

BASES DE DATOS MATISSE

DB
DB.io
Database of
Databases

Matisse
The Post-Relational SQL Database

INTRODUCCIÓN

Matisse es un Sistema Gestor de Bases de Datos Orientado a Objetos (SGBOO) desarrollado por ADB Inc. Fue lanzado por primera vez en 1995 y desde entonces ha sido utilizado por una amplia gama de aplicaciones, desde aplicaciones empresariales hasta aplicaciones web.

Matisse se basa en un modelo de datos orientado a objetos, lo que significa que los datos se representan como objetos. Esto ofrece una serie de ventajas, como una mayor flexibilidad y escalabilidad, así como un mejor rendimiento para aplicaciones que manipulan datos complejos.

En este trabajo, exploraremos las características y ventajas de Matisse, así como su integración con el lenguaje de programación Java. También proporcionaremos nuestra opinión sobre el SGBD.

CARACTERÍSTICAS Y VENTAJAS

Matisse ofrece una amplia gama de características y ventajas, entre las que se incluyen:

Modelo de datos orientado a objetos: Matisse se basa en un modelo de datos orientado a objetos, lo que significa que los datos se representan como objetos. Esto ofrece una serie de ventajas, como una mayor flexibilidad y escalabilidad, así como un mejor rendimiento para aplicaciones que manipulan datos complejos.

Soporte para objetos nativos: Matisse admite objetos nativos, lo que significa que los objetos se almacenan en su forma original, sin necesidad de mapeo relacional de objetos. Esto puede mejorar el rendimiento y la eficiencia de las aplicaciones.

Soporte para XML: Matisse admite XML, lo que lo hace adecuado para aplicaciones que requieren el intercambio de datos con otras aplicaciones o sistemas.

Arquitectura cliente-servidor: Matisse utiliza una arquitectura cliente-servidor, lo que permite a los clientes acceder a los datos desde cualquier lugar.

Seguridad: Matisse proporciona una serie de funciones de seguridad para proteger los datos, como la autenticación, la autorización y el cifrado.

Para la plataforma Windows, la base de datos tiene una herramienta para modelar llamada Matisse Modelador de Base de Datos, pudiendo hacer un modelado totalmente orientado a objetos, con sus clases, atributos, métodos y relaciones entre ellos. Las clases, donde puede guardar el modelo, imprimirlo y exportarlo como una imagen. Como la base de datos está activa, automáticamente los cambios realizados en el modelo se llevan a cabo en ella.

Además de la herramienta de modelado, también existe la herramienta para gestión del banco llamado Matisse Gerente de Empresa. Con esta herramienta es posible monitorizar los distintos Esquemas de base de datos creados. Las principales características son:

- Activar o deshabilitar esquemas;
- Monitor ejecución de los regímenes;
- Gestionar el tamaño de los regímenes;
- Gestionar las clases, metaclasses y datos de los esquemas,
- Gestionar la seguridad, los usuarios de los sistemas bancarios;
- Hacer copias de seguridad y registros de los esquemas, etc.

Además, en cada esquema es posible importar y exportar ODL, que son lenguajes de descripción de objetos utilizados en el modelado de la base de datos, y DDL's, que son los lenguajes de definición de datos como SQL y en el caso de de BDOO's al lenguaje OQL. Por lo tanto, es posible que el usuario cambie por completo o parcialmente las clases y otras estructuras contenidas en un esquema bancario.

La diferencia con Matisse es que puedes usar rutinas para manipular objetos en la base de datos o para usar rutinas SQL con del lenguaje OQL.

PRECIO

El precio de Matisse depende de la cantidad de datos que se almacenen y de la cantidad de usuarios que accedan a la base de datos. Para obtener información sobre los precios, puede ponerse en contacto con ADB Inc.

Características adicionales

Además de las características y ventajas mencionadas anteriormente, Matisse también ofrece una serie de características adicionales, como:

Soporte para transacciones: Matisse admite transacciones, lo que garantiza que las operaciones se realicen de forma consistente.

Soporte para recuperación de desastres: Matisse proporciona funciones de recuperación de desastres para proteger los datos en caso de un desastre.

Soporte para replicación: Matisse permite replicar los datos en varios servidores, lo que puede mejorar la disponibilidad y el rendimiento.

INTEGRACIÓN CON JAVA

```
public class MyObject {  
    private int id;  
    private String name;  
  
    public MyObject(int id, String name) {  
        this.id = id;  
        this.name = name;  
    }  
  
    public int getId() {  
        return id;  
    }  
  
    public void setId(int id) {  
        this.id = id;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
}
```

Para almacenar un objeto de MyObject en Matisse, puede utilizar el siguiente código:

```
MatisseDatabase db = new MatisseDatabase();  
MyObject object = new MyObject(1, "My Object");  
db.insertObject(object);
```

Para recuperar un objeto de MyObject de Matisse, puede utilizar el siguiente código:

```
MyObject object = db.getObjectById(1);
```

INSTALACIÓN DEL SERVIDOR

Para poder crear el servidor primero debemos descargar el instalador desde la página oficial de Matisse



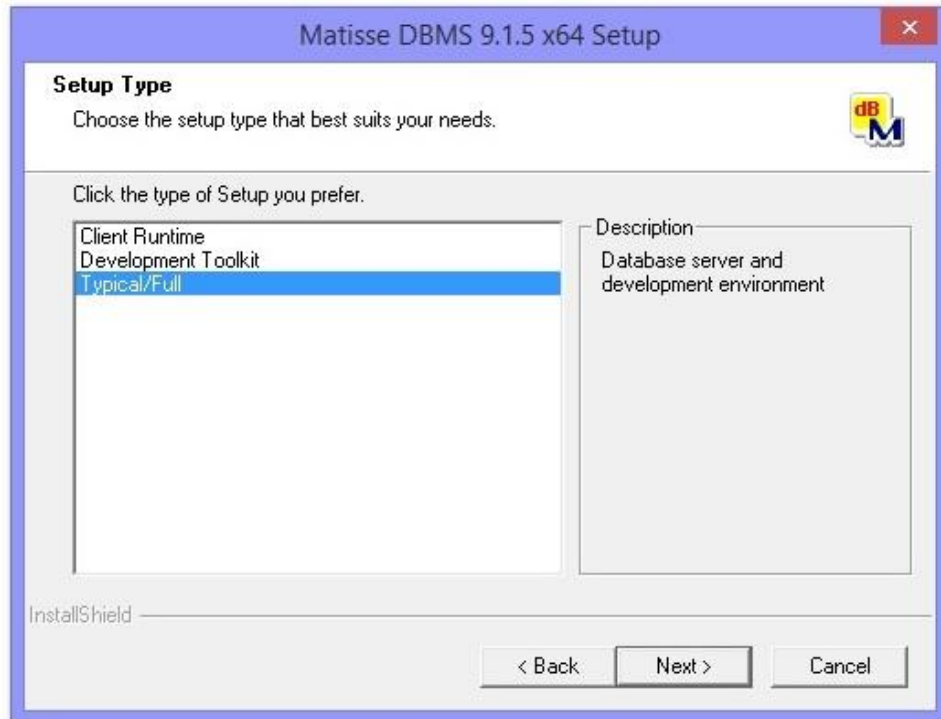
Para instalar la versión 9.1.5 (la versión más reciente en el sitio web) En la plataforma Windows, es necesario instalar .NET 3.0 o superior y Java Runtime Environment 1.7.0 o superior y al menos Windows Vista.

La instalación es muy sencilla: con dos clics se completa el procedimiento. En el ejemplo de este artículo lo instalamos en Windows 8 de 64 bits, pero No hay diferencia con la versión de 32 bits. El único requisito previo es la instalación del JRE 1.7.0, porque si ya está instalado en la máquina, Matisse no instala su propio JRE. El procedimiento de instalación debe realizarse siempre como usuario administrador, de lo contrario la instalación no funcionará.

Ilustraremos la instalación de la base de datos Matisse, Matisse DBMS, la herramienta de modelado, Matisse Database Modeler, y el para la interacción con aplicaciones .Net, Matisse Enlace .Net. El primer paso es aceptar los términos del contrato y A continuación, debe elegir el nombre del usuario y la organización, ya que La figura 8 muestra. La información de este paso dependen del usuario.

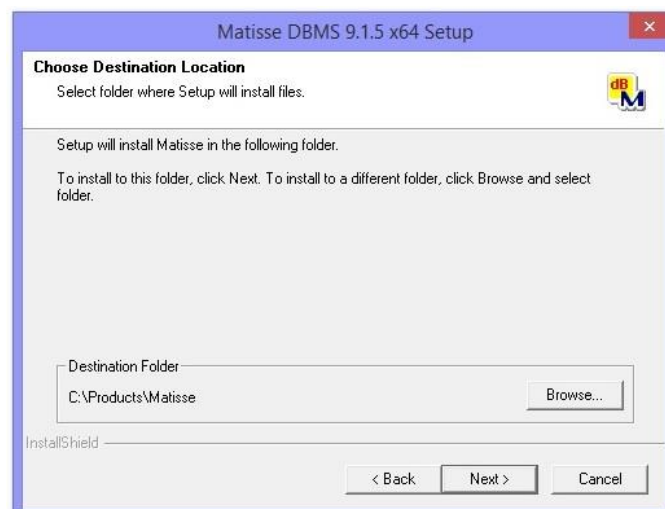
The image shows the 'Matisse DBMS 9.1.5 x64 Setup' window. The title bar indicates the version and architecture. The main window has a 'Customer Information' section with the instruction 'Please enter your information.' and a sub-instruction 'Please enter your name and the name of the company for which you work.' There are two text input fields: 'User Name:' with the value 'Jullian' and 'Company Name:' with the value 'Home'. At the bottom, there are three buttons: '< Back', 'Next >', and 'Cancel'. The 'InstallShield' logo is visible in the bottom left corner.

El siguiente paso es elegir el tipo de instalación: tipo de Instalación típica o con todas las funciones con todas las funciones, como el modo cliente, Servidor y herramienta para la gestión bancaria. Elija la opción de instalación típico o completo



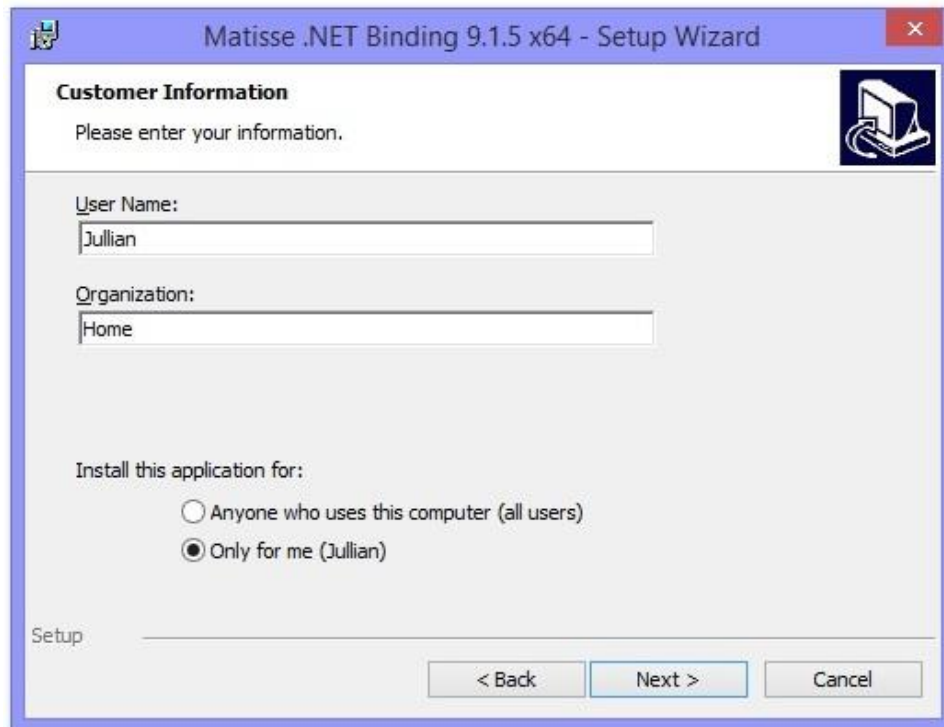
El siguiente y último paso es elegir el directorio de storage, que por defecto Matisse establece la ruta "C:\Products\Matisse" para evitar conflictos de virtualización en Windows Vista. En las otras versiones de la Windows puede usar la ruta "C:\Archivos de programa\Matisse", pero cuidado, porque en este caso tendrás que crear el archivo directorio llamado "Matisse" manualmente.

Una nota importante es que la desinstalación del asiento debe ser realizada por el propio instalador y como usuario administrador, de lo contrario, la desinstalación no se realiza correctamente.



El siguiente paso es instalar la herramienta de enlace Matisse.Net para ejecutar el Banca en aplicaciones .NET. El procedimiento de instalación es similar al mencionado encima.

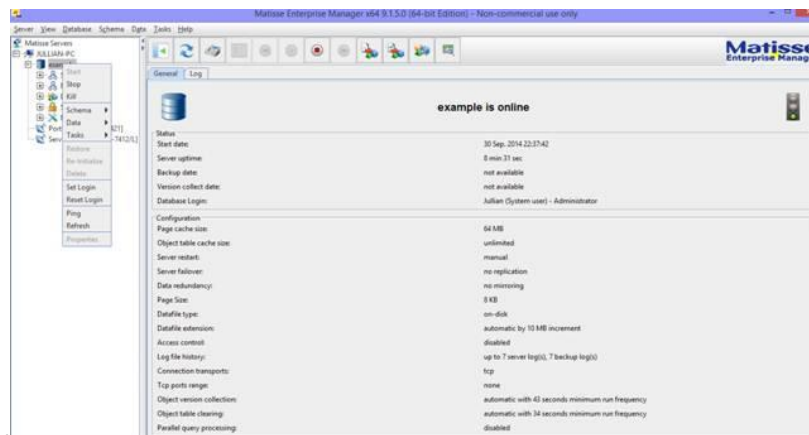
Primero, debe aceptar los términos del contrato. En A continuación, debe elegir el usuario y la organización, que deben ser los mismos que los utilizados en la instalación anterior (queda a discreción del usuario que la instalación sea para el usuario administrador o para otros usuarios del equipo



La última herramienta que se instala es la banco, Matisse Database Modeler, que También debe realizarse como usuario administrador. Los pasos de instalación son los mismos que en la herramienta anterior, incluidas las pantallas de navegación para instalación

Ahora vamos a configurarlo conexión a Matisse en el IDE de Netbeans y Visual Studio 2013, ambas versiones 64 bits. En primer lugar, es necesario iniciar manualmente un esquema bancario que has almacenado. En este ejemplo, vamos a usar el esquema de banco "ejemplo", hacia Inicializar un esquema bancario Haga clic con el botón derecho en el icono Esquemático y elija la opción "Inicio"

ACCESO A DATOS 2º DAM



El siguiente paso es configurar la conexión con el base de datos en su aplicación Java en el IDE de Netbeans. En esta publicación solo se le mostrará el conexión manual usando JDBC, hecha por línea de código y no que configuración de los servicios bancarios ofrecidos por el IDE a través de drivers.

Cree una aplicación Java estándar. Una vez hecho esto, importe el archivo Matisse ejecutable.jar que es ubicado dentro del directorio lib, dentro del directorio Matisse para dentro de la carpeta Libraries presente en el archivo su aplicación.



El siguiente paso es importar las bibliotecas desde el archivo recursos utilizados por el controlador JDBC de Java y Matisse, como se ilustra en el Código de listado 1. Importe el archivo Bibliotecas Java.SQL.* y Com.Matisse.* en la aplicación (líneas 2 y 3). Una vez hecho esto, cree cadenas para representar el host y el nombre del banco que desea utilizar (línea 7). Crear un objeto que represente el banco de Matisse Data y pasar como parámetros el host y el nombre del esquema bancario y abra la conexión con el banco, como se muestra en las líneas 9 y 10. A continuación, cree un conector JDBC, asignando un conector JDBC del banco Matisse por la función dada (línea 12). Después de eso, simplemente use los procedimientos comunes, como crear parámetro, rutina SQL, ejecución de la rutina SQL e impresión de los datos, como vemos en las líneas 15 a 22. Recuerde crear un control de excepciones para la conexión con la base de datos y para la ejecución de rutinas SQL.

Finalmente, cierre la conexión a la base de datos, al igual que en el archivo Línea 23.


```
import com.matisse.*;
import java.sql.*;

public class Test{
    public static void main (String[] arg) {
        String host="localhost", dbname="example"; //información sobre el host y el nombre de la base de datos
        try {
            MtDatabase db = new MtDatabase (host, dbname); //crear una instancia de base de datos
            db.open(); //open the connection to the bank
            System.out.println("Recuperar nombres de clases a través de una conexión JDBC obtenidos a través de una conexión MtDatabase");
            Connection jcon = db.getConnection(); //abrir conexión JDBC
            try {
                Statement stmt = jcon.createStatement(); //creación de procedimientos
                String query = "SELECT * FROM MtClass"; //creamos el comando sql
                ResultSet rs = stmt.executeQuery(query); //ejecutamos el comando sql
                System.out.println("result of: " + query);
                //Imprimir metaclasses presentes en el esquema de base de datos seleccionado
                while (rs.next()) {
                    System.out.println(rs.getString("MtName"));
                }
            } catch (SQLException e) {
                System.out.println("SQLException thrown: " + e.getMessage());
            }
        } catch (MtException mte) {
            System.out.println("MtException : " + mte.getMessage());
        }
    }
}
```

El Listado 2 ilustra la conexión utilizando JDBC solamente. Observe que en la línea 2 el Biblioteca Java.SQL.*. Una vez hecho esto, cree cadenas para representar el host y el nombre del esquema bancario que desee (línea 7). A continuación, configuremos una clase con el nombre com.matisse.sql.MtDriver, porque esto class representa el conductor para conectarse al banco Matisse (línea 10). En A continuación, cree una URL de conexión (línea 11) y cree un conector JDBC que pase como parámetro, la URL de conexión creada (línea 13). Los otros procedimientos son: Patrones JDBC como la creación de parámetros (línea 16), script SQL (línea 17) y Ejecución del script (línea 18). El bucle while se utiliza para imprimir el nombre de las clases presentes en el bucle Banco conectado. Recuerde colocar estos comandos dentro de los bloques para Excepciones.

Listado 2. Código en Java describiendo una conexión a la base de datos Matisse mediante el controlador JDBC.

```
package testejdbc;
import java.sql.*;

public class testeJDBC {

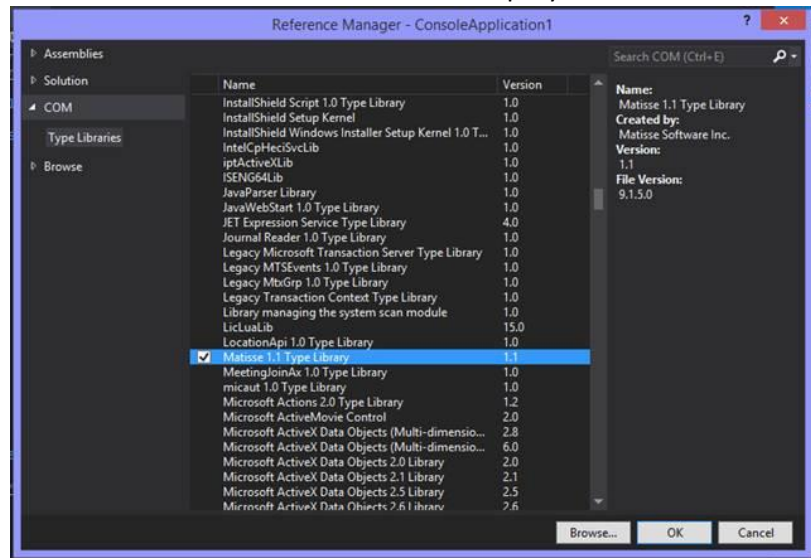
    public static void main (String[] args) {
        String host="localhost", dbname="example"; //información sobre el host y el nombre de la base de datos
        System.out.println("===== Conexión a DBC puro =====\n");
        try {
            Class.forName("com.matisse.sql.MtDriver");
            String url = "jdbc:mt://" + host + "/" + dbname; //Crear url
            System.out.println("Recuperación de nombres de clases a través de una conexión JDBC: " + url);
            Connection jcon = DriverManager.getConnection(url);

            // Acceso regular JDBC
            Statement stmt = jcon.createStatement(); //Creación de parámetro
            String query = "SELECT MtName FROM MtClass"; //Script SQL
            ResultSet rs = stmt.executeQuery(query); //Ejecución del Script
            System.out.println("Resultado de: " + query);
            while (rs.next()) {
                System.out.println(rs.getString("MtName"));
            }
            System.out.println("\nÉxito.");
        } catch (ClassNotFoundException e) {
            System.out.println("Clase de controlador Matisse JDBC no encontrada");
        } catch (SQLException e) {
            System.out.println("SQLException thrown: " + e.getMessage());
        }
    }
}
```

Ahora vamos a crear una conexión simple en aplicaciones .Net por Visual Studio 2013, específicamente una aplicación de consola en el OMB Es necesario tener instalada la herramienta de encuadernación Matisse.Net, de lo contrario no podrás conectarte a la base de datos de Matisse, ya que esta herramienta habilita la dll responsable del controlador de conexión de Matisse en Visual Studio. O El nombre de este archivo dll es Matisse 1.1 Type Library

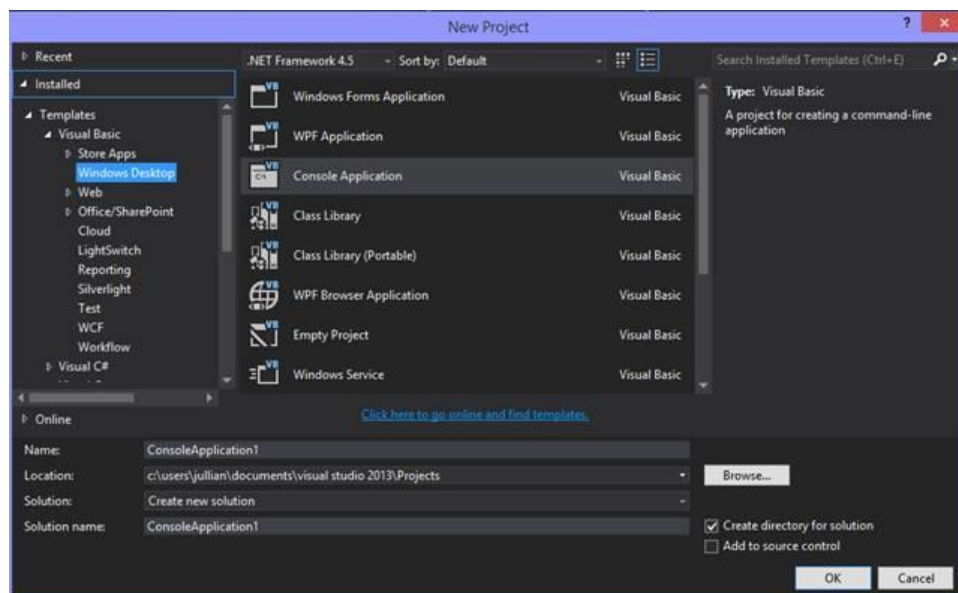
ACCESO A DATOS 2º DAM

Después de instalar esta herramienta automáticamente, el Visual Studio (que se está instalando) reconoce esta dll, porque está incorporado en la lista de dll's del framework .Net. Por lo tanto, no es necesario añadir referencias a la su proyecto o solución.



Una vez realizado este procedimiento, vamos a crear una solución de aplicación de consola VB

A continuación, cree una clase que represente el controlador de conexión al banco de Matisse y pasar como parámetros el host y el nombre del esquema con el que desea trabajar: en este ejemplo, es el host localhost y el ejemplo del esquema de base de datos (línea 9). Una vez hecho esto, el Los siguientes procedimientos son estándar como cualquier otro tipo de conexión para el Banco de Datos. Abra la conexión (línea 11), ejecute las transacciones (línea 12), confirme o cancelar las transacciones (línea 18) y cerrar la conexión (línea 19).



Ejemplo de código para la conexión

```
Imports System
Imports Matisse

Module Module1
    sub Main()
        Try
            'Creación de la clase que representa la conexión con el banco Matisse
            'Los parámetros pasados son el host y el nombre del esquema utilizado.
            Dim db As MtDatabase = new MtDatabase("localhost", "example")

            db.Open() 'Apertura de conexión
            db.StartTransaction() 'Ejecución de transacciones

            'Acceso a la base de datos para lectura/escritura
            Console.WriteLine("Conexión de lectura y escritura para acceder a " & _
                db.ToString())

            db.Commit() 'Confirmación de ejecución de transacción
            db.Close() 'Conexión cercana

        Catch mte as MtException
            Console.WriteLine("MtException : " & mte.Message)
        End Try
    End Sub
End Module
```

OPINIÓN

Matisse es una base de datos que me parece interesante pero en mi opinión se podrían utilizar programas mas actuales, me ha costado bastante encontrar la documentación para hacer el trabajo