

Instituto Universitario Aeronáutico
Facultad de Ingeniería



TESIS DE GRADO

Sistema de monitoreo de napas
freáticas mediante freatómetros
electrónicos

Fernando Menendez
Córdoba - 2016

Agradecimientos

A mi familia, por haberme brindado la posibilidad de realizar la carrera de Ingeniero en Informática, por su apoyo, esfuerzo y colaboración a lo largo de estos años.

Al Profesor Gabriel Muñoz, quién desinteresadamente aceptó ser el tutor académico del presente proyecto, colaborando con su experiencia y conocimiento.

A los profesionales de distintas áreas, quienes desinteresadamente colaboraron brindando información y despejando dudas sobre el objeto de estudio.

A todos los docentes que he tenido a lo largo de la carrera, los cuales me brindaron las herramientas técnicas, morales y humanas que hoy me ayudan a desenvolver en mi vida profesional.

Sistema de Monitoreo de napas freáticas
mediante freatómetros electrónicos.

Resumen

El siguiente trabajo final de grado presenta el proceso de desarrollo de un sistema de monitorización en tiempo real del nivel freático a lo largo de un territorio. El mismo abarca desde el estudio del fenómeno en sí, sus causas y consecuencias, las tecnologías disponibles para llevar a cabo el proceso de medición, almacenamiento de los datos y acceso a los mismos, hasta el análisis y diseño, puesta en marcha y pruebas del sistema propuesto.

El proyecto básicamente se basa en el desarrollo de un prototipo de freatómetro, elemento cuya tarea es calcular la distancia desde el nivel de una acumulación de agua hacia la superficie y un servidor de datos, el cual deberá principalmente almacenar la información registrada por los freatómetros, y permitir un acceso a los registros actuales e históricos desde un terminal con acceso a internet.

El principal motivo por el que se decidió llevar a cabo este proyecto, es principalmente debido al alto nivel freático que se ha registrado en los últimos años en ciudades y pueblos del interior de la provincia de Córdoba y Santa Fe, los cuales generan numerosos problemas, principalmente edilicios. Los precarios métodos actuales en algunos casos, y totalmente nulos en otros sobre llevar adelante un monitoreo del fenómeno, alentaron al desarrollo de este sistema.

El resultado del proyecto es un prototipo de freatómetro electrónico, el cual captura el nivel con respecto a la superficie de un recipiente que contiene agua, basado en tecnologías de medición de distancias mediante ultrasonido, conjunto a un servidor de datos con sus respectivas aplicaciones, que interactúan con el primero tanto para obtener datos como para insertar nuevos. Además de un estudio teórico sobre como se deberá realizar la instalación de los mismos, la cobertura de la red de freatómetros para la ciudad de Monte Maíz (Córdoba) y futuras aplicaciones a entornos rurales.

Índice de contenido

Introducción.....	3
1. Antecedentes.....	3
2. Situación Problemática.....	3
3. Problema.....	5
4. Objeto de estudio y campo de acción.....	5
5. Aporte teórico.....	5
6. Aporte práctico.....	5
7. Métodos y medios de investigación científica.....	6
8. Métodos y medios de ingeniería.....	6
Objetivos.....	7
1. Objetivos generales y específicos.....	7
2. Alcance y limitaciones.....	7
Primera Parte: Marco Contextual.....	8
1. Entorno del objeto de estudio.....	8
2. Relación del tesista con el objeto de estudio.....	9
3. Análisis de los problemas observados.....	9
4. Antecedentes de proyecto similares.....	10
Segunda Parte: Marco Teórico.....	13
1. Marco teórico del objeto de estudio.....	13
2. Marco teórico del campo de acción.....	15
2.1 Campo de acción del dispositivo de Medición o Freatómetro.....	15
2.2 Campo de acción del servidor de datos.....	25
3. Diagnostico.....	29
3.1 Rango freático.....	29
3.2 Variación de la distancia con respecto a la Temperatura.....	30
Tercera Parte: Modelo Teórico.....	32
1. Planificación.....	32
2. Requerimientos.....	35
2.1 Sistema de monitorización.....	35
2.2 Freatómetro.....	38
3. Análisis de requerimientos.....	40
4. Diseño.....	47
4.1 Diagrama de clases del Modelo.....	47
4.2 Diagrama de clases del Servidor de Datos.....	48
5. Diseño de Interfaces Gráficas.....	49
Cuarta Parte: Concreción del Modelo.....	52
1. Implementación.....	52
1.1 Implementación de Prototipo de Freatómetro :.....	52
1.2 Implementación de Servidor :.....	57
1.3 Implementación de Webservice:.....	58
1.4 Implementación del Monitor Web.....	64
1.5 Implementación de Aplicación Móvil.....	68
1.6 Envío y Recepción de Alertas.....	70
2. Pruebas.....	72
2.1 Pruebas de Comunicación.....	72

2.2 Pruebas de medición.....	72
2.3 Pruebas de variación en el tiempo.....	73
3. Puesta en marcha.....	74
4. Prefactibilidad.....	75
Conclusión.....	77
Bibliografía.....	78

Introducción

1. Antecedentes

Las abundantes precipitaciones que ha sufrido tanto la provincia de Córdoba como gran parte del territorio argentino en los últimos años trajeron aparejados muchos problemas debido a, principalmente, la falta de infraestructura para afrontarlos. El principal problema, claro está, son las inundaciones que azotaron tanto a los pueblos y ciudades como a zonas rurales, aunque este no es el único problema presente.

En varias de las edificaciones que se encuentran dentro de las zonas afectadas, como casas, edificios, galpones, salones y hasta calles y rutas, se comenzaron a hacer presentes varios problemas edilicios como rajaduras o vencimiento de las paredes, levantamiento del suelo y fractura de pilares, produciendo desde entrada de humedad en las edificaciones hasta peligro de derrumbe.

La principal causa de este problema es el alto nivel de napa freática que se hace presente en estas zonas, en algunos casos muy cercana a la superficie terrestre, producto de las ya nombradas abundantes precipitaciones.

Una napa freática es una acumulación de agua subterránea que se encuentra a una profundidad relativamente pequeña bajo el nivel del suelo. Concretamente es un acuífero, con la diferencia de que los acuíferos pueden estar también a mayores profundidades.

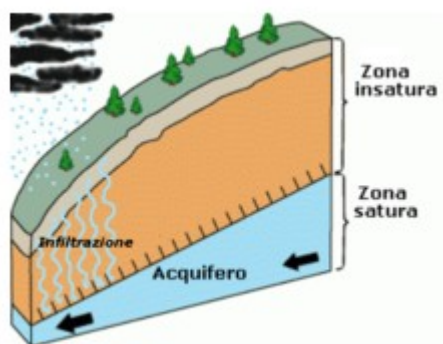
Este proyecto presenta un sistema de monitoreo del nivel de napa freática tanto actual como histórico, el cual permita mediante un terminal; ya sea una computadora, smartphone, tablet, etc. consultar la información a fin de reducir la incertidumbre y tomar decisiones.

El trabajo de realizar la medición del nivel de napa con respecto a la superficie, es realizado por un **freatómetro**, basado estructuralmente en un modelo tradicional y básico propuesto por el I.N.T.A (Instituto Nacional de Tecnología Agropecuaria). Actualmente es la herramienta más utilizada para realizar este tipo de mediciones, principalmente por los productores agropecuarios con el fin de realizar ajustes a las técnicas de siembra y cosecha.

Básicamente, el medidor consiste en un tubo de PVC enterrado en la tierra, por donde ingresa el agua, la cual hace elevar o disminuir la altura de un flotante (o boya) que permite visualizar la altura del nivel de napa.

2. Situación Problemática

Una napa freática es una acumulación de agua subterránea que se encuentra a una profundidad relativamente pequeña bajo el nivel del suelo. Su altura, con respecto a la superficie, varía de un lugar a otro principalmente por la distancia al nivel del mar en donde se encuentra el terreno, lo cual podría determinarse como constante, y por otros fenómenos propios de cada región que hacen variar el nivel freático.



Esquema representativo de la ubicación de una napa freática

Para la región de la llanura pampeana, en donde existe actualmente una fuerte actividad agrícola, la disminución del consumo del agua de las napas freáticas y el aumento del caudal de las precipitaciones, son las dos principales causas del creciente nivel freático.

La primera de las causas, se genera principalmente por el avance de la agricultura por sobre la ganadería, lo que genera una mayor cantidad de territorio en el cual se siembra actual y principalmente soja, en lugar de pasturas perennes y pastizales, que se utilizan para alimentar al ganado. Las pasturas y pastizales consumen mucha mayor cantidad de agua que por ejemplo las plantas de soja, maíz, trigo, etc. Otras variables que modificaron también el consumo son la aplicación de la siembra directa y la poca rotación de cultivos.

El aumento de las precipitaciones, otra de las variables principales y más comúnmente conocida, es producto principalmente del calentamiento global y la deforestación.

El nivel de napa freática al estar cerca de la superficie terrestre produce el hundimiento del suelo en donde se apoyan los cimientos de las edificaciones que se encuentran en la región, produciendo rajaduras o vencimiento de las paredes, levantamiento del suelo y fractura de pilares, logrando desde entrada de humedad hasta peligro de derrumbe.

La localidad de Monte Maíz, se encuentra al sudeste de la provincia de Córdoba, dentro del departamento Unión. Cuenta con aproximadamente 10.000 habitantes y un crecimiento constante debido a la demanda laboral por parte del sector agro-industrial y agropecuario, entre otras cosas.

Dentro de la localidad, debido a que se encuentra en una zona en donde el problema del alto nivel de napa freática se hace presente, se vienen registrando desde hace un tiempo numerosos problemas edilicios como los anteriormente mencionados, los cuales despiertan gran preocupación debido al peligro que significan.

La falta de infraestructura para capturar la información sobre este fenómeno, sumado también a la falta de conocimiento sobre el mismo, generan una gran incertidumbre en los lugares donde es necesario aplicar inversión (como la instalación de cámaras de desagote) o realizar trabajos para solucionar el problema y asentar información actual e histórica de los terrenos que permitan tomar decisiones a futuro, teniendo en cuenta

futuras construcciones.

La localidad de Monte Maíz no tiene registro sobre esto, ni actual ni histórico, por lo que la información sobre el estado de las napas a lo largo del terreno de la ciudad es prácticamente Nula.

3. Problema

La prácticamente nula información tanto actual como histórica sobre los niveles de napa freática por parte de las autoridades de ciudades y pueblos, hacen que haya una incertidumbre total a la hora de tomar decisiones, y también de realizar inversiones de mejoras edilicias, como de desagüe de las napas, a fin de disminuir el nivel de las mismas, sumado también a la incertidumbre en terrenos en donde se construirán nuevas edificaciones.

4. Objeto de estudio y campo de acción

El objeto de estudio es básicamente la medición del nivel de napa freática de forma automática y precisa, y que esta misma información capturada sea accesible desde cualquier punto a través de un terminal con conexión a internet.

Teniendo en cuenta este objeto de estudio, se identifican dos campos de acción principales:

- El primero consiste en la aplicación de microcontroladores programables y tecnologías de medición de distancias mediante ultrasonido a la captura y procesamiento de distancias de los niveles de napas freáticas con respecto a la superficie, las cuales sean autónomos y automáticos.
- El segundo radica en la elaboración de un servidor de datos conjunto a la utilización de redes TCP/IP que permitan una comunicación desde y hacia los módulos que realizan las mediciones. Junto a esto, el desarrollo de una aplicación que muestre los datos obtenidos, actuales e históricos.

5. Aporte teórico

El aporte teórico general del proyecto consiste en la aplicación de tecnología a la medición del nivel freático, ya que los procedimientos actuales de medición de los mismos son bastante precarios y manuales.

La aplicación de tecnología a este proceso de medición permite automatizar el mismo y lograr un acceso permanente y desde cualquier sitio, contando con un terminal con acceso a internet, además de lograr una mejor precisión en el estudio del fenómeno.

6. Aporte práctico

La implementación completa del proyecto beneficiará tanto a los habitantes de la localidad de Monte Maíz, como también a aquellas localidades que se encuentren con el problema del alto nivel de napas freáticas y deseen disponer de un sistema para controlar el mismo. El sistema aporta información a autoridades municipales e investigadores, que permitirá desde la toma de decisiones para solucionar el problema, aplicando obras e inversiones, con el fin evitar tanto actuales como futuros problemas edilicios en la localidad hasta información para el estudio científico del fenómeno.

También, la implementación del sistema en ambientes rurales, beneficia a los productores

agropecuarios, brindando información para mantener un registro histórico sobre el fenómeno y principalmente para el perfeccionamiento de técnicas de siembra y cosecha.

7. Métodos y medios de investigación científica

A fin de entender el entorno en el cual se desenvuelve el presente proyecto de tesis, los métodos empíricos para entender el mismo y reducir los riesgos se basan en entrevistas a distintos profesionales relacionados al objeto de estudio. También se ha realizado una exhaustiva recopilación de información sobre proyectos similares y antecedentes de los problemas observados a lo largo de los últimos años.

La determinación de las formas de automatizar el proceso de medición, tecnologías aplicables a el mismo, posibles riesgos, tanto en el proceso de desarrollo del mismo como en el funcionamiento final, también se llevan adelante mediante entrevistas y la recopilación de información tanto a entes profesionales como a bibliografía de confianza.

8. Métodos y medios de ingeniería

El desarrollo del sistema se hace bajo la metodología del Modelo Incremental, trabajando conjuntamente con los profesionales en el área de la Municipalidad de Monte Maíz, teniendo en cuenta la retroalimentación constante con respecto a los incrementos del modelo. Se utilizan algunos conceptos de la metodología Scrum, como reuniones semanales frecuentes a fin de corregir errores y mitigar posibles riesgos, tanto con el personal de la Municipalidad de Monte Maíz como con el tutor docente del proyecto de tesis.

Objetivos

1. Objetivos generales y específicos

El objetivo general consiste en la elaboración y puesta en marcha de un sistema que permita una forma sistemática de medición, almacenamiento y visualización de los niveles de napa freática a lo largo de un territorio, el cual brinde información con el fin de reducir la incertidumbre para la toma de decisiones, tanto para la mitigación de problemas futuros como para resolución de problemas actuales, con respecto a consecuencias de un alto nivel freático. El sistema estará compuesto por freatómetros, los cuales serán los encargados de realizar la medición y un servidor central que se encargará del almacenamiento, gestión y aprovisionamiento de los datos medidos por los freatómetros. Los objetivos específicos son:

- Diseñar un prototipo de freatómetro electrónico conjunto a un software embebido para su microcontrolador, el cual sea capaz de medir con precisión, utilizando tecnologías basadas en ultrasonido, la distancia a un determinado nivel de agua dentro de la superficie terrestre.
- Determinar en forma teórica, la cantidad de freatómetros necesarios para cubrir el territorio, de forma tal que la información que brinden los mismos sean precisas y evitar redundancias.
- Implementar un servidor con herramientas de gráficos, bases de datos y sistemas de alertas sobre un Sistema Operativo Linux, el cual pueda ser accedido desde cualquier parte, por medio de Internet.
- Aplicar una tecnología de comunicación entre los módulos que realizarán la medición (freatómetros) y el servidor que almacenará y mostrará los datos.

2. Alcance y limitaciones.

El proyecto se limita a la construcción y puesta en marcha de un prototipo de freatómetro a escala, el cual se encargará de medir diferentes niveles de agua, acumulados cada un tiempo determinado, y enviar esta información hacia un servidor que se encargará tanto de capturar y almacenar estas mediciones, como de mostrar la información tanto actual como histórica del fenómeno.

Quedan fuera del alcance del proyecto:

- La instalación real de los freatómetros en el suelo.
- La determinación del tipo de energía que se utilizará para alimentar a los módulos de los freatómetros.
- Las protecciones al hardware que se deben implementar, para que el mismo soporte las consecuencias de estar a la intemperie.
- Los métodos y medios de mantenimiento de los freatómetros

Primera Parte: Marco Contextual

1. Entorno del objeto de estudio

La aplicación del objeto de estudio se llevará a cabo en la localidad de Monte Maíz, la cual se encuentra en el departamento Unión, al sudeste de la provincia de Córdoba.



Ubicación geográfica de localidad de Monte Maíz. Google Maps ®

Las principales actividades de la región son la agricultura y ganadería, al igual que en gran parte de la llanura pampeana, aunque además, presenta una gran actividad industrial debido a agro-industrias que se hacen presente en la misma. A diferencia de muchas localidades aledañas y de similares características, Monte Maíz se encuentra en un proceso de crecimiento, tanto demográfico como de expansión geográfica, debido a las oportunidades laborales que se ofrecen.

El problema del alto nivel freático y sus consecuencias, se viene registrando desde hace un tiempo en gran parte del sudeste cordobés, como también en la llanura pampeana. Antes de esto, el nivel freático en la zona oscilaba entre 7 y 10 metros con respecto a la superficie.

Las principales causas de que este fenómeno sea cada vez más pronunciado en la localidad, es principalmente a las nombradas en la introducción, tanto el menor consumo del agua por parte del sector agropecuario, como el aumento del caudal de precipitaciones.

Otras causas, específicas de la localidad, fueron la instalación de las redes de agua potable, lo que produjo que se dejen de utilizar pozos de agua y bombas de

extracción de agua, comúnmente instalados en los hogares con el fin de obtener agua potable desde las napas freáticas. También el desmonte que se produjo dentro de la ciudad, producto del crecimiento demográfico de la misma, el cual demandó una mayor cantidad de construcciones edilicias, trajo aparejado lógicamente una menor cantidad de árboles que consuman el agua de las napas.

2. Relación del tesista con el objeto de estudio

La principal relación que une al tesista con el presente objeto de estudio es el interés que ha despertado, a través de las experiencias y estudios realizados en la facultad, la utilización e investigación sobre microcontroladores programables, sobre todo en lo que se refiere al proyecto Arduino. La principal inclinación al estudio y aplicación de microcontroladores, es para su utilización en la domótica y en la captura y medición de fenómenos.

Además, el tesista es oriundo de una ciudad en la cual el alto nivel freático se ha hecho presente en los últimos años, conociendo tanto sus consecuencias como la preocupación de sus habitantes, por los riesgos que genera el mismo. Esto ha llevado a interiorizarse en el estudio del fenómeno e identificar la necesidad de contar con un sistema, el cual permita monitorizarlo más precisa y fácilmente que las formas tradicionales existentes.

3. Análisis de los problemas observados

Desde hace aproximadamente 15 años, se comenzaron a observar dentro de la ciudad, problemas en estructuras edilicias, principalmente en el centro de la ciudad. En los últimos tiempos, la frecuencia de estos problemas fue creciendo año tras año. La razón de los mismos, claro está, es el alto nivel freático, lo que produce una presión ascendente hacia el piso, produciendo en muchos casos rajaduras o el desvanecimiento de los mismos, y trae aparejado problemas en los cimientos y paredes de las edificaciones, como rajaduras y posibles derrumbes, que en muchos casos hasta fue necesaria la instalación urgente de pilotes para evitarlos, debido al riesgo que significan.



Hundimiento de los cimientos de una casa.



Vereda desplomada en el centro de la ciudad de Monte Maíz



Pilotes instalados en casas, para evitar derrumbes debido al hundimiento de los cimientos

Muchos edificios comerciales de la ciudad de Monte Maíz, como la Mutual de Asociados del Club Deportivo Argentino, la sede Social del Club Atlético Lambert y en donde se encontraba anteriormente la Escuela Primaria, debieron suspender sus actividades para realizar trabajos a fin de reducir el riesgo de derrumbe y las entradas de humedad, producto de las rajaduras en las paredes, debido a este fenómeno. Un histórico edificio de la ciudad en el año 2014, debió ser completamente evacuado, suspendiendo todas sus actividades, debido al peligro de derrumbe que se presentaba en el mismo. La iglesia de la ciudad y numerosas casas particulares también sufrieron las consecuencias del alto nivel freático.

Otro de los grandes problemas, es la prácticamente nula información sobre los niveles freáticos, por debajo de terrenos que se utilizarán a futuro para construir nuevas edificaciones. La localidad de Monte Maíz se encuentra en un proceso de crecimiento, por lo que hay muchas edificaciones por construirse o en proceso de construcción, en las cuales se desconoce el nivel freático. Recientemente se realizó un gran loteo, para futuras construcciones de viviendas, en los cuales tampoco existe un registro ni actual, ni histórico, del nivel de napas freáticas por debajo de los terrenos.

4. Antecedentes de proyecto similares

El antecedente un proyecto similar, quizás el más conocido y utilizado generalmente por productores agropecuarios, para realizar mediciones del nivel de capa freática, es el modelo freatímetros propuestos por el I.N.T.A (Instituto Nacional de Tecnología Agropecuaria). El Instituto Nacional de Tecnología Agropecuaria es un organismo estatal descentralizado con autarquía operativa y financiera, dependiente del Ministerio de Agroindustria de la Nación. Fue creado en 1956 y desde entonces desarrolla acciones de investigación e innovación tecnológica en las cadenas de valor, regiones y territorios para mejorar la competitividad y el desarrollo rural sustentable del país.

Si bien hay varios modelos propuestos, sólo varían unos de otros en el tamaño de sus componentes, debido a que están propuestos para diferentes regiones del país. El modelo

propuesto consta básicamente de cuatro componentes:



Elementos del modelo de freatómetro propuesto por el I.N.T.A

- Varilla y tapa: la varilla es metálica, hueca. La tapa es plástica y sirve para proteger el extremo superior de la varilla.
- Flotante: es de plástico de 90 mm a 120 mm de diámetro y 110 mm a 130 mm de longitud. Su función es mantener la varilla en flotación por lo que su tamaño es muy importante. El flotante se une a la varilla mediante un trozo de tornillo.
- Tubo guía: es de PVC de 16 mm de diámetro y 20 cm de longitud. Su función es mantener la varilla en posición vertical y permitir su deslizamiento a medida que suba o baje el nivel freático. La parte superior del tubo guía es el punto de referencia donde se determina el nivel freático.
- Tubo ranurado: es de PVC de 110 mm a 200mm de diámetro y de 2 a 5 metros de longitud. Debe estar ranurado casi en su totalidad. Su función es evitar el desmoronamiento del pozo y guiar al flotante en su desplazamiento.
- Tapa del freatómetro: es de PVC y su diámetro es según el diámetro del tubo ranurado.

Su funcionamiento es muy sencillo: el tubo ranurado se coloca enterrado en el suelo, en la superficie donde se desea realizar la medición del nivel de napa. El agua proveniente de la napa freática, accede dentro del tubo por medio de las ranuras, elevando o disminuyendo el flotante con la varilla, la cual contiene una escala de colores que permite determinar el nivel napa freática.



Freatímetro instalado en el suelo, el sistema de colores de la varilla permite estimar a simple vista el nivel freático

El modelo de freatímetro propuesto en este proyecto de tesis, se basa en este modelo tradicional propuesto por el I.N.T.A en lo que se refiere a la estructura, debido a que se ha demostrado mediante su utilización la resistencia del mismo, la facilidad de acceso a los materiales y el bajo costo que representa.

Basado en este instrumento, el 20 de noviembre de 2015 el grupo NAPAS Marcos Juárez a cargo del I.N.T.A instaló una red de 14 freatímetros en la ciudad de Marcos Juárez, Provincia de Córdoba, con el objetivo de monitorear mensualmente el nivel de napa freática, conocer a qué profundidad se encuentra y delimitar las zonas con un nivel freático más cercano a la superficie.

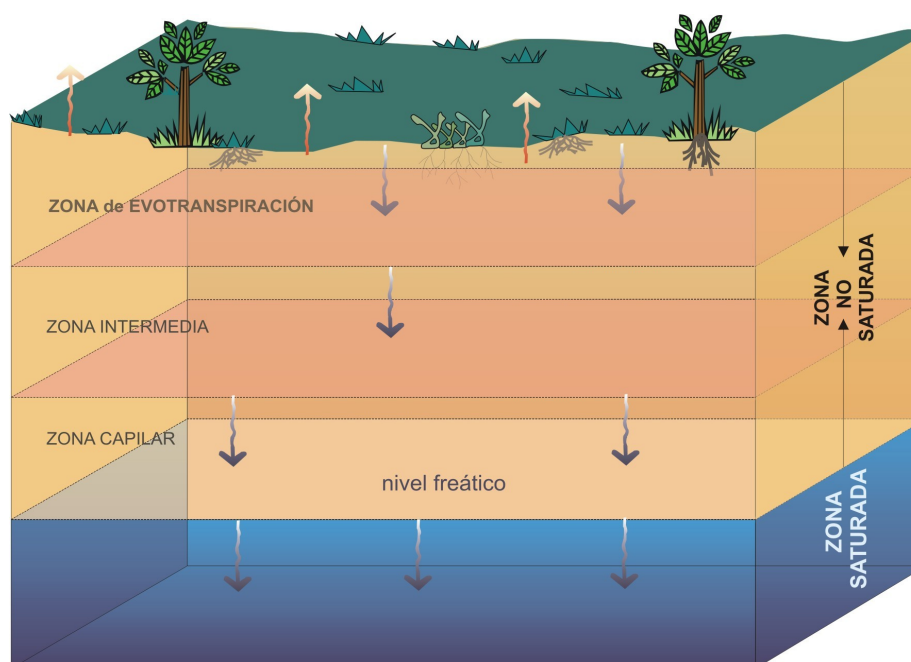
El proceso de captura de datos consiste en registrar mensualmente el nivel de cada uno de los freatímetros instalados, midiendo manualmente la distancia y registrando los resultados en planillas. Cada freatímetro posee un nombre simbólico que lo representa y fueron distribuidos estratégicamente en la ciudad. El grupo publica mensualmente los resultados obtenidos, establece una escala de riesgo para los mismos y realiza comparaciones con respecto a meses anteriores.

Es importante destacar este proyecto, debido a que además de poseer similitudes y coincidir en el objeto de estudio, las localidades de Monte Maíz (Ciudad para la cual se presenta este proyecto de tesis) y Marcos Juárez se encuentran relativamente cerca, compartiendo muchas características que pueden ser aprovechadas como información para mitigar riesgos y reducir la incertidumbre.

Segunda Parte: Marco Teórico

1. Marco teórico del objeto de estudio.

El objeto de estudio del presente proyecto es, en pocas palabras, la automatización de la medición de los distintos niveles freáticos a lo largo de un territorio, por medio de una red de freatómetros electrónicos, los cuales sean autónomos y de comportamiento automático. El concepto de nivel freático, hace referencia a la distancia de la zona saturada con respecto a la superficie, como se puede observar en la siguiente imagen:



Esquema de la zona interna de la tierra hacia la zona saturada, donde se ubica la napa freática

La necesidad de contar con este sistema de monitoreo, es principalmente por los ya nombrados problemas que traen aparejados el ascenso del nivel freático por sobre las localidades y ciudades de la región de la llanura pampeana, sector en donde se produjo mas acentuadamente el fenómeno.

Las principales causas del ascenso del nivel freático en la región son la disminución del consumo del agua de las napas freáticas y el aumento de las precipitaciones.

Un estudio publicado por el I.N.T.A (Bertram, N. & Chiacchiera, S, EEA Marcos Juárez) sugiere que los cambios producidos a niveles de sistemas productivos de la región, caracterizados por una disminución en el consumo de agua (más cultivos anuales, menos pasturas perennes y pastizales, mayor eficiencia en el uso del agua, etc.) generaron excedentes hídricos que fueron incorporándose regularmente al agua subterránea, determinando su acercamiento a la superficie.

Existen numerosos trabajos que documentan el crecimiento de la superficie agrícola,

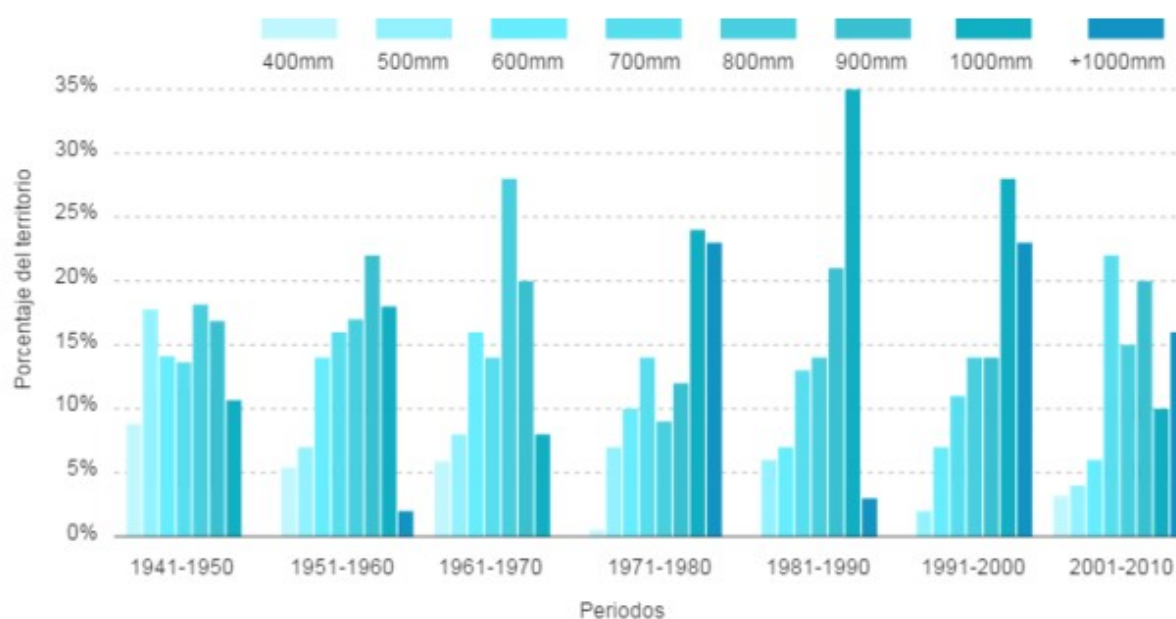
principalmente soja, en detrimento de aquella destinada a la actividad ganadera ocurrido en los últimos años. En términos generales, 10 millones de hectáreas han pasado de la actividad ganadera o mixta a la agrícola pura en la región pampeana.

Este desplazamiento y concentración de la ganadería no sólo tuvo efectos negativos directos sobre esta producción, sino que a su vez presentó efectos indirectos asociados al consumo de agua. Millones de hectáreas de pasturas perennes y pastizales que consumían agua durante los doce meses del año fueron cambiadas por cultivos anuales que, en el mejor de los casos lo hacen durante un tercio o la mitad de ese tiempo, pasando de consumir anualmente 1500-2000 mm a 500-800 mm. De esta manera, las pasturas y pastizales generaban menores ingresos de agua a las napas y, en aquellos lugares o períodos en los que éstas se acercaban a la superficie, se registraba un consumo más intenso, mientras que los cultivos anuales modificaron dicho balance, incrementando el ingreso de agua y limitando los egresos por consumo, empujando los niveles freáticos hacia la superficie.

Adicionalmente, las rotaciones de cultivos agrícolas se han empobrecido, disminuyendo la proporción de gramíneas, afectando no sólo el consumo de agua del cultivo actual y de los sucesivos, sino también el proceso de infiltración.

La siembra directa colaboró muchísimo con el incremento y estabilización de rendimientos, reduciendo las pérdidas de agua. Esta técnica, que comenzó incipientemente, con muy poca superficie en el país y la región a principios de la década de los 90, en la actualidad alcanza aproximadamente el 80% de la superficie cultivada del país y cerca del 100% en departamentos como Marcos Juárez. La implementación de la tecnología RR a mediados de la década del '90 favoreció la expansión del cultivo de soja, que hoy representa cerca del 80% de la totalidad de los cultivos de verano que se siembran en el departamento. A esto se suma la rápida adopción de barbechos limpios por más de seis meses en el año, los cuales también colaboraron con la reducción de las pérdidas de agua.

Con respecto al aumento de las precipitaciones, una investigación de la Facultad de Ciencias Agropecuarias (Oliva, E., UNCiencia) de la Universidad Nacional de Córdoba asegura que entre 1941 y 2010 la variable climática que más cambió en la región central de Argentina no fue la temperatura, sino la cantidad de lluvia anual. "Mientras la tendencia de la temperatura media anual sólo acusa un aumento significativo en menos del 10% del territorio analizado, el 57% de la región central de Argentina presenta una tendencia creciente del total de lluvia anual, lo que atenuó de manera significativa la condición de aridez.



El gráfico señala, para cada década, en qué porcentaje de territorio se registró cada valor de precipitación anual media. (UNCiencia, UNC)

2. Marco teórico del campo de acción.

Como se determinó en la introducción, se identifican dos campos de acción en este proyecto:

- La elaboración de un dispositivo de medición automática de los niveles de napa freática que comunique los datos capturados hacia un servidor de datos, objeto reconocido como freatómetro.
- La configuración de un servidor accesible desde cualquier lugar mediante un terminal, que permita tanto recibir los datos como mostrarlos de manera clara y entendible.

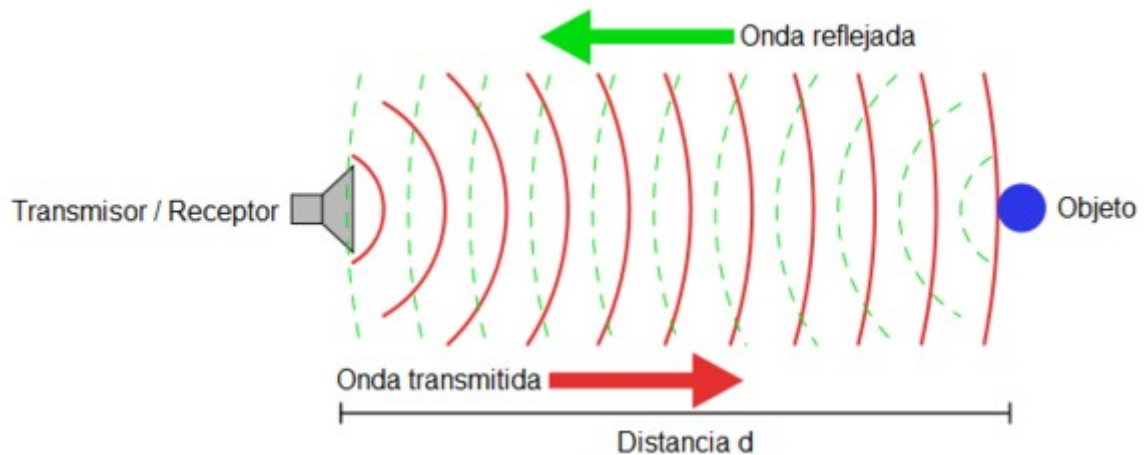
2.1 Campo de acción del dispositivo de Medición o Freatómetro.

La funcionalidad del freatómetro propuesto a construir requiere básicamente de tres procesos:

2.1.1 Medir la distancia entre la superficie terrestre y la napa freática (Nivel Freático):

La tecnología propuesta para la medición de distancias será el ultrasonido, el cual se propaga en forma de onda, de la misma manera que el sonido, con la diferencia que este no es audible por humanos debido a que se encuentra en una frecuencia mayor a 20 Khz, la aproximada mayor frecuencia audible por humanos.

El funcionamiento básico de los ultrasonidos como medidores de distancia se muestra en el siguiente esquema, donde un transmisor emite un pulso de ultrasonido que rebota sobre un determinado objeto, y la reflexión de ese pulso es detectada por un receptor de ultrasonidos:



Esquema de transmisión de la onda y recepción de la onda reflejada por el transductor.

La mayoría de los sensores de ultrasonido de bajo coste se basan en la emisión de un pulso de ultrasonido cuyo lóbulo, o campo de acción, es de forma cónica. Midiendo el tiempo que transcurre entre la emisión del sonido y la percepción del eco se puede establecer la distancia a la que se encuentra el obstáculo que ha producido la reflexión de la onda sonora, mediante la fórmula:

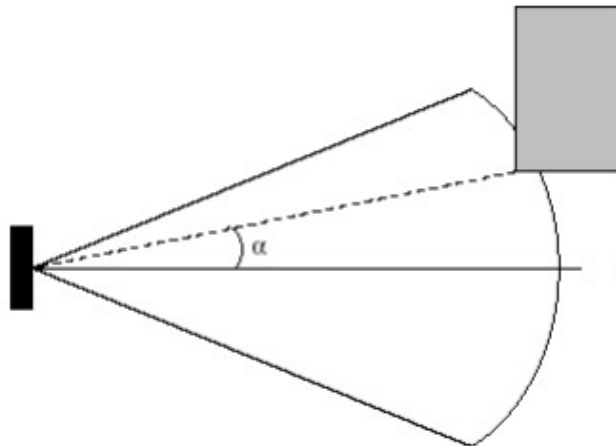
$$(1) \quad d = \frac{1}{2} V \cdot t$$

donde **V** es la velocidad del sonido en el aire y **t** es el tiempo transcurrido entre la emisión y recepción del pulso. Este modelo es también aplicable cuando el objeto en el cual se produce la reflexión es un líquido, debido a que la misma produce el eco por el cambio de densidad con respecto a el aire.

Existen factores inherentes tanto a los ultrasonidos como al mundo real, que influyen de una forma determinante en las medidas realizadas. Por tanto, es necesario un conocimiento de las diversas fuentes de incertidumbre que afectan a las medidas para poder tratarlas de forma adecuada, minimizando su efecto en el conocimiento del entorno que se desea adquirir. Estas son:

- El campo de actuación del pulso que se emite desde un transmisor de ultrasonido tiene forma cónica. El pulso que se recibe como respuesta al eco del sonido indica la presencia del objeto más cercano que se encuentra dentro del cono acústico y no especifica en ningún momento la localización angular del mismo. Aunque la

máxima probabilidad es que el objeto detectado esté sobre el eje central del cono acústico, la probabilidad de que el eco se haya producido por un objeto presente en la periferia del eje central no es en absoluto despreciable y debe ser tomada en cuenta.



El transductor puede detectar elementos que se encuentren a un ángulo distinto de el ángulo 0°

- La cantidad de energía acústica reflejada por el obstáculo depende en gran medida de la estructura de su superficie. Para obtener una reflexión altamente difusa del obstáculo, el tamaño de las irregularidades sobre la superficie reflectora debe ser comparable a la longitud de onda de la onda de ultrasonido incidente.
- Los factores ambientales tienen una gran repercusión sobre las medidas: Las ondas de ultrasonido se mueven por un medio material que es el aire. La densidad del aire depende de la temperatura, influyendo este factor sobre la velocidad de propagación de la onda según la expresión:

$$(2) \quad V_s = V_{so} \sqrt{1 + \frac{T}{273}}$$

siendo **V_{so}** la velocidad de propagación de la onda sonora a 0 °C, y **T** la temperatura absoluta (grados Kelvin).

En conclusión, la variación de la temperatura, tendrá repercusión en el cálculo final de la distancia. Esto afecta, sobre todo, cuando en un futuro se deberán comparar diferentes cálculos obtenidos en distintos días, donde claramente está, la temperatura variará.

Casi la totalidad de los transductores de ultrasonido traducen su medición de la distancia a un pulso eléctrico, variando su voltaje o tensión según la distancia obtenida. Este tipo de respuesta se conoce como una señal analógica.

Se dice que un sistema es analógico cuando las magnitudes de la señal se representan mediante variables continuas, esto es, análogas a las magnitudes que dan lugar a la generación de esta señal. Un sistema analógico contiene dispositivos que manipulan cantidades físicas representadas en forma analógica. En un sistema de este tipo, las cantidades varían sobre un intervalo continuo de valores.

Las señales digitales, en contraste con las señales analógicas, no varían en forma continua, sino que cambian en pasos o en incrementos discretos. La mayoría de las señales digitales utilizan códigos binarios o de dos estados (Generalmente 1 y 0).

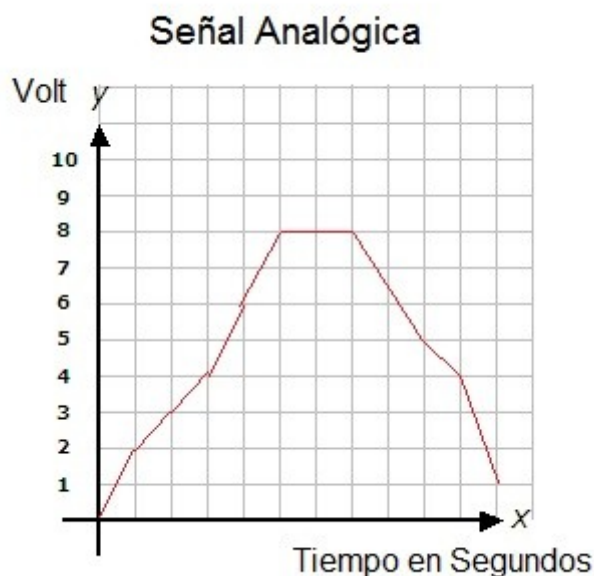
Basándose en que los transductores de ultrasonido son un sistema en sí, estos producen una salida como respuesta a un impulso de entrada, el cual activa el momento en el cual se debe disparar la señal ultrasónica transmitida, para luego recibir su reflexión o eco y así realizar el cálculo de la distancia de forma electrónica. La respuesta a este cálculo, o la salida de este sistema, se manifiesta como una señal analógica donde la variación del voltaje se relaciona con la distancia medida.

Muchos de estos dispositivos, en su gran mayoría, traen consigo un conversor Analógico-Digital, el cual básicamente se encarga de traducir la señal analógica a una señal digital, un valor continuo a un valor discreto. Esto se produce principalmente debido a que esta respuesta servirá como entrada a una unidad de procesamiento, que solo admite señales discretas, es decir, valores digitales.

Para realizar esa tarea, el conversor ADC (Analog-to-Digital Converter - Conversor Analógico Digital) tiene que efectuar los siguientes procesos:

- A.- Muestreo de la señal analógica.**
- B.- Cuantización de la propia señal**
- C.- Codificación del resultado de la cuantización, en código binario.**

A.- Muestreo de la señal analógica:



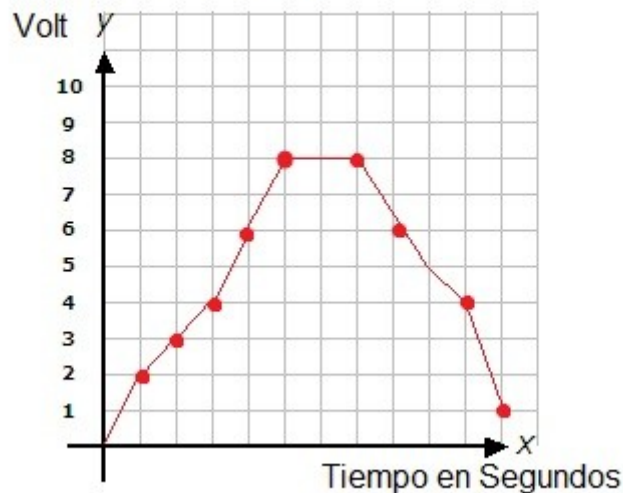
Muestreo gráficamente de la señal analógica

Para convertir una señal analógica en digital, el primer paso consiste en realizar un muestreo (sampling) de la misma, o lo que es igual a tomar diferentes muestras de tensiones o voltajes en diferentes puntos de la onda senoidal, la cual es la portadora de la señal analógica. La frecuencia a la que se realiza el muestreo se denomina tasa o frecuencia de muestreo y se mide generalmente en Kiloherz (kHz). A mayor cantidad de muestras tomadas, mayor calidad y fidelidad tendrá la señal digital resultante.

Durante el proceso de muestreo se asignan valores numéricos equivalentes a la tensión o voltaje existente en diferentes puntos de la senoide, con la finalidad de realizar a continuación el proceso de cuantización.

Por tanto, una señal cuyo muestreo se realice a 24 kHz (24.000 veces por segundo), tendrá menos calidad y fidelidad que otra realizada a 48 kHz (48.000 veces por segundo). Sin embargo, mientras mayor sea el número de muestras tomadas, mayor será también el ancho de banda necesario para transmitir una señal digital.

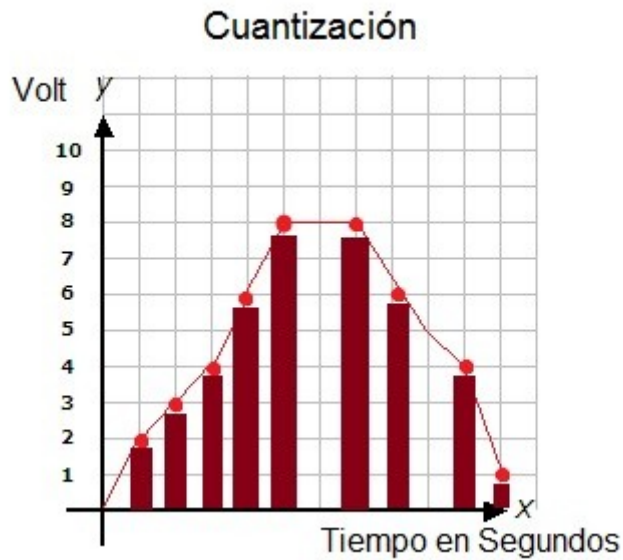
Muestreo (Sampling)



Samping de la señal análoga

B.- Cuantización de la señal analógica

Una vez realizado el muestreo, el siguiente paso es la cuantización (quantization) de la señal analógica. Para esta parte del proceso los valores continuos se convierten en series de valores numéricos decimales discretos correspondientes a los diferentes niveles o variaciones de voltajes que contiene la señal analógica original.

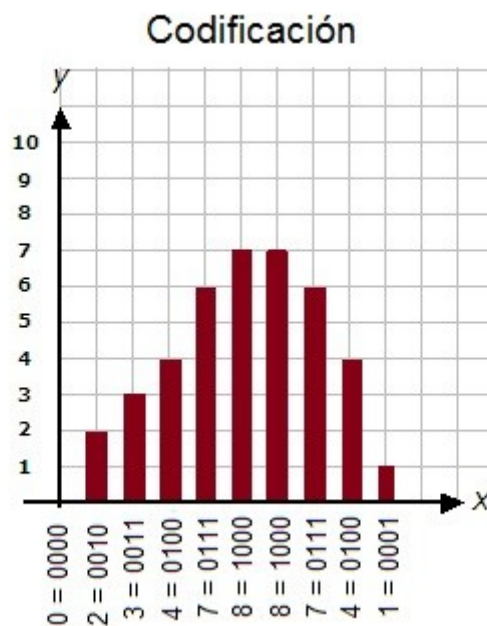


Cuantización gráficamente de la señal analógica

Por tanto, la cuantización representa el componente de muestreo de las variaciones de valores de tensiones o voltajes tomados en diferentes puntos de la onda sinusoidal, que permite medirlos y asignarles sus correspondientes valores en el sistema numérico decimal, antes de convertir esos valores en sistema numérico binario.

C.- Codificación de la señal en código binario

Después de realizada la cuantización, los valores de las tomas de voltajes se representan numéricamente por medio de códigos y estándares previamente establecidos. Lo más común es codificar la señal digital en código numérico binario o una serie de bits.



Codificación binaria de la señal

Cabe destacar que mientras mayor sea la cantidad de bits que se dispongan, para representar los respectivos valores continuos obtenidos en el muestreo de la misma, mayor será la fidelidad con la que se digitalice la señal continua.

Finalmente, es necesario aclarar que la mayoría de los transductores envían su respuesta en función del **tiempo** en el que la señal ultrasónica emitida causó el rebote en algún objeto y esta fue recibida por el receptor. Será trabajo del dispositivo que reciba resta respuesta calcular luego la distancia, la cual no es más que introducir el tiempo en la siguiente fórmula, ya anteriormente nombrada :

$$(1) \quad d = \frac{1}{2} V \cdot t$$

2.1.2 Controlar el tiempo de medición y procesar los datos obtenidos

Los transductores de ultrasonido requieren de una señal de activación para que el mismo inicie el proceso de medición que se explicó anteriormente, el cual una vez concluido, responden su resultado también como un pulso eléctrico. Aquellos que poseen un conversor analógico-digital lo harán en forma digital, ya que el proceso de conversión es interno al transductor, y los que no lo poseen lo harán de forma analógica.

A fin de controlar tanto los tiempos para los cuales se deberá efectuar el cálculo de la distancia hacia el nivel freático mediante ultrasonido, como también el procesamiento de la respuesta, ya sea analógica o digital, con el objetivo de enviarla luego hacia el servidor por medio de Internet, será necesario un microcontrolador programable que permita definir entradas y salidas, temporizar eventos y realizar comunicaciones. Para estos requerimientos, se ha determinado la utilización de un microcontrolador del proyecto Arduino.

Arduino es una compañía de hardware libre, la cual desarrolla placas de desarrollo que integran un microcontrolador y un entorno de desarrollo (IDE), diseñado para facilitar el uso de la electrónica en proyectos multidisciplinarios. El hardware consiste en una placa de circuito impreso con un microcontrolador, usualmente de la marca Atmel, y puertos digitales y analógicos de entrada/salida.

Por otro lado, el software consiste en un entorno de desarrollo (IDE) basado en el entorno de Processing y lenguaje de programación basado en C. La placa es programable y permite configurar sus puertos como entrada o salida. En la mayoría del gran abanico de placas que el proyecto ofrece, en todas se encuentra al menos un conversor analógico-digital y al menos un puerto de comunicación Serial, componentes necesarios para el procesamiento y la comunicación del freatímetro.

El conversor analógico-digital, se encuentra presente para ser utilizado en algunos de sus puertos, también llamados pines, mediante los cuales se puede realizar una entrada de una señal analógica hacia la placa, esto sólo será necesario en caso que el transductor de ultrasonido no posea el conversor A-D y la respuesta llegue como un pulso análogo.

El puerto Serial, es necesario para la comunicación de los datos obtenidos hacia algún

módulo externo que tenga la capacidad de conectarse a una red TCP/IP, y que la misma posea acceso a internet con la finalidad de comunicar el resultado en un servidor de datos. La interfaz que ofrecen la mayoría de los estos módulos, sea cual sea su tecnología (Ethernet, WI-FI, GPRS, etc), para enviar y recibir datos, es un puerto de comunicación Serial.

El concepto de comunicación Serial es sencillo. El puerto serial envía y recibe bytes de información un bit a la vez. Aun y cuando esto es más lento que la comunicación en paralelo.

Típicamente, la comunicación serial se utiliza para transmitir datos en formato ASCII. Para realizar la comunicación se utilizan 3 líneas de transmisión:

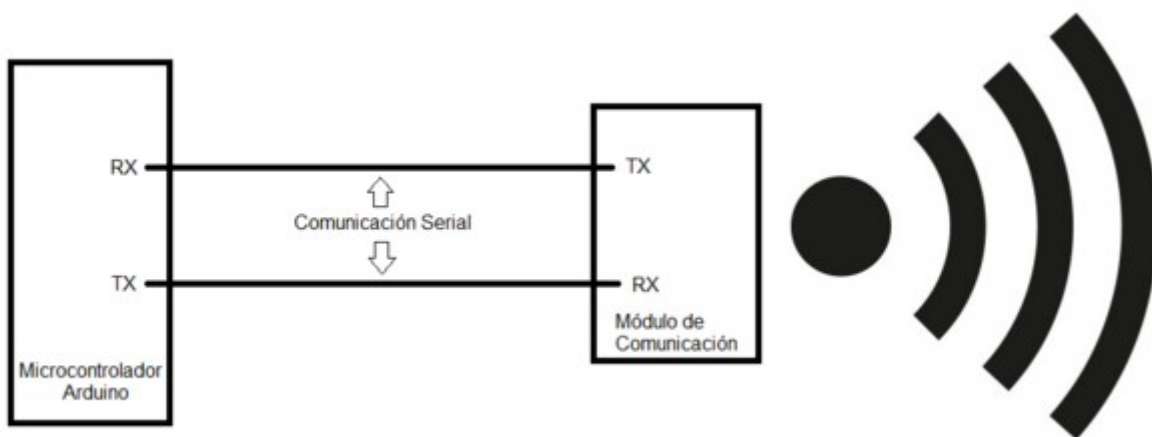
- Tierra (o referencia).
- Transmisión, el cual se lo conoce también como **TX**.
- Recepción, el cual se lo conoce también como **RX**.

Debido a que la transmisión es asincrónica, es posible enviar datos por una línea mientras se reciben datos por otra. William Stallngs en su Libro "Comunicaciones y Redes de Computadores" establece que las características más importantes de la comunicación serial son la velocidad de transmisión, los bits de datos, los bits de parada, y la paridad. Para que dos puertos se puedan comunicar, es necesario que las características sean iguales.

1. **Velocidad de transmisión (Baud rate):** Indica el número de bits por segundo que se transfieren, y se mide en baudios (bauds). Por ejemplo, 300 baudios representa 300 bits por segundo. Cuando se hace referencia a los ciclos de reloj se está hablando de la velocidad de transmisión.
2. **Bits de datos:** Se refiere a la cantidad de bits en la transmisión. Cuando la computadora envía un paquete de información, el tamaño de ese paquete no necesariamente será de 8 bits. Las cantidades más comunes de bits por paquete son 5, 7 y 8 bits. El número de bits que se envía depende en el tipo de información que se transfiere. Por ejemplo, el ASCII estándar tiene un rango de 0 a 127, es decir, utiliza 7 bits; para ASCII extendido es de 0 a 255, lo que utiliza 8 bits. Si el tipo de datos que se está transfiriendo es texto simple (ASCII estándar), entonces es suficiente con utilizar 7 bits por paquete para la comunicación. Un paquete se refiere a una transferencia de byte, incluyendo los bits de inicio/parada, bits de datos, y paridad. Debido a que el número actual de bits depende en el protocolo que se seleccione, el término paquete se usar para referirse a todos los casos.
3. **Bits de parada:** Usado para indicar el fin de la comunicación de un solo paquete. Los valores típicos son 1, 1.5 o 2 bits. Debido a la manera como se transfiere la información a través de las líneas de comunicación y que cada dispositivo tiene su propio reloj, es posible que los dos dispositivos no estén sincronizados. Por lo tanto, los bits de parada no sólo indican el fin de la transmisión sino además dan un margen de tolerancia para esa diferencia de los relojes. Mientras más bits de parada se usen, mayor será la tolerancia a la sincronía de los relojes, sin embargo la transmisión será más lenta.

4. **Paridad:** Es una forma sencilla de verificar si hay errores en la transmisión serial. Existen cuatro tipos de paridad: par, impar, marcada y espaciada. La opción de no usar paridad alguna también está disponible. Esto permite al dispositivo receptor conocer de antemano el estado de un bit, lo que serviría para determinar si hay ruido que esté afectando de manera negativa la transmisión de los datos, o si los relojes de los dispositivos no están sincronizados.

De esta manera, siempre y cuando el módulo se encuentre conectado a una red, y que esta misma tenga conexión a Internet, el microcontrolador podrá comunicar sus mediciones obtenidas hacia el servidor de datos. Esquemáticamente, la comunicación Serial entre el microcontrolador Arduino y el módulo de Comunicación es la siguiente:



Comunicación Serial entre el microcontrolador Arduino y el módulo de comunicación Wi-Fi

Como se puede observar, el puerto de transmisión Serial en el microcontrolador Arduino (TX) envía datos hacia el puerto Receptor Serial en el módulo de Comunicación (RX), inversamente el módulo de Comunicación envía mediante su puerto de Transmisión (TX) al puerto Receptor Serial del microcontrolador Arduino (RX).

El módulo de comunicación recibirá desde el microcontrolador Arduino tanto **tramas de control o configuración** como **tramas de datos**.

Las tramas de control que el microcontrolador Arduino envíe hacia el Módulo de comunicación permiten tanto configurar al segundo como establecer parámetros con los que luego se realizará la comunicación TCP/IP hacia el servidor, como pueden ser abrir una conexión TCP hacia una determinada dirección IP y puerto, configurar parámetros de red (como pueden ser el SSID al cual conectarse, en caso de tratarse de protocolo WI-FI), iniciar el envío de datos, cerrar la comunicación, etc.

Las tramas de datos que el microcontrolador envíe hacia el el módulo de comunicación, estas serán automáticamente enviadas hacia donde se estableció previamente (dirección IP y puerto) utilizando las tramas de control.

De esta manera, se abstrae de los requisitos que cada uno de los protocolos de Capa de

enlace de Datos, perteneciente al Modelo OSI, como lo son Ethernet, Wi-Fi, GPRS, etc. requieren para realizar una comunicación. Así, las tramas de datos que el módulo de comunicación recibe desde el microcontrolador Arduino (Canal inferior del esquema) serán automáticamente enviadas hacia la red, utilizando su tecnología y protocolo.

De la misma manera, es posible realizar el procedimiento inverso, en caso que se quiera enviar una trama de datos hacia el módulo de comunicación, este deberá tener abierto un puerto y asignada una IP, la cual se corresponda con la red a la cual está conectada. Las tramas de datos que el módulo reciba desde la red en esa dirección y puerto, las enviará automáticamente por su transmisor Serial hacia el microcontrolador Arduino (Canal superior del esquema).

Otra característica importante y necesaria que posee este tipo de microcontroladores es la posibilidad de configurar **Timers o interrupciones programadas**. Esto consiste en definir una frecuencia para la cual se ejecutará un conjunto de instrucciones. La forma en la que el microcontrolador se dota de esta capacidad, es utilizando la frecuencia de un cristal oscilador que le determina al microprocesador la cantidad de instrucciones que puede procesar por segundo. El cristal que poseen la mayoría de las placas del proyecto Arduino tienen una frecuencia de 16 Mhz, lo que determina que el procesador ejecutará 16.000 instrucciones por segundo. Basándose en esto, el microcontrolador permite configurar cada cuanto tiempo se desea ejecutar cierta funcionalidad con una alta precisión, pero no totalmente absoluta.

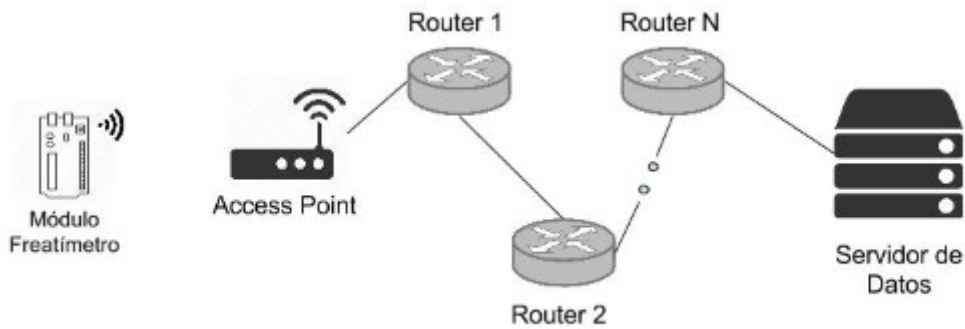
Esto será necesario para establecer con qué frecuencia se deberá enviar un pulso al módulo de medición de ultrasonido para que realice su proceso de medición y devuelva su resultado, que luego será enviado hacia Internet por el módulo de comunicación, utilizando su puerto de comunicación Serial como se explicó anteriormente.

2.1.3 Enviar hacia Internet los datos obtenidos

Se ha determinado como tecnología de comunicación a Wi-Fi, para comunicar las mediciones hacia el servidor de datos. Entonces, el módulo de comunicación que se introdujo anteriormente se comunicará hacia la red a través de esta tecnología.

La comunicación mediante el protocolo Wi-Fi está definida por IEEE en su estándar IEEE 802.11. Los estándares IEEE 802.11b, IEEE 802.11g e IEEE 802.11n utilizan la banda de 2,4 GHz, con una velocidad de hasta 11 Mbit/s, 54 Mbit/s y 300 Mbit/s, respectivamente. Se trata básicamente de una comunicación inalámbrica entre un terminal y un punto de Acceso (Access Point). Los puntos de acceso son dispositivos que generan un "set de servicio", que podría definirse como una "Red wifi" a la que se pueden conectar otros dispositivos. Permiten, en resumen, conectar dispositivos de forma inalámbrica a una red existente.

El siguiente esquema muestra cómo se logrará comunicar los datos obtenidos por el freatímetro hacia el servidor de Datos, llamando como Módulo Freatímetro al sistema compuesto por el microcontrolador Arduino, el transductor de ultrasonido y el Módulo de comunicación Wi-Fi en comunicación hacia el anterior por medio de sus puertos de Comunicación Serial:



El recorrido de los paquetes de datos que envía el freatímetro hasta llegar al servidor.

El primer paso es la comunicación desde el Módulo Freatímetro hacia el Access Point. Será necesario entonces, configurar en el Módulo Freatímetro el SSID (Service Set Identifier) y su respectiva contraseña para que el mismo pueda realizar su conexión hacia el Access-Point.

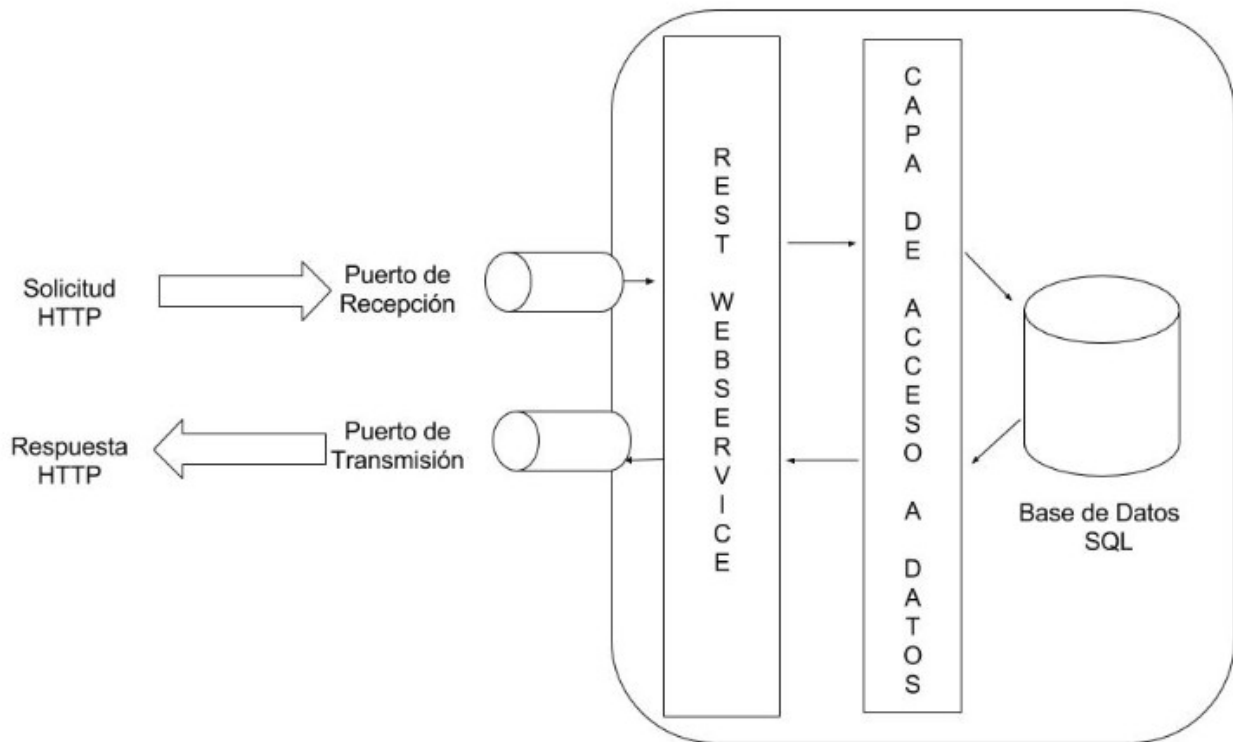
Una vez establecida la conexión, el Módulo Freatímetro enviará una trama TCP a la dirección IP del servidor de datos, la cual debe ser una IP pública y fija. Esta trama será primero transmitida hacia el Access-Point, el cual deberá estar conectado a una Red y será trabajo de los Routers, encaminar la trama enviada hacia el servidor de datos, pasando a través de Internet.

2.2 Campo de acción del servidor de datos

2.2.1 Servidor y protocolo de Comunicación.

El servidor de datos deberá tener la capacidad tanto de recibir datos por parte de los Freatímetros como también de recibir peticiones y enviar los datos requeridos hacia las aplicaciones encargadas de mostrar los mismos.

Como primer medida será necesario que el servidor sea accesible desde cualquier terminal conectado a Internet, ya que tanto los Freatímetros que enviarán los datos como la aplicación que requiera los mismos alcanzará el servidor si este se encuentra en Internet, esto se traduce a que el mismo deberá tener una dirección IP, o en su defecto un nombre asociado a la misma, que sea Pública y fija. La arquitectura propuesta es la siguiente:



Arquitectura del servidor, utilizando el protocolo HTTP (Puerto 80 - TCP) Para recibir peticiones y enviar respuestas.

El protocolo de comunicación que se utilizará a nivel Capa de Aplicación, es el protocolo HTTP, por lo que será necesario disponer de un Servicio que reciba las solicitudes y envíe las respuestas en base a las primeras.

Una solicitud HTTP es un conjunto de líneas que se envía al servidor. Este Incluye:

- **Una línea de solicitud:** es una línea que especifica el tipo de documento solicitado, el método que se aplicará y la versión del protocolo utilizada. La línea está formada por tres elementos que deben estar separados por un espacio:
 - el método
 - la dirección URL
 - la versión del protocolo utilizada por el cliente (por lo general, HTTP/1.0)
- **Los campos del encabezado de solicitud:** es un conjunto de líneas opcionales que permiten aportar información adicional sobre la solicitud y/o el cliente (navegador, sistema operativo, etc.). Cada una de estas líneas está formada por un nombre que describe el tipo de encabezado, seguido de dos puntos (:) y el valor del encabezado.
- **El cuerpo de la solicitud:** es un conjunto de líneas opcionales que deben estar

separadas de las líneas precedentes por una línea en blanco y, por ejemplo, permiten que se envíen datos por un comando POST durante la transmisión de datos al servidor utilizando un formulario.

Los métodos que el protocolo dispone son los siguientes:

Método	Descripción
GET	Solicita el recurso ubicado en la URL específica
HEAD	Solicita el encabezado del recurso ubicado en la URL especificada
POST	Envía datos al programa ubicado en la URL especificada
PUT	Envía datos a la URL especificada
DELETE	Borra el recurso ubicado en la URL especificada

Una respuesta HTTP es un conjunto de líneas que el servidor envía. Está constituida por :

- **Una línea de estado:** Es una línea que especifica la versión del protocolo utilizada y el estado de la solicitud en proceso mediante un texto explicativo y un código. La línea está compuesta por tres elementos que deben estar separados por un espacio: La línea está formada por tres elementos que deben estar separados por un espacio:
 - la versión del protocolo utilizada
 - el código de estado
 - el significado del código
- **Los campos del encabezado de respuesta:** Es un conjunto de líneas opcionales que permiten aportar información adicional sobre la respuesta y/o el servidor. Cada una de estas líneas está compuesta por un nombre que califica el tipo de encabezado, seguido por dos puntos (:) y por el valor del encabezado. Cada una de estas líneas está formada por un nombre que describe el tipo de encabezado, seguido de dos puntos (:) y el valor del encabezado.
- **El cuerpo de la respuesta:** Contiene el documento solicitado.

2.2.2 Procesamiento de solicitudes y Respuestas:

El procesamiento de las respuestas en base a las solicitudes entrantes en el servidor se hará a través de un **WebService RESTful**.

Un Servicio web REST se utiliza en el sentido más amplio para describir cualquier interfaz entre sistemas que utilice directamente [HTTP](#) para obtener datos o indicar la ejecución de operaciones sobre los datos, en cualquier formato ([XML](#), [JSON](#), etc).

Por medio de una URI, la cual se encuentra en la línea de solicitud HTTP, se especifica la ejecución que se desea realizar junto a los métodos que el protocolo HTTP dispone. Las propiedades de un WebService RESTful son las siguientes:

- Un webservice está compuesto de recursos.
- Cada recurso tiene un identificador único global, que es su **URI**. Usando una URL se puede alcanzar cualquier recurso en la web.
- Dada una URI, y mediante el protocolo **HTTP**, se pueden operar sobre estos recursos. La operación a realizar se especifica mediante el **método** del protocolo HTTP. Mediante cabeceras especiales como Accept o Content-Type se puede especificar que representaciones entiende el servidor y el cliente y que representación se usa en un mensaje concreto para transporta el estado del recurso.
- Los métodos del protocolo HTTP se referencian con las siguientes operaciones sobre un recurso:
 - El método **GET** hace la operación READ.
 - El método **DELETE** hace la operación DELETE.
 - El método **PUT** se usa normalmente para hacer UPDATE
 - El método **POST** se usa normalmente para hacer CREATE.
- Las representaciones a utilizar se especifican mediante los llamados tipos mime. La mayoría de los tipos MIME son estándares, como XML o JSON. El usar tipos mime estándar facilita la interoperabilidad.

La definición de las URI junto a los métodos para realizar operaciones sobre los recursos, se la conoce como **API**. La comunicación entre diferentes sistemas hacia y desde un API por medio de tipos **MIME**, permiten flexibilidad e independencia en la arquitectura de los sistemas para comunicarse entre sí.

El tipo MIME que se determinó utilizar para realizar la comunicación desde y hacia la API, el cual permitirá tanto a los freatímetros realizar inserciones de datos, como al Sistema de monitoreo consultar los mismos a fin de mostrarlos desde un navegador web, es **JSON** (acrónimo de JavaScript Object Notation). La principal causa del mismo es su mejor claridad y legibilidad con respecto a XML, además de ser más compacto y producir menos tráfico en la Red. Actualmente grandes empresas como Google recomiendan la utilización del mismo. Los tipos de datos disponibles con JSON son:

- **Números**: Se permiten números negativos y opcionalmente pueden contener parte fraccional separada por puntos. Ejemplo: 123.456
- **Cadenas**: Representan secuencias de cero o más caracteres. Se ponen entre doble comilla y se permiten cadenas de escape. Ejemplo: "Hola"
- **Booleanos**: Representan valores booleanos y pueden tener dos valores: true y false, verdadero y falso respectivamente.
- **null**: Representan el valor nulo.
- **Vector**: Representa una lista ordenada de cero o más valores los cuales pueden ser de cualquier tipo. Los valores se separan por comas y el vector se inserta entre corchetes.
- **Objetos**: Son colecciones no ordenadas de pares de la forma <nombre>:<valor> separados por comas y puestas entre llaves. El nombre tiene que ser una cadena y entre ellas. El valor puede ser de cualquier tipo

El almacenamiento de los datos ingresados para luego ser consultados y mostrados en

forma de gráficos, tablas o medidores por medio del sistema de Monitoreo se realizará en una Base de datos SQL.

La capa intermedia que se puede observar en la figura NÚMERO, llamada Capa de Acceso a Datos permite al sistema abstraer del tipo de Base de Datos implementada y definir las formas de acceso a la misma, tanto para inserción, actualización, modificación, eliminación o consulta de los datos.

3. Diagnostico.

3.1 Rango freático

Como punto de referencia, para establecer el rango de niveles freáticos, se posee como información las mediciones realizadas por el ya nombrado Grupo Napas Marcos Juárez perteneciente al I.N.T.A. Se toma este proyecto como referencia para establecer ciertos parámetros debido a que la distancia hacia la ciudad de Monte Maíz es de aproximadamente 100 Km, por lo que variables como la variación de la temperatura, precipitaciones y la altura por sobre el nivel del mar se pueden considerar prácticamente las mismas, aprovechando su cercanía.

Las mediciones del nivel freático obtenidas en la ciudad de Marcos Juárez para los meses de Abril y Marzo del año 2016 fueron los siguientes:

Ubicación de Freatómetros	Mediciones 04/2016	Mediciones 03/2016
1 - Centro periférico de Salud	0,75 m	1,50 m
2 - Pellegrini y Sarmiento Predio Coyspu	1,44 m	1,80 m
3 - PL. Funes y Belgrano	1,28 m	1,73 m
4 - Yrigoyen y Lago Henry Dellarossa Predio Coyspu	1.48 m	1.74 m
5 - Digeno Parque Industrial	1.25 m	1.65 m
6 - Las Colonias y Consituyentes Predio Coyspu	3.64 m	3.60 m
7 - Plaza de la Parroquia Barrio Lavalle	1.60 m	2.15 m
8 - Club El Panal	0.60 m	1.00 m
9 - Santa Fe y Fuerza Aérea	1.46 m	1.68 m
10 - Consorcio Caminero	0.50 m	0.80 m
11 - Barrio Tubero - Reinaudi S.	0.95 m	1.30 m
12 - Municipalidad de Marcos Juárez	3.40 m	3.70 m
13 - Cementerio	1.30 m	1.70 m
14 - Paseo de las Colonias	3.50 m	3.50 m

Como resultado se obtuvo un promedio de 1.82 m, la mayor distancia registrada con respecto a la superficie es de 3.7 m y la menor es de 0.5 m.

Teniendo en cuenta estos datos, se llega a la conclusión de que será necesario contar con un transductor de ultrasonido que tenga la capacidad de medir **distancias de hasta aproximadamente 4.5 metros y 0.2 metros como mínimo.**

3.2 Variación de la distancia con respecto a la Temperatura

Como se aclaró dentro del marco teórico, en donde se explicó el proceso de cálculo de distancia por ultrasonido, la temperatura del ambiente influye en el cálculo de la distancia. Teniendo en cuenta que las mínimas temperaturas anuales registradas en la región son de aproximadamente -2°C , y las máximas temperaturas anuales alcanzan los 40°C aproximados.

Si se supone una velocidad de onda de 343 m/s del sonido transmitido a 0°C , se obtienen, teniendo en cuenta las máximas y mínimas temperaturas de la región, las siguientes velocidades de propagación (Teniendo en cuenta la fórmula 2) :

$$V_{min}=343\text{ m/s}\sqrt{1+\frac{271,15}{273}}=484,25\text{ m/s}$$

$$V_{max}=343\text{ m/s}\sqrt{1+\frac{313,15}{273}}=502,59\text{ m/s}$$

Donde **Vmin** es la velocidad de propagación para la mínima temperatura ($271,15\text{ K}$ equivalentes a -2°C) y **Vmax** es la velocidad de propagación para la máxima temperatura ($313,15\text{ K}$ equivalentes a 40°C).

Suponiendo una distancia **d = 2 m**, se puede determinar la variación del tiempo para cada una de las velocidades de propagación, despejando el tiempo t de la fórmula 1:

$$T_{min}=\frac{2\cdot d}{V_{min}}=8,26\times 10^{-3}\text{ s}$$

$$T_{max}=\frac{2\cdot d}{V_{max}}=7,963\times 10^{-3}\text{ s}$$

Finalmente, se puede determinar cómo impacta la variación del tiempo, producto de la variación de la velocidad de propagación de la onda en el cálculo de una misma distancia:

$$D_{min}=\frac{1}{2}V_{min}\cdot T_{min}=1,9999525\text{ m}$$

$$D_{max}=\frac{1}{2}V_{max}\cdot T_{max}=2,00010\text{ m}$$

Como se puede observar, la diferencia entre **Dmin** y **Dmax** es casi despreciable en lo que refiere a la medición de distancias de napas freáticas. Consultado a personal de I.N.T.A, se tolera un error de hasta 10 cm, por lo que la tecnología de ultrasonido seleccionada para medir las distancias, bajo las características ambientales en donde se encuentra la localidad de Monte Maíz, se la puede considerar muy precisa.

Tercera Parte: Modelo Teórico

1. Planificación

La adopción del modelo Incremental para el desarrollo del presente proyecto de tesis, permite ir avanzando de manera conjunta con los usuarios y profesionales que utilizarán este sistema. De esta manera, a medida que se avanza en el proyecto, se mantiene una retroalimentación constante que permite mitigar riesgos, corregir errores y cambiar o perfeccionar requerimientos, con la finalidad de que los mismos al ser detectados a tiempo, no arrastren problemas a futuro que comprometan el resultado final del proyecto.

El total de las actividades planeadas, conjunto a su tiempo estimado, se han agrupado en cuatro iteraciones, al cabo de cada una de ellas se ha planeado una entrega al personal que utilizará el sistema a fin de obtener la ya nombrada retroalimentación.

La unidad de tiempo utilizada para medir el esfuerzo estimado es en días, teniendo en cuenta también, que todas las actividades identificadas son realizadas por la misma persona. Debido a esto, es en vano realizar una programación de recursos.

Tarea	Inicio	Fin	Duración	Completado	Dependencias
0) Sistema de Monitoreo de Napas Freáticas	2/2/16 8:00 a.m.	2/6/16 5:00 p.m.	17s 3d	0 %	
1) Iteración 1	2/2/16 8:00 a.m.	19/2/16 5:00 p.m.	2s 4d	0 %	
1.1) Identificación de requerimientos	2/2/16 8:00 a.m.	3/2/16 5:00 p.m.	2d	0 %	
1.2) Análisis de Requerimientos Identificados	4/2/16 8:00 a.m.	5/2/16 5:00 p.m.	2d	0 %	1.1
1.3) Análisis de Estrategias de Acceso a Datos	8/2/16 8:00 a.m.	8/2/16 5:00 p.m.	1d	0 %	1.2
1.4) Estudio de Tecnologías FrontEnd	8/2/16 8:00 a.m.	9/2/16 5:00 p.m.	2d	0 %	1.2
1.5) Estudio de Microcontroladores aplicables	8/2/16 8:00 a.m.	10/2/16 5:00 p.m.	3d	0 %	1.2
1.6) Estudio de Tecnologías de Medición	8/2/16 8:00 a.m.	10/2/16 5:00 p.m.	3d	0 %	1.2
1.7) Estudio de Tecnologías Aplicables de Comunicación	8/2/16 8:00 a.m.	8/2/16 5:00 p.m.	1d	0 %	1.2
1.8) Análisis de Componentes a Aplicar	11/2/16 8:00 a.m.	12/2/16 5:00 p.m.	2d	0 %	1.6, 1.5, 1.4, 1.3, 1.7
1.9) Estudio de Componentes Seleccionados	15/2/16 8:00 a.m.	19/2/16 5:00 p.m.	1s	0 %	1.8

2) Iteración 2	22/2/16 8:00 a.m.	25/3/16 5:00 p.m.	5s	0 %	1
2.1) Construcción Prototipo de Interfaces Gráficas	22/2/16 8:00 a.m.	24/2/16 5:00 p.m.	3d	0 %	1
2.2) Configuración de Servidor de Datos	22/2/16 8:00 a.m.	25/2/16 5:00 p.m.	4d	0 %	
2.3) Elaboración de Base de Datos	22/2/16 8:00 a.m.	22/2/16 5:00 p.m.	1d	0 %	
2.4) Desarrollo de Capa de Acceso a Datos	25/2/16 8:00 a.m.	25/2/16 5:00 p.m.	1d	0 %	2.2FF, 2.3
2.5) Desarrollo Webservice	26/2/16 8:00 a.m.	1/3/16 5:00 p.m.	3d	0 %	2.4
2.6) Desarrollo FrontEnd	2/3/16 8:00 a.m.	25/3/16 5:00 p.m.	3s 3d	0 %	2.5, 2.1
3) Iteración 3	28/3/16 8:00 a.m.	29/4/16 5:00 p.m.	5s	0 %	2
3.1) Integración de Componentes hacia Microcontrolador	28/3/16 8:00 a.m.	15/4/16 5:00 p.m.	3s	0 %	
3.2) Desarrollo de Software Embebido	18/4/16 8:00 a.m.	29/4/16 5:00 p.m.	2s	0 %	3.1
3.3) Desarrollo de prototipo Estructural	18/4/16 8:00 a.m.	27/4/16 5:00 p.m.	1s 3d	0 %	3.1
4) Iteración 4	2/5/16 8:00 a.m.	2/6/16 5:00 p.m.	4s 4d	0 %	3
4.1) Pruebas Modulares (Caja Blanca)	2/5/16 8:00 a.m.	13/5/16 5:00 p.m.	2s	0 %	
4.2) Pruebas de Integración de Módulos	16/5/16 8:00 a.m.	20/5/16 5:00 p.m.	1s	0 %	4.1
4.3) Prueba de Funcionamiento de Sistema (Caja negra)	23/5/16 8:00 a.m.	27/5/16 5:00 p.m.	1s	0 %	4.2
4.4) Análisis de resultados Obtenidos	30/5/16 8:00 a.m.	2/6/16 5:00 p.m.	4d	0 %	4.3

La programación de las actividades da como resultado el siguiente diagrama de Gantt:

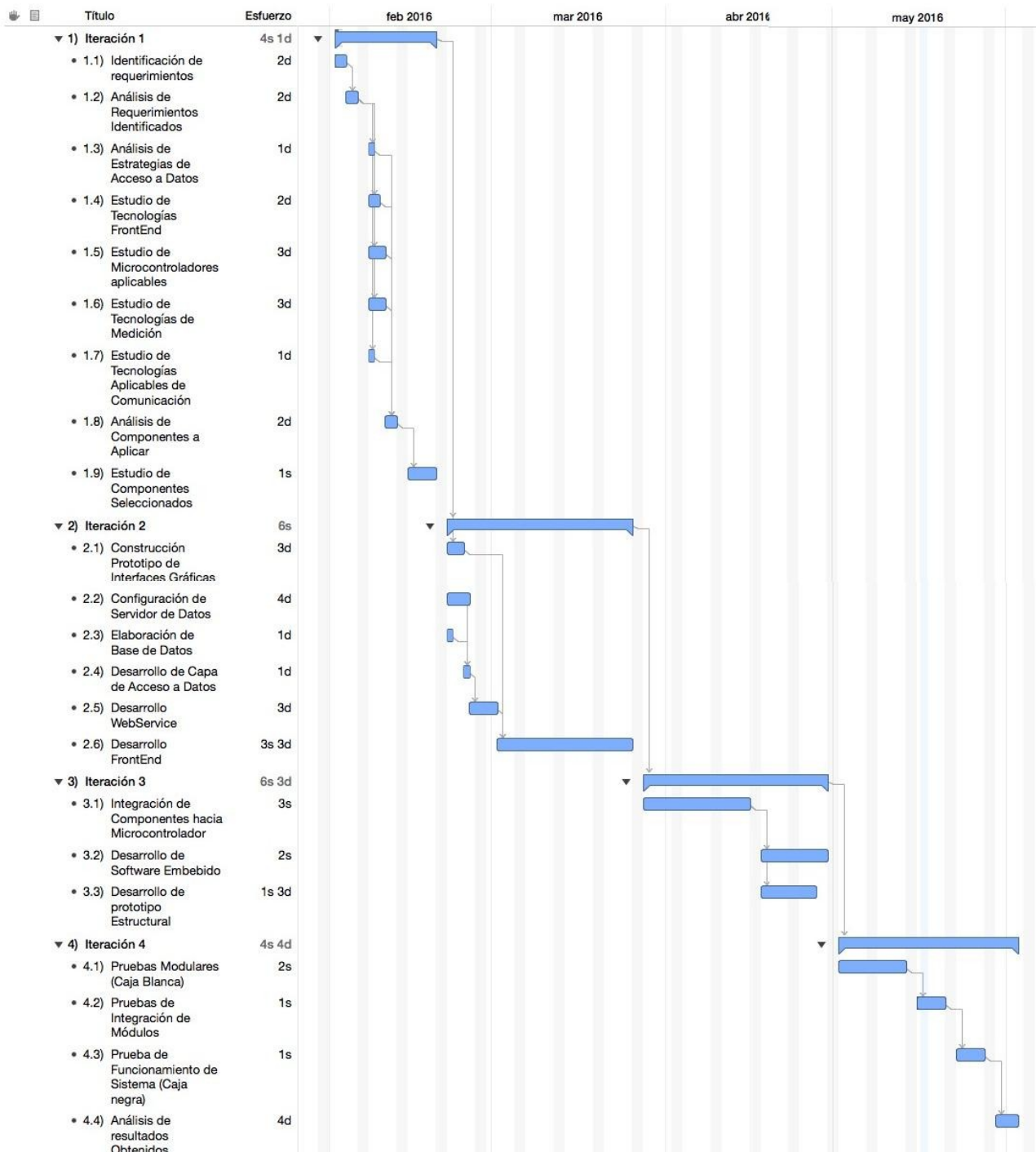


Diagrama de Gantt producto de la planificación de Actividades.

2. Requerimientos

2.1 Sistema de monitorización

El proceso de identificación de casos de uso, conjunto a la determinación de los actores que ejecutarán el mismo, es el primer paso a llevar a cabo para determinar las funcionalidades que debe tener el sistema con una granularidad gruesa. De esta manera, quedarán expuestos los requerimientos del sistema.

Con la ayuda de un diagrama de casos de uso, se puede analizar y comunicar:

- Los escenarios en los que el sistema o aplicación interactúa con personas, organizaciones o sistemas externos.
- Los objetivos que el sistema o aplicación contribuye a lograr.
- El ámbito del sistema.

Requerimientos Funcionales:

El siguiente diagrama de Casos de uso muestra los diferentes actores que interactúan con el sistema de monitorización de Napas Freáticas. Es importante aclarar, que desde el punto de vista del Sistema de monitorización en sí, cada uno de los Freatímetros son un sistema externo al mismo, por lo que se lo considera como un actor que interactúa con el sistema de Monitorización, produciendo entradas al mismo.

Cada actor en el modelo, representa una manera en la cual un agente externo se relaciona con el sistema, obteniendo una salida o respuesta, en base a una entrada, debido a esto, un actor no necesariamente representa una persona y una persona, puede ser para el sistema, varios actores.

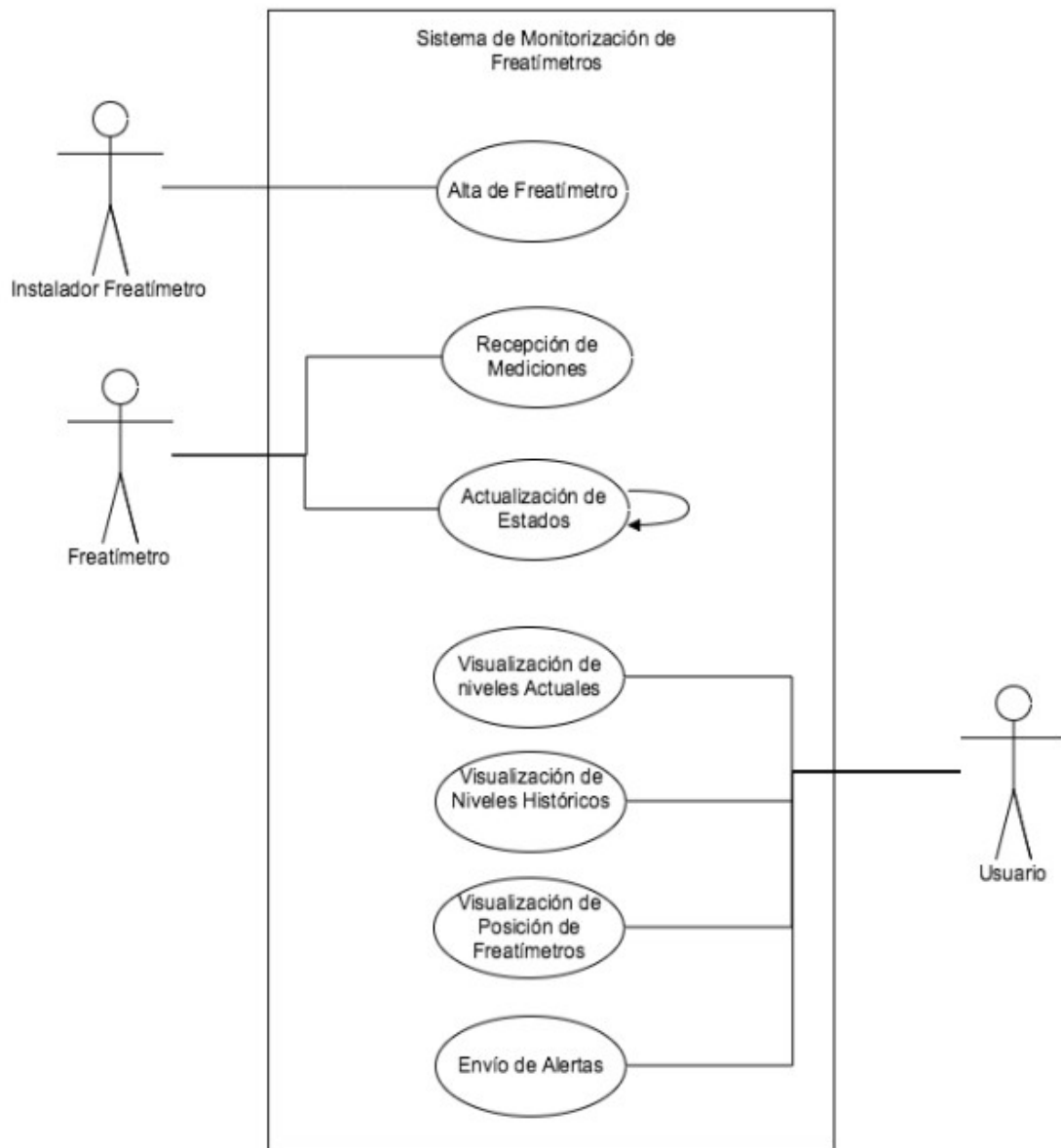


Diagrama de Casos de Uso del servidor de Datos

1 - Alta de Freatímetro :

Una vez instalado el freatímetro en el suelo, el actor Instalador de Freatímetro deberá dar de alta el mismo en el sistema, para que luego este comience a enviar las mediciones. Cada freatímetro tendrá un identificador único, y será necesario enviar también la Latitud y Longitud donde fue instalado el mismo y un nombre descriptivo para su fácil identificación. A raíz de este requerimiento, el mejor dispositivo para realizar esta acción es mediante un smartphone que posea GPS, debido a que hoy en día es ampliamente utilizado y posee el hardware requerido.

Entonces, será necesaria una aplicación para dispositivos móviles, la cual permita al actor Instalador de Freatímetro cargar el Identificador único del freatímetro, su nombre representativo, y tomará desde GPS su posición actual para determinar la latitud y longitud donde se está instalando el mismo.

Estos datos al ser recibidos por el servidor del Sistema de Monitorización deberán ser

almacenados de manera persistente en una Base de datos.

El caso de uso también debe ingresar un estado inicial al freatómetro recientemente agregado.

2 - Recepción de Mediciones :

Cada uno de los los freatómetros dados de alta comenzará a enviar las mediciones del nivel freático cada cierto tiempo definido. El Servidor del Sistema de Monitorización deberá recibir estas mediciones, por lo que será necesario además de recibir el valor de la medición en sí, el identificador de que freatómetro está enviando la misma. Estos datos deben ser almacenados de manera persistente en una Base de datos para su posterior explotación. Al momento de almacenar los mismos es necesario también registrar fecha y hora del evento.

3 - Actualización de Estados

Cada freatómetro dado de alta en el sistema, deberá tener un estado asociado para determinar el funcionamiento del mismo. Los estados requeridos son:

- Estado de Inicio: Cuando el freatómetro fue dado de alta pero aún no envió ninguna medición.
- Estado Activo: El freatómetro a enviado mediciones recientemente.
- Estado de Alerta: El freatómetro no ha enviado mediciones después de HORAS desde la última medición.
- Estado de Error: El freatómetro no ha enviado mediciones después de HORAS desde la última medición.

Cuando se efectúe tanto la alta del nuevo freatómetro, como la recepción de las mediciones por parte del mismo, se deberán actualizar los estados. El sistema deberá también tener la capacidad de actualizarse automáticamente (referencia a sí mismo en el diagrama de casos de uso) para establecer cuando un estado debe cambiar a estado de Alerta o estado de Error.

4 - Visualización de Niveles Actuales

Se requiere la capacidad de mostrar el nivel de cada uno de los freatómetros instalados en una misma pantalla, conjunto a una regla de colores que permita identificar fácilmente la gravedad del nivel freático.

5 - Visualización de Niveles Históricos

Mediante gráficos conocidos como histogramas, se requiere mostrar las mediciones históricas de cada uno de los freatómetros, con el fin de estudiar la variación del fenómeno en el tiempo y realizar comparaciones. Se debe poder contar con la opción de ajustar el tiempo en el histograma de manera diaria, mensual y anual.

6 - Visualización de Posición de Freatómetros

Mediante la utilización de la latitud y longitud cargada en cada freatómetro desde su alta, se deberán mostrar en un mapa la ubicación de cada uno de los freatómetros, permitiendo visualizar su nombre y también el nivel freático actual.

7 - Envío de Alertas

Será necesario notificar a usuarios sobre ciertos eventos que ocurran. Uno de ellos será en el momento en que por alguna razón, un freatímetro dado de alta pase a estado de Alerta o Error. Otro evento que deberá ser notificado, es cuando en cierto freatímetro se registre que el nivel freático es menor a un determinado valor, lo que presupone cierto riesgo en una determinada zona. La forma de notificaciones se realizará hacia la aplicación Móvil por notificaciones Push.

Requerimientos No Funcionales:

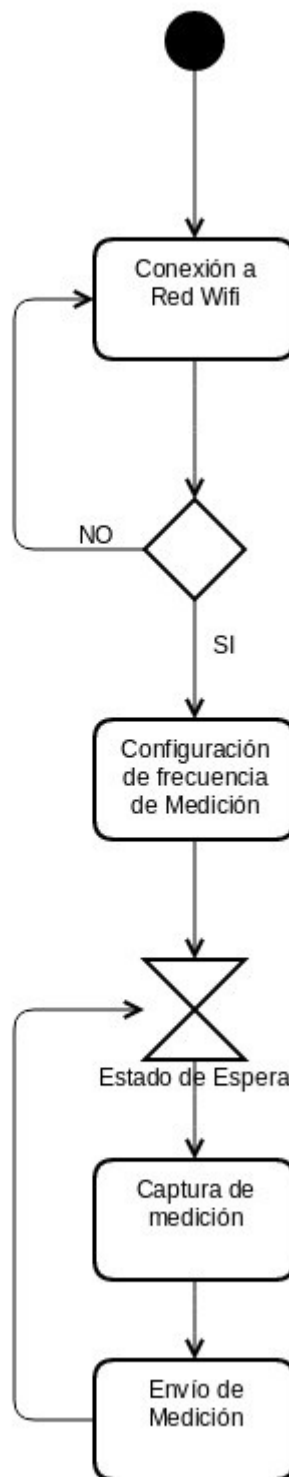
Se requiere que el Sistema de monitorización contemple los siguientes requerimientos:

- Las pantallas de la aplicación se deben adaptar perfectamente a Móviles, tablets y PC.
- Acceso permanente desde cualquier sitio donde se posea una conexión a Internet.
- Performance en la consulta y muestreo de datos.
- Agilidad en la carga y funcionamiento.

2.2 Freatímetro

A raíz de que en su comportamiento, los freatímetros no reciben interacción alguna de su entorno (Su comportamiento es automático, programado previamente), no es posible identificar los requerimientos funcionales del mismo mediante Casos de Uso.

De todas maneras, el requerimiento funcional, se puede definir como que se espera del mismo la medición del nivel freático y el envío de los datos capturados hacia el servidor cada un tiempo previamente programado. Teniendo en cuenta esto, se puede definir su ciclo operativo mediante el siguiente diagrama de Actividad:



El diagrama de actividad detalla el comportamiento que debería tener el freatómetro electrónico

Como se puede observar, es un proceso que no tiene fin, es decir, mientras el dispositivo contenga energía y conexión a internet realizará su tarea. El proceso comienza con la conexión del freatímetro, mediante su módulo Wi-Fi a una red. De esta manera será necesario configurar en el microcontrolador, tanto el SSID como su Contraseña de acceso. En caso que no se logre la conexión el dispositivo quedará intentándolo continuamente, si se logra la misma, se procede a la Configuración de la frecuencia de medición, es decir, cada cuanto tiempo se deberá capturar la distancia del nivel freático. Una vez configurada la misma, el freatímetro entra en estado de espera, a fin que se cumpla el tiempo configurado anteriormente. Cuando el tiempo se cumple, se procede, en primer paso a Capturar el nivel freático mediante el sensor de ultrasonido, y luego, a enviar los datos hacia el servidor por medio del módulo Wi-Fi. Concluido esto, el freatímetro vuelve a estado de espera y ejecuta nuevamente el ciclo.

3. Análisis de requerimientos

Teniendo en cuenta cada uno de los casos de uso detectados en la captura de requerimientos, es necesario determinar qué o cuáles componentes serán los encargados de realizar cada una de las funcionalidades requeridas. El siguiente diagrama de paquetes muestra los componentes por los que se compone el sistema completo, su composición por otros sub-paquetes y la relación entre los mismos :

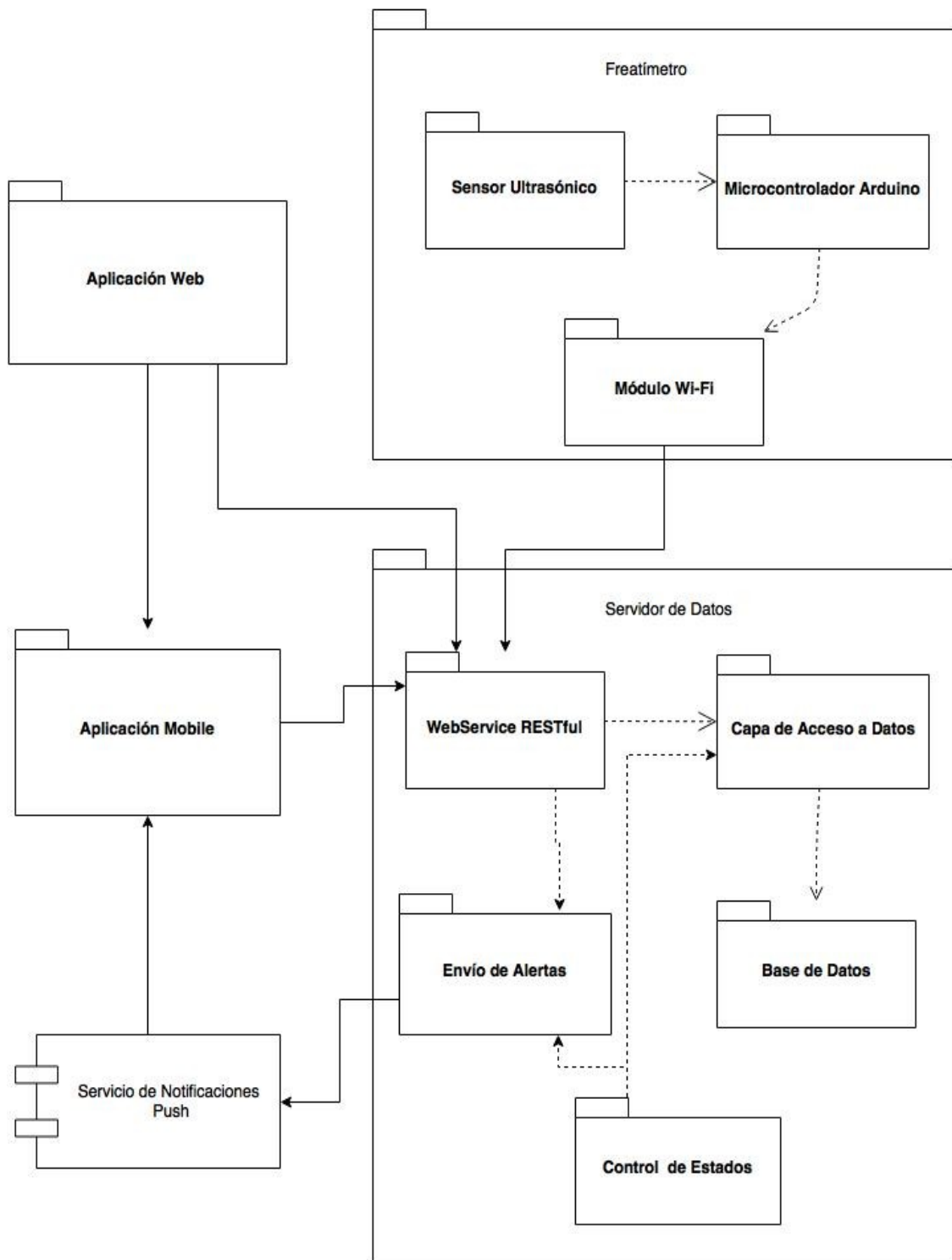


Ilustración 1: Diagrama de Paquetes del sistema Completo, con sus respectivos sub-paquetes

La estrategia de la utilización de un servidor de Datos externo a las Aplicaciones Web (Monitorizar los datos) y Móvil (Dar de alta los medidores y recibir las alertas), las cuales se comunican por medio de objetos JSON hacia el Webservice RESTful, permite centralizar todos los datos en un solo lugar, como también la estrategia de inserción, actualización, borrado o consulta de los mismos. La única manera de acceder a los datos es por medio de los métodos que se establecen en el Webservice.

Otra ventaja importante de esta estrategia de diseño, es que permite agregar fácilmente nuevos módulos al sistema. Por ejemplo, si en un futuro se requiere la implementación de aplicaciones móviles nativas que realicen el trabajo de mostrar los niveles actuales e históricos (Requerimiento de la aplicación web) solo será necesario obtener los datos desde el Webservice, invocando a los métodos del mismo.

Aplicación Web:

Este paquete desplegará una interfaz web que permite consultar los datos actuales de los niveles freáticos de cada uno de los freatímetros, los datos históricos que se representarán en forma de gráficos, la posición de cada freatímetro en un mapa e información acerca de cada uno de los freatímetros. Los datos se obtendrán haciendo peticiones hacia el Servidor de datos, más precisamente hacia el módulo que contiene el Webservice RESTful, el cual actúa como interfaz de entrada, por medio de peticiones HTTP y objetos JSON tanto para las peticiones como para interpretar las respuestas.

Freatímetro:

La interacción de sus tres componentes principales permiten comunicar al servidor el nivel freático en donde fue instalado el freatímetro. El microcontrolador Arduino actúa como intermediador entre el sensor ultrasónico y el Módulo de Comunicación Wi-Fi, determinando en qué momento se debe ejecutar la medición (Por medio de un Timer interno, previamente configurado), interpretando su respuesta y enviándola a través del Módulo Wi-Fi hacia el servidor de datos. Nuevamente el Webservice RESTful será el encargado de recibir la misma y procesarla internamente.

Aplicación Móvil:

El paquete de Aplicación móvil, tendrá las funciones de dar de Alta un Freatímetro, evento en el cual se deberá comunicar hacia el Servidor de Datos, por medio del Webservice que como en los casos anteriores, actuará como interfaz. Para realizarlo deberá enviar su Localización geográfica (Utilizando el hardware del dispositivo), el identificador del Freatímetro y un nombre que lo represente. También tendrá función de recibir las alertas en forma de notificaciones Push, por medio de un Servidor de Notificaciones Push externo, que actúa como interconector entre el paquete de Envío de Alertas del Servidor de Datos y la aplicación.

La relación entre la Aplicación Web y la Aplicación móvil, es que la segunda contendrá un fácil acceso a la primera, aprovechando su capacidad de ser responsiva a dispositivos móviles como tablets y smartphones.

Servidor de Datos

1. **Webservice RESTful:** Actúa como interfaz entre el Servidor de datos y cualquier módulo o paquete externo que desee comunicarse con el mismo. Recibe peticiones HTTP. La forma de envío de datos y sus respuestas a las primeras, es mediante el formato JSON. De esta manera, abstrae de los tipos de tecnologías o lenguajes que utilizan los demás elementos que se comunican hacia el Webservice. Se comunica con el paquete de Capa de Acceso a datos para almacenar persistentemente la información o para requerir datos y luego enviarlos, en forma de JSON, como respuesta a las peticiones recibidas.
2. **Capa de Acceso a Datos:** De la misma manera que el Webservice RESTful actúa como interfaz entre cualquier elemento externo y el Servidor de datos, este paquete actúa como interfaz entre cualquier paquete interno y la Base de datos que requiera datos de la misma. Se abstrae de la tecnología de Base de Datos que se esté utilizando (Si es SQL, o NoSQL, por ejemplo) o del motor de base de datos de la misma. La Capa de Acceso a Datos recibe peticiones que pueden tener objetos propios del lenguaje, y su respuesta también será en forma de objetos del lenguaje.
3. **Base de Datos:** En forma de tablas, se contiene los datos de forma persistente para su explotación. Se comunica con el paquete de Capa de Acceso a Datos, el cual manejará las consultas a realizarse, las inserciones, actualizaciones o eliminación de Datos.
4. **Envío de Alertas:** A fin de comunicar a los dispositivos móviles los eventos que deban ser notificados, el paquete de Envío de Alertas deberá comunicarse con el módulo externo de Servicio de notificaciones Push, para que luego este, envíe la alerta a los dispositivos. Se realiza también mediante una comunicación HTTP. Su relación con el paquete de Webservice RESTful permite que este, al recibir el nivel freático por parte de alguno de los freatómetros, comunique al primero el nivel freático recibido, y este determinará si es necesario realizar el envío de notificaciones o no, y el tipo de alerta.
5. **Control de Estados:** Cada vez que la interfaz del Servidor de Datos recibe datos desde los freatómetros, este paquete deberá actualizar los tiempos, para que una vez transcurridos los mismos, se actualicen los estados de los Freatímetros dados de alta, en caso de que no se reciban nuevas mediciones, identificando así que alguno de los medidores ha dejado de funcionar o que por alguna causa, no comunicó su nivel freático.

Como se puede ver, para resolver cada uno de los requerimientos del sistema, se producen interacciones entre cada uno de estos componentes, comenzando generalmente por una entrada de alguno de los actores identificados, y finalizando por una salida o respuesta en base a la primera.

Los siguientes diagramas de secuencia muestran la interacción de los componentes que resuelven cada uno de los requerimientos propuestos, desde el punto de vista del Sistema de Monitoreo.

1. Caso de Uso: Alta de freatómetro:

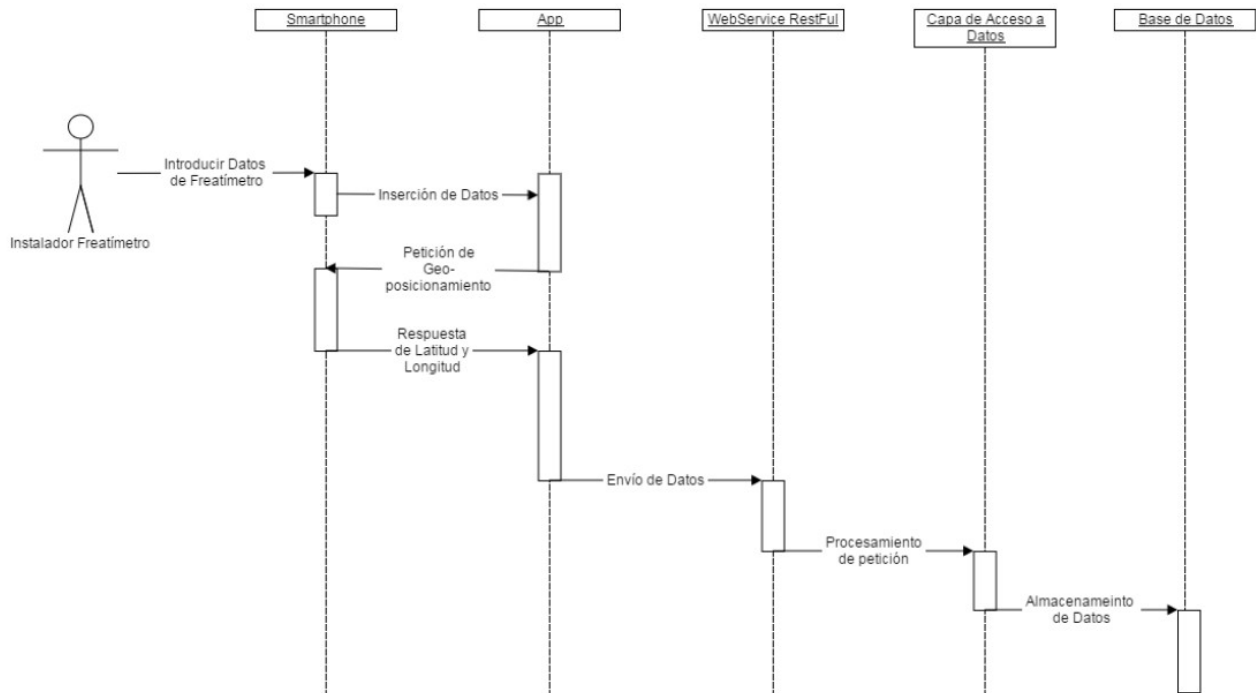


Diagrama de Secuencia para el Caso de Uso de Alta de Freatímetro

2. Casos de Uso: Visualización de niveles Actuales, Visualización de Niveles Históricos y Visualización de posición de Freatímetros.

En estos tres casos de uso, se puede considerar que la interacción entre los componentes es la misma, ya que más allá de que los datos sean distintos, intervienen siempre los mismos componentes, de la misma manera:

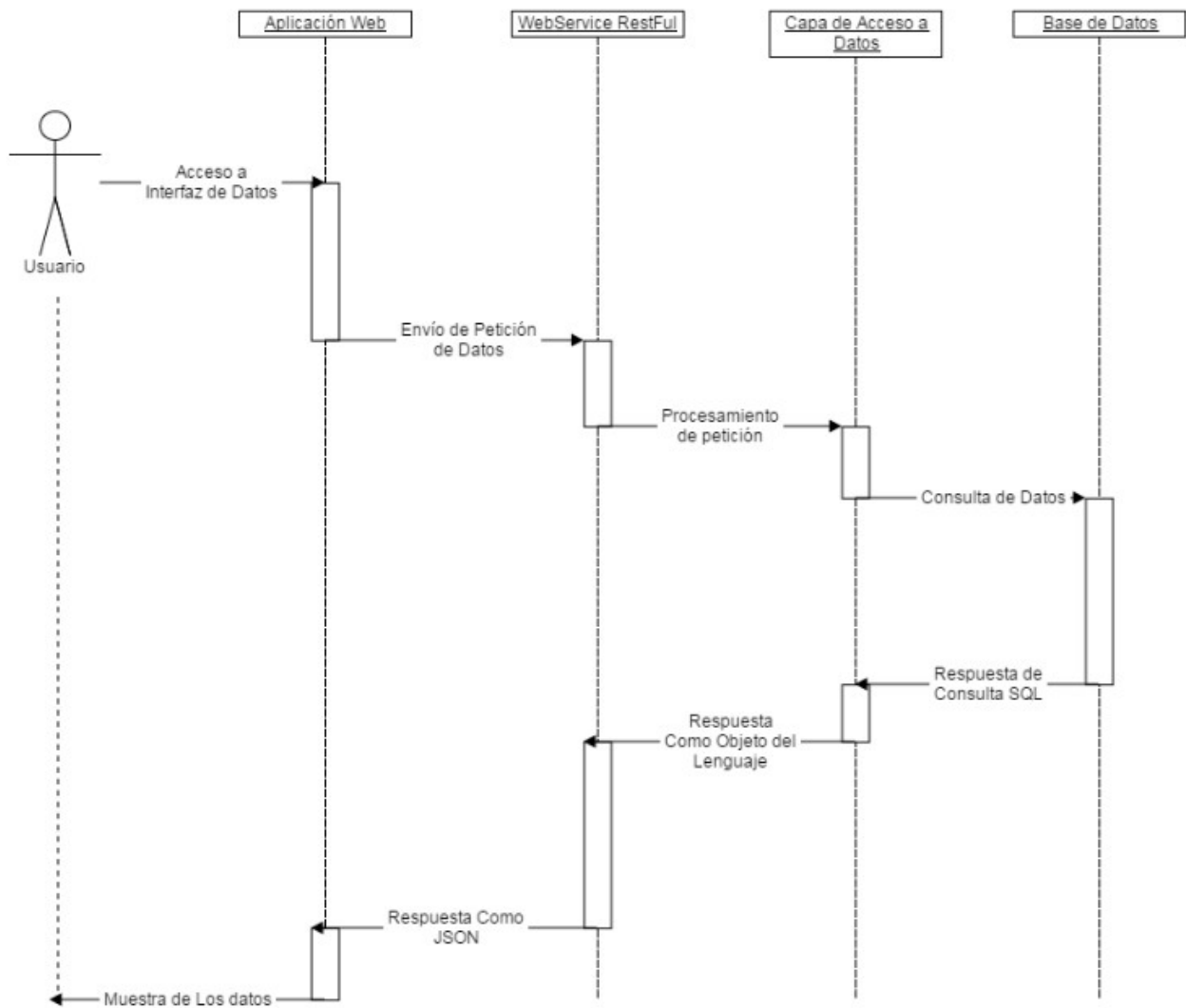


Diagrama de Secuencia para los Casos de Uso: Visualización de niveles Actuales, Visualización de Niveles Históricos y Visualización de posición de Freatómetros

3. Recepción de mediciones

Como se detalló anteriormente, para el Sistema de monitorización, cada freatómetro es un agente externo al mismo, por lo que se lo considera un actor del sistema.

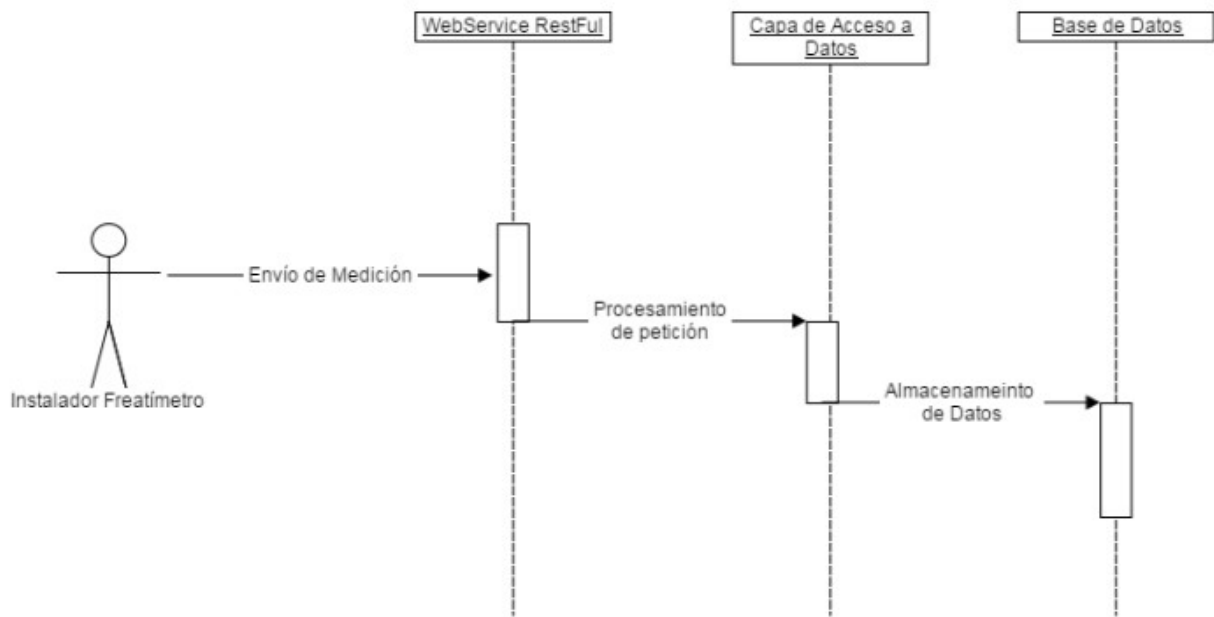


Diagrama de Secuencia para el Caso de Uso Recepción de Mediciones

4. Envío de Alertas

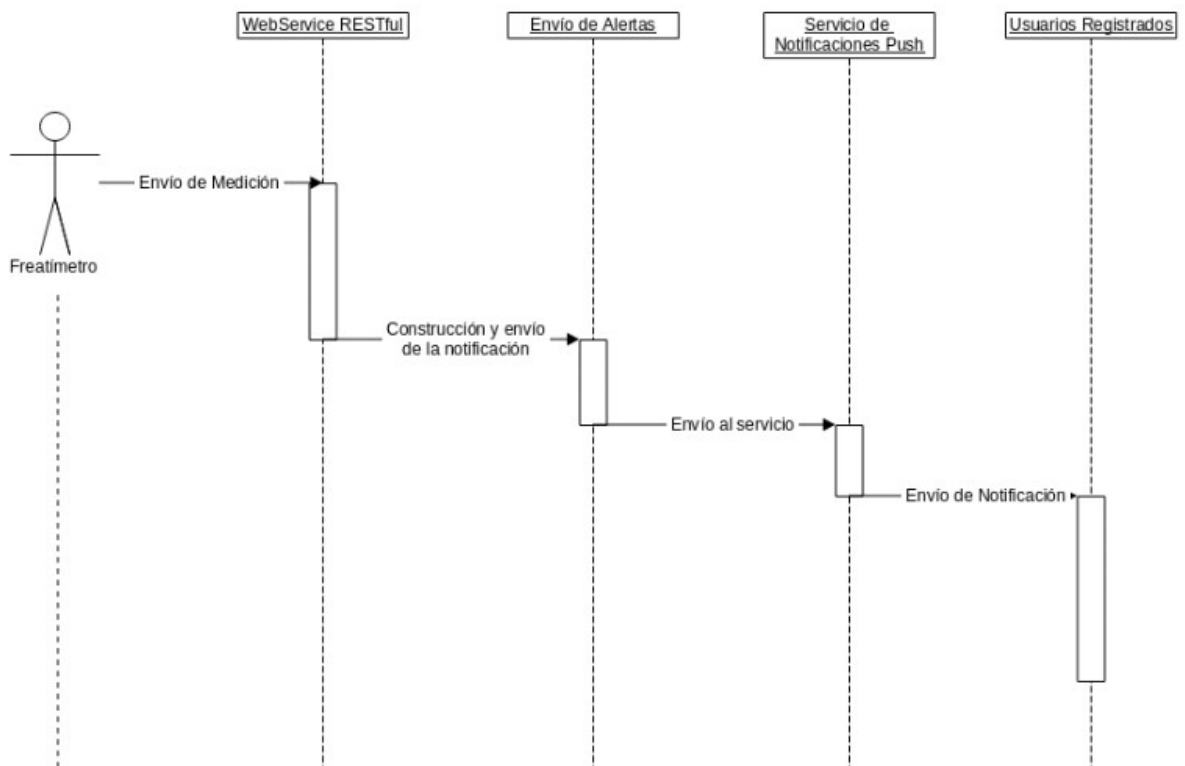


Diagrama de Clases para el Caso de Uso Envío de Alertas

4. Diseño

4.1 Diagrama de clases del Modelo

El siguiente diagrama de clases muestra la manera en que se representarán aquellos entes del mundo real que se hacen presentes en el sistema. Interactuando con el mismo o siendo parte de él, su representación en forma de clases permite una mayor claridad en el diseño del sistema, identificando sus atributos y métodos. Además, de una manera u otra, los lenguajes que se utilizarán en su implementación, serán todos orientados a objetos, por lo que esta manera de representarlos permite una muy buena aproximación para que finalmente sean materializados en código. Cada atributo que compone una clase, tiene un nombre y un tipo:

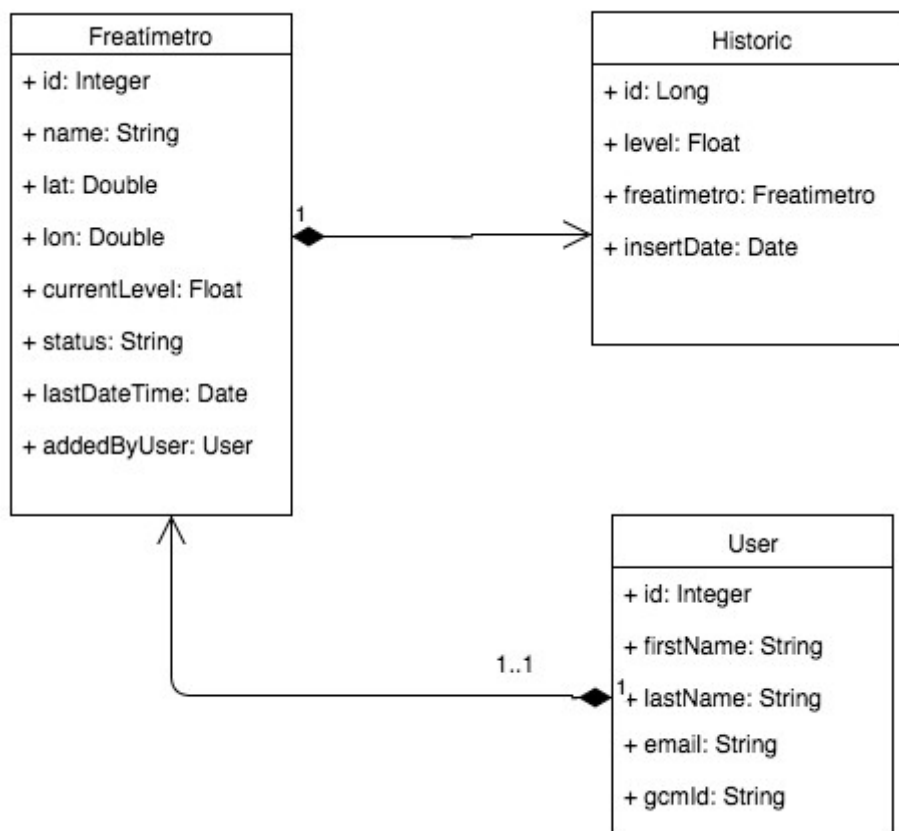


Diagrama de Clases del Modelo propuesto

Freatimetro:

- id : (Integer) : Identifica unívocamente a un Dispositivo Freatímetro.
- name : (String) : Nombre autodescriptivo, para una mejor identificación del Freatímetro.
- lat : (Double) : Latitud en donde fue instalado.
- lon : (Double) : Longitud en donde fue instalado.
- currentLevel : (Float) : Último nivel freático registrado.

- status : (String) : Representa el estado en el que se encuentra el Freatímetro. Init, Activo, Alerta y Error son los valores que harán referencia al estado actual en el que se encuentra el mismo.
- lastDateTime : (Date) : La fecha y horario en que se registró la última medición.
- addedByUser : (User) : El usuario que dio de alta el freatímetro a través de la aplicación Móvil.

User:

- id : (Integer) : Identifica unívocamente a un Usuario.
- firstName : (String) : Nombre del usuario.
- lastName : (String) : Apellido del usuario.
- email : (String) : Dirección E-mail del usuario.
- gcmlId : (String) : El ID del Servicio de Google Cloud Messaging, el cual se debe utilizar para enviar notificaciones Push a los dispositivos móviles, con las alertas que ejecute el sistema.

Historic:

- id : (Long) : Identifica unívocamente una medición de nivel freático.
- level : (Float) : Nivel freático registrado.
- freatimetro : (Freatímetro) : El freatímetro al que corresponde este nivel freático registrado.
- insertDate : (Date) : Fecha y Hora en la que se registró la medición de nivel freático.

Es importante aclarar, que los atributos `currentLevel` y `lastDateTime`, pertenecientes a la clase `Freatimetro`, se podrían obtener de la relación con la clase `Historic`. La decisión de tener estos campos en esta clase, es por una cuestión de rendimiento, ya que será muy frecuente requerir todos los freatímetros dados de alta, con su nivel actual. Con este diseño, al momento de realizar esta consulta será necesario solamente consultar a la Base de Datos sobre una sola tabla. De forma contraria, cada vez que se requiera realizar esta consulta se deberá también acceder a la tabla `Historic` y buscar el último registro perteneciente a un determinado freatímetro.

Como desventaja, cada vez que ingrese un nuevo registro de nivel freático, será necesario entonces realizar la actualización de ambos atributos a fin de mantener la consistencia, además de realizar la inserción de un nuevo objeto `Historic`.

Tanto el Servidor de datos como las aplicaciones Web y Móvil, deberán implementar estas clases, ya que forman parte del modelo y serán requeridas en su funcionamiento.

4.2 Diagrama de clases del Servidor de Datos

Las clases `FreatimetroService`, `HistoricService` y `UserService` conforman el paquete de `WebService RESTful` en el diagrama de paquetes. Su función es servir de interfaz entre las aplicaciones externas que se comunican hacia el servidor por medio del protocolo HTTP. De esta manera, mediante la definición de sus métodos, quedan definidos los métodos del `WebService` y solo a través de ellos, será posible interactuar con el servidor

de Datos. Quedará luego, en la implementación del mismo, definir de qué manera se pasarán los parámetros de cada uno de los métodos.

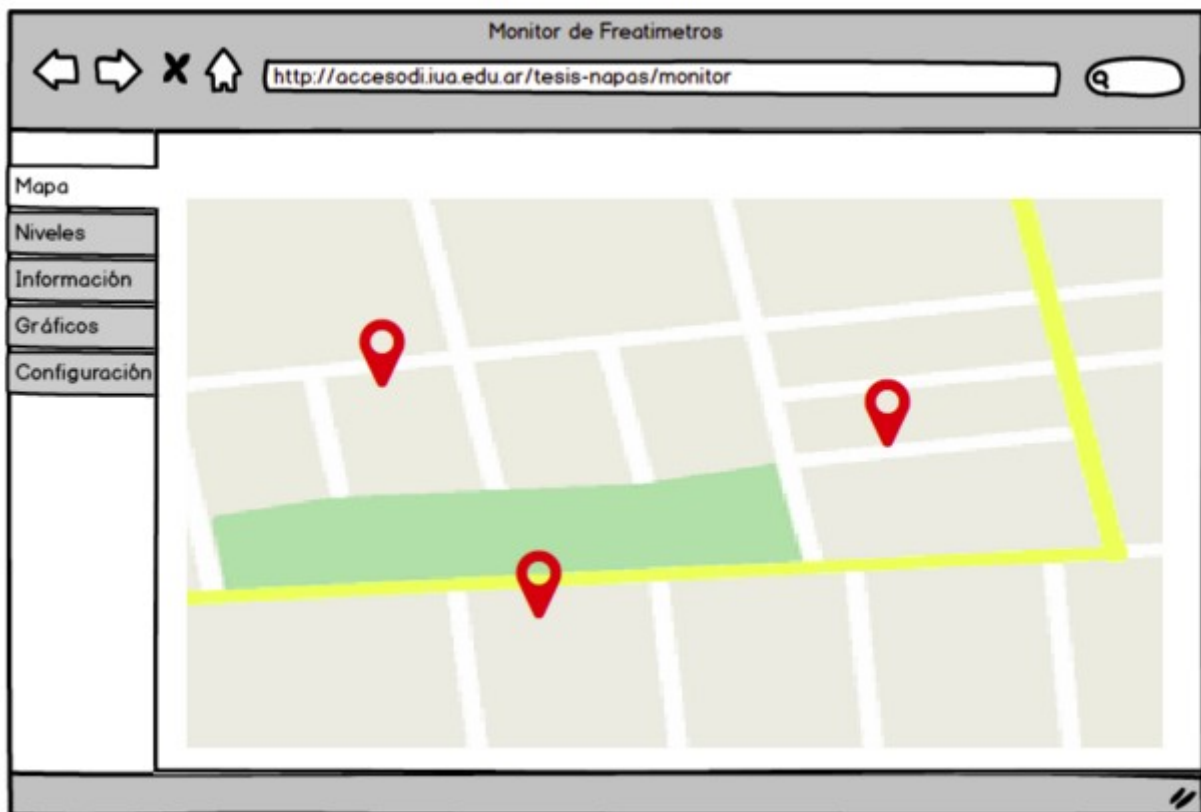
Las clases FreatimetroDAO, UserDAO e HistoricDAO pertenecen al paquete de Capa de Acceso de Datos (La terminología DAO al final de cada uno del nombre de las clases, hace referencia al término Data Access Object, nombre que generalmente se le da a las clases que interactuarán con la Base De Datos), su función es poder realizar operaciones sobre la Base de datos para la clase del modelo que representan. De esta manera, la clase FreatimetroDAO por ejemplo, tiene la funcionalidad para actualizar, insertar, eliminar o realizar consultas en la Base de Datos sobre la clase Freatimetro del modelo.

La clase sendAlert tendrá la responsabilidad de enviar a los dispositivos de los usuarios dados de alta, por medio del servicio de notificaciones Push, las alertas con respecto al estado de los freatímetros, como también sobre ciertos niveles freáticos registrados que sean necesarios ser notificados.

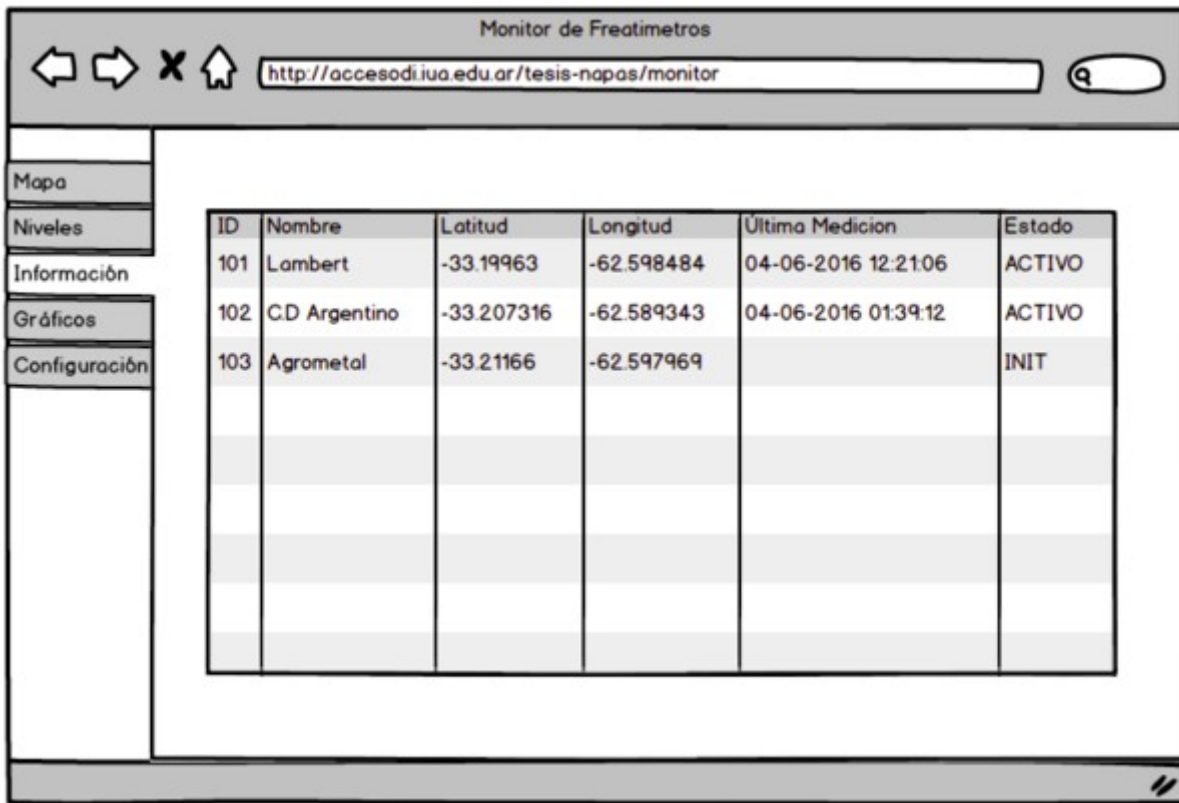
5. Diseño de Interfaces Gráficas

Una buena aproximación hacia la materialización del diseño propuesto, es también realizar maquetas o esquemas acerca de cómo se deberían ver las interfaces gráficas de aquellos componentes que tendrán interacción con el usuario. Para el caso de este proyecto, se han definido esquemáticamente cómo se verán tanto las interfaces de usuario de cada una de las pantallas tanto para la aplicación de monitorización web como la aplicación móvil.

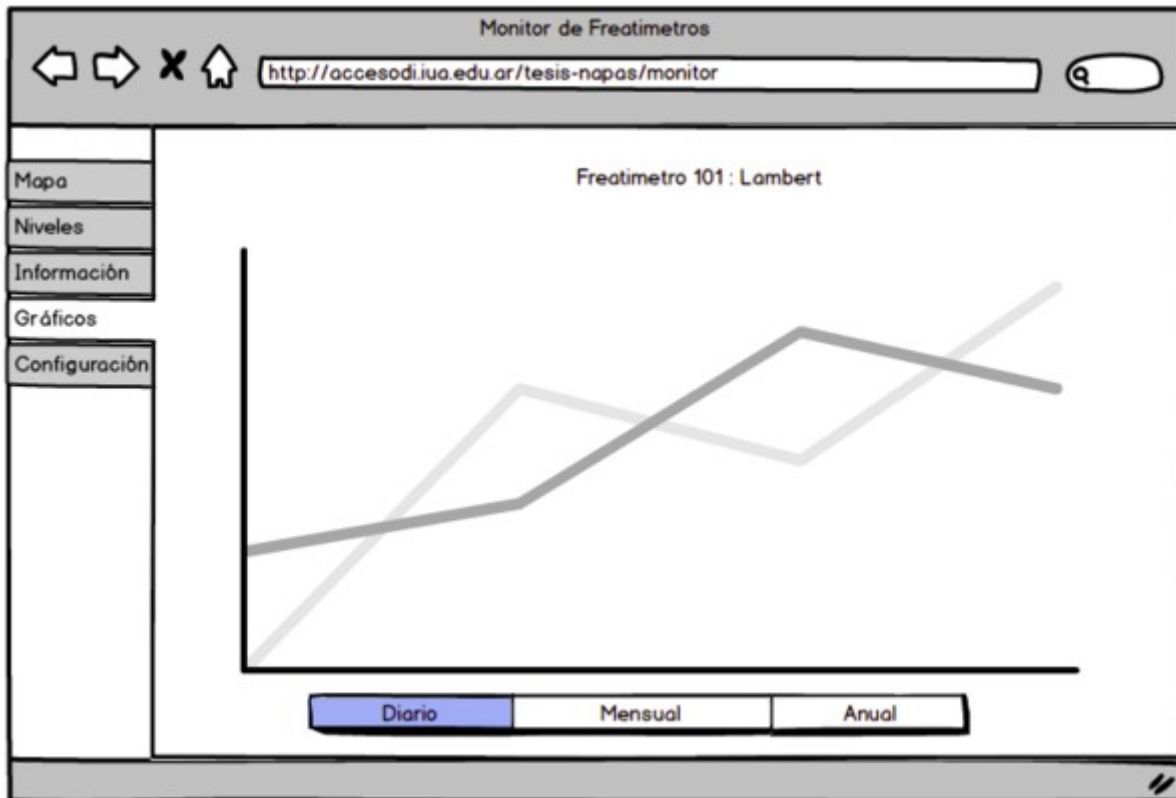
Monitor Web:



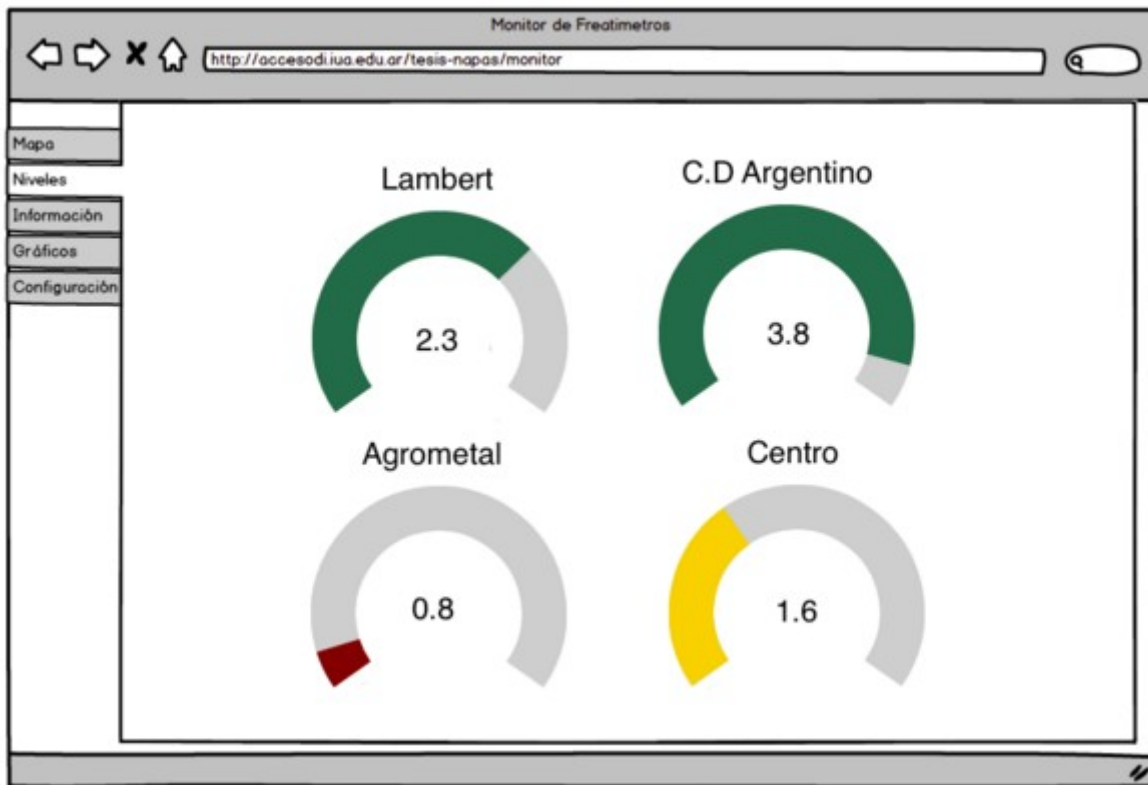
Esquema de la vista Mapa del monitor web



Esquema de la vista Información del monitor web



Esquema de la vista Gráficos del monitor web



Esquema de la vista Niveles del monitor web

Aplicación Móvil:



Maqueta de como se deberá ver la aplicación Móvil.

Cuarta Parte: Concreción del Modelo

1. Implementación

1.1 Implementación de Prototipo de Freatímetro :

Estructura y entorno:

Estructuralmente, el prototipo de freatímetro construido está basado en el propuesto por el I.N.T.A y presentado anteriormente. Se trata de un modelo a escala, el cual está compuesto por un tubo de PVC de 11 cm de diámetro y 1.40 m de altura. El tubo de PVC se encuentra ranurado para permitir el ingreso y salida de agua.

En su parte superior, se encuentra, bajo una tapa de PVC y dentro de una caja estanca, el circuito electrónico en el cual se encuentran los componentes que conforman el sistema electrónico de medición mediante ultrasonido.

En su parte inferior, un cuenco en el cual ingresa el tubo de PVC, preparado para entrar y salir agua mediante canillas, actuará como simulador del nivel freático. De esta manera, la variación de agua dentro del cuenco deberá ser capturada por el sensor de ultrasonido. El modelo final propuesto es como el que se muestra en la siguiente imagen:

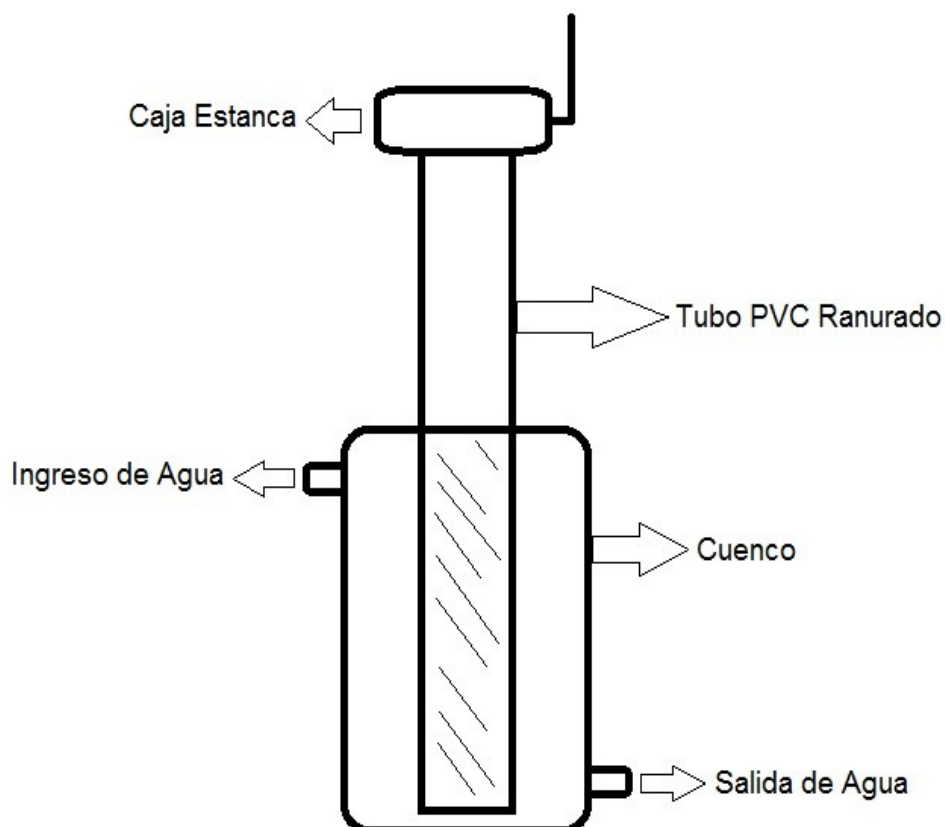


Diagrama del prototipo de freatímetro propuesto

Es importante destacar que el prototipo trabaja con una fuente de alimentación conectada a la red eléctrica. Debido a que se ha dejado fuera del alcance el estudio y las maneras de alimentación, teniendo en cuenta que los freatímetros instalados en el suelo generalmente no poseerán una fuente de alimentación desde una red eléctrica.

Componentes Electrónicos:

Sensor de Distancia Ultrasónico:

El componente seleccionado para actuar como sensor de distancia mediante ultrasonido, teniendo en cuenta los requerimientos que se identificaron anteriormente, es el modelo Micropik HC - SR04, el cual posee estas características:

- Voltaje: DC 5 V
- Corriente de trabajo: 15mA
- Frecuencias de trabajo: 40Hz
- Rango máximo: 4m
- Rango mínimo: 2cm
- Dimensiones: 45*20*15mm



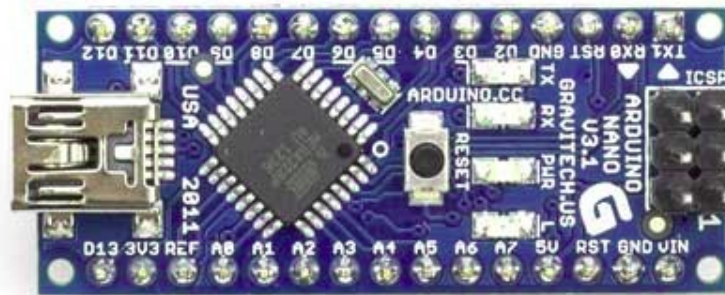
Sensor de distancias por Ultrasonido Micropik HC - SR04

Además de cumplir con los rangos máximos y mínimos de distancias que se requieren, este módulo es ampliamente utilizado en proyectos junto a microcontroladores Arduino, por lo que se pueden aprovechar las buenas experiencias de estos proyectos y sus resultados positivos para mitigar ciertos riesgos. Otra ventaja, es que es un componente económico y de fácil acceso.

Microcontrolador:

El microcontrolador que se decidió utilizar, es un Arduino Nano. Al igual que el módulo de ultrasonido, este microcontrolador posee como ventajas su fácil acceso al mismo y un costo relativamente bajo, además de ser ampliamente uno de los microcontroladores del proyecto Arduino más utilizados. Sus características más destacadas son:

- Microprocesador ATmega328 con cargador de inicio preprogramado.
- Tensión de entrada (recomendada): +7 a + 12 V.
- Tensión de entrada (límites): +6 a + 20 V.
- Tensión de trabajo : 5 V.
- 14 pines GPIO (de los que 6 ofrecen salida PWM).
- 6 pines de entrada analógica.
- Corriente DC por pin de E/S: 40 mA.
- Memoria Flash de 32 KB (2 KB para cargador de inicio).
- SRAM de 2 KB.
- EEPROM de 1 KB.
- Puerto de comunicación Serial.
- Frecuencia de reloj: 16 MHZ.
- Dimensiones: 0,73" x 1,7" .

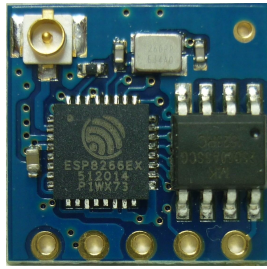


Microcontrolador Arduino Nano v2

Como se puede observar en sus características, este microcontrolador cumple con los requerimientos que se identificaron anteriormente; posee un puerto de comunicación Serial para ser utilizado en la comunicación hacia el módulo Wi-Fi, interrupción mediante temporizadores por hardware y puertos de entrada y salida tanto digitales como analógicos para la comunicación hacia el módulo de ultrasonido.

Módulo de Comunicación Wi-Fi :

El módulo de comunicación Wi-Fi seleccionado para el prototipo de freatímetro es el módulo Adafruit ESP8266 Serial WiFi Module (ESP-05):



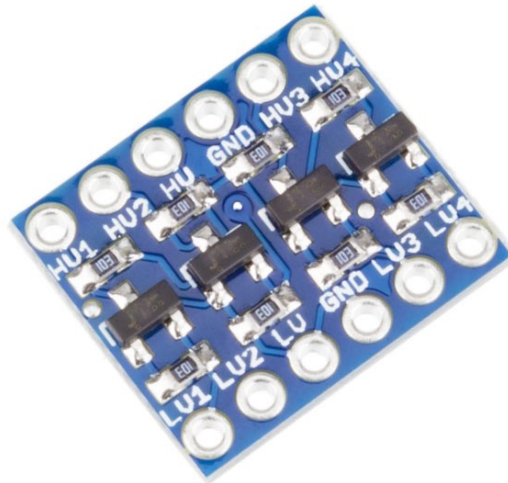
*Módulo de
Comunicación Wi-
Fi Adafruit
ESP8266*

- Interface Serial UART
- Norma : 802.11 B/G/N
- Tensión de trabajo: 3.3 V
- WIFI Direct (P2P)
- Trabajo bajo protocolo TCP/IP
- Comunicación mediante comandos AT
- Soporta Modos: Access Point, Station y Access Point+Station
- Conector para antena UFL

Adaptador de Tensiones (Level Shifter):

Debido a que, como se puede observar en las especificaciones anteriores, la tensión o voltaje de trabajo del microcontrolador Arduino Nano (5V) es diferente a la del Módulo Wi-Fi (3.3 V, como la mayoría de los componentes microelectrónicos actuales) será necesario adaptar la misma bidireccionalmente, para que la comunicación Serial entre los dos componentes sea posible. Sin esta adaptación de tensiones, al microcontrolador le ingresaría por medio de su puerto de entrada Serial un valor digital con una tensión menor a la de trabajo, y no será capaz de interpretar el valor discreto. En la otra dirección, el módulo Wi-Fi recibirá en su puerto de entrada Serial una tensión mayor a la de su tensión de trabajo, lo que podría dañar el mismo.

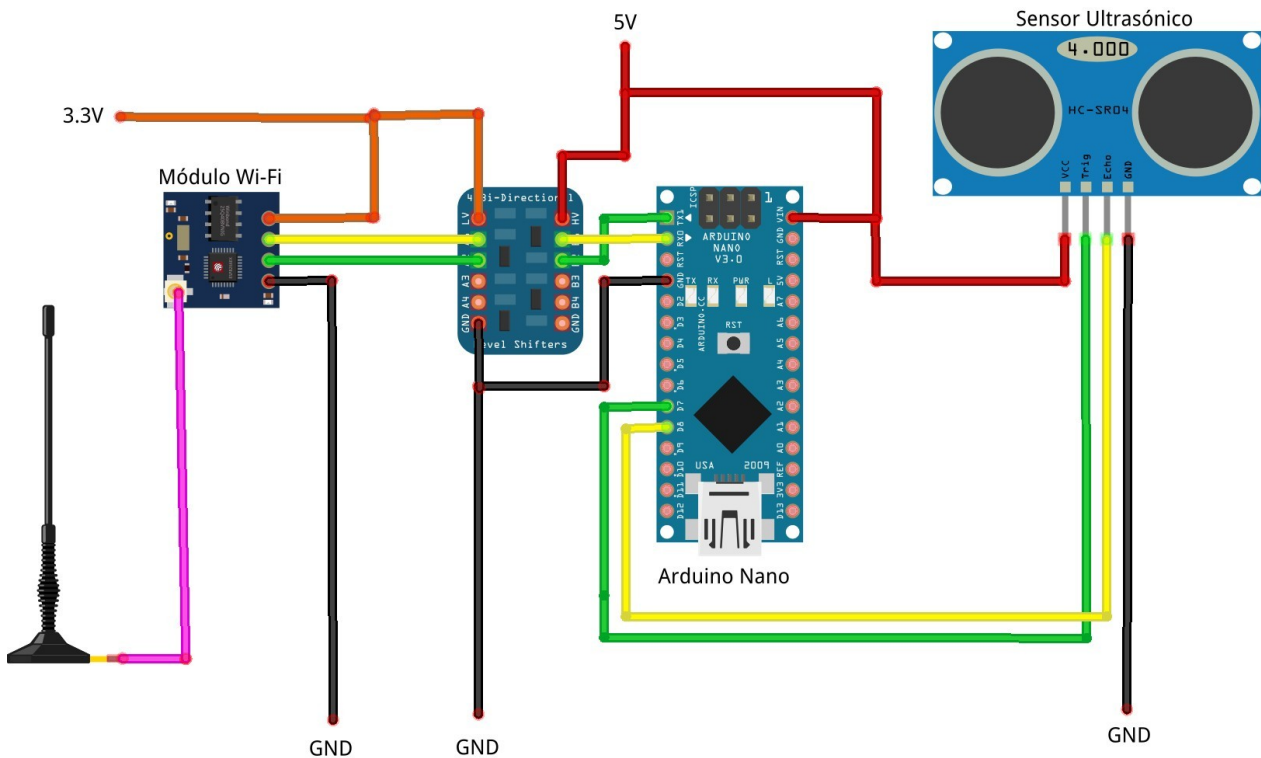
Mediante este componente, denominado Level Shifter, se realiza la conversión de tensiones para una comunicación digital. El componente seleccionado es un Addicore Logic-Level Shifter de 4 canales:



Adaptador de tensiones Addicore Logic-Level Shifter

Como se puede observar en la imagen superior, el mismo debe estar alimentado por ambas tensiones (tanto 5V como 3.3V), debido a que las utiliza como referencia para realizar la conversión. Las tensiones que ingresen por los canales de 5V, serán salida en sus respectivos canales de 3.3V, de forma inversa, aquellas tensiones que ingresen por los canales de 3.3 V, serán salida en su respectivo canal de 5V. De esta manera, tanto el módulo de comunicación Wi-Fi como el microcontrolador Arduino logran interpretar lo que el otro dispositivo envía, a pesar de que se haga con diferentes tensiones.

Esquema de conexión de Componentes:



fritzing

Esquema de Conexión de componentes, utilizando la herramienta Fritzing (www.fritzing.com)

Esquemáticamente, la imagen superior muestra cómo se interconectan los componentes anteriormente detallados para lograr el funcionamiento del Freatímetro electrónico. Como se puede observar, hay líneas de datos para la comunicación entre los componentes y líneas de tensión para alimentar los módulos y como referencia para las comunicaciones digitales. Debido a la necesidad de contar con dos líneas de tensiones diferentes, y que la de mayor tensión o voltaje es de 5V, queda definido que la alimentación del sistema completo deberá ser de 5V. Luego a través de un regulador de tensión, se debe adaptar la tensión hacia los 3.3V requeridos tanto para el módulo Wi-Fi como para la tensión de referencia necesaria en el Level Shifter.

Dentro del firmware en el microcontrolador Arduino Nano, deberá estar embebido un identificador numérico, que identifica unívocamente a cada uno de los freatímetros, conjunto al SSID y contraseña de la red inalámbrica a la que se deberá conectar el módulo Wi-Fi para la transmisión de los datos.

1.2 Implementación de Servidor :

El Instituto Universitario Aeronáutico ha brindando un servidor virtual para el alojamiento y despliegue de los módulos que deben ser accedidos desde Internet. El mismo cuenta con un sistema operativo Ubuntu 14.04 LTS. Su acceso para la administración del mismo se realiza mediante el servicio SSH, conjunto a la utilización de una red VPN.

La decisión de utilizar esta versión de sistema operativo, es principalmente debido a que por cuestiones de rendimiento, seguridad y operatividad, son ampliamente utilizados sistemas operativos basados en Linux por sobre otros sistemas para actuar en servidores. Ubuntu es un sistema operativo basado en Linux, ampliamente utilizado a nivel mundial y gratuito.

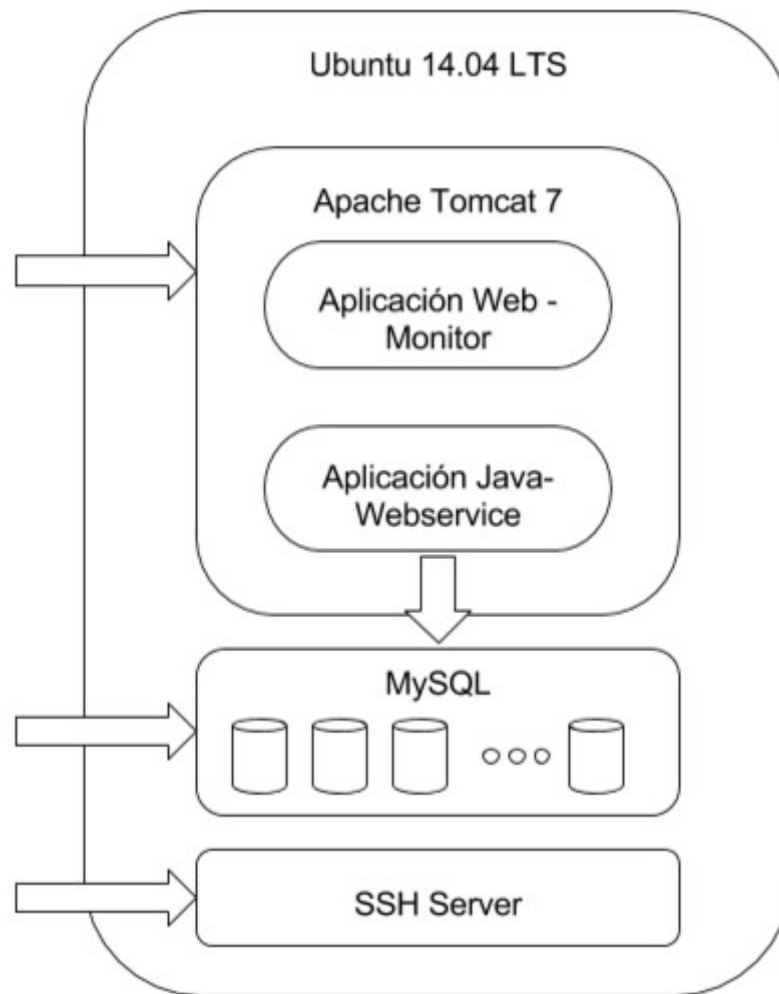
En el mismo, se encuentra un servidor Apache Tomcat 7 tanto para el despliegue como para servir los módulos. La aplicación web de monitoreo y el Webservice, podrán ser accedidos mediante este servicio.

La decisión de utilizar Apache Tomcat, es debido a que posee la capacidad de desplegar aplicaciones desarrolladas bajo Java, además de su fácil administración y amplia utilización.

Como Sistema de gestión de bases de datos, el cual permite el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos, se utiliza MySQL.

Se utiliza este sistema de gestión de bases de datos, ya que dentro de las opciones gratuitas, es una de las más potentes y utilizadas, además de que el autor del proyecto tiene experiencia sobre esta tecnología.

Esquemáticamente, con la adopción de estos componentes, el servidor donde se aloja el sistema es de la siguiente manera:



Esquema de Servicios y Componentes del servidor de datos

1.3 Implementación de Webservice:

Tecnologías:

El desarrollo del webservice se realizó bajo el lenguaje de programación Java, aprovechando la API de desarrollo para servicios web RestFUL JAX-RS, la cual es un módulo del entorno JAVA Enterprise Edition 6 (JAVA EE6).

El principal motivo de la decisión de utilizar esta tecnología es el conocimiento y la experiencia sobre esta tecnología por el autor de la tesis, estudiada a lo largo de la carrera de grado en varias materias de la misma, además de ser ampliamente utilizada a nivel mundial.

Teniendo en cuenta el esquema de módulos y paquetes planteado en la etapa de Diseño, para la simplificación del desarrollo de la capa de servicio, encargada de la recepción y respuesta de peticiones HTTP, se utiliza el framework **Jersey**.

Jersey provee una API propia que extiende del estandar JAX-RS para la producción de webservices RestFUL, el cual permite un desarrollo más claro, ágil y mantenible.

Para la capa de acceso a datos, encargada de realizar el acceso y las operaciones hacia la base de datos, se utilizó para su producción el framework **Hibernate**.

Hibernate es una herramienta de Mapeo objeto-relacional (ORM) para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en las clases de modelo de las entidades que permiten establecer estas relaciones.

De esta manera, la aplicación Java se abstrae totalmente de con qué sistema de gestión de base de datos se están realizando las operaciones de persistencia.

La aplicación se construyó bajo la herramienta **Maven**, la cual provee un entorno para la construcción, estructuración y administración de dependencias.

Maven utiliza un Project Object Model (POM) para describir el proyecto de software a construir, sus dependencias de otros módulos y componentes externos, y el orden de construcción de los elementos. Viene con objetivos predefinidos para realizar ciertas tareas claramente definidas, como la compilación del código y su empaquetado.

Trae consigo muchas plantillas de estructuras de aplicaciones para su ágil desarrollo, se utiliza la estructura clásica de una aplicación web, con su configuración estándar para ser desplegada en Apache Tomcat.

Implementación:

Los métodos declarados en el diseño de clases de la capa de servicios en la etapa de diseño, definen la interfaz del servidor de datos, mediante el protocolo HTTP. La definición completa de la interfaz, conocida como API, debe definir:

- El método de la petición (GET, POST, PUT, DELETE, etc)
- La URI o dirección del recurso
- Los parámetros a enviar (en caso de ser por medio del método GET, los parámetros se especifican junto a la URI)

Teniendo en cuenta esto, la API del Webservice quedó definida de la siguiente manera:

URI	Método	Parámetros	Descripción
/api/freatimetro/getAll	GET	--	Devuelve una lista con objetos de tipo Freatímetro
/api/freatimetro/insert	POST	{ id: Integer, lat: Double, lon: Double, Name: String }	Inserta un nuevo Freatímetro
/api/freatimetro/update	PUT	{ id: Integer, lat: Double, lon: Double, Name: String }	Actualiza un freatímetro existente

/api/freatimero/get?id=ID	GET	ID: Integer	Devuelve el freatímetro con el ID requerido
/api/freatimetro/setLevel?id=ID&level=LEVEL	GET	ID : Integer LEVEL: Integer	Envía una medición, este es el método al que accederán los freatímetros, enviando sus mediciones. El parámetro LEVEL es el nivel capturado, en centímetros.
api/historic/getDay?id=ID	GET	ID : Integer	Devuelve todos los registros de mediciones registradas por el freatímetro con el ID enviado, entre la fecha y hora de la petición y exactamente un día atrás.
api/historic/getMonth?id=ID	GET	ID : Integer	Devuelve todos los registros de mediciones registradas por el freatímetro con el ID enviado, entre la fecha y hora de la petición y exactamente un mes atrás.
api/historic/getYear?id=ID	GET	ID : Integer	Devuelve todos los registros de mediciones registradas por el freatímetro con el ID enviado, entre la fecha y hora de la petición y exactamente un año atrás.
api/user/get?id=ID	GET	ID : Integer	Retorna el usuario al que le corresponde el ID enviado.
api/user/insert	POST	{ firstName: String, lastName: String, email : String, password : String gcmlId : String }	Inserta un nuevo usuario.

api/user/update	PUT	{ Id : Integer firstName: String, lastName: String, email : String, password : String, gcmlId : String }	Actualiza los datos de un usuario.
-----------------	-----	---	------------------------------------

Con la utilización de Jersey, a las clases de la capa de servicio del webservice, solo se deben implementar las anotaciones correspondientes para que actúen como se ha definido anteriormente la API. Las anotaciones que provee Jersey para definir la API son las siguientes:

@PATH("Nombre del Path") : Define el Path o ruta hacia el recurso. Se define a nivel de clase.

@GET : Define el acceso al método mediante un método GET correspondiente al protocolo HTTP.

@POST : Define el acceso al método mediante un método POST correspondiente al protocolo HTTP.

@PUT : Define el acceso al método mediante un método PUT correspondiente al protocolo HTTP.

@DELETE : Define el acceso al método mediante un método DELETE correspondiente al protocolo HTTP.

@Produces(Tipo) : La respuesta del método de la clase, se traduce al tipo del parámetro. Para el caso de este proyecto, el tipo de respuesta será traducido siempre a objetos JSON.

@Consumes(Tipo) : Declara el tipo en el que se recibirá el cuerpo del mensaje HTTP cuando este es POST o PUT. También será siempre de Tipo JSON.

@QueryParam(Nombre del parámetro) : Se traducen los parámetros recibidos desde la URI, generalmente utilizados en peticiones GET.

De esta manera, estructuralmente las clases de la capa de servicio quedan definidas de esta manera:

FreatimetroService:

`@Path("/api/freatimetro")`

`public class FreatimetroService {`

`@GET`

`@Path("/get")`

`@Produces(MediaType.APPLICATION_JSON)`

`public Response get(@QueryParam("id") Integer id){`

`...`

`}`

`@GET`

`@Path("/getAll")`

`@Produces(MediaType.APPLICATION_JSON)`

`public Response getAll(){`

`...`

`}`

`@PUT`

```

    @Path("/update")
    @Produces(MediaType.APPLICATION_JSON)
    @Consumes(MediaType.APPLICATION_JSON)
    public Response updateFreatimetro(Freatimetro freatimetro){
        ...
    }

    @POST
    @Path("/insert")
    @Produces(MediaType.APPLICATION_JSON)
    @Consumes(MediaType.APPLICATION_JSON)
    public Response addFreatimetro(Freatimetro freatimetro){
        ...
    }

    @GET
    @Path("/setLevel")
    @Produces(MediaType.APPLICATION_JSON)
    public Response setLevel(@QueryParam("id") Integer id,
        @QueryParam("level") Integer level){
        ...
    }
}

```

HistoricService:

```

@Path("/api/historic")
public class HistoricService {

    @GET
    @Path("/getDay")
    @Produces(MediaType.APPLICATION_JSON)
    public Response getDay(@QueryParam("id") Integer id){
        ...
    }

    @GET
    @Path("/getMonth")
    @Produces(MediaType.APPLICATION_JSON)
    public Response getMonth(@QueryParam("id") Integer id){
        ...
    }

    @GET
    @Path("/getYear")
    @Produces(MediaType.APPLICATION_JSON)
    public Response getYear(@QueryParam("id") Integer id){
        ...
    }
}

```

UserService:

```

@Path("/api/user")
public class HistoricService {

    @POST
    @Path("/insert")
    @Produces(MediaType.APPLICATION_JSON)
    public Response getDay(User user){
        ...
    }
}

```

```

    }

    @POST
    @Path("/login")
    @Produces(MediaType.APPLICATION_JSON)
    public Response getMonth(RequestLogin login){
        ...
    }
    @GET
    @Path("/getUser")
    @Produces(MediaType.APPLICATION_JSON)
    public Response getYear(@QueryParam("id") Integer id){
        ...
    }
    @PUT
    @Path("/update")
    @Produces(MediaType.APPLICATION_JSON)
    public Response getDay(User user){
        ...
    }
}

```

Para la capa de acceso a datos, es necesario declarar que clases Java del modelo se corresponden con las tablas de la base de datos SQL, y que atributos de cada una de esas clases se corresponden con las columnas de cada tabla. Con el uso de **Hibernate** como framework, esta asociación también se realiza mediante anotaciones. De esta manera, las clases del modelo quedan establecidas de la siguiente manera:

Freatimetro:

```

@Entity
@Table(name = "FREATIMETRO")
public class Freatimetro {

    @Id
    @Column(name = "ID")
    private Integer id;

    @Column(name = "ID_NAME")
    private String name;

    @Column(name = "LAT")
    private double lat;

    @Column(name = "LON")
    private double lon;

    @Column(name = "CURRENT_METER")
    private float currentLevel;

    @Column(name = "STATUS")
    private String status;

    @Column(name = "LAST_DATE_TIME")
    private Date lastDateTime;

    @OneToOne

```

```

    @JoinColumn(name = "ID_USER")
    private User addedByUser;
}

```

Historic:

```

@Entity
@Table(name = "HISTORIC")
public class Historic {

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    @Column(name = "ID")
    private Long id;

    @Column(name = "LEVEL")
    private float level;

    @Column(name = "INSERT_DATE_TIME")
    private Date insertDate;

    @OneToOne
    @JoinColumn(name = "ID_FREATIMETRO")
    private Freatimetro freatimetro;
}

```

User:

```

@Entity
@Table(name = "USER")
public class User {

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    @Column(name = "ID")
    private Integer id;

    @Column(name = "FIRST_NAME")
    private String firstName;

    @Column(name = "LAST_NAME")
    private String lastName;

    @Column(name = "EMAIL")
    private String email;

    @Column(name = "GCM_ID")
    private String GCMid;
}

```

1.4 Implementación del Monitor Web**Tecnologías:**

La implementación del monitor Web fue realizada en Javascript, conjunto al framework AngularJS. Mediante la utilización de esta tecnología, la aplicación se ejecuta del lado del cliente, y luego consume los datos invocando a los métodos de la API definida en el

webservice para mostrarlos en un navegador web.

La decisión de optar por esta tecnología, es principalmente su agilidad para el desarrollo, código mantenible, y su adaptación para la implementación del patrón de Diseño Modelo-Vista-Controlador (MVC - Model-View-Controller).

La estrategia del patrón MVC consiste en definir una Vista por cada una de las pantallas definidas en la etapa de diseño, la cual contendrá los elementos necesarios para interactuar con el usuario. Cada una de estas vistas, tiene asociado un controlador, desarrollado en el lenguaje Javascript, utilizando AngularJS, el cual es el encargado de gestionar los eventos y las comunicaciones que se aplican en la Vista.

Los datos representados en cada una de las vistas, por medio de su controlador, serán objetos de las clases definidas en el diagrama de clases del modelo, en la etapa de diseño.

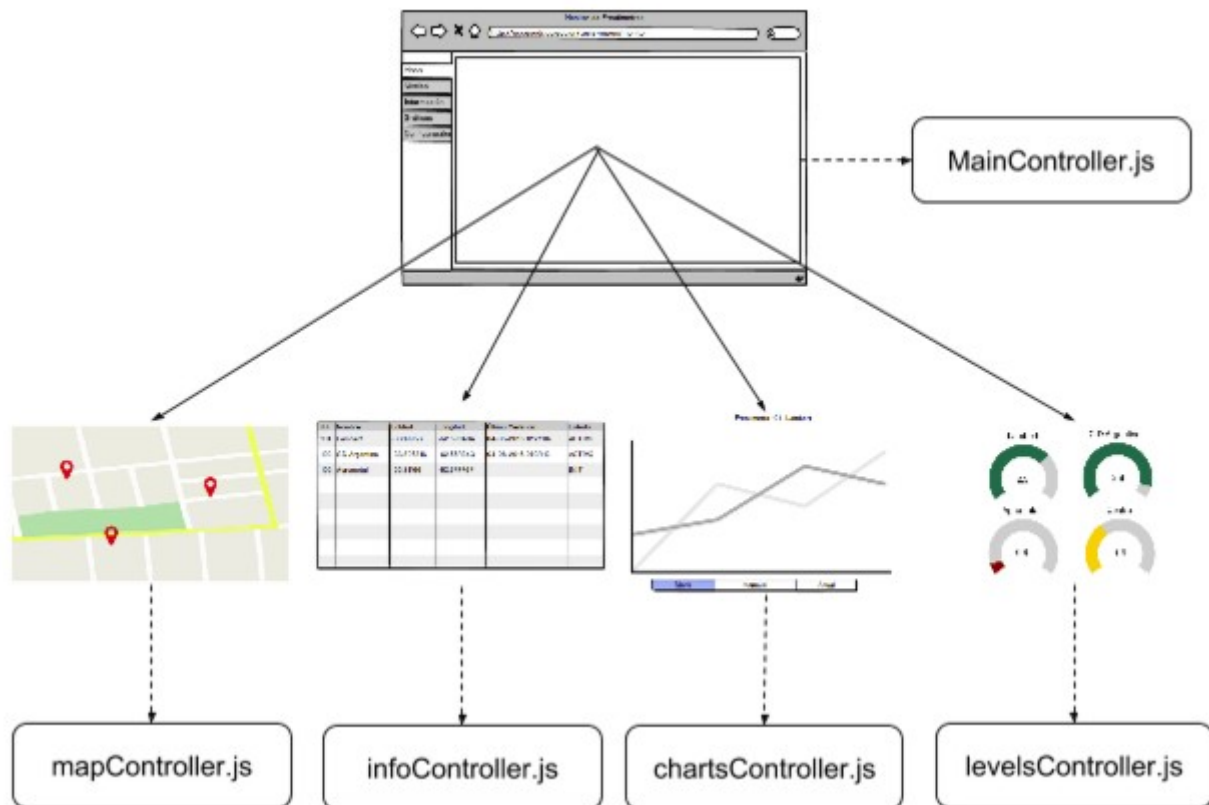
Implementación:

Teniendo en cuenta los diagramas de interfaces definidos, se debió implementar cada una de las vistas, con su respectivo controlador.

AngularJS posee la característica de definir un controlador padre, y varios controladores hijos al anterior, de manera tal que las propiedades que se definen en el controlador padre, son accesibles en sus controladores hijos, de la misma manera que sucede en la mayoría de los lenguajes de programación orientados a objetos, en los cuales existe la herencia.

Debido a que la mayoría de las vistas debe mostrar, de diferentes maneras, la información de los freatímetros dados de alta, y además, en el diseño de las interfaces de usuario, claramente se puede observar una vista padre, la cual contiene el menú que permite acceder a cada una de las vistas, la estrategia es requerir la información desde el controlador de la vista padre y luego, acceder a los datos desde cada uno de los controladores de las vistas hijas.

Esquemáticamente, la distribución de vistas y sus controladores es de la siguiente manera:



Distribución de vistas y sus Controladores de manera jerárquica

La estrategia de mantener un controlador padre, tiene la principal ventaja de que solo se realiza una petición hacia el webservice en el mismo, y luego estos datos son accesibles desde cada uno de los controladores hijos, por lo que aumenta considerablemente el rendimiento. Otra ventaja, es que AngularJS permite referenciar directamente desde los controladores hijos, las variables donde se almacenan los datos en el controlador padre. De esta manera, al actualizar los datos en el controlador padre, automáticamente se actualizarán también en las vistas manejadas por los controladores hijos.

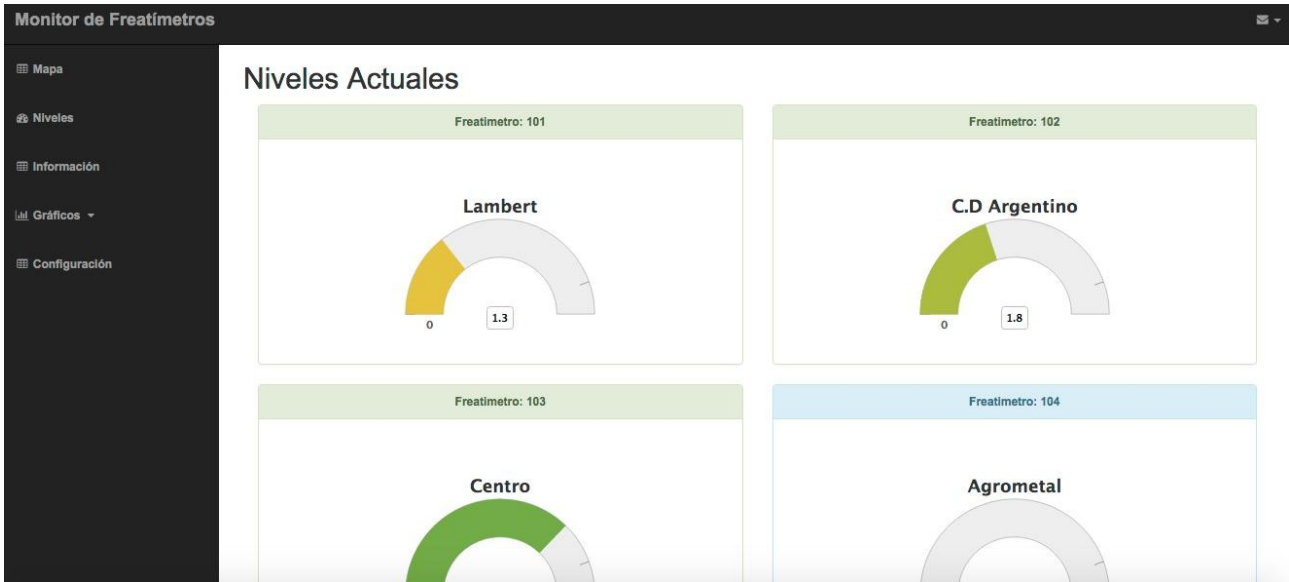
Así, solo el controlador `chartsController`, el cual maneja las vistas de los gráficos históricos de cada uno de los freatímetros, deberá implementar una llamada propia hacia el webservice para obtener los datos.

Debido a que la aplicación web deberá adaptarse a diferentes dispositivos, como computadoras, tablets, smartphones, televisores, etc. Se optó por un desarrollo de las vistas con la característica de ser responsivas. Esto significa, que las vistas se adaptarán automáticamente al tamaño de la pantalla y dispositivo en el cual se requiera la aplicación web.

Las vistas desarrolladas en HTML5, conjunto a hojas de estilo CSS, permiten que esto sea posible. La utilización del framework CSS Twitter Bootstrap, permitió realizar esta característica de manera ágil y muy óptima, además de brindar componentes amigables a la interfaz web. La misma es totalmente de código abierto y es actualmente la opción más utilizada a nivel mundial para el desarrollo de aplicaciones web basadas en vistas HTML.

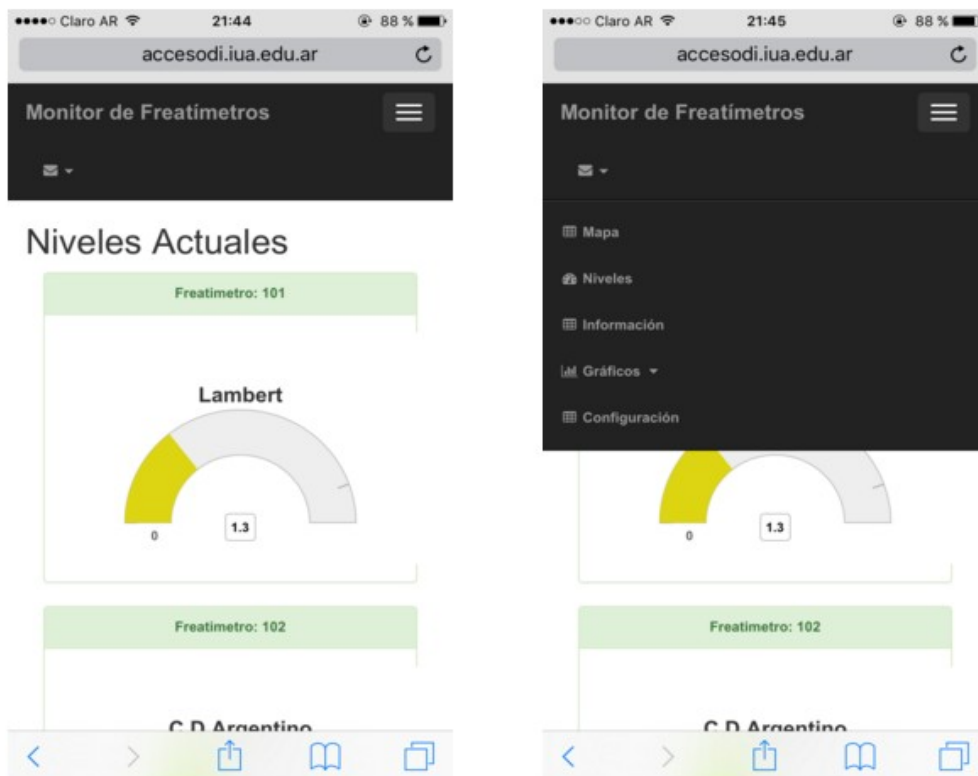
De esta manera, se obtienen vistas adaptadas a diferentes dispositivos, de la siguiente manera:

Desde Navegador web en Computadora:



Vista del monitor Web desde un navegador de una Computadora

Desde Smartphone:



1.5 Implementación de Aplicación Móvil

Como se detalló en la etapa de diseño, la aplicación móvil requerida tiene tres funcionalidades principales:

- Dar de alta los freatómetros, aprovechando la integración de GPS que poseen la mayoría de los smartphones de la actualidad.
- Recibir notificaciones sobre eventos importantes como alertas de niveles freáticos y cambios de estado de los freatómetros.
- Un acceso desde la aplicación móvil a la aplicación de monitoreo web, aprovechando su capacidad de ser responsiva.

Tecnologías:

El desarrollo de la aplicación se realizó bajo la plataforma **Android**. La principal decisión es debido a que es, junto a iOS, el sistema operativo más utilizado por smartphones en la actualidad y claramente el sistema operativo más utilizado por las personas tanto en el entorno en donde se desarrolla este proyecto, como en el país en general.

De la misma manera que con la aplicación web, el patrón de diseño Modelo-Vista-Controlador (MVC), se aplica en el desarrollo de la aplicación móvil.

Como se detalló en el diseño de las interfaces, se puede observar que la aplicación posee tres vistas principales, cada una con su correspondiente controlador que maneje a la misma.

Cada vista con su respectivo controlador en Android recibe el nombre de Actividad (Activity), y cada una de ellas es totalmente independiente de las demás. De esta manera, esquemáticamente, la asociación Vista-Controlador queda definida de la siguiente manera:



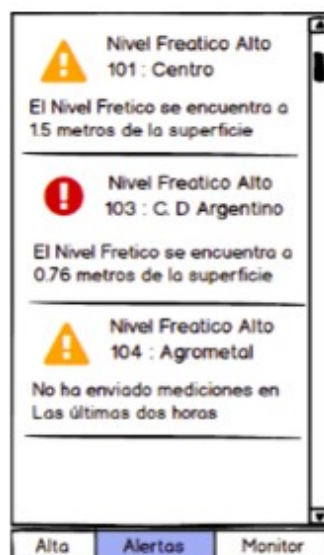
LoginActivity



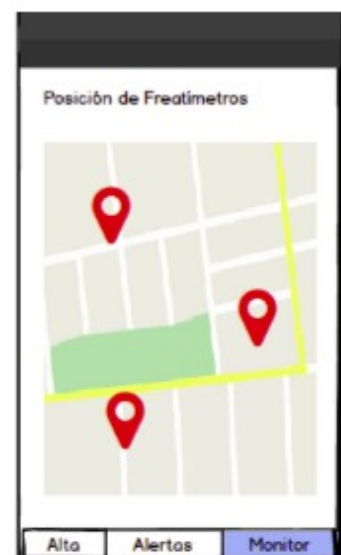
RegisterActivity



AltaActivity



AlertaActivity



MonitorActivity

Las vistas con sus respectivos Controladores, llamados Activity en Android, para la aplicación Móvil.

El controlador **LoginActivity**, permite el inicio de sesión de un usuario registrado, a través del ingreso de un nombre de usuario, y su correspondiente contraseña.

El controlador **RegisterActivity** posee un formulario de los campos requeridos para que un nuevo usuario sea dado de alta, luego de esto, si los datos ingresados son correctos y permitidos, se envía la petición de registro de un nuevo usuario al servidor y si es aceptado ya se podrá iniciar sesión con el mismo.

El controlador **AltaActivity**, realiza, además de controlar cada uno de los componentes visuales de la vista, la utilización del módulo GPS para obtener las coordenadas posicionales y el manejo de peticiones HTTP, para el envío de peticiones hacia el webservice, con el fin de dar de alta nuevos freatímetros.

El controlador **AlertaActivity**, muestra las alertas recibidas, almacenadas persistentemente. Es importante aclarar que también fue necesaria la elaboración de un Servicio (Service en Android) el cual se ejecuta en segundo plano, y almacena persistentemente cada una de las alertas recibidas.

Por último, el **MonitorActivity** solamente controla y direcciona la URL del monitor web al único componente que posee la vista, el cual es un **WebView**. El componente WebView permite mostrar una aplicación web de la misma manera que lo realiza un navegador web de un dispositivo móvil, embebida dentro de la aplicación. Permite un acceso rápido al monitor. De esta manera, se reutiliza el desarrollo del monitor web, aprovechando su capacidad de responsividad.

1.6 Envío y Recepción de Alertas

Las aplicaciones móviles desarrolladas en Android, poseen la capacidad de recibir alertas, conocidas como Notificaciones Push. De esta manera, se puede comunicar al usuario de un smartphone sobre algún evento. Existen varios servicios disponibles para realizar el envío de Notificaciones Push, pero en Android el más comúnmente utilizado y que mejor se adapta al sistema operativo es la plataforma de Google Cloud Messaging (GCM).

El sistema completo propuesto está basado en la arquitectura Cliente - Servidor. La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro agente, el servidor, quien le da respuesta.

Para el envío de alertas desde el servidor hacia cada uno de los clientes, que en este caso serán los dispositivos móviles, se requiere realizar el proceso inverso, es decir, que el servidor envíe un mensaje hacia los clientes sin recibir ninguna petición para contestar en consecuencia a la misma.

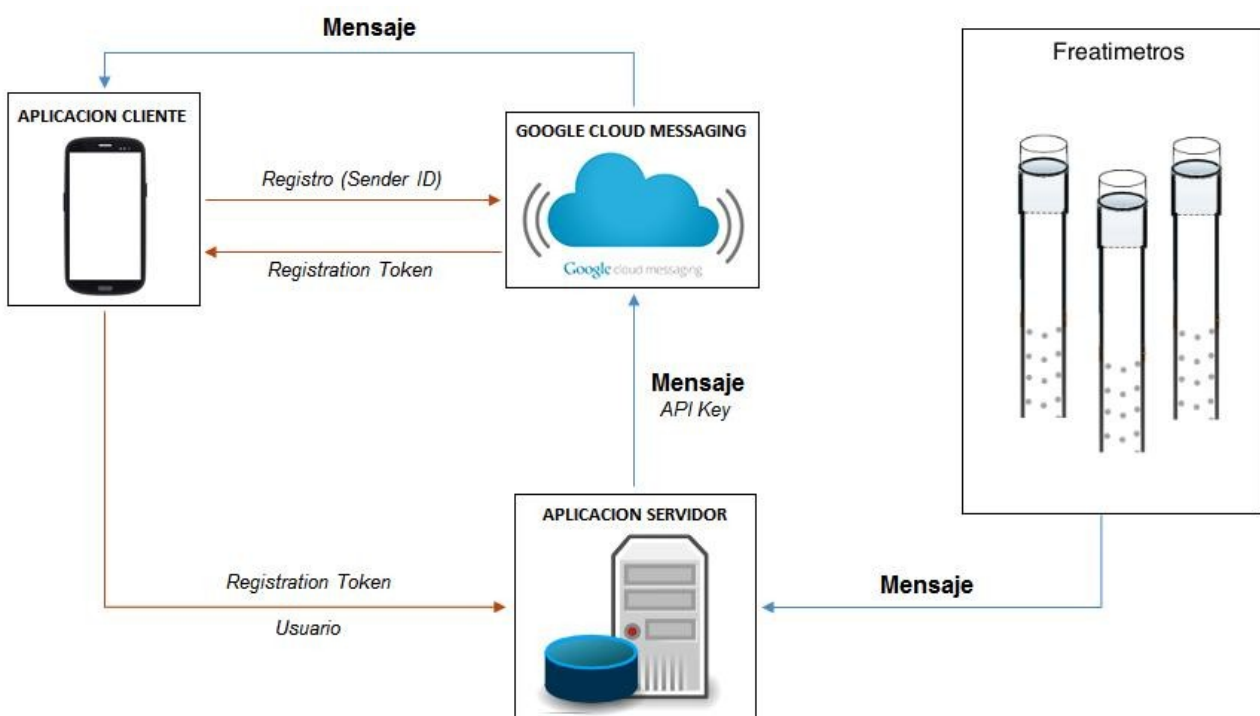
Es aquí donde entra en juego el sistema de Google Cloud Messaging (GCM), el cual actúa como intermediador entre el servidor y cada uno de los dispositivos clientes.

Cuando el servidor detecte que es necesario comunicar algún evento a los dispositivos clientes, este enviará un mensaje hacia la plataforma GCM, el cual comunicará el mensaje a cada uno de los dispositivos móviles.

En el mensaje, se debe comunicar a qué dispositivos clientes se debe enviar el mensaje, y el identificador de la aplicación móvil, conocido como API KEY.

Cuando un usuario se da de alta desde la aplicación Android, envía conjunto a sus datos, su identificador del servicio de notificaciones, conocido como Registration Token, el cual se toma directamente cuando la aplicación se registra en su servidor GCM. De esta manera, el servidor cuando requiera enviar alguna notificación hacia sus clientes, envía a la plataforma GCM una lista de todos los Registration Token registrados en la base de datos de los usuarios, y este se encarga de comunicar el mensaje a estos dispositivos móviles.

En la siguiente imagen se puede observar el proceso completo:



Esquema de funcionamiento del Sistema de Google Cloud Messaging

Para comenzar, la aplicación móvil Android cliente se registra en su plataforma GCM, y la misma devuelve un identificador, conocido como **Registration Token**, el cual es accesible desde cualquier contexto en la aplicación Android. Al darse de alta un usuario desde la aplicación móvil, se envían sus datos conjunto al Registration Token hacia el servidor, el cual almacena persistentemente esta información. En el momento que los Freatímetros dados de alta envían sus mediciones hacia el servidor y estas requieren ser notificadas a los usuarios, o el mismo detecta un cambio de estado en alguno de los freatímetros (Ya que por alguna razón no contestan), este envía un mensaje al servidor GCM, el cual contiene el API KEY de la aplicación y todos los Registration Token almacenados, de cada uno de los usuarios que se dieron de alta, conjunto a la información que se requiere informar. Este mensaje cuando es recibido por el Servidor GCM, automáticamente envía el mensaje a cada uno de los clientes móviles, los cuales lo reciben en forma de notificación Push.

2. Pruebas

2.1 Pruebas de Comunicación

La comunicación mediante el módulo Wi-Fi y un Access-Point, fue una de las pruebas que se llevo adelante, variando la distancia entre ambos, con la finalidad de obtener el porcentaje de paquetes perdidos con respecto a los recibidos, utilizando el comando Ping. De esta manera, se corrió el test durante cierto tiempo y los resultados obtenidos fueron los siguientes:

Distancia	Paquetes Enviados	Paquetes recibidos	Efectividad
2m	150	150	100%
4m	150	150	100%
8m	150	133	88%
12m	150	129	86%
15m	150	11	7%
20m	150	0	0%

El porcentaje de efectividad se calcula realizando :

$$\frac{\text{Paquetes Recibidos}}{\text{Paquetes enviados}} \times 100$$

Se puede observar claramente que la perdida de paquetes comienza a producirse entre una distancia de mayor de 8 metros. La conclusión entonces, claro está, es que la distancia no debe ser mayor que esos 8 metros debido a que se estarían perdiendo mediciones, aunque debido al porcentaje de efectividad, estas pérdidas pueden ser tolerables. A partir de los 12 metros de distancia la perdida es muy grande, por lo que no debería pasar.

Es importante aclarar que existen en el mercado una gran cantidad de módulos Wi-Fi con interfaz serial que se adaptarían fácilmente al circuito propuesto.

2.2 Pruebas de medición

La prueba quizás mas importante a realizarse, se trató de las pruebas de medición, con la finalidad de detectar y determinar, que tan preciso es tanto la tecnología de ultrasonido propuesta, como el sensor ultrasónico que se determinó utilizar.

La primera de las pruebas fue variar el nivel de agua dentro del prototipo, capturar la distancia mediante el módulo, y compararlos con la distancia real medida a través de una cinta métrica. Se tomaron muestras de cada una de las distancias, para determinar el comportamiento y los resultados obtenidos fueron los siguientes:

Distancia real	Distancia capturada	Error:
0.80 m	0.80 m	0 m
0.90 m	0.90 m	0 m
1.00 m	0.99 m	-0.01 m
1.10 m	1.10 m	0 m
1.20 m	1.21 m	0.01 m
1.30 m	1.31 m	0.01 m
1.40 m	1.40 m	0 m
1.50 m	1.52 m	0.02 m
1.60 m	1.60 m	0 m
1.70 m	1.58 m	- 0.02 m

Teniendo en cuenta que se tolera un error de distancia de hasta 0.10 m, se puede determinar que tanto la tecnología seleccionada como el sensor ultrasónico elegido son altamente precisos. Se registraron errores de no más de 0.02 metros.

2.3 Pruebas de variación en el tiempo

La ultima de las pruebas a realizarse fue mantener midiendo el freatímetro electrónico a una distancia fija, y observar su comportamiento a lo largo del tiempo. Aprovechando los gráficos para observar la variación del nivel freático histórico, que se desarrolló en el monitor web, se puede ver el comportamiento del freatímetro electrónico. Se enviaron durante cierto tiempo mediciones cada 20 segundos y los resultados obtenidos fueron los siguientes:

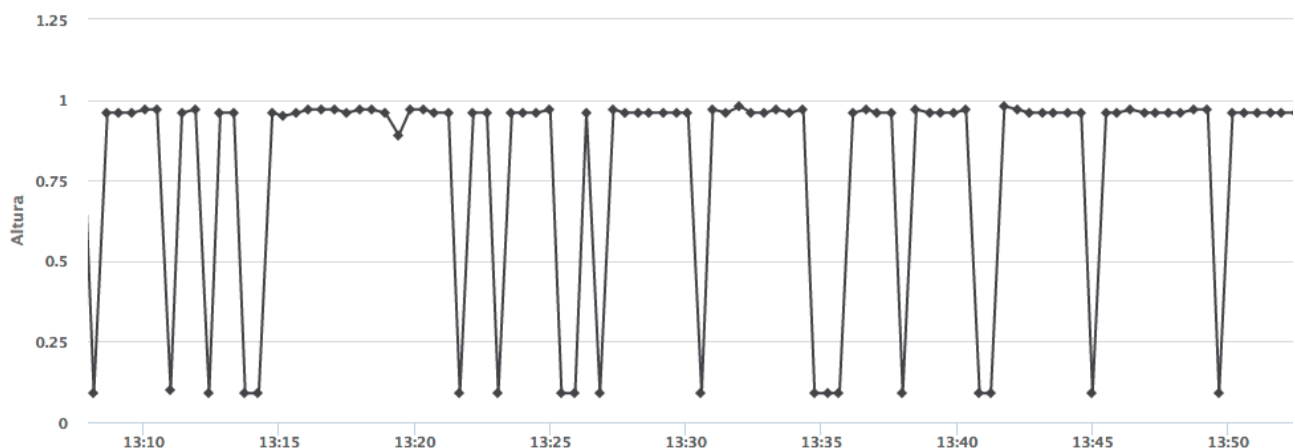


Gráfico de mediciones a lo largo del tiempo. Se pueden observar picos hacia abajo que no deberían hacerse presentes.

Como se puede observar, los resultados obtenidos muestran unos picos hacia abajo que nada tienen que ver con la distancia realmente medida. El problema detectado fue sobre la electrónica del medidor, mas precisamente en la corriente de alimentación del módulo.

Una vez corregido el mismo, los resultados obtenidos son los siguientes:

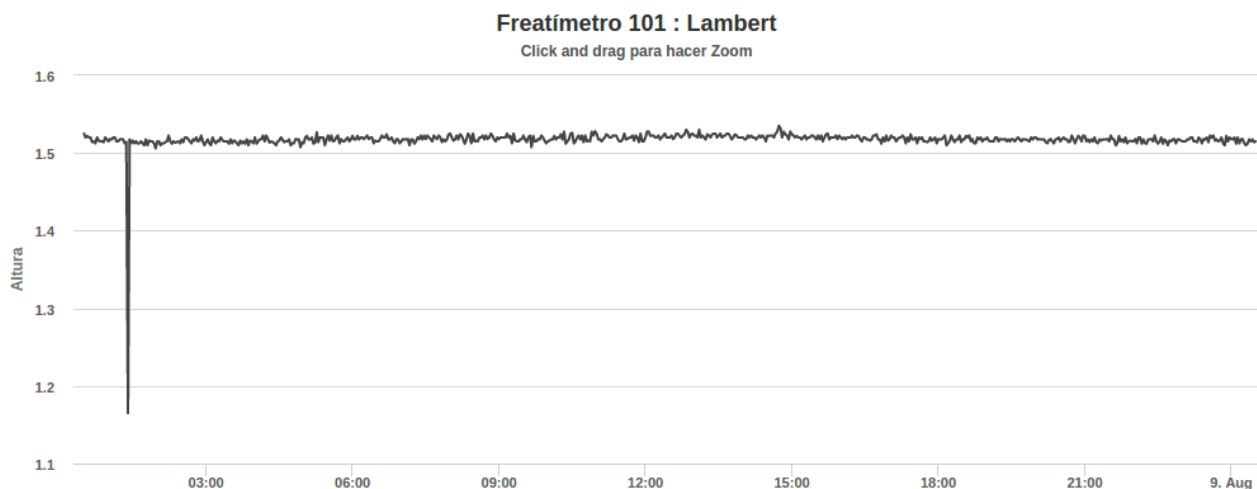


Gráfico histórico de mediciones, ya con las correcciones realizadas, refleja de mejor manera la realidad.

Luego de la corrección, ahora la variación en el tiempo si refleja más a la realidad. Como se puede observar, las variaciones no son de más de un par de centímetros, dentro del rango tolerable de error.

3. Puesta en marcha

Con el servidor preparado para comenzar a recibir datos, los pasos a seguir serán instalar cada uno de los freatímetros necesarios, este proceso lleva adelante los siguientes pasos:

1. Definir la ubicación del freatímetro
2. Asegurarse que en la ubicación existe una red Wi-Fi con el SSID y contraseña previamente configurada en cada uno de los freatímetros. En caso que no exista se deberá instalar un Access-Point que provea la red anterior. Los módulos Wi-Fi de los medidores soportan los protocolos de cifrado actualmente utilizados, como lo son WEP, WPA y WPA-2. Un punto muy importante, es que el Access-Point que brinde la red deberá también contar con el servicio de DHCP (Dynamic Host Configuration Protocol) ya que será necesario que le provean al módulo Wi-Fi la configuración de los parámetros de red necesarios, como lo son su dirección IPv4, máscara de sub-red y puerta de enlace.
3. La energía que alimentará a la electrónica del freatímetro, es uno de los temas que queda fuera del alcance del proyecto, la alternativa más óptima sería la utilización de paneles solares que mantengan alimentado al módulo. Baterías o un acceso a red eléctrica son algunas de las opciones que se podrían utilizar también. Es necesario que el freatímetro sea alimentado con una tensión de 5V, como quedó determinado en la implementación del módulo medidor, por las características de sus componentes.
4. Una vez definido el lugar y que el mismo cuente con los aspectos anteriores, será necesario realizar la perforación en el suelo y la posterior colocación del tubo de PVC ranurado, en el cual variará el nivel freático. En su parte superior, se debe colocar el módulo medidor completo.

- Es muy importante que la parte superior del tubo PVC, en donde se encuentra la caja estanca con la electrónica dentro, haya exactamente 30 cm hasta el nivel de suelo, ya que en el caso de no ser así y que la distancia distinta, se estarían capturando valores que no reflejan la realidad del fenómeno.
5. Antes de alimentar al módulo y que este comience su funcionamiento, se deberá utilizar la aplicación móvil en el lugar físico donde se instaló el freatómetro y proceder a darlo de alta en el sistema, cada freatómetro contendrá un identificador, el cual consta de un número, que se deberá colocar en el campo ID de la aplicación móvil.
 6. Una vez dado de alta el freatómetro en el sistema, se deberá alimentar de energía al freatómetro. El mismo se verá desde el monitor web en estado INIT, lo cual significa que se ha dado de alta, pero aún no recibió ninguna medición del nivel freático. Cuando este, envíe su primera medición hacia el servidor, pasará a estado ACTIVO.

La capacidad del sistema de detectar cuando un freatómetro, por alguna razón, ha dejado de enviar sus mediciones, a través de sus estados, y las notificaciones enviadas a los usuarios en base al cambio de los mismos, da la ventaja de que solo será necesario volver físicamente hacia el lugar donde fue instalado el freatómetro cuando se detecte que este no está enviando mediciones. Algunos de los problemas más comunes, que podrían hacer que esto suceda son:

- Problemas en el Access-Point vinculado al módulo Wi-Fi del freatómetro, ya sean eléctricos o de Red. En este caso será necesario chequear si el Access-Point se encuentra emitiendo la red específica, y si la misma posee acceso a Internet
- Problemas de alimentación en el freatómetro, lo cual hará que el mismo esté apagado. Será necesario entonces revisar las formas de alimentación hacia el freatómetro, y que las mismas poseen la energía con la tensión necesaria.
- Puede que por alguna razón, el freatómetro se encuentre con energía y conectado a la red Wi-Fi con acceso a Internet, aunque esté sin enviar datos ni ejecutando su rutina. Será necesario entonces reiniciar el mismo, desconectando y alimentando nuevamente el módulo medidor.

4. Prefactibilidad

Las ventajas que ofrece este sistema con respecto al resto de las formas que se utilizan actualmente para tener un control sobre el nivel freático de un territorio, lo ubican como una gran alternativa en el momento que se requiera la necesidad de contar con esta información.

El sistema ofrece una alternativa mucho más controlable, precisa, cómoda y accesible que el resto de las formas utilizadas actualmente.

Su aplicación en entornos urbanos es totalmente factible, ya que representa un costo muy accesible para los municipios de las localidades que requieran contar con el servicio.

Este sistema de monitoreo, con algunas modificaciones, se podría adaptar perfectamente a entornos rurales, con el fin de mantener un registro histórico y las referencias de los niveles freáticos actuales para tomar decisiones en el perfeccionamiento de las técnicas de siembra y cosecha.

Consultado a productores agropecuarios, el nivel freático en la actualidad es una variable a considerar y en muchos casos sólo se estima, y en otros se utilizan los modelos tradicionales de mediciones, los cuales no aportan ni la precisión, ni las ventajas que ofrece este proyecto.

Las modificaciones técnicas a tener en cuenta para la instalación de una red de freatímetros a lo largo de un territorio el cual se utilizará para actividad agrícola, en primer medida es reemplazar la tecnología de comunicación, ya que sería muy laborioso y costoso poder mantener una red Wi-Fi a lo largo del territorio, por lo que esta tecnología queda totalmente descartada. La alternativa que mejor se adaptaría, es reemplazar los módulos de comunicación Wi-Fi por módulos de comunicación GSM, utilizando las redes de telefonía para así comunicar las mediciones. Existen módulos GSM que funcionan exactamente de la misma manera que los módulos Wi-Fi propuestos en este proyecto, es decir, con una comunicación Serial hacia el microcontrolador y utilizando su propia tecnología para comunicarse hacia la red. Esta modificación se adaptaría bien para territorios con cobertura de redes de telefonía, las cuales no siempre cubren territorios lejanos a ciudades, como lo son los territorios dedicados para la actividad agrícola, debido a que, justamente, están pensadas para abastecer de servicio a gran cantidad de personas que se encuentran comúnmente viviendo en ciudades.

Otra alternativa, la cual cambia un poco la estructura de este proyecto, sería que cada medidor capture y almacene localmente sus mediciones, y luego mediante algún módulo de comunicación como el actual módulo Wi-Fi, actuando como Access-Point, o Bluetooth, una persona se acerque físicamente hasta cada medidor con un dispositivo móvil como un smartphone, tablet o notebook, y capture todas las mediciones realizadas por el medidor para luego insertarlas en el servidor. Para esta solución, se debe contemplar que es necesario algún medio de almacenamiento para salvar las mediciones, al menos hasta que se envíen. Con esta última alternativa se pierden algunas ventajas que ofrece el sistema propuesto.

Otra cuestión a tener en cuenta para su adaptación a entornos rurales, será las formas de alimentación de los módulos, ya que las maneras de realizarlo son mucho más restringidas que en el proyecto propuesto para entornos urbanos.

Conclusión

En la realización de este proyecto de tesis se aplicaron muchos de los conocimientos que fueron adquiridos a lo largo de todo el cursado de la carrera, abarcando programación, tanto en un lenguaje orientado a objetos como Java, y otro de scripting como JavaScript. La implementación de una base de datos, la programación de sistemas embebidos a placas, la aplicación de módulos externos a las mismas, diferentes tipos de comunicación entre componentes, el análisis de requerimientos, la utilización de elementos de documentación, la aplicación de patrones, la configuración de un servidor y la utilización de diferentes frameworks y herramientas que permitieron el desarrollo ágil de aplicaciones web y móviles.

Una de las conclusiones, una vez concluido el mismo, fue la capacidad de entender que en la actualidad, muchos fenómenos y eventos que ocurren en el mundo, pueden ser capturados, procesando esos datos y enviándolos hacia receptores, los cuales transformen esos datos en información que sirva para tomar decisiones y reducir la incertidumbre. Así como en este caso se mide el nivel freático de un determinado lugar, la experiencia al realizar el mismo, permite imaginar la posibilidad de interactuar con muchísimos eventos y fenómenos que ocurren a diario, sabiendo que en la actualidad, existe una gran cantidad de componentes que permiten, de manera muy sencilla en algunos casos, capturar estos datos y transformarlos en información.

Otra conclusión, es entender la importancia de un trabajo en equipo compuesto por varias personas, cosa que en este proyecto, por diferentes razones no se llevo adelante. La especialización de las personas en diferentes roles, relacionados con las distintas tecnologías que se utilizaron, hubiera permitido un desarrollo mas ágil y con menos complicaciones. Como ventaja de esto, la experiencia de haber estado en absolutamente todos los roles necesarios para llevar adelante el mismo, dan una visión del proceso completo y permiten entenderlo desde todos los puntos de vista.

Finalmente, la necesidad de tener que consultar a diferentes profesionales, y a raíz de esto, mantener varias comunicaciones con los mismos, sobre distintos conocimientos que se refieren al objeto de estudio en sí, fueron una experiencia muy positiva que permitieron entender la manera en la se debe desenvolver un profesional.

Bibliografía

Stallings, W. (2004). *Comunicación y Redes de Computadores*. Madrid: Pearson.

Margolis, M. (2011). *Arduino Cookbook, 2nd Edition*. New York: O'Reilly Media.

Seshadri, S. & Green, B. (2014). *AngularJS: Up and Running*. New York: O'Reilly Media.

Booch G., Rumbaugh J., Jacobson I. (2001). *Lenguaje Unificado de Modelado, 2da Edición*. Pearson.

Bertram, N. & Chiacchiera, S. (2015). *Ascenso de napas en la Región Pampeana: ¿Consecuencia de los cambios en el uso de la tierra?*. 2016, de I.N.T.A Sitio web: http://inta.gob.ar/sites/default/files/script-tmp-inta_napas_mjz_13.pdf

Oliva, E.. (2014). *El incremento de las lluvias, el cambio climático más importante en el centro de Argentina*. 2016, de UNCiencia, Universidad Nacional de Córdoba Sitio web: <http://m.unciencia.unc.edu.ar/2014/septiembre/climatologia-1941-2010/el-incremento-de-las-lluvias-el-cambio-climatico-mas-importante-en-el-centro-de-argentina>

Pérez De Diego, D. (2006). *Sensores de Distancia por Ultrasonidos*. 2016, de ALCABOT, Robótica de la Universidad de Alcalá Sitio web: <http://www.alcabot.com/alcabot/seminario2006/Trabajos/DiegoPerezDeDiego.pdf>

Requena, A. M. & Nordenstrom, G. . (2012). *Freatímetro de fabricación casera*. 2016, de INTA (Instituto Nacional de Tecnología Agropecuaria) Sitio web: <http://inta.gob.ar/documentos/freatimetro-de-fabricacion-casera>

HighSoft. (2016). *Highcharts, Highstock and Highmaps documentation*. 2016, de HighSoft Sitio web: <http://www.highcharts.com/docs>

Google. (2016). *Try Cloud Messaging for Android*. 2016, de Google inc. Sitio web: <https://developers.google.com/cloud-messaging/android/start>