

Capítulo 4

Georap Compact Edition

“Tenía un maravilloso talento para empacar estrechamente el pensamiento, haciéndolo portable ”

Thomas Babington Macaulay

4.1 Introducción

Tal como se explico en el capítulo anterior este proyecto se concebió como una mejora a la solución global del posicionamiento, requeríamos un software capaz de comunicarse con un receptor GPS de código, obtener nuestra posición, conectarse con una estación que envíe correcciones RTCM, realizar estas correcciones mejorando el error de posicionamiento, enviar esa posición a una central de control que nos localiza en un mapa de Google Maps y almacenar esta posición en una base de datos para ser consultada de forma gráfica a través de Internet por cualquier persona incluido el propio usuario. Estas tareas se dividieron en dos proyectos diferentes el software que realiza las correcciones y el servidor que muestra la posición y la almacena en una base de datos. El motivo de esta división es la división de funciones entre el usuario móvil y el controlador. El usuario-móvil dispone de un Pocket PC, un dispositivo muy limitado que en ningún caso asemeja sus posibilidades a las de un PC, por tanto el software implementado debía cumplir unos requisitos muy exactos, que nos permitieran cumplir exactamente y únicamente el objetivo real de nuestro proyecto, crear un sistema portable capaz de realizar correcciones en cualquier lugar del planeta. En este capítulo vamos a tratar el desarrollo de este software, GEORAP CE, y en el siguiente capítulo desarrollaremos la parte del servidor.

GEORAP CE cumple unas funciones muy concretas, se comunica con un dispositivo GPS a través de Bluetooth (también permite la conexión por puerto serie a estos dispositivos GPS siempre que la Pocket PC disponga de esta conexión en su base de alimentación) imprescindible para poder desplazarnos sin necesidad de cables y poder utilizar el software mientras nos desplazamos, al obtener las posiciones crea un socket para comunicarse con una estación RTCM más cercana, recibe las correcciones

y las efectúa sobre las posiciones anteriormente obtenidas mostrándolas en la pantalla de la PDA.

4.2 Descripción de tecnologías

El software de GEORAP CE ha sido desarrollado en Java 2 SE, esta decisión se basó en que esta parte del proyecto consiste en la adaptación para PDA de un software ya existente GEORAP, que fue programado en este lenguaje. Sin embargo, durante el desarrollo del proyecto, se vio que GEORAP debido a varias razones como su peso, requisitos de memoria RAM y formato de mapas utilizados entre otros, no fue programado con la posibilidad de adaptarlo a un dispositivo con las limitaciones de una PDA, y es posible que hubiera sido mejor idea haber utilizado un lenguaje mas orientado a la programación en pequeños dispositivos como Java 2 ME o C Embedded Tools, lo que nos habría aportado mayores ventajas como son: la posibilidad de portarlo a otros dispositivos aún mas sencillos como terminales móviles, o el empleo de imágenes de bajo peso para dispositivos de bajos recursos.

Nuestro software se comunica con el receptor GPS a través del protocolo NMEA 0183, y recibe las correcciones de las estaciones de referencia en formato RTCM.

4.2.1 Lenguaje de programación Java

4.2.1.1 MOTIVACION DE JAVA

Las principales razones que llevaron a crear Java son:

1. Creciente necesidad de interfaces mucho más cómodas e intuitivas que los sistemas de ventanas que proliferaban hasta el momento.
2. Fiabilidad del código y facilidad de desarrollo. Gosling, actual vicepresidente de Sun Microsystems, observó que muchas de las características que ofrecían C o C++ aumentaban de forma alarmante el gran coste de pruebas y depuración. Por ello se decidió a crear un lenguaje de programación donde intentaba solucionar los fallos que encontraba en C++.
3. Enorme diversidad de controladores electrónicos. Los dispositivos electrónicos se controlan mediante la utilización de microprocesadores de bajo precio y reducidas prestaciones, que varían cada poco tiempo y que utilizan diversos conjuntos de instrucciones. Java permite escribir un código común para todos los dispositivos.

4.2.1.2 CARACTERÍSTICAS Y FUNCIONAMIENTO DE JAVA

El lenguaje fue diseñado con las siguientes características en mente:

1. Simple. Elimina la complejidad de los lenguajes como "C" y da paso al contexto de los lenguajes modernos orientados a objetos.
2. Sintaxis conocida. Como la mayoría de los programadores están acostumbrados a programar en C o en C++, la sintaxis de Java es muy similar al de estos.
3. Robusto. El sistema de Java maneja directamente la memoria de la computadora. No hay que preocuparse por punteros, memoria que no se esté utilizando, etc. Java realiza todo esto sin necesidad de que uno se lo indique.
4. Seguro. El sistema de Java tiene ciertas políticas que evitan que se puedan codificar virus con este lenguaje. Existen muchas restricciones, especialmente para los “applets⁶”, que limitan lo que se puede y no puede hacer con los recursos críticos de una computadora.
5. Portable. Como el código compilado de Java (conocido como *byte code*) es interpretado, un programa compilado de Java puede ser utilizado por cualquier computadora que tenga implementado el interprete de Java.

(6) Pequeño programa hecho en Java

6. Independiente a la arquitectura. Al compilar un programa en Java, el código resultante es un *byte code*. Este código es interpretado por diferentes computadoras de igual manera, solamente hay que implementar un intérprete para cada plataforma. De esa manera Java logra ser un lenguaje que no depende de una arquitectura computacional definida.
7. Multihilo. Un lenguaje que soporta múltiples hilos es un lenguaje que puede ejecutar diferentes líneas de código al mismo tiempo.
8. Interpretado. Java corre en máquina virtual, por lo tanto es interpretado.
9. Dinámico. Java no requiere que compile todas las clases de un programa para que este funcione. Si realizas una modificación a una clase Java se encarga de realizar un Dynamic Bynding o un Dynamic Loading para encontrar las clases.

Java puede funcionar como una aplicación sola o como un "applet". Los applets de Java se pueden "pegar" a una página de Web (HTML), y con esto puedes tener un programa que cualquier persona que tenga un navegador compatible podrá usar.

Java funciona de la siguiente manera: El compilador de Java deja el programa en un Pseudo-código (no es código maquina) y luego el intérprete de Java ejecuta el programa (lo que se conoce como el "Java Virtual Machine"). Por eso Java es multiplataforma, existe un intérprete para cada máquina diferente.

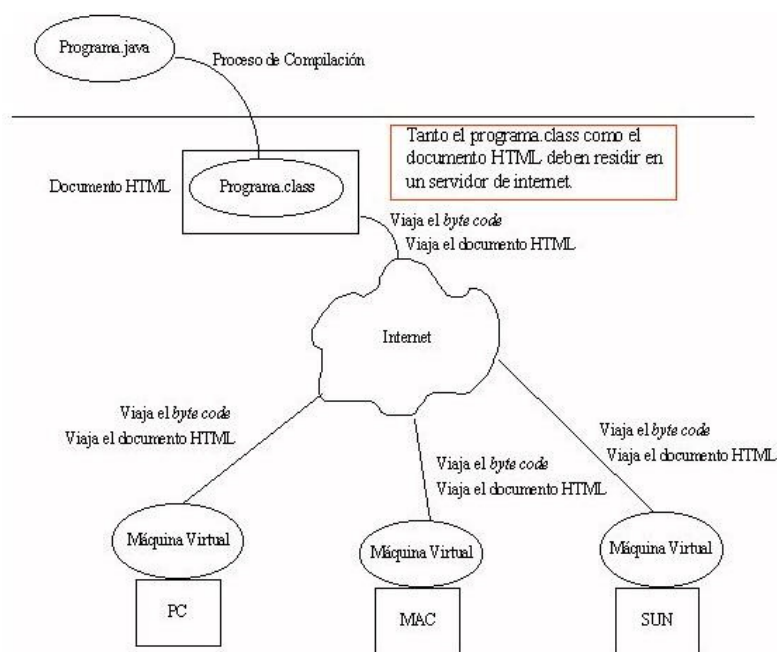


Figura2. Funcionamiento de Java, mismo código para múltiples máquinas.

4.2.1.3 VERSIONES DE JAVA

La empresa Sun, dispuesta a proporcionar las herramientas necesarias para cubrir las necesidades de todos los usuarios, creó distintas versiones de Java de acuerdo a las necesidades de cada uno. Según esto nos encontramos con que el paquete Java 2 lo podemos dividir en 3 ediciones distintas. J2SE (Java Standard Edition) orientada al desarrollo de aplicaciones independientes de la plataforma, J2EE (Java Enterprise Edition) orientada al entorno empresarial y J2ME (Java Micro Edition) orientada a dispositivos con capacidades restringidas. Veamos cuáles son las características de cada una de las versiones:

Java 2 Platform, Standard Edition (J2SE): Esta edición de Java es la que en cierta forma recoge la iniciativa original del lenguaje Java. Tiene las siguientes características:

- Inspirado inicialmente en C++, pero con componentes de alto nivel, como soporte nativo de cadenas de caracteres y recolector de basura.
- Código independiente de la plataforma, precompilado a bytecodes intermedio y ejecutado en el cliente por una JVM (Java Virtual Machine).
- Modelo de seguridad tipo *sandbox* proporcionado por la JVM.
- Abstracción del sistema operativo subyacente mediante un juego completo de APIs de programación.

Esta versión de Java contiene el conjunto básico de herramientas usadas para desarrollar Java Applets, así como las APIs orientadas a la programación de aplicaciones de usuario final: Interfaz gráfica de usuario, multimedia, redes de comunicación, etc.

Java 2 Platform, Enterprise Edition (J2EE): Esta versión está orientada al entorno empresarial. El software empresarial tiene unas características propias marcadas: está pensado no para ser ejecutado en un equipo, sino para ejecutarse sobre una red de ordenadores de manera distribuida y remota mediante EJBs (Enterprise Java Beans). De hecho, el sistema se monta sobre varias unidades o aplicaciones. En muchos casos, además, el software empresarial requiere que se sea capaz de integrar datos provenientes de entornos heterogéneos. Esta edición está orientada especialmente al desarrollo de servicios web, servicios de nombres, persistencia de objetos, XML, autenticación, APIs para la gestión de transacciones, etc. El cometido

de esta especificación es ampliar la J2SE para dar soporte a los requisitos de las aplicaciones de empresa.

Java 2 Platform, Micro Edition (J2ME): Esta versión de Java está enfocada a la aplicación de la tecnología Java en dispositivos electrónicos con capacidades computacionales y gráficas muy reducidas, tales como teléfonos móviles, PDAs o electrodomésticos inteligentes. Esta edición tiene unos componentes básicos que la diferencian de las otras versiones, como el uso de una máquina virtual denominada KVM (Kilo Virtual Machine, debido a que requiere sólo unos pocos Kilobytes de memoria para funcionar) en vez del uso de la JVM clásica, inclusión de un pequeño y rápido recolector de basura y otras diferencias que ya iremos viendo más adelante.

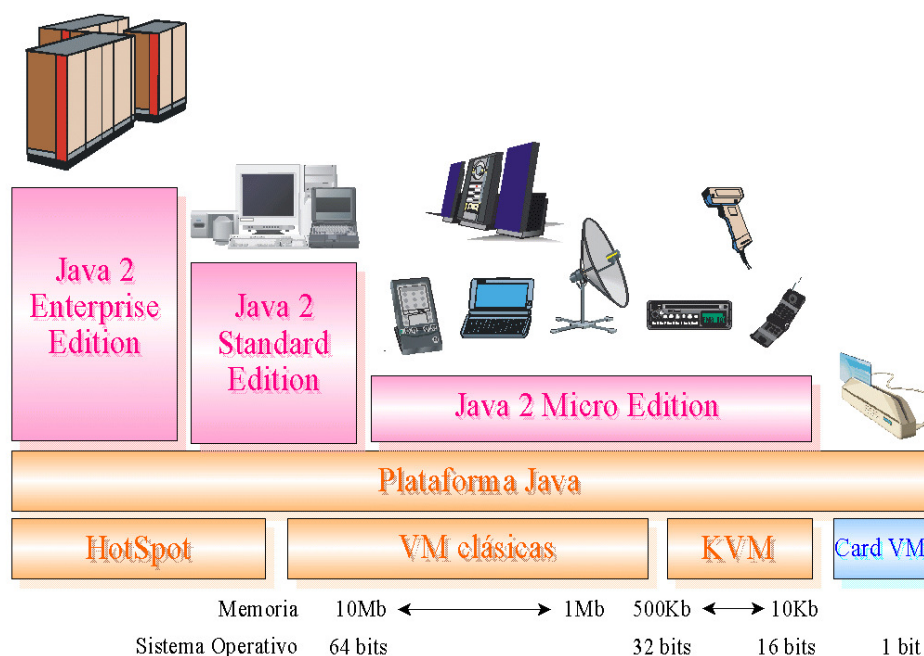


Figura 3. Arquitectura de la plataforma Java 2 de Sun.

4.3 Descripción global del programa

4.3.1 Descripción infraestructura

El programa GEORAP CE se distribuye en un solo archivo georapce.jar, que se puede ejecutar en cualquier máquina que tenga instalada una maquina virtual de java, este software ha sido concebido para ser usado en un Pocket PC, por lo que si lo ejecutamos en un PC, a pesar de que funcionará sin ningún problema, puede resultar de una interfaz muy simple.

El archivo georapce.jar es un archivo ejecutable comprimido que ha sido creado a partir de todas las clases que se han programado independientemente, vamos a ver la estructura de interna, georapce.jar:

- **GPS.class:** *Contiene el método principal y lanza el programa.*
- **Apariencia.class:** *Define el aspecto visual del programa.*
- **Autor.class:** *Muestra los créditos del programa.*
- **Paquete Bluetooth:** *Permite la comunicación mediante interfaz Bluetooth. Está compuesto por:*
 - Conexion.class
 - Bluetooth.class
 - Bluetooth\$Listener.class
 - Bluetooth\$CheckBoxListener.class

- **Paquete Datos:** *Permite el intercambio de datos entre las distintas clases del programa. Compuesto por:*
 - Datos.class
- **Paquete NMEA:** *Lee los datos recibidos del receptor GPS. Compuesto por:*
 - NMEA.class
- **Paquete Operaciones:** *Realiza las transformaciones de coordenadas y resto de operaciones (medias, desviaciones típicas...). Compuesto por:*
 - Transformar.class
- **Paquete Principal:** *Muestra los datos por la interfaz gráfica. Compuesto por:*
 - Estadisticas.class
 - Geoestadisticas.class
 - Infocoord.class
 - Inforest.class
 - Informacion.class
 - Oyente.class
 - Principal.class

- Satelites.class
 - UTMestadisticas.class
 - XYZestadisticas.class
- **Paquete RS232:** *Proporciona la posibilidad de conectar por puerto serie. Compuesto por:*
 - ConexionSerie.class
 - ExcepcionSerie.class
 - ParametrosSerie.class
 - Rs232.class
 - VerDatos.class
 - ConfSerie.class
 - ConfSerie\$CheckBoxListener.class
- **Paquete RTCM:** *Controla todo lo referente a la conexión con la estación de referencia y la recepción de correcciones. Compuesto por:*
 - AutoTcpIp.class
 - ComTcpIp.class
 - DatosRTCM.class

- Fichero.class
- RTCM.class
- TcpIp.class
- Datos_Estaciones.txt

A continuación vamos a comenzar a explicar que labor desempeña cada una de las clases implementadas en este software.

4.3.2 Descripción detallada y visual de clases

4.3.2.1 GPS.class

Esta clase invoca el método “main” o principal, por tanto es la clase que comienza la ejecución del programa, lanza el programa, nos muestra por pantalla la interfaz gráfica de éste y se queda a la espera de que realicemos alguna acción, ya sea cambiar de pestaña, pulsar algún botón o cerrar el programa.

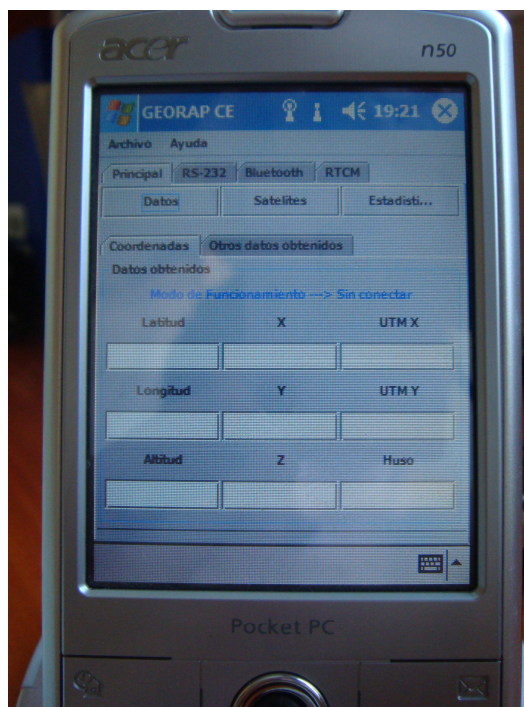


Figura 4. Pantalla de inicio de GEORAP CE

4.3.2.2 Apariencia.class

Esta clase es bastante sencilla y se limita a definir el aspecto visual del programa, en ella se definen los colores de la interfaz, así como los diferentes tipos y formatos de letra.

4.3.2.3 Autor.class

Esta clase se ejecuta al pulsar el botón “Acerca de GEORAP CE” situado en el menú ayuda y lanza una pantalla con la información acerca del programa.



Figura 5. Pantalla mostrada por la clase Autor.

4.3.2.4 PAQUETE BLUETOOTH

El paquete Bluetooth está formado por cuatro clases: Conexión.class, Bluetooth.class, Bluetooth&Listener.class y Bluetooth&CheckBoxListener.class. Este paquete nos proporciona la interfaz de comunicación principal de nuestro programa, permitiéndonos conectarnos a un receptor GPS de forma inalámbrica y comunicarnos con él.

A esta utilidad del programa podemos acceder pulsando sobre la pestaña Bluetooth, mostrándose así la interfaz gráfica con la que podemos establecer la conexión.

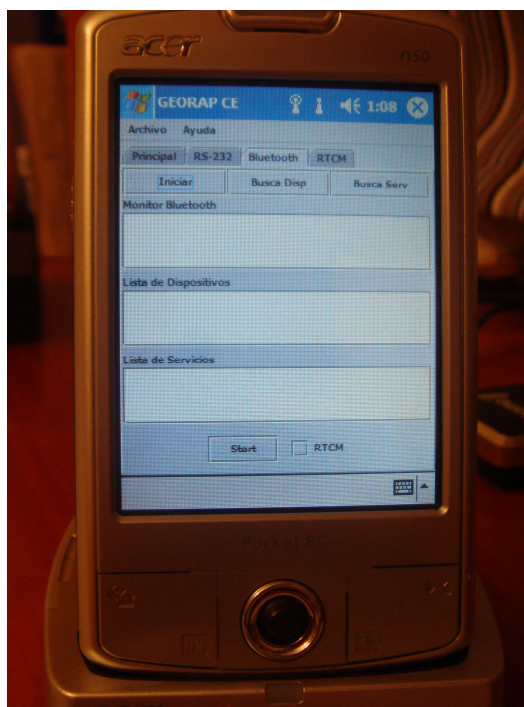


Figura 6. Interfaz de la pestaña Bluetooth

Como podemos ver esta interfaz presenta cuatro botones, tres cuadros de diálogo y un check-box. La funcionalidad de cada uno de éstos es la siguiente:

- El botón *Iniciar* se encargará de buscar el dispositivo bluetooth local del equipo y activarlo, este proceso se irá detallando en el cuadro de diálogo “Monitor Bluetooth”
- El botón *Busca Disp* se encarga de buscar dispositivos cercanos que tengan activada con la conexión bluetooth, y los mostrará en el cuadro de diálogo “Lista de dispositivos”.
- El botón *Busca Serv* se encargará de buscar que servicios de comunicación ofrece el dispositivo que hayamos seleccionado en el

cuadro de diálogo “Lista de Dispositivos”, y los mostrará en “Lista de Servicios”.

- El botón *Start/Stop* dará comienzo a la comunicación Bluetooth, una vez hayamos seleccionado tanto el dispositivo como el servicio. En el cuadro de diálogo “Monitor Bluetooth” se irán mostrando todos los mensajes NMEA recibidos del receptor GPS al que nos hayamos conectado. Mientras esté conectado el botón pasará a tener la palabra *Stop* y pulsando sobre él se terminará la conexión Bluetooth y volverá a cambiar a la palabra *Start*.
- Por último la check-box *RTCM* deberá estar activada si queremos poder realizar correcciones diferenciales sobre los datos obtenidos.

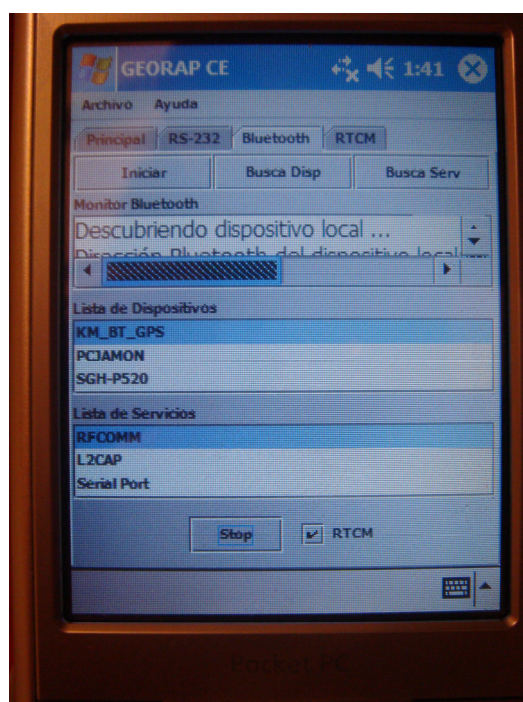


Figura 7. Interfaz Bluetooth conectada al GPS

4.3.2.5 PAQUETE RS232

Este paquete está compuesto por las siguientes clases: `ConexionSerie.class`, `ExcepcionSerie.class`, `ParametrosSerie.class`, `Rs232.class`, `VerDatos.class`, `ConfSerie.class` y `ConfSerie$CheckBoxListener.class`.

Estas clases nos proporcionan la posibilidad de establecer una conexión con nuestro receptor GPS a través del puerto serie, así como la correspondiente interfaz gráfica para gestionar y supervisar esta conexión. La posibilidad de conectar mediante una interfaz serie ha sido incluida en el programa mas con una finalidad de prueba y testeo del sistema, que como un efecto práctico, ya que las Pocket PC no disponen de un puerto serie en si mismas, sino que seria necesario, por ejemplo, disponer de ésta en su base/cargador para poder conectarla mediante esta interfaz (dependiendo el modelo). Como ya hemos venido hablando uno de los objetivos de este proyecto era la posibilidad de poder utilizar el software en cualquier parte ya estemos haciendo senderismo, en nuestro vehículo, en una ruta en bicicleta... y obviamente en estos supuestos no seria de utilidad tener que ir cargando con dos dispositivos conectados por un puerto serie. Pero aún así nos ha parecido útil poder dotar de esta interfaz ya sea para realizar pruebas de funcionamiento o para usarlo con navegadores que no dispongan de conexión bluetooth en ubicaciones puntuales.

Vamos a ver a continuación la interfaz gráfica de la pestaña RS232 así como la funcionalidad de cada uno de los elementos que la integran. Esta pantalla muestra dos menús diferentes según el botón que hayamos pulsado *Config* o *I/O*. Por defecto se muestra el menú del botón *Config*.

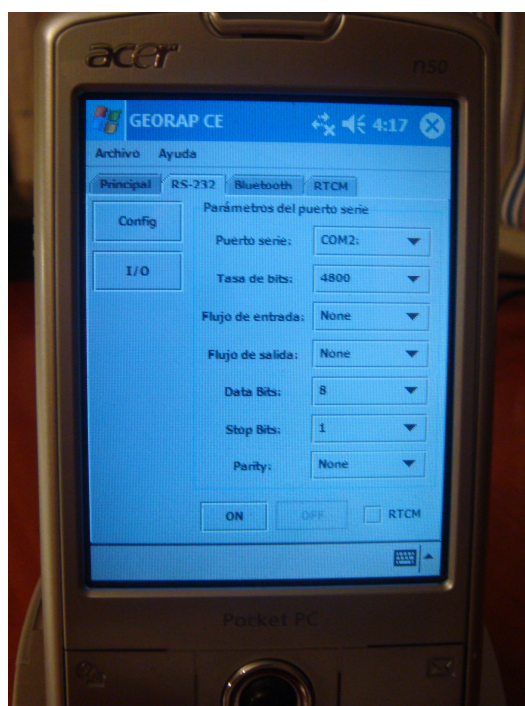


Figura 8. Interfaz gráfica por defecto de la pestaña RS-232.

En la pantalla de configuración aparece un donde podremos seleccionar los parámetros que definen una comunicación serie, a saber, Nombre del puerto, Tasa de bits, flujos de entrada/salida, Bits de datos, stop y paridad. Una vez definamos todos estos parámetros podremos comenzar la comunicación pulsando el botón ON y detenerla pulsando OFF. Además podemos habilitar el envío de correcciones RTCM habilitando la casilla diseñada a tal efecto.

Una vez configurada y establecida la conexión podemos visualizar el intercambio de datos pulsando el botón I/O, mediante el cual aparecerá un cuadro de diálogo en el que podremos seleccionar ver los datos enviados o los recibidos.

Tal y como sucedía con la interfaz de comunicación Bluetooth el intercambio de mensajes se basará en el protocolo NMEA. Para tratar estos mensajes ha sido definida una clase específica que se tratará a continuación.

4.3.2.6 PAQUETE NMEA

El paquete NMEA consta de una única clase que realiza una función muy específica. Una vez realizada una conexión entre nuestro Pocket PC y nuestro receptor GPS, con cualquiera de las dos interfaces definidas anteriormente, los datos son recibidos del receptor GPS vendrán en formato NMEA. Ya definimos en un apartado anterior cual es la sintaxis de este tipo de mensajes. La clase NMEA se ha programado para leer y reconocer estos mensajes, así como para extraer de ellos la información que nos interesa, tal como las coordenadas de posición y algunos otros datos que nos serán de gran utilidad como los satélites de los que estamos recibiendo y su relación SNR (señal a ruido). Toda esta información es mostrada en la sus respectivos campos dentro de la pestaña “Principal”. Además de esto la clase NMEA también va imprimiendo por pantalla la información tal y como la va recibiendo siempre que estemos trabajando en modo consola, esto puede ayudarnos a saber porque no se recibió un dato o si en alguna línea NMEA algún campo vino vacío y por eso no obtenemos el dato en la pestaña “Principal”.

4.3.2.7 PAQUETE RTCM

Este paquete se compone de seis clases (AutoTcpIp.class, ComTcpIp.class, DatosRTCM.class, Fichero.class, RTCM.class y TcpIp.class), y de un archivo de texto: Datos_Estaciones.txt.

El paquete RTCM es el encargado de establecer la conexión TCP/IP con la estación de referencia y recibir estas correcciones, su funcionamiento es el siguiente.

Una vez hayamos establecido una conexión con nuestro receptor GPS, ya sea mediante Bluetooth o mediante el puerto serie, y siempre y cuando hayamos habilitado la casilla *RTCM*, debemos dirigirnos a la pestaña “RTCM” de la interfaz principal del programa desplegándonos entonces una ventana tal como la mostrada en la figura 9.

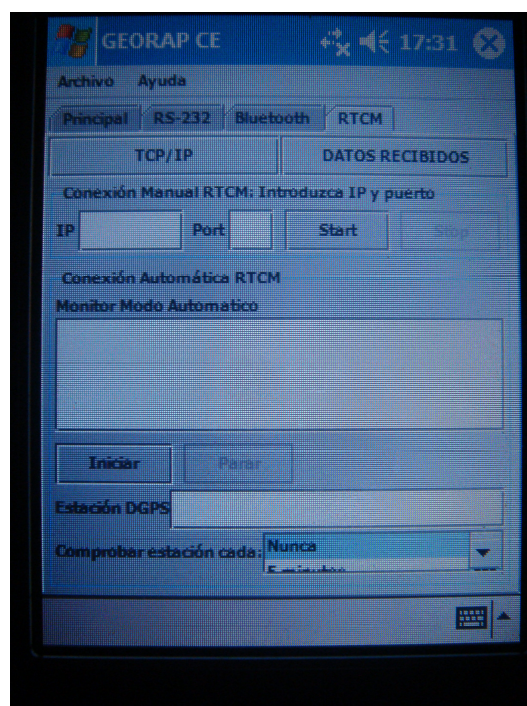


Figura 9. Interfaz gráfica de la pestaña RTCM.

Una vez aquí el programa nos da dos opciones, o bien podemos introducir de forma manual la IP y el puerto donde está escuchando el servidor de correcciones RTCM, o podemos comenzar el modo automático.

Si optamos por el modo manual simplemente debemos conocer la IP de la estación de referencia y el puerto por el cual se está escuchando, una vez introducidos

pulsaremos *Start*, esto hará que nuestro programa abra un socket y comience la comunicación con la estación, para desconectarnos de ésta solo tendremos que pulsar el botón *Stop*, todas las acciones llevadas a cabo se irán registrando en el cuadro de diálogo inferior.

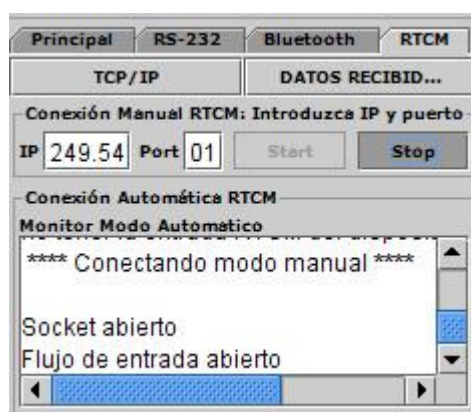


Figura 10. Conectado a la estación RTCM de forma manual

La otra opción es conectar de forma automática, de esa manera solo tendremos que preocuparnos de marcar cada cuanto tiempo queremos que se compruebe que estamos conectados a la estación más cercana y pulsar el botón *Iniciar*, de este modo nuestro programa leerá el archivo *Datos_Estaciones.txt* para obtener las IP y puertos a los que conectarse, además nosotros podremos modificar este archivo para eliminar estaciones obsoletas o añadir nuevas.

4.3.2.8 PAQUETE DATOS

En este paquete está definida una única clase, la clase *Datos*, esta clase sirve para que todas las demás puedan intercambiar datos entre si, en ella están declaradas todas las posibles variables que interesan a más de una clase como pueden ser los

valores de latitud, longitud, altitud... Además de todos los métodos get/set para obtener el valor almacenado en dichas variables o establecerlo.

4.3.2.9 PAQUETE PRINCIPAL

El paquete Principal está formado por diez clases. Este paquete se encarga de mostrar todos los datos que hemos obtenido de nuestra comunicación con el receptor GPS, así como otros datos calculados a través de las clases del paquete Operaciones, que veremos más adelante.

La interfaz gráfica que proporciona este conjunto de clases está dividida en tres pantallas (Datos, Satélites y Estadísticas), a las cuales se accede a través de sus correspondientes botones.

La interfaz gráfica mostrada por defecto es la correspondiente a los datos, esta interfaz está compuesta a su vez de dos pestañas. La primera de ellas muestra los datos referentes a nuestra posición en diferentes sistemas de coordenadas: geográficas en WGS84, geocéntricas (XYZ) y UTM, así como el modo de funcionamiento del dispositivo. La segunda muestra otro conjunto de datos también obtenidos de los mensajes NMEA como son la ondulación del geoide, PDOP, HDOP, VDOP...

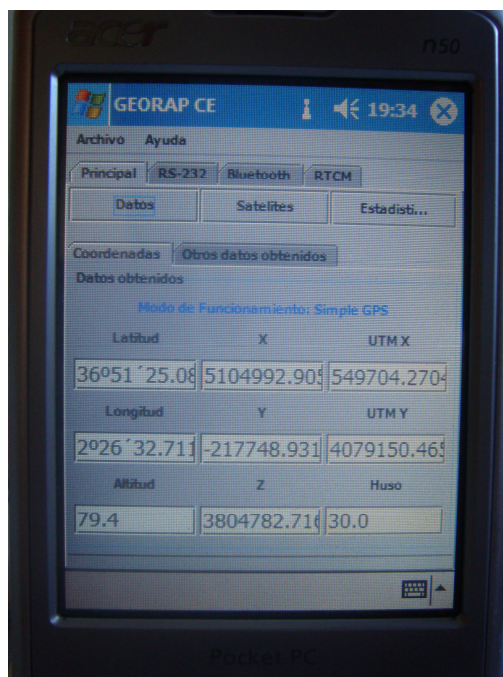


Figura 11. Pantalla que muestra las coordenadas obtenidas en distintos sistemas de referencia.

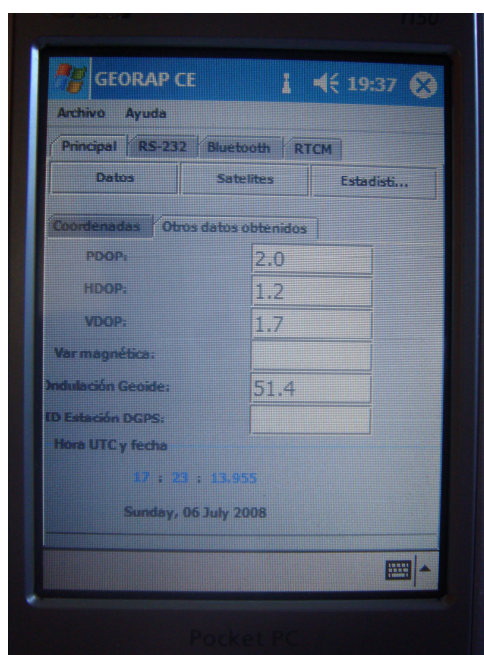


Figura 12. Resto de datos obtenidos.

El botón *Satélites* nos muestra una pantalla donde se recogen los cuatro satélites que estamos usando para localizar nuestra posición, la relación SNR (Señal a ruido) en decibelios, y el total de satélites con los que tenemos visibilidad directa.

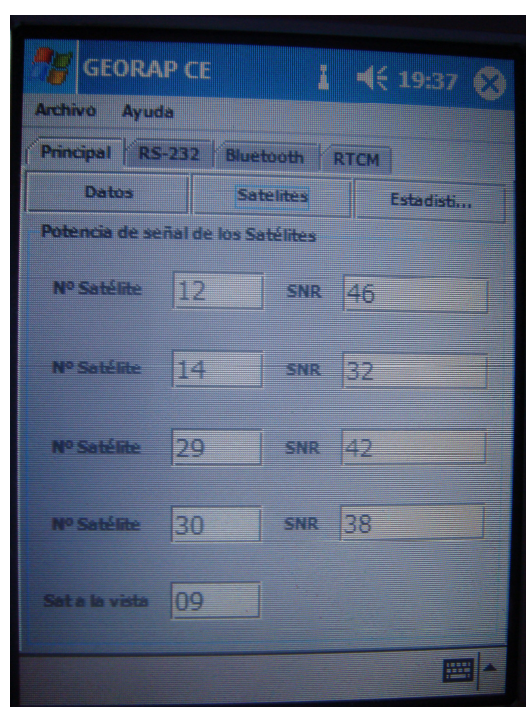


Figura 13. Pantalla con información de los satélites.

Por último, en la pantalla *Estadísticas* tenemos tres pestañas en cada una de las cuales se ha calculado la media y la desviación típica de todas las muestras de coordenadas que nos ha ido enviando el receptor GPS. El receptor GPS manda su posición a la PDA cada cierto tiempo, al desconectarnos de éste el software calcula la media y la desviación típica de todas estas muestras de posición, que han sido enviadas periódicamente por el receptor.

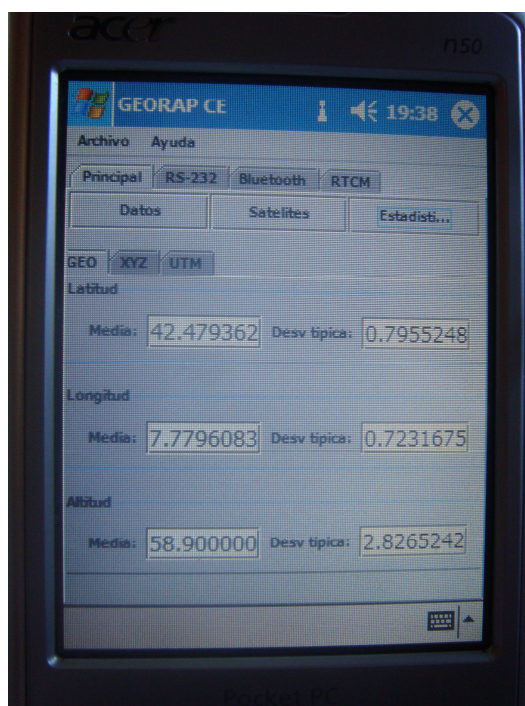


Figura 14. Pantalla donde se muestra la media y la desviación típica de las coordenadas.

4.3.2.10 PAQUETE OPERACIONES

El paquete *Operaciones* está constituido únicamente por la clase *transformar.class*, y su labor consiste en implementar los algoritmos necesarios para transformar las coordenadas geodésicas WGS84 recibidas de nuestro receptor GPS a los otros dos sistemas de coordenadas que hemos utilizado: XYZ (geocéntricas) y UTM.