Using System;

using System.Linq;

using System.Text;

public class JobLogger

{

private static bool \_logToFile;

private static bool \_logToConsole;

private static bool \_logMessage;

private static bool \_logWarning;

private static bool \_logError;

private static bool LogToDatabase; **es mejor mantener el standard en la declaración de variables con el prefijo \_**

private bool \_initialized;

public JobLogger(bool logToFile, bool logToConsole, bool logToDatabase, bool

logMessage, bool logWarning, bool logError)

{

\_logError = logError;

\_logMessage = logMessage;

\_logWarning = logWarning;

LogToDatabase = logToDatabase;

\_logToFile = logToFile;

\_logToConsole = logToConsole;

}

public static void LogMessage(string message, bool message, bool warning, bool

error)

{

message.Trim();

if (message == null || message.Length == 0)

{

return;

}

if (!\_logToConsole && !\_logToFile && !LogToDatabase)

{

throw new Exception("Invalid configuration");

}

if ((!\_logError && !\_logMessage && !\_logWarning) || (!message && !warning

&& !error))

{

throw new Exception("Error or Warning or Message must be specified");

}

System.Data.SqlClient.SqlConnection connection = new

System.Data.SqlClient.SqlConnection(System.Configuration.ConfigurationManager.AppS

ettings["ConnectionString"]);

connection.Open();

int t;

if (message && \_logMessage)

{

t = 1;

}

if (error && \_logError)

{

t = 2;

}

if (warning && \_logWarning)

{

t = 3;

}

System.Data.SqlClient.SqlCommand command = new

System.Data.SqlClient.SqlCommand("Insert into Log Values('" + message + "', " +

t.ToString() + ")");

command.ExecuteNonQuery();

string l;

if

(!System.IO.File.Exists(System.Configuration.ConfigurationManager.AppSettings["Log

FileDirectory"] + "LogFile" + DateTime.Now.ToShortDateString() + ".txt"))

{

l =

System.IO.File.ReadAllText(System.Configuration.ConfigurationManager.AppSettings["

LogFileDirectory"] + "LogFile" + DateTime.Now.ToShortDateString() + ".txt");

}

if (error && \_logError)

{

l = l + DateTime.Now.ToShortDateString() + message;

}

if (warning && \_logWarning)

{

l = l + DateTime.Now.ToShortDateString() + message;

}

if (message && \_logMessage)

{

l = l + DateTime.Now.ToShortDateString() + message;

}

System.IO.File.WriteAllText(System.Configuration.ConfigurationManager.AppSettings[

"LogFileDirectory"] + "LogFile" + DateTime.Now.ToShortDateString() + ".txt", l);

if (error && \_logError)

{

Console.ForegroundColor = ConsoleColor.Red;

}

if (warning && \_logWarning)

{

Console.ForegroundColor = ConsoleColor.Yellow;

}

if (message && \_logMessage)

{

Console.ForegroundColor = ConsoleColor.White;

}

Console.WriteLine(DateTime.Now.ToShortDateString() + message);

}

} \_\_