

TFC .NET

PROYECTO FINAL DE CARRERA
Manual Instalación.

HOTEL SPA



Raúl Taravillo Puras

Consultor: David Gañán Jiménez

Mayo-Junio 2010.

CONTENIDO

1. CONSIDERACIONES PREVIAS.....	3
2. MÓDULOS, LIBRERÍAS Y EJECUTABLES DE LAS APLICACIONES.....	5
3. INSTALACIÓN APLICACIÓN “eHotelSPANetServicio”	6
4. INSTALACIÓN APLICACIÓN “eHotelSPANetAdmin”	7
5. INSTALACIÓN EN EL SERVIDOR.....	8
CONFIGURACIÓN IIS.....	9
CONFIGURACIÓN SQL SERVER 2008.....	13

1. CONSIDERACIONES PREVIAS

Para el desarrollo del proyecto, se ha necesitado:

- Aplicación cliente, situado en cada habitación y en otros sitios del hotel. Aplicación para que el cliente del hotel interactúe.
- Aplicación cliente para los encargados del hotel, para que puedan gestionar las peticiones de sus clientes.
- Comunicación (Servicios Webs). La lógica del negocio estará en el servidor junto con la base de datos.

El software se desarrolló usando la plataforma de desarrollo de Microsoft .NET Framework 3.5, se han usado las siguientes tecnologías:

- WPF (Windows Presentation Foundation).
- WCF (Windows Communications Foundation).
- IIS (Internet Information Server). Servidor de páginas web avanzado de la plataforma Windows. Para nuestro propósito (WCF) se han utilizado los directorios virtuales.
- ADO.NET.
- LINQ, LINQ TO SQL.
- Entorno de desarrollo Visual Studio 2008 y Microsoft Expression Blend 2.
- Lenguaje de desarrollo C#.
- Base de datos Microsoft SQL Server 2008.
- El proyecto se ha desarrollado sobre el sistema operativo Windows XP Profesional SP3 con .NET Framework 3.5
- WPF Toolkit - February 2010 Release. (<http://wpf.codeplex.com/releases/view/40535>)

Es importante señalar que estas aplicaciones utilizan imágenes, dado que los diferentes módulos van a ser utilizados dentro de la misma red, se ha decidido no pasar las imágenes a través de WCF y utilizar algún directorio compartido por el servidor donde todos los terminales tengan acceso a él. De esta forma estamos liberando al servicio WCF de un paso masivo de datos que en este caso podemos ahorrarnos.

Junto con el código y este manual se adjunta un fichero script de la base de datos con inserts de algunas tablas para poder realizar el juego de pruebas.

También hay que tener en cuenta que el desarrollo del este software parte del hecho de que existen módulos para el mantenimiento de las tablas de clientes, usuarios, habitaciones y la relación de habitaciones y clientes. Por este motivo para poder probar el software se añaden inserts para estas tablas.

2. MÓDULOS, LIBRERÍAS Y EJECUTABLES DE LAS APLICACIONES

La aplicación recibe el nombre de **eHotelSPANet**.



El resultado final de la implementación han sido las siguientes librerías y aplicaciones:

- ✚ **eHotelSPANetAdmin.** Aplicación destinada a los trabajadores del hotel/Spa. Proporciona las pantallas necesarias para realizar el mantenimiento de:
 - Tipo de Servicios
 - Servicios
 - Calendarios
 - Horarios
 - Gestión sobre las peticiones de los clientes.

- ✚ **eHotelSPANetServicio.** Aplicación instalada en las habitaciones y otras instancias del hotel, y que será utilizada por los clientes para realizar las peticiones de los servicios ofertados por el hotel/Spa. Esta aplicación está especialmente diseñada para su uso táctil.

- ✚ **eHotelSPANetClass.** Librería que contiene las principales clases resultantes del estudio de las entidades.

- ✚ **eHotelSPANetComun.** Librería de soporte para los módulos de la aplicación, con funciones genéricas.

- ✚ **eHotelSPANetTeclado.** Librería que contiene un teclado táctil que se muestra por pantalla. Válido para introducir texto a través de pantallas táctiles.

- ✚ **WCFHotelSPANetAdmin.** Librería utilizada como servicios de WCF. Esta librería proporciona el acceso a los datos de la base de datos a la aplicación eHotelSPANetAdmin.

- ✚ **WCFHotelSPANetServicio.** Librería utilizada como servicios de WCF. Esta librería proporciona el acceso a los datos de la base de datos a la aplicación eHotelSPANetServicio.
- ✚ **eHotelSPANetDatos.** Librería que tiene el acceso a la base de datos, y es la encargada de gestionar las peticiones que se deben realizar sobre los datos de la base de datos.

3. INSTALACIÓN APLICACIÓN “EHOTELSPANETSERVICIO”

eHotelSPANetServicio Es la aplicación que utilizarán los clientes del Hotel/Spa.

Necesidades:

- NET Framework 3.5
- Librería WPF Toolkit - February 2010 Release (WPFToolkit.dll)
- Aplicación **eHotelSPANetServicio**
- Librería **eHotelSPANetClass**
- Librería **eHotelSPANetComun**
- Librería **eHotelSPANetTeclado**
- Acceso a recursos de red del servidor para la lectura de imágenes.

Fichero configuración “.config” de la aplicación

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <system.serviceModel>
    <bindings>
      <wsHttpBinding>
        <binding name="WSHttpBinding_IGestorServicios" closeTimeout="00:01:00"
          openTimeout="00:01:00" receiveTimeout="00:10:00" sendTimeout="00:01:00"
          bypassProxyOnLocal="false" transactionFlow="false"
          hostNameComparisonMode="StrongWildcard"
          maxBufferPoolSize="524288" maxReceivedMessageSize="2000000"
          messageEncoding="Text" textEncoding="utf-8" useDefaultWebProxy="true"
          allowCookies="false">
          <readerQuotas maxDepth="32" maxStringContentLength="8192"
            maxArrayLength="16384"
            maxBytesPerRead="4096" maxNameTableCharCount="16384" />
          <reliableSession ordered="true" inactivityTimeout="00:10:00"
            enabled="false" />
          <security mode="Message">
            <transport clientCredentialType="Windows" proxyCredentialType="None"
              realm="" />
            <message clientCredentialType="Windows"
              negotiateServiceCredential="true"
              algorithmSuite="Default" establishSecurityContext="true" />
          </security>
        </binding>
      </wsHttpBinding>
    </bindings>
    <client>
      <endpoint address="http://pccasita/WCFHotelSPANetServicio/GestorServicios.svc"
        binding="wsHttpBinding" bindingConfiguration="WSHttpBinding_IGestorServicios"
        contract="ServiceServicio.IGestorServicios"
        name="WSHttpBinding_IGestorServicios">
      </endpoint>
    </client>
  </system.serviceModel>
</configuration>
```

```
<identity>
  <dns value="localhost" />
</identity>
</endpoint>
</client>
</system.serviceModel>
</configuration>
```

Desde este fichero podemos modificar la configuración de la aplicación. Debemos destacar al menos tres apartados:

Establecemos una cantidad superior de los datos que podemos recibir.

```
maxReceivedMessageSize="2000000"
```

Indicamos la conexión con el servicio WCF.

```
<endpoint address="http://pccasita/WCFHotelSPANetServicio/GestorServicios.svc"
```

Establecemos como nos autenticaremos en el servidor para acceder a los servicios WCF.

```
<security mode="Message">
  <transport clientCredentialType="Windows" proxyCredentialType="None"
    realm="" />
  <message clientCredentialType="Windows" negotiateServiceCredential="true"
    algorithmSuite="Default" establishSecurityContext="true" />
</security>
```

4. INSTALACIÓN APLICACIÓN “EHOTELSPANETADMIN”

eHotelSPANetAdmin. Es la aplicación que utilizará el personal del Hotel/Spa.

Necesidades:

- NET Framework 3.5
- Librería WPF Toolkit - February 2010 Release (WPFToolkit.dll)
- Aplicación **eHotelSPANetAdmin**
- Librería **eHotelSPANetClass**
- Librería **eHotelSPANetComun**
- Acceso a recursos de red del servidor para la lectura y escritura de imágenes.

Fichero configuración “.config” de la aplicación

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <system.serviceModel>
    <bindings>
      <wsHttpBinding>
        <binding name="WSHttpBinding_IGestorAdministrativo" closeTimeout="00:01:00"
          openTimeout="00:01:00" receiveTimeout="00:10:00" sendTimeout="00:01:00"
          bypassProxyOnLocal="false" transactionFlow="false"
          hostNameComparisonMode="StrongWildcard"
          maxBufferPoolSize="524288" maxReceivedMessageSize="2000000"
          messageEncoding="Text" textEncoding="utf-8" useDefaultWebProxy="true"
          allowCookies="false">
          <readerQuotas maxDepth="32" maxStringContentLength="8192"
            maxArrayLength="16384"
            maxBytesPerRead="4096" maxNameTableCharCount="16384" />
          <reliableSession ordered="true" inactivityTimeout="00:10:00"
            enabled="false" />
        </binding>
      </wsHttpBinding>
    </bindings>
  </system.serviceModel>
</configuration>
```

```
<security mode="Message">
  <transport clientCredentialType="Windows" proxyCredentialType="None"
    realm="" />
  <message clientCredentialType="Windows"
    negotiateServiceCredential="true"
    algorithmSuite="Default" establishSecurityContext="true" />
</security>
</binding>
</wsHttpBinding>
</bindings>
<client>
  <endpoint address="http://pccasita/WCFHotelSPANetAdmin/GestorAdministrativo.svc"
    binding="wsHttpBinding"
    bindingConfiguration="WSHttpBinding_IGestorAdministrativo"
    contract="ServiceAdmin.IGestorAdministrativo"
    name="WSHttpBinding_IGestorAdministrativo">
    <identity>
      <dns value="localhost" />
    </identity>
  </endpoint>
</client>
</system.serviceModel>
</configuration>
```

Desde este fichero podemos modificar la configuración de la aplicación. Debemos destacar al menos tres apartados:

Establecemos una cantidad superior de los datos que podemos recibir.

```
maxReceivedMessageSize="2000000"
```

Indicamos la conexión con el servicio WCF.

```
<endpoint address="http://pccasita/WCFHotelSPANetAdmin/GestorAdministrativo.svc"
```

Establecemos como nos autenticaremos en el servidor para acceder a los servicios WCF.

```
<security mode="Message">
  <transport clientCredentialType="Windows" proxyCredentialType="None"
    realm="" />
  <message clientCredentialType="Windows" negotiateServiceCredential="true"
    algorithmSuite="Default" establishSecurityContext="true" />
</security>
```

5. INSTALACIÓN EN EL SERVIDOR

El servidor es el encargado de recibir y de proporcionar los datos que piden las aplicaciones “eHotelSPANetAdmin” y “eHotelSPANetServicio” a través de dos servicios.

- ✓ **WCFHotelSPANetAdmin.** Librería utilizada como servicios de WCF. Esta librería proporciona el acceso a los datos de la base de datos a la aplicación eHotelSPANetAdmin.

- ✓ **WCFHotelSPANetServicio**. Librería utilizada como servicios de WCF. Esta librería proporciona el acceso a los datos de la base de datos a la aplicación eHotelSPANetServicio.

Necesidades:

- NET Framework 3.5
- Librería WPF Toolkit - February 2010 Release (WPFToolkit.dll)
- Librería **WCFHotelSPANetAdmin**
- Librería **WCFHotelSPANetServicio**
- Librería **eHotelSPANetClass**
- Librería **eHotelSPANetDatos**
- Recursos de red del servidor para la lectura y escritura de imágenes.
- IIS. Configuración de Internet Information Server para los dos servicios.
- SQL Server 2008.
- Base de datos eHotelSPA (El script de generación se proporciona en un fichero adjunto).

CONFIGURACIÓN IIS

Para ambos servicios la configuración es idéntica.

Fichero de configuración "Web.Config"
Mostramos los datos más significativos:

```
<system.serviceModel>
  <services>
    <service behaviorConfiguration="WCFHotelSPANetAdmin.Service1Behavior"
      name="WCFHotelSPANetAdmin.GestorAdministrativo">
      <endpoint address="" binding="wsHttpBinding"
        contract="WCFHotelSPANetAdmin.IGestorAdministrativo"
        bindingConfiguration="wsHttpBinding_IServicio1">
        <identity>
          <dns value="localhost" />
        </identity>
      </endpoint>
      <endpoint address="mex" binding="mexHttpBinding" contract="IMetadataExchange"/>
    </service>
  </services>

  <bindings>
    <wsHttpBinding>
      <binding name="wsHttpBinding_IServicio1"
        maxReceivedMessageSize="2000000"/>
    </wsHttpBinding>
  </bindings>
</system.serviceModel>
```

Establecemos una cantidad superior de los datos que podemos recibir.

```
maxReceivedMessageSize="2000000"
```

Autenticación utilizado ASP.NET para identificar a un usuario entrante.

```
<authentication mode="Windows" />
```

El fichero de configuración “.svc” ya nos lo crea el propio Visual Studio. Este fichero tiene que está en la raíz del directorio del servicio

GestorAdministrativo.svc:

```
<%@ ServiceHost Language="C#" Debug="true"  
Service="WCFHotelSPANetAdmin.GestorAdministrativo"  
CodeBehind="GestorAdministrativo.svc.cs" %>
```

GestorServicios.svc:

```
<%@ ServiceHost Language="C#" Debug="true"  
Service="WCFHotelSPANetServicio.GestorServicios"  
CodeBehind="GestorServicios.svc.cs" %>
```

Para cada servicio tenemos que tener un directorio “bin” dentro de la carpeta del servicio con las librerías necesarias compiladas.

... \WCFHotelSPANetAdmin\bin

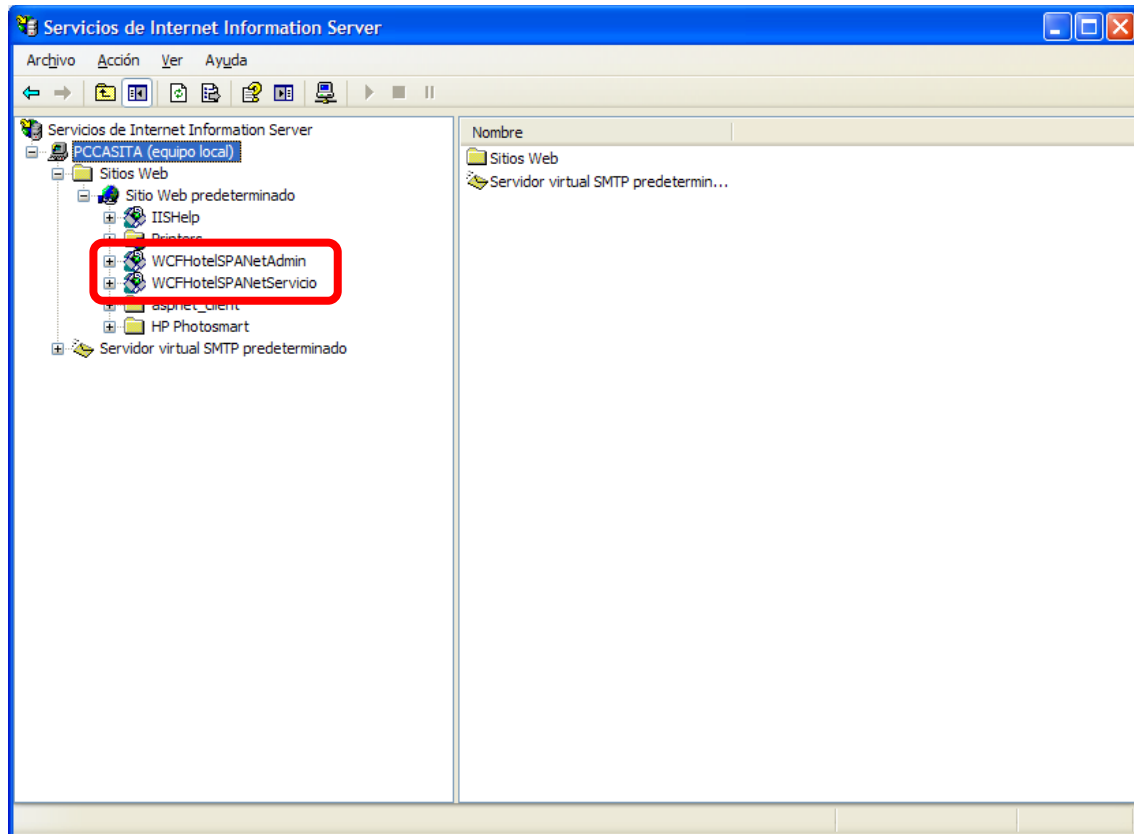
- **WCFHotelSPANetAdmin.dll**
- eHotelSPANetClass.dll
- eHotelSPANetDatos.dll

... \WCFHotelSPANetServicio\bin

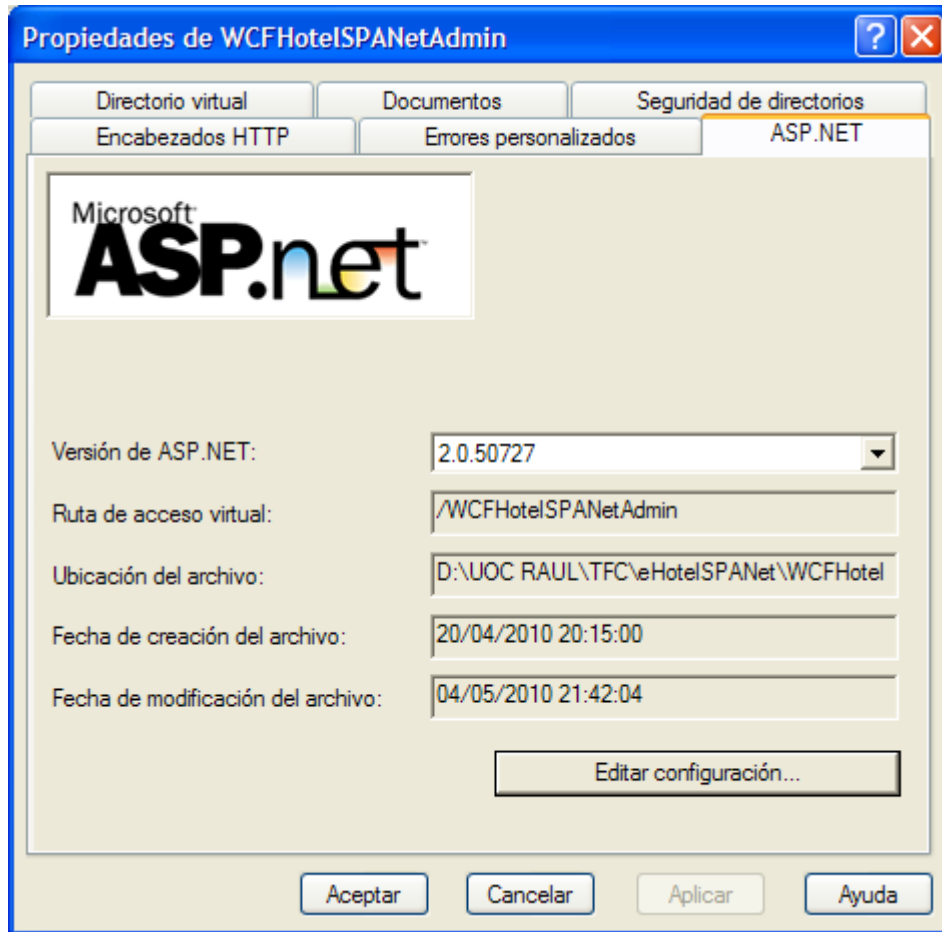
- **WCFHotelSPANetServicio.dll**
- eHotelSPANetClass.dll
- eHotelSPANetDatos.dll

Nos queda crear un directorio virtual en IIS para cada uno de nuestros servicios.

En este caso estamos utilizando IIS v5.1



Hay que señalar que es importante configurar el apartado ASP.NET:



CONFIGURACIÓN SQL SERVER 2008

Por último nos queda configurar el gestor de bases de datos SQL Server.
El script de generación de la base de datos se adjunta en otro fichero.

Debemos otorgar permisos al usuario ASPNET ya que este usuario es con el que se realizarán las conexiones remotas.

