

2017



**Eduardo González
Calvillo.**

Ingeniería en Tecnologías de la Información y las Comunicaciones.

Apoyo técnico al laboratorio de modelaje y sensores remotos – desarrollo de sistema de gestión de estaciones automatizadas.

Instituto nacional de investigaciones forestales, agrícolas y pecuarias (inifap).

Asesor: Ing. Juan Fernando Gómez Gutiérrez.

Octubre 2017.

RESUMEN

La Red Nacional de Estaciones Agroclimatológicas (RNEA) del INIFAP (Instituto Nacional de Investigaciones Forestales Agrícolas Y Pecuarias) opera de manera adecuada, registrando los datos directamente en el terreno cada 15 minutos de las variables meteorológicas por lo que es fundamental que la medición de la variabilidad de la temperatura, precipitación, humedad relativa, radiación solar y, dirección y velocidad de viento. Con la infraestructura actual, y a través de la aplicación de protocolos de análisis de datos, el portal <http://clima.inifap.gob.mx> pone a disposición de los usuarios en general la base de datos libre y sin restricciones de acceso.

La puesta a disposición de la base de datos es con la premisa de que los tomadores de decisiones en el sector primario dispongan de información confiable y en tiempo cercano al real de casi 1,100 estaciones distribuidas en el territorio nacional. Alcanza una cobertura total de la frontera agrícola del país, es decir, que la Red Nacional de Estaciones Agroclimatológicas (RNEA) opera de manera adecuada, lo cual ha sido posible con el apoyo de las Fundaciones Produce de cada entidad federativa, lo que ha permitido que se encuentre en funcionamiento y dentro de los estándares de calidad de la información.

El presente Sistema de Gestión de Estaciones Automatizadas tiene como objetivo detectar atrasos en el envío de datos de las estaciones. Y verificar las razones de los mismos. La actualización de la información de la Red Nacional de Estaciones Agroclimatológicas Automatizadas, se realiza a través en primera instancia a las bases locales de captura de información en cada uno de los estados mediante el uso de la frecuencia de uso libre UHF y algunas cuentan con GPRS, una vez que ya se encuentra registrada en la base de datos local, esta es transmitida y almacenada al Laboratorio Nacional de Modelaje y Sensores Remotos a través de Internet. La recuperación de datos se hace mediante sistemas alineados al motor de base de datos SQL Server 2008 R2, la aplicación se desarrolló en el lenguaje de programación c# con el framework de asp.net y para Dispositivos móviles iOS y Android. Se utilizó el IDE Visual Studio 2012 Professional generar los procesos para estas aplicaciones.

Con este conjunto de aplicaciones toman las decisiones al momento, y se generan planes de mantenimiento para la corrección y/o prevención de fallas.

ABSTRACT

The present project aims to implement a system to detect connections agrometeorological stations and disconnection. These stations use communication protocols, data retrieval is done by the database engine SQL Server 2008 R2, the application will be developed in the programming language C # with asp.net framework and Visual Studio 2012 IDE will be used for programming. It is the main reason for this, which spontaneously stations are disconnected from the server, and so no data or information needed for the main application that is responsible for displaying meteorological data users are received. With this application the information in time and will disconnect any station is detected and given prompt maintenance, avoiding loss of information.

Tabla de Contenido

RESUMEN	3
ABSTRACT	4
Lista de figuras	6
1. INTRODUCCIÓN.....	7
2. JUSTIFICACIÓN.....	9
3. PLANTEAMIENTO DEL PROBLEMA.....	10
4. OBJETIVO	13
4.1 Objetivos específicos	13
4.2 Alcances del proyecto	13
5. ANTECEDENTES.....	14
6. MARCO TEÓRICO	15
6.3 Estaciones meteorológicas automatizadas	18
7 METODOLOGÍA	20
8 Desarrollo de la propuesta	27
9 Comprobación del funcionamiento correcto del Sistema de Gestión de Estaciones	28
9.1 Desarrollo del programa de conexión y recepción de datos por medio de Arduino (Trabajo realizado por el inifap).	33
9.2 Configuración de los módulos X-Bee® serie 1.....	37
9.3 Desarrollo del software en LabVIEW®.....	37
8 RESULTADOS	40
9 CONCLUSIONES	43
10 CONCLUSIONS	43
11 CRONOGRAMA DE ACTIVIDADES	44
12 REFERENCIAS.....	45

Lista de figuras

Figura 1. Distribución espacial de la Red Nacional de Estaciones Agrometeorológicas administradas por el INIFAP.. ¡Error! Marcador no definido.8
Figura 2. Muestra el proceso de lógica del negocio en su nivel más detallado.....	¡Error! Marcador no definido.22
Figura 3. Muestra el proceso de la capa de datos.....	¡Error! Marcador no definido.23
Figura 4. Muestra el modelado de procesos en su nivel general.	¡Error! Marcador no definido.25
Figura 5. Muestra parte del código requerido para la generación de las vistas para móvil.....	¡Error! Marcador no definido.27
Figura 6. Programación de Modelos.....	28
Figura 7. Vista para web.....	28
Figura 8. Muestra parte del código requerido para la generación de las vistas para web ¡Error! Marcador no definido.9
Figura 9. Vista Web.....	¡Error! Marcador no definido.9
Figura 10. Vista Web.....	¡Error! Marcador no definido.0
Figura 11. Detalle gráfico de los procesos	¡Error! Marcador no definido.
Figura 12. Estructura void setup() de Arduino.....	¡Error! Marcador no definido.38
Figura 13. Lectura de valores de los sensores y envió por módulo X-Bee®	¡Error! Marcador no definido.39
Figura 14. Guardado de datos en memoria micro SD.	¡Error! Marcador no definido.40
Figura 15. Función del sensor de precipitación.....	3441
Figura 16. Comprobación de módulos X-Bee® serie 1	3542
Figura 17. Configuración serial en LabVIEW®.....	3642

Figura 18. Ciclo while secundario.....	3643
Figura 19. Diagrama de bloques de graficas de visualización.....	3743
Figura 20 Diagrama de bloques de archivo en Excel.	3843

1. INTRODUCCIÓN

La agricultura es altamente dependiente del clima, siendo la radiación solar, temperatura y la precipitación los factores que limitan o favorecen el crecimiento de los cultivos. En México como en muchos otros países, el desarrollo económico derivado de las actividades agropecuarias es frecuentemente afectado por la presencia de fenómenos climáticos adversos.

La disponibilidad de información de las condiciones meteorológicas prevalecientes en una región dada, permite elaborar programas y definir acciones específicas que coadyuven a la mitigación de los efectos de fenómenos climáticos adversos como sequía, inundaciones, ráfagas de viento, heladas, granizo, entre otras; por lo que es fundamental que la medición de la variabilidad de la temperatura, precipitación, humedad relativa, radiación solar, dirección y velocidad de viento, alcance una cobertura total de la frontera agrícola del país, es decir, que la Red Nacional de Estaciones Agroclimatológicas (RNEA) opere de manera adecuada, lo cual ha sido posible con el apoyo de las Fundaciones Produce de cada entidad federativa, lo que ha permitido que se encuentre en funcionamiento y dentro de los estándares de calidad de la información.

Por otra parte, el conocimiento de las condiciones del clima que se presentan en las áreas donde se practican la actividades primarias es parte fundamental para la toma de decisiones en la agricultura moderna que requiere información cuasi en tiempo real para incidir en los procesos de producción, ya que el clima es uno de los componentes ambientales en la adaptación, distribución y productividad de los seres vivos.

En el presente proyecto se realizara la construcción de un sistema de comunicación entre las estaciones agrometeorológicas del INIFAP, el cual monitoree la entrada de datos tanto como la conexión de dichas estaciones, existen y se cuenta con un poco más de 1100 estaciones a nivel nacional. El sistema Integral móvil y web que se desarrolló, monitorea la recepción de datos de dichas estaciones. El motivo de esto es facilitar la detección de atrasos en él envío de datos. Y realizar los procesos apropiados para la planeación de mantenimientos y/o corrección, y puesta en marcha de la estación.

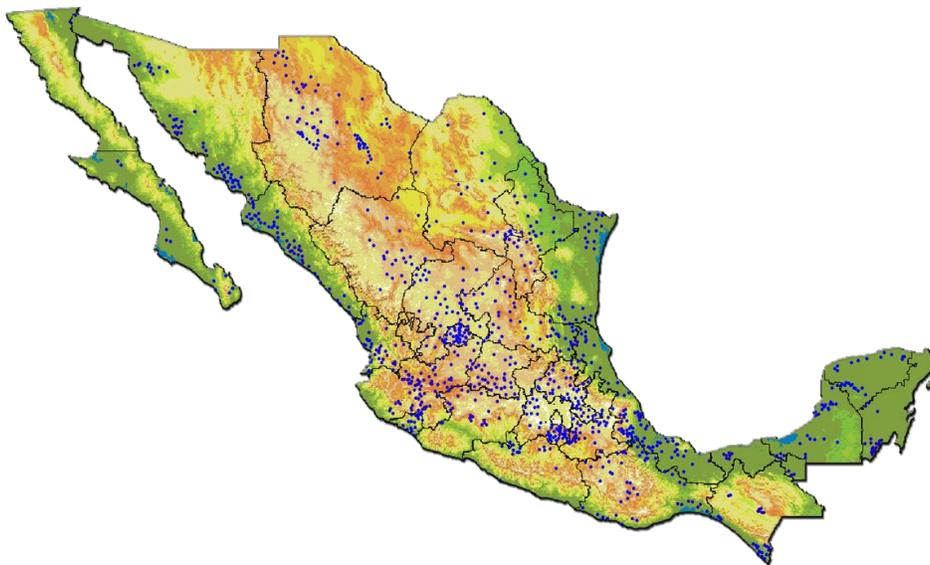


Figura 1. Distribución espacial de la Red Nacional de Estaciones Agrometeorológicas administradas por el INIFAP.

2. JUSTIFICACIÓN

La Red Nacional de Estaciones Agroclimatológicas están conformadas por una serie de sensores para la medición e instrumentos de fuentes de energía (panel solar y batería recargable), estos fueron diseñados, calibrados y con una vida útil para funcionar dentro de un rango de tolerancia; por lo que no monitorear y detectar posibles fallas, pone en una situación crítica el funcionamiento de los sensores y los componentes auxiliares de una estación meteorológica.

Por otra parte, la falta de Aplicaciones web y móviles, para generar planes de mantenimiento preventivo y correctivo a la red, es una de las causas principales de la falta de seguimiento de las estaciones. Además la actualización de la base de datos climatológicos de la red nacional de estaciones está sujeta a los riesgos de no tener una debida planeación, para la puesta en marcha de estaciones caídas.

De acuerdo a lo anterior, existe la posibilidad de perder la comunicación con las redes estatales, si no se genera un Sistema de Gestión de estaciones automatizadas, para reducir el tiempo en la detección de fallas en el registro de los datos de las estaciones.

3. PLANTEAMIENTO DEL PROBLEMA

El INIFAP en los últimos años no contaba con un sistema que detectara estaciones caídas, desconectadas o desaparecidas, esto dificultaba la búsqueda puesto que se tenía que hacer manualmente, realizando la búsqueda dentro de la base de datos directamente, esto demoraba tiempo ya que era generar la consulta a la base y buscar. Generaba conflicto ya que para realizar la consulta tenía que hacerlo personal del área de sistemas, esto resultaba una pérdida de tiempo para el avance de proyectos, así como atraso de los mismos a la fecha de entrega. Al generar el sistema lo podrán consultar colaboradores del laboratorio, independientemente de si son del área de sistemas o no, esto es de gran ayuda ya que ahorrara tiempo de los del área de sistemas.

De acuerdo con las estadísticas de la Organización para la Alimentación y la Agricultura (FAO, por sus siglas en inglés), hasta el año 2009 se cultivaron a nivel mundial aproximadamente 1,527 millones de hectáreas de las cuales, el 19.7 % se destinó para agricultura de riego y el 80.3 % restante para agricultura de temporal o secano (FAO, 2011). En México la superficie agrícola sigue una tendencia similar puesto que se cultivan alrededor de 22 millones de hectáreas, de las cuales la agricultura de riego comprende el 29.5 % y la de temporal el 70.5 % (CONAGUA, 2012; SIAP,

2013). La superficie establecida bajo riego a nivel nacional oscila en 6.5 millones de hectáreas, distribuidas en 85 distritos de riego y 39,500 unidades de riego (CONAGUA, 2011).

En las zonas áridas y semiáridas de México, se produce la mayor cantidad de alimentos y se obtienen los mejores rendimientos, donde se encuentra la mayor superficie bajo riego y donde la intensificación de la variabilidad climática ha generado incertidumbre en los volúmenes de agua disponible, ocasionando inestabilidad en la productividad de varios cultivos (Flores et al., 2013). Gran parte de la aplicación del riego se realiza de acuerdo al criterio que el agricultor ha adquirido como experiencia y

al manejo que ofrece cada parcela, ya que la programación científica del riego en la mayoría de los casos se realiza aplicando los procedimientos propuestos por la FAO debido a que existe nula o escaso material bibliográfico generado localmente (Sifuentes y Ojeda, 2005). El riego es indispensable para asegurar rendimientos comerciales aún en zonas con precipitación alta (Wanjura y Upchurch, 2000). Debido a que la competencia por la disponibilidad de agua es cada vez más fuerte, existe la necesidad de generar un reconocimiento económico, social y ambiental al buen uso del agua (Flores et al., 2014). Por tal motivo, para que una calendarización y la supervisión del servicio de riego en grandes zonas sean efectivas, es importante el ajuste de la frecuencia y cantidad del riego con base en factores del suelo, ambiente, planta y manejo del riego, lo cual demanda de modelos de fácil implantación en los sistemas computacionales (Ojeda et al., 2006).

El desarrollo y popularización de las computadoras proporciona herramientas para almacenar grandes volúmenes de datos y realizar cálculos numerosos. Lo anterior permite el desarrollo de sistemas computacionales para analizar el comportamiento y respuesta biológica de los sistemas de producción agrícola bajo escenarios de diferente manejo así como en condiciones climáticas contrastantes. Por lo cual, los modelos calibrados y validados son un herramienta computacional viable como apoyo para la toma de decisiones en la aplicación de insumos y labores agrícolas que se realizan en la parcela.

La seguridad alimentaria es un reto en México para producir suficiente cantidad de alimentos, con calidad alta y precio accesible. La agricultura de riego es importante para responder al aumento de la demanda de alimentos de la población que día con día aumenta; aunque la superficie de cultivo ha permanecido casi estable en las últimas cuatro décadas (Ojeda et al., 2012). El maíz es cultivado en la mayoría de las zonas con la modalidad de producción de riego y temporal, pero México no es autosuficiente e importa alrededor de 8 millones de toneladas anualmente, esto es la tercera parte de sus necesidades (SAGARPA, 2007). Algunos factores externos e internos que complicaron el abastecimiento de maíz y la reducción de su producción en el 2011, fue

el aumento de la demanda para producir etanol en la última década, aumento de los costos de producción, variabilidad climática y recurrencia de sequías, que aumentan la presión en los recursos hídricos para uso agrícola (SIAP, 2012).

En otro contexto, el cultivo de papa (*Solanum tuberosum* L.) ocupa el cuarto lugar en superficie sembrada en México, superado únicamente por los granos básicos: maíz (*Zea mays* L.), frijol (*Phaseolus vulgaris* L.) y trigo (*Triticum aestivum*); estableciéndose una superficie aproximada a las

70,000 hectáreas (SAGARPA, 2006). En el norte de Sinaloa se establecen alrededor de 14,000 hectáreas y se obtiene un rendimiento promedio de 25 ton/ha (CONPAPA, 2003). A pesar de lo anterior, el manejo tradicional del cultivo de papa es cada vez menos funcional debido a la alta variabilidad climática en las zonas productoras del noroeste de México, provocando aplicaciones excesivas de insumos, contaminación y baja rentabilidad (Sifuentes et al., 2013).

Por lo anterior, se requieren de herramientas que permitan analizar estaciones automatizadas. A través de un portal web y dispositivos móviles, Para evitar el deterioro de las mismas.

4. OBJETIVO

Que el Sistema de Gestión de Estaciones provea al LNMySR, del soporte técnico para la elaboración y/o identificación de fallas en el envío de datos de las estaciones automatizadas que pertenecen a dicha red.

4.1 Objetivos específicos

- Visualizar en el momento estaciones que presenten algún Retraso en los datos.
- Implementar el Sistema de Gestión de estaciones en plataforma web y móvil.

4.2 Alcances del proyecto

El alcance del Proyecto es a nivel Nacional. Siempre y cuando exista conectividad a Internet para el funcionamiento del Sistema de Gestión.

5. ANTECEDENTES

Ante la recurrencia de eventos climáticos adversos que afectan las actividades agroalimentarias del país, la SAGARPA con el apoyo del INIFAP y de las Fundaciones Produce de los Estados, implementaron la Red Nacional de Estaciones Estatales Agroclimatológicas. Esta red tiene como objetivo principal disponer de infraestructura para el registro automatizado del tiempo meteorológico, con la finalidad de ofrecer a los productores agropecuarios información meteorológica a tiempo real, aplicada a los procesos de producción y protección ante clima adverso. Desde que inició el proyecto en el año 2000, con el apoyo de INIFAP y las Fundaciones Produce de los Estados se dieron a la tarea de implementar la Red Nacional de Estaciones Agrometeorológicas, con la finalidad de ofrecer a lo sector agropecuario y a la población en general el registro de información agroclimatológica de una forma automatizada.

Actualmente la Red Nacional de Estaciones Agrometeorológicas cuenta con un poco más de 1100 estaciones operando en 29 estados de la República Mexicana y además se cuenta con 21 estaciones más localizadas en los países de Belice, Guatemala y Honduras.

Objetivos de la Red Nacional de Estaciones Agrometeorológicas:

- Disponer de infraestructura para el registro automatizado del clima, con la finalidad de ofrecer a los productores agropecuarios información meteorológica en tiempo real, aplicada a los procesos de producción y protección ante clima adverso.
- Apoyar las decisiones relacionadas con la Seguridad Alimentaria Nacional por parte de las instituciones del Sector Agropecuario en la estimación de las cosechas esperadas ante dificultades climáticas.
- Informar a la población civil de las condiciones climatológicas adversas a fin de realizar medidas de prevención ante el riesgo de la pérdida de vidas humanas.

6. MARCO TEÓRICO

6.1 Internet

El internet es un conjunto descentralizado de redes de comunicación interconectadas que utilizan la familia de protocolos TCP/IP, lo cual garantiza que las redes físicas heterogéneas que la componen como una red lógica única de alcance mundial. Sus orígenes se remontan a 1969, cuando se estableció la primera conexión de computadoras, conocida como Arpanet, entre tres universidades en California (Estados Unidos).

Uno de los servicios que más éxito ha tenido en internet ha sido la World Wide Web (WWW o la Web), hasta tal punto que es habitual la confusión entre ambos términos. La WWW es un conjunto de protocolos que permite, de forma sencilla, la consulta remota de archivos de hipertexto. Esta fue un desarrollo posterior (1990) y utiliza internet como medio de transmisión.

Existen, por tanto, muchos otros servicios y protocolos en internet, aparte de la Web: el envío de correo electrónico (SMTP), la transmisión de archivos (FTP y P2P), las conversaciones en línea (IRC), la mensajería instantánea y presencia, la transmisión de contenido y comunicación multimedia, telefonía (VoIP), televisión (IPTV), los boletines electrónicos (NNTP), el acceso remoto a otros dispositivos (SSH y Telnet) o los juegos en línea.

Sirve de enlace entre redes y permite ampliar su cobertura al hacerlas parte de una "red global". Esta red global tiene la característica de que utiliza un lenguaje común que garantiza la intercomunicación de los diferentes participantes; este lenguaje común o protocolo (un protocolo es el lenguaje que utilizan las computadoras al compartir recursos) se conoce como TCP/IP.

Los Proveedores de Servicios de Internet (ISP) conectan a clientes, quienes representan la parte más baja en la jerarquía de enrutamiento, con otros clientes de otros ISP a través de capas de red más altas o del mismo nivel. En lo alto de la jerarquía de enrutamiento están las redes de capa 1, grandes compañías de

telecomunicaciones que intercambian tráfico directamente con otras a través de acuerdos de interconexión. Redes de capa 2 y de más bajo nivel compran tráfico de Internet de otros proveedores para alcanzar al menos algunas partes del Internet mundial, aunque también pueden participar en la interconexión. Un ISP puede usar un único proveedor para la conectividad o implementar multihoming para conseguir redundancia y balanceo de carga. Los puntos neutros tienen las cargas más importantes de tráfico y tienen conexiones físicas a múltiples ISP.

Los ordenadores y routers utilizan las tablas de enrutamiento para dirigir los paquetes IP entre las máquinas conectadas localmente. Las tablas pueden ser construidas de forma manual o automáticamente a través de DHCP para un equipo individual o un protocolo de enrutamiento para los routers de sí mismos. En un solo homed situaciones, una ruta por defecto por lo general apunta hacia "arriba" hacia un ISP proporciona el transporte. De más alto nivel de los ISP utilizan el Border Gateway Protocol para solucionar rutas de acceso a un determinado rango de direcciones IP a través de las complejas conexiones de la Internet global.

Las instituciones académicas, las grandes empresas, gobiernos y otras organizaciones pueden realizar el mismo papel que los ISP, con la participación en el intercambio de tráfico y tránsito de la compra en nombre de sus redes internas de las computadoras individuales. Las redes de investigación tienden a interconectarse en subredes grandes como GEANT, GLORIAD, Internet2, y de investigación nacional del Reino Unido y la red de la educación, Janet. Estos a su vez se construyen alrededor de las redes más pequeñas (véase la lista de organizaciones académicas de redes informáticas).

No todas las redes de ordenadores están conectados a Internet. Por ejemplo, algunos clasificados los sitios web de los Estados sólo son accesibles desde redes seguras independientes.

6.2 Dispositivos Móviles

Un dispositivo móvil es un aparato de pequeño tamaño, con algunas capacidades de procesamiento, alimentación autónoma, con conexión permanente o intermitente a una red, con memoria limitada, diseñados específicamente para una función, pero que pueden llevar a cabo otras funciones más generales. Dado el variado número de niveles de funcionalidad asociado con dispositivos móviles, en el 2005, T38 y DuPont Global Mobility Innovation Team propusieron varios estándares para la definición de dispositivos móviles (Martínez, 2011).

- **Dispositivo móvil de datos limitados (Limited Data Mobile Device):** dispositivos que tienen una pantalla pequeña, principalmente basada en pantalla de tipo texto con servicios de datos generalmente limitados a SMS y acceso WAP. Un típico ejemplo de este tipo de dispositivos son los teléfonos móviles.
- **Dispositivo móvil de datos básicos (Basic Data Mobile Device):** dispositivos que tienen una pantalla de mediano tamaño, (entre 120 x 120 y 240 x 240 píxeles), menú o navegación basada en íconos por medio de una "rueda" o cursor, y que ofrecen acceso a e-mail, lista de direcciones, SMS, y un navegador web básico. Un típico ejemplo de este tipo de dispositivos son las BlackBerrys y los Teléfonos Inteligentes.
- **Dispositivo móvil de datos mejorados (Enhanced Data Mobile Device):** dispositivos que tienen pantallas de medianas a grandes (por encima de los 240 x 120 píxeles), navegación de tipo stylus, y que ofrecen las mismas características que el "Dispositivo Móvil de Datos Básicos" (Basic Data Mobile Devices) más aplicaciones nativas y aplicaciones corporativas usuales, en versión móvil. Este tipo de dispositivos incluyen el sistema operativo como Windows Mobile, Android, iPhone OS (iOS).

Un sistema operativo móvil o SO móvil es un sistema operativo que controla un dispositivo móvil al igual que las computadoras utilizan Windows o Linux entre otros. Sin embargo, los sistemas operativos móviles son bastantes más simples y están más orientados a la conectividad inalámbrica, los formatos multimedia para móviles y las diferentes maneras de introducir información en ellos.

Android está basado en una versión modificada del Kernel de Linux. Al contrario que otros sistemas operativos para dispositivos móviles como iOS o Windows Phone, Android se desarrolla de forma abierta y se puede acceder tanto al código fuente como al listado de incidencias donde se pueden visualizar problemas aún no resueltos y reportar problemas nuevos (Martínez, 2011).

6.3 Estaciones meteorológicas automatizadas

Las estaciones meteorológicas automatizadas (EMA's) son instalaciones destinadas a llevar el registro de variables climatológicas, con los cuales se puede realizar predicciones, fines comunicativos a la sociedad en general, apoyo en la toma de decisiones. La EMA's es colocada para lograr un ahorro en el trabajo y en tiempo, y sobre todo para realizar la lectura en lugares que son de difícil acceso o que se encuentran muy alejados de la instalación en donde se operan dichas estaciones.

Las principales variables que se miden en una EMA's son:

- Temperatura.
- Radiación solar.
- Presión atmosférica.
- Humedad relativa.
- Velocidad y dirección de viento.

Los sensores para llevar a cabo la medición de las variables deben de ser resistentes a condiciones extremas ya que por lo regular las estaciones se colocan a la intemperie y están expuestas a las condiciones climáticas que existan en el lugar en donde son colocadas. Las EMA's se clasifican de acuerdo a la aplicación que se le vaya a dar, algunas de las clasificaciones son las siguientes:

- Agrometeorología: se utilizan para contribuir en la predicción de cosechas, con información de lluvias, la temperatura en días próximos, etc.
- Meteorología aeronáutica: se dedica a estudiar los fenómenos que afectan las aeronaves.
- Hidrometeorología: se relaciona con el estudio de la hidrología, que es la que estudia la distribución del agua en la atmosfera y en la corteza terrestre.

Las lecturas de dichos sensores se dan en un tiempo que se define por el operador de la estación, y así también se decide cada cuanto lapso se muestra a los usuarios.

Las EMA's se pueden implementar formando redes de varias estaciones, y entre más estaciones se encuentren en la red la información que se genere en ellas será mucho más precisa y confiable al abarcar un gran terreno de lecturas climatológicas.

7 METODOLOGÍA

Para la creación de este Sistema de Gestión de Estaciones, el proceso se plasma en la integración de las siguientes fases del ciclo de vida del software: planeación y análisis, diseño, desarrollo, pruebas e implementación de la aplicación para dispositivos móviles que sea de calidad, cumpliendo plazos y costos.

7.1 Planeación y análisis

En esta fase es importante considerar las diferentes velocidades de Internet, además la limitación de ciertas capacidades de procesamiento de la información; por lo cual es necesario considerar que la aplicación móvil y web, sea capaz de procesar en la computadora y/o dispositivo los datos a visualizar, un inadecuado procesamiento de estos podría afectar en el óptimo desempeño del Sistema. Otra situación importante es una eficiente interacción entre la base de datos y la computadora y/o móvil, que brinda a través de Internet los datos. Por lo tanto, en esta fase se consideró lo siguiente:

- Estudio de las limitaciones y alcances de la tecnología web y móvil.
Actualmente, existe una demanda cada vez mayor por parte de los usuarios de Internet a una disponibilidad incondicional de conexión. En este ámbito existen proveedores que ofrecen planes de banda ancha móvil y conexión fija. Pero la realidad en el mercado Mexicano es que aunque las empresas estén invirtiendo en una infraestructura física y lógica para tener la mayor cobertura posible. No todo el territorio Nacional tiene cobertura y una velocidad adecuada de Internet (COFETEL, 2013). Sin embargo estas limitaciones llevaron a la búsqueda y desarrollo de procesos para hacer eficiente el Sistema de Gestión, la visualización de los datos. Esto apoyado con pruebas de conexión para ver el comportamiento de la aplicación y el tiempo de respuesta.
- Estudio de la metodología Business Process Management.

Aunque propiamente esta metodología está enfocada al modelado de procesos (Harmon P. A. 2007), la combinación de esta con el diseño y desarrollo de la aplicación permitió dar un panorama alineado a los objetivos del Laboratorio Nacional de Modelaje. Para llegar a un Sistema de Gestión necesitamos que cumpla con los requisitos que busca esta organización. A continuación se muestran algunos de los procesos modelados del Sistema:

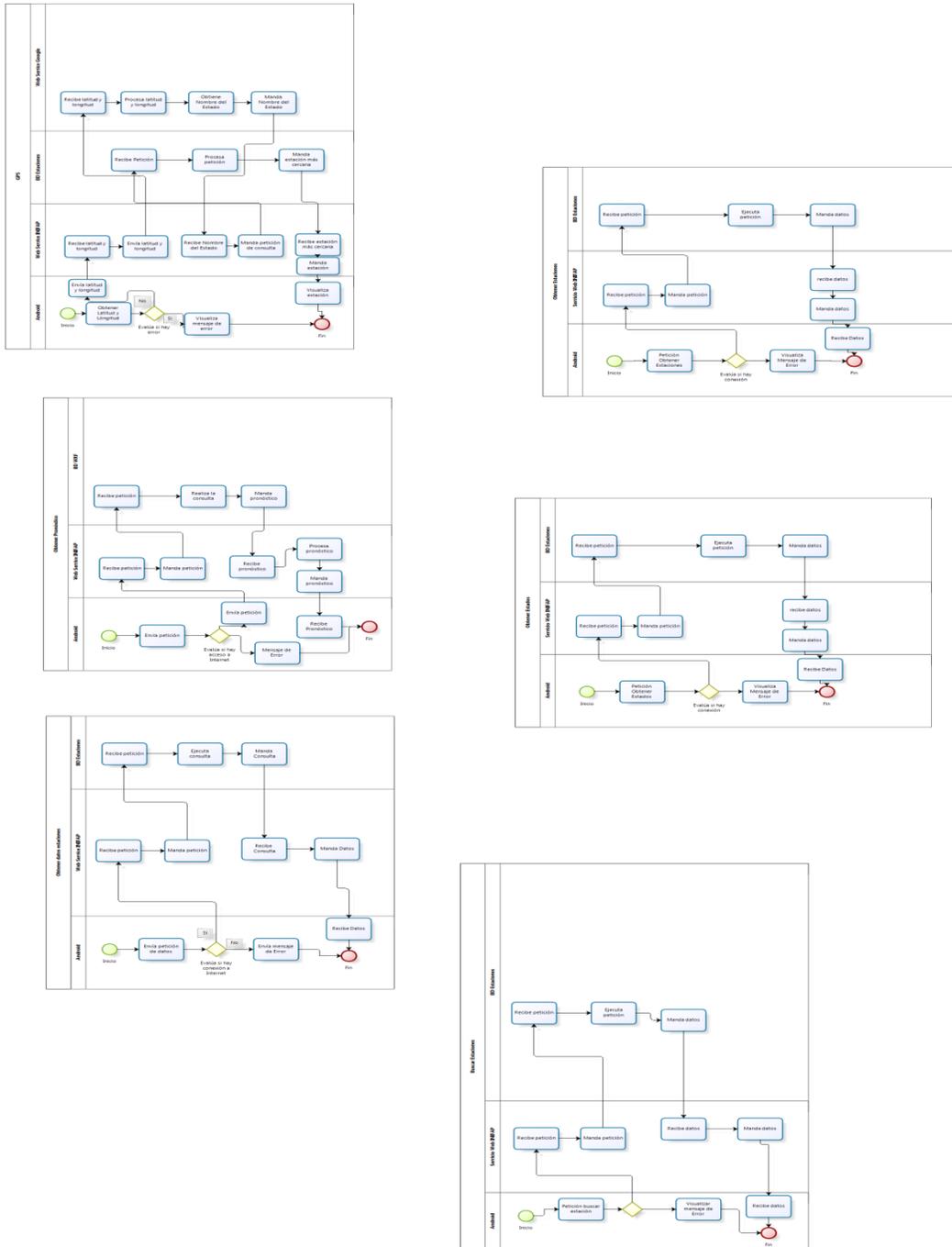


Figura 2. Muestra el proceso de lógica del negocio en su nivel más detallado.

En la siguiente imagen se muestra la base de datos en la cual se realizó el análisis para poder llegar, al desarrollo de la capa de datos que apoya, a la toma de decisiones para implementar al personal técnico del LNMySR.

- Velocidades de Internet: se realizaron pruebas de comunicación en diferentes zonas agrícolas del Estado de Aguascalientes, lo cual permitió que el protocolo de comunicación en Internet era la mejor opción debido a que existen zonas que la conexión es muy deficiente o inexistente.
- Software libre y Análisis de factibilidad del SO Android : es necesario el análisis de factibilidad de los entornos de diseño así como el desarrollo de los procesos y codificación. En la actualidad, Android es una plataforma tecnológica para los dispositivos móviles que incorporan la tecnología Java que cuya característica fundamental es aprovechar al máximo los escasos recursos con los que cuentan los dispositivos portátiles. Android es un software libre (Meier, 2010) que proporciona mecanismos integrales para garantizar un entorno de diseño y desarrollo eficiente. Que es Adaptable para el uso de cualquier Base de datos utilizada en el Laboratorio Nacional de Modelaje y Sensores Remotos del INIFAP.
- Infraestructura de hardware y software con la que cuenta el Laboratorio Nacional de Modelaje y Sensores Remotos del INIFAP.

El Laboratorio Nacional de Modelaje y Sensores Remotos cuenta con la siguiente:

Infraestructura de Hardware:

- 5 Dell Precision T7610
- 1 Unidad de almacenamiento masivo marca Dell modelo PowerVault md3600i
- 1 UPS marca APC modelo surt8000rmlxlt6u de 8Kva

- 1 Fortigate modelo 100D
- 2 Switch Cisco modelo ws-c2960s-24ps-l
- 1 Chasis IBM Blade modelo E con 14 navajas blades.
- 4 Servidores IBM System x3755 M3

Infraestructura de Software:

- Visual Estudio 2013
- SQL Server 2012
- A2A
- Software ARCGIS
- Ubuntu 14.04 LTS
- WRF
- Caja Negra (Software desarrollado en el Laboratorio Nacional de Modelaje para administración de las estaciones automatizadas)

En base a la infraestructura con la que cuenta el LNMSR, se determinó que el SO Android se adapta mejor a estas características, para desarrollar una aplicación móvil que cumpla con los requerimientos mínimos de comunicación y velocidad de internet para la visualización de datos climáticos de las estaciones automatizadas.

- Requerimientos generales de los usuarios.

Por los puntos expuestos anteriormente, es necesario considerar que el Sistema de gestión de estaciones sea capaz de procesar en los diferentes dispositivos los datos a visualizar.

7.2 Diseño

En esta segunda fase se identificó un proceso general llamado diseño, que fue donde se desglosaron los siguientes 3 subprocesos del proceso antes mencionado (Figura 3) donde se partió para modelar los subprocesos en su forma más detallada, que permitió describir todos los aspectos de la aplicación a desarrollar, apoyada en una metodología de Gestión de Procesos de Negocio (Business Process Management o BPM, por sus siglas en inglés) cuya efectividad ha sido comprobada en la práctica internacional. El objetivo es mejorar la eficiencia a través de la sistematización de los procesos de

negocio que se deben modelar, automatizar y optimizar de forma continua (Harmon, 2007). A lo largo del diseño se evalúa la calidad de los procesos de la aplicación móvil con un conjunto de revisiones técnicas, para garantizar que todos los procesos sean eficientes y den una completa idea de lo que es la aplicación web y móvil.

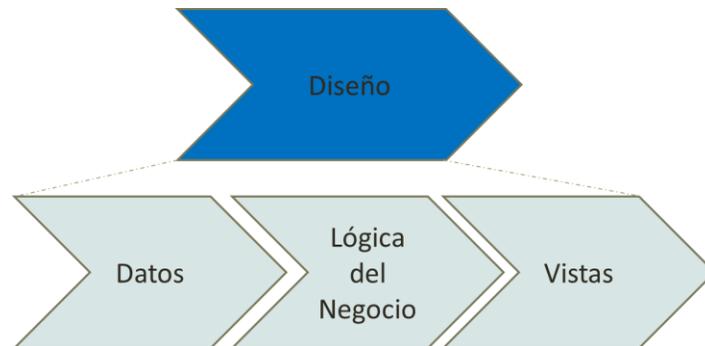


Figura 4. Muestra el modelado de procesos en su nivel general.

7.3 Desarrollo

Esta fase tiene como propósito la producción eficaz y eficiente de un producto, utilizando los procesos modelados en la fase de diseño. Para llevar a cabo la codificación de la misma (Figura 5), se elige el lenguaje Java porque es multiplataforma y el conjunto de herramientas SDK (Software Development Kit) que comprende un depurador de código, biblioteca, un simulador de teléfono basado en QEMU (Murphy, 2009). Las plataformas de desarrollo soportadas incluyen Linux (cualquier distribución moderna), Mac OS X 10.4.9 o posterior, y Windows XP o posterior. La plataforma integral de desarrollo (IDE, Integrated Development Environment) soportada oficialmente es Android Studio, aunque también puede utilizarse un editor de texto para escribir ficheros Java y XML y utilizar comandos en un terminal (se necesitan los paquetes JDK, Java Development Kit y Apache Ant) para crear y depurar aplicaciones. Además, pueden controlarse dispositivos Android que estén conectados (e.g. reiniciarlos, instalar aplicaciones en remoto). Las herramientas antes mencionadas

facilitan trasladar los procesos modelados de manera eficiente y confiable. Así como para dispositivos iOS, se utilizó Xcode 7.0.

Por otra parte la interacción entre la base de datos y el SO Android. Se realizó también un mediante Visual Studio 2012; que brinda a través de Internet los datos a la aplicación.

El uso de las metodologías orientadas a objetos junto con los métodos de descripción formal se ha revelado como una forma prometedora de tratar la complejidad de los sistemas en dispositivos móviles, puesto que son altamente complejos. Actualmente estas metodologías son soportadas por un conjunto de herramientas que permiten la especificación, simulación y validación de los aspectos funcionales de estos sistemas. No obstante, la mayoría de estas metodologías no tienen en cuenta los aspectos no funcionales tales como la interacción con los dispositivos físicos y las restricciones de tiempo real que son especialmente importantes en el contexto de este tipo de sistemas. En esta propuesta se presenta una metodología que se basa en la combinación de otras notaciones y metodologías (BPM, UML, POO, etc.) junto con la integración de técnicas de análisis de planificación en el contexto de técnicas de descripción formal. Por lo tanto, la metodología presta especial atención a la transición del modelo de objetos al modelo de tareas, teniendo en cuenta aspectos de tiempo real y de integración con los dispositivos físicos.

En la figura siguiente se muestran los modelos realizados en Visual Studio 2013.

7.4 Pruebas e implementación

Se realizaron pruebas de ejecución por separado y en su conjunto, de cada uno de los módulos para asegurar una eficiente respuesta de la aplicación. Para demostrar en diferentes escenarios que la aplicación no de respuestas inesperadas por motivos de diferentes versiones de SO Android e iOS, y así mismo se administren los activos de la App móvil de forma eficaz y puntual para asegurar el buen funcionamiento de esta. Si existiese algún problema es importante identificarlo, evaluarlo y corregirlo para posteriormente implementar las correcciones en la App. El área de impacto de la aplicación en Android abarca toda la República Mexicana, pues en todas las entidades federativas existen estaciones agroclimatológicas y se puede decir, que es donde está establecida la Red Nacional de Estaciones Agroclimatológicas Automatizadas del INIFAP.

8 Desarrollo de la propuesta

Se desarrolló el programa en el IDE de Visual Studio 2013 proporcionado por el INIFAP, para la programación de la página web, que será la encargada de mostrar los datos a los usuarios dentro del Laboratorio, después se procederá a la programación de los módulos móvil y web, mediante el software Android Studio y X Code 7.0.

9 Comprobación del funcionamiento correcto del Sistema de Gestión de Estaciones

La presente descripción muestra el lenguaje que se utilizó en la representación de los procesos de Datos, Lógica del Negocio y Vistas, con sus respectivos actores involucrados.

Vistas:

1.- Selección Estación

- Sistema de Gestión de Estaciones
- BD LNMySR

- Datos Estaciones

- Usuario
- Sistema de Gestión de Estaciones
- Protocolo de Comunicación
- BD LNMySR

Descripción de los Símbolos utilizados para modelar los procesos en su nivel más detallado

Simbología	Descripción
 Símbolo Pool	<p>Es el área de actividad de un proceso el cual representan participantes en un Diagrama de procesos de Negocio interactivo o Colaborativo. Un participante se define como un rol de negocios general, por ejemplo un comprador, vendedor, transportista o proveedor.</p> <p>Consideraciones:</p> <p>Dentro de un pool pueden ir otros pool para describir con claridad cada participante o rol.</p>

 Inicio	<p>El evento de inicio señala un proceso para comenzar. Cada proceso puede tener uno y sólo un evento de inicio.</p> <p>Consideraciones: Esta figura sólo puede aparecer una vez por proceso.</p>
 Descripción de la tarea ó subproceso realizado	<p>Se usa para describir una tarea que se puede dividir en sub-tareas, el cual al interior de la figura se pone el nombre del sub-proceso</p> <p>Consideraciones: se utiliza cuando el trabajo en el proceso se modela con más detalle en un nivel de proceso inferior</p>
 Flujo	<p>Una secuencia de flujo se utiliza para mostrar el orden en que las actividades en un proceso se llevará a cabo.</p> <p>Consideraciones: Una tarea puede llegar a usar varios flujos de secuencia para comunicarse con otras tareas o compuertas.</p>
 Condicional	<p>Desvía el flujo de trabajo en todas la vías que cumplen con la condición o</p>

	las que no lo hacen.
 Descripción Si	Indica el flujo que toma el condicional si se cumple el condicional.
 Descripción No	Indica el flujo que toma el condicional si no cumple el condicional.
 Fin	El evento de fin completa el proceso o Sub-proceso, un proceso puede tener múltiples eventos de fin.

Reglas

- 1.- Este tipo de reglas se asocian directamente con los procesos, es decir, corresponden a reglas de transición, de asignación, de eventos o condiciones de visibilidad, obligatoriedad o edición de atributos dentro de las formas. Las reglas de ejecución, permiten que el proceso lleve a cabo cada regla durante su ejecución, además las reglas de ejecución pueden invocar cualquier regla de librería definida con el interés de realizar una particularización de la misma.
- 2.- Una función puede agregar y/o modificar un valor.
- 3.- Un proceso puede representar un conjunto de funciones.
- 4.- En el rol se define el actor que ejecuta o tiene que ver con las actividades.
- 5.- En un proceso pueden existir uno o más procesos.
- 6.- Por ningún motivo al empezar un proceso o subproceso se debe Expresar en la parte extremo derecha (horizontal)
- 7.- Los Procesos y subprocesos comienzan en la parte extremo izquierda horizontal.
- 8.- Cualquier otro símbolo ajeno a este lenguaje no tendrá ningún valor o significado en los símbolos que se vean afectados.

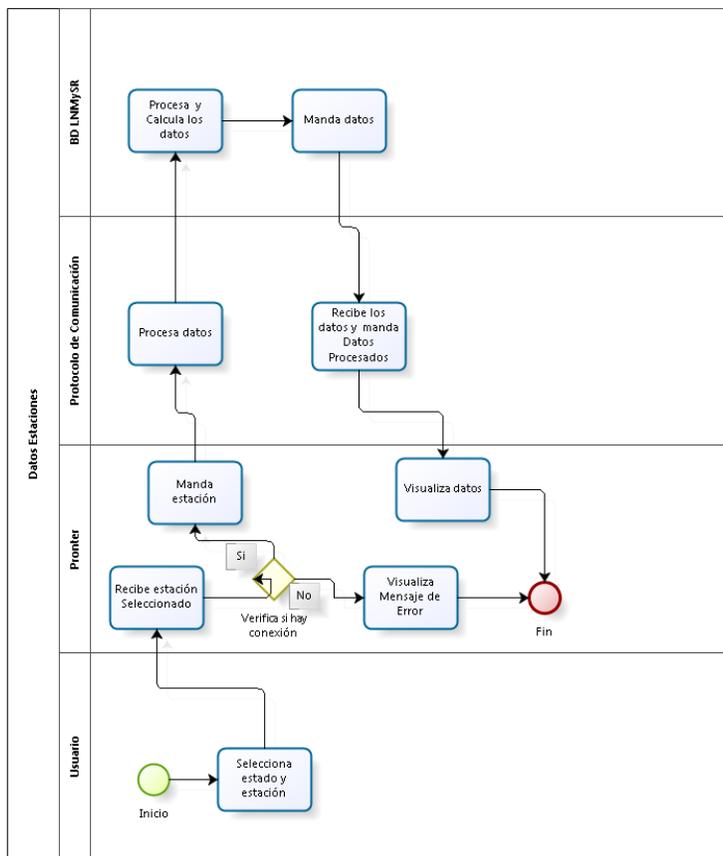
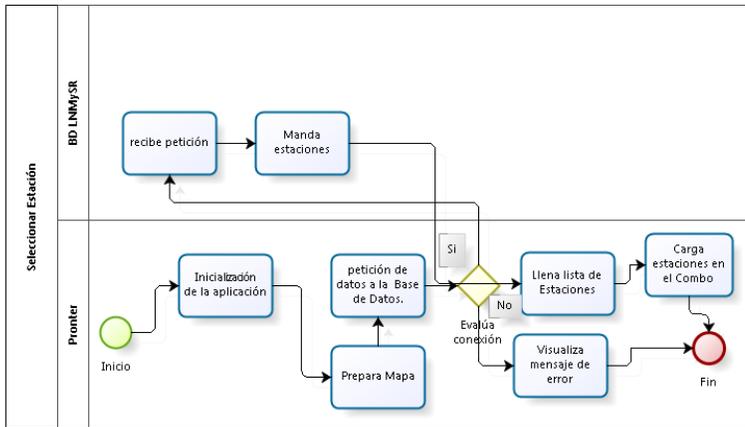


Figura 11. Detalle gráfico de los procesos

9.1 Desarrollo del programa de conexión y recepción de datos por medio de Arduino (Trabajo realizado por el inifap).

A continuación se describe el desarrollo de un prototipo de estación, que coadyuvará con el sistema de Gestión de estaciones, para controlar de manera automática, una posible falla técnica, ya que los técnicos encargados de esta actividad, han disminuido considerablemente.

Antes de la realización del programa en el software, se tuvo que realizar una simulación para poder verificar si el programa estaba realizado de forma correcta, antes de poder utilizarlo en el circuito real. Para la simulación de los sensores de temperatura, humedad relativa, radiación solar, dirección y velocidad de viento se utilizaron potenciómetros y para el sensor de precipitación se utilizó un botón. En este esquema se puso el RTC ds1307 y la memoria micro SD (tarjeta MMC en Isis® de Proteus®), para poder simular la escritura y verificar que la hora y fecha que nos proporcione el RTC ds1307 sea el correcto, solo que la escritura en la memoria micros SD no se pudo hacer en la simulación por lo que eso se tuvo que comprobar hasta que se montó el circuito real, pero se mantuvo en la simulación para ilustrar como se realizara el circuito real. Después de realizar las conexiones en la simulación se procedió a realizar el programa en el software de Arduino.

Posteriormente se tiene que declarar las variables en las cuales se guardan los datos que son recibidos de los sensores, las variables son declaradas en los pines correspondientes del sensor, en las que 5 de ellas son declaradas las entradas analógicas, la entrada del sensor de precipitación es declarada como constante mediante la sentencia #define y que se utilizara en el pin 2, que es utilizado para la modulación por ancho de pulso(PWM) y esto se debe a que es un sensor que en envía un pulso mediante un reed switch.

En la siguiente parte del programa lo que se realizo fue inicializar las librerías de software serial Serial.begin(9600) a una velocidad de 9600 baudios, la librería de la

memoria microSD (SD.begin(53)), que se seleccionó en el pin 53 que es para la función Chip Select(CS), se inicializo la librería wire mediante Wire.begin() que es necesaria para la utilización del RTC ds1307, en esta parte también se selecciona que todos los pines en donde fueron declaradas las variables que leerán los valores de los sensores serán utilizados como entradas (INPUT) mediante la función pinMode, en el caso del sensor de precipitación se indica que la variable que leerá ese dato será en una entrada digital mediante la función digitalWrite() y que estará en estado alto (HIGH), para este sensor como se utilizara en los pines de modulación por ancho de pulso (PWM) se creó una función especial llamada cantidad para poder realizar el conteo del sensor de precipitación, el cual se realizar mediante la función attachInterrupt(0,cantidad,FALLING) que es una función que manda una interrupción cuando se cumple una condición declarada, en donde el 0 indica el número de interrupción, cantidad es el nombre de la función que actuara al efectuarse la interrupción, y FALLING es la condición de que la interrupción se dará cuando el pin cambie de un estado alto (HIGH) a un estado bajo (LOW). En la última sección de esta parte se escribirán los encabezados que deberá de llevar el archivo que contendrá los registros en la memoria micro SD para que el usuario identifique que valor es el que se está guardando en cada columna.

Cabe mencionar que todas estas instrucciones deberán de estar dentro de la función void setup().

```

void setup()
{
  pinMode(temp, INPUT);           //pone los pines como entrada
  pinMode(hum, INPUT);
  pinMode(rad, INPUT);
  pinMode(vel, INPUT);
  pinMode(dir, INPUT);
  pinMode(pluviometro, INPUT);
  digitalWrite(pluviometro, HIGH); //pone los pines como entrada
  attachInterrupt(0, cantidad, FALLING); //manda a la estructura de pluviometro
  Wire.begin();                  //inicializa reloj y fecha
  Serial.begin(9600);            //inicializa comunicacion seria
  setSyncProvider(RTC.get());
  SD.begin(53);

  //escritura de los encabezados en el archivo de la SD
  datos=SD.open("prueba1.xls", FILE_WRITE); //abre el archivo en donde se guardaran los datos en la SD
  datos.print("TEMPERATURA\t"); //-----escribe en el archivo los encabezados-----
  datos.print("HUMEDAD\t");
  datos.print("RADIACION\t");
  datos.print("VELOCIDAD DEL VIENTO\t");
  datos.print("DIRECCION DEL VIENTO\t");
  datos.print("PRECIPITACION\t");
  datos.print("HORA\t");
  datos.println("FECHA");
  datos.close(); //-----escribe en el archivo los encabezados-----
}

```

Figura 12.Estructura void setup() de Arduino.

La siguiente parte del código realizado fue dentro de la estructura void loop(), que es la parte donde se realizan las instrucciones de manera cíclica, en este parte es en donde se realiza la lectura de los sensores para su conversión a señal digital, y así mandar ese valor por el modulo X-Bee® Serie 1 que se encuentra en la estación y ser recibido por el modulo que se encuentra dentro de las instalaciones, para después ser conectado al max3232 que se encargara de que la señal sea convertida para poder llevar la señal por el cable RS-232 hacia el equipo de cómputo y se pueda visualizar mediante el software LabVIEW®. El código realizado se muestra en la siguiente figura.

```

void loop()
{
  restemp=analogRead(temp); //-----lectura de los sensores e imprime los valores leídos en el puerto serie-----
  restemp1=analogRead(temp);
  restemp=restemp/8;
  if (restemp<100&&restemp>=10){Serial.print("0");}
  if (restemp<10&&restemp>=0){Serial.print("00");}
  Serial.println(restemp);
  reshum=analogRead(hum);
  reshum1=analogRead(hum);
  reshum=reshum/8;
  if (reshum<100&&reshum>=10){Serial.print("0");}
  if (reshum<10){Serial.print("00");}
  Serial.println(reshum);
  resrad=analogRead(rad);
  resrad1=analogRead(rad);
  resrad=resrad/8;
  if (resrad<100&&resrad>=10){Serial.print("0");}
  if (resrad<10){Serial.print("00");}
  Serial.println(resrad);
  resvel=analogRead(vel);
  resvel1=analogRead(vel);
  resvel=resvel/8;
  if (resvel<100&&resvel>=10){Serial.print("0");}
  if (resvel<10){Serial.print("00");}
  Serial.println(resvel);
  resdir=analogRead(dir);
  resdir1=analogRead(dir);
  resdir=resdir/8;
  if (resdir<100&&resdir>=10){Serial.print("0");}
  if (resdir<10){Serial.print("00");}
  Serial.println(resdir);
  if (conteo<10){Serial.print("00");}
  if (conteo<100&&conteo>=10){Serial.print("0");}
  Serial.println(conteo); //-----lectura de los sensores e imprime los valores leídos en el puerto serie-----
}

```

Figura 13. Lectura de valores de los sensores y envío por módulo X-Bee®.

Como se puede ver en la figura anterior, el valor leído por la variable procedente del sensor se divide entre 8, esto es debido a que la placa Arduino toma para los valores digitales el rango de 0 hasta 1023, y los sensores toman un valor de 0 hasta 128 y por ese motivo se realizó esta operación, y también se puede ver en la figura anterior que se colocaron dos condiciones if, el motivo de ello fue que al momento de leer los datos recibidos en LabVIEW® no se mostraban en la variable correspondiente ya que el valor de la dirección y velocidad del viento variaban constantemente y la visualización no se realizaba de manera correcta, es por ello que al momento de que en cualquier momento que la lectura de cualquier sensor fuera menor a 10 o 100, se colocarían dos o un cero respectivamente.

En la segunda parte de la estructura void loop(), se realizó la escritura de los valores de los sensores en la memoria micro SD, y en esta parte no se necesitó de realizar la operación de división del valor entre 8, ya que para esta parte se utilizó la función map(), que esta función realiza un re-mapeo del valor que ingresa a la variable, en la siguiente figura se denota la estructura de esta parte y de cómo se utiliza la función map().

```

datos=SD.open("prueba1.xls",FILE_WRITE); //abre el archivo en donde se guardaran los datos en la SD
restempi=map(restempi,0,1023,-40,60);
datos.print(restempi);
datos.print("\t"); //-----escribe los datos de los sensores en el archivo de la SD-----
reshuml=map(reshuml,0,1023,0,100);
datos.print(reshuml);
datos.print("\t");
resradl=map(resradl,0,1023,0,1400);
datos.print(resradl);
datos.print("\t");
resvell=map(resvell,0,1023,0,220);
datos.print(resvell);
datos.print("\t");
resdirl=map(resdirl,0,1023,0,359);
datos.print(resdirl);
datos.print("\t");
datos.print(conteo);
datos.print("\t");
datos.print(hour());
datos.print(":");
if (minute()<10){datos.print("0");}
datos.print(minute());
datos.print(":");
if (second()<10){datos.print("0");}
datos.print(second());
datos.print("\t");
if (day()<10){datos.print("0");}
datos.print(day());
datos.print("/");
if (month()<10){datos.print("0");}
datos.print(month());
datos.print("/");
datos.print(year());
datos.println(); //-----escribe los datos de los sensores en el archivo de la SD-----
datos.close(); //cierra el archivo de la SD
delay (1000);
}

```

Figura 14.Guardado de datos en memoria micro SD.

En la última parte del programa realizado en Arduino se realizó la función en donde se llevaría a cabo el conteo de los pulsos del sensor de precipitación, esta función se realizara cada vez que el pin 2 detecte un cambio en el voltaje de entrada (de estado bajo a estado alto, LOW-HIGH) y mediante la función attachInterrupt() y así la variable tendrá un incremento de 1 cada vez que sucedo eso. En la figura siguiente se muestra la función del sensor de precipitación.

```

//programa de pluviometro-----
void cantidad() //funcion para realizar el conteo del pluviometro
{
conteo++;
}

```

Figura 15.Función del sensor de precipitación.

9.2 Configuración de los módulos X-Bee® serie 1

Los módulos X-Bee® serie 1 vienen por defecto con la configuración en modo transparente, la cual es suficiente para la aplicación que se utilizó en este proyecto, por lo tanto solo se debió de comprobar que entre los dos módulos a utilizar existiera una comunicación aceptable y esto se llevó a cabo mediante el software X-CTU, lo cual se muestra en la siguiente figura.

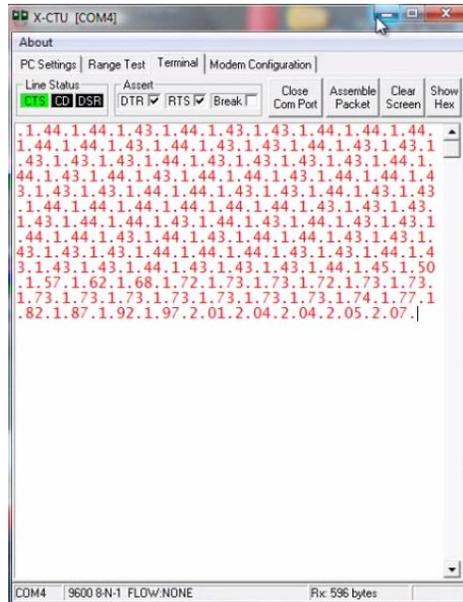


Figura 16. Comprobación de módulos X-Bee® serie 1.

En la figura anterior se muestra que el modulo receptor está recibiendo los datos del modulo emisor.

9.3 Desarrollo del software en LabVIEW®

En el software LabVIEW® será en donde el usuario podrá visualizar los datos en forma gráfica y en tiempo real, y el Sistema de Gestión de Estaciones interacciona para que la información recibida de los sensores. La versión LabVIEW® que se utilizó fue la 2012. En una primera parte del programa realizado se tuvo que inicializar la comunicación serial, en donde se especificara porque puerto se estará llevando a cabo, que velocidad de comunicación tendrá, cuanto cantidad de bits será la que estará leyendo, todo esto se lleva a cabo mediante el icono de la figura siguiente.

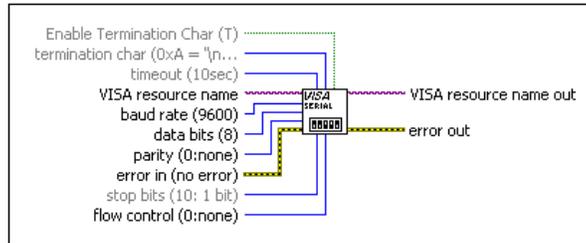


Figura 171.Configuración serial en LabVIEW®.

Al inicializar a configuración serial lo siguiente fue colocar un ciclo while general que realice todas las instrucciones de una forma cíclica hasta que el usuario presione el botón de stop, en ese ciclo while se colocó dentro otro ciclo while que realice la función de estar recibiendo los datos que estén ingresando y acomodarlos en forma correcta en el arreglo que corresponde, dentro de este ciclo while secundario también al momento de separar los valores de cada sensor se manda a la función de convertir cadenas a numero para que así se muestre el valor recibido. En la siguiente figura se muestra la descripción anterior.

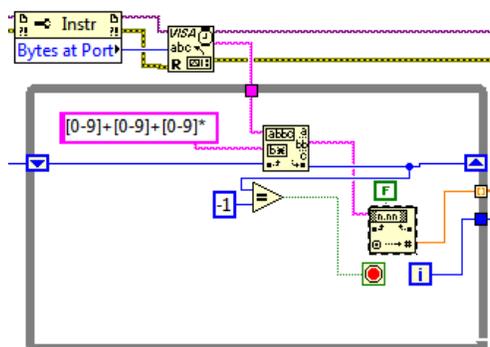


Figura 28.Ciclo while secundario.

En la siguiente parte del diagrama de bloques se realizaron las operaciones necesarias para que el valor de la señal de entrada de los sensores se muestra ya con el valor correcto, que será el que el usuario estará visualizando mediante gráficas y un valor numérico en el panel frontal del LabVIEW®. A continuación se muestra el diagrama de bloques realizado para esta parte.

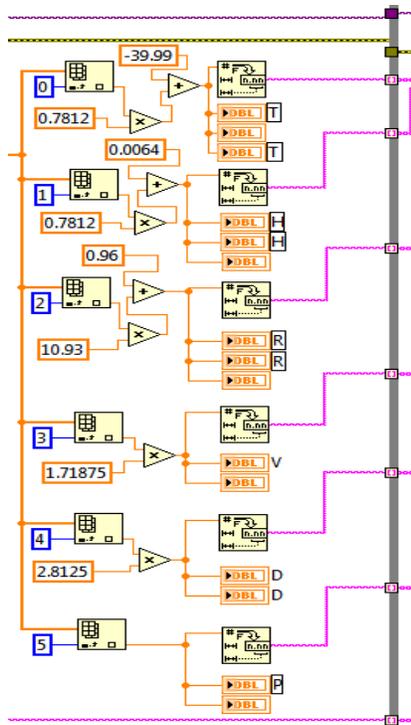


Figura 193.Diagrama de bloques de graficas de visualización.

En la última parte del diagrama de bloque lo que se realizó fue colocar un arreglo el cual estará guardando los mismos datos de los sensores en un archivo de Excel, el cual contiene los datos de los sensores, con fecha y hora en que se realiza la lectura. La estructura del diagrama de bloques de lo anterior se muestra a continuación.

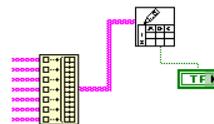


Figura 20.Diagrama de bloques de archivo en Excel.

8 RESULTADOS

Se trabajó para que el Sistema de Gestión de estaciones sea de fácil manejo, una vez inicializado en la web o en el dispositivo móvil, genera un entorno amigable y fácil de usar, que indica al usuario y/o técnico que está iniciando sus diferentes procesos para poder interactuar a través de los diferentes mecanismos para la visualización de la consulta de datos.

Se recomienda, que las metodologías para el desarrollo de aplicaciones en dispositivos móviles están en proceso de maduración se deben cuidar los siguientes puntos:

Análisis

- Fijar un objetivo(os) que se desee llegar con la aplicación web y móvil de la manera más específica y medible posible. Uno o varios objetivos son los que van a dirigir el resto de las actividades y procesos que se van a realizar. Por lo tanto una incorrecta fijación va a lastrar el resto de estrategias que se empleen, además que solo siendo específico se puede medir los resultados y contrastarlos con los objetivos para saber si se han llevado a cabo los resultados esperados con la aplicación móvil.
- Definir los macro procesos principales y evitar la duplicidad entre ellos. Cuidar que exista coherencia entre los objetivos de los subprocesos, procesos y macroprocesos.
- Desglosar los macro procesos y procesos hasta su nivel más bajo. Coherencia entre los objetivos de los subprocesos, procesos y macro procesos.
- Utilizar Business Process Management. Para representar los procesos desglosados. Esta herramienta permite que un número mayor de personas, independientemente de su formación profesional puedan entender y comprender los procesos representados de la aplicación.
- Definir un protocolo de comunicación entre la aplicación Android y la Base de datos. Para aportar interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.

Diseño

- Priorizar la simpleza. Las aplicaciones móviles deben ofrecer sólo los elementos esenciales para un uso adecuado. El uso de elementos decorativos o funcionales excesivos sólo fuerza los límites del dispositivo: una pantalla pequeña, una conexión inalámbrica (inestable) y un procesador no muy potente.
- Proteger al usuario. La seguridad también es importante en plataformas móviles. La aplicación no debe recolectar del usuario más datos que los estrictamente necesarios. El uso de SSL y HTTPS es especialmente crucial: los usuarios pueden estar conectados a través de conexiones WiFi fáciles de interceptar.
- Usar un estilo universal. La aplicación debería estar disponible para las versiones Android más utilizadas. Por eso, hay que evitar utilizar un estilo visual inspirado en el de alguno de ellos en particular.
- Cuidar la usabilidad. Es probable que no todos usen la aplicación tal como los desarrolladores imaginaron, especialmente si está dirigida a dispositivos móviles, cuyas reglas de operación todavía están en proceso de maduración. Es importante, antes de entregar la aplicación al cliente, observar cómo se la utiliza dentro de distintos focus groups convocados para tal fin (y compuestos por usuarios con distintos niveles de experiencia y habilidad).

SistemaReportesRNEA - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM SQL TOOLS TEST ANALYZE WINDOW HELP

objdb.cs Dreportes.cs Web.config HomeController.cs MonitorearVariables.cshtml RouteConfig.cs ReportesController.cs Reporte.cshtml Index.cshtml

```

[System.ActionAttribute]
public ActionResult ValidarSensores(string From, string To, string Sensores, string Cantidad)
{
    List<SelectListItem> itemsSensores = new List<SelectListItem>();

    itemsSensores.Add(new SelectListItem { Text = "Dirección del Viento", Value = "dirv" });
    itemsSensores.Add(new SelectListItem { Text = "Velocidad del Viento", Value = "velv" });
    itemsSensores.Add(new SelectListItem { Text = "Radiación Global", Value = "radg" });
    itemsSensores.Add(new SelectListItem { Text = "Humedad Relativa", Value = "humr" });
    itemsSensores.Add(new SelectListItem { Text = "Temperatura (tmax)", Value = "tmax" });
    itemsSensores.Add(new SelectListItem { Text = "Temperatura (tmin)", Value = "tmin" });

    ViewBag.Sensores = itemsSensores;

    //var obj = objdb.ObjRedclima.MdoConsultarEventosVariables(new MReporteEventosVariables("", "", "", 10, null, "dirv", DateTime.Parse("1/01/2014"), DateTime.Parse("
    if (From != "" && From != null && To != "" && To != null)
    {

```

91%

Error List

0 Errors 2 Warnings 0 Messages

Description	File	Line	Column	Project
Validation (HTML5): Element 'title' occurs too few times.	HtmlPage.html	1	8	WebSitePruebas
Validation (HTML5): Attribute 'align' is not a valid attribute of element 'div'.	Reporte.cshtml	16	6	MVCReportes

Package Manager Console | Web Publish Activity | Error List | Immediate Window | Output | Find Results 1

Ready Ln 112 Col 13 Ch 13 INS

localhost:50781/?_RequestVerificationToken=fkOpN9BdzmYV-snHExeEvu0BZ3TtPpbV1JaYgWgpDDF4--RmTUPSE-D

Cantidad

From

To

Estacion	Municipio	Mes	Año	Cantidad	Eventos
El Becerro	Aguascalientes	4	2015	0.01	6
El Becerro	Aguascalientes	5	2015	0.01	7
El Chiquihuitero	Calvillo	10	2013	0.33	8
La Salada	Aguascalientes	10	2014	0.00	31
La Salada	Aguascalientes	11	2014	0.00	29
La Salada	Aguascalientes	12	2014	0.00	16
Presa Calles	San José de Gracia	3	2013	0.00	8
Presa Calles	San José de Gracia	6	2014	0.00	30
Presa Calles	San José de Gracia	6	2015	0.02	6
Presa Calles	San José de Gracia	7	2014	0.00	30
Presa Calles	San José de Gracia	7	2015	0.02	7
Presa Calles	San José de Gracia	8	2013	0.03	7
Presa Calles	San José de Gracia	8	2014	0.00	10
Presa Calles	San José de Gracia	9	2013	0.06	6
Presa Calles	San José de Gracia	9	2015	0.01	11
Presa Calles	San José de Gracia	10	2014	0.00	14
Santa Anita	Aguascalientes	10	2013	0.00	6
Santa Lucía	San Francisco de los Romo	4	2015	0.00	10

9 CONCLUSIONES

En este proyecto se desarrolló un sistema de control de estaciones, y se dejó una capa de datos, para la generación y pruebas de la aplicaciones en Android e iOS para visualizar atrasos en las estaciones correctamente, la cual tiene el propósito de monitorear las principales variables climáticas de una región específica, por medio de este sistema de consulta con la intención de apoyar en la toma de decisiones de los técnicos a Nivel Nacional.

El sistema fue y es probado usando las tecnologías más recientes para facilitar un manejo adecuado o amigable por parte de usuario, así como proveer los datos en tiempo y forma.

10 CONCLUSIONS

In this project, a system control stations developed, and a data layer is left, for the generation and testing of applications on Android and iOS to display delays in the stations properly, which aims to monitor the main variables climate of a specific region, through this system of consultation with the intent to support the decision of the technicians at the National Level.

The system was and is tested using the latest technologies to facilitate proper or friendly management by the user and provide data in a timely manner.

11 CRONOGRAMA DE ACTIVIDADES

Actividades por quincena.	Ene-1a	Ene-2a	Feb-1a	Feb-2a	Mar-1a	Mar-2a	Abr-1a	Abr-2a	May-1a	May-2a	Jun-1a	Jun-2a
Introducción al ámbito laboral del INIFAP.												
Conocimiento de las estaciones automatizadas.												
Recabación de datos para la realización del sistema.												
Diseño del Sistema De Gestión De Estaciones.												
Desarrollo del Sistema De Gestión de Estaciones.												
Pruebas del Sistema.												
Implementación y redacción del informe.												

12 REFERENCIAS

- Adcon Telemetry*. (2 de Febrero de 2014). Obtenido de <http://www.adcon.at/>
- Arduino*. (1 de Febrero de 2014). Obtenido de www.arduino.cc
- Evans, B. W. (Agosto de 2007). *Arduino Notebook: A beginner's Reference*. Creative Commons.
- Ingenieria aplicada*. (5 de Febrero de 2014). Obtenido de <http://alvarounal.blogspot.mx>
- INIFAP. (1 de Febrero de 2014). *Red de Estaciones del INIFAP*. Obtenido de <http://clima.inifap.gob.mx/redinifap/>
- Instruments, N. (9 de Febrero de 2014). *National Instruments México*. Obtenido de <http://mexico.ni.com/>
- Instruments, N. (s.f.). *LabVIEW User Manual*.
- Integrated, M. (2007). 3.0V to 5.5V, Low-Power, up to 1Mbps, True RS-232.
- Integrated, M. (2008). DS1307 64 x 8, Serial, I2C Real-Time Clock.
- International, D. (10 de Febrero de 2014). *Digi International*. Obtenido de <http://www.digi.com/>
- Jesús Treviño García, Jesús Manuel Silva Salazar. (Mayo de 2011). Red de Sensores Inalámbrica Genérica.
- Martin Habel, George Bricker, Daniel Harris. (2010). *Getting Started whit XBee RF Modules*.
- Tdrobotica. (2014 de Febrero de 5). *Cosas de Mecatronica*. Obtenido de www.tdrobotica.co
- Telemetry, A. (Mayo de 2009). Data Sheet Rain Gauge-Standard.
- Telemetry, A. (Enero de 2011). Data Sheet Pyranometer SP-Lite.
- Telemetry, A. (Noviembre de 2012). ADCON TR1 Air Temperature & Relative Humidity Sensor.

Telemetry, A. (Noviembre de 2012). Data Sheet Wind Speed and Direction Sensors Model Pro 10/2.

Telemetry, A. (Octubre de 2013). Data Sheet ADCON Solar Panels.

Telemetry, A. (Enero de 2011). Data Sheet Adcon 3.3 Ah NiMH Battery pack.