

# Instituto Tecnológico de Tuxtla Gutiérrez

*Instrumentación electrónica para  
Biorreactores EGSB para el  
tratamiento de fuentes  
agroindustriales.*

**ASESOR:**

Dr. Héctor Ricardo Hernández de León.

**ALUMNOS:**

Rubén Darío Caballero Serrano.

María De Los Ángeles Gómez López.

**INGENERIA ELECTRONICA  
RESIDENCIA PROFESIONAL**

---

## Contenido

Capítulo 1. Generalidades.....	1
1.1 Introducción.....	1
1.2 Información de la institución donde se desarrolló el proyecto.....	2
<b>1.2.1 Historia del ITTG.....</b>	<b>2</b>
<b>1.2.2 Misión.....</b>	<b>3</b>
<b>1.2.3 Visión.....</b>	<b>3</b>
<b>1.2.4 Localización.....</b>	<b>3</b>
<b>1.2.5 Área específica relacionada directamente con el proyecto.....</b>	<b>4</b>
1.3 Antecedentes.....	5
1.4 Planteamiento del problema.....	6
1.5 Nombre del proyecto.....	6
1.6 Objetivos.....	7
1.6.1 Objetivo general.....	7
1.6.2 Objetivos específicos.....	7
1.7 Justificación del proyecto.....	8
1.8.- Delimitación del tema.....	8
Capítulo 2. Fundamento teórico.....	9
2.1 Programación en Arduino.....	9
2.2 Características.....	9
2.3 Estructura básica de un programa.....	10
2.4 Entradas/Salidas digitales y analógicas.....	11
2.5 Tipos de Datos.....	12
2.6 funciones de tiempo.....	12
2.7 LabView.....	13
2.7.1 Utilizando LabVIEW.....	13
2.7.2 LabVIEW para el Medio Ambiente.....	14
2.7.3 Propiedad de los componentes.....	15
2.8 Microcontrolador.....	21
2.9 Arduino MEGA 2560.....	21
2.10 Estructura básica de un programa.....	22
2.10.1 Tipos de Datos.....	23
2.10.2 Funciones de Tiempo.....	23
2.11 pH Electrode model: ASP200-2-1M/BNC serialNO: E130722224.....	24

2.12 Sensor de temperatura DS18B20. ....	25
2.13 Sensor de oxígeno disuelto.....	27
Capitulo3. Desarrollo e implementación del proyecto. ....	29
3.1 conexión de sensor de pH al arduino.....	29
3.1.1 Programación en arduino. ....	30
3.1.2 <i>conexión del sensor DS18B20 estanco al arduino.</i> ....	33
3.1.3 programacion en arduino. ....	34
3.1.4 conexión de sensor de oxígeno disuelto al arduino. ....	35
3.1.5 programacion en arduino. ....	36
CAPITULO 4. Anexo de imágenes.....	37
5. <i>Observaciones y sugerencias.</i> .....	40
6.-Conclusión.....	41
7.- Referencias bibliográficas.....	42





# Capítulo 1. Generalidades.

## 1.1 Introducción.

La importancia de tratar aguas contaminadas es sumamente importante, ya que es una alta cantidad de agua la cual se está desperdiciando. En el tecnológico de Tuxtla Gutiérrez existe un proyecto llamado biorreactor tipo EGSB, su funcionamiento es descontaminar el agua hasta el punto de que pueda ser utilizada nuevamente. Existen muchas formas de contaminación en el agua, en este caso los biorreactores que están en el tecnológico tratan con el residuo de tres procesos diferentes, el agua que se contamina cuando se hace el licor, el de los lácteos y el del café. Las especies de sulfuros se derivan de la oxidación anaeróbica de materia orgánica. El  $H_2S$  es una sustancia venenosa y corrosiva que debe eliminarse de las aguas residuales antes de la descarga y los gases ricos en energía antes de la combustión o la transferencia a gasoductos. , tienen tecnología compleja y pueden tener un impacto negativo en el medio ambiente. Para hacer este proceso, es necesario tener el monitoreo de tres factores que influyen en la degradación de dichas bacterias en el agua, que son el pH, temperatura y oxígeno. Este monitoreo lo hacen manualmente, implica que se tiene que estar destapando constantemente los recipientes donde se está tratando el agua y eso también es una amenaza ya que no le debe de llegar el oxígeno. Es por eso que se le implementaron tres sensores los cuales por medio de una gráfica te describe el comportamiento de todo el día de dichas aguas. A continuación veremos esta implementación.

## 1.2 Información de la institución donde se desarrolló el proyecto

### 1.2.1 Historia del ITTG.

El Instituto Tecnológico de Tuxtla Gutiérrez es una institución pública dependiente de la Secretaría de Educación Pública. Imparte 9 licenciaturas y 2 programas de posgrado en las áreas de Ingeniería, Tecnología y Ciencias Administrativas.

El Instituto Tecnológico de Tuxtla Gutiérrez fue fundado el 24 de octubre de 1972, se encuentra ubicado en el centro del estado de Chiapas, que por su gran variedad de climas y suelos es propio para el cultivo de muy diversas especies vegetales nativas y adaptadas, con lo que se puede desarrollar la agroindustria, así como muchos otros procesos industriales a partir del gran potencial que ofrece este estado, en el cual el Instituto debe constituirse en el sujeto de cambio, al presentar alternativas de desarrollo sustentable del estado en forma multidisciplinaria.

La matrícula escolar en licenciatura se conforma de 1 982 estudiantes: 1 473 hombres y 509 mujeres. En el posgrado hay 78 alumnos: 68 hombres y 10 mujeres.

La población de egresados en el nivel de licenciatura es de 2 345 alumnos, de ellos 1 892 son hombres y 453 son mujeres.

Los principales laboratorios con los que cuenta el Instituto Tecnológico de Tuxtla Gutiérrez son:

- ✓ Microbiología
- ✓ Biotecnología
- ✓ Química
- ✓ Química pesada
- ✓ Mecánica
- ✓ Sistemas computacionales
- ✓ Ingeniería industrial
- ✓ Plantas piloto
- ✓ Polo Tecnológico Nacional

## Proyectos de vinculación exitosos con la industria.

- ✓ Aprovechamiento integral del timbre.
- ✓ Incremento en el contenido de sacarosa en la caña de azúcar.
- ✓ Efectos del boro en la producción del cacahuete.
- ✓ Factibilidad técnica en la extracción de aceites esenciales de la flora chiapaneca.

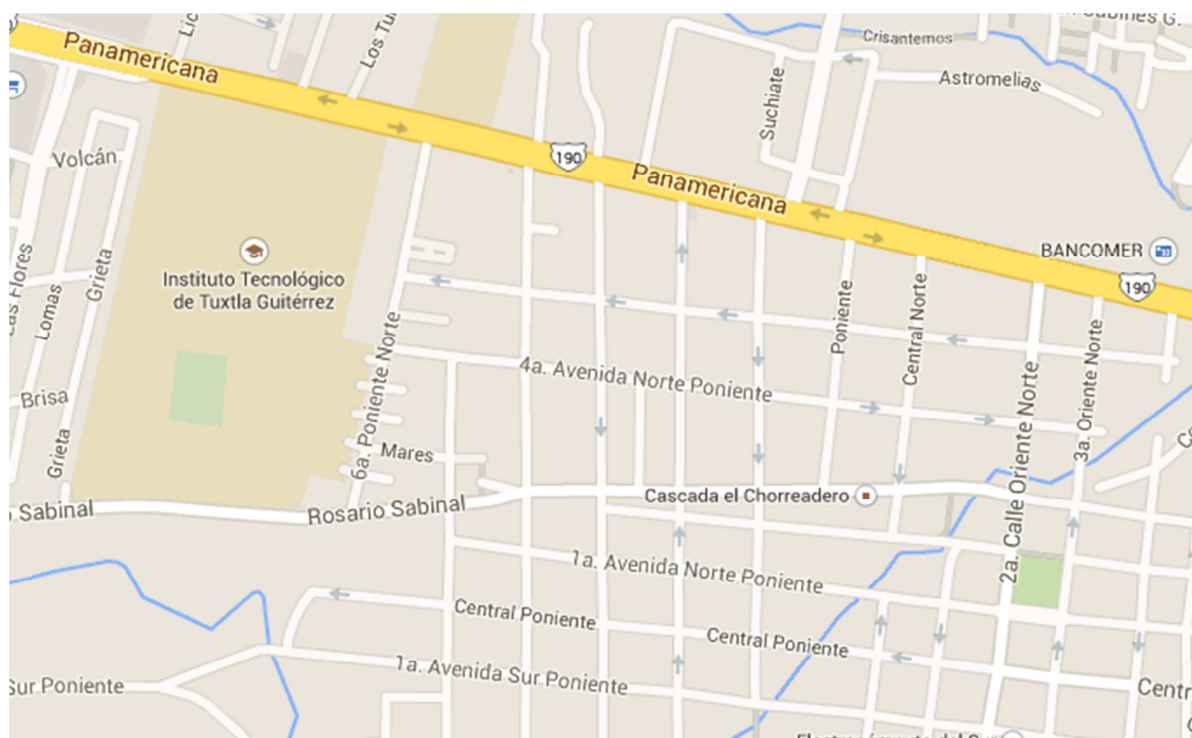
### 1.2.2 Misión.

Formar de manera integral profesionistas de excelencia en el campo de la ciencia y la tecnología con actitud emprendedora, respeto al medio ambiente y apego a los valores éticos.

### 1.2.3 Visión.

Ser una Institución de excelencia en la educación superior tecnológica del Sureste, comprometida con el desarrollo socioeconómico sustentable de la región.

### 1.2.4 Localización.



**Fig. 1.1 Carretera Panamericana Km 1080. Tuxtla Gutiérrez Chiapas, C.P. 29050.**

### **1.2.5 Área específica relacionada directamente con el proyecto.**

El laboratorio de Ingeniería Electrónica cuenta con 8 áreas destinadas a la investigación, desarrollo e implementación de proyectos escolares, 5 de las cuales son educativas, en las que los docentes imparten clases de: electrónica digital, PLC, electrónica de potencia, instrumentación, mediciones eléctricas, programación en c y electrónicas analógicas. Dos más son para investigación, una pertenece al grupo de IEEE y la otra es de desarrollo de circuitos electrónicos. La última área está destinada para que los alumnos desarrollen sus actividades en extra clase. Parte del área que se ocupó en la realización de éste proyecto fueron las de IEEE, las de Electrónicas Analógicas y desarrollo de circuitos electrónicos y el almacén de Mecatrónica.

La carrera de Ingeniería Electrónica tiene como misión “formar profesionales de excelencia con competencias en el ámbito de la Ingeniería Electrónica, motivados para la promoción del desarrollo profesional y el conocimiento científico y tecnológico, con actitud emprendedora, respeto al medio ambiente y apego a los valores cívicos y éticos”.

### 1.3 Antecedentes .

- ❖ La eliminación del carbón y de los nutrientes forma aguas residuales domésticas de baja resistencia por el método de tratamiento integrado por filtración de lecho de zeolita de cama granular expandida (EGSB)

Por: Xiao – Ming Li, Liang Guo, Qi Yang, Guang-Ming Zeng, De – Xiang Liao.

Facultad de ciencias ambientales e ingeniería, universidad humana, hunan Changsha.

3 de Abril 2006.

- ❖ Reactancia de lecho de lodo granular expandido anaerobio (EGSB) Para la eliminación del sulfuro por desnitrificación autotrófica.

Carlos Dinamarca.

Departamento de Procesos, Energía y Medio Ambiente; Facultad de Tecnología; Telemark University College Kjølnes anillo 56, 3918 Porsgrunn, Noruega.

2014.

- ❖ Startup and operation of anaerobic EGSB reactor treating palm oil mill effluent

ZHANG Yejian<sup>1</sup>, YAN Li<sup>2</sup>, CHI Lina<sup>1</sup>, LONG Xiuhua<sup>1</sup>,  
MEI Zhijian<sup>1</sup>, ZHANG Zhenjia<sup>1</sup>,<sup>\*</sup>

1. School of Environmental Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: yjzhang2007@yahoo.com  
2. College of Life and Environmental Science, Wenzhou University, Wenzhou 325000, China

Received 27 August 2007; revised 1 November 2007; accepted 23 January 2008

## **1.4 Planteamiento del problema.**

Las aguas contaminadas deben de tener un ambiente específico, tanto como pH, temperatura y oxígeno disuelto. Para poder tener esas tres lecturas en un biorreactor tipo EGSB, es necesario destapar este equipo y hacer las lecturas manual mente, teniendo en cuenta que es un equipo el cual no se puede estar destapando a todas horas y aparte que el agua no tiene un comportamiento constante en el transcurso del día, es decir, cambia dependiendo del clima en el exterior, esto hace que las lecturas requeridas para obtener el resultado requerido sean de gran dificultad y riesgosa.

Tener visibilidad del comportamiento del agua tratada en el biorreactor tipo EGSB. Por lo tanto demostrar por medio de una gráfica el comportamiento del pH, temperatura y oxígeno disuelto del agua que se está tratando en el biorreactor tipo EGSB.

## **1.5 Nombre del proyecto.**

Instrumentación electrónica para Biorreactores EGSB para el tratamiento de fuentes agroindustriales.

## **1.6 Objetivos.**

### **1.6.1 Objetivo general.**

Diseñar y construir una interfaz electrónica para la medición las variables características (oxígeno disuelto, pH, temperatura) de un Biorreactor EGSB para el tratamiento de fluentes agroindustriales.

### **1.6.2 Objetivos específicos.**

1. Conocer la teoría básica de Biorreactores tipo EGSB (Expanded Granular Sludge Bed - Reactores de manto de lodo granular expandido).
2. Diseñar y construir una interfaz electrónica aplicada al Biorreactor tipo EGSB mediante la identificación de las variables características del proceso, como el oxígeno disuelto, pH y temperatura.
3. Desarrollar una interfaz gráfica de usuario que permita el monitoreo de las variables características del Biorreactor tipo EGSB, utilizando interconectividad inalámbrica de sensores.

## **1.7 Justificación del proyecto.**

La ingeniería de procesos moderna se basa en el uso de modelos matemáticos rigurosos para realizar tareas de análisis, diseño, optimización y control. En el caso de bioprocesos, estos modelos suelen tener un carácter dinámico y no lineal, los cuales son tratados como procesos complejos. Los reactores de manto de lodo granular expandido (EGSB) presentan un gran potencial en el tratamiento de aguas con diferente carga contaminante y gran adaptabilidad a diferentes temperaturas de operación. En la actualidad, la utilización de herramientas computacionales para la adquisición y análisis de datos con el objetivo de realizar un verdadero diagnóstico y caracterización electrónica de birreactores EGSB se ha incrementado. En estos programas es posible el manejo de distintos tipos de datos y gráficos que representan los comportamientos descritos por los modelos. El diagnóstico basado en alarmas y una interfaz gráfica de usuario del proceso pueden ser usados como una importante herramienta para lograr un diseño óptimo.

## **1.8.- Delimitación del tema.**

En este proyecto se implementaron tres sensores diferentes los cuales facilitaron la lectura de los cambios de estados de dichas aguas tratadas y una gráfica la cual funciona las 24 horas del día para analizar su reacción en toda la madrugada. Los sensores implementados son: de pH, oxígeno disuelto y temperatura.



# Capítulo 2. Fundamento teórico.

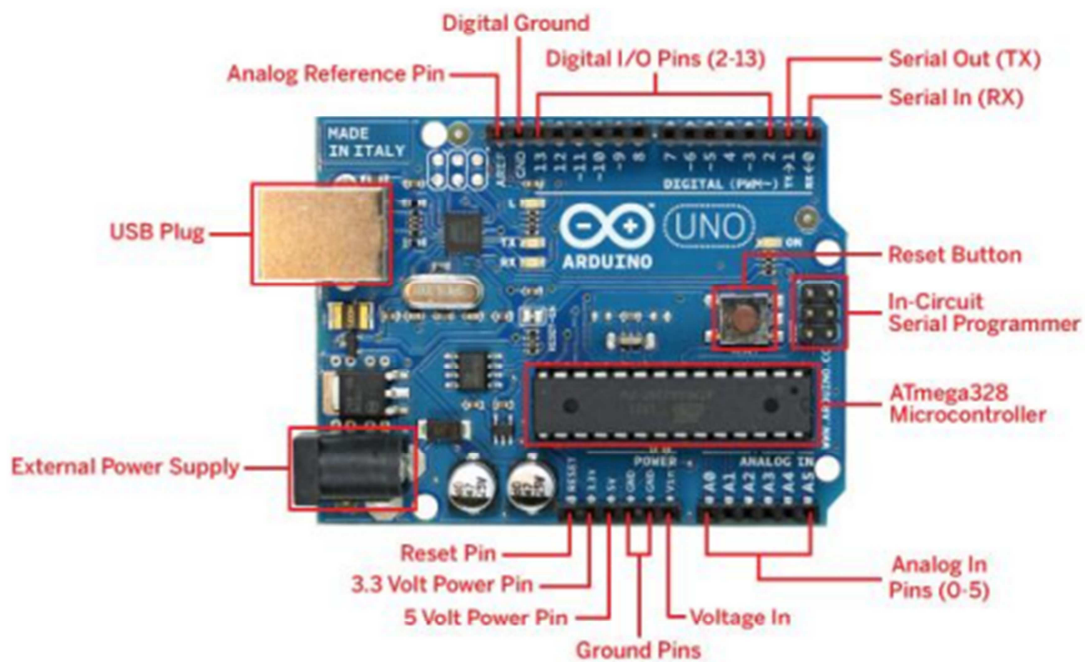
## 2.1 Programación en Arduino.

Arduino es una plataforma para computación física de código abierto basada en una simple tarjeta I/O y un entorno de desarrollo que implementa el procesamiento/idioma de cableado (Processing/Wiring). Arduino puede ser usado para desarrollar objetos autónomos interactivos o puede ser conectado a software en su ordenador (por ejemplo, Flash, Processing, MaxMSP). El IDE de código abierto puede ser descargado de forma gratuita (en la actualidad para Mac OS X, Windows y Linux).

## 2.2 Características:

Microcontrolador	ATmega328
Voltaje operativo	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (limite)	6-20V
Pines digitales E/S	14 (de los cuales 6 proporcionan salida PWM)
Pines de entradas analógicas	6
Corriente continua para pines E/S	40 mA
Corriente continua para pines de 3.3V	50 mA
Memoria Flash	32 KB (ATmega328) de los cuales 0.5 KB son para el bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Velocidad de reloj	16 MHz

A continuación se muestra en la figura 2 donde están ubicados los elementos más importantes que componen la placa Arduino Uno que son descritos de arriba abajo y de izquierda a derecha:



**Fig. 1 Elementos de la placa Arduino Uno.**

## 2.3 Estructura básica de un programa.

La estructura básica de programación de Arduino es bastante simple y divide la ejecución en dos partes: Setup, Loop

Setup() constituye la preparación del programa y loop() es la ejecución. En la función Setup() se incluye la declaración de variables y se trata de la primera función que se ejecuta en el programa. Esta función se ejecuta una única vez y es empleada para configurar las terminales de salida mediante pinMode(). La función loop() incluye el código a ser ejecutado continuamente (leyendo las entradas de la placa, salidas, etc.).

```

void setup()
{
  //Preparacion del Programa
}

void loop()
{
  //Ejecución
}

```

**Nota:** Al igual que en C se pueden introducir bloques de comentarios con /\* ... \*/.

## 2.4 Entradas/Salidas digitales y analógicas.

**digitalRead(pin).** Lee el valor desde un pin digital específico. Devuelve un valor HIGH o LOW.

**digitalWrite(pin, value).** Introduce un nivel alto (HIGH) o bajo (LOW) en el pin digital especificado. De nuevo, el pin puede ser especificado con una variable o una constante.

**analogRead(pin).** Lee el valor desde el pin analógico especificado con una resolución de 10 bits. Esta función solo funciona en los pines analógicos (0-5). El valor resultante es un entero de 0 a 1023. Los pines analógicos, a diferencia de los digitales no necesitan declararse previamente como INPUT o OUTPUT.

**analogWrite(pin, value).** Escribe un valor pseudo-analógico usando modulación por ancho de pulso (PWM) en un pin de salida marcado como PWM.

## 2.5 Tipos de Datos.

Arduino permite manejar los siguientes tipos de datos:

**Byte.** Almacena un valor numérico de 8 bits. Tienen un rango de 0-255.

**Int.** Almacena un valor entero de 16 bits con un rango de 32,767 a -32,768.

**Long.** Valor entero almacenado en 32 bits con un rango de 2,147,483,647 a – 2,147,483, 648.

**Float.** Tipo coma flotante almacenado en 32 bits con un rango de 3.4028235E+38 a -3.4028235E+38.

**Arrays.** Se trata de una colección de valores que pueden ser accedidos con un número de índice (el primer valor del índice es 0).

## 2.6 funciones de tiempo.

**delay(ms).** Realiza una pausa en el programa la cantidad de tiempo en milisegundos especificada en el parámetro (máximo 1000, mínimo 1).

**millis().** Devuelve la cantidad de milisegundos que lleva la placa Arduino ejecutando el programa actual como un valor long unsigned. Después de 9 horas el contador vuelve a 0.

## **2.7 LabView.**

El software LabVIEW es ideal para cualquier sistema de medidas y control, es una plataforma y entorno de desarrollo para diseñar sistemas, resolver problemas, productividad acelerada y constante innovación. Con un lenguaje de programación visual gráfico. Recomendado para sistemas hardware y software de pruebas, control y diseño, simulado o real y embebido, pues acelera la productividad. El lenguaje que usa se llama lenguaje G, donde la G simboliza que es lenguaje Gráfico.

Los programas desarrollados con LabVIEW se llaman Instrumentos Virtuales, o VIs, y su origen provenía del control de instrumentos, aunque hoy en día se ha expandido ampliamente no sólo al control de todo tipo de electrónica (Instrumentación electrónica) sino también a su programación embebida, comunicaciones, matemáticas, etc. Un lema tradicional de LabVIEW es: "La potencia está en el Software", que con la aparición de los sistemas multinúcleo se ha hecho aún más potente. Entre sus objetivos están el reducir el tiempo de desarrollo de aplicaciones de todo tipo (no sólo en ámbitos de Pruebas, Control y Diseño) y el permitir la entrada a la informática a profesionales de cualquier otro campo. LabVIEW consigue combinarse con todo tipo de software y hardware, tanto del propio fabricante -tarjetas de adquisición de datos, PAC, Visión, instrumentos y otro Hardware- como de otros fabricantes. [4]

### **2.7.1 Utilizando LabVIEW.**

En este capítulo se explican los conceptos básicos en LabVIEW.

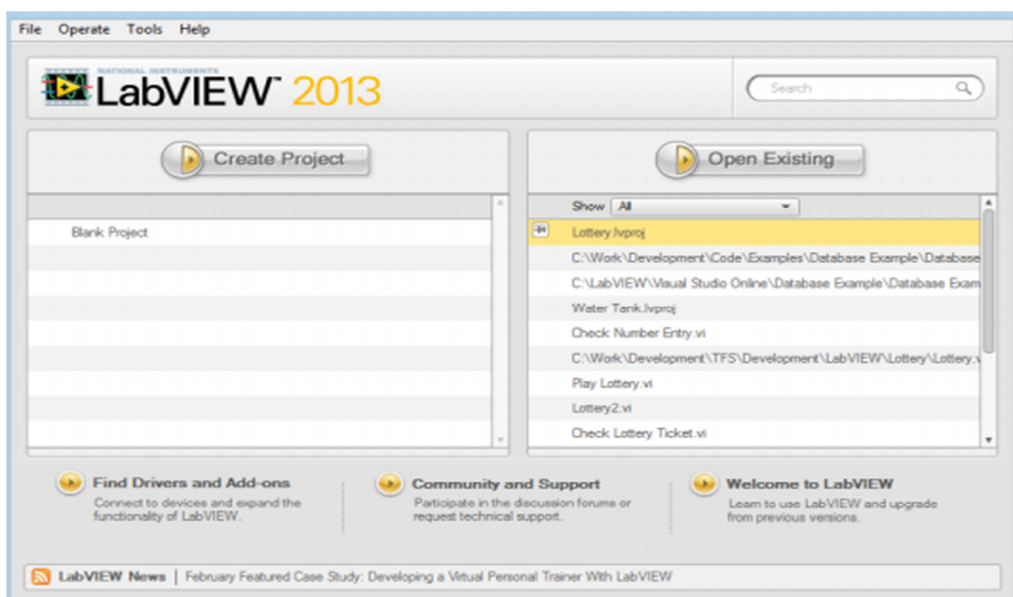
Los temas son los siguientes:

- El LabVIEW Medio Ambiente
- Panel Frontal y Diagrama de Bloques
- Paletas: Paleta de control, funciones de la paleta, paleta de Herramientas
- Tipos de datos
- Nodos de Propiedad

## 2.7.2 LabVIEW para el Medio Ambiente.

Programas de LabVIEW se llaman Instrumentos Virtuales o VIs, porque su apariencia y funcionamiento imitan los instrumentos físicos, como osciloscopios y multímetros. LabVIEW contiene un amplio conjunto de herramientas para la adquisición de analizar, visualizar y almacenar los datos, así como herramientas para ayudarle a solucionar su código.

Al abrir LabVIEW, primera vez que vienen a la ventana "Getting Started"



**Fig. 2.3 Software LabVIEW**

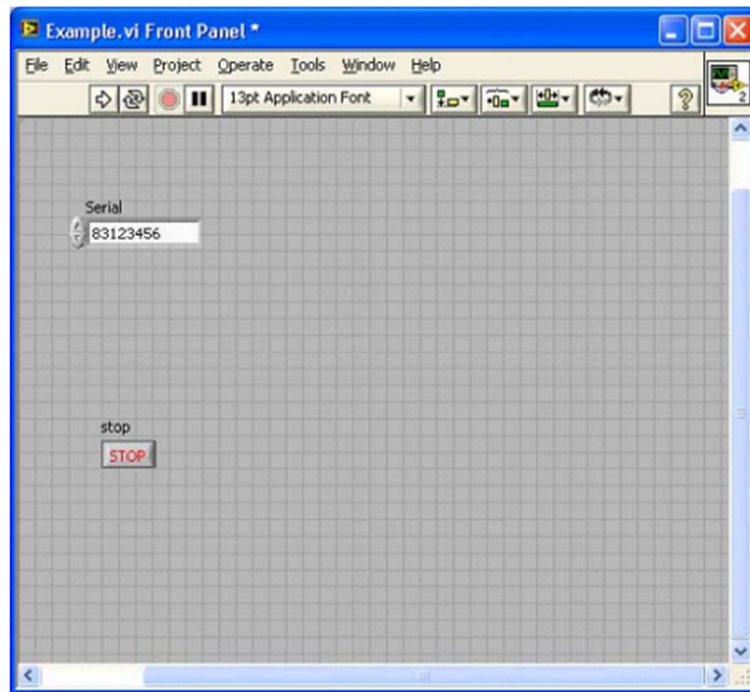
Con el fin de crear un nuevo VI, seleccione "En blanco VI" o con el fin de crear un nuevo proyecto de LabVIEW, seleccione "proyecto vacío".

Al abrir un VI en blanco, aparece una ventana de panel frontal sin título. Esta ventana muestra el panel frontal y es una de las dos ventanas de LabVIEW que se utilizan para construir un VI. La otra ventana contiene el diagrama de bloques. Las secciones siguientes describen el panel frontal y el diagrama de bloques.

### 2.7.3 Propiedad de los componentes.

- Panel frontal

Cuando haya creado un nuevo VI o seleccionado existente, aparecerá el panel frontal y el diagrama de bloques para que especifique VI.



**Fig.2.4 Panel Frontal**

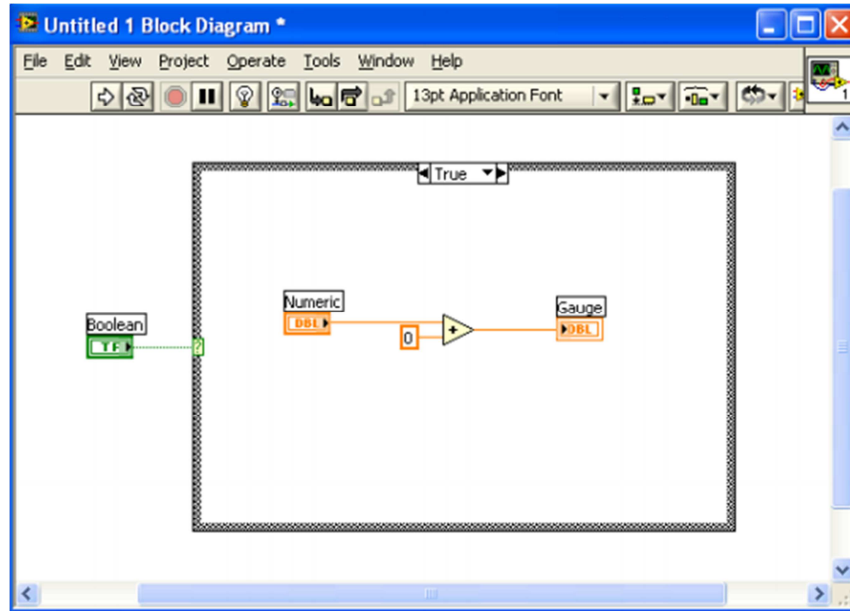
En LabVIEW, usted construye una interfaz de usuario, o el panel frontal, con controles e indicadores, que son los terminales de entrada y salida interactivas. Los controles son perillas, botones, diales y otros dispositivos de entrada. Los indicadores son gráficas, LEDs y otras pantallas.

Los controles simulan dispositivos de entrada del instrumento y de suministro de datos al diagrama de bloques. Los indicadores simulan los dispositivos de salida de instrumentos y datos de visualización del diagrama de bloques adquiere o genera.

- Diagrama de bloques

Después de construir la interfaz de usuario, se agrega código usando VIs y estructuras para controlar los objetos del panel frontal. El diagrama de bloques

contiene el código. En cierto modo, el diagrama de bloques se asemeja a un diagrama de flujo.



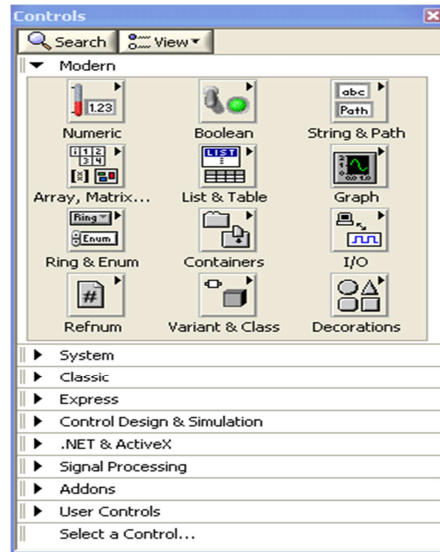
**Fig. 2.5 Diagrama de bloque**

Después de construir el panel frontal, se agrega código utilizando representaciones gráficas de funciones para controlar los objetos del panel frontal. El diagrama de bloques contiene el código fuente gráfica. Objetos del panel frontal aparecen como terminales, en el diagrama de bloques. Objetos de diagrama de bloques incluyen terminales, subVIs, funciones, constantes, estructuras y cables, que transfieren datos entre otros objetos de diagrama de bloques.

#### - Controla la gama de colores

Los controles y las funciones paletas contienen sub paletas de objetos que puede utilizar para crear un VI. Al hacer clic en un icono de la paleta de sub, la totalidad de los cambios en la gama de colores de la paleta sub seleccionado. Para utilizar un objeto en las gamas de colores, haga clic en el objeto y colocarlo en el panel frontal o diagrama de bloques. La paleta de controles sólo está disponible en el panel frontal. La paleta de controles contiene los controles e indicadores que se utilizan para construir el panel frontal.

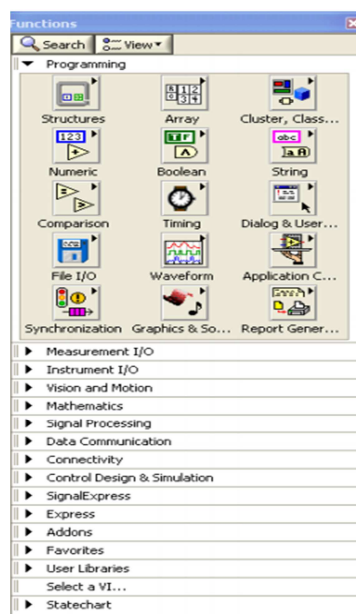




**Fig. 2.6 LabVIEW incluye diferentes controles para interactuar con la aplicación.**

#### - Paleta de Función

La paleta de funciones está disponible sólo en el diagrama de bloques. La paleta de funciones contiene los VIs y funciones que utiliza para construir el diagrama de bloques.




**Fig. 2.7 Paleta de función**


## - Barra de herramientas


A continuación vemos la Barra de Herramientas de LabVIEW:





Los comportamientos de los diferentes botones son los siguientes:


 Haga clic en el botón Ejecutar para ejecutar un VI. LabVIEW compila el VI, si es necesario. Puede ejecutar un VI si el botón Run aparece como un sólido flecha blanca. La flecha blanca sólida, que se muestra más arriba, también indica que puede utilizar el VI como un subVI si crea un panel conector para el VI.

 Mientras el VI se ejecuta, el botón Run aparece como se muestra a la izquierda si el VI es un VI de alto nivel, lo que significa que no tiene personas que llaman y por lo tanto no es un subVI.


 Si el VI que se está ejecutando es un subVI, el botón Run aparece como se muestra a la izquierda.

 El botón Run aparece roto, muestra a la izquierda, cuando el VI que está creando o editando contiene errores. Si el botón Ejecutar todavía aparece roto después de acabado el cableado del diagrama de bloques, el VI está roto y no se puede ejecutar. Haga clic en este botón para mostrar la ventana de la lista de errores, que enumera todos los errores y advertencias.

 Haga clic en el botón Ejecutar Continualmente, se muestra a la izquierda, para ejecutar el VI hasta que se cancela o hacer una pausa en la ejecución. También puede hacer clic en el botón de nuevo para desactivar el funcionamiento continuo.

 Mientras el VI se ejecuta, el botón Abort Execution, muestra a la izquierda, aparece. Haga clic en este botón para detener el VI inmediatamente si no hay otra forma de detener el VI. Si más de un nivel superior corriendo VI utiliza el VI, se atenúa el botón.

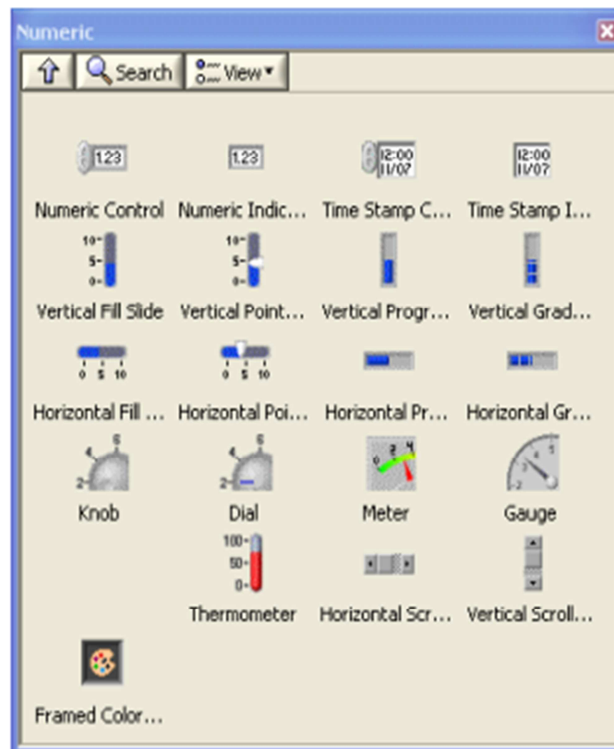
Nota: Evite utilizar el botón Abort Execution para detener un VI. O deje que el VI complete sus datos de flujo o el diseño de un método para detener el VI mediante programación. Al hacerlo, el VI se encuentra en un estado conocido. Por ejemplo, colocar un botón en el panel frontal que detiene el VI cuando hace clic en él.

 Haga clic en el botón de pausa, se muestra a la izquierda, para hacer una pausa en un VI corriendo. Al hacer clic en el botón Pausa, LabVIEW destaca en el diagrama de bloques de la ubicación donde se detuvo la ejecución, y

el botón Pausa aparece en rojo. Haga clic en el botón de nuevo para continuar ejecutando el VI.

#### Paleta Sub Numérico

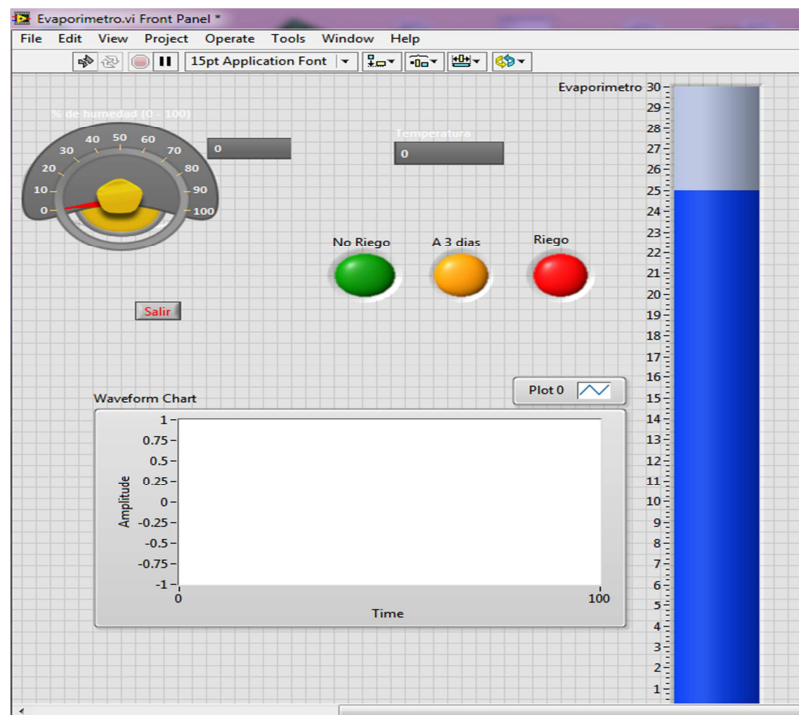
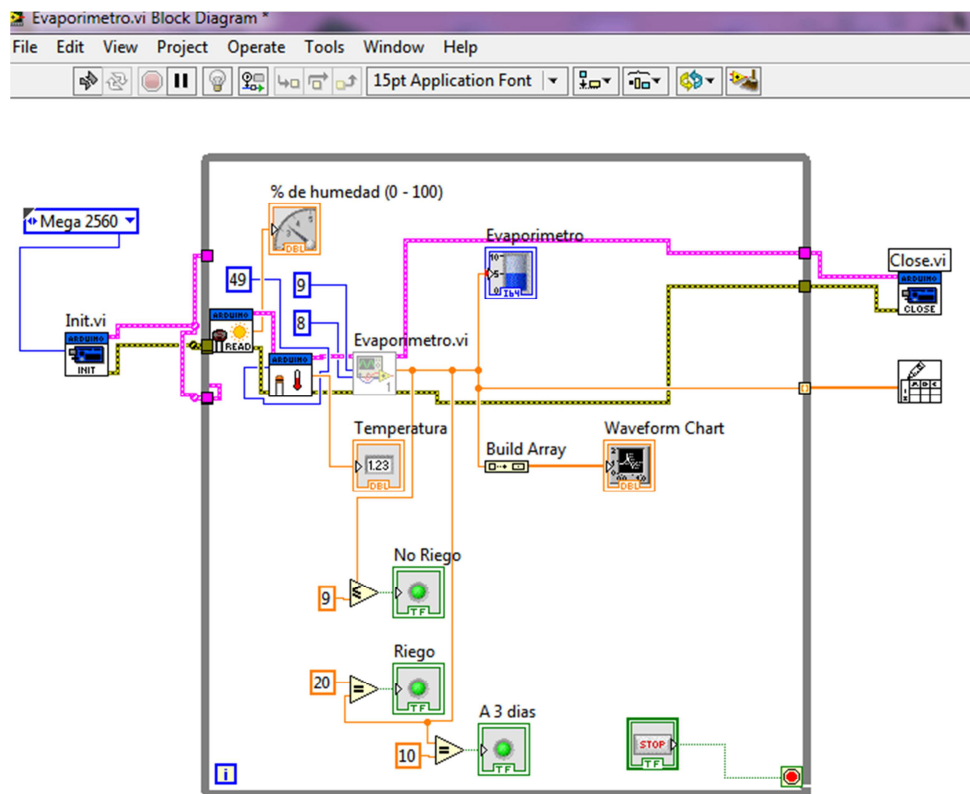
"Control Numérico" y "indicador numérico" son los objetos más utilizados en la paleta sub numérico.



**Fig. 2.8 Paleta Sub-Numérico**

#### - Interfaz de usuario

Después de programar en el software NI LabVIEW, generalmente se construyó una interfaz de usuario. Visualizando los datos de entrada adquiridos en tiempo real, la siguiente figura muestra el diseño y programación (diagrama de bloques) como se fue desarrollando la interfaz de usuario (datos que captura en la pantalla).



**Fig.2.9 Programas en el plano frontal.**

## 2.8 Microcontrolador.

Tomando en cuenta las características de los diferentes microcontroladores se seleccionó el que tiene más ventajas para su uso. Este circuito integrado forma parte de las placas Arduino mega, que pertenece a la categoría de los microcontroladores.

Presenta características muy versátiles que le permiten aplicaciones en un gran número de variedades, además de contar con módulos Ethernet con puerto RJ-45 para la conexión a internet y la posibilidad de enviar datos. Arduino mega 2560 [1]

## 2.9 Arduino MEGA 2560.

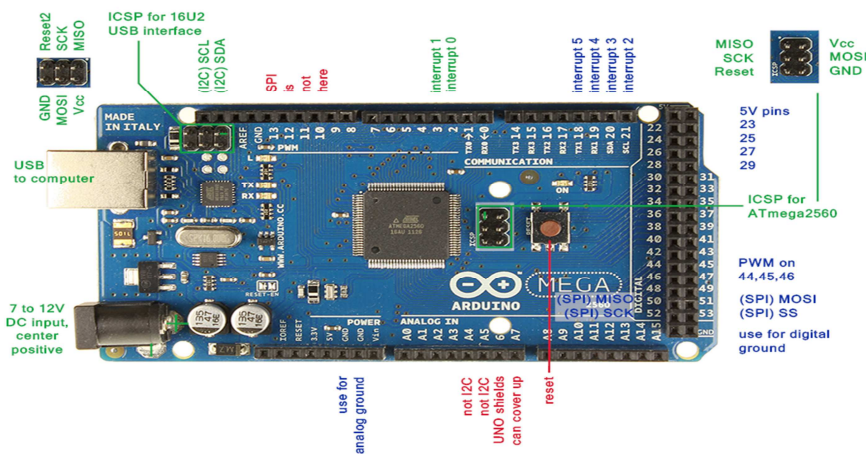
Arduino es una plataforma para computación física de código abierto basada en una simple tarjeta I/O y un entorno de desarrollo que implementa el procesamiento/idioma de cableado (Processing/Wiring). Arduino puede ser usado para desarrollar objetos autónomos interactivos o puede ser conectado a software en su ordenador (por ejemplo, Flash, Processing, MaxMSP). El IDE de código abierto puede ser descargado de forma gratuita (en la actualidad para Mac OS X, Windows y Linux).

### Características:

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

**Tabla 2.1 Características eléctricas del Arduino MEGA 2560**

A continuación se muestra en la Ilustración 2 donde están ubicados los elementos más importantes que componen la placa Arduino MEGA 2560 que son descritos de arriba abajo y de izquierda a derecha: V



**Fig 2.10 Estructura del Arduino Mega 2560**

## 2.10 Estructura básica de un programa.

La estructura básica de programación de Arduino es bastante simple y divide la ejecución en dos partes:

Setup  
Loop

Setup() constituye la preparación del programa y loop() es la ejecución. En la función Setup() se incluye la declaración de variables y se trata de la primera función que se ejecuta en el programa. Esta función se ejecuta una única vez y es empleada para configurar las terminales de salida mediante pinMode(). La función loop() incluye el código a ser ejecutado continuamente (leyendo las entradas de la placa, salidas, etc.).

```
void setup()
{
  //Preparacion del Programa
}

void loop()
{
  //Ejecución
}
```

**Nota:** Al igual que en C se pueden introducir bloques de comentarios con /\* ... \*/.

## Entradas/Salidas digitales y analógicas.

**digitalRead(pin).** Lee el valor desde un pin digital específico. Devuelve un valor HIGH o LOW.

**digitalWrite(pin, value).** Introduce un nivel alto (HIGH) o bajo (LOW) en el pin digital especificado. De nuevo, el pin puede ser especificado con una variable o una constante.

**analogRead(pin).** Lee el valor desde el pin analógico especificado con una resolución de 10 bits. Esta función solo funciona en los pines analógicos (0-5). El valor resultante es un entero de 0 a 1023. Los pines analógicos, a diferencia de los digitales no necesitan declararse previamente como INPUT o OUTPUT.

**analogWrite(pin, value).** Escribe un valor pseudo-analógico usando modulación por ancho de pulso (PWM) en un pin de salida marcado como PWM.

### 2.10.1 Tipos de Datos.

Arduino permite manejar los siguientes tipos de datos:

**Byte.** Almacena un valor numérico de 8 bits. Tienen un rango de 0-255.

**Int.** Almacena un valor entero de 16 bits con un rango de 32,767 a -32,768.

**Long.** Valor entero almacenado en 32 bits con un rango de 2,147,483,647 a – 2,147,483, 648.

**Float.** Tipo coma flotante almacenado en 32 bits con un rango de 3.4028235E+38 a -3.4028235E+38.

**Arrays.** Se trata de una colección de valores que pueden ser accedidos con un numero de índice (el primer valor del índice es 0).

### 2.10.2 Funciones de Tiempo.

**delay(ms).** Realiza una pausa en el programa la cantidad de tiempo en milisegundos especificada en el parámetro (máximo 1000, mínimo 1).

**millis().** Devuelve la cantidad de milisegundos que lleva la placa Arduino ejecutando el programa actual como un valor long unsigned. Después de 9 horas el contador vuelve a 0. [1]

**2.11 pH Electrode model: ASP200-2-1M/BNC serialNO:  
E130722224.**



**Descripción del producto:**

El 3550 es un electrodo de combinación económica ideal para sus aplicaciones de uso general en el laboratorio o en el campo. Tiene las cualidades resistentes al impacto de un cuerpo de epoxi.

El electrodo proporciona una respuesta rápida, estable y es ideal para las lecturas de pH prolongados. El diseño sellado, relleno de gel prácticamente no requiere mantenimiento ya que no hay necesidad de rellenar la cámara de referencia.

El 3550 medidas 120 mm (largo) x 12 mm (diámetro), puede soportar temperaturas de hasta 80 ° C y mide los niveles de pH de 0 a 14.

Tiene un cable de 1 metro terminada por un conector BNC.

**Especificaciones del producto:**

Propiedades de los electrodos:

Tipo de electrodo de combinación sencilla - Junction

Rango de medición de pH de 0 a 14

Tiempo de respuesta del electrodo Max 10 s

Min temperatura de funcionamiento 0 ° C

Temperatura de funcionamiento máxima 80 ° C.



**Propiedades físicas:**

Tubo exterior de material plástico resistente al impacto

Tipo de conector BNC

Longitud 120 mm

12 mm de diámetro

Longitud del cable 1 m

**2.12 Sensor de temperatura DS18B20.**

El sensor DS18B20 permite medir temperaturas de hasta 125°C de forma fácil y además está sellado en un **envoltorio estanco** que permite sumergirlo en un líquido o protegerlo de la intemperie. Dado que es un sensor digital, la señal leída no se degrada debido a la distancia del cableado. Puede funcionar en modo 1-Wire con una precisión de  $\pm 0.5^{\circ}\text{C}$  con una resolución de 12 bits. También pueden utilizarse varios sensores sobre el mismo pin ya que internamente viene programado con un ID único de 64 bits para diferenciarlos. El rango de funcionamiento es de 3 a 5V por lo que se puede utilizar en prácticamente cualquier sistema de que use microcontroladores.

### Características del cable:

- Tubo de acero inoxidable de 6mm de diámetro por 30mm de largo
- Largo: 91cm
- Diámetro: 4mm
- Contiene un sensor de temperatura DS18B20

### Conexión:

En función de la producción, los cables del sensor pueden variar pero seguirán según estas especificaciones:

- Si tienes 4 cables: Rojo es Vcc (positivo), Negro es GND (negativo) y Blanco es el cable de datos. La malla es GND.
- Si tienes 3 cables: Rojo es Vcc (positivo), Azul/Negro es GND (negativo) y Amarillo/Blanco es el cable de datos.

### Características del sensor DS18B20:

- Rango de temperatura: -55 a 125°C
- Resolución: de 9 a 12 bits (configurable)
- Interfaz 1-Wire (Puede funcionar con un solo pin)
- Identificador interno único de 64 bits
- Múltiples sensores pueden compartir el mismo pin
- Precisión:  $\pm 0.5^{\circ}\text{C}$  (de  $-10^{\circ}\text{C}$  a  $+85^{\circ}\text{C}$ )
- Tiempo de captura inferior a 750ms
- Alimentación: 3.0V a 5.5V

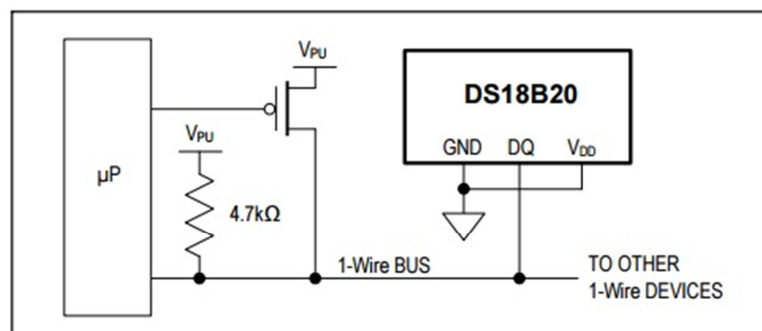


Diagrama de conexión

## 2.13 Sensor de oxígeno disuelto.



La sonda oxígeno disuelto galvánica es un dispositivo pasivo que genera un pequeño voltaje de 0 mV a 47mv dependiendo de la saturación de oxígeno de la membrana de detección HDPE. Este voltaje puede ser fácilmente leído por un análogo o por multímetro al convertidor digital.

El sensor es un tubo con una varilla de zinc (ánodo) sumergido en un electrolito. El elemento de detección es la membrana de detección HDPE comprimido contra un disco de plata (cátodo).

El oxígeno disuelto se expresa en mg / l. Hay muchos factores que deben ser tomados en cuenta en la lectura de oxígeno disuelto, como la salinidad y la temperatura. Por lo tanto, no hay ecuación lineal simple que nos devuelva el valor de oxígeno disuelto a partir de la tensión de salida de la sonda.

**Especificaciones:**

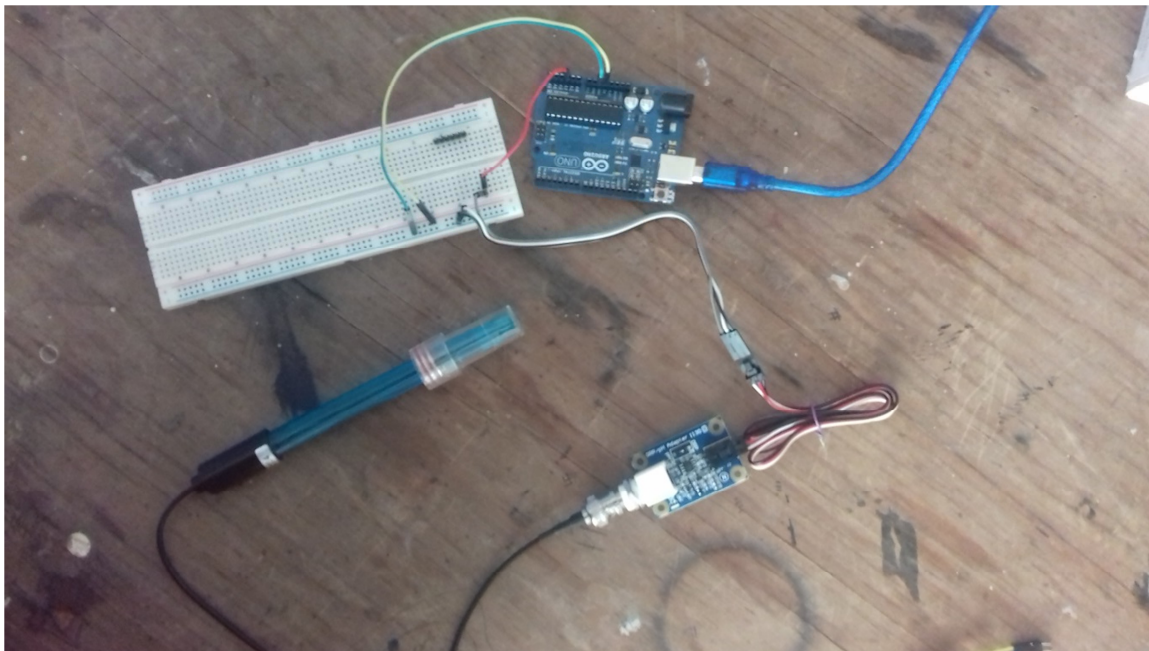
- Rango: 0-20 mg / L.
- Temperatura máxima: 50 ° C.
- Presión máxima: 690 kPa (100 psi).
- La profundidad máxima de 60 m (197 pies).
- Longitud del cable: 1 metro.
- Conector BNC.

**Características:**

- Dimensiones: 16.50x116.00 mm.
- Peso: 52 g.
- Marca: Atlas Scientific.
- Cantidad: 1 pieza.

## Capítulo 3. Desarrollo e implementación del proyecto.

### 3.1 conexión de sensor de pH al arduino.



El sensor pH cuenta con tres puertos (rojo positivo, negro tierra y blanco recibe la señal) se alimentó con 5 volts y se utilizó un puerto analógico del arduino (A2) para obtener una medida más exacta.

### 3.1.1 Programación en arduino.

```
/*  
  
# This sample code is used to test the pH electrode.  
  
# Date   : 2017.01.10.  
  
# Product: ASP200-2-1M/BNC  
  
# SKU    : E130722224   */  
  
#define SensorPin A2      //pH electrode output to Arduino Analog Input 2  
  
#define Offset 0.00       //deviation compensate  
  
#define LED 13  
  
#define samplingInterval 20  
  
#define printInterval 800  
  
#define ArrayLenth 40    //times of collection  
  
int pHArray[ArrayLenth]; //Store the average value of the sensor feedback  
  
int pHArrayIndex=0;  
  
void setup(void)  
{  
    pinMode(LED,OUTPUT);  
  
    Serial.begin(9600);  
  
    Serial.println("pH meter experiment!"); //Test the serial monitor  
}  
  
void loop(void)  
{  
  
    static unsigned long samplingTime = millis();  
  
    static unsigned long printTime = millis();  
  
    static float pHValue,voltage;
```

```

if(millis()-samplingTime > samplingInterval)
{
    pHArray[pHArrayIndex++]=analogRead(SensorPin);
    if(pHArrayIndex==ArrayLenth)pHArrayIndex=0;
    voltage = avergearray(pHArray, ArrayLenth)*5.0/1024;
    pHValue = 3.5*voltage+Offset;
    samplingTime=millis();
}

if(millis() - printTime > printInterval) //Every 800 milliseconds, print a numerical,
convert the state of the LED indicator
{
    Serial.print("Voltage:");
        Serial.print(voltage,2);
        Serial.print("   pH value: ");
    Serial.println(pHValue,2);
        digitalWrite(LED,digitalRead(LED)^1);
        printTime=millis(); }
}

double avergearray(int* arr, int number){
    int i;
    int max,min;
    double avg;
    long amount=0;
    if(number<=0){
        Serial.println("Error number for the array to avraging!/n");
    }
}

```

```

    return 0;
}
if(number<5){ //less than 5, calculated directly statistics
    for(i=0;i<number;i++){
        amount+=arr[i];
    }
    avg = amount/number;
    return avg;
}else{
    if(arr[0]<arr[1]){
        min = arr[0];max=arr[1];
    }
    else{
        min=arr[1];max=arr[0];
    }
    for(i=2;i<number;i++){
        if(arr[i]<min){
            amount+=min;    //arr<min
            min=arr[i];
        }else {
            if(arr[i]>max){
                amount+=max;    //arr>max
                max=arr[i];
            }else{
                amount+=arr[i]; //min<=arr<=max
            }
        }
    }
}

```



```

    }
  }//if
} //for

avg = (double)amount/(number-2);

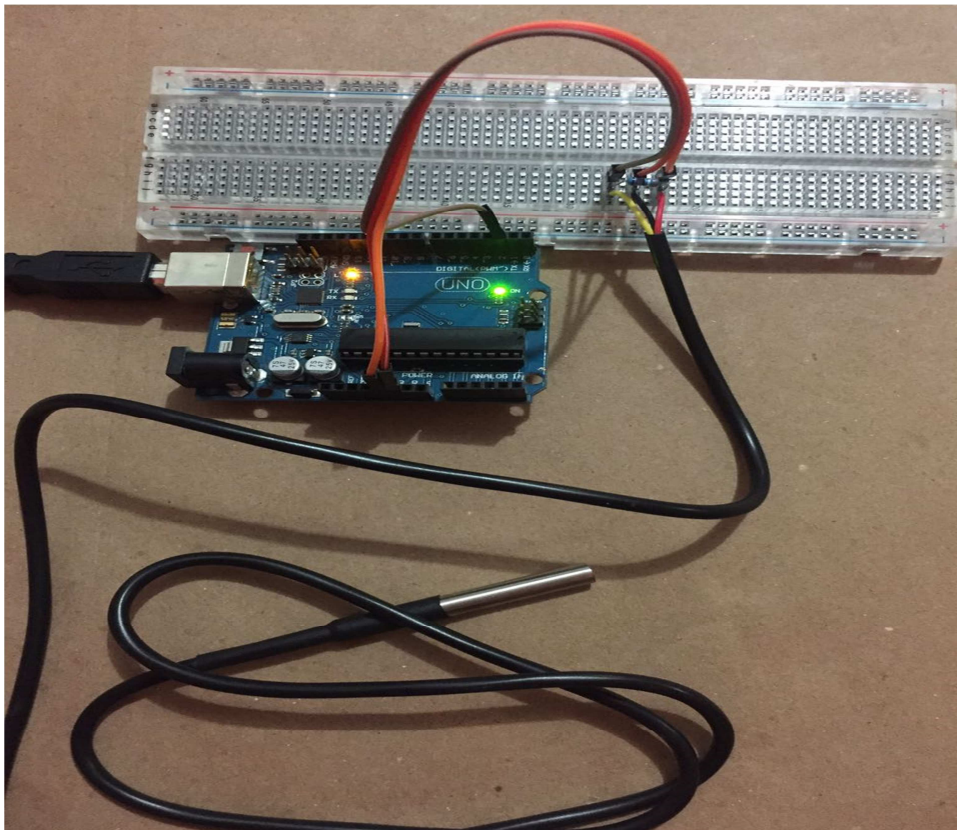
} //if

return avg;

}

```

### 3.1.2 *conexión del sensor DS18B20 estanco al arduino.*



Al igual que el sensor de pH, el sensor de temperatura cuenta con tres puertos (positivo, tierra y la señal).

### 3.1.3 programacion en arduino.

```
#include <OneWire.h>
```

```
#include <DallasTemperature.h>
```

```
OneWire ourWire(2);
```

```
DallasTemperature sensors(&ourWire);
```

```
void setup() {
```

```
  delay(1000);
```

```
  Serial.begin(9600);
```

```
  sensors.begin();
```

```
}
```

```
void loop() {
```

```
  sensors.requestTemperatures();
```

```
  float temp= sensors.getTempCByIndex(0);
```

```
  Serial.print("Temperatura= ");
```

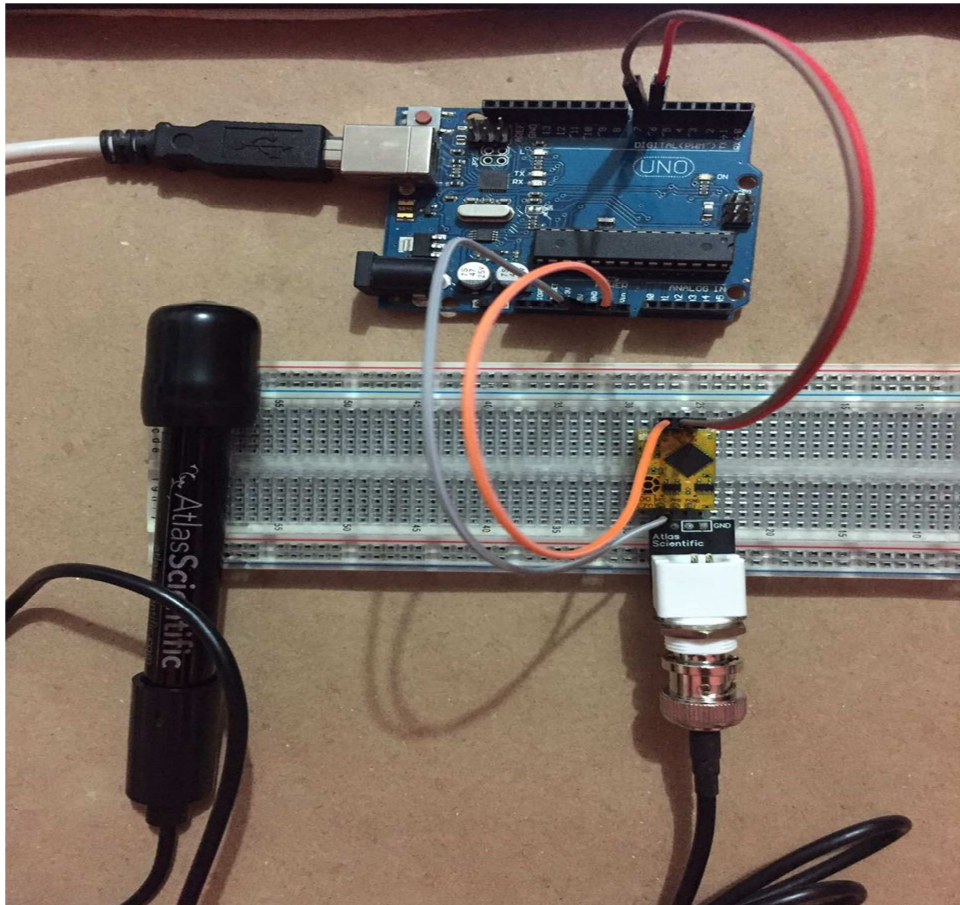
```
  Serial.print(temp);
```

```
  Serial.println(" C");
```

```
  delay(100);
```

```
}
```

### 3.1.4 conexión de sensor de oxígeno disuelto al arduino.

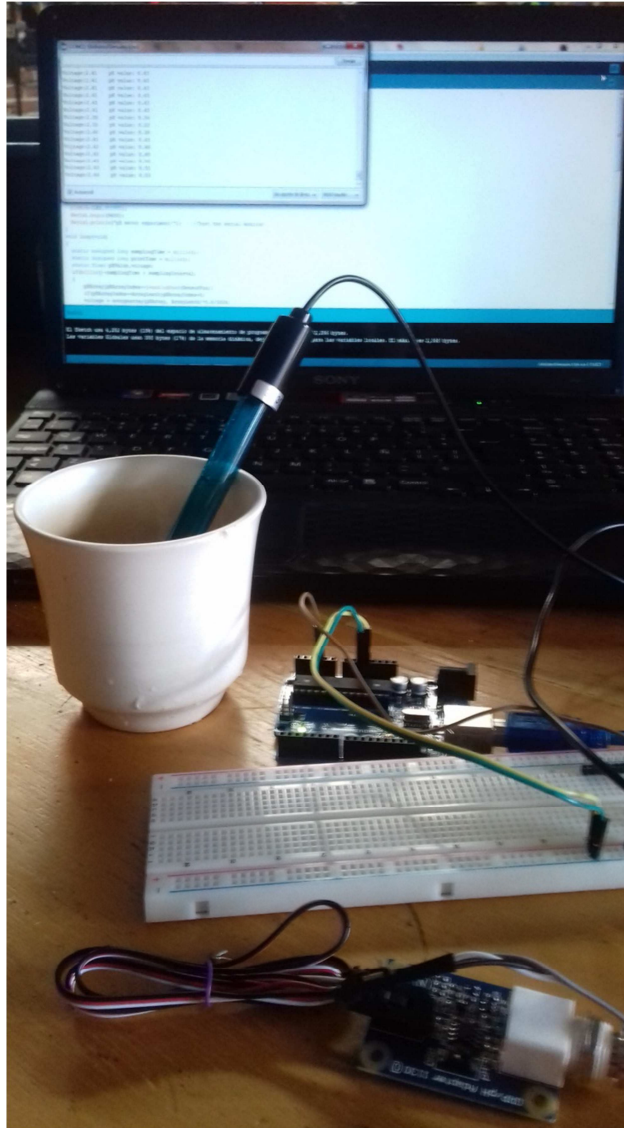


A diferencia de los sensores pasados, en el sensor de oxígeno disuelto se utilizaron dos señales de salida

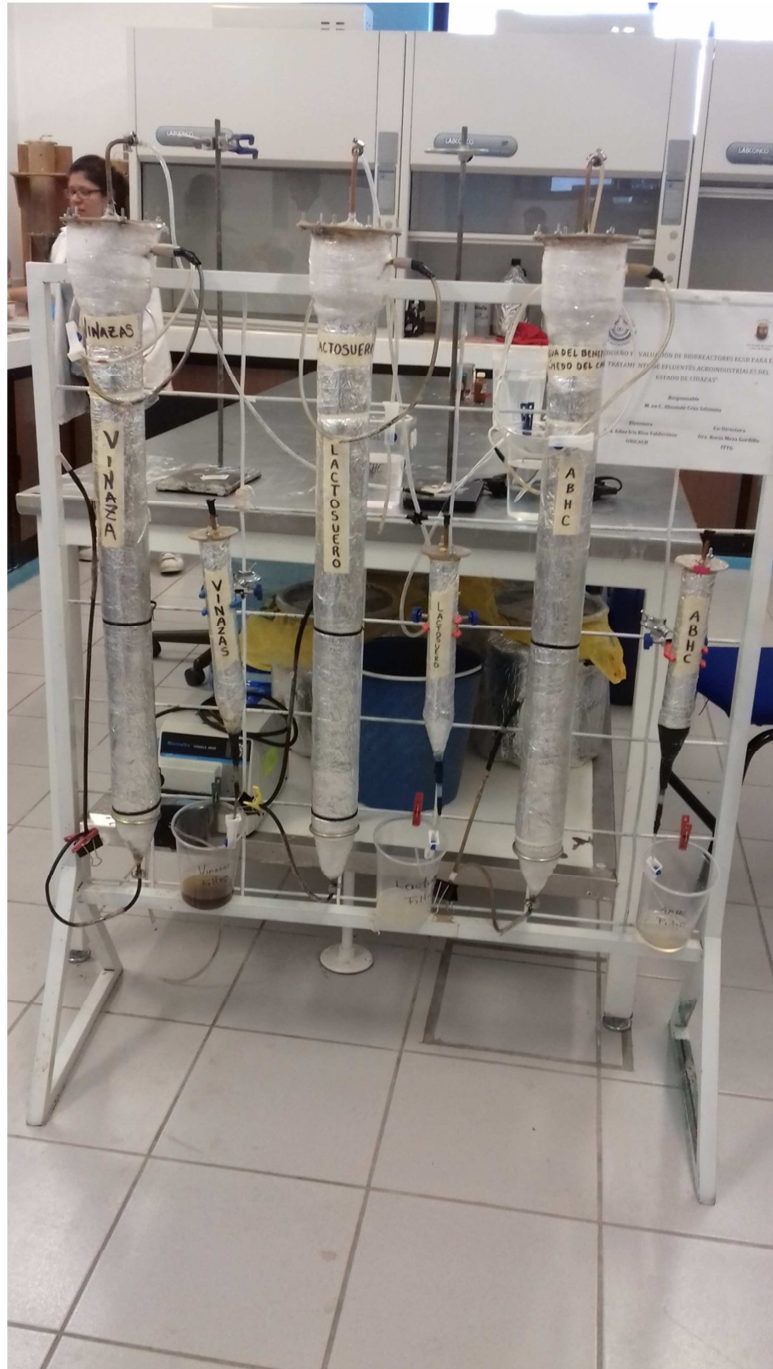
### 3.1.5 programacion en arduino.

```
#include <
Serial3.begin(9600);
void getOxigeno(){
  if (Serial3.available() ){
    while (sensor_DO_stringcomplete==false) {
      char inchar_od = (char)Serial3.read();
      sensorDOstring += inchar_od;
      if(inchar_od == '\r') {
        sensor_DO_stringcomplete = true;
      }
    }
    if(sensor_DO_stringcomplete){
      Data_OD=sensorDOstring;
      sensorDOstring="";
      sensor_DO_stringcomplete=false;
      Serial.print("I received: ");
      Serial.println(Data_OD);
    }
  }
}
```

## CAPITULO 4. Anexo de imágenes.

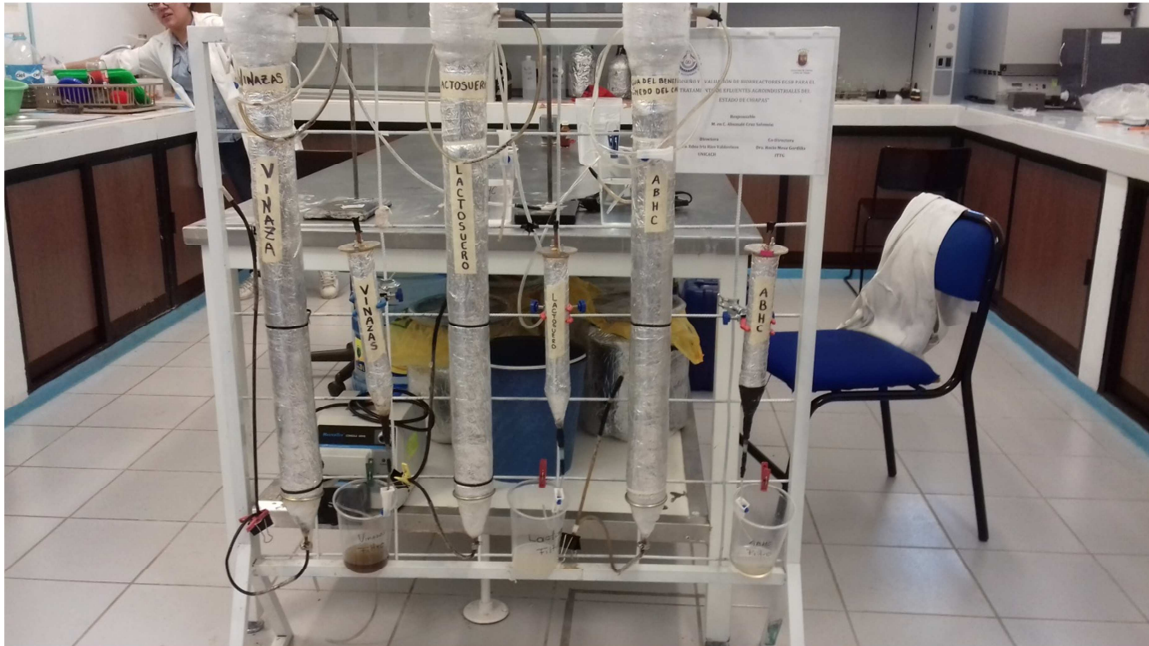


Prueba de la medición del pH del agua del tubo.

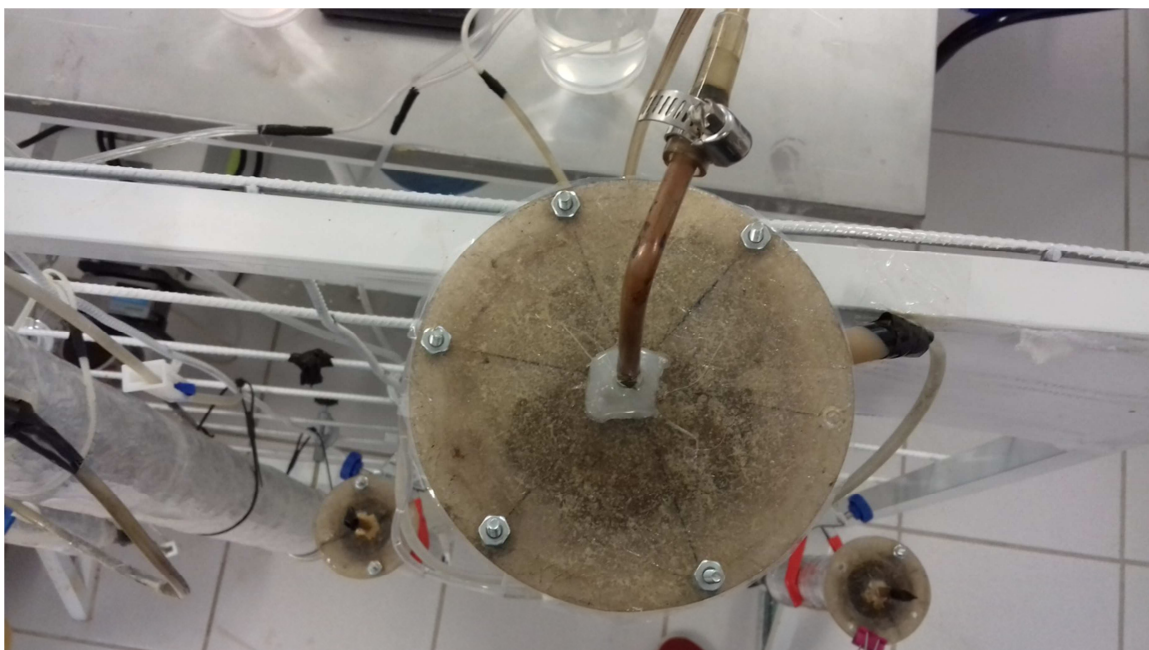


Biorreactor tipo EGSB





Biorreactor tipo EGSB



Biorreactor tipo EGSB.

## ***5. Observaciones y sugerencias.***

Como pudimos observar, en el transcurso del desarrollo de este proyecto, nos dimos cuenta que el tratado de aguas residuales no es una tarea fácil y es un trabajo el cual se le debe dedicar mucho tiempo para estar en constante chequeo del cambio que sufren dichas aguas. Pero con la ayuda de la tecnología este chequeo se puede hacer más fácil, certero y seguro, pues el biorreactor es un equipo el cual no se puede estar destapando a cada rato y además solo puedes obtener el resultado del pH, temperatura y oxígeno disuelto una sola vez, y si quieres saber nueva mente esos tres datos, debes de hacer otra vez este proceso. Con la instalación de estos tres sensores se redujeron todos esos problemas, su instalación es permanente y eso te da a tener un análisis del comportamiento de las aguas residuales las 24 horas del día.



## **6.-Conclusión.**

El trato de las aguas contaminadas, hoy en día ya es de suma importancia ya que existen muchas industrias que ocasionan este problema y es mucho lo que contaminan. Los factores que influyen en el agua contaminada son tres, que como ya vimos son, el pH, la temperatura y el oxígeno disuelto, es por eso que se debe de tener en constante monitoreo.

Los bioreactores son un buen recurso para tratarlas y así poder reutilizar esas aguas. Y ya vimos que si le instalamos los sensores de pH, temperatura y oxígeno disuelto, El proceso de reciclado se hace más fácil, porque tienes un control y análisis de la forma en que se comportan dichas aguas contaminadas.

## 7.- Referencias bibliográficas.

[https://www.dfrobot.com/wiki/index.php/PH\\_meter\\_V1.1\\_SKU:SEN0161](https://www.dfrobot.com/wiki/index.php/PH_meter_V1.1_SKU:SEN0161)

<https://hetpro-store.com/sensor-de-oxigeno-disuelto/>

Anaerobic expanded granular sludge bed (EGSB) reactor for the removal of sulphide by autotrophic denitrification

Startup and operation of anaerobic EGSB reactor treating palm oil mill effluent

Removal of carbon and nutrients from low strength domestic wastewater by expanded granular sludge bed-zeolite bed filtration (EGSB-ZBF) integrated treatment concept

Effects of ORP%2c recycling rate%2c and HRT on simultaneous sulfate reduction and sulfur production in expanded granular sludge bed (EGSB) reactors under micro-aerobic conditions for treating molasses distillery wastewa