiva.

La Disponibilidad Selectiva supone una alteración o manipulación de la información que los satélites de la constelación GPS envían a los usuarios en su mensaje de navegación, manipulación que realiza el Departamento de Defensa de los Estados Unidos (DoD). Se actúa sobre los estados de los relojes y parámetros orbitales. Trabajando con posicionamiento relativo o diferencial se puede eliminar este error.

Se dice que el gobierno de Clinton en la guerra del golfo, altero dicha información, para que el enemigo tuviera datos erróneos de la posición de las tropas enemigas.

I.II.IV.IV. Pérdidas de Ciclos.

Las pérdidas de ciclos suponen un salto en el registro de las medidas de fase, producido por alguna interrupción o pérdida de la señal enviada por el satélite. Estas pérdidas de ciclos pueden ser causadas por la obstrucción de la señal del satélite debido a la presencia de árboles, edificios, puentes, montañas, etc. Esta causa es la más frecuente, pero también pueden ser debidas a una baja SNR (calidad señal-ruido) debido a unas malas condiciones ionosféricas, efecto multipath, receptores en movimiento, o baja elevación del satélite. Otra causa puede ser un fallo en el software del receptor, que conduce a un procesamiento incorrecto de la señal. Una última causa de pérdida de ciclo, aunque suele darse en raras ocasiones, es aquella debida a un mal funcionamiento del oscilador del satélite.

La detección de una pérdida de ciclo y su reparación requiere la localización del salto y determinación de su tamaño. La detección se lleva a cabo por medio de un chequeo o test de cantidad, estos test pueden ser medida de la fase en bruto, combinaciones de fase, combinaciones de código y fase, etc. Una vez determinado el tamaño de la pérdida de ciclo, la reparación se hace corrigiendo a todas las observaciones de fase siguientes para este satélite y su portadora, según una cantidad fija. El software interno del receptor es capaz (in situ) de detectar y corregir las pérdidas de ciclo.

I.II.IV.V. Efecto Multipath.

El efecto multipath o multicamino es causado principalmente por múltiples reflexiones de la señal emitida por el satélite en superficies cercanas al receptor. Estas señales reflejadas que se superponen a la señal directa son siempre más largas, ya que tienen un tiempo de propagación más largo y pueden distorsionar significativamente la amplitud y forma de la onda. Este efecto puede ser considerablemente reducido eligiendo puntos de estación protegidos de reflexiones (edificios, vehículos, árboles, etc.), es decir, evitar las superficies reflectantes en las proximidades del receptor y un apropiado diseño de la antena, también la utilización de planos de tierra, que reducen las interferencias de señales con baja elevación o incluso con elevación negativa, que son las que provocan el multipath (ver imagen 1.6.). En otras palabras, se intenta reducir la intensidad de las señales secundarias y aislar a la señal directa. El efecto multipath depende de la frecuencia de la portadora. Por lo tanto, las medidas de fase se verán menos afectadas que las medidas de código, donde el efecto multipath puede alcanzar hasta el nivel de metro.

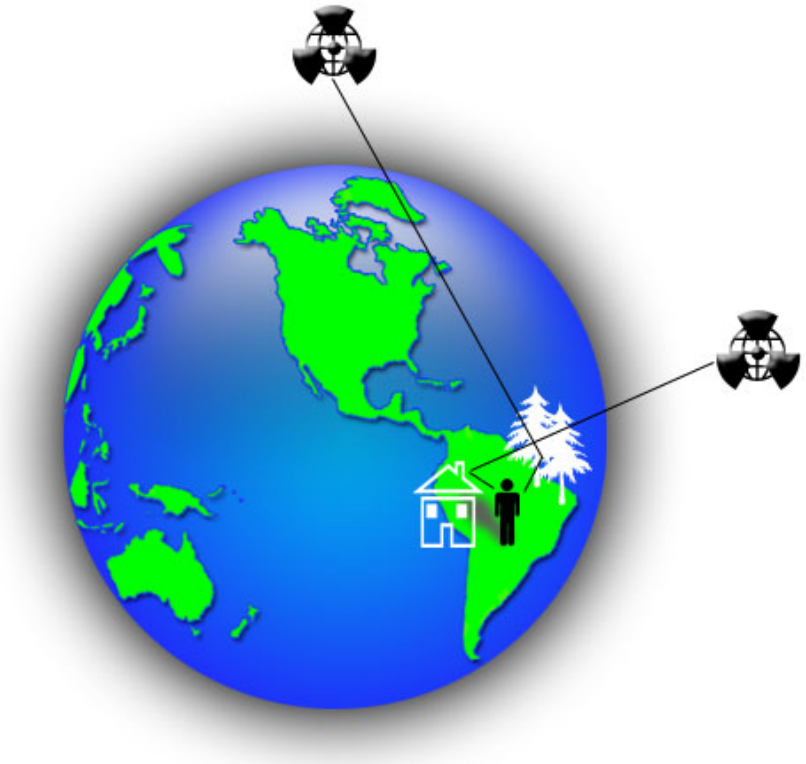


Imagen 1. 6. Efecto Multipath

I.II.V. Errores relativos al receptor.

Cuando un receptor recibe una señal de un satélite, en ese momento su reloj interno tendrá un desfase o error con respecto a la Escala de Tiempo. Este error afectará a todas las medidas de seudodistancias realizadas para cada época.

Los errores en los osciladores de los receptores los podemos eliminar trabajando con posicionamiento relativo por medidas de fase, planteando las ecuaciones de dobles diferencias.

Asimismo, problemas con la antena debido a mal uso del dispositivo o fallas en el software

I.II.VI. Dilución de la Precisión (DOP).

La geometría de los satélites visibles es un factor importante a la hora de conseguir altas precisiones en el posicionamiento de un punto. Dicha geometría cambia con el tiempo como consecuencia del movimiento orbital de los satélites. Un factor que mide la bondad de esta geometría es el denominado factor de dilución de la precisión (dilution of precision , DOP).

El valor del DOP puede ser interpretado geométricamente como el volumen del cuerpo formado por los satélites y el receptor. Cuanto mayor sea el volumen de este cuerpo

mejor será la geometría, y por lo tanto menor será el valor del DOP, siendo el valor ideal la unidad. Como ya se vio anteriormente, el valor del DOP es el factor por el que debe ser multiplicado el error obtenido en las seudodistancias para obtener el error final en el posicionamiento. Los valores de DOP más utilizados son los siguientes:

- GDOP: Dilución de precisión en posición y estado del reloj.
- PDOP: Dilución de precisión en posición.
- TDOP: Dilución de precisión en el estado del reloj.
- HDOP: Dilución de precisión en planimetría.
- VDOP: Dilución de precisión en altimetría.
- RDOP: Dilución de precisión relativa entre dos puntos.

Tan importante es el posicionamiento de satélites como la geometría que describen entre ellos y emiten sus señales, me explico. En la siguiente imagen 1.7., podemos observar como cuatro satélites emiten señales válidas a un receptor GPS.

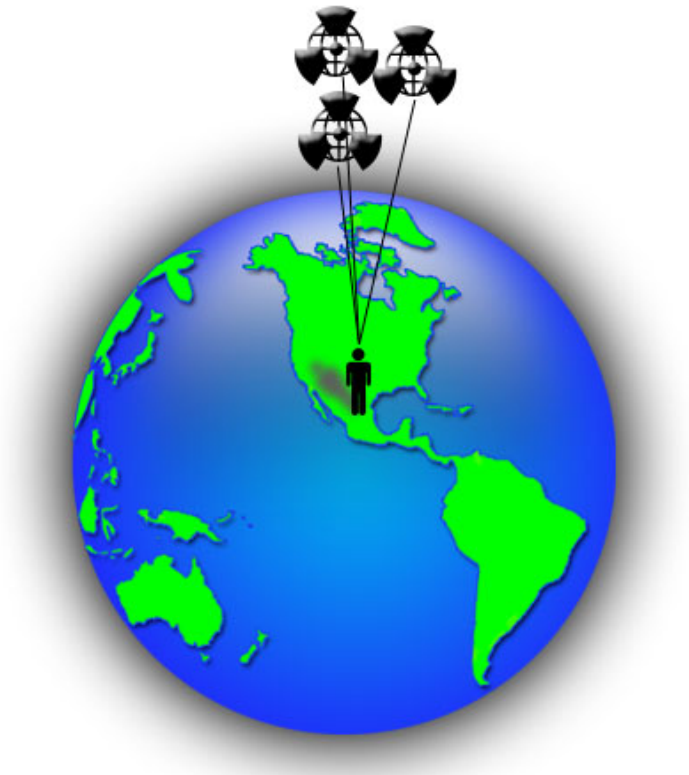


Imagen 1. 7. Geometría A de los Satélites

Sin embargo la geometría de sus señales no es óptima debido a la proximidad que existe entre ellos. En la imagen 1.8., ocurre todo lo contrario.

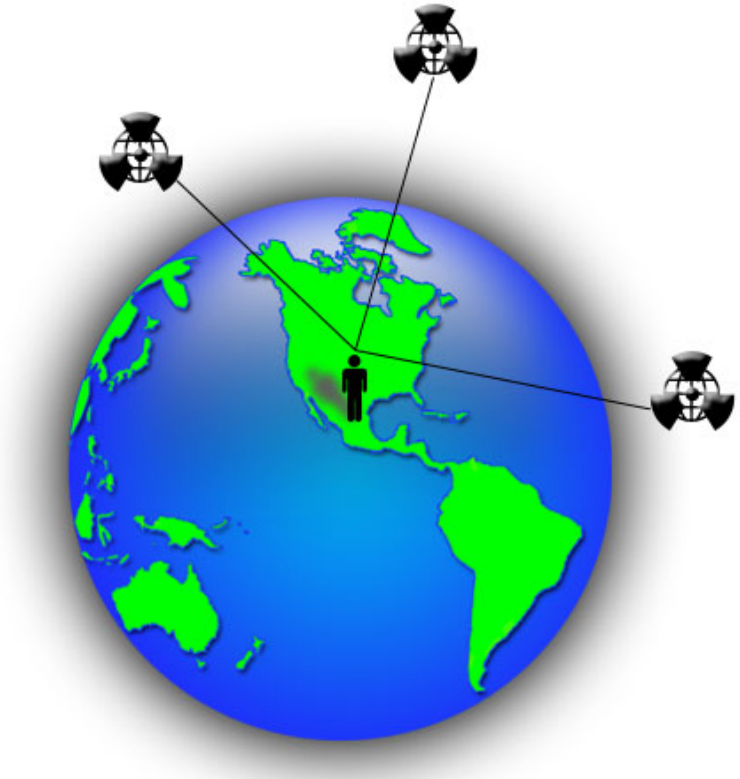


Imagen 1. 8. Geometría B de los Satélites

Las señales provenientes de los satélites geométricamente mejor posicionados generarán una mayor fiabilidad del posicionamiento. Así pues:

La dilución de precisión (DOP) se emplea en cartografía y describe la precisión del GPS en base a la geometría de los satélites. Cuando la señal DOP es alta, los satélites están muy cerca entre sí con lo que la precisión disminuye y el valor DOP aumenta. Si por el contrario los satélites son distantes, la precisión aumenta y con ello el valor DOP disminuye. Existen diluciones de precisión para el posicionamiento horizontal (HDOP), vertical (VDOP), el de posicionamiento (PDOP) y el de tiempo (TDOP) con lo que dentro del cada uno de dichos aspectos puede ofrecer un valor distinto. Los obstáculos urbanos y naturales puede aumentar el DOP de la señal del GPS.

La corrección de estos errores se llevan a cabo mediante complejas formulas fisicomatemáticas específicas para cada fuente de error. La mayoría de estos errores son transparentes para nosotros y ni siquiera debemos tenerlos en cuenta, en la mayoría de circunstancias. Muchos de estas correcciones podríamos obtenerlas mediante las sentencias NMEA recibidas (pero nunca intervenir)

I.II.VII. GPS Diferencial

DGPS. Aunque su traducción es “GPS diferencial”, se utiliza esta terminología para trabajos diferenciales en los que solamente intervienen medidas de código (observables de tiempo). Cierto es, que con las actuales técnicas de posicionamiento conjunto GPS/GLONASS este término no es correcto, pero se sigue utilizando.

El DGPS (Differential GPS), o GPS diferencial, es un sistema que proporciona a los receptores de GPS correcciones de los datos recibidos de los satélites GPS, con el fin de proporcionar una mayor precisión en la posición calculada. Se concibió fundamentalmente debido la introducción de la disponibilidad selectiva (SA).

El fundamento radica en el hecho de que los errores producidos por el sistema GPS afectan por igual (o de forma muy similar) a los receptores situados próximos entre sí. Los errores están fuertemente correlacionados en los receptores próximos.

Un receptor GPS fijo en tierra (referencia) que conoce exactamente su posición basándose en otras técnicas, recibe la posición dada por el sistema GPS, y calcula los errores producidos por el sistema GPS, comparándola con la suya, conocida de antemano como lo muestra la imagen 1.9. Este receptor transmite la corrección de errores a los receptores próximos a él, y así estos pueden, a su vez, corregir también los errores producidos por el sistema dentro del área de cobertura de transmisión de señales del equipo GPS de referencia. La estructura DGPS quedaría de la siguiente manera:

- Estación monitorizada (referencia), que conoce su posición con una precisión muy alta. Esta estación está compuesta por:
  - Un receptor GPS
  - Un microprocesador, para calcular los errores del sistema GPS y para generar la estructura del mensaje que se envía a los receptores.
  - Transmisor, para establecer un enlace de datos unidireccional hacia los receptores de los usuarios finales.
- Equipo de usuario, compuesto por un receptor DGPS (GPS + receptor del enlace de datos desde la estación monitorizada).

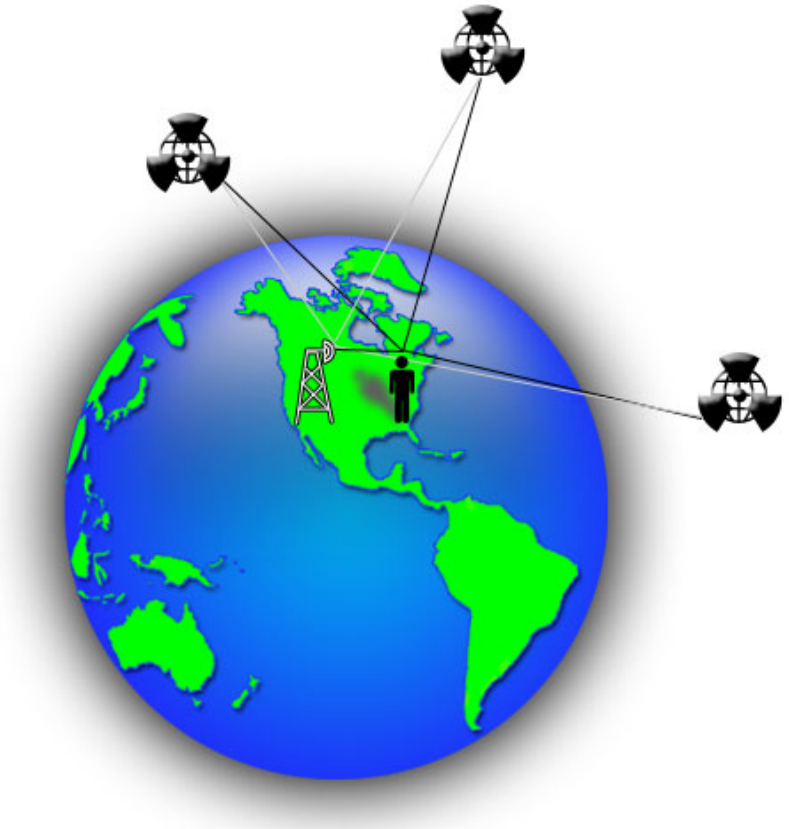


Imagen 1. 9. GPS Diferencial

Existen varias formas de obtener las correcciones DGPS. Las más usadas son:

- Recibidas por radio, a través de algún canal preparado para ello, como el RDS en una emisora de FM.
- Descargadas de Internet, o con una conexión inalámbrica.
- Proporcionadas por algún sistema de satélites diseñado para tal efecto. En Estados Unidos existe el WAAS, en Europa el EGNOS y en Japón el MSAS, todos compatibles entre sí.

En los mensajes que se envían a los receptores próximos se pueden incluir los siguientes tipos de correcciones,

- Una corrección directamente aplicada a la posición. Esto tiene el inconveniente de que tanto el usuario como la estación monitora deberán emplear los mismos satélites, pues las correcciones se basan en esos mismos satélites.
- Una corrección aplicada a las pseudodistancias de cada uno de los satélites visibles. En este caso el usuario podrá hacer la corrección con los 4 satélites de mejor relación señal-ruido (S/N). Esta corrección es más flexible.

El error producido por la disponibilidad selectiva (SA) varía incluso más rápido que la velocidad de transmisión de los datos. Por ello, junto con el mensaje que se envía de correcciones, también se envía el tiempo de validez de las correcciones y sus tendencias. Por tanto, el receptor deberá hacer algún tipo de interpolación para corregir los errores producidos.

Si se deseara incrementar el área de cobertura de correcciones DGPS y, al mismo tiempo, minimizar el número de receptores de referencia fijos, será necesario modelar las variaciones espaciales y temporales de los errores. En tal caso estaríamos hablando del GPS diferencial de área amplia.

Con el DGPS se pueden corregir en parte los errores debidos a:

- Disponibilidad selectiva (eliminada a partir del año 2000).
- Propagación por la ionosfera - troposfera.
- Errores en la posición del satélite (efemérides).
- Errores producidos por problemas en el reloj del satélite.
- Para que las correcciones DGPS sean válidas, el receptor tiene que estar relativamente cerca de alguna estación DGPS; generalmente, a menos de 1.000 km. La precisión lograda puede ser de unos dos metros en latitud y longitud, y unos 3 m en altitud.

Existe la posibilidad de trabajar en DGPS con un único receptor, al que se le debe sumar una unidad de control y un transmisor/receptor de radiofrecuencia que emite los datos de observación a una estación central de referencia, que envía datos de posicionamiento en formato RTCM o RTCA a la estación móvil, obteniendo la posición en tiempo real.

I.III. Consideraciones Finales y Aplicaciones del GPS

I.III.I. Consideraciones Finales del GPS

El Sistema de Posicionamiento por Satélite, ya sea con GPS, GLONASS o GPS/GLONASS, es una herramienta imprescindible en la sociedad de nuestros días, y que los técnicos en todas las materias afectadas deben saber tratar, manipular y ejecutar correctamente, ya que supone, como hemos dicho, un adelanto en la calidad y rendimiento de los trabajos respecto a los métodos clásicos, que nunca se deben abandonar, pero que la evolución de otras técnicas obliga a ir dejando a un lado y recurrir a técnicas, no sólo más modernas, sino más fructíferas y que en un futuro cercano estarán en el idioma y rutina cotidiano de los profesionales de estos campos.

Se citan a continuación las ventajas que ofrece el posicionamiento por satélite en nuestro trabajo:

- No es necesaria la intervisibilidad entre estaciones, ya que el sistema de medida es indirecto entre ellas y directo a los satélites. Esto reduce el número de estacionamientos al poder salvar los obstáculos y reduce los errores accidentales y sistemáticos al no tener que realizar punterías ni tantos estacionamientos con intervisibilidad entre los puntos. En definitiva, se reduce el tiempo de observación y los errores que se producen en ella. Debemos añadir además que la observación nocturna es totalmente operativa.
- Al trabajar con ondas de radio, estas no sufren efectos significativos a causa de la niebla, lluvia, fríos y calores extremos, y otros tipos de incidencias.
- El rango de distancias que se pueden alcanzar es mucho mayor, al no ser medidas directas. El mejor de los distanciómetros no supera los 4-5 Km de distancia, además del error que introduce. Con el posicionamiento por satélite podemos medir bases desde unos pocos metros hasta centenas y miles de Km.
- Dado que no se dispone de sistemas ópticos, su fragilidad es menor y su mantenimiento y calibración no es requerido con la frecuencia que lo requieren los instrumentos ópticos. Los costes de mantenimiento por ello son menores.
- El servicio de las señales que ofrecen los sectores espaciales y de control es totalmente gratuito, lo que supone sólo desembolsos en instrumentación de observación, cálculo y gastos para I+D.
- La obtención de los resultados es rápida, máxime si sumamos la obtención de los mismos en tiempo real (RTK). Además, las observaciones y los resultados son interpretables y tienen comprobación.
- La variedad de métodos de posicionamiento hace que sean sistemas apropiados y aptos para cualquier tipo de trabajo.

Por otro lado, los inconvenientes más relevantes son :

- No puede ser utilizado en obras subterráneas y a cielo cerrado.
- Tiene dificultades de uso en zonas urbanas, cerradas, con altos edificios y zonas arboladas y boscosas, debido a las continuas pérdidas de la señal de los satélites. Este problema, no obstante, se está solucionando, y de forma satisfactoria, con el uso combinado de las constelaciones GPS y GLONASS para mantener siempre cinco o más satélites sobre el horizonte.

I.III.II. Aplicaciones del GPS

- Servicio de guardacostas

  El Servicio de Guardacostas de EE.UU. es el responsable de proporcionar todas las ayudas de navegación. El huracán BOB que azotó la costa este de EE.UU. en 1991 destrozó o desplazó un gran número de boyas. La situación era peligrosa, pues los barcos iban a puerto confiados en unas boyas que ya no existían o estaban cambiadas de sitio.

  El Servicio de Guardacostas equipó uno de sus barcos de mantenimiento de boyas con un receptor DGPS y reposicionaron las boyas de nuevo, en tan solo unos días. A lo largo de este año se espera esté implantado el sistema DGPS para toda la costa de EE.UU.

- Aviación

  Algunos experimentos realizados por la NASA y por las FAA de EE.UU. contribuyeron al aterrizaje de helicópteros y aviones de pasajeros mediante DGPS como único sistema guía, sin las radiobalizas tradicionales.

  En la actualidad los sistemas de aterrizaje con poca visibilidad son tan caros que sólo están disponibles en los mayores aeropuertos. El DGPS es tan barato que lo puede instalar cualquier aeropuerto. La mejora de seguridad de vuelo es tremenda. Como referencia se puede citar Canadá, donde el sistema GPS ha sustituido al habitual, conocido como Omega.

- Gestión de los recursos naturales

  La gestión del uso y protección de los bosques es una gran tarea. Su estudio topográfico es difícil, sin embargo hay que medir constantemente parcelas de árboles, ya sea por asunto de su conservación o por ventas a empresas madereras.

  El Servicio Forestal de EE.UU. ha sido uno de los pioneros del DGPS. Hacen medidas con GPS desde helicópteros. Otras aplicaciones son: topografía de galerías de minas, de superficies de pantanos y de zonas para pesca.

  Otro caso es el control de incendios en los bosques.

- Exploración costera

  Las empresas petrolíferas gastan enormes cantidades de dinero en la exploración del fondo de los océanos en busca de lugares idóneos para perforar. El problema, es que una vez el barco encuentra un lugar de perforación, su tripulación necesita llevar a ese punto los dispositivos de perforación, lo cual no es fácil llegar al mismo sitio, al no haber posibilidad de poner marcas de referencia, y apartarse unos metros significa muchos millones de gasto de más. Para solucionar este problema usan el GPS.

Otra utilidad es para mantener a los barcos en las rutas exactas. También se usan para el levantamiento topográfico de los puertos. En EE.UU. es obligatorio que los barcos petroleros lleven GPS por motivos de seguridad.

- Gestión transporte y flotas

  Con este sistema el controlador de una flota puede llevar la cuenta de cada vehículo, el resultado es una más estricta adhesión al horario y una mejor supervisión. A las empresas de transporte, flotas de servicios y servicios de seguridad pública les gusta saber la posición de sus vehículos incluso al extremo de conocer el nombre de la calle. La solución es DGPS. También se usa en los ferrocarriles

- Agricultura

  El GPS está abriendo una nueva era de "agricultura de precisión". Un agricultor puede analizar las condiciones del suelo en cada parcela, y compilar un mapa de las demandas de fertilizante. Este mapa se digitaliza y se registra en ordenador. La máquina que adiciona los productos químicos al terreno, va con un GPS y su posición se correlaciona con los datos previamente digitalizados, añadiendo en cada punto la cantidad exacta de fertilizante. Se beneficia el agricultor con menos gasto y el medio ambiente evitando un exceso de productos químicos. También se puede aplicar a la fumigación aérea.

- Seguridad

  Para los servicios de bomberos y policía el tiempo de respuesta es muy importante. Con DGPS se pueden guiar los vehículos con gran precisión. Los planos de rutas centralizadas ofrecen a los controladores un mejor conocimiento de la forma en que están desplegados sus efectivos.

  Otro punto importante y tema de esta tesis es la localización del dispositivo, el cual esta asociado a una persona que lo utiliza para fines de comunicación. Por lo tanto si conocemos la ubicación del dispositivo, conocemos la ubicación del usuario, ya sea para el caso de emergencias tales como robo, secuestro etc, o simplemente para compartir su ubicación a familiares, amigos etc.

- Servicio de información de tiempo mundial

  La señal del sistema GPS contiene también la información de tiempo proporcionado por el reloj atómico abordo del satélite y ya que todos los relojes de los satélites están sincronizados se puede usar esta información para sincronizar cualquier evento temporal a nivel mundial evitando que las diferencias horarias o los errores humanos interfieran en la ejecución de estos eventos a la hora y día indicados por la fuente.

I.IV. Historia de .NET y Windows Mobile

I.IV.I. Historia de .NET

Visual Studio es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones Web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C++, Visual C# y Visual J# utilizan el mismo entorno de desarrollo integrado (IDE), que les permite compartir herramientas y facilita la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes aprovechan las funciones de .NET Framework, que ofrece acceso a tecnologías clave para simplificar el desarrollo de aplicaciones Web ASP y Servicios Web XML.

Visual Studio incluye un nuevo diseñador de páginas Web denominado Visual Web Developer que incluye muchas mejoras para la creación y edición de páginas Web ASP.NET y páginas HTML.

El entorno integrado de Visual Studio incluye herramientas destinadas a dispositivos como los PDA y Smartphone. Entre las mejoras se encuentran tiempos de ejecución de dispositivos nativos y herramientas de Visual C++, diseñadores administrados que proporcionan un modo WYSIWYG mejorado específico para cada plataforma y compatibilidad con varios factores de forma, un nuevo emulador, herramientas de control de datos similares al escritorio, y proyectos de implementación para el usuario final que eliminan la edición manual de los archivos .inf.

Los formularios Windows Forms sirven para crear aplicaciones de Microsoft Windows en .NET Framework. Este marco de trabajo proporciona un conjunto de clases claro, orientado a objetos y ampliable, que permite desarrollar complejas aplicaciones para Windows. Además, los formularios Windows Forms pueden actuar como interfaz de usuario local en una solución distribuida de varios niveles. Para obtener más información, vea Introducción a los formularios Windows Forms.

I.IV.II. Evolución de .NET

- Visual Studio .NET (2002).

  Se publicó en 2002 y fue la primera versión de Visual Studio en introducir el framework .NET. Esta versión de Visual Studio introdujo, junto con el Framework .NET tres nuevos lenguajes de programación, C#, VB.NET y Visual J#.

  En esta primera versión de Visual Studio .NET se podían programar aplicaciones Windows.Forms (aplicaciones de escritorio) y aplicaciones ASP.NET (Aplicaciones Web).

- Visual Studio .NET 2003.

  Se publicó en 2003 fue una actualización menor de Visual Studio .NET, básicamente propiciada por la introducción de la versión 1.1 del Framework .NET.

En esta versión se añadió por primera vez la posibilidad de programar para dispositivos móviles usando .NET, ya fuera usando el Compact Framework, o ASP.NET...

- Visual Studio 2005

  Se publicó el 4 de Octubre de 2005, se basó en el framework .NET 2.0. Añade soporte de 64-bit (x86-64: AMD64 e Intel 64, e IA-64: Itanium)

  Se publican las Ediciones: Express, Standard, Professional, Tools for Office, y 5 ediciones Visual Studio Team System (Architects, Software Developers, Testers, y Database Professionals)

  La versión interna de Visual Studio 2005 es la 8.0, mientras que el formato del archivo es la 9.0.

  A partir de la introducción en el mercado de la versión 2005 de Visual Studio Microsoft publicó lo que se conoce como ediciones Express de distintos programas. Las versiones Express son versiones limitadas pero gratuitas, pensadas para usos no profesionales (principiantes, aficionados y pequeños negocios), existiendo una edición independiente para cada lenguaje.

  Visual Basic Express Edition es una versión de Visual Studio limitada. Esta versión permite sólo programar en VB.NET, y además limita el tipo de proyectos que se pueden desarrollar. Visual Web Developer Express Edition permite programar páginas ASP.NET en VB.

  Se lanzó el service Pack 1 para Visual Studio 2005 el 14 de Diciembre de 2006.

- Visual Studio 2008

  El IDE de Visual Studio 2008 permite trabajar contra 3 .NET frameworks diferentes:

  .NET Framework 2.0
  .NET Framework 3.0
  .NET Framework 3.5

  Es muy fácil de usar gracias al desarrollo de hardware. Además, integra el framework ASP.NET AJAX para el desarrollo de webs con tecnología AJAX.

I.IV.III. El entorno .NET Framework

.NET Framework es un entorno multilenguaje que permite generar, implantar y ejecutar aplicaciones y Servicios Web XML. Consta de tres partes principales:

- Common Language Runtime

  A pesar de su nombre, el motor en tiempo de ejecución desempeña una función tanto durante la ejecución como durante el desarrollo de los componentes.

Cuando el componente se está ejecutando, el motor en tiempo de ejecución es responsable de administrar la asignación de memoria, iniciar y detener subprocesos y procesos, y hacer cumplir la directiva de seguridad, así como satisfacer las posibles dependencias del componente sobre otros componentes. Durante el desarrollo, el papel del motor en tiempo de ejecución cambia ligeramente; a causa de la gran automatización que permite (por ejemplo, en la administración de memoria), el motor simplifica el trabajo del desarrollador, especialmente al compararlo con la situación actual de la tecnología COM. En concreto, funciones tales como la reflexión reducen de forma espectacular la cantidad de código que debe escribir el desarrollador para convertir la lógica de empresa en componentes reutilizables.

- Clases de programación unificadas.

  El entorno de trabajo ofrece a los desarrolladores un conjunto unificado, orientado a objetos, jerárquico y extensible de bibliotecas de clases (API). Actualmente, los desarrolladores de C++ utilizan las Microsoft Foundation Classes y los desarrolladores de Java utilizan las Windows Foundation Classes. El entorno de trabajo unifica estos modelos dispares y ofrece a los programadores de Visual Basic y JScript la posibilidad de tener también acceso a las bibliotecas de clases. Con la creación de un conjunto de API comunes para todos los lenguajes de programación, Common Language Runtime permite la herencia, el control de errores y la depuración entre lenguajes. Todos los lenguajes de programación, desde JScript a C++, pueden tener acceso al entorno de trabajo de forma parecida y los desarrolladores pueden elegir libremente el lenguaje que desean utilizar.

- ASP.NET.

  Construye las clases de programación de .NET Framework, lo que proporciona un modelo de aplicación Web con un conjunto de controles e infraestructura que facilitan la generación de aplicaciones Web. ASP.NET incluye un conjunto de controles que encapsulan elementos comunes de interfaz de usuario de HTML, como cuadros de texto, botones y cuadros de lista. Sin embargo, dichos controles se ejecutan en el servidor Web, y representan la interfaz de usuario en el explorador como HTML. En el servidor, los controles exponen un modelo de programación orientado a objetos que proporciona la riqueza de la programación orientada a objetos al desarrollador Web. ASP.NET también proporciona servicios de infraestructura, como la administración de estado y el reciclaje de procesos, que reduce aún más la cantidad de código que debe escribir el desarrollador y aumenta la confiabilidad de la aplicación. Asimismo, ASP.NET utiliza estos mismos conceptos para permitir a los desarrolladores la entrega de software como un servicio. Al utilizar características de Servicios Web XML, los desarrolladores de ASP.NET pueden escribir su lógica empresarial y utilizar la infraestructura de ASP.NET para entregar ese servicio a través de SOAP. Para obtener más información, vea Introducción a la programación de servicios Web XML en código administrado.

I.IV.IV. Arquitectura de .NET Compact Framework

.NET Compact Framework hereda la arquitectura .NET Framework completa de Common Language Runtime para ejecutar código administrado. Proporciona interoperabilidad con el sistema operativo Windows CE de un dispositivo para tener acceso a funciones nativas e integrar los componentes nativos favoritos en una aplicación.

Puede ejecutar aplicaciones nativas y administradas de manera simultánea. El host del dominio de aplicación, que también es una aplicación nativa, inicia una instancia del Common Language Runtime para ejecutar el código administrado. En la imagen 1.10. se resume la arquitectura de la plataforma .NET Compact Framework.

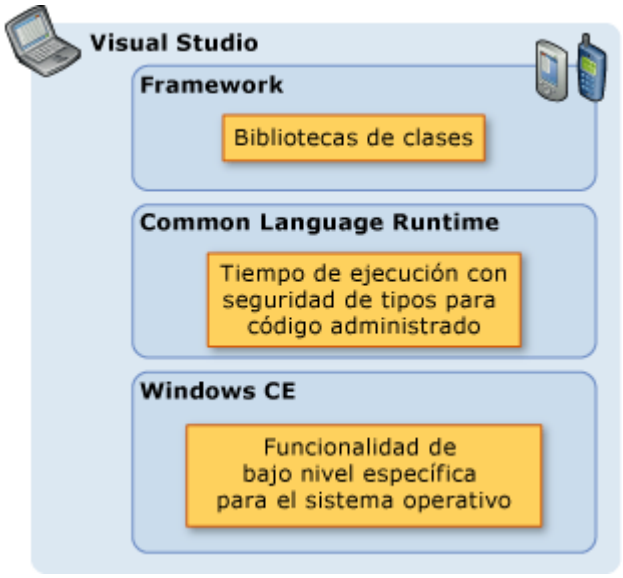


Imagen 1. 10. Arquitectura del Net Compact Framework

.NET Compact Framework utiliza el sistema operativo Windows CE para la funcionalidad central y para diversas características específicas de dispositivos. Varios tipos y ensamblados, como los de los formularios Windows Forms, gráficos, dibujos y servicios Web, se han recompilado para que se ejecuten eficazmente en los dispositivos, en lugar de copiarse de .NET Framework completo.

.NET Compact Framework ofrece la siguiente interoperabilidad con Windows CE:

- Compatibilidad con seguridad nativa.
- Integración completa con programas de instalación nativos.
- Interoperabilidad con código nativo mediante la interoperabilidad COM y la invocación de plataformas.

También el Common Language Runtime (CLR) de .NET Compact Framework se ha vuelto a generar para permitir que los recursos restringidos se ejecuten en memoria limitada y lograr un uso eficaz de la energía.

Entre Windows CE y el Common Language Runtime existe una capa de adaptación de plataforma, que no aparece en la ilustración, para asignar las interfaces de servicios y

dispositivos necesarias para CLR y Framework a los servicios e interfaces de Windows CE.

.NET Compact Framework es un subconjunto de .NET Framework pero también contiene características diseñadas en exclusiva. Ofrece prestaciones y facilidad de uso para acercar a los desarrolladores de aplicaciones nativas para dispositivos a .NET Framework, y para acercar a quienes desarrollan aplicaciones de escritorio a los dispositivos. NET Compact Framework implementa aproximadamente un 30 por ciento de la biblioteca de clases completa de .NET Framework y contiene asimismo las características y clases específicas del desarrollo móvil e incrustado

Desarrollar aplicaciones para dispositivos inteligentes con Microsoft Visual Studio 2005 es tan sencillo como desarrollar aplicaciones para escritorio. El desarrollo de dispositivos inteligentes en Visual Studio incluye un conjunto de emuladores y tipos de proyecto que cubren el desarrollo para Pocket PC, Smartphone, PDAs y Windows CE incrustado.

I.IV.V. Historia de Windows Mobile

Windows Mobile es un sistema operativo compacto, con una suite de aplicaciones básicas para dispositivos móviles basados en la API Win32 de Microsoft. Los dispositivos que llevan Windows Mobile son Pocket PC, Smartphones y Media Center portátil. Ha sido diseñado para ser similar a las versiones de escritorio de Windows.

Tanto Windows Mobile para Pocket PC, como Windows Mobile para Smartphone, poseen bastantes aspectos parecidos. En la pantalla "Hoy" nos mostrará la fecha actual, la información del dueño, las citas próximas, los mensajes E-mail, y las tareas. En la parte inferior aparecerá, generalmente, una barra con dos botones. También incluye una barra que incluye iconos para notificar el estado del Bluetooth, batería, cobertura, etc. Este tema predeterminado puede ser cambiado añadiendo o eliminando complementos, como por ejemplo, alarma, temperatura, estado de la batería.

Treo 700w con Windows Mobile 5.En la barra de tareas muestra: la hora actual, el volumen y el estado de la conectividad. Cuando un programa o un mensaje están abiertos el espacio en blanco, en el que estaba el reloj se convierte en una "ok" o un icono de cerrar (x). La característica principal de la barra de tareas es el botón de Inicio, que está diseñado para que sea parecido al botón de Inicio de las versiones de escritorio de Windows. El menú de Inicio ofrece programas abiertos recientemente, nueve entradas del menú personalizadas, y accesos directos a programas, ajustes, búsquedas, y ayuda.

Las versiones Pocket PC incluyen en Windows Mobile aplicaciones de Microsoft Office. Éstos incluyen Pocket Word y Pocket Excel. En Windows Mobile 5.0 se incluye Pocket PowerPoint. Estas versiones incluyen muchas de las características que se utilizan en versiones de escritorio, pero algunas otras características como la inserción de las tablas e imágenes no se han incluido versiones anteriores a Windows 5.0. ActiveSync tiene la capacidad de convertir archivos de versiones de escritorio a archivos compatibles con Pocket PC.

Outlook Mobile es también un programa que viene con Windows Mobile. Esto incluye tareas, calendario, contactos, y la bandeja de entrada. Microsoft Outlook para las versiones de escritorio se incluye a veces en los CD-ROM's del fabricante del Pocket PC.

Windows Media Player for Windows Mobile se añade con el software. Actualmente, todas las Pocket PC incluyen la versión 9 del reproductor, pero la versión 10 se ha incluido con un hardware más nuevo y con las nuevas versiones de Windows Mobile. Para algunos dispositivos, la versión 10 está disponible para su descarga solo para determinados dispositivos - éstos incluyen los dispositivos de la gama de Dell Axim. Windows Media Player reproduce: WMA, WMV , MP3, y AVI. Los archivos MPEG actualmente no están soportados, y se debe descargar un programa de terceros para reproducirlos, y los archivos de WAV se reproducen en un reproductor por separado. Algunas versiones son también capaces de reproducir M4A.

Algunas versiones del Windows Mobile han sido:

- Windows Mobile 6.

  Antes con el nombre en clave Crossbow es la última versión de la plataforma Windows Mobile y fue lanzado el 12 de febrero del 2007 en el 3GSM World Congress 2007. Ofrece tres versiones: Windows Mobile 6 Standard para Smartphones (teléfonos sin pantalla táctil), Windows Mobile 6 Professional para PDAs con la funcionalidad del teléfono (Pocket PC Phone Edition), y Windows Mobile 6 Classic para PDAs sin telefonía IP.[1] Utiliza Windows CE 5.2 y ligado fuertemente a los productos: Windows Vista, Windows Live, Microsoft Office y Exchange 2007. Tiene las siguientes caracteristicas:

  - Basado en Windows CE 5.0 (versión 5.2)
  - Soporta las resoluciones 800x480 y 320x320.
  - Opción de 1:1 en la páginas web
  - Acceso de escritorio remoto mejorado
  - Desarrollo y distribución de aplicaciones más rápido y más fácil.
  - Soporte VoIP con los codec del audio AEC (Acoustic Echo Cancelling) y MSRT
  - Windows Live para Windows Mobile.
  - Opción de mejora de la experiencia del cliente
  - La pila Bluetooth de Microsoft ha mejorado notablemente.
  - Cifrado de la tarjeta de almacenamiento - Windows Mobile 6 para Pocket PC y Smartphone soportan el cifrado de los datos almacenados en tarjetas externas de almacenamiento.
  - Outlook Mobile ahora soporta HTML .
  - Capacidad de buscar para contactos en Exchange Server Address Book.
  - Soporte AJAX, JavaScript y XMLDOM en Internet Explorer Mobile.
  - .NET Compact Framework v2 SP1 en la ROM.
  - SQL Server Compact Edition en la ROM.

- Windows Mobile 5.

  Anteriormente con el nombre en clave "Magneto", salió al mercado el 9 de mayo del 2005. Utiliza Windows CE 5.0 y utiliza .NET Compact Framework 1.0 SP2 - una plataforma de desarrollo .NET para los programas basados en .NET que utiliza. Tiene las siguientes caracteristicas:

  - Una nueva versión de Office llamada "Office Mobile".
  - Se agregará una versión de Powerpoint denominada "Powerpoint Mobile".
  - Excel Mobile añade la capacidad de ver representaciones gráficas.
  - Word Mobile incluirá la capacidad de insertar tablas y gráficos.
  - Reproductor "Windows Media 10 Mobile".
  - Identificador de llamadas con fotos.
  - Un paquete multimedia que facilitará la administración de vídeos y fotos.
  - Ayuda mejorada de Bluetooth.
  - Interfaz de administración GPS para los programas de navegación instalados.
  - Mejoras de la funcionalidad de "Microsoft Exchange Server" las mejoras
  - Soporte para teclados QWERTY incluido por defecto.
  - ActiveSync 4.2, prometiendo 10-15% de aumento de la velocidad en la sincronización de datos.
  - Cliente para PPTP y L2TP/IPsec VPNs.
  - La memoria no volátil (ROM) está disponible en Pocket PC permitiendo un aumento de la batería. Anteriormente más del 50% (suficiente para 72 horas de almacenaje) de energía de la batería se reservaba para mantener datos en la memoria RAM (volátil). Los dispositivos basados en Windows usa la memoria RAM como su medio de almacenaje primario al uso de memoria flash.

- Windows Mobile 2003 Second Edition.

  También conocida como Windows Mobile 2003SE, salió el 24 de marzo, 2004 y la Dell Axim x30 fue la primera en tenerlo. Incluye un número de mejoras sobre su precursor, como:

  - La opción de cambiar la orientación de la pantalla. Esto no está disponible en la versión de Smartphone.
  - Soporte para una resolución de pantalla VGA (640×480). También se apoya un nuevo Factor de forma del cuadrado (240×240 y 480×480 para las pantallas de VGA), que favorece a los fabricantes que desean incluir un teclado hardware. Aunque no era su idea original, Microsoft decidió agregarla debido a la presión de fabricantes del Pocket PC.
  - Soporte para Wi-Fi.
  - Windows 2003SE Mobile utiliza Windows CE 4.21.111

- Windows Mobile 2003.

  Fue lanzada el 23 de junio, 2003, y era el primer lanzamiento bajo el nombre Windows Mobile. Vino en tres ediciones diferentes. Dos de estas ediciones son muy similares: Windows Mobile 2003 Pocket PC Edition y Windows Mobile 2003 Pocket PC Phone Edition, este último diseñado para los Pocket PC que tienen características de teléfonos móviles (como HTC's Himalaya, distribuido en muchos países como Qtek, XDA, MDA o VPA).

  Windows Mobile 2003 Smartphone Edition, es una plataforma substancialmente diferente ya que está limitada por las características especiales de este tipo de dispositivos. Algunas de estas limitaciones son: funcionamiento por teclas al no disponer de pantalla táctil, resolución de pantalla más baja, modelo de seguridad que impide instalar aplicaciones no firmadas y modelo de memoria diferente (diferente tipo de memoria y menor cantidad).

  Windows Mobile 2003 es conocido también como Windows CE 4.20.

- PocketPC 2002. Utiliza Windows CE 3.0. Diseñado para dispositivos Pocket PC con pantalla 240 × 320 (QVGA) (sin teclado), Windows Mobile 2002 era, como el lanzamiento original PocketPC 2000, una entidad independiente en la gama de dispositivos Microsoft Embedded. Con los lanzamientos futuros, las líneas de Pocket PC y Smartphone chocaban cada vez más, mientras que los términos de licencia se relajaron permitiendo que los OEMs se aprovecharan de las ideas más innovadoras de diseño.

I.V. Historia de los dispositivos Móviles

La línea entre lo que es un dispositivo móvil y lo que no lo es puede ser un poco difusa, pero en general, se pueden definir como aquellos micro-ordenadores que son lo suficientemente ligeros como para ser transportados por una persona, y que disponen de la capacidad de batería suficiente como para poder funcionar de forma autónoma. Normalmente, son versiones limitadas en prestaciones, y por tanto en funcionalidades, de los ordenadores portátiles o de sobremesa. Por cierto, los ordenadores portátiles no se consideran como dispositivos móviles, ya que consumen más batería y suelen ser un poco más pesados de lo que se espera de algo pensado para llevar siempre encima

A grandes rasgos, y dependiendo del tamaño los dispositivos se pueden dividir en tres clases.

- Teléfonos. Son los más pequeños de la casa, y por tanto los más ligeros y más transportables. En general, también son los más baratos, aunque un teléfono de gama alta puede superar en precio a muchos de sus hermanos mayores, las PDAs. Su función primordial era clara: recibir y realizar llamadas; aunque parece que dentro de poco va a comenzar a ser complicado encontrar teléfono que sirvan para eso. Funcionalidades propias de ordenadores, o de dispositivos de otro tipo, como la grabación y edición de vídeo, realización de fotografías, lectura de documentos, localización en mapas, navegación por Internet, y muchas cosas más, son no sólo habituales, sino esperadas en cualquier teléfono moderno.

- PDAs, organizadores electrónicos u ordenadores de mano. Su nombre (PDA) significa Personal Digital Assistant (asistente personal digital), un término acuñado en sus primeros años de historia, pero que resume bien su funcionalidad principal, que es servir como organizadores, con agenda, calendario, gestión de contactos, y que posteriormente han ido creciendo, de forma que actualmente sirven tanto como aparatos en los que leer un libro como en los que encontrarse en un mapa. La línea que los separa de los teléfonos es cada vez más difusa.

- Consolas. En realidad esta categoría debería llamarse "dispositivos orientados a jugar", porque son más que simples consolas. Los dos ejemplos actualmente en el mercado son la Sony PlayStation Portable (PSP) y la Nintendo DS, que no sólo sirven para jugar, sino que integran algunas de las funcionalidades típicas de una PDA, como reproducción de archivos multimedia, integración con agenda y calendario, o navegador de Internet

El primer dispositivo móvil, para muchos, es la Newton, desarrollada y comercializada por Apple, y que estuvo a la venta entre 1993 y 1998. La Newton era un dispositivo revolucionario para su tiempo, que implementaba un sistema de reconocimiento de escritura y que podía sincronizarse con un ordenador de sobremesa (de Apple, claro está). Fue tan revolucionaria, y se adelantó tanto a su tiempo, que fue un fracaso comercial, por lo que terminó retirándose del mercado. Pero aunque la Newton pueda considerarse como la primera PDA, desde luego no fue el primer dispositivo portátil programable.

Durante los años 80, tanto Casio como Hewlett-Packard desarrollaron y comercializaron varias calculadoras programables, que si bien no tenían la capacidad de sincronizar sus datos con un ordenador de sobremesa, sí tenían capacidades gráficas, y accesorios que tal vez puedan sonar extravagantes ahora, como impresoras, o tarjetas de memoria extraíbles de 1Kb

Las calculadoras programables de Casio rivalizaban en prestaciones con las de Hewlett-Packard. Si la FX-750P podría considerarse como el buque insignia de la marca japonesa, la serie HP48 lo era para los californianos. Las calculadores de esta serie, que se dividían en dos ramas, la S para los modelos estándar y la G para los de mayor funcionalidad, estuvo en producción entre los años 1990 y 2003. Las especificaciones comunes a todos los modelos de la gama eran una pantalla de 131x63 píxeles, un puerto de comunicaciones por infrarrojos y otro serie de 4 pines, y 512 KB de memoria. El modelo más alto de la gama, la HP 48GX, soportaba dos tarjetas de expansión de memoria, de forma que se podía llegar a acumular un total de 5MB.

Tras la muerte de la Newton, nacieron los dos dispositivos que durante unos años dominaron el mercado: la Pilot y el PocketPC.

Los dispositivos de Palm se adelantaron a los de Microsoft en un par de años. En parte por ello, su salida a producción fue un éxito de ventas, llegando incluso a acumular, en 2001, un 60 por ciento del mercado americano. Sin embargo, Palm se resintió del declive global del mercado de las PDA, comenzando un declive que coincidió prácticamente en el tiempo con la entrada en el mercado del gigante de la informática de consumo: Microsoft.

En el año 2000 vio la luz el primer PocketPC, el hijo de la Newton, y que al contrario que ésta, sí ha sobrevivido hasta hoy, pese a no ser un éxito comercial de grandes dimensiones, gracias a la continuidad y el apoyo decidido de Microsoft.

Los primeros PocketPC tenían como sistema operativo el llamado Windows CE 3.0. Por su nombre podría parecer que era una versión aligerada del sistema operativo más utilizado en el mundo, pero en realidad no tenía mucho que ver con éste.

La mayor razón para el éxito del PocketPC ha sido su relativa sencillez de manejo y su integración con ordenadores de escritorio basados en Windows. La integración es tal que para realizar una sincronización entre ambos basta con conectar el PocketPC al ordenador con un cable. Actualmente, los PocketPC y las Palm tienen pantallas de resolución VGA, en prácticamente todos los casos incorporan protocolos de comunicaciones inalámbricos, como Bluetooth o Wifi, o unidades de GPS. Por prestaciones, son, sin duda, los hermanos mayores del mundo de la movilidad.

Pero si las PDAs son las reinas en lo referente a prestaciones, en cuanto a unidades en el mercado, el rey indiscutible es el teléfono móvil. ¿Quién no tiene un teléfono móvil? En realidad, lo raro es que haya quien no tenga un teléfono totalmente funcional guardado en un cajón, porque se ha comprado otro que, en el plazo de unos meses, multiplicaba por diez o por cien las prestaciones de su antiguo Terminal.

Si la Newton, la Palm y el PocketPC han sido los que han abierto el camino, los que realmente han entrado con toda la fuerza posible a ocupar ese mercado han sido los teléfonos móviles. Actualmente, se calcula que en España hay más terminales móviles (unos 44 millones) que habitantes. En el 37% de los hogares hay un móvil, en el 34% hay dos terminales, y en el 17% hay tres, según datos de Cetelem de finales de 2005.
Sin embargo, los móviles que verdaderamente pueden considerarse equivalentes a las PDAs son los de gama alta, o smartphones.

El término smartphone es engañoso. Su traducción literal sería “teléfonos inteligentes”, y se utiliza indiscriminadamente para hacer referencia a cualquier teléfono de gama alta, englobando tanto a dispositivos de la Serie 60 o superior de Symbian, como a los que funcionan bajo Windows Mobile o bajo Palm OS. De todas formas, la marca comercial SmartPhone es propiedad de Microsoft. Porque, efectivamente, hay una línea de teléfonos que funcionan bajo Windows Mobile. Por un lado, los PocketPC Phone Edition, que son híbridos de teléfono y PocketPC y por otro los llamados comercialmente SmartPhones. La diferencia entre ambos está, fundamentalmente, en la pantalla táctil. Los PocketPC Phone Edition la tienen, mientras que los SmartPhones no, y que se manejan de forma similar a los móviles Symbian de gama alta. Tanto PocketPC Phone Edition como PocketPC se engloban dentro de la gama Windows Mobile de Microsoft.

También Palm mantiene su irreductible nicho de mercado dentro de los teléfonos de gama alta: la serie Treo. Los Treo son híbridos de teléfono y PDA, con teclado QWERTY, y funcionan, salvo uno de los modelos (en concreto el Treo 700w, que utiliza PocketPC Phone Edition) bajo PalmOS.

En todo caso, el mercado de los teléfonos de gama alta estaba copado, hasta hace muy poco, por los modelos con sistema Symbian. Sin embargo, en los últimos dos años hemos asistido a la irrupción del representante por excelencia del "techno-lust" el iPhone de Apple. Si Apple fue la que abrió el camino con la Newton, es el que ha vuelto a revolucionar el mercado con el iPhone, en parte porque ha sido el único fabricante que realmente ha conseguido desarrollar un dispositivo que integrara a todos los anteriores existentes por separado (teléfono, reproductor mp3 y PDA), reinventando la forma en la que se interactúa con él.

I.V.I. Plataformas y lenguajes soportados

Cada una de las plataformas tiene sus particularidades, no sólo en cuanto al manejo del dispositivo por el usuario, sino también a la hora de desarrollar aplicaciones para las mismas.

No será lo mismo programar una aplicación para Windows Mobile que para Symbian, entre otras cosas porque no todas las plataformas soportan los mismos lenguajes de programación. Algunas plataformas soportadas son las siguientes:

- Windows Mobile

  Como ya hemos visto con anterioridad, el sistema operativo en el que están basados los PocketPC actualmente se llama Windows Mobile. Este sistema se pretende vender como una versión muy aligerada de Windows, pero en realidad no tiene mucho que ver con él. El paradigma de funcionamiento es similar, basado en ventanas, aunque éstas se comportan de forma muy diferente a como lo hacen en su hermano mayor de escritorio. Por ejemplo, al cerrar la ventana de un programa éste no se cierra realmente, sino que se sigue ejecutando como si se hubiera minimizado.

  El punto fuerte de estos dispositivos es que ofrecen funcionalidades similares a las de sus hermanos mayores, Por ejemplo, se pueden editar documentos de word, hojas de cálculo de excel, leer libros en formato pdf o chm, recibir y enviar correo electrónico, manejar una agenda, la libreta de contactos, sincronizar datos con el PC, navegar por Internet, utilizar un GPS... en definitiva, casi lo mismo que en un PC, pero sin teclado y con la pantalla bastante pequeña.

  La entrada de datos se realiza a través de la pantalla, que es táctil, y gracias al sistema de reconocimiento de escritura que implementan, que permite trabajar de una forma bastante natural y rápida. La mayoría de las aplicaciones para estos dispositivos se desarrollan en .Net, la plataforma de desarrollo de Microsoft, o directamente en C++, aunque aún quedan, como reminiscencia de sus inicios como Windows CE, algunas aplicaciones escritas en Embedded C++ o Embedded Basic, que eran dos entornos de desarrollo basados respectivamente en C++ y Basic, pero con muchas limitaciones.

  Pero en la actualidad casi todos los desarrollos para Windows Mobile se realizan en C++ o en .Net, basándose en el Compact Framework. El principal problema a la hora de desarrollar aplicaciones para estos dispositivos es el precio de las

herramientas de desarrollo. Estas herramientas (Visual Studio, un entorno de desarrollo que soporta varios lenguajes como C#, C++, J#, JScript o ASP .Net) son de altísima calidad, pero de un precio que no está al alcance de todos los que se quieran aventurar a escribir una aplicación para PocketPC.

Por otra parte, el hecho de que Microsoft esté detrás de la plataforma, dándole todo su apoyo, se nota en la cantidad y la calidad de la documentación disponible para los desarrolladores. Newsletters, una sección sólo para dispositivos en el MSDN, blogs de los ingenieros de Microsoft que trabajan en Windows Mobile, actualizaciones de la documentación en DVDs. Tal vez por eso mismo, y pese al precio del entorno de desarrollo, la competencia es muy alta. Hay muchas compañías que producen aplicaciones para Windows Mobile, y que invierten mucho dinero en su desarrollo. No es, por tanto, un nicho de mercado en el que sea sencillo introducirse.

En cualquier caso, siempre que se vaya a desarrollar aplicaciones para esta plataforma, deben tenerse en cuenta las limitaciones específicas de la misma, sobre todo a la hora de intentar desarrollar interfaces que sean lo más point-and-click posibles, y que necesiten de la menor introducción de textos posible.

- Symbian

  Si los PocketPC están basados en un sistema operativo de Microsoft llamado Windows Mobile, la gran mayoría de los teléfonos móviles funcionan gracias a un sistema operativo llamado Symbian. Hace unos años, a estos teléfonos se les llamaba con el nombre genérico de SmartPhones.

  Symbian es un consorcio en el que participan los mayores fabricantes de teléfonos móviles, con Nokia a la cabeza. Sony Ericsson, BenQ (que hace poco absorbió a Siemens), Fujitsu, Lenovo, Motorola, Panasonic, Samsung o Sharp están representados en Symbian, y son por tanto, partícipes en el desarrollo y la expansión del sistema.

  De todos estos fabricantes, el que mayor cuota de mercado tiene es Nokia (que con algunas fluctuaciones, viene siendo desde hace años de alrededor del 40%), que además es el fabricante que primero apostó por este sistema operativo, y el que más uso hace de él. Por tanto, a partir de ahora, al referirme a teléfonos Symbian, lo haré sobre todo a teléfonos Nokia.

  Symbian es un sistema operativo escrito en C++, por lo que presenta muy bajo consumo de recursos del dispositivo, a la vez que se ejecuta con gran rapidez. El sistema operativo Symbian se presenta en varios "sabores". En concreto, Nokia divide sus dispositivos Symbian en tres familias, que se llaman respectivamente Serie 40, Serie 60 y Serie 80.

  La Serie 40 es la que agrupa a los teléfonos Symbian con pantallas más pequeñas (en general, hasta de 240x320 píxeles), y tiene ciertas limitaciones sobre las otras series, sobre todo en lo referente a la cantidad de recursos del teléfono que puede poner a disposición de las aplicaciones que se ejecuten sobre él.

La Serie 60 es la más extendida, y es donde suelen estar los teléfonos de gama media y alta de Nokia, a los que muchas veces se hace referencia como teléfonos multimedia. Con alguna excepción, suelen tener pantallas más grandes que los de la serie 40, y suelen ser teléfonos específicamente optimizados para ejecutar aplicaciones J2ME. La Serie 80, finalmente, es la de los llamados Communicators, esos teléfonos que se abren longitudinalmente y que esconden teclados QWERTY. En realidad son un paso intermedio entre el teléfono y la PDA. Aparte de la implementación que hace Nokia de Symbian, Sony Ericsson implementa otro sabor más de ese sistema, llamado UIQ, que se maneja por teclado o a través de una pantalla táctil. Estos dispositivos suelen implementar sistemas de reconocimiento de escritura, y los más conocidos son, como he dicho, la serie P de Sony Ericsson.

Actualmente, en el mercado de los teléfonos móviles Symbian es el sistema más extendido, aunque no el único. La mayoría de los fabricantes siguen implementando sus propios sistemas, por lo que sigue sin haber una uniformidad o coherencia entre dispositivos.

Las tres grandes posibilidades tecnológicas a la hora de desarrollar aplicaciones para teléfonos móviles son C++, J2ME y Flash Lite.

- C++

  Es el lenguaje de programación por excelencia para aplicaciones que necesitan extraer el máximo del terminal, tanto en capacidad de procesamiento y por lo tanto en velocidad de ejecución, como en utilizar las posibilidades de hardware que ofrezca el dispositivo. El ejemplo más claro de aplicación candidata a ser realizada en C++ sería un juego de conducción, con capacidades multiusuario a través de bluetooth.

  Es ideal, por tanto, para aplicaciones críticas (como por ejemplo, sistemas operativos para teléfonos como Symbian). C++ es un lenguaje de programación, de los que se llaman orientados a objetos, que en realidad es una evolución del lenguaje más utilizado en la historia de la informática: el lenguaje C. Desarrollar en C++ es bastante complicado para casi cualquier programador. Hacen falta, por tanto, programadores de un perfil muy especializado o muy alto, y el lenguaje en sí no hace

- J2ME

  J2ME es un subconjunto del lenguaje Java. ¿Pero qué es el lenguaje Java? Es un lenguaje de programación desarrollado a principios de los años noventa por James Gosling y algunos de sus colegas de Sun Microsystems. Las principales características de este lenguaje son, por un lado, que es de los llamados orientados a objetos, lo que supuestamente permite desarrollar aplicaciones más complejas con mayor facilidad, y

por otro, que fue diseñado para ser independiente de la plataforma, lo que supuso una gran novedad en su momento

.

Sin embargo, el caso de J2ME es especial, ya que es de todo, menos independiente de la plataforma. La especificación J2ME está dividida en dos grandes grupos, dependiendo de la cantidad de funcionalidades para las que se quiera dar soporte. Esos grupos son las llamadas Configuraciones (en toda la documentación sobre J2ME se suele hacer referencia a ellas por su nombre en inglés: Configurations). Hay dos configuraciones, que dividen la plataforma en dos grandes grupos de dispositivos.

Por un lado, los dispositivos más potentes, son los que soportan la Configuración CDC. Durante mucho tiempo en este grupo sólo cabían las PDAs (dispositivos como la Palm, por ejemplo), dadas las exigencias de memoria y tamaño de pantalla para cumplir con la especificación.

La otra configuración, la llamada CLDC, es la que agrupaba a los dispositivos con menor capacidad de procesamiento, y es donde siempre han estado los móviles.

- Flash Lite

Flash Lite es un adaptación de la plataforma Flash para dispositivos móviles. Actualmente coexisten varias versiones, Flash Lite 1.1, Flash Lite 2.0 y Flash Lite 2.0. Flash Lite 1.1, es la versión más extendida, sobre todo en Asia, y basada en la versión 4 de player de flash. Y es la más extendida por dos razones primordiales; en primer lugar porque es la que más tiempo lleva disponible, y en segundo lugar porque es la que necesita de hardware menos potente para su ejecución.

Al estar basado en la versión 4 del player de flash, sólo soporta la sintaxis de ActionScript de Flash 4, una sintaxis bastante extraña y alejada de las convenciones de programación modernas. Esa forma de programar necesita de vinculaciones muy fuertes entre el código y los gráficos, por lo que es muy difícil que los desarrolladores que no estén acostumbrados a ella se puedan adaptar con facilidad.

Además, para desarrollar cualquier aplicación con cierta interactividad, termina siendo necesario repartir el código entre multitud de elementos gráficos, por lo que la complejidad del desarrollo crece exponencialmente con la complejidad de la aplicación. Por no hablar del coste de mantenimiento.

Flash Lite 2.0 está basado en el Flash Player 7. En realidad, la funcionalidad que permite es muy similar a la de cualquier aplicación flash basada en esa versión del player. Por tanto, ahora es posible guardar datos en la memoria del teléfono o cargar ficheros XML con estructuras de datos complejas a través de la conexión GPS. Para el programador la mejora ha sido también sustancial, ya que puede aplicar las técnicas

modernas de programación: programación orientada a objetos, patrones de diseño, aplicaciones dirigidas por eventos.

- Iphone OS.

  El iPhone, el último en llegar, rompe también con las líneas maestras en cuanto a desarrollo se refiere, marcadas por sus antecesores.

  El sistema operativo utilizado es una versión aligerada de Mac OS X, el sistema detrás de los ordenadores de Apple, y por tanto, las herramientas de desarrollo que se deben utilizar son las mismas que para trabajar en escritorio.

  Objective-C es el lenguaje que debe utilizarse para desarrollar aplicaciones nativas para iPhone OS es Objective-C, un superset de C (una especie de C enriquecido), de forma que el desarrollador se apoya en un extenso Framework orientado a objetos, escalable y altamente modular, llamado Cocoa.

  En realidad, Cocoa es una colección de frameworks, que proporcionan todas las piezas necesarias para construir una aplicación: desde elementos de interfaz hasta gestión de tráfico de red.

  Una aplicación para iPhone, casualmente desarrollada por el autor. Pese a lo modular y extenso de los frameworks, el mayor problema con el que se encuentran los recién llegados a la plataforma es que la curva de aprendizaje de Objective-C es bastante acusada. Especialmente si, además, se viene de un entorno de desarrollo Windows, ya que no sólo hay que cambiar de lenguaje, sino de forma de desarrollo por completo.

  Además, para poder desarrollar aplicaciones para iPhone es necesario estar registrado (previo pago) como desarrollador con Apple, para así poder obtener los certificados digitales necesarios para que las aplicaciones funcionen en los dispositivos. Por si fuera poco, la única forma de poner esas aplicaciones en esos dispositivos, es a través de una tienda de aplicaciones controlada por Apple.

- Android

  Android es un sistema operativo para teléfonos, basado en el núcleo de Linux, y que, aunque disponible para cualquier fabricante como open-source, actualmente es el motor de los dispositivos comercializados por Google (en el momento de escribir este texto, sólo existe un dispositivo, el conocido en Asia como HTC Dream, y en el resto del mundo como G1).

  Android también proporciona al desarrollador un completo Framework Java, un intento de normalización de la selva de especificaciones en que se ha convertido J2ME, que al igual que en el caso del iPhone, está orientado a facilitar y hacer más rápido el desarrollo, por un lado, y a proporcionar un “look and feel” específico y reconocible de la plataforma.

También se proporciona una tienda de aplicaciones, donde se pueden vender desarrollos comerciales. En este momento, aún está por ver si la plataforma termina por alcanzar el momento suficiente como para terminar despegando.

I.V.II. Navegadores soportados

Hemos hablado de aplicaciones, pero no de Internet. se puede navegar por Internet desde un dispositivo móvil. Pero, como tantas otras cosas en estos ordenadores, con ciertas limitaciones. Y esas limitaciones vienen dadas por los navegadores a utilizar.

En el caso de los PocketPC, Windows Mobile incluye una versión de Internet Explorer muy rebajada de funcionalidad, y que no soporta muchas de las cosas que soporta su hermano mayor.

También, desde hace relativamente poco tiempo, hay una versión de Firefox, llamada Mínimo, pero que sólo funciona bajo Windows Mobile 5.0, y que es un poco propensa a leaks de memoria, como por otra parte cabe esperar de una aplicación cuyo número de versión es 0,013

En los dispositivos Symbian, sin embargo, el navegador que se está convirtiendo en la mejor opción para muchos usuarios es Opera. Opera Mobile es un navegador sólido, que soporta los estándares web, y que permite navegar sitios web "normales", que no tienen porqué ser WAP necesariamente. Y del que también hay versión para Windows Mobile.

En todos los casos, debe tenerse en cuenta, a la hora de desarrollar webs para dispositivos móviles, una serie de aspectos que harán que la web desarrollada sea accesible sin problemas para el usuario.

Por un lado es muy importante separar contenido de presentación. La utilización de marcado XHTML estricto en las páginas, junto con hojas de estilo en cascada para formatear los datos, permite, por un lado, que los navegadores sean capaces de renderizar la página sin problemas, y por otro, asignar una css específica para dispositivos móviles.

De esa forma, se pueden servir a los usuarios que utilicen, por ejemplo, un teléfono para visitar una web páginas sin imágenes, que se cargarán en menos tiempo, y serán más ligeras, por lo que le costarán menos dinero.

Además, hay que tener en cuenta que es bastante normal que los usuarios deshabiliten la carga de imágenes a la hora de navegar por Internet desde un dispositivo móvil, para de esa forma reducir no sólo los precios de espera sino los costes de la conexión, que en los casos de GPRS o 3G se tarifica por Kb. Por tanto, a la hora de plantear la estructura de la web, debemos evitar depender de imágenes para que se pueda navegar de forma correcta. Es buena idea, por tanto, hacer que todos los elementos de los menús sean sólo texto.

Tampoco tiene sentido plantear efectos dinámicos, como rollovers, y hay que ser extremadamente cuidadosos a la hora de asignar los estilos de los enlaces visitados y no visitados, para que sean visibles.

También hay que ser precavidos con los tipos de archivos que se muestran. Por ejemplo, no se deben incluir archivos swf, quicktime, o cualquier otro que no sea estrictamente html. También hay que tener cuidado con los posibles archivos que se cuelguen para descarga (zip, rar, incluso archivos de Word), que pueden no ser legibles para el dispositivo.

En el caso del iPhone, la versión de Safari incluida permite navegar webs sin hojas de estilo especiales. Sin embargo, al no soportar ningún plugin externo, el único contenido multimedia visualizable es el que esté en formato quicktime.

CAPITULO II. FUNDAMENTOS DEL NMEA

II.I. Introducción

II.I.I. Protocolo NMEA 0183 y Protocolo SiRF.

Muchos de los receptores GPS actuales permiten seleccionar el protocolo de salida entre el tradicional NMEA-0183 y el nuevo protocolo SIRF. Ambos protocolos permiten transferir los datos recibidos por el receptor GPS a un determinado programa instalado en un ordenador o PDA. Algunos de estos datos son: posición, altitud, satélites en cobertura, intensidad de su señal, hora, fecha, etc.

El protocolo NMEA-183 (National Marine Electronics Association) es un protocolo estándar, prácticamente incorporado en todos los receptores GPS y admitido por la gran mayoría de los programas que permiten conexión a un GPS. Precisamente esta estandarización y su amplia difusión es la cualidad más destacable de este protocolo.

La mayor parte de los receptores envían de forma continua sentencias NMEA 0183, normalmente un grupo por segundo. Dichas sentencias, comienzan siempre por el signo "$" y están formadas por letras, números y signos o sea caracteres ASCII. La configuración del puerto típica de este protocolo es:

Velocidad	4800 baudios
Bits de Datos	8
Paridad	None
Bits de Parada	1
HandShake	None

Cabe mencionar que puede aceptar otras velocidades, de echo actualmente existe la NMEA 0183 V.4.00 llamada NMEA 0183-HS (High Speed) Versión 1.01, la cual opera a una velocidad de 38.4 K-baud rate, con aproximadamente 59 nuevas sentencias.

Por otro lado, la empresa SiRF fundada por Kanwar Chadha, desarrolló un nuevo sistema más moderno de procesamiento de la señal de los satélites. El protocolo SiRF presenta una transferencia de datos entre el GPS y el ordenador más fluida, tiempos menores en la adquisición, mayor rapidez a la hora de calcular la posición y más precisión, además de permitir a los distintos programas un manejo más completo del receptor GPS; ya que admite el envió de comandos que permiten, por ejemplo, activar o desactivar el sistema WAAS/EGNOS, inicializar el receptor, ajustar una serie de parámetros del receptor, etc.

Velocidad	57,600 baudios
Bits de Datos	8
Paridad	None
Bits de Parada	1
HandShake	None

Se puede configurar cualquier otra velocidad. La transmisión de datos se realiza en formato binario. Otra característica del SiRF es que en cada segundo se actualiza la señal de los satélites individualmente, esto puede ser útil para hacer un control preciso de la señal recibida de cada satélite en todo momento.

De todas formas tenga presente de que en la actualidad existen una gran mayoría de programas que no aceptan dicho protocolo, como por ejemplo el Route 66, OziExplorer, Autoroute, etc. Sin embargo no dudamos de que en un futuro inmediato aparezcan nuevos programas que aprovechen al máximo las posibilidades de este novedoso protocolo.

La elección de uno u otro protocolo, dependerá realmente del software que utilice. No obstante, hoy por hoy el protocolo NMEA es el más compatible con la mayoría de los programas y está especialmente recomendado cuando tenga mas de un programa instalado y alguno de ellos no soporte SiRF, ya que de lo contrario tendría que estar cambiando la configuración del GPS al cambiar de programa.

II.I.II. Protocolo NMEA 0183.

NMEA 0183 es una especificación eléctrica y de datos combinada para la comunicación entre los dispositivos electrónicos marinos por ejemplo sounder del eco, sonares, Anemómetro (velocidad y dirección de los vientos), girocompás, piloto automático, GPS receptores y muchos otros tipos de instrumentos, dicho protocolo ha sido definido y mantenido por NMEA (National Marine Electronics Association) la cual es una asociación sin fines de lucro de fabricantes, distribuidores, instituciones educacionales y otros interesados en equipos periféricos marinos

Los equipos GPS que se apegan al estándar NMEA 0183 usan un formato simple ASCII y son compatibles con protocolos RS232, sin embargo, estrictamente hablando, el estándar NMEA no es RS232, por lo que recomiendan la conformidad con la EIA-422.

Esta seria una cadena de sentencias típicas de cadenas ASCII del estándar NMEA 0183

$<CR><LF>
MRK,0<CR><LF>
ZDA,123336.8069,17,06,2001,13.0<CR><LF>
GLL,4916.45,N,12311.12,W,225444,A,*1D<CR><LF>[1]
VTG,218.7,T,2.38,H,0.18,V<CR><LF>
SGD,-1.0,G,-1.0,M<CR><LF>
SYS,3T,9<CR><LF>
ZEV,0.28745E-006<CR><LF>
NSV,2,00,000,00,0.0,00.0,00,00,D<CR><LF>
NSV,7,00,000,00,0.0,00.0,00,00,D<CR><LF>
NSV,28,00,000,00,0.0,00.0,00,00,N<CR><LF>
NSV,1,00,000,00,0.0,00.0,00,00,D<CR><LF>
NSV,13,00,000,00,0.0,00.0,00,00,D<CR><LF>
NSV,4,00,000,00,0.0,00.0,00,00,N<CR><LF>
NSV,25,00,000,00,0.0,00.0,00,00,N<CR><LF>
&

II.II. Sentencias

II.II.I. Identificadores de Equipos Transmisores (Talker Identifiers)

Algunos identificadores de la tabla 2.1.fueron extraídas del estándar NMEA 0183 Versión 2.0 y 2.1, Actualmente existe la versión NMEA 0183 3.01 y NMEA HS 1.00

Identificador	Equipo
AG	Autopilot – General
AP	Autopilot – Magnetic
CD	Communications – Digital Selective Calling (DSC)
CR	Communications – Receiver / Beacon Receiver
CS	Communications – Satellite
CT	Communications – Radio-Telephone (MF/HF)
CV	Communications – Radio-Telephone (VHF)
CX	Communications – Scanning Receiver
DF	Direction Finder
EC	Electronic Chart Display & Information System (ECDIS)
EP	Emergency Position Indicating Beacon (EPIRB)
ER	Engine Room Monitoring Systems
GP	Global Positioning System (GPS)
HC	Heading – Magnetic Compass
HE	Heading – North Seeking Gyro
HN	Heading – Non North Seeking Gyro
II	Integrated Instrumentation
IN	Integrated Navigation
LC	Loran C
P	Proprietary Code
RA	RADAR and/or ARPA
SD	Sounder, Depth
SN	Electronic Positioning System, other/general
SS	Sounder, Scanning
TI	Turn Rate Indicator
VD	Velocity Sensor, Doppler, other/general
DM	Velocity Sensor, Speed Log, Water, Magnetic
VW	Velocity Sensor, Speed Log, Water, Mechanical
WI	Weather Instruments
YX	Transducer
ZA	Timekeeper – Atomic Clock
ZC	Timekeeper – Chronometer
ZQ	Timekeeper – Quartz
ZV	Timekeeper – Radio Update, WWV or

Tabla 2.1. Talker Identifiers

II.II.II Identificadores de Sentencias (Sentence Identifiers)

Identificador	Descripción de la Sentencia
AAM	Waypoint Arrival Alarm
ALM	GPS Almanac Data
APA	Autopilot Sentence “A”
APB	Autopilot Sentence “B”
ASD	Autopilot System Data
BEC	Bearing & Distance to Waypoint – Dead Reckoning
BOD	Bearing – Waypoint to Waypoint
BWC	Bearing & Distance to Waypoint – Latitude N/S, Longitude E/W, UTC, Status
BWR	Bearing & Distance to Waypoint – Rhumb Line Latitude N/S, Longitude E/W, UTC, Status
BWW	Bearing – Waypoint to Waypoint
DBK	Depth Below Keel
DBS	Depth Below Surface
DBT	Depth Below Transducer
DCN	Decca Position
DPT	Heading – Deviation & Variation
DSC	Digitañ Selective Calling Information
DSE	Extended DSC
DSI	DSC Transponder Initiate
DSR	DSC Transponder Response
DTM	Datum Reference
FSI	Frecuency Set Information
GBS	GPS Satellite Fault Detection
GGA	Global Positioning system Fix Data. Time, Position and fix related data for a GPS receiver
GLC	Geographic Position, Locan-C
GLL	Geographic Position – Latitude/Longitude
GRS	GPS Range Residuals
GST	GPS Pseudorange Noise Statics
GSA	GPS DOP and Active Satellites
GSV	Satellites in View
GTD	Geographic Location in Time Differences
GXA	Transit Position-Latitude/Longitude, Location and Time of Transit Fix at Waypoint
HDG	Heading-Deviation & Variation
HDM	Heading-Magnetic
HDT	Heading-True
HSC	Heading Steering Command
LCD	Loran-C Signal Data
MSK	MSK Receiver Interface (for DGPS Beacon Receivers)
MSS	MSK Receiver Signal Status
MWD	Wind Direction & Speed
MTW	Water Temperature
MWV	Wind Speed and Angle
OLN	Omega Lane Numbers
OSD	Own Ship Data
ROO	Waypoints in Active Route
RMA	Recommended Minimum Navigation Information
RMB	Recommended Minimum Navigation Information
RMC	Recommended Minimum Navigation Information
ROT	Rate of Turn
RPM	Revolutions
RSA	Rudder Sensor Angle
RSD	Radar System Data

RTE	Routes
SFI	Scanning Frecuency Information
STN	Multiple Data ID
TLL	Target Latitude and Longitude
TRF	Transit Fix Data
TTM	Tracked Target Message
VBW	Dual Ground/Water Speed
VDR	Set and Drift
VHW	Water Speed and Heading
VLW	Distance Traveled Through Water
VPW	Speed-Measured Parallel to Wind
VTG	Track Made Good and Ground Speed
VWR	Relative Wind Speed and Angle
WCV	Waypoint Closure Velocity
WDC	Distance to Waypoint-Great Circle
WDR	Distance to Waypoint-Rhumb Line
WNC	Distance to Waypoint- Waypoint
WPL	Waypoint Location
XDR	Cross Track Error-Dead Reckoning
XTE	Cross-Track Error-Measured
XTR	Cross Track Error-Dead Reckoning
ZDA	Time & Date-UTC, Day, Month, Year and Local Time Zone
ZDL	Time and Distance to Variable Point
ZFO	UTC & Time from Origin Waypoint
ZTG	UTC & Time to Destination Waypoint

Tabla 2.2.Sentence Identifiers

II.II.III Formato General de las Sentencias

Todos los datos se transmiten en forma de oraciones, frases o sentencias en formato ASCII. Cada sentencia inicia con el símbolo "$" y termina con <CR><LF> es decir <retorno de carro><Salto de línea>. Se cuentan con tres tipos de sentencias básicas. Sentencias de Envió "Talker Sentences", Sentencias de Propiedad "Propietary sentences" y Sentencias de Consulta "Query Sentences".

- Sentencias de Envió "Talker Sentences":

  El formato general de este tipo de sentencias es:

  $TTSSS, D1,D2,….<CR><LF>

  Donde TT es el identificador del equipo transmisor, SSS es el identificador de la sentencia, D1,D2,…., son campos de datos que son específicos o predefinidos dependiendo de la sentencia usada. Como parte de los campos de datos, se agrega un campo adicional checksum para control, el cual esta identificado por el símbolo "*"y dos dígitos hexadecimales que representar la OR exclusiva de todos los caracteres excluyendo $ y *. La sentencia termina con <CR><LF>.

Un ejemplo de esta sentencia seria:

$HCHDM,238,M<CR><LF>

"HC"	Es el Identificador del equipo transmisor HC quiere decir "Heading - Magnetic Compass", que traducido es Compás Magnético
"HDM"	Determina el Identificador de la Sentencia el cual es "Heading-Magnetic"
"238"	Es el valor en grados
"M"	M=Magnético

Si los datos de un campo no existen, se omite el campo, pero la separación por comas se sigue enviando sin espacio entre ellos

- Sentencias de Propiedad "Propietary sentences"

  La norma permite a los fabricantes definir formatos de sentencias propietarias, estas frases comienzan con $P

  Un ejemplo de estas sentencias propietarias seria la correspondiente a la empresa Garmin con la que obtiene la Velocidad 3D.

  $PGRMV,XX,YY,ZZ*HH<CR><LF>

XX	Velocidad Este m/s
YY	Velocidad Norte m/s
ZZ	Velocidad UP m/s
*HH	CheckSum

- Sentencias de Consulta "Query Sentences"

  Este tipo de sentencias, son necesarias para pedir a los dispositivos una sentencia en particular. Por ejemplo:

  CCGPQ $,GCA<CR><LF>

  Donde el dispositivo Solicitante "CC" (ordenador), le esta requiriendo al dispositivo "GP" (Unidad de GPS), la sentencia "GGA". Una vez enviada dicha instrucción, el GPS transmitirá esta sentencia una vez por segundo hasta que una consulta diferente sea requerida

De la tabla 2.2. Identificadores de Sentencias, las sentencias más usadas para dispositivos GPS son los siguientes:

GPAAM	Waypoint Arrival Alarm	GPRMA	Recommended Loran data
GPALM	Almanac data	GPRMB	Recommended navigation data for gps
GPAPA	Auto Pilot A sentence	GPRMC	Recommended minimum data for gps
GPAPB	Auto Pilot B sentence	GPRTE	route message
GPBOD	Bearing Origin to Destination	GPTRF	Transit Fix Data
GPBWC	Bearing using Great Circle route	GPSTN	Multiple Data ID
GPDTM	Datum being used	GPVBW	dual Ground / Water Spped
GPGGA	Fix information	GPVTG	Vector track an Speed over the Ground
GPGLL	Lat/Lon data	GPWCV	Waypoint closure velocity (Velocity Made Good)
GPGRS	GPS Range Residuals	GPWPL	Waypoint Location information
GPGSA	Overall Satellite data	GPXTC	cross track error
GPGST	Pseudorange Noise Statistics	GPXTE	measured cross track error
GPGSV	Detailed Satellite data	GPZTG	Zulu (UTC) time and time to go (to destination)
GPMSK	Send control for a beacon receiver	GPZDA	Date and Time
GPMSS	Beacon receiver status information		

Las siguientes sentencias son soportadas por algunos receptores de GPS

HCHDG	Compass Output	PSLIB	Remote Control for DGPS receivers

Asimismo, tenemos algunas sentencias propietarias de los receptores Garmin.

PGRME	Estimated Error	PGRMZ	Altitude
PGRMM	Map Datum	PSLIB	Beacon receive control

II.III. Descodificación de algunas sentencias NMEA

II.III.I. Sentencias de Posición

$GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,*47

GP	Global Positioning System (GPS)	
GGA	Global Positioning system Fix Date, Time, Position and fix related data for a GPS receiver	
123519	Time UTC	12:35:19 UTC
4807.038	Latitude	49 deg 07.038’
N	N=North S=South)	North
01131.000	Longitude	11 deg 31.000’
E	E=East W=West	East
1	Fix Quality: 0=Invalid 1=GPS Fix 2=DGPS Fix 3=PPS fix 4=Real Time Kinematic 5=Float RTK 6=Estimated 7=Manual Input Mode 8=Simulation mode	GPS Fix
08	Number of satellites being tracked	8 satellites (nota 1)
0.9	Horizontal Dilution of Position	0.9 HDOP
545.4	Altitude, above mean sea level	545.4 meters
M	Units of altitude	Meters
46.9	Height of geoid (mean sea level) Above WGS84 elipsoid	46.9 meters
M	Units of Height	Meters
(empty)	time in seconds since last DGPS update	
(empty)	DGPS station ID number	
*47	CheckSum date, always begins with *	

Nota 1. Recordemos que para el cálculo de trilateracion, es necesario tener por lo menos 4 satelites en vista

$GPGSA,A,3,04,05,,09,12,,,24,,,,,2.5,1.3,2.1*39

GP	Global Positioning System (GPS)	
GSA	GPS DOP and Active Satellites	
A	Selection Mode A=Autoselection of 2D, 3D M=Manual	Autoselection
3	Mode 1=No fix 2=2D fix 3=3D fix	3D Fix
04,05,,09…	Ident n of 12 satellite used for fix	
2.5	PDOP in meters	2.5m PDOP
1.3	HDOP in meters	1.3m HDOP
2.1	VDOP in meters	2.1m VDOP
*39	Checksum data	

$GPGSV,2,1,08,01,40,083,46,02,17,308,41,12,07,344,39,14,22,228,45*75

GP	Global Positioning System (GPS)
GSV	Satellites in View
2	Number of sentences
1	Sentence 1 of 2
08	Number of Satellites in view
01	Satellite PRN (id) number
40	Elevation in degreess
083	Azimuth in degrees
46	SNR in dB-higher is better
More satellites infos	
*75	Checksum

$GPRMC,123519,A,4807.038,N,01131.000,E,022.4,084.4,230394,003.11,W*6A

GP	Global Positioning System (GPS)	
RMC	Recommended Minimum Sentence C	
123519	Time UTC	12:35:19 UTC
A	Status	Active
	A=Active	
	V=Void	
4807.038	Latitude	49 deg 07.038’
N	N=North	North
	S=South)	
01131.000	Longitude	11 deg 31.000’
E	E=East	East
	W=West	
022.4	Speed over the ground in Knots	022.4 Knots
084.4	Track angle in degrees True	084.4 deg
230394	Date	23/03/94
003.1	Magnetic variation in degrees	003.1 deg
W	E=East	West
	W=West	
*6A	Checksum	

$GPGLL,4916.45,N,12311.12,W,225444,A,*1D

GP	Global Positioning System (GPS)	
GLL	Geographic Position – Latitude/Longitude	
4916.45	Latitude	49 deg 16.45’
N	N=North	North
	S=South)	
12311.12	Longitude	123 deg 11.12’
W	E=East	West
	W=West	
225444	Fix taken at time UTC	22:54:44
A	Status	Active
	A=Active	
	V=Void	
*1D	Checksum	

$GPVTG,054.7,T,034.4,M,005.5,N,010.2,K*48

GP	Global Positioning System (GPS)
VTG	Track Made Good and Ground Speed
054.7,T	True track made good in degrees
034.4,M	Magnetic Track made good
005.5,N	Ground Speed in Knots (N=Knots)
010.2,K	Ground Speed in Km per hour (K=KMxH)
*48	Checksum

II.III.II. Sentencias de Navegación

$GPWPL,4807.038,N,01131.000,E,WPTNME*5C

GP	Global Positioning System (GPS)
WPL	Waypoint Location
4807.038,N	Latitude
01131.000,E	Longitude
WPTNME	Waypoint Name
*5C	Checksum

$GPAAM,A,A,0.10,N,WPTNME*32

GP	Global Positioning System (GPS)
AAM	Arrival Alarm
A	Arrival circle entered
A	Perpendicular passed
0.10	Circle radius
N	Nautical miles
WPTNME	Waypoint name
*32	Checksum

$GPAPB,A,A,0.10,R,N,V,V,011,M,DEST,011,M,011,M*3C

GP	Global Positioning System (GPS)
APB	Autopilot format B
A	Loran-C blink/SNR warning, general warning
A	Loran-C cycle warning
0.10	Cross-track error distance
R	Steer Right to correct (or L for Left)
N	Cross-track error units - nautical miles (K for kilometers)
V	Arrival alarm - circle
V	Arrival alarm - perpendicular
011,M	Magnetic bearing, origin to destination
DEST	Destination waypoint ID
011,M	Magnetic bearing, present position to destination
011,M	Magnetic heading to steer (bearings could True as 033,T)
*3C	Checksum

$GPBOD,045.,T,023.,M,DEST,START*01

GP	Global Positioning System (GPS)
BOD	Bearing - origin to destination waypoint
045,T	Bearing 045 True from "START" to "DEST"
023.,M	Bearing 023 Magnetic from "START" to "DEST"
DEST	Destination waypoint ID
START	Origin waypoint ID
*01	Checksum

$GPBWC,225444,4917.24,N,12309.57,W,051.9,T,031.6,M,001.3,N,004*29

GP	Global Positioning System (GPS)
BWC	Bearing and distance to waypoint - great circle
225444	UTC time of fix 22:54:44
4917.24,N	Latitude of waypoint
12309.57,W	Longitude of waypoint
051.9,T	Bearing to waypoint, degrees true
031.6,M	Bearing to waypoint, degrees magnetic
001.3,N	Distance to waypoint, Nautical miles
004	Waypoint ID
*29	Checksum

$GPRMB,A,0.66,L,003,004,4917.24,N,12309.57,W,001.3,052.5,000.5,V*20

GP	Global Positioning System (GPS)
RMB	Recommended minimum navigation information
A	Data status A = OK, V = Void (warning)
0.66,L	Cross-track error (nautical miles, 9.99 max), L=Left, R=Right
003	Origin waypoint ID
004	Destination waypoint ID
4917.24,N	Destination waypoint latitude
12309.57,W	Destination waypoint longitude
001.3	Range to destination, nautical miles (999.9 max)
052.5	True bearing to destination
000.5	Velocity towards destination, knots
V	Arrival alarm A = arrived, V = not arrived
*20	checksum

$GPRTE,2,1,c,0,W3IWI,DRIVWY,32CEDR,32-29,32BKLD,32-I95,32-US1,BW-32,BW-198*69

GP	Global Positioning System (GPS)
RTE	Waypoints in active route
2	Total number of sentences needed for full data
1	This is sentence 1 of 2
c	Type c = complete list of waypoints in this route w = first listed waypoint is start of current leg
0	Route identifier
W3IWI,...	Waypoint identifiers (names)
*69	Checksum

$GPXTE,A,A,0.67,L,N*6F

GP	Global Positioning System (GPS)
XTE	Cross track error, measured
A	General warning flag V = warning (Loran-C Blink or SNR warning)
A	Not used for GPS (Loran-C cycle lock flag)
0.67	Cross track error distance
L	Steer left to correct error (or R for right)
N	Distance units - Nautical miles
*6F	Checksum

II.III.III. Otro Sentencias necesarias y/o propietarias

$GPALM,A.B,C.D,E,F,hh,hhhh,...

GP	Global Positioning System (GPS)
ALM	Almanac Data being sent
A	Total number of messages
B	Message number
C	Satellite PRN number
D	GPS week number (0-1023)
E	Satellite health (bits 17-24 of message)
F	Eccentricity
hh	t index OA, almanac reference time
hhhh	Sigma index 1, inclination angle
...	OMEGADOT rate of right ascension
	SQRA(A) root of semi-major axis
	Omega, argument of perigee
	Omega index 0, longitude of ascension node
	M index 0, mean anomaly
	a index f0, clock parameter
	a index f1, clock parameter

$PGRME,15.0,M,45.0,M,25.0,M*1C

PGRME	Sentencia Propietaria Garmin
15.0,M	Estimated horizontal position error in meters (HPE)
45.0,M	Estimated vertical error (VPE) in meters
25.0,M	Overall spherical equivalent position error

$PSNY,0,00,05,500,06,06,06,06*14

PSNY	Sentencia Propietaria Sony
0	Preamp (external antenna) status
	0 = Normal
	1 = Open
	2 = shorted
00	Geodesic system (datum) 0-25, 0 = WGS84
05	Elevation mask in degrees
500	Speed Limit in Km
06	PDOP limit with DGPS on
06	HDOP limit with DGPS on
06	PDOP limit with DGPS off
06	HDOP limit with DGPS off
*14	Checksum

CAPITULO III. DISEÑO DEL SOFTWARE DEL DISPOSITIVO MOVIL

III.I. Análisis de Requerimientos

Se cuenta los siguientes recursos de equipo y software para realizar el Sistema, los cuales se tiene las siguientes características:

- Dispositivo Portátil HTC Mogul
    - con Cpu Qualcomm 7500 (MSM 7500), 400Mhz, RAM 64 MB, Flash 256 MB, expansión de 2 GB, resolución de pantalla de 240 X 320, modelo TITA100
    - Windows Mobile 6.1.
    - Office Mobile
    - Bluetooth, GPS, Cámara de 2MP, WIFI y acceso a Internet (EVDO), tecnología CDMA.
    - Microsoft Active Sync 4.5

- Plan de renta con la empresa Iusacell, el cual incluye Navegación 3G en Internet limitado (EVDO)

- Dominio y Hosting en www.elaion.com.mx con las siguientes características

    - 1 GB Almacenamiento en disco duro
    - Trafico de 3MB/Mes
    - Soporte para 10 bases de Datos Mysql (Client API Version 5.0.58)
    - Soporte PHP
    - Soporte Apache (Apache API Versión 20051115)
    - Panel de Control y administración en https://elaion.com.mx:8443

Con la disponibilidad de estos recursos, el sistema debe ser diseñado para obtener los siguientes resultados:

- La aplicación del Dispositivo Móvil, será capaz de interactuar con el módulo GPS que trae integrado y su operación tendrá las siguientes características:

    - Obtener la posición global referente a los parámetros de Latitud, Longitud y Velocidad.
    - Obtener información adicional como margen de error en la información de posición, información acerca de características de las señales emitidas, numero de Satélites, fecha y hora.
    - Enviar la posición global a través de Internet para su monitoreo o rastreo remoto.
    - Que en caso de que el usuario lo decida, el sistema podrá enviar dicha posición Global a través de un mensaje SMS a un teléfono predeterminado.
    - Configuración manual de los parámetros del módulo GPS
    - Identificación del Equipo, mediante un nombre asignado por el usuario

- La aplicación para WEB, deberá tener las siguientes características

    - Mostrar las posiciones registradas que el dispositivo móvil le envió para su rastreo y monitoreo.
    - Dichas posiciones deberán ordenarse por nombre del dispositivo y por día
    - Su Posición Global, deberá ser en modo grafico a través de imágenes o mapas digitales

Con los siguientes recursos disponibles, emplearemos el método de diseño de software que se expone a continuación es el método de solución de problemas mediante técnicas de ingeniería.

- Definición o Análisis del Problema

    En esta etapa, también conocida como Especificación de Requerimientos, se establece el problema, aclarándolo lo más posible. Es la parte más crítica de la solución. Amerita un estudio cuidadoso. Se deben identificar las teorías, fundamentos y/o principios matemáticos, físicos o de cualquier índole que permitan fundamentar satisfactoriamente el problema. Se deben eliminar los aspectos poco importantes para el planteamiento del problema

- Diseño

  Desarrollar una lista de pasos llamados algoritmo para la solución, verificando que el problema se resuelve. Es la parte más difícil del proceso de solución del problema Debe verificarse que es correcto el algoritmo antes de continuar Se auxilia de técnicas de diseño

- Implementación

  Esta etapa consiste en implementar o escribir el algoritmo como un programa de computadora en un lenguaje de programación, convirtiendo cada paso del algoritmo en instrucciones en el lenguaje de programación.

  Se requiere el conocimiento de un lenguaje de programación particular en lo referente a su gramática, sintáxis y semántica, para ello se recomienda leer el manual del programador o su equivalente y utilizarlo como consulta siempre que sea necesario.

- Verificación y Prueba

  Esta etapa consiste en probar el programa completo y verificar que trabaja como se esperaba. Se deben probar cada una de las funciones primero por separado y luego en conjunto. Se debe probar el programa completo con distintos conjuntos de datos de prueba. En caso de que haya errores repetir el modelo hasta la satisfacción de los requerimientos

III.II. Plataforma de Desarrollo

Con base en el análisis de la información, se modela la arquitectura de la solución y su interrelación (Imagen 3.1.)

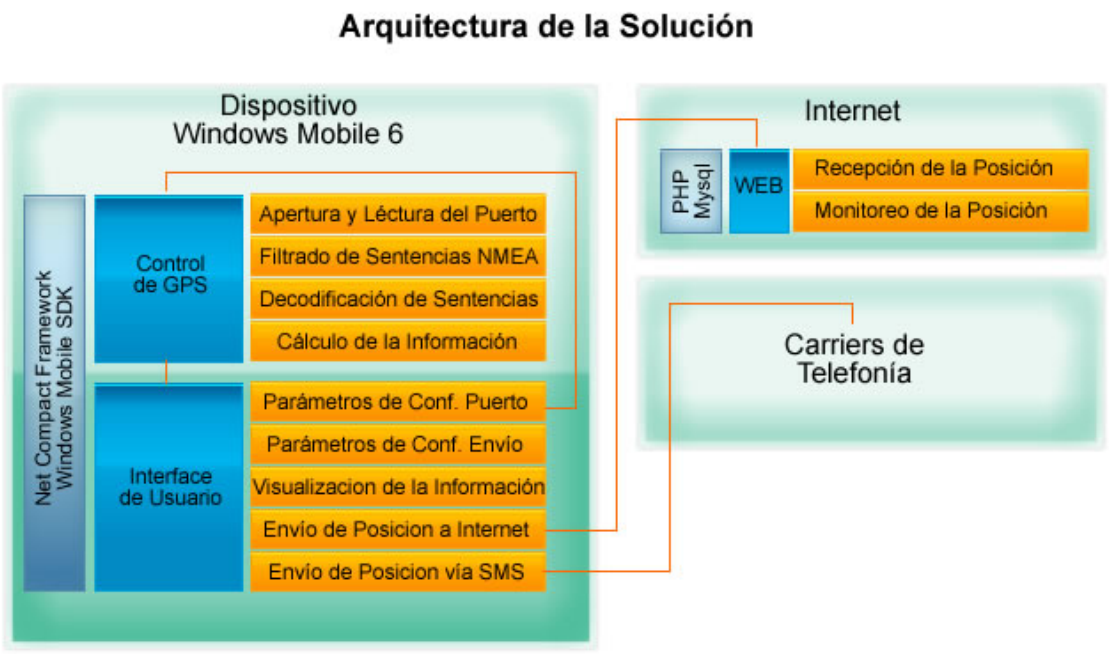


Imagen 3.1. Arquitectura de la Solución

Con base en esta Arquitectura, utilizaremos el siguiente Software

- Microsoft Visual Basic 2005, con Microsoft Net Compact Framework 2.0
- Windows Mobile 6 SDK
- PHP
- Mysql
- Apache

III.III. Módulo de Configuración

En algunos casos, los dispositivos móviles no traen integrado el receptor de GPS y adquieren por separado dichos receptores los cuales se alimentan del dispositivo y se conectan a través de bluetooth.

Ya sea que tenga o no incorporado el receptor de GPS, el puerto disponible para su lectura, es muy variado, al igual que la velocidad de transferencia.

Por lo tanto la función principal del módulo es que permita a cada dispositivo modificar sus parámetros los cuales son:

- Puerto asignado al GPS
- BaudRate (Velocidad de Transferencia)
- DataBits (Bits de Datos)
- Parity (Paridad)
- Stop Bits (Bits de Parada)
- HandShake

De la misma manera, cada dispositivo deberá tener un nombre para su identificación en Internet, y de un numero telefónico para el envió de mensajes SMS. Por lo que en este mismo módulo se podrán cambiar dichos parámetros

Todos estos parámetros o valores, se almacenaran para su actualización y lectura en un archivo de texto plano (txt), llamado gps.txt, el cual tiene al final del archivo la sentencia FIN, el cual le dirá al interprete el momento en que dejara de leer las líneas.

De la misma manera si encuentra la palabra PORT, traerá todos los parámetros de configuración de GPS. Y si encuentra la palabra EEQUIPO, traerá los parámetros de nombre de equipo y numero de teléfono para envió de SMS. Las siguientes líneas muestran un ejemplo típico del contenido del archivo gps.txt

```
PORT,4,BAUDRATE,4800,DATABITS,8,PARITY,0,STOPBITS,1,HANDSHAKE,0
EEQUIPO,HTCMogul,ESMS,55
Parity.None = 0
Parity.Odd = 1
Parity.Even = 2
Parity.Mark = 3
Parity.Space = 4
StopBits.None = 0
StopBits.One = 1
StopBits.Two = 2
StopBits.OnePointFive = 3
Handshake.None = 0
Handshake.XOnXOff = 1
Handshake.RequestToSend = 2
Handshake.RequestToSendXOnXOff = 3
FIN
```

La interfase de usuario en modo diseño para esta sección o módulo se presenta en la imagen 3.1.

Imagen 3.1. Sección de Configuración

Como norma, de la solución extraeremos solo el código correspondiente al módulo. Para ver el código completo, diríjase a la sección de anexos.

En el evento de Form Load, tenemos el siguiente código

```
    Private Sub frmGPSWEB_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Dim Parametros() As String

......
        'Lee un archivo de Configuraciones iniciales,
        'las cuales dependen de cada dispositivo

        'En este caso, lee la linea del archivo que empieza con
        'La cadena PORT la cual tiene el siguiente format.
        'Port,4,BaudRate,4800,DataBits,8,Parity,0,StopBits,1,HandShake,0
        Parametros = LeeArchivoTexto("Program Files/GPSWEB/gps.txt", "PORT")
        'Esta funcion me devuelve una matriz con los valores de la cadena
        'Parametro("PORT","4",BaudRate,4800,DataBits,8,....)
        'El archivo tiene que tener una linea con la Etiqueta FIN, para
        'poder detener la lectura
        If Parametros(0) <> "FIN" Then
            'Si el contenido del archivo no tiene la etiqueta FIN
            'entonces inicializa los Objetos de la forma con los valores
            'contenidos en la matriz
            cboPort.SelectedIndex = CInt(Parametros(1))
```

```
            txtBaudRate.Text = Parametros(3)
            txtDataBits.Text = Parametros(5)
            cboParity.SelectedIndex = CInt(Parametros(7))
            cboStopBits.SelectedIndex = CInt(Parametros(9))
            cboHandshake.SelectedIndex = CInt(Parametros(11))
        Else
            'Si no encuentra los parametros configurados,
            'Le pone parametros Iniciales a los objetos por default
            'Para que despues el usuario los modifique y presione el boton
            'Grabar Configuración
            cboPort.SelectedIndex = 4
            txtBaudRate.Text = "4800"
            txtDataBits.Text = "8"
            cboParity.SelectedIndex = 0
            cboStopBits.SelectedIndex = 1
            cboHandshake.SelectedIndex = 0
        End If
        'De la misma manera que con PORT, ahora leemos la cadena EEQUIPO
        'Esta cadena tiene el nombre del equipo y el telefono predeterminado
        'para envio de SMS
        Parametros = LeeArchivoTexto("Program Files/GPSWEB/gps.txt", "EEQUIPO")
        If Parametros(0) <> "FIN" Then
            txtEEquipo.Text = Parametros(1)
            txtESMS.Text = Parametros(3)
        Else
            'Si no encuentra los parametros configurados,
            'Le da pone parametros Iniciales por default
            txtEEquipo.Text = "123456789012345"
            txtESMS.Text = "55"
        End If

        'Abrimos el Puerto con las configuraciones en los objetos de la forma
        Try
            If SerialPort1.IsOpen Then
                'Si esta abierto el puerto lo cierra
                SerialPort1.Close()
            Else
                'Establece los parametros y abre el puerto
                SerialPort1.PortName = cboPort.SelectedItem.ToString
                SerialPort1.BaudRate = CInt(txtBaudRate.Text)
                SerialPort1.DataBits = CInt(txtDataBits.Text)
                SerialPort1.Parity = CType(cboParity.SelectedIndex,
System.IO.Ports.Parity)
                SerialPort1.StopBits = CType(cboStopBits.SelectedIndex,
System.IO.Ports.StopBits)
                SerialPort1.Handshake = CType(cboHandshake.SelectedIndex,
System.IO.Ports.Handshake)
                SerialPort1.Open()
            End If
        Catch ex As Exception
            ' Si no puede abrir el puerto, envia el mensaje
            MessageBox.Show("No se ha podido Abrir el Puerto: " &
cboPort.SelectedItem.ToString, "Atención")
            Exit Sub
        End Try
    End Sub
```

La función para abrir el archivo y devolver la cadena buscada es el siguiente, el cual esta dentro de un módulo.

```
    'Lee las configuraciones Iniciales
    Public Function LeeArchivoTexto(ByVal Archivo As String, ByVal Encabezado As String)
As String()
        'Recibe como parametros el Nombre del Archivo, y la cadena buscar
        ' Si la encuentra, trae la linea y la almacena en un vector, el cual devuelve
        Dim Parametros() As String
        Dim Cadena As String
        Cadena = "FIN,,"
        'Carga el fichero
        If System.IO.File.Exists(Archivo) = False Then
            MsgBox("El archivo " & Archivo & " no existe", MsgBoxStyle.OkOnly)
        Else
            Dim sr As New System.IO.StreamReader(Archivo, System.Text.Encoding.Default,
True)
            While sr.Peek() <> -1
```

```
                Cadena = sr.ReadLine
                ' Si reconoce la cadena de configuracion del puerto, la extrae
                ' e inicializa los objetos asociados
                ' La cadena recibida puede tener el siguiente Formato entre otras
                ' Port,1,BaudRate,4800,DataBits,8,Parity=None,StopBits,1,HandShake,None
                If Mid(Cadena, 1, Len(Encabezado)) = Encabezado Then
                    Exit While
                End If
            End While
            sr.Close()
        End If
        'Esta funcion recibe la cadena y nos devuelve la cadena pero en matriz
        Parametros = DivideSentencia(Cadena)
        Return Parametros
    End Function

    'Esta funcion nos devuelve una matriz unidimensional basada en cero que
    'contiene un número especificado de subcadenas
    'Es decir nos devuelve un elemento en el arreglo por cada palabra separada
    ' por comas en la sentencia
    Public Function DivideSentencia(ByVal sentencia As String) As String()
        Return Split(sentencia, ",")
    End Function
```

Ya que el puerto fue configurado y los objetos tienen los parámetros o valores del archivo, el usuario puede ver sus parámetros de configuración, modificarlos y darle clic al boton de GRABAR CONFIGURACION, el cual desencadena el siguiente evento

```
....
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdGrabar.Click
        Dim oSW As New System.IO.StreamWriter("Program Files/GPSWEB/gps.txt", False,
System.Text.Encoding.Default)
        '        Dim sr As New System.IO.StreamWriter("Program Files/GPSWEB/gps.txt",
System.Text.Encoding.Default, True)

        Dim Linea As String
        'Necesitamos el inico de la forma
        'Imports System.IO
        'Port,4,BaudRate,4800,DataBits,8,Parity,0,StopBits,1,HandShake,0
        Linea = ""
        Linea = Linea & "PORT," & cboPort.SelectedIndex
        Linea = Linea & ",BAUDRATE," & Trim(txtBaudRate.Text)
        Linea = Linea & ",DATABITS," & Trim(txtDataBits.Text)
        Linea = Linea & ",PARITY," & cboParity.SelectedIndex
        Linea = Linea & ",STOPBITS," & cboStopBits.SelectedIndex
        Linea = Linea & ",HANDSHAKE," & cboHandshake.SelectedIndex
        oSW.WriteLine(Linea)
        Linea = ""
        Linea = Linea & "EEQUIPO," & Trim(txtEEquipo.Text)
        Linea = Linea & ",ESMS," & Trim(txtESMS.Text)
        oSW.WriteLine(Linea)

        'Estas lineas siempre va al final como Informacion
        Linea = "Parity.None = 0"
        oSW.WriteLine(Linea)
        Linea = "Parity.Odd = 1"
        oSW.WriteLine(Linea)
        Linea = "Parity.Even = 2"
        oSW.WriteLine(Linea)
        Linea = "Parity.Mark = 3"
        oSW.WriteLine(Linea)
        Linea = "Parity.Space = 4"
        oSW.WriteLine(Linea)
        Linea = "StopBits.None = 0"
        oSW.WriteLine(Linea)
        Linea = "StopBits.One = 1"
        oSW.WriteLine(Linea)
        Linea = "StopBits.Two = 2"
        oSW.WriteLine(Linea)
        Linea = "StopBits.OnePointFive = 3"
        oSW.WriteLine(Linea)
        Linea = "Handshake.None = 0"
        oSW.WriteLine(Linea)
        Linea = "Handshake.XOnXOff = 1"
```

```
            oSW.WriteLine(Linea)
            Linea = "Handshake.RequestToSend = 2"
            oSW.WriteLine(Linea)
            Linea = "Handshake.RequestToSendXOnXOff = 3"
            oSW.WriteLine(Linea)
            Linea = "FIN"
            oSW.WriteLine(Linea)
            oSW.Close()
            MsgBox("Actualización Correcta", MsgBoxStyle.OkOnly, "Atención")
        End Sub
.....
```

III.IV. Módulo de Adquisición de Datos

Ya que se tiene comunicación con el receptor de GPS, el módulo de adquisición de datos, deberá comunicarse con el receptor de GPS y traer los valores siguientes valores y almacenarlos en sus objetos o variables respectivas.

- Información de posición y velocidad:
  - Latitud
  - Longitud
  - Velocidad

- Información de Satélites:
  - Numero de Satélites visibles
  - PRN (Pseudos-Random Noise)
  - Elevación
  - Azimuth
  - SNR (Signal to Noise Ratio)

- Fecha y Hora del Reloj Atómico

- Dilución de la Presición DOP
  - HDOP (Horizontal Dilution of Precision)
  - VDOP (Vertical Dilution of Precision)

Las interfases de usuario en modo diseño para estas secciones o módulos se presenta en las imágenes 3.2. y 3.3.

Imagen 3.2. Sección de Control

Imagen 3.2. Sección de mas Información

Cuando la forma fue inicializada, se configuraron valores de variables, las cuales mostramos algunas dentro del código del evento Form Load

```
.....
Public Class frmGPSWEB
    'Definimos estas variables para conocer si tenemos datos validos
    'de estas sentencias y salir del ciclo de lectura del puerto
    Public bRMC As Boolean
    Public bGSV As Boolean
    Public bGSA As Boolean
    Private Sub frmGPSWEB_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Dim Parametros() As String
        'Se inicializan los valores de los objetos en la interface de usuario
        bRMC = False
        bGSV = False
        bGSA = False

        TabControl1.TabPages(0).Text = "Control"
        TabControl1.TabPages(1).Text = "Mas Info."
        TabControl1.TabPages(2).Text = "Configuraciones"
        TabControl1.TabPages(0).Show()

        txtSentenciaNMEA.Text = ""

        lblLatitude.Text = ""
        lblLatitudeS.Text = ""
        lblLongitude.Text = ""
```

```
        lblLongitudeS.Text = ""
        lblVelocidad.Text = "0 KmH"

        lblNumSat.Text = ""
        lblid1.Text = ""
        lblel1.Text = ""
        lblaz1.Text = ""
        lblsr1.Text = "0 KmH"

        lblid2.Text = ""
        lblel2.Text = ""
        lblaz2.Text = ""
        lblsr2.Text = "0 KmH"

        lblid3.Text = ""
        lblel3.Text = ""
        lblaz3.Text = ""
        lblsr3.Text = "0 KmH"

        lblid4.Text = ""
        lblel4.Text = ""
        lblaz4.Text = ""
        lblsr4.Text = "0 KmH"

        lblReloj.Text = ""

        lblHDOP.Text = ""
        lblVDOP.Text = ""
……
    End Sub
```

Este módulo inicia cuando el usuario le da clic al botón de OBTENER DATOS GPS y desencadena el interprete de tipos de sentencia NMEA

...

```
    Private Sub cmdObtener_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdObtener.Click
        Dim Dato As String
        Dim contador As Integer
        Dim Bandera As Boolean
        bRMC = False
        bGSV = False
        bGSA = False

        Bandera = False
        contador = 0
        txtSentenciaNMEA.Text = ""
        Dato = ""
        '        txtMensaje.Text = "Entro Click"
        While Bandera = False
            If SerialPort1.IsOpen Then
                Try
                    Dato = Trim(Mid(SerialPort1.ReadLine, 1, 250))
                    If Len(Trim(Dato)) <> 0 Then
                        'txtSentenciaNMEA.Text = Mid(Dato, 1, Len(Trim(Dato)) - 1)
                        txtSentenciaNMEA.Text = Mid(Dato, 1, Len(Trim(Dato)) - 1) &
ControlChars.CrLf & txtSentenciaNMEA.Text

                        IdentificadorNMEA(Mid(Dato, 1, Len(Trim(Dato)) - 1))
                        ' Se sale del ciclo hasta que las tres sentencias sentencias
hayan traido datos
                        If bRMC And bGSV And bGSA Then
                            Bandera = True
                        End If
                    End If
                Catch ex As Exception
                    'Continua el Ciclo
                End Try

            End If
            contador = contador + 1
            If contador = 50 Then
                txtSentenciaNMEA.Text = ""
            End If
            If contador >= 100 Then
```

```
                If Len(Trim(lblLatitude.Text)) = 0 Then
                    MessageBox.Show("Señal GPS Débil", "Atención")
                End If
                Bandera = True
            End If
        End While

    End Sub
…..
    Private Function IdentificadorNMEA(ByVal sentencia As String) As Boolean
        'Antes de Identificar la sentencia, se procede a checar que el CHECKSUM,
        'Sea igual a Xor de todos los caracteres
        If Not ChecksumValido(sentencia) Then
            Return False
        End If
        'Comparamos el primer elemento de la Sentencia
        Select Case DivideSentencia(sentencia)(0)
            Case "$GPRMC"
                'RMC      Recommended Minimum Sentence C
                If DecodificacionGPRMC(sentencia) Then
                    Return True
                Else
                    Return False
                End If
            Case "$GPGSV"
                'GSV  Satellites in View
                If DecodificacionGPGSV(sentencia) Then
                    Return True
                Else
                    Return False
                End If
            Case "$GPGSA"
                'GSA      GPS DOP and Active Satellites
                If DecodificacionGPGSA(sentencia) Then
                    Return True
                Else
                    Return False
                End If
            Case Else
                ' cualquier Otra Sentencia
                Return False
        End Select
    End Function

    Private Function DecodificacionGPRMC(ByVal sentencia As String) As Boolean
        Dim Parametros() As String = DivideSentencia(sentencia)
        '$GPRMC,123519,A,4807.038,N,01131.000,E,022.4,084.4,230394,003.11,W*6A
        '0  GP            Global Positioning System (GPS)
        '   RMC           Recommended Minimum Sentence C
        '1  123519           Time UTC                                12:35:19 UTC
        '2  A                Status,A = Active, V = Void
        '3  4807.038    Latitude
        '4  N                    N=North, S=South)
        '5  01131.000 Longitude
        '6  E                E=East, W = West
        '7  022.4            Speed over the ground in Knots
        '8  084.4            Track angle in degrees True
        '9  230394           Date
        '10 003.1            Magnetic variation in degrees
        '11 W                    E=East,W = West
        '11 *6A           Checksum

        'YA que reconocimos la sentecia GPRMC y la funcion de los parametros,
calcularemos los
        'datos sigientes

        'LATITUD Y LONGITUD

        'En esta parte verificamos que los parametros para obtener Latitud y Longitud no
esten vacios
        If Parametros(3) <> "" And Parametros(4) <> "" And Parametros(5) <> "" And
Parametros(6) <> "" Then

            'Extraemos y convertimos Latitud en grados minutos y segundos
            Dim Latitude As String
            'Horas
            Latitude = Mid(Parametros(3), 1, 2) & " "
```

```
            'Horas & Minutos
            Latitude = Latitude & Mid(Parametros(3), 3, 2) & " "
            'Horas & Minutos & Segundos
            Latitude = Latitude & Mid(CStr(CDbl(Mid(Parametros(3), 5, 6)) * (60)), 1, 5)
            'Horas & Minutos $ Segundos & Hemisferio
            Latitude = Latitude & " " & Parametros(4)

            'Extraemos y convertimos Longitud en grados minutos y segundos
            Dim Longitude As String
            'Horas
            Longitude = Mid(Parametros(5), 1, 3) & " "
            'Horas & Minutos
            Longitude = Longitude & Mid(Parametros(5), 4, 2) & " "
            'Horas & minutos & segundos
            Longitude = Longitude & Mid(CStr(CDbl(Mid(Parametros(5), 6, 6)) * (60)), 1,
5)
            'Horas & Minutos $ Segundos & Hemisferio
            Longitude = Longitude & " " & Parametros(6)

            'Ponemos los calculos en los objetos para el usuario
            lblLatitude.Text = Latitude
            lblLatitudeS.Text = Parametros(3) & Parametros(4)

            lblLongitude.Text = Longitude
            lblLongitudeS.Text = Parametros(5) & Parametros(6)

        Else
            Return False
        End If
        'Cada satélite GPS contiene cuatro relojes atómicos que se utiliza para las
        'transmisiones. Por lo que estos relojes atómicos se puede utilizar para
sincronizar
        'el reloj del dispositivo con una precisión de milisegundos.
        If Parametros(1) <> "" Then
            Dim UtcHours As Integer = CType(Parametros(1).Substring(0, 2), Integer)
            Dim UtcMinutes As Integer = CType(Parametros(1).Substring(2, 2), Integer)
            Dim UtcSeconds As Integer = CType(Parametros(1).Substring(4, 2), Integer)
            Dim UtcMilliseconds As Integer
            ' Si tiene el parametro milisegundos se extraen
            If Parametros(1).Length > 7 Then UtcMilliseconds =
CType(Parametros(1).Substring(7), Integer)
            Dim Today As DateTime = System.DateTime.Now.ToUniversalTime
            Dim SatelliteTime As New System.DateTime(Today.Year, Today.Month, Today.Day,
UtcHours, UtcMinutes, UtcSeconds, UtcMilliseconds)
            'Traemos la fecha y la conversion de la hora UTC con la hora local
            'configurada en el dispositivo
            lblReloj.Text = CType(SatelliteTime.ToLocalTime(), String)
        End If
        ' VELOCIDAD
        ' La informacion esta en Knots, por lo que basta cob multiplicarlo
        ' por 1,852 para convertirla a KMH
        If Parametros(7) <> "" Then
            '1 Knot=1.852KmH
            '1 Knot=1.150779448MillasxH
            Dim Velocidad As Double
            Velocidad = CDbl(Parametros(7)) * 1.852
            lblVelocidad.Text = Trim(CStr(Velocidad)) & " KmH"
        End If

        ' SUFICIENCIA EN EL CALCULO
        'Hemos visto que cuando se cuentan con al menos 3 satelites, el margen de error
        'puede ser grande, por lo tanto esta sentencia tiene un parametro (2) el cual
nos
        'indica si la fuerza de la señal de esos satelites es suficiente para darnos una
        'posicion viable
        If Parametros(2) <> "" Then
            Select Case Parametros(2)
                Case "A"
                    ' Return True
                Case "V"
                    Return False
            End Select
        End If
        bRMC = True
        Return True

    End Function
```

```
    Private Function DecodificacionGPGSV(ByVal Sentencia As String) As Boolean
        Dim IdNumber As String
        Dim Azimuth As String
        Dim Elevation As String
        Dim SignalToNoiseRatio As String
        Dim Parametros() As String = DivideSentencia(Sentencia)
        '$GPGSV,2,1,08,01,40,083,46,02,17,308,41,12,07,344,39,14,22,228,45*75
        '0   GP              Global Positioning System (GPS)
        '0  GSV       Satellites in View
        '1  2         Number of sentences
        '2  1         Sentence 1 of 2
        '3  08         Number of Satellites in view
        '4  01         Satellite PRN (id) number. Pseudo-Random Noise
        '5  40         Elevation in degreess
        '6  083        Azimuth in degrees
        '7  46:          SNR in(dB - higher Is better)
        '          More satellites infos
        'X  *75              Checksum

        'Por lo regular se tienen 4 bloques de informacion, cada bloque tiene
id.satelite, Elevacion, Acimut y fuera de la señal
        Dim Count As Integer
        For Count = 1 To 4
            ' Si tiene Parametros la sentencia?
            If (Parametros.Length - 1) >= (Count * 4 + 3) Then
                If Parametros(Count * 4) <> "" And Parametros(Count * 4 + 1) <> "" And
Parametros(Count * 4 + 2) <> "" And Parametros(Count * 4 + 3) <> "" Then
                    IdNumber = Parametros(Count * 4)
                    Elevation = Parametros(Count * 4 + 1)
                    Azimuth = Parametros(Count * 4 + 2)
                    SignalToNoiseRatio = Mid(Parametros(Count * 4 + 3), 1, 2)
                    'Hay veces que el ultimo parametro solo viene con el checksum
                    'por eso filtramos el checksum, y si no trae nada le ponemos 00
                    If Mid(SignalToNoiseRatio, 1, 1) = "*" Then   ' Si no tiene
                        SignalToNoiseRatio = "00"
                    End If
                    'El parametro SignalToNoiseRatio en teoria tiene el valor de 0 a 99
en donde
                    ' 0 representa una señal debil y 99 una señal Maxima
                    'Pero se ha probado este parametro en cielos totalmente despejado y
el valor maximo que se
                    'ha obtenido es 50, por lo tanto para este Proyecto se trabajara con
un maximo de 50.
                    'Si se tiene localidades donde la señal sea superior a 50, solamente
se cambiara
                    'el valor maximo del objeto de medicion
                    '
                    If Count = 1 Then
                        lblid1.Text = IdNumber
                        lblel1.Text = Elevation & " º"
                        lblaz1.Text = Azimuth & " º"
                        lblsr1.Text = SignalToNoiseRatio & " dB"
                        Pb1.Value = CType(SignalToNoiseRatio, Integer)
                    End If
                    If Count = 2 Then
                        lblid2.Text = IdNumber
                        lblel2.Text = Elevation & " º"
                        lblaz2.Text = Azimuth & " º"
                        lblsr2.Text = SignalToNoiseRatio & " dB"
                        Pb2.Value = CType(SignalToNoiseRatio, Integer)
                    End If
                    If Count = 3 Then
                        lblid3.Text = IdNumber
                        lblel3.Text = Elevation & " º"
                        lblaz3.Text = Azimuth & " º"
                        lblsr3.Text = SignalToNoiseRatio & " dB"
                        Pb3.Value = CType(SignalToNoiseRatio, Integer)
                    End If
                    If Count = 4 Then
                        lblid4.Text = IdNumber
                        lblel4.Text = Elevation & " º"
                        lblaz4.Text = Azimuth & " º"
                        lblsr4.Text = SignalToNoiseRatio & " dB"
                        Pb4.Value = CType(SignalToNoiseRatio, Integer)
                    End If
                End If
                lblNumSat.Text = Parametros(3)
```

```
            End If
        Next
        bGSV = True
        Return True
    End Function

    Public Function DecodificacionGPGSA(ByVal sentencia As String) As Boolean
        Dim Parametros() As String = DivideSentencia(sentencia)
        '$GPGSA,A,3,04,05,,09,12,,,24,,,,,2.5,1.3,2.1*39
        '0  GP        Global Positioning System (GPS)
        '0  GSA           GPS DOP and Active Satellites
        '1  A         Selection Mode A=Autoselection of  2D o 3D, M = Manual
        '2  3         Mode 1=No fix,2=2D fix,       3=3D fix
        '3-14  04,05,,09…     Ident  n of 12 satellite used for fix
        '15 2.5               PDOP in meters                    2.5m PDOP
        '16 1.3               HDOP in meters                    1.3m HDOP
        '17 2.1               VDOP in meters                    2.1m VDOP
        '17 *39               Checksum data
        If Parametros(16) <> "" Then
            lblHDOP.Text = Parametros(16) & " mts"
        End If
        If Parametros(17) <> "" Then
            lblVDOP.Text = Mid(Parametros(17), 1, InStr(Parametros(17), "*") - 1) & "
mts"
        End If
        'No se tienen rangos muy claros en cuanto a que valores son los ideales, ya que
depende del
        'tipo de Aplicacion, pero aqui se muestra una tabla que puede darnos una idea
        '1      Ideal       Este es el más alto el nivel de confianza puede ser
utilizado
        '                   para aplicaciones que demandan la máxima precisión en todo
momento.
        '2-3    Excelente   En este nivel de confianza, las mediciones de posición se
consideran lo
        '                   suficientemente precisas para cumplir con todos, pero las
aplicaciones más sensibles
        '4-6    Bueno       Representa un nivel que marca el mínimo adecuado para la
toma de decisiones
        '                   podría ser usado para hacer sugerencias en la navegación
para el usuario
        '7-20   Moderado    Solo para estimaciones
        '21o+   Pobre       Deberia descartarse
        bGSA = True
        Return True
    End Function
…..
End Class
```

Como mencionamos en el capitulo II, al final de cada sentencia NMEA, existe un parámetro llamado Checksum, el cual nos sirve para saber si nuestra sentencia leída a través del puerto esta completa y correcta.

La función de CheckSum esta en un módulo con el siguiente código

```
    'Esta funcion Checa que el parametro CheckSum coincida con el XOR de
    'todos los caracteres de la sentencia excluyendo $ y *.
    'Devolviendo True si es una sentencia Valida y false en caso contrario
    Public Function ChecksumValido(ByVal sentencia As String) As Boolean
        ' Loop through all chars to get a checksum
        Dim Checksum As Integer
        Dim Caracter As Char

        For Each Caracter In sentencia
            Select Case Caracter
                'Ignoramos el Simbolo $ y *
                Case "$"c
                Case "*"c
                    'Cuando encuentra * deja de calcular
                    Exit For
                Case Else
                    ' Calcula el XOR de los caracteres validos
                    If Checksum = 0 Then
                        Checksum = Convert.ToByte(Caracter)
                    Else
```

```
                    Checksum = Checksum Xor Convert.ToByte(Caracter)
                End If
            End Select
        Next
        ' Devuelve la comparacion entre el numero despues del simbolo * de la sentencia
        ' y el valor calculado, convertido en 2 numeros hexadecimales
        Return sentencia.Substring(sentencia.IndexOf("*") + 1) = Checksum.ToString("X2")
    End Function
```

III.V Módulo de envió de datos vía Internet

Ya que se obtuvieron los datos y que están almacenados en variables, el módulo de envió de datos vía Internet, deberá generar una sentencia de dirección URI (Uniform Resource Identifier) valido.

```
    Private Sub cmdEnviar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdEnviar.Click
        'En la pagina www.elaion.com.mx/gpsweb/, esta una pagina inspos.php que recibe
        'Los parametros de latitude, longtude, etc. y los graba en su base de datos
        Dim siteUri As New Uri("http://www.elaion.com.mx/gpsweb/inspos.php?equipo=" &
txtEEquipo.Text & "&latitud=" & lblLatitude.Text & "&longitud=" & lblLongitude.Text &
"&velocidad=" & lblVelocidad.Text & "&fecha=" & Format(Now(), "yyyy/MM/dd HH:mm:ss"))
        'Si el servidor esta fuera de linea o existen muchas transacciones pendientes,
alerta al usuario
        If WebBrowser.IsOffline Or WebBrowser.IsBusy Then
            MsgBox("No hay conexion a internet, o esta ocupado el servidor",
MsgBoxStyle.Critical, "Error de Conexion")
        Else
            WebBrowser.Navigate(siteUri)
            MsgBox("Posicion registrada correctamente", MsgBoxStyle.OkOnly, "Atención")
        End If
    End Sub
```

III.VI. Módulo de envió de posición vía SMS

Ya que se obtuvieron los datos y que están almacenados en variables, este módulo enviara un mensaje SMS al número telefónico configurado.

```
    Private Sub cmdSMS_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles cmdSMS.Click
        'Para este evento, necesitamos el Microsoft.WindowsMobile.PocketOutlook
        'Creamo el Mensaje a enviar por SMS
        Dim smsMessage As New SmsMessage()
        Dim Mensaje As String
        'Si no esta vacio y que sea que tenga mas de 8 numeros
        If Len(Trim(txtESMS.Text)) <> 0 And Len(Trim(txtESMS.Text)) >= 8 Then
            'Establecemos el cuerpo y el destinatario del mismo
            Mensaje = txtEEquipo.Text & " te ha enviado su posicion. "
            Mensaje = Mensaje & "Latitud: " & lblLatitude.Text & ". "
            Mensaje = Mensaje & "Longitud: " & lblLongitude.Text & ". "

            smsMessage.Body = Mensaje
            Dim recipient As New Recipient(txtESMS.Text)
            smsMessage.To.Add(recipient)
            'Le decimos que queremos un aviso de entrega para este SMS
            'smsMessage.RequestDeliveryReport = True
            'Enviamos el SMS
            smsMessage.Send()
            MessageBox.Show("Mensaje SMS Enviado")
        Else
            MsgBox("Número Telefónico Invalido", MsgBoxStyle.Critical, "Atención")
        End If
   End Sub
```

CAPITULO IV. DISEÑO DEL PORTAL DE INTERNET

IV.I. Diseño de la Base de Datos

La base de datos quedara de la siguiente manera:

Nombre de la Base de Datos	GPSWEB
Nombre del usuario	admingpsweb

Para darnos una idea del entorno del panel de control del dominio www.elaion.com.mx y los pasos para generar administrar una base de datos, adicionaremos las imágenes de dicho panel de control.

Por lo tanto para generar la Base de Datos, le damos clic en el menú principal (lado derecho de la pantalla) al icono **inicio** y en el apartado de Aplicaciones y Servicios, le daremos clic en el icono **Base de Datos** como lo muestra la imagen 4.1.

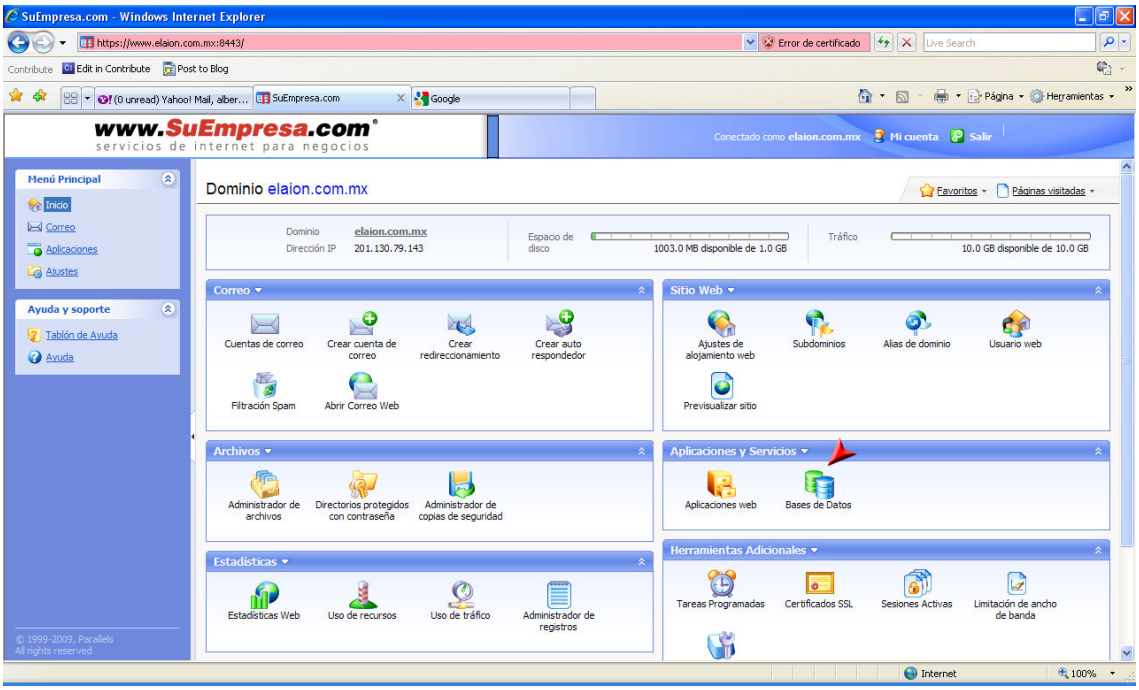


Imagen 4.1. Creación de Base de Datos Paso A.
(Fuente: Panel de Administración del Sitio www.elaion.com.mx)

Hecho lo anterior, llamara otra ventana como la imagen 4.2. en donde seleccionaremos el icono de **Añadir una nueva Base de Datos**

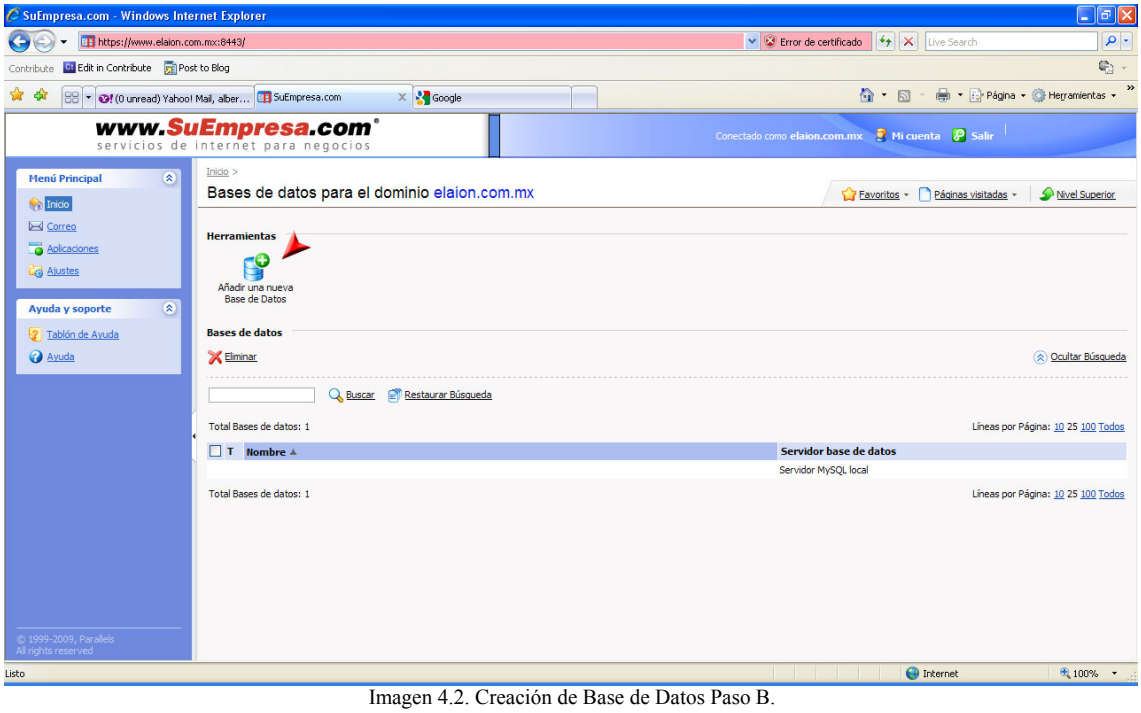


Imagen 4.2. Creación de Base de Datos Paso B.
(Fuente: Panel de Administración del Sitio www.elaion.com.mx)

Después de esto, nos preguntara por el nombre de la base de datos, el cual será GPSWEB del tipo MYSQL. Y damos clic al botón Aceptar.

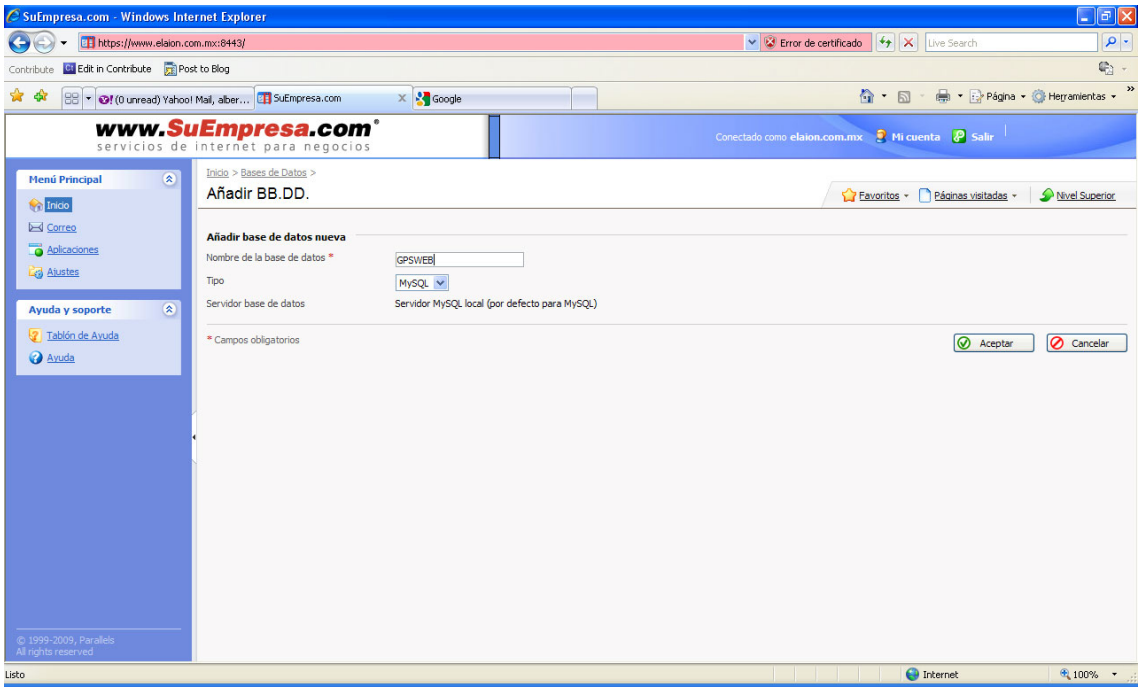


Imagen 4.3. Creación de Base de Datos Paso C.
(Fuente: Panel de Administración del Sitio www.elaion.com.mx)

Con ayuda de este panel de administración, ha sido creada la base de datos, solo nos falta crear un usuario para esta base de datos. Al darle aceptar en el paso anterior (imagen 4.3.), nos direcciono a otra ventana en donde se le dará clic en el botón **Añadir Usuario de BB.DD.** como se muestra en la imagen 4.4.

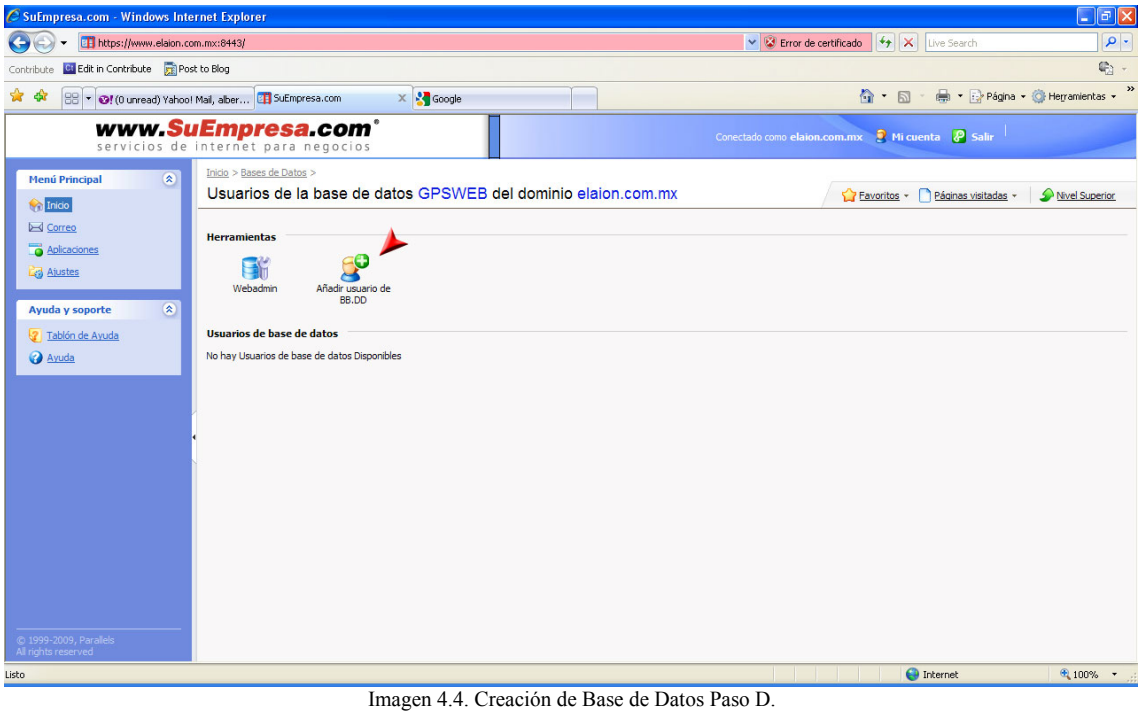


Imagen 4.4. Creación de Base de Datos Paso D.
(Fuente: Panel de Administración del Sitio www.elaion.com.mx)

El nombre de usuario a crear será admingpsweb con su respectiva contraseña

Una vez generada la base de datos y conforme al análisis de requerimientos del Capitulo III, se creara la tabla rastreos_gps con las caracteristicas mostradas en la Tabla 4.1.

rastreo_gps

Campo	Tipo(longitud)	Propiedades
id_rastreo	Integer	Not Null, Primary Key, Auto-Increment
Equipo	Char(15)	Not Null
Latitud	Char(15)	Not Null
Longitud	Char(15)	Not Null
Velocidad	Char(10)	Not Null
Fecha	DateTime	Not Null

Tabla 4.1.Caracteristicas de la tabla.

Para generar esta tabla, le damos clic en el menú principal (lado derecho de la pantalla) al icono **inicio** y en el apartado de Aplicaciones y Servicios, le daremos clic en el icono **Base de Datos** como lo muestra la imagen 4.1. Después de esto nos seleccionamos la Base de datos GPSWEB y le damos clic al icono **WebAdmin,** como lo muestra la imagen 4.5.

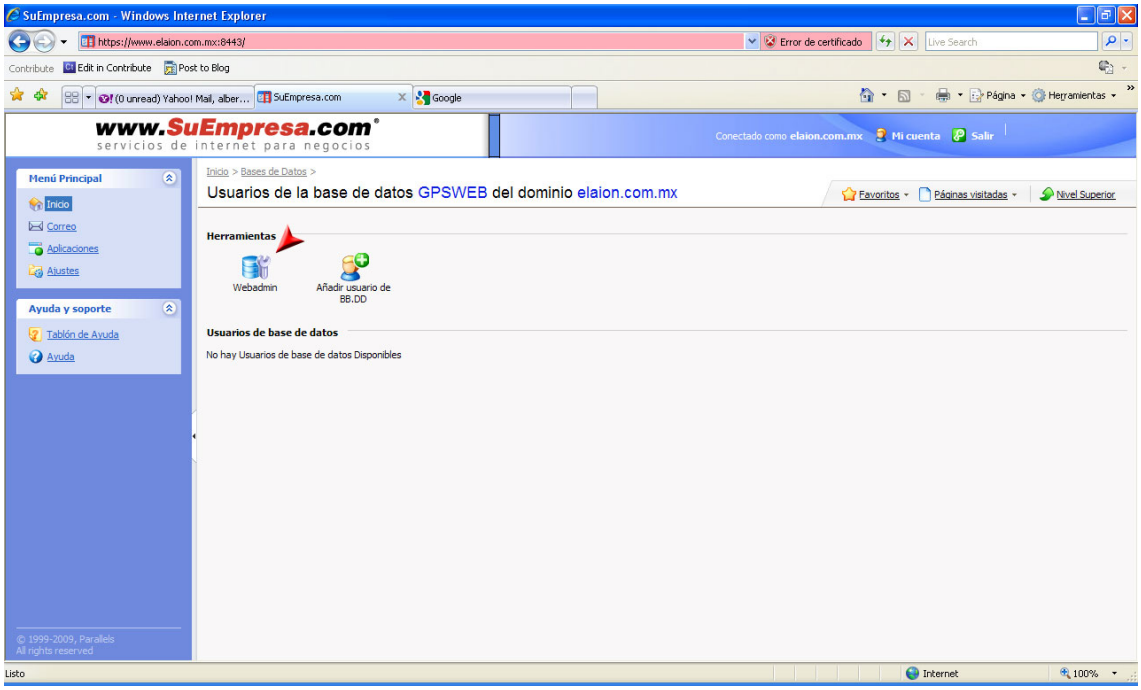


Imagen 4.5. PHPMYADMIN
(Fuente: Panel de Administración del Sitio www.elaion.com.mx)

Con esto, nos abrio una ventana nueva de adminisracion de Base de datos llamado PHPMYADMIN, con el cual seleccionaremos en la parte derecha del menú, nuestra base de datos GPSWEB() y le daremos clic a la pestaña de SQL para ejecutar el siguiente código como lo muestra la imagen 4.6.

```
CREATE TABLE ‘rastreos_gps{
‘id_rastreo’ INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
‘equipo’ CHAR(15) NOT NULL,
‘latitud’ CHAR(15) NOT NULL,
‘longitud’ CHAR(15) NOT NULL,
‘velocidad’ CHAR(10) NOT NULL,
‘fecha’ DATETIME NOT NULL
} ENGINE=MYISAM;
```

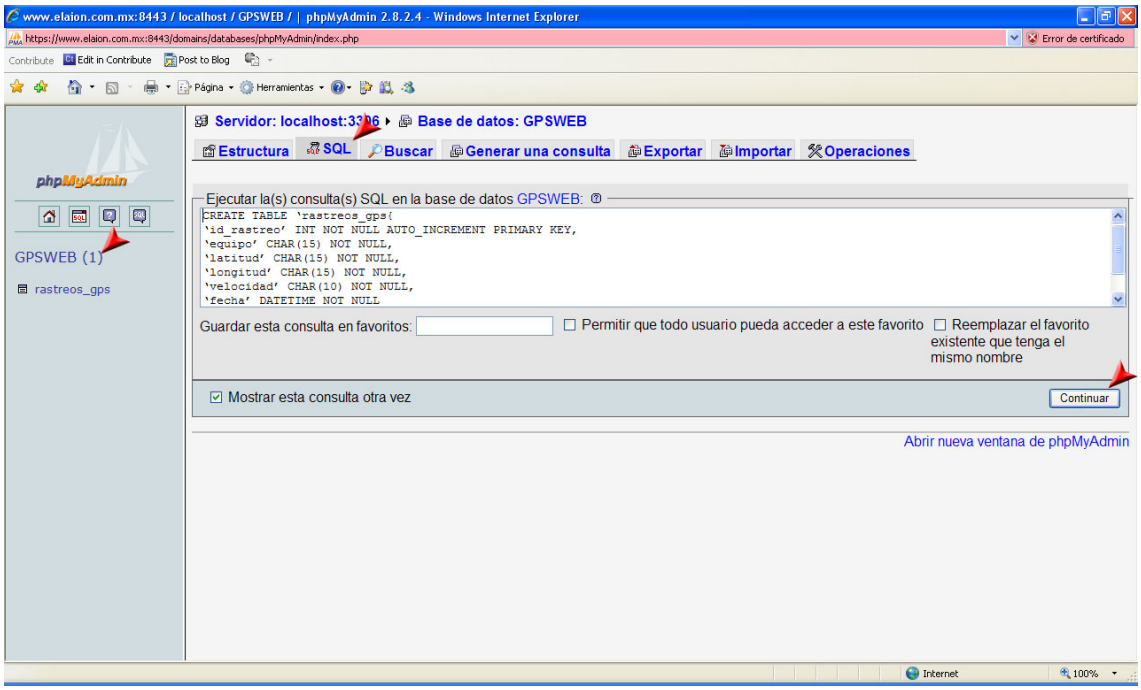


Imagen 4.6. Ejecución de Sentencias SQL
(Fuente: Panel de Administración del Sitio www.elaion.com.mx)

Al ejecutar esta sentencia SQL, ha quedado generada la tabla rastreo_gps, dentro de la base de datos GPSWEB

IV.II. Módulo de Adquisición de la información

Este módulo será el encargado de recibir la información como la latitud, longitud, velocidad y fecha que envía el dispositivo móvil.

Como ya vimos en el capitulo III, el Dispositivo móvil envía una dirección URI, con la siguiente estructura:

http://www.elaion.com.mx/gpsweb/inspos.php?equipo=*datoequipo*&latitud=*datolatitud*&longitud=*datolongitud*&velocidad=*datovelocidad*&fecha=*datofecha*

Traduciendo la anterior dirección URI, nos indica que va a ejecutar la página llamada inspos.php que esta en www.elaion.com.mx, conteniendo las variables equipo, latitud, longitud, velocidad, fecha y sus respectivos valores. Por lo tanto la pagina inspos.php, recogerá dichas variables y valores y los insertara en la tabla rastreos_gps de la base de datos GPSWEB.

Inspos.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>GPSWEB: Actualización de Registros</title>
<style type="text/css">
<!--
body,td,th {
        font-size: 12px;
}
body {
        margin-left: 0px;
        margin-top: 0px;
        margin-right: 0px;
        margin-bottom: 0px;
}
-->
</style></head>

</head>

<body>
<?php
// Recolectamos las variables con sus valores
$equipo2=$HTTP_GET_VARS["equipo"];
$latitud2=$HTTP_GET_VARS["latitud"];
$longitud2=$HTTP_GET_VARS["longitud"];
$velocidad2=$HTTP_GET_VARS["velocidad"];
$fecha2=$HTTP_GET_VARS["fecha"];

  // Nos conectamos al Administrador de Base de Datos con el usuario creado
  if (!($conid=mysql_connect("localhost","admingpsweb","agpsw")))
    {
      // Si no se puede conectar manda el siguiente mensaje
      echo "Error al conectarse a  la base de datos.";
      exit();
    }

  // Seleccionamos la Base de Datos GPSWEB
  if (!mysql_select_db("GPSWEB",$conid))
    {
     // Si no se puede realizar manda el siguiente mensaje
     echo "Error seleccionando la base de datos.";
    exit();
    }
```

```
   // Ya que se selecciono la Base, insertamos el registro en la tabla rastreos_gps
   $query="Insert into rastreos_gps(equipo,latitud,longitud,velocidad,fecha) values
('$equipo2','$latitud2','$longitud2','$velocidad2','$fecha2')";
   // Ejecutamos la sentencia SQL
   $rsid=mysql_query($query,$conid);

  // Si se Inserto o no el dato, informamos al usuario
  if ($rsid==false){
      echo "El dato no se pudo insertar<BR>";
      echo mysql_errno().":".mysql_error()."<BR>";
      exit();
      }

   echo "Dato Insertado";
   exit();
?>

</body>
</html>
```

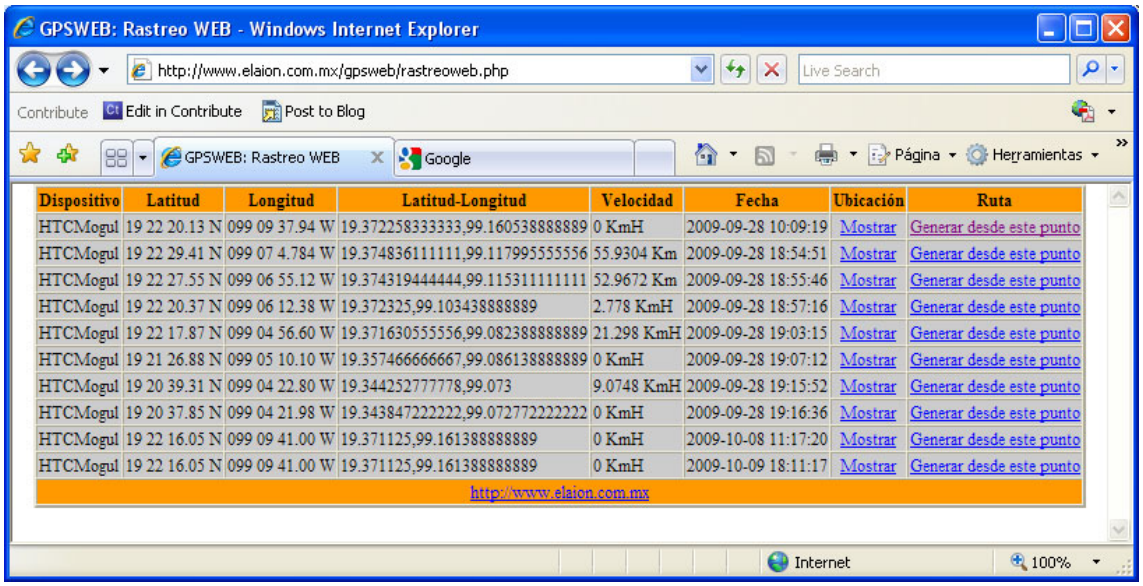
IV.III. Módulo de Monitoreo de la Información

Este módulo será el encargado de mostrar los registros de la tabla rastreos_gps que contiene la información de equipo, latitud, longitud, velocidad y fecha que envía el dispositivo móvil.

La dirección electrónica para ver dicho módulo es:

http://www.elaion.com.mx/gpsweb/rastreoweb.php

La vista de la página se muestra en la Imagen 4.7.



Dispositivo	Latitud	Longitud	Latitud-Longitud	Velocidad	Fecha	Ubicación	Ruta
HTCMogul	19 22 20.13 N	099 09 37.94 W	19.372258333333,99.160538888889	0 KmH	2009-09-28 10:09:19	Mostrar	Generar desde este punto
HTCMogul	19 22 29.41 N	099 07 4.784 W	19.374836111111,99.117995555556	55.9304 Km	2009-09-28 18:54:51	Mostrar	Generar desde este punto
HTCMogul	19 22 27.55 N	099 06 55.12 W	19.374319444444,99.115311111111	52.9672 Km	2009-09-28 18:55:46	Mostrar	Generar desde este punto
HTCMogul	19 22 20.37 N	099 06 12.38 W	19.372325,99.103438888889	2.778 KmH	2009-09-28 18:57:16	Mostrar	Generar desde este punto
HTCMogul	19 22 17.87 N	099 04 56.60 W	19.371630555556,99.082388888889	21.298 KmH	2009-09-28 19:03:15	Mostrar	Generar desde este punto
HTCMogul	19 21 26.88 N	099 05 10.10 W	19.357466666667,99.086138888889	0 KmH	2009-09-28 19:07:12	Mostrar	Generar desde este punto
HTCMogul	19 20 39.31 N	099 04 22.80 W	19.344252777778,99.073	9.0748 KmH	2009-09-28 19:15:52	Mostrar	Generar desde este punto
HTCMogul	19 20 37.85 N	099 04 21.98 W	19.343847222222,99.072772222222	0 KmH	2009-09-28 19:16:36	Mostrar	Generar desde este punto
HTCMogul	19 22 16.05 N	099 09 41.00 W	19.371125,99.161388888889	0 KmH	2009-10-08 11:17:20	Mostrar	Generar desde este punto
HTCMogul	19 22 16.05 N	099 09 41.00 W	19.371125,99.161388888889	0 KmH	2009-10-09 18:11:17	Mostrar	Generar desde este punto
http://www.elaion.com.mx							

Imagen 4.7. Vista preeliminar de rastreoweb.php
(Los datos pueden variar)

Básicamente presenta la información que el dispositivo móvil envía y la muestra en columnas con la siguiente funcionalidad:

- Dispositivo:
  Como podemos recordar del Capitulo III.III, la aplicación del dispositivo móvil cuenta con un módulo de configuración en donde se captura el identificador del equipo en WEB. Este valor es el que se muestra en esta columna.

  Si la aplicación se instalara en varios dispositivos móviles con identificadores diferentes, los registros quedarían identificados en esta columna

- Latitud:
  Muestra el valor de la latitud en Horas, Minutos y Segundos y su referencia

- Longitud:
  Muestra el valor de la longitud en Horas, Minutos y Segundos y su referencia

  Algunas aplicaciones de mapas digitales solicitan este tipo de formato en las coordenadas para poder presentar la ubicación del punto. Ej. Google Maps

- Latitud-Longitud:
  Como el valor de la latitud y longitud es Horas minutos y Segundos, en este campo transformamos estos valores mediante la siguiente ecuación

  Horas + (Minutos/60) + (Segundos/3600)

  Y si el valor de referencia es S=SUR o S=South o W=West u O=OESTE, se le asigna un signo negativo

  Esto es debido a que algunas otras aplicaciones de mapas digitales solicitan este tipo de formato en las coordenadas para poder presentar la ubicación del punto. Ej. Google Latitud para Móviles

- Velocidad:
  Indica el valor de la velocidad que tenia el dispositivo al momento de enviar su ubicación transformado en KM/H

- Fecha y Hora:
  Indica el momento en el tiempo en el que se registro la ubicación

- Ubicación y Ruta se definirán con mas detalle en la siguiente sección

rastreoweb.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>GPSWEB: Rastreo WEB</title>
<style type="text/css">
<!--
body,td,th {
	font-size: 12px;
}
body {
	margin-left: 0px;
	margin-top: 0px;
	margin-right: 0px;
	margin-bottom: 0px;
}
-->
</style></head>

<body>
<?php

// Nos conectamos a la Base de Datos
$link = mysql_connect('localhost','admingpsweb','agpsw');
if (!$link)
 {
   die('No se pudo conectar con la base de datos: ' . mysql_error());
 }

if (!mysql_select_db("GPSWEB",$link))
 {
   echo "Error seleccionando la base de datos.";
   exit();
 }
// Ya que seleccionamos la Base de Datos, realizamos un select a la tabla rastreos_gps
// ordenada por equipo, id_rastreo y fecha para mostrarla en la pagina
$result = mysql_query("SELECT * FROM `rastreos_gps` order by equipo,id_rastreo,fecha",$link);
if (!$result) {
   die('No se pudo ejecutar la consulta:' . mysql_error());
}

 echo "<table align='center' cellspacing=0 border=2>
      <tr>
                        <td align=center bgcolor=#FF9900><b>Dispositivo</font></b></td>
                        <td align=center bgcolor=#FF9900><b>Latitud</font></b></td>
                        <td align=center bgcolor=#FF9900><b>Longitud</font></b></td>
                        <td align=center bgcolor=#FF9900><b>Latitud-Longitud</font></b></td>
                        <td align=center bgcolor=#FF9900><b>Velocidad</b></td>
                        <td align=center bgcolor=#FF9900><b>Fecha</b></td>
                        <td align=center bgcolor=#FF9900><b>Ubicación</b></td>
                        <td align=center bgcolor=#FF9900><b>Ruta</b></td>

                        </tr>";

 // Por cada registro de la tabla, generamos un renglon en la pagina
 while($registro=mysql_fetch_array($result))
  {
  $palabras="";
	$latitudde="";
  $signo="";
  $palabras=split(" ",$registro["latitud"]);
  // Si la palabra es Oeste, West y South o Sur, le pone un negativo al valor
  if($palabras[3]=="O" || $palabras[3]=="W"  || $palabras[3]=="S")
   {$signo="-";}
   // Google earth trabaja con grados decimales
   // Nuestra informacion tiene el siguiente formato: HH MM SS.SS
   // Lo modificamos de la siguiente manera
   // signo HH + (MM/60) + (SS.SS/3600)
   $latitudde=$signo.($palabras[0]+($palabras[1]/60)+($palabras[2]/3600));


  $palabras="";
```

```
            $longitudde="";
    $signo="";
    $palabras=split(" ",$registro["longitud"]);
    // Si la palabra es Oeste, West y South o Sur, le pone un negativo al valor
    if($palabras[3]=="O" || $palabras[3]=="W"  || $palabras[3]=="S")
     {$signo="-";}
     // Google earth trabaja con grados decimales
     // Nuestra informacion tiene el siguiente formato: HH MM SS.SS
     // Lo modificamos de la siguiente manera
     // signo HH + (MM/60) + (SS.SS/3600)
     $longitudde=$signo.($palabras[0]+($palabras[1]/60)+($palabras[2]/3600));

      echo "<tr><td align=center bgcolor=#CCCCCC>".$registro["equipo"]."</td>";
      echo "<td bgcolor=#CCCCCC>".$registro["latitud"]."</td>";
      echo "<td bgcolor=#CCCCCC>".$registro["longitud"]."</td>";
      echo "<td bgcolor=#CCCCCC>".$latitudde.",".$longitudde."</td>";
      echo "<td bgcolor=#CCCCCC>".$registro["velocidad"]."</td>";
      echo "<td bgcolor=#CCCCCC>".$registro["fecha"]."</td>";

     //Generamos un vinculo para que esta direccion pueda mostrar la ubicacion en mapas digitales

$url1="http://maps.google.com/maps?q=".$registro["latitud"].",".$registro["longitud"]."(".$registro["equipo"].")&iwloc=A&hl=es";

     //Generamos un vinculo para que esta direccion genere una ruta
     $url2="verruta.php?id_rastreo=".$registro["id_rastreo"]."&equipo=".$registro["equipo"]."&fecha=".$registro["fecha"];

     // http://maps.google.com/maps?q=19+23+40.80+N,099+04+43.23W(MI+SITIO)&iwloc=A&hl=es
     echo "<td align=center bgcolor=#CCCCCC><a href='$url1'>Mostrar</a></td>";
     echo "<td align=center bgcolor=#CCCCCC><a href='$url2'>Generar desde este punto</a></td>";

     echo "</tr>";

   }
  echo "<tr><td align='center' bgcolor=#FF9900 colspan=8><a
href='http://www.elaion.com.mx'>http://www.elaion.com.mx</a></tr></td></table>";

  //Liberamos los Recurso de Base de datos y consulta
  mysql_free_result($result);

mysql_close($link);
?>
</body>
</html>
```

IV.III. Visualización Gráfica de la Posición

Este módulo será el encargado de mostrar las ubicaciones de forma grafica a travez de las aplicaciones de terceros llamada Google Maps y Google Earth.

De la sección anterior, nos falto por explicar la función de las ultimas dos columnas, las cuales definiremos a continuación

- Ubicación:
  Presenta un link con los valores necesarios para abrir la dirección electrónica del google maps y presentarnos la ubicación en el mapa digital con una marca de posición gráfica y nombre del dispositivo, tal como lo muestra la imagen 4.8.

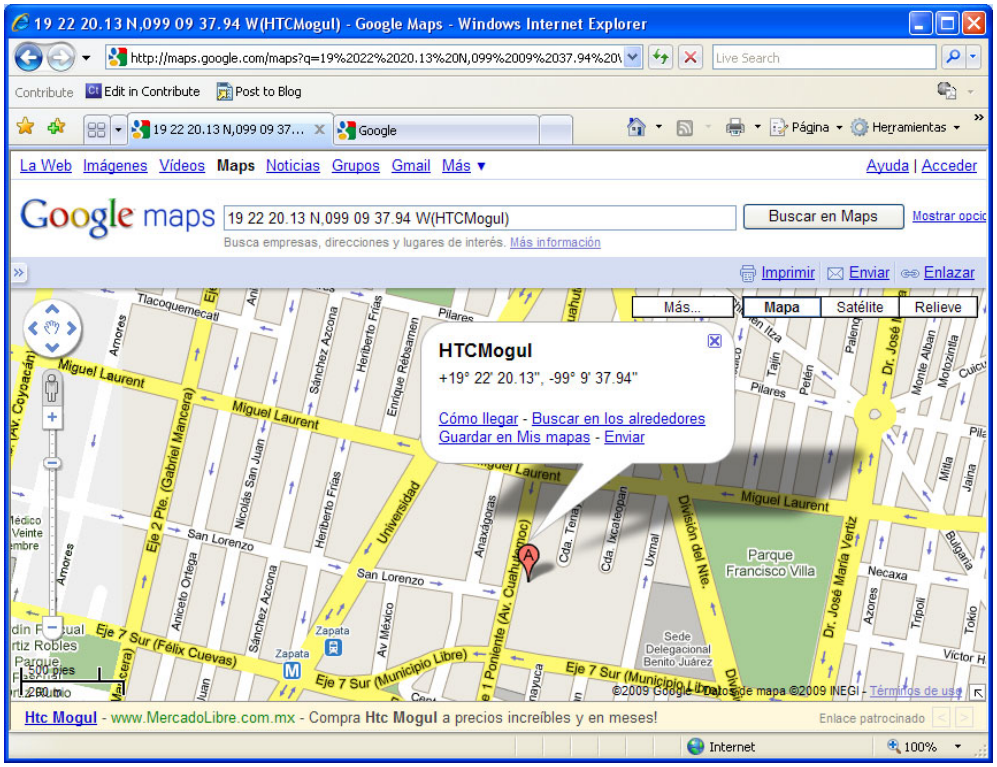


Imagen 4.8. Ubicación gráfica del dispositivo móvil
(Fuente: Google Maps)

- Ruta:
  Dependiendo del registro que seleccionemos, el sistema traerá todos los registros de ese punto en adelante, de ese día y de ese dispositivo en particular, con la finalidad de generar un archivo KML (Keyhole Markup Lenguaje) que es compatible con google earth para trazar una ruta visible que siguió el dispositivo móvil, y definiendo el punto de inicio y punto final mediante marcadores de posición.

  Este link ejecuta el archivo verruta.php el cual genera el archivo ruta.kml y muestra el link para su descarga en la PC (ver imagen 4.9.). Solamente se tiene que asegurar el usuario que cuando se descargue tenga la extensión KML, en caso contrario bastara con renombrar la extensión

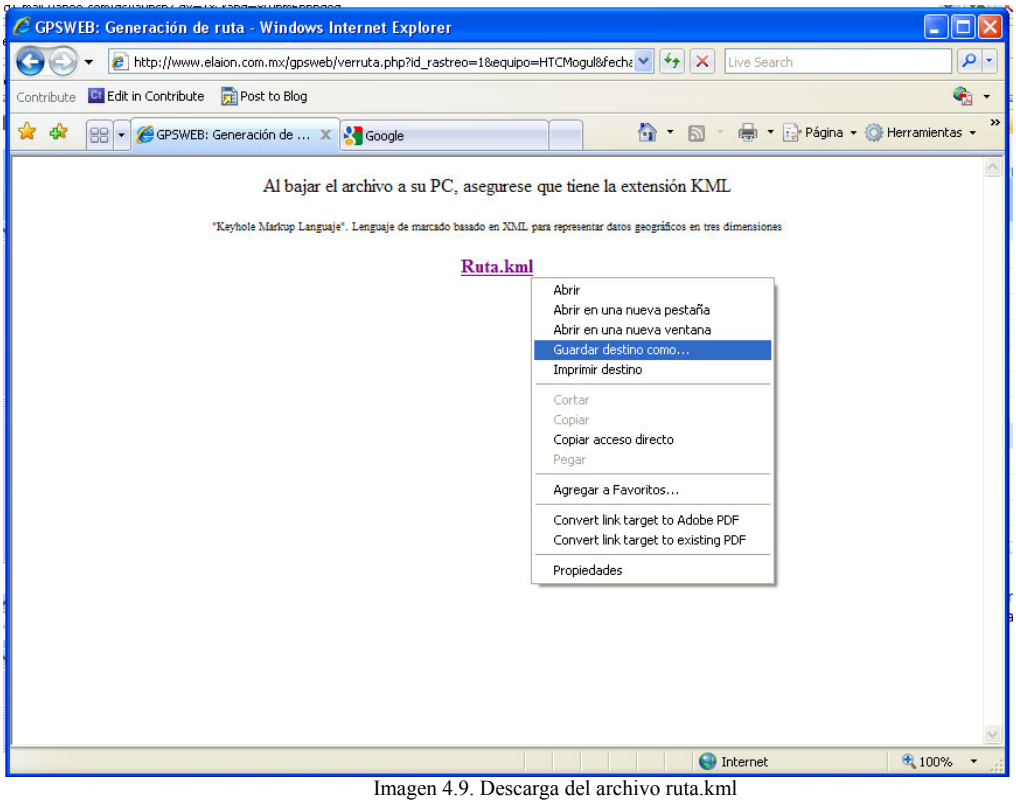


Imagen 4.9. Descarga del archivo ruta.kml

En caso de que el usuario le de clic al link, este mostrara el código o contenido de dicho archivo

Ya que tenemos el archivo ruta.kml en la pc, bastara con darle doble clic al archivo y el sistema operativo lo asociara con la aplicación google earth (es necesario instalar dicha aplicación) y nos mostrara la ruta seguida por el dispositivo como lo muestra la imagen 4.10.

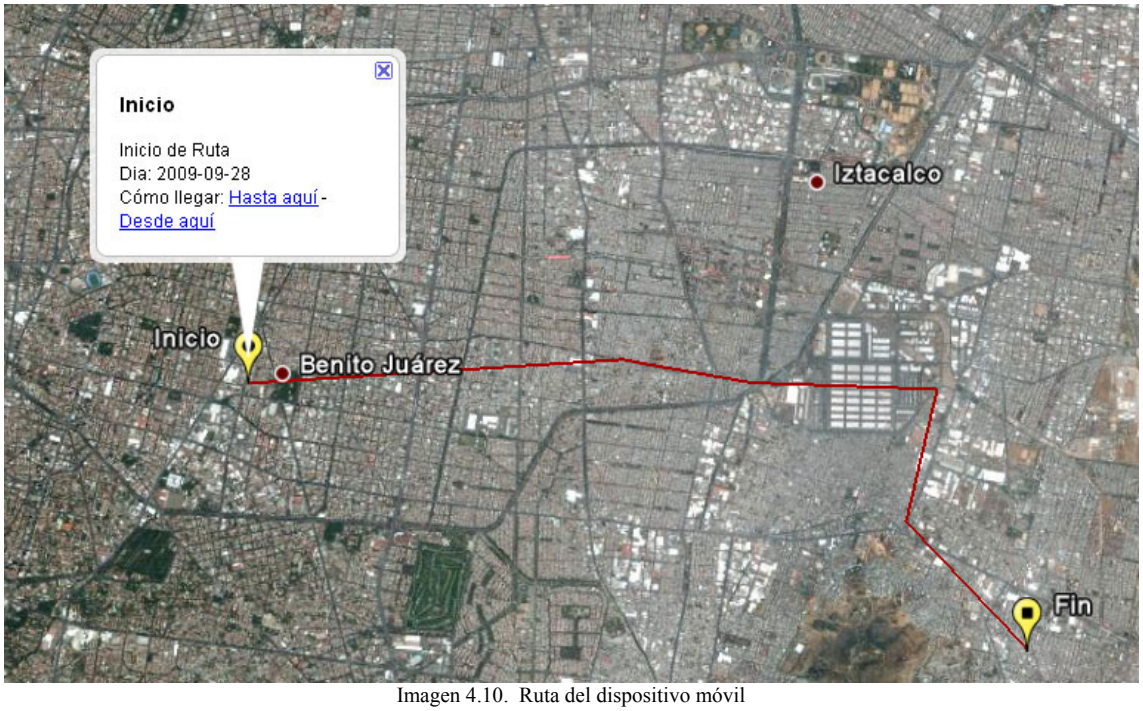


Imagen 4.10. Ruta del dispositivo móvil
(Fuente: Google Earth)

verruta.php

```
<?php
 // Recolectamos las variables
$id_rastreo2=$HTTP_GET_VARS["id_rastreo"];
$equipo2=$HTTP_GET_VARS["equipo"];
$fecha2=$HTTP_GET_VARS["fecha"];

// Nos conectamos a la Base de Datos
$link = mysql_connect('localhost','admingpsweb','agpsw');
if (!$link)
 {
   die('No se pudo conectar a la base de datos: ' . mysql_error());
 }

if (!mysql_select_db("GPSWEB",$link))
 {
   echo "Error seleccionando la base de datos.";
   exit();
 }

// Ya que seleccionamos la Base de Datos, realizamos un select a la tabla rastreos_gps
// Con la condicion de sea el registro, el equipo, y la fecha dia que seleccionamos
// ordenada por equipo, id_rastreo y fecha para mostrarla en la pagina

$query="SELECT * FROM `rastreos_gps` where id_rastreo>= ".$id_rastreo2." and equipo='".$equipo2."' and DATE_FORMAT(
fecha, '%Y-%m-%d' ) ='".substr($fecha2,0,10)."'  order by equipo,id_rastreo,fecha";
$result = mysql_query($query,$link);
if (!$result) {
   die('No se pudo ejecutar la sentencia:' . mysql_error());
}

// Por cada registro generaremos las coordenadas,
// con las caracteristicas siguientes "longitud,latitud,0"
$tagcoordenadas="";

// Para establecer gr\x{fffd}ficamente una marca en la ruta inicial y final,
// guardamos en estas variables el primer y ultimo registro
$latitudini="";
$longitudini="";
$latitudfin="";
$longitudfin="";

// Exploramos registro por registro de la consulta solicitada
while($registro=mysql_fetch_array($result))
 {
 $palabras="";
 $signo="";
 $palabras=split(" ",$registro["latitud"]);
 // Si la palabra es Oeste, West y South o Sur, le pone un negativo al valor
 if($palabras[3]=="O" || $palabras[3]=="W"  || $palabras[3]=="S")
  {$signo="-";}
  // Google earth trabaja con grados decimales
  // Nuestra informacion tiene el siguiente formato: HH MM SS.SS
  // Lo modificamos de la siguiente manera
  // signo HH + (MM/60) + (SS.SS/3600)
  $latitud=$signo.($palabras[0]+($palabras[1]/60)+($palabras[2]/3600));

  //Obtenemos el registro Inicial y Final de la latitud de la ruta
  // Para poder colocar una marca en la aplicaci\x{fffd}n
  if($latitudini=="")
  {$latitudini=$latitud;}
 $latitudfin=$latitud;

 $palabras="";
 $signo="";
 $palabras=split(" ",$registro["longitud"]);
 // Si la palabra es Oeste, West y South o Sur, le pone un negativo al valor
 if($palabras[3]=="O" || $palabras[3]=="W"  || $palabras[3]=="S")
  {$signo="-";}
 // Google earth trabaja con grados decimales
 // Nuestra informacion tiene el siguiente formato: HH MM SS.SS
 // Lo modificamos de la siguiente manera
 // signo HH + (MM/60) + (SS.SS/3600)
```

```
 $longitud=$signo.($palabras[0]+($palabras[1]/60)+($palabras[2]/3600));

 //la variable tagcoordenadas, trae todos los registros de la ruta a seguir
 //teniendo la siguiente formato longitud, latitud, 0
 $tagcoordenadas=$tagcoordenadas."                                    ".$longitud.",".$latitud.",0  ";

 //Obtenemos el registro Inicial y Final de la Longitud de la ruta
 // Para poder colocar una marca en la aplicacion
 if($longitudini=="")
  {$longitudini=$longitud;}
 $longitudfin=$longitud;
}

//Teniendo Toda la informacion de la Ruta, generaremos el archivo KML

$filename="ruta.kml";
//Creamos o abrimos el archivo
if (!$fileh = fopen($filename, 'w'))
 {
  echo "<p>No se puede abrir el archivo para guardar su texto. Por favor, si el problema persiste contacte con el
administrador.</p>";
  exit;
 }
// Introducimos Datos con formato KML (Keyhole Markup Language)
// Es un lenguaje de marcado basado en XML para representar
// datos geograficos en tres dimensiones
fwrite($fileh,'<?xml version="1.0" encoding="UTF-8"?>'."\r\n");
fwrite($fileh,'<kml xmlns="http://www.opengis.net/kml/2.2" xmlns:gx="http://www.google.com/kml/ext/2.2"
xmlns:kml="http://www.opengis.net/kml/2.2" xmlns:atom="http://www.w3.org/2005/Atom">'."\r\n");
fwrite($fileh,'<Document>'."\r\n");
fwrite($fileh,'          <name>Ruta.kml</name>'."\r\n");
fwrite($fileh,'          <StyleMap id="msn_ylw-pushpin">'."\r\n");
fwrite($fileh,'                    <Pair>'."\r\n");
fwrite($fileh,'                              <key>normal</key>'."\r\n");
fwrite($fileh,'                              <styleUrl>#sn_ylw-pushpin</styleUrl>'."\r\n");
fwrite($fileh,'                    </Pair>'."\r\n");
fwrite($fileh,'                    <Pair>'."\r\n");
fwrite($fileh,'                              <key>highlight</key>'."\r\n");
fwrite($fileh,'                              <styleUrl>#sh_ylw-pushpin</styleUrl>'."\r\n");
fwrite($fileh,'                    </Pair>'."\r\n");
fwrite($fileh,'          </StyleMap>'."\r\n");
fwrite($fileh,'          <Style id="sh_ylw-pushpin">'."\r\n");
fwrite($fileh,'                    <IconStyle>'."\r\n");
fwrite($fileh,'                              <scale>1.3</scale>'."\r\n");
fwrite($fileh,'                              <Icon>'."\r\n");
fwrite($fileh,'                                        <href>http://maps.google.com/mapfiles/kml/pushpin/ylw-
pushpin.png</href>'."\r\n");
fwrite($fileh,'                              </Icon>'."\r\n");
fwrite($fileh,'                              <hotSpot x="20" y="2" xunits="pixels" yunits="pixels"/>'."\r\n");
fwrite($fileh,'                    </IconStyle>'."\r\n");
fwrite($fileh,'                    <LineStyle>'."\r\n");
fwrite($fileh,'                              <color>ff0000aa</color>'."\r\n");
fwrite($fileh,'                              <width>2</width>'."\r\n");
fwrite($fileh,'                    </LineStyle>'."\r\n");
fwrite($fileh,'          </Style>'."\r\n");
fwrite($fileh,'          <Style id="sn_ylw-pushpin">'."\r\n");
fwrite($fileh,'                    <IconStyle>'."\r\n");
fwrite($fileh,'                              <scale>1.1</scale>'."\r\n");
fwrite($fileh,'                              <Icon>'."\r\n");
fwrite($fileh,'                                        <href>http://maps.google.com/mapfiles/kml/pushpin/ylw-
pushpin.png</href>'."\r\n");
fwrite($fileh,'                              </Icon>'."\r\n");
fwrite($fileh,'                              <hotSpot x="20" y="2" xunits="pixels" yunits="pixels"/>'."\r\n");
fwrite($fileh,'                    </IconStyle>'."\r\n");
fwrite($fileh,'                    <LineStyle>'."\r\n");
fwrite($fileh,'                              <color>ff0000aa</color>'."\r\n");
fwrite($fileh,'                              <width>2</width>'."\r\n");
fwrite($fileh,'                    </LineStyle>'."\r\n");
fwrite($fileh,'          </Style>'."\r\n");
fwrite($fileh,'          <Style id="sh_ylw-circle">'."\r\n");
fwrite($fileh,'                    <IconStyle>'."\r\n");
fwrite($fileh,'                              <scale>1.3</scale>'."\r\n");
fwrite($fileh,'                              <Icon>'."\r\n");
fwrite($fileh,'                                        <href>http://maps.google.com/mapfiles/kml/paddle/ylw-
circle.png</href>'."\r\n");
fwrite($fileh,'                              </Icon>'."\r\n");
```

```
fwrite($fileh,'                                <hotSpot x="32" y="1" xunits="pixels" yunits="pixels"/>'."\r\n");
fwrite($fileh,'                        </IconStyle>'."\r\n");
fwrite($fileh,'                        <ListStyle>'."\r\n");
fwrite($fileh,'                                <ItemIcon>'."\r\n");
fwrite($fileh,'                                        <href>http://maps.google.com/mapfiles/kml/paddle/ylw-circle-
lv.png</href>'."\r\n");
fwrite($fileh,'                                </ItemIcon>'."\r\n");
fwrite($fileh,'                        </ListStyle>'."\r\n");
fwrite($fileh,'        </Style>'."\r\n");
fwrite($fileh,'        <Style id="sn_ylw-circle">'."\r\n");
fwrite($fileh,'                        <IconStyle>'."\r\n");
fwrite($fileh,'                                <scale>1.1</scale>'."\r\n");
fwrite($fileh,'                                <Icon>'."\r\n");
fwrite($fileh,'                                        <href>http://maps.google.com/mapfiles/kml/paddle/ylw-
circle.png</href>'."\r\n");
fwrite($fileh,'                                </Icon>'."\r\n");
fwrite($fileh,'                                <hotSpot x="32" y="1" xunits="pixels" yunits="pixels"/>'."\r\n");
fwrite($fileh,'                        </IconStyle>'."\r\n");
fwrite($fileh,'                        <ListStyle>'."\r\n");
fwrite($fileh,'                                <ItemIcon>'."\r\n");
fwrite($fileh,'                                        <href>http://maps.google.com/mapfiles/kml/paddle/ylw-circle-
lv.png</href>'."\r\n");
fwrite($fileh,'                                </ItemIcon>'."\r\n");
fwrite($fileh,'                        </ListStyle>'."\r\n");
fwrite($fileh,'        </Style>'."\r\n");
fwrite($fileh,'        <StyleMap id="msn_ylw-circle">'."\r\n");
fwrite($fileh,'                        <Pair>'."\r\n");
fwrite($fileh,'                                <key>normal</key>'."\r\n");
fwrite($fileh,'                                <styleUrl>#sn_ylw-circle</styleUrl>'."\r\n");
fwrite($fileh,'                        </Pair>'."\r\n");
fwrite($fileh,'                        <Pair>'."\r\n");
fwrite($fileh,'                                <key>highlight</key>'."\r\n");
fwrite($fileh,'                                <styleUrl>#sh_ylw-circle</styleUrl>'."\r\n");
fwrite($fileh,'                        </Pair>'."\r\n");
fwrite($fileh,'        </StyleMap>'."\r\n");
fwrite($fileh,'        <Style id="sh_ylw-square">'."\r\n");
fwrite($fileh,'                        <IconStyle>'."\r\n");
fwrite($fileh,'                                <scale>1.3</scale>'."\r\n");
fwrite($fileh,'                                <Icon>'."\r\n");
fwrite($fileh,'                                        <href>http://maps.google.com/mapfiles/kml/paddle/ylw-
square.png</href>'."\r\n");
fwrite($fileh,'                                </Icon>'."\r\n");
fwrite($fileh,'                                <hotSpot x="32" y="1" xunits="pixels" yunits="pixels"/>'."\r\n");
fwrite($fileh,'                        </IconStyle>'."\r\n");
fwrite($fileh,'                        <ListStyle>'."\r\n");
fwrite($fileh,'                                <ItemIcon>'."\r\n");
fwrite($fileh,'                                        <href>http://maps.google.com/mapfiles/kml/paddle/ylw-square-
lv.png</href>'."\r\n");
fwrite($fileh,'                                </ItemIcon>'."\r\n");
fwrite($fileh,'                        </ListStyle>'."\r\n");
fwrite($fileh,'        </Style>'."\r\n");
fwrite($fileh,'        <StyleMap id="msn_ylw-square">'."\r\n");
fwrite($fileh,'                        <Pair>'."\r\n");
fwrite($fileh,'                                <key>normal</key>'."\r\n");
fwrite($fileh,'                                <styleUrl>#sn_ylw-square</styleUrl>'."\r\n");
fwrite($fileh,'                        </Pair>'."\r\n");
fwrite($fileh,'                        <Pair>'."\r\n");
fwrite($fileh,'                                <key>highlight</key>'."\r\n");
fwrite($fileh,'                                <styleUrl>#sh_ylw-square</styleUrl>'."\r\n");
fwrite($fileh,'                        </Pair>'."\r\n");
fwrite($fileh,'        </StyleMap>'."\r\n");
fwrite($fileh,'        <Style id="sn_ylw-square">'."\r\n");
fwrite($fileh,'                        <IconStyle>'."\r\n");
fwrite($fileh,'                                <scale>1.1</scale>'."\r\n");
fwrite($fileh,'                                <Icon>'."\r\n");
fwrite($fileh,'                                        <href>http://maps.google.com/mapfiles/kml/paddle/ylw-
square.png</href>'."\r\n");
fwrite($fileh,'                                </Icon>'."\r\n");
fwrite($fileh,'                                <hotSpot x="32" y="1" xunits="pixels" yunits="pixels"/>'."\r\n");
fwrite($fileh,'                        </IconStyle>'."\r\n");
fwrite($fileh,'                        <ListStyle>'."\r\n");
fwrite($fileh,'                                <ItemIcon>'."\r\n");
fwrite($fileh,'                                        <href>http://maps.google.com/mapfiles/kml/paddle/ylw-square-
lv.png</href>'."\r\n");
fwrite($fileh,'                                </ItemIcon>'."\r\n");
fwrite($fileh,'                        </ListStyle>'."\r\n");
```

```
fwrite($fileh,'		</Style>'."\r\n");
fwrite($fileh,'		<Placemark>'."\r\n");
fwrite($fileh,'			<name>Ruta</name>'."\r\n");
fwrite($fileh,'			<styleUrl>#msn_ylw-pushpin</styleUrl>'."\r\n");
fwrite($fileh,'			<LineString>'."\r\n");
fwrite($fileh,'				<tessellate>1</tessellate>'."\r\n");
fwrite($fileh,'				<coordinates>'."\r\n");
fwrite($fileh,'				'.$tagcoordenadas."\r\n");
fwrite($fileh,'				 </coordinates>'."\r\n");
fwrite($fileh,'			</LineString>'."\r\n");
fwrite($fileh,'		</Placemark>'."\r\n");
fwrite($fileh,'		'."\r\n");
fwrite($fileh,'		<Placemark>'."\r\n");
fwrite($fileh,'			<name>Inicio</name>'."\r\n");
fwrite($fileh,'			<description>Inicio de Ruta'."\r\n");
fwrite($fileh,'			Dia: '.substr($fecha2,0,10)."\r\n");
fwrite($fileh,'			</description>'."\r\n");
fwrite($fileh,'			<LookAt>'."\r\n");
fwrite($fileh,'				<longitude>'.$longitudini.'</longitude>'."\r\n");
fwrite($fileh,'				<latitude>'.$latitudini.'</latitude>'."\r\n");
fwrite($fileh,'				<altitude>0</altitude>'."\r\n");
fwrite($fileh,'				<range>327.6664158876513</range>'."\r\n");
fwrite($fileh,'				<tilt>15.84192145146256</tilt>'."\r\n");
fwrite($fileh,'				<heading>5.631553392556575e-006</heading>'."\r\n");
fwrite($fileh,'				<altitudeMode>relativeToGround</altitudeMode>'."\r\n");
fwrite($fileh,'				<gx:altitudeMode>relativeToSeaFloor</gx:altitudeMode>'."\r\n");
fwrite($fileh,'			</LookAt>'."\r\n");
fwrite($fileh,'			<styleUrl>#msn_ylw-circle</styleUrl>'."\r\n");
fwrite($fileh,'			<Point>'."\r\n");
fwrite($fileh,'				<coordinates>'.$longitudini.','.$latitudini.',0 </coordinates>'."\r\n");
fwrite($fileh,'			</Point>'."\r\n");
fwrite($fileh,'		</Placemark>'."\r\n");
fwrite($fileh,"."\r\n");
fwrite($fileh,'		<Placemark>'."\r\n");
fwrite($fileh,'			<name>Fin</name>'."\r\n");
fwrite($fileh,'			<description>Fin de Ruta'."\r\n");
fwrite($fileh,'			Dia: '.substr($fecha2,0,10)."\r\n");
fwrite($fileh,'			</description>'."\r\n");
fwrite($fileh,'			<LookAt>'."\r\n");
fwrite($fileh,'				<longitude>'.$longitudfin.'</longitude>'."\r\n");
fwrite($fileh,'				<latitude>'.$latitudfin.'</latitude>'."\r\n");
fwrite($fileh,'				<altitude>0</altitude>'."\r\n");
fwrite($fileh,'				<range>269.5626065115161</range>'."\r\n");
fwrite($fileh,'				<tilt>8.545643487755228</tilt>'."\r\n");
fwrite($fileh,'				<heading>7.503555492219565e-007</heading>'."\r\n");
fwrite($fileh,'				<altitudeMode>relativeToGround</altitudeMode>'."\r\n");
fwrite($fileh,'				<gx:altitudeMode>relativeToSeaFloor</gx:altitudeMode>'."\r\n");
fwrite($fileh,'			</LookAt>'."\r\n");
fwrite($fileh,'			<styleUrl>#msn_ylw-square</styleUrl>'."\r\n");
fwrite($fileh,'			<Point>'."\r\n");
fwrite($fileh,'				<coordinates>'.$longitudfin.','.$latitudfin.',0 </coordinates>'."\r\n");
fwrite($fileh,'			</Point>'."\r\n");
fwrite($fileh,'		</Placemark>'."\r\n");
fwrite($fileh,"."\r\n");
fwrite($fileh,'		</Document>'."\r\n");
fwrite($fileh,'</kml>'."\r\n");

//Cerramos el Archivo
fclose($fileh);


 //Liberamos los Recurso de Base de datos y consulta
mysql_free_result($result);

mysql_close($link);

?>

<HTML>
<HEAD>
<TITLE>GPSWEB: Generaci&#65533;n de ruta</TITLE>
<style type="text/css">
<!--
.style3 {
	font-size: 18px;
```

```
            font-weight: bold;
            color: #FFFF00;
}
.Estilo1 {font-size: xx-small}
-->
</style>
</HEAD>
<body>
  <div align="center">
    <p>Al bajar el archivo a su PC, asegurese que tiene la extensi&oacute;n KML</p>
    <p class="Estilo1"> &quot;Keyhole Markup Languaje&quot;. Lenguaje de marcado basado en XML para representar datos
geogr&aacute;ficos en tres   dimensiones</p>
    <p><a href="ruta.kml"><strong>Ruta.kml</strong></a>
      <br>
      </p>
  </div>
</BODY>
</HTML>
```

CONCLUSIONES

Diseñar un proyecto que involucre sistemas de geolocalización a través de un dispositivo móvil y su integración hacia un medio de comunicación tan importante y universal como lo es Internet, requiere de conocimientos y técnicas que solamente se pueden adquirir a través de la disciplina de la Ingeniería.

Además, hemos tratado de una manera sencilla y práctica la construcción de este sistema y hemos aprendido que al diseñar cualquier aplicación de GPS, debemos tener en cuenta que la señal es extremadamente débil y que por esta causa, solo la encontramos disponible al aire libre debido a que dicha señal no penetra en la mayoría de los materiales densos, por este motivo, la señal no existe dentro de edificios o viviendas de concreto, a menos que nos encontremos muy cerca de ventanas con cristal de baja densidad..

Asimismo, tendremos en cuenta que toma un poco de tiempo el adquirir la señal suficiente para que el receptor de GPS calcule nuestras coordenadas, debido a varios factores como la sensibilidad de nuestro receptor o a zonas urbanas donde existan edificios que obstruyan la señal.

Una de las observaciones que no dejaremos escapar, es que el valor teórico del parámetro SignalToNoiseRatio de la sentencia GPGSV es de 0 a 99, en donde 0 representa una señal debil y 99 representa una señal maxima. Pero nunca se logro a lo largo de las pruebas de este proyecto, conseguir una medicion mayor a 50, aún en dias con cielos totalmente abiertos, por lo que se decidio cambiar este parámetro teórico de 0 a 50. Por lo que si en alguna localidad o pais se logran mediciones superiores a 50, solo bastará cambiar este valor.

Otra observación importante cuando teniamos cielos cerrados por lluvia y se lograban ver al menos 3 satelites, el margen de error en la medicion era de aproximadamente 4 a 5 metros. Pero al obtener este error en la medicion, nos dimos cuenta que el parámetro 2 de la sentencia GPRMC llamado Estatus, contenia un valor de “V”, por lo que se decidio codificar este parámetro para que en caso de que el valor fuera “V”, nos excluyera la posición obtenida.

Terminamos con sugerir que google earth y google maps cuentan con interfases API, las cuales pueden ser muy útiles en la construcción de sistemas GPS empresariales y de ésta manera estaremos en el proceso de crecimiento de esta aplicación, el cual nos deja un gran sabor de boca y esperando en un futuro poderlo comercializar siendo nuestro principal objetivo el competir con un costo accesible.

ANEXOS

Código Fuente completo de la aplicación para el dispositivo Móvil en Visual Studio

Código de la forma frmGPSWEB.vb

```
'Se agrega esta REFERENCIA para SMS
Imports Microsoft.WindowsMobile.PocketOutlook
Public Class frmGPSWEB
    'Definimos estas variables para conocer si tenemos datos validos
    'de estas sentencias y salir del ciclo de lectura del puerto
    Public bRMC As Boolean
    Public bGSV As Boolean
    Public bGSA As Boolean


    Private Sub frmGPSWEB_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Dim Parametros() As String
        'Se inicializan los valores de los objetos en la interface de usuario
        bRMC = False
        bGSV = False
        bGSA = False

        TabControl1.TabPages(0).Text = "Control"
        TabControl1.TabPages(1).Text = "Mas Info."
        TabControl1.TabPages(2).Text = "Configuraciones"
        TabControl1.TabPages(0).Show()

        txtSentenciaNMEA.Text = ""

        lblLatitude.Text = ""
        lblLatitudeS.Text = ""
        lblLongitude.Text = ""
        lblLongitudeS.Text = ""
        lblVelocidad.Text = "0 KmH"

        lblNumSat.Text = ""
        lblid1.Text = ""
        lblel1.Text = ""
        lblaz1.Text = ""
        lblsr1.Text = "0 KmH"

        lblid2.Text = ""
        lblel2.Text = ""
        lblaz2.Text = ""
        lblsr2.Text = "0 KmH"

        lblid3.Text = ""
        lblel3.Text = ""
        lblaz3.Text = ""
        lblsr3.Text = "0 KmH"

        lblid4.Text = ""
        lblel4.Text = ""
        lblaz4.Text = ""
        lblsr4.Text = "0 KmH"

        lblReloj.Text = ""

        lblHDOP.Text = ""
        lblVDOP.Text = ""

        'Lee un archivo de Configuraciones iniciales,
        'las cuales dependen de cada dispositivo

        'En este caso, lee la linea del archivo que empieza con
        'La cadena PORT la cual tiene el siguiente format.
        'Port,4,BaudRate,4800,DataBits,8,Parity,0,StopBits,1,HandShake,0
        Parametros = LeeArchivoTexto("Program Files/GPSWEB/gps.txt", "PORT")
        'Esta funcion me devuelve una matriz con los valores de la cadena
        'Parametro("PORT","4",BaudRate,4800,DataBits,8,....)
        'El archivo tiene que tener una linea con la Etiqueta FIN, para
```

```
        'poder detener la lectura
        If Parametros(0) <> "FIN" Then
            'Si el contenido del archivo no tiene la etiqueta FIN
            'entonces inicializa los Objetos de la forma con los valores
            'contenidos en la matriz
            cboPort.SelectedIndex = CInt(Parametros(1))
            txtBaudRate.Text = Parametros(3)
            txtDataBits.Text = Parametros(5)
            cboParity.SelectedIndex = CInt(Parametros(7))
            cboStopBits.SelectedIndex = CInt(Parametros(9))
            cboHandshake.SelectedIndex = CInt(Parametros(11))
        Else
            'Si no encuentra los parametros configurados,
            'Le pone parametros Iniciales a los objetos por default
            'Para que despues el usuario los modifique y presione el boton
            'Grabar Configuración
            cboPort.SelectedIndex = 4
            txtBaudRate.Text = "4800"
            txtDataBits.Text = "8"
            cboParity.SelectedIndex = 0
            cboStopBits.SelectedIndex = 1
            cboHandshake.SelectedIndex = 0
        End If
        'De la misma manera que con PORT, ahora leemos la cadena EEQUIPO
        'Esta cadena tiene el nombre del equipo y el telefono predeterminado
        'para envio de SMS
        Parametros = LeeArchivoTexto("Program Files/GPSWEB/gps.txt", "EEQUIPO")
        If Parametros(0) <> "FIN" Then
            txtEEquipo.Text = Parametros(1)
            txtESMS.Text = Parametros(3)
        Else
            'Si no encuentra los parametros configurados,
            'Le da pone parametros Iniciales por default
            txtEEquipo.Text = "123456789012345"
            txtESMS.Text = "55"
        End If

        'Abrimos el Puerto con las configuraciones en los objetos de la forma
        Try
            If SerialPort1.IsOpen Then
                'Si esta abierto el puerto lo cierra
                SerialPort1.Close()
            Else
                'Establece los parametros y abre el puerto
                SerialPort1.PortName = cboPort.SelectedItem.ToString
                SerialPort1.BaudRate = CInt(txtBaudRate.Text)
                SerialPort1.DataBits = CInt(txtDataBits.Text)
                SerialPort1.Parity = CType(cboParity.SelectedIndex,
System.IO.Ports.Parity)
                SerialPort1.StopBits = CType(cboStopBits.SelectedIndex,
System.IO.Ports.StopBits)
                SerialPort1.Handshake = CType(cboHandshake.SelectedIndex,
System.IO.Ports.Handshake)
                SerialPort1.Open()
            End If
        Catch ex As Exception
            ' Si no puede abrir el puerto, envia el mensaje
            MessageBox.Show("No se ha podido Abrir el Puerto: " &
cboPort.SelectedItem.ToString, "Atención")
            Exit Sub
        End Try


    End Sub

    Private Sub cmdObtener_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdObtener.Click
        Dim Dato As String
        Dim contador As Integer
        Dim Bandera As Boolean
        bRMC = False
        bGSV = False
        bGSA = False

        Bandera = False
        contador = 0
        txtSentenciaNMEA.Text = ""
```

```
        Dato = ""
        '        txtMensaje.Text = "Entro Click"
        While Bandera = False
            If SerialPort1.IsOpen Then
                Try
                    Dato = Trim(Mid(SerialPort1.ReadLine, 1, 250))
                    If Len(Trim(Dato)) <> 0 Then
                        'txtSentenciaNMEA.Text = Mid(Dato, 1, Len(Trim(Dato)) - 1)
                        txtSentenciaNMEA.Text = Mid(Dato, 1, Len(Trim(Dato)) - 1) &
ControlChars.CrLf & txtSentenciaNMEA.Text

                        IdentificadorNMEA(Mid(Dato, 1, Len(Trim(Dato)) - 1))
                        ' Se sale del ciclo hasta que las tres sentencias sentencias
hayan traido datos
                        If bRMC And bGSV And bGSA Then
                            Bandera = True
                        End If
                    End If
                Catch ex As Exception
                    'Continua el Ciclo
                End Try

            End If
            contador = contador + 1
            If contador = 50 Then
                txtSentenciaNMEA.Text = ""
            End If
            If contador >= 100 Then
                If Len(Trim(lblLatitude.Text)) = 0 Then
                    MessageBox.Show("Señal GPS Débil", "Atención")
                End If
                Bandera = True
            End If
        End While

    End Sub

    Private Sub lblParam_ParentChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles lblParam.ParentChanged

    End Sub

    Private Sub TabPage2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TabPage3.Click

    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdGrabar.Click
        Dim oSW As New System.IO.StreamWriter("Program Files/GPSWEB/gps.txt", False,
System.Text.Encoding.Default)
        '        Dim sr As New System.IO.StreamWriter("Program Files/GPSWEB/gps.txt",
System.Text.Encoding.Default, True)

        Dim Linea As String
        'Necesitamos el inico de la forma
        'Imports System.IO
        'Port,4,BaudRate,4800,DataBits,8,Parity,0,StopBits,1,HandShake,0
        Linea = ""
        Linea = Linea & "PORT," & cboPort.SelectedIndex
        Linea = Linea & ",BAUDRATE," & Trim(txtBaudRate.Text)
        Linea = Linea & ",DATABITS," & Trim(txtDataBits.Text)
        Linea = Linea & ",PARITY," & cboParity.SelectedIndex
        Linea = Linea & ",STOPBITS," & cboStopBits.SelectedIndex
        Linea = Linea & ",HANDSHAKE," & cboHandshake.SelectedIndex
        oSW.WriteLine(Linea)
        Linea = ""
        Linea = Linea & "EEQUIPO," & Trim(txtEEquipo.Text)
        Linea = Linea & ",ESMS," & Trim(txtESMS.Text)
        oSW.WriteLine(Linea)

        'Estas lineas siempre va al final como Informacion
        Linea = "Parity.None = 0"
        oSW.WriteLine(Linea)
        Linea = "Parity.Odd = 1"
        oSW.WriteLine(Linea)
        Linea = "Parity.Even = 2"
```

```
        oSW.WriteLine(Linea)
        Linea = "Parity.Mark = 3"
        oSW.WriteLine(Linea)
        Linea = "Parity.Space = 4"
        oSW.WriteLine(Linea)
        Linea = "StopBits.None = 0"
        oSW.WriteLine(Linea)
        Linea = "StopBits.One = 1"
        oSW.WriteLine(Linea)
        Linea = "StopBits.Two = 2"
        oSW.WriteLine(Linea)
        Linea = "StopBits.OnePointFive = 3"
        oSW.WriteLine(Linea)
        Linea = "Handshake.None = 0"
        oSW.WriteLine(Linea)
        Linea = "Handshake.XOnXOff = 1"
        oSW.WriteLine(Linea)
        Linea = "Handshake.RequestToSend = 2"
        oSW.WriteLine(Linea)
        Linea = "Handshake.RequestToSendXOnXOff = 3"
        oSW.WriteLine(Linea)
        Linea = "FIN"
        oSW.WriteLine(Linea)
        oSW.Close()
        MsgBox("Actualización Correcta", MsgBoxStyle.OkOnly, "Atención")
    End Sub

    Private Sub Label7_ParentChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label7.ParentChanged

    End Sub

    Private Sub txtBaudRate_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles txtBaudRate.TextChanged

    End Sub
    Private Function IdentificadorNMEA(ByVal sentencia As String) As Boolean
        'Antes de Identificar la sentencia, se procede a checar que el CHECKSUM,
        'Sea igual a Xor de todos los caracteres
        If Not ChecksumValido(sentencia) Then
            Return False
        End If
        'Comparamos el primer elemento de la Sentencia
        Select Case DivideSentencia(sentencia)(0)
            Case "$GPRMC"
                'RMC      Recommended Minimum Sentence C
                If DecodificacionGPRMC(sentencia) Then
                    Return True
                Else
                    Return False
                End If
            Case "$GPGSV"
                'GSV  Satellites in View
                If DecodificacionGPGSV(sentencia) Then
                    Return True
                Else
                    Return False
                End If
            Case "$GPGSA"
                'GSA      GPS DOP and Active Satellites
                If DecodificacionGPGSA(sentencia) Then
                    Return True
                Else
                    Return False
                End If
            Case Else
                ' cualquier Otra Sentencia
                Return False
        End Select
    End Function
    Private Function DecodificacionGPRMC(ByVal sentencia As String) As Boolean
        Dim Parametros() As String = DivideSentencia(sentencia)
        '$GPRMC,123519,A,4807.038,N,01131.000,E,022.4,084.4,230394,003.11,W*6A
        '0  GP            Global Positioning System (GPS)
        '   RMC           Recommended Minimum Sentence C
        '1  123519           Time UTC                                12:35:19 UTC
        '2  A                Status,A = Active, V = Void
```

```
        '3  4807.038    Latitude
        '4  N                   N=North, S=South)
        '5  01131.000 Longitude
        '6  E               E=East, W = West
        '7  022.4           Speed over the ground in Knots
        '8  084.4           Track angle in degrees True
        '9  230394          Date
        '10 003.1           Magnetic variation in degrees
        '11 W                   E=East,W = West
        '11 *6A          Checksum

        'YA que reconocimos la sentecia GPRMC y la funcion de los parametros,
calcularemos los
        'datos sigientes

        'LATITUD Y LONGITUD

        'En esta parte verificamos que los parametros para obtener Latitud y Longitud no
esten vacios
        If Parametros(3) <> "" And Parametros(4) <> "" And Parametros(5) <> "" And
Parametros(6) <> "" Then

            'Extraemos y convertimos Latitud en grados minutos y segundos
            Dim Latitude As String
            'Horas
            Latitude = Mid(Parametros(3), 1, 2) & " "
            'Horas & Minutos
            Latitude = Latitude & Mid(Parametros(3), 3, 2) & " "
            'Horas & Minutos & Segundos
            Latitude = Latitude & Mid(CStr(CDbl(Mid(Parametros(3), 5, 6)) * (60)), 1, 5)
            'Horas & Minutos $ Segundos & Hemisferio
            Latitude = Latitude & " " & Parametros(4)

            'Extraemos y convertimos Longitud en grados minutos y segundos
            Dim Longitude As String
            'Horas
            Longitude = Mid(Parametros(5), 1, 3) & " "
            'Horas & Minutos
            Longitude = Longitude & Mid(Parametros(5), 4, 2) & " "
            'Horas & minutos & segundos
            Longitude = Longitude & Mid(CStr(CDbl(Mid(Parametros(5), 6, 6)) * (60)), 1,
5)
            'Horas & Minutos $ Segundos & Hemisferio
            Longitude = Longitude & " " & Parametros(6)

            'Ponemos los calculos en los objetos para el usuario
            lblLatitude.Text = Latitude
            lblLatitudeS.Text = Parametros(3) & Parametros(4)

            lblLongitude.Text = Longitude
            lblLongitudeS.Text = Parametros(5) & Parametros(6)

        Else
            Return False
        End If
        'Cada satélite GPS contiene cuatro relojes atómicos que se utiliza para las
        'transmisiones. Por lo que estos relojes atómicos se puede utilizar para
sincronizar
        'el reloj del dispositivo con una precisión de milisegundos.
        If Parametros(1) <> "" Then
            Dim UtcHours As Integer = CType(Parametros(1).Substring(0, 2), Integer)
            Dim UtcMinutes As Integer = CType(Parametros(1).Substring(2, 2), Integer)
            Dim UtcSeconds As Integer = CType(Parametros(1).Substring(4, 2), Integer)
            Dim UtcMilliseconds As Integer
            ' Si tiene el parametro milisegundos se extraen
            If Parametros(1).Length > 7 Then UtcMilliseconds =
CType(Parametros(1).Substring(7), Integer)
            Dim Today As DateTime = System.DateTime.Now.ToUniversalTime
            Dim SatelliteTime As New System.DateTime(Today.Year, Today.Month, Today.Day,
UtcHours, UtcMinutes, UtcSeconds, UtcMilliseconds)
            'Traemos la fecha y la conversion de la hora UTC con la hora local
            'configurada en el dispositivo
            lblReloj.Text = CType(SatelliteTime.ToLocalTime(), String)
        End If
        ' VELOCIDAD
        ' La informacion esta en Knots, por lo que basta cob multiplicarlo
        ' por 1,852 para convertirla a KMH
```

```
        If Parametros(7) <> "" Then
            '1 Knot=1.852KmH
            '1 Knot=1.150779448MillasxH
            Dim Velocidad As Double
            Velocidad = CDbl(Parametros(7)) * 1.852
            lblVelocidad.Text = Trim(CStr(Velocidad)) & " KmH"
        End If

        ' SUFICIENCIA EN EL CALCULO
        'Hemos visto que cuando se cuentan con al menos 3 satelites, el margen de error
        'puede ser grande, por lo tanto esta sentencia tiene un parametro (2) el cual
nos
        'indica si la fuerza de la señal de esos satelites es suficiente para darnos una
        'posicion viable
        If Parametros(2) <> "" Then
            Select Case Parametros(2)
                Case "A"
                    ' Return True
                Case "V"
                    Return False
            End Select
        End If
        bRMC = True
        Return True

    End Function
    Private Function DecodificacionGPGSV(ByVal Sentencia As String) As Boolean
        Dim IdNumber As String
        Dim Azimuth As String
        Dim Elevation As String
        Dim SignalToNoiseRatio As String
        Dim Parametros() As String = DivideSentencia(Sentencia)
        '$GPGSV,2,1,08,01,40,083,46,02,17,308,41,12,07,344,39,14,22,228,45*75
        '0   GP              Global Positioning System (GPS)
        '0  GSV       Satellites in View
        '1  2         Number of sentences
        '2  1         Sentence 1 of 2
        '3  08         Number of Satellites in view
        '4  01         Satellite PRN (id) number. Pseudo-Random Noise
        '5  40         Elevation in degreess
        '6  083       Azimuth in degrees
        '7  46:     SNR in(dB - higher Is better) Relacion Señal Ruido
        '          More satellites infos
        'X  *75              Checksum

        'Por lo regular se tienen 4 bloques de informacion, cada bloque tiene
id.satelite, Elevacion, Acimut y fuera de la señal
        Dim Count As Integer
        For Count = 1 To 4
            ' Si tiene Parametros la sentencia?
            If (Parametros.Length - 1) >= (Count * 4 + 3) Then
                If Parametros(Count * 4) <> "" And Parametros(Count * 4 + 1) <> "" And
Parametros(Count * 4 + 2) <> "" And Parametros(Count * 4 + 3) <> "" Then
                    IdNumber = Parametros(Count * 4)
                    Elevation = Parametros(Count * 4 + 1)
                    Azimuth = Parametros(Count * 4 + 2)
                    SignalToNoiseRatio = Mid(Parametros(Count * 4 + 3), 1, 2)
                    'Hay veces que el ultimo parametro solo viene con el checksum
                    'por eso filtramos el checksum, y si no trae nada le ponemos 00
                    If Mid(SignalToNoiseRatio, 1, 1) = "*" Then   ' Si no tiene
                        SignalToNoiseRatio = "00"
                    End If
                    'El parametro SignalToNoiseRatio en teoria tiene el valor de 0 a 99
en donde
                    ' 0 representa una señal debil y 99 una señal Maxima
                    'Pero se ha probado este parametro en cielos totalmente despejado y
el valor maximo que se
                    'ha obtenido es 50, por lo tanto para este Proyecto se trabajara con
un maximo de 50.
                    'Si se tiene localidades donde la señal sea superior a 50, solamente
se cambiara
                    'el valor maximo del objeto de medicion
                    '
                    If Count = 1 Then
                        lblid1.Text = IdNumber
                        lblel1.Text = Elevation & " º"
                        lblaz1.Text = Azimuth & " º"
```

```
                    lblsr1.Text = SignalToNoiseRatio & " dB"
                    Pb1.Value = CType(SignalToNoiseRatio, Integer)
                End If
                If Count = 2 Then
                    lblid2.Text = IdNumber
                    lblel2.Text = Elevation & " º"
                    lblaz2.Text = Azimuth & " º"
                    lblsr2.Text = SignalToNoiseRatio & " dB"
                    Pb2.Value = CType(SignalToNoiseRatio, Integer)
                End If
                If Count = 3 Then
                    lblid3.Text = IdNumber
                    lblel3.Text = Elevation & " º"
                    lblaz3.Text = Azimuth & " º"
                    lblsr3.Text = SignalToNoiseRatio & " dB"
                    Pb3.Value = CType(SignalToNoiseRatio, Integer)
                End If
                If Count = 4 Then
                    lblid4.Text = IdNumber
                    lblel4.Text = Elevation & " º"
                    lblaz4.Text = Azimuth & " º"
                    lblsr4.Text = SignalToNoiseRatio & " dB"
                    Pb4.Value = CType(SignalToNoiseRatio, Integer)
                End If
            End If
            lblNumSat.Text = Parametros(3)
        End If
    Next
    bGSV = True
    Return True
End Function

Public Function DecodificacionGPGSA(ByVal sentencia As String) As Boolean
    Dim Parametros() As String = DivideSentencia(sentencia)
    '$GPGSA,A,3,04,05,,09,12,,,24,,,,,2.5,1.3,2.1*39
    '0  GP         Global Positioning System (GPS)
    '0  GSA            GPS DOP and Active Satellites
    '1  A          Selection Mode A=Autoselection of  2D o 3D, M = Manual
    '2  3          Mode 1=No fix,2=2D fix,       3=3D fix
    '3-14  04,05,,09…     Ident  n of 12 satellite used for fix
    '15 2.5               PDOP in meters                        2.5m PDOP
    '16 1.3               HDOP in meters                        1.3m HDOP
    '17 2.1               VDOP in meters                        2.1m VDOP
    '17 *39               Checksum data
    If Parametros(16) <> "" Then
        lblHDOP.Text = Parametros(16) & " mts"
    End If
    If Parametros(17) <> "" Then
        lblVDOP.Text = Mid(Parametros(17), 1, InStr(Parametros(17), "*") - 1) & "
mts"
    End If
    'No se tienen rangos muy claros en cuanto a que valores son los ideales, ya que
depende del
    'tipo de Aplicacion, pero aqui se muestra una tabla que puede darnos una idea
    '1      Ideal       Este es el más alto el nivel de confianza puede ser
utilizado
    '                   para aplicaciones que demandan la máxima precisión en todo
momento.
    '2-3    Excelente   En este nivel de confianza, las mediciones de posición se
consideran lo
    '                   suficientemente precisas para cumplir con todos, pero las
aplicaciones más sensibles
    '4-6    Bueno       Representa un nivel que marca el mínimo adecuado para la
toma de decisiones
    '                   podría ser usado para hacer sugerencias en la navegación
para el usuario
    '7-20   Moderado    Solo para estimaciones
    '21o+   Pobre       Deberia descartarse
    bGSA = True
    Return True
End Function

Private Sub TabPage2_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TabPage2.Click

End Sub
```

```
    Private Sub Label21_ParentChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label21.ParentChanged

    End Sub

    Private Sub cmdEnviar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdEnviar.Click
        'En la pagina www.elaion.com.mx/gpsweb/, esta una pagina inspos.php que recibe
        'Los parametros de latitude, longtude, etc. y los graba en su base de datos
        Dim siteUri As New Uri("http://www.elaion.com.mx/gpsweb/inspos.php?equipo=" &
txtEEquipo.Text & "&latitud=" & lblLatitude.Text & "&longitud=" & lblLongitude.Text &
"&velocidad=" & lblVelocidad.Text & "&fecha=" & Format(Now(), "yyyy/MM/dd HH:mm:ss"))
        'Si el servidor esta fuera de linea o existen muchas transacciones pendientes,
alerta al usuario
        If WebBrowser.IsOffline Or WebBrowser.IsBusy Then
            MsgBox("No hay conexion a internet, o esta ocupado el servidor",
MsgBoxStyle.Critical, "Error de Conexion")
        Else
            WebBrowser.Navigate(siteUri)
            MsgBox("Posicion registrada correctamente", MsgBoxStyle.OkOnly, "Atención")
        End If
    End Sub

    Private Sub cmdSMS_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles cmdSMS.Click
        'Para este evento, necesitamos el Microsoft.WindowsMobile.PocketOutlook
        'Creamo el Mensaje a enviar por SMS
        Dim smsMessage As New SmsMessage()
        Dim Mensaje As String
        'Si no esta vacio y que sea que tenga mas de 8 numeros
        If Len(Trim(txtESMS.Text)) <> 0 And Len(Trim(txtESMS.Text)) >= 8 Then
            'Establecemos el cuerpo y el destinatario del mismo
            Mensaje = txtEEquipo.Text & " te ha enviado su posicion. "
            Mensaje = Mensaje & "Latitud: " & lblLatitude.Text & ". "
            Mensaje = Mensaje & "Longitud: " & lblLongitude.Text & ". "

            smsMessage.Body = Mensaje
            Dim recipient As New Recipient(txtESMS.Text)
            smsMessage.To.Add(recipient)
            'Le decimos que queremos un aviso de entrega para este SMS
            'smsMessage.RequestDeliveryReport = True
            'Enviamos el SMS
            smsMessage.Send()
            MessageBox.Show("Mensaje SMS Enviado")
        Else
            MsgBox("Número Telefónico Invalido", MsgBoxStyle.Critical, "Atención")
        End If


    End Sub

    Private Sub TabPage1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TabPage1.Click

    End Sub
End Class
```

Código del módulo modGPSWEB.vb

```
Module modGPSWEB
    'Lee las configuraciones Iniciales
    Public Function LeeArchivoTexto(ByVal Archivo As String, ByVal Encabezado As String)
As String()
        'Recibe como parametros el Nombre del Archivo, y la cadena buscar
        ' Si la encuentra, trae la linea y la almacena en un vector, el cual devuelve
        Dim Parametros() As String
        Dim Cadena As String
        Cadena = "FIN,,"
        'Carga el fichero
        If System.IO.File.Exists(Archivo) = False Then
            MsgBox("El archivo " & Archivo & " no existe", MsgBoxStyle.OkOnly)
        Else
            Dim sr As New System.IO.StreamReader(Archivo, System.Text.Encoding.Default,
True)
            While sr.Peek() <> -1
                Cadena = sr.ReadLine
                ' Si reconoce la cadena de configuracion del puerto, la extrae
                ' e inicializa los objetos asociados
                ' La cadena recibida puede tener el siguiente Formato entre otras
                ' Port,1,BaudRate,4800,DataBits,8,Parity=None,StopBits,1,HandShake,None
                If Mid(Cadena, 1, Len(Encabezado)) = Encabezado Then
                    Exit While
                End If
            End While
            sr.Close()
        End If
        'Esta funcion recibe la cadena y nos devuelve la cadena pero en matriz
        Parametros = DivideSentencia(Cadena)
        Return Parametros
    End Function
    'Esta funcion nos devuelve una matriz unidimensional basada en cero que
    'contiene un número especificado de subcadenas
    'Es decir nos devuelve un elemento en el arreglo por cada palabra separada
    ' por comas en la sentencia
    Public Function DivideSentencia(ByVal sentencia As String) As String()
        Return Split(sentencia, ",")
    End Function
    'Esta funcion Checa que el parametro CheckSum coincida con el XOR de
    'todos los caracteres de la sentencia excluyendo $ y *.
    'Devolviendo True si es una sentencia Valida y false en caso contrario
    Public Function ChecksumValido(ByVal sentencia As String) As Boolean
        ' Loop through all chars to get a checksum
        Dim Checksum As Integer
        Dim Caracter As Char

        For Each Caracter In sentencia
            Select Case Caracter
                'Ignoramos el Simbolo $ y *
                Case "$"c
                Case "*"c
                    'Cuando encuentra * deja de calcular
                    Exit For
                Case Else
                    ' Calcula el XOR de los caracteres validos
                    If Checksum = 0 Then
                        Checksum = Convert.ToByte(Caracter)
                    Else
                        Checksum = Checksum Xor Convert.ToByte(Caracter)
                    End If
            End Select
        Next
        ' Devuelve la comparacion entre el numero despues del simbolo * de la sentencia
        ' y el valor calculado, convertido en 2 numeros hexadecimales
        Return sentencia.Substring(sentencia.IndexOf("*") + 1) = Checksum.ToString("X2")
    End Function


End Module
```

Código Fuente completo de la aplicación para el portal WEB

Código del archivo inspos.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>GPSWEB: Actualización de Registros</title>
<style type="text/css">
<!--
body,td,th {
        font-size: 12px;
}
body {
        margin-left: 0px;
        margin-top: 0px;
        margin-right: 0px;
        margin-bottom: 0px;
}
-->
</style></head>

</head>

<body>
<?php
// Recolectamos las variables con sus valores
$equipo2=$HTTP_GET_VARS["equipo"];
$latitud2=$HTTP_GET_VARS["latitud"];
$longitud2=$HTTP_GET_VARS["longitud"];
$velocidad2=$HTTP_GET_VARS["velocidad"];
$fecha2=$HTTP_GET_VARS["fecha"];

  // Nos conectamos al Administrador de Base de Datos con el usuario creado
  if (!($conid=mysql_connect("localhost","admingpsweb","agpsw")))
    {
      // Si no se puede conectar manda el siguiente mensaje
      echo "Error al conectarse a  la base de datos.";
      exit();
    }

  // Seleccionamos la Base de Datos GPSWEB
  if (!mysql_select_db("GPSWEB",$conid))
    {
     // Si no se puede realizar manda el siguiente mensaje
     echo "Error seleccionando la base de datos.";
    exit();
    }

  // Ya que se selecciono la Base, insertamos el registro en la tabla rastreos_gps
  $query="Insert into rastreos_gps(equipo,latitud,longitud,velocidad,fecha) values
('$equipo2','$latitud2','$longitud2','$velocidad2','$fecha2')";
  // Ejecutamos la sentencia SQL
  $rsid=mysql_query($query,$conid);

 // Si se Inserto o no el dato, informamos al usuario
 if ($rsid==false){
    echo "El dato no se pudo insertar<BR>";
    echo mysql_errno().":".mysql_error()."<BR>";
    exit();
    }

  echo "Dato Insertado";
  exit();
?>

</body>
</html>
```

Código del archivo rastreoweb.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>GPSWEB: Rastreo WEB</title>
<style type="text/css">
<!--
body,td,th {
        font-size: 12px;
}
body {
        margin-left: 0px;
        margin-top: 0px;
        margin-right: 0px;
        margin-bottom: 0px;
}
-->
</style></head>

<body>
<?php

// Nos conectamos a la Base de Datos
$link = mysql_connect('localhost','admingpsweb','agpsw');
if (!$link)
 {
   die('No se pudo conectar con la base de datos: ' . mysql_error());
 }

if (!mysql_select_db("GPSWEB",$link))
 {
   echo "Error seleccionando la base de datos.";
   exit();
 }
// Ya que seleccionamos la Base de Datos, realizamos un select a la tabla rastreos_gps
// ordenada por equipo, id_rastreo y fecha para mostrarla en la pagina
$result = mysql_query("SELECT * FROM `rastreos_gps` order by equipo,id_rastreo,fecha",$link);
if (!$result) {
   die('No se pudo ejecutar la consulta:' . mysql_error());
}

 echo "<table align='center' cellspacing=0 border=2>
      <tr>
                        <td align=center bgcolor=#FF9900><b>Dispositivo</font></b></td>
                        <td align=center bgcolor=#FF9900><b>Latitud</font></b></td>
                        <td align=center bgcolor=#FF9900><b>Longitud</font></b></td>
                        <td align=center bgcolor=#FF9900><b>Latitud-Longitud</font></b></td>
                        <td align=center bgcolor=#FF9900><b>Velocidad</b></td>
                        <td align=center bgcolor=#FF9900><b>Fecha</b></td>
                        <td align=center bgcolor=#FF9900><b>Ubicación</b></td>
                        <td align=center bgcolor=#FF9900><b>Ruta</b></td>

                        </tr>";

 // Por cada registro de la tabla, generamos un renglon en la pagina
 while($registro=mysql_fetch_array($result))
  {
  $palabras="";
        $latitudde="";
  $signo="";
  $palabras=split(" ",$registro["latitud"]);
  // Si la palabra es Oeste, West y South o Sur, le pone un negativo al valor
  if($palabras[3]=="O" || $palabras[3]=="W"  || $palabras[3]=="S")
   {$signo="-";}
   // Google earth trabaja con grados decimales
   // Nuestra informacion tiene el siguiente formato: HH MM SS.SS
   // Lo modificamos de la siguiente manera
   // signo HH + (MM/60) + (SS.SS/3600)
   $latitudde=$signo.($palabras[0]+($palabras[1]/60)+($palabras[2]/3600));


  $palabras="";
```

```
            $longitudde="";
    $signo="";
    $palabras=split(" ",$registro["longitud"]);
    // Si la palabra es Oeste, West y South o Sur, le pone un negativo al valor
    if($palabras[3]=="O" || $palabras[3]=="W"  || $palabras[3]=="S")
     {$signo="-";}
     // Google earth trabaja con grados decimales
     // Nuestra informacion tiene el siguiente formato: HH MM SS.SS
     // Lo modificamos de la siguiente manera
     // signo HH + (MM/60) + (SS.SS/3600)
     $longitudde=$signo.($palabras[0]+($palabras[1]/60)+($palabras[2]/3600));

      echo "<tr><td align=center bgcolor=#CCCCCC>".$registro["equipo"]."</td>";
      echo "<td bgcolor=#CCCCCC>".$registro["latitud"]."</td>";
      echo "<td bgcolor=#CCCCCC>".$registro["longitud"]."</td>";
      echo "<td bgcolor=#CCCCCC>".$latitudde.",".$longitudde."</td>";
      echo "<td bgcolor=#CCCCCC>".$registro["velocidad"]."</td>";
      echo "<td bgcolor=#CCCCCC>".$registro["fecha"]."</td>";

     //Generamos un vinculo para que esta direccion pueda mostrar la ubicacion en mapas digitales

$url1="http://maps.google.com/maps?q=".$registro["latitud"].",".$registro["longitud"]."(".$registro["equipo"].")&iwloc=A&hl=es";

     //Generamos un vinculo para que esta direccion genere una ruta
     $url2="verruta.php?id_rastreo=".$registro["id_rastreo"]."&equipo=".$registro["equipo"]."&fecha=".$registro["fecha"];

     // http://maps.google.com/maps?q=19+23+40.80+N,099+04+43.23W(MI+SITIO)&iwloc=A&hl=es
     echo "<td align=center bgcolor=#CCCCCC><a href='$url1'>Mostrar</a></td>";
     echo "<td align=center bgcolor=#CCCCCC><a href='$url2'>Generar desde este punto</a></td>";

     echo "</tr>";

   }
  echo "<tr><td align='center' bgcolor=#FF9900 colspan=8><a
href='http://www.elaion.com.mx'>http://www.elaion.com.mx</a></tr></td></table>";

  //Liberamos los Recurso de Base de datos y consulta
  mysql_free_result($result);

mysql_close($link);
?>
</body>
</html>
```

Código del archivo verruta.php

```
<?php
 // Recolectamos las variables
$id_rastreo2=$HTTP_GET_VARS["id_rastreo"];
$equipo2=$HTTP_GET_VARS["equipo"];
$fecha2=$HTTP_GET_VARS["fecha"];

// Nos conectamos a la Base de Datos
$link = mysql_connect('localhost','admingpsweb','agpsw');
if (!$link)
 {
   die('No se pudo conectar a la base de datos: ' . mysql_error());
 }

if (!mysql_select_db("GPSWEB",$link))
 {
   echo "Error seleccionando la base de datos.";
   exit();
 }

// Ya que seleccionamos la Base de Datos, realizamos un select a la tabla rastreos_gps
// Con la condicion de sea el registro, el equipo, y la fecha dia que seleccionamos
// ordenada por equipo, id_rastreo y fecha para mostrarla en la pagina

$query="SELECT * FROM `rastreos_gps` where id_rastreo>= ".$id_rastreo2." and equipo='".$equipo2."' and DATE_FORMAT(
fecha, '%Y-%m-%d' ) ='".substr($fecha2,0,10)."'  order by equipo,id_rastreo,fecha";
$result = mysql_query($query,$link);
if (!$result) {
   die('No se pudo ejecutar la sentencia:' . mysql_error());
}

// Por cada registro generaremos las coordenadas,
// con las caracteristicas siguientes "longitud,latitud,0"
$tagcoordenadas="";

// Para establecer graficamente una marca en la ruta inicial y final,
// guardamos en estas variables el primer y ultimo registro
$latitudini="";
$longitudini="";
$latitudfin="";
$longitudfin="";

// Exploramos registro por registro de la consulta solicitada
while($registro=mysql_fetch_array($result))
 {
 $palabras="";
 $signo="";
 $palabras=split(" ",$registro["latitud"]);
 // Si la palabra es Oeste, West y South o Sur, le pone un negativo al valor
 if($palabras[3]=="O" || $palabras[3]=="W"  || $palabras[3]=="S")
  {$signo="-";}
  // Google earth trabaja con grados decimales
  // Nuestra informacion tiene el siguiente formato: HH MM SS.SS
  // Lo modificamos de la siguiente manera
  // signo HH + (MM/60) + (SS.SS/3600)
  $latitud=$signo.($palabras[0]+($palabras[1]/60)+($palabras[2]/3600));

  //Obtenemos el registro Inicial y Final de la latitud de la ruta
  // Para poder colocar una marca en la aplicacion
  if($latitudini=="")
  {$latitudini=$latitud;}
 $latitudfin=$latitud;

 $palabras="";
 $signo="";
 $palabras=split(" ",$registro["longitud"]);
 // Si la palabra es Oeste, West y South o Sur, le pone un negativo al valor
 if($palabras[3]=="O" || $palabras[3]=="W"  || $palabras[3]=="S")
  {$signo="-";}
 // Google earth trabaja con grados decimales
 // Nuestra informacion tiene el siguiente formato: HH MM SS.SS
 // Lo modificamos de la siguiente manera
 // signo HH + (MM/60) + (SS.SS/3600)
```

```
 $longitud=$signo.($palabras[0]+($palabras[1]/60)+($palabras[2]/3600));

 //la variable tagcoordenadas, trae todos los registros de la ruta a seguir
 //teniendo la siguiente formato longitud, latitud, 0
 $tagcoordenadas=$tagcoordenadas."                                    ".$longitud.",".$latitud.",0  ";

 //Obtenemos el registro Inicial y Final de la Longitud de la ruta
 // Para poder colocar una marca en la aplicacion
 if($longitudini=="")
  {$longitudini=$longitud;}
 $longitudfin=$longitud;
}

//Teniendo Toda la informacion de la Ruta, generaremos el archivo KML

$filename="ruta.kml";
//Creamos o abrimos el archivo
if (!$fileh = fopen($filename, 'w'))
 {
  echo "<p>No se puede abrir el archivo para guardar su texto. Por favor, si el problema persiste contacte con el
administrador.</p>";
  exit;
 }
// Introducimos Datos con formato KML (Keyhole Markup Language)
// Es un lenguaje de marcado basado en XML para representar
// datos geograficos en tres dimensiones
fwrite($fileh,'<?xml version="1.0" encoding="UTF-8"?>'."\r\n");
fwrite($fileh,'<kml xmlns="http://www.opengis.net/kml/2.2" xmlns:gx="http://www.google.com/kml/ext/2.2"
xmlns:kml="http://www.opengis.net/kml/2.2" xmlns:atom="http://www.w3.org/2005/Atom">'."\r\n");
fwrite($fileh,'<Document>'."\r\n");
fwrite($fileh,'          <name>Ruta.kml</name>'."\r\n");
fwrite($fileh,'          <StyleMap id="msn_ylw-pushpin">'."\r\n");
fwrite($fileh,'                    <Pair>'."\r\n");
fwrite($fileh,'                              <key>normal</key>'."\r\n");
fwrite($fileh,'                              <styleUrl>#sn_ylw-pushpin</styleUrl>'."\r\n");
fwrite($fileh,'                    </Pair>'."\r\n");
fwrite($fileh,'                    <Pair>'."\r\n");
fwrite($fileh,'                              <key>highlight</key>'."\r\n");
fwrite($fileh,'                              <styleUrl>#sh_ylw-pushpin</styleUrl>'."\r\n");
fwrite($fileh,'                    </Pair>'."\r\n");
fwrite($fileh,'          </StyleMap>'."\r\n");
fwrite($fileh,'          <Style id="sh_ylw-pushpin">'."\r\n");
fwrite($fileh,'                    <IconStyle>'."\r\n");
fwrite($fileh,'                              <scale>1.3</scale>'."\r\n");
fwrite($fileh,'                              <Icon>'."\r\n");
fwrite($fileh,'                                        <href>http://maps.google.com/mapfiles/kml/pushpin/ylw-
pushpin.png</href>'."\r\n");
fwrite($fileh,'                              </Icon>'."\r\n");
fwrite($fileh,'                              <hotSpot x="20" y="2" xunits="pixels" yunits="pixels"/>'."\r\n");
fwrite($fileh,'                    </IconStyle>'."\r\n");
fwrite($fileh,'                    <LineStyle>'."\r\n");
fwrite($fileh,'                              <color>ff0000aa</color>'."\r\n");
fwrite($fileh,'                              <width>2</width>'."\r\n");
fwrite($fileh,'                    </LineStyle>'."\r\n");
fwrite($fileh,'          </Style>'."\r\n");
fwrite($fileh,'          <Style id="sn_ylw-pushpin">'."\r\n");
fwrite($fileh,'                    <IconStyle>'."\r\n");
fwrite($fileh,'                              <scale>1.1</scale>'."\r\n");
fwrite($fileh,'                              <Icon>'."\r\n");
fwrite($fileh,'                                        <href>http://maps.google.com/mapfiles/kml/pushpin/ylw-
pushpin.png</href>'."\r\n");
fwrite($fileh,'                              </Icon>'."\r\n");
fwrite($fileh,'                              <hotSpot x="20" y="2" xunits="pixels" yunits="pixels"/>'."\r\n");
fwrite($fileh,'                    </IconStyle>'."\r\n");
fwrite($fileh,'                    <LineStyle>'."\r\n");
fwrite($fileh,'                              <color>ff0000aa</color>'."\r\n");
fwrite($fileh,'                              <width>2</width>'."\r\n");
fwrite($fileh,'                    </LineStyle>'."\r\n");
fwrite($fileh,'          </Style>'."\r\n");
fwrite($fileh,'          <Style id="sh_ylw-circle">'."\r\n");
fwrite($fileh,'                    <IconStyle>'."\r\n");
fwrite($fileh,'                              <scale>1.3</scale>'."\r\n");
fwrite($fileh,'                              <Icon>'."\r\n");
fwrite($fileh,'                                        <href>http://maps.google.com/mapfiles/kml/paddle/ylw-
circle.png</href>'."\r\n");
fwrite($fileh,'                              </Icon>'."\r\n");
```

```
fwrite($fileh,'                                <hotSpot x="32" y="1" xunits="pixels" yunits="pixels"/>'."\r\n");
fwrite($fileh,'                        </IconStyle>'."\r\n");
fwrite($fileh,'                        <ListStyle>'."\r\n");
fwrite($fileh,'                                <ItemIcon>'."\r\n");
fwrite($fileh,'                                        <href>http://maps.google.com/mapfiles/kml/paddle/ylw-circle-
lv.png</href>'."\r\n");
fwrite($fileh,'                                </ItemIcon>'."\r\n");
fwrite($fileh,'                        </ListStyle>'."\r\n");
fwrite($fileh,'        </Style>'."\r\n");
fwrite($fileh,'        <Style id="sn_ylw-circle">'."\r\n");
fwrite($fileh,'                        <IconStyle>'."\r\n");
fwrite($fileh,'                                <scale>1.1</scale>'."\r\n");
fwrite($fileh,'                                <Icon>'."\r\n");
fwrite($fileh,'                                        <href>http://maps.google.com/mapfiles/kml/paddle/ylw-
circle.png</href>'."\r\n");
fwrite($fileh,'                                </Icon>'."\r\n");
fwrite($fileh,'                                <hotSpot x="32" y="1" xunits="pixels" yunits="pixels"/>'."\r\n");
fwrite($fileh,'                        </IconStyle>'."\r\n");
fwrite($fileh,'                        <ListStyle>'."\r\n");
fwrite($fileh,'                                <ItemIcon>'."\r\n");
fwrite($fileh,'                                        <href>http://maps.google.com/mapfiles/kml/paddle/ylw-circle-
lv.png</href>'."\r\n");
fwrite($fileh,'                                </ItemIcon>'."\r\n");
fwrite($fileh,'                        </ListStyle>'."\r\n");
fwrite($fileh,'        </Style>'."\r\n");
fwrite($fileh,'        <StyleMap id="msn_ylw-circle">'."\r\n");
fwrite($fileh,'                        <Pair>'."\r\n");
fwrite($fileh,'                                <key>normal</key>'."\r\n");
fwrite($fileh,'                                <styleUrl>#sn_ylw-circle</styleUrl>'."\r\n");
fwrite($fileh,'                        </Pair>'."\r\n");
fwrite($fileh,'                        <Pair>'."\r\n");
fwrite($fileh,'                                <key>highlight</key>'."\r\n");
fwrite($fileh,'                                <styleUrl>#sh_ylw-circle</styleUrl>'."\r\n");
fwrite($fileh,'                        </Pair>'."\r\n");
fwrite($fileh,'        </StyleMap>'."\r\n");
fwrite($fileh,'        <Style id="sh_ylw-square">'."\r\n");
fwrite($fileh,'                        <IconStyle>'."\r\n");
fwrite($fileh,'                                <scale>1.3</scale>'."\r\n");
fwrite($fileh,'                                <Icon>'."\r\n");
fwrite($fileh,'                                        <href>http://maps.google.com/mapfiles/kml/paddle/ylw-
square.png</href>'."\r\n");
fwrite($fileh,'                                </Icon>'."\r\n");
fwrite($fileh,'                                <hotSpot x="32" y="1" xunits="pixels" yunits="pixels"/>'."\r\n");
fwrite($fileh,'                        </IconStyle>'."\r\n");
fwrite($fileh,'                        <ListStyle>'."\r\n");
fwrite($fileh,'                                <ItemIcon>'."\r\n");
fwrite($fileh,'                                        <href>http://maps.google.com/mapfiles/kml/paddle/ylw-square-
lv.png</href>'."\r\n");
fwrite($fileh,'                                </ItemIcon>'."\r\n");
fwrite($fileh,'                        </ListStyle>'."\r\n");
fwrite($fileh,'        </Style>'."\r\n");
fwrite($fileh,'        <StyleMap id="msn_ylw-square">'."\r\n");
fwrite($fileh,'                        <Pair>'."\r\n");
fwrite($fileh,'                                <key>normal</key>'."\r\n");
fwrite($fileh,'                                <styleUrl>#sn_ylw-square</styleUrl>'."\r\n");
fwrite($fileh,'                        </Pair>'."\r\n");
fwrite($fileh,'                        <Pair>'."\r\n");
fwrite($fileh,'                                <key>highlight</key>'."\r\n");
fwrite($fileh,'                                <styleUrl>#sh_ylw-square</styleUrl>'."\r\n");
fwrite($fileh,'                        </Pair>'."\r\n");
fwrite($fileh,'        </StyleMap>'."\r\n");
fwrite($fileh,'        <Style id="sn_ylw-square">'."\r\n");
fwrite($fileh,'                        <IconStyle>'."\r\n");
fwrite($fileh,'                                <scale>1.1</scale>'."\r\n");
fwrite($fileh,'                                <Icon>'."\r\n");
fwrite($fileh,'                                        <href>http://maps.google.com/mapfiles/kml/paddle/ylw-
square.png</href>'."\r\n");
fwrite($fileh,'                                </Icon>'."\r\n");
fwrite($fileh,'                                <hotSpot x="32" y="1" xunits="pixels" yunits="pixels"/>'."\r\n");
fwrite($fileh,'                        </IconStyle>'."\r\n");
fwrite($fileh,'                        <ListStyle>'."\r\n");
fwrite($fileh,'                                <ItemIcon>'."\r\n");
fwrite($fileh,'                                        <href>http://maps.google.com/mapfiles/kml/paddle/ylw-square-
lv.png</href>'."\r\n");
fwrite($fileh,'                                </ItemIcon>'."\r\n");
fwrite($fileh,'                        </ListStyle>'."\r\n");
```

```
fwrite($fileh,'		</Style>'."\r\n");
fwrite($fileh,'		<Placemark>'."\r\n");
fwrite($fileh,'			<name>Ruta</name>'."\r\n");
fwrite($fileh,'			<styleUrl>#msn_ylw-pushpin</styleUrl>'."\r\n");
fwrite($fileh,'			<LineString>'."\r\n");
fwrite($fileh,'				<tessellate>1</tessellate>'."\r\n");
fwrite($fileh,'				<coordinates>'."\r\n");
fwrite($fileh,'				'.$tagcoordenadas."\r\n");
fwrite($fileh,'				 </coordinates>'."\r\n");
fwrite($fileh,'			</LineString>'."\r\n");
fwrite($fileh,'		</Placemark>'."\r\n");
fwrite($fileh,'		'."\r\n");
fwrite($fileh,'		<Placemark>'."\r\n");
fwrite($fileh,'			<name>Inicio</name>'."\r\n");
fwrite($fileh,'			<description>Inicio de Ruta'."\r\n");
fwrite($fileh,'			Dia: '.substr($fecha2,0,10)."\r\n");
fwrite($fileh,'			</description>'."\r\n");
fwrite($fileh,'			<LookAt>'."\r\n");
fwrite($fileh,'				<longitude>'.$longitudini.'</longitude>'."\r\n");
fwrite($fileh,'				<latitude>'.$latitudini.'</latitude>'."\r\n");
fwrite($fileh,'				<altitude>0</altitude>'."\r\n");
fwrite($fileh,'				<range>327.6664158876513</range>'."\r\n");
fwrite($fileh,'				<tilt>15.84192145146256</tilt>'."\r\n");
fwrite($fileh,'				<heading>5.631553392556575e-006</heading>'."\r\n");
fwrite($fileh,'				<altitudeMode>relativeToGround</altitudeMode>'."\r\n");
fwrite($fileh,'				<gx:altitudeMode>relativeToSeaFloor</gx:altitudeMode>'."\r\n");
fwrite($fileh,'			</LookAt>'."\r\n");
fwrite($fileh,'			<styleUrl>#msn_ylw-circle</styleUrl>'."\r\n");
fwrite($fileh,'			<Point>'."\r\n");
fwrite($fileh,'				<coordinates>'.$longitudini.','.$latitudini.',0 </coordinates>'."\r\n");
fwrite($fileh,'			</Point>'."\r\n");
fwrite($fileh,'		</Placemark>'."\r\n");
fwrite($fileh,"."\r\n");
fwrite($fileh,'		<Placemark>'."\r\n");
fwrite($fileh,'			<name>Fin</name>'."\r\n");
fwrite($fileh,'			<description>Fin de Ruta'."\r\n");
fwrite($fileh,'			Dia: '.substr($fecha2,0,10)."\r\n");
fwrite($fileh,'			</description>'."\r\n");
fwrite($fileh,'			<LookAt>'."\r\n");
fwrite($fileh,'				<longitude>'.$longitudfin.'</longitude>'."\r\n");
fwrite($fileh,'				<latitude>'.$latitudfin.'</latitude>'."\r\n");
fwrite($fileh,'				<altitude>0</altitude>'."\r\n");
fwrite($fileh,'				<range>269.5626065115161</range>'."\r\n");
fwrite($fileh,'				<tilt>8.545643487755228</tilt>'."\r\n");
fwrite($fileh,'				<heading>7.503555492219565e-007</heading>'."\r\n");
fwrite($fileh,'				<altitudeMode>relativeToGround</altitudeMode>'."\r\n");
fwrite($fileh,'				<gx:altitudeMode>relativeToSeaFloor</gx:altitudeMode>'."\r\n");
fwrite($fileh,'			</LookAt>'."\r\n");
fwrite($fileh,'			<styleUrl>#msn_ylw-square</styleUrl>'."\r\n");
fwrite($fileh,'			<Point>'."\r\n");
fwrite($fileh,'				<coordinates>'.$longitudfin.','.$latitudfin.',0 </coordinates>'."\r\n");
fwrite($fileh,'			</Point>'."\r\n");
fwrite($fileh,'		</Placemark>'."\r\n");
fwrite($fileh,"."\r\n");
fwrite($fileh,'		</Document>'."\r\n");
fwrite($fileh,'</kml>'."\r\n");

//Cerramos el Archivo
fclose($fileh);


 //Liberamos los Recurso de Base de datos y consulta
mysql_free_result($result);

mysql_close($link);

?>

<HTML>
<HEAD>
<TITLE>GPSWEB: Generacion de ruta</TITLE>
<style type="text/css">
<!--
.style3 {
		font-size: 18px;
		font-weight: bold;
```

```
            color: #FFFF00;
}
.Estilo1 {font-size: xx-small}
-->
</style>
</HEAD>
<body>
  <div align="center">
    <p>Al bajar el archivo a su PC, asegurese que tiene la extensi&oacute;n KML</p>
    <p class="Estilo1"> &quot;Keyhole Markup Languaje&quot;. Lenguaje de marcado basado en XML para representar datos
geogr&aacute;ficos en tres   dimensiones</p>
    <p><a href="ruta.kml"><strong>Ruta.kml</strong></a>
      <br>
       </p>
  </div>
</BODY>
</HTML>
```

BIBLIOGRAFIA

GPS & GLONASS, Descripción y Aplicaciones, Madrid España 1998
http://www.isa.cie.uva.es/gps/
http://geeks.ms/blogs/jmtorres/archive/2009/01/21/cuan-preciso-es-la-senyal-de-nuestros-receptores-gps.aspx
http://www.elgps.com/documentos/comofuncionagps/comofuncionagps.html
http://electronics.howstuffworks.com/gadgets/travel/gps.htm
http://es.wikipedia.org/wiki/GPS
http://msdn.microsoft.com/es-mx/library/fx6bk1f4(VS.80).aspx
http://www.nmea.org/content/nmea_standards/nmea_083_v_400.asp
http://earth.google.com/userguide/v4/ug_editing.html
http://maps.google.com
http://code.google.com/intl/es/apis/kml/documentation/kmlreference.html
http://www.elaion.com.mx/gpsweb/rastreoweb.php
https://elaion.com.mx:8443