

Universidad ORT Uruguay
Facultad de Ingeniería

Sistema de posicionamiento satelital con exactitud de centímetros para soluciones en agricultura de precisión

Entregado como requisito para la obtención del título de Ingeniero en
Electrónica

Mauro Güisso – 180591

José Aldabalde – 181874

Ian Herrera – 195330

Tutor: Nicolás Cremona

2019

Declaración de autoría

Nosotros, Mauro Güisso, José Aldabalde y Ian Herrera, declaramos que el trabajo que se presenta en esta obra es de nuestra propia mano. Podemos asegurar que:

- La obra fue producida en su totalidad mientras realizábamos el proyecto final de la carrera Ingeniería en Electrónica.
- Cuando hemos consultado trabajos publicados por otros, lo hemos atribuido con claridad.
- Cuando hemos citado obras de otros autores, hemos indicado las fuentes. Con excepción de dichas citas, la obra es enteramente nuestra.
- En la obra, hemos acusado recibo de las ayudas recibidas.
- Cuando la obra se basa en trabajo realizado conjuntamente con otros, hemos explicado claramente qué fue contribuido por otros, y qué fue contribuido por nosotros
- Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.



.....

Mauro Güisso
26 de febrero de 2019



José Aldabalde
26 de febrero de 2019



.....

Ian Herrera
26 de febrero de 2019

Agradecimientos

Deseamos agradecer a nuestras familias y amigos por el apoyo brindado a lo largo de nuestra carrera.

A la Universidad ORT por brindarnos la posibilidad de desarrollarnos como profesionales.

A la empresa SmartWay por confiar en nosotros la realización del proyecto y el trato recibido por su parte.

Por último, queremos agradecer a nuestro tutor Nicolás Cremona por el apoyo brindado a lo largo del proyecto.

Abstract

En el presente documento se expone la investigación, desarrollo e implementación de un sistema de posicionamiento GNSS adicionado con correcciones RTK, aplicado a la agricultura de precisión.

El proyecto fue presentado por la empresa SmartWay con el objetivo de proveer soluciones en agricultura de precisión. El sistema propuesto debe ser capaz de brindar información de posicionamiento de alta precisión en tiempo real, utilizando NB-IoT como medio de acceso a la red celular, GNSS para la obtención de la información de posición y técnicas de corrección de posición RTK.

Para cumplir con los requerimientos establecidos, se diseñó un sistema independiente compuesto por dos dispositivos. El primero, la estación de referencia, es capaz de proveer la información necesaria para la realización de correcciones RTK. Esta información es usada por el segundo dispositivo (cliente RTK) para proveer al usuario su posición con precisión de centímetros en tiempo real.

Para su implementación fue necesaria la utilización de múltiples lenguajes de programación para el desarrollo de diferentes librerías, que permitieron la integración de diversos componentes de hardware. Adicionalmente se debieron modificar una variedad de archivos de configuración de un sistema Linux para otorgar funcionamiento autónomo del sistema.

Se logró construir un sistema capaz de realizar correcciones RTK satisfaciendo todos los requerimientos del cliente, permitiendo la obtención de correcciones por diversos medios y entidades, comunicándose a través de internet o un radio enlace, utilizando tanto estaciones de referencia públicas como privadas.

Palabras clave

RTK, RTKReady, GNSS, estación de referencia, cliente RTK, posición, precisión, correcciones, tiempo real, NTRIP, *fix*, *float*.

Índice

1	Motivación	8
1.1	Conceptos básicos de agricultura y agricultura de precisión.....	8
1.2	Planteo del problema	10
2	Objetivos del proyecto	12
2.1	Requerimientos del proyecto	12
3	Teoría de tecnologías utilizadas	14
3.1	Introducción sobre posicionamiento.....	14
3.1.1	¿Cómo se calcula la posición de un GNSS?	16
3.1.2	¿Cómo se calculan las distancias a los satélites?	18
3.1.3	Protocolo de datos para receptores GNSS.....	23
3.1.4	Sincronización de relojes	24
3.1.5	Dilution of Precision	25
3.1.6	Fuentes de error GNSS.....	26
3.2	Sistemas de aumentación GNSS.....	29
3.2.1	Satellite-Based Augmentation System (SBAS)	30
3.2.2	Ground-based Augmentation System (GBAS)	31
3.2.3	DGNSS clásico.....	32
3.2.4	Precise Point Positioning.....	32
3.2.5	RTK.....	33
3.2.6	Comparativa de los sistemas de aumentación	35
3.2.7	Transmisión de correcciones	37
3.2.8	Software disponible.....	39
3.3	LPWAN.....	43
4	Solución propuesta	45
4.1	RTKReady	48
4.1.1	Funcionalidades.....	49

4.1.2	Hardware	55
4.1.3	Software	65
5	Pruebas	81
5.1	Pruebas de precisión	81
5.2	Pruebas de alcance del radio enlace	85
5.3	Conexión de estación de referencia a NTRIP Caster	86
5.4	Pruebas de estabilidad	87
5.5	Pruebas de conectividad a LTE Cat M1	89
6	Conclusiones	93
6.1	Forma de trabajo	93
6.2	Precio final.....	95
6.3	Aprendizaje.....	96
7	Mejoras.....	99
8	Glosario	101
9	Referencias bibliográficas	104
A.	Evolución del sistema.....	109
B.	Protocolo de configuraciones del sistema	114
C.	AT Commands	117
D.	Protocolo SPI.....	118
E.	Transformación de formato Latitud y Longitud.....	120
F.	Protocolo NMEA.....	121
G.	Manual de usuario	123
H.	Procedimiento de instalación	127

1 Motivación

En este capítulo se introducen los conceptos de agricultura, agricultura de precisión y las mejoras tecnológicas en el mercado.

1.1 Conceptos básicos de agricultura y agricultura de precisión

La agricultura es un conjunto de actividades destinadas a la transformación del medio ambiente, con el fin de hacerlo más apto para el crecimiento de las siembras. Es decir, el trabajo de adaptar el suelo y el tratamiento del cultivo, generalmente con el propósito de obtener alimentos.

Los orígenes de la agricultura se remontan al periodo Neolítico entre 8.000 a 10.000 años a. C., es en este periodo donde se encontraron los primeros indicios de las transiciones de sociedades de cazadores-recolectores a sociedades agrícolas. Existen varias teorías sobre las causas de estas transiciones, entre ellas se consideran cambios climáticos, la escasez de caza y/o alimentos de recolección. La agricultura permitió un aumento en la densidad de población, generado por el aumento de la disponibilidad de alimentos.

Desde poco después de sus orígenes hasta el día de hoy, la enorme mayoría de las sociedades depende de la agricultura para su supervivencia. Teniendo un papel vital en toda sociedad, la agricultura evolucionó pasando por varias etapas según la sociedad y su entorno en la cual se desarrolló. Esta dependencia se retroalimenta de los beneficios generados por la agricultura, es decir al aumentar la disponibilidad de alimentos aumenta la población, al aumentar la población es necesario aumentar la disponibilidad de alimentos. Esta dependencia genera una constante evolución de la agricultura adaptándose a diferentes periodos de la historia humana.

Es de gran importancia destacar la revolución verde, esta se desarrolló entre 1950 y 1980 en países en vías de desarrollo. Hace referencia al gran aumento en la producción agrícola debido al uso de variedades de alto rendimiento de cosecha (conocido por sus siglas en inglés HYVs) en especial por la incorporación de organismos genéticamente modificados (por sus siglas en inglés GMO). La revolución verde también es consecuencia del incremento del uso de sustancias químicas, tanto fertilizantes para cubrir las necesidades de la planta y pesticidas para luchar contra plagas y enfermedades.

Uno de los problemas presentes en la agricultura que es de interés abordar en este proyecto es su sostenibilidad. Se entiende que, para poder satisfacer la gran demanda de productos agrícolas, el medio ambiente tuvo que pagar un costo muy alto. El uso GMO amenaza la base genética de la agricultura y reduce la biodiversidad. Además, el uso de pesticidas y fertilizantes en exceso amenazan la sostenibilidad de la agricultura. El excedente de producto, es decir lo que la planta no pudo absorber llega a los ríos y de ahí van a parar a los océanos. Estos productos químicos matan peces y generan algas que consumen el oxígeno de las aguas, desequilibrando el ecosistema con consecuencias fatales para la vida.

Según la docente de la Maestría de Ciencias Ambientales de la facultad de Ciencias de la República Oriental del Uruguay, es imposible que Uruguay se mantenga siendo un país agroexportador sin la utilización de fertilizantes o plaguicidas [1]. Como Uruguay hay muchos países que se encuentran en esta situación, y existe una necesidad de satisfacer la demanda de productos agrícolas sin comprometer el medio ambiente por el uso de agroquímicos. Hasta el momento no existen alternativas para remplazar los agroquímicos, generando la necesidad de utilizarlos en las medidas justas para disminuir su impacto en el medio ambiente.

Con el fin de poder cumplir con la demanda y la sostenibilidad, surgió la necesidad de más precisión en los sistemas de agricultura. Esta necesidad crea un nuevo mercado: “la agricultura de precisión”, el cual tiene como misión optimizar la gestión ambiental, económica y tecnológica en el sector agronómico [2].



Ilustración 1 - Concepto de agricultura de precisión [3]

1.2 Planteo del problema

Los servicios de posicionamiento GNSS son una de las tecnologías fundamentales para el desarrollo de la agricultura de precisión. Si bien estos sistemas permiten a los usuarios ubicarse en cualquier parte del mundo, las posiciones obtenidas presentan desviaciones cercanas a los 10m (ver capítulo 3.1.6), surgiendo la necesidad de utilizar técnicas que incrementen la precisión de la información.

Hablando en términos de agricultura de precisión, al aumentar la exactitud de la información de posición se consigue aumentar la cantidad y calidad de automatismos de control de las maquinarias, esto simplifica la labor de los tractoristas y permite extender las jornadas laborales al posibilitar trabajar en condiciones de reducida visibilidad o de noche.

Utilizar la información de posición corregida en las maquinarias de agricultura, se aumenta significativamente la eficiencia en las prácticas de siembra, cosecha y pulverización. Permitiendo, en cualquier etapa, distribuir los insumos de forma localizada, y en consecuencia optimizar las pasadas. Esto minimiza los solapamientos y huecos durante la aplicación de insumos, el uso innecesario de químicos y permite explotar al máximo el potencial de cosecha.

Todos estos factores permiten reducir los costos de producción, disminuir el impacto ambiental y asegurar una producción estable, lo que mejora la rentabilidad y sostenibilidad del negocio agrícola.

Con el tiempo se comienzan a utilizar técnicas para mejorar la exactitud de las posiciones GNSS y acceder a todos los beneficios que esto posee. Entre estas se destaca RTK, la cual se basa en la realización de correcciones a posiciones GNSS, para lograr niveles elevados de precisión.

A grandes rasgos los sistemas RTK están compuestos de una estación de referencia (*Base Station*) y un cliente RTK (*Rover*) del cual se quiere conocer su posición con alta precisión. La estación de referencia debe conocer su posición con un alto nivel de precisión, ser capaz de posicionarse mediante sistemas GNSS convencionales y mediante algún medio de comunicación, debe transmitir toda esta información al cliente RTK. Este último, con algoritmos complejos, procesa la información proveniente de la estación de referencia en

conjunto con su posición obtenida utilizando sistemas GNSS convencionales, calculando su posición con desviaciones del orden de centímetros, siempre que este se encuentre en las cercanías de la estación de referencia (70 Km máximo).

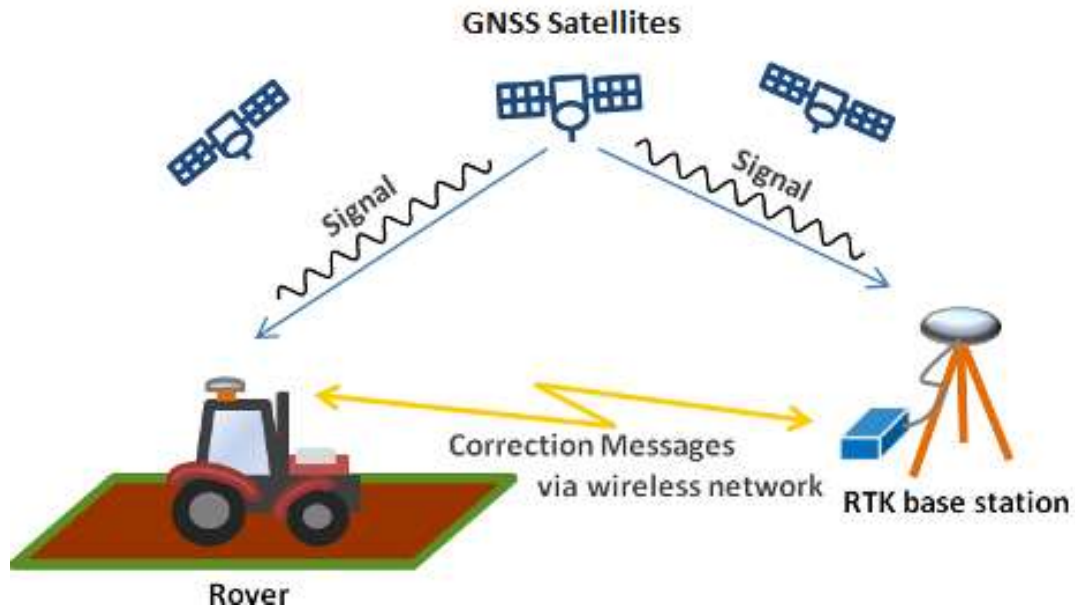


Ilustración 2 - Concepto básico de RTK [4]

La empresa SmartWay con el objetivo de proveer soluciones en agricultura de precisión, plantea el deseo de obtener un sistema que sea capaz de proporcionar en tiempo real, a su hardware actual de telemetría, con información de posicionamiento de alta precisión. Para ello propone construir un sistema de correcciones RTK, utilizando receptores GNSS para la obtención de la información de posición y módulos NB-IoT como medio de acceso a la red celular.

2 Objetivos del proyecto

Este proyecto abarca el estudio e integración de diferentes tecnologías y dispositivos con el fin de implementar un sistema que beneficie al área de la agricultura de precisión.

El objetivo general del proyecto es aumentar la eficiencia de la práctica de la agricultura de precisión al implementar un sistema que sea capaz de proveer a maquinarias agrícolas, su información de posición con precisión de centímetros en tiempo real. Para ello se construirá un sistema capaz de realizar correcciones RTK de posiciones GNSS convencionales, el cual requerirá de la implementación de al menos una estación de referencia y un cliente RTK.

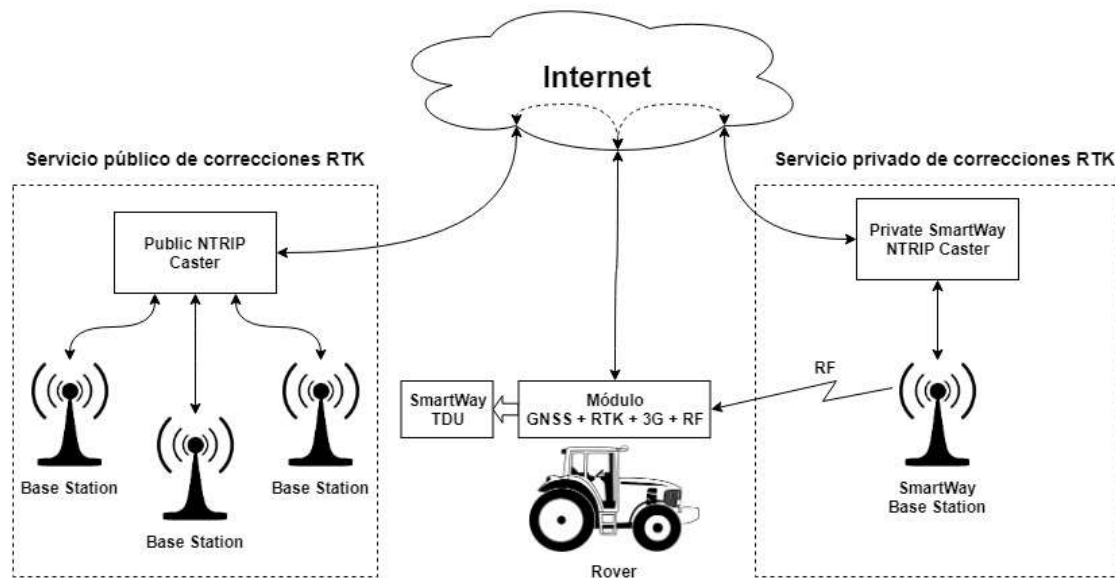


Ilustración 3 - Modelo de sistema planteado

2.1 Requerimientos del proyecto

Para su realización, el proyecto deberá abarcar:

1. Diseño y construcción de un cliente RTK. Este debe ser capaz de, consumir la información brindada por las estaciones de referencia para realizar la corrección de su posición y poder así entregar ésta al usuario. La información proveniente de las estaciones de referencia se podrá adquirir por medio de una conexión a internet o utilizando un radio enlace.

2. Diseño y construcción de una estación de referencia. Esta debe ser capaz de, proveer de la información necesaria para que los clientes RTK puedan corregir su posición. Esta información podrá ser otorgada mediante un servidor de correcciones por medio de una conexión a internet o mediante un radio enlace directo a los clientes RTK.
3. Diseñar e implementar todos los mecanismos de comunicación entre los dispositivos mencionados anteriormente bajo protocolos estándar de sistemas RTK. Esto le permite al usuario, utilizar sistemas y/o herramientas de carácter público en su beneficio.

Bajo todas las condiciones anteriores, el sistema deberá contar con las siguientes características:

1. Simple y amigable para facilitar a usuarios inexpertos que adquieran y utilicen el producto.
2. Adaptable a las necesidades de los usuarios.
3. Robusto, para la independización del usuario que lo utilice.
4. Basado en software y/o librerías de código abierto.

Adicionalmente, se busca que el sistema sea modular, escalable, compatible y de costo reducido.

- Modular: que sea posible reemplazar tecnologías sin mayores inconvenientes de integración con software y hardware existentes.
- Escalable: agregar nuevas funcionalidades o que nuevas versiones del software utilizado pueda ser fácilmente añadido. Esto requiere, que todas las librerías o paquetes de software de código abierto utilizados se modifiquen lo menos posibles.
- Compatible: siempre que sea posible, utilizar medios de comunicación e intercomunicación (interfaces, protocolos de capas de datos, protocolos de capa física, etc.) del dispositivo de carácter estándar.
- Costo reducido: debido a que el usuario, debe incorporar un módulo por cada maquinaria que trabaje de forma paralela, se debe considerar que el costo de implementación no sea elevado.

3 Teoría de tecnologías utilizadas

En este capítulo se detallarán todas las tecnologías utilizadas e investigadas durante la realización del proyecto.

3.1 Introducción sobre posicionamiento

Se entiende por GNSS a un grupo de tecnologías (sistemas satelitales) que permiten determinar las coordenadas geográficas y altitud de un punto dado del globo terrestre. Estas coordenadas se obtienen como resultado de la recepción de señales provenientes de un grupo de satélites que orbitan sobre una zona determinada a la cual brindan cobertura. Se los conoce como constelación de satélites, siendo las más conocidas:

- Sistema de Posicionamiento Global (GPS).
- Sistema Orbital Mundial de Navegación por Satélite (GLONASS).
- Galileo.
- BeiDou.

En la Tabla 1 se realiza una pequeña comparación de estos sistemas, donde se hace importante notar la precisión de estos.

Sistema	GPS	GLONASS	Galileo	BeiDou
Dueño	Estados Unidos	Rusia	Unión Europea	China
Cobertura	Global	Global	Global a partir de 2020	Global a partir de 2020
Satélites	32	24	26	23
Estatus	Operativo	Operativo	Operativo desde 2016 (6 satélites más a lanzarse en 2020)	Operativo desde 2018 (8 satélites más a lanzarse en 2020)
Precisión	15m	4.5m - 7.4m	1m	10m

Tabla 1 - Comparativa de constelaciones GNSS

Sin embargo, indiferentemente de la constelación, la estructura de los sistemas GNSS se divide en tres segmentos:

- 1) Segmento espacial: este segmento está compuesto únicamente por satélites, los cuales envían señales de radio UHF o VHF difundiendo a modo de *broadcast* la información requerida por los demás segmentos. Dichos satélites orbitan alrededor de la tierra enviando la información necesaria para que el segmento de usuario pueda posicionarse. La cantidad de satélites en órbita garantizan una cobertura global en todo momento.
- 2) Segmento de control: este segmento es una red de estaciones ubicadas en tierra, comprendida por estaciones de control (*Master Control Stations*), estaciones de envío de datos (*Data Uploading Stations*) y estaciones base o de monitoreo (*Base Stations*). Las estaciones de monitoreo se encuentran distribuidas geográficamente y se encargan de monitorear y recibir las señales provenientes de los satélites, estas retransmiten la información a las estaciones de control. Con la información recibida, las estaciones de control analizan la información y corrigen los parámetros de la órbita y de los relojes de alta precisión en los satélites, realizando esta comunicación a través de las estaciones de envío de datos.
- 3) Segmento de usuario: este segmento consiste en equipamiento capaz de recibir las señales enviadas por los satélites GNSS, y procesarlas para obtener información de tiempo y ubicación actual del dispositivo. Los dispositivos de este segmento deben de contar con antenas y receptores, los cuales pueden o no estar físicamente integrados.
 - a) Antena activa: se encargan de filtrar y recibir las señales de radio enviadas por los satélites y transmitirlos al receptor.
 - b) Receptor: utilizan las señales captadas por las antenas, para calcular la información de tiempo y posición. Estos pueden estar diseñados para usar las señales de una o varias constelaciones GNSS.



Ilustración 4 - Estructura de sistemas GNSS [5]

3.1.1 ¿Cómo se calcula la posición de un GNSS?

El principio de funcionamiento de un GNSS es un método matemático denominado trilateración, el cual consiste en determinar la posición de un objeto conociendo la distancia a un mínimo de puntos de referencia distintos. Matemáticamente, para obtener una posición en casos bidimensionales, basta con conocer la distancia a tres puntos de referencia, sin embargo, para casos tridimensionales se requiere de un mínimo de cuatro puntos de referencia.

Geoméricamente con la utilización de este método se deduce que:

- La posición deseada está en algún punto del cascarón de una esfera, si se conoce la distancia a un punto de referencia.
- La posición deseada estará en algún punto de la circunferencia resultante de la intersección de dos esferas, si se conoce la distancia a dos puntos de referencia.
- La posición deseada estará en alguno de los dos puntos resultantes de la intersección de una circunferencia (intersección de dos esferas) y una esfera (ver Ilustración 5), si se conoce la distancia a tres puntos de referencia.

- La posición deseada será dada por la intersección de una esfera con uno de los dos puntos generado por la intersección de tres esferas, si se conoce la distancia a cuatro puntos de referencia.

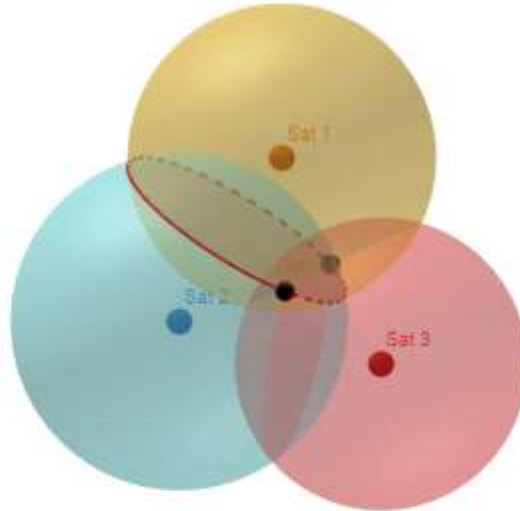


Ilustración 5 - Método de trilateración

Haciendo referencia a la geolocalización, conociendo la distancia a tres satélites se puede determinar la posición, exenta de información de altura, conservando aquel punto que es acorde al globo terrestre. Esto es posible ya que el globo terrestre constituye la cuarta esfera de intersección. En caso de requerirse la información de altura, se debe utilizar un mínimo de cuatro satélites en el cálculo de la posición.

Para plantear un método analítico de trilateración, se expresan las distancias a los puntos de referencia como ecuaciones de esferas. Resolviendo el sistema de ecuaciones planteado se obtiene la posición deseada.

Conociendo las distancias de los puntos de referencia, en el caso de un GNSS las distancias de los satélites, y aplicando el método de trilateración se obtiene las coordenadas de cualquier punto en el globo terrestre.

3.1.2 ¿Cómo se calculan las distancias a los satélites?

Se entiende por observable GNSS, como la medida de la distancia entre el receptor y el satélite. Dicha distancia derivada de medidas de tiempo y de las diferencias de fase. Estas medidas son basadas en comparaciones entre la señal recibida y una réplica de la señal generada en el receptor.

Para el caso particular de GPS, los satélites emiten señales moduladas sobre portadoras en las frecuencias $L1=1575.42$ MHz y $L2=1227.6$ MHz. Si bien otras constelaciones usan frecuencias próximas a estas (ver Ilustración 6), indiferentemente de su constelación, estas son nombradas en base a su cercanía con las portadoras de GPS, generalizando a todas las portadoras en L1 y L2.

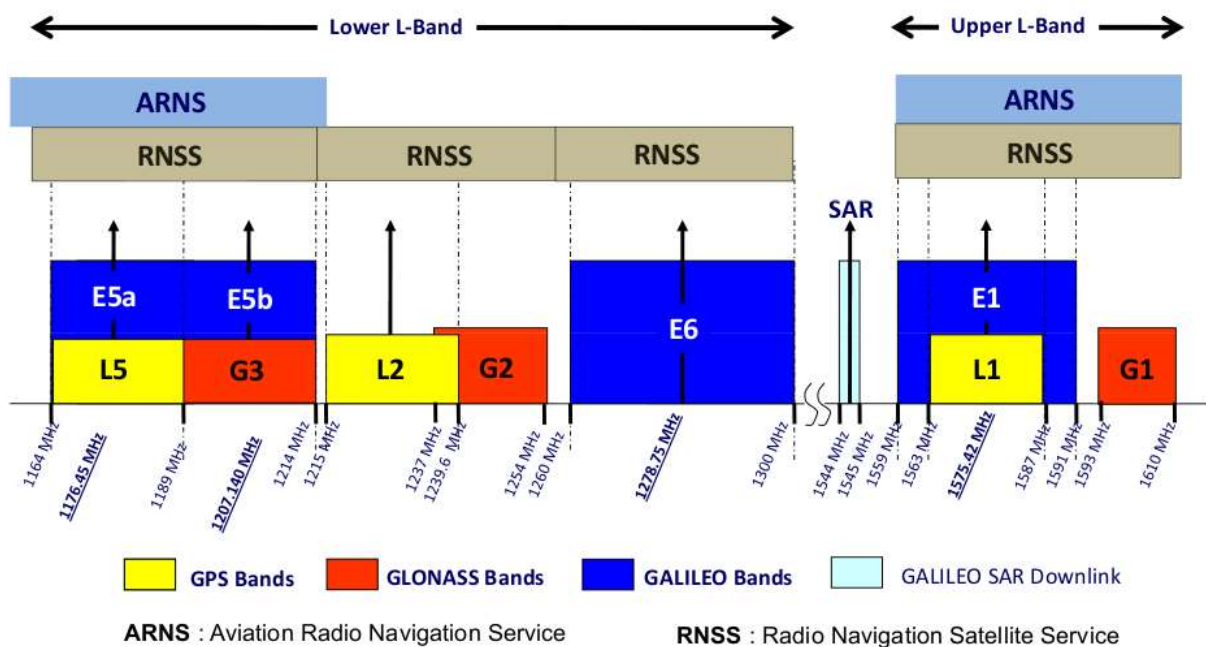


Ilustración 6 - División del espectro de frecuencias de sistemas GNSS [6]

Sobre estas portadoras se modulan los códigos y mensajes a ser enviados.

- Código C/A (*Coarse Acquisition*), también llamado SPS (*Standard Positioning Service*), este código es proporcionado para usos civiles.
- Código P (*Precise*), también llamado PPS (*Precise Positioning Service*), este código es de uso militar, está encriptado y no es utilizable por receptores civiles. Los códigos

SPS y PPS son denominados códigos PRN (*Pseudo Random Noise*), ya que son muy complejos y a simple vista se parecen a ruido.

- Mensajes de navegación, que contiene las efemérides, estas proporcionan la situación orbital del satélite, correcciones de reloj y otros parámetros del sistema.

En la Ilustración 7 se representa la estructura de la señal transmitida por los satélites de la constelación GPS. Cabe destacar que un chip constituye una secuencia de n bits. En este tipo de modulación cada chip abarca únicamente un bit, resultando en que 1.023 Mchips por segundo equivalen a 1.023 Mbps.

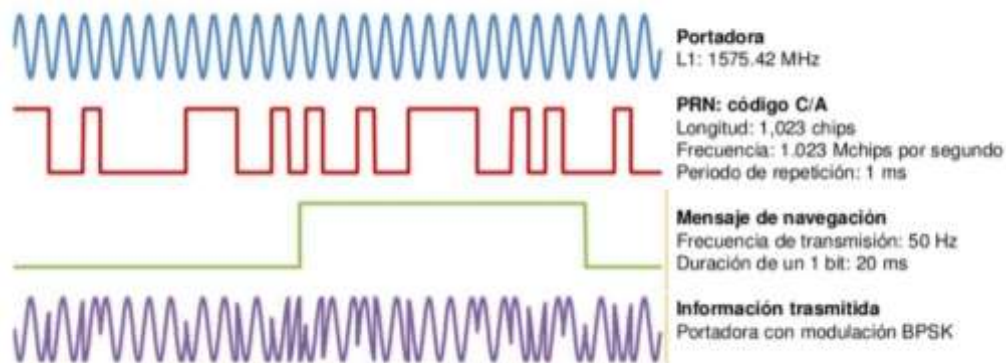


Ilustración 7 - Estructura de las señales GPS [6]

Desde un punto de vista genérico se puede definir al observable básico GNSS como el tiempo que demora la señal emitida por el satélite en ser detectada por el receptor, en otras palabras, el tiempo de propagación de la señal.

Tiempo de Propagación = Tiempo de detección de la señal en el receptor - Tiempo de emisión de la señal en el satélite

Conociendo el tiempo de propagación, se puede calcular la distancia al satélite al multiplicar este tiempo por la velocidad de la luz.

Un factor importante a tener en cuenta es que, para calcular el tiempo de propagación de la señal intervienen dos relojes, el reloj del receptor y el del satélite. Por este motivo es de suma importancia que estos relojes estén sincronizados, para lo cual existen métodos que se detallarán en el capítulo 3.1.4.

Como se mencionó anteriormente, para realizar las medidas de distancia es necesario conocer las observables GNSS, las cuales se puede agrupar en tres grupos:

- Observables de tiempo
 - Código C/A modulado sobre la portadora L1
 - Código P modulado sobre la portadora L1
 - Código P modulado sobre la portadora L2
- Observables de diferencia de fase de la portadora
 - Diferencia de fase de la portadora L1
 - Diferencia de fase de la portadora L2
- Observable Doppler
 - Desplazamiento Doppler de la portadora L1
 - Desplazamiento Doppler de la portadora L2

Para estas tres observables existen métodos que permiten medir las distancias a los satélites desde un receptor. Medida de pseudo-distancia Doppler que utiliza las observables Doppler, medida de pseudo-distancia de código que utiliza las observables de tiempo, y medida de pseudo-distancia de fase que utiliza las observables de fase de la portadora.

3.1.2.1 Pseudo-distancia de código

Para obtener las pseudo-distancias de código, el receptor GNSS genera una réplica exacta del código de la señal enviada por el satélite y compara los códigos mediante una correlación. Es decir, calcula el desfasaje de tiempo necesario para alinear ambos códigos, dicho desfasaje corresponde al tiempo de propagación de la señal.

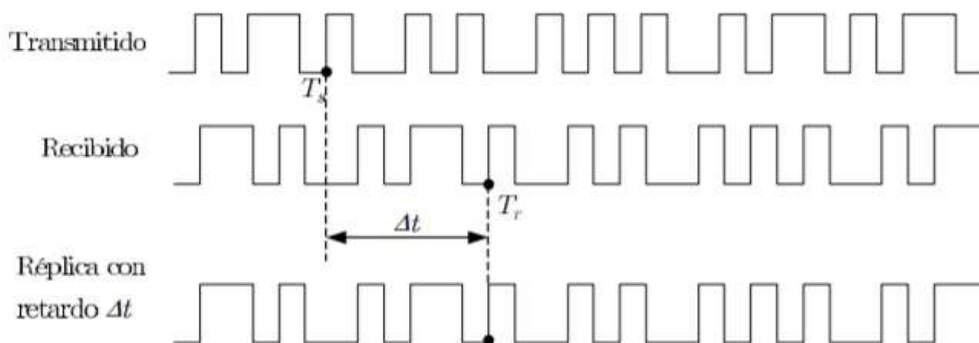


Ilustración 8 - Sincronización entre el código recibido y la replica [7]

Como se mencionó antes si se multiplica el tiempo de propagación por la velocidad de la luz, se obtiene un cálculo simplificado de las pseudo-distancias entre los satélites y el receptor. Para que estos cálculos sean más precisos se deben considerar varios tipos de errores que suceden a lo largo de todo el proceso. En particular se deben tener en cuenta los errores inducidos por el sincronismo entre los relojes del receptor y los satélites, el *multipath* de las señales, efectos relativistas y el retardo inducido en las señales emitidas por los satélites al atravesar la ionosfera y troposfera. En el capítulo 3.1.6 se profundizará sobre los efectos de error en las señales GNSS.

3.1.2.2 Pseudo-distancia de fase

La pseudo-distancia de fase es la distancia entre el satélite y el receptor expresada como un número entero de ciclos n de la portadora más una diferencia de fase. A esta diferencia de fase se le denomina observable de fase, y es determinada por el receptor al comparar la señal de la portadora recibida con una generada internamente por el mismo. Como se muestra en la Ilustración 9 esta observable toma diferentes valores ϕ_t para cada tiempo t de la órbita de un satélite.

Hay que tener en cuenta que en el momento que el receptor comienza a seguir la fase recibida de la portadora, desconoce la cantidad de ciclos enteros n que transcurrieron desde el instante que se emitió la señal hasta que fue recibida. Al número entero de ciclos n se le denomina ambigüedad de ciclos inicial.

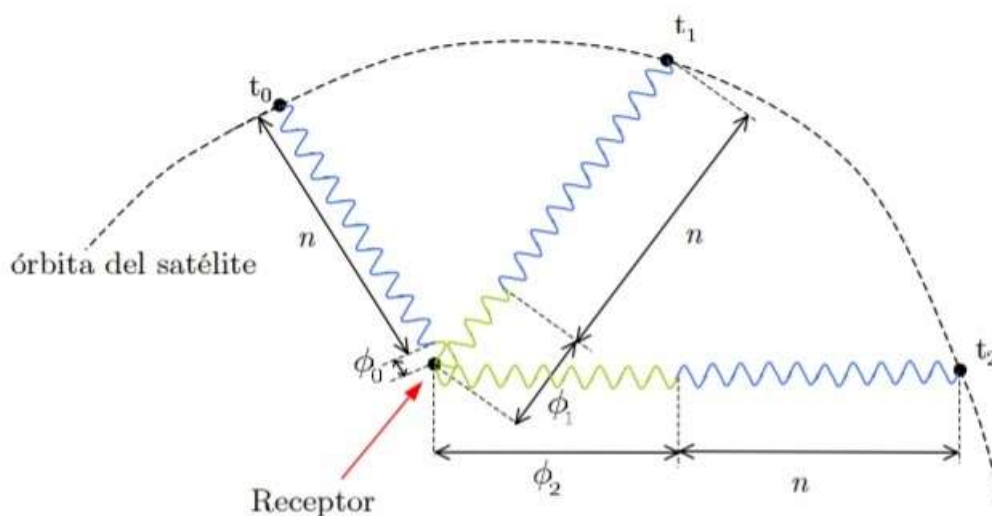


Ilustración 9 - Persistencia de la ambigüedad a lo largo del tiempo [7]

En la figura anterior se puede observar cómo persiste la ambigüedad de ciclo inicial. Para resolver esta ambigüedad existen varios métodos nada triviales que requieren un alto costo computacional. En primera instancia, las ambigüedades iniciales se resuelven parcialmente con números reales, a la posición resultante durante esta primera estimación se la clasifica como *float* y llegan a una precisión del orden de metros. Para esto, utiliza un filtro basado en el algoritmo de Kalman extendido (EKF por sus siglas en inglés), el cual a partir construye una ventana de soluciones, que supone contiene las ambigüedades enteras correctas. Este algoritmo recursivo expresado como una máquina de estados, utiliza las mediciones de entrada actual, el estado calculado previamente y una matriz de incertidumbres para calcular predicciones de los nuevos estados del algoritmo. Con estos estados se obtiene la ventana de soluciones misionada anteriormente.

Una vez realizada esta primera aproximación, hay que resolver la ambigüedad a un número entero de ciclos, a este procedimiento se le denomina *Integer Ambiguity Resolution (IAR)*. Con esta nueva estimación se obtiene la posición con una precisión del orden de pocos centímetros y se la clasifica como *fix*. Si bien existen varios métodos IARs, el que se destaca es el método *Least square AMBiguity Decorrelation Adjustment (LAMBDA)*.

Este método, utilizando la técnica de análisis numérico mínimos cuadrados, reduce la venta de soluciones obteniendo un vector de ambigüedades que se consideran son las iniciales. A este vector se lo califica de forma numérica a partir de una relación con los dos vectores previos de mayor calidad. Si la clasificación supera un umbral se considera una solución *fix*, en caso contrario permanece en *float*. Con estas ambigüedades y las observables de fase se obtienen las pseudo-distancia de fase que permiten resolver el cálculo de la posición.

En las medidas de pseudo-distancia de fase actúan los mismos errores mencionados en las medidas de pseudo-distancia de código, sin embargo, se puede lograr una mayor precisión con las medidas de fase. El motivo principal en el aumento de la precisión al utilizar la medida de pseudo-distancia de fase radica íntegramente en la frecuencia de la señal a ser medida.

Considerando el caso de la transmisión de señales GPS utilizando el código C/A modulado sobre la portadora L1, en donde la portadora tiene una frecuencia de 1575.42 MHz y el código C/A se modula a razón de 1.023 MHz, se tiene que, la longitud onda de la portadora es de

0.19 m y la longitud que abarca la transmisión de 1 bit es de 293.05 m, tal como se observa en la Ilustración 10. En esta ilustración las señales de la portadora y modulación del código C/A se graficaron en diferentes escalas de tiempo para facilitar la visualización de la comparativa.

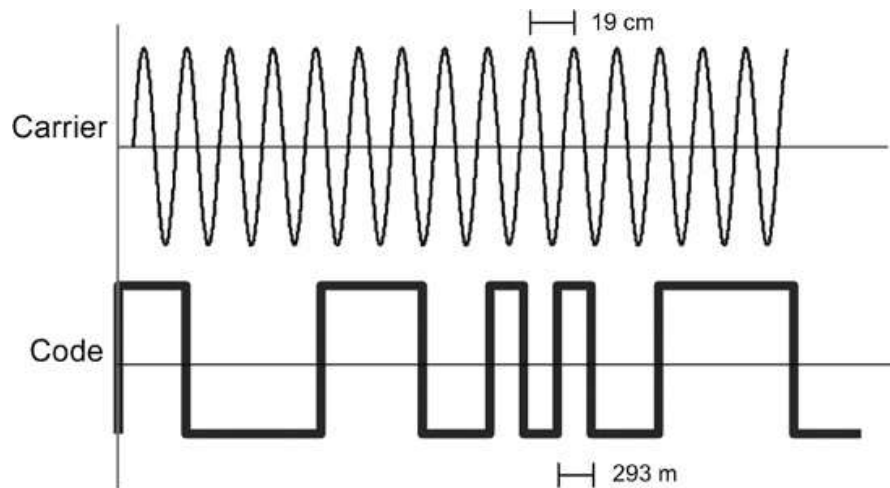


Ilustración 10 - Comparativa de la medición de fase de la portadora y medición de código [8]

Dado que la medición de fase de la portadora intrínsecamente es una referencia de longitud mucho más exacta que la medición realizada con las observables de código, la distancia a los satélites resulta más precisa si se utiliza la pseudo-distancia de fase.

3.1.2.3 Pseudo-distancia Doppler

Los primeros modelos propuestos para el posicionamiento mediante GPS eran basados en medidas de pseudo-distancia Doppler, utilizando como principio la variación de frecuencia detectada debido a la diferencia de velocidades entre el receptor y el emisor, este efecto es denominado efecto Doppler. Este método es poco preciso ya que depende de las variaciones de las distancias entre los satélites y el receptor, siendo estas relativamente pequeñas en relación con la distancia entre estos, no obstante, es un posicionamiento rápido y muy útil al determinar en tiempo real la velocidad de un receptor, siendo esta última su principal funcionalidad hoy en día.

3.1.3 Protocolo de datos para receptores GNSS

Comúnmente, todos los fabricantes de receptores GNSS desarrollan y mantienen su propio formato para transmisión GNSS.

Con el fin de evitar incompatibilidades entre datos y formatos de mensajes entre receptores GNSS/GPS, la Asociación Nacional de Electrónica Marina de los Estados Unidos (NMEA por sus siglas en inglés) desarrolla el estándar NMEA 0183. Este estándar tiene como particularidad, que se limita a la transferencia de información de navegación a información de posición, velocidad, ubicación temporal y datos de los satélites.

3.1.4 Sincronización de relojes

Para explicar el problema de sincronización de los relojes del receptor con los satélites, suponga un espacio bidimensional como muestra la Ilustración 11. En la figura se puede observar dos tipos de circunferencias, las de trazo discontinuo representan las pseudo-distancias con errores de sincronización de reloj y las de trazo continuo donde los relojes del satélite y el receptor están sincronizados.

Considerando el caso donde los relojes no están sincronizados, la intersección de los dos primeros satélites tendría como posible solución los puntos A' y B'. Al intentar intersecar el tercer satélite, se crea un sistema que no tiene solución, ya que no coincide con ninguno de los puntos mencionados.

Para resolver esto el receptor comienza a variar su reloj lo que se traduce en la variación del radio de las tres circunferencias (medida de pseudo-distancia). Una vez las tres circunferencias intersecan en un punto (punto A), el receptor culmina la sincronización de su reloj con los de los satélites, posibilitando la resolución de su posición.

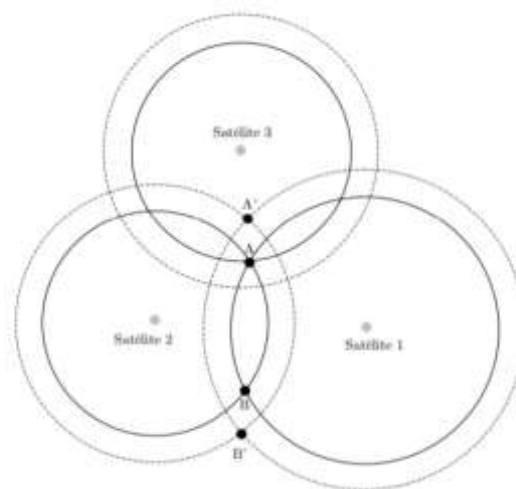


Ilustración 11 - Posicionamiento bidimensional con desviaciones de reloj en receptores [7]

3.1.5 Dilution of Precision

Un factor importante que influye en la precisión de la posición del receptor es la geometría de los satélites, a este parámetro se lo denomina *Dilution of Precision* (DOP). Suponga en un plano bidimensional la intersección de dos circunferencias contemplando un rango de posibles medidas de estas, esto genera una zona donde estaría la posible intersección. En la Ilustración 12 se muestra esta zona para un DOP favorable y en la Ilustración 13 para un DOP desfavorable.

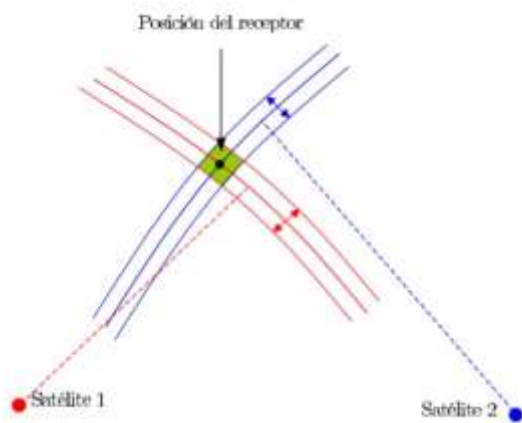


Ilustración 12 - Dispersión de la precisión con geometría favorable [7]

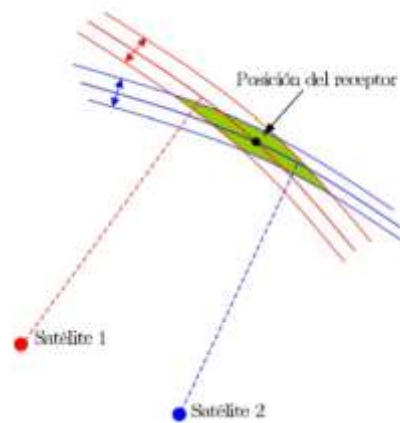


Ilustración 13 - Dispersión de la precisión con geometría desfavorable [7]

Se observa que para un DOP desfavorable el área de la posible intersección es mayor que para un DOP favorable, lo que se traduce a un error mayor en la posición del receptor. Los receptores calculan diferentes valores DOP para diferentes geometrías y a partir de dichos cálculos se eligen los satélites a utilizar.

Para la elección de los satélites a utilizar se pueden calcular diferentes tipos de DOP:

- *Vertical DOP* - Proporciona la degradación de la exactitud en la dirección vertical.
- *Horizontal DOP* - Proporciona la degradación de la exactitud en la dirección horizontal.
- *Position DOP* - Proporciona la degradación de la exactitud en posición 3D.
- *Time DOP* - Proporciona la degradación de la exactitud en los tiempos.
- *Geometric DOP* - Proporciona la degradación de la exactitud en posición 3D y en los tiempos.

3.1.6 Fuentes de error GNSS

Debido a la gran variedad de factores que intervienen en los sistemas GNSS, se tienen múltiples fuentes de error en el posicionamiento. Estos errores hacen imposible el computo de la posición exacta de los dispositivos. A continuación, se detallarán las principales fuentes de error GNSS donde sus magnitudes se representan en la Tabla 2.

Fuente	Rango de error
Reloj de satélites	± 2.0 m
Errores orbitales	± 2.5 m
Demoras ionosféricas	± 5.0 m
Demoras troposféricas	± 0.5 m
Ruido en el receptor	± 0.3 m
Multipath	± 1.0 m

Tabla 2 - Fuentes de error GNSS

3.1.6.1 Reloj de los satélites

Los relojes en los satélites GNSS son muy precisos, pero pueden experimentar un pequeño desvío que resulta en un error significativo en la posición calculada por el receptor. Debido a esto, el segmento de control monitorea y corrige el desfase del reloj.

Para obtener una posición aún más exacta, el receptor debe compensar el error del reloj. Para su realización, el receptor puede descargar información precisa del reloj del satélite desde un sistema SBAS o de un proveedor de servicio PPP, o utilizar un mecanismo de corrección como DGNSS o RTK.

3.1.6.2 Errores orbitales

Los satélites GNSS viajan en órbitas muy precisas, sin embargo, al igual que sus relojes, las órbitas pueden variar en una pequeña cantidad, que resulta en un error significativo en la posición calculada por el receptor.

El segmento de control GNSS en tierra, monitorea de forma continua la órbita de los satélites, enviando las correcciones de sus efemérides siempre que se detecte un desvío.

Al igual que los errores de reloj, para obtener una posición aún más exacta, el receptor debe compensar los errores orbitales utilizando los mismos mecanismos anteriormente utilizados.

3.1.6.3 Demoras ionosféricas

La ionosfera es la capa de la atmósfera que se encuentra en el rango de 80 - 600 kilómetros por encima de la tierra. Esta capa contiene partículas cargadas eléctricamente llamados iones, los cuales retrasan las señales del satélite causando un error significativo en la posición del satélite. Por lo general, estos errores son de aproximadamente 5 metros, pero pueden ser mayores durante períodos de alta actividad ionosférica.

Las demoras ionosféricas varían dependiendo de la actividad solar, la época del año, la estación, la hora del día y la ubicación, haciendo muy difícil predecir cuánto afecta al cálculo de la posición. Adicionalmente, estas demoras varían dependiendo en la frecuencia de radio utilizada para transmitir las señales.

Los receptores que pueden recibir más de una frecuencia GNSS (ej. L1 y L2), pueden utilizar esto a su favor, debido a que pueden determinar la demora ionosférica mediante la comparación de las distintas señales, y eliminarla del cálculo de posición.

Los que se limitan a una sola frecuencia, utilizan modelos de la ionosfera para reducir sus errores. Sin embargo, debido a la naturaleza variable de las demoras ionosféricas, estos modelos no son tan efectivos como usar receptores de múltiples frecuencias.

Las condiciones ionosféricas son muy similares dentro de un área local, permitiendo compensar el retraso utilizando mecanismos de corrección como DGNSS o RTK.

3.1.6.4 Demoras troposféricas

La troposfera es la capa de la atmósfera más cercana a la superficie de la tierra. Las demoras introducidas por esta capa varían a causa de un cambio de humedad, temperatura y presión atmosférica en la troposfera. Dado que estas variaciones son muy similares dentro de un área local, es posible compensar el retraso utilizando mecanismos de corrección como DGNSS o RTK.

3.1.6.5 Ruido en el receptor

El ruido en el receptor hace referencia al error de posición causado por el Hardware y Software de este. Receptores de mayor calidad tienden a tener menos ruido que los receptores de menor costo.

3.1.6.6 Multipath

Las trayectorias múltiples ocurren cuando una señal GNSS es reflejada en un objeto, como la pared de un edificio. Debido a esto, las señales reflejadas se propagan una distancia mayor, llegando al receptor ligeramente retrasado. Esta señal demorada puede causar que el receptor calcule una posición incorrecta.

La forma más sencilla de reducir este tipo de errores es posar la antena GNSS en una ubicación alejada a superficies reflectivas. Cuando esto no es posible, las demoras son manejadas por el receptor GNSS y su antena.

Por lo general, demoras largas se manejan por el receptor GNSS, mientras que demoras ligeras se manejan dentro de la antena receptora.

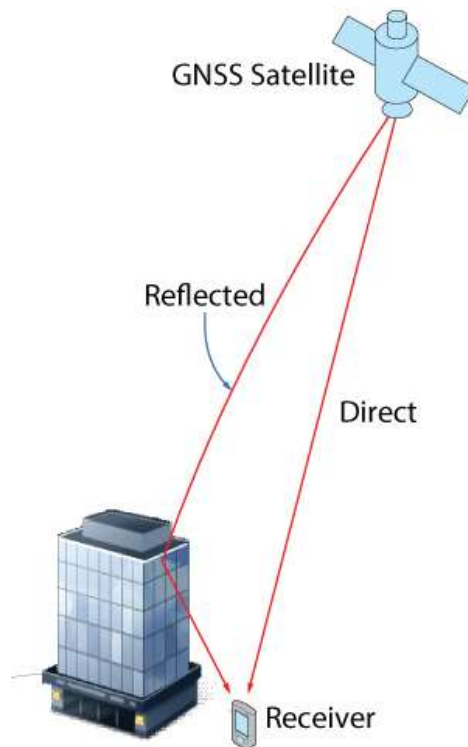


Ilustración 14 - Multipath en señales GNSS [5]

3.2 Sistemas de aumentación GNSS

Debido a que las constelaciones GNSS no fueron elaboradas para satisfacer requisitos estrictos sobre posicionamiento, se han desarrollado varios sistemas que mejoran, "aumentan", el rendimiento de los sistemas de navegación haciendo uso de información externa a los sistemas GNSS. Estos sistemas denominados Sistemas de aumentación GNSS permite satisfacer requisitos estrictos de:

- Precisión: Diferencia entre la posición estimada y la real.
- Integridad: Confianza sobre la información total proporcionada.
- Continuidad: Funcionamiento sin interrupciones no programados.
- Disponibilidad: Es la parte del tiempo en la cual el sistema presenta simultaneidad entre las características requeridos.

Por ejemplo, los errores en sistemas GNSS mencionados anteriormente pueden introducir varios metros de error en la posición de un avión y deben corregirse en tiempo real debido a la precisión necesaria en lugares de poca o ninguna visibilidad. Si bien la motivación inicial para el desarrollo de estos sistemas fue la seguridad en la aviación civil, hoy en día se utilizan en diversas prácticas que abarcan desde aplicaciones recreativas hasta industriales, como es el caso de la agricultura de precisión.

Estos sistemas de aumentación se pueden agrupar según distintos criterios:

- Según el sistema de referencia:
 - Absoluto: La posición se calcula utilizando las medidas de pseudo-distancia de código de un único receptor.
 - Relativo o Diferencial: La posición se calcula utilizando las medidas de pseudo-distancia de código y/o fase de al menos dos receptores.
- Según el movimiento del receptor:
 - Estático.
 - Cinemático.
- Según el observable utilizado:
 - Pseudo-distancia de código.
 - Pseudo-distancia de fase.

- Según el tiempo de la obtención de las coordenadas:
 - En tiempo real.
 - Postproceso: las coordenadas se obtienen una vez finalizada la observación.

A continuación, se dará una breve descripción de los métodos de aumentación GNSS más utilizados en la actualidad.

3.2.1 Satellite-Based Augmentation System (SBAS)

Este sistema se basa en una red de estaciones terrestres ubicadas en posiciones conocidas sobre el área al que brindan cobertura los satélites geostacionarios (situados sobre un punto fijo sobre el globo terrestre). Estas estaciones terrestres monitorean las señales de las constelaciones GNSS y las retransmiten a las instalaciones de procesamiento. Estas últimas realizan y envían las correcciones a los satélites, donde son retransmitidas en forma de broadcast hacia los usuarios finales.

Dado que la posición de las estaciones terrestres es conocida, la instalación de procesamiento puede calcular con precisión las correcciones de un área amplia.

Este sistema es más apropiado cuando los usuarios a los que se quiere dar servicio se extienden sobre un área demasiado grande. Esto se logra con el uso de satélites adicionales para la transmisión de correcciones.

Su virtud principal consiste en que provee los medios para mejorar la precisión, integridad y disponibilidad de señales GNSS primarias.

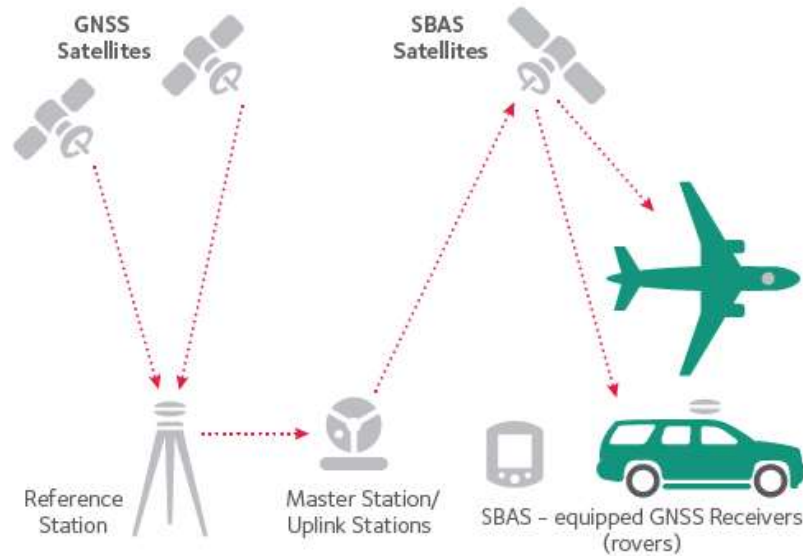


Ilustración 15 - Estructura de sistemas SBAS [9]

Los sistemas SBAS que están en funcionamiento en la actualidad o que están siendo desarrollados son:

- *Wide Area Augmentation System (WAAS)*, gestionado por el Departamento de Defensa de los Estados Unidos.
- *European Geostationary Navigation Overlay Service (EGNOS)*, administrado por la Agencia Espacial Europea.
- *Wide Area GPS Enhancement (WAGE)*, que transmite más precisión en los datos de efemérides y reloj de los satélites destinados a uso militar.
- *Multi-Functional Satellite Augmentation System (MSAS)*, operado por Japón.
- StarFire, Gestionado por la empresa John Deere.
- *Quasi-Zenith Satellite System (QZSS)*, propuesto por Japón.
- *GPS and GEO Augmented Navigation (GAGAN)*, planeado por la India.

3.2.2 Ground-based Augmentation System (GBAS)

Al igual que SBAS, se componen de una red de estaciones terrestres, de las que se conoce su posición de forma precisa. Estas reciben los datos de las constelaciones primarias y las retransmiten a las instalaciones de procesamiento, donde conociendo la posición de las estaciones, se transmite en broadcast las correcciones diferenciales e información relacionada con la integridad.

A diferencia de los sistemas SBAS en la que se transmiten a los satélites geostacionarios, estos transmiten la información directamente a los usuarios finales. Esto se debe a que estos sistemas no fueron diseñados para dar servicio sobre amplias regiones geográficas.

Si bien el objetivo principal de GBAS es proporcionar una garantía de integridad, también aumenta la precisión con errores de posición por debajo de 1m.

3.2.3 DGNSS clásico

El método diferencial clásico consta de un receptor GNSS móvil del que se quiere conocer su posición y de una estación de referencia de la cual se conoce su posición con precisión.

En este método, también conocido como DGPS cuando se utiliza únicamente la constelación GPS, la posición de la estación de referencia se determina con un alto grado de precisión utilizando técnicas convencionales de agrimensura. Luego, la Estación Base determina las distancias a los satélites GNSS a la vista, utilizando técnicas de posicionamiento basada en observables de código y la ubicación de los satélites conociendo las orbitas a partir de las efemérides y tiempo de los satélites.

Una vez conocidos los rangos, la estación de referencia calcula un factor de corrección de estos para cada satélite observado, basado la diferencia obtenida entre su posición conocida y la calculada por la recepción GNSS. Esta información es transmitida a los Rover para que estos la incorporen en los cálculos de su posición.

Esta técnica tiene precisión del orden de 1 metro para usuarios en el rango de algunas decenas de kilómetros desde la Estación Base, creciendo a un ritmo de 1 metro por cada 150 kilómetros de separación.

3.2.4 Precise Point Positioning

Precise Point Positioning (PPP) es una técnica de posicionamiento que proporciona un alto nivel de precisión de un receptor GNSS. Esta solución depende de información precisa acerca de la señal de reloj y órbitas de los satélites GNSS. Para ello, una red de estaciones de referencia situadas en tierra monitorea las constelaciones GNSS y genera correcciones a partir de las observables de fase a los satélites. Estas correcciones son enviadas a los receptores

finales a través de satélites o internet, lográndose así un nivel de precisión de hasta 3 centímetros sin restricción del área de cobertura.

Existen varios proveedores de servicios PPP, los cuales operan la red de estaciones situadas en tierra. Algunos de ellos son VERIPOS, TerraStar, OmniSTAR y StarFire.

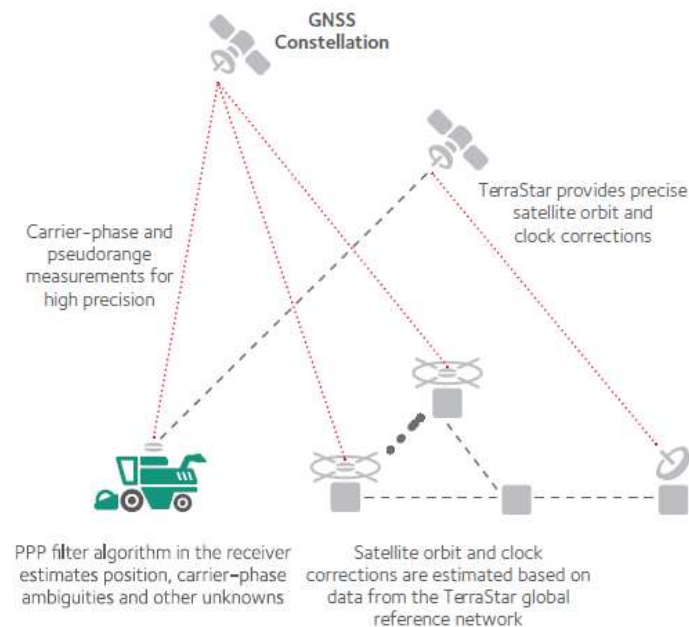


Ilustración 16 - Estructura de sistemas PPP [9]

3.2.5 RTK

Al igual que DGNSS, este es un método diferencial de corrección GNSS, el cual requiere para su funcionamiento de un mínimo dos receptores. Uno funcionando como estación de referencia y uno como receptor móvil o Rover como suele ser nombrado.

Esta técnica provee un alto rendimiento de posicionamiento en las cercanías de las estaciones de referencia, basándose en la transmisión de medidas de pseudo-distancia de fase de la portadora.

Para determinar la ubicación del Rover con alta precisión, el sistema se basa en diferentes principios que se describen a continuación.

En las cercanías de una ubicación, los principales errores en el procesamiento de la señal GNSS son constantes. Por este motivo, tanto el Rover como la estación de referencia

presentan errores similares en el cálculo de su posición. Al conocer la posición real de la estación de referencia, se genera un diferencial que permite al Rover eliminar los errores constantes durante el procesamiento de las correcciones.

Los errores que se incluyen en este principio son el reloj de los satélites, los errores orbitales, el retardo ionosférico y el retardo troposférico. Sin embargo, el único error que es verdaderamente constante a medida que se distancian ambos receptores, es el del reloj del satélite, siendo el error troposférico el primero en descorrelacionarse a cabo de unos pocos kilómetros de separación. Este es el motivo por el cual se degrada el nivel de precisión al aumentar la distancia entre el Rover y la estación de referencia.

Si bien el ruido en las mediciones de fase de la portadora es mucho menor que el de las mediciones de pseudo-código, el procesamiento de la fase de la portadora es ambiguo porque se desconoce el número total de ciclos entre el satélite y el Rover. Debido a esto es necesario un proceso complicado llamado "resolución de ambigüedad", que permite determinar el rango hasta los satélites GNSS.

Para ello, la estación de referencia debe enviar al Rover su ubicación conocida junto con las observables de código y observables de fase de la portadora que ésta recibe de los satélites. Con esta información, el Rover puede arreglar las ambigüedades de la fase y determinar su ubicación con respecto a la estación de referencia.

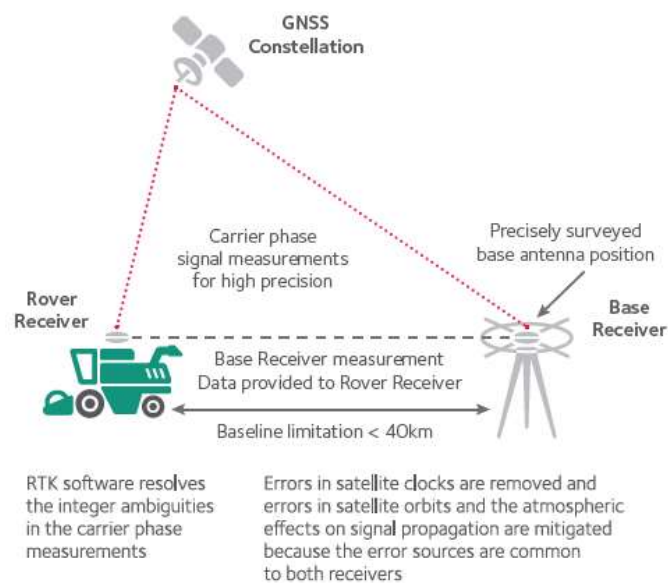


Ilustración 17 - Estructura de sistemas RTK [9]

Utilizando esta técnica se pueden alcanzar precisiones de unos pocos centímetros para distancias de hasta 10 o 20 kilómetros de las estaciones de referencia. Sus limitaciones más importantes son, la distancia entre los receptores, tiempo de convergencia de la solución cercano a 10 minutos y que la detección de las señales GNSS debe ser continuo a fin de evitar el reinicio de la resolución de ambigüedad.

Recientemente, han surgido métodos como la Red RTK y el *Wide Area RTK* (WARTK) que permiten mejorar estas limitaciones.

La Red RTK, se basa en el suministro de información (posición, observables) de una red de estaciones base de tal forma que las mediciones de fase se proveen con ambigüedades constantes, esto tiene como ventaja que el Rover puede cambiar de una Estación Base a otra sin la necesidad de reiniciar los filtros de corrección de ambigüedad.

El método de WARTK incrementa el área de servicio de RTK o de la Red RTK, a partir, de estaciones de referencia separadas una distancia entre los 500 y 900 kilómetros.

La diferencia principal a RTK, es que no se supone que el retardo ionosférico sea el mismo en ambos receptores, principio en el que se basa RTK, sino que se transmite a los usuarios a partir de mensajes adicionales.

Actualmente este sistema no se encuentra operativo, aunque se han llevado a cabo proyectos que demuestran su viabilidad.

3.2.6 Comparativa de los sistemas de aumentación

Durante la investigación de los sistemas de aumento se hizo notorio que estos están fuertemente diferenciados de acuerdo con el equipamiento requerido, el método de corrección utilizado y el costo de la solución.

Cabe aclarar que el sistema GBAS, siendo que fue desarrollado en específico para el campo de la aviación, es un sistema con rápido tiempo de convergencia y precisión cercana a 1m, pero proporciona correcciones para una reducida área de cobertura a costos elevados de implementación. Por este motivo se descartó como sistema de aumentación útil para la implementación del sistema.

3.2.6.1 Equipamiento requerido

En cuanto al equipamiento necesario, los sistemas DGNS y RTK son muy similares ya que ambos requieren una estación de referencia situada en una ubicación conocida, un Rover que recibe correcciones de estación de referencia y un canal de comunicación entre los receptores. Sin embargo, la implementación de RTK requiere receptores GNSS más avanzados que proporcionen información sobre los observables de fase de la portadora y no solo de los observables de código. En ambos casos existe una limitación de cobertura, la cual restringe a los Rover a mantenerse en las cercanías de las estaciones de referencia, siendo DGNS el método más permisivo alcanzando distancias de hasta 1000 Km.

En contraste, los sistemas SBAS y PPP requieren únicamente de receptor GNSS, teniendo la posibilidad de adicionar una conexión a internet para la descarga de correcciones en caso de que se requiera. En el caso de SBAS la cobertura está restringida por el área funcionamiento de satélite geostacionario, en contraste con PPP el cual no tiene limitaciones en este sentido.

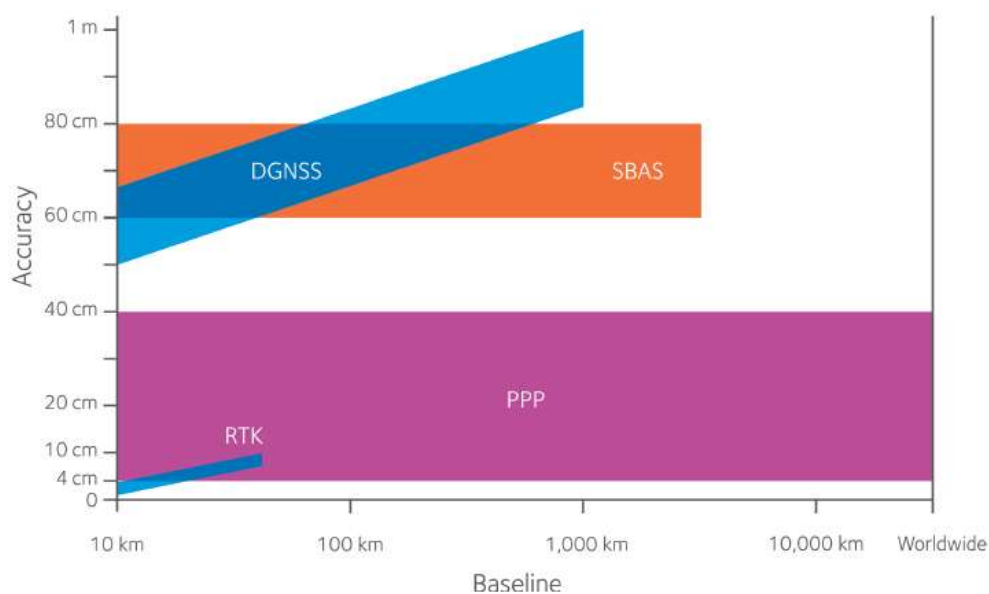


Ilustración 18 - Precisión y rangos de cobertura de sistemas de aumentación [9]

3.2.6.2 Métodos de corrección

En este aspecto, los sistemas DGNS y SBAS utilizan métodos basados en los observables de código, obteniendo así un grado de precisión menor que los sistemas RTK y PPP, los cuales utilizan los observables de fase. Como consecuencia, estos últimos poseen un tiempo de convergencia mayor, siendo PPP el de mayor tiempo.

3.2.6.3 Costos de servicio

En cuanto a los costos, el sistema PPP requiere de una suscripción de pago para acceder a la señal de correcciones, mientras que los sistemas SBAS, DGNSS y RTK utilizan servicios gratuitos, o en el caso de los dos últimos, existe la posibilidad de integrar estaciones de referencia propia, por lo tanto, el costo de su implementación dependerá de los requerimientos de cada sistema y de cada usuario.

3.2.6.4 Elección de sistema de aumentación

Una vez culminada la investigación acerca de los distintos tipos de sistemas de aumento, se determinó que la elección inicial por parte de la empresa SmartWay en la que se pretende utilizar RTK como mecanismo de corrección es el más adecuado. Por lo tanto, de aquí en adelante, la investigación planteada se hará de acuerdo con los requerimientos necesario para la implementación de este método.

3.2.7 Transmisión de correcciones

Para poder utilizar los sistemas de aumentación es necesario que el Rover obtenga por algún medio información extra de la recibida por las constelaciones GNSS. Dado que esta información puede provenir de estaciones de referencia cercanas o de sistemas privados con estaciones distribuidas local o globalmente, se han desarrollado métodos y protocolos para la transmisión y recepción de información de correcciones.

La implementación de nuevos protocolos y formatos fue propulsada por las fuertes demandas de aplicaciones que realicen correcciones en tiempo real y que el protocolo NMEA 0183 no es capaz de transmitir la información requerida para correcciones, por ejemplo, la transmisión de datos de observables para correcciones diferenciales.

Debido a esto múltiples empresas como John Deere, Trimble o Novatel utilizan o han desarrollado protocolos propietarios como NTC, CMRX y RTCA respectivamente para posibilitar la trasmisión de correcciones.

Dada la necesidad de un formato industrial estándar para este tipo de mensajes, el comité 104 de la organización RTCM, genera el formato RTCM en sus distintas versiones, que permite la transmisión de correcciones diferenciales a los usuarios de manera estándar.

Para que la transmisión de estos mensajes sea posible, es necesario contar con un enlace que transmita en tiempo real los datos obtenidos por el receptor de la estación de referencia al receptor móvil (Rover).

Desde hace muchos años los métodos que se utilizaban para la transmisión de correcciones los radio enlaces en las bandas UHF o VHF, pero con el desarrollo de las tecnologías de comunicación inalámbrica, internet es actualmente el medio alternativo preferido de transmisión de correcciones GNSS.

3.2.7.1 Internet como medio de transmisión de correcciones

A finales del año 2004, se desarrolla por la BKG (Agencia Federal Alemana de Cartografía y Geodesia) el protocolo NTRIP. Este protocolo fue aceptado por el comité RTCM como el estándar para transmisión de correcciones en tiempo.

NTRIP es un protocolo para la transferencia de correcciones entre estación de referencia y Rover utilizando Internet. El mismo se basa en la transferencia de mensajes RTCM bajo el protocolo *HyperText Transfer Protocol* (HTTP) en su versión 1.1 y su principal utilidad se encuentra en zonas donde links de radios tradicionales no funcionan de buena forma debido a árboles, colinas, edificios.

Al posibilitar las correcciones mediante la conexión a internet, se proporciona un método rápido y fiable para la transmisión en cualquier zona cubierta por una red de telefonía móvil.

Si bien este método tiene varios beneficios en cuanto a la adquisición de correcciones, para su implementación, se debe respetar la arquitectura estándar de los servicios que las proveen. Esta arquitectura se compone de tres partes:

NTRIP Server: Es el terminal conectado al receptor GNSS en la estación de referencia. Su función es la de establecer una conexión a un NTRIP Caster posibilitando el flujo de datos RTCM desde la estación de referencia al Caster. En la actualidad, los dispositivos GNSS que se diseñan especialmente para funcionar como estación de referencia, incluyen en su Firmware la funcionalidad de NTRIP Server.

NTRIP Clients: Es la terminal en el Rover que se encarga de realizar las peticiones de datos RTCM al NTRIP Caster y entregarla al Receptor GNSS.

NTRIP Caster: Es lo que se conoce como Servidor en una arquitectura Cliente - Servidor. El mismo se encarga, por una parte, de gestionar los flujos enviados por los NTRIP Server, y por otra, responder a las peticiones de los NTRIP Clients.

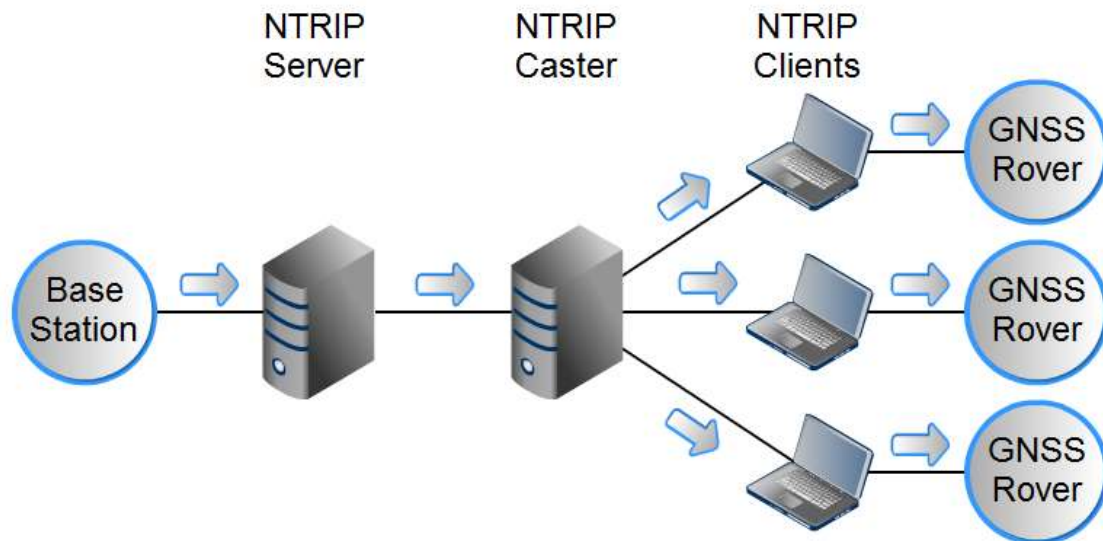


Ilustración 19 - Arquitectura estándar de los servicios de correcciones en tiempo real utilizando internet [10]

3.2.8 Software disponible

En esta etapa se realizará un breve resumen de los softwares más relevantes encontrados para la implementación de un sistema de correcciones.

MyWay RTKs

Es un sistema diseñado especialmente para agricultura, permitiendo a los usuarios recibir correcciones RTK de estaciones de referencia privadas a través de Internet. Como desventaja principal, se trata de un Software pago el cual no tiene estaciones de referencia que brinden correcciones para Uruguay.

EZSurv

Es un Software capaz de realizar correcciones pos-proceso.

Eos GNSS Tools

Es un Software del tipo NTRIP Client, que permite recibir correcciones GNSS de un NTRIP Caster o de un servidor TCP/IP de Internet.

GNSS Analysis and Positioning Software (GAPS)

Software desarrollado por la Universidad de New Brunswick con el fin de proveer a los usuarios de una herramienta correcciones PPP, gratuita y en línea, capaz de estimar posiciones y otros parámetros de interés.

BKG NTRIP Client (BNC)

Es un programa de código abierto del tipo NTRIP Client diseñado principalmente para recibir flujos de datos desde cualquier NTRIP Caster. Una vez adquirida la información permite transferirla por un puerto serie o IP, alimentando una aplicación de correcciones. Adicionalmente puede calcular en tiempo real una solución de PPP a partir de flujos RTCM o archivos RINEX.

BKG Professional NTRIP Caster

Es un *software* pago del tipo NTRIP Caster que permite la transmisión de flujos RTCM en tiempo real a través de Internet. Se ejecuta en sistemas Linux y ha sido probado con éxito en varias distribuciones este.

Tiene la capacidad de admitir más de 100 NTRIP Servers y más de 2000 NTRIP Clients simultáneamente.

Lefebure NTRIP Client

Es un software del tipo NTRIP Client que permite la obtención de mensajes RTCM desde un Caster a través de Internet, para luego enviarlas de forma serial a una aplicación de correcciones. A su vez, permite la descarga de la tabla de *mountpoints* disponibles. Si bien es de código abierto, este *software* fue diseñado solo para sistemas Windows.

SNIP

Es una aplicación del tipo NTRIP Caster que facilita la transferencia de correcciones a través de Internet. El mismo permite transmitir correcciones GNSS a múltiples usuarios de forma simultánea, realizar revisiones en tiempo real de estadísticas detalladas. Adicionalmente, posee un servicio público RTK2go, en el cual se pueden publicar y adquirir información de correcciones de forma gratuita.

RTKLib

Es un paquete de *software* de código abierto con funcionalidades de posicionamiento estándar y de alta precisión utilizando GNSS. Posee aplicaciones funcionales para Windows y algunas distribuciones de Linux del tipo NTRIP Client, NTRIP Server y NTRIP Caster para un único *mountpoint*.

Este software provee de librerías y aplicaciones que implementan múltiples métodos de posicionamiento entre los que se encuentran el posicionamiento estándar, DGNSS, RTK y PPP, utilizando las constelaciones GPS, GLONASS, BeiDou, entre otras.

Adicionalmente, es capaz de: procesar varios formatos y protocolos estándar de GNSS (NMEA, RINEX, BINEX, RTCM 3.1 y 2.3, etc.), procesar mensajes propietarios de receptores GNSS (Fabricantes: NovAtel, Hemisphere, Ublox, SkyTraq, JAVAD, Furuno y NVS) y proveer de comunicación exterior por medio de interfaz serial, TCP/IP, NTRIP, FTP/HTTP y almacenamiento local de archivos.

Servicio de correcciones en Uruguay

En 1995 el Servicio Geográfico Militar (SGM), se integra al proyecto de Sistema de Referencia Geocéntrico para las Américas (SIRGAS) con el objetivo de establecer y mantener una red nacional de estaciones de referencia, actualmente este servicio lleva el nombre de Red Geodésica Nacional Activa de la República Oriental del Uruguay (REGNA-ROU).

A partir del 1° de enero del 2019 el Servicio Geográfico Militar (SGM) pasa a llamarse Instituto Geográfico Militar (IGM).

Actualmente, este servicio cuenta con 27 estaciones de referencia activas (puntos rojos en la Ilustración 20), 23 situadas dentro del territorio nacional, tres dentro del territorio argentino

cercano a la frontera y una en la Estación Científica Antártica Artigas. Cada una de estas estaciones se conecta por Internet a un servidor central (NTRIP Caster) situado en la IGM, permitiendo que las observables se encuentran disponibles en la Web para su acceso de forma libre y gratuita.

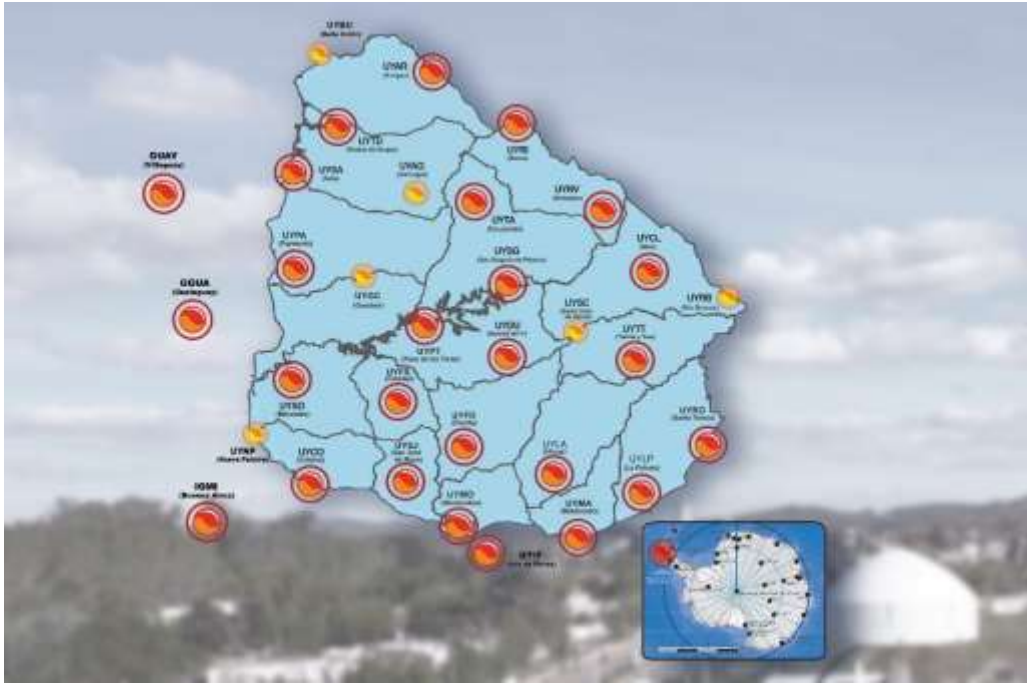


Ilustración 20 - Estaciones de referencia del IGM

Si bien el servicio REGNA-ROU actualmente no provee cobertura nacional completa con las estaciones de referencia de forma individual, se está planeado integrar seis nuevas estaciones (puntos amarillos en la Ilustración 20). Si bien añadir nuevas estaciones mejoran la cobertura del servicio, actualmente éste provee cobertura nacional completa utilizando mecanismos de redes RTK *Virtual Reference Stations* (VRS) los cuales aseguran la cobertura en áreas internas a un grupo de estaciones de referencia determinadas (ver Ilustración 21).

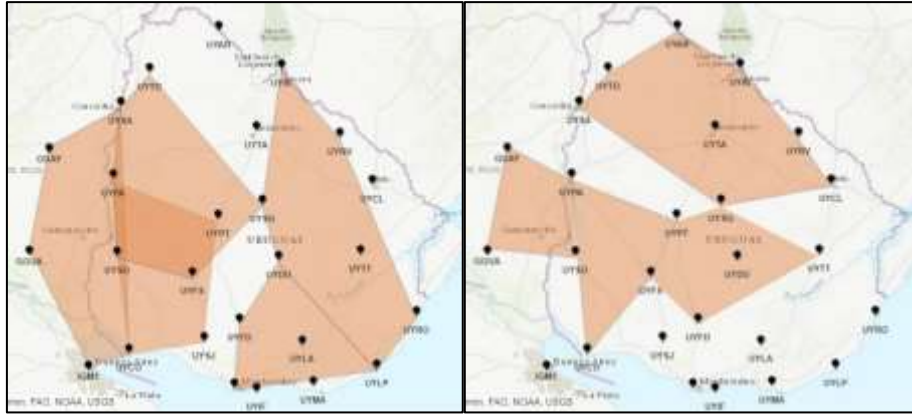


Ilustración 21 - Áreas de cobertura de las diferentes VRS

3.3 LPWAN

Existen muchas opciones de tecnologías LPWAN siendo las de interés de estudio LTE Cat M1 y Cat NB1. La diferencia principal entre ellas se encuentra en que la tecnología Cat NB1 no opera en la banda LTE, significando que los operadores deben realizar una inversión inicial para despliegue de la misma, mientras que LTE Cat M1 puede incorporarse fácilmente a las redes LTE existente, beneficiándose de todas las características de seguridad y privacidad de la misma, como la confidencialidad de la identidad del usuario, la autenticación de la entidad, la integridad de los datos y la identificación del equipo móvil.

De todas formas, Cat NB1 es auspiciado como la opción potencialmente menos costosa, ya que elimina la necesidad de una puerta de enlace, motivo por el cual, grandes empresas de telefonía como Huawei, Ericsson, Qualcomm y Vodafone están investigando activamente y haciendo un esfuerzo para comercializar Cat NB1.

Este tipo de redes fueron diseñadas para brindar conectividad a un número de dispositivos potencialmente alto, los cuales requieren pequeñas cantidades de datos durante largos periodos de tiempo, disminuyendo así el consumo de energía y alargando la vida útil del dispositivo. Una vez establecida la conexión a la red, esta tecnología introduce dos estados para el control de los recursos durante la ausencia de tráfico, PSM (*Power Save Mode*) y eDRX (*Extended Discontinuous Reception*), con el objetivo de reducir el consumo total de energía.

El eDRX está diseñado para reducir el consumo de energía del dispositivo mientras espera por mensajes en el canal. En este modo, el dispositivo no puede solicitar nuevos recursos a la red,

sin embargo, se realiza un seguimiento para mantener la sincronización de la red y para determinar si hay datos pendientes de enlace para ser recibidos. Cuando el eDRX expira, el dispositivo se mueve a PSM. En este estado, el dispositivo cierra el enlace de radio y, por lo tanto, dado que la red no puede acceder al dispositivo, las notificaciones no serán recibidas. Sin embargo, el dispositivo puede reanudar la conexión en cualquier momento, lo que implica un proceso con mucha menos sobrecarga, debido a que el dispositivo guarda la información de contexto.

En relación con el nivel de cobertura, este es mejorado permitiendo conectarse en zonas remotas o inaccesibles, estimando una mejora en la ganancia de +20dB en comparación con el servicio general de paquetes de radio (GPRS).

Para lograr esto, se introduce un mecanismo de repetición de tramas. Estas repeticiones ocurren tanto en los canales de enlace ascendente como en los de enlace descendente, siendo determinadas por la estación base de acuerdo con la intensidad de la señal recibida y reportada por el dispositivo. Sobre esto, la estación base establece una categoría para el dispositivo, llamada *Coverage Enhancement Level* (ECL), que determina el número de repeticiones llegando hasta tres niveles, desde ECL0 para operación normal a ECL2 para el peor de los casos.

Un beneficio notorio en la utilización de estas tecnologías es la gran variedad de campos en los que puede ser aplicado, integrando estos:

- Casas y edificios: permiten la medición de consumo inteligente de electricidad, gas y agua, alarmas contra intrusos e incendios, iluminación y climatización inteligente.
- Infraestructuras de ciudades: permite controlar el alumbrado de las calles, los espacios de estacionamiento gratuitos y monitorear las condiciones ambientales.
- Sanidad: permite monitorear parámetros de salud mediante dispositivos implantados.
- Agricultura y Ganadería: permitiendo mayor precisión en técnicas agrícolas, incorporación de sensores atmosféricos y monitorización del terreno, con el objetivo de conseguir una granja autónoma y conectada.
- Seguimiento de personas, animales (ganado) u objetos.
- Fábricas: permitiendo a las industrias ser mucho más eficientes, autónomas y productivas.

4 Solución propuesta

Para cumplir con todos los objetivos planteados se propone construir un sistema, al cual se le denominó RTKReady. Este, basado en la arquitectura básica de sistemas de correcciones RTK, se compone de el cliente RTK y la estación de referencia.

Para diseñar el marco general de funcionalidades del sistema, se consideraron una variedad de escenarios en los que la utilización del sistema sería de gran productividad. Esto permitió seleccionar las tecnologías y dispositivos a integrar, con el fin de abarcar y cumplir con todos los objetivos y requerimientos propuestos.

Dado que existen servicios en internet de carácter público, que permiten la adquisición de la información requerida para llevar a cabo las correcciones de posicionamiento RTK, siempre que se tenga cobertura de una red celular el escenario de funcionamiento del sistema puede ser representado por la Ilustración 22. En este, alcanza con integrar un único dispositivo (cliente RTK), el cual descargue la información de estos servidores y realice las correcciones de la posición proporcionada por su receptor GNSS. Este dispositivo debe ser capaz de, obtener su posición utilizando un receptor GNSS, establecer una conexión a los servidores de correcciones RTK mediante internet para lo cual se utilizará un módulo celular, realizar las correcciones de su posición utilizando técnicas RTK y proporcionar estas por medio de una interfaz serial para que aplicaciones externas la puedan utilizar.

Independientemente del escenario de trabajo, dado que no se pudo satisfacer el requerimiento inicial de SmartWay que consistía en la utilización de NB-IoT como tecnología de acceso a la red celular, por los motivos descritos en el capítulo 4.1.2.3. Se propuso como objetivo secundario el desarrollar las librerías necesarias para que el hardware de telemetría (TDU) de SmartWay tenga la posibilidad de interactuar con un módulo que implemente esta tecnología.

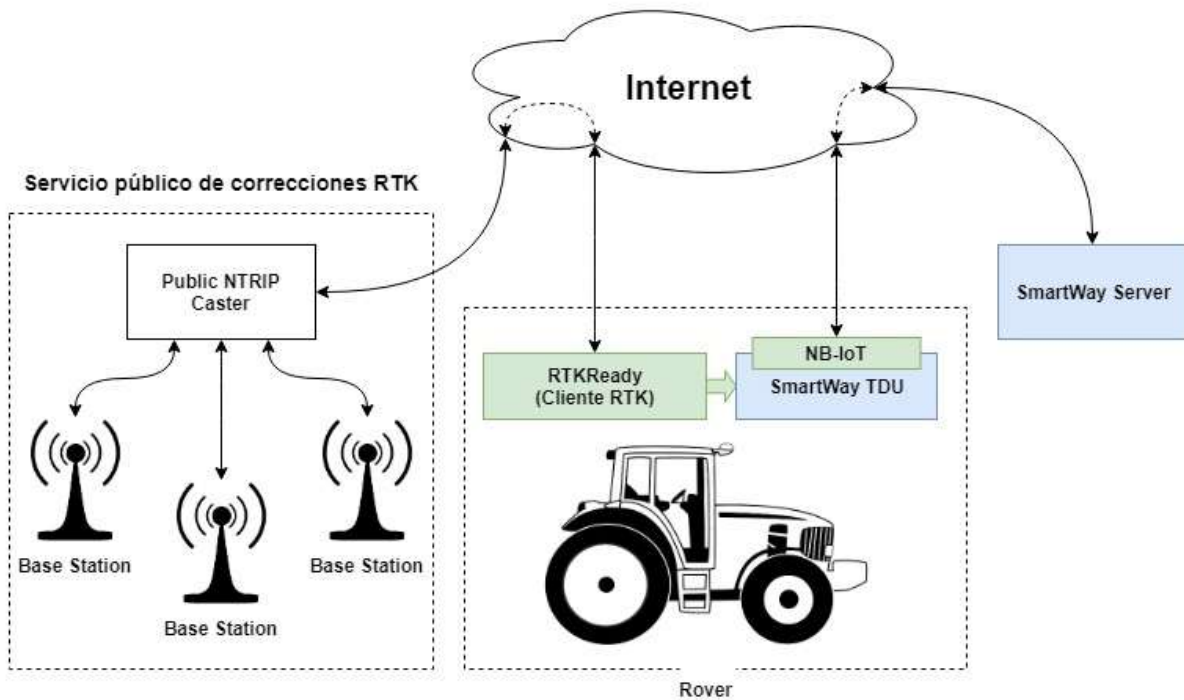


Ilustración 22 - Integración de RTKReady con servicios de correcciones públicas

Aunque el sistema en estas condiciones permite realizar correcciones RTK en la información de posición, se presentan varias desventajas que imposibilitarían la realización de estas en otros escenarios.

Al considerar como escenarios posibles, la ausencia de cobertura de la red celular, el cierre permanente o temporal de servicios públicos de correcciones o la inexistencia de estaciones de referencia públicas que proporcionen cobertura al área de aplicación del sistema, se hizo notar la necesidad de integración de un nuevo dispositivo y método de comunicación que posibiliten las correcciones en estas condiciones.

En este contexto de funcionamiento, representado por la Ilustración 23, se incorporan al sistema estaciones de referencia privadas con la capacidad publicar la información relevante para las correcciones. Esta incorporación, permite independizarse por completo de los servicios de correcciones públicos al utilizar servidores NTRIP Caster privados para publicar la información de las estaciones de referencia, y elimina la restricción de cobertura de la red celular al facilitar la utilización de un radio enlace para publicar de esta información. Este último, requiere que el cliente RTK incorpore mecanismos que posibiliten la adquisición de datos por este medio.

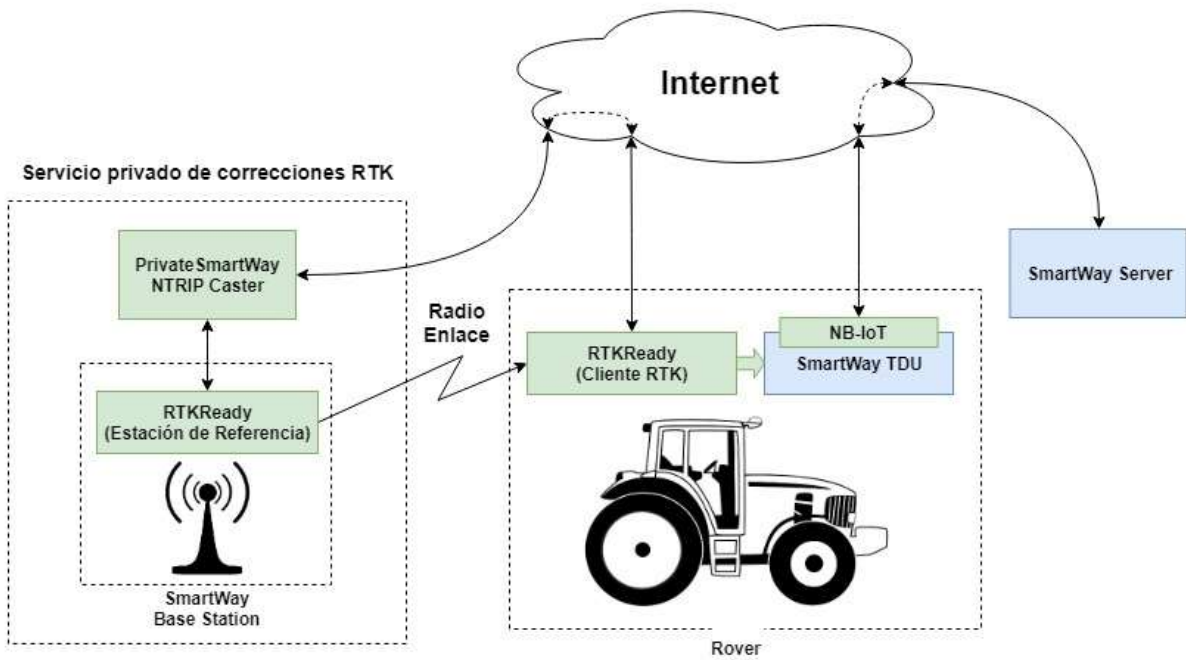


Ilustración 23 - Integración de RTKReady con servicios de correcciones privados

Como factor adicional, la incorporación de estaciones de referencia privadas permite que estas sean ubicadas dentro del área de implementación del sistema, asegurando un óptimo funcionamiento de las correcciones. Si bien, el área de cobertura RTK de una estación de referencia abarca aproximadamente 50 km de radio pudiéndose extender a 100 km, cuanto más alejado se encuentre el cliente RTK de esta, menor será el grado de precisión de la corrección. Esto puede ser contraproducente para ciertas aplicaciones donde las exigencias del nivel de precisión son elevadas.

En cuanto a la información de posición, debido a que el sistema se pensó para ser funcional en áreas con y sin cobertura de la red celular, la entrega de la posición corregida se realiza únicamente de forma local. Es decir, que el sistema se enfoca en proporcionar la información de posición precisa a dispositivos conectados físicamente al sistema.

4.1 RTKReady

Dado que en este capítulo se detallarán todos los aspectos de la implementación de RTKReady, para facilidad del lector se utilizaron nombres nemotécnicos para identificar a todo software desarrollado.

Como se mencionó anteriormente, el sistema se compone en su totalidad de clientes RTK y estaciones de referencia.

El cliente RTK o Rover como se los suele denominar, es el dispositivo encargado de realizar las correcciones de la posición y entregar ésta a un dispositivo o servicio que el usuario conecte.

La estación de referencia o *Base Station*, es la encargada de proporcionar la información relevante para que los clientes RTK puedan realizar las correcciones.

La Ilustración 24 representa las interconexiones de todas las piezas de hardware y los medios de comunicación entre ambos dispositivos.

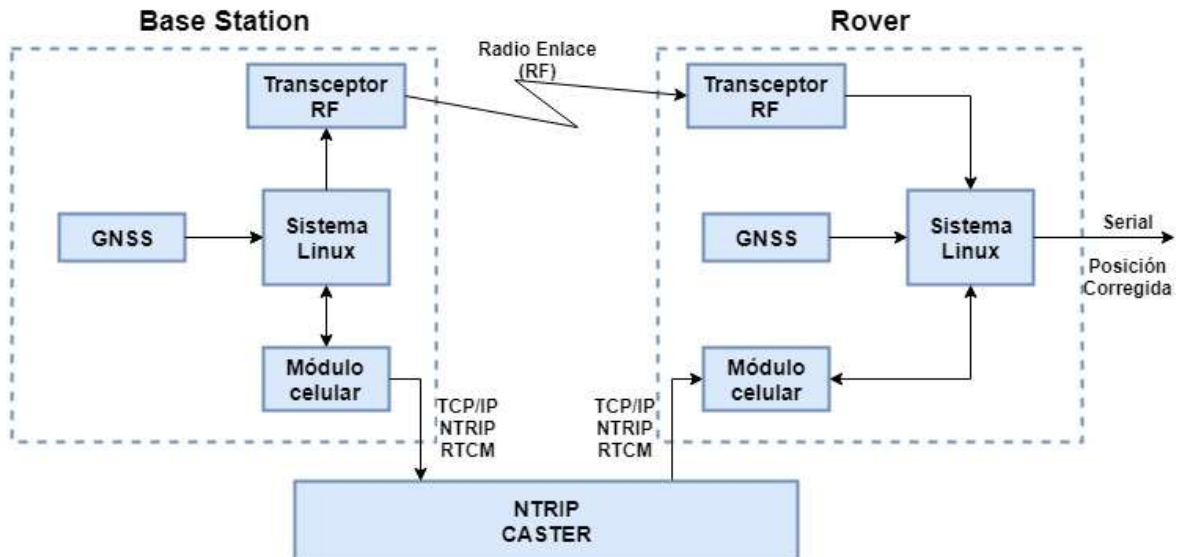


Ilustración 24 - Modelo de sistema RTKReady

Como se puede apreciar en la ilustración anterior, ambos dispositivos se componen de, un Sistema Linux, un receptor GNSS, un módulo celular y un transceptor de radiofrecuencia.

Esto hace notorio la reutilización de hardware y software para el control de los mismos, lo que beneficia en gran medida al desarrollo e implementación del sistema.

Debido a que cada uno de los dispositivos utilizados permite implementar funcionalidades específicas del sistema, se posibilita el fraccionamiento de la capa de software, en bloques destinados a tareas específicas mediante el control de cada dispositivo.

En este contexto, el receptor GNSS es el encargado de proveer a ambos dispositivos con la información de observables de fase y código requerida para el cálculo de correcciones RTK.

Para la comunicación entre la estación de referencia y el cliente RTK se utilizan el transceptor RF y el módulo celular. El módulo celular posibilita el intercambio de la información desde las estaciones de referencias, tanto públicas como privadas, necesarias para las correcciones del cliente RTK utilizando un servidor NTRIP Caster. En adicción este módulo, provee de un método para configurar remotamente los dispositivos a través de mensajes SMS. En contraste, la integración del transceptor RF permite el intercambio de esta información de forma directa entre el cliente RTK y estaciones de referencia privadas a través de un radio enlace, independizando al sistema del requerimiento de cobertura de la red celular.

Por último, el sistema Linux, mediante el software desarrollado, permite controlar a cada uno de los dispositivos e interfaces utilizadas con el fin de proveer todas las funcionalidades del sistema de manera robusta y confiable.

En los capítulos siguientes se argumentan las elecciones realizadas en cuanto a *hardware* y *software* durante la etapa de diseño. A fin de facilitar el entendimiento de estos, a continuación, se listarán las tecnologías utilizadas que fueron cruciales durante estas etapas.

En cuanto a *software*, se debe resaltar la utilización de RTKLib ya que fue el punto de partida para el diseño del sistema, y en cuanto *hardware*, se debe destacar el uso de Raspberry Pi como sistema Linux debido a que su utilización es crucial para la implementación de ciertas funcionalidades.

4.1.1 Funcionalidades

En esta etapa se dará una breve descripción de todas las características de funcionamiento inherentes al sistema.

4.1.1.1 Configuración

Para lograr una funcionalidad completa es necesario realizar pequeñas configuraciones al sistema, estas dependen de: el dispositivo a configurar, la ubicación, geografía, necesidades y/o conveniencia de cada usuario.

Con este fin, el sistema provee de dos medios por los cuales el usuario podrá realizar las configuraciones y/o solicitudes al sistema, uno de forma local y otro de forma remota. Cabe aclarar que ambos métodos están protegidos por un usuario y una contraseña que el usuario deberá configurar de forma local en la primera ejecución del sistema.

Para acceder al método de configuración local, el usuario debe conectarse al punto de acceso Wifi proveído por el dispositivo e ingresar a su página de configuración (ver Ilustración 25), para ello, basta con abrir cualquier navegador web, acceder a la dirección 192.168.0.1 e indicar las credenciales de inicio de sesión del sistema en la página de acceso.

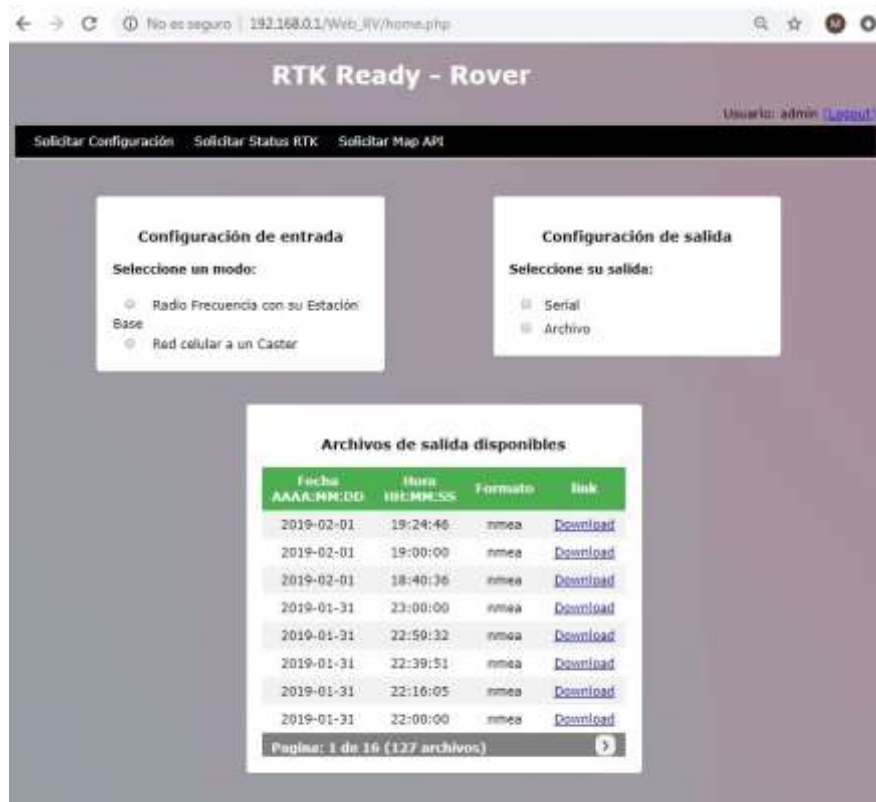


Ilustración 25 - Interfaz de configuración local de un cliente RTK

En el método de configuración remota, el usuario envía al dispositivo mensajes SMS especificando usuario y contraseña, seguido de las configuraciones a realizar bajo un

protocolo propietario. En caso de que el usuario y contraseña coincidan, y se detecte un error de sintaxis en las líneas de configuración, el sistema responde indicando el error.



Ilustración 26 - Solicitud de estado de forma remota

En el caso del cliente RTK, las configuraciones abarcan:

1. Elección del método para la obtención de correcciones provenientes de las estaciones de referencia (descargas desde un servidor NTRIP Caster en internet o mediante un radio enlace directo) y la información asociada a los mismos.
2. Elección de el o los métodos de salida para la información de posición corregida. Estos constituyen, salida por medio de una interfaz serial con velocidad configurable o por almacenamiento local en archivos, en ambos casos se puede seleccionar el formato de los datos de una lista predefinida.

Para el caso de la estación de referencia, las configuraciones son:

1. Elección de el o los métodos para la publicación de correcciones (subir a un servidor NTRIP Caster en internet o envió a los clientes RTK mediante un radio enlace) y toda información inherente a los mismos.
2. Ingreso de la posición actual de la estación de referencia mediante una posición definida por el usuario o indicar método de auto-posicionamiento, caso en el cual la información solicitada es idéntica a la configuración 1 del cliente RTK.

Ilustración 27 - Configuración local de estación de referencia

Una vez iniciado y configurado el sistema, es imprescindible tener la capacidad de conocer los parámetros actuales de funcionamiento de los dispositivos, para solucionar esto, el sistema es capaz de devolver información relevante sobre su estado actual y configuraciones que rigen los parámetros de su ejecución, a estas consultas se la identificará como peticiones (se observa un ejemplo en la Ilustración 26 realizado de manera remota).

Si bien ambos métodos permiten configurar completamente el dispositivo, el método de configuración local presenta funcionalidades extra cuando se trata del cliente RTK. Al acceder a la interfaz local, se tiene la posibilidad de descargar archivos de salida de posicionamiento del sistema, además de acceder a un reporte de estado actual con información adicional.

4.1.1.2 Información de salida

Con el fin de brindar una solución que cumpla con las necesidades de los usuarios se ofrecen dos alternativas para obtener las correcciones RTK, uno por medio de una interfaz serial y otro por medio de transferencia de archivos. En ambos casos, se provee únicamente de la información de posición, siendo ésta entregada cada 1 segundo.

Dada la importancia de ofrecer una solución en tiempo real y de fácil integración con otros dispositivo o aplicaciones que el usuario posea, se propuso entregar las correcciones RTK por medio de una interfaz serial. Dicha interfaz se rige por la norma *Recommended Standard 232* (RS-232), la cual fue elegida con el objetivo de estandarizar la comunicación. Con el mismo fin se permite configurar la velocidad de transmisión bajo velocidades estándares, evitando posibles incompatibilidades.

En caso de que el usuario quiera un histórico de las posiciones o no se tenga la posibilidad de conectarse mediante la interfaz serial, también se puede guardar las posiciones en forma de archivos de texto directamente en la memoria de la Raspberry Pi, para ser descargados posteriormente a través de la interfaz de configuración local tal como se muestra en la Ilustración 28.

The screenshot shows a web interface titled "Archivos de salida disponibles". It contains a table with four columns: "Fecha", "Hora", "Formato", and "link". The table lists several files, all in "nmea" format, with dates ranging from 2019-01-31 to 2019-02-01. Each row has a "Download" link. At the bottom, there is a pagination bar showing "Pagina: 1 de 16 (127 archivos)" and a right arrow button.

Fecha AAAA:MM:DD	Hora HH:MM:SS	Formato	link
2019-02-01	19:24:46	nmea	Download
2019-02-01	19:00:00	nmea	Download
2019-02-01	18:40:36	nmea	Download
2019-01-31	23:00:00	nmea	Download
2019-01-31	22:59:32	nmea	Download
2019-01-31	22:39:51	nmea	Download
2019-01-31	22:16:05	nmea	Download
2019-01-31	22:00:00	nmea	Download

Pagina: 1 de 16 (127 archivos) >

Ilustración 28 - Descarga local de archivos de salida

Dado que los archivos son manipulados por los usuarios, se limita el almacenamiento de posiciones a un máximo de 4 horas de trabajo continuas, generando un nuevo archivo una vez cumplidas las 4 horas o se reinicie o reconfigure el sistema. Esto se realiza para evitar la creación de archivos excesivamente grandes o que sean difíciles de utilizar.

En cuanto al protocolo de la capa de datos, ambos métodos proveen de la posición bajo el protocolo estándar NMEA 0183, o directamente las posiciones en el formato Latitud/Longitud/Altura o x/y/z (tomando el centro de la tierra como origen de coordenadas), siendo estas configurables según las necesidades de los clientes.

4.1.1.3 Periféricos

Los periféricos, son los dispositivos o componentes que posibilitan la interacción física entre el usuario y el sistema. Estos son capaces de proporcionar un rápido diagnóstico del estado del sistema, facilitar el acceso a la interfaz de configuración local y proveer de un medio de reinicio manual.

Como se mencionó anteriormente, es de vital importancia conocer el estado del sistema, teniendo en consideración que el usuario habitual puede no tener acceso a las interfaces de configuración, se provee de un método de diagnóstico rápido y sencillo mediante el uso de indicadores led. Estos permiten determinar si el sistema está encendido y proporcionan información sobre el nivel de corrección de la posición, actividad en la recepción GNSS y estado de los enlaces de radiofrecuencia e internet (ver Tabla 3).

Posición	Función	Color	Modo
1	Sistema Encendido	Azul	Estático
2	Nivel de corrección	Rojo – <i>Single</i> Anaranjado – <i>Float</i> Verde – <i>Fix</i>	Estático
3	Enlace RF	Verde	Parpadeante (*)
4	Enlace a NTRIP Caster	Verde	Parpadeante (*)
5	Actividad GNSS	Verde	Parpadeante (*)
(*) Frecuencia 1Hz, ciclo de encendido proporcional a velocidad del enlace (0 - 1,9 Kbps)			

Tabla 3 - Descripción de los indicadores led

Para facilitar el control del sistema al usuario, se provee de un pulsador. Al presionarlo, se activan diferentes mecanismos que permiten realizar un reinicio del dispositivo o habilitar el acceso a las configuraciones locales. Si la pulsación se prolonga por más de diez segundos, se produce el cierre y apertura de todas las capas software asociadas al sistema RTKReady, lo que permite reiniciar el sistema sin necesidad reiniciar el sistema Linux.

Para habilitar el acceso a la interfaz de configuración local, se debe presionar el pulsador, por un tiempo menor a diez segundos. Esto produce la activación de la interfaz Wifi y los servicios imprescindibles que otorgan acceso a esta interfaz de configuración.

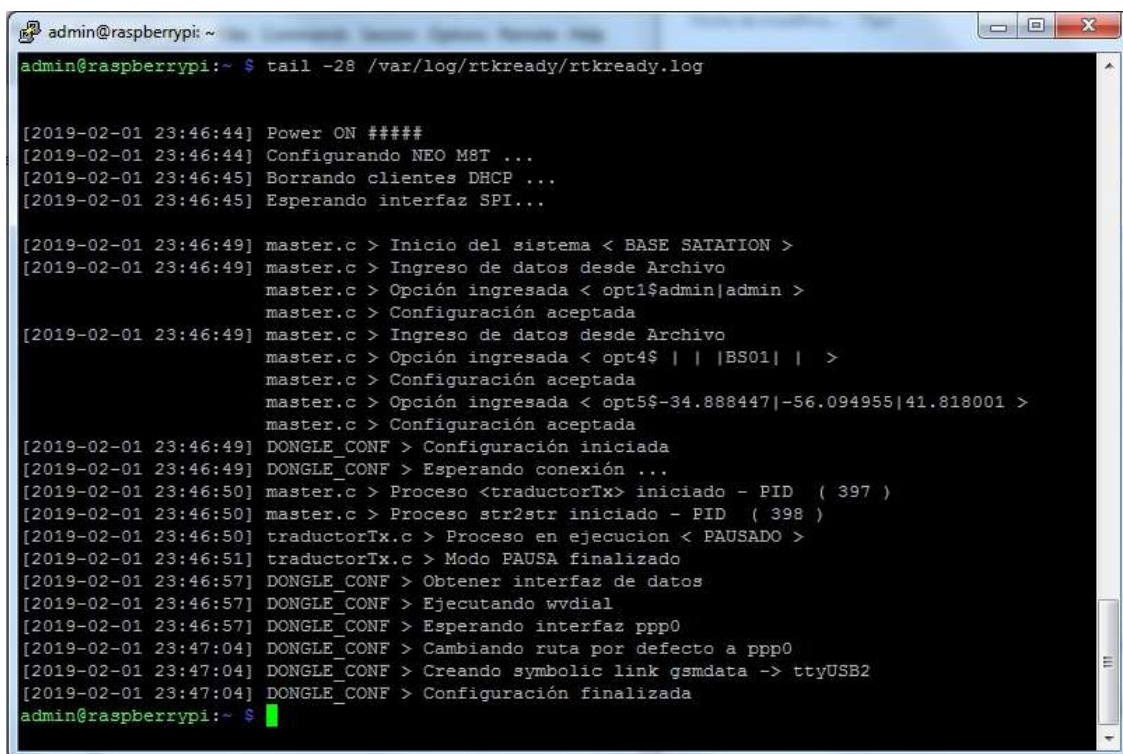
Dado que no se requieren configuraciones habituales, la interfaz de configuración local se deshabilita siempre que no haya usuarios conectados, esto involucra el apagado de la interfaz Wifi y el cierre de todos los servicios inherentes. Esto permite reducir la carga sobre el

sistema Linux, pudiéndose reactivar siempre que el usuario lo requiera presionando nuevamente el pulsador.

4.1.1.4 Administración

A fin de diagnosticar problemas específicos que puedan surgir durante la utilización de los dispositivos, se debe contemplar opciones de administración del sistema en general para que sean revisadas por técnicos especializados. Para administrar el sistema, los técnicos deben conectarse al punto de acceso Wifi provisto por los dispositivos y acceder al sistema Linux utilizando el protocolo SSH. Si por algún motivo la conexión Wifi es insostenible, siempre se puede acceder añadiendo un monitor y un teclado.

Para simplificar la tarea de administración de los dispositivos, se genera un archivo con información de todos los eventos relevantes durante la ejecución del sistema.



```
admin@raspberrypi:~ $ tail -28 /var/log/rtkready/rtkready.log

[2019-02-01 23:46:44] Power ON #####
[2019-02-01 23:46:44] Configurando NEO M8T ...
[2019-02-01 23:46:45] Borrando clientes DHCP ...
[2019-02-01 23:46:45] Esperando interfaz SPI...

[2019-02-01 23:46:49] master.c > Inicio del sistema < BASE SATATION >
[2019-02-01 23:46:49] master.c > Ingreso de datos desde Archivo
master.c > Opción ingresada < opt1$admin|admin >
master.c > Configuración aceptada
[2019-02-01 23:46:49] master.c > Ingreso de datos desde Archivo
master.c > Opción ingresada < opt4$ | | |BS01| | >
master.c > Configuración aceptada
master.c > Opción ingresada < opt5$-34.888447|-56.094955|41.818001 >
master.c > Configuración aceptada
[2019-02-01 23:46:49] DONGLE_CONF > Configuración iniciada
[2019-02-01 23:46:49] DONGLE_CONF > Esperando conexión ...
[2019-02-01 23:46:50] master.c > Proceso <traductorTx> iniciado - PID ( 397 )
[2019-02-01 23:46:50] master.c > Proceso str2str iniciado - PID ( 398 )
[2019-02-01 23:46:50] traductorTx.c > Proceso en ejecución < PAUSADO >
[2019-02-01 23:46:51] traductorTx.c > Modo PAUSA finalizado
[2019-02-01 23:46:57] DONGLE_CONF > Obtener interfaz de datos
[2019-02-01 23:46:57] DONGLE_CONF > Ejecutando wvdial
[2019-02-01 23:46:57] DONGLE_CONF > Esperando interfaz ppp0
[2019-02-01 23:47:04] DONGLE_CONF > Cambiando ruta por defecto a ppp0
[2019-02-01 23:47:04] DONGLE_CONF > Creando symbolic link gsmdata -> ttyUSB2
[2019-02-01 23:47:04] DONGLE_CONF > Configuración finalizada
admin@raspberrypi:~ $
```

Ilustración 29 - Visualización del archivo de eventos

4.1.2 Hardware

En este capítulo se describen todos los argumentos y motivos para la elección de cada una de las piezas de hardware integradas en el sistema.

4.1.2.1 Sistema Linux

Para la implementación del sistema se decidió utilizar un sistema Linux como centro de cómputo principal, ya que no solo permite el uso de librerías sino también el uso directo de paquetes de software que agilicen la implementación.

Al buscar sistemas Linux de bajo costo que proporcionarían al menos cuatro interfaces UART nativas, nos encontramos que, la gran mayoría presentaban una única interfaz UART adicionando interfaces como SPI e I2C. Sin embargo, aquellas que presentaban la cantidad requerida de interfaces UART no estaban del todo desarrolladas a nivel de integración Software-Hardware o no poseían críticas positivas de la comunidad.

Como no se encontró un dispositivo que proporciona cuatro interfaces UART, se decidió sustituir por una interfaz I2C, SPI y simulación de interfaz serial en los GPIO. Con las características anteriores se evaluaron diversos dispositivos, siendo los más relevantes Raspberry Pi, Banana PI, Orange Pi.

Se decidió utilizar una Raspberry Pi Zero W (ver Ilustración 30) ya que además de cumplir con los requisitos anteriores, esta tiene una larga trayectoria y se caracteriza por ser un sistema muy estable.



Ilustración 30 - Raspberry Pi Zero W

Una característica secundaria que afirmó la elección de la Raspberry Pi Zero W fue que esta posee un módulo Wifi integrado, el cual es de gran utilidad para el acceso local al dispositivo.

4.1.2.2 Receptor GNSS

La elección del receptor GNSS es crucial a la hora de implementar un sistema RTK, este debe ser capaz de obtener las observables de código y fase para posibilitar las correcciones.

En el mercado actual existen varias empresas que ofrecen receptores GNSS con esta característica, entre las cuales se encuentran SkyTraq, Novatel, NVS Technologies y Ublox entre otras. Sin embargo, la gran mayoría lo hace únicamente bajo protocolos propietarios o añadiendo RTCM 3 solo en los receptores con funcionalidades avanzadas.

Para la realización del proyecto, se optó por utilizar el receptor Ublox NEO-M8T (ver Ilustración 31), el cual proporciona la información de las observables bajo el protocolo UBX propietario de Ublox. Para su elección se tuvo en consideración que es el receptor de menor costo que ofrece compatibilidad con las librerías de RTKLIB a nivel de protocolo, además de los buenos resultados en pruebas de campo reportados por la comunidad.



Ilustración 31 - Placa de desarrollo Ublox NEO-M8T

Si bien el módulo provee de USB, UART, SPI e I2C como interfaces de comunicación, al adquirir una placa de evaluación que simplificará las pruebas iniciales de concepto, se optó por una con puerto USB integrado. Esto, debido a la estructura interna del receptor, deja inhabilitada la interfaz SPI dejando únicamente activas las interfaces USB, UART e I2C.

Debido a la limitada cantidad de interfaces UART que provee la Raspberry Pi, la interconexión entre ésta y el receptor GNSS, se realizaría utilizando la interfaz I2C. Luego de una exhaustiva etapa de pruebas en la que no se obtuvieron los resultados esperados, se realizó una nueva investigación de la tecnología, en la que se encontró que el controlador de I2C utilizado por Raspberry Pi no soporta la comunicación con dispositivos esclavos que utilicen el mecanismo de *Clock Stretching*, el cual es implementado por el receptor GNSS de Ublox. Este mecanismo accionado por el esclavo permite retener la señal de reloj para enlentecer la comunicación y asegurarse que el máster espere hasta que sea posible transmitir la información.

Dado este impedimento, para el cual no existe solución, se descarta la idea de comunicar el receptor GNSS por medio de I2C, realizándose por interfaz UART.

4.1.2.3 Módulo celular

Dada la propuesta de SmartWay, se pretendía integrar al sistema un módulo celular que implementará NB-IoT, el cual sería el encargado de proporcionar conexión a internet al sistema.

Debido a que ningún operador en Uruguay provee acceso a esta tecnología y que las interfaces proveídas por la Raspberry pi son escasas, se decidió utilizar un módem USB compatible con Raspberry Pi que implemente EGPRS o 3G.

La utilización de un módem compatible con Raspberry Pi trae consigo varias ventajas, en primera instancia asegura una sencilla configuración e implementación de la conexión a internet al utilizar software libre ya existente y como la conexión del módem se realiza por medio de una interfaz USB estándar, cada usuario es libre de decidir si desea adquirir el módem o no, dependiendo de sus necesidades.

Sin embargo, una desventaja notoria, es que, al ser un dispositivo extraíble, requieren de software dedicado que soporte conexiones y desconexiones aleatorias.

Durante la investigación de módems, se descubrió que gran variedad de módems USB Huawei eran compatibles con Raspberry Pi. Dado que se tenían en adquisición dos módems Huawei modelo E3131 y E1756, siendo únicamente el modelo E3131 identificado como

compatible, se procedió a realizar pruebas con ambos donde se obtuvieron excelentes resultados.

Cabe aclarar que la elección de estos módems no es restrictiva, ya que existe una gran variedad de fabricantes que presentan compatibilidad y la integración de cualquiera de estos es posible bajo pequeñas modificaciones de configuración.



Ilustración 32 - Modem Huawei E3131



Ilustración 33 - Modem Huawei E1756

4.1.2.4 Transceptor de radio frecuencia

Para determinar el módulo transceptor a ser utilizado hubo que considerar varios factores, siendo estos: alcance, potencia, espectro de frecuencias, tipo de modulación e interfaces de comunicación disponibles.

Lo primero a definir es la máxima distancia a la cual se quiere transmitir la información de la estación base. Se estimó que un alcance de 4 km cumpliría con las demandas de los usuarios, fuera de este rango se tendrá que recurrir a uno o varios repetidores que incrementen la potencia de la señal transmitida o a la integración de nuevas estaciones de referencia.

Si bien, la utilización de otra estación de referencia puede ser una alternativa a un repetidor, la utilización de este último trae consigo varias ventajas. Este permite, ampliar el área de cobertura del radio enlace sin necesidad de reconfigurar el sistema evitando el reinicio de las correcciones al cambiar de estación de referencia, evitar el uso de servidores externos que implemente técnicas complejas de redes RTK al reunir todas las estaciones de referencia, y

reducir el costo de implementación ya que éste requiere de menor cantidad de piezas de hardware.

A continuación, se procede a la elección de la potencia y el espectro de frecuencias del módulo, siendo estas restringidas por la Unidad Reguladora de Servicios de Comunicaciones (URSEC). La URSEC establece bandas del espectro de frecuencias que son de libre utilización bajo restricciones de máxima potencia emitida, a modo de no generar interferencia con diversos servicios que utilizan radiofrecuencia. Las frecuencias de libre acceso son 860 MHz, 915 MHz, 2.4 GHz y 5.8 GHz siendo la potencia máxima permitida en emisión inferior a 36 dBm.

Dado que las pérdidas en transmisión son directamente proporcionales al logaritmo de la frecuencia de modulación, se optó por utilizar módulos transceptores de las frecuencias más bajas. Al buscar en el mercado se vio que los transceptores de 915 MHz eran los más abundantes y económicos. Se destacaron varios módulos entre ellos: Xbee Pro 53B, Shield RFM95 LoRa, RFM95 LoRa y LoRa Bee, tomando en cuenta sus características, costos, envíos, y si deben ser homologados.

Se optó por el módulo LoRa Bee del fabricante Dragino el cual integra el transceptor RFM95. Las características determinantes para la elección fueron la posibilidad de proveer SPI como interfaz de comunicación, su alta sensibilidad siendo ésta -148 dBm y además ofrece varios tipos de modulación como: FSK, GFSK, MSK, GMSK, LoRa y OOK.

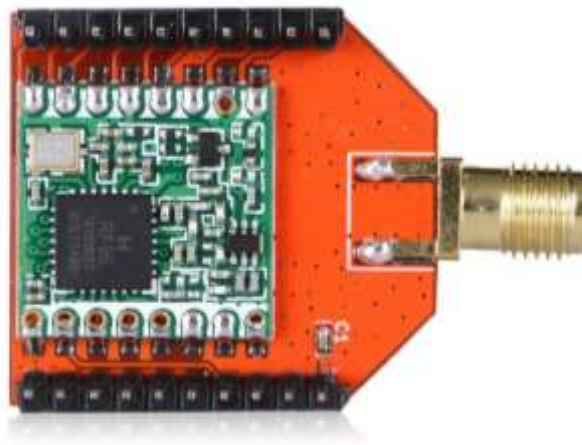


Ilustración 34 - Transceptor de radiofrecuencia LoRa Bee

A propósitos del proyecto se decidió implementar la modulación LoRa ya que esta fue diseñada con el objetivo de tener un bajo consumo de potencia y un alto alcance en su cobertura, siendo estas características ideales para la estación base. Al utilizar modulación LoRa se limita las velocidades de transferencia de datos a unos 50 Kbps aproximadamente, esto no tendría impactos en el rendimiento de RTKReady, pero si el usuario deseara implementar otra modulación se podría realizar con mínimos ajustes de código.

4.1.2.5 Antenas

A continuación, se procede a explicar las características que deben tener las antenas para un correcto funcionamiento de RTKReady, ya sean las antenas de los receptores GNSS o las antenas de los transceptores de radio frecuencia.

Para un óptimo rendimiento de cualquier sistema RTK se busca que las antenas tengan la mayor ganancia posible, la capacidad de soportar múltiples constelaciones y además para cada constelación deberá soportar la mayor cantidad de frecuencias.

Las antenas GNSS que fueron utilizadas en el proyecto fueron proporcionadas por SmartWay, siendo estas A.40 Hercules y A.30 Ultima, ambas del fabricante Taoglas. Dichas antenas tienen una ganancia típica de 27 dB a unos 3.0 V y soportan únicamente la banda L1 para las constelaciones de GPS, GLONASS y Galileo.



Ilustración 35 - Antena GNSS A.40 Hercules, Taoglas

Ilustración 36 - Antena GNSS A.30 Ultima, Taoglas

Haciendo referencia a las antenas utilizadas por los transceptores de radiofrecuencia, lo buscado en sus características son alta ganancia y la capacidad de trabajar a la frecuencia de 915 MHz.

En el proyecto se optó por utilizar las antenas Delta 6A del fabricante Sireta para la estación base y ANT-GSMSTUB4 de RF Solutions para el cliente RTK. Ambas antenas son capaces de trabajar a la frecuencia de 915 MHz, tienen ganancia cercana a los 4 dBi y además son de los más económicos en el mercado.



Ilustración 37 – Antena Delta 6A, Sireta



Ilustración 38 – Antena ANT-GSMSTUB4, RF Solutions

Adicionalmente para el sistema RTKReady se busca que todas las antenas sean lo suficientemente robustas para trabajar en ambientes hostiles, en condiciones de alta humedad o incluso bajo lluvia. Debido a la utilidad y locación de cada uno de los dispositivos se tuvo en consideración la longitud del cable conector, siendo estos de 3m de longitud para el caso del cliente RTK, ya que el dispositivo se ubica en el interior de las maquinarias y las antenas en el exterior.

4.1.2.6 Regulador de tensión

Para la alimentación de los dispositivos se requiere de un regulador de tensión de 5V. Debido a que las maquinarias agrícolas proveen de suministros de tensión de 12V o 24V, se utilizó un regulador *Step Down* el cual suministra una tensión constante de 5V para cargas menores a 2A.

4.1.2.7 Conversor RS-232

Dado que información de salida se proveerá por medio de una interfaz serial y la Raspberry Pi provee la salida utilizando niveles *Transistor-Transistor Logic* (TTL) de 3.3V, a fin de estandarizar la comunicación y proveer mayor inmunidad al ruido, se adicionó un conversor que integra un chip MAX232 el cual convierte de serial TTL a RS-232.

4.1.2.8 Prototipo

Para realización del sistema, se construyó un prototipo a utilizarse para el cliente RTK y un prototipo a utilizarse como estación de referencia. Estos, reúnen todos los componentes de hardware ya mencionados en una estructura que permite: el desplazamiento, montaje en cualquier lugar de forma rápida y sencilla, y evitar desconexiones involuntarias que afectan la integridad de las pruebas.

Los componentes que se ilustran en cada uno de los sistemas son: leds de indicación, las antenas GNSS y RF, el Modem 3G, los dispositivos NEO-M8T, LoRa Bee y la Raspberry Pi. La diferencia entre ambos dispositivos se encuentra en que el cliente RTK provee de una interfaz de salida serial para lo cual integra un módulo conversor RS-232.



Ilustración 39 - Prototipo de cliente RTK



Ilustración 40 - Prototipo de estacion de referencia

Adicionalmente, se pretende utilizar un torno CNC (*Computer Numerical Control*) para integrar al sistema en un PCB (*Printed Circuit Board*) de doble capa como prototipo final. Sin embargo, debido a problemas de tiempo no fue posible culminar la integración final priorizando etapas de mayor importancia.

Para este último prototipo se realizó el diseño del PCB doble capa y la construcción de la misma en el torno CNC (ver Ilustración 41 e Ilustración 42), faltando únicamente la integración de todas las partes (ver Ilustración 43).

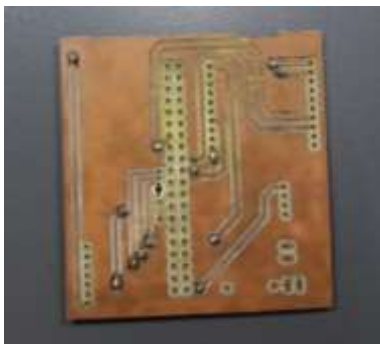


Ilustración 41 - Construcción de PCB lado inferior



Ilustración 42 - Construcción de PCB lado superior



Ilustración 43 - Componentes para el desarrollo del prototipo final

4.1.3 Software

Dado que la base del sistema es la integración de hardware y software, la elección y/o diseño de software es crucial a fin de proveer estabilidad al sistema. Durante este capítulo se detallarán los motivos de la elección de software crítico y argumentos del modelado de software a implementar.

4.1.3.1 RTKLib

Durante la búsqueda de librerías y/o software que realizará correcciones RTK se encontraron tres opciones de interés: RTKLib, NTRIP Client de Lefebure y BKG NTRIP Client.

Al investigar las funcionalidades que estos proveían como sistemas de correcciones RTK, se optó por utilizar RTKLIB. Siendo que todas las opciones eran de código abierto, RTKLib presentó mayor versatilidad y cantidad de funcionalidades, no solo para las funciones en el Rover, sino que también para las estaciones base. Otro factor importante a la hora de realizar la elección fue el gran volumen de información disponible debido a su amplio uso en la comunidad.

Como funcionalidades relevantes, RTKLib provee aplicaciones funcionales de NTRIP Client y NTRIP Server para sistemas Windows y Linux, las cuales son de gran relevancia para la implementación del sistema. La aplicación NTRIP Client, facilita la adquisición de datos provenientes de los NTRIP Caster y el cómputo de las correcciones requeridas por el cliente RTK. En el caso del NTRIP Server, provee de todas las funcionalidades necesarias para obtener y publicar en un NTRIP Caster la información necesaria para las correcciones.

Adicionalmente, las aplicaciones con interfaz gráfica para Windows simplificaron en gran medida la etapa de prueba de conceptos, ya que estas facilitan la configuración, conexión y seguimiento del funcionamiento de los dispositivos involucrados en las correcciones.

A modo de identificación, de ahora en más, se hará referencia a las aplicaciones que auspician de NTRIP Client y NTRIP Server, con los nombres utilizados por RTKLib, siendo estos rtkrcv y str2str respectivamente.

Las funcionalidades, métodos y protocolos de conexión implementados por estas aplicaciones, en su gran mayoría, definieron el contexto de conexión de otros procesos con estas. Por tal motivo, a continuación, se detallará toda la información relevante asociado al funcionamiento de las aplicaciones utilizadas de RTKLib.

En cuanto a rtkrcv, esta es la aplicación utilizada por cliente RTK para la realización de las correcciones. Con este fin, provee de múltiples métodos para el ingreso y salida de datos, de los cuales se utilizarán los más convenientes y de mayor utilidad para el correcto funcionamiento del sistema.

Para la obtención de los datos GNSS a corregir, al realizarse de manera local, se utiliza el método de recepción serial directo a la interfaz UART conectada al receptor GNSS.

Para la obtención de correcciones provenientes de las estaciones de referencia, se utilizan dos métodos, en caso utilizarse el radioenlace, las correcciones se obtienen mediante una interfaz serial que se encuentra conectada al proceso de control de módulo LoRa Bee y si se utiliza un servidor de correcciones en internet, la aplicación posibilita la obtención de correcciones desde los NTRIP Server cumpliendo la función de NTRIP Client.

Por último, para la salida de la posición corregida, rtkrcv permite: el almacenamiento en archivos y la salida por medio de interfaz serial la cual se conectará al Traductor Serial. En el

caso de los archivos, otorga la posibilidad de nombrarlos en base a su fecha y hora de creación.

En el caso de str2str, la información de posición proveída por el receptor GNSS se obtiene por medio de una interfaz serial, convierte esta información de mensajes propietarios de Ublox a mensajes RTCM y proporciona diferentes mecanismos para el envío de estos al cliente RTK. Vale aclarar, que los tipos de mensaje RTCM y la frecuencia con la que estos son enviados es configurable.

Para la publicación de las correcciones, se utilizan dos métodos, en el caso de publicarse mediante radioenlace, los mensajes RTCM son entregados utilizando una conexión serial directa al proceso de control del módulo LoRa Bee y si se publican en servidor de correcciones en internet, str2str permite conectarse a un NTRIP Caster cumpliendo la función de NTRIP Server.

Como factor adicional, ambas aplicaciones durante su ejecución proporcionan información relacionada al estado y velocidades de las conexiones establecidas, así como también datos de funcionamiento general. Esta información se utilizará a fin de proveer un reporte del funcionamiento del sistema al usuario siempre que éste lo solicite.

4.1.3.2 Diseño base

Dado que el software para la realización de las correcciones RTK es crucial para la implementación del sistema, una vez realizada la elección de RTKLib, fue necesario investigar, identificar y evaluar todas las funcionalidades, medios de comunicación y protocolos provistas por este Software. Esto asegura que las capas de software a implementar sean compatibles en cuanto a comunicación entre procesos y sin funcionalidades redundantes.

Culminada esta etapa, conociendo las interfaces de comunicación de las piezas de hardware y las funcionalidades requeridas, se construyó un diagrama de bloques para el cliente RTK (ver Ilustración 44) y otro para la estación de referencia (ver Ilustración 45), que representa los métodos de interacción entre todos los bloques de software y la conexión de estos con las piezas de hardware externas.

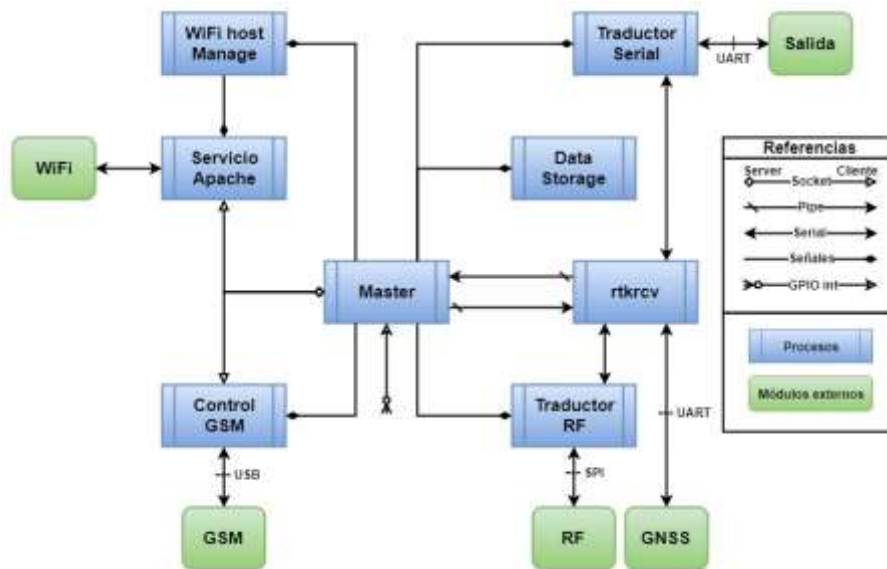


Ilustración 44 - Diagrama de bloques de cliente RTK

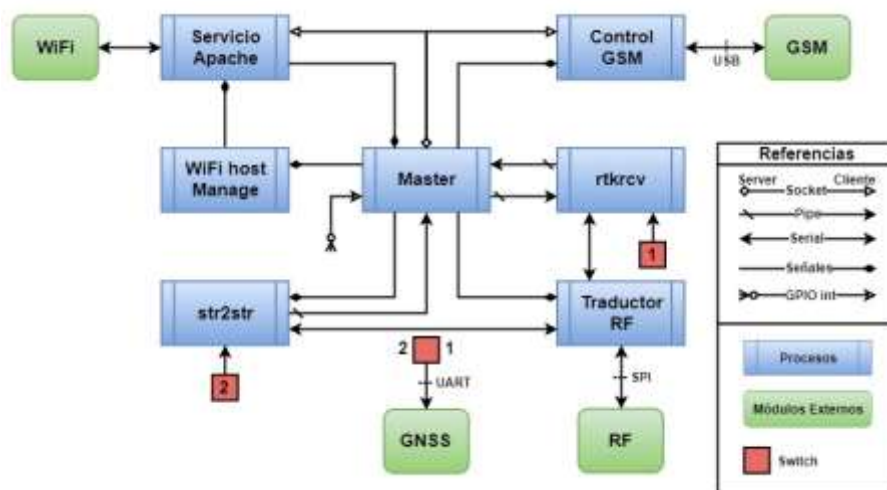


Ilustración 45 - Diagrama de bloques de estación de referencia

Comparando las dos ilustraciones anteriores, se puede apreciar que ambos dispositivos comparten una cantidad importante de bloques funcionales. Esto se debe a que la funcionalidad de auto-posicionamiento de la estación de referencia requiere de gran parte de las funcionalidades provistas por el cliente RTK y como se había mencionado anteriormente, la reutilización de hardware trae consigo una reutilización importante del software diseñado para su control. Vale aclarar que, la interfaz UART utilizada para comunicarse con el receptor GNSS debe intercambiarse entre, rtkrcv para auto-posicionarse (*switch* 1 de Ilustración 45) y

str2str para funcionar exclusivamente como estación de referencia (*switch* 2 de Ilustración 45).

Otro beneficio notorio de este modelo de sistema es la programación modular, la cual facilita la división de tareas entre los integrantes del equipo de desarrollo. Esto se logra predefiniendo los métodos y protocolos de comunicación entre los diferentes procesos y así independizar la implementación de los mismo. Adicionalmente, ante el reemplazo o sustitución de una tecnología, únicamente se debe reemplazar o modificar el bloque de software desarrollado para su control, sin necesidad de modificar otras etapas del sistema o bajo modificaciones mínimas.

Una vez diagramada la totalidad del sistema, se procede a la implementación de cada una de las partes, a fin de cumplir con todos los requerimientos.

En los capítulos siguientes, se describirán las funcionalidades básicas de todos los procesos involucrados en la implementación del sistema.

4.1.3.3 Master

Este proceso desarrollado en lenguaje C, se ejecuta al encender la Raspberry Pi y se caracteriza por ser, el proceso principal del sistema, el más complejo a nivel de software desarrollado y el que integra la mayor cantidad de funcionalidades.

A grandes rasgos el Master se encarga de:

- Iniciar, monitorear y detener los demás procesos, dependiendo del estado y los parámetros de funcionamiento del sistema.
- Posibilitar la adquisición de configuraciones, actualización de los parámetros de funcionamiento del sistema y persistencia de estos ante el apagado del sistema.
- Control de los periféricos del sistema (indicadores led y pulsador).

Siendo que los parámetros de funcionamiento del sistema son configurables, el proceso Master es capaz de iniciar, monitorear y detener la totalidad de los procesos, con el fin de mantener activos únicamente los procesos indispensables para satisfacer con las necesidades, peticiones y/o configuraciones indicadas por el usuario.

Con el fin de posibilitar la recepción de peticiones y configuraciones, el Master provee de un servidor socket local, en donde los procesos encargados de interactuar con los usuarios se conectarán a modo de cliente. Esto asegura, que el Master sufrirá modificaciones mínimas o nulas ante la eliminación, modificación, reemplazo o agregación de un método de interacción con el usuario (interfaces de configuración).

En relación con la comunicación entre el servidor y los clientes, esta se realiza utilizando un protocolo propietario, en el cual se identifica cada mensaje con una cabecera específica y se añade toda la información inherente al mensaje separada por un carácter especial. Para identificar cada conexión y dejar registro de lo ocurrido, cada cliente se debe identificar al inicio de la conexión con una cabecera única que será validada por el Master, permitiendo a este controlar y registrar las conexiones activas. Esta cabecera consiste en un carácter único para cada proceso, '3' en el caso del proceso Control GSM o 'W' en el Servicio Apache, seguido de un salto de línea.

Otra funcionalidad de gran importancia es la persistencia de las configuraciones del sistema, lo cual requiere que el sistema se inicie bajo los parámetros de funcionamiento que poseía con anterioridad a su último apagado. Para lograr esto, el Master almacena las configuraciones en un archivo durante su ejecución y las restablece cada vez que se inicia el sistema.

Como funcionalidad final, este proceso es el encargado de controlar los periféricos del sistema. En el caso de los indicadores led, el proceso utiliza toda la información relevante a su alcance para indicar el estado del sistema mediante encendido y apagado de indicadores led. Para el caso del pulsador, este proceso es el encargado de determinar la duración de la pulsación y con esto realizar las acciones adecuadas al evento, tal como se explicó en el capítulo 4.1.1.3.

4.1.3.4 Traductor RF

Desde un punto de vista genérico, el Traductor RF facilita la comunicación entre el módulo LoRa Bee y las aplicaciones de RTKLib, ya sea rtkrcv o str2str. El Traductor RF engloba dos procesos, dependiendo si el sistema se ejecuta como cliente RTK o como estación de referencia, iniciándose como Traductor Tx o Traductor Rx respectivamente. Para el desarrollo de estos se utilizó el lenguaje C.

En ambos casos los procesos se comunican a través de la interfaz SPI con el módulo LoRa Bee para obtener o enviar información por radio frecuencia según la configuración del sistema. Por el otro lado los procesos Traductor Tx y Traductor Rx se comunican con los procesos str2str y rtkrcv respectivamente a través de una pseudo-terminal, la cual permite la comunicación emulando una interfaz serial.

El Traductor Tx se comunica con el proceso str2str, con el objetivo de obtener los mensajes RTCM, para luego ser transmitidos al cliente RTK. Adicionalmente el Traductor Tx se encarga de armar la cabecera de los mensajes a ser enviados y dividir los mensajes RTCM en tramas. Dicha cabecera consiste en el *mountpoint* (identificación de la estación base), cantidad de tramas y número actual de la trama siendo enviada. La necesidad de dividir los mensajes RTCM en tramas es consecuencia de que el módulo LoRa Bee es capaz de enviar mensajes de hasta 256 bytes y los mensajes RTCM están compuestos de un largo variable con un máximo de 1029 bytes. Una vez creadas las tramas de los mensajes RTCM el traductor Tx se los entrega al LoRa Bee para su posterior transmisión, vale aclarar que no se envían dos mensajes RTCM dentro de una misma trama.

El enviar cada mensaje RTCM de manera independiente tiene la ventaja que al perderse un paquete se pierde únicamente un mensaje RTCM, pero se paga el costo de no aprovechar plenamente el canal ya que la última trama puede no ocupar el total de bytes disponibles a ser enviados.

Análogamente la función del Traductor Rx es obtener todas las tramas detectadas por el LoRa Bee, descartando mensajes provenientes de un *mountpoint* no deseado y los cuales les falta alguna trama. Una vez filtrados y reconstruidos los mensajes RTCM el Traductor Rx procede a entregar los mensajes al proceso rtkrcv el cual los utiliza para realizar correcciones RTK.

Debido a que el módulo LoRa Bee transmite en forma de *broadcast* y que el Traductor Rx discrimina los paquetes por el identificador de la estación de referencia (*mountpoint*), es posible utilizar varios Rover contra una estación de referencia. En este aspecto, es de vital importancia darle una identificación única a cada estación referencia, para evitar que un cliente RTK adquiera correcciones de múltiples estaciones simultáneamente, lo cual genera errores en el cálculo de su posición.

4.1.3.5 Control GSM

Este proceso programado en lenguaje C, se encarga de la comunicación por medio de comandos AT con el módulo celular, el cual se encuentra conectado al puerto USB. El mismo, será capaz de recibir y validar configuraciones y/o peticiones enviadas por un usuario a través de mensajes SMS.

El módulo celular, durante la etapa inicial del proceso, fue configurado para notificar al Control GSM ni bien se recibe un mensaje SMS. Una vez recibido un mensaje, el control GSM se comporta de acuerdo con el diagrama de la Ilustración 46 para procesar este. En caso de que no se reciba ningún mensaje, el proceso espera por un tiempo determinado antes de consultar por nuevas SMS notificaciones.

A grandes rasgos, la secuencia ante la recepción de un mensaje involucra, la autenticación del usuario, la verificación de la sintaxis (descrita en el Anexo B) y la respuesta al usuario.

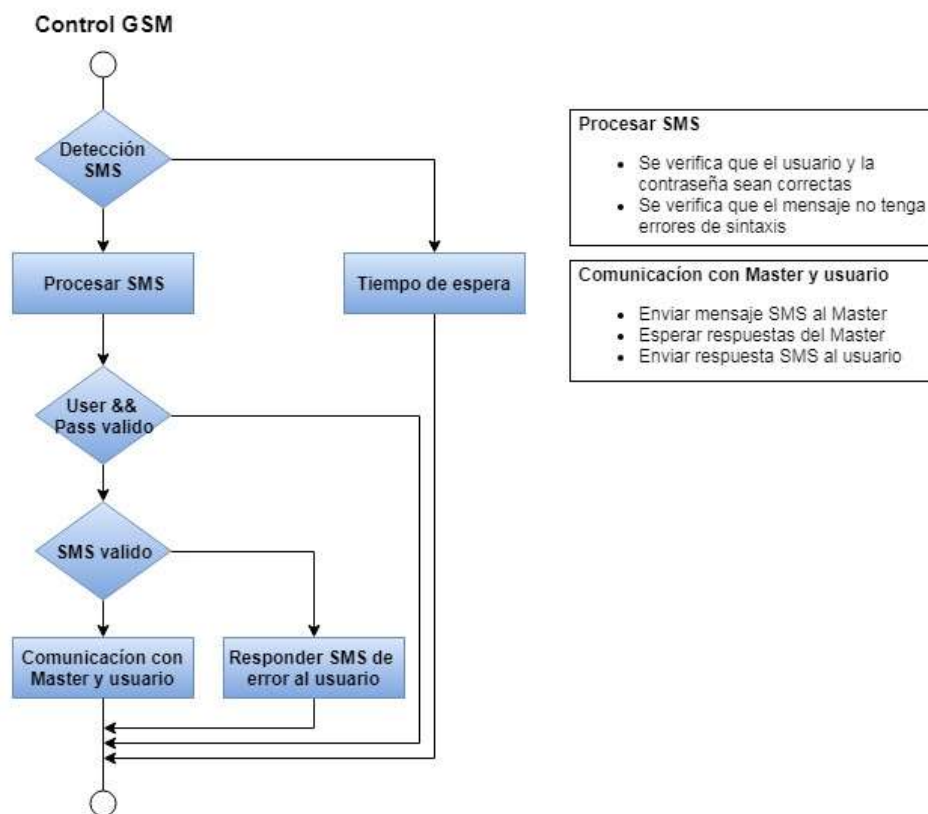


Ilustración 46 - Diagrama de flujo del Control GSM ante la recepción de un mensaje

La autenticación del usuario, donde usuario y contraseña se esperan en las primeras dos líneas del mensaje, se realiza para prevenir que usuarios no habilitados realicen cambios en el funcionamiento del sistema, debido a esto, si el mensaje no proviene de un usuario autorizado, este es descartado de inmediato sin proveer ningún tipo de respuesta. A fin de no procesar mensajes que luego deberán ser descartados, el Control GSM realiza la validación de la sintaxis del mensaje solo si se determina que este proviene de un usuario habilitado. Si la validación del mensaje falla, se notifica al usuario los errores cometidos mediante una respuesta SMS (ver Ilustración 47), descartando el mensaje en su totalidad. Un mensaje recibido se toma como válido, una vez se completa con éxito la etapa de autenticación y la de validación de sintaxis.

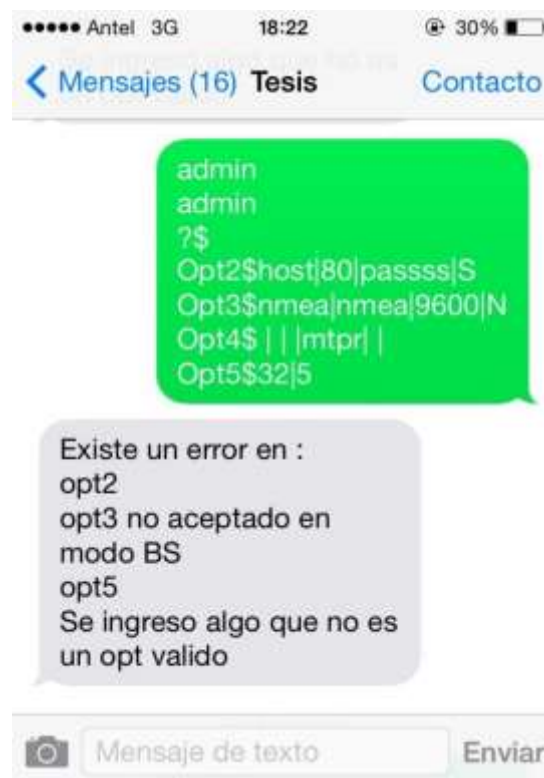


Ilustración 47 - Envío de configuraciones con campos incorrectos

A la recepción de un mensaje válido, el Control GSM se comunica por socket con el Master a fin de transmitir todas las configuraciones y peticiones recibidas. Una vez se envía toda la información al Master, si alguna de estas involucra una petición, se espera la respuesta para luego construir y enviar el SMS al usuario. Debido al costo de los SMS, se optó por no contestar al usuario cuando se realizan únicamente configuraciones correctas. Dado a que los SMS se procesan en orden de escritura, si el usuario desea verificar las configuraciones

realizadas, puede hacerlo en un mismo mensaje, añadiendo al final de este una petición de configuración actual.

A causa de que el sistema tiene dos modos de funcionamiento, el Control GSM será iniciado en modo cliente RTK o estación base según su naturaleza. Cada modo permite configuraciones diferente, por ende, si está en modo cliente RTK deberá rechazar los mensajes destinados a configurar una estación referencia y viceversa.

Además de las responsabilidades ya mencionadas, el Control GSM es el encargado de codificar y decodificar en *Protocol Data Unit* (PDU) los mensajes SMS. Se decidió utilizar este modo ya que el modo texto falla a la hora de decodificar caracteres especiales (ej.: acentuaciones de diversos tipos), al decodificarlo manualmente se facilita el control de dichos caracteres.

A fin de facilitar su implementación, se busca un sistema que realice la conversión de un mensaje en modo texto a modo PDU y viceversa. En este sentido, se encontró una librería de código abierto y desarrollada en C, que nos permite de forma sencilla codificar y decodificar mensajes en PDU.

4.1.3.6 Data Storage

Dado que el sistema puede correr por largos períodos sin la intervención del usuario, se vio la necesidad de limitar la cantidad de datos que se guardarán dentro de un archivo salida, como también el espacio total ocupado por el conjunto de estos, y de esta forma, asegurar que siempre se encuentre espacio disponible en disco para almacenamiento. Con el fin descrito anteriormente se programó en Shell Script el proceso Data Storage.

Este proceso tiene dos funcionalidades básicas, en primera instancia eliminar el archivo de mayor antigüedad siempre que se supere el espacio asignado para el almacenamiento de estos, y por último renombra los archivos que ya fueron cerrados por rtkrcv, indicando que ya pueden ser descargados por el usuario.

Para el control del espacio asignado, se configuró a rtkrcv para que almacene los archivos en un directorio utilizado únicamente para este fin. Esto permite al Data Storage controlar periódicamente el volumen de este directorio y eliminar los archivos de mayor antigüedad en caso de que se requiera.

Para evitar que el usuario descargue archivos incompletos, el sistema no permite la descarga de los archivos que se están generando en la ejecución actual. Para esto, se tomaron varias acciones. Se configuró a rtkrcv para que nombre a los archivos de salida con el encabezado “RW=” añadiendo fecha y hora de creación. Se programó la página de configuración local para que excluya de los archivos descargables a aquellos que contengan el encabezado mencionado. Por último, el proceso Data Storage, utilizando herramientas del sistema Linux, remueve el encabezado de todos los archivos que no están bajo escritura de rtkrcv, posibilitando entonces la descarga de estos.

Si bien el usuario es el responsable de adquirir los archivos antes que sean eliminados por falta de espacio de almacenamiento, el sistema se diseñó, para asegurar un mínimo de 270 horas de almacenamiento para archivos en formato NMEA, variando en caso de utilizarse otro protocolo de salida.

4.1.3.7 Wifi Host Manager

Este proceso programado en Shell Script es el encargado de, encender la Wifi propia de la Raspberry, generar con esta un punto de acceso y levantar los servicios necesarios para que el usuario pueda acceder a la página de configuración local.

Con el fin de ahorrar el consumo de energía del sistema y liberar poder de computo, cuando no se detectan usuarios conectados a la Wifi por un lapso de tiempo, esta se cierra, siendo necesario levantarla manualmente presionando por un tiempo menor a 10 segundo el pulsador de configuración, tal como se explicó en el capítulo 4.1.1.3.

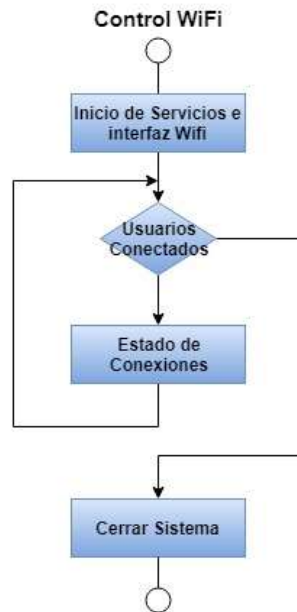


Ilustración 48 - Diagrama de flujo del proceso Wifi Host Manager

Para lograr el funcionamiento óptimo de este proceso se orientó su programación en base al diagrama de la Ilustración 48, donde en primera instancia el programa inicia todos los servicios indispensables para proveer el acceso a la interfaz de configuración local, estos integran, el servicio Apache, los servicios de configuración del punto de acceso y la activación de la interfaz Wifi de la Raspberry Pi.

Una vez iniciado todos los servicios, se registra cada 20 segundos las conexiones activas al punto de acceso. En caso de que no se detecten conexiones por un periodo continuo de 1 minuto, se inicia la culminación del proceso, el cual involucra el cierre de los servicios iniciados y la desactivación de la interfaz Wifi.

4.1.3.8 Traductor serial

Debido a la escasez de interfaces serial nativas en la Raspberry Pi se vio la necesidad de simular una salida serial por software a través de los GPIO de la misma, para realizar esto se utilizó `Soft_uart`.

El módulo `Soft_uart` utiliza una técnica denominada *bit banging*, esta consiste en utilizar software para codificar la información en señales y pulsos a través de los pines GPIO, además

de controlar la sincronización, tiempos, umbrales y protocolos necesarios para lograr la comunicación.

Se decidió usar dicho módulo ya que es de código abierto y además al generar la interfaz simulada, lo hace en el directorio estándar para dispositivos en los sistemas Linux, facilitando su integración con RTKLib.

4.1.3.9 Servicio Apache

Este servicio capaz de responder a las peticiones de contenido Web utilizando el protocolo HTTP, se utilizó con el fin de proveer un interfaz de configuración local amigable para el usuario. Para ello se programó la página de configuración local utilizando el lenguaje de programación Web PHP¹, la cual es capaz de comunicarse con el proceso Master por medio de sockets con el fin de enviar las configuraciones y peticiones.

A fin de lograr la comunicación, la página de configuración debe dialogar con el Master utilizando el protocolo interno por el cual este último se rige. Dado que esta interacción se realiza utilizando un mecanismo de comunicación internos al sistema Linux (servidor socket local), en caso de fallos o pérdida de conexión, se establecieron tiempos reducidos (2 segundos) para que la página de configuración establezca la conexión como inactiva. En estos casos se notifica inmediatamente al usuario mediante mensajes de error, facilitando así la detección y diagnóstico de problemas.

En adición, durante la primera ejecución del sistema, donde se deben establecer las credenciales de este (usuario y contraseña), la página de configuración es la única interfaz capaz de proporcionar un método para el ingreso de estas. Una vez se establecen las credenciales, estas son almacenadas en un archivo para futuros inicios del sistema y enviadas simultáneamente al Master para que este reanude su ejecución.

Es de importancia aclarar que, para acceder a la etapa de configuraciones, el usuario deberá pasar previamente por una etapa en donde se le solicitaran las credenciales de acceso del sistema.

¹ Acrónimo recursivo de PHP Hypertext Preprocessor.

El servicio Apache, siendo esencial para el correcto funcionamiento de la interfaz de configuración local, es controlado por el proceso Wifi Host Manager.

4.1.3.10 Software adicional utilizado

Si bien todas las capas de *software* descritas anteriormente conforman la base del sistema, para el correcto funcionamiento de este, se debieron utilizar librerías y *software* adicionales que permitieron simplificar la comunicación o configuración previa de las piezas de *hardware*.

Dhcpd, Dnsmasq y Hostapd

Estos son un conjunto de *software* que posibilitaron la implementación de un punto de acceso Wifi provisto por la Raspberry Pi, lo cual permite, a cualquier dispositivo externo que posea una interfaz de red Wifi, conectarse por este medio a la Raspberry Pi.

Dhcpd es el *software* encargado de configurar una dirección IP estática en la interfaz de red Wifi de la Raspberry Pi. En la configuración actual del sistema está establecida la dirección 192.168.0.1. Dnsmasq es el encargado de proveer el servicio *Dynamic Host Configuration Protocol* (DHCP), que permite proporcionarle de forma automática una dirección IP a cualquier dispositivo que se conecte al punto de acceso Wifi. Por último, Hostapd es el encargado configurar la interfaz Wifi para que esta actué como punto de acceso.

DongleConf

Debido a que la conexión y desconexión del modem USB puede darse en cualquier momento, en cada evento de conexión, el sistema requiere una gran variedad de configuraciones para que este dispositivo adquiriera completa funcionalidad. Para lograr esto, se programó un Shell Script nombrado DongleConf, trabajando de forma conjunta con las herramientas udev, At y wvdial, que se detallan a continuación.

Para satisfacer la necesidad de que el sistema actúe cuando se realiza la conexión del modem USB, se utilizó udev. Esta herramienta permite ejecutar procesos de corta duración ante la detección de un evento de conexión/desconexión de dispositivos externos en interfaces seriales, dentro de los que se incluye las conexiones USB.

Con el objetivo de obtener una interfaz de conexión a Internet para la adquisición o publicación de correcciones, se utiliza Wvdial. Esta herramienta utiliza un protocolo de nivel de enlace, conocido como *Point to Point Protocol*, que permite crear y ejecutar una red IP a través de un enlace serial, permitiendo al sistema Linux acceder a los recursos de red del modem.

El proceso DongleConf es el encargado de, ejecutar Wvdial, configurar la interfaz generada por este como la interfaz por defecto de salida a internet y crear un archivo puente, que será utilizado para la conexión del proceso Control GSM, redirigiendo la comunicación a la interfaz de configuración del modem.

Dado que Wvdial se ejecuta de forma ininterrumpida y recordando que udev restringe la duración de los procesos, se requirió que DongleConf se independice de la ejecución de udev, para lo cual se utilizó la herramienta At.

ConfigNEO-M8T

Este es un proceso programado en Shell Script que tiene como objetivo realizar las configuraciones para que el receptor NEO-M8T pueda otorgar en su salida la información necesaria para realizar correcciones RTK.

Este proceso se comunica con el receptor únicamente en el inicio del sistema a través de la interfaz UART, utilizando un protocolo propietario de Ublox, especificando la información de salida y el protocolo de esta.

StartRTKReady

Es un proceso programado en Shell Script que se ejecuta automáticamente en el encendido de la Raspberry Pi, este tiene como objetivo ejecutar el proceso de configuración del receptor GNSS (ConfigNEO-M8T), esperar a que se complete la inicialización de las interfaces físicas de la Raspberry (USB, SPI, UART, etc.) e iniciar el proceso Master sin la intervención del usuario.

WiringPi

Esta es una librería desarrollada en lenguaje C para Raspbian² la cual contiene una gran variedad de funciones, dentro de las que se encuentran el control y lectura de pines GPIO, el manejo de diversas interfaces de comunicación (UART, I2C, SPI), y la medición de tiempos en milisegundos.

La misma es utilizada por el proceso Traductor RF con el fin de facilitar, la comunicación SPI y las interrupciones por *hardware* utilizadas en la comunicación entre este y el transceptor LoRa Bee. Adicionalmente es utilizada por el Master, para el manejo de Leds y pulsadores a través de pines GPIO y para el manejo de tiempos utilizado en comunicaciones internas.

² Distribución de Linux utilizada por Raspberry Pi

5 Pruebas

En este capítulo, se expondrá todo el procedimiento de las pruebas realizadas, incluyendo los resultados obtenidos y la comparación con los esperados. Los ensayos realizados consistieron en, pruebas de desempeño general de RTKReady y pruebas de conectividad a la red celular, utilizando el módulo Quectel BG96 bajo el protocolo LTE Cat M1.

Para validar el funcionamiento del sistema de RTKReady, se realizaron pruebas enfocadas en dos aspectos cruciales, las pruebas de precisión, pruebas de alcance del radioenlace, pruebas de conexión de estación de referencia a un NTRIP Caster y pruebas de estabilidad general del sistema.

5.1 Pruebas de precisión

El objetivo de esta prueba es observar la precisión de la posición provista por RTKReady. Con este fin, se registran las posiciones, mientras se desplaza la antena del receptor GNSS sobre una hoja de papel impresa con las siglas RTK. Al graficar estas posiciones se espera obtener boceto de las siglas impresas.

Para el registro de las posiciones, se almaceno un archivo utilizando el cliente RTK en formato NMEA, el cual luego se procesó con la aplicación rtkplot³. La cual permite graficar todas las posiciones en una escala conveniente para visualizar los resultados obtenidos.

Esta prueba se realizó en una zona urbana, específicamente arriba de la azotea de un edificio utilizando las estaciones de la IGM como estación de referencia. Se utilizaron letras de 10 cm y 18 cm de altura, como se observa en la Ilustración 49 e Ilustración 50 respectivamente.

³ Aplicación con interfaz grafica para Windows provista por RTKLib la cual permite graficar las posiciones llegando a escalas de milímetros.



Ilustración 49 - Impresión para pruebas de precisión con letra de 10 cm de altura



Ilustración 50 - Impresión para pruebas de precisión con letras de 18 cm de altura

Se puede observar en la Ilustración 51 e Ilustración 52 que las desviaciones obtenidas están dentro del rango de error de 3 cm promedio para correcciones RTK, adicionando posibles desviaciones introducidas por el operario al desplazar la antena GNSS.

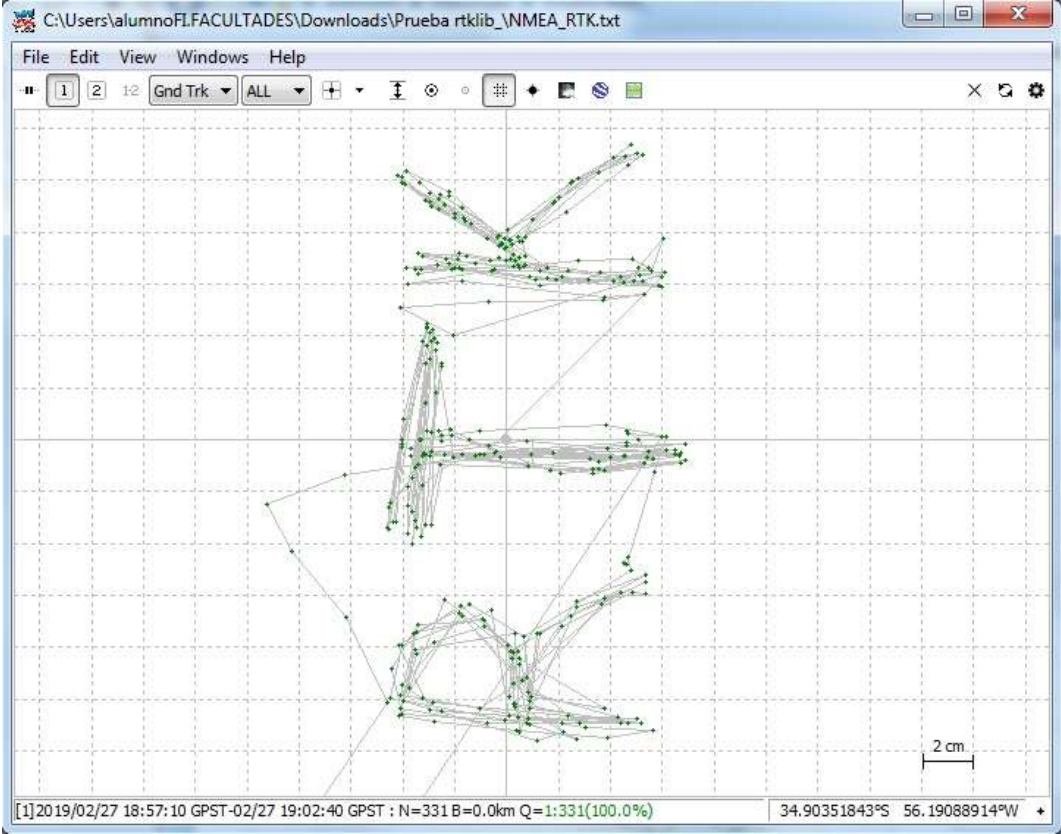


Ilustración 51 - Resultado de prueba de precisión con letras de 10 cm de altura

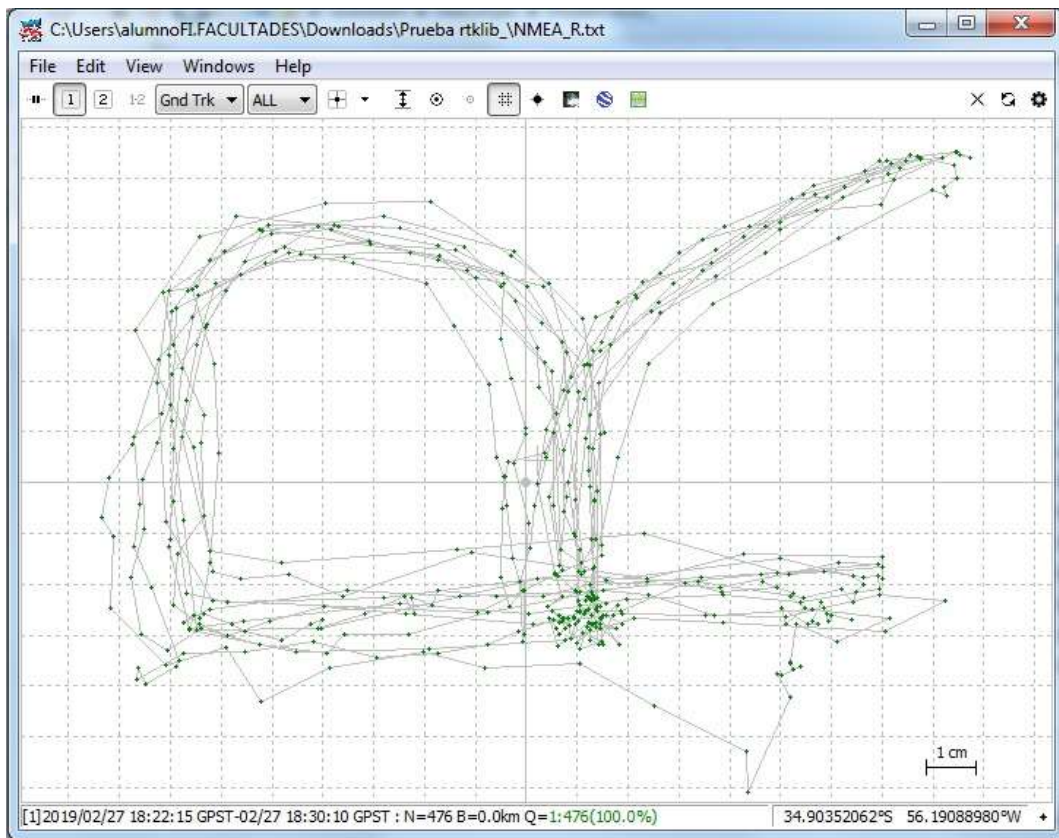


Ilustración 52 - Resultado de prueba de precisión con letras de 18 cm de altura

Durante las pruebas realizadas, tal como se mencionó anteriormente, se observó que el nivel de precisión en el posicionamiento es del orden de los 3 cm, permitiendo concluir que el sistema RTKReady alcanza los requerimientos de precisión de un sistema de correcciones RTK.

5.2 Pruebas de alcance del radio enlace

El objetivo de esta prueba era determinar los límites máximos de separación entre la estación de referencia y el cliente RTK en cual este último aún recibe datos de observables, al utilizar el radio enlace como mecanismo de comunicación entre los dispositivos. Para su realización, se busca un camino libre de obstáculos que interfieran en la línea de visión entre ambos dispositivos tal como se muestra en la Ilustración 53.



Ilustración 53 - Locación para pruebas del alcance del radio enlace

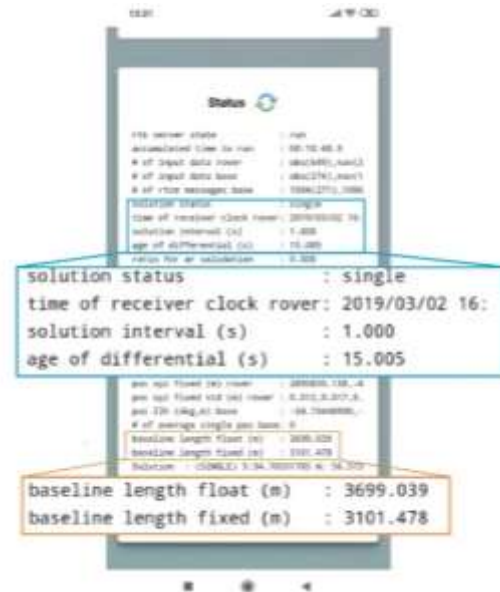


Ilustración 54 - Estatus de cliente RTK en pruebas de alcance del radio enlace

Para verificar el alcance del radioenlace se observaron los mensajes RTCM detectados por el Rover enviados desde la estación de referencia, dichos mensajes pueden ser observados en el estatus de RTKReady. Una vez que se dejaron de recibir mensajes RTCM se consideró que se había llegado al alcance máximo del radioenlace.

El estatus del cliente RTK que se muestra en la Ilustración 54, proporciona la máxima distancia alcanzada en donde todavía existía comunicación entre los dispositivos. Esta se ve representada por el parámetro *baseline length float*, que indica la distancia a la estación de referencia de la última posición corregida con precisión de metros (*float*).

Se consideró exitosa esta prueba, debido a que se logró una distancia máxima de separación de unos 3.8 km acercándose al valor objetivo de 4.0 km.

Es de importancia aclarar que durante esta prueba solo se contempló la recepción continua de mensajes RTCM provenientes desde la estación de referencia, por lo cual, no se esperó la convergencia de la solución de correcciones y no se previó que el área de recorrida estuviera libre de obstáculos que interferían con las señales GNSS.

5.3 Conexión de estación de referencia a NTRIP Caster

El objetivo de esta prueba fue utilizar un NTRIP Caster público con el fin de registrar una estación de referencia. Para la realización de esta prueba se utilizó el servicio de rtk2go, debido que para el proyecto no se adquirió un servidor propio que cumpla esta funcionalidad.

En las ilustraciones se mostrará todas las estaciones de referencia que se encuentran en este Caster, y en particular, la utilizada para la prueba, tomando como nombre identificador “TesisBS01”.



Ilustración 55 - Tabla de NTRIP Servers de rtk2go

Consideramos que la prueba fue exitosa, debido a que se pudo ingresar la estación de referencia en el Caster y solicitar la información proveniente de esta desde un Rover.

5.4 Pruebas de estabilidad

El objetivo de las pruebas de estabilidad era observar el funcionamiento del sistema RTKReady bajo diferentes condiciones y entornos al moverse durante un tiempo prolongado.

En esta prueba se realizó un recorrido por un camino vecinal, utilizando como estación de referencia la estación UYMO de la IGM, que se encontraba a una distancia de 35Km aproximadamente. Se utilizó el cliente RTK para registrar el movimiento de un vehículo a una velocidad promedio de 30 Km/h durante una hora de recorrido. En el comienzo de esta, se mantuvo estática la posición del vehículo por un periodo de tiempo que asegurará la convergencia de las correcciones.

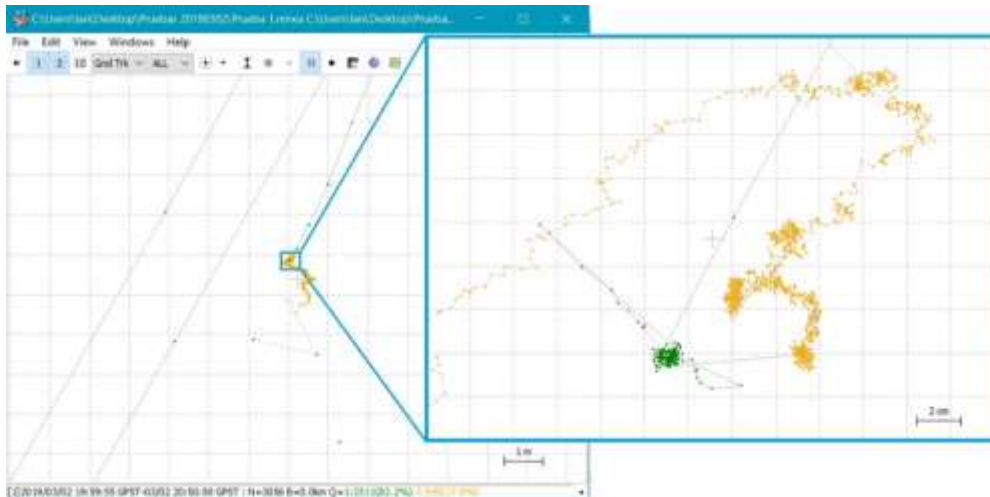


Ilustración 56 - Periodo de inmovilidad al comienzo de la pruebas de estabilidad

En la Ilustración 56 se puede observar el principio de la prueba donde las posiciones indicadas con puntos amarillos representan correcciones parciales (*float*). Estas se mantuvieron por aproximadamente 12 minutos, tiempo a partir del cual las correcciones pasan a ser completas (*fix*), indicándose las posiciones con puntos verdes en la ilustración.

Una vez estabilizadas las correcciones, se comenzó el recorrido con el vehículo tal como se muestra en la Ilustración 57 e Ilustración 58. Para facilitar la visualización de las pruebas, en la Ilustración 57 se utilizó color rojo para indicar posiciones con correcciones completas y amarillo para las parciales.

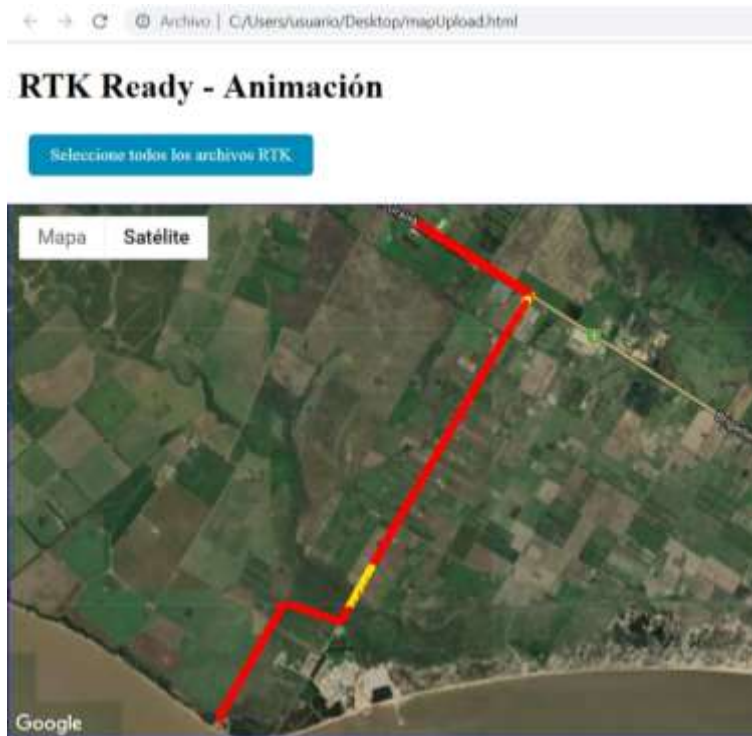


Ilustración 57 - Recorrido de la prueba de estabilidad

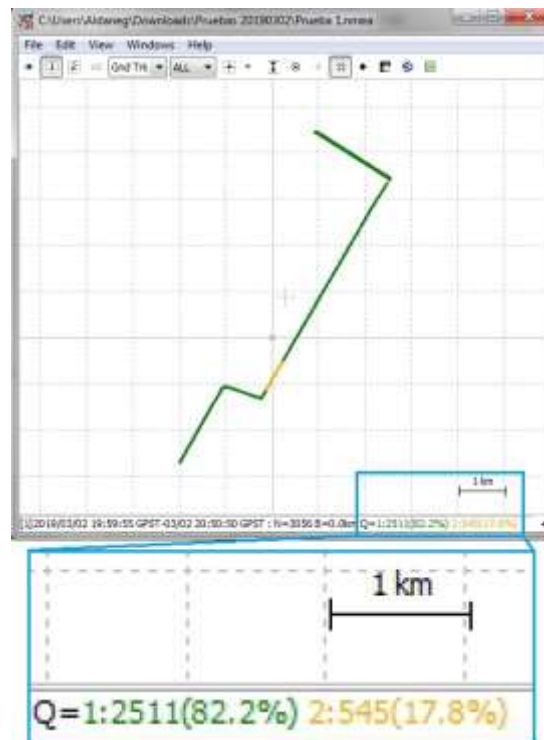


Ilustración 58 - Recorrido de la prueba de estabilidad visualizado en rtkplot

Observando con mayor detalle los parámetros proporcionados por la Ilustración 58, se puede notar que durante el transcurso de toda la prueba se obtuvieron un 82.2% de correcciones

completas y un 17.8% de correcciones parciales. Dentro de estos valores se deben considerar los 12 minutos de convergencia de las correcciones iniciales.

Con esta información presente se determinó que el sistema cumplía con los requerimientos de estabilidad propuestos asegurando un funcionamiento continuo proveyendo correcciones de posición.

Con el fin de apreciar el nivel de precisión de las correcciones, en la Ilustración 59 se puede notar el cambio de carril del vehículo de prueba al aproximarse al retorno.



Ilustración 59 - Precisión del recorrido en la prueba de estabilidad

5.5 Pruebas de conectividad a LTE Cat M1

Dado que, como objetivo secundario, se había propuesto el desarrollo de librerías que posibiliten la comunicación entre la aplicación SmartWay y un módulo celular que implementara NB-IoT. Eran necesarias pruebas de conectividad de esta tecnología que aseguraran el correcto entendimiento de las configuraciones y protocolos a utilizar.

Si bien no fue posible realizar prueba alguna con esta tecnología en particular, en el mes de febrero, la empresa de telecomunicaciones Movistar pone en funcionamiento de manera provisoria, una radio base que implementa LTE Cat M1 en el departamento de Montevideo. Esto permitió realizar pruebas de conexión a una red LPWA que actualmente está en desarrollo a nivel global, la cual ofrece prestaciones similares a las de NB-IoT.

Para la realización de estas pruebas se utilizó una placa de evaluación que integra un módulo Quectel BG96 el cual implementa tecnologías de conexión EGPRS, LTE Cat M1 y NB-IoT. Esta placa de desarrollo fue provista por la empresa internacional Ericsson el cual es un referente mundial en la prestación de insumos para telecomunicaciones.

La prueba realizada consta, de la verificación de las configuraciones del módulo BG96 para posibilitar la conexión a la red LTE Cat M1 a una radio base y utilizar esta, para obtener conectividad a internet.

Debido a inconvenientes en la etapa de configuración del módulo, se solicitó ayuda a un funcionario de Ericsson que nos indicó que, si bien las configuraciones realizadas eran correctas, la secuencia no era la debida.

En la Ilustración 60 se puede observar la secuencia de las configuraciones realizadas mediante comandos AT utilizando el *software* de Quectel, en conjunto con la información de la radio base a la cual se conectó y la dirección IP asignada.

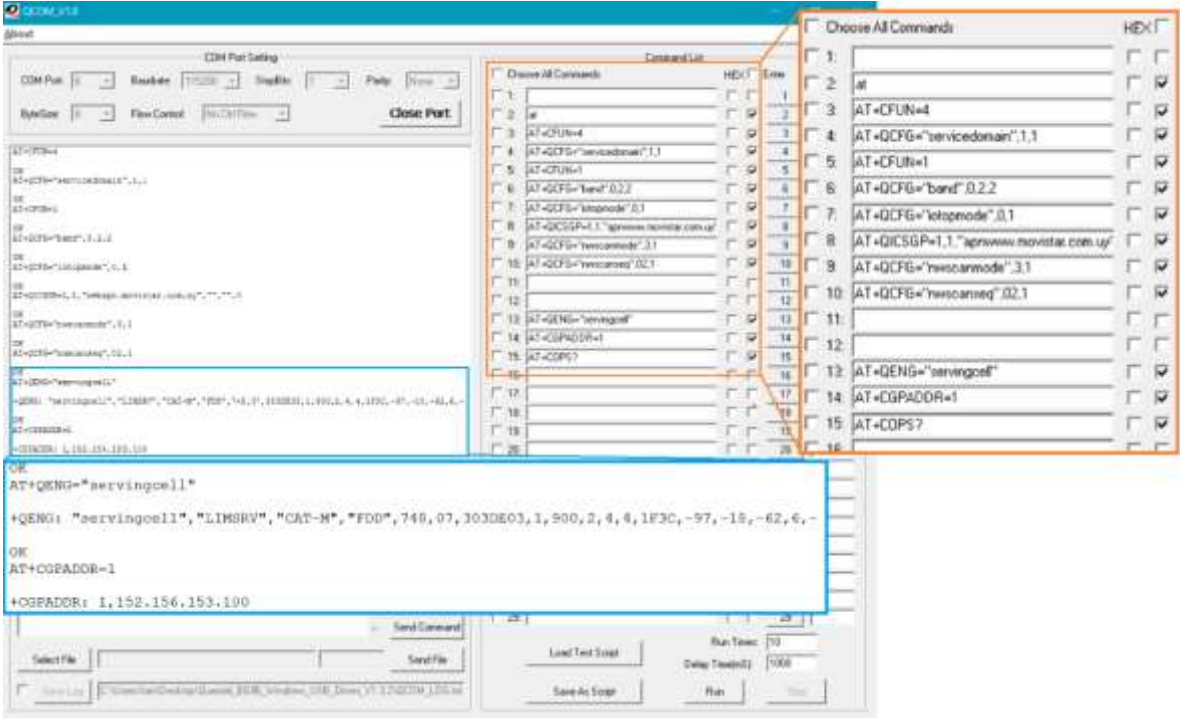


Ilustración 60 - Prueba de conexión a radio base LTE Cat M1

Para verificar la conectividad del módulo, se realizó un ping⁴ a www.google.com y se descargó de dos formas diferentes una página Web. Dicha descarga se realizó utilizando los comandos HTTP propios del módulo y enviado las consultas HTTP a partir de la apertura manual de un socket, siendo esta última un requerimiento de la empresa SmartWay. En la Ilustración 61 se puede observar la prueba de ping a www.google.com y la descarga de la página Web <http://www.example.com> mediante la apertura manual de sockets.

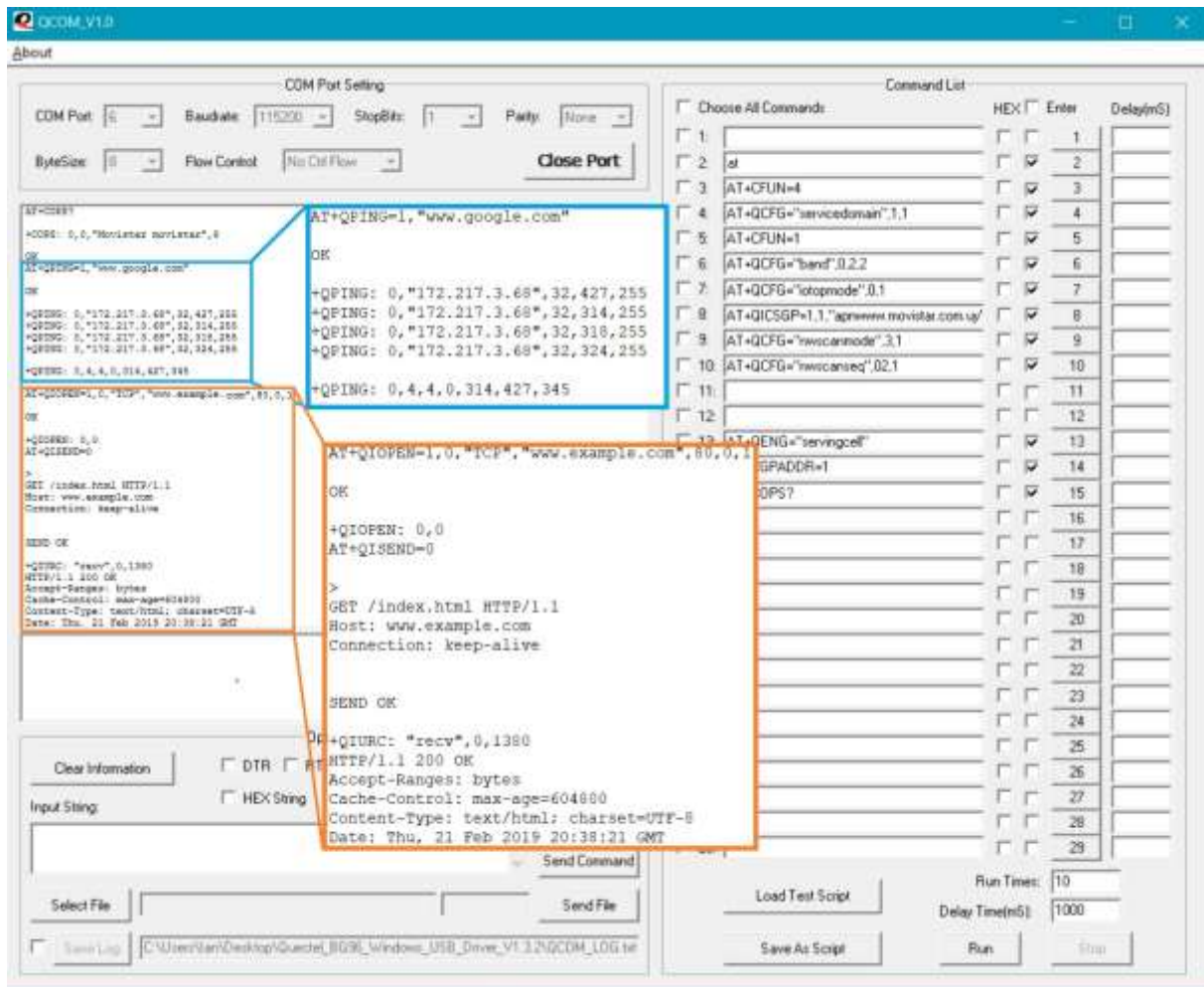


Ilustración 61 Prueba de conectividad a internet utilizando LTE Cat M1

Como resultados de esta prueba, se considera exitosa la configuración del módulo Quectel BG96 para lograr acceso a internet, utilizando la red celular bajo el protocolo LTE Cat M1.

⁴ Ping es una utilidad de diagnóstico que permite comprobar el estado de la conexión de red entre dos dispositivos o computadoras.

Sin bien, debido a problemas de tiempos, no fue posible la implementación de librerías para el control de este módulo, se documentó todo los procedimientos y configuraciones necesarios para la implementación de estas.

6 Conclusiones

A modo de culminación del proyecto, en este capítulo se detallarán los métodos y forma de trabajo a lo largo del proyecto, los costos de implementación del sistema y una breve reflexión de los conocimientos adquiridos.

6.1 Forma de trabajo

La forma de trabajo a lo largo del proyecto fue cambiando, adaptándose a la etapa en la que este se encontraba, separando tareas para agilizar la programación, trabajando en conjunto para solucionar problemas puntuales o intercambiando tareas para recibir *feedback*.

Durante todo el proyecto se realizaron reuniones periódicas con el cliente y tutor, con el fin de garantizar que se cumplan las necesidades del cliente y proteger los intereses de los integrantes del proyecto, en dichas reuniones se discutieron avances, modificaciones y plazos a seguir. Para cada reunión se realizó una bitácora de los puntos hablados y las modificaciones acordadas.

Al comienzo del proyecto en la etapa de investigación, ya que ningún integrante tenía conocimientos previos sobre sistemas GNSS se decidió trabajar en conjunto, adquiriendo los conceptos necesarios para adquirir una comprensión global sobre estos sistemas, posibilitando el diseño de RTKReady.

Durante la etapa de diseño se listaron todas las obligaciones del sistema, para luego agruparlas y determinar qué proceso será responsable de ejecutarlas. De esta forma se fueron definiendo los mismos para satisfacer con todas las obligaciones. Una vez definidos todos los procesos con sus respectivas obligaciones se procede a diagramar todas las comunicaciones entre estos y las piezas de hardware.

Durante la etapa de diseño se listaron y agruparon todas las obligaciones del sistema, de esta forma se fueron definiendo los procesos necesarios para satisfacer con todas las obligaciones. Una vez definidos todos los procesos con sus respectivas obligaciones, se diagramaron las comunicaciones entre estos y las piezas de hardware, permitiendo iniciar con la etapa de implementación.

Al comenzar con la etapa de implementación, se vio la necesidad de utilizar una herramienta que facilitara la gestión del proyecto, para esto se utilizó Trello.

Trello es una herramienta en la nube para la organización y gestión de proyectos, la cual permite crear listas de tareas a realizar indicando fecha de inicio y finalización, notificando en caso de retrasos en las culminaciones de estas.

La utilización de esta herramienta permitió visualizar el proyecto como un conjunto de etapas, simplificando la división de tareas entre los integrantes del proyecto y controlar los plazos de finalización de dichas etapas. Al gestionar el proyecto con Trello se logró un método de trabajo dinámico, eficiente e integral, posibilitando independencia de los integrantes al realizar las tareas.

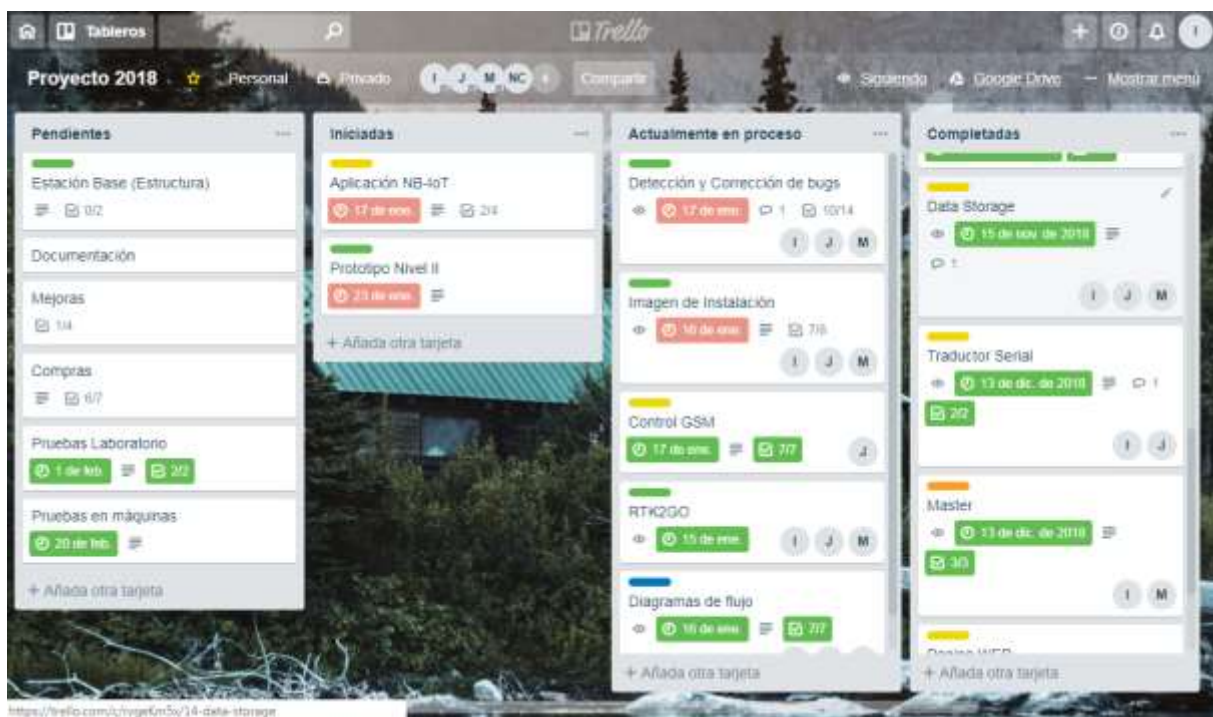


Ilustración 62 - Gestión del proyecto con Trello

Por último, las etapas de prueba se realizaron en conjunto, de manera que ante algún inconveniente este pueda ser diagnosticado y/o solucionado rápidamente. De forma similar se realizó la documentación, con excepción que esta estuvo presente a lo largo de todo el proyecto, realizando informes periódicos con las decisiones y actividades realizadas.

6.2 Precio final

Durante el proyecto se realizaron tres compras al exterior para la adquisición de equipamientos. En primera instancia, se adquirieron dos módulos GNSS necesarios para las pruebas iniciales del sistema de correcciones. En segunda instancia, se adquieren dos Raspberry Pi Zero W y dos módulos LoRa Bee obteniendo todas las partes esenciales para la implementación. Por último, se adquirieron las antenas RF, conversores RS232 y algunos componentes necesarios para la finalización del sistema.

Una limitante a la hora de la elección de *hardware* para el diseño del sistema fue atenerse a un presupuesto, por lo tanto, algunos dispositivos pueden ser cambiados por otros que presentan mejores prestaciones como es el caso de las antenas GNSS o las antenas RF.

En este punto, creemos que realizamos una buena tarea en la administración del recurso económico eligiendo de forma eficiente los módulos adquiridos y llegando de forma justa al presupuesto brindado.

En la Tabla 4 se muestra un resumen de los artículos comprados, mostrando: el precio de la unidad, cantidades compradas y proveedor del cual fueron adquiridos.

Artículo	Precio unidad	Cantidad	Proveedor
UBLOX NEO-M8T TIME & RAW receiver board with SMA	U\$S 74.99	2	CSG Shop
Raspberry Pi Zero W + 16 GB SD	U\$S 19.95	2	PiShop
LoRa Bee 915	U\$S 27.00	2	RobotShop
Modem 3G USB Huawei ^(*1)	~ U\$S 22.00	2	Amazon
Antenas GNSS ^(*2)	~ U\$S 50.00	2	Taoglas
Antena RF Delta 6A, Siretta	U\$S 16.68	1	DigiKey
Antena RF ANT-GSMSTUB4, RF Solutions	U\$S 13.77	1	DigiKey
Conversor RS-232	U\$S 10.10	1	DigiKey
StepDown 5V 2A ^(*2)	~ U\$S 15.00	2	

Caja plástica	U\$S 7.10	2	DigiKey
Componentes varios	~ U\$S 38.00		
~ Costo aproximado. (*1) Previamente adquirido por los integrantes del equipo. (*2) Previamente adquirido por SmartWay.			

Tabla 4 - Resumen de costos

Teniendo en consideración el costo de cada componente, se tiene que la inversión de *hardware* requerida para la construcción de cada dispositivo, cliente RTK o estación de referencia, es de unos U\$S 262. Dado que los componentes fueron adquiridos en reducidas cantidades, el costo de los mismo se regía por precios unitarios, sin embargo, estos se reducirían notoriamente ante la compra por volumen. A su vez, los usuarios que no estén interesados en adquirir correcciones por medio de internet ni configurar el sistema de forma remota, pueden excluir la adquisición del módem USB, reduciendo el precio final del sistema.

Como factor final, se deben considerar gastos adicionales como, costos de envíos y costos de trámites administrativos como homologaciones, tramites de importación, etc.

Se debe considerar el costo relacionado a la cantidad de horas de trabajo. En cuanto a la etapa de desarrollo de *software* e integración de las piezas *hardware*, se estima que, entre los tres integrantes del proyecto, se dedicaron un total de 2160 horas de trabajo.

6.3 Aprendizaje

Durante todo el proyecto se tuvo la constante tarea de incorporar nuevos conceptos e ideas que permitieron evolucionar de manera constante y acertada el desarrollo de este. En primera instancia se tuvo que comprender la necesidad de tener una alta precisión en el posicionamiento GNSS en el área de la agricultura de precisión, ya sea con el fin de reducir costos en la aplicación de insumos, garantizar una producción estable o disminuir el impacto ambiental producido por los agroquímicos.

Debido a la cantidad de tecnologías utilizadas para la implementación del sistema propuesto, se adquirieron conocimientos en diferentes áreas técnicas.

De forma teórica, se comprendió el funcionamiento de la red GNSS abarcando el sistema espacial (constelaciones), estructura terrestre, el método de posicionamiento con sus posibles

causas de error en la obtención de posiciones, y los mecanismos o sistemas que permiten mejorar su rendimiento. Además, se logró un amplio conocimiento de todos los protocolos utilizados para posicionamiento basado en sistemas GNSS y sus correcciones.

En relación con lo anterior y como punto inicial, fue necesario adquirir conocimiento relacionado al funcionamiento de un receptor GNSS Ublox NEO-M8T, en particular los protocolos utilizados por este. Adicionalmente, se logró comprender el funcionamiento de un sistema RTK, entendiendo el papel que cumple cada parte, abarcando la estación de referencia, el cliente RTK y los protocolos utilizados para el intercambio de información. Como factor de gran relevancia, se obtuvieron conocimientos sobre los procedimientos necesarios para la obtención de correcciones RTK, parámetros de importancia que influyen en las mismas y la configuración de las variables del sistema que afectan el rendimiento.

Para la realización de correcciones RTK utilizando internet como mecanismo de comunicación, fue necesario incorporar conocimientos sobre la administración, funcionamiento e interacción de las diferentes partes de los sistemas NTRIP, esto incluye conocimientos sobre la creación de usuarios y permisos que actúan en los servidores NTRIP Caster, para lograr administrar de manera correcta las conexiones de los NTRIP Clients y los NTRIP Servers.

De acuerdo con las funcionalidades planteadas, fue necesaria la investigación de la distribución Linux propia de la Raspberry, en donde se debió configurar el sistema RTKReady para iniciar en el arranque, manipular los eventos en los dispositivos seriales, y configurar una interfaz de red como un punto de acceso para otros dispositivos. También se vio la necesidad adquirir conocimientos para la comunicación y sincronización entre los procesos del sistema.

La cantidad de dispositivos utilizados y la escasez de interfaces seriales de comunicación de la Raspberry determinó el aprendizaje de distintos protocolos de comunicación, como SPI, I2C y la utilización de los pines de *Input/Output* con este fin. Para cada dispositivo con el que se comunicó, se tuvo que adquirir el conocimiento del protocolo propietario necesario para realizar las configuraciones deseadas.

En cuanto los módulos celulares, se comprendió la configuración del módulo 3G/GSM a partir de comandos AT para establecer la conexión a la red celular. Esto facilitó el

entendimiento de las configuraciones solicitadas por las librerías y software utilizados para el control de los mismos. En adición, se comprendieron en su totalidad las configuraciones necesarias para conectarse a una red LTE Cat M1, adquiriendo conocimiento sobre IoT y las tecnologías de vanguardia que esto impulsa.

Desde el punto de vista del desarrollo Web, se aprendieron tecnologías para el diseño de una página forma sencilla, destacándose la instalación y configuración del servidor Apache, y el uso de lenguajes como HTML⁵, JavaScript, CSS⁶ y PHP. A su vez se aprendió y utilizó la técnica *media query* para el diseño *responsive*, es decir, adaptativo a los distintos dispositivos celulares, tabletas y equipos de escritorio.

Consideramos también como punto de suma importancia en el aprendizaje, la posibilidad de interactuar con un cliente, en este caso con la empresa SmartWay para la cual se realiza el proyecto. En este contexto, fue de suma importancia discutir cada paso del proyecto con el fin de satisfacer sus necesidades, sin descuidar los intereses de ambas partes.

⁵ Acrónimo para *HyperText Markup Language*

⁶ Acrónimo para *Cascading Style Sheets*

7 Mejoras

En esta sección expondremos posibles mejoras del sistema que ya sea por tiempo o presupuesto no fueron implementadas en la solución pero que pueden incrementar el valor de este.

Parte de las funcionalidades que pueden ser agregadas al sistema, requieren únicamente de una ampliación de la capa de software implementado. Para facilitar el uso, seguimiento y administración del sistema, se deberán modificar los siguientes aspectos:

- Reemplazar el método tedioso de configuración remoto a través de mensajes SMS, por un método más sencillo para el usuario en el cual todos los Clientes RTK y estaciones de referencia se conectan a un servidor central, posibilitando la configuración de forma remota por este medio. A fin de facilitar la labor de administración, integrar la posibilidad de acceder a los archivos con el registro de eventos del sistema.
- Permitir la creación de usuarios con niveles jerárquicos, discriminado entre la posibilidad de configurar o únicamente visualizar el estado del sistema.
- Facilitar la administración y modificación de contraseñas sin la necesidad de intervención de un administrador.
- Descargar la tabla de *mountpoints* disponibles cuando un usuario se conecta a un NTRIP Caster. De este modo no se depende de la obtención previa de esta a partir de terceros (Manuales, descarga Web, etc.).

Ya que la estación de referencia debe funcionar de forma autónoma por largos períodos de tiempo, sería conveniente utilizar energías renovables para su alimentación.

Como no se puede garantizar que dos o más estaciones de referencia estén en funcionamiento próximas entre sí, sería de gran conveniencia para la integridad de la información, que cada dispositivo verifique la inactividad del canal de radio frecuencia antes de proceder con la transmisión de datos.

Implementar en RTKLib técnicas de Redes RTK para posibilitar la conmutación entre estaciones de referencia privadas, esto facilita el uso del sistema, aumenta área de cobertura y el nivel de las de correcciones.

A fin de mejorar el rendimiento general del sistema algunos dispositivos deberían ser reemplazados por otros que presentan mejores prestaciones. En el caso de las antenas GNSS, la adquisición de antenas multi-constelaciones y multi-frecuencias mejoraría el grado de precisión de la solución y el tiempo de convergencia. En el caso de las antenas RF, antenas con mayor ganancia proveen una mayor área de cobertura. Y por último a fin mejorar eficiencia energética del sistema y ampliar la cobertura a la red celular, sería conveniente reemplazar los modem USB por módulos celulares que utilicen tecnologías de red LPWA.

8 Glosario

GMO: Organismos genéticamente modificados (*Genetically Modified Organism*), hace referencia a cualquier organismo cuya genética ha sido modificada a través de técnicas de ingeniería genética.

HYV: Variedad de alto rendimiento (*High Yielding Varieties*), son aquellas variedades de una siembra que tiene un rendimiento óptimo en alguno de sus aspectos, por ejemplo: resistencia a enfermedades, maduración a temprana edad, entre otras posibilidades.

GNSS: Sistema global de navegación por satélite (*Global Navigation Satellite System*), es un sistema que mediante el envío de señales satelitales provee a los usuarios servicios de posición, navegación y tiempo.

GPS: Sistema de posicionamiento global (*Global Positioning System*), es un sistema de carácter GNSS desarrollada, mantenida y operada por la Fuerza Aérea de los Estados Unidos.

GLONASS: Acrónimo para *GLO*bal *NA*avigation *Sat*ellite *Sy*stem traducido del ruso, es un sistema GNSS desarrollado por la Unión Soviética, siendo hoy administrado por la Federación Rusa. Constituye el homólogo del GPS estadounidense.

LPWAN: Tipo de red inalámbrica diseñada para permitir comunicaciones de largo alcance con baja velocidad y potencia (*Low Power Wide Area Network*). Se presenta como una opción válida de protocolo de transporte para ciertas necesidades de IoT.

NB-IoT: *Narrow Band – Internet of Things*, es la primera tecnología del tipo LPWAN implementada para brindar conectividad de Internet a objetos cotidianos, los cuales requieren un tráfico de datos no muy alto.

LoRa: *Long Range*, es una tecnología de modulación en radio frecuencia patentada para aplicaciones LPWAN, permitiendo conexiones a largas distancias utilizando bandas de frecuencia que no requieren licencia (ISM).

RTCM: Comisión técnica de radio para los servicios marítimos (*Radio Technical Commission for Maritime Services*), es una organización que provee estándares para los

sistemas de radionavegación marítima y de radiocomunicaciones. En particular, el comité especial 104 (SC-104) es el encargado de generar un estándar para las correcciones de GNSS.

NTRIP: Protocolo estándar utilizado para el transporte de mensajes RTCM vía Internet (*Transport of RTCM via Internet Protocol*).

RTK: Navegación cinética satelital en tiempo real (*Real Time Kinematic*), es una técnica de aumento GNSS en tiempo real, que genera correcciones a partir del uso de medidas de fase de la portadora y una estación de referencia.

MountPoint: Nombre identificador de cada estación de referencia.

UART: *Universal Asynchronous Receiver-Transmitter*, es un dispositivo utilizado para la comunicación serial asíncrona, en el cual, el manejo de señales eléctricas se realiza mediante un driver externo al dispositivo.

SPI: *Serial Peripheral Interface*, es un estándar de comunicación síncrono, utilizado principalmente para la transferencia de información entre dispositivos electrónicos.

I2C: *Inter-Integrated Circuit*, es un estándar de comunicación síncrono, utilizado principalmente para la transferencia de información entre circuitos integrados.

GPIO: Entrada/salida de propósito general (*General Purpose Input/Output*), es un conjunto de pines o conexiones de entrada/salida cuyo comportamiento puede ser controlado para cumplir una amplia gama de propósitos.

Unix: Familias de sistema operativo multi usuario, multiproceso y multitarea.

Linux: Familia de sistemas operativos basados en Unix.

Socket: Mecanismo que permite el intercambio de datos entre procesos que pueden estar o no en una misma computadora.

Shell script: Es un lenguaje de programación basado en comandos que se ejecutan sobre una terminal de Unix.

UHF: *Ultra High Frequency*, es la banda del espectro que ocupa el rango de frecuencias entre 300 MHz a 3 GHz.

VHF: *Very High Frequency*, es la banda del espectro que ocupa el rango de frecuencias entre 30 MHz a 300 MHz.

9 Referencias bibliográficas

- [1] E. Observador, «Agroquímicos: un mal necesario,» Diciembre 2016. [En línea]. Available: <https://www.elobservador.com.uy/nota/agroquimicos-un-mal-necesario-20161210500>.
- [2] I. T. d. C. y. Leon, «Agricultura de precisión,» [En línea]. Available: <http://www.itacyl.es/documents/20143/0/TIC+Agricultura+de+precision.pdf/9719bb38-90c7-4470-c50d-71ea15e434e6>.
- [3] Ublox, «u-blox F9: Taking GNSS precision to the next level,» [En línea]. Available: <https://www.u-blox.com/en/publication/beyond-stories/u-blox-f9-taking-gnss-precision-next-level>.
- [4] I. Magellan Systems Japan, « High Precision GNSS RTK Solution,» [En línea]. Available: <http://www.magellan.jp/english/item/index1.html>.
- [5] NovAtel, «An Introduction to GNSS,» [En línea]. Available: <https://www.novatel.com/an-introduction-to-gnss/chapter-1-gnss-overview/section-1/>.
- [6] J. J. Z. a. M. H.-P. J. Sanz Subirana, «GNSS Signals,» 2011. [En línea]. Available: https://gssc.esa.int/navipedia/index.php/GNSS_signal.
- [7] U. F. Plazaola, «Técnicas de resolución de la ambigüedad de las medidas de fase de sistemas de navegación por satélite,» [En línea]. Available: <https://riuma.uma.es/xmlui/bitstream/handle/10630/4632/UFp.pdf?sequence=1>.
- [8] Quora, «What are code and carrier phase measurement in GPS?,» Septiembre 2015. [En línea]. Available: <https://www.quora.com/What-are-code-and-carrier-phase-measurement-in-GPS>.
- [9] NovAtel, «NovAtel CORRECT,» [En línea]. Available:

- <https://www.novatel.com/assets/Documents/Papers/NovAtelCORRECT-Brochure.pdf>.
- [10] Lefebure, «NTRIP Component Names,» Junio 2011. [En línea]. Available: <http://lefebure.com/articles/ntrip-names/>.
- [11] rtklibexplorer, «Exploring precision GPS/GNSS with RTKLIB open source software and low-cost u-blox GNSS receivers,» [En línea]. Available: <https://rtklibexplorer.wordpress.com/>.
- [12] RaspberryPi, «Setting up a Raspberry Pi as an access point in a standalone network (NAT),» [En línea]. Available: <https://www.raspberrypi.org/documentation/configuration/wireless/access-point.md>.
- [13] RaspberryPi, «How to disable automatic wifi connection on wlan0 (Stretch)?,» [En línea]. Available: <https://raspberrypi.stackexchange.com/questions/82130/how-to-disable-automatic-wifi-connection-on-wlan0-stretch>.
- [14] W. Pi, «GPIO Interface library for the Raspberry Pi,» [En línea]. Available: <http://wiringpi.com/>.
- [15] StackOverflow, «How to read/write to Linux pseudoterminals using separate processes but without forking?,» [En línea]. Available: <https://stackoverflow.com/questions/23459520/how-to-read-write-to-linux-pseudoterminals-using-separate-processes-but-without>.
- [16] M. E. R. B. Tamara Giménez Rodríguez, «SISTEMA DE POSICIONAMIENTO GLOBAL,» 2009. [En línea]. Available: https://webs.um.es/bussons/GPSresumen_TamaraElena.pdf.
- [17] J. J. Z. J. S. S. M. Hernández-Pajares, «Procesado de Datos GPS: código y fase Algoritmos, Técnicas y Recetas,» Febrero 2008. [En línea]. Available: http://gage.upc.edu/sites/default/files/TEACHING_MATERIAL/GPS_BOOK/CASTELLANO/PDGPS/LIBRO_PDGPS_gAGE_NAV_08.pdf.

- [18] D. A. G. Álvarez, «Sistema GNSS,» Enero 2008. [En línea]. Available: <http://arantxa.ii.uam.es/~jms/pfcsteleco/lecturas/20080125DavidGarcia.pdf>.
- [19] SEDICI, «El Sistema de Posicionamiento Global,» [En línea]. Available: http://sedici.unlp.edu.ar/bitstream/handle/10915/1998/2_-_El_Sistema_de_Posicionamiento_Global.pdf?sequence=6.
- [20] U. N. O. f. O. S. Affairs, «Sistemas mundiales de navegación por satélite,» 2013. [En línea]. Available: http://www.unoosa.org/pdf/icg/2013/Ed_GNSS_S_ebook.pdf.
- [21] E. Gómez Treviño, «Trilateración: Sismos, GPS, rayos y teléfonos celulares, y la XIX Olimpiada de Ciencias de la Tierra,» 2014. [En línea]. Available: <https://www.ugm.org.mx/publicaciones/geos/pdf/geos14-2/trilateracion-34-2.pdf>.
- [22] O. U. P. d. Madrid, «PRÁCTICAS GPS,» [En línea]. Available: http://ocw.upm.es/ingenieria-cartografica-geodesica-y-fotogrametria/topografia-cartografia-y-geodesia/contenidos/PRACTICAS/GPS/practicas_gps.pdf.
- [23] VITRIKO, «No me llames Dolores llámame LoRa,» [En línea]. Available: <https://vitriko.eu/lora/>.
- [24] D. Cerebrum, «Bit banging,» [En línea]. Available: <https://digitalcerebrum.wordpress.com/random-tech-info/bit-banging/>.
- [25] Navipedia, «Ground-Based Augmentation System (GBAS),» 2011. [En línea]. Available: [https://gssc.esa.int/navipedia/index.php/Ground-Based_Augmentation_System_\(GBAS\)](https://gssc.esa.int/navipedia/index.php/Ground-Based_Augmentation_System_(GBAS)).
- [26] J. M. R. Medina, «Implementación del Differential Global Positioning System con correcciones simples y dobles en pseudodistancias,» 2014. [En línea]. Available: <http://bibing.us.es/proyectos/abreproy/60237/fichero/Proyecto+fin+de+carrera+-+Jos%C3%A9+Mar%C3%ADa+Rivera+Medina+%2817-09-14%29.pdf>.
- [27] Navipedia, «Wide Area RTK (WARTK),» 2011. [En línea]. Available:

- [https://gssc.esa.int/navipedia/index.php/Wide_Area_RTK_\(WARTK\)](https://gssc.esa.int/navipedia/index.php/Wide_Area_RTK_(WARTK)).
- [28] Navipedia, «WARTK Standards,» 2011. [En línea]. Available: https://gssc.esa.int/navipedia/index.php/WARTK_Standards.
- [29] Navipedia, «RTK Fundamentals,» 2011. [En línea]. Available: https://gssc.esa.int/navipedia/index.php/RTK_Fundamentals.
- [30] Snip, «NTIRP CASTERS MADE EASY,» [En línea]. Available: <https://www.use-snip.com/>.
- [31] I. G. Militar, «Estaciones,» [En línea]. Available: <http://www.igm.gub.uy/geoportal/estaciones/>.
- [32] Agroicultura, «LA AGRICULTURA EN LA HISTORIA DE LA HUMANIDAD,» Octubre 2018. [En línea]. Available: <https://agroicultura.com/general/la-agricultura-en-la-historia-de-la-humanidad/>.
- [33] T. F. Club, «How to setup a USB 3G Modem on Raspberry PI using usb_modeswitch and wvdial,» Noviembre 2013. [En línea]. Available: <https://www.thefanclub.co.za/how-to/how-setup-usb-3g-modem-raspberry-pi-using-usbmodeswitch-and-wvdial>.
- [34] Hackaday, «WHAT COULD GO WRONG? I2C EDITION,» Julio 2016. [En línea]. Available: <https://hackaday.com/2016/07/19/what-could-go-wrong-i2c-edition/>.
- [35] Lefebure, «NMEA 0183 Sentences,» Diciembre 2013. [En línea]. Available: <http://lefebure.com/articles/nmea/>.
- [36] Lefebure, «NMEA GGA Sentences,» Diciembre 2013. [En línea]. Available: <http://lefebure.com/articles/nmea-gga/>.
- [37] A. Mechatronic, «Raspberry Pi I2C clock-stretching bug,» Agosto 2013. [En línea]. Available: <http://www.advamation.com/knowhow/raspberrypi/rpi-i2c-bug.html>.

- [38] Qampo, «Evolución de la agricultura,» Agosto 2017. [En línea]. Available: <https://qampo.es/blog/evolucion-la-agricultura/>.
- [39] E. Solera, «Modulación LoRa: Long Range Modulation,» Agosto 2018. [En línea]. Available: <https://medium.com/pruebas-de-laboratorio-de-la-modulaci%C3%B3n-lora/modulaci%C3%B3n-lora-4ad74cabd59e>.
- [40] A. Upale, «LTE Cat M1 vs NB-IoT vs LoRa - Comparing LPWANs,» Septiembre 2018. [En línea]. Available: <https://www.semiconductorstore.com/blog/2018/LTE-Cat-M1-vs-NB-IoT-vs-LoRa-Comparing-LPWANs-Symmetry-Blog/3496/>.
- [41] Wikipedia, «GBAS,» Noviembre 2015. [En línea]. Available: <https://es.wikipedia.org/wiki/GBAS>.
- [42] Wikipedia, «Satellite_navigation,» Marzo 2019. [En línea]. Available: https://en.wikipedia.org/wiki/Satellite_navigation.
- [43] Wikipedia, «SBAS,» Diciembre 2017. [En línea]. Available: <https://es.wikipedia.org/wiki/SBAS>.
- [44] Wikipedia, «Sistema global de navegación por satélite,» Octubre 2018. [En línea]. Available: https://es.wikipedia.org/wiki/Sistema_global_de_navegaci%C3%B3n_por_sat%C3%A9lite.
- [45] Aristasur, «CÓMO FUNCIONA EL SISTEMA DE POSICIONAMIENTO GPS,» Septiembre 2016. [En línea]. Available: <https://www.aristasur.com/contenido/como-funciona-el-sistema-de-posicionamiento-gps>.
- [46] F. A. A. B. Borja Martínez, «Exploring the Performance Boundaries of NB-IoT,» Febrero 2019. [En línea]. Available: <https://arxiv.org/pdf/1810.00847.pdf>.

A. Evolución del sistema

Desde el comienzo de la etapa de investigación, el modelo global del sistema sufrió modificaciones frecuentes.

La idea inicial era adaptar el *hardware* y *software* utilizado por aplicación de SmartWay con el fin de realizar correcciones RTK. Debido a que el receptor GNSS que esta utiliza no ofrece la información necesaria para implementar RTK, se planteó integrar el mínimo de hardware y software en la aplicación a fin de posibilitar la realización de correcciones.

Como primera solución a este inconveniente, se propuso utilizar módulos GNSS que ya implementaran RTK, pero dado su alto costo esta idea fue remplazada rápidamente, con la de utilizar receptores GNSS de bajo costo en conjunto con librerías de código abierto.

Una vez se decidió utilizar RTKLib como software de correcciones surgió como modelo inicial de sistema el representado por la Ilustración 63, donde el cálculo de las correcciones se da en un servidor y la obtención de datos se realiza con el mínimo hardware requerido.

En este escenario, la recolección de datos en los clientes RTK y las estaciones de referencia se realiza con un módulo GNSS de bajo costo, un módulo NB-IoT como puerta de conexión IP y un microcontrolador para comunicar y controlar ambos módulos.

En este caso, como el cálculo de las correcciones RTK es externo al cliente RTK y a las estaciones de referencia, la interconexión de estos a través de un radioenlace queda obsoleta.

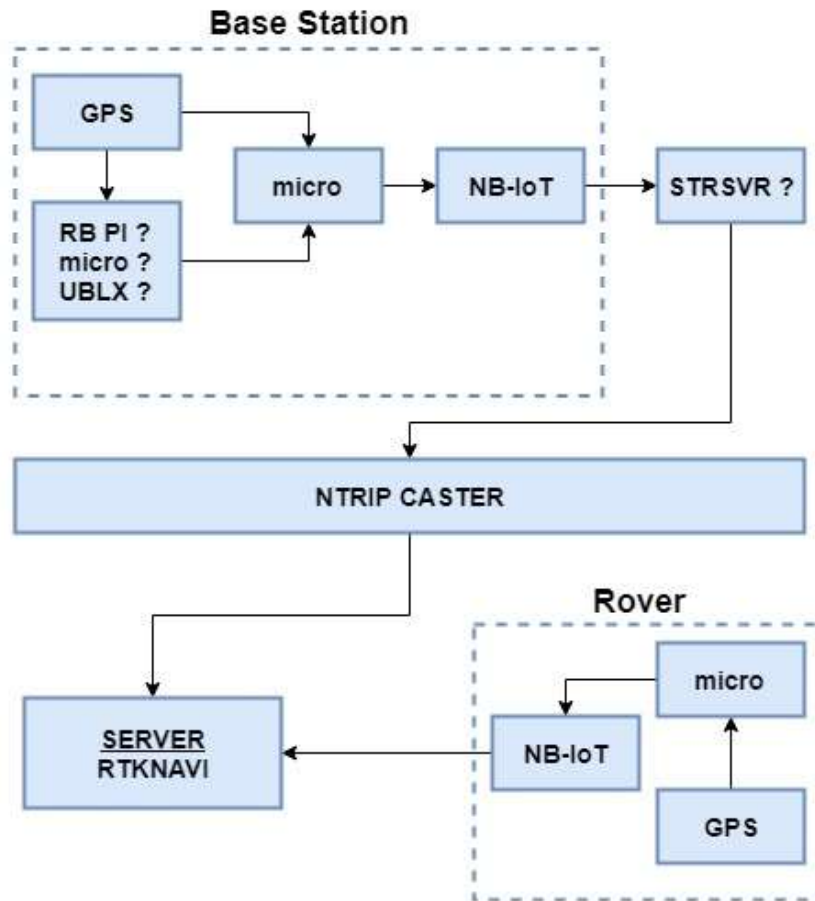


Ilustración 63 - Diagrama de evolución 1

Al investigar y discutir sobre el modelo de sistema planteado, surgieron varios puntos nuevos que no se habían considerado.

En el caso de las estaciones base, siendo que una de ellas puede otorgar información utilizable para las correcciones a un ilimitado número de Clientes RTK dentro de un radio de 50 Km, se puso en duda si realmente era necesario disminuir costos de hardware a cambio de complejizar la implementación. Con esto en mente surgieron las dos opciones mostradas en la Ilustración 64.

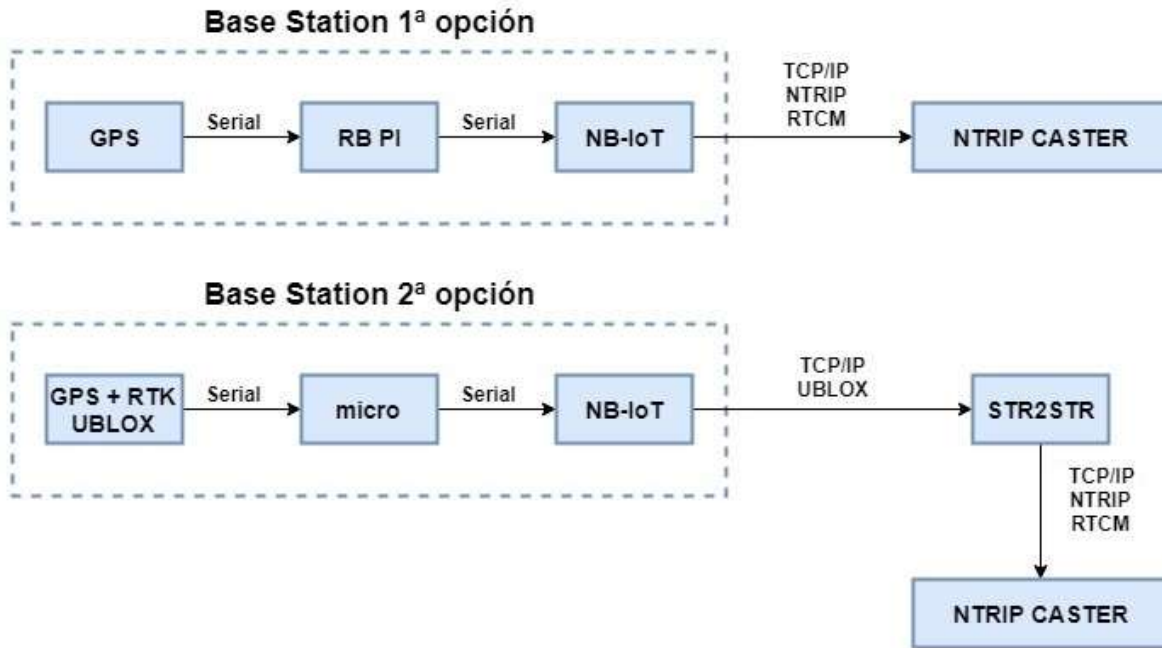


Ilustración 64 - Diagrama de evolución 2

La primera utiliza un sistema Linux y ejecuta software de RTKLib lo cual simplifica en gran medida la implementación del dispositivo. En la segunda se mantiene como prioridad el costo de hardware para la implementación.

En el caso del cliente RTK, como el volumen de dispositivos por cliente es potencialmente mayor que el de estaciones base, se pretende mantener bajo costo de hardware para la implementación, pero se consideró una posible necesidad de obtener las correcciones de la posición a bordo del Rover para una aplicación del cliente. De aquí, surgieron las dos opciones siguientes vistas en la Ilustración 65.

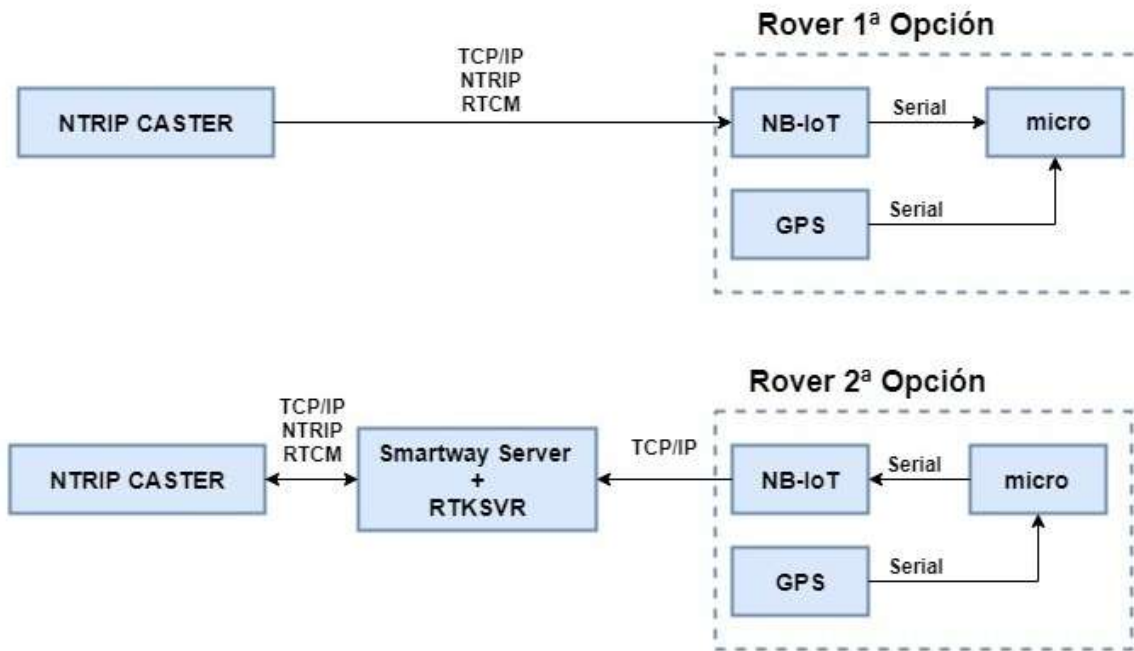


Ilustración 65 - Diagrama de evolución 3

En la primera, el microcontrolador utiliza las librerías de RTKLib y realiza las correcciones RTK en tiempo real a bordo del cliente RTK y la segunda mantiene la idea de que las correcciones se realicen en un servidor externo a los dispositivos.

Al evaluar estas opciones se dieron a conocer nuevos inconvenientes en los modelos del sistema.

En primera instancia, integrar estos sistemas con las aplicaciones de los clientes requería de un trabajo personalizado para cada uno de ellos. Por lo tanto, se optó por una solución que no requiera dicha integración, donde la única interacción, es el envío de la posición corregidas sobre un canal serial utilizando protocolos configurables.

Otra desventaja, es la fuerte restricción por el acceso a la red celular de las áreas de productividad (principalmente áreas rurales), para solucionar dicho problema se retoma la idea de una comunicación de radioenlace entre la estación referencia y los Clientes RTK.

Considerando lo anterior, se propuso implementar un sistema nombrado RTKReady (ver Ilustración 24), el cual deberá ser simple, robusto, de código abierto y con un mínimo de añadido de código.

En este modelo de sistema, el Rover y las estaciones base tienen como soporte principal un sistema Linux que ejecuta software de RTKLIB y se comunica con todos los módulos por medio de una interfaz serial.

B. Protocolo de configuraciones del sistema

Para las configuraciones del usuario, fue necesario implementar un protocolo sencillo que permitiera realizar dos acciones, tanto por mensaje texto como a través de la interfaz local de configuraciones. Este protocolo se implementa en la comunicación de procesos por medio de un socket, además de en la interfaz celular del usuario.

En este protocolo las acciones posibles son:

- Realizar una petición
- Realizar configuraciones

En relación a los mensajes, estos se componen de dos partes, el encabezado y los valores de configuración, por lo tanto, se definen tres tipos de separadores que permiten diferenciar cada parte de los mismos:

- \$ - Indica la finalización del encabezado.
- | - Separa cada valor de configuración.
- \n – Indica la finalización de un mensaje.

Realizar una petición: El protocolo permite dos encabezados.

Encabezado	Descripción	Utilización
Opt?	Pedido de Status del sistema.	Opt?\$\n
Opt!	Pedido de configuración actual de entrada y salida.	Opt!?\n

Tabla 5 - Formato estatus y configuración actual

Realización de configuraciones: El protocolo permite cinco configuraciones, dependiendo de a qué sistema se le está brindando servicio, es decir, Rover o Estación Base.

Encabezado	Valores de configuración
Opt1	Usuario y Contraseña
Opt2	Host, Puerto, Mount Point, Usuario, Contraseña
Opt3	Formato serial, Formato de archivo, Velocidad de transmisión serial (BaudRate), Enviar solución Single
Opt4	Host, Puerto, Mount Point Caster, Mount Point RF, Usuario, Contraseña
Opt5	Latitud, Longitud, Altura

Tabla 6 - Realización de configuraciones

Utilización	Descripción
Opt1\$User Pass\n	Configuración de Usuario y Contraseña.
Opt2\$Host Port Mntp User Pass\n	Configuración Radiofrecuencia y Caster.
Opt3\$Serial File Baud Single\n	Configuración Serial y Archivo (Válido solo en Rover).
Opt4\$Host Port MntpC MntpRF User Pass\n	Configuración de auto posicionamiento (Válido solo en Estación Base).
Opt5\$Lat Long High\n	Configuración de posicionamiento manual (Válido solo en Estación Base).

Tabla 7 - Formato de las configuraciones

El protocolo para las configuraciones realizadas a través de mensajes SMS es idéntico al anterior, con la excepción que el mensaje Opt1\$User|Pass\n solo es aceptado durante la configuración inicial del sistema.

C. AT Commands

Acrónimo para “ATtention Commands”, hace referencia al set de instrucciones utilizadas para la comunicación con un modem. Existen dos tipos de comandos:

- Comandos básicos: Son todos aquellos comandos los cuales no van continuados por ‘+’. Ejemplos: “AT”, “ATD”, “ATE”.
- Comandos Extendidos: Son todos aquellos comandos los cuales van continuados por ‘+’..

Dentro de los comandos extendidos, se pueden encontrar cuatro tipos de comandos:

Sintaxis	Descripción	Tipo
AT+<command_name>=?	Chequea si <command_name> es soportado por el modem.	Testeo
AT+<command_name>?	Chequea la configuración actual de <command_name>	Lectura
AT+<command_name>=value1, value2, ..., valueN	Modifica la configuración del <command_name>	Seteo
AT+<command_name>=value1, value2, ..., valueN	Modifica la configuración del <command_name>	Ejecución

Tabla 8 - AT Command: comandos extendidos

Como se pueden notar, los comandos de configuración y ejecución tienen la misma sintaxis. La diferencia entre ellos se encuentra en que los comandos de ejecución no almacenan los valores de los parámetros asignados a ellos. Por lo tanto, no hay un comando de lectura disponible para recuperar los valores de los últimos parámetros asignados a un comando de ejecución.

D. Protocolo SPI

Es un protocolo de comunicación que permite la transferencia de información entre dispositivos. Este es un protocolo síncrono sobre un bus de datos, es decir, la transferencia se realiza acorde a una señal de reloj. Al dispositivo que genera la señal de reloj, generalmente, el microcontrolador, se lo conoce como Maestro y al otro como Esclavo, siendo posible la transferencia a más de uno.

Por lo tanto, para que esta comunicación sea bidireccional, y a múltiples esclavos se disponen de cuatro señales:

- SCK: Señal de reloj generada por el Maestro.
- MOSI: *Master Out/Slave In*. Datos enviados de Maestro a Esclavo.
- MISO: *Master In/Slave Out*. Datos enviados de Esclavo a Maestro.
- SS: *Slave Select*. Señal de control de esclavo, es decir, le dice al mismo cuando despertar para enviar o recibir. Es necesario contar con un pin SS por cada Esclavo.

Comunicación Lora Bee – PI Zero W

Para lograr esta comunicación no solo se debió investigar sobre el protocolo, sino también el cómo es implementado por el módulo LoRaBee. En este sentido, provee de tres modos de acceso a los registros por SPI:

Acceso SINGLE:

- Se utiliza para escritura/lectura de un solo registro.
- Se envían dos bytes en el cual el primero contiene la dirección de memoria del registro a escribir/leer.
 - Escritura: El segundo byte contiene el dato a escribir.
 - Lectura: El segundo byte se envía para que el módulo pueda responder con el estado del registro.

Acceso BURST:

- Se utiliza para escritura/lectura de registros contiguos.

- Se envían múltiples bytes en donde el primero contiene la dirección de memoria del primer registro a modificar/leer.
 - La dirección se incrementa automáticamente internamente entre cada byte.
 - Los restantes bytes se envían para escritura en los registros o para que el módulo pueda responder con los estados de los registros.

Acceso FIFO:

- Se utiliza para la escritura/lectura en el registro FIFO.
- La dirección no se incrementa automáticamente, sino que se memoriza y no necesita ser enviada entre cada byte de datos, es decir, las operaciones de lectura y escritura se realizan siempre sobre el mismo registro.

En todos los modos de acceso, el primer bit de la dirección del registro contiene la operación a realizar por el Maestro:

- 1 – Escritura.
- 0 – Lectura.

Dado estos tres métodos de acceso al dispositivo, se crearon dos funciones incluidas en la librería Traductor RF, que proporcionan estos tipos de acceso de forma fácil.

E. Transformación de formato Latitud y Longitud

Debido a la utilización de una API de Google para mostrar el recorrido de un Rover a partir de los archivos de salida, se hizo necesaria la conversión del formato de latitud y longitud brindado por NMEA. Esto se debe a que, como se muestra en el Anexo F, este brinda coordenadas de latitud y longitud en formato grados y minutos decimales los cuales deben ser convertidos a grado decimal.

La ecuación para esta transformación se muestra a continuación:

$$\text{Grados decimales} = \text{grados} + \text{minutos}/60$$

Cabe resaltar que las latitudes que se indican como Sur, deben incorporar el signo negativo en el formato grado decimal. A si como también, las longitudes que se indican como Oeste.

F. Protocolo NMEA

A la hora de utilizar NMEA como formato de salida para las correcciones, ya sea, la salida serial o por archivo, fue necesario comprender la información que brinda.

Para comenzar, el formato se divide en sentencias, en donde, cada una de ellas se encuentra separada por dos caracteres, un retorno de carro (ASCII 13, ‘\r’) y un salto de línea (ASCII 10, ‘\n’), y cada una de estas sentencias puede ser considerado como un conjunto de tres secciones, la cabecera, los datos, y el *checksum*.

La cabecera tiene un largo fijo de 6 caracteres comenzando siempre con el carácter ‘\$’. Luego los caracteres restantes indican la constelación que brinda el servicio, también conocido como Talker ID, y el tipo de mensaje como por ejemplo GGA, GSV, GSA.

Los posibles Talker ID que se encontraron son:

- GP para GPS.
- GL para GLONASS.
- GN cuando se utiliza información de más de una constelación.

Una vez terminado el envío de datos los cuales varían de acuerdo al tipo de mensaje que se envía, se termina la sentencia con el carácter ‘*’ continuado por el *checksum*. Este es utilizado para determinar la validez del mensaje, es decir, si el mensaje recibido es el mismo al mensaje enviado.

Dado que en el sistema nos interesa la información asociada al nivel de corrección de la posición, se analiza a continuación el mensaje que corresponde a esta información.

```
0      1      2  3      4      5 6 7 8      9  10 11 12 13 14 15  
$ - -GGA,hhmmss.ss,lllll,a,yyyyy.yyyyyyy,a,x,xx,x.x,xxx.xxx,M,xx.xx,M,x.x,xxxx,* hh
```

A continuación, se procede a explicar punto a punto la sentencia GGA del protocolo NMEA.

1. Hora UTC.
2. Latitud en grados y minutos decimales.
3. N o S (Norte o Sur).
4. Longitud en grados y minutos decimales.

5. E o W (Este o Oeste).
6. Indicador de calidad.
 - a. 0 - Inválido (Posición no disponible).
 - b. 1 - *Fix* GNSS autónomo (No se utilizan datos de correcciones).
 - c. 2 - *Fix* DGNSS.
 - d. 3 - *Fix* PPS.
 - e. 4 - *Fix* RTK.
 - f. 5 - *Float* RTK.
 - g. 6 - (Desconocido)
 - h. 7 - Modo de ingreso manual.
 - i. 8 - Modo de simulación.
 - j. 9 - *Fix* DGNSS de NovAtel cuando se utiliza el sistema WAAS.
7. Cantidad de satélites a la vista.
8. Horizontal DOP.
9. Altura de la antena por encima/debajo del nivel del mar nominal (geoide)
10. Unidad de altura antena (metros)
11. Separación geoidal. Hace referencia a la diferencia en metros entre el elipsoide de referencia (datum) y el nivel del mar nominal (geoide). Valores negativos indican que el nivel del mar se encuentra por debajo del elipsoide de referencia.
12. Unidad de la separación geoidal.
13. Edad de la corrección para sistemas DGNSS y RTK en segundos.
14. ID de la estación de referencia.
15. Checksum.

G. Manual de usuario

En esta sección se dará a conocer el procedimiento para que, una vez adquirido el producto, el usuario sea capaz de entender y configurar el sistema de forma de sacar provecho de todas las funcionalidades del mismo, aprovechando todo su potencial.

El mismo se compondrá de tres secciones en común que le permitirán realizar la conexión física de cualquiera de los dispositivos, realizar el ingreso por primera vez al sistema, e ingresar en la misma red, para luego mostrar las posibles configuraciones de cada sistema (Estación de referencia o Cliente RTK).

Cabe mencionar que cada configuración se puede realizar a partir del método local o remoto siguiendo el Anexo B.

Conexión física del sistema:

Como se hace notar el sistema se entrega a usuario casi armado, siendo necesario únicamente la conexión de la antena GNSS además de la alimentación del mismo. Opcionalmente, podrá conectar la antena RF y el Dongle USB.

Para la conexión de la antena GNSS, véase la Ilustración 31, Ilustración 35 y Ilustración 36.

En caso de ser necesario para su implementación, es decir, si su dispositivo dará/recibirá flujos GNSS por un enlace de radio frecuencia, véase la Ilustración 34, Ilustración 37 e Ilustración 38 para la conexión de la antena RF.

En caso de ser necesario para su implementación, es decir, si su dispositivo dará/recibirá flujos GNSS a partir de un NTRIP Caster, deberá conectar el Dongle USB hasta el puerto USB mini. Como podrá notar que el dispositivo cuenta con dos puertos debiendo ser introducido en el que se encuentra situado hacia el interior o centro del mismo. A su vez, se hará necesario el uso de un adaptador USB a mini USB el cual será proporcionado.

Primera puesta en marcha:

La primera vez que conecta el sistema deberá realizar configuraciones para su funcionamiento. Para ello, se debe acceder al sistema por el método local de configuración ingresando en la misma red de la Raspberry. Véase la sección Conectarse a la misma Red

Nótese que, para esta primera configuración, no es posible la configuración por el método remoto. Esto se debe a que para esta configuración todavía no se tiene configurado un usuario y contraseña.

Una vez ingresado en la misma, debe ingresar la IP 192.168.0.1 desde cualquier navegador. Allí aparecerá la página para la creación de usuario en donde deberá especificar usuario y contraseña que se utilizará. Recuerde que estos datos son necesarios para realizar futuras configuraciones, por lo que no deben ser olvidados.

Conexión a la misma red:

Para realizar el ingreso a la red que brinda la Raspberry, deberá presionar el pulsador que enciende la Wifi como punto de acceso, y dirigirse a la configuración de redes en su portátil o dispositivo móvil. Podrá notar que ahora le aparecerá una red llamada “Base Station” o “RTK Ready” dependiendo de qué dispositivo se esté usando.

Configuraciones de estación base:

Dado que el rendimiento del sistema de posicionamiento (RTK) se encuentra ligado a la posibilidad de establecer la posición de esta estación base con precisión, se brindan dos formas para llevarlo a cabo, el posicionamiento manual y el automático.

En el posicionamiento manual, el usuario deberá proporcionar latitud, longitud y altura.

En el posicionamiento automático, la estación base se transforma en un Rover, siendo necesario realizar configuraciones propias del Rover. Ver sección Configuraciones Rover

Una vez posicionada la estación base, se debe ingresar la forma en la que la misma se dará a conocer. Cabe destacar que esta configuración se puede realizar en cualquier momento, sin embargo, no tomará efecto hasta esta esté en una posición fija, es decir, se haya terminado el auto posicionamiento o se haya ingresado una manual.

Las formas en la que se puede dar a conocer, es decir, por donde enviará los flujos GNSS son a partir de un Caster y/o por radio frecuencia.

Para la configuración de envío de flujos por medio de un NTRIP Caster se pedirá:

- Host: Es la dirección IP o URL que identifica en la Internet al servidor que corre un NTRIP Caster.
- Puerto: Es un número por el cual el servidor escucha peticiones de este tipo.
- *MountPoint*: Un nombre conocido por el usuario con el que se guardará la estación en el NTRIP Caster, el cual es utilizado para la obtención del flujo GNSS.

User y Contraseña: Datos de autenticación necesarios para publicar la información en el NTRIP Caster. El mismo campo no es obligatorio ya que ciertos *Casters* permiten la conexión sin credenciales.

Para la configuración de envío de flujos por medio de un enlace de radio, se pedirá solamente el *MountPoint*, el cual puede ser distinto al utilizado para el NTRIP Caster. Al igual que en el Caster, es el nombre con el cual puede ser referido en el Rover para la obtención del flujo.

Configuraciones de Rover:

Para poner en funcionamiento el Rover, se debe indicar de donde obtendrá los datos para realizar las correcciones en su posición. Los métodos posibles son a partir de un Caster a una estación de referencia propia o de uso público, o por un enlace de radio frecuencia a una estación propia.

Para la configuración de envío de flujos por medio de un NTRIP Caster se pedirá:

- Host: Es la dirección IP o URL que identifica en la Internet, al servidor que corre un NTRIP Caster. Es posible utilizar el servicio de correcciones del Uruguay brindado por la REGNA-ROU.
- Puerto: Es un número por el cual el servidor propio, y el de la REGNA-ROU, escuchan peticiones de este tipo.
- *MountPoint*: El nombre que tiene asignado una estación base en el NTRIP Caster.
- Usuario y Contraseña: Datos para autenticación a la hora de obtener el flujo. El mismo campo no es obligatorio ya que el Caster puede ser de acceso libre a todo público.
- *nmeaToBase*: Indica si se quiere enviar su posición a la estación base. Resaltamos que esta opción se implementó como una mejora para una solución de red sin la necesidad de modificar los protocolos.

Para la configuración de recepción de flujos por medio de un enlace de radio, debe haber realizado previamente la configuración de la estación base. En caso de ser requerido, véase la sección configuraciones de estación base. Para esta configuración se pedirá el *mountpoint* elegido para la estación base.

Una vez elegido el método de obtención de flujos, es posible configurar la salida. Cabe destacar que la salida la podrá configurar en cualquier momento, teniendo en cuenta que la configuración de la misma causará el reinicio sistema.

Las opciones que se brindan son la salida serial y en forma de archivo. En ambos casos, se deben suministrar:

- Formato: Formato de envío o almacenamiento.
- *Solution Single*: Este campo indica si debe enviar la posición GPS sin ninguna corrección si la misma no se encuentra disponible.

A su vez, para la salida serial se debe configurar la velocidad del envío de datos.

H. Procedimiento de instalación

Como medida de seguridad, es muy recomendable modificar el usuario que la distribución Raspbian tiene por defecto. Los pasos se muestran a continuación utilizando como ejemplo el nombre de usuario “admin”, el cual que se mantendrá durante todo el procedimiento de instalación:

1. Crear usuario nuevo
`sudo adduser admin`
2. Añadir el usuario nuevo a grupos
`sudo usermod -aG sudo tty gpio spi admin`
3. Eliminar el usuario por defecto
`sudo deluser pi`

Instalación de paquetes para el correcto funcionamiento del sistema:

```
sudo apt-get install dnsmasq hostapd dhcpd wvdial usb-modeswitch ppp at apache2 php  
raspberrypi-kernel-headers
```

Añadir/eliminar al usuario de un grupo, necesario para las librerías wvdial, soft_uart y apache:

1. `sudo usermod -aG dip dialout admin`
2. `gpasswd -d admin root`

Habilitar Raspberry como servidor SSH:

1. `sudo raspi-config`
2. Interfacing Options -> SSH -> enable SSH server

A continuación, se mostrarán configuraciones adicionales necesarias en algunos paquetes. Como recomendación puede ser útil realizar un respaldo de los archivos antes de ser modificados.

Configuración para establecer punto de acceso

1. Detener procesos

```
sudo systemctl stop dnsmasq
sudo systemctl stop hostapd
```
2. Configurar una IP estática en wlan0

```
sudo vim /etc/dhcpd.conf
interface wlan0
static ip_address=192.168.4.1/24
nohook wpa_supplicant
```
3. Reiniciar dhcpd para que tome las configuraciones de wlan0 nuevas

```
sudo service dhcpd restart
```
4. Configurar dnsmasq aceptar solamente peticiones DHCP y DNS

```
sudo vim /etc/dnsmasq.conf
interface=wlan0
dhcp-range=192.168.0.2,192.168.0.5,255.255.255.0,24h
```
5. Configurar hostapd para transformar esta interfaz en un punto de acceso

```
sudo vim /etc/hostapd/hostapd.conf
interface=wlan0
driver=nl80211
ssid=BaseStation
hw_mode=g
channel=7
wmm_enabled=0
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=T3s1s2018
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rns_pairwise=CCMP
```
6. Indicar donde encontrar el archivo de configuración

```
sudo vim /etc/default/hostapd
#Reemplazar linea [#DAEMON_CONF]
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

7. Iniciar los servicios

```
sudo systemctl start dnsmasq
sudo systemctl start hostapd
```

Configuración Dongle 3G

1. Editar archivo de configuración de discado al modem

```
sudo vim /etc/wvdial.conf
[Dialer Default]
Modem = /dev/gsmmodem
Init1 = ATZ
Init2 = ATQ0 V1 E1 S0=0 &C1 &D2
Init3 = AT+CGDCONT=1, "IP", "antel.lte"
Stupid Mode = 1
Modem Type = Analog Modem
Phone = *99#
New PPPD = yes
ISDN = 0
Username = { }
Password = { }
Baud = 9600
```

2. Configuración de eventos

```
vim /lib/udev/rules.d/40-usb_modeswitch.rules
Añadir: RUN+="/bin/bash /usr/local/bin/dongle.bash reload"
Lugar: al final de la línea comenzando con en ATTRS{idVendor}=="12d1"
```

Configuración Apache2

1. Ejecución Web con usuario y grupo determinados

```
sudo vim /etc/apache2/envvars
export APACHE_RUN_USER=admin
export APACHE_RUN_GROUP=admin
```
2. Redirección página de apache por defecto a página propia

```
sudo vim /var/www/html/index.html
<script>
    location.href = "./Web_XX/login.php";
</script>
```
3. Eliminar Index de carpetas

```
sudo vim /etc/apache2/apache2.conf
# Deny show Index on folders
<Directory /var/www>
    Options -Indexes
    AllowOverride All
    Order allow,deny
    Allow from all
</Directory>
```
4. Reinicio proceso

```
sudo service apache2 restart
```

Instalación y configuración soft_uart

1. `git clone https://github.com/adrianomarto/soft_uart`
2. `cd soft_uart`

```
make
sudo make install
```