



Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería Mecánica Eléctrica

**DISEÑO DEL SISTEMA DE ADQUISICIÓN DE DATOS DEL INTERCAMBIADOR  
DE CALOR DE CONCHA Y TUBOS DEL LABORATORIO DE QUÍMICA DE LA  
FACULTAD DE INGENIERÍA, USAC**

**Leonel Alejandro Echeverría González**

Asesorado por el Ing. Byron Odilio Arrivillaga Méndez

Guatemala, septiembre de 2015

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DISEÑO DEL SISTEMA DE ADQUISICIÓN DE DATOS DEL INTERCAMBIADOR  
DE CALOR DE CONCHA Y TUBOS DEL LABORATORIO DE QUÍMICA DE LA  
FACULTAD DE INGENIERÍA, USAC**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA  
POR

**LEONEL ALEJANDRO ECHEVERRÍA GONZÁLEZ**  
ASESORADO POR EL ING. BYRON ODILIO ARRIVILLAGA MÉNDEZ

AL CONFERÍRSELE EL TÍTULO DE

**INGENIERO EN ELECTRÓNICA**

GUATEMALA, SEPTIEMBRE DE 2015

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANO	Ing. Pedro Antonio Aguilar Polanco
VOCAL I	Ing. Angel Roberto Sic García
VOCAL II	Ing. Pablo Christian de León Rodríguez
VOCAL III	Inga. Elvia Miriam Ruballos Samayoa
VOCAL IV	Br. Narda Lucía Pacay Barrientos
VOCAL V	Br. Walter Rafael Véliz Muñoz
SECRETARIA	Inga. Lesbia Magalí Herrera López

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Ing. Murphy Olympto Paiz Recinos
EXAMINADOR	Ing. Byron Odilio Arrivillaga Méndez
EXAMINADOR	Ing. Carlos Eduardo Guzmán Salazar
EXAMINADOR	Ing. Julio Rolando Barrios Archila
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

## **HONORABLE TRIBUNAL EXAMINADOR**

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

**DISEÑO DEL SISTEMA DE ADQUISICIÓN DE DATOS DEL INTERCAMBIADOR  
DE CALOR DE CONCHA Y TUBOS DEL LABORATORIO DE QUÍMICA DE LA  
FACULTAD DE INGENIERÍA, USAC**

Tema que me fuera asignado por la Dirección de la Escuela de Mecánica Eléctrica, con fecha 06 de febrero de 2015.



**Leonel Alejandro Echeverría González**

Guatemala, 23 de junio de 2015

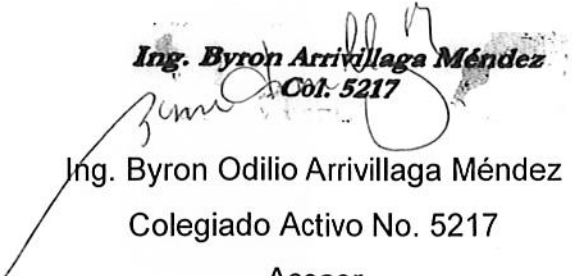
Ingeniero  
Guillermo Antonio Puentes Romero  
Director de Escuela  
Escuela de Mecánica Eléctrica  
Facultad de Ingeniería  
Presente

Estimado Ingeniero Puentes:

Por este medio quiero informarle que de acuerdo a lo estipulado en la dirección de escuela, se procedió a la asesoría y revisión del trabajo de graduación titulado: **“DISEÑO DEL SISTEMA DE ADQUISICIÓN DE DATOS DEL INTERCAMBIADOR DE CALOR DE CONCHA Y TUBOS DEL LABORATORIO DE QUÍMICA DE LA FACULTAD DE INGENIERÍA, USAC.”**, desarrollado por el estudiante Leonel Alejandro Echeverría González, identificado con carné 200915036, previo a optar al título de Ingeniero Electrónico.

Debido a lo anterior, tanto el autor como el asesor, son responsables del contenido y conclusiones del mismo y por medio de la presente me permito aprobarlo como requisito para la graduación del autor.

Atentamente,

  
**Ing. Byron Arrivillaga Méndez**  
**Cól. 5217**

Ing. Byron Odilio Arrivillaga Méndez

Colegiado Activo No. 5217

Asesor



Ref. EIME 57. 2015

Guatemala, 15 de JULIO 2015.

Señor Director

Ing. Guillermo Antonio Puente Romero  
Escuela de Ingeniería Mecánica Eléctrica  
Facultad de Ingeniería, USAC.

Señor Director:

Me permito dar aprobación al trabajo de Graduación titulado:  
DISEÑO DEL SISTEMA DE ADQUISICIÓN DE DATOS DEL  
INTERCAMBIADOR DE CALOR DE CONCHA Y TUBOS DEL  
LABORATORIO DE QUÍMICA DE LA FACULTAD DE  
INGENIERÍA, USAC, del estudiante Leonel Alejandro Echeverría  
González, que cumple con los requisitos establecidos para tal fin.

Sin otro particular, aprovecho la oportunidad para saludarle.

Atentamente,  
D Y ENSEÑAD A TODOS

Ing. Carlos Eduardo Guzmán Salazar  
Coordinador Área Electrónica



SFG



FACULTAD DE INGENIERIA

REF.EIME 57 2015.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto Bueno del Coordinador de Área, al trabajo de Graduación del estudiante; LEONEL ALEJANDRO ECHEVERRÍA GONZÁLEZ, titulado: DISEÑO DEL SISTEMA DE ADQUISICIÓN DE DATOS DEL INTERCAMBIADOR DE CALOR DE CONCHA Y TUBOS DEL LABORATORIO DE QUÍMICA DE LA FACULTAD DE INGENIERÍA, USAC, procede a la autorización del mismo.

Ing. Guillermo Antonio Puente Romero

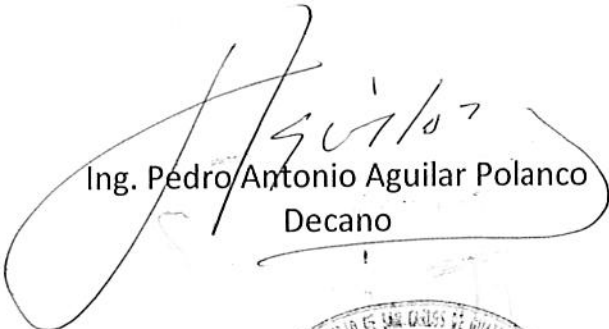


GUATEMALA, 31 DE JULIO 2,015.



El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al Trabajo de Graduación titulado: **DISEÑO DEL SISTEMA DE ADQUISICIÓN DE DATOS DEL INTERCAMBIADOR DE CALOR DE CONCHA Y TUBOS DEL LABORATORIO DE QUÍMICA DE LA FACULTAD DE INGENIERÍA, USAC**, presentado por el estudiante universitario: **Leonel Alejandro Echeverría González**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:

  
Ing. Pedro Antonio Aguilar Polanco  
Decano

Guatemala, septiembre de 2015

/gdech





# ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES .....	V
LISTA DE SÍMBOLOS .....	VII
GLOSARIO .....	IX
RESUMEN .....	XV
OBJETIVOS.....	XVII
INTRODUCCIÓN .....	XIX
1. MARCO TEÓRICO .....	1
1.1. Fundamentos de un sistema de adquisición de datos .....	1
1.1.1. Conversión de señales físicas a señales eléctricas .....	1
1.1.1.1. Transductores .....	2
1.1.2. Conversión analógica a digital .....	2
1.1.2.1. Muestreo .....	3
1.1.2.2. Cuantización .....	4
1.1.2.3. Codificación.....	4
1.1.3. Procesamiento de datos .....	5
1.1.3.1. Transmisión de datos .....	5
1.1.3.2. Almacenamiento de datos.....	6
1.1.3.3. Presentación de datos.....	6
1.2. Componentes del diseño del sistema de adquisición de datos .....	6
1.2.1. Circuito integrado DS18B20 .....	6
1.2.1.1. Protocolo de comunicación 1-Wire.....	7
1.2.2. Microcontrolador.....	9
1.2.2.1. Arduino Nano .....	10
1.2.3. Módulo Bluetooth HC-05 .....	10

1.2.3.1.	Comunicación serial TTL.....	11
1.2.3.2.	Protocolo de comunicación Bluetooth .....	11
2.	GENERALIDADES DEL DISEÑO DEL SISTEMA DE ADQUISICIÓN DE DATOS .....	13
2.1.	Funcionamiento del intercambiador de calor de concha y tubos ...	13
2.2.	Diseño del sistema de adquisición de datos .....	15
2.2.1.	Módulo de sensores.....	16
2.2.2.	Unidad de control de sensores.....	16
2.2.3.	Módulo de comunicación Bluetooth .....	16
2.2.4.	Dispositivo de almacenamiento y presentación .....	16
3.	DISEÑO Y ESPECIFICACIONES DE HARDWARE.....	17
3.1.	Módulo de sensores .....	17
3.2.	Unidad de control de sensores .....	18
3.3.	Módulo de comunicación Bluetooth .....	20
3.4.	Fuente de alimentación del sistema de adquisición de datos.....	20
4.	DISEÑO Y ESPECIFICACIONES DE SOFTWARE .....	21
4.1.	Desarrollo para Arduino Nano.....	21
4.1.1.	Librerías para Arduino.....	21
4.1.1.1.	Librería OneWire .....	22
4.1.1.2.	Librería Dallas Temperature.....	22
4.2.	Desarrollo para PC .....	24
4.2.1.	Lenguaje de programación Visual Basic .....	24
4.2.2.	Especificaciones de objetos en el código fuente.....	24
4.2.2.1.	Objeto de puerto serial .....	25
4.2.2.2.	Objeto cuadro de diálogo grabar .....	25
4.2.2.3.	Objeto vista cuadrícula de datos .....	26

4.2.3.	Almacenamiento de datos.....	28
4.2.3.1.	Archivo de valores separados por coma .....	28
4.3.	Configuración y utilización del software para usuario final.....	29
4.3.1.	Instalación y emparejamiento del dispositivo Bluetooth ..	29
4.3.2.	Conexión de la aplicación .....	34
4.3.3.	Uso de la aplicación .....	37
4.3.3.1.	Lectura de datos.....	37
4.3.3.2.	Eliminación de datos .....	38
4.3.3.3.	Guardar datos .....	39
CONCLUSIONES .....		41
RECOMENDACIONES .....		43
BIBLIOGRAFÍA .....		45
APÉNDICE.....		47
ANEXOS.....		53



# ÍNDICE DE ILUSTRACIONES

## FIGURAS

1.	Ejemplo de señal analógica .....	2
2.	Ejemplo de muestreo .....	3
3.	Ejemplo de cuantización y codificación.....	4
4.	Circuito integrado DS18B20 encapsulado metálico .....	7
5.	Esquema de una MicroLAN .....	9
6.	Arduino Nano.....	10
7.	Módulo Bluetooth HC-05.....	11
8.	Termómetros del intercambiador de calor de concha y tubos .....	14
9.	Módulos del diseño del sistema de adquisición de datos .....	15
10.	Diagrama esquemático módulo de sensores.....	17
11.	Placa de circuito impreso de sensores DS18B20 .....	18
12.	Placa de circuito impreso Arduino Nano .....	19
13.	Pines de conexión para módulo Bluetooth HC-05 .....	20
14.	Propiedades del objeto de puerto serial.....	25
15.	Propiedades del cuadro de diálogo grabar .....	26
16.	Propiedades del objeto vista cuadrícula de datos.....	27
17.	Dispositivos e impresoras .....	29
18.	Agregar un dispositivo .....	30
19.	Buscando dispositivos .....	31
20.	Seleccionar dispositivo HC-05 .....	31
21.	Escribir código de emparejamiento, paso 1 .....	32
22.	Escribir código de emparejamiento, paso 2 .....	33
23.	Notificación de instalación del dispositivo .....	33

24.	Instalación del software de controlador.....	34
25.	Ventana del sistema de adquisición de datos .....	35
26.	Selección de puerto serial.....	35
27.	Conectar a puerto serial virtual .....	36
28.	Conexión exitosa a puerto serial virtual .....	36
29.	Lectura de datos .....	37
30.	Borrar selección.....	38
31.	Guardar datos.....	39

## **TABLAS**

I.	Pines de conexión de Arduino Nano .....	18
----	---	----

## LISTA DE SÍMBOLOS

<b>Símbolo</b>	<b>Significado</b>
<b>A</b>	Amperio
<b>U</b>	Circuito Integrado
<b>GHz</b>	Gigahertz
<b>°C</b>	Grados Celsius
<b>Hz</b>	Hertz
<b>KB</b>	Kilobytes
<b>®</b>	Marca registrada
<b>MHz</b>	Megahertz
<b>mA</b>	Miliamperio
<b>Ω</b>	Ohm
<b>Q</b>	Quetzales (moneda guatemalteca)
<b>Rx</b>	Receptor
<b>R</b>	Resistencia
<b>GND</b>	Tierra o voltaje de referencia
<b>Tx</b>	Transmisor
<b>V</b>	Volt
<b>Vdd</b>	Voltaje de alimentación positivo





## GLOSARIO

<b>Archivo CSV</b>	<p>(<i>Comma-separated values</i>) es un tipo de documento en formato abierto sencillo para representar datos en forma de tabla, en las que las columnas se separan por comas y las filas por un caracter de cambio de línea.</p>
<b>Arduino</b>	<p>Es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.</p>
<b>Banda ISM</b>	<p>(<i>Industrial, Scientific and Medical</i>) son bandas reservadas internacionalmente para uso no comercial de radiofrecuencia electromagnética en áreas industrial, científica y médica.</p>
<b>Baudio</b>	<p>Es una unidad de velocidad de transmisión de datos utilizada en comunicaciones que denota la cantidad de símbolos transmitidos en un intervalo de tiempo, un baudio puede contener varios bits.</p>
<b>Bit</b>	<p>Es la unidad mínima de información que se almacena en memoria o es transmitida y únicamente puede tener dos estados lógicos, cero o uno.</p>

<b>Bluetooth</b>	Es una especificación industrial para redes inalámbricas de área personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos, mediante un enlace por radiofrecuencia en la banda ISM de los 2,4 GHz.
<b>Bootloader</b>	Es un programa sencillo (que no tiene la totalidad de las funcionalidades de un sistema operativo) diseñado exclusivamente para preparar todo lo que necesita el sistema operativo para funcionar.
<b>Bus de datos</b>	Es un sistema digital que transfiere datos entre componentes electrónicos. Está formado por cables o pistas en un circuito impreso, dispositivos como resistores y condensadores además de circuitos integrados.
<b>Circuito integrado</b>	También conocido como chip o microchip. Es una estructura de pequeñas dimensiones de material semiconductor, de algunos milímetros cuadrados de área, sobre la que se fabrican circuitos electrónicos.
<b>EEPROM</b>	Son las siglas de <i>Electrically Erasable Programmable Read-Only Memory</i> (ROM programable y borrable eléctricamente). Es un tipo de memoria ROM que puede ser programada, borrada y reprogramada eléctricamente.

<b>Memoria flash</b>	Es derivada de la memoria EEPROM, permite la lectura y escritura de múltiples posiciones de memoria en la misma operación. Gracias a ello, la tecnología flash, siempre mediante impulsos eléctricos, permite velocidades de funcionamiento muy superiores frente a la tecnología EEPROM.
<b>MicroUSB</b>	Es un tipo de conector del estándar USB, que permite la alimentación eléctrica y transmisión de datos en dispositivos electrónicos.
<b>Placa de circuito impreso</b>	Es la superficie constituida por caminos, pistas o buses de material conductor laminadas sobre una base no conductora. El circuito impreso se utiliza para conectar eléctricamente a través de los caminos conductores, y sostener mecánicamente, por medio de la base, un conjunto de componentes electrónicos.
<b>RAM</b>	Siglas en inglés de <i>Random Access Memory</i> (significa Memoria de Acceso Aleatorio) es la memoria donde se cargan las instrucciones que ejecuta el procesador o cualquier unidad de cómputo, en el tiempo real.
<b>ROM</b>	Siglas en inglés de <i>Read-Only Memory</i> , es un tipo de memoria de solo lectura.

<b>Resistencia <i>pull-up</i></b>	Resistencia de polarización conectada a un voltaje positivo para elevar la tensión de entrada para un dispositivo digital.
<b>RS-232</b>	Es una interfaz que designa una norma para el intercambio de una serie de datos binarios entre un DTE ( <i>Data Terminal Equipment</i> , Equipo Terminal de Datos) y un DCE ( <i>Data Communication Equipment</i> , Equipo de Comunicación de Datos).
<b>Sensor</b>	Dispositivo capaz de transformar un determinado tipo de energía de entrada a una señal eléctrica de salida.
<b>SRAM</b>	SRAM son las siglas de la voz inglesa <i>Static Random Access Memory</i> , que significa memoria estática de acceso aleatorio (o RAM estática). Denomina a un tipo de tecnología de memoria RAM basada en semiconductores, capaz de mantener los datos, mientras siga alimentada, sin necesidad de circuito de refresco.
<b>UHF</b>	Siglas del inglés <i>Ultra High Frequency</i> , frecuencia ultra alta. Es una banda del espectro electromagnético que ocupa el rango de frecuencias de 300 MHz a 3 GHz.

## **USB**

El bus universal en serie (en inglés: *Universal Serial Bus*), es un bus estándar industrial que define los cables, conectores y protocolos usados en un bus para conectar, comunicar y proveer de alimentación eléctrica entre computadoras, periféricos y dispositivos electrónicos.



## RESUMEN

En la actualidad, los avances en el equipo electrónico han hecho posible la optimización de procedimientos que requieren de tiempo y esfuerzo de personas. Uno de los lugares beneficiados con dichos avances tecnológicos es el campo de la educación, específicamente los laboratorios, ya que poseen equipo en el que se toman mediciones manualmente, para su posterior análisis.

En el primer capítulo se presenta una descripción del sistema de adquisición de datos. También un detalle de las tres partes que lo componen: la conversión de señales físicas a señales eléctricas, la conversión analógica a digital y el procesamiento de los datos. Asimismo, se plantean las generalidades de los componentes que integran el diseño del sistema de adquisición de datos.

Dependiendo de las necesidades de los lugares en donde se implementa un sistema de adquisición de datos, se desarrollan distintos tipos de soluciones, es por eso que en el segundo capítulo, se detalla el funcionamiento del intercambiador de calor de concha y tubos. Además de las generalidades del diseño del sistema de adquisición de datos de temperatura, el cual se hizo de forma modular para su versatilidad, escalabilidad y facilidad de mantenimiento. También se describe la funcionalidad de cada módulo y los componentes que lo conforman.

Después de proponer el diseño general, en el tercer capítulo se presentan los diseños y especificaciones de hardware, con los diagramas de circuitería, las placas de circuito impreso e información de la configuración en los dispositivos y

circuitería. También se detallan las formas de interconexión de los cuatro módulos y los pines ocupados correspondientes a los dispositivos utilizados.

En el cuarto capítulo se describe el diseño y las especificaciones del software, esto incluye las herramientas de desarrollo junto con las ventajas que justifican su utilización. Asimismo, se presentan detalles de implementación de librerías con información sobre las funciones relevantes, detalles de los objetos instanciados y propiedades de los mismos, tanto del desarrollo para el microcontrolador Arduino Nano como para el software de PC.

Para finalizar, el capítulo cuatro plantea una sección dedicada a los pasos necesarios para la configuración y utilización del software para usuarios finales, con capturas de pantalla y procedimientos detallados. Esto para el óptimo desempeño del software, que manipulan los usuarios finales del sistema de adquisición de datos.



# OBJETIVOS

## General

Diseñar un sistema de adquisición de datos para el intercambiador de calor de calor de concha y tubos del Laboratorio de Química de la Facultad de Ingeniería de la Usac.

## Específicos

1. Proponer el diseño de un sistema de adquisición de datos.
2. Presentar las generalidades del diseño del sistema de adquisición de datos del intercambiador de calor de concha y tubos.
3. Especificar las conexiones de hardware del sistema de adquisición de datos propuesto.
4. Detallar la implementación y desarrollo de software necesarios en el diseño del sistema de adquisición de datos del intercambiador de calor de concha y tubos.



## INTRODUCCIÓN

En el mundo contemporáneo es necesario realizar una gran variedad de procesos en los que se requiere de la toma de mediciones y adquisición de datos para su posterior procesamiento.

Con los avances tecnológicos y la subsecuente minimización de sus costos, es cada vez más fácil y asequible la implementación de sistemas electrónicos en distintos aspectos de la vida cotidiana. Tal es el caso del ámbito académico, en el cual se deben de mantener estándares de educación congruentes con estos avances.

La tecnología permite la optimización de procesos que son repetitivos y propensos a errores humanos. Un ejemplo es la toma de mediciones en los laboratorios, que se complementa y beneficia ampliamente de la automatización con sistemas electrónicos. Estos permiten que el tiempo que consume la adquisición de datos, así como también las equivocaciones debidas al factor humano se reduzcan considerablemente.

El presente trabajo de graduación propone el diseño del sistema de adquisición de datos de temperatura del intercambiador de calor de concha y tubos, uno de los equipos del Laboratorio de Química de la Facultad de Ingeniería de la Usac. El cual consiste de catorce sensores de temperatura digitales que se comunican y operan desde un microcontrolador, que envía los datos para su almacenamiento y análisis posterior en un archivo estándar de valores separados por comas.



# **1. MARCO TEÓRICO**

## **1.1. Fundamentos de un sistema de adquisición de datos**

Un sistema de adquisición de datos es un proceso por el que se recolectan datos para documentación o análisis de algún fenómeno. En su forma más simple una persona que, por ejemplo, escribe en una hoja de papel las mediciones de un termómetro, está realizando un proceso de adquisición de datos.

Conforme la tecnología ha avanzado, este procedimiento se ha hecho más preciso, versátil y confiable a través de equipo electrónico. Este permite realizar la tarea en menos tiempo y reduce la cantidad de errores en comparación con la recopilación de datos en forma manual.

Los sistemas de adquisición de datos electrónicos están compuestos por tres subsistemas, que son: la conversión de señales físicas a señales eléctricas, la conversión analógica a digital y el procesamiento de datos.

### **1.1.1. Conversión de señales físicas a señales eléctricas**

Para que sea posible la manipulación y procesamiento de los datos en un sistema electrónico, es necesario que las magnitudes del fenómeno físico de interés se conviertan en señales eléctricas, como voltaje o corriente. Los encargados de realizar esta conversión son los transductores.

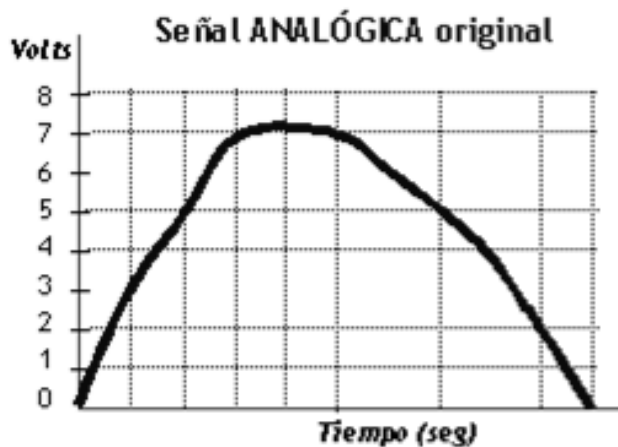
### 1.1.1.1. Transductores

Existen distintos tipos de transductores utilizados para detectar un amplio rango de fenómenos físicos como el movimiento, energía térmica, energía magnética, energía mecánica, energía eléctrica, entre otras. Ellos se encargan de transformar un tipo de energía a otro, en este caso a señales eléctricas para ser utilizadas por el equipo electrónico.

### 1.1.2. Conversión analógica a digital

Las señales eléctricas provenientes de los transductores, que pueden ser señales de voltaje o de corriente, son continuas en el tiempo y en amplitud, es decir que son análogas, véase la figura 1. Por lo tanto aún no pueden ser procesadas, manipuladas o almacenadas por el equipo electrónico digital.

Figura 1. Ejemplo de señal analógica



Fuente: MARTÍNEZ, Evelio. *Conversión analógico-digital*. [www.eveliux.com](http://www.eveliux.com). Consulta: 19 de febrero de 2015.

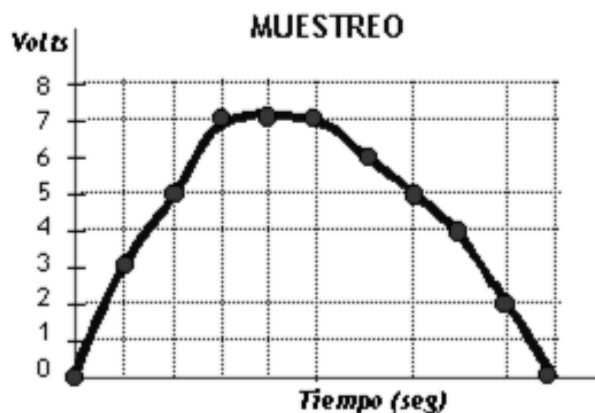
La conversión de señales análogas a señales digitales, es decir, a señales que son discretas en tiempo y amplitud, hace más fácil y eficiente el almacenamiento, procesamiento y manipulación de las señales en general. Por ello, resulta de gran utilidad en los sistemas de adquisición de datos.

Para realizar esta conversión se necesita de un conversor analógico a digital, el cual se compone de tres etapas: el muestreo, la cuantización y la codificación.

### 1.1.2.1. Muestreo

Se denomina etapa de muestreo, cuando se mide la amplitud de la señal analógica en intervalos de tiempo definidos, es decir que se toma una muestra de la señal en periodos de tiempo constantes. Al número de muestras por segundo se le denomina tasa de muestreo, y su unidad de medida son los hertz (Hz). Al finalizar la etapa de muestreo, la señal es discreta en el tiempo y continua en amplitud, véase la figura 2.

Figura 2. Ejemplo de muestreo

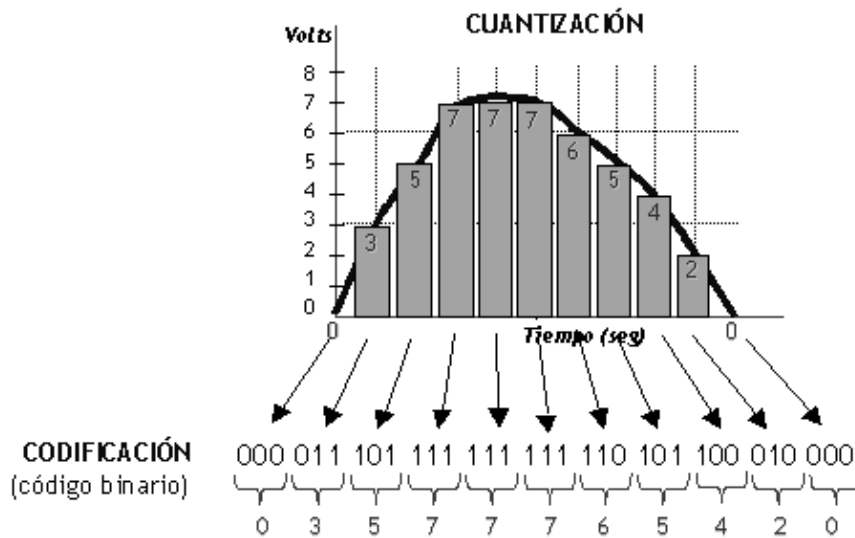


Fuente: MARTÍNEZ, Evelio. *Conversión analógico-digital*. [www.eveliux.com](http://www.eveliux.com). Consulta: 19 de febrero de 2015.

### 1.1.2.2. Cuantización

La cuantización es el proceso de convertir las amplitudes de la señal continua en amplitudes discretas. Esto quiere decir que mientras el muestreo representa el tiempo de captura de la señal, la cuantización es el componente de amplitud del muestreo. En otras palabras, el muestreo mide el tiempo (el número de muestras por segundo) y la cuantización mide la amplitud en forma discreta, es decir que se tienen niveles de cuantización a los que se aproxima la señal continua en amplitud, véase la figura 3.

Figura 3. Ejemplo de cuantización y codificación



Fuente: MARTÍNEZ, Evelio. *Conversión analógico-digital*. www.eveliux.com. Consulta: 19 de febrero de 2015.

### 1.1.2.3. Codificación

La codificación es el proceso donde se le asigna a cada nivel de cuantización un código (por ejemplo el código binario, como en el ejemplo de la figura 3).



El número de niveles de cuantización disponibles en la codificación binaria, está dado por  $2^N$  donde N es el número de bits utilizados en la codificación. En el ejemplo de la figura 3, se aprecia una codificación binaria de 3 bits, por lo que tiene un número de niveles de cuantización de  $2^3 = 8$ .

Al número de bits por muestra de la señal se le denomina longitud de palabra, mientras más grande la longitud de palabra, mejor es la representación de la señal original en su forma digital.

### **1.1.3. Procesamiento de datos**

Dentro de un sistema de adquisición de datos, la parte del procesamiento de datos es la más importante. Esta permite el almacenamiento, el análisis, la presentación y todos los tratamientos que sean requeridos para la información recopilada en el sistema.

Es por esto que existe una gran variedad de técnicas, instrumentos y dispositivos para el procesamiento de los datos. Dentro de los procedimientos más comunes para el tratamiento de los datos se encuentran los enumerados a continuación.

#### **1.1.3.1. Transmisión de datos**

La transmisión de los datos, a través de un canal de comunicaciones, es uno de los procesos más comunes dentro de un sistema de adquisición de datos. Esto es debido a que permite separar física y lógicamente el equipo que se encarga de adquirir los datos del equipo encargado de procesarlos.

### **1.1.3.2. Almacenamiento de datos**

El almacenamiento de los datos brinda a los sistemas la capacidad de la utilización de los datos tiempo después de haberlos adquirido. Esto genera bases de datos, hace copias de respaldo, entre otras.

### **1.1.3.3. Presentación de datos**

Permite a los sistemas mostrar los datos a los usuarios finales, hacer interfaces de comunicación para la interconexión de sistemas, y otros.

Estos tipos de procesamiento de datos pueden ser utilizados individualmente o si es necesario, dependiendo de los requerimientos de la aplicación, hacer una combinación de los mismos.

## **1.2. Componentes del diseño del sistema de adquisición de datos**

A continuación se detallan las características generales y las especificaciones técnicas más importantes de los componentes utilizados en el sistema de adquisición de datos.

### **1.2.1. Circuito integrado DS18B20**

Contiene un sensor que procesa las magnitudes de temperatura, realiza la conversión analógica a digital y las envía por medio de comunicación 1-Wire a un microcontrolador centralizado. Es configurable para muestrear temperaturas desde 9 hasta 12 bits en grados Celsius. Contiene funcionalidad de alarma con puntos de referencia superior e inferior, con memoria no volátil y programable por el usuario. Tiene un rango de temperatura de -55 °C a +125 °C y una

exactitud de  $\pm 0,5$  °C sobre el rango de -10 °C a +85 °C. También puede utilizar como fuente de alimentación la línea de datos, eliminando la necesidad de otra fuente de poder.

El encapsulado del circuito integrado DS18B20 utilizado en el diseño del sistema de adquisición de datos es una sonda de tipo metálico, hecha de acero inoxidable ideal para aplicaciones en las que se requiera resistencia a la humedad, véase la figura 4.

Figura 4. **Circuito integrado DS18B20 encapsulado metálico**



Fuente: *DS18B20 encapsulado metálico*. [www.abcelectronica.net](http://www.abcelectronica.net). Consulta: 5 de marzo de 2015.

#### **1.2.1.1. Protocolo de comunicación 1-Wire**

1-Wire es un sistema de bus de comunicaciones para dispositivos electrónicos, diseñado por Dallas Semiconductor que provee transmisiones de datos a bajas velocidades, señalización y alimentación a través de un solo cable. 1-Wire es similar en concepto a I2C, pero con velocidades más bajas y menor rango. Es típicamente usado para comunicarse con dispositivos pequeños y de bajo costo, tales como termómetros digitales e instrumentación

climatológica. Una red de dispositivos 1-Wire con un dispositivo maestro asociado es llamada una MicroLAN.

Una característica distintiva del bus es la posibilidad de usar solo dos cables: el de datos y el de tierra. Para lograr eso, los dispositivos 1-Wire incluyen un capacitor de 800 pF para almacenar carga, y para encender el dispositivo durante los periodos donde la línea de datos está activa.

En cualquier MicroLAN existe siempre un dispositivo maestro encargado de todo, puede ser una PC o un microcontrolador. El maestro inicia actividades en el bus de datos, simplificando y evitando las colisiones dentro del bus. Protocolos implementados por software son los que detectan colisiones. Después de una colisión, el maestro intenta nuevamente la comunicación requerida.

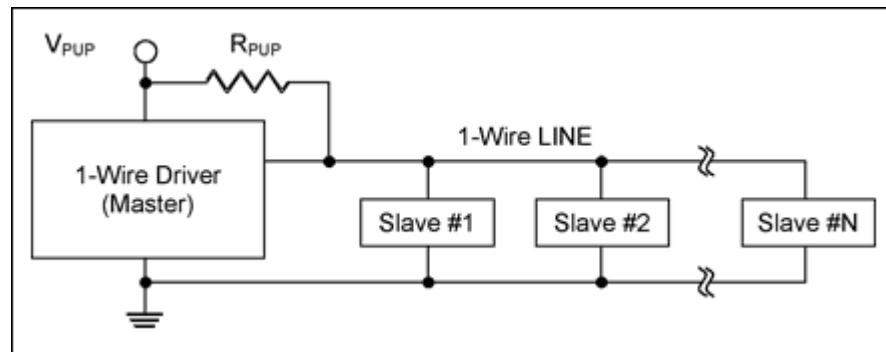
Muchos dispositivos pueden compartir el mismo bus. Cada dispositivo en el bus tiene un número serial único de 64 bits. El byte menos significativo del número serial es un número de 8 bits que indica el tipo de dispositivo. El byte más significativo es un código cíclico de detección de errores estándar para el bus 1-Wire.

Para identificar a los dispositivos, el maestro emite un comando de enumeración, y luego una dirección, escuchando después de cada bit de una dirección. Si un esclavo tiene todos los bits de la dirección hasta el momento, devuelve un 0. El maestro usa este comportamiento simple para buscar sistemáticamente por secuencias válidas de bits.

El proceso es mucho más rápido que una búsqueda con fuerza bruta para todos los números posibles con 64 bits, porque al momento que un bit inválido

es detectado, se sabe que todos los bits de direcciones siguientes son inválidos. En la figura 5 se puede ver el diagrama esquemático típico de una MicroLAN.

Figura 5. **Esquema de una MicroLAN**



Fuente: *1-Wire*. [www.maximintegrated.com](http://www.maximintegrated.com). Consulta: 14 de marzo de 2015.

Cada DS18B20 tiene un código serial único de 64 bits, que permite funcionar a múltiples DS18B20 en el mismo bus 1-Wire. Esto simplifica el uso de un solo microprocesador para controlar varios DS18B20 distribuidos sobre un área grande.

### 1.2.2. **Microcontrolador**

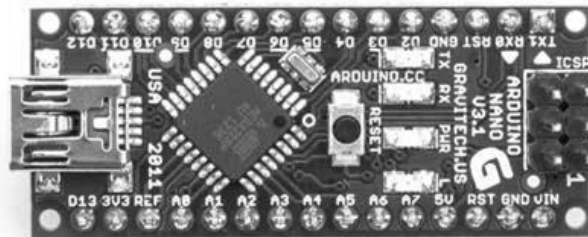
Es un componente electrónico que posee todas las características de un computador a pequeña escala, contenidas en un solo circuito integrado. Es decir, tiene un núcleo con procesador, memoria volátil, memoria no volátil y periféricos de entrada/salida programables.

Son dispositivos de bajo costo, que permiten realizar funciones que no requieren de grandes cantidades de procesamiento o memoria y hacen posible el diseño de sistemas embebidos confiables y de dimensiones pequeñas.

### 1.2.2.1. Arduino Nano

Es una placa de desarrollo, compacta y completa que está basada en el microcontrolador ATmega328, véase figura 6. El voltaje de entrada del dispositivo es de 6 a 20 V. Voltaje de operación de 5 V. Posee 14 pines de entrada y salida, con corriente máxima de 40 mA por pin. Memoria Flash de 32 KB de los que se utilizan 2 KB para el *bootloader*. SRAM de 2 KB. EEPROM de 1 KB. Reloj de 16 Mhz.

Figura 6. Arduino Nano



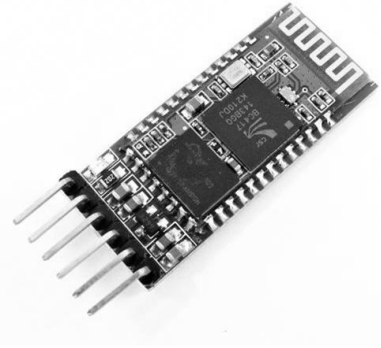
Fuente: *Arduino Nano*. [www.arduino.cc](http://www.arduino.cc). Consulta: 18 de marzo de 2015.

### 1.2.3. Módulo Bluetooth HC-05

Es un módulo de comunicaciones que se encarga de la conversión del protocolo serial con niveles de voltaje TTL, a un canal de comunicaciones Bluetooth. Este módulo posee dos modos de uso: maestro y esclavo. La utilización de este dispositivo hace posible la transmisión de datos de una forma modular, inalámbrica, sencilla y económica en los sistemas en los que se utiliza.

Las velocidades en baudios soportadas por el módulo son las siguientes: 9600, 19200, 38400, 57600, 115200, 230400, 460800. Ver figura 7.

Figura 7. **Módulo Bluetooth HC-05**



Fuente: *Módulo Bluetooth master / slave (HC-05)*. [www.botscience.net](http://www.botscience.net). Consulta: 26 de marzo de 2015.

### **1.2.3.1. Comunicación serial TTL**

La comunicación serial consiste en el envío de los datos un bit a la vez, secuencialmente, a través de un canal de comunicaciones. Los niveles de voltaje TTL o lógica transistor-transistor se encuentran entre 0 V, para un 0 lógico y Vcc para un 1 lógico. Aquí el Vcc puede ser 3,3 V o 5 V dependiendo de los dispositivos.

### **1.2.3.2. Protocolo de comunicaciones Bluetooth**

Bluetooth es un estándar de tecnología inalámbrica para comunicación en distancias cortas. Utiliza ondas de radio con longitud de onda corta UHF en la banda ISM de los 2,4 a los 2,485 GHz. Es utilizado en una gran variedad de dispositivos, desde computadoras personales, teléfonos móviles, hasta sistemas embebidos. Fue creado originalmente como una alternativa a los cables de datos del protocolo de comunicación RS-232.





## **2. GENERALIDADES DEL DISEÑO DEL SISTEMA DE ADQUISICIÓN DE DATOS**

En el Laboratorio de Química de la Facultad de Ingeniería de la Usac se tiene el intercambiador de calor de concha y tubos, el cual es un equipo de laboratorio utilizado en las prácticas de los estudiantes de la Facultad. El objetivo de las prácticas es tomar mediciones de temperatura de dos líquidos en 14 puntos del recorrido de los mismos. Estas mediciones se realizan periódicamente durante el transcurso de la práctica y se recopilan los datos manualmente.

### **2.1. Funcionamiento del intercambiador de calor de concha y tubos**

El proceso inicia tomando agua de una pileta por medio de tuberías, estas llevan el líquido hasta el intercambiador de calor. Una tubería es de agua fría o temperatura ambiente y la otra pasa por un proceso de condensado donde se calienta a la temperatura determinada por el operario.

Posteriormente el agua fría pasa por un sistema donde se aumenta la presión a un valor estipulado y luego circula dentro del tubo de mayor radio. Esto en cada uno de los tubos del intercambiador, siendo posible seleccionar en qué momento trabaja cada tubo o si trabajan en serie dos o más. El agua caliente pasa por un sistema de presión similar y circula hacia los tubos de menor radio, que se encuentran dentro del tubo de radio mayor, pudiéndose seleccionar de igual manera qué tubo del intercambiador de calor es el que trabaja.

El agua a temperatura ambiente que circula por el tubo mayor, enfría los tubos menores y así se genera el intercambio de calor.

Existe un termómetro en la entrada del agua caliente hacia los tubos menores y uno a la salida. Esto con el objetivo de verificar la reducción de calor por medio del agua fría en el tubo de radio mayor. Debido a que se tienen cinco tubos se adquieren en total diez termómetros en esta área. Véase figura 8.

Figura 8. **Termómetros del intercambiador de calor de concha y tubos**



Fuente: Laboratorio de Química de la Facultad de Ingeniería.

Asimismo, existen dos termómetros en la tubería de agua fría uno en la entrada y el otro en la salida, para verificar que el agua entrante se calienta debido al intercambio de calor. Se tienen dos termómetros más para la temperatura del área del cálculo de la presión.

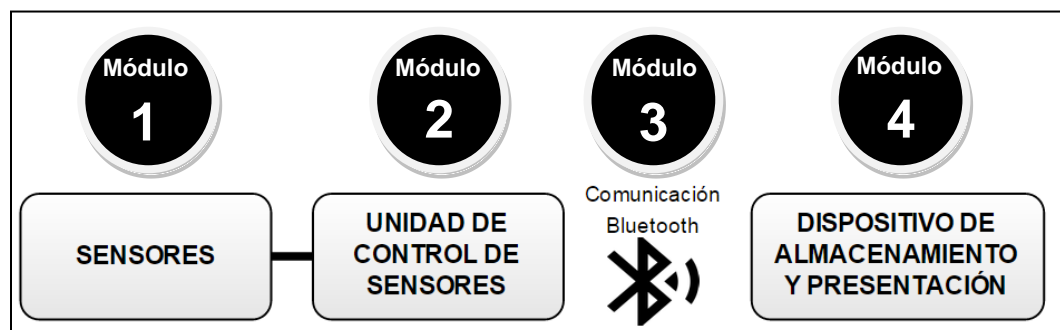
Los rangos de temperatura en los que trabaja el intercambiador van desde 0 °C hasta los 100 °C. La temperatura está dada por el aparato de medición y este es un termómetro de mercurio con resolución de 1 °C.

## 2.2. Diseño del sistema de adquisición de datos

Se plantea el diseño de un sistema de adquisición de datos. Este permite minimizar el tiempo y cantidad de trabajo para obtener las mediciones de temperatura correspondientes del intercambiador de calor de concha y tubos.

El diseño está desarrollado de forma modular permitiendo facilitar su mantenimiento preventivo y correctivo. Esto a través del tiempo, consistiendo en cuatro módulos, véase figura 9.

Figura 9. Módulos del diseño del sistema de adquisición de datos



Fuente: elaboración propia, con programa Microsoft Word.

### **2.2.1. Módulo de sensores**

En este módulo se localizan los 14 sensores de temperatura digitales DS18B20. Estos recopilan las mediciones correspondientes a los diferentes puntos del intercambiador de calor de concha y tubos. Los envían a la unidad de control de sensores por medio de un bus de datos que utiliza el protocolo de comunicación serial 1-Wire.

### **2.2.2. Unidad de control de sensores**

Este módulo es el encargado de controlar el flujo de información en el bus de datos de los sensores. También recibe comandos para iniciar las mediciones y luego envía los datos por comunicación serial TTL al módulo de comunicación Bluetooth. En esta etapa se utiliza la placa de desarrollo Arduino Nano como controlador.

### **2.2.3. Módulo de comunicación Bluetooth**

Este módulo es el encargado de hacer la conversión de la comunicación serial TTL, de la unidad de control de sensores a comunicación inalámbrica Bluetooth. El módulo Bluetooth HC-05 es utilizado en esta etapa.

### **2.2.4. Dispositivo de almacenamiento y presentación**

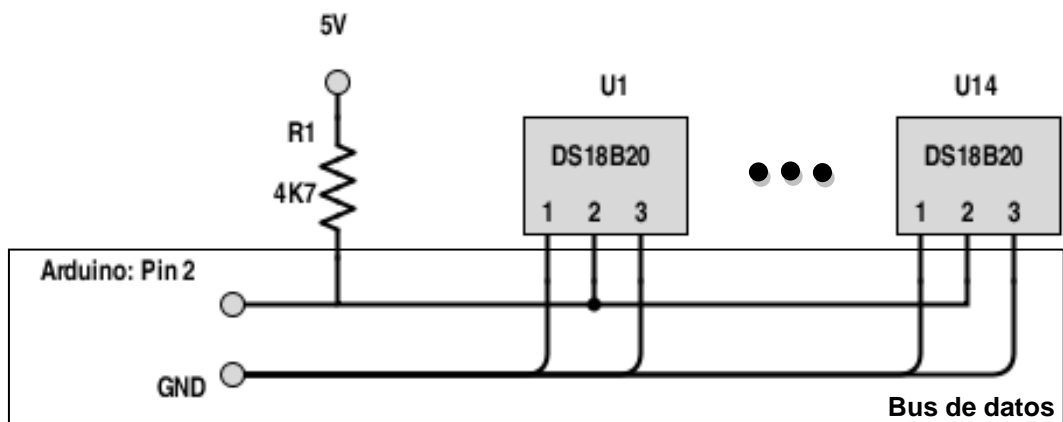
Este módulo es el encargado de enviar los comandos de iniciación de toma de mediciones y recibir información por medio de la comunicación Bluetooth. Al recibir los datos los almacena, los presenta al usuario y brinda opciones de exportación para su posterior procesamiento. En esta etapa, el encargado es el software desarrollado para PC.

### 3. DISEÑO Y ESPECIFICACIONES DE HARDWARE

#### 3.1. Módulo de sensores

Los sensores DS18B20 utilizan la configuración predeterminada para una precisión de 9 bits. También se habilita el modo de alimentación parásita, enviando los pines 1 y 3 a tierra y colocando una resistencia de *pull-up* de 4 700  $\Omega$  a 5 V y al pin 2 del Arduino Nano y pin 2 de los 14 sensores DS18B20, véase la figura 10.

Figura 10. Diagrama esquemático módulo de sensores



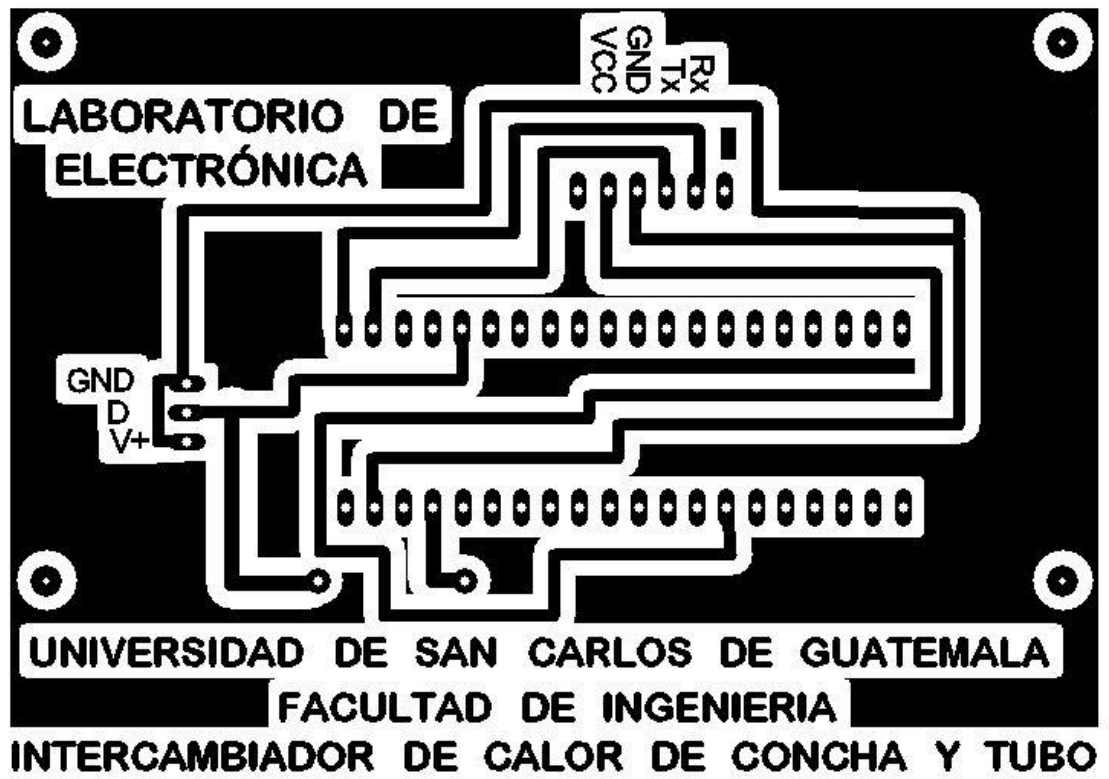
Fuente: elaboración propia, empleando Scheme-it.

El bus de datos incluye la placa de circuito impreso de sensores DS18B20, véase la figura 11, y el cableado de los mismos.



La unidad de control de sensores incluye también la placa de circuito impreso que integra al Arduino Nano, con el módulo Bluetooth HC-05 y el bus de datos del módulo de sensores, véase la figura 12.

Figura 12. Placa de circuito impreso Arduino Nano



Fuente: Equipo de electrónica aplicada 1.

En la parte superior, de la figura 11, se aprecian los conectores para el módulo Bluetooth HC-05. En la misma figura, del lado izquierdo, se encuentran los conectores para el bus de datos del módulo de sensores. Y en la parte central se encuentran los conectores para la placa de desarrollo Arduino Nano.

### 3.3. Módulo de comunicación Bluetooth

Está compuesto por el circuito integrado Bluetooth HC-05. Configurado como dispositivo Bluetooth esclavo, con el pin de emparejamiento: 1234. Para la comunicación serial TTL se configuró a 9 600 baudios. Véase la figura 13, para los pines de conexión al diseño.

Figura 13. Pines de conexión para módulo Bluetooth HC-05



Fuente: *HC-05 Bluetooth Module*. [www.fabtolab.com](http://www.fabtolab.com). Consulta: 3 de abril de 2015.

### 3.4. Fuente de alimentación del sistema de adquisición de datos

El sistema de adquisición de datos es alimentado a través del conector hembra microUSB de la placa de desarrollo Arduino Nano, por una fuente con conector macho microUSB estándar de 5 V y 1,5 A.

Los costos del equipo y materiales necesarios para la implementación del diseño están detallados en el anexo I.



## **4. DISEÑO Y ESPECIFICACIONES DE SOFTWARE**

El software necesario para el correcto funcionamiento del sistema de adquisición de datos incluye el desarrollo para el microcontrolador Arduino Nano. Esto es la unidad de control de sensores; y el desarrollo para PC, que es el dispositivo de presentación y almacenamiento.

### **4.1. Desarrollo para Arduino Nano**

El desarrollo del software para Arduino Nano se hace con el Ambiente de Desarrollo de Arduino. Se incluye un editor de texto para escribir código, un área para mensajes, una consola de texto, una barra de herramientas con botones para las funciones más comunes y una serie de menús para el resto de funciones. Esta herramienta se conecta al hardware de Arduino para subir los programas desarrollados y comunicarse con ellos.

#### **4.1.1. Librerías para Arduino**

Las librerías proveen funcionalidad extra para utilizar en los programas para Arduino. Con esto se facilita el trabajar con hardware y datos, con funciones y códigos comunes y reutilizables plasmados dentro de la estructura de la librería. Esto permite hacer el código del programa final más simple, ordenado, legible e incluso más fácil de optimizar, actualizar y modificar en futuros mantenimientos correctivos y preventivos.

Las librerías utilizadas para la programación de la placa de desarrollo Arduino Nano, en el diseño del sistema de adquisición de datos del intercambiador de calor de concha y tubos, se enumeran y describen a continuación.

#### **4.1.1.1. Librería OneWire**

Esta permite utilizar el protocolo de comunicación 1-Wire de una forma sencilla y rápida, utilizando uno o más pines de entrada/salida como buses de datos 1-Wire.

Los requerimientos de hardware para el correcto funcionamiento de esta librería son: una resistencia de *pull-up* de 4 700  $\Omega$ , conectada entre el pin utilizado como bus de datos del Arduino y 5 V; y conectar todos los dispositivos 1-Wire al bus de datos y a tierra.

La función básica para la utilización de esta librería es la creación de un objeto tipo OneWire. La sintaxis necesaria es la siguiente: *OneWire miObjetoOneWire(pin)*, donde *miObjetoOneWire* es el objeto de tipo OneWire y *pin* es el número de pin de entrada/salida de la placa de desarrollo Arduino Nano que sirve como bus de datos.

#### **4.1.1.2. Librería Dallas Temperature**

Es una extensión de la librería OneWire, por lo tanto es necesario instalar ambas para su correcto funcionamiento.

Esta librería provee una forma fácil de controlar los sensores de temperatura DS18B20, sin tener limitante de software para el número de

sensores conectados por bus de datos. Las funciones y código utilizado para la programación de la placa de desarrollo Arduino Nano, de esta librería se presentan a continuación.

El tipo de objeto Dallas Temperature se declara con la sintaxis *DallasTemperature sensors(&miObjetoOneWire)*, donde *sensors* es la instancia de Dallas Temperature y *&miObjetoOneWire* es la referencia al objeto de tipo OneWire previamente declarado.

La función *sensors.begin()* donde *sensors* es el objeto DallasTemperature, inicializa los sensores del bus de datos con la opción predeterminada de 9 bits de ancho de palabra para la conversión analógico a digital.

La función *sensors.requestTemperatures()* donde *sensors* es el objeto DallasTemperature, envía una petición global de toma de temperatura a todos los dispositivos en el bus.

La función *sensors.getTempCByIndex(ind)* donde *sensors* es el objeto DallasTemperature, devuelve la temperatura en grados Celsius del sensor con el índice entero *ind*, con base cero en el bus de datos.

El código fuente completo para la placa de desarrollo Arduino Nano, puede ser consultado en el apéndice I.

## **4.2. Desarrollo para PC**

El desarrollo del software para PC se realiza con la herramienta Microsoft Visual Studio Express 2012 for Windows Desktop<sup>®</sup>. La cual permite la creación de aplicaciones de escritorio para el sistema operativo Microsoft Windows<sup>®</sup> en sus diferentes versiones, utilizando como lenguaje de programación C#, C++ o Visual Basic.

### **4.2.1. Lenguaje de programación Visual Basic**

El lenguaje de programación utilizado para desarrollar el software para PC es Visual Basic. Esto proporciona herramientas para facilitar y agilizar el desarrollo de aplicaciones con funcionalidades de uso común sobre el sistema operativo Microsoft Windows<sup>®</sup> en sus diferentes versiones.

### **4.2.2. Especificaciones de objetos en el código fuente**

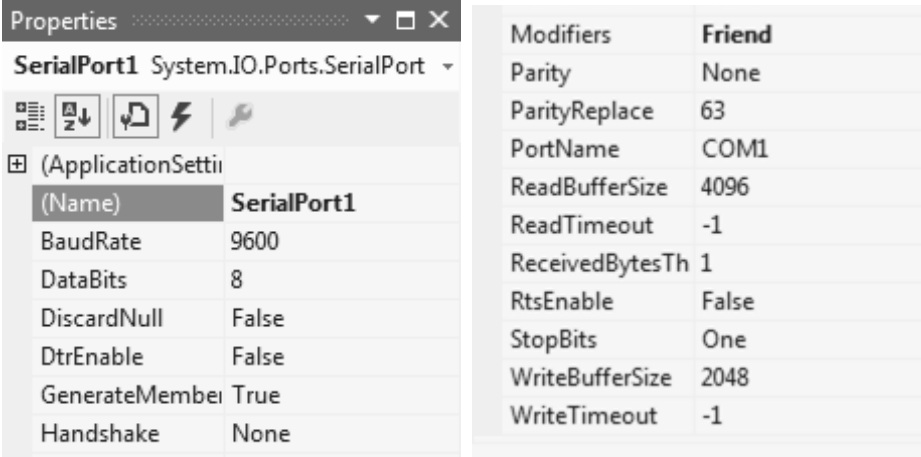
Por la naturaleza de simplicidad del ambiente de desarrollo de Microsoft Visual Studio Express 2012 for Windows Desktop<sup>®</sup>, las propiedades de los objetos en Visual Basic pueden ser definidas con ayuda de la interfaz gráfica.

A continuación se enumeran las propiedades de los objetos, necesarias para el correcto funcionamiento del software, definidas de esa forma. El resto del código fuente puede ser consultado en el apéndice II. La lógica de programación para el desarrollo de la aplicación puede ser consultada en el apéndice III.

#### 4.2.2.1. Objeto de puerto serial

Para utilizar y modificar las funciones y propiedades de los puertos serial instalados en el sistema operativo se tienen los objetos tipo SerialPort, el objeto en el código fuente tiene por nombre SerialPort1 y las propiedades y sus valores se encuentran en la figura 14.

Figura 14. Propiedades del objeto de puerto serial



The image shows a screenshot of the Visual Studio Properties window for a SerialPort1 object. The window title is 'Properties' and the object is identified as 'SerialPort1 System.IO.Ports.SerialPort'. Below the title bar, there are several icons for actions like 'Copy', 'Paste', 'Refresh', and 'Undo'. The main area of the window is divided into two panes. The left pane shows the 'ApplicationSettings' section with a table of properties for 'SerialPort1'. The right pane shows a table of properties for the 'Friend' modifier.

(Name)	SerialPort1
BaudRate	9600
DataBits	8
DiscardNull	False
DtrEnable	False
GenerateMember	True
Handshake	None

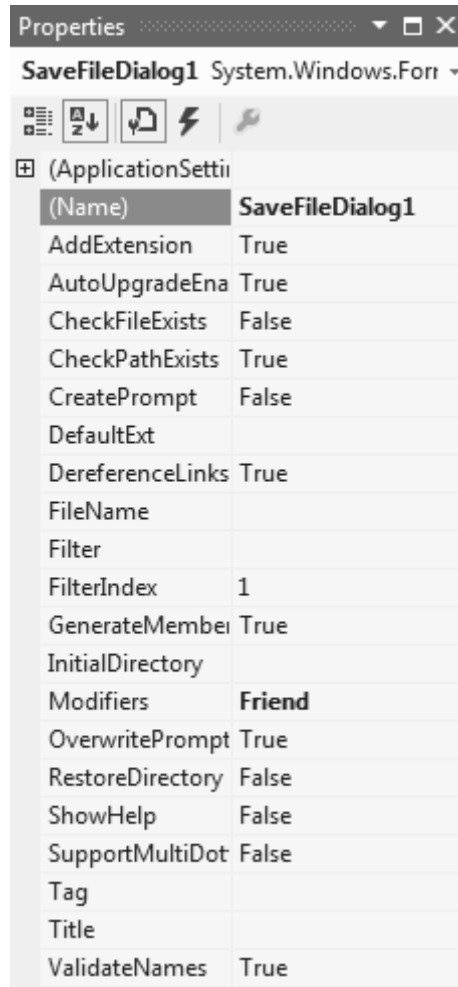
Modifiers	Friend
Parity	None
ParityReplace	63
PortName	COM1
ReadBufferSize	4096
ReadTimeout	-1
ReceivedBytesTh	1
RtsEnable	False
StopBits	One
WriteBufferSize	2048
WriteTimeout	-1

Fuente: elaboración propia.

#### 4.2.2.2. Objeto cuadro de diálogo grabar

Visual Basic permite crear objetos que se encargan de manejar distintos tipos de tareas, el cuadro de diálogo que solicita al usuario una locación para almacenar un archivo, se realiza con el tipo de objeto SaveFileDialog. En el código fuente, el objeto tiene por nombre SaveFileDialog1 y sus propiedades con valores se enumeran en la figura 15. Este objeto se utiliza para grabar el archivo de valores separados por coma, generado por el programa.

Figura 15. **Propiedades del cuadro de diálogo grabar**



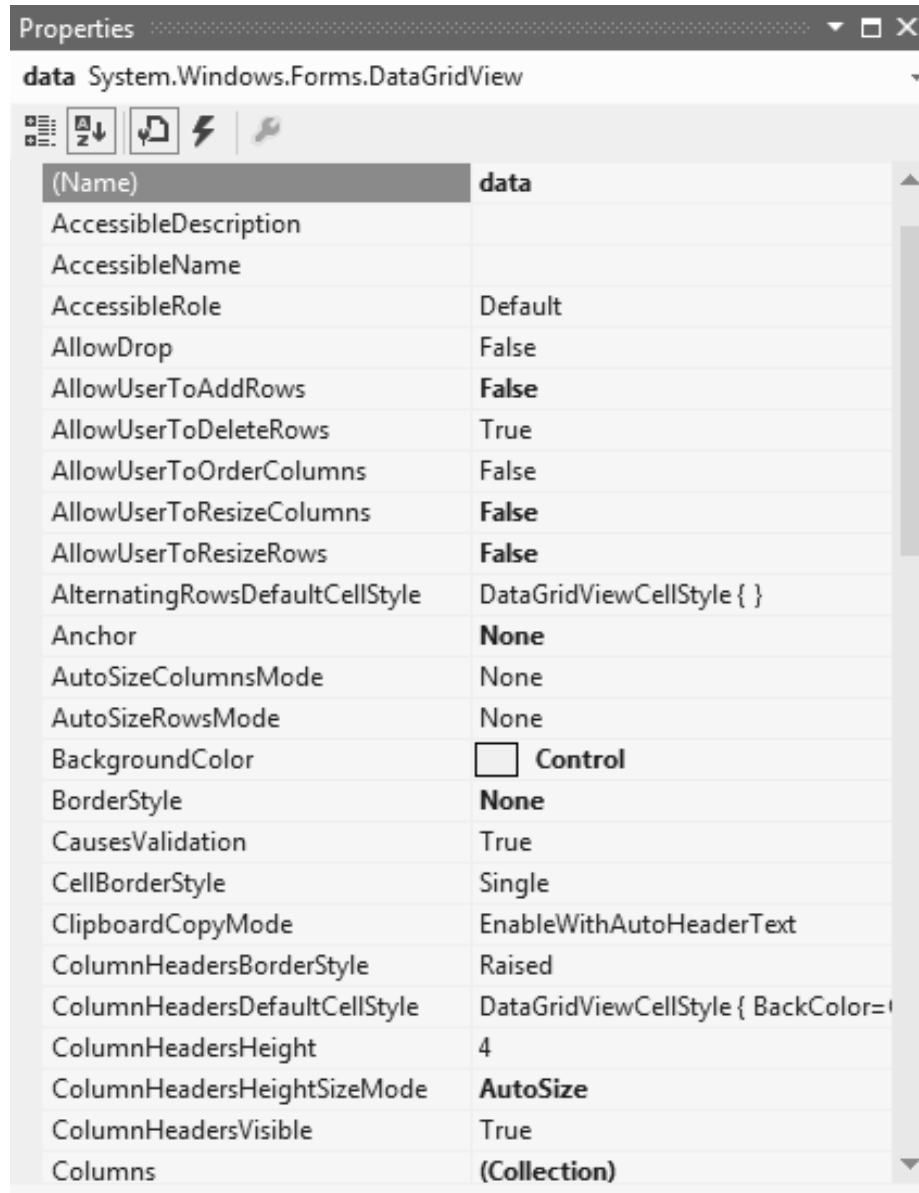
Fuente: elaboración propia.

#### 4.2.2.3. **Objeto vista cuadrícula de datos**

El objeto vista cuadrícula de datos permite al usuario visualizar los datos pertinentes en un espacio delimitado por una cuadrícula. En este espacio se muestran los datos de los 14 sensores DS18B20, así como también la hora a la

que se tomó la medición. En el código fuente se le da el nombre de data y sus propiedades pueden verse en la figura 16.

Figura 16. **Propiedades del objeto vista cuadrícula de datos**



Continuación de la figura 16.

RowHeadersVisible	False
RowHeadersWidth	41
RowHeadersWidthSizeMode	DisableResizing
RowsDefaultCellStyle	DataGridViewCellStyle { }
RowTemplate	DataGridViewRow { Index=-1 }
ScrollBars	Vertical
SelectionMode	FullRowSelect
ShowCellErrors	True
ShowCellToolTips	True
ShowEditingIcon	False
ShowRowErrors	True
Size	720, 388
StandardTab	False
TabIndex	4
TabStop	True
Tag	
UseWaitCursor	False
VirtualMode	False
Visible	True

Fuente: elaboración propia.

### 4.2.3. Almacenamiento de datos

La aplicación permite el almacenamiento de los datos adquiridos mediante los sensores de temperatura, en un archivo de valores separados por coma, generado a petición del usuario.

#### 4.2.3.1. Archivo de valores separados por coma

Los archivos de valores separados por coma son archivos de formato abierto sencillo, para representar datos en forma de tabla. Las columnas son separadas por comas y las filas por saltos de línea.



El formato de valores separados por coma o CSV (*comma-separated values*) es muy versátil y permite ser abierto en una diversidad de programas, como en hojas de cálculo. Esto permite la manipulación y análisis de los datos de forma eficiente y sencilla.

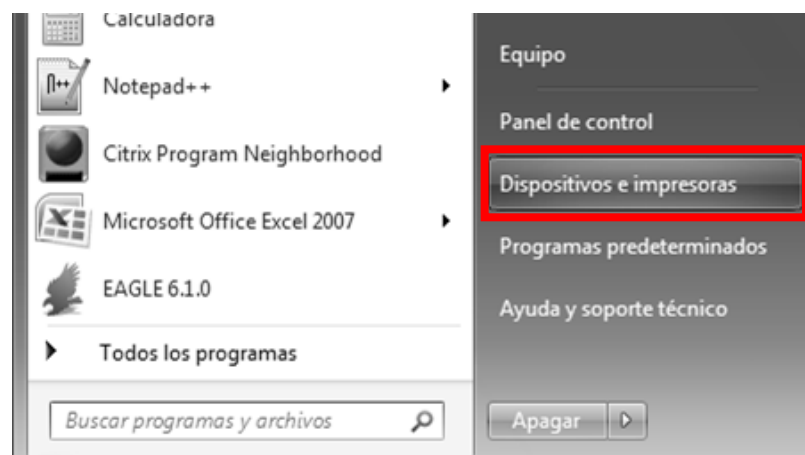
### 4.3. Configuración y utilización del software para usuario final

A continuación se describen los pasos necesarios para configurar y utilizar correctamente el software de PC para los usuarios finales del sistema de adquisición de datos del intercambiador de calor de concha y tubos.

#### 4.3.1. Instalación y emparejamiento del dispositivo Bluetooth

Antes de utilizar el software para PC se deben emparejar el módulo Bluetooth HC-05 con el equipo que se va a utilizar. Para esto es necesario ir a la configuración de Dispositivos e impresoras desde el menú inicio, como se muestra en la figura 17.

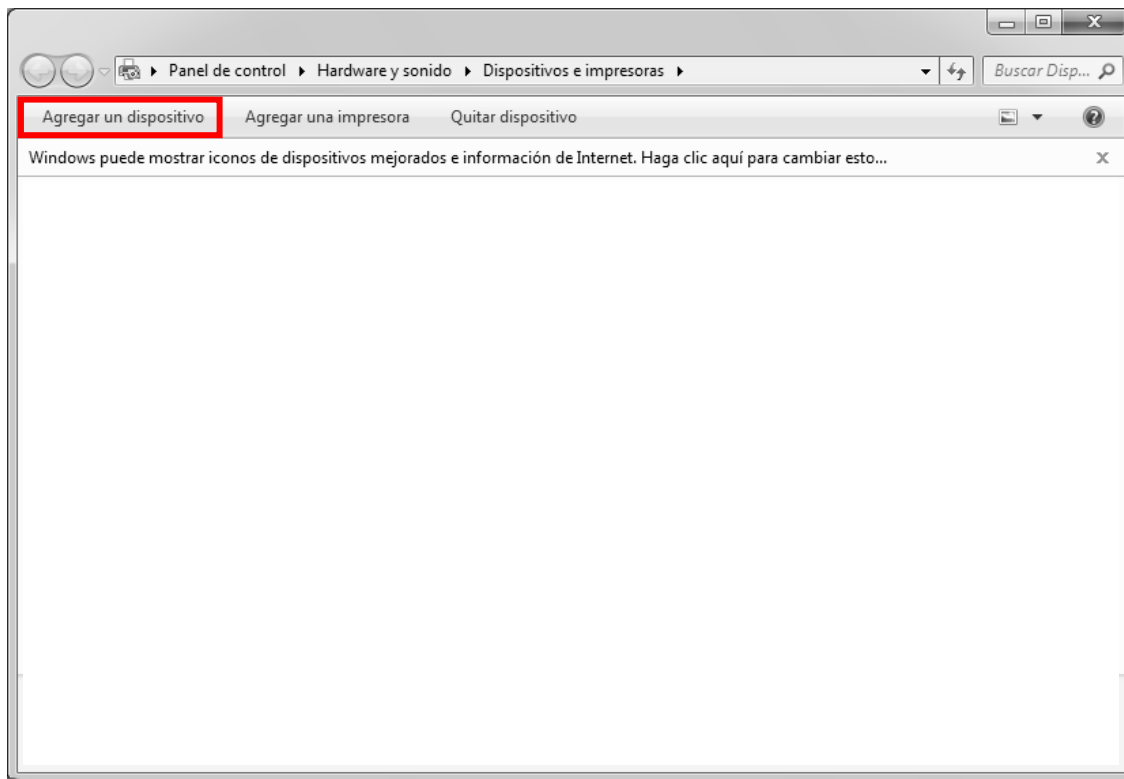
Figura 17. **Dispositivos e impresoras**



Fuente: elaboración propia.

Seleccionar el botón Agregar un dispositivo ubicado en la esquina superior izquierda, como se muestra en la figura 18.

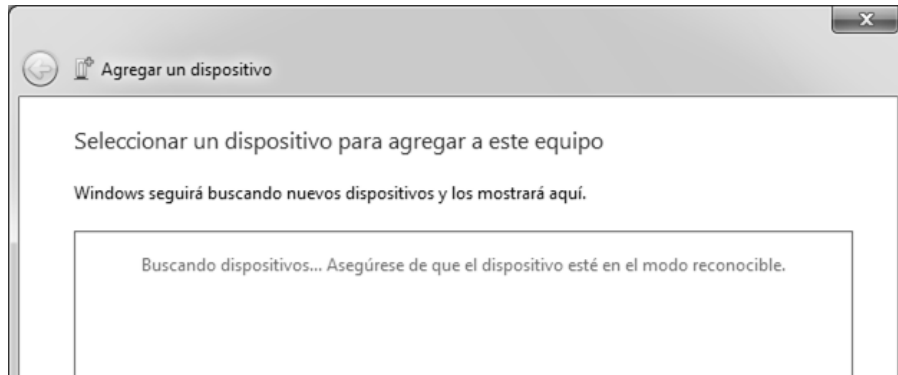
Figura 18. **Agregar un dispositivo**



Fuente: elaboración propia.

Aparecerá una pantalla, véase figura 19, en donde el sistema operativo buscará y mostrará los dispositivos al alcance. En esta etapa debe asegurarse que el hardware del sistema de adquisición de datos esté conectado, funcionando y al alcance del equipo.

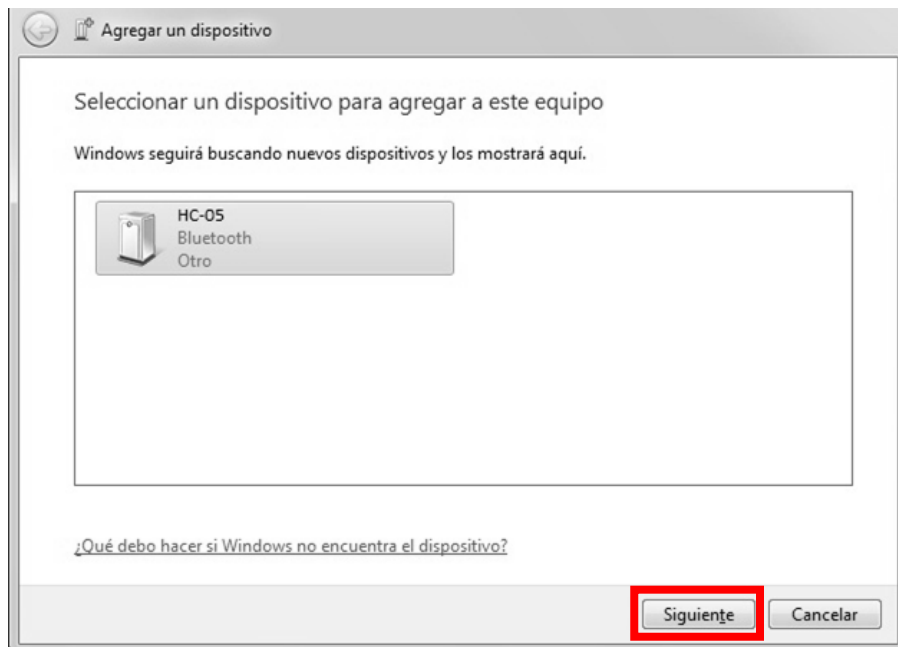
Figura 19. **Buscando dispositivos**



Fuente: elaboración propia.

Cuando se haya encontrado el dispositivo HC-05, seleccionarlo y presionar el botón Siguiente de la ventana. Véase figura 20.

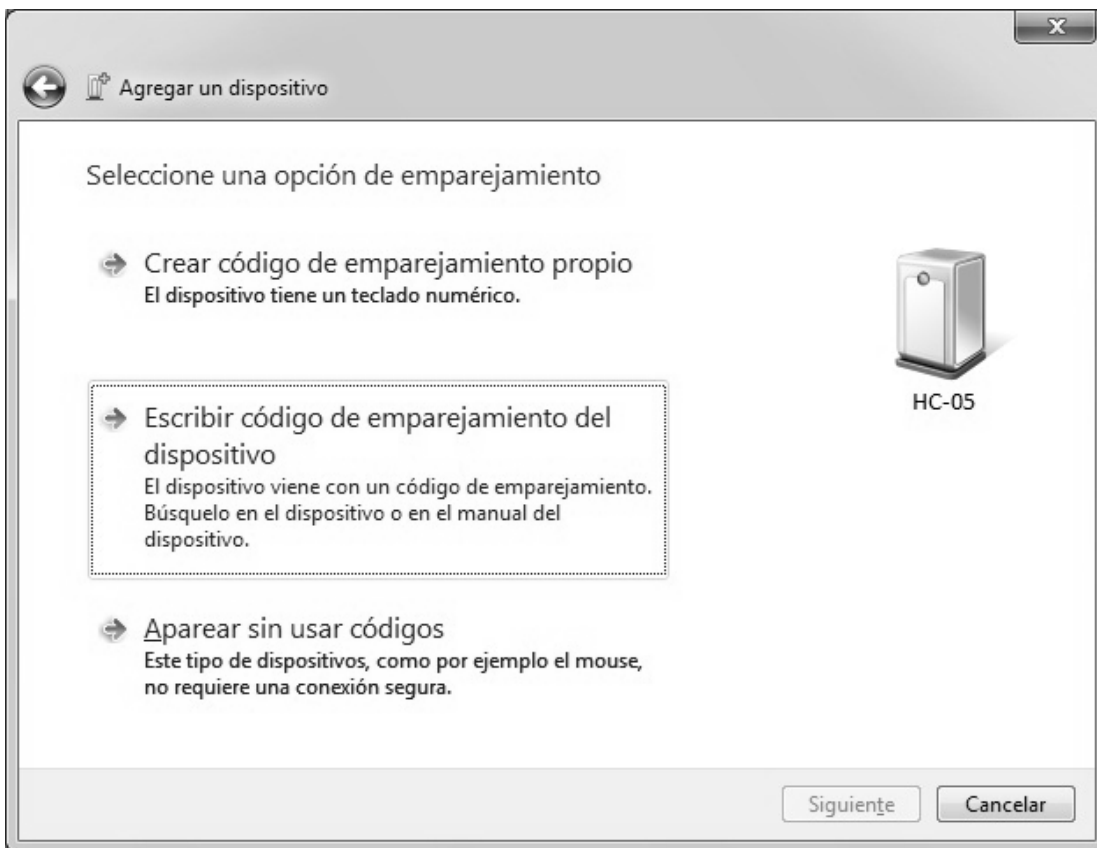
Figura 20. **Seleccionar dispositivo HC-05**



Fuente: elaboración propia.

En la siguiente ventana, véase figura 21, seleccionar la opción: Escribir código de emparejamiento del dispositivo. El dispositivo viene con un código de emparejamiento. Búsquelo en el dispositivo o en el manual del dispositivo.

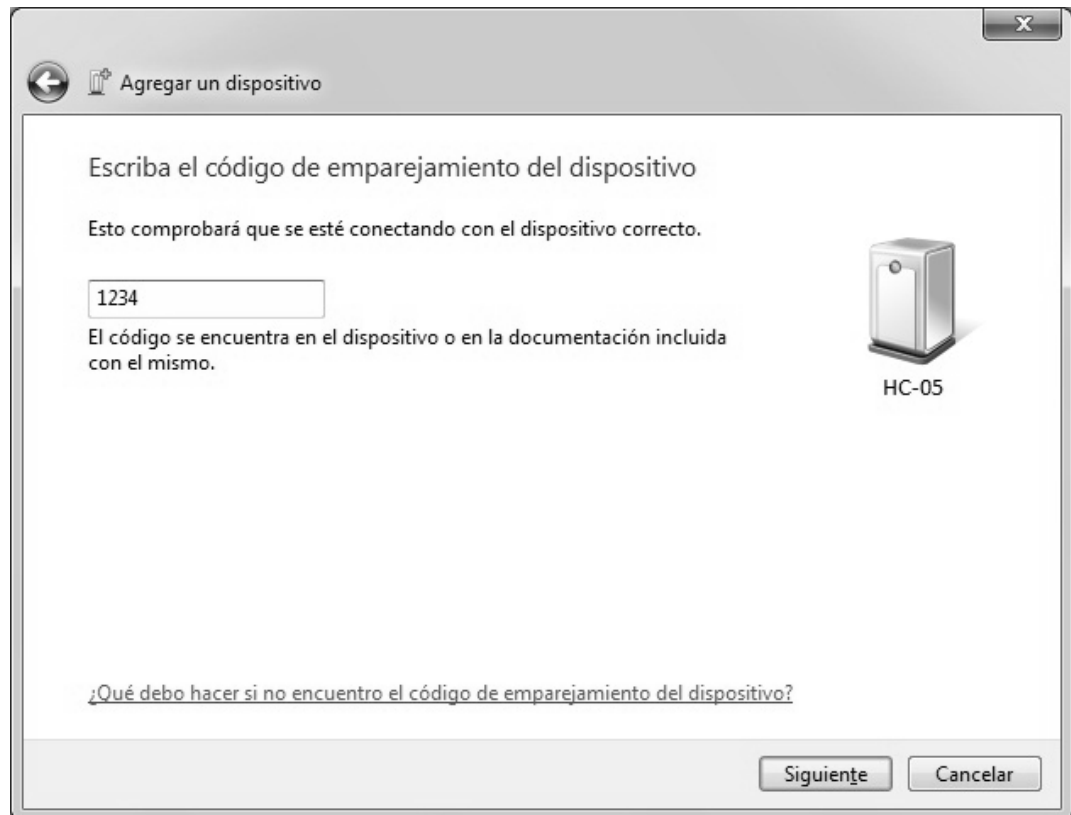
Figura 21. **Escribir código de emparejamiento, paso 1**



Fuente: elaboración propia.

A continuación ingresar en el cuadro de texto el código de emparejamiento: 1234. Véase la figura 22. Y luego presionar el botón siguiente.

Figura 22. **Escribir código de emparejamiento, paso 2**



Fuente: elaboración propia.

El dispositivo se agregará al sistema operativo y se instalarán los controladores de los puertos seriales virtuales. Hacer clic en el ícono de notificación de instalación del dispositivo, en la barra de tareas, esquina inferior derecha. Véase la figura 23.

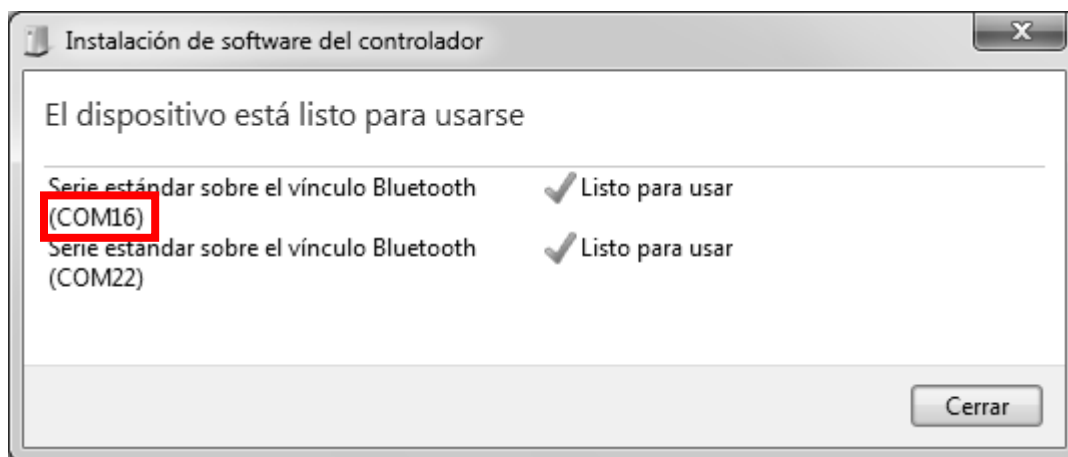
Figura 23. **Notificación de instalación del dispositivo**



Fuente: elaboración propia.

Una ventana emergerá con los nombres de los puertos seriales virtuales instalados. El primer nombre de puerto, que se muestra entre paréntesis, se utilizará para conectar el software con el sistema de adquisición de datos. Véase figura 24.

Figura 24. **Instalación de software del controlador**

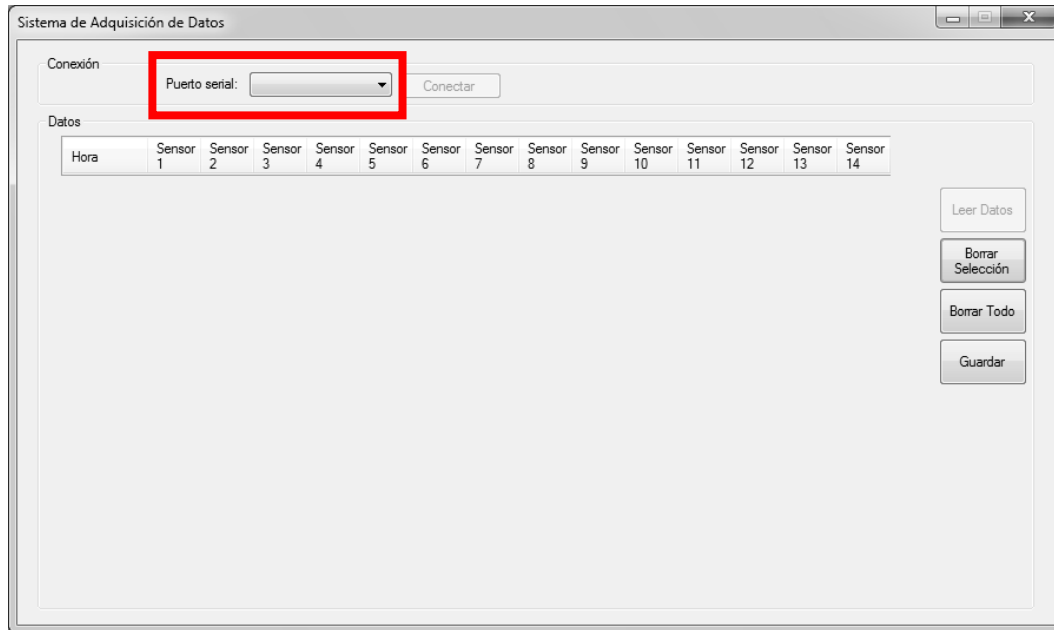


Fuente: elaboración propia.

#### 4.3.2. **Conexión de la aplicación**

Abrir el archivo ejecutable DAQ.exe en el que se mostrará la ventana del software desarrollado para el sistema de adquisición de datos. Hacer clic en la barra de menú desplegable Puerto Serial, en la sección de conexión de la ventana. Véase la figura 25.

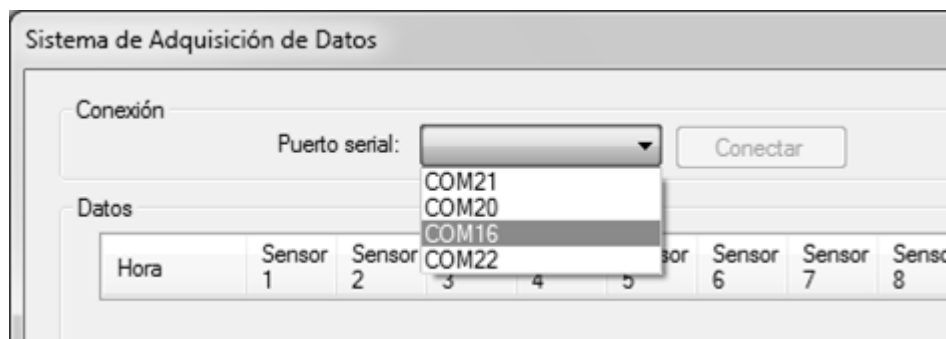
Figura 25. **Ventana del sistema de adquisición de datos**



Fuente: elaboración propia.

En la lista se mostrarán los puertos seriales disponibles, seleccionar el puerto serial virtual que se instaló con el emparejamiento de dispositivos Bluetooth. Véase figura 26.

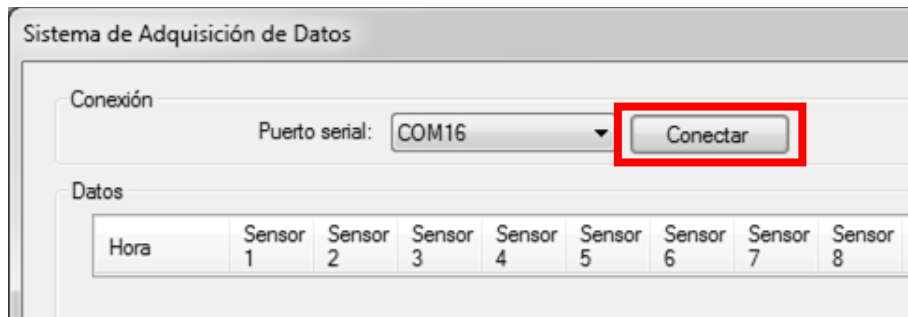
Figura 26. **Selección de puerto serial**



Fuente: elaboración propia.

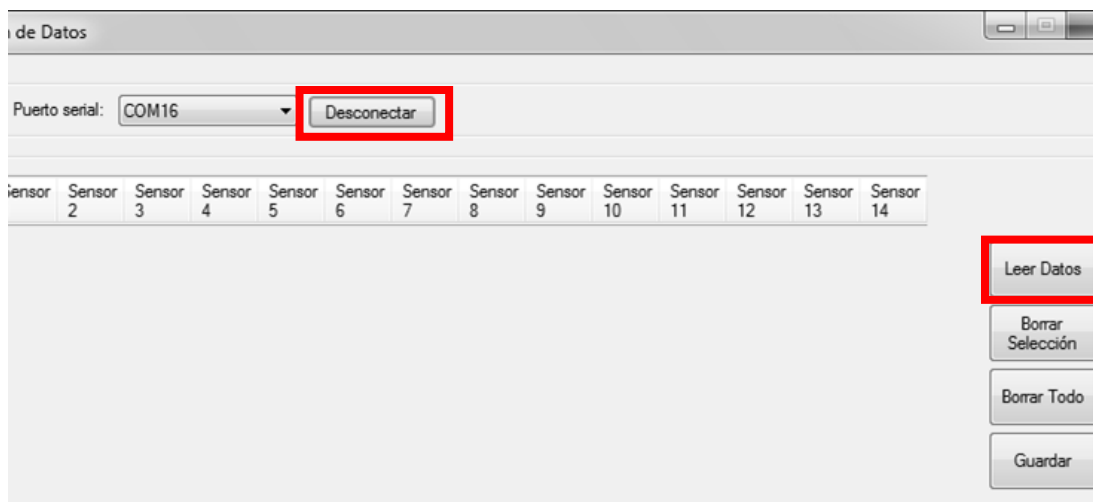
Después de seleccionar el puerto serial se habilitará el botón Conectar, hacer clic sobre este botón. Véase figura 27.

Figura 27. **Conectar a puerto serial virtual**



Fuente: elaboración propia.

Figura 28. **Conexión exitosa a puerto serial virtual**



Fuente: elaboración propia.

Al momento en que se habilitan los botones Desconectar y Leer Datos la conexión al sistema de adquisición de datos ha sido exitosa. Véase la figura 28.



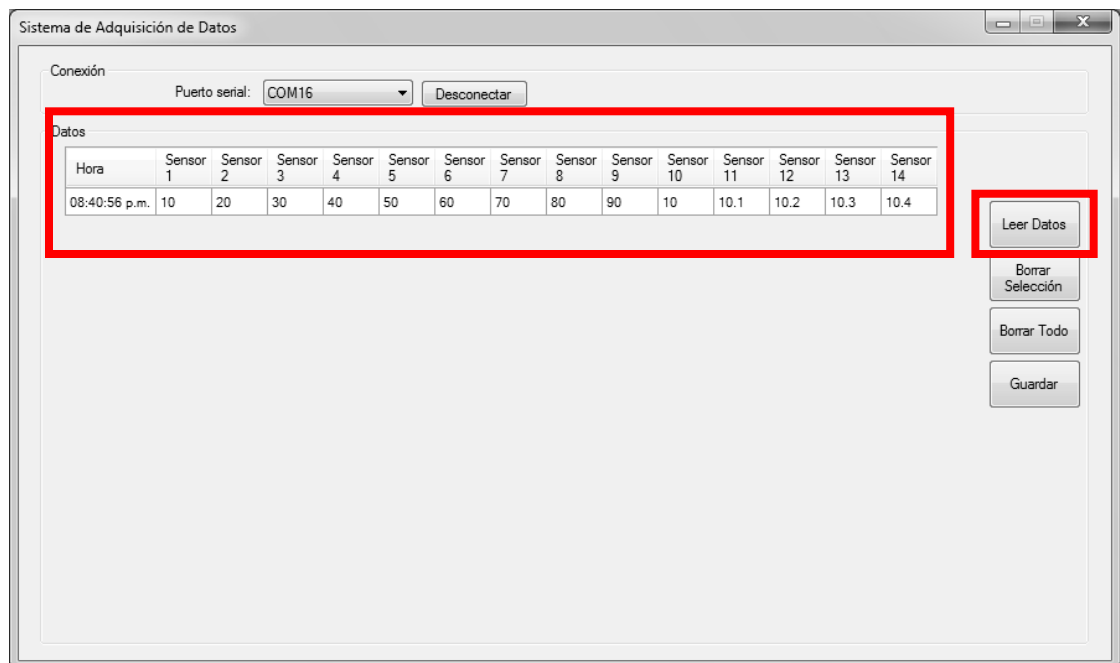
### 4.3.3. Uso de la aplicación

Al momento de conectar la aplicación al hardware, la interfaz gráfica del sistema de adquisición de datos permite realizar una serie de operaciones esenciales para el propósito del programa. A continuación se enumeran dichas operaciones.

#### 4.3.3.1. Lectura de datos

Para leer los datos de temperatura de los 14 sensores, hacer clic el botón Leer Datos. Después de unos momentos se mostrará, en la tabla de la izquierda, la hora de la medición junto con los grados Celsius correspondientes a los sensores de temperatura DS18B20. Véase figura 29.

Figura 29. Lectura de datos

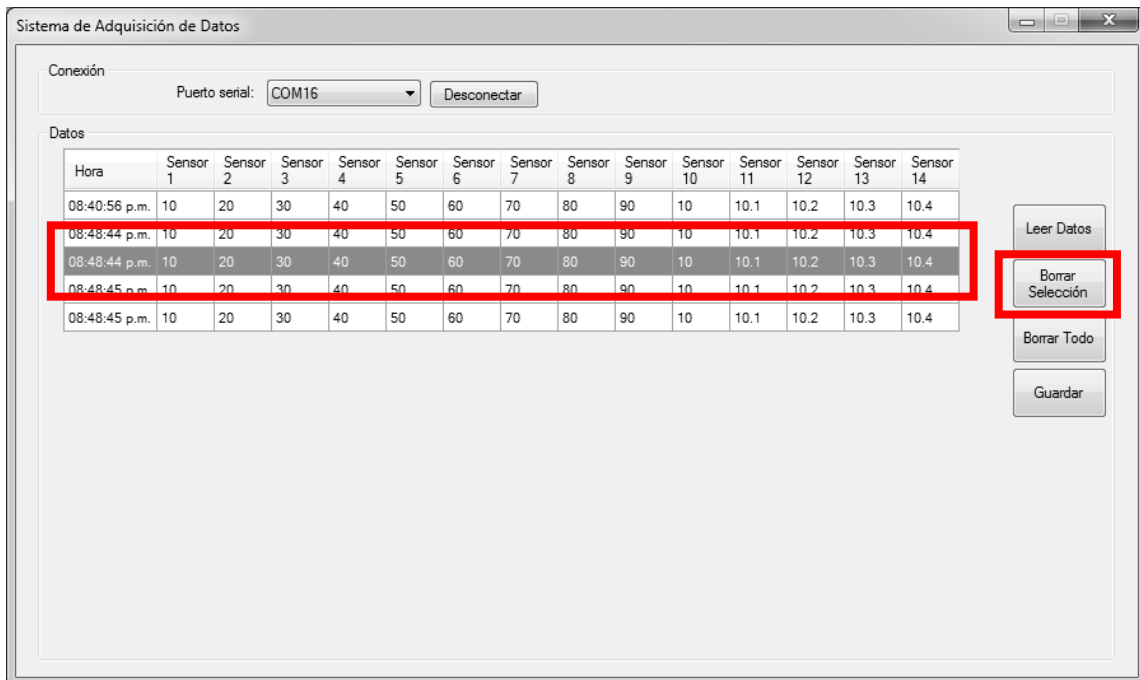


Fuente: elaboración propia.

### 4.3.3.2. Eliminación de datos

La aplicación permite borrar líneas de datos individualmente. El procedimiento consiste en seleccionar la línea de datos que se desea eliminar y presionar el botón Borrar Selección. Véase figura 30.

Figura 30. Borrar selección



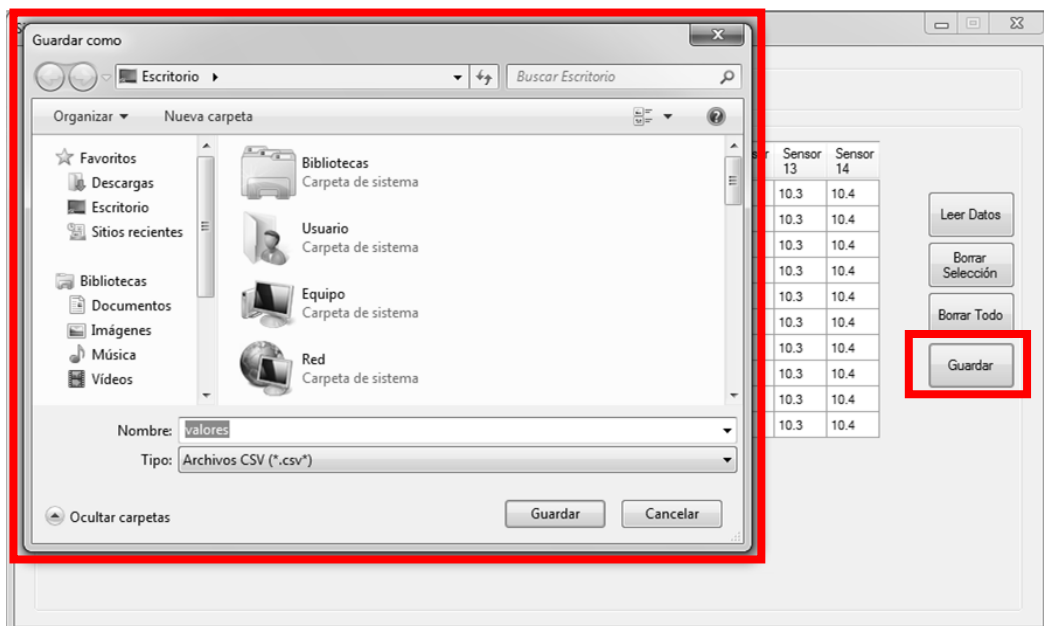
Fuente: elaboración propia.

La aplicación también permite la eliminación de todos los datos de la tabla. El procedimiento consiste en presionar el botón Borrar Todo. Luego, aparecerá un cuadro de diálogo preguntando si está seguro de querer borrar todos los datos, al hacer clic en el botón Sí; se eliminarán todos los datos. Si se desea cancelar la operación se debe hacer clic en el botón No.

### 4.3.3.3. Guardar datos

La aplicación permite exportar los datos de la tabla, a un archivo CSV. Para realizar este procedimiento, después de haber hecho la lectura de los datos de interés, se presiona el botón Guardar. Se mostrará un cuadro de diálogo para seleccionar el destino y el nombre del archivo. Véase figura 31.

Figura 31. Guardar datos



Fuente: elaboración propia.

Como último paso, hacer clic en el botón Guardar del cuadro de diálogo y el archivo se abrirá con el programa por defecto para los archivos CSV.



## CONCLUSIONES

1. El diseño del sistema de adquisición de datos contribuye con la formación académica de los estudiantes del Laboratorio de Química, de la Facultad de Ingeniería de la USAC.
2. El sistema de adquisición de datos de temperatura reduce el tiempo en que se realizan mediciones y elimina los errores debidos al factor humano durante las prácticas en el intercambiador de calor de concha y tubos.
3. El circuito integrado DS18B20 realiza la conversión analógica a digital de los datos de temperatura y los transmite de forma serial. Esto evita problemas con la atenuación de señales y hace la comunicación más confiable.
4. El protocolo de comunicación 1-Wire permite la conexión de varios dispositivos a través de un solo bus de datos.
5. El protocolo de comunicación 1-Wire hace posible la alimentación parásita a través de la línea de datos, en dispositivos como el DS18B20. Esto simplifica y reduce costos en el diseño de los circuitos en los que se utiliza.
6. La placa de desarrollo Arduino Nano es un dispositivo de bajo costo con las prestaciones necesarias para aplicaciones en sistemas embebidos de mediana complejidad.

7. El módulo Bluetooth HC-05 facilita la comunicación entre dispositivos, haciéndola inalámbrica, sencilla, segura y confiable.
8. El diseño modular del sistema de adquisición de datos facilita la actualización, el mantenimiento preventivo y el mantenimiento correctivo de hardware y software.
9. El diseño del sistema de adquisición de datos está compuesto por circuitería simple y pocos dispositivos especializados, dando como resultado un sistema robusto y confiable.
10. Las librerías para la plataforma Arduino hacen posible que el tiempo de desarrollo y la complejidad del código se minimicen. Esto facilita la corrección de errores, la actualización y la optimización de código.
11. La creación de aplicaciones de escritorio para Microsoft Windows<sup>®</sup>, utilizando la herramienta Microsoft Visual Studio Express 2012 for Windows Desktop<sup>®</sup> y el lenguaje de programación Visual Basic, reducen tiempos de desarrollo, facilita la corrección de errores y permite la utilización de controles nativos del sistema operativo.

## RECOMENDACIONES

1. Considerar que el sistema de adquisición de datos debe recibir inspecciones periódicas visuales y de funcionamiento, para asegurar que las prácticas de laboratorio sean llevadas a cabo satisfactoriamente y sin retrasos o imprevistos.
2. Tomar en cuenta que el único equipo electrónico utilizado en el diseño del sistema de adquisición de datos capaz de soportar humedad, es el circuito integrado DS18B20 con encapsulado metálico. El resto del equipo debe estar protegido y fuera del alcance de la humedad y demás condiciones ambientales no favorables.
3. Para el óptimo funcionamiento del sistema de adquisición de datos se sugiere darle al usuario final del equipo una inducción con las indicaciones de la configuración y utilización del software, como se presenta en el capítulo 4, inciso 4,3 Configuración y utilización del software para usuario final.
4. Es importante evaluar la utilidad presentada por el diseño del sistema de adquisición de datos de temperatura del intercambiador de calor de concha y tubos, para considerar la implementación de sistemas de adquisición de datos para otros equipos del laboratorio, basados en el presente diseño.

5. Evaluar la necesidad del desarrollo de la aplicación del módulo de almacenamiento y presentación: en otros sistemas operativos, para hacer más amplia la población estudiantil a la que está dirigido el sistema; y con otras tecnologías, tomando en cuenta que existen opciones de código abierto y multiplataforma. Para esto se propone el apéndice III. Lógica de aplicación del dispositivo de almacenamiento y presentación.



## BIBLIOGRAFÍA

1. DI PAOLO EMILIO, Maurizio. *Data Acquisition Systems: From Fundamentals to Applied Design*. Estados Unidos, Nueva York: Springer Science & Business Media, 2013. 135 p.
2. MARTÍNEZ, Evelio. *Conversión Analógico-Digital (ADC)*. [en línea]. <<http://www.eveliux.com/mx/Conversion-Analogico-Digital-ADC.html>>. [Consulta: mayo de 2015].
3. MAXIM INTEGRATED. DS18B20 – *HIGH-PRECISION 1-WIRE DIGITAL THERMOMETER* [en línea]. <<http://datasheets.maximintegrated.com/en/ds/DS18S20.pdf>>. [Consulta: mayo de 2015].
4. MILES BURTON. *Dallas Temperature Control Library*. [en línea]. <[http://milesburton.com/Main\\_Page?title=Dallas\\_Temperature\\_Control\\_Library](http://milesburton.com/Main_Page?title=Dallas_Temperature_Control_Library)>. [Consulta: mayo de 2015].
5. NATIONAL INSTRUMENTS. *Adquisición de Datos (DAQ)*. [en línea] <<http://www.ni.com/data-acquisition/esa/>>. [Consulta: abril de 2015].
6. NATIONAL INSTRUMENTS. *¿Qué es adquisición de datos?* [en línea]. <<http://www.ni.com/data-acquisition/what-is/esa/>> [Consulta: abril de 2015].

7. OMEGA. *Data acquisition systems*. [en línea]. <<http://www.omega.com/techref/pdf/dasintro.pdf>> [Consulta: abril de 2015].
8. PJRC ELECTRONIC PROJECTS. *OneWire Library*. [en línea]. <[http://www.pjrc.com/teensy/td\\_libs\\_OneWire.html](http://www.pjrc.com/teensy/td_libs_OneWire.html)>. [Consulta: mayo de 2015].
9. UNITED ELECTRONIC INDUSTRIES. *Data acquisition systems*. [en línea] <<http://www.ueidaq.com/data-acquisition-systems.html>>. [Consulta: abril de 2015].

## APÉNDICE

### I. Código fuente de placa de desarrollo Arduino Nano

```
1  #include <OneWire.h>
2  #include <DallasTemperature.h>
3  #define Pin 2 //Se declara el pin donde se conectará la DATA
4  char input;
5  OneWire ourWire(Pin); //Se establece el pin declarado como
6                          //bus para la comunicación OneWire
7  DallasTemperature sensors(&ourWire); //Se instancia la librería
8                          //DallasTemperature
9  void setup() {
10     delay(1000);
11     Serial.begin(9600);
12     sensors.begin(); //Se inician los sensores
13 }
14
15 void loop() {
16     //Prepara el sensor para la lectura
17     sensors.requestTemperatures();
18     input=Serial.read();
19     if(input=='1'){
20
21         //Se lee e imprime la temperatura en grados Celsius
22         Serial.print("#");
23         Serial.print(sensors.getTempCByIndex(0));
24         Serial.print(",");
25         Serial.print(sensors.getTempCByIndex(1));
26         Serial.print(",");
27         Serial.print(sensors.getTempCByIndex(2));
28         Serial.print(",");
29         Serial.print(sensors.getTempCByIndex(3));
30         Serial.print(",");
31         Serial.print(sensors.getTempCByIndex(4));
32         Serial.print(",");
33         Serial.print(sensors.getTempCByIndex(5));
34         Serial.print(",");
35         Serial.print(sensors.getTempCByIndex(6));
36         Serial.print(",");
37         Serial.print(sensors.getTempCByIndex(7));
38         Serial.print(",");
39         Serial.print(sensors.getTempCByIndex(8));
40         Serial.print(",");
41         Serial.print(sensors.getTempCByIndex(9));
42         Serial.print(",");
43         Serial.print(sensors.getTempCByIndex(10));
44         Serial.print(",");
45         Serial.print(sensors.getTempCByIndex(11));
46         Serial.print(",");
47         Serial.print(sensors.getTempCByIndex(12));
48         Serial.print(",");
49         Serial.print(sensors.getTempCByIndex(13));
50         Serial.print("#");
51     }
52 }
```

Fuente: elaboración propia.

## II. Código fuente de aplicación para PC en Visual Basic

```
Imports System
Imports System.Threading
Imports System.IO.Ports
Imports System.ComponentModel
Imports System.Data

Public Class daq
    Dim myPort As Array
    Dim temp As String
    Dim temps() As String

    Delegate Sub SetTextCallBack(ByVal [text] As String)

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        myPort = IO.Ports.SerialPort.GetPortNames()
        portComboBox.Items.AddRange(myPort)
        Dim col As DataGridViewColumn
        data.Columns.Add("hora", "Hora")
        col = data.Columns(0)
        col.Width = 73
        For i As Integer = 2 To 15 Step 1
            data.Columns.Add("sensor" + (i - 1).ToString, "Sensor " + (i - 1).ToString)
            col = data.Columns(i - 1)
            col.Width = 45
        Next
    End Sub

    Private Sub portComboBox_SelectedIndexChanged(sender As Object, e As EventArgs) _
        Handles portComboBox.SelectedIndexChanged
        connect.Enabled = True
    End Sub

    Private Sub connect_Click(sender As Object, e As EventArgs) Handles connect.Click
        On Error GoTo ErrorHandler
        If SerialPort1.IsOpen Then
            readButton.Enabled = False
            SerialPort1.Close()
            connect.Text = "Conectar"
        Else
            SerialPort1.PortName = portComboBox.Text
            SerialPort1.BaudRate = 9600
            SerialPort1.Open()
            readButton.Enabled = True
            connect.Text = "Desconectar"
        End If
        Exit Sub
ErrorHandler:
        MessageBox.Show("Ha ocurrido un error, por favor revise que el dispositivo" + _
            " esté conectado y configurado correctamente", "Error", MessageBoxButtons.OK)

    End Sub

    Private Sub readButton_Click(sender As Object, e As EventArgs) Handles readButton.Click
        On Error GoTo ErrorHandler
        If SerialPort1.IsOpen() Then
            SerialPort1.Write("1")
        Else
            connect.Text = "Conectar"
            readButton.Enabled = False
            MessageBox.Show("El puerto de comunicaciones ha sido desconectado.", _
                "Error", MessageBoxButtons.OK)
        End If
        Exit Sub
    End Sub
```

## Continuación apéndice II.

```
Exit Sub
ErrorHandler:
    MessageBox.Show("Ha ocurrido un error, por favor revise que el dispositivo " + _
        "esté conectado y configurado correctamente", "Error", MessageBoxButtons.OK)

End Sub

Private Sub SerialPort1_DataReceived(sender As Object, e As SerialDataReceivedEventArgs) _
    Handles SerialPort1.DataReceived
    On Error GoTo ErrorHandler
    ReceivedText(SerialPort1.ReadExisting())
    Exit Sub
ErrorHandler:
    MessageBox.Show("Ha ocurrido un error, por favor revise que el dispositivo " + _
        "esté conectado y configurado correctamente", "Error", MessageBoxButtons.OK)
End Sub

Private Sub ReceivedText(ByVal [text] As String) 'input from ReadExisting
    If Me.data.InvokeRequired Then
        Dim x As New SetTextCallBack(AddressOf ReceivedText)
        Me.Invoke(x, New Object() {(text)})
    Else
        temp += text
        If (temp.Length > 14 And temp.Length < 200 And temp(0) = "#" _
            And temp(temp.Length - 1) = "#") Then
            temp = temp.Substring(1, temp.Length - 2)
            temps = Split(temp, ",")
            If temp.Contains("#") Then
                MessageBox.Show("Ha ocurrido un error con la transmisión de los datos, " + _
                    "por favor vuelva a intentarlo.", "Error", MessageBoxButtons.OK)
                temp = ""
                text = ""
            Else
                temps = {DateTime.Now.ToString("hh:mm:ss tt")}.Concat(temps).ToArray
                Me.data.Rows.Add(temps.ToArray)
                temp = ""
                data.ClearSelection()
            End If
        Else
            If (temp.Length >= 200) Then
                temp = ""
                MessageBox.Show("Ha ocurrido un error con la transmisión de los datos, " + _
                    "por favor vuelva a intentarlo.", "Error", MessageBoxButtons.OK)
            End If
        End If
    End If
End Sub

Private Sub saveButton_Click(sender As Object, e As EventArgs) Handles saveButton.Click
    If data.Rows.Count > 0 Then
        SaveFileDialog1.Filter = "Archivos CSV (*.csv)*.csv"
        If SaveFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then
            Dim headers = (From header As DataGridViewColumn In data.Columns.Cast(Of DataGridViewColumn)() _
                Select header.HeaderText).ToArray
            Dim rows = From row As DataGridViewRow In data.Rows.Cast(Of DataGridViewRow)() _
                Where Not row.IsNewRow _
                Select Array.ConvertAll(row.Cells.Cast(Of DataGridViewCell).ToArray, _
                    Function(c) If(c.Value IsNot Nothing, c.Value.ToString, ""))
            Using sw As New IO.StreamWriter(SaveFileDialog1.FileName)
                sw.WriteLine(String.Join(",", headers))
                For Each r In rows
                    sw.WriteLine(String.Join(",", r))
                Next
            End Using
            Process.Start(SaveFileDialog1.FileName)
        End If
    End If
End Sub
```

## Continuación apéndice II.

```
End If
End Sub

Private Sub deleteButton_Click(sender As Object, e As EventArgs) Handles deleteButton.Click
    If data.SelectedRows.Count = 1 Then
        data.Rows.Remove(data.SelectedRows(0))
        data.ClearSelection()
    End If
End Sub

Private Sub deleteAllButton_Click(sender As Object, e As EventArgs) Handles deleteAllButton.Click
    If data.Rows.Count > 0 Then
        If MessageBox.Show("¿Está seguro que desea borrar todos los datos?", "Confirmación", _
            MessageBoxButtons.YesNo) = Windows.Forms.DialogResult.Yes Then
            data.Rows.Clear()
        End If
    End If
End Sub
End Class
```

Fuente: elaboración propia.

### III. Lógica de aplicación del dispositivo de almacenamiento y presentación

La aplicación del dispositivo de almacenamiento y presentación puede ser recreada en otra plataforma de sistema operativo o utilizando otro tipo de tecnologías dependiendo de las necesidades que se presenten, es por eso que a continuación se detalla la lógica que se debe implementar en el desarrollo.

Se debe iniciar por crear un puerto serial virtual utilizando el protocolo de comunicación inalámbrico Bluetooth, las configuraciones del puerto serial virtual se listan a continuación:

1. Baudaje: 9600
2. Sin bit de paridad
3. Un bit de parada

Una vez configurada la comunicación serial y desarrollado el procedimiento de conexión, se envía el caracter "l" (letra L minúscula, con código ASCII decimal 108) para que el módulo de control de sensores inicie la secuencia de pedido de temperaturas.

El módulo de control de sensores responde con una cadena de caracteres de la siguiente forma:

```
#T1,T2,T3,T4,T5,T6,T7,T8,T9,T10,T11,T12,T13,T14#
```

Donde el caracter "#" (numeral, código ASCII decimal 35), es el caracter que indica el inicio de transmisión de las temperaturas.

T1 a T14 son las temperaturas de los sensores correspondientes en grados Celsius, por ejemplo, si la temperatura del sensor 1 es de 26,5 grados Celsius, T1 estaría compuesta por los siguientes cuatro caracteres:

1. “2” (dos, ASCII decimal 50)
2. “6” (seis, ASCII decimal 54)
3. “.” (punto, ASCII decimal 46)
4. “5” (cinco, ASCII decimal 53)

Para delimitar las temperaturas de cada sensor se utiliza el caracter “,” (coma, código ASCII decimal 44) y el caracter que indica la finalización de la transmisión de datos es nuevamente “#” (numeral, código ASCII numeral 35).

Cuando se terminan de recibir los datos del módulo de control de sensores, es posible volver a enviar el caracter “l” (L minúscula, código ASCII decimal 108) para solicitar la toma de nuevas temperaturas, y repetir el proceso.



## ANEXOS

### I. Cotizaciones de dispositivos y materiales separados por proveedor.

- ELECTRONICA ABC

DESCRIPCION	CANTIDAD	PRECIO UNITARIO (Q)	PRECIO TOTAL (Q)
Sensor DS18B20 encapsulado metálico	14	40,00	560,00
Arduino Nano	1	150,00	150,00
Modulo Bluetooth HC-05		113,00	113,00
Cables con conector	15	4,00	60,00
Placa de cobre	1	32,00	32,00
Header	2	4,50	9,00
<b>SUBTOTAL</b>			<b>924,00</b>

- ELECTRONICA STEREN

DESCRIPCION	CANTIDAD	PRECIO UNITARIO (Q)	PRECIO TOTAL (Q)
Thermofit negro 1/8	1 metro	2,50	2,50
Thermofit negro 1/4	3 metros	3,50	10,50
Cable de 3 hilos 24 AWG	2 metros	8,50	17,00
<b>SUBTOTAL</b>			<b>30,00</b>

- ELECTRONICA CORPORACION R & CH ELECTRONICA Y ELECTRICOS.

DESCRIPCION	CANTIDAD	PRECIO UNITARIO (Q)	PRECIO TOTAL (Q)
Caja plástica con tapadera	1	62,00	62,00
Extensión eléctrica 25 ft	1	45,00	45,00
Extensión eléctrica 8 ft	1	8,00	8,00
Socket ZIF	1	40,00	40,00
<b>SUBTOTAL</b>			<b>155,00</b>

- ELECTRONICA BP.

DESCRIPCION	CANTIDAD	PRECIO UNITARIO (Q)	PRECIO TOTAL (Q)
Thermofit 1/16	1 metro	5,00	5,00
Thermofit 3/16	1 metro	5,00	5,00
<b>SUBTOTAL</b>			<b>10,00</b>

- CELASA.

DESCRIPCION	CANTIDAD	PRECIO UNITARIO (Q)	PRECIO TOTAL (Q)
Flexitubo 1"	2 metros	3,45	6,90
<b>SUBTOTAL</b>			<b>6,90</b>

- CEMACO

DESCRIPCION	CANTIDAD	PRECIO UNITARIO (Q)	PRECIO TOTAL (Q)
Cable 4 conductores	3 metros	5,59	16,77
<b>SUBTOTAL</b>			<b>16,77</b>

<b>TOTAL</b>	<b>Q. 1 142,67</b>
--------------	--------------------

Fuente: Grupo colaborador electrónica aplicada 1, Consulta: noviembre 2014.