



UNSAM
UNIVERSIDAD
NACIONAL DE
SAN MARTÍN

Sistema Global de Navegación Satelital y Sensado de Temperatura Para Vehículos

Proyecto Final Integrador – Escuela de Ciencia
y Tecnología – UNSAM – 2016

Tutor: Marcelo Romeo

marcelo.romeo@gmail.com

Autor: Federico Nicolás Human

federico.human@gmail.com



UNSAM
UNIVERSIDAD
NACIONAL DE
SAN MARTÍN

Agradecimientos

A mis padres, Néstor y Alejandra, por haberme formado y acompañado a lo largo de mi vida.

A mi familia, por su apoyo incondicional.

A mis amigos, por brindarme su valioso tiempo y por todos los momentos que compartimos.

A la Universidad Nacional de San Martín, a sus autoridades y profesores, por darme la oportunidad de estudiar y formarme como profesional.



Tabla de Contenidos

<u>Agradecimientos</u>	2
<u>Tabla de Contenidos</u>	3
<u>Lista de Figuras</u>	5
<u>Lista de tablas</u>	7
1. <u>Objetivo</u>	8
2. <u>Selección del Proyecto</u>	9
2.1. <u>Descripción</u>	9
2.2. <u>Argumento</u>	10
3. <u>Gestión de Proyectos</u>	11
3.1. <u>Definición de Proyecto</u>	11
3.2. <u>Etapas de Proyecto</u>	12
3.3. <u>Gestión de Proyecto</u>	13
3.4. <u>Declaración de Alcance</u>	15
3.5. <u>Estructura de Descomposición del Trabajo</u>	17
3.6. <u>Planificación</u>	20
4. <u>Sistema Global de Navegación por Satélite (GNSS)</u>	24
4.1. <u>Introducción</u>	24
4.2. <u>Arquitectura</u>	25
4.3. <u>Computación</u>	26
4.4. <u>Recepción</u>	27
4.5. <u>Señales Satelitales</u>	27
4.6. <u>Errores</u>	28



<u>5.</u>	<u>Sistema Global para las Comunicaciones Móviles (GSM)</u>	31
5.1.	<u>Introducción</u>	31
5.2.	<u>Arquitectura</u>	31
5.3.	<u>Estructura</u>	32
5.4.	<u>Servicio General de Paquetes vía Radio (GPRS)</u>	34
<u>6.</u>	<u>Desarrollo Técnico del Proyecto</u>	36
6.1.	<u>Hardware</u>	36
6.2.	<u>Firmware</u>	39
6.3.	<u>Software</u>	43
6.4.	<u>Servidor Web</u>	46
6.5.	<u>Aplicación Web</u>	48
<u>7.</u>	<u>Conclusiones Finales</u>	50
<u>8.</u>	<u>Referencias</u>	51



Lista de Figuras

<u>Figura 2.1.1: Interacción entre los componentes del sistema.</u>	10
<u>Figura 3.1.1: Relación entre los tres componentes de un proyecto.</u>	11
<u>Figura 3.3.1: Relación entre los procesos en la gestión de un proyecto y las etapas por las cuales atraviesa.</u>	14
<u>Figura 3.6.1: Diagrama de Gantt del proyecto dividido en cuatro etapas.</u>	21
<u>Figura 3.6.2: Diagrama de Gantt de la etapa “Comienzo del proyecto”.</u>	22
<u>Figura 3.6.3: Diagrama de Gantt de la etapa “Organización y preparación”.</u>	22
<u>Figura 3.6.4: Diagrama de Gantt de la etapa “Llevando adelante el proyecto”.</u>	22
<u>Figura 3.6.5: Diagrama de Gantt de la etapa “Cierre del proyecto”.</u>	23
<u>Figura 3.6.6: Recursos asignados a las tareas del proyecto.</u>	23
<u>Figura 4.1.1: Orbitas de los sistemas GNSS actuales.</u>	24
<u>Figura 4.2.1: Segmentos de un sistema GNSS.</u>	25
<u>Figura 4.5.1: Espectro de las bandas L utilizadas por los sistemas GNSS.</u>	28
<u>Figura 4.6.1: Propagación multicamino.</u>	30
<u>Figura 5.2.1: Estructura celular de la red GSM.</u>	32
<u>Figura 5.3.1: Estructura PLMN.</u>	34
<u>Figura 5.4.1: Funciones de control de acceso a la red GPRS.</u>	35
<u>Figura 6.1.1: Diagrama esquemático Raspberry Pi.</u>	36
<u>Figura 6.1.2: Diagrama de conexiones del sistema prototipo.</u>	39
<u>Figura 6.2.1: Diagrama de flujo del firmware.</u>	42
<u>Figura 6.3.1: Arquitectura fundamental del sistema operativo GNU/Linux.</u>	43
<u>Figura 6.3.2: Perspectiva de la arquitectura del núcleo Linux.</u>	44



UNSAM
UNIVERSIDAD
NACIONAL DE
SAN MARTÍN

<u>Figura 6.3.3: Firmware y software de la aplicación.</u>	46
<u>Figura 6.4.1: Flujo de paquetes de datos.</u>	47
<u>Figura 6.4.2: Tablas de la base de datos.</u>	48
<u>Figura 6.5.1: Selección de rango de visualización.</u>	48
<u>Figura 6.5.2: Google Maps API.</u>	49
<u>Figura 6.5.3: Gráfico de temperatura Plotly.</u>	49



UNSAM
UNIVERSIDAD
NACIONAL DE
SAN MARTÍN

Lista de tablas

<u>Tabla 3.4.1: Declaración de alcance.</u>	15
<u>Tabla 3.5.1: Comienzo del proyecto.</u>	17
<u>Tabla 3.5.2: Organización y preparación.</u>	17
<u>Tabla 3.5.3: Llevando adelante el trabajo.</u>	18
<u>Tabla 3.5.4: Cierre del proyecto.</u>	20



UNSAM
UNIVERSIDAD
NACIONAL DE
SAN MARTÍN

1. Objetivo

Realizar un proyecto final integrador que contemple las etapas de gestión de proyectos y a su vez integre contenidos de las distintas asignaturas de la carrera de Ingeniería en Electrónica. Seleccionar el proyecto a desarrollar teniendo en cuenta los requisitos y las reglas de diseño de proyectos, además de la viabilidad del proyecto desde un punto de vista comercial. Realizar la declaración de alcance del proyecto y obtener su aprobación. Desarrollar la estructura de descomposición del trabajo y una planificación del proyecto contemplando los recursos y los costos, y luego establecer el camino crítico para lograr el objetivo. Llevar a cabo el proyecto a través de todas sus etapas. Desarrollar las tecnologías necesarias para la realización del proyecto. Desarrollar los aspectos técnicos del proyecto. Conclusiones finales.



2. Selección del Proyecto

2.1. Descripción

Se seleccionó como proyecto un sistema de posicionamiento global y sensado de temperatura para vehículos. Los sistemas de posicionamiento global permiten al usuario monitorear con exactitud tanto un vehículo como toda una flota.

El sistema de posicionamiento global obtiene la ubicación en coordenadas donde se encuentra el vehículo, además de que informa su velocidad, sentido e inclinación de desplazamiento. El transductor de temperatura permite monitorear la temperatura del recinto donde se lo coloque. El proyecto incluye un LED testigo que cambia su color según dos umbrales establecidos por el usuario. El LED testigo cambia a rojo cuando la temperatura es mayor el umbral superior, a azul cuando es menor al umbral inferior, y verde cuando se encuentra entre los umbrales. Es posible programar alertas de texto o correo electrónico basadas en parámetros establecidos por el usuario por adelantado, por ejemplo, cuando el vehículo excede la velocidad máxima permitida, cuando recorre rutas alternativas, o la temperatura alcanza valores no deseados, entre otras configuraciones.

La información adquirida por el dispositivo instalado en el vehículo se transmite por la red de telefonía móvil hacia un servidor web, que luego la almacena en una base de datos, y posteriormente el usuario la puede interpretar en una interfaz web gráfica, o descargar y procesar. Se pueden realizar todo tipo de cálculos una vez adquirida la información, como distancia promedio recorrida o consumo promedio de combustible. Paralelamente se almacena una copia de toda la información en la memoria física del sistema en caso de existir problemas en la red de telefonía móvil.

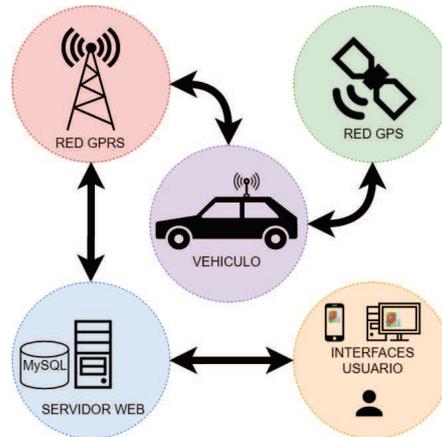


Figura 2.1.1: Interacción entre los componentes del sistema.

2.2. Argumento

El conjunto de todas sus características da una ventaja singular a los departamentos de mantenimiento y logística de una empresa transportista, ofreciendo información clave para mejorar el negocio; es a ese segmento del mercado al que se tiene como consumidor principal. Los sistemas de posicionamiento global que se encuentran actualmente en el mercado local no ofrecen la posibilidad de modificar su hardware, firmware o software, por lo que no existe posibilidad de expandirlos hacia las necesidades puntuales de cada usuario. Se ideó este proyecto para ofrecer una alternativa de valor agregado, y para cubrir las necesidades que puedan surgir. Existe la posibilidad de añadir cualquier tipo de transductor electrónico en conjunto o reemplazo del de temperatura, según las necesidades del transporte en cuestión, dándole versatilidad a la solución ofrecida al usuario. También es posible añadir actuadores o conmutadores en función de las necesidades del usuario.

El desarrollo completo del código en todas sus etapas, y el uso de hardware libre, otorga la versatilidad y simpleza de modificar cualquier segmento para adecuarlo a las necesidades del usuario, pudiendo así adaptar un mismo producto con una mínima inversión a las diferentes necesidades del mercado.



3. Gestión de Proyectos

3.1. Definición de Proyecto

Un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único. La naturaleza temporal de los proyectos indica que un proyecto tiene un principio y un final definidos. El final se alcanza cuando se han alcanzado los objetivos del proyecto, o cuando el proyecto es finalizado porque sus objetivos ya no son deseados o realizables. Generalmente el producto, servicio o resultado creado por el proyecto no es temporal; la mayor parte de los proyectos son emprendidos para crear un resultado duradero.

Un proyecto siempre está compuesto de los siguientes tres componentes:

- *Producto*: Es el resultado único deseado que justifica la existencia del proyecto.
- *Planificación*: Fechas establecidas que dictan los tiempos de comienzo y finalización de las tareas que conforman el proyecto.
- *Recursos*: Cantidad necesaria de personas, presupuesto y otros recursos, que son indispensables para la finalización del proyecto.

Cada uno de los componentes de un proyecto influye sobre los otros dos, y sobre la definición de estos tres componentes se trabaja para alcanzar los resultados deseados.

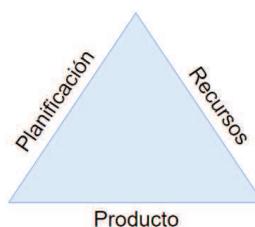


Figura 3.1.1: Relación entre los tres componentes de un proyecto.



Si se deseara reducir la duración del proyecto, sería preciso incrementar los recursos requeridos, o reducir la calidad del producto deseado. Por otra parte, si se deseara reducir los recursos requeridos, tendrían que verse extendida la duración del proyecto, o reducirse la calidad del producto. Finalmente, si fuera necesario incrementar la calidad del producto, tendrían que destinarse mayores recursos o extenderse la duración del proyecto. Encontrar la proporción justa de cada uno de estos tres componentes es clave a la hora de definir un proyecto.

3.2. Etapas de Proyecto

Cualquier proyecto atraviesa las siguientes cuatro etapas:

- *Comienzo del proyecto*: Esta etapa implica generar, evaluar y enmarcar las necesidades del proyecto, y el enfoque general para llevarlo a cabo, además de acordar un plan detallado de realización del proyecto. Los resultados de esta etapa pueden incluir la aprobación para pasar a la siguiente etapa, documentación de la necesidad de la finalización proyecto y estimaciones aproximadas de tiempo y recursos necesarios.
- *Organización y preparación*: Esta etapa implica desarrollar un plan que especifique los resultados deseados, el trabajo que hacer, el tiempo, los costos, y otros recursos necesarios, además de un plan para enfrentar los riesgos clave del proyecto. Los resultados de esta etapa pueden incluir un plan del proyecto que documente los resultados esperados, y el tiempo, los recursos y las tareas necesarias para lograrlos.
- *Llevando adelante el trabajo*: Esta etapa implica establecer el equipo del proyecto y los sistemas de soporte, realizar el trabajo planificado, y monitorear y controlar la performance para asegurar la adherencia al plan. Los resultados de esta etapa pueden incluir resultados del proyecto, reportes del progreso del proyecto, y otras comunicaciones.



- *Cierre del proyecto:* Esta etapa implica evaluar los resultados del proyecto, obtener las aprobaciones necesarias, asignar a los miembros a otros proyectos, cierres financieros, y realización de una evaluación final del proyecto. Los resultados de esta etapa pueden incluir resultados del proyecto, además de recomendaciones y sugerencias para proyectos similares que puedan surgir en el futuro.

3.3. Gestión de Proyecto

La gestión de proyecto es el proceso de guiar un proyecto desde su comienzo a través de su desarrollo hasta su cierre final, aplicando conocimientos, habilidades, herramientas, y técnicas a las actividades del proyecto, para garantizar las necesidades y expectativas de las audiencias interesadas en su realización. La gestión de proyectos incluye cinco conjuntos de procesos:

- *Procesos de iniciación:* Reconocer que un proyecto debe comenzar y comprometerse a realizarlo, clarificar la necesidad de la existencia del proyecto, definir las expectativas sobre el producto del proyecto, calcular los recursos necesarios y el presupuesto, identificar las audiencias involucradas en el proyecto.
- *Procesos de planificación:* Idear y mantener un esquema de trabajo para cumplir la necesidad que el proyecto debe satisfacer, detallar el alcance del proyecto, los plazos, los recursos, los riesgos, así como también el enfoque para las comunicaciones del proyecto, la calidad y la gestión de las compras externas de bienes y servicios necesarios.
- *Procesos de ejecución:* Establecer y gestionar el equipo asignado al proyecto, comunicación y gestión de las audiencias involucradas en el proyecto, y ejecución de los planes del proyecto.
- *Procesos de monitoreo y control:* Seguimiento del desempeño y toma de acciones correctivas necesarias para asegurar que los planes del proyecto se implementen con éxito y se logren los resultados deseados.



- *Procesos de cierre*: Finalizar todas actividades del proyecto, formalizar la aceptación del proyecto por parte de las audiencias interesadas.

Estos cinco grupos de procesos dan soporte al proyecto a través de las cuatro etapas de su ciclo de vida. Los procesos de iniciación apoyan el trabajo a realizar al iniciar el proyecto, y los procesos de planificación respaldan la etapa de organización y preparación. Los procesos de ejecución guían las tareas del proyecto llevadas a cabo al realizar el trabajo, y los procesos de cierre son utilizados para realizar las tareas que ponen fin al proyecto. Se debe considerar también el retroceso desde los procesos de ejecución hacia los procesos de planificación; entonces es necesario regresar a la etapa de organización y preparación para modificar los planes existentes, ya sea para resolver problemas imprevistos, o motivado por nueva información que se adquiere al realizar el trabajo del proyecto. Finalmente, los procesos de monitoreo y control se utilizan en cada una de las cuatro etapas para asegurar que el trabajo se está realizando de acuerdo con lo planeado.



Figura 3.3.1: Relación entre los procesos en la gestión de un proyecto y las etapas por las cuales atraviesa.



3.4. Declaración de Alcance

El objetivo de una declaración de alcance es establecer de forma escrita los resultados que producirá el proyecto, además de las restricciones y las suposiciones bajo las cuales se llevará a cabo. Se deben considerar medidas y especificaciones para determinar si se han cumplido los objetivos. Todas las partes que se ven involucradas en el proyecto deben estar de acuerdo en todos los términos incluidos en la declaración de alcance antes de que se comience a trabajar en el proyecto. Las declaraciones de alcance pueden tomar muchas formas dependiendo del proyecto implementado y la naturaleza de la organización.

Tabla 3.4.1: Declaración de alcance.

Título del proyecto	Gerente de proyecto
Sistema Global de Navegación Satelital y Sensado de Temperatura Para Vehículos	Federico Human
Declaración de Alcance	
Justificación	
Se detectó la necesidad de ofrecer sistemas de posicionamiento global con la posibilidad de expandir el sistema hacia los requisitos de los distintos usuarios, adicionando todo tipo de transductores y/o actuadores, que puedan ser necesarios en un transporte, ya que los existentes en el mercado no ofrecen esta posibilidad.	
Objetivos	
Desarrollar íntegramente el sistema de posicionamiento global y sensado de temperatura para vehículos.	
Descripción del alcance del producto y criterios de aceptación del producto	
Alcance	Criterio de aceptación
El sistema debe funcionar dentro de un vehículo.	Alimentación 12 VDC - 24 VDC.
	Gabinete compacto y resistente a vibraciones.
	Reglas de diseño del proyecto.



El sistema debe ser estable desde el punto de vista del software y firmware.	Utilizar el sistema operativo Linux.
	Reinicio ante cualquier falla.
	Multiprocesamiento y acceso limitado a los recursos.
El sistema debe medir temperatura e indicar rangos en un LED testigo.	Comprobar valores obtenidos. Ver media y desviaciones. Comprobar correcto funcionamiento del LED testigo.
El sistema debe medir posición global.	Comprobar valores obtenidos. Ver media y desviaciones.
El sistema debe almacenar los datos adquiridos en memoria local.	Almacenamiento seguro para no corromper el sistema de almacenamiento.
El sistema debe transmitir los datos a un servidor accesible por el usuario.	Transmisión GPRS al servidor web por métodos HTTP.
	Recepción y almacenamiento de datos en base de datos MySQL.
Crear una interfaz web para usuario que sea simple e intuitiva.	Seleccionar el vehículo a visualizar dentro de la flota.
	Seleccionar el rango de tiempo en un calendario para limitar la visualización de datos.
	Exhibir los datos de posición global en un mapa interactivo.
	Exhibir los datos de temperatura en un gráfico interactivo.
Restricciones	
Limitantes	
1. Solo puede participar activamente en el desarrollo del proyecto el gerente de proyecto, aunque se detallen todas las tareas como si fueran llevadas a cabo por un equipo.	
2. Por el limitante anterior, se consideraron los tiempos de aprendizaje de nuevos conocimientos en el desarrollo de cada una de las tareas.	
3. Se deben respetar las reglas de diseño de proyectos establecidas previamente.	
Necesidades	
1. Presupuesto de \$5000 para la compra del hardware necesario para desarrollar el prototipo.	
2. Herramientas necesarias para el armado del prototipo: destornilladores, taladro, alicate, pinza de punta, soldador y estaño.	
3. Recursos humanos: un gerente de proyecto, un ingeniero electrónico, un programador web, un técnico de compras y un programador de embebidos.	



4. Hosting web para alojar el servidor, que representa un pago mensual de aproximadamente \$100.

3.5. Estructura de Descomposición del Trabajo

La estructura de descomposición del trabajo es una herramienta utilizada en gestión de proyectos, orientada a entregables, que consiste en la descomposición jerárquica del trabajo a ser ejecutado por el equipo del proyecto, para cumplir con los objetivos y crear los entregables requeridos. Su propósito es organizar y definir el alcance total aprobado del proyecto según la declaración de alcance. Finalmente, sirve como base para la planificación del proyecto.

Al descender en los niveles jerárquicos en la estructura de descomposición del trabajo se ven representadas definiciones más detalladas del trabajo a realizar en el proyecto. Los diferentes niveles jerárquicos tienen diferentes nombres, donde al nivel superior se lo llama típicamente proyecto, y al nivel más bajo se lo llama paquete de trabajo. Los niveles intermedios se los conoce como fases, sub-proyectos, asignaciones de trabajo, tareas, sub-tareas, entregables, etc.

Tabla 3.5.1: Comienzo del proyecto.

Estructura de descomposición del trabajo			
Nivel 1	Nivel 2	Nivel 3	Nivel 4
Comienzo del proyecto	Describir el proyecto		
	Justificar el proyecto		
	Detallar las etapas y los procesos del proyecto		

Tabla 3.5.2: Organización y preparación.

Estructura de descomposición del trabajo			
Nivel 1	Nivel 2	Nivel 3	Nivel 4
Organización y preparación	Declaración de alcance	Redactar la declaración de alcance	
		Aprobar la declaración de	



		alcance	
	Estructura de descomposición del trabajo	Desarrollar la estructura de descomposición del trabajo	
	Planificación	Desarrollar el diagrama de Gantt	
		Analizar los costos	
		Analizar los tiempos	

Tabla 3.5.3: Llevando adelante el trabajo.

Estructura de descomposición del trabajo				
Nivel 1	Nivel 2	Nivel 3	Nivel 4	
Llevando adelante el trabajo	Hardware	Detallar los componentes del prototipo	Listar los componentes	
			Solicitar cotizaciones	
			Seleccionar distribuidores	
		Comprar el hardware		Comprar computadora Raspberry Pi B+
				Comprar transductor DS18B20
				Comprar módulo NEO6MV2
				Comprar módulo SIM908
				Comprar fuente 12-24VDC / 12VDC
				Comprar fuente 12VDC / 5VDC
				Comprar CP2102 UART / USB
				Comprar LED RGB
		Conectar el hardware		Comprar pack cable Dupont 20cm F/F
			Comprar tarjeta SD 8GB	
			Fuente 5V - Raspberry Pi	
			Fuente 12/24V - Fuente 5V y SIM908	
			Raspberry Pi – CP2102 - SIM908	
			SIM908 - Antena GPRS	
			Raspberry Pi – NEO6MV2	
			NEO6MV2 - Antena GPS	
	Comprar los elementos para ensamblar el prototipo		Raspberry Pi – DS18B20	
			Raspberry Pi – LED RGB	
			Comprar gabinete plástico	
			Comprar pack separadores	
		Comprar dos extensores U.fl a SMA		
		Comprar pack precintos		
Armar el producto prototipo		Comprar 2 conectores DIN 3P M/F		
		Comprar pack cable 1mm multifilamento		
		Comprar antena GPS y GPRS		
		Realizar perforaciones en el gabinete		
		Instalar los conectores		
Testear el producto prototipo		Pegar los separadores al gabinete		
		Instalar el hardware en el gabinete		
		Test de temperatura		
		Test de GPS		
		Test en campo San Fernando – Ezeiza		
Firmware	Seleccionar el lenguaje	Características principales del lenguaje		
	Instalar las librerías	Justificar la selección		
		Instalar Python		
		Instalar librería GPIO		



			Instalar librería UART	
			Instalar librería DS18B20	
			Instalar librería GPSD	
			Instalar librería NTPD	
		Programar las funciones de hardware	Cambiar el estado de los puertos GPIO	
			Encender y apagar los LED	
			Adquirir temperatura	
			Comunicación UART – NEO6MV2	
			Adquirir información GPS	
			Comunicación UART – SIM908	
			Crear un contexto GPRS	
			Enviar y recibir información por métodos HTTP	
			Enviar y recibir SMS	
			Enviar y recibir correo electrónico	
			Testear las funciones de hardware	Verificar control de puertos GPIO
				Verificar encendido y apagado de LED
		Verificar temperatura adquirida		
		Verificar información GPS		
		Verificar envío y recepción mediante HTTP		
		Verificar envío y recepción de SMS		
		Verificar envío y recepción de correo electrónico		
		Programar las funciones multiprocesamiento	Programar candados	
			Programar colas de datos	
			Programar subprocesos	
			Programar reanudación de subprocesos ante fallas	
			Programar el sistema de registro	
		Testear las funciones multiprocesamiento	Verificar el bloqueo de candados	
			Verificar escritura y lectura en las colas de datos	
			Verificar la reanudación de subprocesos ante fallas	
			Verificar el funcionamiento de los registros	
		Integrar las funciones en un programa final	Integrar las funciones de hardware en los subprocesos	
			Comunicar los subprocesos por colas de datos	
			Bloquear los subprocesos por candados	
			Programar la generación de registros por subproceso	
			Programar el manejo de fallas por subproceso	
		Testear el programa final	Verificar la comunicación entre los subprocesos	
Verificar el bloqueo de los subprocesos				
Verificar la generación de registros de los subprocesos				
Verificar el manejo de fallas de los subprocesos				
Software	Seleccionar el software	Características principales del software		
		Justificar la selección		
	Instalar el software	Instalar Linux Raspbian		
		Instalar SSH		
		Instalar FTP		
	Configurar el software	Instalar Supervisor		
		Configurar Supervisor		
		Configurar GPSD		
		Configurar NTPD		



	Desarrollo Web	Seleccionar el servidor web	Solicitar cotizaciones
			Contratar servicio
		Desarrollar las funciones web básicas	Programar en PHP scripts para métodos HTTP
			Programar en PHP el acceso a la base de datos MySQL
		Desarrollar la interfaz web para los usuarios	Diseñar la página web con HTML y CSS
			Programar en Javascript el uso de Google Maps API
			Programar en Javascript el uso de Plotly API
		Testear la interfaz web	Visualizar la información GPS y temperatura de la prueba en campo

Tabla 3.5.4: Cierre del proyecto.

Estructura de descomposición del trabajo			
Nivel 1	Nivel 2	Nivel 3	Nivel 4
Cierre del proyecto	Redactar el informe del proyecto		
	Redactar las conclusiones del proyecto		

3.6. Planificación

La planificación es el proceso de cuantificar el tiempo y los recursos que el proyecto costará. La finalidad es crear un plan de proyecto que pueda ser utilizado por el gestor del proyecto para acompañar el progreso. Para determinar la duración de cualquier proyecto, previamente se deben conocer las siguientes informaciones:

- *Tareas*: Las necesarias para la realización del proyecto.
- *Secuencia*: El orden en el cual se deben desarrollar las actividades.
- *Duración*: Cuanto tiempo demora cada actividad individual.

La planificación permite estimar con mayor exactitud los costos y recursos necesarios para cada tarea, y optimizarlos de acuerdo a la duración total para cumplir con los objetivos. Una vez establecida y aceptada la planificación, esta se convierte en la línea base de seguimiento del proyecto. El progreso será comparado contra la planificación a lo largo de toda la vida del proyecto.



Inicialmente el espectro del proyecto es definido y los métodos para completar el proyecto son determinados. A partir de la estructura de descomposición del trabajo se obtienen las tareas a realizar en el proyecto, y se puede establecer la duración de cada tarea. Las dependencias lógicas que existen entre las tareas fuerzan la secuencia en la cual se deben llevar a cabo. Finalmente, se establecen los recursos necesarios para la realización de cada tarea, y se calculan los costos.

Para representar los elementos a tener en cuenta al desarrollar la planificación se utiliza el diagrama de Gantt. El diagrama de Gantt es una herramienta grafica que ilustra el cronograma de un proyecto. Expone el tiempo de dedicación previsto y los recursos asignados para cada tarea, y las dependencias lógicas entre las mismas.

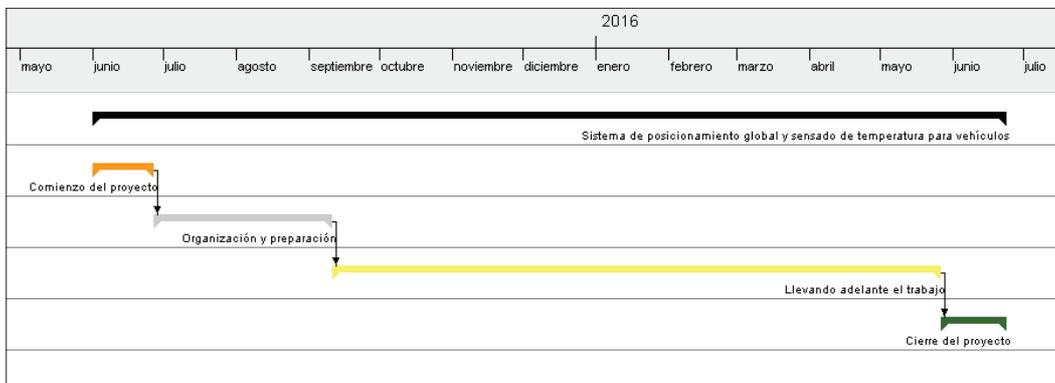


Figura 3.6.1: Diagrama de Gantt del proyecto dividido en cuatro etapas.

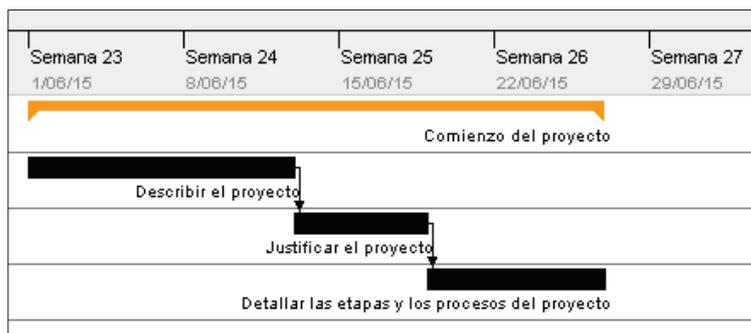




Figura 3.6.2: Diagrama de Gantt de la etapa “Comienzo del proyecto”.

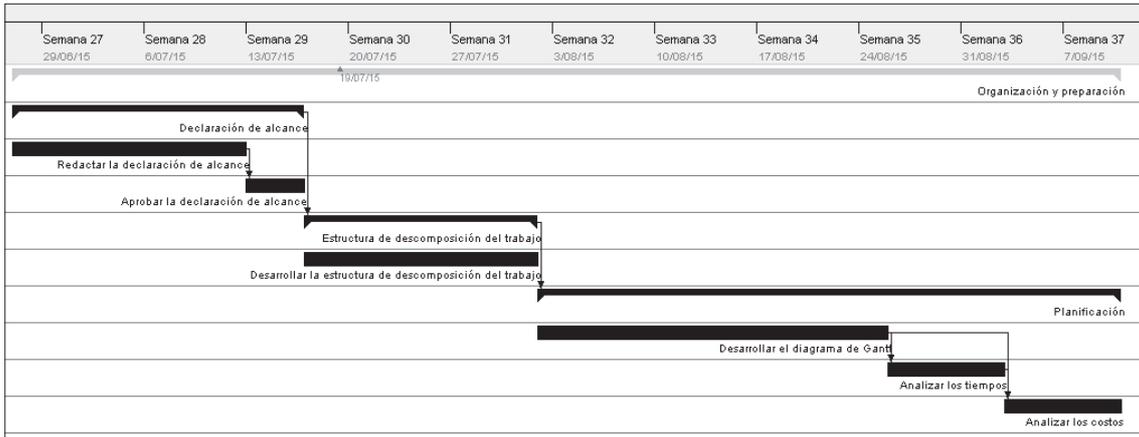


Figura 3.6.3: Diagrama de Gantt de la etapa “Organización y preparación”.

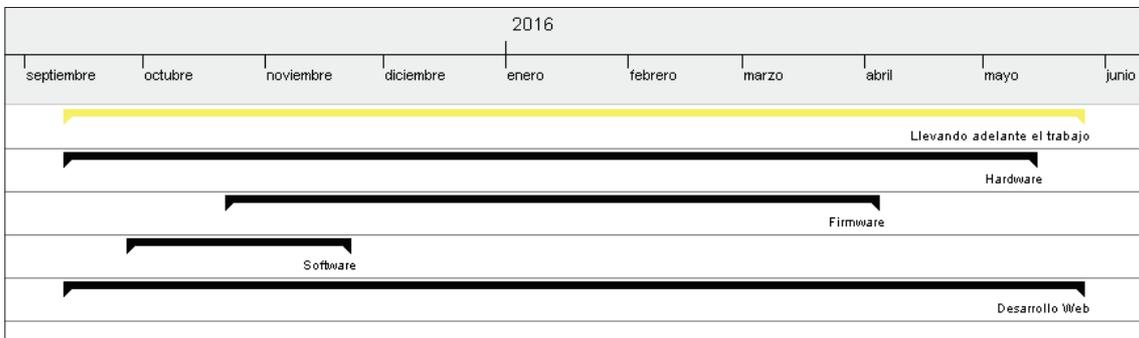


Figura 3.6.4: Diagrama de Gantt de la etapa “Llevando adelante el proyecto”.

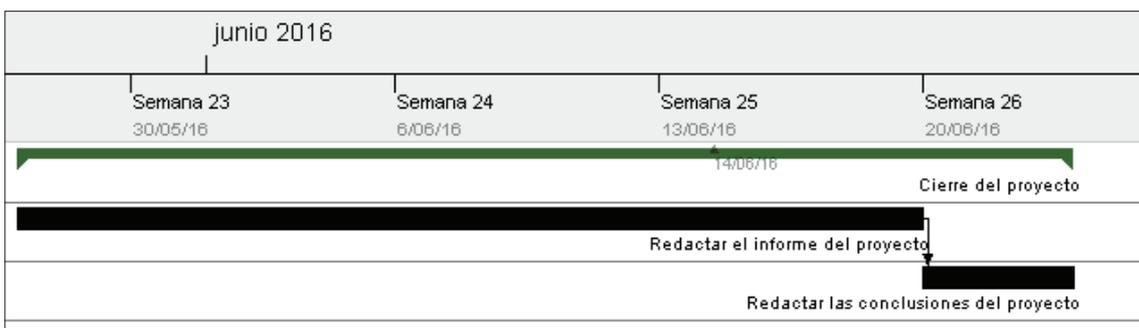


Figura 3.6.5: Diagrama de Gantt de la etapa “Cierre del proyecto”.



El proyecto está compuesto de 140 tareas, y su duración se estimó en un año, utilizando 5 recursos humanos, teniendo la disponibilidad de todos los recursos materiales necesarios listados en la declaración de alcance.

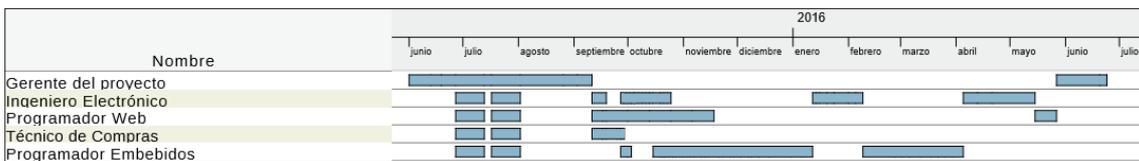


Figura 3.6.6: Recursos asignados a las tareas del proyecto.



4. Sistema Global de Navegación por Satélite (GNSS)

4.1. Introducción

Un sistema global de navegación por satélite (Global Navigation Satellite System - GNSS) es una constelación de satélites que transmite señales de radiofrecuencia utilizadas para posicionar y localizar al receptor en cualquier parte del globo terrestre. Estos sistemas proveen posicionamiento geoespacial autónomo, permitiendo al receptor determinar longitud, latitud, altitud y tiempo local con precisión. En la actualidad existen tres sistemas completamente operativos: NAVSTAR GPS (estadounidense), GLONASS (ruso) y GALILEO (europeo). Los sistemas COMPASS (chino), IRNSS (indio) y QZSS (japonés) todavía no se encuentran funcionales.

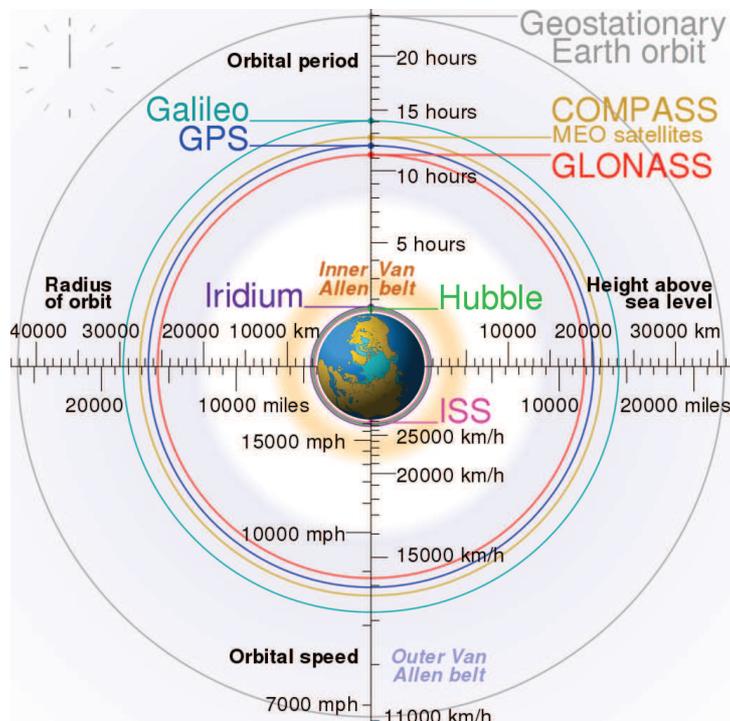


Figura 4.1.1: Órbitas de los sistemas GNSS actuales.

4.2. Arquitectura



Los sistemas GNSS consisten de 3 grandes componentes, el segmento espacial, el segmento de control y el segmento de usuario.

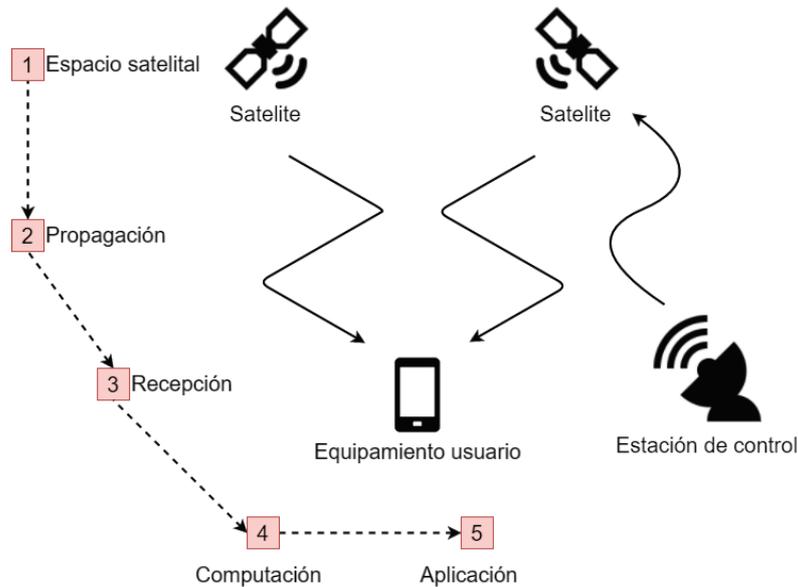


Figura 4.2.1: Segmentos de un sistema GNSS.

El segmento espacial consiste de los satélites orbitando a aproximadamente 20.000 km sobre el nivel del mar. Cada sistema GNSS tiene su propia constelación de satélites, ordenados en orbitas para proveer la cobertura deseada. Cada satélite transmite una señal de radiofrecuencia para identificarse, además de proveer su tiempo, orbita y estado.

El segmento de control comprende una red de estaciones de control, monitoreo y comunicación. La estación de control ajusta los parámetros de la órbita del satélite y los relojes de alta precisión cuando sea necesario para mantener la exactitud. Las estaciones de monitoreo, usualmente instaladas sobre una amplia área geográfica, monitorean los estados y las señales de los satélites, para luego informar a la estación de control. La estación de control analiza las señales y transmite las correcciones a los satélites a través de las estaciones de comunicación.



El segmento de usuario consiste de equipamiento que procesa la señal recibida de los satélites GNSS y la utiliza para obtener información espacial y temporal. Los equipos comprenden desde teléfonos móviles, receptores de autos, hasta equipamiento sofisticado, utilizado para aplicaciones de topografía y cartografía.

4.3. Computación

Al conocer la posición de cuatro satélites y la distancia exacta hacia cada uno es posible determinar la ubicación del receptor. Por cada satélite siendo rastreado, el receptor calcula cuanto tiempo demora la señal del satélite en alcanzarlo, de la forma:

Multiplicando el tiempo de propagación por la velocidad de la luz se obtiene la distancia al satélite. Por cada satélite siendo rastreado, el receptor conoce la ubicación del satélite al momento de transmitir (ya que el satélite transmite además su órbita). Usando métodos geométricos como la trilateración, el receptor calcula su posición. Según el método de trilateración son necesarias tres distancias a puntos conocidos para poder calcular la posición de un objeto en sus tres dimensiones.

Anteriormente se mencionó que son necesarios cuatro satélites como mínimo para calcular la ubicación del receptor, y esto se debe a que los relojes de sincronización del receptor no son tan precisos como los relojes a bordo de los satélites. La mayoría de los relojes utilizados en los receptores son cristales de cuarzo, mientras que los utilizados en satélites son relojes atómicos. Esto introduce un error que fuerza la necesidad de ajustar el reloj del receptor, siendo necesario para esto conocer la posición de, por lo menos, un cuarto satélite. El receptor adelanta o atrasa su reloj de forma que las posiciones calculadas converjan a la misma posición.



4.4. Recepción

Como se discutió previamente, el receptor necesita por lo menos cuatro satélites para obtener su posición. El uso de más satélites, si están disponibles, van a mejorar considerablemente la precisión de la ubicación. Sin embargo, la habilidad del receptor para hacer uso de los satélites adicionales puede estar limitada por su poder computacional. La forma en que el receptor hace uso de los satélites adicionales es generalmente de propiedad intelectual del fabricante del receptor. Los receptores varían en función de cual constelación o constelaciones rastrean, y cuantos satélites pueden utilizar simultáneamente.

Seleccionar la antena de recepción GNSS correcta es un punto clave para optimizar la performance del sistema. La antena debe coincidir con las capacidades y especificaciones del receptor, así como también cumplir las especificaciones propias de la aplicación (tamaño, peso, condiciones ambientales, estrés mecánico, etc.).

4.5. Señales Satelitales

Las señales de los satelitales transmiten su información en las bandas de frecuencia L1, L2 y L5, utilizando acceso múltiple por división de código (Code Division Multiple Access - CDMA) basado en la tecnología de espectro expandido.

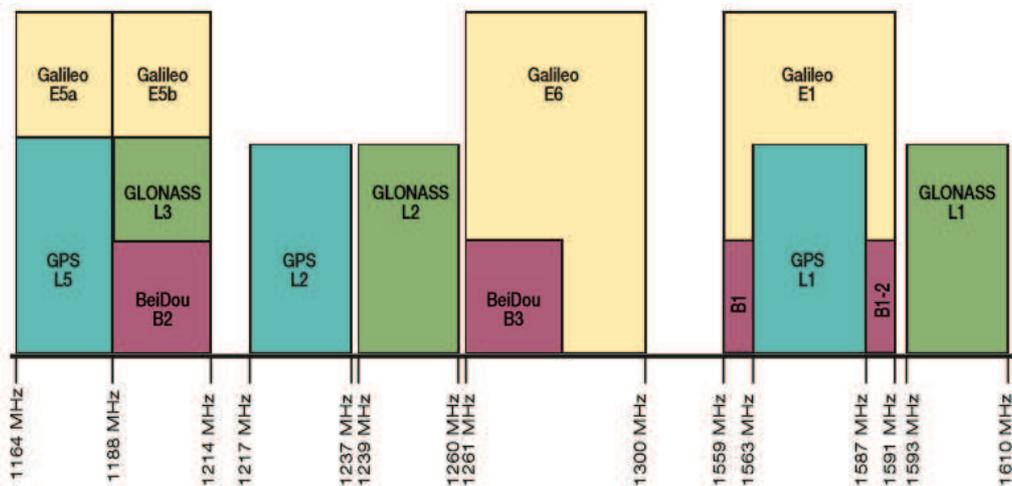


Figura 4.5.1: Espectro de las bandas L utilizadas por los sistemas GNSS.

Las señales satelitales, aunque se encuentren en la misma banda de frecuencia, son moduladas por una secuencia digital pseudoaleatoria única, o código. Cada satélite utiliza un código pseudoaleatorio diferente, conocido por los receptores. Esto les permite a los receptores sincronizarse con la señal CDMA para un satélite en particular. Las señales CDMA son recibidas en niveles muy bajos, sin embargo, a través de la correlación de código, el receptor es capaz de recuperar las señales y la información que contienen.

4.6. Errores

Un receptor GNSS calcula su posición basado en la información recibida de los satélites. Sin embargo, existen varias fuentes de errores que pueden causar que el cálculo de la posición sea incorrecto. Es esencial conocer el tipo de error y el método para mitigarlo, para calcular con precisión la posición del receptor. Se destacan las fuentes de contribución al error de mayor peso:

- *Reloj del satélite (± 2 m)*: Los relojes atómicos en los satélites GNSS son muy precisos, pero se desvían una pequeña cantidad. Una pequeña imprecisión en los relojes satelitales resulta en un error en la posición



calculada por el receptor. El reloj en el satélite es monitoreado por el sistema de control terrestre GNSS y es comparado contra un reloj aún más preciso. De esta forma, el satélite provee al receptor con un estimativo del desvío de su reloj. La estimación tiene una precisión de ± 2 metros.

- *Errores de órbita (± 2.5 m)*: Los satélites viajan en conocidas y muy precisas orbitas. Sin embargo, las orbitas varían en una pequeña cantidad. Una pequeña imprecisión en las orbitas satelitales resulta en un error en la posición calculada por el receptor. El sistema de control terrestre GNSS continuamente monitorea la órbita de los satélites, y si cambia, envía correcciones a los satélites. Existen igualmente pequeños errores en la órbita que resultan en ± 2.5 metros de error.
- *Retardo ionosférico (± 5 m)*: La ionosfera es la capa de la atmosfera entre los 80 km y 600 km sobre la tierra. Esta capa contiene partículas eléctricamente cargadas (iones). Los iones retardan las señales satelitales y causan una cantidad significativa de error en la posición, típicamente ± 5 metros, pero puede ser mayor en periodos de gran actividad ionosférica. Es muy difícil predecir cuanto retardo ionosférico impacta sobre el cálculo de la ubicación, ya que depende de la actividad solar, la época del año, la ubicación y la hora. También varía de acuerdo a la frecuencia de la señal que la atraviesa, por lo que los receptores multibanda pueden comparar las diferencias entre los cálculos de dos bandas y determinar el retardo ionosférico para remover el error de la posición calculada. Para los monobanda se utilizan en los cálculos modelos de la ionosfera, pero no son tan efectivos.
- *Retardo troposférico (± 0.5 m)*: La troposfera es la capa de la atmosfera más cercana a la superficie terrestre. Las variaciones en el retardo troposférico son causadas por la cambiante humedad, temperatura y presión atmosférica en la troposfera. Los receptores GNSS utilizan modelos de la troposfera para estimar el error causado por el retardo.



- *Ruido del receptor ($\pm 0.3 m$)*: El ruido del receptor hace referencia al error en la posición causado por el hardware y software del sistema receptor GNSS.
- *Propagación multicamino ($\pm 1 m$)*: La propagación multicamino ocurre cuando una señal GNSS es reflejada en un objeto, como por ejemplo la pared de un edificio. Como la señal reflejada viaja una distancia mayor para llegar a la antena, la señal reflejada llega al receptor con un pequeño retardo. Esta señal retardada puede causar que el receptor calcule una posición incorrecta. La forma más simple de reducir los errores multicamino es colocando la antena GNSS en una ubicación que este alejada de superficies reflectivas. Cuando esto no es posible, el receptor se encarga de los errores multicamino de largo retardo, y la antena se encarga de los errores multicamino de corto retardo.

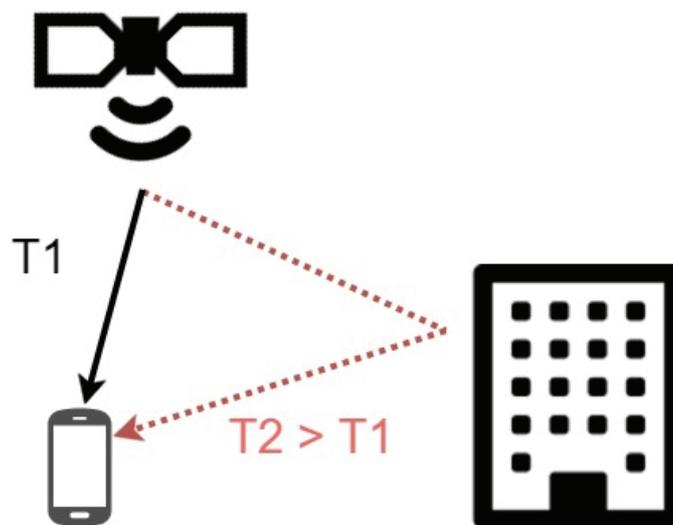


Figura 4.6.1: Propagación multicamino.



5. Sistema Global para las Comunicaciones Móviles (GSM)

5.1. Introducción

El sistema global para las comunicaciones móviles (Global System for Mobile communications - GSM) es un estándar de telefonía móvil digital desarrollado por el Instituto Europeo de Estándares de Telecomunicaciones (European Telecommunications Standards Institute - ETSI) para describir los protocolos de redes de telefonía celular digital de segunda generación 2G utilizados por teléfonos móviles. Las redes GSM se desarrollaron como un reemplazo de las redes analógicas de primera generación 1G. El primer propósito de la red fue optimizar la telefonía de voz full dúplex, aunque luego se expandió para incluir comunicaciones de datos, mediante el servicio general de paquetes vía radio (General Packet Radio Service - GPRS). GSM opera en las bandas de frecuencia 900 MHz y/o 1800 MHz.

5.2. Arquitectura

Como todas las redes modernas, GSM utiliza una estructura celular. El objeto de una estructura celular es dividir el rango de frecuencias disponibles, para asignar por partes el espectro de frecuencia a las estaciones base de transmisión, y reducir el rango de cada estación base para reutilizar las escasas frecuencias tan seguido como sea posible. Una de las metas de la planificación de redes es reducir la interferencia entre las diferentes estaciones base, que depende en gran magnitud de la distancia entre las antenas de estaciones con el mismo canal de frecuencia asignado.

Sin embargo, reutilizar frecuencias presenta desventajas para una red celular:

- Incrementar el número de estaciones base incrementa el costo de infraestructura.



- Las redes celulares requieren que mientras la estación móvil se desplaza, una llamada activa sea entregada de una célula a otra. Este proceso se lo conoce como traspaso (Handover).
- La red tiene que conocer la ubicación aproximada de cada estación móvil, para poder entregar una llamada entrante.
- El sistema requiere una comunicación extensiva entre la estación móvil y la red, así como entre los distintos elementos de la red. Se refiere a estas comunicaciones como señalización (Signaling).

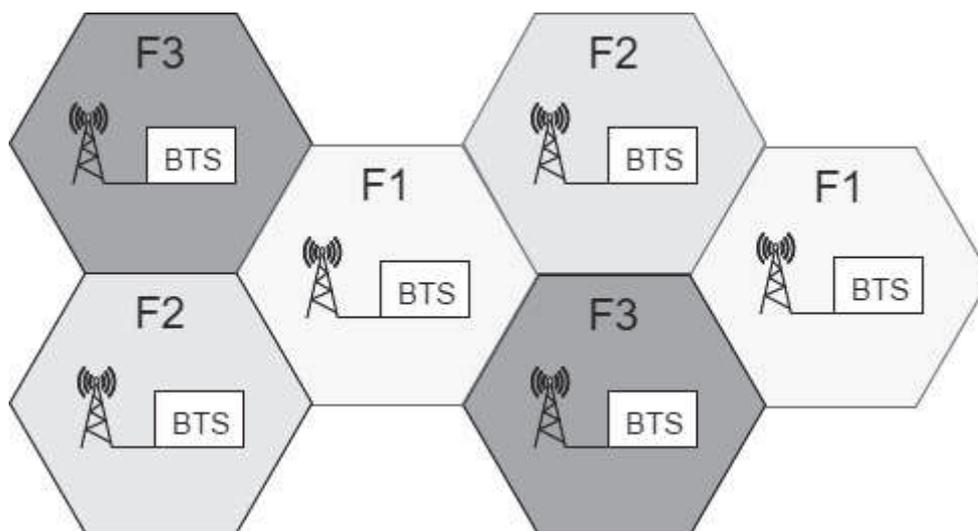


Figura 5.2.1: Estructura celular de la red GSM.

5.3. Estructura

Una red GSM está compuesta de múltiples elementos:

- *Estación móvil (Mobile Station - MS)*: está compuesta por todo el equipamiento de usuario y el software necesario para la comunicación dentro de la red GSM. Todas las terminales GSM poseen una identidad internacional de equipo móvil (International Mobile Station Equipment Identity – IMEI).



- *Módulo de identificación de abonado (Subscriber Identity Module - SIM):* es un circuito integrado que almacena la identidad internacional del abonado a un móvil (International Mobile Subscriber Identity - IMSI), utilizado para identificar y autenticar a los suscriptores.
- *Estación base de transmisión (Base Transceiver Station - BTS):* se encarga de las tareas relacionadas con la radiocomunicación entre la red y la MS.
- *Controlador de estación base (Base Station Controller - BSC):* realiza las tareas centrales y el control del subsistema, referido como subsistema de estación base (Base Station Subsystem - BSS). El BSS comprende la BSC y todas las BTS conectadas, usualmente agrupadas por regiones.
- *Unidad de tasa de transcodificación y adaptación (Transcoding Rate and Adaptation Unit - TRAU):* realiza la compresión de datos, principalmente la compresión y descompresión de datos de llamada.
- *Central de conmutación móvil (Mobile Services Switching Center - MSC):* su principal función es interconectar a los usuarios de la red móvil entre sí. Inicializa, termina y canaliza las llamadas a través de las distintas BSC.
- *Registro de ubicación base (Home Location Register - HLR):* es un repositorio que contiene los datos de los suscriptores. En una red es necesario al menos un HLR.
- *Registro de ubicación de visitante (Visitor Location Register - VLR):* es un repositorio que contiene también información de los suscriptores, pero a diferencia del HLR, solo mantiene la información mientras el suscriptor recorre el área MSC.
- *Registro de identificación de equipo (Equipment Identity Register - EIR):* En esta base se registra el equipamiento utilizado por los usuarios a partir del identificador IMEI, de forma de evitar que equipamiento perdido o robado pueda ser utilizado en forma fraudulenta.



Todos estos elementos combinados forman una red móvil terrestre pública (Public land mobile network - PLMN).

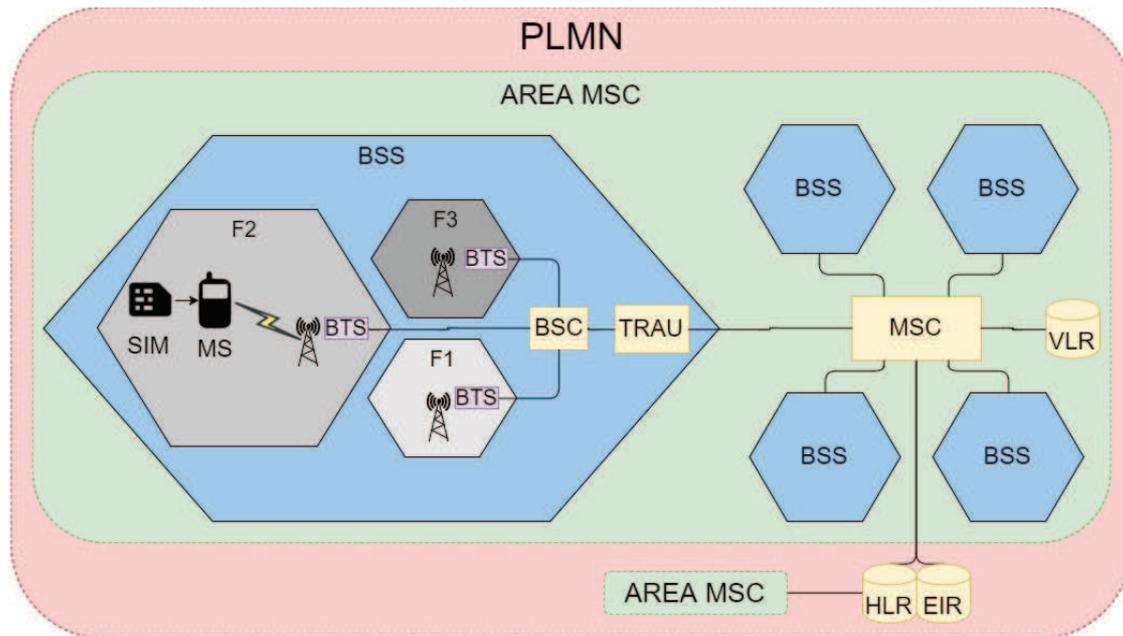


Figura 5.3.1: Estructura PLMN.

5.4. Servicio General de Paquetes vía Radio (GPRS)

El servicio general de paquetes vía radio (General Packet Radio Service - GPRS) es una extensión de GSM creada para la transmisión de datos mediante conmutación de paquetes. Fue la primera arquitectura que permitió a los suscriptores tasas de transmisión de alta velocidad (normalmente entre 56 y 114 kbit/s) y el uso de aplicaciones de datos desde sus MS. Utiliza canales de acceso múltiple por división de tiempo (Time Division Multiple Access - TDMA), por ejemplo, los que se encuentran dentro del sistema GSM.

Para establecer una comunicación GPRS, la MS tiene que registrarse en la red (proceso de registración). El suscriptor debe configurar el nombre del punto de acceso (Access Point Name – APN) para que el dispositivo pueda acceder a la



red. Posteriormente, la red le asigna una dirección IP, lo que permite activar un contexto de paquete de datos (Packet Data Protocol – PDP) y utilizar el protocolo de internet (Internet Protocol – IP) para el intercambio de datos.

Una registración exitosa produce un perfil de servicio valido, similar al almacenado en el registro HLR. Al registrarse se comprueba la identidad de la MS (proceso de autenticación), y se comprueba el perfil de suscriptor en el registro VLR, para dar acceso a los servicios permitidos para ese usuario (proceso de autorización). Además, la red debe comprobar si se encuentran disponibles los recursos necesarios para cumplir con la solicitud de cada suscriptor en forma dinámica (control de admisión) y monitorear el consumo de paquetes de datos (proceso de facturación).



Figura 5.4.1: Funciones de control de acceso a la red GPRS.



6. Desarrollo Técnico del Proyecto

6.1. Hardware

El hardware utilizado en el sistema se escogió de forma que respete las reglas de diseño de proyecto. Se seleccionó hardware libre o de código abierto, es decir, aquel hardware cuyas especificaciones y diagramas esquemáticos son de acceso público de forma gratuita. Además, se buscó que el hardware tenga buen soporte y continuidad de fabricación. Se seleccionaron los siguientes componentes para el desarrollo del prototipo:

- *Raspberry Pi B+*: Es una computadora pequeña o de placa reducida, de bajo costo, desarrollado en Reino Unido. El sistema tiene el tamaño aproximado de una tarjeta de crédito, CPU/GPU ARM11 700 MHz, 512 MB de memoria RAM, conectividad USB, video HDMI, Ethernet, lector de tarjeta SD y un puerto GPIO con múltiples interfaces (I²C, UART, SPI, PCM, DPI, GPCLK, JTAG, 1-WIRE, SDIO). Se seleccionó para el proyecto debido a su gran portafolio de soluciones, su bajo costo y su versatilidad a la hora de programar, ya que cuenta con soporte para C, C++, Python y Java, entre otros. La computadora Raspberry Pi utiliza Raspbian, un sistema operativo basado en el núcleo GNU/Linux. Esta distribución tiene compatibilidad con todo el hardware disponible, además de otorgar la posibilidad de utilizar cualquier software existente para GNU/Linux.

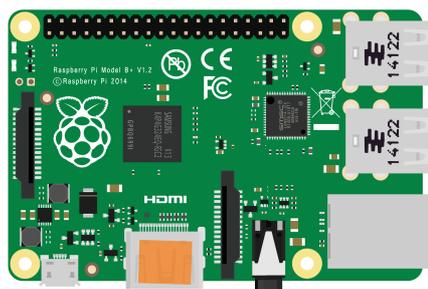




Figura 6.1.1: Diagrama esquemático Raspberry Pi.

- *GPS GY-NEO6MV2*: Es el modulo encargado de obtener la información de posicionamiento global, cálculo de velocidad, dirección, inclinación, además de sincronizar la fecha y hora del Raspberry Pi (sincronización por pulsaciones PPS). Es un módulo económico desarrollado por u-blox con distintos protocolos de comunicación (NMEA, UBX, RTCM) ideal para desarrollo de proyectos. Se lo selecciono por su bajo costo, compatibilidad con Raspberry Pi, gran disponibilidad y su exactitud.
- *SimCom SIM908 GPRS*: Es el modulo encargado de transmitir la información desde el Raspberry Pi al servidor web. Es un módulo de alta performance y bajo consumo desarrollado por SimCom, que además cuenta con un sistema de posicionamiento global. El sistema de posicionamiento integrado no fue utilizado debido a que las redes de telefonía móvil actuales no son muy estables, y el dispositivo debe reiniciarse con una frecuencia mayor que el tiempo de pre-calentamiento del GPS en frio, usualmente 30 segundos. Su reemplazo por cualquier otro modelo de SimCom es intuitivo y esta es la principal ventaja por la cual se lo escogió. El módulo recibe comandos AT y opera en consecuencia. Otra característica importante es que brinda la posibilidad de comunicación mediante protocolos de Internet: TCP/IP, HTTP, FTP y SMTP los cuales son utilizados en todo tipo de aplicaciones. El modelo SIM908 ya se encuentra discontinuado, pero su reemplazo, el SIM800, es compatible, además que su precio es menor.
- *Transductor de temperatura DS18B20*: Es un termómetro digital con resolución programable de 9 a 12 bits. Se comunica mediante el protocolo 1-WIRE, que posibilita la alimentación directamente de la línea de datos, eliminando la necesidad de una fuente de alimentación externa. Cada DS18B20 tiene un código de serie único que permite que múltiples sensores se conecten al mismo bus. Mide temperaturas dentro



del rango -55°C a $+125^{\circ}\text{C}$, y tiene una precisión de 0.5°C para las mediciones que se encuentran en el rango -10°C a $+85^{\circ}\text{C}$. El transductor se encuentra alojado típicamente en una vaina metálica ideal para ambientes húmedos o con presencia de agua.

- *Fuente de alimentación switching DC/DC 12V-24V/12V*: Es la primera etapa que se encuentra entre la entrada de alimentación al sistema y el resto de los componentes. Se buscó que la tensión de entrada variara entre los rangos de baterías de un auto convencional y un camión de transporte. Se utiliza esta primera fuente de 12V para alimentar a la segunda etapa y, por otra parte, al módulo de comunicaciones que tiene picos de consumo al momento de transmitir.
- *Fuente de alimentación switching DC/DC 12V/5V*: Es la segunda etapa que tiene como propósito alimentar la computadora Raspberry Pi, que luego alimenta al resto de los componentes.
- *Módulo UART – USB*: Se utiliza un módulo UART – USB ya que la computadora Raspberry Pi solo cuenta con un solo puerto UART, y dos son los requeridos por la aplicación, uno para el modulo GPS y otro para el módulo de comunicaciones GPRS.
- *Antena GPS y antena GPRS*: Son las dos antenas necesarias para incrementar la ganancia de transmisión y recepción de información. Se ofrece una solución exterior con imanes para adherir al techo del vehículo, la cual proporciona gran conectividad y calidad, y otra solución en la cual las antenas van por dentro del equipo, proporcionando la posibilidad de que el sistema no sea visible desde el exterior del vehículo.



del código. Python es un lenguaje de programación interpretado, de propósito general, dinámico, gratuito y de alto nivel. Su filosofía de diseño enfatiza la legibilidad del código, y su sintaxis permite a los programadores expresar conceptos en menores líneas de código que C++ o Java.

Se desarrolló el programa principal buscando que la concurrencia sea factible, especialmente utilizando la librería de multiprocesamiento de Python. La concurrencia es la propiedad de descomposición de un programa en unidades, las cuales, si son ejecutadas en orden aleatorio, el resultado final permanece igual. Aplicar la concurrencia permite la ejecución paralela de las unidades concurrentes, lo que se refleja como mayor seguridad y control en la búsqueda del resultado final. La librería de multiprocesamiento de Python permite la creación y el control de subprocesos, proporciona elementos de comunicación como son las pipas y las colas, elementos de sincronización como semáforos y candados, espacios de memoria compartida, entre otros.

Al utilizar multiprocesamiento el código se vuelve más extenso y su ejecución más lenta, sin embargo, las ventajas son notables. Por ejemplo, permite limitar el acceso a un recurso mediante el uso de semáforos o candados, controlar de forma exterior los subprocesos, reiniciarlos si se produce una excepción y no detener la ejecución del programa, conservando el contexto para no descartar información. Para comunicar a los subprocesos entre sí de forma segura se utilizan colas de datos, y para limitar el acceso a los recursos de hardware se utilizan candados. Se dividió al programa principal en tres subprocesos, cada uno de los cuales controla un periférico del sistema:

- *Subproceso NEO6MV2*: Este subproceso se encarga de limitar el acceso al módulo de posicionamiento global. El subproceso se comunica con el módulo para obtener la información GPS, luego verifica su calidad, ajusta el reloj interno del sistema Linux y finalmente transmite la información al subproceso de comunicaciones GPRS a través de una cola de datos FIFO.



- *Subproceso DS18B20*: Este subproceso se encarga de limitar el acceso al transductor de temperatura. El subproceso se comunica con el transductor de temperatura para obtener la información, luego se comunica con el LED testigo para encenderlo de acuerdo al rango en el cual se encuentra la medición, y finalmente transmite la información al subproceso de comunicaciones GPRS a través de una cola de datos FIFO.
- *Subproceso SIM908*: Este subproceso se encarga de limitar el acceso al módulo de comunicaciones GPRS. El subproceso se comunica con el módulo para generar el contexto necesario para una comunicación GPRS. Lee la información de las colas de datos FIFO. La información se transmite a un servidor mediante los métodos HTTP GET y POST.

Todos los subprocesos son monitoreados desde el proceso principal, y cuando se detecta que han finalizado se vuelven a ejecutar. Además, se programaron rutinas de salida para todos los subprocesos, para liberar los candados cuando se produce algún error. La información producida por el sistema es capturada por el módulo registrador de Python: códigos de errores, advertencias, excepciones, depuración, además de posición y temperatura. Cada entrada es almacenada con fecha y hora en la memoria SD del sistema, en un archivo con tamaño limitado que descarta la información antigua para hacer lugar a la información actual.

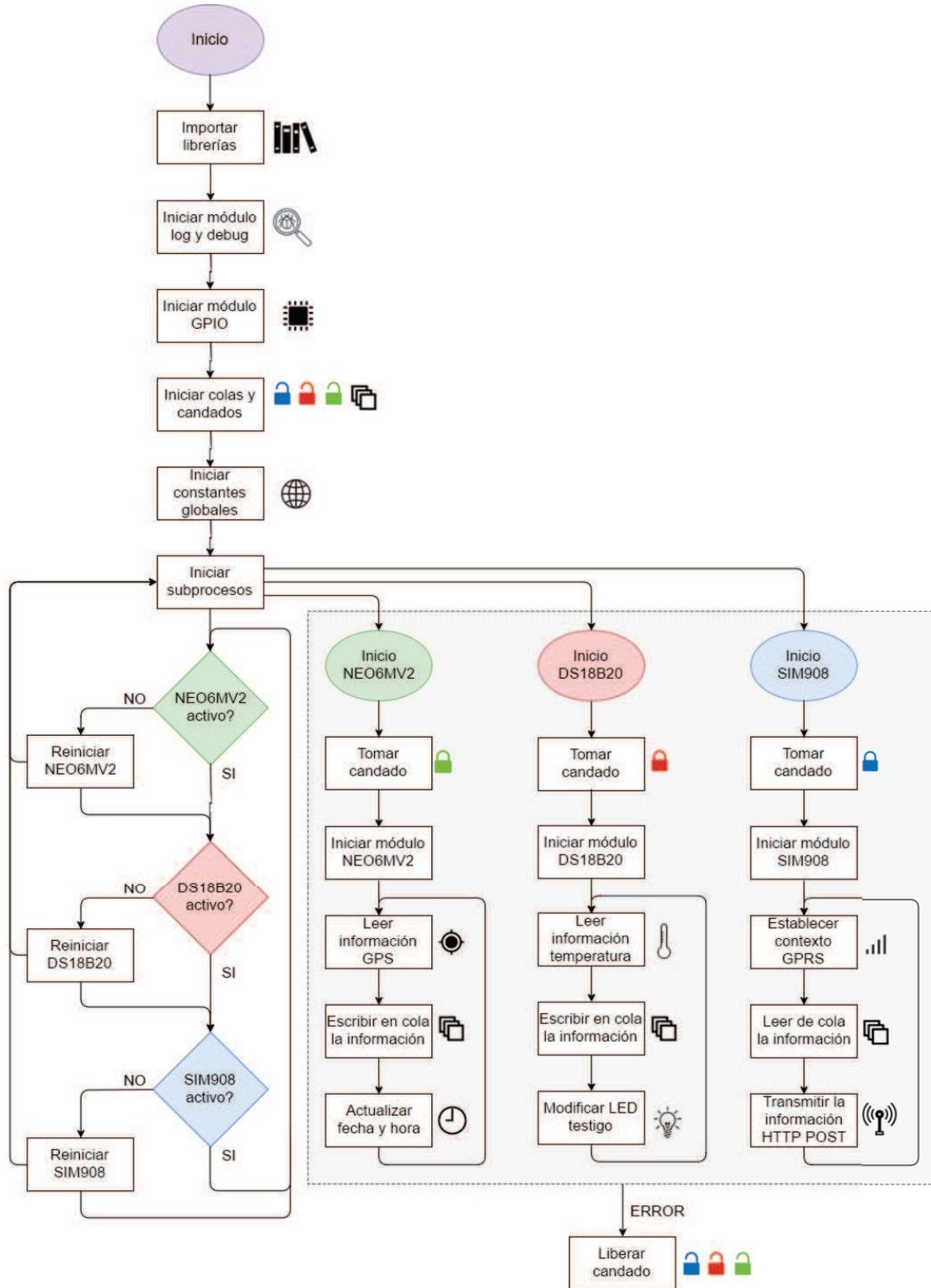


Figura 6.2.1: Diagrama de flujo del firmware.



6.3. Software

Considerando que el desarrollo del proyecto se basa en una computadora Raspberry Pi, se seleccionó como sistema operativo la distribución GNU/Linux Raspbian, la cual se encuentra basada en Debian. GNU/Linux es un sistema operativo basado en Unix, ensamblado bajo el modelo de software libre y de código abierto de distribución gratuita, basado en la colaboración para su desarrollo. La distribución Raspbian se encuentra optimizada para el hardware disponible de la computadora Raspberry Pi, es la más utilizada y por lo tanto la que tiene mayor soporte, además de que es ampliamente recomendada por la fundación Raspberry Pi.

Para comprender mejor el funcionamiento del sistema operativo GNU/Linux, se puede pensar en dos niveles. En la cima se encuentra el espacio de usuario (User Space). En esta capa es donde las aplicaciones del usuario (User Applications) son ejecutadas. Por debajo del espacio de usuario se encuentra el espacio del núcleo (Kernel Space), donde vive el núcleo Linux (Kernel).

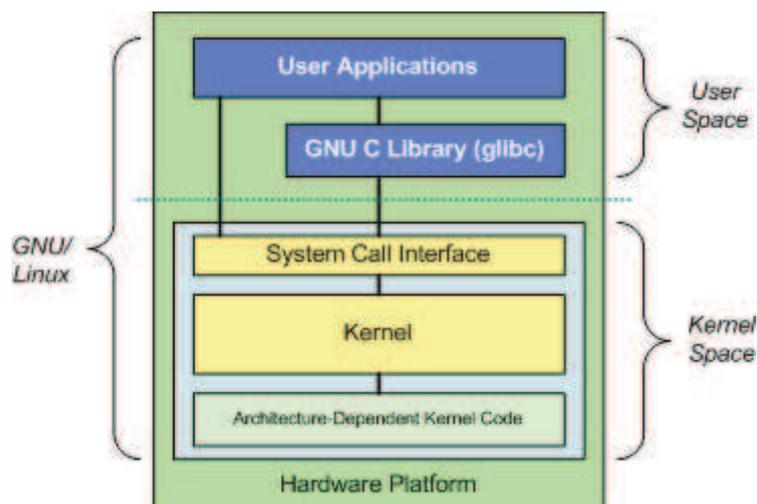


Figura 6.3.1: Arquitectura fundamental del sistema operativo GNU/Linux.



En el espacio de usuario también se encuentra la librería GNU C Library (glibc). Esta librería provee la interfaz de llamada al sistema (System Call Interface - SCI) que se conecta al núcleo y proporciona los mecanismos para la transición entre las aplicaciones del usuario y el núcleo. Este vínculo es importante porque el núcleo y las aplicaciones del usuario ocupan diferentes espacios de direcciones protegidas.

El núcleo de Linux puede dividirse en tres grandes niveles. En la parte superior esta la interfaz de llamada al sistema, que implementa las funciones básicas como leer y escribir. Debajo se encuentra el código del núcleo independiente de la arquitectura (Kernel). Este código es común a todos los tipos de arquitecturas de procesadores soportados por Linux. Finalmente, en el nivel inferior se encuentra el código del núcleo dependiente de la arquitectura (Architecture-Dependent Kernel Code), que hace de código específico de la plataforma.

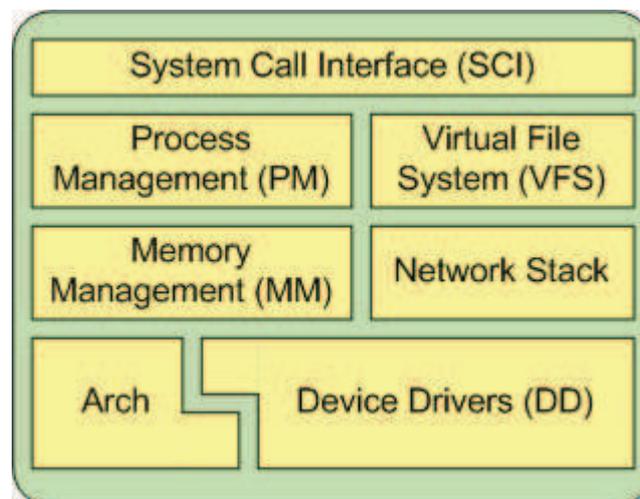


Figura 6.3.2: Perspectiva de la arquitectura del núcleo Linux.

El núcleo de Linux está conformado por distintos subsistemas, algunos de los más relevantes son:



- *Interfaz de llamada al sistema (System Call Interface - SCI)*: es una capa delgada que proporciona los medios para realizar llamadas a funciones desde el espacio de usuario al núcleo Linux.
- *Gerencia de procesos (Process Management - PM)*: el módulo está enfocado en la ejecución de procesos y en la necesidad de compartir el CPU entre los procesos activos. Para esto implementa un algoritmo de planificación que opera a tiempo constante.
- *Gerencia de memoria (Memory Management - MM)*: otro de los recursos importantes controlado por el núcleo de Linux es la memoria. La memoria es administrada en páginas, usualmente de 4KB. Linux incluye los medios para gestionar la memoria disponible, así como los mecanismos de hardware para las asignaciones físicas y virtuales.
- *Sistema de archivos virtual (Virtual File System - VFS)*: proporciona una interfaz de abstracción entre la interfaz de llamada al sistema y los sistemas de archivos soportados por el núcleo Linux.
- *Pila de red (Network Stack - NS)*: sigue una arquitectura en capas modelada a partir de los protocolos de red. El protocolo IP es el protocolo principal que se encuentra por debajo del protocolo TCP. Por encima del TCP se encuentra la capa de sockets, que se invoca a través del SCI.
- *Código basado en la arquitectura (Arch - AR)*: son los elementos que deben considerar la arquitectura para el normal funcionamiento del equipo y para su eficiencia.
- *Controladores de dispositivos (Device Drivers - DD)*: es el código que permiten que un dispositivo de hardware particular sea utilizable en el entorno Linux.

Sobre el sistema operativo Linux se utilizó la aplicación Supervisor, que proporciona un sistema de control de procesos. Supervisor permite al usuario monitorear y controlar procesos en sistemas operativos tipo UNIX. Proporciona



muchas opciones como iniciar los procesos en el arranque, reiniciar los procesos fallidos y la creación automática de registros estándar de salida y de errores.

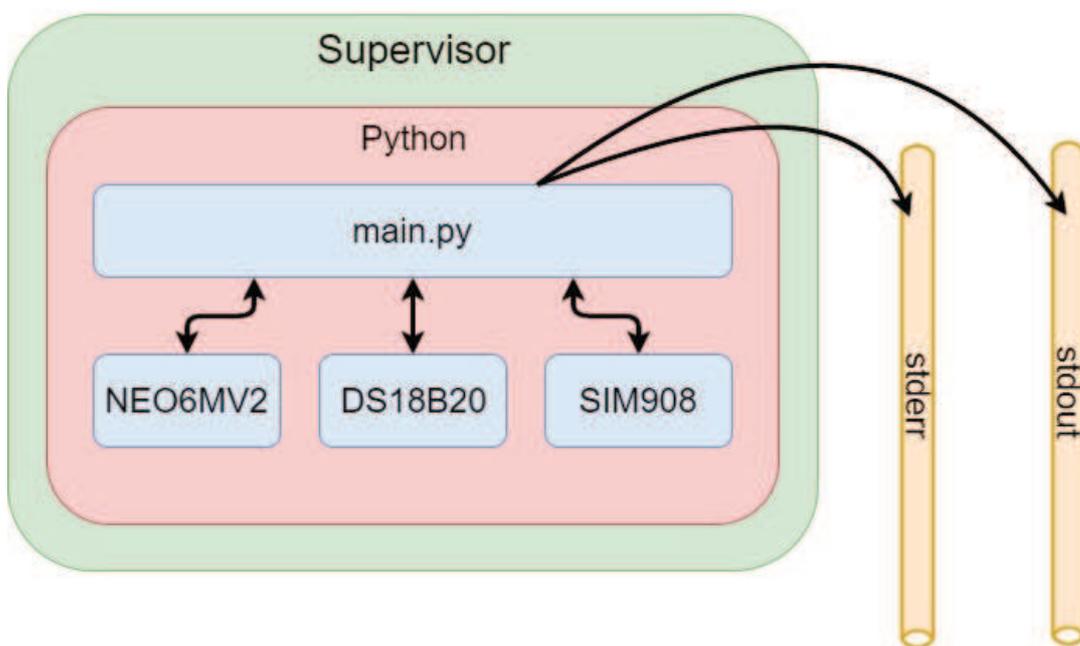


Figura 6.3.3: Firmware y software de la aplicación.

6.4. Servidor Web

Se seleccionó PHP como lenguaje de programación de las tareas del servidor web. PHP es un lenguaje de programación de uso general de código del lado del servidor para desarrollo de aplicaciones web. El módulo GPRS brinda la posibilidad de ejecutar los métodos de petición HTTP GET/POST/HEAD, que son utilizados para comunicar la información entre el sistema y el servidor. PHP procesa todas las peticiones, verifica que la fuente sea correcta mediante el IMEI del dispositivo, y si coincide la información obtenida con un bloque típico



de datos, los almacena en la base de datos MySQL, disponible dentro del servidor web.

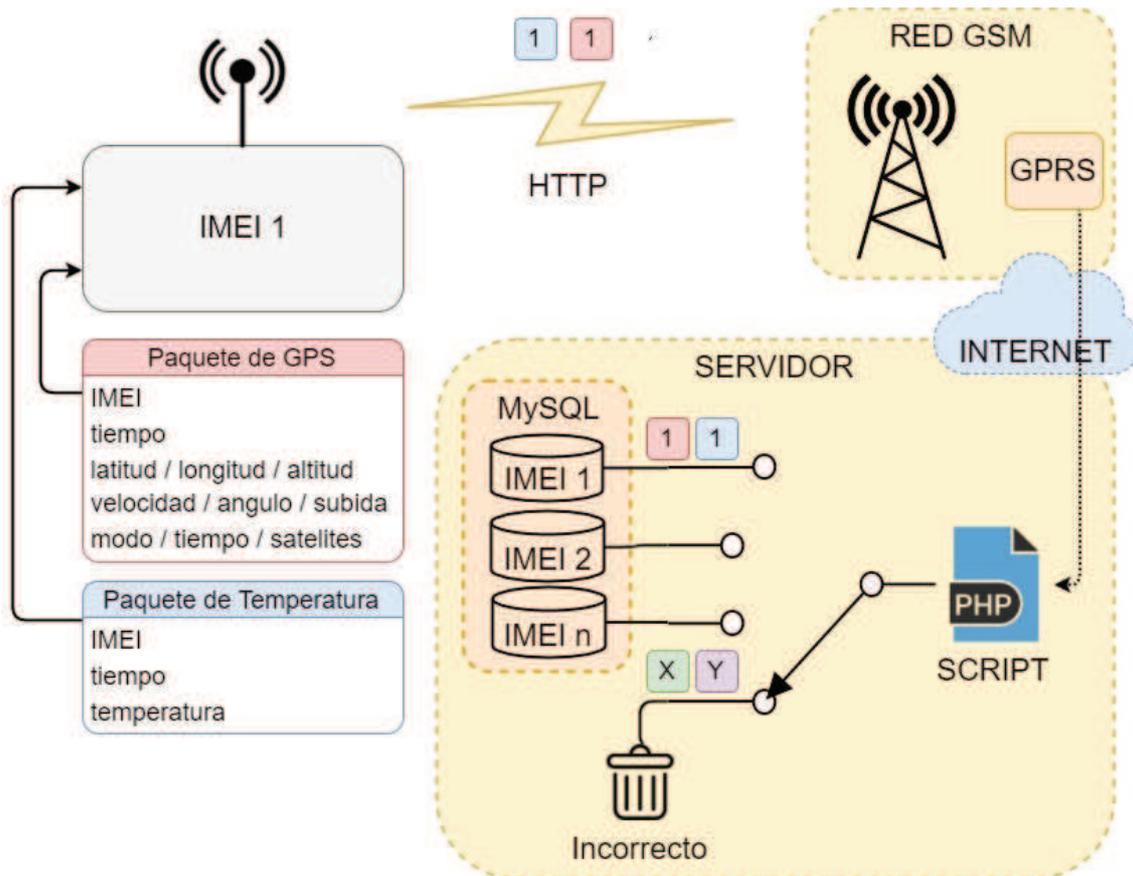


Figura 6.4.1: Flujo de paquetes de datos.

MySQL es un sistema de gestión de bases de datos relacional de código abierto, utilizada mayormente en entornos de desarrollo web. Se crearon dos tablas en la base de datos, una para almacenar toda la información del sistema de posicionamiento global, y otra para almacenar la información del transductor de temperatura. La simpleza de MySQL permite incorporar tablas de forma rápida y eficiente, otorgando versatilidad si fuese necesario modificar el sistema. Si se utilizan múltiples sistemas, la base de datos permite indexar cada entrada por IMEI, creando así tablas únicas para cada sistema.



Figura 6.4.2: Tablas de la base de datos.

6.5. Aplicación Web

La aplicación web fue desarrollada con el objetivo de que el usuario pueda disponer de la información de la forma más simple y compacta posible. La aplicación web hace una petición a la base de datos MySQL según el criterio que el usuario selecciona en el calendario. Luego se visualiza la información de posicionamiento en un mapa de Google Maps API, y la información de temperatura se visualiza en una gráfica de Plotly. El desarrollo de este código combina HTML y JavaScript, usualmente utilizados para la creación de sitios web.

Custom Range: 03/11/2016 12:30 - 03/11/2016 13:00

Today: 11/03/2016 11/03/2016

Last Position: 12 : 30 13 : 00

Yesterday: Nov 2016 Dec 2016

	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
Last 7 Days					3	4	5	27	28	29	30	1	2	3
Last 30 Days	30	31	1	2	3	4	5	4	5	6	7	8	9	10
This Month	6	7	8	9	10	11	12	11	12	13	14	15	16	17
Last Month	13	14	15	16	17	18	19	18	19	20	21	22	23	24
Custom Range	20	21	22	23	24	25	26	25	26	27	28	29	30	31
	27	28	29	30	1	2	3	4	5	6	7			

Apply Cancel



Figura 6.5.1: Selección de rango de visualización.

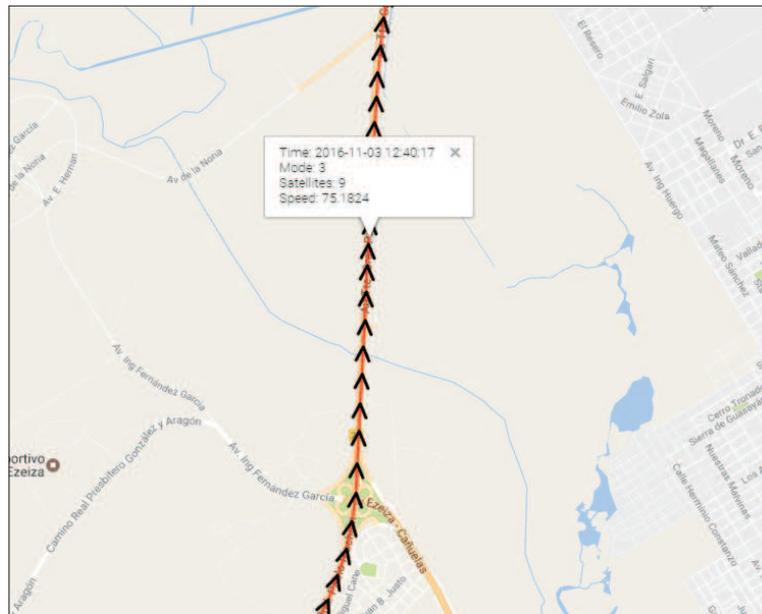


Figura 6.5.2: Google Maps API.

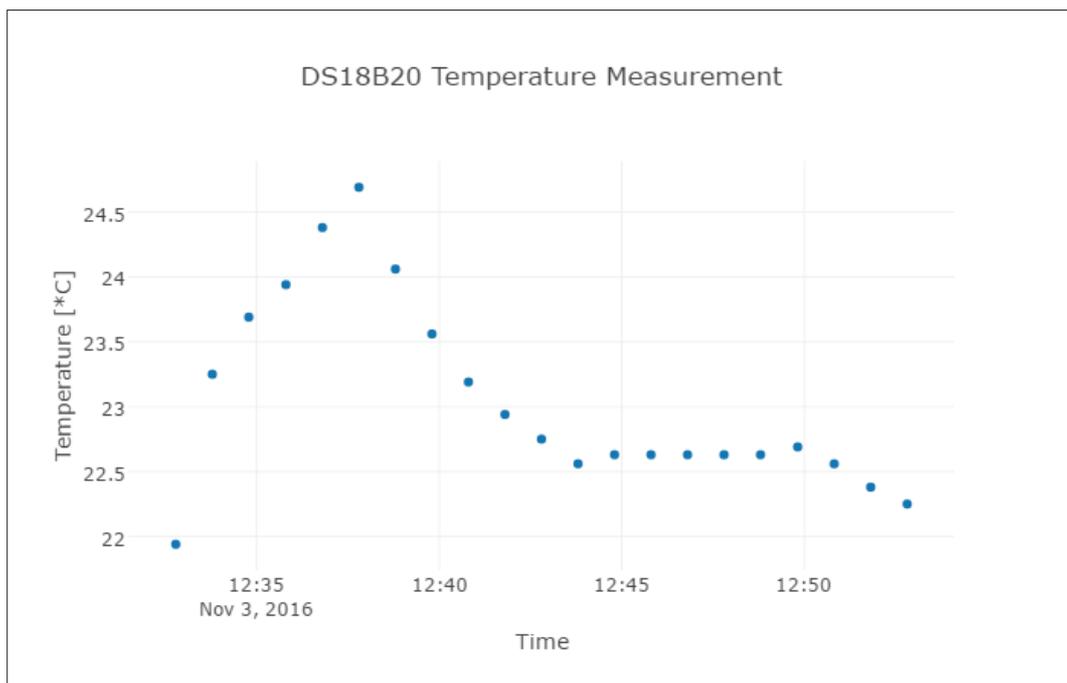


Figura 6.5.3: Gráfico de temperatura Plotly.



7. Conclusiones Finales

Se han logrado los objetivos propuestos. Se presentó un prototipo del sistema que cumple con los requisitos planificados. El prototipo fue probado en un vehículo con resultados favorables. El proyecto final ha sido un gran desafío y me llena de satisfacción haber podido desarrollarlo desde su principio hasta su fin.

Realizar un proyecto final desde el punto de vista de la gerencia de proyectos me otorgó elementos y herramientas que hasta este momento no había considerado pudieran ser tan útiles como resultaron, tanto en mi vida profesional como personal. Los nuevos conocimientos desarrollados me permiten comprender mejor la estructura de las organizaciones y la forma en la que emprenden en sus proyectos. Aplicar metodologías para abordar proyectos de cualquier índole me posibilitan expandirme a nuevos campos laborales.

He adquirido nuevos conocimientos orientados a las telecomunicaciones y los sistemas de información que son muy provechosos para mi perfil laboral. El constante crecimiento del uso de sistemas “internet de las cosas” (Internet of Things – IoT) produce una demanda cada vez mayor de profesionales.

Los conocimientos adquiridos a lo largo de la carrera probaron ser necesarios para el desarrollo completo del trabajo final.



8. Referencias

- Baddeley, G. (2001). *GPS - NMEA sentence information*. Obtenido de <http://aprs.gids.nl/nmea/>
- DigitalOcean. (2013). *How To Install and Manage Supervisor on Ubuntu and Debian VPS*. Obtenido de DigitalOcean Tutorials: <https://www.digitalocean.com/community/tutorials/how-to-install-and-manage-supervisor-on-ubuntu-and-debian-vps>
- Downey, A. (2012). *Think Python*. Massachusetts: Green Tea Press.
- Furrer, T. (2017). *W1thermsensor*. Obtenido de GitHub: <https://github.com/timofurrer/w1thermsensor>
- Gauchat, J. D. (2011). *El gran libro de HTML5, CSS3 y Javascript*. Barcelona: Marcombo.
- Geoff Sanders, L. T. (2003). *GPRS Networks*. West Sussex: John Wiley & Sons.
- Heine, G. (1999). *GSM networks: protocols, terminology, and implementation*. Massachusetts: Artech House.
- Jones, M. (2007). *Anatomy of the Linux kernel*. Obtenido de IBM developerWorks®: <https://www.ibm.com/developerworks/library/l-linux-kernel/>
- Khondker Shajadul Hasan, M. R. (2009). *Cost Effective GPS-GPRS Based Object Tracking System*. Hong Kong: IMECS 2009.
- Libelium. (2016). *Geolocation Tracker (GPRS + GPS) with SIM908 over Arduino and Raspberry Pi*. Obtenido de Cooking Hacks: <https://www.cooking-hacks.com/documentation/tutorials/geolocation-tracker-gprs-gps-geoposition-sim908-arduino-raspberry-pi/>



- Lundh, F. (2007). *Thread Synchronization Mechanisms in Python*. Obtenido de effbot.org: <http://effbot.org/zone/thread-synchronization.htm>
- Mehdi Achour, F. B. (2017). *PHP Manual*. Obtenido de PHP Documentation Group: <https://secure.php.net/manual/en/>
- Nerds Inc. (2013). *Demystifying the uBlox 6 GPS Module*. Obtenido de <https://thenerdcompany.blogspot.com.ar/2013/12/demystifying-ublox-6-gps-module.html>
- NovAtel. (2015). *An Introduction to GNSS, Second Edition*. Alberta: NovAtel.
- Portny, S. E. (2010). *Project Management For Dummies®, 3rd Edition*. New Jersey: Wiley Publishing.
- Project Management Institute. (2013). *Guía de los fundamentos para la dirección de proyectos (guía del PMBOK®) -- Quinta edición*. Pennsylvania: Project Management Institute.
- Raspberry Pi Foundation. (2016). *Raspberry Pi Documentation*. Obtenido de <https://www.raspberrypi.org/documentation/>
- SIMCom Wireless Solutions. (2011). *SIM908 AT Commands Set*. Shanghai: Sim Tech.
- SIMCom Wireless Solutions. (2012). *SIM908 IP Application Note*. Shanghai: Sim Tech.
- Suehring, S. (2002). *MySQL Bible*. New York: Wiley Publishing.