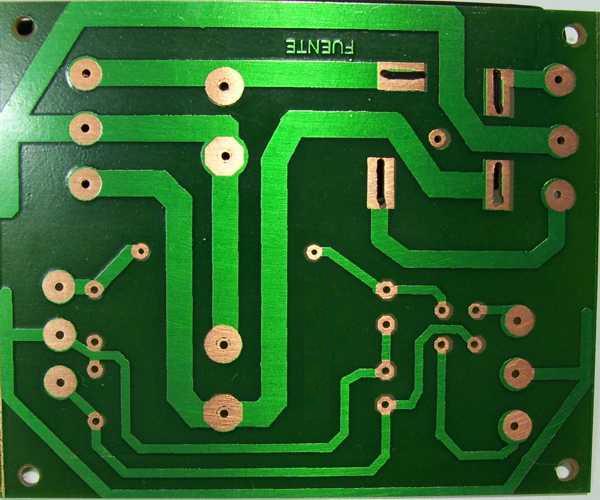
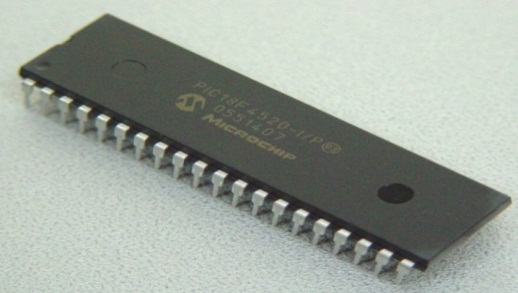
**2012**

**Jonathan Olier**

**Universidad Simón Bolívar**

**10/04/2012**

**Manual para USIBA**

********

Diseño de una Interfaz grafica por conexión USB a una tarjeta electrónica que trabaja con el micro-controlador 18F4550 para recepción señales análogas de un dispositivos Biomédicos.

# *SOBRE EL MANUAL*

Manual Para USIBA:

Versión 1 / Junio 2012

Manual en línea:

<http://www.altiviaot.probandando.com/home/>

Proyecto por Ing. Jonathan Olier e ing. Isabel de la Cruz

* Diseño y Diagramación: Innovador y explicito por el autor
* Elaboración y diseño de la tarjeta electrónica por el Prof. Carlos Ochoa
* Edición: Realizada con respecto a la base del proyecto
* Autor: Jonathan Olier
* Autor: Isabel de la Cruz

*Este trabajo se encuentra bajo una licencia* ***Creative Commons***

***[](../../../../Pictures/licencia%20para%20usiba/88x31.png)***

#### CONTACTO

 [djom202@gmail.com](mailto:djom202@gmail.com)

http://www.altiviaot.probandoando.com /home/correo/



#### REDES SOCIALES

 <http://www.facebook.com/>djom20

 <http://www.twitter.com/djom>20

# INDICE

[*SOBRE EL MANUAL* 1](#_Toc326900976)

[INDICE 2](#_Toc326900977)

[*1.* *INTRODUCCIÓN* 4](#_Toc326900978)

[*2.* *REQUERIMIENTOS* 5](#_Toc326900979)

[2.1. Servidor de bases de datos 5](#_Toc326900980)

[2.1.1. Crear base de datos 5](#_Toc326900981)

[2.1.2. Crear tablas 6](#_Toc326900982)

[2.2. DLL’s 7](#_Toc326900983)

[2.3. Visual Basic.Net y Framework 4 7](#_Toc326900984)

[*3.* *DIAGRAMA DE DISEÑO* 8](#_Toc326900985)

[3.1. Diagrama de componentes 8](#_Toc326900986)

[*4.* *ELABORACIÓN DE LA APLICACIÓN* 9](#_Toc326900987)

[4.1. Módulos de la aplicación 11](#_Toc326900988)

[4.1.1. Pantalla de Presentación 11](#_Toc326900989)

[4.1.2. Interface 12](#_Toc326900990)

[4.1.3. Configuraciones 15](#_Toc326900991)

[4.1.4. Iteraciones 17](#_Toc326900992)

[4.1.5. Acerca de USIBA 17](#_Toc326900993)

[4.2. Módulos de AccesoDatos.dll 18](#_Toc326900994)

[4.2.1. Modulo de Conexión ODBC 19](#_Toc326900995)

[4.2.2. Modulo de Archivo 19](#_Toc326900996)

[4.2.3. Modulo de HIDDLLInterface 19](#_Toc326900997)

[4.2.4. Modulo USB 20](#_Toc326900998)

[*5.* *DIAGRAMAS ELÉCTRICOS* 21](#_Toc326900999)

[5.1. Materiales 21](#_Toc326901000)

[5.2. Circuitos eléctricos 23](#_Toc326901001)

[5.1.0. Circuito de alimentación 24](#_Toc326901002)

[5.1.1. Circuito de entrada (Pulsador) 24](#_Toc326901003)

[5.1.2. Circuito de alimentación del micro-controlador 18F4550 25](#_Toc326901004)

[5.1.3. Circuito reset del micro-controlador 18F4550 25](#_Toc326901005)

[5.1.4. Circuito resonante 26](#_Toc326901006)

[5.1.5. Circuito USB interno 26](#_Toc326901007)

[5.1.6. Circuito de visualización de puerto B 27](#_Toc326901008)

[5.1.7. Circuito de adquisición de datos 28](#_Toc326901009)

[5.1.8. Circuito Completo 29](#_Toc326901010)

[5.3. Imágenes de la Tarjeta electrónica construida 29](#_Toc326901011)

[5.4. Firmware 30](#_Toc326901012)

[5.5. Construcción 33](#_Toc326901013)

[*6.* *FUNCIONAMIENTO* 35](#_Toc326901014)

[6.1. Instalación de la aplicación USIBA 35](#_Toc326901015)

[6.2. Instalación del MySQL ODBC 36](#_Toc326901016)

[6.3. Configuración del Servidor de Bases de Datos 38](#_Toc326901017)

[Nota: 39](#_Toc326901018)

[6.4. Integración del sistema de cómputo 41](#_Toc326901019)

[Esquema de sistema USIBA 42](#_Toc326901020)

# *INTRODUCCIÓN*

En este manual se presenta la elaboración de un sistema de cómputo llamado **USIBA (USb Interface for Biomedical Application)**, como ayuda a la recolección de datos en la Biomedicina, este sistema consta de 4 etapas, la elaboración de la aplicación que interactúa con el micro controlador, la integración del micro controlador con una tarjeta electrónica, la aplicación para su uso y la base de datos para el almacenamiento de la información recolectada para su posterior análisis o reproducción, además se expone y se detalla la comunicación entre una aplicación de escritorio con el dispositivo Biomédico por medio de protocolo USB con la finalidad de poder almacenar los datos y facilitar su manejo, control y disponibilidad ante cualquier circunstancia. Esta investigación se desarrollo en la Universidad Simón Bolívar de Barranquilla-Colombia con el fin de poder fomentar a otros investigadores u estudiantes a fortalecer el ámbito de la Biomedicina y el protocolo USB, además de dar bases para futuras investigaciones.

Describiremos claramente como se desarrollo y construyó el sistema, daremos pautas sobre como manejará el sistema de información y la configuración del sistema en general.

Como primer paso en cualquier instalación y ejecución de alguna aplicación veremos los requisitos necesarios en un equipo de cómputo para poder utilizar esta aplicación. Luego veremos los diagramas tanto de la aplicación como de la tarjeta electrónica que es utilizada para la recepción de los datos para así poder detallar su construcción.

También daremos algunos detalles para tener en cuenta a la hora de su funcionamiento para que trabaje de forma óptima, sin presentar ningún inconveniente a la hora de su ejecución, y al final podrán ver algunas respuestas claras a posibles problemas que podrían ocurrir de forma inesperada.

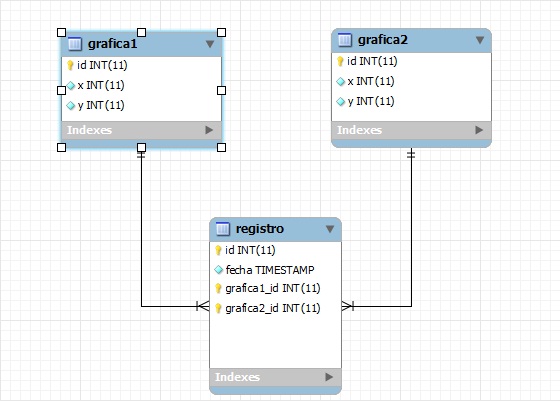
# *REQUERIMIENTOS*

## Servidor de bases de datos

El servidor de bases de datos utilizado es **MySQL en su versión 5.3** por su portabilidad, su fácil manejo, adquisición y de licencia gratuita, es uno de los más utilizados ya que es muy robusto ante las peticiones y consultas además de su gran estabilidad. Con el servidor de MySQL y la interfaz **Phpmyadmin** en **su versión 3.4.5** diseñada para el manejo grafico del servidor MySQL, fue diseñada la estructura de las tablas, sus relaciones y las consultas optimizadas para la inserción de gran cantidad de datos en un tiempo muy corto, utilizadas en **USIBA**, además realiza una conexión tipo MySQL ODBC para la fácil utilización de diversas Bases de Datos y la fácil comunicación de la aplicación con la base de datos.

### Crear base de datos

La base de datos consta solamente de tres tablas en las cuales se almacenan todos los datos recibos en microsegundos, previamente enviados por cualquier dispositivo biomédico y que posteriormente serán graficados por **USIBA**.



### Crear tablas

**SQL (Standard Query Language)** es el lenguaje estándar utilizado para las consultas, por todos los servidores de bases de datos actuales, por lo tanto la creación de las tablas se exporto a fin de obtener el SQL para su próxima reutilización sin necesidad de crearla desde cero.

Código SQL:

|  |  |
| --- | --- |
| -- phpMyAdmin SQL Dump  -- version 3.4.5  -- http://www.phpmyadmin.net  -- Servidor: localhost  -- Tiempo de generación: 25-05-2012 a las 12:05:34  -- Versión del servidor: 5.5.16  -- Versión de PHP: 5.3.8  SET SQL\_MODE="NO\_AUTO\_VALUE\_ON\_ZERO";  SET time\_zone = "+00:00";  /\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;  /\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;  /\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;  /\*!40101 SET NAMES utf8 \*/;  --  -- Base de datos: `investigacion`  --  -- --------------------------------------------------------  --  -- Estructura de tabla para la tabla `grafica1`  --  CREATE TABLE IF NOT EXISTS `grafica1` (  `id` int(11) NOT NULL AUTO\_INCREMENT,  `x` int(11) NOT NULL,  `y` int(11) NOT NULL DEFAULT '0',  PRIMARY KEY (`id`)  ) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO\_INCREMENT=1 ; | -- --------------------------------------------------------  --  -- Estructura de tabla para la tabla `grafica2`  --  CREATE TABLE IF NOT EXISTS `grafica2` (  `id` int(11) NOT NULL AUTO\_INCREMENT,  `x` int(11) NOT NULL,  `y` int(11) NOT NULL DEFAULT '0',  PRIMARY KEY (`id`)  ) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO\_INCREMENT=1 ;  -- --------------------------------------------------------  --  -- Estructura de tabla para la tabla `registro`  --  CREATE TABLE IF NOT EXISTS `registro` (  `id` int(11) NOT NULL AUTO\_INCREMENT,  `id\_grafica1` int(11) NOT NULL,  `id\_grafica2` int(11) NOT NULL,  `fecha` varchar(50) NOT NULL,  PRIMARY KEY (`id`)  ) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO\_INCREMENT=1 ;  /\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;  /\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;  /\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/; |

## DLL’s

La aplicación **USIBA** funciona a base de tres .dll muy sencillas, una es **MySql.Data.dll** que nos permite la utilización de un puente por ODBC para realizar una conexión a la base de datos, **mcHID.dll** que nos brinda los módulos y configuraciones necesarias para el puente entre los **dispositivos USB** por medio de **HID (Human Interface Device)** y la otra es **AccesoDatos.dll** que contiene los módulos de la aplicación en conexión USB, de conexión con la base de datos, manejo de archivos y directorios.

## Visual Basic.Net y Framework 4

**USIBA** fue construido en **Visual Basic.Net** por su amplia facilidad de aprendizaje en un corto tiempo, sus interfaces muy didácticas, la variedad de componentes que proporciona, además de su robusto campo de funciones y herramientas para el diseño de software.

En su última versión, **Visual Basic.Net 2010** también nos presenta el **Framework versión 4**, que no solo trae consigo nuevos componentes y funciones sino también mejoras de los anteriores componentes, de los cuales reutilizamos algunos para el diseño de esta aplicación.

Por lo tanto es importante la instalación de este framework en el equipo de cómputo donde se instalara **USIBA**, para el caso de los diseñadores o programadores si así lo desean podrán instalar el **IDE** del Visual Basic.net para ver su código fuente, ya que es totalmente, **código abierto**.

El Framework 4 se puede descargar desde su página oficial [***aquí***](http://www.microsoft.com/en-us/download/details.aspx?id=17851).

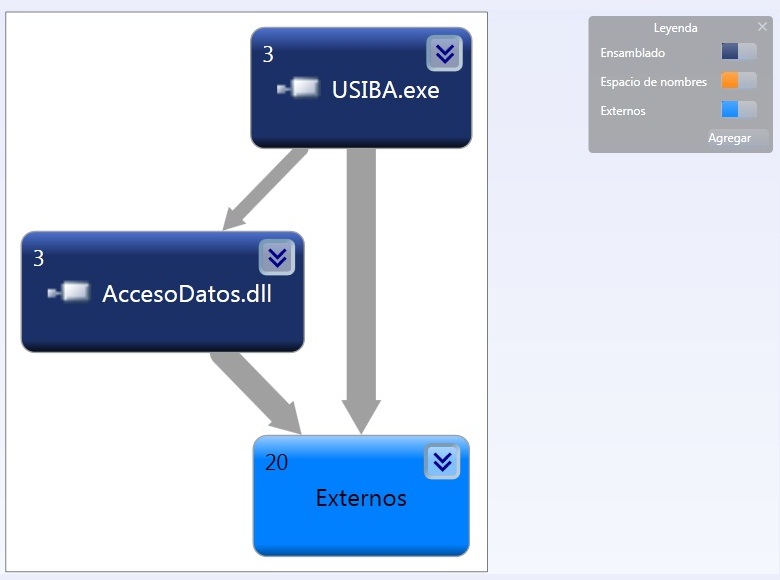
El Visual Basic.net 2010 se puede descargar desde su página oficial [***aquí***](http://www.microsoft.com/en-us/download/details.aspx?id=12752).

# *DIAGRAMA DE DISEÑO*

## Diagrama de componentes

En este diagrama podemos ver que la aplicación consta de tres cosas:

* La aplicación ó ejecutable de **USIBA**
* La .dll llamada **AccesoDatos.dll**
* Unas librerías **externa** que son la .dll **MySql.Data.dll** y **mcHID.dll**

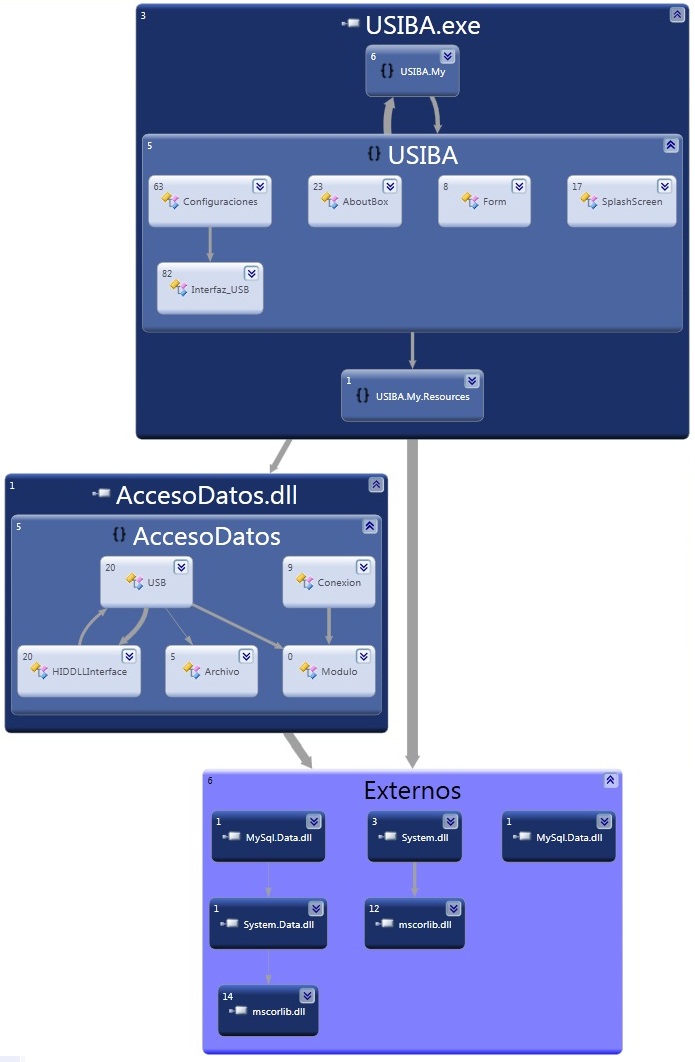


# *ELABORACIÓN DE LA APLICACIÓN*

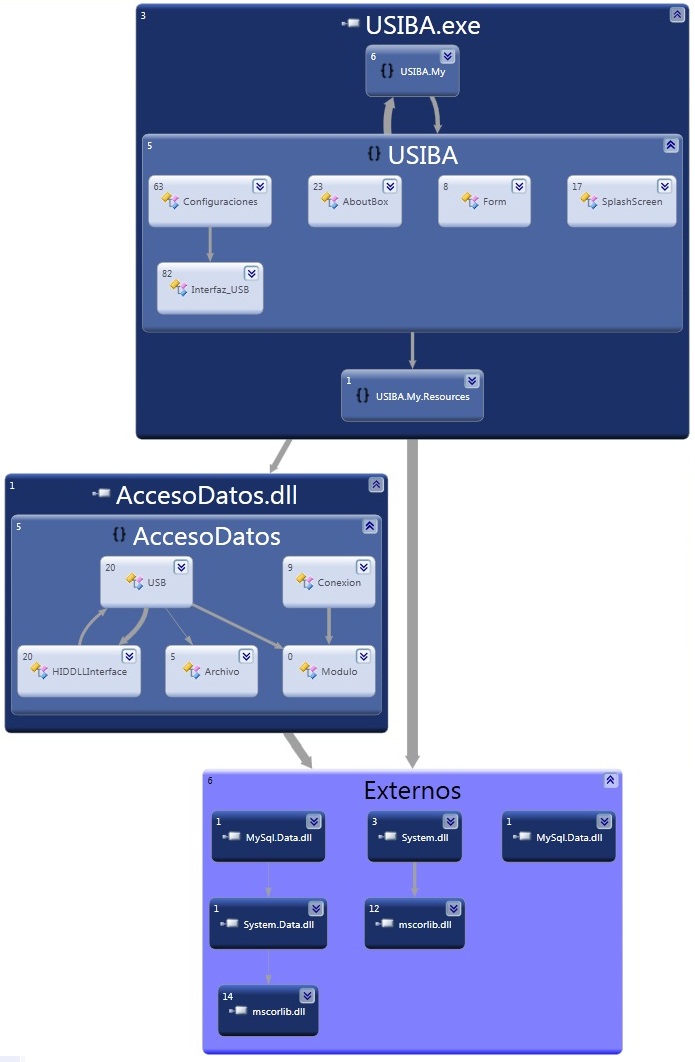
Desarrollo de una interfaz gráfica (**GUI**) con el lenguaje de alto nivel orientado a objetos, **Visual Basic.Net** y el **Framework 4** que permite la interacción con el micro controlador **18F4550** por medio del protocolo **USB (Universal Serial Bus)** junto con el protocolo de comunicación **HID (Human Interface Device)** que comúnmente se refiere a la especificación **USB-HID**, además la realización de la conversión en las señales enviadas por el dispositivo biomédico a través del micro controlador a un tipo de datos leíbles y manejable para cualquier usuario en el equipo de cómputo para su posterior uso y análisis en el campo de la biomédica. Detallaremos cada uno de los módulos construidos además de realizar una breve descripción de como se usaron los componentes externos como **mcHID.dll** y **MySql.Data.dll**.

Para las gráficas estadísticas e información relacionada con los datos recibidos se utilizo el componente o control suministrado por el ambiente de desarrollo de Visual Studio 2010 llamado **Chart** que fue añadido en el **Framework versión 3** pero ahora el la versión 4 del Framework presenta muchas mejoras substanciales como la variedad de gráficos que se pueden realizar hasta la cantidad de graficas en un solo objeto chart.

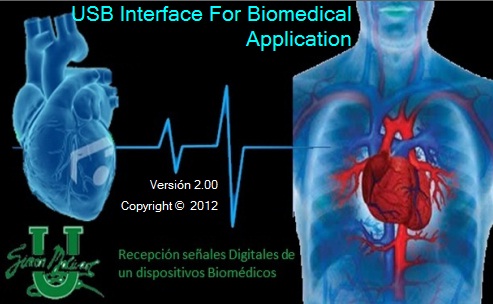
En la comunicación con el protocolo **HID**, existe 2 entidades: el host y el dispositivo. El **dispositivo** es la entidad que interactúa directamente con el humano, como lo hace un teclado o un ratón y en nuestro caso es nuestra tarjeta electrónica con el micro controlador 18F4550. El **host** se comunica con el dispositivo y recibe datos de entrada del dispositivo en las acciones ejecutadas por el humano, para nosotros el host es **USIBA** que recibe los datos de la tarjeta electrónica que a su vez recibe las señales de cualquier dispositivo biomédico. Los datos de salida van del host al dispositivo y luego al humano. El dispositivo como todo posee una serie de identificaciones únicas que son **ID del vendedor, ID del producto**, al momento de la conexión solo se realizara la conexión con aquel dispositivo que contenga estas identificaciones.



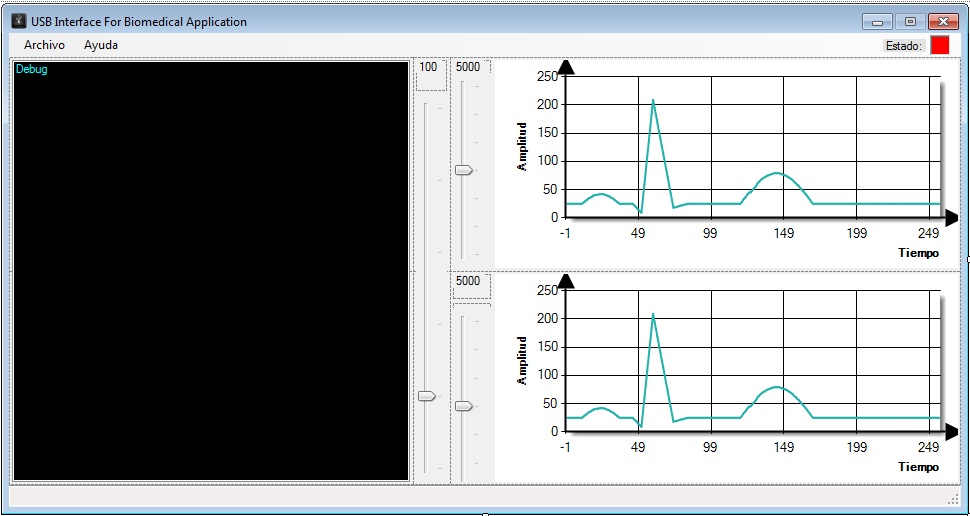
## Módulos de la aplicación



### 4.1.1. Pantalla de Presentación

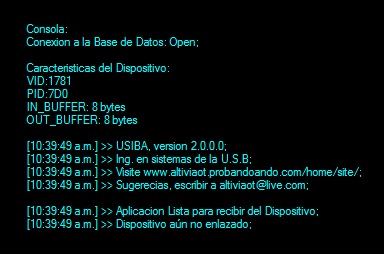


### 4.1.2. Interface

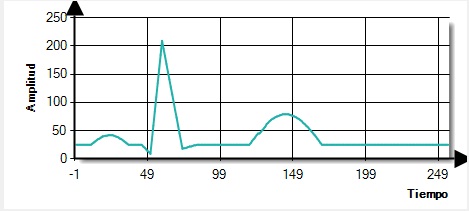


La interfaz consta de 6 partes muy sencillas:

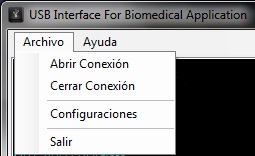
* Empezando por una consola donde se visualizara la serie de instrucciones que se irán ejecutando, por ejemplo en la línea **2** se visualiza en el momento de la ejecución de la aplicación un estado a la conexión a la base de datos ya se **OPEN, CLOSED, BROKEN, CONNECTING, EXECUTING ó FETCHING**, seguido de las caractericas del dispositivo entre ellas **ID del vendedor, ID del producto** y los tamaños de los **BUFFER** utilizados en la aplicación. Después se visualizara las características de la aplicación acompañadas de la hora en que ejecutaron.

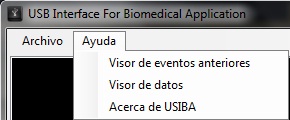


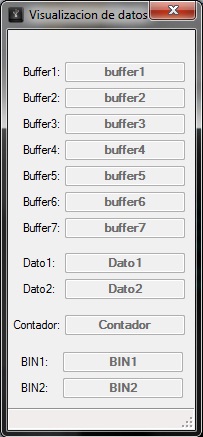
* Después podremos ver una barra deslizante que atraviesa la venta verticalmente que manipulara en cambio de la oscilación en ambos visores por medio de señales enviadas al micro-controlador 18F4550.
* Ahora veremos dos barras deslizantes, ambas al lado izquierdo de cada visor que manipulan la variación de la amplitud en cada grafica respectivamente.
* Seguido podremos ver los dos visores que visualizaran los datos enviados por el micro-controlador 18F4550 en un lapso de tiempo del cual pertenece al eje “x” y en el eje “y” representaremos la variación de la amplitud de los datos.

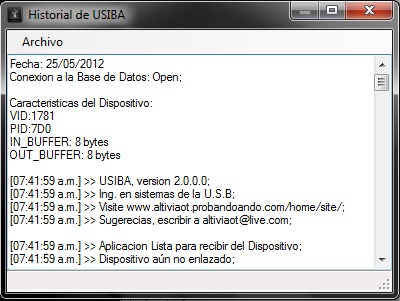


* En la esquina superior derecha de la ventana podemos un recuadro rojo o verde que indica el estado de conexión con la tarjeta electrónica.
* Ahora podremos ver el menú archivo y ayuda, con la opciones de poder abrir la conexión de recepción de datos del dispositivo o cerrarla, la configuración y cerrar la aplicación, además la posibilidad de poder acceder al visor de eventos y el visor de datos.









Nota: en el Visor de eventos, se lograra ver el historial de todas las ocasiones pasadas en que utilizo USIBA, y en el Visor de datos se podrá ver cada uno de los datos recibidos por la tarjeta electrónica de forma numérica.

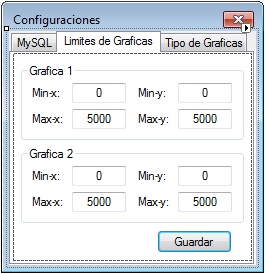
### 4.1.3. Configuraciones

En la opción de configuraciones podremos ver que se abre una ventana en la cual podremos ver dos pestañas, una que nos indica MySQL, lo cual es configuración sobre la conexión del servidor MySQL de bases de datos, entre los cuales vemos los campos habituales como lo son el nombre de la base de datos, la dirección del host de las base de datos, el usuario creado para la manipulación de los datos, el password del usuario para la conexión y el puerto usado por el servidor para la conexión.

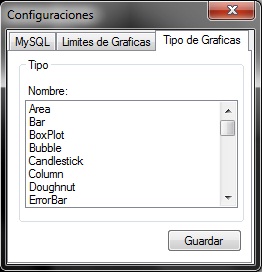


#### Nota: No utilizar el usuario root (Administrador de la Base de Datos) para este tipo de conexión.

En la siguiente pestaña podremos los valores de oscilación de cada visor que se utilizan para poder realizar una mejor vista ó zoom de la grafica, ya que la tarjeta electrónica maneja estos valores en milisegundos que es la frecuencia con que son enviados los datos por **teorema de muestreo de Nyquist-Shannon,** dice que la reconstrucción exacta de una señal periódica continua en banda base a partir de sus muestras, es matemáticamente posible si la señal está limitada en banda y la tasa de muestreo es superior al doble de su ancho de banda, y en nuestro caso como la señal **ECG (electro cardiograma)** tiene componentes en frecuencia de máximo 250Hz por lo que se debe muestrear mínimo a 500Hz que a su vez esta frecuencia es dada por el cristal de 4MHz que se encuentra en la tarjeta electrónica que permite cambiar el tiempo que por defecto trae el micro-controlador 18F4550 para la recepción y envió de datos al equipo de computo, sin embargo este valor puede variar entre 0 y 5000 en USIBA para realizar un zoom a las graficas de mayor rango.

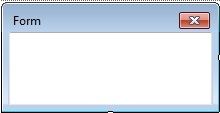


En esta pestaña podemos ver los diferentes tipos de graficas que son posibles de implementar en los visores, el tipo de grafica por defecto es **FASTLINE** pero esta opción esta habilitada para ver la información del paciente ya que los diferentes tipos de graficas proveen diferentes tipos de información de los pacientes.



### 4.1.4. Iteraciones

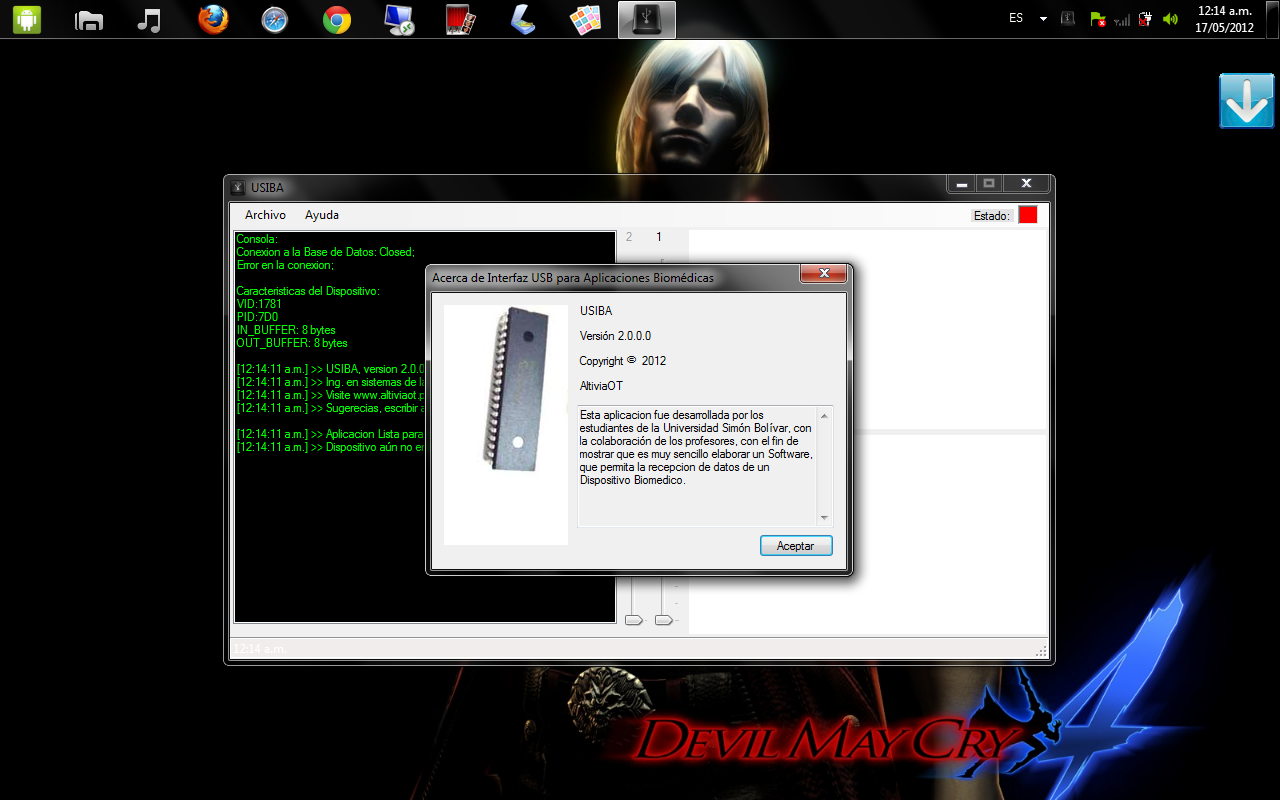
Este modulo es el mas complicado e importante de la aplicación, ya que por mas simple que se vea esta ventana, es aquella que permite la **conexión USB** de la aplicación con la tarjeta electrónica. Este modulo realiza una intervención en el **proceso de comunicación USB** de la aplicación realizado en el modulo **HIDDLLInterface** de la .dll llamada **AccedoDatos.dll** ya que sirve como hilo para sustraer de los procesos del equipo de computo el dispositivo que es conectado por **HID**, además que se le provee el **ID proveedor** y el **ID vendedor** internamente en su código fuente, ya que miles de dispositivos pueden ser conectados por HID que es una tecnología ya muy usual entre una gran variedad de dispositivos USB, si esto no se diera, no se realizaría la conexión.



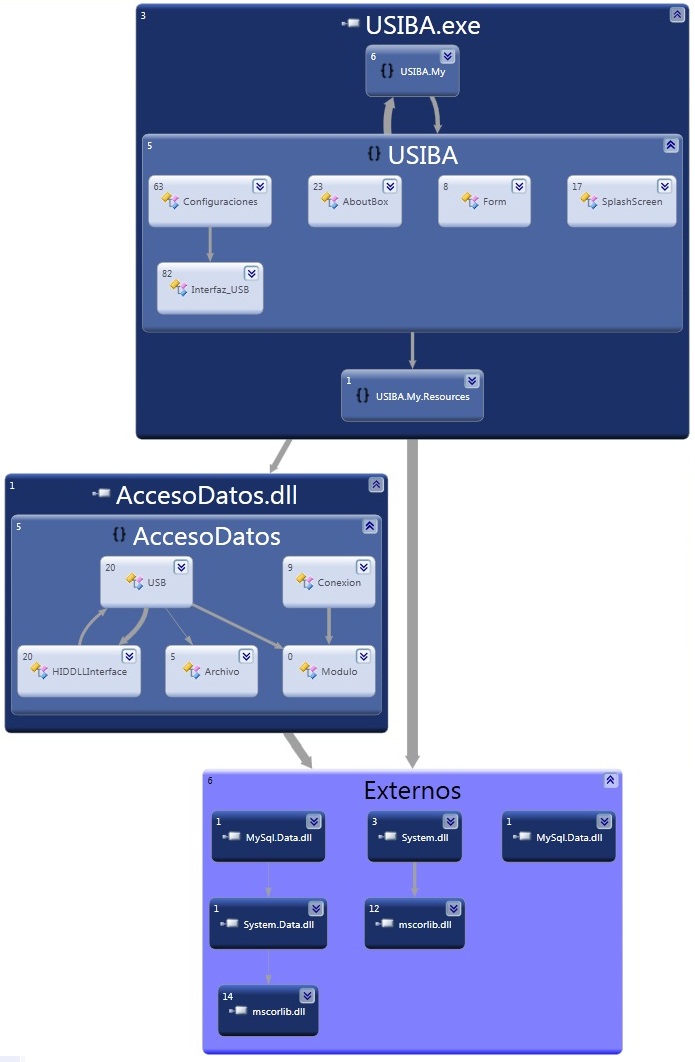
Solo basta con ser llamada esta venta para que se realice el proceso de comunicación USB, ya que el modulo **HIDDLLInterface** reconoce que la venta es lanzada y la oculta ante la vista para realizar un hilo ó un subproceso propio de la aplicación.

### 4.1.5. Acerca de USIBA

Vimos que el área de la biomedicina necesitaba una ardua interacción de los equipos de cómputo para mejorar los procesos de interacción con los equipos biomédicos ya que las herramientas en la actualidad son de costos considerables además que son diseños que no son propios y/o sin licencias para implementar, **USIBA (USb Interface for Biomedical Application)** en su versión 2.0, nació como un proyecto de la catedra de Formativa I y II de la Universidad Simón Bolívar a cargo del Profesor e ingeniero electrónico Calos Ochoa, por consiguiente se diseño **USIBA** para realizar una aplicación que fuera capaz de poder recopilar y almacenar la información de forma ágil y portable de cualquier tipo de dispositivo Biomédico que emitieran señales **análogas** y las convirtiera a **digitales** para su posterior análisis y/o reproducción, además la fundamentación de esta será para dar base a todos los posibles proyectos que apenas están surgiendo o que necesiten fundamentos y/o colaboración científica sobre las aplicaciones que por medio del protocolo USB reciba señales de dispositivos Biomédicos ya sea **ECG (Electro Cardiograma), EEG (Electro Encefalograma) ó EMG**, entre otros.



## 4.2. Módulos de AccesoDatos.dll



### 4.2.1. Modulo de Conexión ODBC

Este modulo provee las funciones y/o procedimientos necesarios para la conexión y desconexión a la bases de datos, además la ejecución de las consultas SQL, el estado de la conexión y de permitir las configuraciones básicas para dicha conexión.

### 4.2.2. Modulo de Archivo

El modulo de Archivo, en pocas palabras es el encargado de poder crear archivos planos con el contenido de información que la aplicación necesite que el usuario tenga almacenado de forma externa al software como tal, en este caso se manejan dos que son el archivo \*.log que viene con el nombre de la fecha del día y la hora en que se ejecuto USIBA y que procesos se llevaron a cabo, además están los archivos \*.backup que contienen todos los datos que no pudieron ser guardados en la base de datos para su posterior inserción de forma manual, de esta forma la inserción posterior no tendrá ninguna repercusión ya que en la base de datos se mantiene una configuración por fecha y hora dándonos así un control de datos a pesar de que estos fueron guardados después.

### 4.2.3. Modulo de HIDDLLInterface

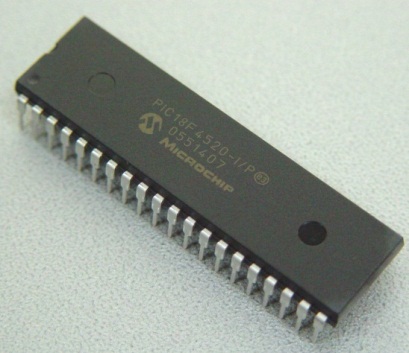
Este modulo no fue diseñado, fue reutilizado de un software llamado **EasyHID Wizard** que permite generar la librería (**mcHID.dll**) y la estructura básica para hacer la conexión de forma muy dinámica y sencilla para este tipo de interconexión USB sin embargo esta desarrollada en visual 6.0, por lo que si se procedió ha convertir el código en visual basic.net para nuestros fines de forma que trabajara de la misma forma, sin afectar su funcionamiento normal.

Este modulo contiene todas las interrupciones necesarias para detectar la conexión y desconexión física y/o lógica de un dispositivo USB por HID. Esta modulo realiza una conexión con la .dll mcHID.dll, que contiene los procedimientos para dicha conexión. El modulo toma los procesos del sistema operativo, hace una interrupción a dichos procesos y procede a verificar la conexión y desconexión física del dispositivo que anteriormente por código se le ha indicado que debe buscar o en el caso del proceso de Windows, estará pendiente y en el momento que ocurra avisara a la aplicación que ya se realizo la conexión o desconexión del dispositivo en cuestión.

### 4.2.4. Modulo USB

El modulo **USB** fue diseñado para contener los procedimientos resultantes de la interrupción de los procesos de Windows con tal de que la información recaudada por dichos procesos sobre el dispositivo conectado además de los datos recaudados llegaran a este. También es por donde se enviaran los datos a la interfaz grafica en parámetros para su posterior visualización.

# *DIAGRAMAS ELÉCTRICOS*



En esta sección del manual veremos los diagramas para la construcción y funcionamiento de la tarjeta electrónica que recibe los señales de un dispositivo biomédico (**ECG, EEG ó EMG**) y los envía al equipo de cómputo por medio de un **micro-controlador 18F4550**.

Nota: La realización o desarrollo del circuito se hizo en un entorno simulado que fue Proteus en su versión 7.

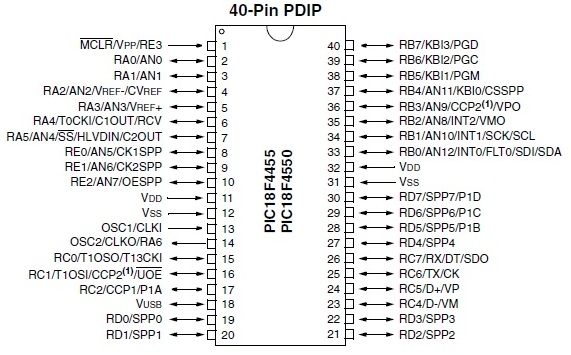
## Materiales

En la siguiente lista de los materiales presentamos una imagen de cada elemento, la cantidad usada para la construcción de la tarjeta electrónica, el precio en pesos colombianos por unidad y por cantidad.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | |  |  |  |  |
| Ítem | **Imagen** | | **Nombre** | **Cantidad** | **Precio c/u** | **Precio** | |
| 1 |  | | ADAPTADOR DE BATERIA DE 5v | 1 | $ 400 | $ 400 |
| 2 |  | | BASE PIC | 1 | $ 500 | $ 500 |
| 3 |  | | BATERIA DE 9v | 1 | $ 5.000 | $ 5.000 |
| 4 |  | | BOTONES | 3 | $ 250 | $ 750 |
| 5 |  | | CABLE TIPO BUS | 1/2' | $ 500 | $ 500 |
| 6 |  | | CABLE USB DE 1 METRO | 1 | $ 3.500 | $ 3.500 |
| 7 |  | | CABLE UTP | 1/4' | $ 750 | $ 750 |
| 8 |  | | CAP 10uF | 1 | $ 250 | $ 250 |
| 9 |  | | CAP 27pF | 2 | $ 200 | $ 400 |
| 10 |  | | CAP 470uF | 2 | $ 300 | $ 600 |
| 11 |  | | CONN H2 | 2 | $ 250 | $ 500 |
| 12 |  | | CONN H3 | 2 | $ 200 | $ 400 |
| 13 |  | | CONN H4 | 1 | $ 250 | $ 250 |
| 14 |  | | CRISTAL 4MHz | 1 | $ 300 | $ 300 |
| 15 |  | | DE7805 | 1 | $ 300 | $ 300 |
| 16 |  | | LED | 10 | $ 50 | $ 500 |
| 17 |  | | micro-controlador 18F4550 | 1 | $ 80.000 | $ 80.000 |
| 18 |  | | PLACA BASE | 1 | $ 3.500 | $ 3.500 |
| 19 |  | | RES 10K | 14 | $ 33 | $ 466 |
| 20 |  | | SOLDADURA | 1 | $ 200 | $ 200 |
|  | | Total: | | | | **$ 99.066** |

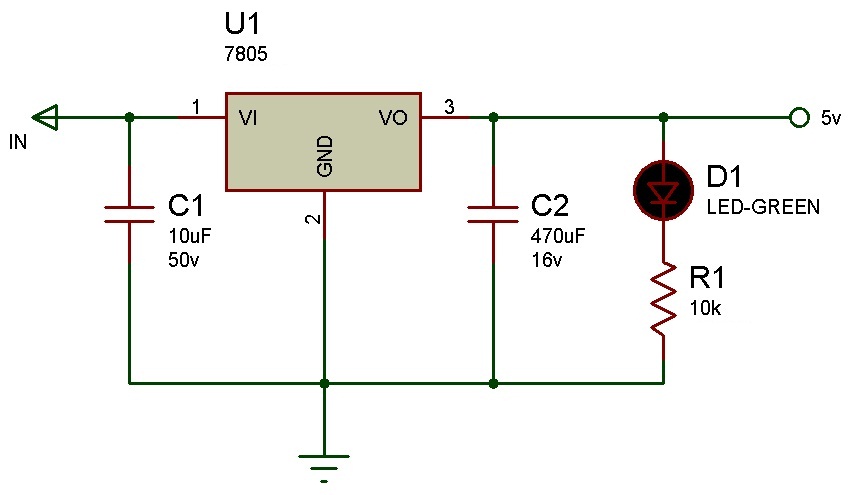
## Circuitos eléctricos

Este es el esquema que nos brinda un amplia visión de los pines que contiene el micro-controlador además de mostrar las configuraciones posibles con el y entre ellas podemos ver donde esta situado el **modulo VUSB** que es la esencia de este proyecto por parte del micro-controlador, ya que este será el que nos permita la comunicación por medio del protocolo USB.



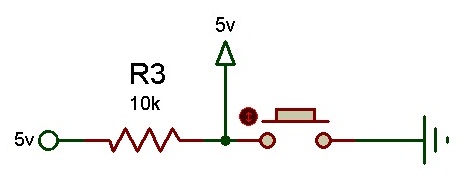
### Circuito de alimentación

En este circuito podemos observar una entrada de 5v que es regulada por un regulador de voltaje 7805 y dos capacitores a cada lado y luego encendido un Led para verificación de contacto ó de corriente existente.



### Circuito de entrada (Pulsador)

Este circuito permite generar unas señales electromagnéticas correspondientes a 1 y 0 lógicos para ser integrados en las entradas del micro-controlador.

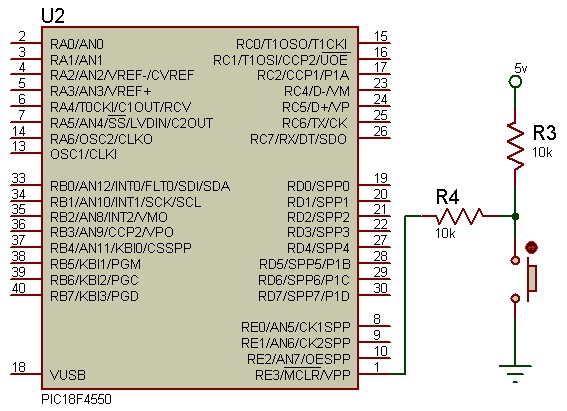


### Circuito de alimentación del micro-controlador 18F4550

En circuito de alimentación del micro-controlador no es más que llevar los 5v a las entradas VDD de cada lado del micro-controlador 18F4550 y tierra a cada entrada VSS de igual manera.

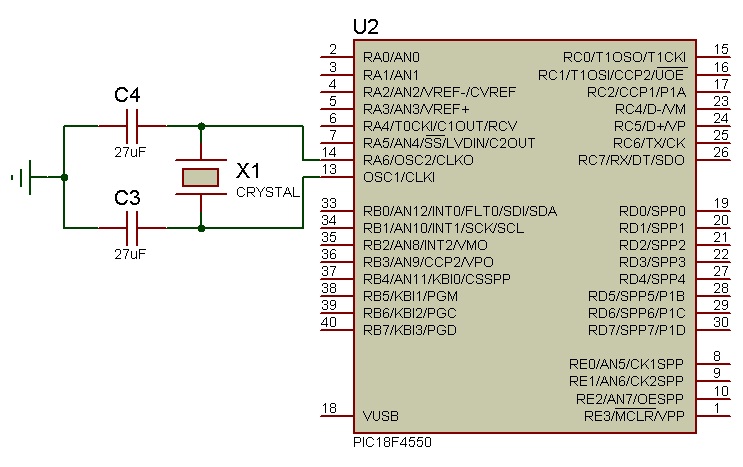
### Circuito reset del micro-controlador 18F4550

En este circuito podemos ver una entrada de 5v que es resistida por dos resistencias que lleva la energía al pin MCLR del micro-controlador que seria su funcionamiento normal, en caso de que el botón sea presionado la energía va directamente a tierra poniendo en 0 la energía, haciendo que el micro-controlador entre en estado de reset y empiece su programación inicial.



### Circuito resonante

En este circuito podemos veremos cómo crear el reloj que permite que el micro-controlador tenga una secuencia en sus procesos que es dado por el cristal y a su vez es alimentado por el micro-controlador y protegida por los capacitadores que re envían la energía a tierra.

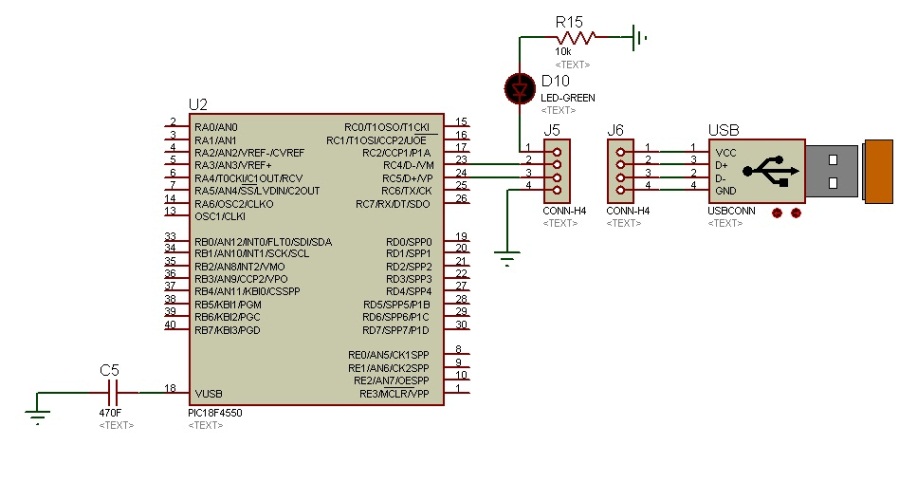


### Circuito USB interno

Aquí podremos ver como activar el modulo interno USB del micro-controlador, creando una regulación de energía sobre el puerto del micro-controlador VUSB, le estamos indicando de esta forma que se active para posterior uso.

Además podemos observar cómo se realizó la conexión del micro-controlador con el cable de re envió de datos que va al equipo de cómputo, vemos a través del puerto USB del cable recibimos cuatro elementos diferentes:

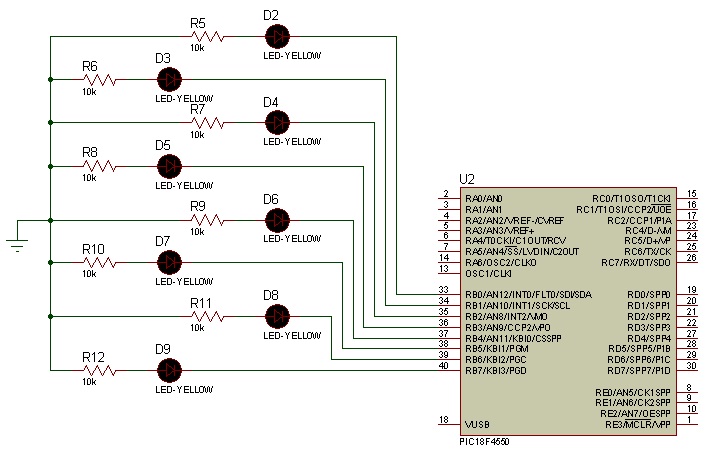
* Un fuente de energía de 5v
* Una fuente de datos +
* Una fuente de datos –
* Un fuente de neutra o tierra

******

Nota: Ya que utilizamos una fuente externa al propio circuito que es la batería de 5v, la energía proporcionada por el puerto USB es direccionada a un LED que a su vez es controlada por una resistencia y simplemente se realiza para la verificación de una buena conexión y/o un correcto funcionamiento del puerto USB y cable que es representado por dos conectores de 4, uno del lado del micro-controlador y otro del lado de cabezal USB del cable.

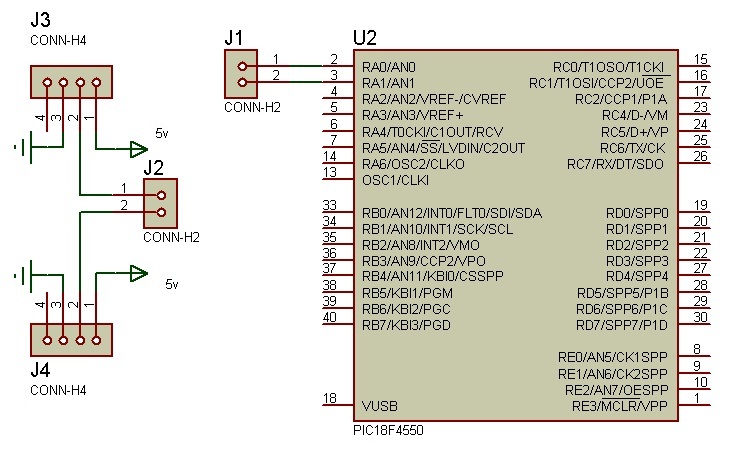
### Circuito de visualización de puerto B

El micro-controlador recibe una serie de datos de los dispositivos biomédico que a su vez son mostrados por una secuencia de Led, permitiendo así ver en que momento y donde se reciben los datos en el micro-controlador.

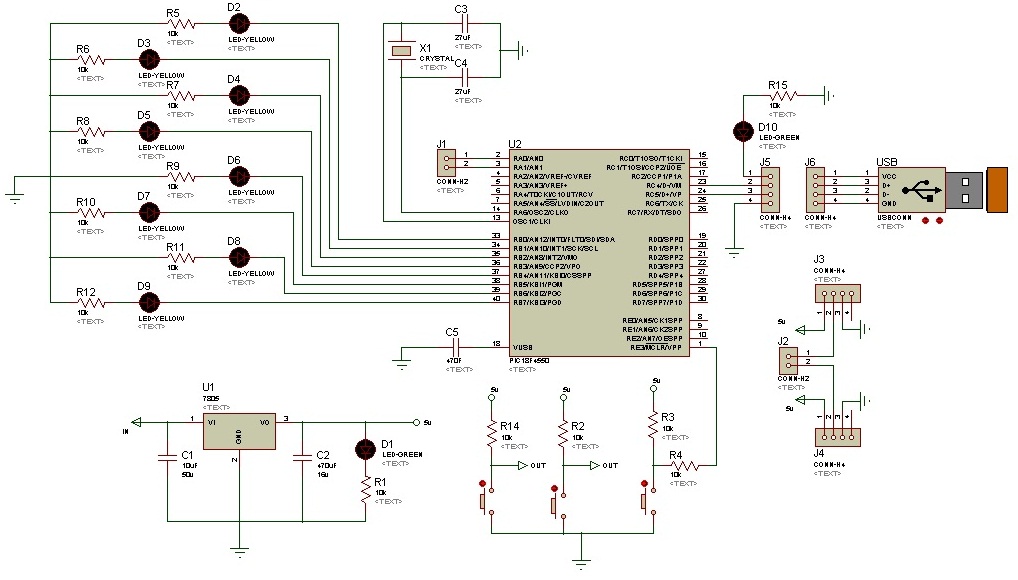


### Circuito de adquisición de datos

En este circuito vemos como por medio de dos conectores de 4 recibimos la señal análoga de un dispositivo biomédico, los conectores son alimentados con una fuente de energía de 5v y tierra para su funcionamiento, además vemos como enviamos los dos canales de la señal a un conector de 2 y recibido por otro conector de 2 para enviarlos al micro-controlador.



### Circuito Completo



## Imágenes de la Tarjeta electrónica construida

Algunas imágenes de la tarjeta construida antes de terminarla y junto a un prototipo construido para pruebas en el centro Biomédico de la universidad.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Firmware

Se desarrolló una aplicación o Firmware para el micro controlador 18F4550 en el lenguaje de programación de bajo nivel Basic, que se cargó en el micro controlador por medio del Proton Compiler para compilar el código en hexadecimal que se requiere para este proyecto, donde se dio las instrucciones básicas para el intercambio del flujo de información que maneja por medio del protocolo USB y así adquiriera los datos enviados por un dispositivo biomédico por medio del Protocolo USB y retransmitirlos a la aplicación de interfaz gráfica.

|  |
| --- |
| Código fuente:  DEFINE OSC 48  USBBufferSizeMax con 8 ' maximum buffer size  USBBufferSizeTX con 8 ' input  USBBufferSizeRX con 8 ' output  ' the USB buffer...  USBBuffer Var Byte[USBBufferSizeMax]  USBBufferCount Var Byte  i var byte ' Agregado para prueba envio datos del micro-controlador al pc  j var byte ' Agregado para prueba envio datos del micro-controlador al pc  TRISB=0 ' TODO EL PUERTO B COMO SALIDA (AGREGADO)  i=0  j=0  ' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ' \* main program loop - remember, you must keep the USB \*  ' \* connection alive with a call to USBService every couple \*  ' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ' 10-bit A/D conversion  ' Connect analog input to channel-0 (RA0)    ' Define ADCIN parameters  ' Set number of bits in result  'DEFINE ADC\_BITS 8  ' Set clock source (3=rc)  'DEFINE ADC\_CLOCK 1  ' Set sampling time in microseconds  'DEFINE ADC\_SAMPLEUS 10  'adcVar VAR WORD ' Create variable to store result  ' Set PORTA to all input  TRISA = %11111111  ' Set up ADCON1  ADCON1 = %00001101 ' AN1 Y AN0 como entradas analogas las demas digitales  ADCON0 = %00000000 ' Se selecciona como canal de entrada AN0  ADCON2 = %10101100 ' Se selecciona 12Tad y Fosc/4  ADCON0.0 = 1 ' Se habilita el modulo A/D  Pause 50 ' Wait .5 second  ' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ' \* main program loop - remember, you must keep the USB \*  ' \* connection alive with a call to USBService every couple \*  ' \* of milliseconds or so... \*  ' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  usbinit ' initialise USB...  ProgramStart:  ' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  'ADCIN 0, adcVar ' Read channel 0  ' do something with adVar here  ' Pause 10 ' Wait.01 second  'PORTB = adcVar    CONVER: ADCON0.1 = 1 ' Se inicia la conversion A/D  VERI:USBService    IF ADCON0.1 = 1 then veri  PORTB= ADRESL    USBService  USBService  USBService  USBService  USBService  USBService  ' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  'gosub DoUSBOut  'gosub DoUSBIn  'PORTB=usbbuffer[7]  'goto ProgramStart    iF PORTD.0=0 THEN  usbbuffer[7]= 00  PORTB.0=0  else  usbbuffer[7]= 01  PORTB.0=1  ENDIF    'usbbuffer[7]= 11 'Pc Recibe BufferIn (8) en el PC ok  Usbbuffer[6]= ADRESH 'PC Recibe BufferIn (7) en el PC ok  usbbuffer[5]= ADRESL 'PC Recibe BufferIn (6) en el PC ok  usbbuffer[4]= ADRESH 'PC Recibe BufferIn (5) en el PC ok  usbbuffer[3]= ADRESL 'PC Recibe BufferIn (4) en el PC ok  usbbuffer[2]= 16 'PC Recibe BufferIn (3) en el PC ok  usbbuffer[1]= 17 'PC Recibe BufferIn (2) en el PC ok    'usbbuffer[9]= 13 'Recibe BufferIn (10) en el PC  'PORTB=11  gosub DoUSBOut 'ojo habilitar    'for i=0 to 10  'usbbuffer[7]= i  'PORTB= i  'for j=0 to 100  'pause 5  'gosub DoUSBOut  'next j  'next i  'PAUSE 5    goto ProgramStart  ' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ' \* receive data from the USB bus \*  ' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  DoUSBIn:  USBBufferCount = USBBufferSizeRX ' RX buffer size  USBService ' keep connection alive  USBIn 1, USBBuffer, USBBufferCount, DoUSBIn ' read data, if available  return    ' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ' \* wait for USB interface to attach \*  ' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  DoUSBOut:  USBBufferCount = USBBufferSizeTX ' TX buffer size  USBService ' keep connection alive  USBOut 1, USBBuffer, USBBufferCount, DoUSBOut ' if bus available, transmit data  return |

## Construcción

Se construyó una tarjeta electrónica con el micro controlador micro-controlador 18F4550 de modo que se implanto la aplicación de bajo nivel en Basic en el mismo. Esta tarjeta electrónica sirve como medio de interacción entre la aplicación de interfaz gráfica (GUI) y el dispositivo Biomédico.

La construcción de realizo en unos cuantos pasos muy simples que describiremos a continuación:

* Construir la tarjeta electrónica, circuito por circuito que fue descrito en el capítulo 5.2 con los materiales que se describieron en el capítulo 5.1 hasta interconectar todos los circuitos completando el circuito descrito en el capítulo 5.1.9.
* Luego de tener la tarjeta electrónica completamente armada y el micro controlador preparado para su inserción en la tarjeta electrónica, se procede a quemar la aplicación en Basic en el micro controlador con el IC-Prog 1.05C que nos facilita cargar y preparar la aplicación para la quemada ya que es compatible con los diferentes puerto (Paralelo, Serie, USB) y además utilizaremos el grabador de micro-controlador **PICSTART PLUS** que su elaboración se encuentra en internet en pasos sencillos y a muy bajo costo.

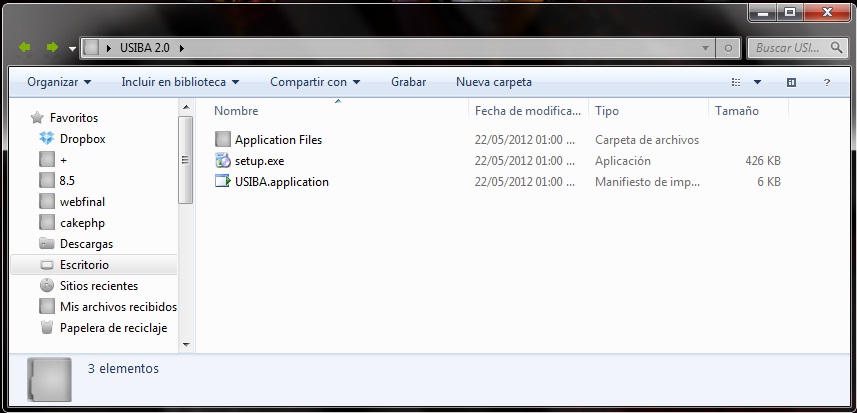
* Se conectara un cable USB al quemador y colocamos el micro-controlador en el zócalo del **PICSTART PLUS** correctamente, para eso no fijamos bien en la muesca que tiene el micro-controlador para no quemarlo ni dañarlo, luego en el IC-Prog previamente configurado damos la orden para quemarlo, incrustando en el micro-controlador la aplicación o Firmware.
* Ya contenida la aplicación en el micro controlador, insertamos el micro-controlador 18F45550 en el zócalo con mucho cuidado teniendo en cuenta la muesca del micro-controlador con la de zócalo y podremos conectar la tarjeta electrónica al puerto USB, verificando así que haya una buena conexión con el equipo de cómputo, además conectar el dispositivo biomédico a la tarjeta electrónica mientras observamos que los Led varían su encendido para demostrar un recepción de datos.

# *FUNCIONAMIENTO*

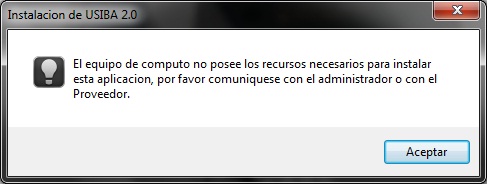
En este capitulo describiremos el funcionamiento del sistema de información, las precauciones que se deben tener en cuenta al momento de la utilización de este.

## Instalación de la aplicación USIBA

Lo primero que haremos será instalar la aplicación en el centro de cómputo, dentro de la carpeta de USIBA 2.0 se encuentran dos archivos y el ejecutable.



Procederemos a ejecutar el instalador que en este caso es el archivo setup.exe, luego el instalador verificará que el equipo de computo contenga todos los recursos en hardware necesarios para la instalación, en caso que haga falta recursos, mostrara una advertencia diciendo que el equipo no posee los recursos necesario.



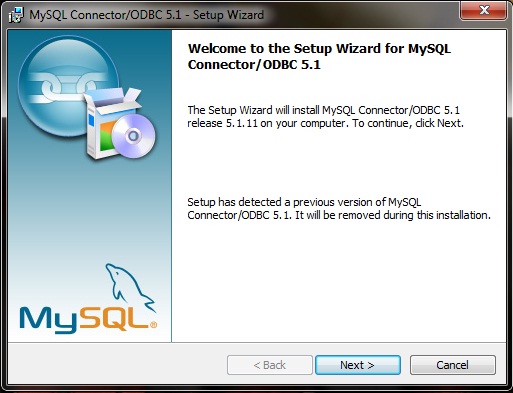
Si el equipo de cómputo satisface las necesidades entonces el instalador procederá a verificar que existan o estén instalados los requisitos previos para el buen funcionamiento de la aplicación como lo son:

* Microsoft .net framework 4
* Windows installer 4.5

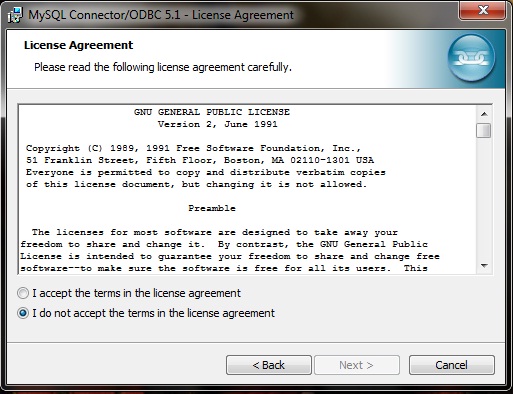
## Instalación del MySQL ODBC

Luego de esto procederemos a instalar el ODBC para la conexión con el servidor de bases de datos.

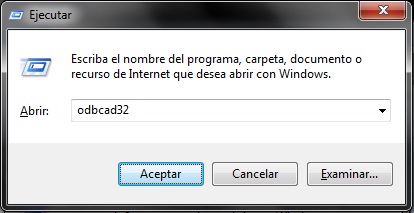
Se puede descargar desde su página oficial [aquí](http://dev.mysql.com/downloads/connector/odbc/).

****

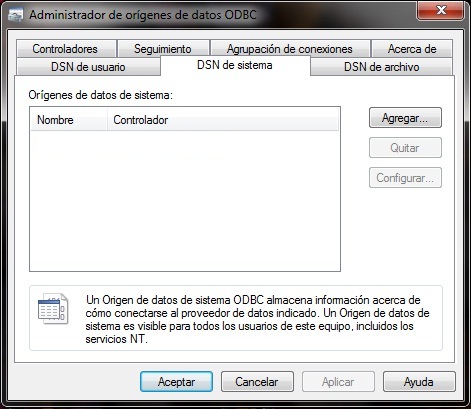
Se descargar el instalador **mysql-connector-odbc-5.x.xx-win32.msi**, luego procedemos ejecutarlo, y configurarlo.



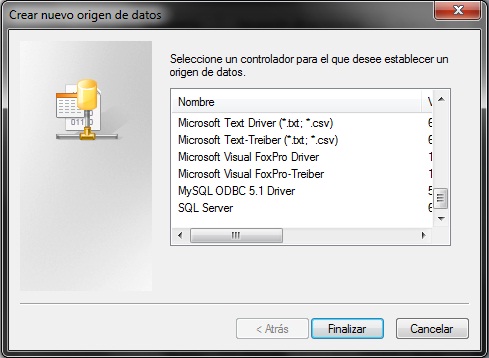
Luego aceptaremos los términos del contrato y procederemos con la instalación, elegiremos la instalación de tipo estándar y finalizaremos las configuraciones e instalación.



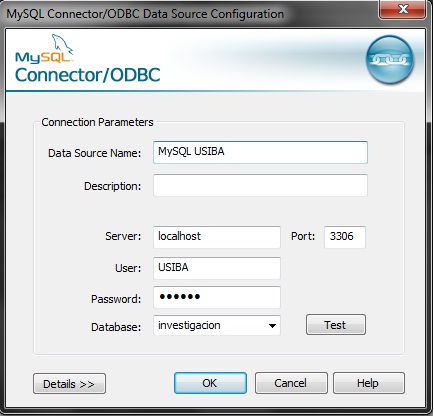
Después de esto procederemos a configurar el puente del ODBC de equipo de computo con la aplicación, para ello presionamos en el teclado las teclas Windows + r seguidas una de la otra para abrir la consola de ejecutar.

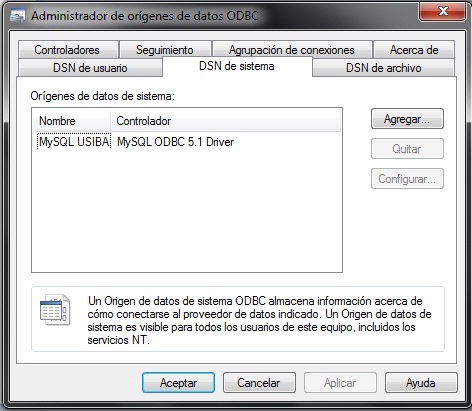


Ahora procedemos a escribir en el cuadro de dialogo **odbcad32**, el comando necesario para abrir el **administrador de orígenes de datos ODBC**. En este cuadro de dialogo iremos a la pestaña **DNS de sistema**, en la cual presionaremos el botón de agregar.



Ahora veremos otro cuadro de dialogo para **crear el origen de datos** en el cual nos pide seleccionar el tipo de conexión que deseamos crear, por lo cual si el paso anterior de instalar el **MySQL ODBC** se hizo correctamente, nos aparecerá en la lista **MySQL ODBC x.x Driver**, el cual se seleccionara y daremos finalizar.

Ahora en este cuadro de dialogo nos pedirá un nombre y una descripción para referenciar el puente, el nombre del server donde esta la bases de datos, colocaremos **localhost** por defecto o colocaremos la ip del host o su **dominio**, luego el usuario de la base de datos y su contraseña de ingreso. Presionamos el botón Test para ver si hay una buena configuración y si nos da satisfactoria podremos seleccionar el nombre de la base de datos que creamos anteriormente, **investigación**.



Ya con esto finalizamos en proceso de la creación del puente por ODBC con lo cual podremos ver listada nuestra conexión, con el nombre que le hemos dado y el tipo de conexión.

## Configuración del Servidor de Bases de Datos

En este paso vamos a ver como configurar y ejecutar nuestro servidor de Bases de datos. Nosotros por facilidad a la hora de la implementación utilizamos un servidor portable como lo es **USBwebserver** en su **versión 8**, este servidor nos provee toda una serie de servidores como el de MySQL y Apache, además de proveernos soporte para **PHP** y la herramienta **Phpmyadmin.**

El USBwebserver v8 se puede descargar desde su página oficial [aquí](http://www.usbwebserver.net/en/).

El **USBwebserver** es una aplicación portable nativa en esta versión. En este caso el servidor de MySQL ya lo tenemos corriendo como se ve en la imagen y de igual forma apache lo tenemos corriendo para poder acceder al Phpmyadmin.

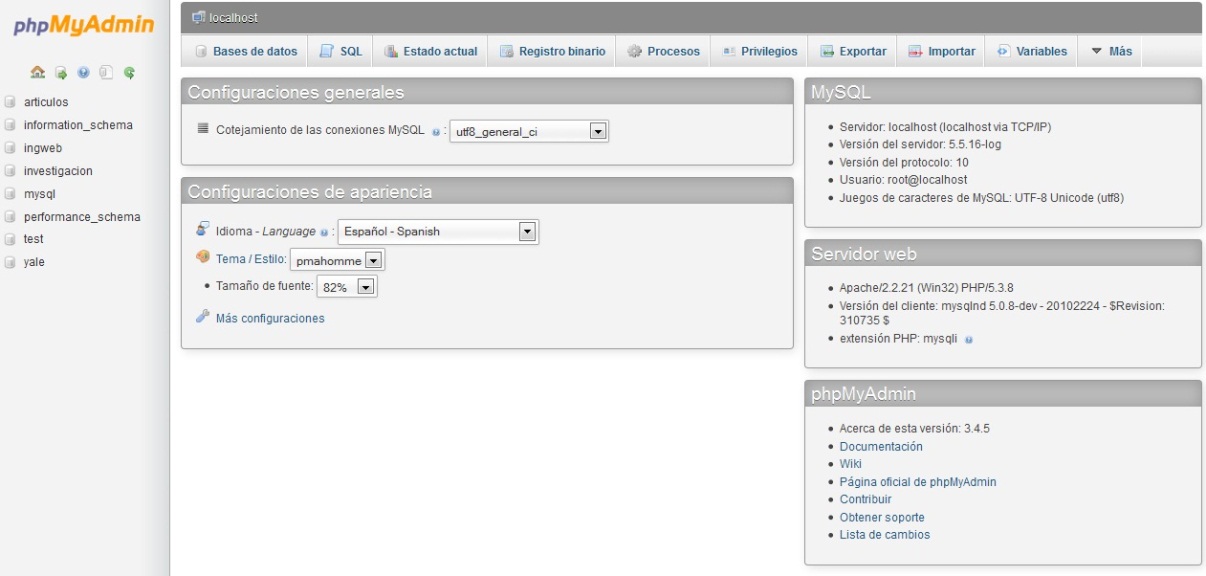


Ahora podremos ver en una de las pestañas del USBwebserver la opción de configuraciones donde podremos ver o modificar el puerto por el cual esta siendo ejecutado el servidor de MYSQL y Apache.

Nota: si el puerto de apache esta establecido en 80, quiere decir que se va a ejecutar en forma normal, pero si en el equipo de computo existe ya otra aplicación ejecutando otro servicio de apache, el USBWebserver no ejecutara el servicio de apache y bloqueara el primer servidor en ejecución, por tal razón tenemos definido el puerto en 8082 para evitar este conflicto. Con el servidor de MySQL no hay problema ya que pueden haber dos servidores corriendo aunque el mismo servidor hace solo funcionar el primero que es ejecutado.



El siguiente paso será abrir el Phpmyadmin, en la pestaña general existe un botón que dice **Phpmyadmin**, al presionarlo nos abrirá el navegador por defecto y nos mostrara la aplicación ya en ejecución.

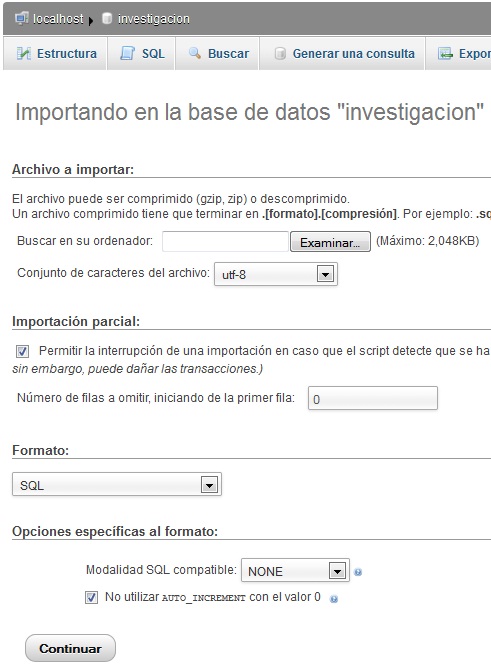


Esta es la interfaz del Phpmyadmin, varia según la versión y el estilo que venga en el USBwebserver.

Procederemos a crear la base de datos, en el link de Bases de Datos, nos pedirá el nombre y le daremos investigación, luego presionamos el botón de crear.



Ya habiendo creada la base de datos, procederemos a crear las tablas, de las cuales crearemos con el scripts que se encuentra en el capitulo 2.1.2, este scripts lo copiaremos y pegaremos en un blog de notas nombrándolo **investigacion.sql**.



Seleccionaremos la base de datos de **investigacion** y luego en la parte superior en el menú esta la opción de importación en la cual podremos ver la opción de examinar con la cual buscaremos el archivo **investigacion.sql** que acabamos de crear y presionaremos el botón de continuar para ejecutar las instrucciones del archivo para crear las tablas.



Ya podremos ver en la estructura las tablas creadas con éxito, y con esto finalizamos la creación de la base de datos.

## Integración del sistema de cómputo

El siguiente paso será ejecutar USIBA en el equipo de computo, además tomar la tarjeta electrónica y conectarla a la fuente de energía, al puerto USB y al dispositivo Biomédico, Con esto nos aseguramos que la tarjeta ya este recibiendo los primeros datos antes de poder poner en funcionamiento a USIBA, además USIBA podrá detectar el dispositivo conectado por USB por medio del protocolo HID y empezar las configuraciones básicas del inicio de transferencia de datos desde el dispositivo y a través de la tarjeta electrónica.

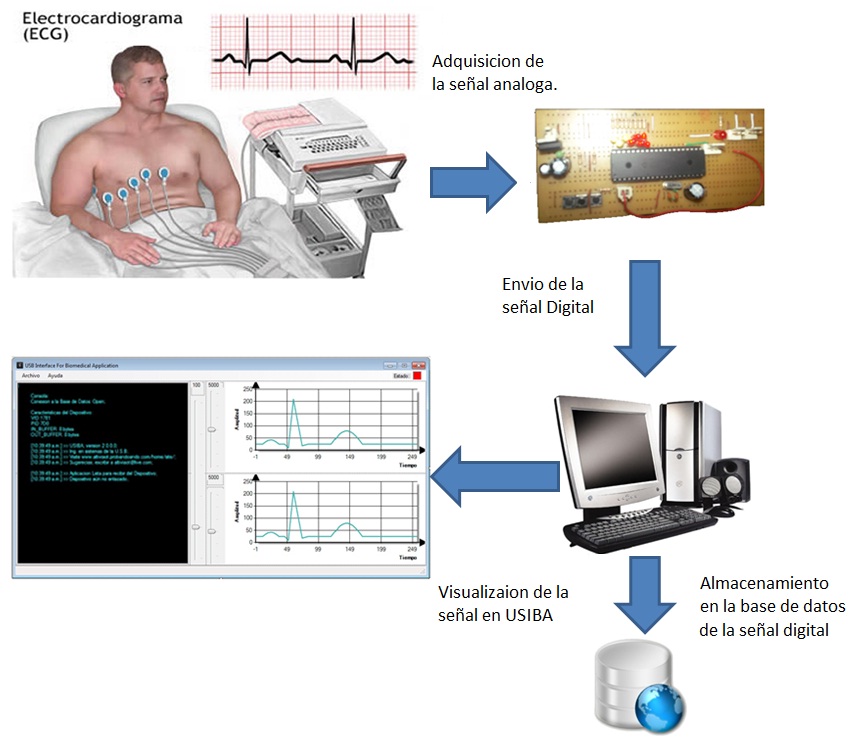
El micro controlador 18F4550 recibirá las señales biomédicas por medio de las entradas disponibles en el micro controlador de forma análoga usando los conversores digitales con una entrada I/O convencional. Los conversores tienen dos señales de entrada Vref+ y Vref- los cuales determinan el rango y a su vez este se convierte en la señal de entrada.

La etapa de adquisición de datos tiene un componente hardware, compuesto por el Conversor A/D mismo y un componente software, que es la rutina encargada de dirigir el funcionamiento del Conversor A/D.

La cantidad de muestras obtenidas es llevada a un archivo binario, el cual es rescatado por USIBA, para llevar a cabo el procesamiento de la señal y su posterior visualización.

Al haber obtenido los datos, USIBA procede a graficarlos. Este proceso es realizado, una vez que se han obtenido las muestras deseadas mediante USBIA, se toma como entrada el archivo producido por la aplicación en Assembler por medio del Conversor A/D. Para llevar a cabo la gráfica de los datos se utilizar un objeto Chart (un objeto estándar dentro de Visual Basic.Net).

### Esquema de sistema USIBA



Con esto damos por finalizada la correcta instalación del sistema de computo, para cualquier duda, comentario o sugerencia por favor comunicarse con el autor.