



SEP

SECRETARÍA DE
EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de la Laguna

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

“Diseño y Desarrollo de un Sistema de Monitoreo Remoto de un Aerogenerador”

POR

Daniel Alberto Maldonado Morales

TESIS

**PRESENTADA COMO REQUISITO PARCIAL PARA OBTENER
EL GRADO DE MAESTRO EN CIENCIAS EN INGENIERÍA
ELÉCTRICA**

DIRECTOR DE TESIS

Dra. Concepción Hernández Flores

CODIRECTOR DE TESIS

Dr. Marco Antonio Arjona López

ISSN: 0188-9060



RIITEC: (15)-TMCIE-2014

Torreón, Coahuila, México,
Noviembre 2014



"2014, Año de Octavio Paz

Torreón, Coah., 26/Noviembre/2014
Dependencia: DEPI/CPCIE
Oficio: DEPI/CPCIE/147/2014
Asunto: Autorización de
impresión de tesis.

C. DANIEL ALBERTO MALDONADO MORALES
CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS EN INGENIERÍA ELÉCTRICA.
PRESENTE

Después de haber sometido a revisión su trabajo de tesis titulado:

"Diseño y desarrollo de un sistema de monitoreo remoto de un acrogenerador"

Habiendo cumplido con todas las indicaciones que el jurado revisor de tesis hizo, se le comunica que se le concede la autorización con número de registro **RIITEC: (15)-TMCIE-2014**, para que proceda a la impresión del mismo.

ATENTAMENTE
EDUCACIÓN TECNOLÓGICA FUENTE DE INNOVACIÓN

DR. JOSÉ LUIS MEZA MEDINA
Jefe de la División de Estudios de Posgrado e Investigación
del Instituto Tecnológico de la Laguna

JLM/HRSK

SECRETARÍA DE
EDUCACIÓN PÚBLICA
INSTITUTO TECNOLÓGICO
de la Laguna
División de Estudios de Posgrado
e Investigación



Agradecimientos

El presente trabajo de tesis se realizó con el apoyo monetario del Consejo Nacional de Ciencia y Tecnología. También se contó con el apoyo del Centro Mexicano de Innovación en Energía Eólica, del Fondo de Sustentabilidad Energética de la Secretaría de Energía, del Instituto de Investigaciones Eléctricas y del Instituto Tecnológico de la Laguna. Agradezco a todos ellos por las facilidades prestadas durante el desarrollo de este proyecto de tesis.

Agradecimientos

Agradezco especialmente a mis padres, Sandra M. Muñoz y Juan de la Riva G. Ya que si no fuera por ellos no estaría donde me encuentro hoy.

También agradezco a mis asesores, la Dra. Concepción Hernández Flores y al Dr. Marco Antonio Arjona López, por todo el apoyo que me han brindado.

Quiero agradecer a mis mejores amigos Jorge Arturo Reyes Ronquillo, a Fernando Arturo Ramírez Sánchez, a Cristian Espinoza Rojas, a Geshel Guerrero López; a mis amigos Manuel Pineda Arciniega, Juan Manolo Fernandez Gallardo, a Edgar García, a David Manríquez y a mis demás compañeros de laboratorio por el apoyo que me brindaron durante la maestría.

Das ist erst die Spitze des Eisbergs...

Índice

CAPÍTULO 1. INTRODUCCIÓN	1
1.1 OBJETIVO DE LA TESIS	1
1.2 JUSTIFICACIÓN	1
1.3 PERFIL DE LA TESIS	2
CAPÍTULO 2. AEROGENERADORES	3
2.1 HISTORIA	3
2.2 TIPOS DE AEROGENERADORES	4
2.3 PARTES DE UN AEROGENERADOR.	8
2.4 SISTEMAS DE MONITORIO PARA UN AEROGENERADOR	9
2.4.1 SCADA	10
2.4.2 LabView	12
2.4.3 Android	15
2.4.4 GSM	17
CAPÍTULO 3. ADQUISICIÓN DE DATOS DE UN AEROGENERADOR	19
3.1 ADQUISICIÓN POR MEDIO DE UN MICROCONTROLADOR Y LABVIEW	19
3.1.1 Microcontrolador	19
3.1.2 Transmisión Serie	21
3.1.3 Adquisición de los datos	22
3.2 ADQUISICIÓN POR MEDIO DE UN DAQ Y LABVIEW	28
3.2.1 DAQ NI USB-6008	28
3.2.2 Creación de un Task	29
3.2.3 Adquisición de Datos	35
3.3 USO DE SMS PARA EL ESTADO DEL GENERADOR	37
3.3.1 Módulo GSM para el envío de información	37
3.3.2 Comandos AT	39
3.3.3 Formato de SMS	41
CAPÍTULO 4. MÉTODOS PARA ALMACENAR LOS DATOS DEL AEROGENERADOR	44
4.1 ARCHIVOS DE TEXTO	44
4.1.1 Archivos XML	45
4.1.2 JSON	46
4.2 BASE DE DATOS	50
4.2.1 Base de Datos en MySQL	51
4.2.2 Creación de Usuario Para la BD Generadores	55
4.2.3 ODBC para la comunicación con LabView y MySQL	57
4.2.4 LabSql	60
4.2.5 Uso de Procedimientos Almacenados para almacenamiento local y remoto de los datos	61
CAPÍTULO 5. PLATAFORMAS PARA LA VISUALIZACIÓN DE LOS DATOS	64
5.1 ANDROID	64
5.1.1 Aplicación para la visualización del estado del aerogenerador a base de SMS.	66
5.1.2 Visualización de los Mensajes	67
5.1.3 BroadcastReceiver	69
5.1.4 Almacenar mensajes en SQLite para Android	70
5.2 APLICACIÓN WEB	70
5.2.1 Obtener datos del Servidor	70

5.2.2	<i>JQuery, Ajax y PHP</i>	72
5.3	APLICACIONES EN LABVIEW	73
5.3.1	<i>Aplicación en computadora local.</i>	74
5.3.2	<i>Aplicación en el servidor remoto</i>	75
CAPÍTULO 6. CONCLUSIONES		77
6.1	RESULTADOS	77
6.2	CONCLUSIONES	78
6.3	TRABAJO FUTURO	80
REFERENCIAS		81

Índice de Figuras

FIGURA 2-1 MOLINO DE VIENTO.	4
FIGURA 2-2 INCIDENCIA DEL VIENTO EN EL AEROGENERADOR, BARLOVENTO Y SOTAVENTO.	5
FIGURA 2-3 AEROGENERADORES DE 1, 2 Y 3 ÁLABES	5
FIGURA 2-4 AEROBOMBA UTILIZADA PARA LA EXTRACCIÓN DE AGUA.	6
FIGURA 2-5 DEL LADO IZQUIERDO SAVONIUS, EN EL CENTRO PANÉMONAS Y POR ÚLTIMO, DEL LADO DERECHO SE ENCUENTRA DARREIUS.	7
FIGURA 2-6 PARTES DE UN AEROGENERADOR.	9
FIGURA 2-7 VISIÓN GENERAL DE LOS PROCESOS DE MONITOREO DE CONDICIONES Y MANTENIMIENTO PARA UN AEROGENERADOR	10
FIGURA 2-8 ESQUEMA DE EJEMPLO DE UN SISTEMA SCADA.	11
FIGURA 2-9 DIAGRAMA DE UN PROGRAMA PARA EL CONTROL DE TEMPERATURA.	13
FIGURA 2-10 MONITOREO DE UNA COMPUTADORA A TRAVÉS DE ANDROID.	16
FIGURA 2-11 APLICACIÓN DEDICADA PARA EL MONITOREO DE AEROGENERADORES.	16
FIGURA 2-12 SISTEMA DE MONITOREO DE VOLTAJE QUE IMPLEMENTA TECNOLOGÍA GPRS PARA ENVIAR SMS.	18
FIGURA 3-1 DIAGRAMA A BLOQUES DE CÓMO SE COMPONE UN MICROCONTROLADOR.	20
FIGURA 3-2 TRANSMISIÓN DE LOS DATOS A TRAVÉS DE UNA LÍNEA.	21
FIGURA 3-3 CONFIGURACIÓN DEL BLOQUE VISA.	23
FIGURA 3-4 VI PARA ENVIAR LOS DATOS POR EL PUERTO SERIE.	24
FIGURA 3-5 VI PARA LEER DATOS POR EL PUERTO SERIE.	24
FIGURA 3-6 BLOQUE DEL CICLO WHILE.	25
FIGURA 3-7 PARÁMETROS PARA LA COMUNICACIÓN SERIE EN LABVIEW.	26
FIGURA 3-8 CICLO PARA INICIAR EL ENVIÓ DE DATOS.	26
FIGURA 3-9 VTS PARA ESTABLECER LA COMUNICACIÓN CON EL MICROCONTROLADOR.	26
FIGURA 3-10 CICLO INFINITO PARA ENVIAR LOS DATOS.	27
FIGURA 3-11 CICLO INFINITO EN LABVIEW.	27
FIGURA 3-12 NI USB-6008.	28
FIGURA 3-13 PANEL DE CONFIGURACIÓN.	29
FIGURA 3-14 CREAR NUEVO TASK PARA EL DISPOSITIVO NI USB-6008.	30
FIGURA 3-15 SELECCIÓN DEL TASK FÍSICO.	31
FIGURA 3-16 TIPO DE SEÑAL DE ENTRADA DEL DISPOSITIVO NI USB-6008.	32
FIGURA 3-17 SELECCIÓN DE LOS CANALES DEL DISPOSITIVO NI USB-6008.	33
FIGURA 3-18 DEFINICIÓN DEL NOMBRE PARA EL TASK.	34
FIGURA 3-19 TASK AGMV CREADO.	34
FIGURA 3-20 ÁREA PARA CONFIGURACIÓN Y VISUALIZACIÓN DE LOS CANALES.	35
FIGURA 3-21 VI PARA ADQUIRIR LOS DATOS DEL DISPOSITIVO NI USB-6008.	35

FIGURA 3-22 BLOQUES PARA LA ADQUISICIÓN, VISUALIZACIÓN Y ALMACENAMIENTO DE LAS SEÑALES.	36
FIGURA 3-23 FORMA VISUAL PARA ASIGNAR EL TASK Y ADQUIRIR LOS DATOS.	36
FIGURA 3-24 A) ES EL LADO FRONTAL DEL MÓDULO GSM, B) ES EL LADO POSTERIOR DEL MÓDULO GSM.	38
FIGURA 3-25 ARQUITECTURA DEL MÓDULO GL865-QUAD.	38
FIGURA 3-26 FUNCIÓN PARA COMPROBAR QUE EL MÓDULO GL865-QUAD ESTE FUNCIONANDO CORRECTAMENTE.	39
FIGURA 3-27 FUNCIÓN PARA BORRAR LOS MENSAJE DE LA BUZÓN DE ENTRADA.	40
FIGURA 3-28 FUNCIÓN PARA DEFINIR EL TIPO DE MENSAJE.	41
FIGURA 3-29 DIAGRAMA DE FLUJO PARA EL FORMATO DE LOS SMS.	42
FIGURA 3-30 FUNCIONES PARA ENVIAR EL MENSAJE.	42
FIGURA 3-31 FUNCIÓN PARA ENVIAR EL SMS.	43
FIGURA 4-1 DIAGRAMA DE UN ARCHIVO XML.	45
FIGURA 4-2 SINTAXIS DE ARCHIVO XML.	46
FIGURA 4-3 ARCHIVO DE MONITOREO EN JSON.	47
FIGURA 4-4 LECTURA DE LOS ARCHIVOS XML.	48
FIGURA 4-5 LECTURA PARA LOS ARCHIVOS JSON.	48
FIGURA 4-6 CARPETAS DONDE SE ALMACENAN LOS ARCHIVOS DE MONITOREO DEL AEROGENERADOR.	49
FIGURA 4-7 TABLA CON INFORMACIÓN ALMACENADA.	50
FIGURA 4-8 ESQUEMA EER PARA LA CREACIÓN DE LA BASE DE DATOS.	52
FIGURA 4-9 TABLA GENERADORES.	53
FIGURA 4-10 TABLA VARIABLES.	53
FIGURA 4-11 REPRESENTACIÓN DE UNO A MUCHOS.	54
FIGURA 4-12 MYSQLWORKBENCH.	54
FIGURA 4-13 MENÚ DE HERRAMIENTAS PARA EJECUTAR LOS SCRIPTS EN SQL.	54
FIGURA 4-14 MENÚ DE MYSQLWORKBENCH PARA ADMINISTRAR LA BD.	55
FIGURA 4-15 AGREGAR USUARIO.	56
FIGURA 4-16 CAMPOS PARA AGREGAR UN USUARIO NUEVO.	56
FIGURA 4-17 ADMINISTRAR ROLES PARA USUARIOS EN LA BD.	57
FIGURA 4-18 PARA ALMACENAR EN LA DB REMOTO.	58
FIGURA 4-19 ACCESO A SERVIDOR REMOTO A BASE DE UN DNS.	59
FIGURA 4-20 ODBC PARA ALMACENAR EN LA BD LOCAL.	60
FIGURA 4-21 ESQUEMA PARA OBTENER LOS GENERADORES DE LA BD LOCAL.	61
FIGURA 4-22 SQL PARA INSERTAR UN AEROGENERADOR.	62
FIGURA 4-23 CREACIÓN DE SP PARA INSERTAR GENERADORES.	62
FIGURA 5-1 IDE ANDROID STUDIO.	65
FIGURA 5-2 ESTADÍSTICAS DE DISTRIBUCIÓN DEL SO ANDROID DEL DÍA 3 DE NOVIEMBRE DEL 2014.	65
FIGURA 5-3 PAQUETES Y CLASES DE LA APLICACIÓN PARA LA VISUALIZACIÓN DEL ESTADO DEL AEROGENERADOR.	66
FIGURA 5-4 VENTANAS PARA MOSTRAR LOS AEROGENERADORES ALMACENADOS Y PARA ALMACENAR AEROGENERADORES.	67

FIGURA 5-5 INFORMACIÓN DEL AEROGENERADOR EN LA BD.	68
FIGURA 5-6 VISUALIZACIÓN DE LA INFORMACIÓN DE LOS SMS.	68
FIGURA 5-7 MÉTODO QUE OBTIENE LOS SMS DE ENTRADA.	69
FIGURA 5-8 DIAGRAMA EER DE LA BD PARA ANDROID.	70
FIGURA 5-9 APLICACIÓN WEB PARA MOSTRAR LOS DATOS ALMACENADOS EN LA BD.	71
FIGURA 5-10 GRÁFICAS DE LOS DATOS DEL VOLTAJE.	71
FIGURA 5-11 SELECCIÓN DE FECHA PARA VISUALIZAR LOS DATOS DE LA BD.	72
FIGURA 5-12 FUNCIÓN EN JQUERY PARA OBTENER LOS DATOS USANDO AJAX.	73
FIGURA 5-13 ARCHIVO PHP PARA OBTENER LOS AEROGENERADORES DE LA BD.	73
FIGURA 5-14 INTERFAZ PARA LA VISUALIZACIÓN DE LOS DATOS EN LA COMPUTADORA LOCAL.	74
FIGURA 5-15 BLOQUES PARA ENVIAR LA INFORMACIÓN VÍA TCP/IP.	74
FIGURA 5-16 APLICACIÓN PARA LA VISUALIZACIÓN REMOTAMENTE DE LOS DATOS DEL AEROGENERADOR.	75
FIGURA 5-17 PROGRAMA A BLOQUES PARA MOSTRAR LA INFORMACIÓN.	76

Resumen

El presente trabajo de tesis presenta el desarrollo de un sistema de monitoreo remoto para un aerogenerador, el cual permitirá sustituir el mantenimiento tradicional con una herramienta de monitoreo que utiliza tecnología de punta y permite hacer frente a los problemas de operación y mantenimiento tradicionales. El sistema utiliza tecnologías basadas en el sistema global para comunicaciones móviles, GSM. Se utilizó el servicio de mensajes cortos, SMS, para implementar una aplicación en un teléfono celular con el sistema operativo Android. También se desarrolló una aplicación en LabView para adquirir datos por medio del puerto serie con un microcontrolador y por un módulo de adquisición de datos NI USB-6008. Se utilizaron bases de datos en MySQL para almacenar los datos de los distintos aerogeneradores; se creó una base de datos tanto en una computadora local de monitoreo como en un servidor externo. Finalmente, se desarrolló una página web en la cual se pueden consultar los datos obtenidos de algún aerogenerador, básicamente desde cualquier parte del mundo. Para obtener los datos desde la página web se necesita una conexión de la página con la base de datos del servidor, la cual se obtiene mediante tecnologías como HTML, JQuery y PHP.

Abstract

The present MSc thesis deals with the development of a remote monitoring system for the monitoring of wind turbine generator variables, which will allow traditional monitoring replacement with a monitoring tool that uses leading edge technology that makes possible to deal with traditional operation and maintenance problems. The developed system employs communication technologies based on GSM and SMS text messages in order to implements an application for a cell phone with Android operating system. It was also developed a LabView application for acquire wind turbine generator data using the serial port and a microcontroller unit together with a NI USB-6008 data acquisition unit. Databases technology and MySQL were used to store different wind turbine generator data; databases were created both, in a monitoring local computer and in an external server. Finally, a web page was developed in which the obtained data from any stored wind turbine generator can be consulted, basically, from any part of the world. In order to obtain data from the web page a connection between the web page and the database server must be established; this connection is obtained by using different technologies such as HTML, JQuery and PHP.

Capítulo 1. Introducción

1.1 Objetivo de la Tesis

El propósito del proyecto de tesis consiste en desarrollar un sistema para efectuar el monitoreo, en forma remota, de las variables asociadas con un aerogenerador. El monitoreo se hará usando el protocolo abierto de mensajes y un servidor; alternativamente se usará también telefonía celular, y el registro se hará en dispositivos móviles o computadoras personales con acceso a internet, para que desde cualquier lugar se pueda efectuar el monitoreo del estado operativo de la turbina, generador eléctrico, convertidor electrónico, velocidad del viento y energía generada. Se emplearán microcontroladores y el sistema operativo Android.

1.2 Justificación

Con el incremento en la población mundial y la escasez de los recursos naturales no renovables como el petróleo y el gas natural, se han buscado nuevos horizontes en cuanto a las fuentes de energía eléctrica. Muchas fuentes pueden no ser muy factibles por la dificultad de construir una planta de ese tipo para generar la energía eléctrica. Las energías provenientes del sol, del viento y del mar representan la mejor solución para los problemas ambientales de nuestro planeta. Problemas tales como el calentamiento global y la lluvia ácida están siendo de gran preocupación para todos los que habitamos este planeta, por lo que nos impulsan a buscar las fuentes de energía más aptas para el lugar en donde vivimos. La energía eólica es una fuente renovable de energía, confiable, accesible, eficiente y de bajo costo y que además, se puede encontrar en el estado de Coahuila.

Los cambios políticos globales y la reestructuración de la industria de la electricidad sugieren que es tiempo de re-examinar el estado en el que se encuentra la energía eólica. Con un promedio anual de crecimiento en generación cada vez mayor, la energía eólica, actualmente no sólo es la tecnología más prometedora de fuente de energía renovable, sino que también es la fuente de energía con más auge en cuanto a crecimiento en el mundo. Una planta de generación de energía eléctrica de tipo eólica contiene entre sus elementos, un rotor formado generalmente por aspas o álabes, que está acoplado a un generador eléctrico y que mediante componentes electrónicos realiza la conexión a la red eléctrica. Adicionalmente puede llevar un sistema de engranajes y la torre que sostiene a la turbina y al generador.

1.3 Perfil de la Tesis

Esta tesis esta conformada por cinco capítulos los cuales se describen a continuación:

Capítulo 1 Introducción

Básicamente este capítulo describe el objetivo por el cual se desarrolló este trabajo de tesis, así como también su justificación.

Capítulo 2 Aerogeneradores

Este capítulo contiene una breve historia del comienzo de los aerogeneradores y un poco de su evolución, los tipos de aerogeneradores que existen y las partes que los componen.

Capítulo 3 Adquisición de Datos de un Aerogenerador

Este capítulo trata los métodos para adquirir las variables que son relevantes de un aerogenerador. Por medio de diferentes medios como la comunicación serie con un microcontrolador o un módulo de adquisición de señales e incluso por comunicación celular.

Capítulo 4 Métodos para almacenar los datos del aerogenerador

En este capítulo se describen los métodos empleados para almacenar los datos que se pueden obtener de un aerogenerador, considerando desde archivos de texto plano hasta la implementación de una base de datos.

Capítulo 5 Plataformas para la visualización de los datos

Este capítulo describe las diferentes plataformas utilizadas para visualizar los datos que se pueden obtener de un aerogenerador, las cuales van desde un teléfono celular, hasta una aplicación para escritorio desarrollada en LabView o una página web alojada en un servidor remoto.

Capítulo 6 Conclusiones

El capítulo seis presenta un resumen del trabajo desarrollado, el cual va desde la justificación del trabajo hasta las tres plataformas que fueron desarrolladas, para permitir la visualización de los datos monitoreados. Este capítulo permite apreciar todas las tecnologías de desarrollo de software que se utilizaron en este trabajo de tesis, las cuales son puestas a disposición de los usuarios para obtener los datos de un aerogenerador y poderlos consultar para tomar las decisiones apropiadas.

Capítulo 2. Aerogeneradores

Los aerogeneradores son máquinas que transforman la energía contenida en la fuerza del viento, en energía mecánica y después esa energía mecánica es transformada en energía eléctrica, por medio de un generador eléctrico [1]. Esta energía se puede usar para hogares o pequeñas comunidades, como pueden ser algunos negocios; también es posible administrarla a las redes eléctricas para que se distribuya a diferentes partes.

2.1 Historia

Durante miles de años, la humanidad ha estado fascinada por el reto de dominar el viento, para alcanzar la movilidad sin límites mediante el uso de las fuerzas de la naturaleza, ampliando así el horizonte del mundo conocido desde la antigüedad.

Por ejemplo un molino es una máquina que transforma el viento en energía aprovechable, la cual proviene de la fuerza del viento sobre unas aspas oblicuas unidas a un eje común. El eje giratorio puede conectarse a varios tipos de maquinaria para moler grano, bombear agua o generar electricidad. Cuando el eje se conecta a una carga, como una bomba, recibe el nombre de molino de viento; en la Figura 2-1 se puede observar un molino de viento usado para moler grano. Si se usa para producir electricidad se le denomina generador de turbina de viento. En 1802 fue la primera vez que se usó el viento para obtener energía eléctrica, Lord Kelvin ensayó asociar un generador eléctrico con un motor eólico, pero fue hasta 1850 con la aparición del dínamo que aparecieron formalmente los aerogeneradores [2].

Posteriormente se hicieron investigaciones aerodinámicas importantes para la Aeronáutica, y fue hasta 1961 que aparecieron los motores eólicos, pero fue hasta 1973, cuando se dio el alza del petróleo [2]; en este punto se fue dando relevancia a la realización de aerogeneradores, creándose lo que se conoce actualmente como parques eólicos, que se pueden conectar a las redes de distribución o a las líneas de transmisión con máquinas desde 300 kW a 3000 kW, agrupados en sectores de 3 a 10 máquinas, aunque también hay zonas aisladas, con una máquina, que se pueden dar en forma combinada con otras tecnologías, como la solar.

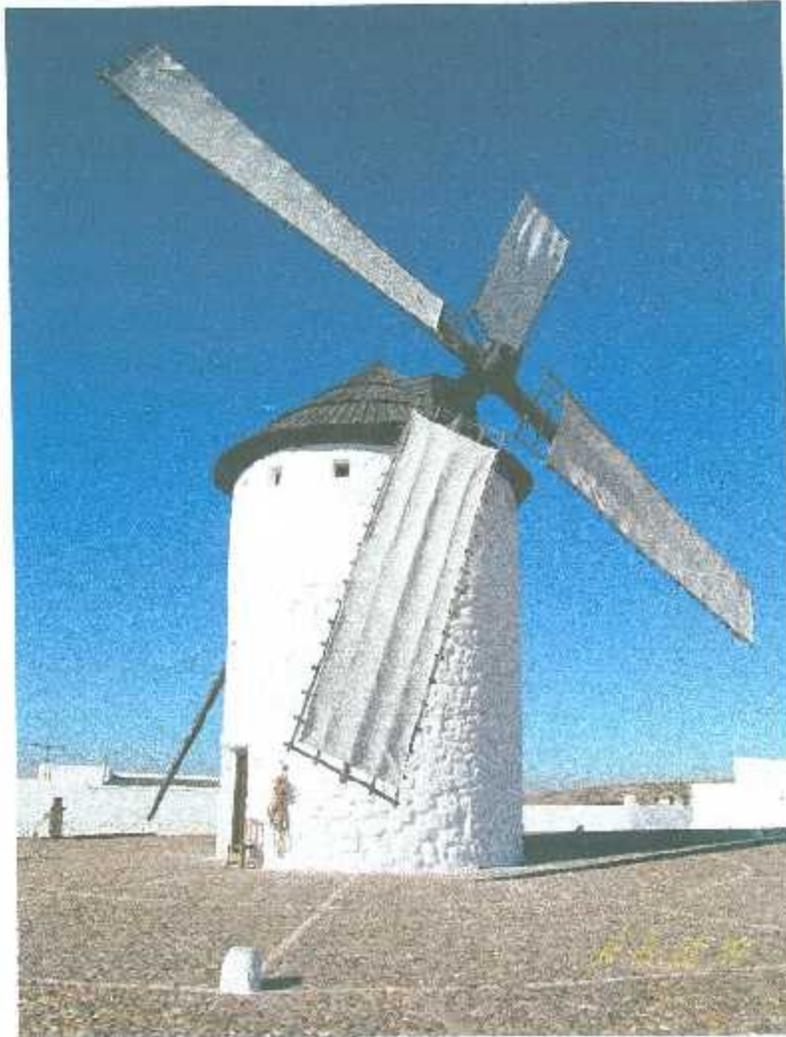


Figura 2-1 Molino de viento.

2.2 Tipos de Aerogeneradores

Los aerogeneradores pueden clasificarse en eje horizontal y eje vertical.

- **Eje Horizontal:** El eje está principalmente paralelo al suelo, son los más utilizados y corresponden a los aerogeneradores que conforman los parques eólicos. La Figura 2-2 muestra la incidencia del viento en estos generadores, siendo dos las formas en las que el viento incide en ellos.

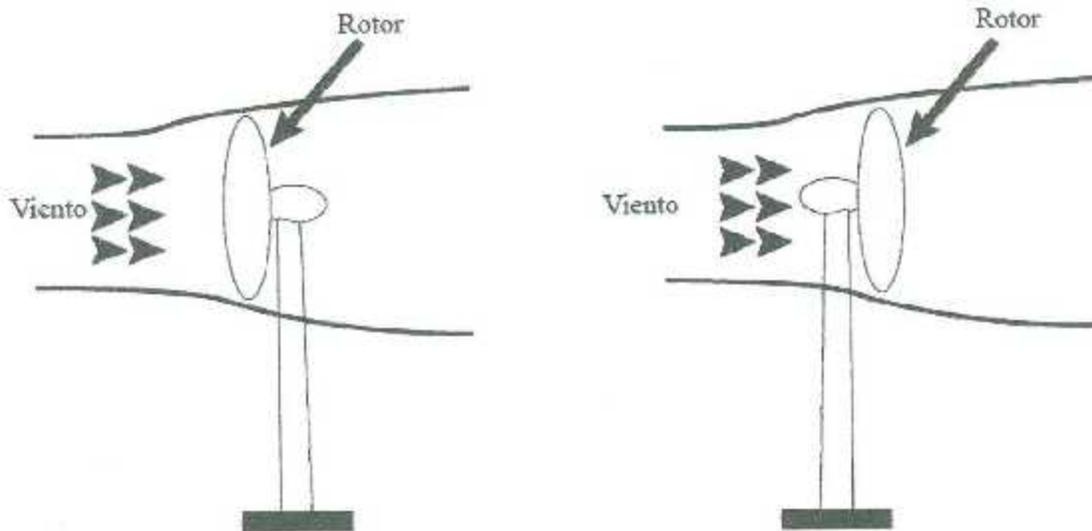


Figura 2-2 Incidencia del viento en el Aerogenerador, barlovento y sotavento.

a) A barlovento.

Cuando el rotor recibe el viento de frente. El aerogenerador necesita un dispositivo orientador para que de esta forma el viento le esté dando de frente constantemente.

b) A sotavento.

En este caso, el viento ataca el lado de la góndola, la cual tiene un diseño especial. El aerogenerador no necesita mecanismo de orientación, ya que el viento lo posiciona de la forma correcta.

El número de álabes en este tipo de aerogeneradores varía desde un álabes hasta tres; en la Figura 2-3 se puede observar el número de álabes en cada generador.

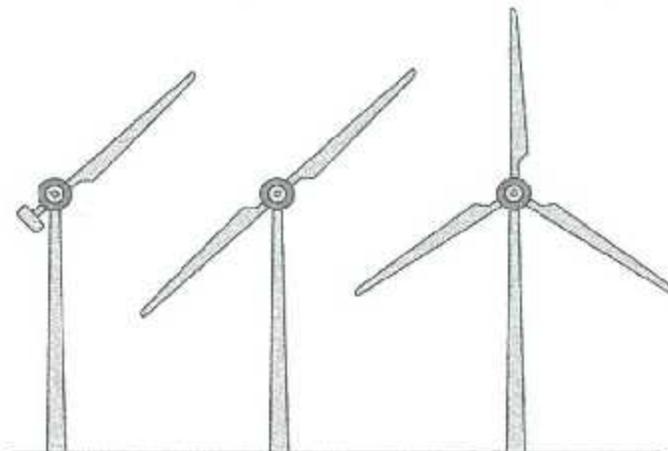


Figura 2-3 Aerogeneradores de 1, 2 y 3 álabes

Cualquiera pensaría que entre mayor sea el número de álabes el aerogenerador producirá mayor potencia, pero en realidad no es así. Si las máquinas están bien diseñadas, la potencia suministrada es más o menos la misma, dependiendo del número de álabes [3]. Usualmente las máquinas con mayor número de álabes son conocidas como *aerobombas* y la función principal de éstas es la de la extracción de agua, y aún hoy se sigue usando el mismo diseño utilizado un siglo atrás. En la Figura 2-4 se puede apreciar cómo es el diseño de una aerobomba.



Figura 2-4 Aerobomba utilizada para la extracción de agua.

- **Eje Vertical:** Los álabes del aerogenerador de eje vertical giran alrededor de su propio eje; estos son muy poco usados para la generación de electricidad debido a que su diseño no les permite girar a altas velocidades; su rendimiento es muy bajo para la generación de energía eléctrica, pero cuentan con ventajas, como la posibilidad de instalarse en el suelo y no necesitan la orientación con respecto al viento, como en los aerogeneradores de eje horizontal, que necesitan estar orientados al viento.

Existen diversas clasificaciones de eje vertical, entre las cuales se encuentran:

a) Savonius.

Su máximo rendimiento se alcanza a bajas revoluciones, debido a la gran resistencia que ofrece el viento. Para la producción de energía eléctrica debe acoplarse un multiplicador al conjunto, el cual reduce enormemente el rendimiento [3].

b) Darrieus.

Se basan en dos o tres arcos que giran alrededor de su eje. Este tipo de generador es un poco más eficaz para producir electricidad; no puede arrancar por sí solo.

c) Panémonas.

Se basan en cuatro o más semicírculos unidos al eje.

En la Figura 2-5 se puede apreciar este tipo de aerogenerador.



Figura 2-5 Del lado izquierdo Savonius, en el centro Panémonas y por último, del lado derecho se encuentra Darrieus.

2.3 Partes de un Aerogenerador.

A continuación se mencionan las partes fundamentales que componen un aerogenerador de eje horizontal.

- **Góndola:** contiene los componentes clave del aerogenerador, incluyendo el multiplicador y el generador eléctrico. En los grandes aerogeneradores, el personal de servicio puede entrar en la góndola desde la torre de la turbina.
- **Álabes:** son los que mueven el eje al recibir el viento de frente. En un aerogenerador de 600 kW, cada álabe mide aproximadamente 20 metros de longitud y su diseño es muy parecido al del ala de un avión.
- **Eje de baja velocidad:** este eje se conecta al rotor con el multiplicador. En un aerogenerador de 600 kW el rotor gira aproximadamente, de unas 19 a 30 r.p.m..
- **Eje de alta velocidad:** gira aproximadamente a 1500 r.p.m., lo que permite el funcionamiento del generador eléctrico. Está equipado con un freno de disco mecánico de emergencia. El freno se utiliza en caso de que falle el freno aerodinámico o durante el mantenimiento del aerogenerador.
- **Generador eléctrico:** suele llamarse generador sincrónico o de inducción.
- **Torre:** la torre del aerogenerador soporta la góndola y el rotor. Generalmente es una ventaja disponer de una torre alta, dado que la velocidad del viento aumenta conforme se aleja del nivel del suelo. Un aerogenerador de 600 kW tiene una altura aproximada de 80 a 100 m.
- **Anemómetro y veleta:** se utilizan para medir la velocidad y la dirección del viento. De esta forma se sabe cuándo se puede conectar el aerogenerador, hacia dónde se debe orientar y también si se debe parar. El aerogenerador se suele conectar cuando el viento alcanza aproximadamente 5 metros por segundo (m/s). Ésta es una variable de monitoreo de suma importancia, ya que si el aerogenerador excede una velocidad de 25 m/s, debe detenerse para proteger la turbina y sus elementos.

En la Figura 2-6 se muestran algunas las partes que componen un aerogenerador.

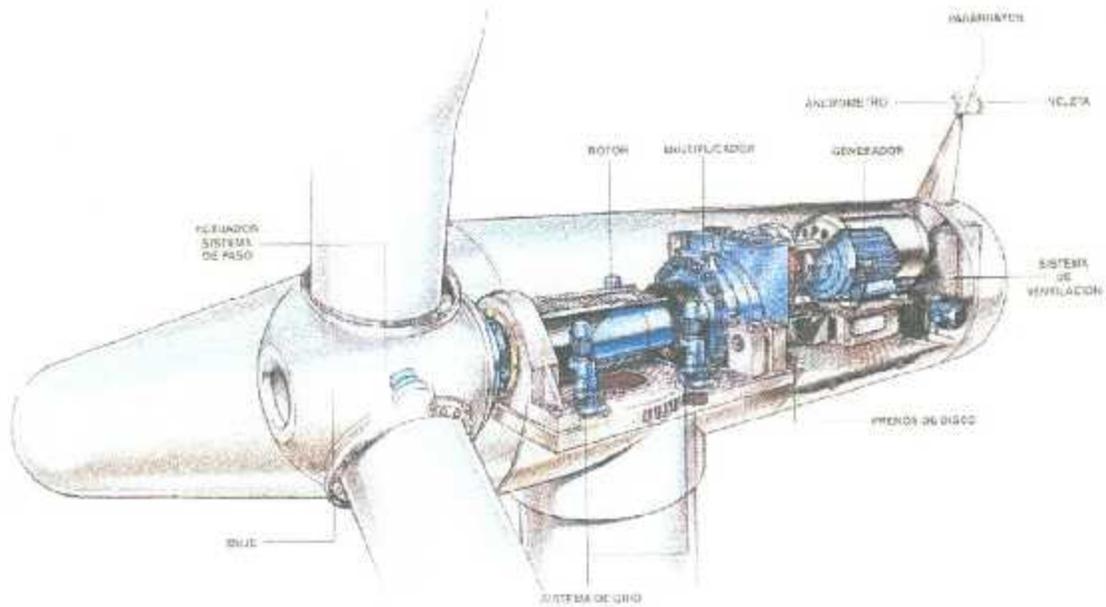


Figura 2-6 Partes de un Aerogenerador.

2.4 Sistemas de Monitoreo para un Aerogenerador

Los sistemas de monitoreo y control son una clase importante de sistemas de tiempo real. Estos comprueban los sensores que proporcionan la información sobre el entorno del sistema y llevan a cabo acciones, dependiendo de la lectura del sensor [4]. Los sistemas de monitoreo realizan una acción cuando se detecta algún valor excepcional del sensor.

A mediados de 1970, investigaciones dieron como resultado que las altas vibraciones causadas por resonancias eran las causantes de problemas como el desgaste del aislante o el agrietamiento por fatiga en los conductores; esto sólo era visible físicamente a la hora que se llevaban trabajos de mantenimiento en los aerogeneradores, y como resultado se podían generar fallos en el aerogenerador durante el tiempo útil de éste.

A finales de 1970 se patentaron sistemas de monitoreo para aerogeneradores a base de sensores ópticos de vibración, los cuales monitorcaban los datos constantemente y eran visualizados en un cuarto de control [5]. Este sistema se utilizó para verificar la estabilidad dinámica, diagnosticar los problemas de vibración relacionados y verificar el bobinado del generador.

Actualmente, el desarrollo de sistemas de monitoreo para aerogeneradores es parte fundamental para el monitoreo de variables como las corrientes o voltajes generados, la temperatura a la que se encuentra el generador, la velocidad de la turbina y la velocidad del viento, control remoto sobre el aerogenerador en caso de fallas, por ejemplo, si el aerogenerador excede la velocidad límite permitida, de esta forma es posible accionar los frenos.

También el monitoreo es de mucha utilidad al momento de la creación de parques eólicos o granjas eólicas, incluso al evaluar la zona donde se instalarán los aerogeneradores; para esto es necesario un estándar global de medición del viento y la tecnología de medición más precisa, para la evaluación de los sitios potenciales de parques eólicos y también para el seguimiento y mantenimiento de parques eólicos existentes [6]. Una vez que el parque eólico se encuentra en funcionamiento se puede detectar qué generador produce más o qué generador produce menos [5], monitorear las fallas o las variables generadas por separado.

Para hacer frente a los problemas de operación y mantenimiento, se están sustituyendo las prácticas de mantenimiento tradicionales, como el mantenimiento periódico y correctivo con el monitoreo y el mantenimiento basado en la condiciones [7]. En la Figura 2-7 se muestra un esquema de la visión general del monitoreo y mantenimiento de un aerogenerador.

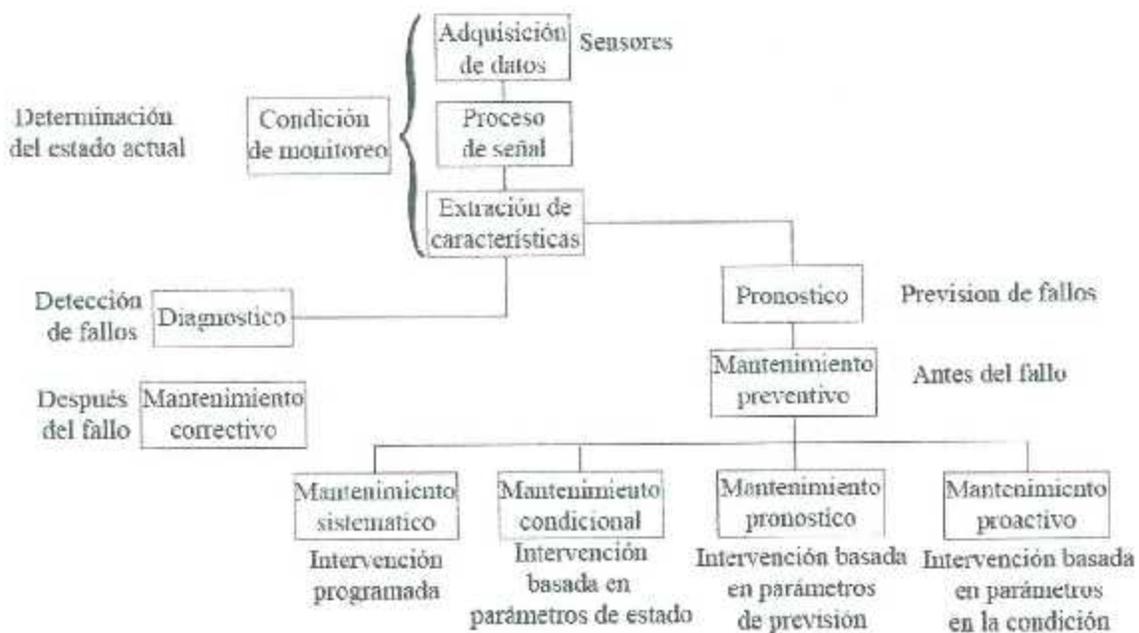


Figura 2-7 Visión general de los procesos de monitoreo de condiciones y mantenimiento para un aerogenerador

A continuación se mencionan algunos métodos para el monitoreo de aerogeneradores.

2.4.1 SCADA

La rápida expansión de los parques eólicos ha generado interés en las operaciones y el mantenimiento. Un aerogenerador operativo sufre diversos cambios de estado, incluida la transformación de un estado normal a un modo de fallo [7]. Se necesitan herramientas de mantenimiento basadas en la condición para identificar fallos potenciales en el sistema.

La supervisión del rendimiento es un método para analizar el rendimiento de un aerogenerador, como el Sistema de Control y Adquisición de Datos (SCADA, por sus siglas en inglés, *Supervisory Control and Data Acquisition*) registra en el sistema varios parámetros de aerogeneradores.

SCADA es un software de computadora que permite controlar y supervisar procesos industriales a distancia, utilizados para el control de producción; se comunica con los dispositivos y controla el proceso de forma automática desde la pantalla de una computadora. Facilita la retroalimentación en tiempo real con controles como sensores y actuadores y esto también permite controlar el proceso automáticamente; en la Figura 2-8 se muestra un esquema básico para la adquisición de datos con un sistema SCADA.

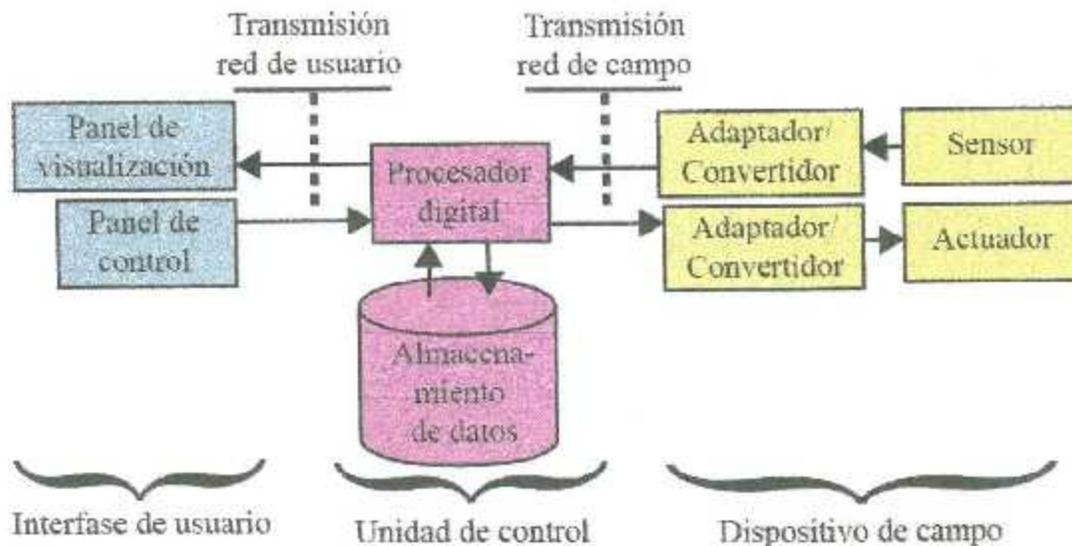


Figura 2-8 Esquema de ejemplo de un sistema SCADA.

Las funciones principales del sistema SCADA son las siguientes:

- **Adquisición de datos:** recolecta, procesa y almacena la información recibida, por ejemplo, monitorea procesos químicos, físicos o de transporte en sistemas de suministro de agua, para controlar la generación y distribución de energía eléctrica, de gas o en oleoductos y otros procesos de distribución.
- **Supervisión:** permite observar desde un monitor la evolución de las variables de control, por ejemplo, la gestión de la producción.
- **Control:** permite modificar la evolución del proceso, actuando, bien sobre los reguladores autónomos básicos (consignas, alarmas, menús, etc.) o bien directamente sobre el proceso mediante las salidas conectadas.
- **Mantenimiento:** evalúa y determina modos de fallo, índices de confiabilidad.

Por otro lado el sistema SCADA ofrece funciones más específicas como :

- **Transmisión:** transmite la información que se recolecta, a través de los dispositivos interconectados y las computadoras.

- Base de datos (BD): gestiona datos con bajos tiempos de acceso, utilizando Conectividad Abierta de Bases de Datos (ODBC, por sus siglas en inglés, *Open Data Base Connectivity*).
- Presentación: representa gráficamente los datos, por ejemplo, la Interfaz del operador o interfaz hombre máquina (HMI, por sus siglas en inglés, *Human Machine Interface*).
- Explotación: utiliza los datos adquiridos para la gestión de la calidad, control estadístico, gestión de la producción y gestión administrativa y financiera.

En ambientes de manufacturación descentralizados, la supervisión en línea y el control industrial de procesos permiten un mayor factor de producción en la industria [8].

En el área de energías renovables en aerogeneradores, el sistema SCADA es utilizado para el monitoreo y estudio de vibraciones, mediciones ópticas de estrés y análisis de partículas en aceite, entre otros [7]. Las variables como la potencia, la velocidad del generador, y la velocidad del rotor, ya antes mencionados, son parámetros de rendimiento, mientras que el ángulo de paso de los álabes y el par del generador son parámetros controlables, la velocidad del viento es la única variable no controlable.

Todas estas variables se transmiten a través del sistema SCADA hacia alguna sala de control donde los ingenieros responsables de analizar esta información en línea determinan si hay algún comportamiento anormal [9]. Un ejemplo de esto es la recopilación de datos de las fallas físicas en las cajas de cambio de los aerogeneradores, ayuda a detectar en una gran flota de turbinas, cuáles son las cajas de cambio con más probabilidades de fracasar [10].

Estos datos pueden ser recolectados a través de un módulo de adquisición de datos (DAQ, por sus siglas en inglés, *Data Acquisition*).

Por otra parte, el sistema de adquisición de datos no debe afectar al funcionamiento normal del aerogenerador y no debería plantear ningún peligro para los ingenieros de mantenimiento; es preferible que no haya ninguna conexión eléctrica entre los dispositivos instalados en la parte superior (góndola) y la parte inferior de la turbina y todas las conexiones eléctricas directas con el controlador, para esto se debe formar una barrera de aislamiento eléctrico [9].

El control de la adquisición de datos y de recolección deben ser activados desde la parte inferior de la turbina.

2.4.2 LabView

LabVIEW (por sus siglas en inglés, *Laboratory Virtual Instrument Engineering Workbench*) es un entorno de programación destinado al desarrollo de aplicaciones, similar a algunos sistemas que utilizan lenguajes de programación como C, C# o Java. Sin embargo, éste presenta una diferencia importante contra dichos programas en el siguiente aspecto, los lenguajes de programación previamente mencionados se basan en líneas de texto para crear el código fuente de un programa, mientras que LabVIEW emplea la

Todos los VI's tienen un panel frontal y un diagrama de bloques. Las paletas contienen las opciones que se emplean para crear y modificar los VI's, como se puede observar en la Figura 2-9.

Las ventajas que proporciona LabView se resumen a continuación:

- Se reduce el tiempo de desarrollo de las aplicaciones al menos de 4 a 10 veces, ya que es muy intuitivo y fácil de aprender.
- Dota de gran flexibilidad al sistema, permitiendo cambios y actualizaciones tanto del hardware (DAQ) como del software.
- Da la posibilidad a los usuarios de crear soluciones completas y complejas.
- Con un único sistema de desarrollo se integran las funciones de adquisición, análisis y presentación de datos.
- El sistema está dotado de un compilador gráfico para lograr la máxima velocidad de ejecución posible.
- Tiene la posibilidad de incorporar aplicaciones escritas en otros lenguajes.

El software LabVIEW se integra con casi cualquier hardware de cualquier proveedor en un solo entorno y ayuda a ahorrar tiempo de desarrollo con características apropiadas y estructura de programación consistente en todo el hardware.

Hoy en día, muchos de los aerogeneradores son conectados a la red eléctrica para el consumo de empresas o usuarios, los cuales son adaptados a este tipo de suministro de energía, esto conlleva a un alto desarrollo en la tecnología para la generación de energía eólica; en comparación con un aerogenerador conectado a la red eléctrica y un aerogenerador no conectado a la red eléctrica se evita el problema de control en la diferencia de tensión, diferencia de fase y diferencia de frecuencia.

Un aerogenerador no conectado a la red por lo general hace uso de la caja de cambios y adopta un rotor de paso fijo sin transmisión, lo que exige una mayor confiabilidad del rotor del aerogenerador [11].

Hoy en día LabView es usado en laboratorios para las pruebas de aerogeneradores tales como la velocidad del viento, la velocidad del rotor y energía generada. Por ejemplo, la adquisición de la corriente es obtenida con un medidor de DC la de aceleración del generador por un sensor de aceleración [11]; estos instrumentos están conectados a LabView mediante DAQ al igual que SCADA.

LabView ofrece una incomparable integración con hardware de adquisición de datos de NI (DAQ) y de terceros, bibliotecas de procesamiento de señales y controles de interfaz de usuario construidas con propósitos específicos para visualización de datos.

⊛

Para el almacenamiento de los datos obtenidos por los aerogeneradores se pueden emplear bases de datos con SQL Server [12] o incluso con MySQL (MySQL es un motor libre para manejo de bases de datos) a través de conexiones ODBC. Una desventaja con LabView es que no contiene bloques para crear bases de datos con MySQL, por lo que

hay quienes proponen usar LabSQL el cual es un conjunto de VI's libres que permiten establecer una conexión ODBC con MySQL, a través de LabView [7]

2.4.3 Android

Android es un SO (*Sistema Operativo*) para teléfonos celulares, que ha tenido gran crecimiento en los últimos años. Los dispositivos con Android ofrecen una manera fácil de transmitir información y presentarla al usuario [13]. En los últimos años hay cada vez más dispositivos con el SO Android.

A diferencia de los sistemas antes mencionados, Android ofrece una portabilidad mayor que la de una computadora, pero también carece de algunos rasgos importantes, como el espacio de almacenamiento para los datos adquiridos; con un dispositivo Android se tiene un espacio limitado de 32 GB, en cambio con una computadora se puede llegar a tener espacio ilimitado de TB de espacio. Para almacenar los datos tampoco se puede tener la velocidad de procesamiento que puede ofrecer una computadora. A pesar de eso los dispositivos Android siguen con un crecimiento exponencial para las diferentes áreas donde se necesite el monitoreo de datos en diferentes campos.

En los últimos años los dispositivos celulares han tenido un rol importante en el tema de monitoreo algunos ejemplos de ellos son:

- En cuestiones médicas para el monitoreo del pulso de una persona y que el dispositivo funcione como un electrocardiógrafo o incluso para la temperatura del paciente [14][15] o por ejemplo, casos en los que se crea un electrocardiógrafo portable para el paciente, donde el dispositivo Android registra los datos y los envía a un servidor donde el doctor puede estar monitoreando la actividad de su paciente [16].
- En cuestiones de seguridad, en la detección de movimiento a base de sensores infrarrojos que detectan la intrusión de personas [17] en el hogar, es implementado para monitorear las casas inteligentes, para el monitoreo de la temperatura, el encendido y apagado de las luces o incluso para el monitoreo de las cámaras instaladas [18].
- En la industria se pueden encontrar aplicaciones para la medición de la humedad y la temperatura en el ambiente [19].

Los teléfonos celulares con Android implementan hardware que también puede ser usado para el desarrollo de las aplicaciones, tal es el caso del GPS para trazar la ruta de un vehículo [20]; en la Figura 2-10 se muestra una aplicación desarrollada en Android para el monitoreo de una computadora.



Figura 2-10 Monitoreo de una computadora a través de Android.

En cuanto a la cuestión de energías renovables Android puede cumplir las necesidades básicas para el monitoreo de un aerogenerador o un parque eólico; en la Figura 2-11 se muestra una aplicación para el monitoreo de aerogeneradores creada por Breeze, compañía dedicada al monitoreo de granjas eólicas y aerogeneradores.

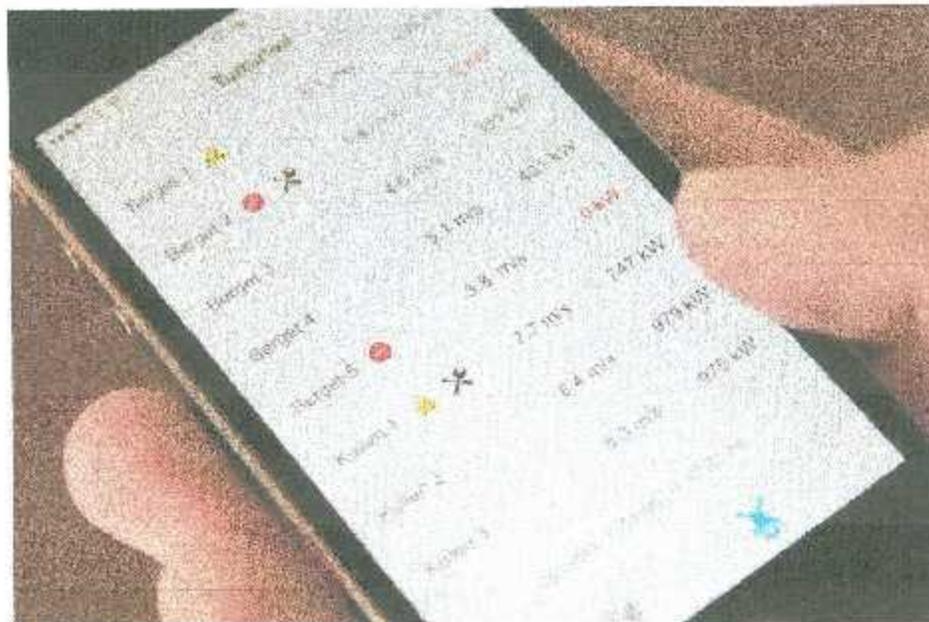


Figura 2-11 Aplicación dedicada para el monitoreo de aerogeneradores.

En algunos caso se opta por crear aplicaciones web que sean visibles en dispositivos Android, para eso es necesario implementar frameworks o tecnologías adicionales. tal es el caso de Mojarrá basado en *JavaServer Face* que se especializa en interfaces para el usuario en páginas web [21] o controles personalizados en las aplicaciones para Android.

Las aplicaciones desarrolladas para Android son capaces de conectarse a un servidor y almacenar la información para que después pueda ser vista por el usuario.

2.4.4 GSM

Sistema Global para comunicaciones Móviles (GSM, por sus siglas en inglés, *Global System for Mobile communications*), es un sistema estándar de telefonía móvil digital.

Las soluciones de adquisición de datos basada en comunicaciones inalámbricas son el foco de muchos investigadores y tienen un gran impacto en dominios como el análisis estructural, el monitoreo en cuestiones médicas, el transporte o los estudios ambientales basados en adquisición remota de datos y sistemas de análisis que ofrecen posibilidades de control de telefonía móvil a través de Servicio General de Paquetes vía Radio (GPRS, por sus siglas en inglés, *General Packet Radio Service*).

Las aplicaciones para teléfonos celulares ofrecen nuevas posibilidades de control remoto y supervisión efectivas. Estas son cada vez más confiables, interesantes y atractivas, como el registro de estacionamientos en la ciudad o monitoreo de una planta de energía. Empresas de instrumentación ofrecen software y hardware con paquetes para el desarrollo de soluciones en dispositivos celulares inteligentes (con el SO Android) [22].

La red GSM es el sistema de red de comunicación basado en tecnología de división de tiempo y de acceso múltiple por división de frecuencia, que ofrece principalmente voz, SMS, datos y otros servicios. GSM tiene características tales como la transmisión rápida de la información a menor costo, por lo que ha sido ampliamente utilizado en el sistema de control remoto [23].

Una conexión GPRS está establecida por la referencia a su nombre del punto de acceso (APN). Con GPRS se pueden utilizar servicios como *Wireless Application Protocol* (WAP) o servicio de mensajes cortos (SMS) por mencionar algunos.

El uso de SMS para el monitoreo de variables puede formar parte fundamental de un sistema de monitoreo de variables de un aerogenerador, ya que si llega a fallar la conexión de internet se pierde la comunicación con el aerogenerador; en cambio la señal de teléfonos celulares es menos propensa a fallar por el gran soporte que tiene por parte de las compañías que brindan este servicio y por otro lado, en estos tiempos la tecnología celular se encuentra en todas partes. En la Figura 2-12 se muestra cómo se implementa la tecnología GPRS para enviar SMS y de esta forma adquirir los datos de un sistema de monitoreo remoto.

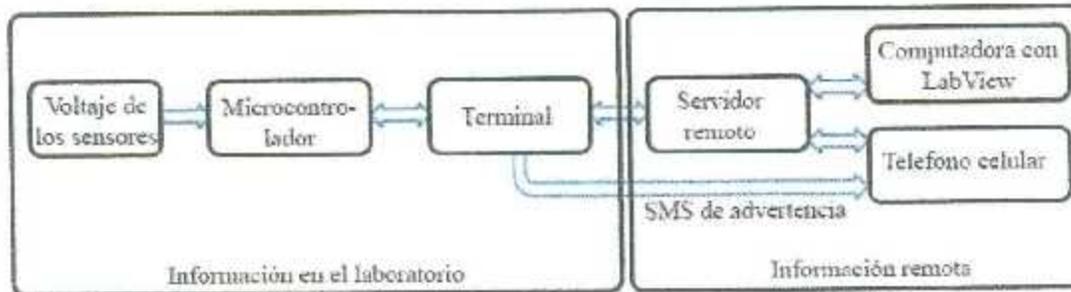


Figura 2-12 Sistema de monitoreo de voltaje que implementa tecnología GPRS para enviar SMS.

Como se observa en la Figura 2-12, el SMS es implementado como una forma de advertencia por si se produce alguna anomalía mientras se monitorea el sistema; en un aerogenerador pueden llegar a ser una parte fundamental del monitoreo de variables, ya que si se llega a producir alguna falla se enviaría un SMS a determinada persona con información específica de lo ocurrido.

El uso de la tecnología GPRS conlleva al uso de módulos GSM para el monitoreo inalámbrico en los aerogeneradores; el propósito general es aprovechar la comunicación inalámbrica que existe entre dispositivos celulares y módulos GSM, de aquí surge la idea de monitorear el aerogenerador a base de pequeños mensajes GSM [23], ya que las aplicaciones móviles no ofrecen nuevas posibilidades de control y monitoreo remoto [24].

Capítulo 3. Adquisición de Datos de un Aerogenerador

Los sistemas de adquisición de datos recogen datos de sensores para su posterior procesamiento y análisis. Estos sistemas se utilizan en circunstancias en las que los sensores han recogido grandes cantidades de datos del entorno del sistema y no es necesario procesar los datos recopilados en tiempo real. Los sistemas de adquisición de datos se usan normalmente en experimentos científicos y sistemas de control de procesos en los que los procesos físicos, tales como una reacción química, ocurren muy rápido [4].

En un aerogenerador se encuentran sensores tales como los de velocidad y dirección del viento, medidores para la corriente, la temperatura, velocidad del rotor y del generador entre otros.

3.1 Adquisición por medio de un Microcontrolador y LabView

3.1.1 Microcontrolador

Desde la invención del circuito integrado, el desarrollo constante de la electrónica digital ha dado lugar a dispositivos cada vez más complejos, entre los cuales se encuentran los microcontroladores, ya sean DSP, PIC o dsPIC, por mencionar algunos. Un microcontrolador es una computadora completa, aunque de alcance limitado, está contenido en el chip de un circuito integrado y se destina a manejar tareas en específico. Las aplicaciones de los microcontroladores son vastas y se puede decir que están limitadas por la imaginación del usuario. Es común encontrar microcontroladores en campos como la robótica y el automatismo, en la industria del entretenimiento, en las telecomunicaciones e incluso en sistemas para el monitoreo.

Los microcontroladores, también denominados de forma resumida como MCU [25], están concebidos fundamentalmente para ser utilizados en aplicaciones puntuales, es decir, en aplicaciones donde se debe realizar un pequeño número de tareas al menor costo posible [26]. Estas aplicaciones son ejecutadas en un programa almacenado permanentemente en su memoria. Por lo tanto, se pueden encontrar en su interior: Unidad Central de Procesamiento (CPU por sus siglas en inglés, *Central Processing Unit*), unidades de memoria (RAM y ROM), puertos de entrada y salida. Estas partes están interconectadas dentro del microcontrolador, y en conjunto forman lo que se conoce como microcomputadora. Se puede decir con toda propiedad que un microcontrolador es una microcomputadora completa encapsulada en un circuito integrado.

Aunque inicialmente todos los microcontroladores adoptaron la arquitectura clásica de von Neumann, en el momento presente se impone la arquitectura Harvard. La arquitectura de von Neumann se caracteriza por disponer de una sola memoria principal donde se almacenan datos e instrucciones de forma indistinta. A dicha memoria se accede a través de un sistema de bus único (direcciones, datos y control), en la Figura 3-1 se puede observar la arquitectura de un microcontrolador.

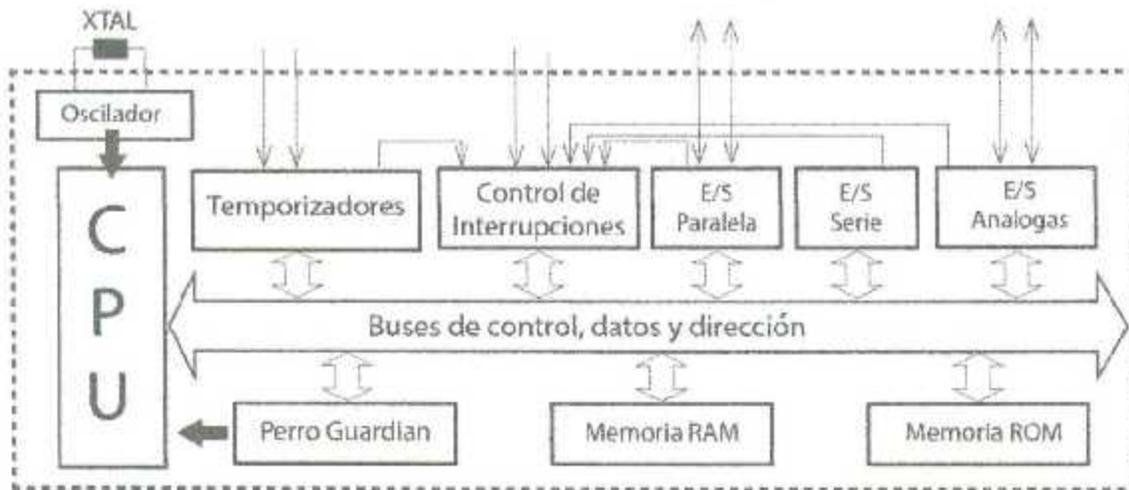


Figura 3-1 Diagrama a bloques de cómo se compone un microcontrolador.

Como se observa en la Figura 3-1, en el microcontrolador existen periféricos de entrada y salida serie y paralelo, así como otros protocolos de transmisión como I2C y UART, estos proporcionan todo lo necesario para que el diseñador tenga las herramientas necesarias al momento de desarrollar su aplicación.

Algunos de los módulos que se pueden encontrar en algunos de los microcontroladores son los siguientes:

- Puertos de E/S.
- Temporizadores.
- Conversos AD de 12 bits de alta resolución.
- Módulo UART.
- Módulo I2C.
- Módulo SPI.

Para el desarrollo de este trabajo de tesis se utilizó el puerto serie con el módulo de transmisión UART (por sus siglas en inglés, *Universal Asynchronous Receiver-Transmitter*) ó USART (por sus siglas en inglés, *Universal Synchronous/Asynchronous Receiver-Transmitter*) dependiendo de las capacidades de manejo de la señal de reloj; está destinado a soportar la comunicación serie asíncrona y síncrona, funciona de forma bidireccional, dependiendo del microcontrolador puede contar con uno o más puertos UART. La comunicación es realizada a través de 2 líneas, una UTX para transmisión y otra URX para recepción de datos.

3.1.2 Transmisión Serie

En transmisión de datos la información se transmite en fragmentos, cada uno de los cuales constituye una muestra digital compuesta por ocho pulsos binarios que reciben cada uno el nombre de bits [25]. Los ocho bits de cada muestra digital se pueden transmitir en paralelo o en serie. Los sistemas de comunicación serie en comparación con los sistemas de comunicación paralelo tienen las siguientes características:

- Transmisión a mayor distancia.
- Menor costo.
- Más sencillos en cuanto al hardware.

La transmisión serie, que es el protocolo tratado en esta tesis, se emplea en los enlaces largos de las redes de datos. En este caso, los ocho bits de cada muestra digital se envían secuencialmente, uno a la vez, sobre un solo circuito de transmisión; como la transmisión serie sólo requiere de un circuito, es mucho más económica para la transmisión de datos sobre enlaces largos, esto conlleva a una gran ventaja ya que la mayoría de los aerogeneradores de gran potencia llegan a tener alturas de 120 m. En la Figura 3-2 se muestra como es el envío de los datos en serie.

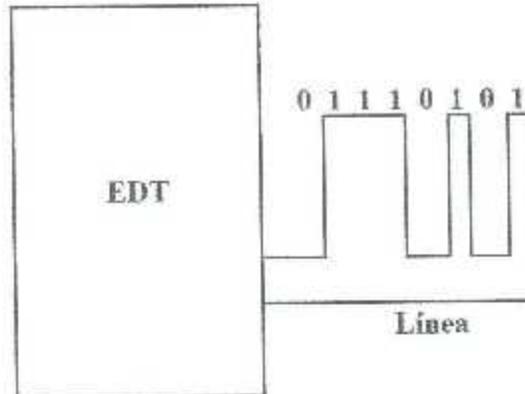


Figura 3-2 Transmisión de los datos a través de una línea.

Para la comunicación serie se pueden encontrar interfaces como RS-232, USB o Firewire, la interfaz usada para este trabajo de tesis es la RS-232.

RS-232 significa, Estándar recomendado (RS, por su siglas en inglés, *Recommended Standar*) y está definido por el Instituto Nacional Estadounidense de Estándares (ANSI, por sus siglas en inglés, *American National Standards Institute*), como la interfaz entre

un equipo terminal de datos y un equipo de comunicación de datos utilizando un intercambio binario en modo serie [27].

Los dispositivos se denominan como Equipo Terminal de Datos (DTE, por sus siglas en inglés, *Data Terminal Equipment*) por ejemplo una computadora y un Circuito terminal de Datos (DCE, por sus siglas en inglés, *Data Circuit-terminating Equipment*) que en este caso podría ser un módem o un microcontrolador.

3.1.3 Adquisición de los datos

La aplicación desarrollada para la adquisición de datos por medio del puerto serie implementa un microcontrolador PIC18F45K22, el cual utiliza el módulo UART para la transmisión serie con la computadora donde se encuentra instalado LabView.

El Ambiente de Desarrollo Integrado (IDE, por sus siglas en inglés, *Integrated Development Environment*) utilizado para desarrollar la aplicación que irá embebida en el microcontrolador es desarrollado por Mikroelektronika también conocida como Mikro en su forma más corta, la cual es una empresa con sede en Serbia Belgrade, dedicada a fabricar y vender microcontroladores, programadores para microcontroladores.

Mikroec maneja tres tipos de IDE's con diferente lenguaje de programación. Estos lenguajes de programación pueden ser C, Basic o Pascal, los cuales vienen con una amplia variedad de librerías para manejar los recursos con los que vienen integrados en los microcontroladores.

En el caso de estudio de esta tesis el IDE elegido para desarrollar fue *MikroC Pro for PIC*, el cual es un compilador ANSI C para dispositivos PIC de Microchip. Se caracteriza por tener un IDE intuitivo, un compilador poderoso con optimizaciones avanzadas SSA (por sus siglas en inglés, *Single Static Assignment*), gran cantidad de librerías de hardware y software, así como herramientas adicionales.

La librería utilizada lleva el nombre de UART, que brinda funciones para la comunicación serie; la funciones utilizadas para la comunicación con LabView por el puerto serie son las siguientes:

- `UART1_Init`: Inicializa el módulo UART deseado con la velocidad de transmisión deseada.
- `UARTx_Data_Ready`: Esta función avisa si hay datos en el buffer de entrada del puerto, si es así regresa un caracter 1, de lo contrario regresa un 0.
- `UARTx_Read`: Esta función regresa un byte que se encuentra en el buffer de entrada del puerto serie. Antes de llamar esta función se debe verificar el puerto serie con el método anterior para ver si hay datos en el buffer.
- `UART_Write_Text`: Esta función se encarga de enviar una cadena de caracteres al puerto serie.

Por otra parte, como se mencionó anteriormente, LabView es un lenguaje, y a la vez, un entorno de programación gráfica en el que se pueden crear aplicaciones de una forma rápida y sencilla. Esta programación basada en bloques, llamada programación G, representa una gran ventaja ya que ahorra tiempo de desarrollo. Usualmente si se necesita de una aplicación que se comunique vía serial con otro dispositivo, se tendrían que importar librerías, crear funciones e inicializar variables; en programación orientada a objetos (POO) se tendrían que crear clases que hereden de las librerías importadas para crear la comunicación, inicializar instancias de objetos, en cambio con LabView no es necesario hacer todo esto.

Las versiones antiguas de LabView incluían unos VI's específicos para manejar el puerto serie, pero en las nuevas versiones estos se han incluido en la librería VISA. A continuación se mencionarán los bloques para la comunicación serie.

El VI que se utiliza para inicializar la configuración serie es el *VISA Configurar Puerto Serial (VISA Configure Serial Port)*, este VI contiene las características necesarias para la comunicación serie tales como la velocidad de transmisión, paridad, o el tipo de control de flujo a utilizar [26]. En la Figura 3-3 se muestra el bloque para la comunicación serie.

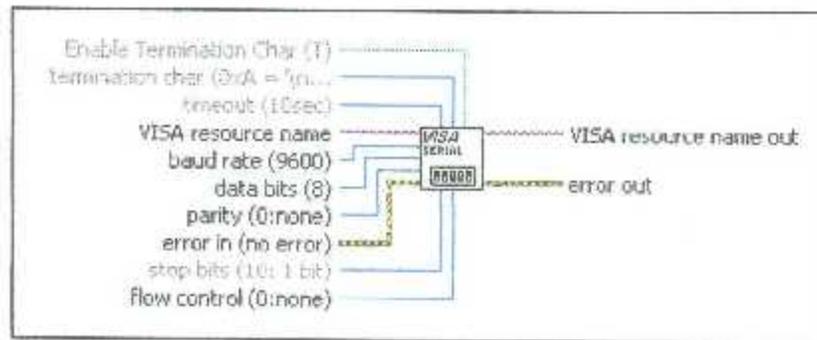


Figura 3-3 Configuración del bloque VISA.

Como se muestra en la Figura 3-3, el VI consta de parámetros necesarios para establecer la comunicación (similares a los de una función programada en C) por el puerto serie, estos parámetros son: VISA Nombre del Recurso, Velocidad en Baudios, Bits de Datos, Paridad y Bits de Parada.

- VISA Nombre del Recurso: Este parámetro será la dirección del puerto RS-232, por ejemplo el COM0 o COM1.
- Velocidad en Baudios: El valor que se introduzca en esta entrada se tomará como la velocidad de transferencia de datos, en baudios, con la que se configura el puerto serie.
- Bits de Datos: Aquí se indica el número de bits, de los bytes recibidos, que se consideran como bits de datos. Es decir, cuantos bits han de ser tomados como bits de datos en cada byte recibido.

- Bits de Parada: Aquí se definen los bits de parada que se desean utilizar en las transferencias.
- Paridad: Aquí se define el tipo de paridad que se desea utilizar en la comunicación. Se introducirá un 0 en caso de no querer emplear ningún tipo de paridad, un 1 para utilizar paridad impar o un 2 para paridad par.

En la Figura 3-4 se muestra el VI de *VISA Escritura (VISA Write)*; en pocas palabras, este bloque se encarga de escribir los datos que se van a transmitir por el puerto serie, desde LabView hacia el microcontrolador.

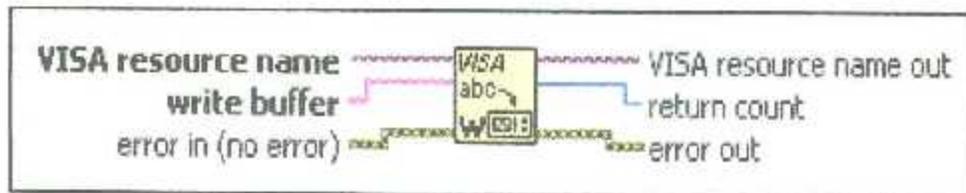


Figura 3-4 VI para enviar los datos por el puerto serie.

- VISA Nombre del Recurso: Es el mismo que se usa en el bloque de Configurar Puerto Serial, y que apunta al puerto serie donde se requiere mandar la cadena de caracteres.
- Buffer de escritura: Aquí se introducen, en forma de caracteres, los datos que se desean enviar por el puerto serie.

El siguiente VI que se utilizó para el desarrollo de la aplicación es el de *VISA Lectura (VISA Read)*, en la Figura 3-5 se pueden apreciar los parámetros que recibe de entrada.

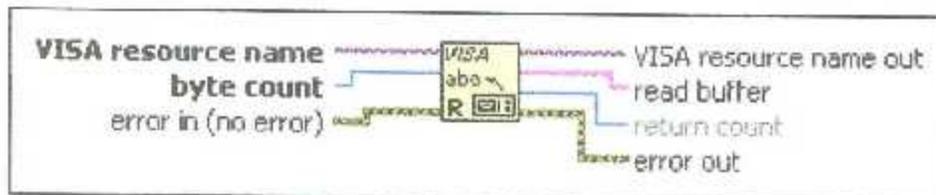


Figura 3-5 VI para leer datos por el puerto serie.

Este VI se utiliza para realizar lecturas de datos recibidos por el puerto serie.

- Contador de Byte: Aquí se indica el número de bytes que se desean leer del puerto serie, si no se indica el número, se leerá el buffer completo que llegó al puerto serie.
- Lectura del Buffer: Una vez que se ejecuta este VI, devuelve en formato de caracteres (String), los datos leídos del buffer de recepción del puerto.

Dentro de la aplicación en LabView se implementó el ciclo *while*; a diferencia del *while* de lenguajes de programación como C#, Java o C++, este es un bloque, comprendido dentro de las denominadas estructuras iterativas, también se encuentra disponible el bloque *for*. La diferencia entre el ciclo *while* de LabView y el de los demás lenguajes de programación es que actúa como un hilo por sí mismo, es decir, si se colocan dos bloques *while* estos se ejecutarán uno independientemente del otro. A continuación en la Figura 3-6 se muestra el ciclo *while*.

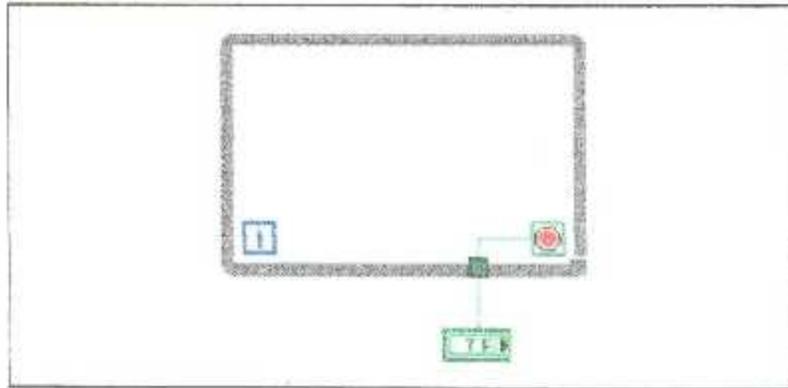


Figura 3-6 Bloque del ciclo While.

En la Figura 3-7 se puede apreciar la configuración necesaria para la comunicación; una vez que se establecen los parámetros necesarios se procederá a establecer la conexión, presionando el botón de *Conectar* de la aplicación desarrollada en LabView, el cual se observa en la Figura 3-7. Al lado de éste se encuentra un indicador LED, el cual cambia a color verde si la comunicación se estableció con éxito. LabView envía un carácter "T" para que el microcontrolador comience a enviar información; en este punto el microcontrolador se encuentra ejecutando un ciclo infinito esperando el carácter por parte de LabView. Después de que LabView envió el carácter, pasa a un estado de un ciclo infinito en espera de un carácter de respuesta del microcontrolador, esta respuesta es sólo de confirmación para LabView, de que está enterado de que empezará a enviar información. En la Figura 3-8 se observa el ciclo que espera el carácter de LabView y en la Figura 3-9 se observa como LabView envía el carácter.

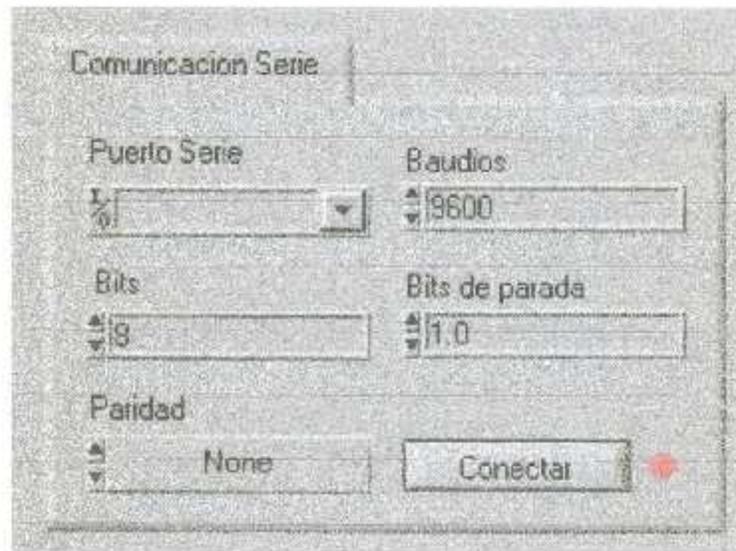


Figura 3-7 Parámetros para la comunicación serie en LabView.

```

while (1)
{
    if(UART1_Data_Ready())
        if (UART1_Read() == 'I')
        {
            UART1_Write('R');
            break;
        }
}

```

Figura 3-8 Ciclo para iniciar el envío de datos.



Figura 3-9 VI's Para establecer la comunicación con el microcontrolador.

Quando finalmente el microcontrolador está enterado de que comenzará a enviar información y LabView está preparado para recibirla, los dos entran en ciclos infinitos.

3.2 Adquisición por medio de un DAQ y LabView

La adquisición de datos es el proceso de medir con una computadora fenómenos físicos o eléctricos como voltaje, corriente, temperatura o presión. Un sistema DAQ está integrado por sensores hardware para la medición y una computadora con software para leer las señales del hardware. Comparados con los sistemas de medidas tradicionales, los sistemas DAQ basados en computadoras aprovechan la potencia del procesamiento, la productividad, la visualización y las habilidades de conectividad de las computadoras, proporcionando una solución de medidas más potente, flexible y rentable.

La función principal de un DAQ es digitalizar señales analógicas entrantes para que la computadora pueda interpretarlas. Los tres componentes clave de un dispositivo DAQ usado para medir una señal son: el circuito de acondicionamiento de señales, el convertidor analógico-digital (ADC, por su siglas en inglés, *Analog to Digital Converter*) y un bus de computadora. Varios dispositivos DAQ incluyen otras funciones para automatizar sistemas de medidas y procesos.

3.2.1 DAQ NI USB-6008

NI USB-6008, es un dispositivo para adquisición de señales fabricado por NI, el cual se utilizó para el monitoreo de señales, este dispositivo ofrece funcionalidad básica para aplicaciones como registro de datos simple, medidas portátiles. Cuenta con diez canales de entradas analógicas, dos canales de salidas analógicas y doce canales de salida y entrada digital; cuenta con una interfaz USB para la comunicación con LabView. En la Figura 3-12 se observa el dispositivo USB-6008.

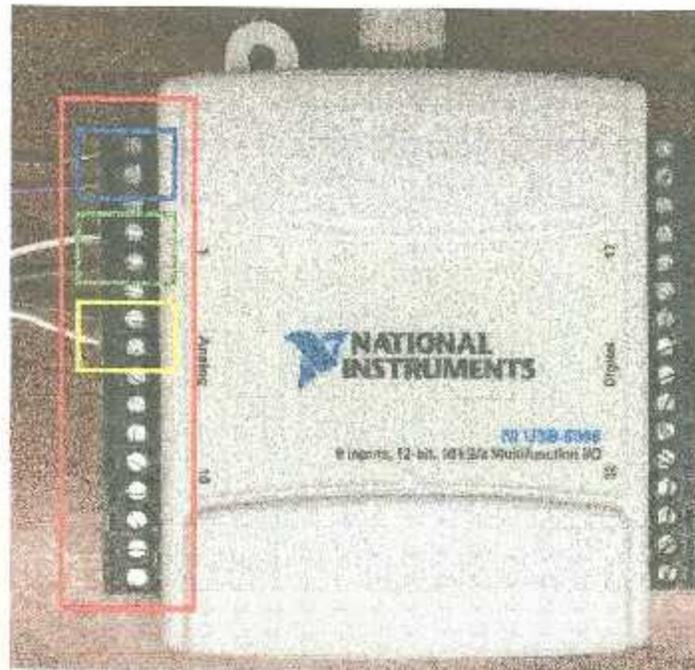


Figura 3-12 NI USB-6008.

En el recuadro rojo de la Figura 3-12 se pueden observar los canales analógicos de entrada y salida; en el recuadro azul entra una señal senoidal generada por un generador de funciones. Ésta simula la fase “A” del voltaje de un aerogenerador. En el recuadro verde se tiene otra señal senoidal, generada por un generador de funciones, que simula la fase “B” del voltaje de un aerogenerador; en el recuadro amarillo pasa lo mismo, sólo que ésta simula la fase “C” del voltaje de un aerogenerador.

El NI USB-6008 se conecta por el puerto USB a la computadora encargada de monitorcar y almacenar los datos, esta computadora es el servidor local del aerogenerador. Los datos son almacenados en la misma BD que se usa en la adquisición de datos con el puerto serie, sólo que la forma de adquirir y manejar los datos para almacenarlos es muy diferente.

3.2.2 Creación de un Task

Para que LabView adquiera los datos del dispositivo NI USB-6008 es necesario crear un Task; esto se hace por medio de MAX software desarrollado por NI y se distribuye junto con LabView, el cual, ayuda a configurar y probar el software y el hardware que se puede utilizar para el desarrollo de aplicaciones [28]. En la Figura 3-13, del lado izquierdo se observa el panel de configuración; aquí se encuentran los siguientes submenús:

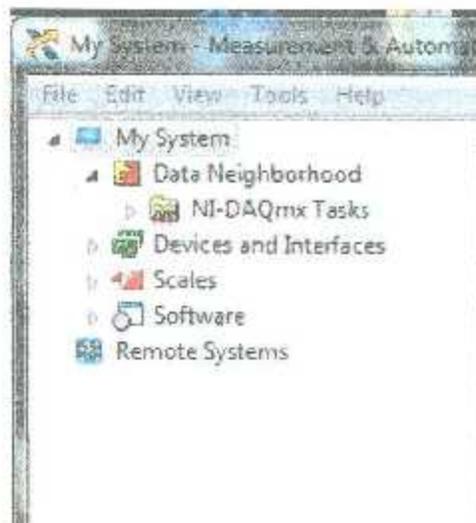


Figura 3-13 Panel de configuración.

- Data Neighborhood : Aquí se crean los canales y tareas para los dispositivos como el NI USB-6008.
- Devices and Interfaces : Aquí se configuran los dispositivos e interfaces instalados en la computadora, tales como tarjetas de adquisición de datos, puertos de comunicación serie, etc. También se pueden crear instrumentos virtuales; éstas son de ayuda cuando se desea probar una aplicación pero no se cuenta con el dispositivo físico.

- Scales : Aquí se crean y configuran fórmulas o tablas de escalado para las tareas, estas escalas usualmente son aplicadas a las muestras adquiridas.
- Software : Aquí se muestra y actualiza el software de NI instalado.

Para crear un nuevo Task, se da *click* derecho sobre el submenú *Data Neighborhood*, y se selecciona *Create New*, como se observa en el recuadro rojo de la Figura 3-14.

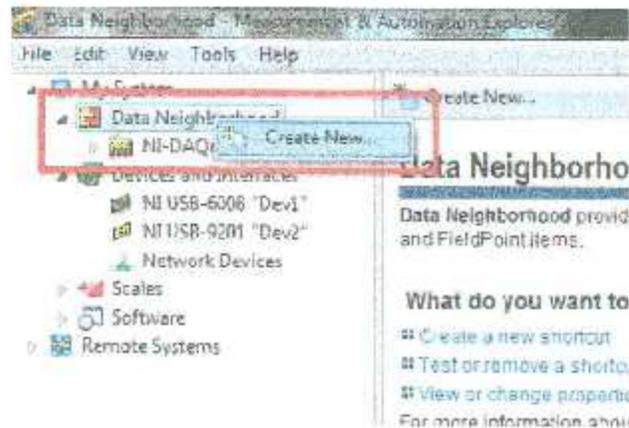


Figura 3-14 Crear nuevo Task para el dispositivo NI USB-6008.

Una vez elegido *Create New*, aparecerá una ventana con dos opciones, la primera es para crear el Task para un dispositivo físico y la segunda opción es para crear un Task de un dispositivo virtual. En el recuadro rojo de la Figura 3-15 se observa la primera opción para la creación del Task.

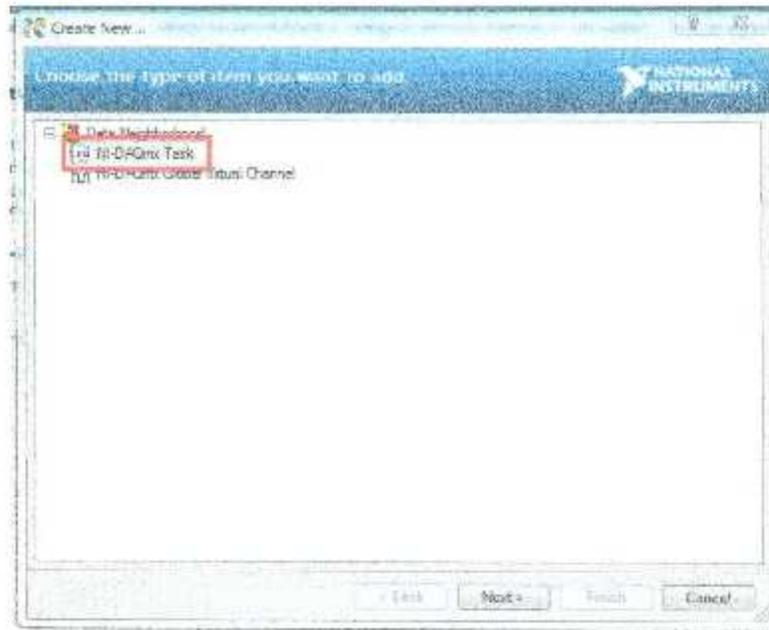


Figura 3-15 Selección del Task físico.

Una vez que se haya seleccionado esta opción se da *click* en el botón siguiente. Después de dar *click* aparecen más submenús; se muestra el submenú que dice *Acquire Signals*, después se muestra el submenú *Analog Input* y por último se selecciona *Voltage* y se da *click* en el botón siguiente. En el recuadro rojo de la Figura 3-16 se pueden apreciar los submenús desplegados para llegar a la opción de *Voltage*, ya que es la señal que se va a simular.

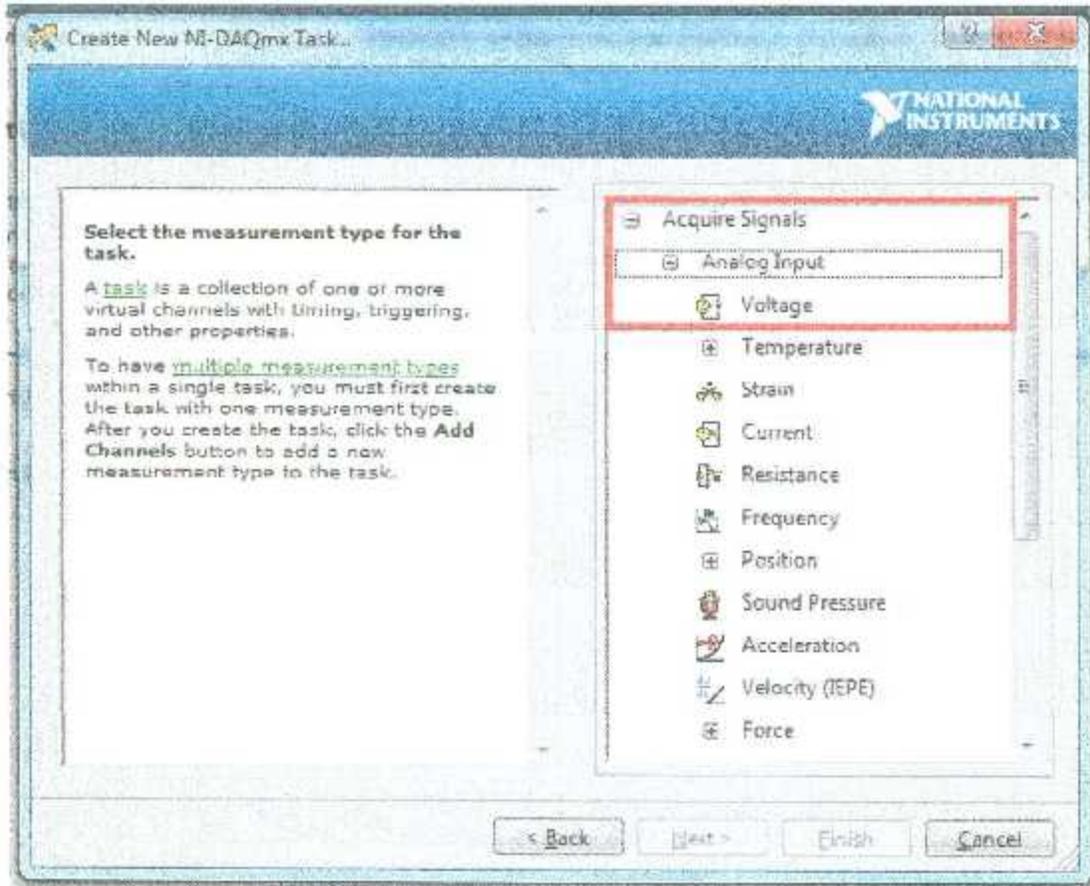


Figura 3-16 Tipo de señal de entrada del dispositivo NI USB-6008.

En la siguiente ventana, al igual que en la anterior, se encuentran submenús desplegables, pero en esta ocasión los submenús pertenecen a los dispositivos conectados a la computadora; se muestra el submenú *Dev 1*, que corresponde al módulo NI USB-6008; aquí se seleccionan los canales que se van a utilizar, en el caso de la aplicación desarrollada sólo se elegirán tres canales, *ai0*, *ai1* y *ai2* y se presiona el botón siguiente, como se muestra en el recuadro rojo de la Figura 3-17.

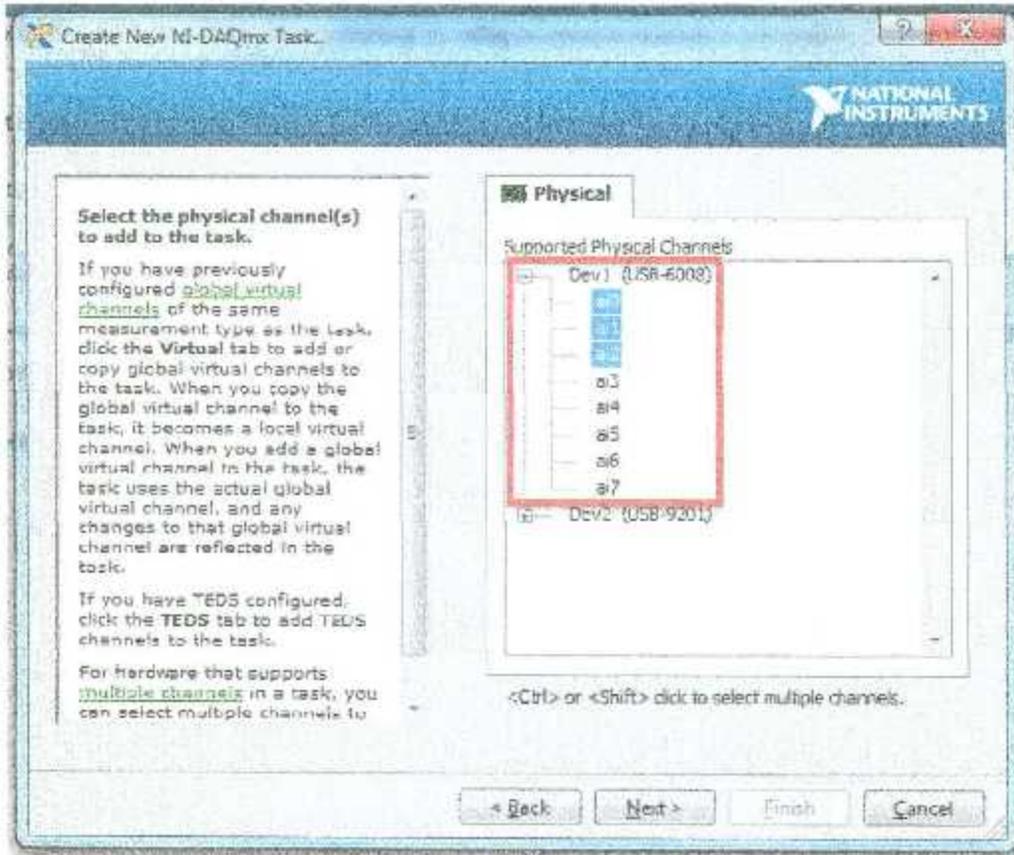


Figura 3-17 Selección de los canales del dispositivo NI USB-6008.

Por último, en la ventana sólo se define el nombre del Task, y se presiona finalizar para concluir con la creación del Task que servirá para la comunicación y adquisición de las señales del dispositivo NI USB-6008 con LabView. En el recuadro rojo de la Figura 3-18 se observa el nombre con que se definió al Task, el cual es *AGMV* (por sus siglas, Aerogenerador Monitoreo Voltaje). En la Figura 3-19 en el recuadro rojo se comprueba la creación del Task, de lado izquierdo en el submenú *NI-DAQmx Tasks*.

Al seleccionar el Task que se creó llevará a una ventana, la cual ayuda a configurar cada uno de los canales que se seleccionaron, por ejemplo, en el recuadro rojo de la Figura 3-20 se asignan los nombres a cada canal por Fase A, Fase B y Fase C, en el recuadro verde de lado superior izquierdo se puede apreciar una opción de guardar, para aplicar los cambios que se hayan hecho al Task, enseguida del recuadro verde se observa un recuadro amarillo en el que se encuentra un botón con el título de *RUN*; este botón sirve para probar las señales que entran en el dispositivo NI USB-6008, las cuales se muestran en un área para graficar, en el recuadro azul, se pueden ver las tres ondas senoidales generadas por los generadores de funciones.

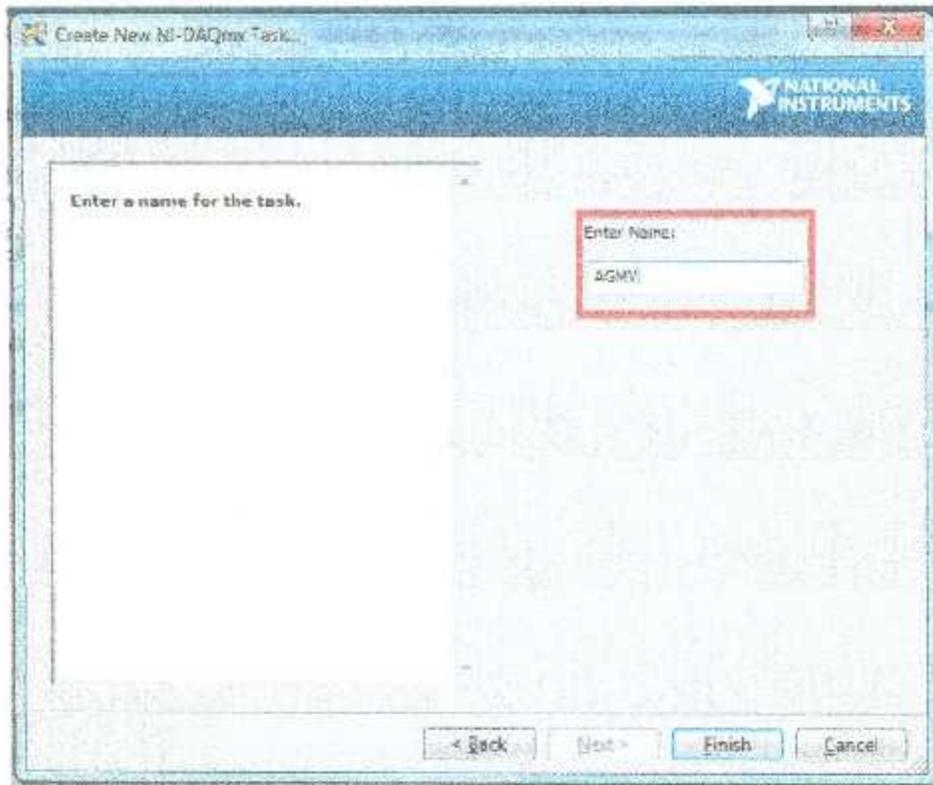


Figura 3-18 Definición del nombre para el Task.



Figura 3-19 Task AGMV creado.

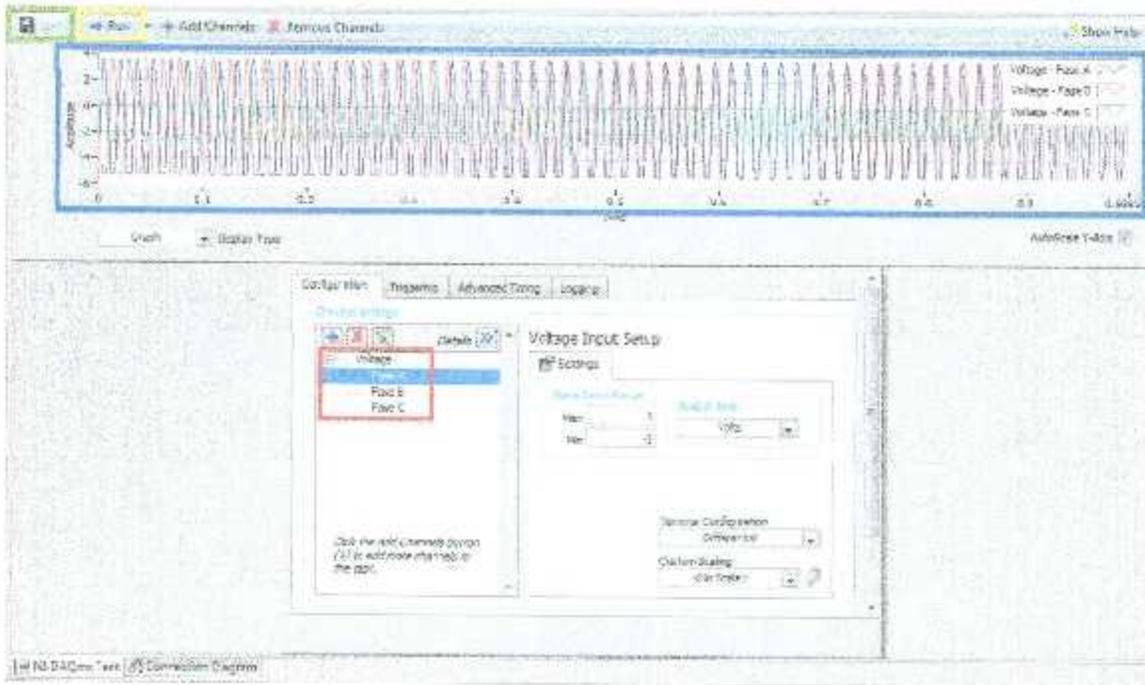


Figura 3-20 Área para configuración y visualización de los canales.

3.2.3 Adquisición de Datos

Para adquirir los datos se usa el VI *DAQmx Read*; en la Figura 3-21 se puede observar que recibe un Task para adquirir los datos del módulo DAQ, es aquí donde se indicará el Task que se creó anteriormente; también se tiene una línea de salida *data*, de esta línea es de donde se obtienen los datos del dispositivo NI USB-6008.

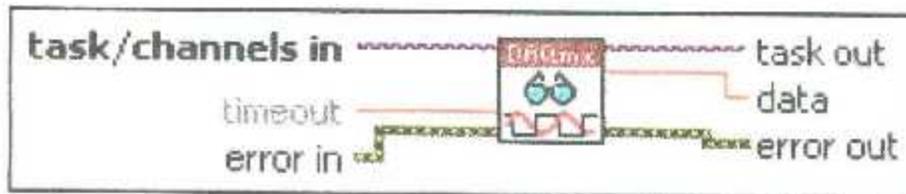


Figura 3-21 VI para adquirir los datos del dispositivo NI USB-6008.

Al igual que con el puerto seric, la forma en que se almacenan los datos en la BD es usando conexión ODBC y usando los VI de LabSQL; en la Figura 3-22 se observa el diagrama a bloques para obtener, visualizar y almacenar los datos.

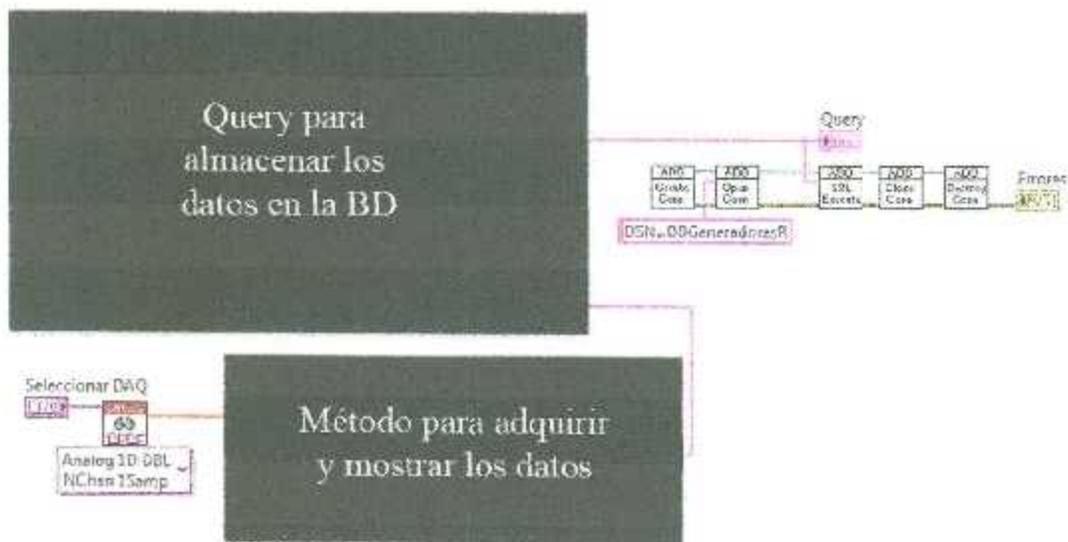


Figura 3-22 Bloques para la adquisición, visualización y almacenamiento de las señales.

Por otro lado, en la parte visual del programa sólo se agregó una sección nueva para adquirir los datos por medio de un DAQ. En el recuadro verde de la Figura 3-23a se tiene un *listbox*; aquí se asigna el Task al VI *DAQmx Read*. En el recuadro rojo de la Figura 3-23b se puede observar que en la lista se encuentra el Task AGMV creado anteriormente, por último, en el recuadro azul de la Figura 3-23c se observa el botón que inicia y detiene la adquisición de datos.

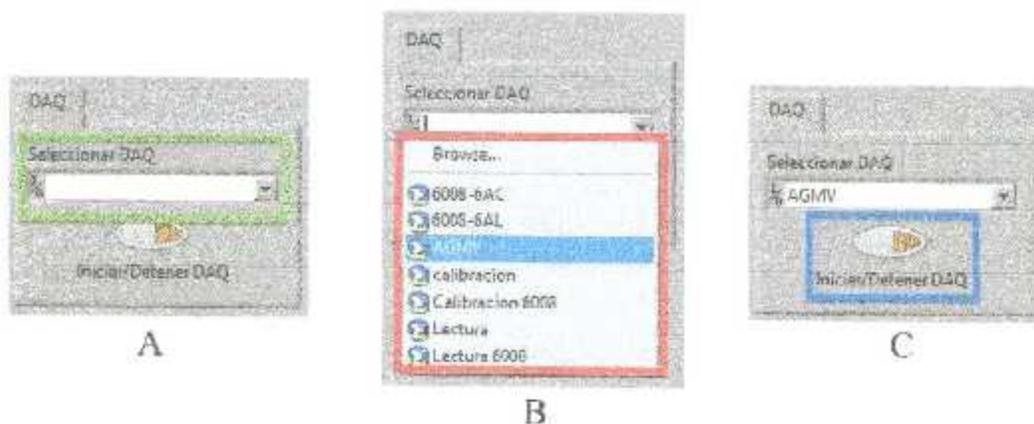


Figura 3-23 Forma visual para asignar el Task y adquirir los datos.

3.3 Uso de SMS para el estado del generador

El Servicio de Mensajería Corta (SMS, por sus siglas en inglés, *Short Message Service*), es un mecanismo que funciona con la red de teléfonos celulares. Una característica de los SMS es que cuando se envía un mensaje, éste se almacena en una BD (dependiendo la compañía que brinda el servicio) por si el receptor no se encuentra disponible poder enviarlo más tarde; estos mensajes pueden ser enviados entre teléfonos celulares dentro de una red GSM [29].

Una desventaja de los SMS es que tienen un tamaño máximo de 160 caracteres, es por eso que en sí el SMS no puede utilizarse para el monitoreo de las variables de un generador, sino para notificar el estado del aerogenerador cada cierto tiempo, incluso notificar de alguna falla del aerogenerador a ciertas personas cuyas decisiones puedan ser importantes.

Los SMS de prueba de estado del aerogenerador son transmitidos por un módulo GSM (GSM Click) fabricado por Mikroe. Este módulo se comunica directamente con un microcontrolador por el puerto serie, el mismo microcontrolador usado para el monitoreo de las señales del aerogenerador el PIC18F45K22.

El microcontrolador se comunica con el módulo GSM a través del puerto serie usando la librería UART, básicamente se envía una cadena de caracteres con la información a enviar en un SMS; esta cadena no debe de pasar los 160 caracteres, es por esta razón que la información enviada es específica.

3.3.1 Módulo GSM para el envío de información

El módulo *GSM Click* cuenta con módulo GL865-QUAD de la compañía Telit, la cual se encarga de proporcionar sistemas integrados para la comunicación celular e internet. En el recuadro rojo de la Figura 3-24a se puede observar el módulo GL865-QUAD.

En el recuadro azul de la Figura 3-24b se observa la entrada para una tarjeta mini-SIM (por sus siglas en inglés, *Subscriber Identity Module*), esta tarjeta SIM puede ser adquirida con cualquier compañía que brinde servicios de comunicación celular.

3.3.2 Comandos AT

Los comandos AT son un subconjunto de órdenes que permiten controlar y configurar módems. Estos comandos comienzan por el comando AT (del inglés, *Attention*) [30] de atención y todos estos comandos terminan con el comando "CR" (por sus siglas en inglés, *Carriage Return*) el cual se puede representar por el número hexadecimal "D" o decimal "13".

Estos comandos son enviados por el microcontrolador a través del puerto serie, por medio del módulo UART, sirven para inicializar el módulo, comprobar si el módulo se encuentra en funcionamiento y definir el tipo de mensaje, por ejemplo que el mensaje a enviar es de tipo texto.

La sintaxis de los comandos AT para el módulo *GSM Click*, son muy similares a los del estándar básico y a los comandos extendidos AT, para GSM, FAX y Hayes AT.

De los cuales se tienen dos tipos de comandos:

- Comandos con parámetros: este tipo de comandos almacenan valores en la memoria para ser usados posteriormente.
- Comandos de acción: este tipo de comandos sirven para ejecutar o probar el módulo.

La sintaxis sería de la siguiente forma:

$AT + \langle \text{comando} \rangle [\langle \text{parámetro} \rangle [\langle \text{parámetro} \rangle]] \langle CR \rangle$

El primer comando que se envía al módulo es para verificar si el módulo GL865-QUAD está funcionando correctamente, en la Figura 3-26 se muestra el código para verificar el módulo.

```
void ModuloListo()
{
    const char atc[] = "AT";
    while(1)
    {
        GM862_Envia(atc);
        Delay_ms(100);
        if (Get_respuesta() == GSM_OK)
            break;
    }
}
```

Figura 3-26 Función para comprobar que el módulo GL865-QUAD este funcionando correctamente.

Como se puede apreciar, en el recuadro verde de la Figura 3-26 se declara un arreglo de tipo *char*, el cual contiene el comando "AT" correspondiente a un comando de tipo acción, ya que se está verificando el módulo GL865-QUAD, el cual responde con una cadena de caracteres por el puerto serie, esta cadena de caracteres corresponde a la palabra "OK" si es correcto, en caso contrario se recibe la palabra "ERROR".

Por esta razón se activan las interrupciones del puerto serie en el microcontrolador, en pocas palabras, cada vez que el módulo GL865-QUAD envía datos por el puerto serie, manda una señal de interrupción, esta interrupción se ve reflejada en una función en el microcontrolador, la cual contiene lo necesario para leer los datos de entrada del puerto serie desde el módulo GL865-QUAD; por ejemplo, si llegase un mensaje de texto al módulo *GSM Click*, el módulo GL865-QUAD se encarga de enviar el mensaje por el puerto serie al microcontrolador, y es aquí cuando la interrupción manda llamar a la función para leer los datos del puerto serie. En la Figura 3-27 se muestra un comando con parámetros.

```
void BorrarMensajes()
{
    const char atc5[] = "AT+CMGD=1,4";
    while(1)
    {
        GM862_Envia(atc5);
        Delay_ms(50);
        if (Get_respuesta() == GSM_OK)
            break;

        Delay_ms(500);
    }
}
```

Figura 3-27 Función para borrar los mensajes de la buzón de entrada.

El comando que se observa en el recuadro verde de la Figura 3-27 sirve para eliminar los mensajes del buzón de entrada del módulo GL865-QUAD. El número 1 y 4 son los parámetros que recibe el comando AT.

El primer parámetro del comando *CMGD*, corresponde al índice del mensaje en memoria, el segundo parámetro contiene banderas predefinidas para borrar los mensajes, de las cuales se tiene que :

- 0 : Elimina el mensaje especificado por el índice.
- 1 : Elimina todos los mensajes almacenados en memoria, dejando todos los mensajes del teléfono celular intactos.

- 2 : Elimina todos los mensajes de la memoria leídos, dejando los mensajes no leídos en el celular.
- 3 : Elimina todos los mensajes de la memoria leídos, elimina los mensajes leídos del celular y deja los mensajes sin leer.
- 4 : Elimina todos los mensajes de la memoria.

El comando que se muestra en la Figura 3-28 ayuda a definir el tipo de SMS que enviará el módulo GL865-QUAD, de los cuales se tienen dos tipos:

- Mensajes de texto plano: estos mensajes están compuestos por una cadena de caracteres ASCII.
- Protocolo de Descripción de Unidad: (PDU, por sus siglas en inglés, Protocol Description Unit), es un formato estándar para los SMS, el mensaje es codificado, carácter por carácter, en su forma hexadecimal.

```
void MensajesTipoTXT()
{
    const char atc[] = "AT+CMGF=1";
    GM862_Envia(atc);
    EsperaRespuesta(GSM_OK);
}
```

Figura 3-28 Función para definir el tipo de mensaje.

Como se observa en el recuadro verde de la Figura 3-28, el comando *CMGF* recibe sólo un parámetro, que puede ser "0" para los mensajes tipo PDU y "1" para los mensajes tipo texto. En el trabajo desarrollado en esta tesis se trabaja con SMS de tipo texto para el estado del monitoreo del aerogenerador.

3.3.3 Formato de SMS

El formato para los SMS es un formato compacto con cierta información, ya que los SMS están limitados a 160 caracteres, como ya anteriormente se había mencionado. El fin de enviar información a base de SMS es por si en alguna situación la comunicación con el servidor remoto llegara a fallar y también porque la comunicación celular abarca más lugares. El formato de los SMS está dado por una función que al finalizar de dar el formato manda llamar a las funciones necesarias para enviar el mensaje al módulo GL865-QUAD; estos mensajes servirán para dos propósitos, mantener al usuario informado cada cierto tiempo de algunos parámetros del aerogenerador y como medio de alerta por si llegase a ocurrir algún problema en el aerogenerador. En la Figura 3-29 se muestra el diagrama para el formato y enviar los SMS.

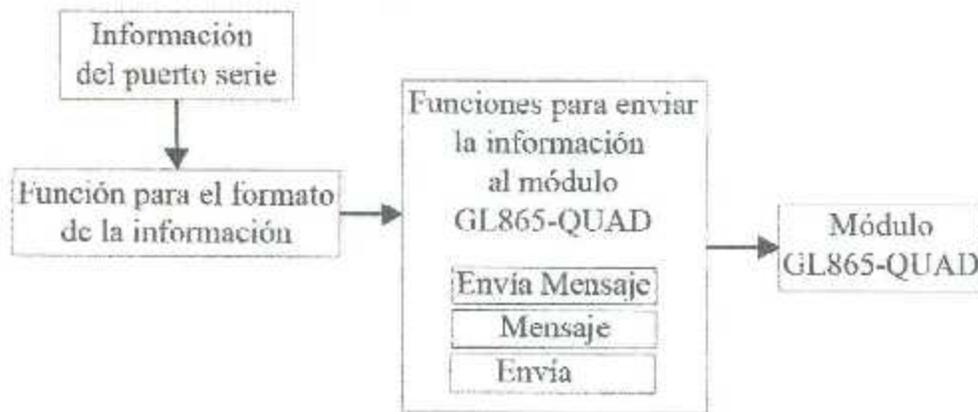


Figura 3-29 Diagrama de flujo para el formato de los SMS.

Las funciones que se observan en la Figura 3-29 sirven para mandar llamar el comando que se encarga de enviar el SMS. En la Figura 3-30, se muestra la función *EnviaMensaje()* que es llamada después de que se le da formato a la información, ésta a su vez manda llamar dos funciones más como, se observa en el recuadro verde. La función *Mensaje()*, tiene como propósito detener las interrupciones del puerto serie hasta que el SMS con formato se concatene a un arreglo de caracteres, ya que esta variable es la que se enviará al módulo GL865-QUAD; en el recuadro rojo de la Figura 3-30 se observa la variable que contiene el SMS con formato.

```

void EnviaMensaje()
{
    Mensaje(_mensaje);
    Enviar(_mensaje);
}

unsigned Mensaje(char* _mensaje)
{
    RC1IE_bit = 0;

    _mensaje[0] = '\0';
    strcat(_mensaje, msnInf);
    //strcat(_mensaje, "\r\n"); //(\CR + LF)

    RC1IE_bit = 1;

    return strlen(_mensaje);
}
  
```

Figura 3-30 Funciones para enviar el mensaje.

En la Figura 3-31 se observa la función encargada de mandar el SMS, en el recuadro verde se observa el comando necesario para advertir al módulo GL865-QUAD que se va a enviar un mensaje, en el recuadro rojo se observa la variable que contiene el número de

celular al que será enviado el SMS, después en el recuadro azul se encuentra la variable que contiene el SMS a enviar. Como se puede apreciar, el mensaje se deposita en la memoria del módulo GL865-QUAD, al final se espera una respuesta OK de que el mensaje ha sido enviado o ERROR si sucede lo contrario.

```
void Enviar(char* _mensaje)
{
    char atc[] = "AT+CMGS=\"";

    char atcAux[33];
    atcAux[0] = 0;

    strcat(atcAux, atc);
    strcat(atcAux, _numeroCelular);
    strcat(atcAux, "\"");

    GM862_Send_Ram(atcAux);

    GM862_Send_Ram(_mensaje); // Enviamos contenido del mensaje
    UART_Wr_Ptr(0x1A);
    UART_Wr_Ptr(0x0D); // CR
    EsperaRespuesta(GSM_OK); // Espera Respuesta OK
}
```

Figura 3-31 función para enviar el SMS.

Capítulo 4. Métodos para almacenar los

Datos del Aerogenerador

Los datos son los testimonios de la información que se tiene acerca de un hecho, son el reflejo de las condiciones en que se encuentra una situación dada [31], por ejemplo, el registro de nacimiento de una persona.

Existe una gran variedad de tipos de datos y por lo general cada uno tiene sus propias características para ser procesados o registrados. Los tipos de datos que son de interés en este caso, son aquellos que son capaces de ser procesados por una computadora, con el propósito de lograr resultados (o nuevos datos) que proporcionen una información útil para lograr algún objetivo. Un ejemplo de esto es el monitoreo de un aerogenerador.

La importancia que ha logrado alcanzar el procesamiento de datos se debe principalmente:

- Al gran volumen de datos que puede contener una computadora.
- A la gran rapidez que se tratan los datos.
- Al alto grado de confiabilidad en los resultados.
- Al respaldo externo de los datos procesados.
- Al orden que se le puede dar a los datos.

Existen dos tipos de formas para almacenar la información, como caracteres de texto o información binaria, por lo cual se tienen dos tipos de archivo [32].

- Archivos de texto.
- Archivos binarios.

En cualquiera de los dos tipos de archivos la información sigue estando ordenada en cadenas de bytes terminadas con el carácter *Fin de Archivo* (EOF, por sus siglas en inglés, *End Of File*). Sólo que la diferencia radica en cómo los programas que leen el archivo interpretan la información almacenada.

4.1 Archivos de Texto

Un archivo de texto es un conjunto de bytes representados por caracteres [32], lo que quiere decir que todo archivo de texto puede ser visible por cualquier editor de texto.

4.1.1 Archivos XML

Lenguaje de Marcas Extensible (XML, por sus siglas en inglés, *eXtensible Markup Language*), es un lenguaje de etiquetas creado para almacenar datos de forma legible, XML a diferencia de HTML no se ha creado para ser interpretado sólo para internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas: éste se puede usar en bases de datos, editores de texto, hojas de cálculo, creación de las interfaces de usuario en Android, etc.

Los archivos XML tienen una sintaxis simple a base de etiquetas, la cuales a diferencia de HTML no vienen predefinidas, el usuario es libre de crear sus propias etiquetas, ya que éstas son las que contienen la información que se almacena en el archivo, lo que conlleva a que las etiquetas sean definidas de acuerdo a la información que se almacena, lo que hace al archivo auto descriptivo [33], como las etiquetas pueden contener otras etiquetas dentro de ellas, esto hace que los archivos XML tengan una estructura de árbol; en la Figura 4-1 se muestra cómo se compone un archivo, si se separan desde la etiqueta raíz hasta las ramas, las demás etiquetas.

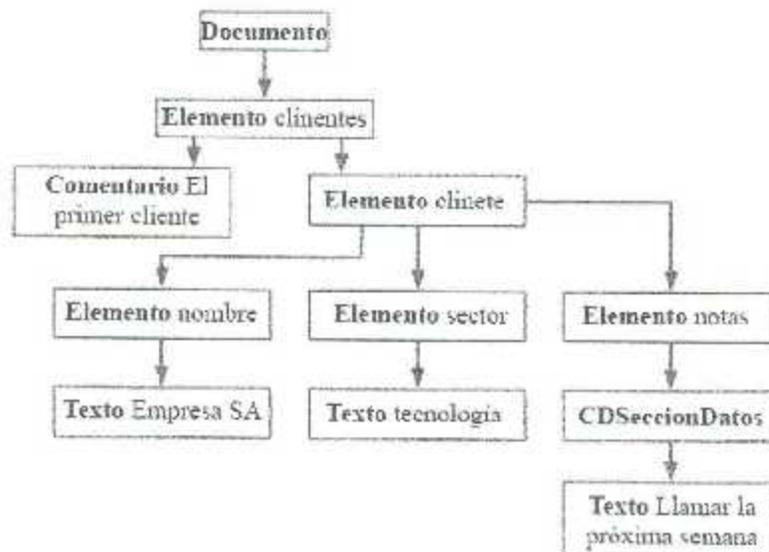


Figura 4-1 Diagrama de un archivo XML.

En la Figura 4-2 se muestra el ejemplo de un archivo para almacenar las variables de un aerogenerador.

```

<?xml version = "1.0" encoding = "UTF-8"?>
<VelocidadViento>
  <fecha>
    2014-06-19 16:11:23
  </fecha>
  <!-- Velocidad máxima y mínima de las últimas 24 horas -->
  <velMax>
    12.9
  </velMax>
  <velMin>
    8.2
  </velMin>
  <datos name="velocidad">
    12.5,
    12.5,
    8.5,
    11.5,
    10.5,
    10.2,
    8.2
  </datos>
</VelocidadViento>

```

Figura 4-2 Sintaxis de Archivo XML.

Para cada variable que se obtiene del aerogenerador se crea una etiqueta para identificar qué tipo de dato es el que se almacenó. En este caso, como se observa en la Figura 4-2, el tipo de variables: velocidad del viento (el archivo contiene pocos datos para poder observar las etiquetas), velocidad máxima, velocidad mínima, fecha y hora.

Por otra parte también cada una de las variables del aerogenerador tiene su propio archivo, es decir, la corriente, el voltaje, la potencia activa y reactiva se almacenan en archivos diferentes, y cada archivo tiene como nombre la fecha en que se tomaron los datos, por ejemplo, los datos almacenados del día diecinueve de junio del 2014, tiene como nombre 19-6-2014.xml, y este almacena todos los datos recibidos ese día.

4.1.2 JSON

Notación de Objetos para JavaScript (JSON, del inglés, *JavaScript Object Notation*), es un formato utilizado para serializar datos. Este suele usarse en sitios web como un formato para el intercambio de datos [34]. Las ventajas que ofrece JSON sobre XML es que puede soportar tipos básicos como:

- Numéricos: enteros, reales.
- Cadenas: doble comilla o simple comilla.
- Booleanos: falso y verdadero.
- Objetos: se refiere a una colección de datos.
- Nulo: se refiere a un objeto o valor nulo.

También, los lenguajes que soportan JSON son cada vez son más, a continuación se mencionan algunos de estos lenguajes.

- LabView.
- ActionScript.
- C.
- C++.
- C#.
- Java.
- JavaScript.
- Objective-C.

Como ejemplo se tomará el archivo de la Figura 4-2 y se pasará a la notación JSON. En la Figura 4-3 se muestra el archivo JSON para el monitoreo de la velocidad del viento.

```
{
  "VelocidadViento":
  {
    "fecha" : "2014-06-19 16:11:23",
    /* Velocidad maxima y minima de las ultimas 24 horas */
    "velMax" : 12.9,
    "velMin" : 8.2,
    "datos" :
    [
      12.5,
      12.5,
      8.5,
      11.5,
      10.5,
      10.2,
      8.2
    ]
  }
}
```

Figura 4-3 Archivo de monitoreo en JSON.

Como se puede apreciar entre la Figura 4-2 y Figura 4-3, el cambio de formato es muy notable, en JSON la fecha está representada como un *string*, las velocidad máxima y mínima alcanzadas por el aerogenerador están representadas como tipo numérico, y los datos de monitoreo están representados como un arreglo de tipo numérico.

Vistas las diferencias entre un archivo de tipo XML y tipo JSON, los dos fueron almacenados en un servidor externo para después desplegar la información almacenada en una pagina web, los tipos de archivo fueron subidos al servidor usando FTP, y a la hora de acceder a los datos es cuando se apreció la diferencia entre los tipos de archivo.

```

function LeerVelocidad()
{
    var _velocidad = [];
    $.ajax({
        type: "GET",
        url: "../Monitoreo/Viento/Velocidad/19-6-2014.xml",
        dataType: "xml",
        success: function(xml)
        {
            $($.xml).find('VelocidadViento').each(function()
            {
                var _vXML = $($.xml).find('datos').text();
                _velocidad = _vXML.split(',');
            });

            var datos = [];
            for (var i = 0; i < _velocidad.length; i++)
                datos.push(parseFloat(_velocidad[i]));

            GraficaVelocidad(datos);
        }
    });
};

```

Figura 4-4 Lectura de los archivos XML.

Como se observa en el recuadro rojo de la Figura 4-4, para poder obtener los datos XML es necesario buscar la etiqueta *VelocidadViento*, y después que se haya encontrado la etiqueta se busca la etiqueta *datos*, ya que es la etiqueta que contiene todos los datos de monitoreo de la velocidad del viento. La forma en la que se accede a la información de este modo es porque como se mencionó anteriormente, los archivos XML forman un árbol, de acuerdo a las etiquetas que se definen.

Una vez que se encuentra la etiqueta que contiene los datos, se separan esos datos con el método *split()*; este método lo único que hace es separar una cadena de caracteres en base a un delimitador, en este caso el delimitador es la coma, como se observa en la Figura 4-4. Una vez que los datos han sido separados se convierten en tipo flotante, uno por uno y se almacenan en un arreglo (ya que se encuentran en formato string) para después ser mostrados en un navegador web.

```

function LeerVelocidad()
{
    $.ajax({
        type: "GET",
        url: "../Monitoreo/Viento/Velocidad/19-6-2014.json",
        dataType: "text",
        success: function(json)
        {
            var _velocidadJSON = JSON.parse(json);

            GraficaVelocidad(_velocidadJSON.datos);
        }
    });
};

```

Figura 4-5 Lectura para los archivos JSON.

Por otra parte, como se muestra en la Figura 4-5, para acceder a los datos de un archivo JSON primero se tiene que de-serializar el archivo y esto se hace con el método *parse()*, una vez que se obtiene el contenido del archivo, sólo basta con acceder a la etiqueta *datos*; ésta por default devolverá un arreglo flotante con todos los datos, sin tener que utilizar el método *split()* como en el archivo XML y mucho menos tener que convertir cada valor a tipo flotante, ya que por defecto en el archivo JSON ya viene definido de tipo numérico.

Una desventaja que se tiene con este tipo de almacenamiento de datos es que cada vez que se suben los archivos al servidor via FTP, se tiene que crear una carpeta para cada una de las variables monitoreadas del aerogenerador y por eso es que se crea una carpeta en el servidor con el nombre de *Monitoreo*, la cual contiene las carpetas de cada una de las variables que se obtienen del generador. Esto se puede apreciar en la Figura 4-6. Otra desventaja es que al momento de subir los archivos, esto se puede demorar por la gran cantidad de datos almacenada en ellos.

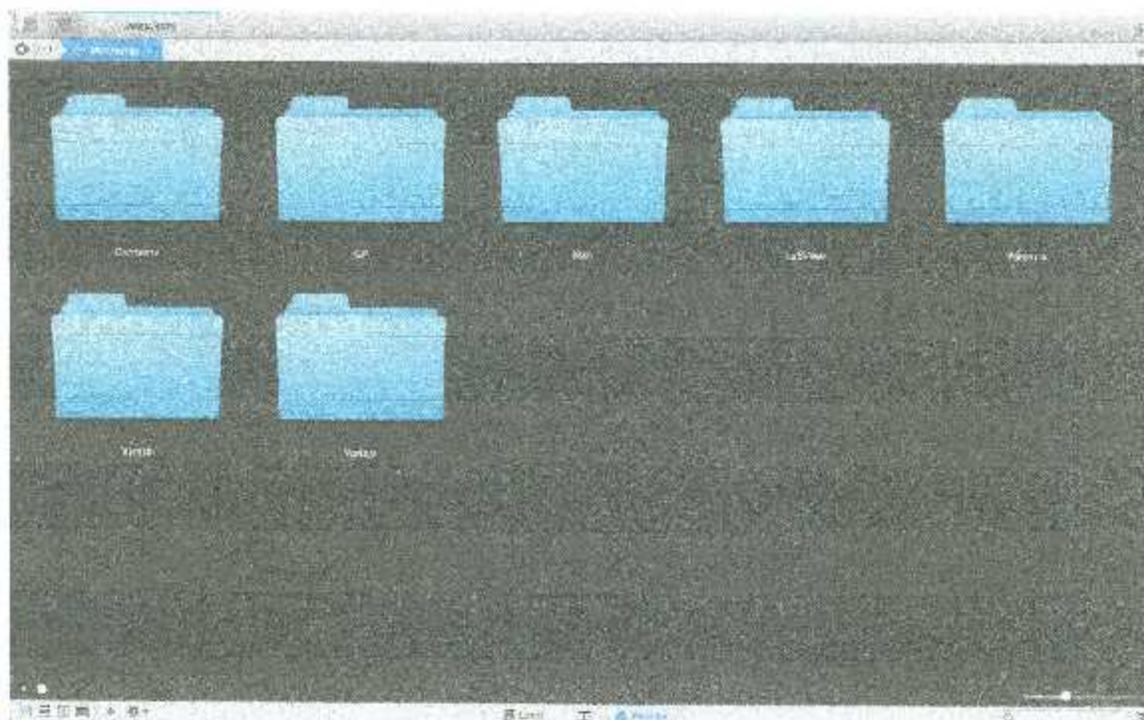


Figura 4-6 Carpetas donde se almacenan los archivos de monitoreo del aerogenerador.

4.2 Base de datos

Una base de datos es un contenedor multifuncional donde se puede almacenar y gestionar información en tablas [35].

A finales de los años 60's, Edgar Frank Codd definió las bases del *Modelo Relacional*, de esta forma es como los datos se estructuran de forma lógica para formar relaciones (tablas). El objetivo fundamental de este modelo, es mantener una independencia estructural lógica al modo de almacenamiento de la información [36]. El modelo relacional es el fundamento de las BD modernas [37].

Las tablas o relaciones son la estructura básica del modelo relacional [36] [37], en pocas palabras, los datos se representan en forma de tablas cuyo contenido varía con el tiempo. Cada columna de la tabla almacena información sobre una propiedad determinada de la tabla, a estas columnas por lo regular se les conoce como *atributos* y estos pueden aparecer en la relación en cualquier orden [38]. Por otra parte, a cada una de las filas de la tabla se les conoce como *tupla*; en la Figura 4-7 se puede observar una tabla que contiene información de una BD.

		Columnas (Atributos)					
Filas (Tuplas)		id	Nombre	NumCell	Potencia	Latitud	Longitud
6		6	Ada	8711580979	50	25.5329	-103.436
7		7	Alan	8718878040	40	-31.9	115.86

Figura 4-7 Tabla con información almacenada.

Cada tabla de la BD está formada por dos partes:

- **Cabecera:** Es un conjunto de partes *atributo:dominio*, donde cada atributo pertenece a un único dominio subyacente y todos los atributos son distintos; en pocas palabras, no hay dos atributos que tengan el mismo nombre.
- **Cuerpo:** Es el conjunto variable de tuplas, donde cada tupla es un conjunto de partes *atributo:valor*, donde cada valor varía con el tiempo

Como se mencionó anteriormente, una BD almacena información en forma de tablas relacionales, pero para crear la BD y almacenar la información en ella, se necesita de un Sistema Gestor de Bases de Datos (DBMS, por sus siglas en inglés, *Data Base Management System*), ya que estos DBMS nos permiten ciertas funciones como:

- **Definición:** Esta función permite a los diseñadores de BD especificar los elementos que la integran, la estructura y las relaciones que existen entre las tablas.
- **Manipulación:** Esta función permite obtener información o actualizarla.

- **Control:** Esta función facilita administrar usuarios, como por ejemplo, brindar funciones de servicios tales como, privilegios a usuarios, copias de seguridad, protección a accesos no autorizados, etc.

Cada vez que se desea información de la BD, es necesario hacer una consulta a las tablas; una consulta es la forma de acceder a la información desde el DBMS; cada vez que se necesite borrar tablas, eliminar elementos, modificar los datos almacenados o insertar nuevos elementos, se realiza una actualización.

Entre los DBMS existentes se encuentran los siguientes:

- **SQL Server:** Es el DBMS de Microsoft utilizado en Windows y Windows Server.
- **MySQL Workbench:** Es el DBMS de Oracle para administrar las BD's en MySQL, este software se encuentra disponible para Windows, Mac OS X y Linux.
- **Informix:** Es el DBMS de IBM, el software está disponible para Unix, Mac OS X, Windows y Linux.

Para poder tener una comunicación entre el DBMS y la BD hace falta un lenguaje intermedio, al cual se le conoce como Lenguaje Estructurado de Consultas (SQL, por sus siglas en inglés, *Structured Query Language*). Por medio de SQL se pueden manipular los datos almacenados en la BD, por ejemplo, crear consultas para obtener la información contenida en las tablas de la BD [39], actualizar la BD: insertar información, borrar o modificar la información. SQL no es un lenguaje de programación, ya que a diferencia de los lenguajes de programación, éste no produce un programa ejecutable entendible para la computadora, sólo es interpretado por un DBMS.

Para el desarrollo de esta tesis se utilizó *MySQL Workbench* de Oracle, ya que las BD creadas están hechas con MySQL y LabView tiene los VT's para una comunicación con MySQL.

4.2.1 Base de Datos en MySQL

La BD para almacenar los distintos aerogeneradores y los datos de monitoreo de cada aerogenerador ha sido creada en MySQL, como anteriormente se mencionó; éste ofrece el beneficio de ser software libre al igual que *MySQL Workbench*, para mantener administrada la BD.

La BD se crea tanto en la computadora de monitoreo como en el servidor externo, con la diferencia de que en la máquina donde se monitorea el servidor sólo se almacena un solo aerogenerador, y en la máquina servidor se almacenan todos los aerogeneradores que se conectan a éste.

Para crear las tablas de la BD se crea un esquema de Entidad Relación Mejorado (EER, por sus siglas en inglés, *Enhanced Entity-Relationship*), el cual es una mejora del

modelo Entidad Relación (ER, por sus siglas en inglés, *Entity-Relationship*), el cual es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información, así como sus interrelaciones y propiedades. No se profundizará en el tema de EER, ya que es un tema bastante amplio y no es tema de desarrollo de esta tesis, sin embargo, si se describirán las partes que incorporan al modelo EER para crear la BD. En la Figura 4-8 se muestra el esquema EER de la base de datos que almacena tanto los generadores como las variables de monitoreo para cada generador almacenado.



Figura 4-8 Esquema EER para la creación de la base de datos.

La BD consta de tres tablas:

- Generadores: En esta tabla se almacenan los generadores y sus atributos, tales como el nombre del generador, la potencia, la ubicación y la altura.
- VariablesG: En esta tabla se almacena las variables de monitoreo de cada generador.
- Datos24Horas: En esta tabla se almacenan los valores relevantes monitoreados en un día, así como la velocidad máxima y mínima del viento, el valor de los kWh del día y la velocidad máxima y mínima del generador.

Como se observa, la tabla *Generadores* se relaciona con las otras dos tablas, la forma en que lo hace es por medio de una *Llave Primaria* (PK, por sus siglas en inglés, *Primary Key*); una PK sirve como identificador único de una entidad, es el identificador único para cada aerogenerador almacenado. La forma en la que se identifican los datos de monitoreo del aerogenerador es que existe una *Llave Foránea* (FK, por sus siglas en inglés, *Foreign Key*) como atributo en las tablas *VariablesG* y *Datos24Horas*, cuyo valor debe corresponder a la PK de la tabla *Generadores*.

<u>id</u>	Nombre	NumCell	Potencia	Latitud	Longitud
6	Ada	8711580979	50	25.5329	-103.436
7	Alan	8718878040	40	-31.9	115.86

Figura 4-9 Tabla Generadores.

En la Figura 4-9 se puede observar el atributo *id*, este atributo es la PK de la tabla *Generadores*; como mencionó anteriormente, ésta ayuda a relacionar las dos tablas con ella. En la Figura 4-10 se muestran algunos valores de la tabla *VariablesG*.

<u>idG</u>	Fecha	Tiempo	Va	Vb	Vc	Ca
6	2014-10-13	13:46:48	0	0	0	0
7	2014-10-13	13:46:49	3.68125	3.68125	3.68125	3.68125
7	2014-10-13	13:46:50	6.84547	6.84547	6.84547	6.84547
7	2014-10-13	13:46:50	9.04827	9.04827	9.04827	9.04827
7	2014-10-13	13:46:50	9.98026	9.98026	9.98026	9.98026
7	2014-10-13	13:46:51	9.51056	9.51056	9.51056	9.51056
7	2014-10-13	13:46:51	7.70514	7.70514	7.70514	7.70514
7	2014-10-13	13:46:52	4.81754	4.81754	4.81754	4.81754
6	2014-10-13	13:46:52	1.25334	1.25334	1.25334	1.25334
6	2014-10-13	13:46:52	-2.48689	-2.48689	-2.48689	-2.48689
6	2014-10-13	13:46:53	-5.87784	-5.87784	-5.87784	-5.87784
6	2014-10-13	13:46:53	-8.44327	-8.44327	-8.44327	-8.44327
6	2014-10-13	13:46:54	-9.82287	-9.82287	-9.82287	-9.82287
6	2014-10-13	13:46:54	-9.82288	-9.82288	-9.82288	-9.82288
6	2014-10-13	13:46:54	-8.44329	-8.44329	-8.44329	-8.44329
6	2014-10-13	13:46:55	-5.87788	-5.87788	-5.87788	-5.87788

Figura 4-10 Tabla VariablesG.

Como se observó en la Figura 4-10, se tiene un atributo *idG*, el cual es la FK de la tabla *VariablesG* y hace referencia a la PK de la tabla *Generadores*. Esto quiere decir que las tuplas con los *idG* son los valores de monitoreo de cada aerogenerador; por ejemplo, en el cuadro rojo de la Figura 4-10 se observa que esas tuplas son los valores del aerogenerador Alan, cuyo *id* es el siete, y los valores del aerogenerador Ada, son aquellos que se encuentran en el recuadro azul.

Ahora la relación que se observa en el esquema EER de la Figura 4-8 es de uno a muchos, como se muestra en la Figura 4-11, en pocas palabras la implementación de una PK y una FK en las tablas ayuda a relacionar que un aerogenerador tendrá muchos valores.



Figura 4-11 Representación de uno a muchos.

Una vez definido el esquema EER que define las tablas de la BD, se pasa a implementar SQL para crear la BD. En la Figura 4-12 se observa el editor *MySQLWorkbench* (nuestro DBMS),

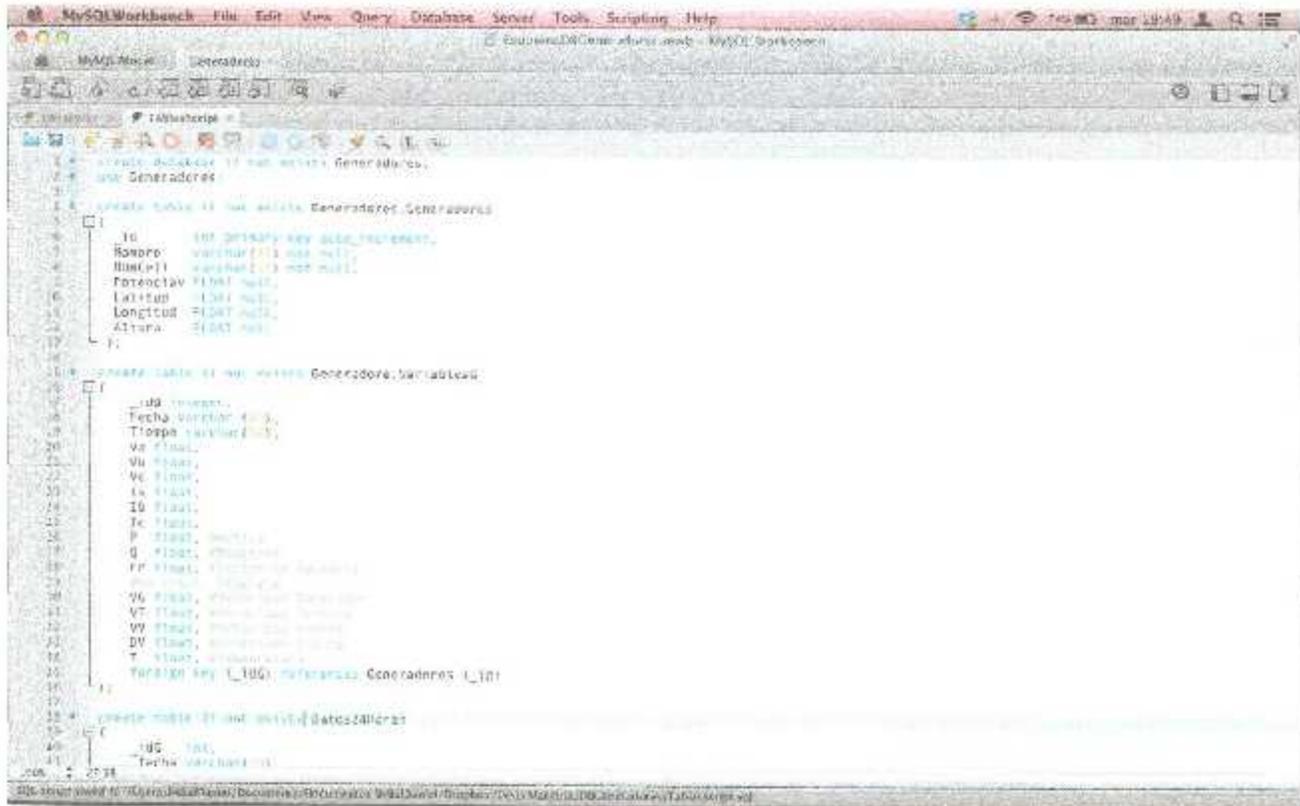


Figura 4-12 MySQLWorkbench.

Como se puede observar en la Figura 4-12, la forma de crear una BD es relativamente simple. Una vez que se introdujeron las instrucciones para crear la BD y las tablas que contendrán la información, se ejecuta el Script, como se muestra en la Figura 4-13.



Figura 4-13 Menú de Herramientas para ejecutar los scripts en SQL.

En el recuadro verde de la Figura 4-13 se muestra el ícono en forma de rayo para ejecutar el script que crea la BD.

A simple vista se observa que una BD es una forma mucho más eficiente para almacenar información que la de generar archivos de texto para almacenarla. La información se mantiene más organizada y no es necesario primero guardar el archivo localmente y después subirlo al servidor; la información se almacena directamente en el servidor externo y en el local.

4.2.2 Creación de Usuario Para la BD Generadores

Crear usuario es de suma importancia a la hora de administrar una BD ya que esto ayuda a controlar los accesos a la BD, y brinda roles en la misma; en la BD del servidor remoto se crea un usuario para cada generador. En la Figura 4-14 se muestra el menú a seleccionar para crear usuarios.

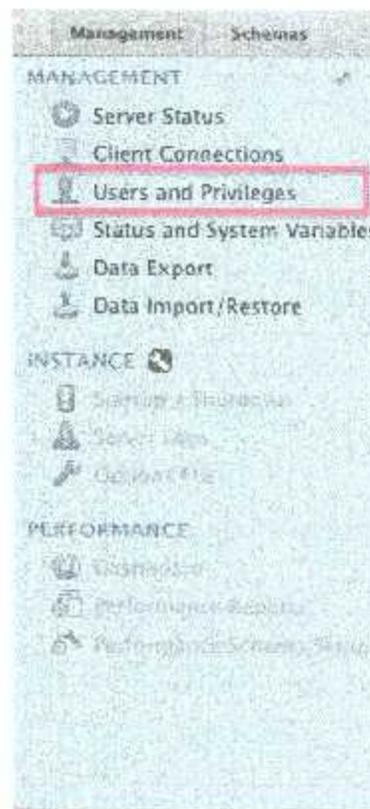


Figura 4-14 Menú de MySQL Workbench para administrar la BD.

Como se muestra en la Figura 4-14, se tiene la opción para administrar los usuarios de las BD, como se observa en el recuadro rojo. Una vez que se seleccionan los usuarios y privilegios, en la parte inferior se puede apreciar un botón para agregar un usuario, esto se puede apreciar en la Figura 4-15, en el recuadro rojo.



Figura 4-15 Agregar usuario.

Después de presionar el botón de agregar usuario aparecerá un formulario donde se pueden definir los parámetros del mismo a agregar. Como se observa en el recuadro rojo de la Figura 4-16, se tienen campos como el nombre de usuario, el tipo de autenticación, los límites del host y la contraseña; antes de agregar el usuario hay que desplazarse a la pestaña administrador de roles.

Details for account newuser@%

Figura 4-16 Campos para agregar un usuario nuevo.

En la Figura 4-17 se observan los roles en el que el usuario tendrá derechos a acceder en la BD.

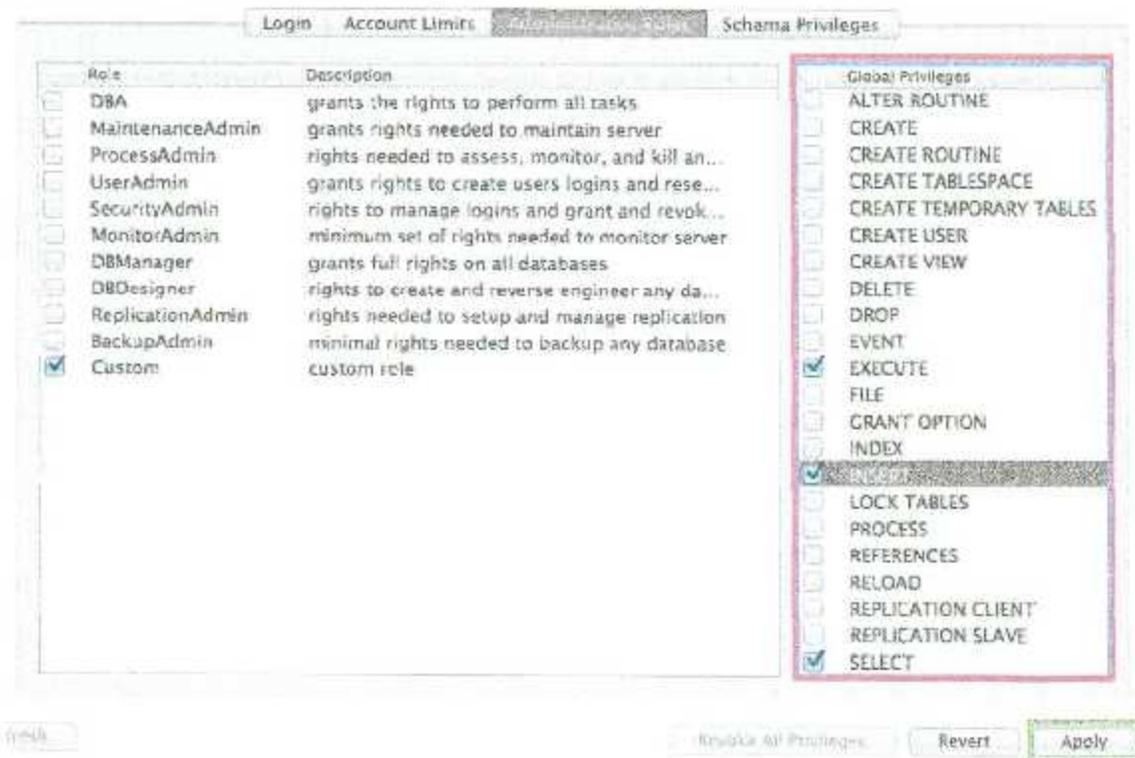


Figura 4-17 Administrar roles para usuarios en la BD.

En la Figura 4-17, en el recuadro rojo se seleccionan sólo las opciones de insertar, ejecutar y seleccionar (consulta), porque el usuario que se está creando es para el servidor remoto, así que lo único que se desea que haga en la BD es insertar datos, seleccionar los datos y ejecutar; con ejecutar se hace referencia a ejecutar Procedimientos Almacenados (SP, por sus siglas en inglés, *Store Procedures*). Un SP es una forma en que SQL hace más eficiente la forma par acceder a la información.

Una vez hecho todo esto se agrega el usuario presionando el botón *Apply* que se observa en el recuadro verde de la Figura 4-17.

4.2.3 ODBC para la comunicación con LabView y MySQL

ODBC es un estándar de acceso a las BD. ODBC sirve como coordinador de las transacciones entre el cliente, en este caso LabView, y la BD en *MySQL*. El cliente realizará llamadas a la BD por medio del ODBC en el equipo servidor, el controlador ODBC traduce esas llamadas en el formato requerido por la BD. En pocas palabras ODBC actúa como el intermediario entre el cliente y la base de datos [40].

Una conexión ODBC está definida por un Nombre de Origen de Datos (DSN, por sus siglas en inglés, *Data Source Name*). Para crear un ODBC es necesario ir a panel de

control de Windows, después a herramientas administrativas, estando ahí, se seleccionan Orígenes de datos (ODBC). En la Figura 4-18 se muestra la ventana para crear un DSN.

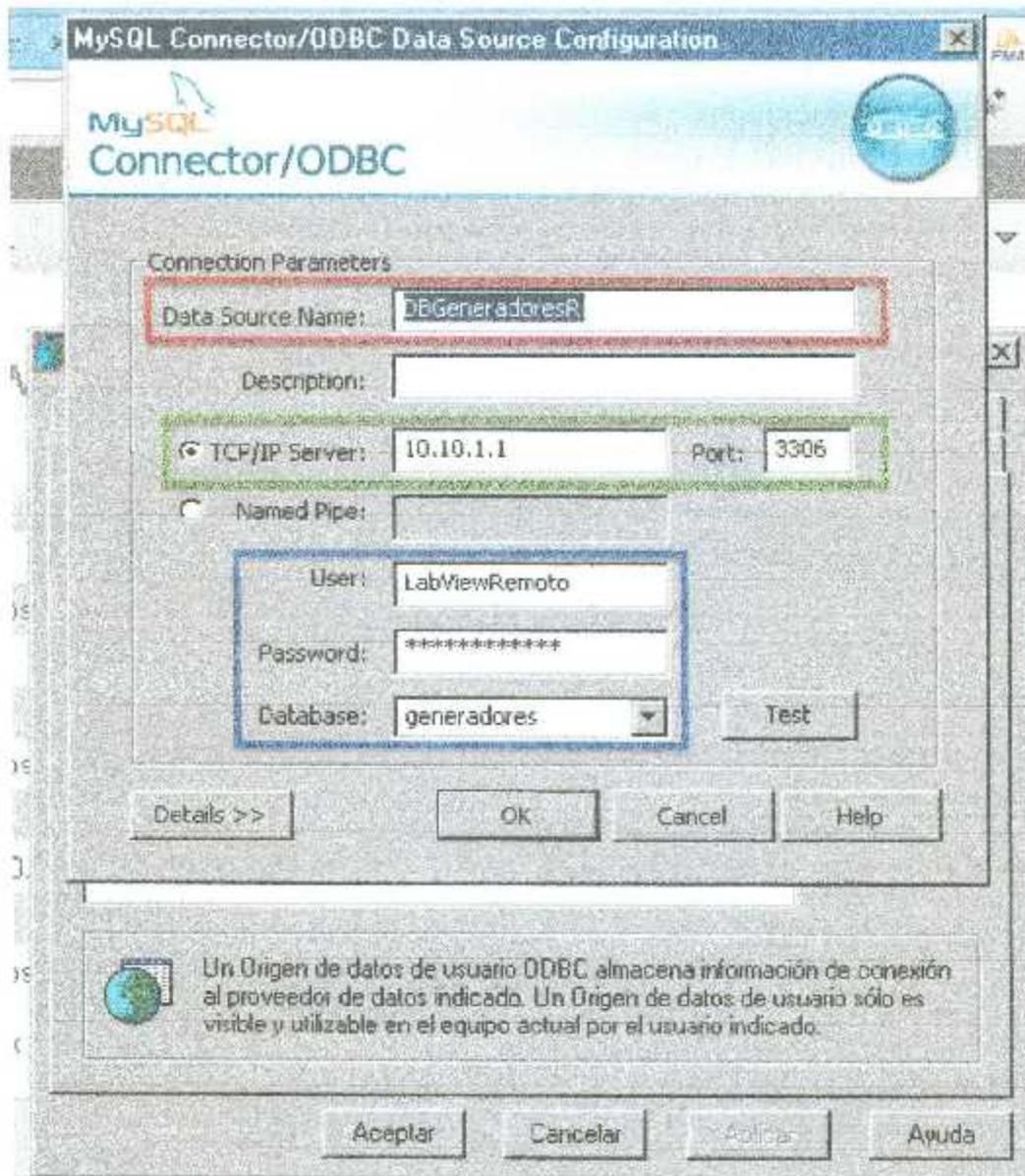


Figura 4-18 para almacenar en la BD remota.

Como se observa en la Figura 4-18, en el recuadro rojo, se introduce el DSN, en el recuadro verde se introduce la dirección IP o el nombre del dominio del servidor remoto, en este caso la dirección del servidor remoto es 10.10.1.1; en la Figura 4-19 se muestra otro servidor remoto a base de un Sistema de Nombres de Dominio (DNS, por sus siglas en inglés, *Domain Name System*), como se muestra en el recuadro verde. El puerto por defecto que usa *MySQL* es el 3306, y por último, en el recuadro azul se introduce el usuario y contraseña creados para almacenar la información en la BD. El usuario que se

utiliza en la Figura 4-18 es un usuario agregado de la misma forma, previamente descrita, como se agregó con *MySQL Workbench*, con los mismos roles, sólo que con diferente nombre.

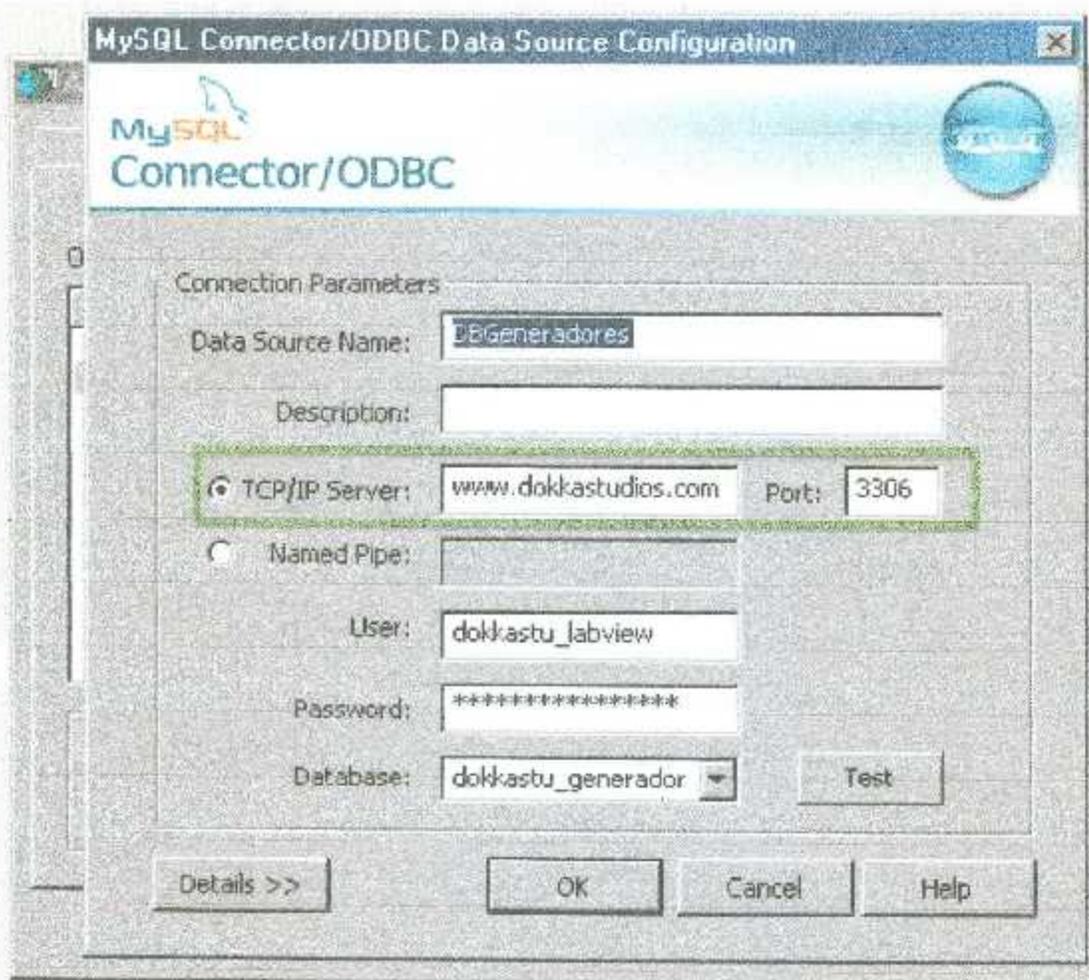


Figura 4-19 Acceso a servidor remoto a base de un DNS.

En la Figura 4-20 se tienen los mismos campos a llenar para crear un DSN, sólo que este ODBC es el que se utiliza para almacenar los datos en la BD local, es por eso que en el recuadro verde se introduce *localhost*, haciendo referencia al servidor local.

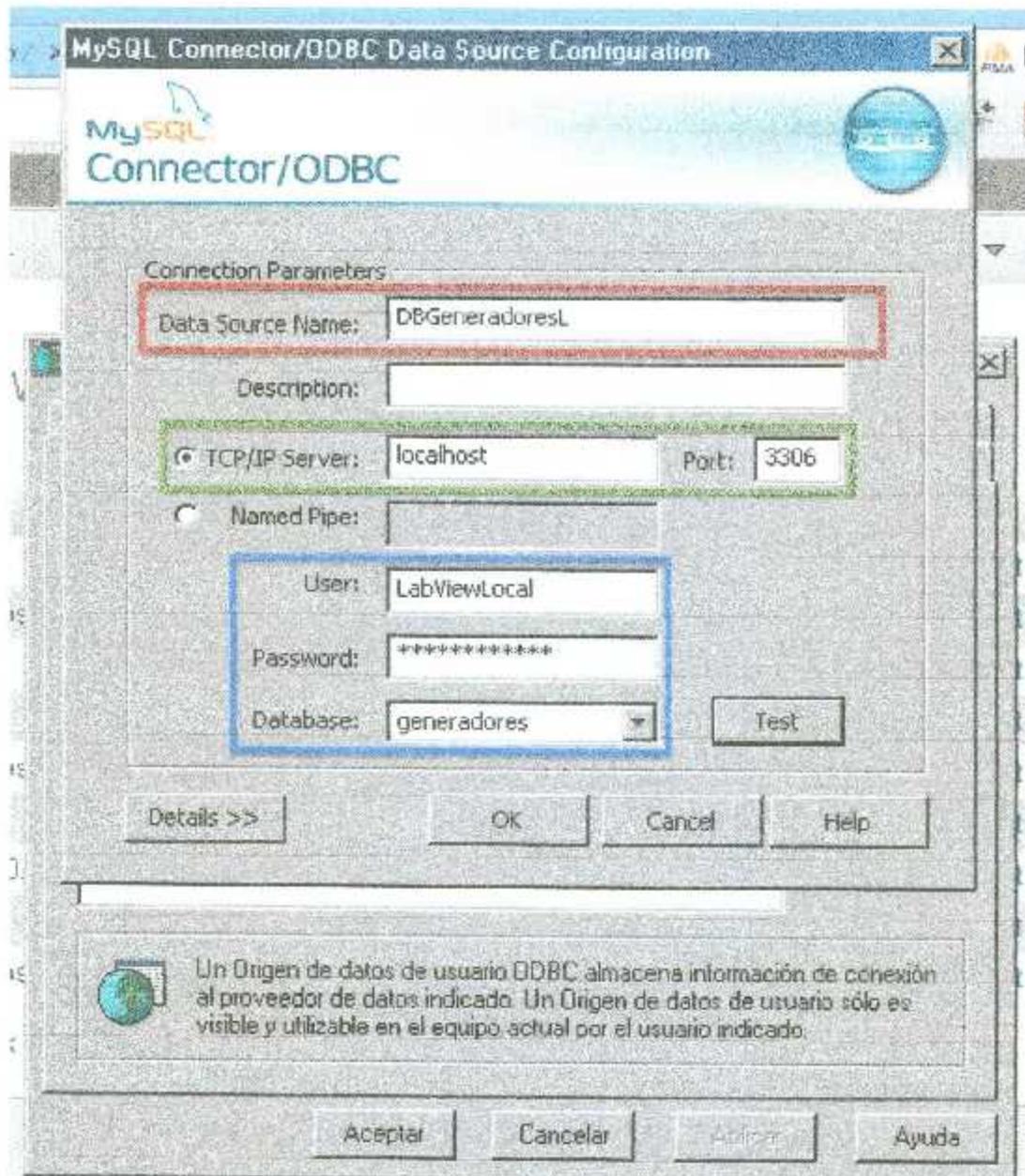


Figura 4-20 ODBC para almacenar en la BD local.

4.2.4 LabSql

LabSQL es un paquete de VI's libre que ayuda en la comunicación de LabView con MySQL, ya que LabView no cuenta con VI's para la conexión con MySQL [9]. Los VI's que se utilizan para la realizar la conexión y almacenar los datos son los siguientes:

- Crear Conexión: Este VI se encarga de crear una conexión para MySQL.
- Abrir Conexión: Este se encarga de abrir la conexión, aquí es donde se le da como parámetro el DSN creado para la BD local o remota.

- Ejecuta SQL: Este VI es el que permite ejecutar las consultas SQL y por otra parte, regresa el resultado de la consulta ejecutada, regresa una matriz de *string*.
- Cerrar Comunicación: Con este VI se cierra la comunicación con MySQL.
- Destruye Comunicación: Con este VI se destruye la comunicación con MySQL.

En la Figura 4-21 se observan los VI's para la comunicación con MySQL en LabView, en el recuadro verde el VI para crear la comunicación, en el recuadro rojo el VI para establecer la comunicación con MySQL, como se observa en el recuadro rosa inferior, se manda llamar el DSN que contiene el usuario previamente creado, en el recuadro azul se encuentra el VI para ejecutar los comandos SQL; en este caso el comando usado es el que se encuentra en el recuadro rosa superior, el cual devuelve una matriz con los generadores almacenados y sus datos; por último en el recuadro amarillo se cierra la comunicación y se destruye.

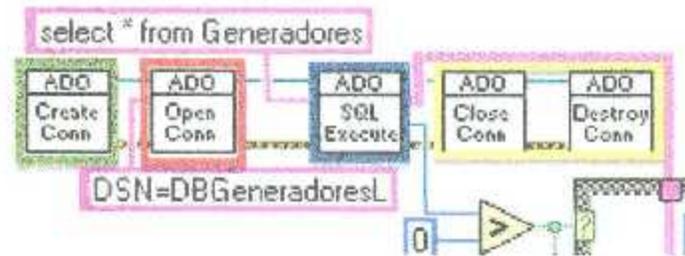


Figura 4-21 Esquema para obtener los generadores de la BD local.

4.2.5 Uso de Procedimientos Almacenados para almacenamiento local y remoto de los datos

Los SP son como las funciones en C, ayudan a hacer actualizaciones en la BD; estos procedimientos reciben parámetros, devuelven valores, y pueden ejecutarse remotamente, con sus propios derechos de acceso, estos se llaman con la palabra reservada *call* seguida del nombre del SP o con la instrucción *Execute*. Una ventaja de los SP es que estos se guardan en la memoria caché en forma compilada [41], por lo que aumenta el rendimiento cuando se vuelven a ejecutar. También ayudan para la protección de la información de las tablas de las que está compuesta la BD.

Por ejemplo, se toma la tabla Generadores de la BD, para insertar un generador nuevo en la BD, la inserción del generador en SQL sería como se muestra en la Figura 4-22. Si se desea que un usuario sólo actualice a ciertas tablas en la BD, sería imposible, porque el privilegio de insertar funciona para todas las tablas y si se le quita ya no se podrá insertar valores en ninguna tabla, entonces, si se quiere que el usuario sólo inserte en la tabla de generadores, se tendrá que crear un SP; entonces al usuario se le quita el privilegio de insertar y sólo se le da el privilegio de ejecutar, de esta forma sólo podrá ejecutar los SP que se le asignen. En la Figura 4-23 se muestra cómo se crea un SP, para insertar en la tabla generadores.

```

insert into Generadores
(
    Nombre,
    NumCell,
    Potencia,
    Latitud,
    Longitud,
    Altura
)
values
(
    'Generador Ejemplo',
    '8711002233',
    50,
    0.0,
    0.0,
    100
);

```

Figura 4-22 SQL para insertar un aerogenerador.

```

CREATE PROCEDURE Generadores.addGenerador
(
    _nombre varchar(50),
    _numCell varchar(15),
    _potencia float,
    _latitud float,
    _longitud float,
    _altura float
)
BEGIN
insert into Generadores
(
    Nombre,
    NumCell,
    Potencia,
    Latitud,
    Longitud,
    Altura
)
values
(
    _nombre,
    _numCell,
    _potencia,
    _latitud,
    _longitud,
    _altura
);
END

```

Figura 4-23 Creación de SP para insertar generadores.

Como se puede observar en la Figura 4-23, se definen parámetros de entrada como una función en C y éstos se pasan a los valores de la tabla, para insertar un aerogenerador con el SP se manda llamar de la siguiente forma:

```
call addGenerador(Generador Ejemplo, '8711942235', 50, 0.0, 0.0, 100);
```

Los datos de monitoreo que se almacenan en el servidor remoto y local se hacen por medio de SP, esto es, para aumentar el nivel de seguridad y la velocidad en la que se almacenan los datos.

Capítulo 5. Plataformas para la visualización de los datos

La visualización de los datos es la parte donde se observan los datos que se están monitoreando en un sistema. Presentar los datos de una forma gráfica ayuda a los usuarios a ver patrones y relaciones en grandes cantidades de datos que serían difíciles de descubrir si los datos se presentaran como listas tradicionales de texto plano [42].

Es por eso que se desarrollaron tres sistemas de visualización de los datos de monitoreo del acrogenerador, cada uno de los tres sistemas está desarrollado sobre una plataforma diferente.

5.1 Android

Como se mencionó anteriormente, Android es un SO libre que ha tenido un gran crecimiento en los últimos años y junto con él también, también han tenido un gran crecimiento los dispositivos móviles, tales como teléfonos celulares, tablets, relojes y TV. El responsable de este sistema operativo viene de la mano de la empresa Google. Este SO es usado por la mayoría de las compañías de teléfonos más importantes como, Samsung, LG, HTC, por mencionar algunas.

Google proporciona un Kit de Desarrollo de Software (SDK, de sus siglas en inglés, *Software Development Kit*) que consta de un conjunto de librerías para desarrollar aplicaciones para dispositivos celulares o tablets, así como también un IDE llamado *Android Studio* por parte de Google, éste está basado en la tecnología *IntelliJ* [43] (entorno de desarrollo para aplicaciones), el cual se muestra en la Figura 5-1.

El lenguaje de programación para las aplicaciones desarrolladas en *Android Studio* es Java, el cual es un lenguaje de programación libre. Java es usado como lenguaje para desarrollar aplicaciones en diferentes SO, como Mac OS X, Windows, Linux y Android, sólo que la única diferencia con este último, como ya se mencionó, es para dispositivos celulares y tablets.

Actualmente Android se encuentra en su versión 5.0, denominada *Lollipop*; en la Figura 5-2 se muestran las estadísticas tomadas de la página de Google de la distribución de las versiones del SO Android.

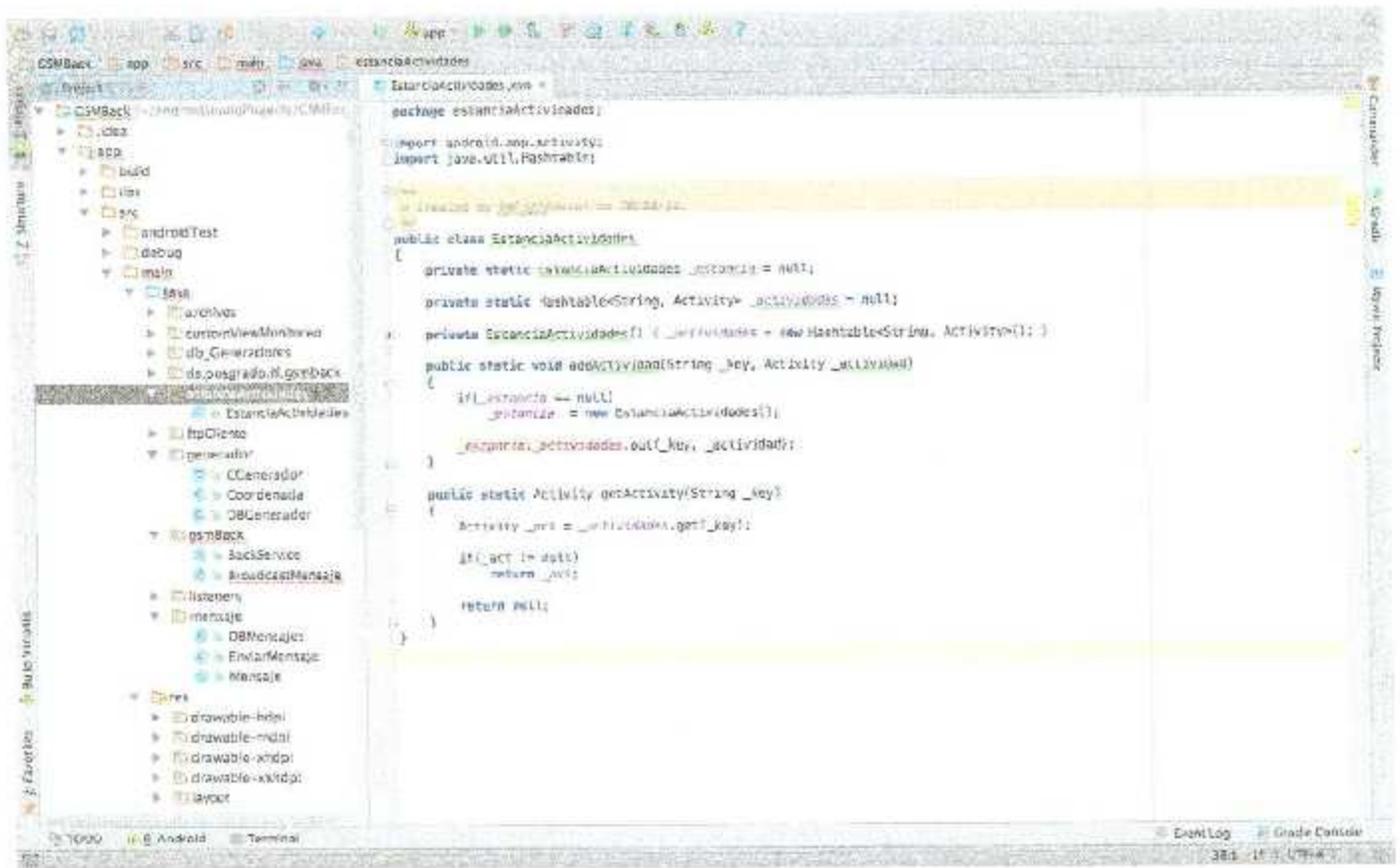


Figura 5-1 IDE Android Studio.

Version	Codename	API	Distribucion
2.2	Froyo	8	0.6%
2.3.3- 2.3.7	Gingerbread	10	9.8%
4.0.3- 4.0.4	Ice Cream Sandwich	15	8.5%
4.1.x	Jelly Bean	16	22.8%
4.2.x		17	20.8%
4.3		18	7.3%
4.4	KitKat	19	30.2%

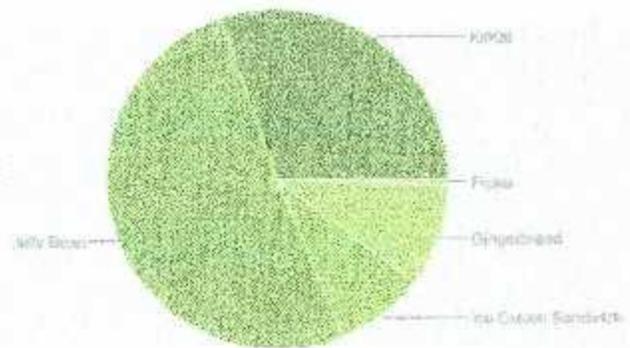


Figura 5-2 Estadísticas de distribución del SO Android del día 3 de Noviembre del 2014.

5.1.1 Aplicación para la visualización del estado del aerogenerador a base de SMS.

Como se mencionó anteriormente, uno de los métodos para el monitoreo del estado del aerogenerador es a base de SMS; estos SMS son enviados por un módulo *GSM Click* a través de las redes celulares hasta un teléfono celular. La aplicación puede ser instalada en cualquier dispositivo celular con una versión de *Android Jelly Bean* (4.1 en adelante), esto es porque las librerías que implementan las versiones anteriores a esta versión son obsoletas. Un ejemplo de éstas es la que maneja los mapas que se despliegan en el celular, una versión anterior a *Android Jelly Bean* maneja clases y métodos (funciones en C) con diferentes especificaciones, por esta razón la aplicación desarrollada en Android fue para la versión *Jelly Bean*, y de esta forma evitar problemas de compatibilidad en futuras modificaciones del software.

La aplicación final consta de treinta y tres clases; una clase es una técnica de la POO que encapsula los atributos y comportamiento de un atributo específico. Cada clase tiene un rol específico para la aplicación, tal como el manejo de los mensajes entrantes, el almacenamiento en la BD de los mensajes y la visualización de los mensajes; en la Figura 5-3 se muestran los paquetes con algunas de las clases que componen la aplicación.

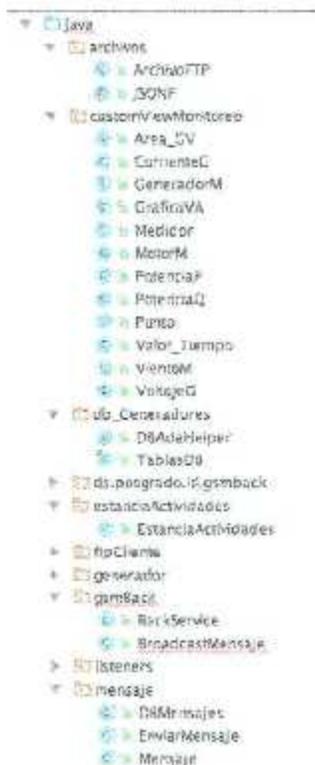


Figura 5-3 Paquetes y clases de la aplicación para la visualización del estado del aerogenerador.

La aplicación consta de seis ventanas, en la Figura 5-4a se tiene la primera ventana de la aplicación, ésta muestra los aerogeneradores almacenados en la BD. En el recuadro azul se puede observar que se tiene la opción de agregar aerogeneradores directamente en esta ventana, en la Figura 5-4b se muestran los campos que se deben de llenar para almacenar el aerogenerador. En el recuadro rojo se observa la opción para almacenar el aerogenerador, una vez que el aerogenerador se almacenó, se despliega un mensaje como el que se observa en la Figura 5-4c.



Figura 5-4 Ventanas para mostrar los aerogeneradores almacenados y para almacenar aerogeneradores.

Si se selecciona uno de los aerogeneradores que se muestran en la Figura 5-4a se muestra la información ya almacenada en la BD de ese aerogenerador, en la Figura 5-5a se puede observar cómo se muestran los datos, incluso un mapa con un indicador de la ubicación del aerogenerador, en la Figura 5-5b, en el recuadro rojo se pueden observar opciones para el generador almacenado, tales como, modificar sus datos, eliminar el generador de la BD, ver los SMS de monitoreo o de alerta.

5.1.2 Visualización de los Mensajes

Para la visualización de SMS del estado del generador se usaron componentes personalizados, ya que los componentes que vienen incluidos en el IDE no son suficientes para poder visualizar la información de los SMS del estado del aerogenerador.

Si se selecciona una de estas dos opciones de ver SMS de monitoreo o alerta, se despliega una ventana más que muestra los SMS; en este caso los SMS que se observan son SMS de prueba enviados desde el módulo *GSM Click*. Por ejemplo, si se selecciona

la opción de SMS de monitoreo se dirigirá a la ventana donde se encuentra una lista de estos SMS, en la Figura 5-6a en el recuadro rojo se muestra uno de los SMS de ejemplo para el monitoreo del estado del aerogenerador. En la Figura 5-6b y Figura 5-6c, se puede observar cómo es que se muestra la información de los SMS; los medidores analógicos que se observan ahí, son los controles personalizados desarrollados para la aplicación.

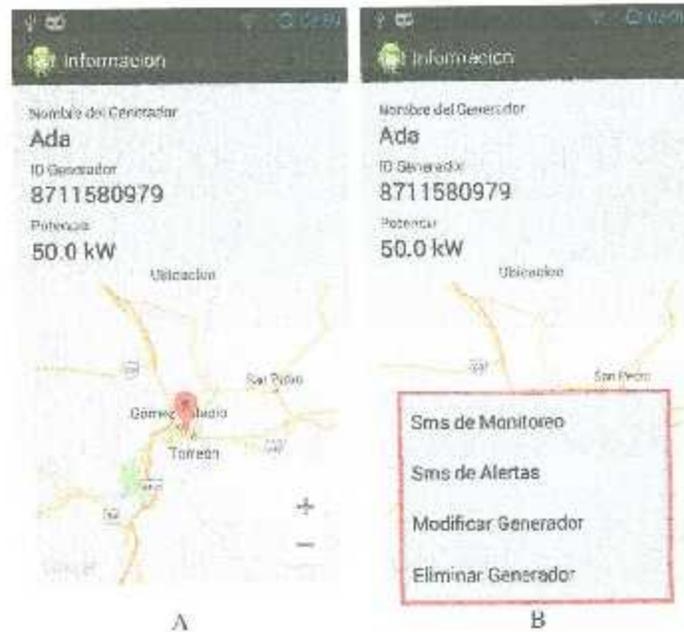


Figura 5-5 Información del aerogenerador en la BD.

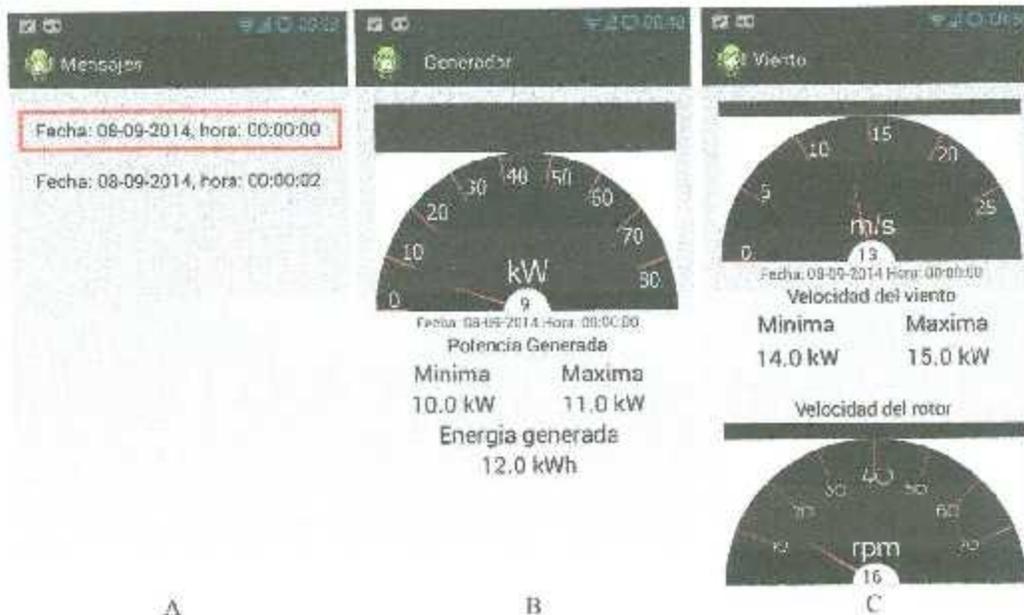


Figura 5-6 Visualización de la información de los SMS.

5.1.3 BroadcastReceiver

Básicamente, cuando se cierra una aplicación en Android el SO mata el proceso de la aplicación, pero si en la aplicación se define un *BroadcastReceiver* (clase especial de Android para el manejo de procesos en segundo plano), ésta sirve para mantener un proceso en ejecución, inclusive si la aplicación se ha cerrado, de esta forma, cuando llegue un SMS, este proceso advierte a la aplicación, la cual tomará el SMS para checar si es algún SMS del estado de algún aerogenerador o de advertencia. En la Figura 5-7 se muestra el método que tiene la llamada de un mensaje entrante.

```
public void onReceive(Context _context, Intent intent)
{
    Generador _activMensajes = (Generador) InstanciaActividades.getActivity("Generador");
    LlegoMensaje _numeroMensaje = (LlegoMensaje) _activMensajes;
    _activMensajes = null;

    Bundle _bundle = intent.getExtras();
    String _fecha = getFormatoFH("Fecha");
    String _hora = getFormatoFH("");

    try
    {
        if(_bundle != null)
        {
            Object[] _datosPDUS = (Object[]) _bundle.get("pdus");
            SmsMessage _mensajeActual = null;

            for (int i = 0; i < _datosPDUS.length; i++)
                _mensajeActual = SmsMessage.createFromPdu((byte[]) _datosPDUS[i]);

            String _numero = _mensajeActual.getDisplayOriginatingAddress();
            String _mensaje = _mensajeActual.getDisplayMessageBody();

            Toast _aviso = null;

            if(DBGenerador.ExisteCelular(_context, _numero))
            {
                _aviso = getAvisoMensaje(DBMensajes.addMensaje(_context, new Mensaje(_mensaje, _hora, _fecha), _numero), _context);
                _aviso.show();

                _ftpArch = new ArchivoFTP();
                _ftpArch.ArchivoJSONF.getArchivoJSONF(DBMensajes.getLastMensaje(_context));
            }
            else
            {
                _aviso = Toast.makeText(_context, "No existe el Aerogenerador: " + _numero, _duracion);
                _aviso.show();
            }
        }
    }
    catch (Exception e)
    {
        Log.e(_tagClass, "Error en el Mensaje de Entrada");
    }
}
```

Figura 5-7 Método que obtiene los SMS de entrada.

El método verifica el número del SMS de entrada, si éste existe en la BD, almacena el SMS, de lo contrario lo ignora; este método es llamado por todos los SMS de entrada en el dispositivo celular y no todos los SMS contienen los datos de estado del aerogenerador.

5.1.4 Almacenar mensajes en SQLite para Android

Todos los SMS del estado del aerogenerador son almacenados en una BD SQLite, este motor de BD es similar al que se usó en LabView para almacenar los datos de monitoreo y los aerogeneradores vinculados con el dispositivo celular. La diferencia es que SQLite sólo tiene soporte para Android, en la Figura 5-8 se muestra el EER de la BD en Android.



Figura 5-8 Diagrama EER de la BD para Android.

Como se puede observar en la Figura 5-8, la BD está compuesta por tres tablas:

- TipoSMS: Esta tabla almacena los tipos de SMS que puede llegar a tener el aerogenerador, por ejemplo, SMS de estado o SMS de alerta.
- SMS: Esta tabla almacena los mensajes que llegan desde el aerogenerador.
- Generadores: En esta tabla se almacenan los aerogeneradores asociados al dispositivo celular.

5.2 Aplicación Web

Como se mencionó anteriormente, MySQL es un motor de BD usado también para servidores web, esto da la posibilidad de acceder a los datos desde una aplicación web, brindando una gran ventaja, ya que se puede acceder a una página web básicamente desde cualquier parte del mundo.

La forma en que se almacenan los datos es por medio los VT's LabSQL y de la conexión ODBC creados anteriormente.

5.2.1 Obtener datos del Servidor

Para obtener los datos desde la página web se necesita una conexión de la página con la BD del servidor, esa conexión se obtiene mediante Ajax y PHP para mostrar los datos; para mandar llamar Ajax y mostrar los datos se utiliza JQuery. En la Figura 5-9 se muestra la aplicación Web para mostrar los datos almacenados en la BD del servidor remoto.

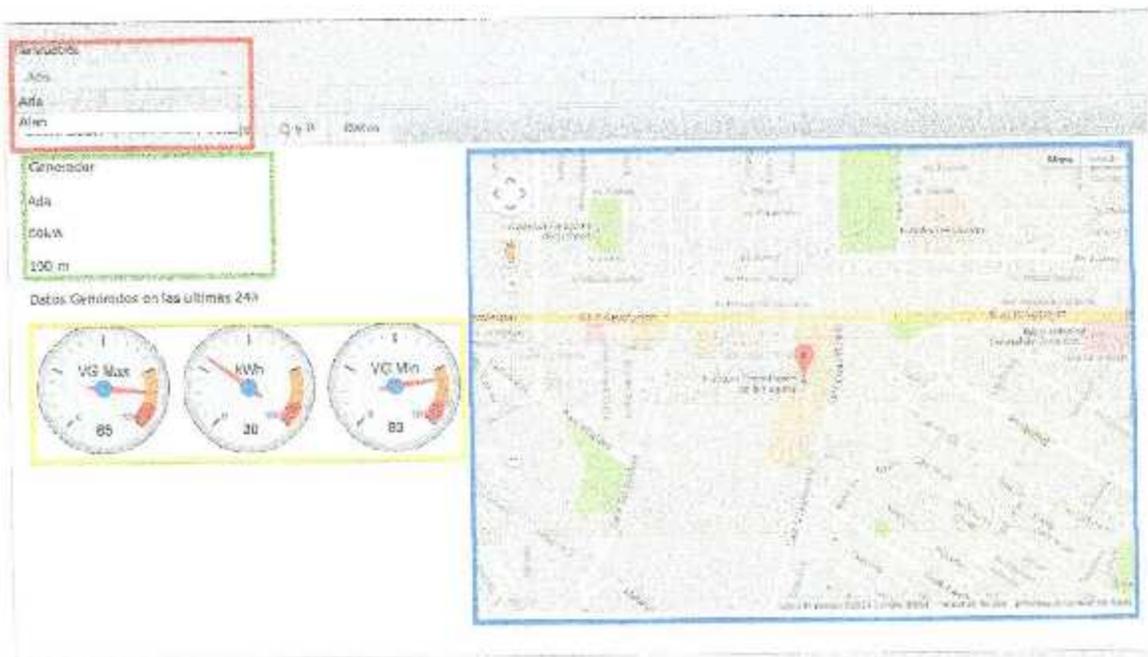


Figura 5-9 Aplicación Web para mostrar los datos almacenados en la BD.

En la Figura 5-9, en el recuadro rojo se, pueden seleccionar los aerogeneradores almacenados en la BD, después consultar sus datos almacenados; en el recuadro verde se observan los datos del generador seleccionado, en el recuadro amarillo se muestran unos indicadores de los valores relevantes almacenados en las últimas 24 horas parecidos a los datos que se muestran en la aplicación de SMS. Por último, en el recuadro azul se muestra la ubicación del aerogenerador seleccionado.

En la Figura 5-10 se muestra el voltaje correspondiente al aerogenerador almacenado en la BD.

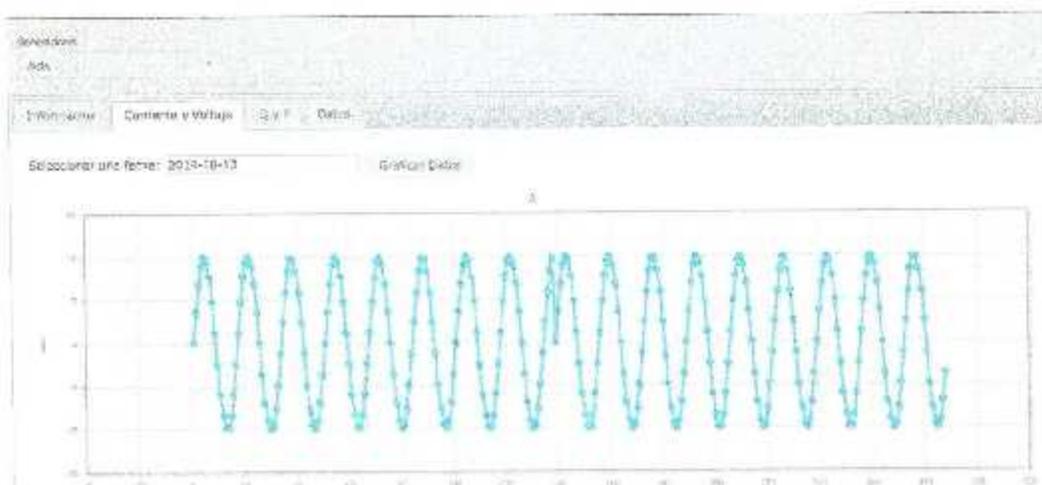


Figura 5-10 Gráficas de los datos del voltaje.

La información se muestra seleccionando la fecha de los datos que se quieren mostrar, como se observa en el recuadro rojo de la Figura 5-11; una vez seleccionada la fecha se da *click* en el botón *Graficar Datos*, como se observa en el recuadro verde.



Figura 5-11 Selección de fecha para visualizar los datos de la BD.

5.2.2 JQuery, Ajax y PHP

JQuery es una librería libre basada en JavaScript, la cual está diseñada para simplificar la creación de aplicaciones de JavaScript, ésta facilita la selección de elementos y la creación de animaciones, maneja eventos e implementa Ajax [44].

Por otra parte, Ajax es una técnica de programación Web fundamental para cualquier sitio web que ofrezca un mayor dinamismo y un menor tiempo de espera para el usuario. Sitios Web como Google Mail, Facebook y Yahoo! están desarrollados con Ajax [45].

Por último, para obtener los datos de la BD es necesario hacer una conexión con la BD utilizando PHP, como se mencionó anteriormente, la razón por la que la conexión se hace con PHP es porque se ejecuta del lado del servidor, su sintaxis es parecida a la del lenguaje de programación C [46].

En la Figura 5-12 se muestra una función en JQuery implementando Ajax para obtener los aerogeneradores de la BD.

```

function getGeneradores()
{
    $.ajax(
    {
        type: "POST",
        url: "../PHP/getGeneradores.php",
        dataType: "text",
        success: function(jsonDatos)
        {
            var _genDB = JSON.parse(jsonDatos);
            _gen5 = _genDB;

            _lat = _genDB.Generadores[0].Latitud;
            _long = _genDB.Generadores[0].Longitud;

            google.maps.event.addListener(window, 'load', initMapa(_lat, _long));
        }
    });
}

```

Figura 5-12 Función en JQuery para obtener los datos usando Ajax.

En el recuadro azul de la Figura 5-12 se observa la forma de llamar un archivo PHP por medio de Ajax, el cual regresa la información en formato JSON, como se observa en el recuadro amarillo, previamente ya se explicó como obtener los datos de un archivo tipo JSON.

En la Figura 5-13 se muestra el archivo, en PHP, que se conecta con la BD y obtiene los aerogeneradores.

```

<?php
include('Conexion.php');

$json = array();
$query = "select * from getGeneradores";
_consulta = $mysqlConx -> query($query);

while($val = $consulta -> fetch_assoc())
    $json[] = $val;

$mysqlConx ->close();
echo '{"Generadores":'.json_encode($json).'}';

```

Figura 5-13 Archivo PHP para obtener los aerogeneradores de la BD.

5.3 Aplicaciones en LabView

Por último, la tercera plataforma para visualizar los datos es por medio de LabView; la visualización consta de dos aplicaciones que se conectan vía TCP/IP, una aplicación se encuentra en la computadora local (donde se encontraría el aerogenerador) y la otra se encuentra en el servidor remoto.

5.3.1 Aplicación en computadora local.

La aplicación donde se visualizan los datos que van a ser obtenidos del aerogenerador es la misma que obtiene los datos vía serie o por medio del módulo NI USB-6008, y es la misma que se encarga de almacenar los datos en la BD local y en la BD remota, esta aplicación cuenta con varios VI's tales como gráficas y medidores analógicos. En la Figura 5-14 se muestra la interfaz de usuario para visualizar los datos, en el recuadro rojo se observan los datos que llegan desde el puerto serie, estos datos son datos de prueba generados por el PIC18F45K22.

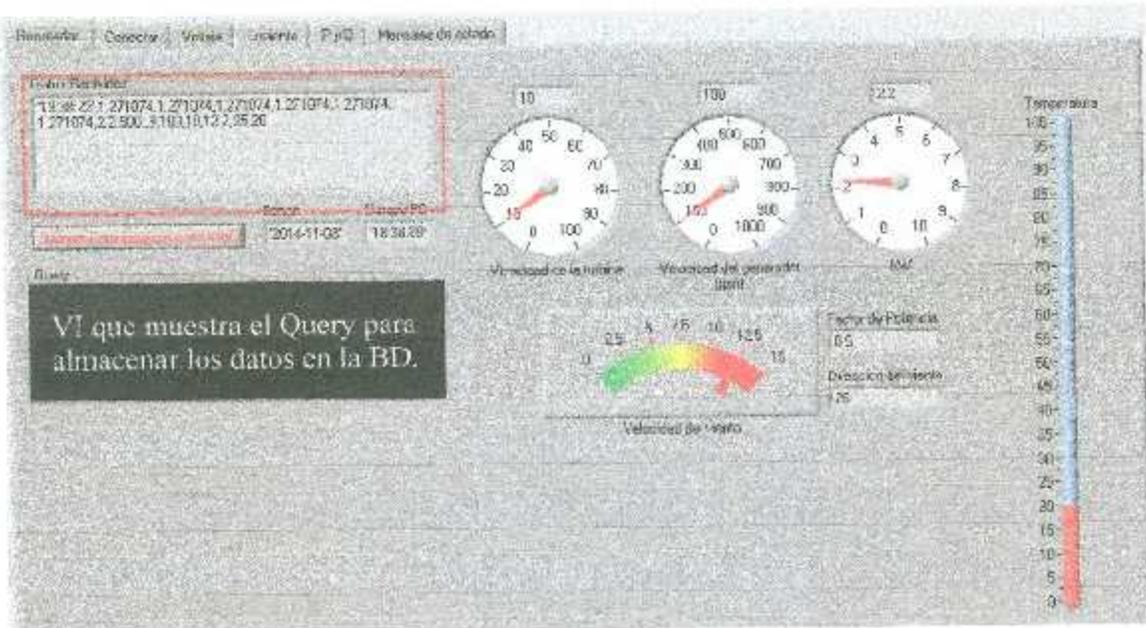


Figura 5-14 Interfaz para la visualización de los datos en la computadora local.

En la Figura 5-15 se muestra el programa a bloques para enviar la información vía TCP/IP.

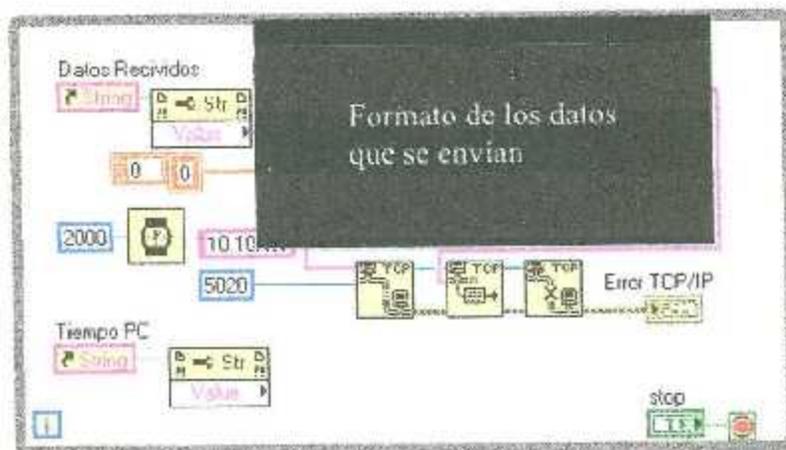


Figura 5-15 Bloques para enviar la información vía TCP/IP.

5.3.2 Aplicación en el servidor remoto

En la Figura 5-16 se muestra la interfaz grafica de la aplicación en servidor remoto desarrollada en LabView, en el recuadro amarillo se muestran los datos del aerogenerador de la BD local, en los recuadros azules se muestran los datos de prueba que llegan por el puerto TCP/IP, por último, en el recuadro rojo se muestran los demás datos almacenados en la BD, dependiendo del aerogenerador seleccionado.

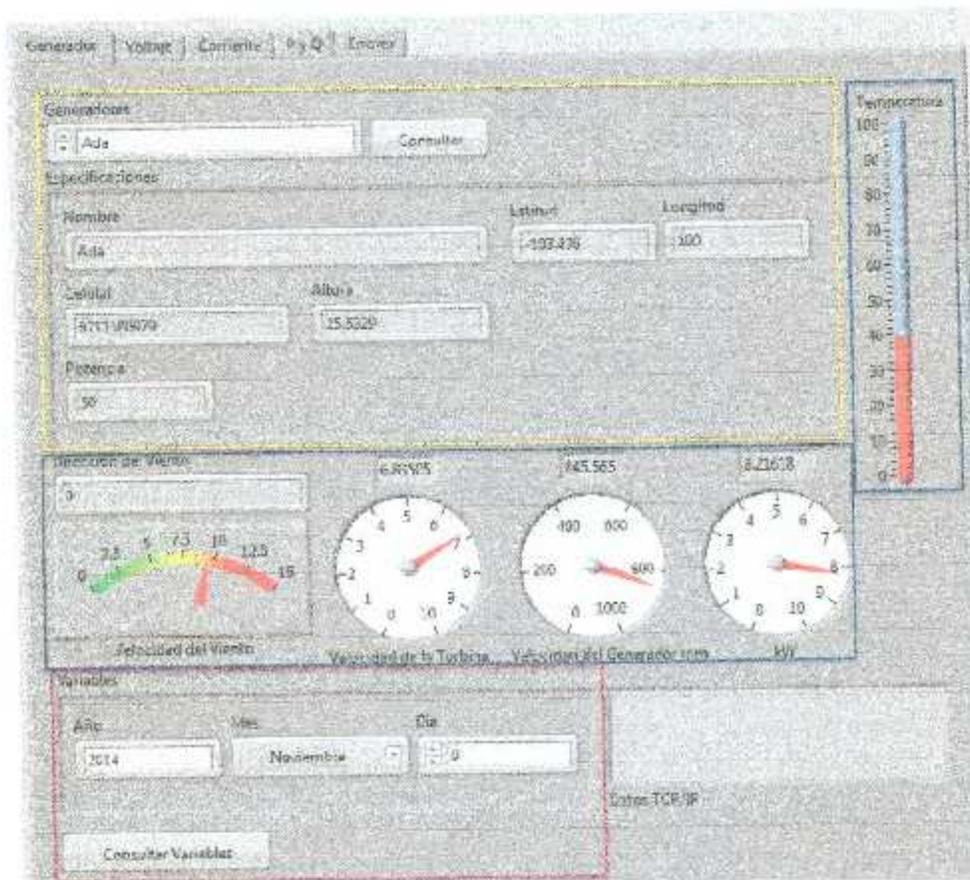


Figura 5-16 Aplicación para la visualización remotamente de los datos del aerogenerador.

En la Figura 5-17 se muestra el programa a bloques para recibir la información desde la computadora local.

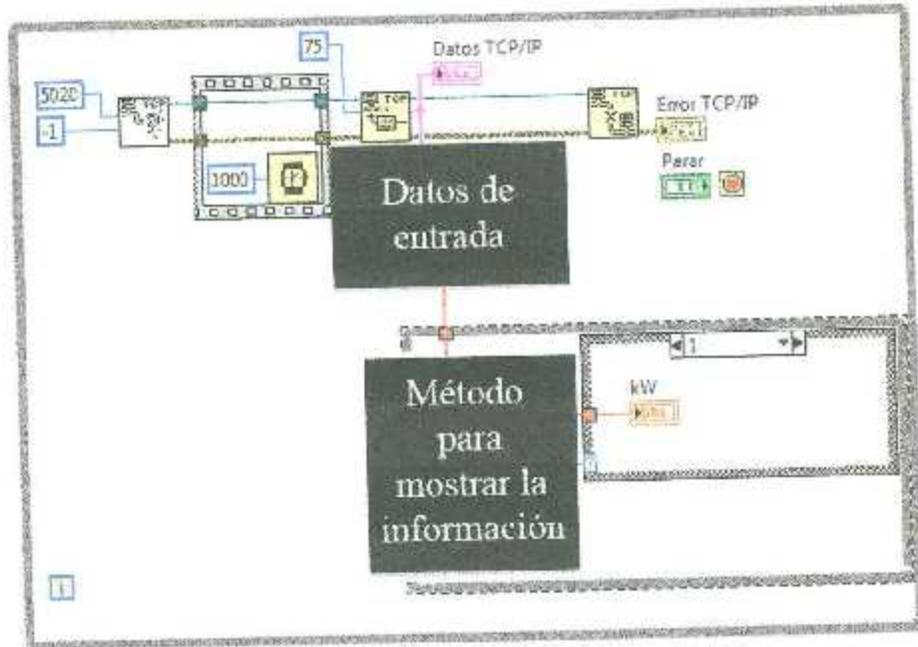


Figura 5-17 Programa a bloques para mostrar la información.

Capítulo 6. Conclusiones

6.1 Resultados

El Sistema de Monitoreo Remoto desarrollado en esta tesis, no alcanzó a llevarse a la etapa de pruebas directas con un aerogenerador físico, debido al tiempo que tomó desarrollar los diferentes sistemas para el monitoreo del mismo.

Para probar cada una de las tecnologías desarrolladas se generaron datos de prueba; por ejemplo, para adquirir los datos del microcontrolador se optó por conectar generadores de funciones a las entradas del convertidor analógico-digital, la señal tenía un proceso para después ser enviada por el puerto serie a la computadora con LabView, de modo que las variables pudieran ser mostradas y almacenadas en la base de datos. De igual forma, las señales que se adquieren por medio del DAQ NI-6008 son generadas con generadores de funciones, los cuales se pueden observar en la Figura 6-1.

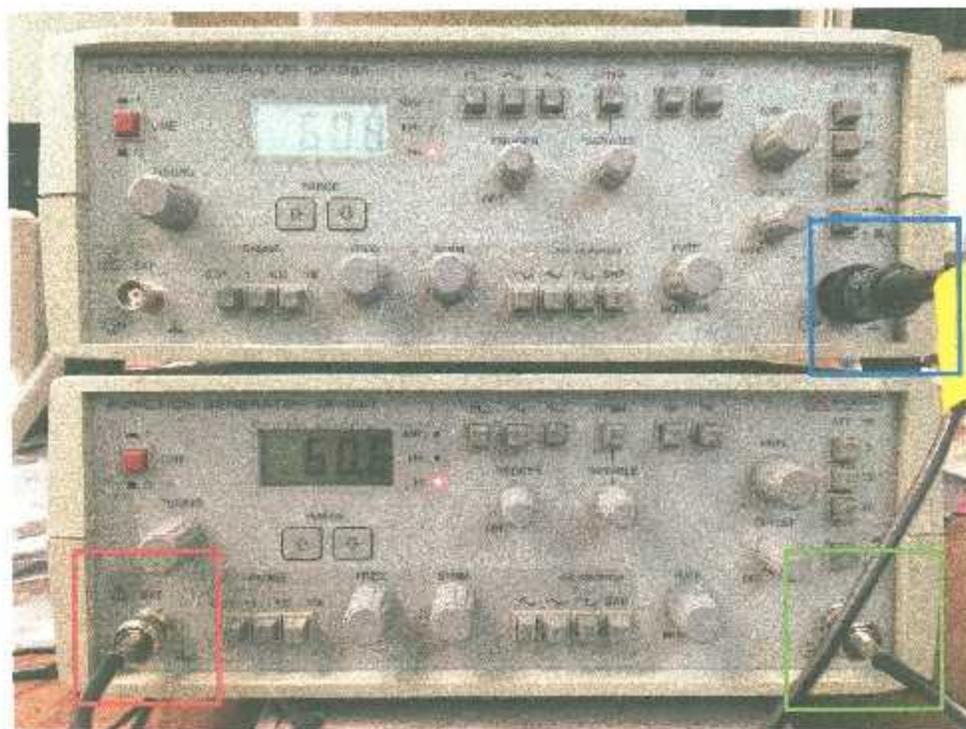


Figura 6-1 Generadores de funciones para las pruebas de del sistema de monitoreo.

Como se puede apreciar en la Figura 6-1, en el recuadro azul se tiene la salida de la fase A, en el recuadro rojo la salida de la fase B y por último en el recuadro verde la salida de la fase C. Estos generadores de funciones ayudaron con las pruebas para la adquisición de datos con el microcontrolador y el DAQ.

Por otra parte, al momento de realizar las pruebas para almacenar los datos recibidos, uno de los métodos propuestos para almacenar los datos fue a base de archivos de texto con formato en XML y JSON, el cual al final se descartó por la forma en que se almacenaban los datos. Los datos para los archivos XML y JSON se almacenaban en el Disco Duro (DD) de la computadora, para después almacenarlos en el servidor remoto por medio de FTP; uno de los problemas que surgieron con este método fue que a la hora de crear los archivos, cuando ocurría un problema no controlado mientras se creaba el archivo, éste resultaba dañado, haciendo que los datos no tuvieran el formato correcto, lo cual dificultaba la lectura de los archivos.

Con la implementación de una base de datos se eliminó la necesidad de crear archivos en la computadora local y en el servidor remoto, ya que los datos se almacenaban al mismo tiempo en los dos destinos, el único inconveniente fue que si se perdía la conexión con el servidor remoto, los datos sólo se almacenaban en la base de datos local, en una tabla temporal hasta que se restableciera la comunicación con el servidor remoto.

Otro inconveniente a la hora de almacenar los datos en el servidor remoto donde se encuentra la página web es que actualmente éste es un servidor contratado, en el cual no se tienen privilegios de administrador, por lo cual se tienen restringidas algunas operaciones sobre la base de datos; también se cuenta con un espacio limitado de 10 GB de almacenamiento en el DD, para la BD y los archivos que constituyen la página web.

6.2 Conclusiones

Los sistemas de monitoreo y control son una clase importante de sistemas de tiempo real que realizan una acción cuando se detecta algún valor excepcional del sensor. Actualmente el desarrollo de sistemas de monitoreo para aerogeneradores es parte fundamental para el monitoreo de variables como las corrientes o voltajes generados, la temperatura a la que se encuentra el generador, la velocidad de la turbina y la velocidad del viento o control remoto sobre el aerogenerador en caso de fallas, por mencionar solamente algunas. También el monitoreo es de mucha utilidad al momento de la creación de parques eólicos, para la evaluación de los sitios potenciales de instalación y también para el seguimiento y mantenimiento de los parques eólicos existentes.

Para hacer frente a los problemas de operación y mantenimiento, se están sustituyendo las prácticas de mantenimiento tradicionales, como el mantenimiento periódico por tiempos, con el monitoreo y el mantenimiento basado en las condiciones monitoreadas. Por ejemplo, un aerogenerador de gran potencia suele estar ubicado en partes lejanas a la sociedad y puede llegar a tener una altura de hasta 100 m, inicialmente esto dificultaba las labores de mantenimiento para un aerogenerador, lo cual ocasionaba que el personal visitara los aerogeneradores cada cierto tiempo programado, por lo que si el aerogenerador sufría de averías antes del tiempo programado, el personal no se percataba.

Como proyecto de tesis, se propuso el diseño y desarrollo de un sistema de monitoreo remoto de un aerogenerador, para lo cual fue necesario trabajar con diferentes

plataformas para la adquisición de los datos que pudieran ser obtenidos de un aerogenerador.

Como resultado, se desarrolló una aplicación en Android para un teléfono celular, la cual implementa el uso de la tecnología GSM; esta tecnología ofrece la ventaja de tener una gran área de cobertura y emplea SMS para poder monitorear el estado de un aerogenerador. Sin embargo, los SMS sólo pueden estar compuestos por 160 caracteres, de tal modo que no pueden llegar a ser el sistema principal para el monitoreo de un aerogenerador, pero sí pueden utilizarse para notificar el estado del aerogenerador cada cierto tiempo, notificando por ejemplo de alguna falla del aerogenerador a ciertas personas cuyas decisiones pudieran ser importantes.

También se desarrolló una aplicación con LabView, la cual es capaz de obtener datos por medio del puerto serie usando un microcontrolador y por medio de un DAQ. La aplicación desarrollada también tiene la capacidad de almacenar en una base de datos local y en una base de datos remota los datos adquiridos. La forma en que se almacenan los datos es por medio de comunicaciones ODBC utilizando MySQL para la base de datos, la cual se crea tanto en la computadora de monitoreo como en un servidor externo; en la máquina donde se monitorea el servidor sólo se almacenan los datos de un solo aerogenerador, y en la máquina servidor se almacenan los datos de todos los aerogeneradores que se conectan a éste.

Se desarrollaron tres sistemas para visualizar los datos monitoreados, cada uno de los cuales se desarrolló sobre una plataforma diferente. La primera plataforma se desarrolló en el sistema operativo Android y permite visualizar los datos monitoreados desde un teléfono celular; permite monitorear el estado del aerogenerador mediante el uso de SMS enviados a través de las redes celulares. La segunda plataforma da la posibilidad de acceder a una página web, básicamente desde cualquier parte del mundo. Se desarrolló una página web en la cual se pueden consultar los datos obtenidos de algún aerogenerador. Para obtener los datos desde la página web se necesita una conexión de la página con la base de datos del servidor, la cual se obtiene mediante tecnologías como HTML, JQuery y PHP para mostrar los datos. La tercera plataforma se desarrolló con LabView y cuenta con la capacidad de conectarse a cualquier servidor; consta de dos aplicaciones que se conectan vía TCP/IP: una aplicación se encontrará en la computadora local, donde se encuentra el aerogenerador y la otra en el servidor remoto.

Como resultado del trabajo de tesis se obtuvo un sistema de monitoreo remoto que permite sustituir el mantenimiento preventivo tradicional con una herramienta de monitoreo que utiliza tecnología de punta y permite hacer frente a los problemas de operación y mantenimiento tradicionales. El monitoreo se realizó usando el protocolo abierto de mensajes y un servidor, se utilizó telefonía celular y se usó el acceso a internet para que desde cualquier lugar se pueda efectuar el monitoreo del estado del aerogenerador, lo cual es posible gracias al uso de microcontroladores, Android, TCP/IP, HTML, PHP y tecnología de bases de datos.

6.3 Trabajo Futuro

Como trabajo futuro se contempla la posibilidad de utilizar un servidor con mayores capacidades y con una IP dedicada, este servidor tendrá el sistema de monitoreo para acceder en forma remota a los datos almacenados de un aerogenerador físico y completamente funcional. También es necesario evaluar el comportamiento del sistema ante casos de fallas, tanto de la red como de los sistemas de adquisición y cómo afecta esto a la consistencia de los datos almacenados.

Referencias

- [1] José Roldán Vilorio, *Energías renovables: Lo que hay que saber*. S.A. Ediciones Paraninfo, 2nd Ed. Madrid España, 2012.
- [2] Gilberto Enriquez, *Tecnologías de generación de energía eléctrica* Limusa, 1^{er}. Ed. México 2009.
- [3] José María Escudero, José María Escudero López, *Manual de energía eólica/ Guide to Wind Energy*. Mundi-Prensa Libros, 2^a Ed. Madrid 2003.
- [4] Ian Sommerville, *Ingeniería del software*. Pearson Educación. S.A. 7^a Ed. Madrid 2005.
- [5] Miller, R.C. Lemak, T.A. Osborne, R.L. Repplier, C.C. Morrison, D.R. Boomgaard, D.J. "Two Recent developments in Monitors for large Turbine Generators". IEEE Transactions. on Energy Conversion, Vol. 3, Agosto 2006, pp 653-659.
- [6] Aiswarya, V. Prakash, N.K. "Wind Turbine Instrumentation System using LabVIEW". Global Humanitarian Technology Conference: South Asia Satellite (GHTC-SAS), 2013 IEEE, pp 218-222.
- [7] Kusiak, A. Verma, A. "A Data-Mining Approach to Monitoring Wind Turbines". IEEE Transactions on. Sustainable Energy, Vol. 3, Diciembre 2011, pp 150-157.
- [8] Nguyen-Vu Truong. Duc-Lung Vu "Remote Monitoring and Control of Industrial Process Via Wireless Network and Android Platform". International Conference on. Control, Automation and Information Sciences (ICCAIS) 2012, pp 340-343.
- [9] Swiszczy, G. Cruden, A. ; Booth, C. ; Leithead, W. "A Data Acquisition Platform for the Development of a Wind Turbine Condition Monitoring System". International Conference on. Condition Monitoring and Diagnosis, 2008. CMD 2008, pp 1358-1361.
- [10] Barton, J.P. Watson, S.J. "Analysis of electrical power data for condition monitoring of a small wind turbine". Renewable Power Generation, IET, Vol 7, Junio 2013, pp 341-349.

- [11] Dongxiang Jiang, Qian Huang, Liangyou Hong. "Test System of a Small Wind Turbine under Laboratory Conditions". World Non-Grid-Connected Wind Power and Energy Conference, 2009. WNWEC 2009, pp 1-4.
- [12] Ma Chenyuan, Zhao Daoli, Wan Tianhu, Wu Luochang, Li Yanfeng. "Research based on labview on wind turbine monitoring system". International Conference on Sustainable Power Generation and Supply (SUPERGEN 2012), Septiembre 2012, pp 1-4.
- [13] Kovalcik, M. Fecilak, P. ; Jakab, F. "Control and monitoring system of Small Water and Wind Power Plant". Conference on Emerging eLearning Technologies and Applications (ICETA), Octubre 2013 IEEE 11th International, pp 235-238.
- [14] Huiliang Shang, Qi Zhang, Mimin Jin, Wenxin Wang, Jiajun Zhu, Qingsheng Kong. "A Portable Pulse Signal Acquiring and Monitoring System based on Android Platform". Conference on Bioinformatics and Biomedicine (BIBM), Diciembre 2013 IEEE International, pp 226-227.
- [15] Weibo Song, Hong Yu, Ce Liang, Qihua Wang, Yunfeng Shi. "Body Monitoring System Design Based on Android Smartphone". World Congress on Information and Communication Technologies (WICT), Octubre 2012, pp 1147-1151.
- [16] Liang-Hung Wang, Tsung-Yen Chen, Shuenn-Yuh Lee, Huan Chen. "Implementation of a Personal Health Monitoring System in Cardiology Application". Conference on Circuits and Systems (APCCAS), Diciembre 2012 IEEE Asia Pacific, pp 236-239.
- [17] Zeki, A.M. Elnour, E.F. Ibrahim, A.A. Haruna, C. Abdulkareem, S. "Automatic Interactive Security Monitoring System". International Conference on Research and Innovation in Information Systems (ICRIIS), Noviembre 2013, pp 215-220.
- [18] De Luca, G. Lillo, P. Mainetti, L. Mighali, V. Patrono, L. Sergi, I. "The use of NFC and Android technologies to enable a KNX-based Smart Home". International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Septiembre 2013, pp 1-7.
- [19] Wenqing Ge, Hua Lai, Mingjie Qi, Linlin Cui. "Design of Temperature and Humidity Monitoring Terminal System Based on Android". International Conference

- on. System Science, Engineering Design and Manufacturing Informatization (ICSEM), Octubre 2012, Vol 1, pp 98-100.
- [20] Sostaric, D. Martinovic, G. Zagar, D. "GPS Tracking of Self-balancing Vehicle for Extreme Environment based on AndroidOS". Telecommunications Forum (TELFOR), Noviembre 2012, pp 1060-1063.
- [21] Christos Petsos. Kostas Kalogirou. Evangelos Bekiaris. "Android Application Front-end for an Energy Brokerage Agent". International Conference on. Power Systems, Energy, Environment, 2014, pp 105-110.
- [22] Kovalcik, M. Fecilak, P. Jakab, F. "Control and monitoring system of Small Water and Wind Power Plant". International Conference on. Emerging eLearning Technologies and Applications (ICETA), Octubre 2013 IEEE, pp 235-238.
- [23] Wang Cheng-gang. Wang Zi-zhen. "Design and Implementation of MW-class Wind Turbine Fault Monitoring System Based on GSM Short Message". International Conference on. Computer, Mechatronics, Control and Electronic Engineering (CMCE), Agosto 2010, Vol. 6, pp 5-8.
- [24] Raul Ionel. Gabriel VasIU. Septimiu Mischic. "GPRS based data acquisition and analysis system with mobile phone control". ELSEVIER Measurement Julio 2011. Vol. 45, pp 1462-1470.
- [25] Enrique Herrera Pérez, *Tecnologías y redes de transmisión de datos*. Limusa, 1ª Ed. Mexico 2003.
- [26] Joaquín del Río Fernández, Shahram Shariat-Panahi, David Sarria Gandul, Antonio Manuel Lázaro, *LABVIEW Programación para sistemas de instrumentación*. Alfaomega, 2ª Ed. Madrid España 2011.
- [27] José Lajara Vizcaino, José Pelegrí Sebastián, *LABVIEW Entorno Grafico de Programación*. Marcombo S.A. 2ª Ed, Barcelona España 2011.
- [28] José Miguel Molina Martínez, Manuel Jiménez Buendía, *Programación Gráfica para Ingenieros*. Marcombo, 1ª Ed. Barcelona España 2010.
- [29] David Mint, *Servicio de Mensajes Cortos (SMS) el Mercado Telefónico de España*. The Wilcox Trading Company, 1ª Ed. España 2013.
- [30] García Rodrigo, Gregorio Morales Santiago, *Instalaciones de radiocomunicaciones*. Ediciones Paraninfo S.A. 1ª Ed. Madrid España 2012.

- [31] Antonio Arranz, *Administración de datos y archivos por computadora/Administration Computer Files and Data*, Megabyte Noriega Editores, 2ª Ed. México 2002.
- [32] Pedro Alonso Jordá, Fernando García Granada, Eva Onaindía de la Rivaherrera, *Diseño e implementación de programas en lenguaje C*. Universitat Politècnica de València, 1ª Ed. España 1998.
- [33] Juan Jesús Tortajada Cordero, *La guía definitiva de XML: ¡¡XML, JSON y mucho más!!*. Autor-Editor. 2014.
- [34] Sebastian Bassi, *Python en 8 clases: Aprendiendo a programar con Python*. Genes Digitales, 4ª Ed. 2013.
- [35] Jesús Martín Alloza, *Access 2003*. Innovación y Cualificación S.L. 1ª Ed. España 2005.
- [36] Ma. Victoria Nevado Cabello, *Introducción a las Bases de Datos Relacionales*. Visión Libros, España 2010.
- [37] C. J. Date, Sergio Luis María. Ruiz Faudón, *Introducción a los sistemas de bases de datos*. Pearson Educación, 7ª Ed. México 2001.
- [38] David M. Kroenke, *Procesamiento de bases de datos: fundamentos, diseño e implementación*. Pearson Educación, 8ª Ed. México 2003.
- [39] Luís Flores Gil, *Didáctica de Nuevas Tecnologías en la ESO Programación y Matemáticas VOL II*. Publicalibros.com, 2008.
- [40] Jim Boyce, *Microsoft Windows XP Power Productivity*. Sybex, 2ª Ed. Estados Unidos 2005..
- [41] Jérôme Gabillaud, *SQL Server 2008 - SQL, Transact SQL: Diseño y creación de una base de datos*. Ediciones ENI, Barcelona España 2009..
- [42] Kenneth C. Laudon, Jane Price Laudon, *Sistemas de información gerencial: administración de la empresa digital*. Pearson Educación, 8ª Ed. México 20004.
- [43] Barbara Hohensee, *Introducción A Android Studio. Incluye Proyectos Reales Y El Código Fuente*. Babelcube Inc. 2014.
- [44] J.D Gauchat, *HTML5 para Masterminds: Cómo aprovechar HTML5 para crear increíbles sitios*. MinBooks, 1ª Ed. 2014.

- [45] Maximiliano R. Firtman, *Ajax Web 2.0 con jQuery para profesionales*. Alfaomega, 2a Ed. 2011.
- [46] Pedro Muñoz Rodríguez, *Mantenimiento De Portales De Información*. Visión Libros, 1a Ed. 2013.

