**Implementación a manera de ejemplo: Calculadora que guarda resultados en DB MySql**

[https://www.paypal.com/en_US/i/btn/btn_donate_LG.gif](http://vertigo.alojamientogratuito.info/wordpress/?p=317)Ayuda para que siempre encuentres lo que buscas!

**Conocimientos previos:**

* Saber crear una base de datos en MySql a nivel básico
  + Leer esto sin no sabes: <https://vertigo2040.wordpress.com/2011/01/01/creando-una-base-de-datos-en-mysql/>
  + Saber trabajar con Matlab a nivel básico
  + Saber trabajar con Xammp
  + Saber entrar a aplicación: Orígenes de datos ODBC

**Necesitamos :**

* Aplicación de conexión a MySql por medio de conector Connector/ODBC
* Pagina de descarga: <http://www.mysql.com/downloads/connector/>
* Tener activo la aplicación “Orígenes de datos ODBC”
* Entorno de trabajo Win 7 (para este tutorial, funciona también con Xp)
* Matlab 2009 o 2010a/b

**Entorno de Problema**

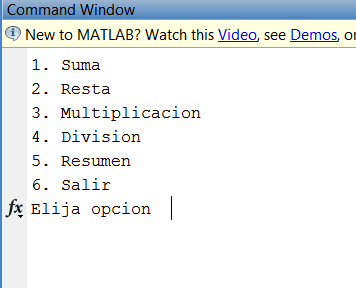
Debemos implementar una calculadora que realize sumas, restas, multiplicacion y division y que los resultados de estas operaciones se guarden en una base de datos que en este caso sera una base creada en Mysql

**Propuesta de Solucion**

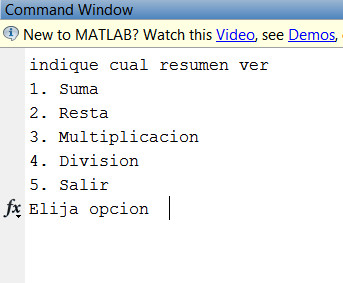
Lo que proponemos es una calculadora simple y en modo texto, claro que con el conocimiento adquirido por este solucion sera de utilidad para la implementacion grafica que es solamente la creacion de los botones y demas opciones graficas que se conectaran a lo que se tenga en modo texto.

Antes de empezar a desgranar el codigo de matlab, hay que ver que es lo que realmente hace para que no perdais el tiempo en un codigo que no os interesara.

Pues bien

[](https://vertigo2040.files.wordpress.com/2011/01/1.png)Menu principal

en la primera figura podeis visualizar el menu principal de la aplicacion en matlab, sus 4 opciones de operaciones basicas que al ser accesada por el numero que  las presede, ademas de una 5ta opcion que es el resumen el cual nos dara el listado de resultados guardados en la base de datos por cada operacion dada (ver figura 2)

[](https://vertigo2040.files.wordpress.com/2011/01/2.png)

podemos decir tambien que este programa o script como se le llama a los codigos de matlab es muy simple, pues solamente pide uno a uno los valores a evaluar( limite de 2 variables numericas a evaluarse) al hacer la suma o resta u otra operacion el resultado lo mostrara en pantalla y mandara este mismo valor a la base de datos de Mysql.

para la figura 2 este menu se  base en pedir de la base de datos de Mysql los datos grabados por el primer menu (figura 1) segun lo que se requiera, esto es, solo mostrara los valores que fueron grabados en digamos suma y si quiere verse los grabados en resta debera ser accesado de nuevo en el menu para ver el siguiente valor de busqueda, estas busquedas las mostrara en pantalla (modo texto)

**La descarga**

[](http://www.megaupload.com/?d=0Q6910II)

[DESCARGA EL SCRIPT](http://www.megaupload.com/?d=0Q6910II)

**el codigo**

Os recuerdo que al pegarlo este codigo en wordpress pues se desformatio de algunos caracteres que matlab usa y que al volverlo a copiar les dara algunos errores que deberan arreglar, pero os aconsejo mejor descargar el punto m de arriba o descargar este archivo de word [calculadora](https://vertigo2040.files.wordpress.com/2011/01/calculadora.docx), por tiempo no os podre explicar paso a paso el codigo, pero os aseguro que lo entenderan, que rayos!, la primeras 2 lineas son vairables que defini porque me daban problemas los ciclos while y debian ser inicializados a 0 sus variables de retorno, los demas codigos  son el encabezado que el querybuilder crea para configurar la base de datos(si no teneis el conector obdc, ni una base de datos en mysql, dificilmente les correra).

pese a que el codigo no les correra en la etapa de peticion a la base de datos, el programa es 100% funcional si:

1. tienen el conector obdc ver: <https://vertigo2040.wordpress.com/2011/01/13/conexion-de-mysql-en-matlab/>
2. tienen una base de datos que tenga las tablas suma, resta, multiplicacion y division ver: <https://vertigo2040.wordpress.com/2011/01/01/creando-una-base-de-datos-en-mysql/>
3. tener Xammp activado

de otra manera solo sera de referencia para ver como se graba hacia la base de datos y como se pide datos de esta para mostrarla en matlab

opcion = 0;

opcion2=0;

**encabezado de configuraciones de query builder para trabajar con el conector obsc para mysql**

% Set preferences with setdbprefs.

s.DataReturnFormat = ‘cellarray’;

s.ErrorHandling = ‘store’;

s.NullNumberRead = ‘NaN’;

s.NullNumberWrite = ‘NaN’;

s.NullStringRead = ‘null’;

s.NullStringWrite = ‘null’;

s.JDBCDataSourceFile = ”;

s.UseRegistryForSources = ‘yes’;

s.TempDirForRegistryOutput = ‘C:\Users\sonia\AppData\Local\Temp’;

s.DefaultRowPreFetch = ‘10000’;

setdbprefs(s)

% Make connection to database.  Note that the password has been omitted.

% Using ODBC driver.

%conn = database(‘127.0.0.1′,”,’password’);

**prime while o ciclo, este es el menu**

while opcion ~= 6

clc

fprintf(‘1. Suma \n’);

fprintf(‘2. Resta \n’);

fprintf(‘3. Multiplicacion \n’);

fprintf(‘4. Division \n’);

fprintf(‘5. Resumen \n’);

fprintf(‘6. Salir \n’);

opcion=input(‘Elija opcion  ‘);

switch opcion

case 1

x=input(‘Ingrese primer numero  ‘);

y=input(‘Ingrese segundo numero  ‘);

S1=x+y;

fprintf(‘La suma es: %f \n’,S1);

%se conecta a la base de datos por el conector obdc

conn = database(‘127.0.0.1′,”,’password’);

% Write data to database. escribe los datos a la base

insert(conn,’suma’,{‘Suma’},S1)

% Close database connection. cierra la coneccion

close(conn)

pause();

case 2

x=input(‘Ingrese primer numero  ‘);

y=input(‘Ingrese segundo numero  ‘);

S1=x-y;

fprintf(‘La diferencia es: %f \n’,S1);

conn = database(‘127.0.0.1′,”,’password’);

% Write data to database.

insert(conn,’resta’,{‘resta’},S1)

% Close database connection.

close(conn)

pause();

case 3

x=input(‘Ingrese primer factor  ‘);

y=input(‘Ingrese segundo factor  ‘);

S1=x\*y;

fprintf(‘El producto es: %f \n’,S1);

conn = database(‘127.0.0.1′,”,’password’);

% Write data to database.

insert(conn,’multiplicacion’,{‘multiplicacion’},S1)

% Close database connection.

close(conn)

pause();

case 4

x=input(‘Ingrese dividendo  ‘);

y=input(‘Ingrese divisor  ‘);

S1=x/y;

fprintf(‘La suma es: %f \n’,S1);

conn = database(‘127.0.0.1′,”,’password’);

% Write data to database.

insert(conn,’division’,{‘division’},S1)

% Close database connection.

close(conn)

pause();

case 5

while opcion2 ~= 5

clc

fprintf(‘indique cual resumen ver\n’);

fprintf(‘1. Suma \n’);

fprintf(‘2. Resta \n’);

fprintf(‘3. Multiplicacion \n’);

fprintf(‘4. Division \n’);

fprintf(‘5. Salir \n’);

opcion2=input(‘Elija opcion  ‘);

switch opcion2

case 1

conn = database(‘127.0.0.1′,”,’password’);

% Read data from database.

e = exec(conn,’SELECT ALL Suma FROM suma’);

e = fetch(e);

close(e)

% Assign data to output variable.

A = e.Data

pause();

% Close database connection.

close(conn)

case 2

conn = database(‘127.0.0.1′,”,’password’);

% Read data from database.

e = exec(conn,’SELECT ALL Resta FROM resta’);

e = fetch(e);

close(e)

% Assign data to output variable.

A = e.Data

pause();

% Close database connection.

close(conn)

case 3

conn = database(‘127.0.0.1′,”,’password’);

% Read data from database.

e = exec(conn,’SELECT ALL Multiplicacion FROM multiplicacion’);

e = fetch(e);

close(e)

% Assign data to output variable.

A = e.Data

pause();

% Close database connection.

close(conn)

case 4

conn = database(‘127.0.0.1′,”,’password’);

% Read data from database.

e = exec(conn,’SELECT ALL Division FROM division’);

e = fetch(e);

close(e)

% Assign data to output variable.

A = e.Data

pause();

% Close database connection.

close(conn)

case 5

opcion2=5;

otherwise

fprintf(‘Gracias..Adios  \n’);

end

end

otherwise

fprintf(‘Gracias..Adios  \n’);

end

end

<https://vertigo2040.wordpress.com/2011/01/24/implementacion-a-manera-de-ejemplo-calculadora-que-guarda-resultados-en-db-mysql/>

**Conexión De MySQL en Matlab**

[https://www.paypal.com/en_US/i/btn/btn_donate_LG.gif](http://vertigo.alojamientogratuito.info/wordpress/?p=317)Ayuda para que siempre encuentres lo que buscas!

Conexión De MySQL en Matlab

**Conocimientos previos:**

* Saber crear una base de datos en MySql a nivel básico
  + Leer esto sin no sabes: <https://vertigo2040.wordpress.com/2011/01/01/creando-una-base-de-datos-en-mysql/>
  + Saber trabajar con Matlab a nivel básico
  + Saber trabajar con Xammp
  + Saber entrar a aplicación: Orígenes de datos ODBC (no indispensable)

**Necesitamos :**

* Aplicación de conexión a MySql por medio de conector Connector/ODBC
* Pagina de descarga: <http://www.mysql.com/downloads/connector/>
* Tener activo la aplicación “Orígenes de datos ODBC”
* Entorno de trabajo Win 7 (para este tutorial, funciona también con Xp)
* Matlab 2009 o 2010a/b

**Entorno de Problema**

Queremos conectar una base de datos del tipo Mysql a Matlab, para poder accesar a ella para consultas y también poder ingresarles valores  para agregar mas datos.

Resolucion del Problema

A manera de Indice de contenidos veremos esto

* Instalacion de Aplicaciones ODBC
* Configuracion de nuestra primera base con el conector ODBC
* Conexión con Matlab, con el Wizard “querybuilder”
* Creacion de nuestra primera consulta a la base de datos con “querybuilder”
* Cracion de nuestra primera inserción de datos a base con “querybuilder”
* Generacion de nuestros archivos \*.m
* Implementación a manera de ejemplo: Calculadora que guarda resultados en DB MySql

**Instalacion de Aplicaciones ODBC:**

Esta aplicación esta en el repositorio de MySql.com que es una herramienta para poder conectar esta base de datos (MySql) a cualquier entorno de programación sin necesidad de entrar de lleno a un intrincado proceso de manipulación de bases de datos.

Primero descargamos el archivo de la solución planteada:

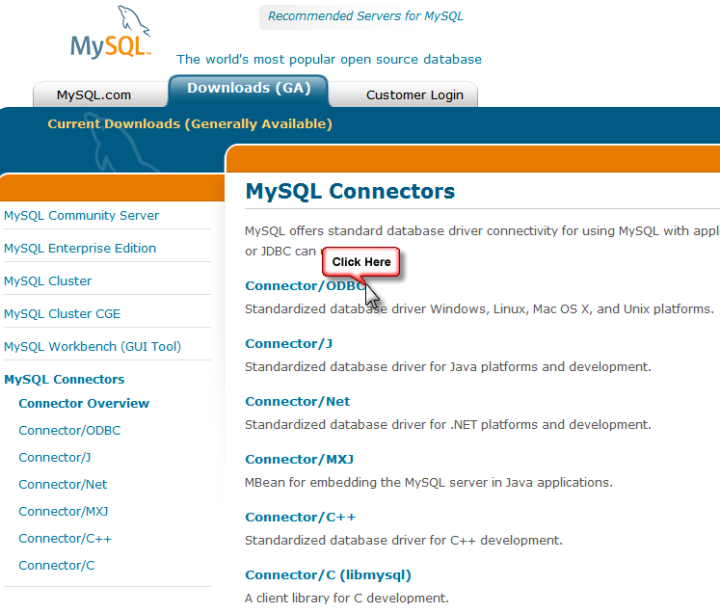
[](https://vertigo2040.files.wordpress.com/2011/01/fig-1-coneccion-mysql.png)

Figura 1: <http://www.mysql.com/downloads/connector/>

Esto os dara un ejecutable llamado “mysql-connector-odbc-5.1.8-winx64” o “mysql-connector-odbc-5.1.8-winx32” según el que sea para sus procesador 64bit o 32bit según la versión que este en ese momento deberá ser esta o superior a la 5.1.8. procedemos a instalarlo

[](https://vertigo2040.files.wordpress.com/2011/01/fig-2-instalacion-conector.png)

Figura 2

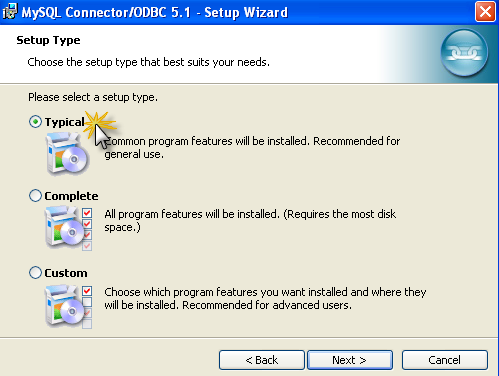
[](https://vertigo2040.files.wordpress.com/2011/01/fig-3-instalacion-conector.png)

Figura 3

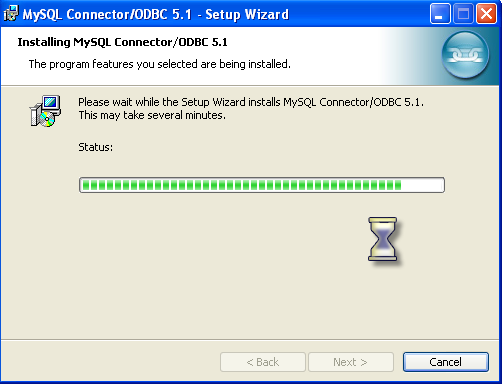
[](https://vertigo2040.files.wordpress.com/2011/01/fig-4-instalacion-conector.png)

Figura 4

**Configuracion de nuestra primera base con el conector ODBC**

Al terminar instalación tendremos nuestro conector en el repositorio de window. Ahora procedemos a configurarlo, entrando a

* Inicio
* Panel de control
* Todos los elementos de Panel de control ( si la vista no esta completa en panel de control)
* Herramientas administrativas
* Orígenes de datos ODBC

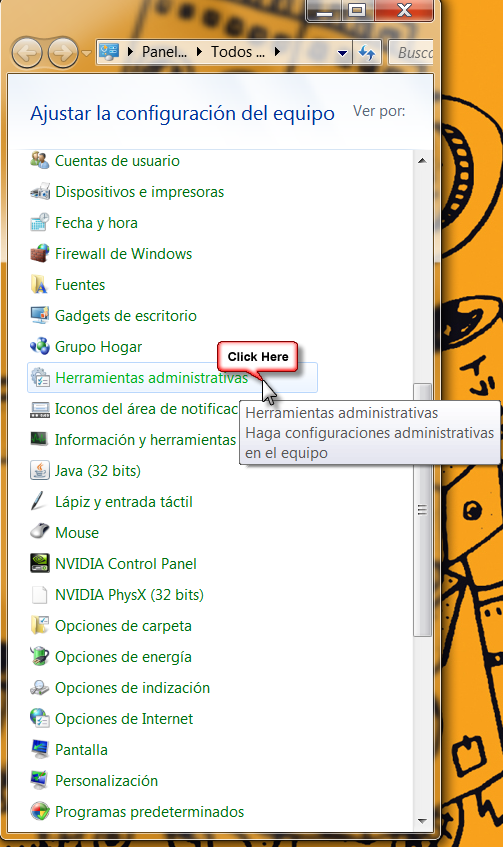
[](https://vertigo2040.files.wordpress.com/2011/01/5-herramientas-admin.png)

Figura 5

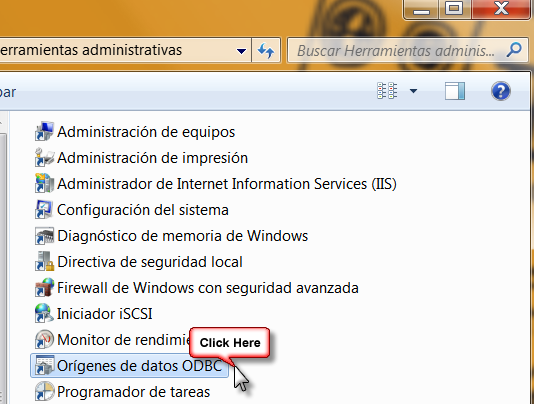
[](https://vertigo2040.files.wordpress.com/2011/01/6-odbc.png)

Figura 6

Antes de entrar a ella, debes saber que tienes que tener corriendo los 2 servicios de Xammp, Apache y Mysql(figura 7), como aver creado tu primera base de datos, para nuestro ejemplo tenemos creado una base de datos llamada “totales” la cual posee 5 columnas (figura 8)

* Id
* Suma
* Resta
* Multiplicación
* Division

Las que nos servirán para guardar resultados de estas operaciones

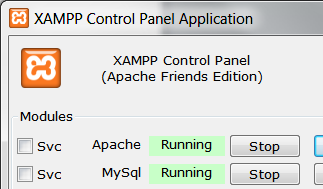
[](https://vertigo2040.files.wordpress.com/2011/01/7-xammp-runnig.png)

Figura 7

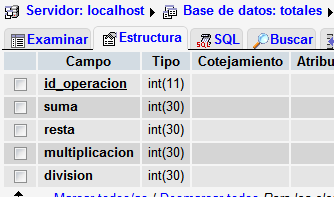
[](https://vertigo2040.files.wordpress.com/2011/01/8-base.png)

Figura 8

Teniendo esto en cuenta vamos a el apartado antes dicho (Panel de control\Todos los elementos de Panel de control\Herramientas administrativas), entramos a “Orígenes de datos ODBC”

Ahí seguimos estos pasos:

* Pestaña DSN Usurio
* Click botón Agregar ( si os dais cuenta tenemos MySQL ODBC 5.1 ahí)
* Seleccionar “MySQL ODBC5.1 Driver” (figura 9)
* Poner finalizar
* Te saldara otra ventana (figura 10)
  + Aquí debes ingresar
  + Data Source name: el nombre de tu servidor donde alojas las paginas (este nombre es un referencia, puedes poner cualquiera, pero este nombre sera reconocido por los entornos de programacion y no el de tu base de Mysql)
  + User: el usuario con que quieres accesar a la base
  + Password: la contraseña si tiene este usuario
  + En la casilla database debes de elegir tu base de datos, pues por cada conexion que añadas solamente una base sera reconocida.

Existe en esta misma ventana (figura 10) otra casilla llamada “database” esta te listara todas las bases de datos  que contenga este servidor y por supuesto aparecerá la tuya creada, dale a test y si todo a salido bien te lo indicara con éxito!

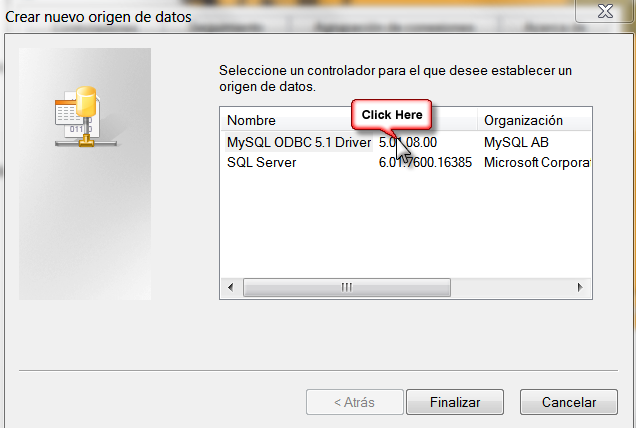
[](https://vertigo2040.files.wordpress.com/2011/01/9-agregar.png)

Figura 9

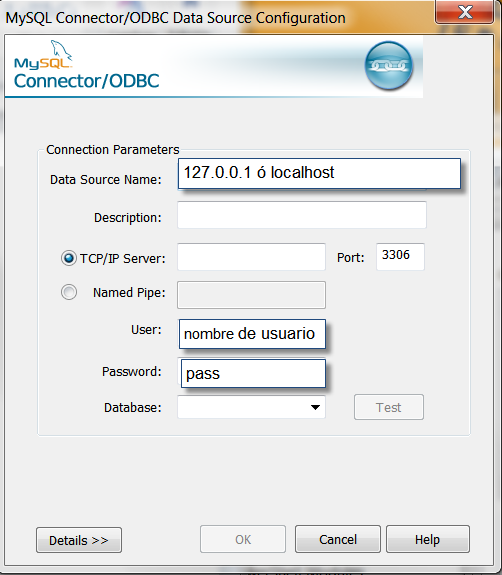
[](https://vertigo2040.files.wordpress.com/2011/01/10-usuario.png)

Figura 10

**Conexión con Matlab, con el Wizard “querybuilder”**

Si todo a salido bien tendrá tu conexión con MySQL totalmente limpia y sin problemas, ahora para que tengas la opción de interactuar con ella en MATLAB este tiene una serie de toolbox que te ayudaran en este sentido. Pero nosotros hablaremos del **“querybuilder” (figura 11)** este es una aplicación que sale en modo grafico en el entorno de trabajo de matlab.

* Entra al entorno de matlab
* Teclea:  **querybuilder**

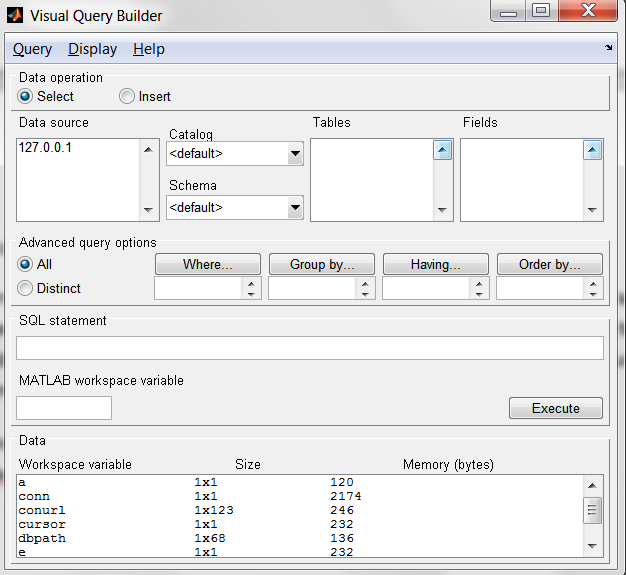
[](https://vertigo2040.files.wordpress.com/2011/01/11-querybuilder.png)

Figura 11

Si os dai cuenta saldrá ya nuestro servidor listado, si no es asi entonces la configuración anterior donde administrábamos nuestra base por medio del conector fallo.

Rapidamente

* Data operation: Select, para pedir info de la base y Insert, para insertar algún valor
* Data source: la ubicación de nuestras bases de datos que utilizan los conectores ODBC
* Catalog: lista las tablas que contiene esta base de datos seleccionada en DATA SOURCE
* Schema: si tiene alguna logia especial
* Tables:  los contenidos en cada uno de estas bases que existen en catalog
* Fields: los contenidos en cada una de las Tables
* Advanced Query option: cosas mas avanzadas para la gestión de datos
* Sql Statement: nos da el comando para trabajarlo desde consola de matlab
* MATLAB workspace variable: este nos sirve para asignar una variable de retorno después de solicitar una búsqueda en la base de datos y si esta en opción insert sirve para ingresar desde consola de Matlab un valor determinado partiendo de una variable creada anteriormente en matlab.
* Data: mira todas las variables creadas en el entorno Matlab.

**Creacion de nuestra primera consulta a la base de datos con “querybuilder”**

**Querybuilder** es un entorno grafico en matlab que nos ayuda muchísimo para trabajar con las bases de datos, no solo sirve para Mysql, sino para cualquier otra que este instalada correctamente en los conectores ODBC

Bueno como lo mencionábamos tenemos en este servidor una base llamada totales que contiene un campo llamado operaciones, la cual contiene 5 columas, id, suma, resta, multiplicación y división, las cuales contendrán los resultados de estas operación realizadas en el entorno Matlab (figura 12) esto lo veremos si selecionamos nuestro servidor

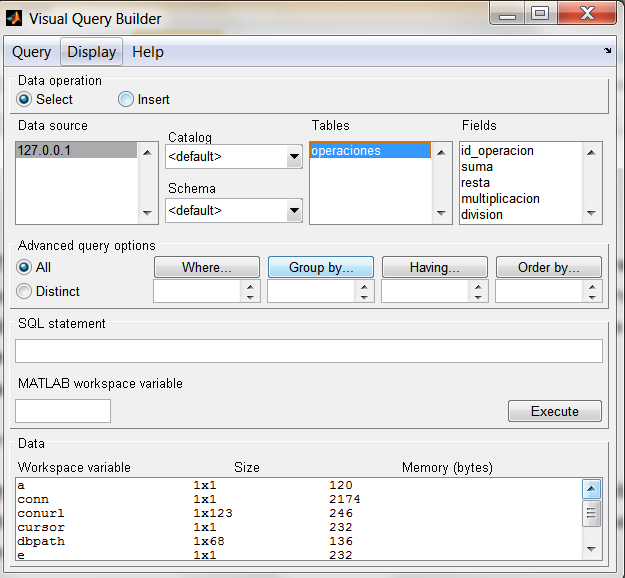
[](https://vertigo2040.files.wordpress.com/2011/01/12-entorno-grafico.png)

Figura12

Nuestra base esta casi limpia, solo el campo resta tiene un valor el “1”, entonces vamos a buscarlo

* Seleccionamos nuestro servidor (127.0.0.1)
* Selecionamos nuestra lista de bases y buscamos la base “totales”(nosotros no lo hacemos porque esta como default configurada)
* Selecionamos nuestra tabla (operaciones)
* Selecionamos en nuestra field (resta)

Esto creara en el espacio SQL Statemnet un código para ingreso en consola matlab, nosotros debemos de asignar una variable de retorno o dicho de otra manera una variable donde guardaremos lo que busque de la base, le pondremos “Var1” por ejemplo, luego buscamos el botón “Execute”, esto crea la variable Var1, hace la búsqueda en la base, exactamente en resta y guarda todo lo que encuentra ahí (todos los datos de esa pila) en la variable “Var1” la cual aparecerá en “workspace variable” en esta misma ventana, para ver lo que trajo solo hay que darle clik encima de esa variable creada y veras el contenido que trajo de la base de datos.( Figura 13 )

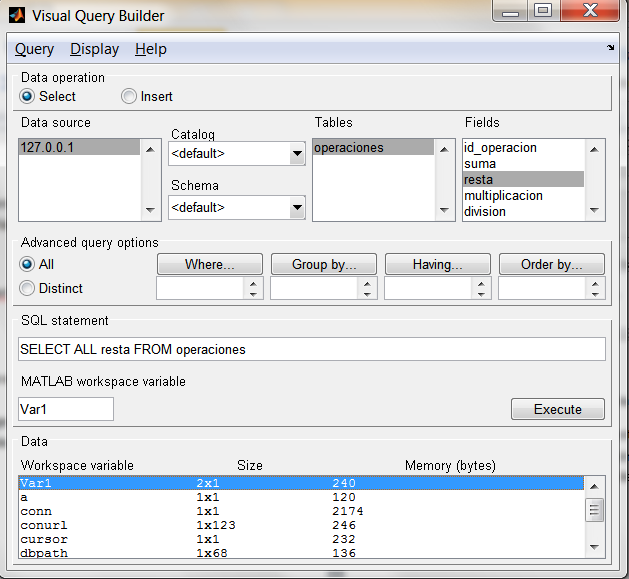
[](https://vertigo2040.files.wordpress.com/2011/01/13-busqueda.png)

Figura 13

**Cracion de nuestra primera inserción de datos a base con “querybuilder”**

Al igual que como en una búsqueda se utilizara este entorno grafico, la variante será que en “Data operation” se cambia de “Select” a “Insert” esto anulara algunas opciones de la pantalla, dejando solamente las necesarias para ingresar datos a la base.

Bamos a ingresar u numero, el  “4” a la base de datos en la casilla resta, la misma que utilizamos en el tema anterior y que solo contenia el numero “1” dentro de ella.

* Crear una variable en en el entorno de matlab (en consola) y grabarle el  numero que querrais, a nuestro ejemplo 4
  + export\_data = {4};
* esta variable “export\_data” se copia en el apartado MATLAB Workspace variable

Si os dais cuenta también nos proporciona un código para que podamos ingresarlo en consola, pues solo nos quedaría ponerle al botón Execute y este programin nos hara el favor de subirlo a nuestra base de datos. Ahora solo nos quedaría buscarlo y verificar si lo a hecho bien. Recordad que debéis primero elegir de nuevo el servidor y el lugar donde quereis los datos a guardar y si os pregunta si os guarda el dato anterio decidle que no y limpiara toda vairable dentro de esta ventana. Primero pones la variable creada y luego le das en execute. (figura 14)

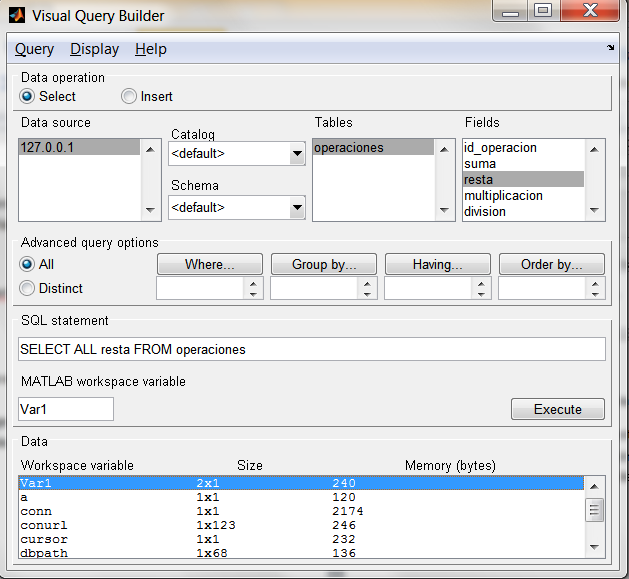
[](https://vertigo2040.files.wordpress.com/2011/01/14-muetra-de-datos.png)

Figura 14

**Generacion de nuestros archivos \*.m**

Que lindo todo lo que hace el entorno grafico de manipulación de base de datos, pero y como lo convertimos a un ejecutable para matlab, pues sencillo, el  mismo entorno de lo deja hacer y te lo crea desde una de sus opciones.

* Opción “Query”
* Click en “Generate M-File”

y de asi te hable el editor de instrucciones con las configuraciones y todo para accesar a la instrucción que pusiste. Osea si realizaste una búsqueda, al hacer esta opción te generara el código a una búsqueda y si lo haz hecho para una inserción de datos pues ese código genereara, pero eso es bueno, pues entenderas cuales son las variantes de queda petición y lo que realmente no puede dejar de ir en el código. (figura 15)

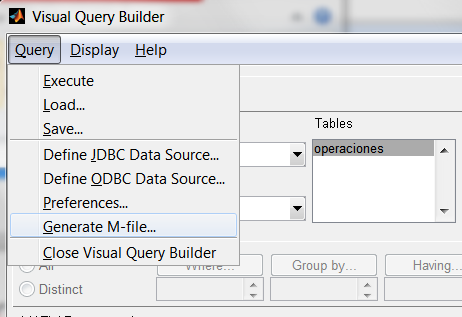
[](https://vertigo2040.files.wordpress.com/2011/01/15-punto-m.png)

figura 15

estos codigos a partir de que tengas una base de datos como la de nuestro ejemplo.

código de Busqueda:

% Set preferences with setdbprefs.

s.DataReturnFormat = ‘cellarray’;

s.ErrorHandling = ‘store’;

s.NullNumberRead = ‘NaN’;

s.NullNumberWrite = ‘NaN’;

s.NullStringRead = ‘null’;

s.NullStringWrite = ‘null’;

s.JDBCDataSourceFile = ”;

s.UseRegistryForSources = ‘yes’;

s.TempDirForRegistryOutput = ‘C:\Users\ADMINI~1\AppData\Local\Temp’;

s.DefaultRowPreFetch = ‘10000’;

setdbprefs(s)

% Make connection to database.  Note that the password has been omitted.

% Using ODBC driver.

conn = database(‘127.0.0.1′,”,’password’);

% Read data from database.

e = exec(conn,’SELECT ALL resta FROM operaciones’);

e = fetch(e);

close(e)

% Assign data to output variable.

a = e.Data

% Close database connection.

close(conn)

código de inserción teniendo una variable llamada export\_data con algún valor

% Set preferences with setdbprefs.

s.DataReturnFormat = ‘cellarray’;

s.ErrorHandling = ‘store’;

s.NullNumberRead = ‘NaN’;

s.NullNumberWrite = ‘NaN’;

s.NullStringRead = ‘null’;

s.NullStringWrite = ‘null’;

s.JDBCDataSourceFile = ”;

s.UseRegistryForSources = ‘yes’;

s.TempDirForRegistryOutput = ‘C:\Users\ADMINI~1\AppData\Local\Temp’;

s.DefaultRowPreFetch = ‘10000’;

setdbprefs(s)

% Make connection to database.  Note that the password has been omitted.

% Using ODBC driver.

conn = database(‘127.0.0.1′,”,’password’);

% Write data to database.

insert(conn,’operaciones’,{‘resta’},export\_data)

% Close database connection.

close(conn)