

Field Remote Control

Diego Valentín Purgue

Universidad de Mendoza

Aristóbulo del Valle 1.215 San José – Guaymallén - Mendoza- Argentina

diegopurgue@hotmail.com

ABSTRACT

El objetivo de la presente investigación es desarrollar un sistema basado en telemetría, que permita realizar diversos controles remotos sobre variables ambientales, proporcionando una herramienta para controlar en forma remota cultivos alejados geográficamente del ámbito cotidiano. Esto permitiría que tanto ingenieros agrónomos como así también los propietarios de fincas, quienes actualmente deben viajar hasta sus cultivos, realicen su trabajo de control diario, como es el seguimiento y control de diferentes variables como por ejemplo: temperatura, haciendo posible realizar sus actividades con modalidad de teletrabajo reduciendo la contaminación ambiental al disminuir el tráfico vehicular y en consecuencia los costos particulares.

Dicha solución es económica, práctica y escalable, no solo se puede satisfacer una necesidad de manera económica, sino también ofrecer un punto de entrada a la investigación, mejora y desarrollo de soluciones innovadoras y prácticas para la motivación del teletrabajo aplicado a la agronomía y agricultura.

PALABRAS CLAVES

Tecnocultivos
Telemetría
Teletrabajo en Agronomía
Teleinformática
Economía
Social
Contaminación
Proyecto autosustentable

1. INTRODUCCIÓN

El objetivo de este trabajo es desarrollar un sistema que permita realizar diversos controles remotos en forma práctica y económica. Además permitiría que los ingenieros agrónomos o encargados de controlar cultivos, quienes deben viajar cotidianamente hasta los mismos, realicen seguimientos de diferentes variables de interés como por ejemplo: temperatura, humedad, presión, velocidad del viento, crecimiento de tallo, riego, etc.; posibilitando de esta forma que el usuario pueda realizar sus tareas de control y medición a distancia en donde se encuentre.

El sistema permite que el propietario de una finca o lugar de cultivo reciba, en horas críticas previamente definidas, los valores de temperatura que se registraron en ese período.

Independientemente de los reportes preestablecidos, se enviará una alarma al propietario en caso de que el sistema registre un valor de temperatura crítico u otra variable crítica para el cultivo bajo control.

El sistema consta de un "script" programado en el microcontrolador que será el encargado de procesar los valores de temperatura capturados por el sensor de temperatura y enviarlos vía "sms" utilizando un MODEM gsm al servidor donde los datos serán procesados por medio de una aplicación de escritorio. La misma se encargará de guardar los datos en una base de datos para usos estadísticos de la región, y de realizar la lógica de envíos de alarma, envío de sms (al propietario), e-mail, como así también actualizar los registros de últimos valores registrados. El sistema también posee una herramienta mediante la cual el administrador del sistema podrá realizar la administración de usuarios, equipos, cultivos.

Lo que motiva el desarrollo de este software es satisfacer la necesidad presente en los profesionales de la rama de Agronomía de poder controlar en forma práctica y económica variables micro-climáticas de sus cultivos, como así también posibilitar el teletrabajo para esta rama de la ciencia promoviendo el desarrollo agronómico y fortaleciendo el aprovechamiento máximo de recursos naturales como son las tierras alejadas imposibilitadas al acceso cotidiano por parte de la mano del hombre, sin dañar el medio ambiente, ya que los componentes utilizados no generan ningún tipo de perjuicio ni contaminación al mismo.

El problema es que actualmente dichos profesionales poseen incomodidad para poder tener un control de variables climáticas (en este caso temperatura) de sus propiedades, tampoco tienen la posibilidad de aprovechar los beneficios que puede aportar la telemetría, informática y la telecomunicación en la administración y control de las mismas,

provocando pérdidas millonarias por año en catástrofes climáticas que causan la pérdida de los frutos de los cultivos, perjudicando así la economía nacional. Actualmente dichos profesionales ó personal adjunto deben realizar múltiples, costosos y fatigadores viajes para poder realizar controles y mediciones necesarias para la toma de decisiones, y a veces no alcanzan a predecir el comportamiento climático para poder tomar medidas preventivas sobre una posible catástrofe climática.

Se observa también la fuerte demanda de la implementación de un sistema de “auto-protección” contra uno de los problemas más grandes en la agronomía: las heladas. Como el sistema detecta y advierte valores de temperaturas críticos para los diferentes cultivos, continuando el desarrollo de este proyecto se podría agregar la funcionalidad para activar automáticamente mecanismos de defensa como por ejemplo: ventiladores, riego por aspersión, calentadores, etc., provocando con esto que el dueño de la propiedad tenga siempre controlado y fuera de peligro sus cultivos.

Como propuesta de solución a estos inconvenientes se desarrollará este sistema basándose en conceptos de agronomía, informáticos y teleinformáticas.

2. DESARROLLO

1.1 Objetivo

El sistema sustentable a desarrollar permite el monitoreo remoto implementando gran parte de los fundamentos explicados con anterioridad. Esta aplicación permite recibir reportes de temperaturas medidas en forma remota y enviar las mismas a los usuarios correspondientes, también controla la relación temperatura – cultivo, de modo tal que si alguno de los valores medidos es un valor que sobrepasa el umbral de temperatura crítica para un determinado cultivo, el sistema genera automáticamente una alarma, la cual es enviada al usuario o usuarios previamente cargados en el sistema.

1.2 Plataforma de Desarrollo

Para el desarrollo del software se utiliza .NET 2005 como lenguaje de programación.

Se elige este lenguaje ya que su principal ventaja es la de tener interoperabilidad entre diversos lenguajes, esto es, que en Microsoft .NET cuando vamos a desarrollar una aplicación no tenemos que preocuparnos de elegir un lenguaje de programación adecuado que se ajuste mejor para el tipo de aplicación a desarrollar.

Tampoco estamos condicionados si en una empresa la mayoría de los programadores saben un único lenguaje de programación. Con Microsoft .NET todos estos asuntos quedan relegados a un segundo plano porque .NET “entiende” diversos lenguajes de programación y con todos ellos podemos llegar a realizar las mismas acciones e incluso desarrollar una aplicación que esté compuesta de módulos hechos en distintos lenguajes de programación teniendo una completa integración entre todos los módulos. La elección de un lenguaje no es una cuestión prioritaria y pasa a ser una mera elección de gustos del programador.

Como base de datos, se utiliza MySQL con su conector “MySql.Data.msi” para tener acceso desde la aplicación de .net, ya que es la base de datos de código fuente abierto más usada del mundo. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar. La extensiva reutilización del código dentro del software y una aproximación minimalística para producir características funcionalmente ricas, ha dado lugar a un sistema de administración de la base de datos incomparable en velocidad, compactación, estabilidad y facilidad de despliegue. Mientras que del lado del equipo se utilizó “C” como lenguaje de programación del microcontrolador, debido a que el equipo utilizado soportaba este lenguaje de programación.

1.3 Herramientas de Desarrollo

1.3.1 Diseño

Para la etapa de diseño se utilizara las herramientas “Enterprise Architect” y “ERwin”. Ambas son herramientas de modelado que soportan el diseño de software desarrollado en .NET (c#) permitiendo importar clases ya codificadas a diagramas y viceversa.

Con ellas se puede diseñar aplicaciones de gran tamaño y realizar todos los diagramas UML necesarios para una buena implementación.

1.3.2 Codificación

Para la codificación de estas aplicaciones se utiliza como entorno de desarrollo integrado Visual Studio .NET 2005. Este nos permite utilizar .NET Framework para desarrollar nuestras aplicaciones de una forma rápida y visual. En Visual Studio .NET podemos crear muchos tipos de aplicaciones, tanto aplicaciones de ASP.NET, de Servicios Web, como aplicaciones de acceso a bases de datos y todo desde el mismo entorno de desarrollo. Este entorno engloba todos los diferentes lenguajes que soporta el .NET Framework, posee barras de herramientas que se pueden “anclar”

(Docking) y ocultar para hacer más amplio el espacio de pantalla según nuestras necesidades.

Algunas de las características más importantes son:

- Capacidad de colapsar y ampliar partes del código para ver únicamente las partes que necesitamos mientras trabajamos en otras secciones del código, (uso de #region, #endregion).
- Una lista de tareas de lo que nos queda por hacer con el comentario que nosotros le pongamos. Al hacer doble clic en una entrada de la lista iremos a la parte del código correspondiente.
- Auto indentación que incluye las estructuras de control de manera automática.
- Mientras vamos escribiendo código el entorno nos muestra una lista con los métodos o propiedades con los que cuenta un objeto, e incluso con la lista de los parámetros de llamada de los métodos.

1.4 Arquitectura

El sistema FRC, “Field Remote Control” consta de dos subsistemas.

Uno de ellos es la aplicación, desarrollada en .NET, al a cual denominaremos controlador de datos o “handler” de datos, y es la que esta corriendo en un servidor con acceso a Internet, la misma es la encargada de recibir los datos enviados por el equipo remoto, procesar los datos, almacenarlos en la base de datos, chequear valores críticos, enviar reportes programados a usuarios y realizar la administración de usuarios, cultivos y equipos por parte del administrador u operador del sistema.

La otra parte del sistema es el componente remoto, sensor + microcontrolador + MODEM gsm, estos componentes son los encargados de tomar la temperatura, procesar los valores y enviarlos vía sms (de a pares) cada cierto intervalo de tiempo a un servidor de mails, donde el handler de información se encargara de recogerla cada cierto intervalo de tiempo para su procesamiento.

1.5 Características

La transmisión de datos entre el subsistema remoto y el handler de datos se realiza utilizando la tecnología SMS, el microcontrolador envía los datos vía este medio a un servidor de mails, en nuestro caso una cuenta creada en “gmail”, llamada remotefield@gmail.com; luego el envío de información entre el handler de datos y los usuarios se realiza mediante el envío de mails, ya sea a celulares o a cuentas de correo electrónico, según lo hayan especificado los usuarios.

1.5.1 Limitación del sistema:

La limitación del sistema es la seguridad, ya que los canales usados para transmitir los datos (sms y mail) no son canales seguros, no aseguran la confidencialidad de los datos ni la integridad de los mismos. Tampoco se asegura que los datos lleguen en tiempo y en forma ya que no es un sistema de tiempo real y el fin del mismo es informativo.

1.5.2 Inseguridad de transmisión:

El canal usado “SMS” en Latinoamérica no es un canal seguro de transmisión de datos ya que la infraestructura no lo permite, se utilizo este medio ya que la cantidad de información a transmitir es limitada y el fin del sistema es meramente informativo.

1.6 Funcionamiento

A continuación describiremos paso a paso como es el funcionamiento. Como ya se dijo anteriormente, este sistema consta de dos subsistemas los cuales funcionan en conjunto para permitir el monitoreo remoto del lugar.

Ahora mostraremos gráficamente como funciona el sistema en su totalidad.

1.6.1 Etapa I:

El subsistema remoto (compuesto por el microcontrolador, el MODEM GSM y el sensor de temperatura, tal como se muestra en la figura 1), es quien se encarga de recolectar los datos en su forma mas cruda directamente del lugar que se esta controlando. El sensor esta dispuesto a medir la temperatura del lugar donde se encuentra, en cuanto se lo indique el microprocesador.

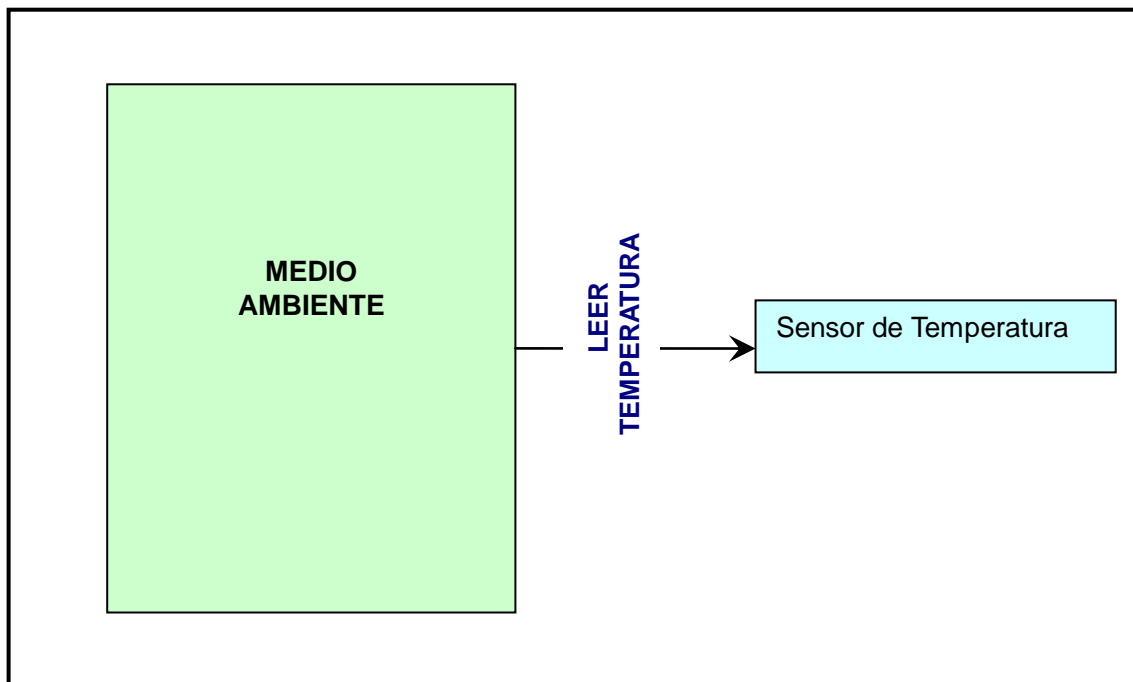


Figura 1. Sensor de temperatura.

1.6.2 Etapa II:

El microcontrolador, tal como se muestra en la figura 2, cada un cierto intervalo fijo de tiempo, envía un comando al sensor para que este lea el valor de temperatura y lo envíe, luego este toma este valor y los envía vía SMS, mediante el MODEM GSM, a un servidor de mail remoto, este proceso se realiza en forma constante y continua en cada uno de los subsistemas remotos instalados, independientemente de la tarea que este realizando el controlador.

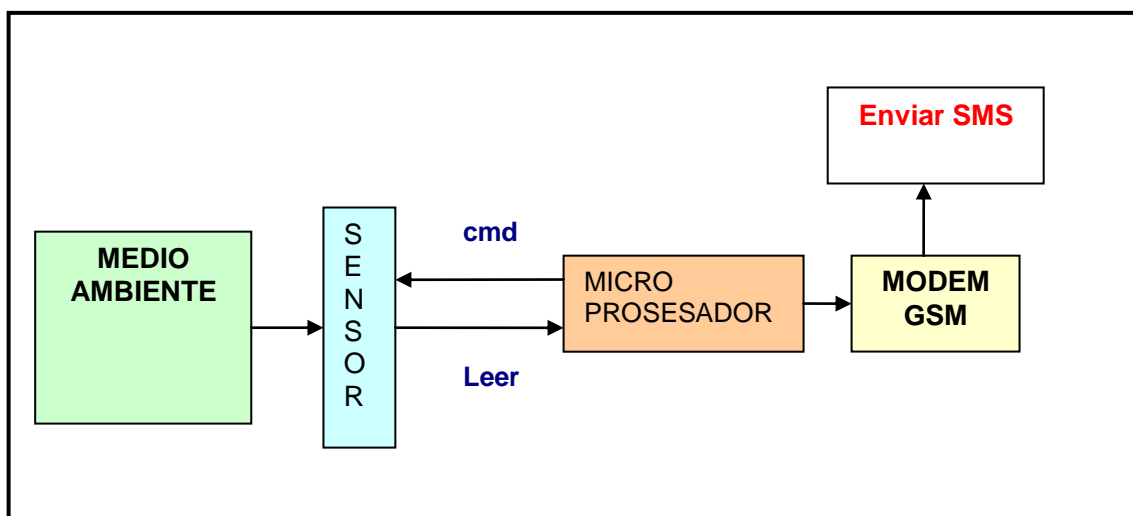


Figura 2. Microprocesador y MODEM GSM.

1.6.3 Etapa III:

Mientras los subsistemas remotos están realizando su trabajo constante de captura y envío de datos al servidor de mails, el subsistema de control de datos (aplicación de .NET), también está realizando su trabajo del lado servidor en forma permanente. El trabajo de este se puede dividir en 4 grandes aspectos: 3 "Threads" de ejecución continua en forma paralela y una interfaz de ABMC (Alta, Baja, Modificación y Consulta) para que utilice el Administrador u Operador del sistema, mediante esta interfaz el administrador, podrá realizar ABMC de las tres entidades principales del sistema: Usuarios, Cultivos y Equipos remotos, en las cuales se podrán realizar diversas tareas como por

ejemplo, asociarlos entre si, modificar prestadoras de envío celular, establecer tiempos de auto reportes para cada equipo, seleccionar medios a los cuales se le deben enviar los reportes al usuario, seleccionar que cultivos controla un equipo, designar el usuario al cual pertenece un equipo, determinar valores de temperaturas críticos de un cultivo, colocar descripciones de las entidades, configurar medios de contacto de los usuarios, etc.

La interfaz también provee un control sobre los procesos fundamentales del FRC, la puesta en marcha o detención de los hilos de chequeo permanente, estos hilos o Threads del sistema son 3:

Thread de recolección de mensajes.

Thread de chequeo de auto-reportes.

Thread de chequeo de alarmas o valores críticos.

1.6.3.1 Thread de recolección de mensajes:

Este hilo, tal como se muestra en la figura 3, es el encargado de realizar la recolección de los mensajes enviados desde el subsistema remoto al servidor de mails.

Una vez que el thread comienza a trabajar, este chequea la existencia de mensajes, para ello se utilizó Microsoft Outlook, como cliente de correo, el cual se configuró para que cada 1 minuto se conecte a un servidor remoto de mails y descargue los mensajes que existieran en este, el hilo utiliza una API para interactuar con el cliente de correo y leer los mensajes que este vaya almacenando, si encuentra mensajes, este hilo crea un objeto de la capa de acceso a datos para insertar los valores en la base de datos, luego si la inserción fue exitosa, entonces se procede a eliminar de la bandeja de entrada los mensajes capturados por el cliente de correo, aquí es donde finaliza un “ciclo” de ejecución de este thread y el mismo procede a “dormir” por un periodo de 30000 milisegundos (30 segundos), al cabo de este tiempo el hilo vuelve a despertar y realiza nuevamente el proceso anteriormente descrito, en caso de que el cliente de correo no haya recibido mensajes, el hilo procede directamente a dormir por 30 segundos.

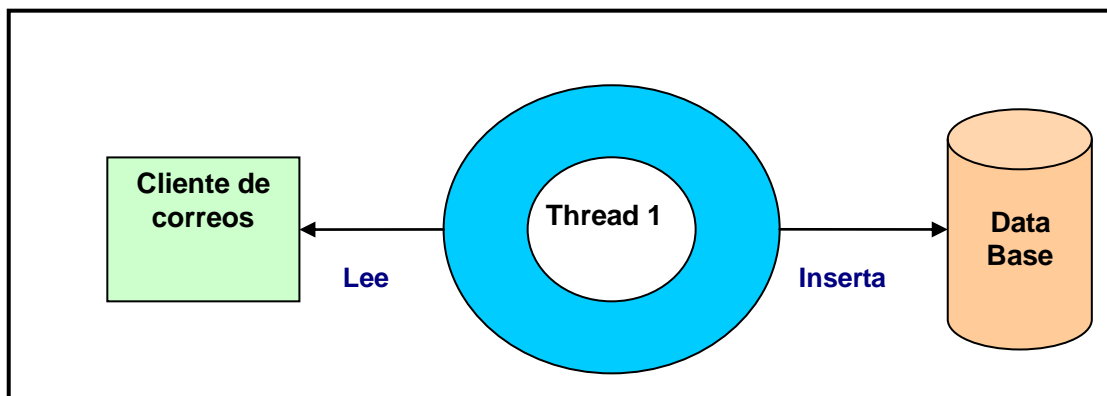


Figura 3. Hilo de lectura de mensajes.

1.6.3.2 Thread de chequeo de auto-reportes:

Este hilo, tal como se muestra en la figura 4, es el encargado de detectar que mensajes almacenados en la base de datos deben ser enviados a sus respectivos usuarios, es decir, este hilo realiza un trabajo exhaustivo de control de tiempo, en donde debe realizar un filtro de los mensajes que han ingresado en la base de datos para detectar que mensajes correspondan que sean enviados en un instante determinado.

El hilo se ejecuta cada 1 minuto, y su proceso de trabajo es el siguiente: primero realiza una consulta a la tabla de temperaturas por aquellas que posean un equipo que este asociado a un usuario, a un cultivo y que el tiempo de envío sea igual o menor al configurado para ese equipo en el campo de auto reporte, una vez obtenidos estos registros procede a enviarlos a cada usuario, a su mail o celular (según hayan sido configurados).

Luego de que los reportes hayan sido enviados a los usuarios que corresponda, el hilo procede a enviar dichos reportes a una tabla de históricos para futuros usos, una vez realizada esta tarea el ciclo de este hilo termina procediendo a pasar a estado de “sleep” (pausa) por 1 minuto, al despertar transcurrido el minuto, el hilo vuelve a realizar el proceso descrito arriba.

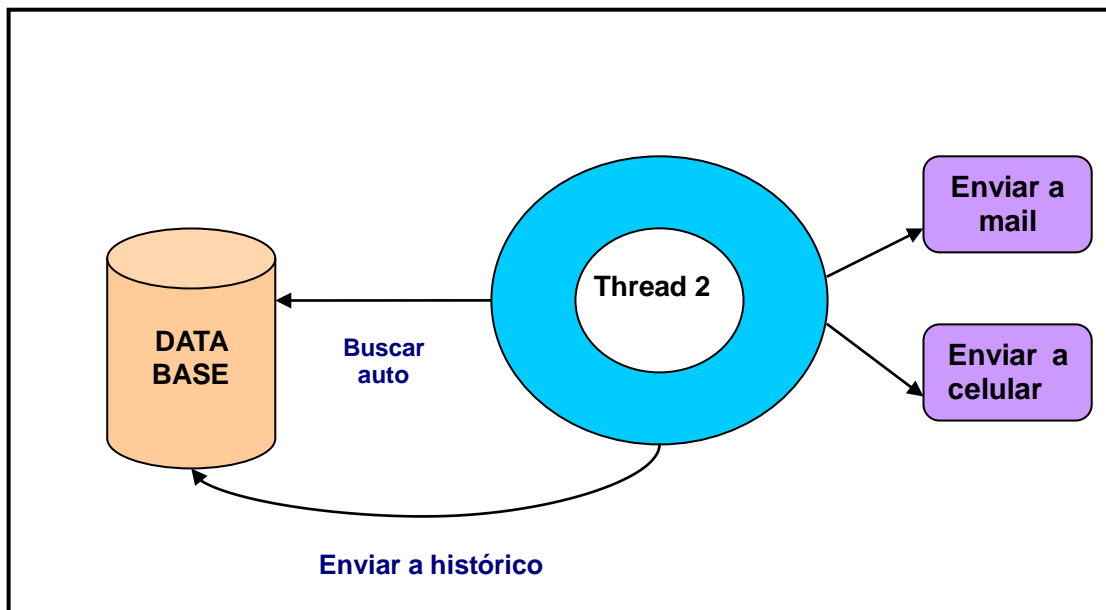


Figura 4. Hilo de chequeo y envío de auto reportes por tiempo.

1.6.3.3 Thread de chequeo de alarmas o valores críticos:

Este hilo, tal como se muestra en la figura 5, es el que realiza el chequeo constante de valores críticos, es decir, que los reportes que ingresan estén dentro del “umbral” normal de temperaturas aceptadas para el cultivo que el equipo esta protegiendo; si el reporte que ingreso esta por debajo de ese umbral, el sistema procede inmediatamente a enviar un reporte de alarma al usuario a los medios que este había programado (mail y/o celular), luego de enviar el reporte de alarma, se procede a marcar el reporte en la base de datos, como que fue enviado como alarma para su futuro procesamiento en auto reportes de tiempo.

El envío de una alarma no modifica el proceso común de envío de auto reportes, el mismo reporte será enviado tanto como alarma como así también como auto reporte en el tiempo que corresponda. Luego de que la alarma ha sido enviada el thread de chequeo de alarmas pasa a estado de “sleep” por diez segundos, transcurrido este tiempo despierta nuevamente y realiza la misma tarea. En caso de no haber alarmas en cada ciclo, el hilo directamente pasa a dormir por otros 10000 milisegundos.

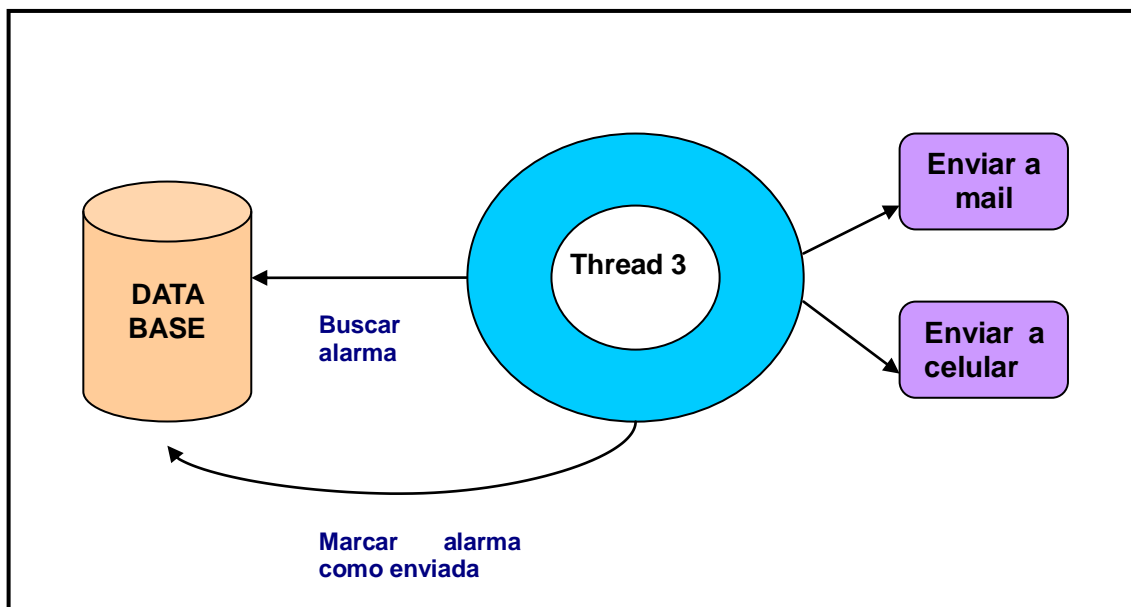


Figura 5. Hilo de busca y envío de alarmas.

3. CONCLUSION

Como conclusión podemos decir que este trabajo es un ejemplo a una solución alternativa de un problema que tiene muy pocas soluciones propuestas y las que existen son costosas e incómodas.

Las soluciones existentes no son desarrolladas en nuestro país, haciendo que esto incremente sus costos y disminuya la posibilidad de brindar soporte técnico de las mismas.

Uno de los objetivos que motivó este trabajo fue investigar y desarrollar un sistema para ponerlo al alcance de profesionales de la rama de agronomía o particulares con necesidad de controlar constantemente parámetros críticos de sus propiedades y facilitar su labor incentivando el teletrabajo como posible metodología para la realización de sus tareas.

Dicha solución es económica, práctica y escalable con el cual no solo se pueda satisfacer una necesidad de manera económica, sino también ofrecer un punto de entrada a la investigación, mejora y desarrollo de soluciones innovadoras y prácticas.

También se puede ver que muchas veces para resolver un problema no siempre hace falta poseer gran cantidad de recursos, sino que se debe utilizar la imaginación, para poder reutilizar elementos de muy bajo costo y muy buenas prestaciones.

APÉNDICE

TELEMETRIA:

Es una tecnología que permite la medición remota de magnitudes físicas y el posterior envío de la información hacia el operador del sistema.

SCRIPT:

En informática, un script, es un guión o conjunto de instrucciones. Permiten la automatización de tareas creando pequeñas utilidades.

MICROCONTROLADOR:

Es un circuito integrado o chip que incluye en su interior las tres unidades funcionales de una computadora: unidad central de procesamiento de información, memoria y unidades de entrada / salida.

SMS:

El servicio de mensajes cortos o SMS (Short Message Service) es un servicio disponible en los teléfonos móviles que permite el envío de mensajes cortos (también conocidos como mensajes de texto, o más coloquialmente, textos) entre teléfonos móviles, teléfonos fijos y otros dispositivos de mano.

MODEM:

(MODULADOR-DEMULADOR) Es un periférico de entrada y salida, que puede ser interno o externo a una computadora, y sirve para conectar una línea telefónica con la computadora. Se utiliza para acceder a Internet u otras redes, realizar llamadas, etc.

.NET:

Es un entorno de desarrollo de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones.

MySQL:

Es un sistema de gestión de bases de datos relacional, con más de seis millones de instalaciones. MySQL es software libre en un esquema de licenciamiento dual.

UML:

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; aún cuando todavía no es un estándar oficial, está apoyado en gran manera por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software.

GSM:

Global "System for Mobile communications" (Sistema Global para las Comunicaciones Móviles), anteriormente conocida como "Group Special Mobile" (GSM, Grupo Especial Móvil) es un estándar mundial para teléfonos móviles digitales.

ROAMING:

Es un concepto utilizado en comunicaciones inalámbricas que está relacionado con la capacidad de un dispositivo para moverse de una zona de cobertura a otra.

TARJETA SIM:

Una tarjeta SIM (acrónimo de Subscriber Identity Module, 'Módulo de Identificación del Suscriptor') es una tarjeta inteligente desmontable usada en teléfonos móviles que almacena de forma segura la clave de servicio del suscriptor usada para identificarse ante la red, de forma que sea posible cambiar la línea de un terminal a otro simplemente cambiando la tarjeta.

FRAMEWORK:

En el desarrollo de software, un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.

THREAD:

Es una característica que permite a una aplicación realizar varias tareas a la vez (concurrentemente). Los distintos hilos de ejecución comparten una serie de recursos tales como el espacio de memoria, los archivos abiertos, situación de autenticación, etc. Esta técnica permite simplificar el diseño de una aplicación que debe llevar a cabo distintas funciones simultáneamente.

API:

Es una interfaz de programación de aplicaciones (del inglés application programming interface) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.