

**Tipo** : Guía de laboratorio  
**Capítulo** : Gestión de excepciones y seguimiento  
**Duración** : 60 minutos

---

## I. OBJETIVO

Administrar y configurar errores en una aplicación Web.

## II. REQUISITOS

Los siguientes elementos de software son necesarios para la realización del laboratorio:

- Windows 8.1
- SQL Server 2016
- Visual Studio 2017

## III. DESCRIPCIÓN

### Ejercicio 1

Implementar excepciones personalizadas Try/Catch y configurar CustomErrors en el Web.Config.

#### Actividades

1. Abrir la solución del capítulo anterior.
2. En el fichero Global.asax, agregamos el siguiente método:

```
protected void Application_Error(Object sender, EventArgs e)
{
    Exception ex = Server.GetLastError();
    if (ex is ThreadAbortException)
        return;
    Response.Redirect("http://localhost/WebForms/Error.aspx");
}
```
3. En el archivo Web.config, ubicamos la etiqueta: `<system.web>` y agregamos estas líneas:

```
<customErrors mode="On"
defaultRedirect="http://localhost/WebForms/Error.aspx"
redirectMode="ResponseRedirect">
    <error redirect="http://localhost/WebForms/Error.aspx"
statusCode="404"/>
</customErrors>
```

4. En el mismo archivo ubicamos la etiqueta: `<system.webServer>` y agregamos la siguiente línea:

```
<httpErrors errorMode="Custom" />
```

5. En la raíz de la solución web agregamos un archivo nuevo llamado: **Error.aspx**

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Error.aspx.cs"
Inherits="WebForms.Error" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent"
runat="server">
    <h1>Ups there is an error.</h1>
</asp:Content>
```

6. Para poder probar nuestro ejemplo agregamos un nuevo WebForm adicional llamado: **ErrorTest.aspx**.

- En el fichero.aspx, agregamos el siguiente código:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="ErrorTest.aspx.cs"
Inherits="WebForms.ErrorTest" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent"
runat="server">
    <asp:Button runat="server" onClick="ErrorButton_Click" Text="Generate
Error" />
</asp:Content>
```

- El code behind debe de quedar de la siguiente manera:

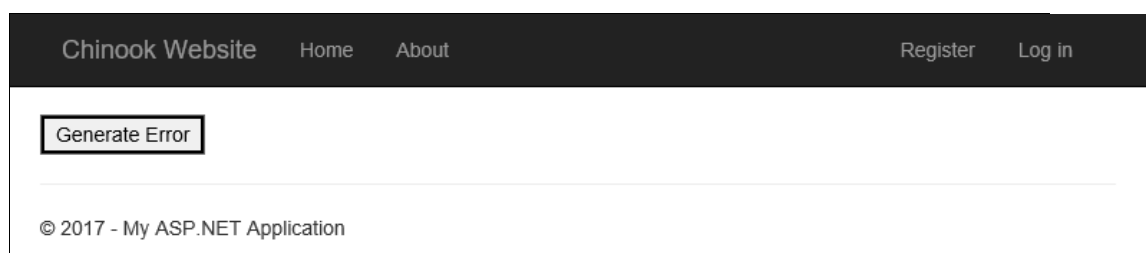
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebForms
{
    public partial class ErrorTest : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

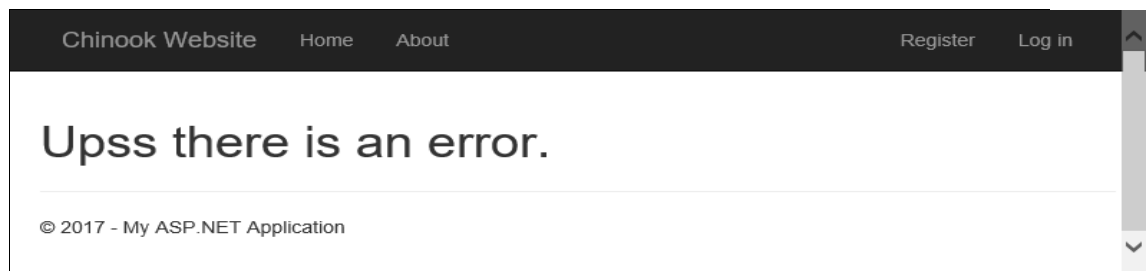
        }

        protected void ErrorButton_Click(object sender, EventArgs e)
        {
            throw new Exception("Este es un error de prueba");
        }
    }
}
```

7. Al ejecutar la aplicación y estar en el webform llamado **ErrorTest.aspx**, se debe observar que al presionar en el botón **Generate Error** nos debe de redirigir a la página de error. (Mirar imagen)



Resultado:



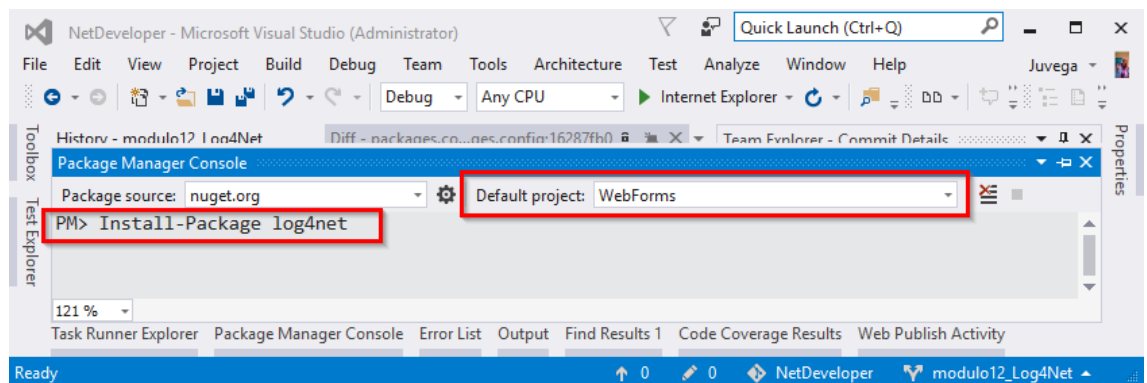
## Ejercicio 2

Configura Log4Net para el logueo de mensajes en un archivo almacenado en el disco duro.

### Actividades

1. Abrimos la solución anterior.
2. Instalamos el paquete log4net en el proyecto Web, desde la consola de instalación de paquetes:

`Install-Package log4net`



3. Finalizada la instalación del paquete, procedemos a configurar el uso del ensamblado:
  - Abrimos el fichero Web.config, ubicamos la etiqueta: `<configSections>` y agregamos la siguiente línea.

```
<section name="log4net"
type="log4net.Config.Log4NetConfigurationSectionHandler, log4net" />
```

- En el mismo archivo, agregamos la siguiente configuración:

```
<log4net>
  <appender name="RollingFile"
type="log4net.Appender.RollingFileAppender">
    <file value="C:\WebForms\Log.txt" />
    <appendToFile value="true" />
    <maximumFileSize value="10000KB" />
    <maxSizeRollBackups value="10" />
    <layout type="log4net.Layout.PatternLayout">
      <conversionPattern value="%date %-5level %logger -
%message%newline" />
    </layout>
  </appender>
</log4net>
```

```

    </appender>
  </root>
  <level value="ALL" />
  <appender-ref ref="RollingFile" />
</root>
</log4net>

```

- El resultado del fichero debe de ser el siguiente:

```

<?xml version="1.0" encoding="utf-8"?>
<!--
  For more information on how to configure your ASP.NET application,
  please visit
  http://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
  <configSections>
    <!-- For more information on Entity Framework configuration, visit
    http://go.microsoft.com/fwlink/?LinkId=237468 -->
    <section name="entityFramework"
    type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection,
    EntityFramework, Version=6.0.0.0, Culture=neutral,
    PublicKeyToken=b77a5c561934e089" requirePermission="false" />
    <section name="log4net"
    type="log4net.Config.Log4NetConfigurationSectionHandler, log4net" />
  </configSections>
  <connectionStrings>
    <add name="ChinookConnection" providerName="System.Data.SqlClient"
    connectionString="Server=.\SQLEXPRESS;Database=Chinook;Trusted_Connect
    ion=True" />
  </connectionStrings>
  <system.web>
    <customErrors mode="On"
    defaultRedirect="http://localhost/WebForms/Error.aspx"
    redirectMode="ResponseRedirect">
      <error redirect="http://localhost/WebForms/Error.aspx"
      statusCode="404" />
    </customErrors>
    <authentication mode="None" />
    <compilation debug="true" targetFramework="4.5.2" />
    <httpRuntime targetFramework="4.5.2" />
    <pages>
      <namespaces>
        <add namespace="System.Web.Optimization" />
        <add namespace="Microsoft.AspNet.Identity" />
      </namespaces>
      <controls>
        <add assembly="Microsoft.AspNet.Web.Optimization.WebForms"
        namespace="Microsoft.AspNet.Web.Optimization.WebForms"
        tagPrefix="webopt" />
      </controls>
    </pages>
    <membership>
      <providers>
        <!--
          ASP.NET Membership is disabled in this template. Please visit
          the following link http://go.microsoft.com/fwlink/?LinkId=301889 to
          learn about the ASP.NET Membership support in this template
        -->
        <clear />
      </providers>
    </membership>
  </system.web>

```

```

    <profile>
      <providers>
        <!--
          ASP.NET Membership Profile is disabled in this template.
          Please visit the following link
          http://go.microsoft.com/fwlink/?LinkId=301889 to learn about the
          ASP.NET Membership support in this template
        -->
        <clear />
      </providers>
    </profile>
    <roleManager>
      <!--
        ASP.NET Membership Role is disabled in this template.
        Please visit the following link
        http://go.microsoft.com/fwlink/?LinkId=301889 to learn about the
        ASP.NET Membership support in this template
      -->
      <providers>
        <clear />
      </providers>
    </roleManager>
    <!--
      If you are deploying to a cloud environment that has
      multiple web server instances,
      you should change session state mode from "InProc" to
      "Custom". In addition,
      change the connection string named "DefaultConnection" to
      connect to an instance
      of SQL Server (including SQL Azure and SQL Compact)
      instead of to SQL Server Express.
    -->
    <sessionState mode="InProc"
    customProvider="DefaultSessionProvider">
      <providers>
        <add name="DefaultSessionProvider"
        type="System.Web.Providers.DefaultSessionStateProvider,
        System.Web.Providers, Version=2.0.0.0, Culture=neutral,
        PublicKeyToken=31bf3856ad364e35"
        connectionStringName="DefaultConnection" />
      </providers>
    </sessionState>
  </system.web>
  <system.webServer>
    <modules>
      <remove name="FormsAuthentication" />
    </modules>
    <httpErrors errorMode="Custom" />
  </system.webServer>
  <runtime>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>
        <assemblyIdentity name="Newtonsoft.Json" culture="neutral"
        publicKeyToken="30ad4fe6b2a6aeed" />
        <bindingRedirect oldVersion="0.0.0.0-6.0.0.0"
        newVersion="6.0.0.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="WebGrease" culture="neutral"
        publicKeyToken="31bf3856ad364e35" />
        <bindingRedirect oldVersion="0.0.0.0-1.5.2.14234"
        newVersion="1.5.2.14234" />
      </dependentAssembly>
    </assemblyBinding>
  </runtime>

```

```

        </dependentAssembly>
        <dependentAssembly>
            <assemblyIdentity name="EntityFramework"
publicKeyToken="b77a5c561934e089" />
            <bindingRedirect oldVersion="0.0.0.0-6.0.0.0"
newVersion="6.0.0.0" />
        </dependentAssembly>
        <dependentAssembly>
            <assemblyIdentity name="Microsoft.Owin" culture="neutral"
publicKeyToken="31bf3856ad364e35" />
            <bindingRedirect oldVersion="0.0.0.0-3.0.1.0"
newVersion="3.0.1.0" />
        </dependentAssembly>
        <dependentAssembly>
            <assemblyIdentity name="Microsoft.Owin.Security.OAuth"
culture="neutral" publicKeyToken="31bf3856ad364e35" />
            <bindingRedirect oldVersion="0.0.0.0-3.0.1.0"
newVersion="3.0.1.0" />
        </dependentAssembly>
        <dependentAssembly>
            <assemblyIdentity name="Microsoft.Owin.Security.Cookies"
culture="neutral" publicKeyToken="31bf3856ad364e35" />
            <bindingRedirect oldVersion="0.0.0.0-3.0.1.0"
newVersion="3.0.1.0" />
        </dependentAssembly>
        <dependentAssembly>
            <assemblyIdentity name="Microsoft.Owin.Security"
culture="neutral" publicKeyToken="31bf3856ad364e35" />
            <bindingRedirect oldVersion="0.0.0.0-3.0.1.0"
newVersion="3.0.1.0" />
        </dependentAssembly>
    </assemblyBinding>
</runtime>
<entityFramework>
    <defaultConnectionFactory
type="System.Data.Entity.Infrastructure.SqlConnectionFactory,
EntityFramework" />
    <providers>
        <provider invariantName="System.Data.SqlClient"
type="System.Data.Entity.SqlServer.SqlProviderServices,
EntityFramework.SqlServer" />
    </providers>
</entityFramework>
<system.codedom>
    <compilers>
        <compiler language="c#;cs;csharp" extension=".cs"
type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.CSharpCodePro
vider, Microsoft.CodeDom.Providers.DotNetCompilerPlatform,
Version=1.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"
warningLevel="4" compilerOptions="/langversion:6
/nowarn:1659;1699;1701" />
        <compiler language="vb;vbs;visualbasic;vbscript" extension=".vb"
type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.VBCodeProvide
r, Microsoft.CodeDom.Providers.DotNetCompilerPlatform,
Version=1.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"
warningLevel="4" compilerOptions="/langversion:14 /nowarn:41008
/define:_MYTYPE=&quot;Web&quot; /optionInfer+" />
    </compilers>
</system.codedom>
<log4net>
    <appender name="RollingFile"
type="log4net.Appender.RollingFileAppender">

```

```

        <file value="C:\WebForms\Log.txt" />
        <appendToFile value="true" />
        <maximumFileSize value="10000KB" />
        <maxSizeRollBackups value="10" />
        <layout type="log4net.Layout.PatternLayout">
            <conversionPattern value="%date %-5level %logger -
%message%newline" />
        </layout>
        </appender>
    <root>
        <level value="ALL" />
        <appender-ref ref="RollingFile" />
    </root>
</log4net>
</configuration>

```

4. Como parte final debemos de editar nuestro archivo Global.asax, que debe de contener las siguientes líneas:

```

using log4net;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading;
using System.Web;
using System.Web.Optimization;
using System.Web.Routing;
using System.Web.Security;
using System.Web.SessionState;

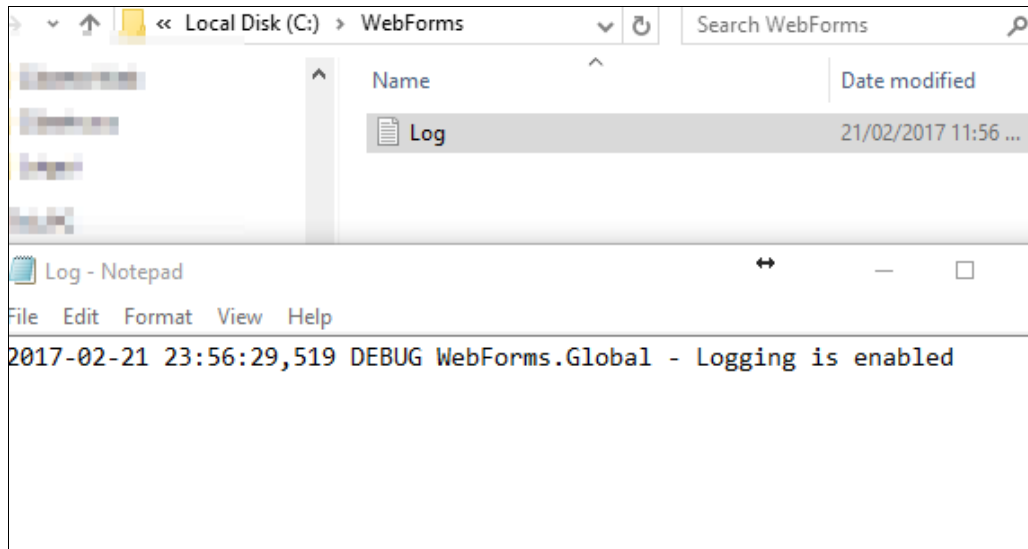
namespace WebForms
{
    public class Global : HttpApplication
    {
        private ILog _logger;
        void Application_Start(object sender, EventArgs e)
        {
            // Code that runs on application startup
            RouteConfig.RegisterRoutes(RouteTable.Routes);
            BundleConfig.RegisterBundles(BundleTable.Bundles);

            log4net.Config.XmlConfigurator.Configure();
            _logger = LogManager.GetLogger(typeof(Global));
            _logger.Debug("Logging is enabled");
        }

        protected void Application_Error(Object sender, EventArgs e)
        {
            Exception ex = Server.GetLastError();
            if (ex is ThreadAbortException)
                return;
            _logger = LogManager.GetLogger(typeof(Global));
            _logger.Error(ex);
            Response.Redirect("http://localhost/WebForms/Error.aspx");
        }
    }
}

```

5. Al ejecutar la aplicación se debe de generar un archivo con el nombre indicado en la configuración de log4net en el web.config: **C:\WebForms\Log.txt**



6. Si hacemos clic en nuestro error de prueba, debemos de visualizar en este fichero el error generado.

```
2017-02-22 00:01:08,729 DEBUG WebForms.Global - Logging is enabled
2017-02-22 00:01:17,342 ERROR WebForms.Global - System.Web.HttpUnhandledException (0x80004005): Exception of type 'System.Web.HttpUnhandledException' was thrown.
---> System.Exception: Este es un error de prueba
   at WebForms.ErrorTest.ErrorButton_Click(Object sender, EventArgs e) in D:\01 Materiales\Developer\NetDeveloper\WebForms\ErrorTest.aspx.cs:line 19
   at System.Web.UI.WebControls.Button.OnClick(EventArgs e)
   at System.Web.UI.WebControls.Button.RaisePostBackEvent(String eventArgument)
   at System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint)
   at System.Web.UI.Page.HandleError(Exception e)
   at System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint)
   at System.Web.UI.Page.ProcessRequest(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint)
   at System.Web.UI.Page.ProcessRequest(HttpContext context)
   at ASP.errorTest_aspx.ProcessRequest(HttpContext context) in C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Temporary ASP.NET Files\webforms\cfa51a23\30e5f84b\
App_Web_jvjegu2a.4.cs:line 0
   at System.Web.HttpApplication.CallHandlerExecutionStep.System.Web.HttpApplication.IExecutionStep.Execute()
   at System.Web.HttpApplication.ExecuteStep(IExecutionStep step, Boolean& completedSynchronously)
```

#### IV. EVALUACIÓN

1. ¿Por qué es recomendable mostrar una página de error en lugar de los detalles de la excepción de la aplicación?
2. ¿En qué momentos es recomendable manejar un error con try/catch en lugar de simplemente no manejar el error?
3. ¿Cuál es la relación/similitud que posee un bloque try/finally versus un bloque using?
4. ¿Qué herramientas conoces/usa para mantener un “log” de aplicación?