

# **Internet Information Server y ASP.NET**

# Indice

<b>Nº-1 Internet Information Server, ASP .NET introducción e instalación.....</b>	<b>1</b>
1. Introducción.....	1
La evolución.....	2
1.1 ¿Qué es el .NET Framework?.....	3
2. Una definición mas profunda de .NET.....	4
2.1 Runtime del lenguaje común.....	5
2.2. Biblioteca de clases.....	6
2.3. Capa de datos.....	6
2.4. Capa de formularios y ASP.....	6
2.5. Capas de CLS y lenguajes.....	7
3. Instalación de ASP.NET.....	7
3.1. ¿Qué es un servidor web?.....	7
3.2. Instalación de IIS.....	15
3.3. Comentarios:.....	21
4. Primeros pasos con IIS.....	21
5. La consola Administrativa de IIS.....	24
6. Propiedades del sitio web.....	33
7. Los directorios virtuales.....	34
7.1 ¿Qué es un directorio virtual?.....	35
7.2 Crear directorios virtuales.....	36
<b>Ejercicios.....</b>	<b>43</b>
Ejercicio 1.....	43
Ejercicio 2.....	43
Ejercicio 3.....	44

# Nº-1 Internet Information Server, ASP .NET introducción e instalación

## 1. Introducción:

ASP.NET no necesita presentación, solo el tiempo le ha hecho situarse en el sitio que se merece. Hace bastantes años Microsoft presentaba el lenguaje ASP. Un lenguaje muy sencillo de utilizar porque se basa en la sencillez de las instrucciones de Visual Basic (rey por entonces de la programación) que permitía realizar páginas dinámicas de una forma sencilla y rápida.

En el otro lado estaba Linux con su PHP que, al ser gratuito, tuvo muchos adeptos, pero su supremacía ya no es tal. En parte es por la excesiva sobrevaloración que se le concede a Linux por el hecho de ser gratuito, pero sigamos con nuestro ASP...

Después de varias evoluciones del lenguaje ASP llegó la revolución con el desarrollo de la plataforma .NET de Microsoft. El modelo de objetos y el entorno de desarrollo introducido por esta tecnología es impresionante. Su potencia y su facilidad está a mucha distancia de PHP y de los componentes Java. Aunque la primera versión era laborioso y un poco confusa, 1.1, las dos últimas versiones la 2.0 y la que trataremos en este curso, la 3.5 son realmente sencillas y sólidas teniendo un grado de coherencia y rendimiento muy elevados.

Para los que sepan un poco de programación avanzar que con ASP.NET podremos:

- Crear páginas web basadas en código compilado
- Utilizar los nuevos lenguajes orientados a objeto (.NET)
- Utilizar variables reales (antes con ASP no existían)
- Crear componentes de servidor utilizando lenguaje .NET
- Utilizar validadores para probar el sistema cliente-servidor.

¿A quién va dirigido este curso? Con este curso pretendemos llegar a dos grupos de usuarios. Por una parte los que nunca han programado en Intranet y quieren aprender este mundillo, y por otro los que ya conocen ASP en sus anteriores versiones y quieren "saltar" al nuevo sistema .NET.

Hay ventajas y desventajas para estos dos grupos. Para los que se quieren iniciar van a aprender una filosofía y técnicas de programación muy ambiciosas y robustas que componen el mejor entorno de desarrollo que existe actualmente. Por contra necesitarán un aprendizaje para entender todas las técnicas en las que se basa .NET para desarrollar aplicaciones, ahí es donde este curso entra en acción: proporcionar un buen nivel a los nuevos programadores para que comiencen a desarrollar en esta plataforma. Para el segundo grupo, como ventaja obvia está el conocimiento de un lenguaje de programación tan extendido en WEB como es ASP. La desventaja es que hay que borrar los conocimientos anteriores y partir de cero, es decir, hay que conocer esta nueva tecnología profundamente para entenderla y trabajar con la nueva filosofía de .NET. (Yo acabo de terminar esta fase)

Eso si, os aviso que varias partes de este curso necesariamente serán técnicas, no es un curso tipo DreamWeber o Flash con cosas de diseño o con resultados rápidos, aquí tendremos que hacer un buen aprendizaje de .NET. Eso si, una vez superado tendréis a vuestra disposición la plataforma mas potente existente para crear páginas web dinámicas. Hace ya tiempo que superó a los servidores java y el mundo de Linux que con su lenguaje PHP se queda demasiado corto.

## Internet Information Server y ASP.NET

En este curso nos centraremos únicamente en ASP.NET 3.5. Esta es la última versión y tiene mejoras sustanciales que hace que deba ser la versión que instalemos obligatoriamente en nuestro servidor Web. Aunque si todavía estás en la versión 2.0 no te preocupes, prácticamente todo lo que vamos a ver funcionará correctamente en esa versión.

Es una versión muy moderna y a la fecha de escribir este manual todavía no ha salido en castellano así que no nos queda mas remedio que empezar con ella en inglés, seguro que en breve podremos instalar la versión en castellano. De todas formas iré mirando a ver si llegamos a tiempo por si apareciese en castellano.

### La evolución

Hace varios años (como diez mas o menos) las aplicaciones se creaban utilizando un mismo lenguaje para todas las tareas y para un sistema operativo concreto. Inicialmente las aplicaciones para Windows se realizaban en C y con llamadas directas a las "tripas" del Windows con la API (¡¡que tiempos mas duros!!) El trabajo era enorme por dos cosas: la complejidad del lenguaje y la necesidad de conocer todos los detalles del sistema operativo para poder programar sobre él.

Ahí nació Visual Basic, como una herramienta que permitía el desarrollo de las interfaces de una forma tremendamente sencilla para lo conocido hasta entonces y con un lenguaje tan popular por entonces como el Basic. Se utilizaban los controles en formato VBX para diseñar las ventanas y luego un mecanismo de métodos y eventos para controlar el código.

La parte profesional se "pasó" al C++, un lenguaje orientado a objetos muy complejo y difícil de mantener. Por otro lado Visual Basic seguía avanzando con los COM (Component Object Model): un modelo de objetos que permitía la reutilización de componentes independiente del lenguaje con el que estuviesen escritos, así transcurrieron las versiones 5 y 6 de Visual Basic que se convirtieron en los entornos mas importantes del panorama Windows.

En las páginas ASP (la tecnología anterior a ASP.NET) utilizadas para la programación de páginas Web dinámicas sucedía lo mismo, estaban basadas en un lenguaje muy limitado y sujeto a la plataforma, además teníamos que acudir enseguida a productos de terceros para ampliar las funcionalidades que no encontrábamos. Aun con todo ASP se desplegó en muy poco tiempo gracias a su lenguaje fácil y con un corto tiempo de aprendizaje como era el VBScript.

### La solución

Con el tiempo además de los COM aparecieron otras técnicas complementarias para ampliar los lenguajes existentes. Pero esta heterogeneidad de tecnologías, muchas veces dictadas por las necesidades tecnológicas del momento, como Internet, hicieron que se produjeran multitud de servicios duplicados, creación de servicios exclusivamente para algunos lenguajes, poca reutilización de código, mas complejidad, ...

La solución definitiva se ha planteado con .NET que consta de una serie de servicios iguales en todos los lenguajes que mantienen la integridad con los desarrollos existentes y hace posible una interoperatividad entre los lenguajes desconocida hasta el momento. Esto es, podemos utilizar varios lenguajes diferentes (los permitidos .NET) y todos tendrán disponibles desde el mismo entorno de desarrollo hasta los controles y componentes de programación. Se acabó que Visual C tenga estos componentes, Visual FoxPro otros o Visual Basic otros... todos parten de la tecnología .NET y comparten por lo tanto los mismos componentes y objetos. Podemos incluso escribir partes distintas de un mismo programa con varios lenguajes .NET.

### ASP.NET y el entorno .NET Framework

Antes de continuar debemos saber un poco de donde sale toda esta tecnología, así que prepárate para leer unos conceptos un poco raros pero que nos ayudarán a situar el lenguaje .NET. Recuerda que ASP.NET es una parte del mundo de desarrollo .NET de Microsoft, así que necesitaremos tener una idea global de .NET para acercarnos, como he comentado, a un concepto mas exacto de este lenguaje.

Allá vamos, pero no te asustes con toda esta teoría, sólo es para situarnos luego cuando empecemos a trabajar irás comprendiendo todos estos conceptos.

ASP.NET usa una jerarquía de clases (los elementos de programación que nos proporciona) que están incluidas en el .NET Framework, por tanto conocer el .NET Framework nos ayudará a conocer al propio ASP.NET, aunque también necesitarás conocer la forma de usar y de hacer del ASP.NET ya que, aunque en el fondo sea lo mismo, el aspecto sintáctico es diferente para cada uno de los lenguajes basados en .NET Framework.

**Importante** Luego... el verdadero núcleo de todo es .NET Framework, un entorno con multitud de clases y objetos disponibles para trabajar con ellos. Sobre este núcleo tendremos los lenguajes de programación y uno de ellos es ASP.NET. Por eso es muy importante conocer que es el .NET Framework

Así que tranquilo primero veamos que eso del .Net Framework que básicamente se puede decir que es el hermano mayor de .NET

**Nota** Luego una vez conocido el .NET Framework podremos crear las páginas web con el lenguaje que queramos, nosotros elegiremos Visual Basic.

## 1.1 ¿Qué es el .NET Framework?

*Literalmente .Net Framework es:*

*" .NET Framework es un entorno para construir, instalar y ejecutar servicios Web y otras aplicaciones.  
Se compone de tres partes principales: el Common Language Runtime, las clases Framework y ASP.NET"*

*"El .NET Framework es un entorno multi-lenguaje para la construcción, distribución y ejecución de Servicios Webs y aplicaciones."*

*"El .NET Framework es una nueva plataforma diseñada para simplificar el desarrollo de aplicaciones en el entorno distribuido de Internet."*

Luego .NET es una nueva filosofía en cuanto a entornos de desarrollo. Este entorno puede utilizar varios lenguajes diferentes para escribir programas, todos se apoyarán en las herramientas que ofrece para conseguir aplicaciones .NET y el alma de este entorno es el .NET Framework

.Net Framework permite el desarrollo de aplicaciones a través del uso de un conjunto de herramientas y servicios que proporciona y que se agrupan en tres bloques:

- El entorno de ejecución común ó Common Language Runtime (CLR)
- Jerarquía de clases de .NET
- Motor de generación de formularios, para crear nuestras aplicaciones

El Common Language Runtime (CLR) es una serie de librerías dinámicas (DLLs), también llamadas assemblies (o ensamblados), que hacen las veces de las DLLs del API de Windows así como las librerías runtime de Visual Basic o C++.

Como seguramente sabrás, cualquier programa Windows depende de una forma u otra de una serie de librerías, ya sea en tiempo de ejecución como a la hora de la compilación (las famosas DLL's). Pues el CLR es eso, una serie de librerías usadas en tiempo de

## Internet Information Server y ASP.NET

ejecución para que nuestros ejecutables o cualquiera basado en .NET puedan funcionar. Se acabó eso de que existan dos tipos de ejecutables: los que son autosuficientes y no dependen de librerías externas o los que necesitan de librerías en tiempo de ejecución para poder funcionar, tal es el caso de las versiones anteriores de Visual Basic. (esto lo entenderán mejor los antiguos programadores de VB)

**Por otro lado, la biblioteca de clases de .NET Framework proporcionan una jerarquía de clases orientadas a objeto disponibles para cualquiera de los lenguajes basados en .NET, incluido el Visual Basic.** Esto quiere decir que a partir de ahora Visual Basic ya no será la "oveja negra" de los lenguajes de programación, sino que tendrá a su disposición todas las clases disponibles para el resto de los lenguajes basados en .NET, (o casi), con lo cual sólo nos diferenciará del resto de programadores en la forma de hacer las cosas: ¡más fáciles!

Los poco flexibles modelos de programación actuales impiden al programador contemplar la idea de un proyecto para Internet que le permita crear aplicaciones distribuidas mas potentes. Estos sistemas utilizaban los tecnologías COM, ASP, ADO, ... que individualmente son buenas pero que no están pensadas para ser ejecutadas en la red. Estos elementos definían la arquitectura Windows DNA (Distributed Internet Architecture) que hasta ahora era el modelo de programación para Internet. Esta tecnología queda ahora desplazada completamente por la arquitectura .NET

Microsoft .NET ofrece una visión orientada a objeto del sistema operativo Windows e incluye cientos de clases que encapsulan los objetos mas importantes del núcleo de Windows. Mas adelante veremos y definiremos las clases pero de una forma resumida digamos que son plantillas para crear objetos, por ejemplo: la clase "botón" me permite crear botones de comando y trabajar con ellos.

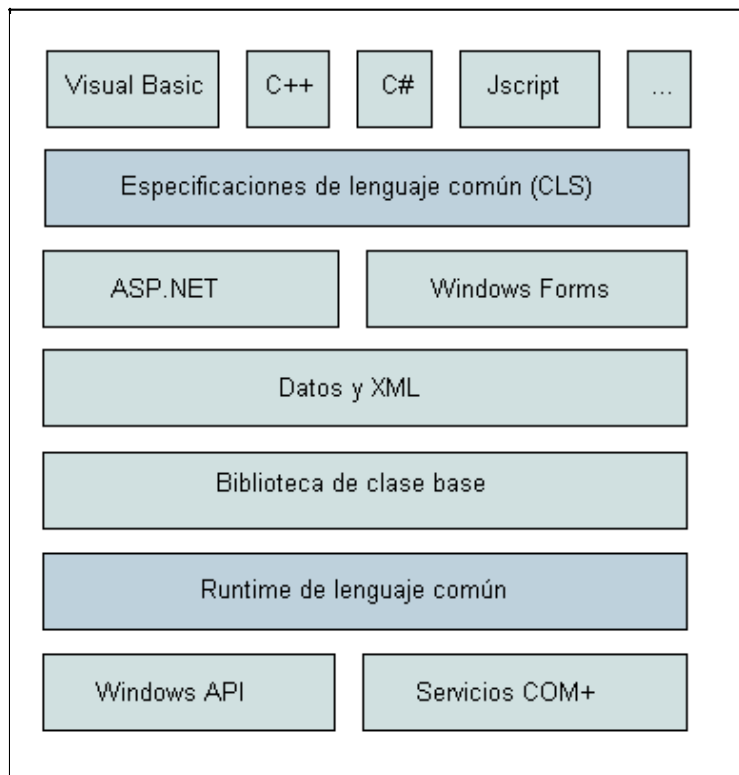
En este curso nos centraremos en la parte de ASP.NET pero no debemos olvidar que .NET es mas que este lenguaje, por ejemplo VB.NET es una potente plataforma de desarrollo de aplicaciones, permite crear potentes programas Windows. Lo bueno es que al terminar este curso tendremos una gran punto de partida para seguir con esta tecnología de .NET.

Espero no haberte liado mucho, mas que nada que esto no es la presentación de un simple lenguaje de programación, es la presentación de cómo funciona nuestro Windows y como lo hará en el futuro apoyado en esta tecnología.

## 2. Una definición mas profunda de .NET

La mejor forma de comprender cómo funciona .NET es analizar las numerosas capas en las que se divide .NET Framework. Veamos la siguiente figura:

## Internet Information Server y ASP.NET



(no te asustes con lo que vas a leer aquí, es obligatorio que lo veamos por encima, al terminar el curso vuelve a leer esto y veras que sencillo era...)

Como hemos comentado antes .NET Framework constituye la base sobre la que se asienta .NET. Para nosotros es la pieza fundamental de esta nueva tecnología y es la que nos va a proporcionar las herramientas y servicios que necesitaremos en nuestros desarrollos. Podemos agrupar en tres bloques el conjunto de herramientas y servicios:

- El runtime de lenguaje común o entorno de ejecución común (CLR)
- Biblioteca de clases base de la plataforma .NET (.NET Framework Base Classes)
- Motor de generación de la interfaz para crear formularios e interfaces de usuario.

Estas son tres de las capas del esquema anterior. Simplemente esa jerarquía me dice que en el nivel mas alto están los lenguajes que vamos a utilizar en nuestro desarrollo con .NET que pueden ser varios (en nuestro caso ASP.NET). Luego traduce esas instrucciones al estándar .NET para poder trabajar con esas instrucciones de una forma independiente al lenguaje con que se escribieron. El siguiente paso es la generación de formularios para Windows o web (ASP.NET). Luego el enlace con bases de datos si las hay. Por fin llegamos a todos los objetos disponibles y por último el runtime de lenguaje común (CLR) que es el que va a ejecutar la aplicación. Veamos ahora más técnicamente de abajo hacia arriba:

### 2.1 Runtime del lenguaje común

El runtime del lenguaje común es la primera capa que pertenece a .NET Framework. Esta capa es la responsable de los servicios básicos de .NET, tales como la administración de memoria, la recolección de los elementos no utilizados, el control estructurado de excepciones y del subprocesamiento múltiple. Si .NET se transporta a otras arquitecturas que no estén basadas en Windows el primer paso sería escribir un runtime del lenguaje para el nuevo equipo. El CLR tiene estas características:

- Proporciona mejoras para el programador que antes tenía que elaborar
- Administra el código en tiempo de ejecución: carga en memoria, liberación de memoria, ...
- Gestiona la seguridad del código ejecutado
- Abre posibilidades a otros fabricantes para incorporar sus lenguajes
- Facilita la distribución e instalación de aplicaciones. Elimina los temibles conflictos de DLL's y versiones de ellas

Es la interfaz entre nuestro código y el sistema operativo, de ahí que si mañana Microsoft hace un CLR en lugar de para Windows, para Macintosh se puedan ejecutar nuestras aplicaciones .NET en este sistema.

## 2.2. Biblioteca de clases

La biblioteca de clases base (BCL) es la parte de .NET Framework que define todos los tipos de datos básicos, tales como System.Object (raíz de la jerarquía de objetos .NET), tipos numéricos y de fechas, tipo string, matrices y colecciones. La BCL contiene también clases que administrarán las características centrales de .NET: entrada/salida de archivos, subprocesamiento, serialización y seguridad. La forma en la que los tipos se implementan en la BCL siguen las especificaciones llamadas **Common Type System** (CTS). Por ejemplo, estas especificaciones dicta la forma en la que un tipo .NET expone campos, propiedades, métodos y sucesos, también cómo un tipo puede heredar otro tipo...

Tranquilo... a lo largo del curso veremos estas clases y por fuerza trabajaremos con ellas porque todo en .NET son clases: desde un gráfico, un botón o nuestro programa.

## 2.3. Capa de datos

La capa de datos y XML contienen las clases .NET que trabajan con bases de datos y con XML. Anteriormente el soporte XML era una compatibilidad proporcionada por un componente externo. en .NET vemos cómo está integrado en su mismo núcleo. Podemos decir que XML es el formato que utiliza .NET para almacenar cualquier tipo de información. La parte de datos es la que se conoce como ADO.NET y es el equivalente en .NET a la tecnología ActiveX Data Object (ADO), ampliamente conocida por los programadores de Visual Basic, Intranets, ...

**Nota para los conocedores de ADO:** Aunque su nombre es muy parecido a ADO.NET son muy diferentes. ADO.NET cubre prácticamente todas las técnicas de bases de datos disponibles (incluyendo cursores del lado del cliente y del servidor, conjuntos de datos desconectados y actualizaciones en diferido). ADO.NET está centrado principalmente en los conjuntos de resultados desconectados (llamados DataSets) Este objeto es mucho mas potente que el objeto ADO Recordset y puede almacenar datos provenientes de varias tablas, relaciones entre diferentes tablas de datos y puede importar y exportar datos en XML.

## 2.4. Capa de formularios y ASP

Las dos capas siguientes son ASP.NET y Windows Forms. Aquí se sitúan todas las clases que podremos utilizar para generar las páginas web en el primer caso y las ventanas estándares o formularios en las aplicaciones de Windows en el segundo caso.

Aunque estén en el mismo nivel las tecnologías son muy distintas. **Web Forms** se ejecuta en el servidor y produce HTML y es la base de las conocidas Intranets donde estas páginas devuelven otras páginas web con conjuntos de resultados u otros datos. **Windows Forms** se ejecuta en el cliente, un equipo Windows.



## Internet Information Server y ASP.NET

Esta es la parte con la que nos quedaremos con ASP.NET en nuestra plataforma de Teleformación dispones del curso VB.NET que trata sólo de Visual Basic.NET como lenguaje para crear aplicaciones. (curso totalmente recomendable y que para suerte o desgracia tuya también lo he desarrollado yo)

La biblioteca de clases de .NET Framework está formada por una colección de ensamblados (o Assembly), cada uno de los cuales comprende una o mas DLL. Veamos que esto de un "ensamblado"...

Un ensamblado consiste en un conjunto de tipos y recursos reunidos para formar la unidad mas elemental de código que puede ejecutar el entorno .NET Framework. Estos ensamblados son los elementos con los que construiremos una aplicación: una aplicación .NET se compone de varios ensamblados. Podemos llamar también a un ensamblado como una DLL lógica (recordamos que una DLL es una biblioteca de enlaces dinámicos). Decimos lógica porque se trata de una lista de ficheros que se referencian en tiempo de ejecución pero que no se compilan para producir un fichero físico, a diferencia de las DLL's tradicionales. Un ensamblado va mucho mas allá ya que puede contener otros recursos además de clases como imágenes, ...

## 2.5. Capas de CLS y lenguajes

Donde están las especificaciones de los lenguajes y su sincronización con el entorno .NET

Después de esta locura de siglas y tecnologías vamos al grano con la instalación y luego realizaremos algún ejemplo sencillo...

## 3. Instalación de ASP.NET

Ya sabemos las tecnología que vamos a utilizar para programar, ahora vamos a conocer qué es un servidor Web y cómo se instala ya que es el que nos va a proporcionar lo necesario para que sirva a los usuarios las páginas web que creamos. Primero vamos a conocer que es un servidor web.

### 3.1. ¿Qué es un servidor web?

Un servidor web es un programa que se ejecuta en un equipo, normalmente un servidor y que proporciona páginas web a los "clientes" que le piden. Los clientes son los navegadores web como Internet Explorer o FireFox

Cuando llamamos a un equipo "servidor" nos referimos a un equipo que va a proporcionar determinados servicios a los usuarios de una red local o de Internet. Lleva un equipamiento de prestaciones mas avanzadas que un equipo de sobremesa y sobre todo incorpora un sistema operativo de servidor (Windows 2000 Server ó Windows 2003 Server).

Los requisitos necesarios son para la parte del servidor un Windows 2000/2003 Server. Y para el equipo cliente, con el que trabajaremos habitualmente utilizaremos un 2000 ó XP professional.

En nuestro equipo "cliente" o en el que vamos a trabajar sólo nos queda utilizar un buen editor de páginas web o editor de sitios web. Front Page 2000/XP es el editor que mejor sincroniza con nuestro IIS para sitios web de páginas HTML normales, por lo tanto debemos ver como se conecta FrontPage con este servidor de páginas Web. Es un producto que como opción forma parte de la suite Office de Microsoft. Finalmente un accesorio que incorpora IIS va a conectar nuestro Front Page con el servidor web IIS. Este accesorio se llama

## Internet Information Server y ASP.NET

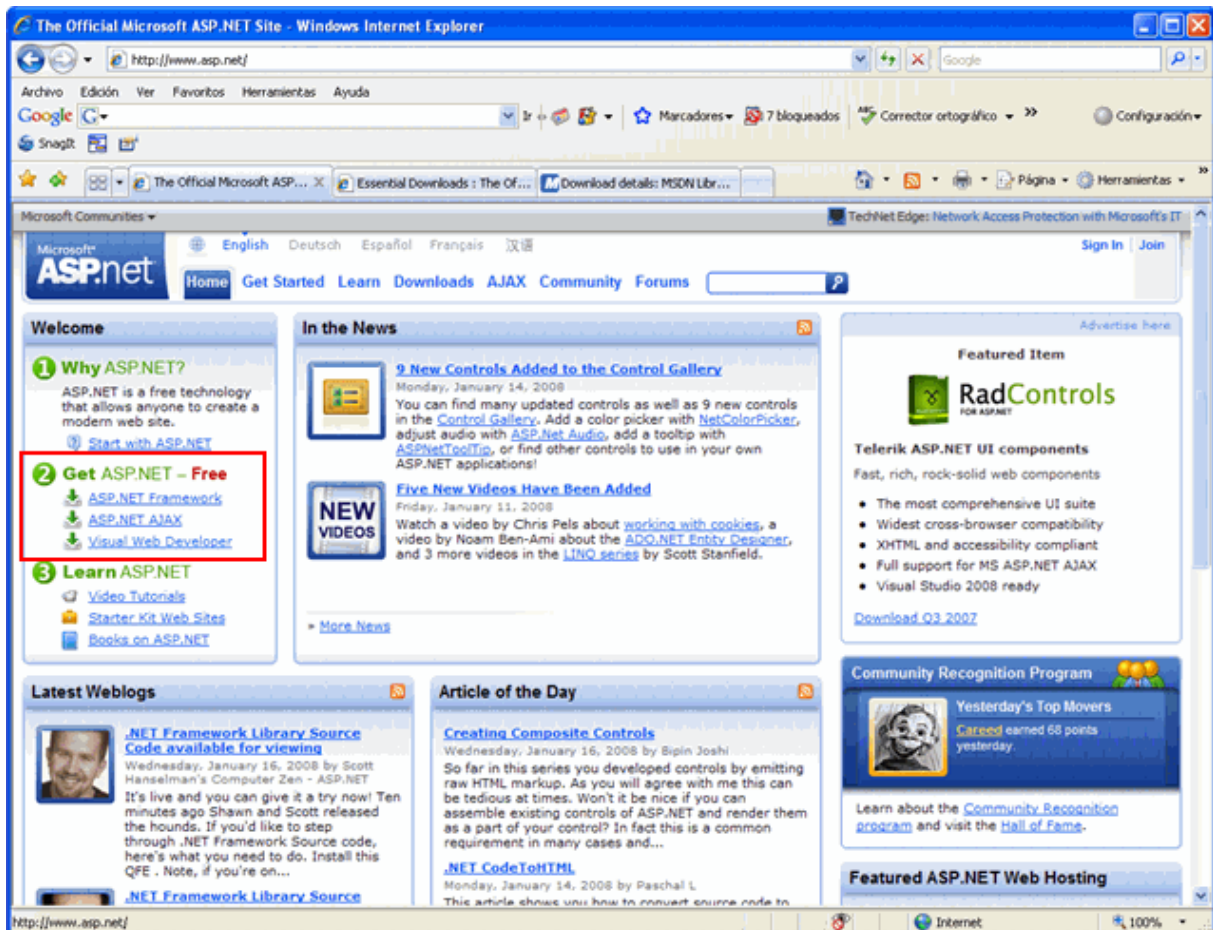
"Extensiones de servidor de Front Page" que veremos posteriormente.

Todo esto es válido para trabajar con las antiguas páginas ASP. Para que esta plataforma trabaje correctamente con ASP.NET necesitamos que el servidor lo incorpore. Actualmente Windows 2000 Server no incorpora soporte para la tecnología ASP y Windows 2003 Server lo hace pero para la versión 1.1. Así que tenemos que descargar el ASP.NET 3.5 e instalarlo en nuestro servidor. Suponemos que el inminente Windows Server 2008 si incorporará "de serie" todo lo relativo a la última versión 3.5

Antes con la primera versión de ASP.NET no había un entorno de desarrollo sencillo, había que construir las páginas "a mano" con FrontPage o con el Visual Studio, entorno muy complejo por entonces. Ahora, al darse cuenta Microsoft (no ahora sino hace unos años) de la importancia del ASP.NET sobre todo a partir de la versión 2.0 desarrolló un entorno de desarrollo llamada "Visual Web Developer" que es el que utilizaremos para la realización de las páginas web. Este mismo curso que antes era de las otras versiones hacía las páginas con FrontPage porque no había otra forma fácil de realizarlas, este año estamos de suerte y desde el principio trabajaremos con el nuevo entorno, que nos proporcionará muchas facilidades en nuestro desarrollo.

ASP.NET Se puede descargar de varios sitios desde el sitio de Microsoft. Pero hay un sitio oficial para ASP.NET que será nuestra base para varias cosas que veremos en este curso. Este es el portal donde vas a encontrar toda las novedades sobre este entorno de programación. Lo malo es que está en inglés, pero bueno, ya sabes en el mundo de la programación es bastante normal.

El sitio web es este: [www.asp.net](http://www.asp.net). En la parte izquierda tenemos el enlace para descargar nuestro ASP.NET 3.5

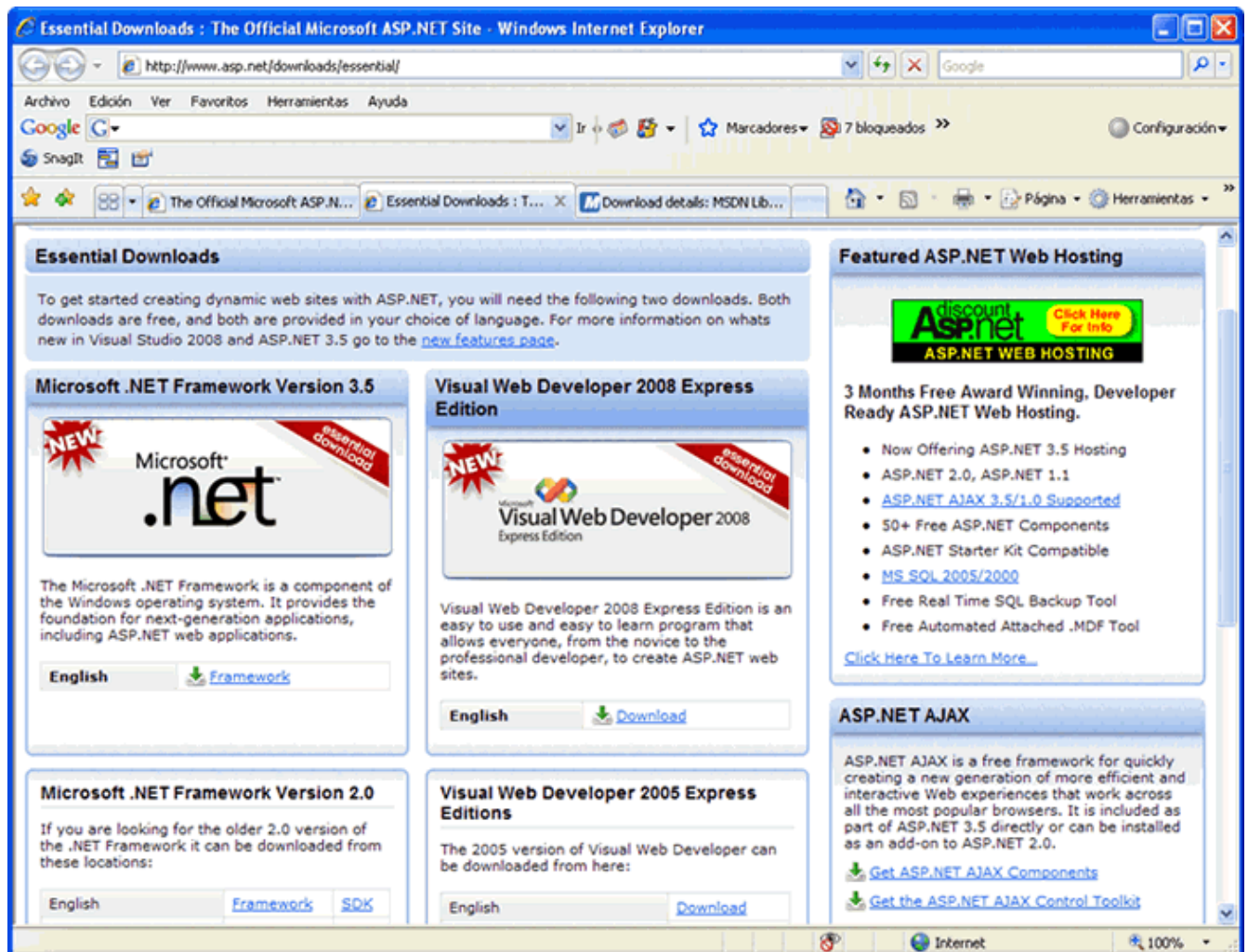


## Internet Information Server y ASP.NET

Antes sólo podíamos descargar lo que es el núcleo del .net framework y teníamos que utilizar cualquier editor, como FrontPage para poder editar las páginas. Por suerte apareció al tiempo un estupendo editor de aspecto muy parecido a los IDE avanzados. Los IDE son los entornos de desarrollo integrados que permiten tener una ayuda inmediata de todos los objetos, su sintaxis y los archivos del proyecto. Se llamaba Web Matrix, era un poco rudimentario pero una gran ayuda para escribir código.

En la versión 2.0 apareció el primer entorno de verdad, que se llamaba Visual Web Developer, que era parte del Visual Studio 2.005. Desde hace muy pocas semanas tenemos disponible ya la última versión de este entorno en su versión Visual Web Developer 2.008. Que es lo que vamos a utilizar ya.

Una vez que hemos seleccionado la opción que te he destacado antes llegamos a la pantalla de descarga:



Vemos a la izquierda el .NET framework. Ojo, esto no es lo que queremos porque esto es sólo para equipos que, por ejemplo, van a ejecutar aplicaciones en Visual Basic 2.008 que requieren por tanto el entorno .NET 3.5 instalado. Nos vamos a la selección de la derecha "Visual Web Developer 2008", que como veis todavía está únicamente en inglés. Seleccionamos esta opción...



**Nota:** Puede que antes de aparecer esta pantalla te solicite la instalación de un pequeño componente llamado "Silverlight". Este programa es el "flash" de Microsoft. El resultado es parecido que el famoso componente multimedia pero hecho por y a medida de Microsoft. Lo instalamos de un momento cuando lo solicite y podremos seguir.

Seleccionamos de la pantalla "Download":



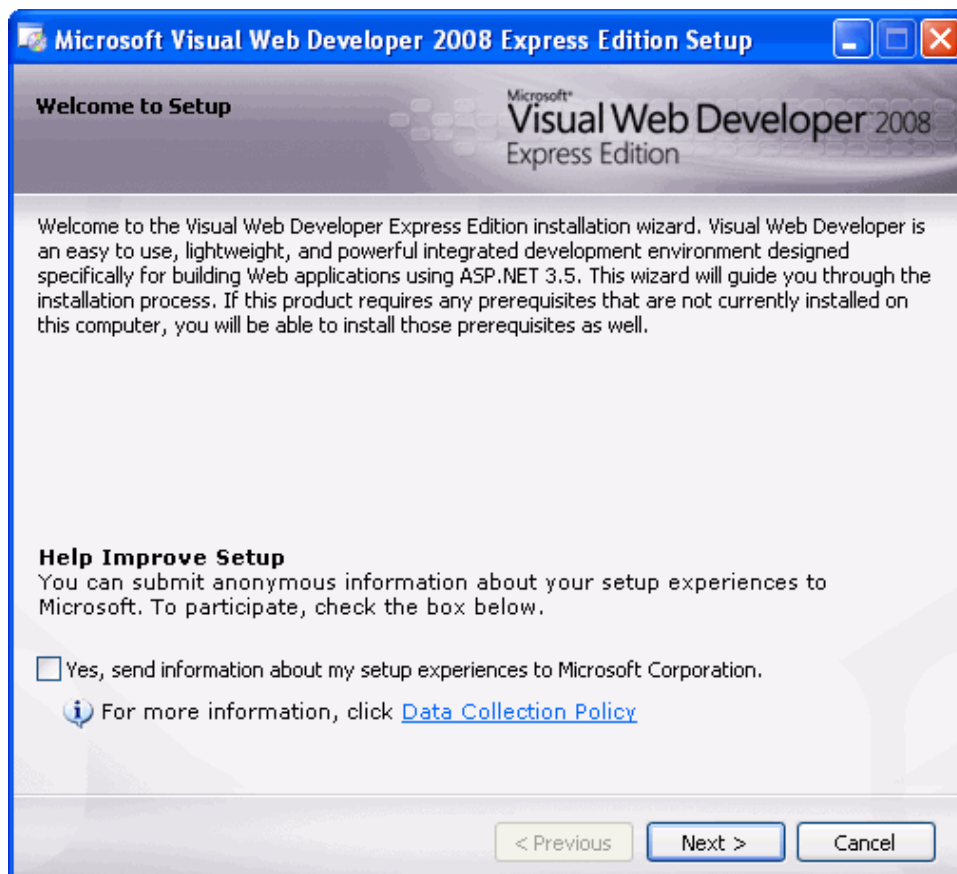
Como puedes debajo, tenemos todos los componentes del Visual Studio 2.008, desde el Visual Basic hasta el servidor de base de datos de Microsoft: SQL Server 2.005 en su versión gratuita "Express edition". En este punto podemos realizar la instalación de dos formas bien en línea en internet, seleccionando "Web Install" o descargando una imagen del DVD de instalación. Yo recomiendo esta segunda opción porque así siempre tendremos una versión del programa en DVD para reinstalaciones u otras tareas.

En este ejemplo vamos a realizar la instalación vía web... así que seleccionamos de la parte inferior de la pantalla anterior el programa "Visual Web Developer 2008", seleccionamos el idioma y pulsamos en "Download". Se nos mostrará una pantalla como esta:





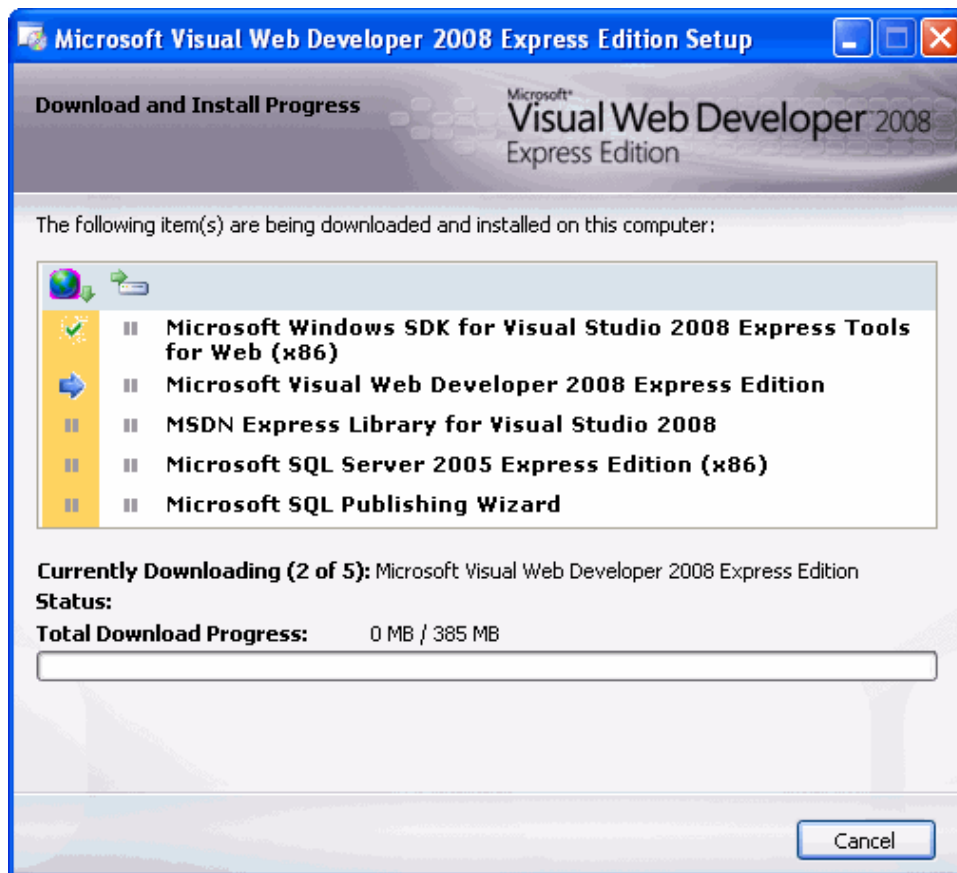
Pulsamos en ejecutar y tendremos el primer mensaje de bienvenida:



Pulsamos en "Next":

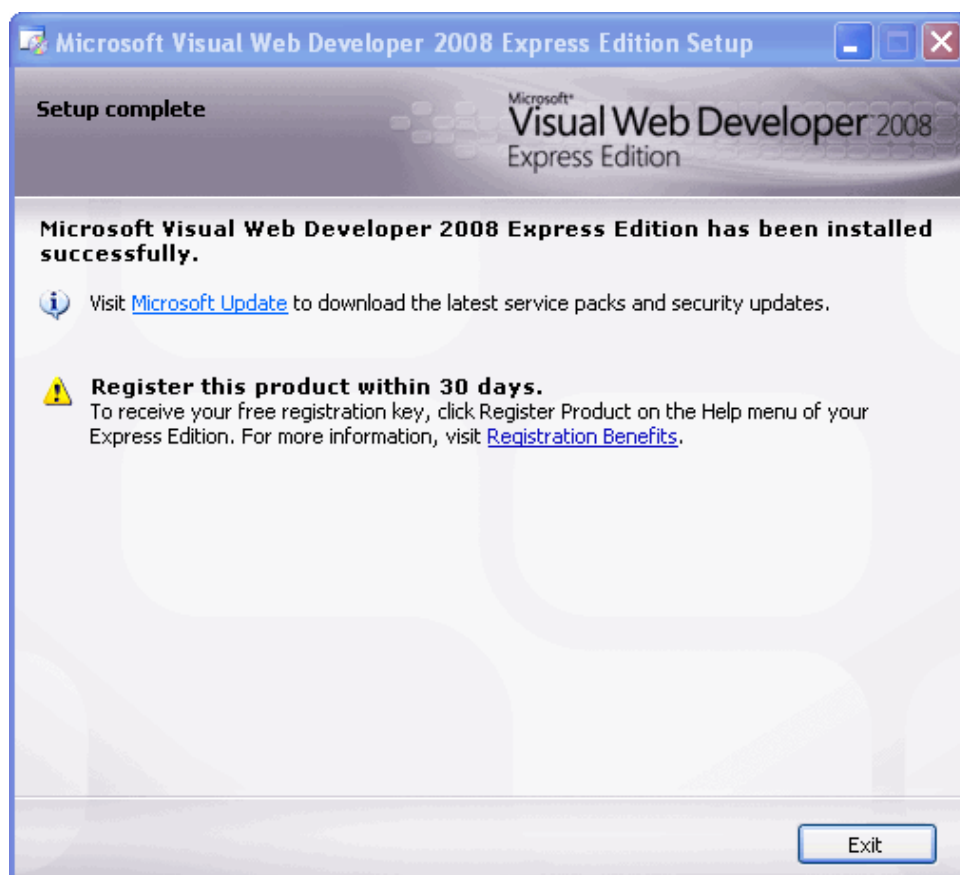


Y marcamos los dos componentes, por un lado MSDN que es la ayuda y ejemplos y por otro lado SQL Server, el servidor de base de datos de Microsoft, al que le dedicaremos un poco mas adelante. Finalmente comienza la instalación:



Y terminamos:





Ya tenemos nuestro programa instalado. Es imprescindible registrarlo para que esté operativo completamente. Ahora vamos con otro tema para instalar el servidor de páginas Web de Microsoft: Internet Information Server

**Nota:** Para desarrollo podemos utilizar el servidor web que incorpora Windows 2000/XP Professional/Vista. Es una versión reducida de la de servidor pero totalmente operativa. Todo lo que veamos en el curso funciona igualmente en esta versión del IIS. Ojo si utilizamos el mismo equipo como servidor web tendremos que cuidarnos de tener un buen equipo con procesador y memoria suficientes pues nos consumirá bastantes recursos.

Otra cosa importante, si optamos por utilizar el IIS de Windows 2000 /XP Professional/Vista recordad que es una versión limitada y sólo proporcionará páginas a cuatro usuarios como máximo.

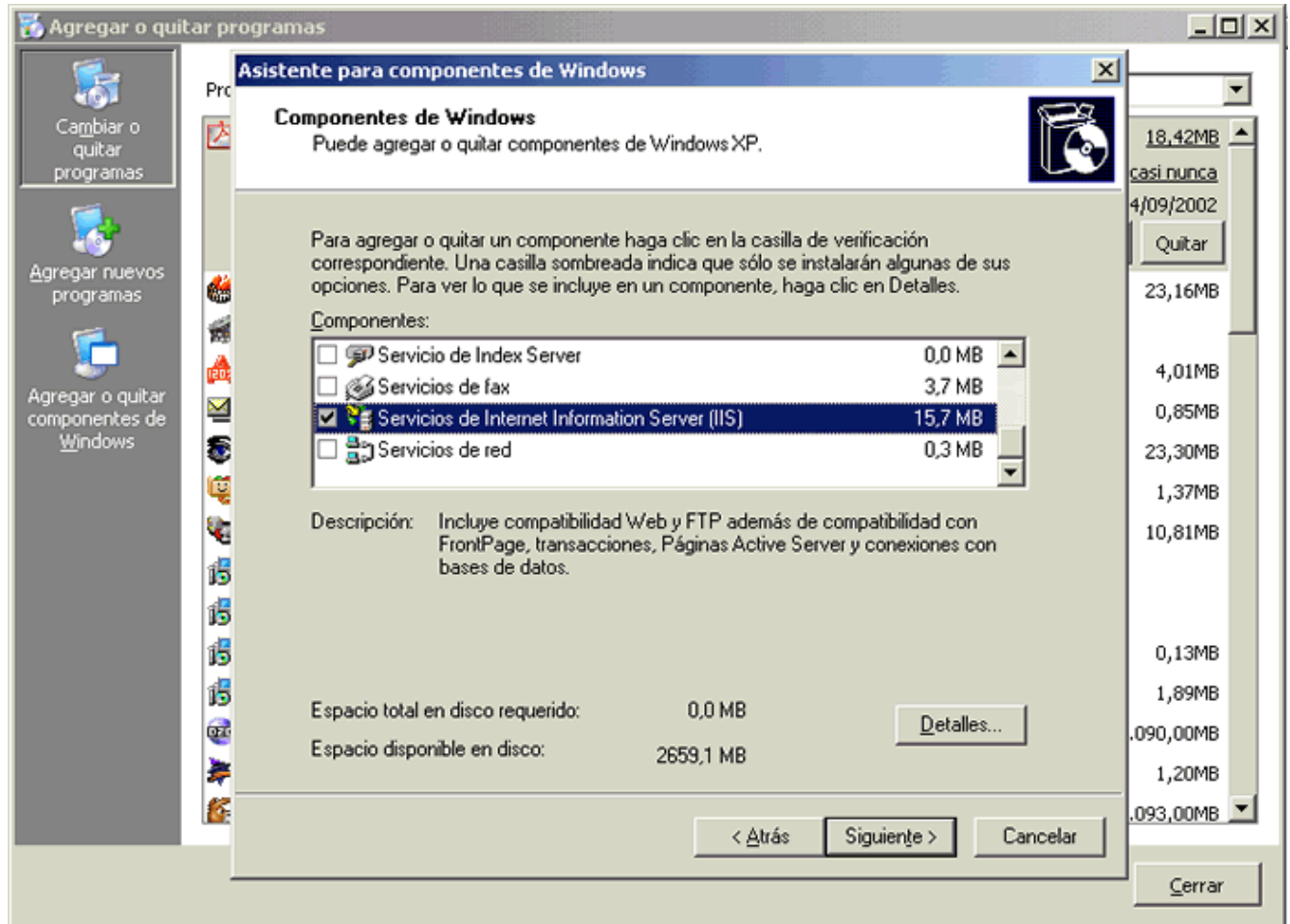
Así que si queremos utilizar nuestro equipo para desarrollar aplicaciones Web tendremos que instalar localmente este componente. Ojo, como verás las páginas web en ASP.NET se ejecutan en el servidor así que este componente sólo debemos instalarlo en el servidor. Pero si vamos a utilizar nuestro equipo local para hacer pruebas éste necesitará este componente para que pueda ejecutar las páginas Web.

### 3.2. Instalación de IIS

Ahora realizaremos la instalación de nuestro servidor Web. Durante este curso nos basaremos en Internet Information Server 6.0 que es el incluido con Windows 2003 Server que será nuestro sistema operativo para el servidor. Sin embargo prácticamente todo lo que se comenta y explica en este curso es válido para las otras versiones menores (2000 Professional, XP y Vista)

### Con Windows 2000 Server

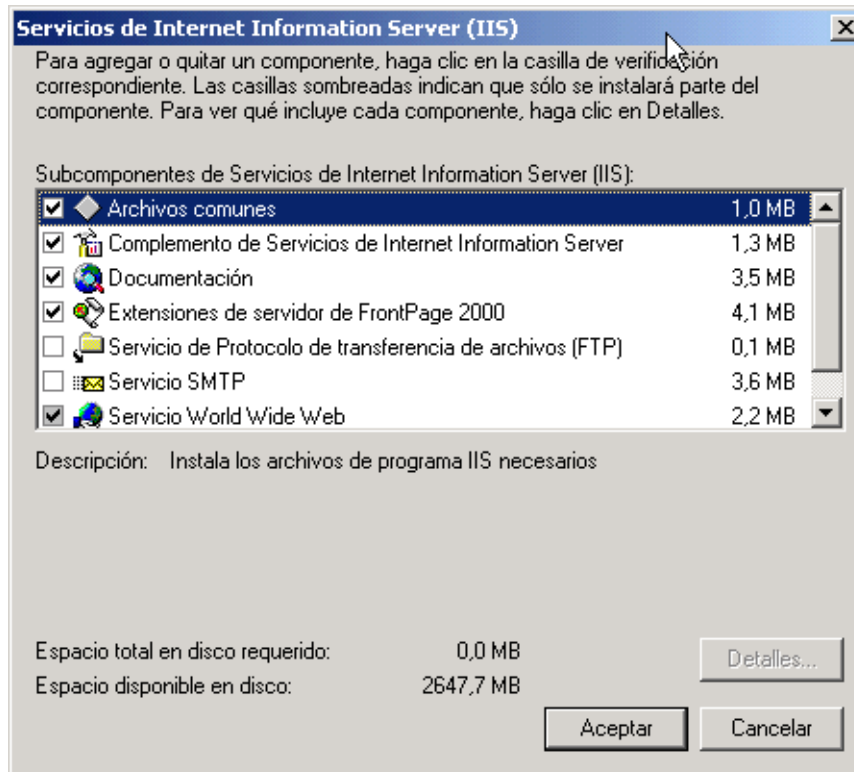
Si estamos trabajando con un Windows 2000 Server el programa ya lo tenemos instalado, vamos a comprobarlo... Hacemos doble clic en la opción "Agregar /Quitar programas" del panel de control de Windows. Una vez abierto seleccionamos la opción "Agregar componentes de Windows"



Si nos desplazamos a la parte inferior podemos ver que está instalado el programa "Servicios de Internet Information Server". Si queremos instalar la versión "Professional" para desarrollo veremos que no está instalada, pulsamos en la casilla de verificación y a continuación el botón siguiente para finalizar la instalación.

Con el botón "Detalles" podemos ver los componentes de estos Servicios de IIS. Vamos a enumerarlos para conocerlos y saber qué estamos instalando:

## Internet Information Server y ASP.NET



<b>Archivos comunes</b>	Ficheros imprescindibles de Internet Information Server. Puesto que IIS además instala otros servicios éstos utilizan estos "archivos comunes". Son pues, obligatorios
<b>Complemento de Servicios de Internet Information Server</b>	Instala la interfaz administrativa o consola de administración. También es un componente imprescindible porque nos va a permitir administrar nuestra Intranet
<b>Documentación</b>	Documentación completa del IIS: su instalación, funcionamiento, opciones, programación, ...
<b>Extensiones de servidor de Front Page 2000</b>	Complemento indispensable para que se comuniquen correctamente el servidor web (IIS) y nuestro editor Front Page 2000. Es imprescindible.
<b>Servicio de Protocolo de transferencia de archivos (FTP)</b>	Módulo adicional para crear un servidor de transferencia de archivos conocido como FTP. En nuestro paso no es necesario puesto que lo realizaremos todo desde la propia Intranet.
<b>Servicio SMTP</b>	Módulo adicional para crear un servicio de transferencia de mensajes o SMTP. Tampoco es necesario para nuestra Intranet
<b>Servicio World Wide Web</b>	Servicio de publicación de páginas web. Es decir nuestro IIS, imprescindible.
<b>Sólo Server: Servicio NNTP</b>	Proporciona un servicio de grupos de noticias o "newsgroup". No es necesario.
<b>Sólo Server: Administrador de servicios Internet HTML</b>	Páginas web con las que podemos administrar el sitio web. Es otra forma alternativa a la consola de administración. No es recomendable porque es

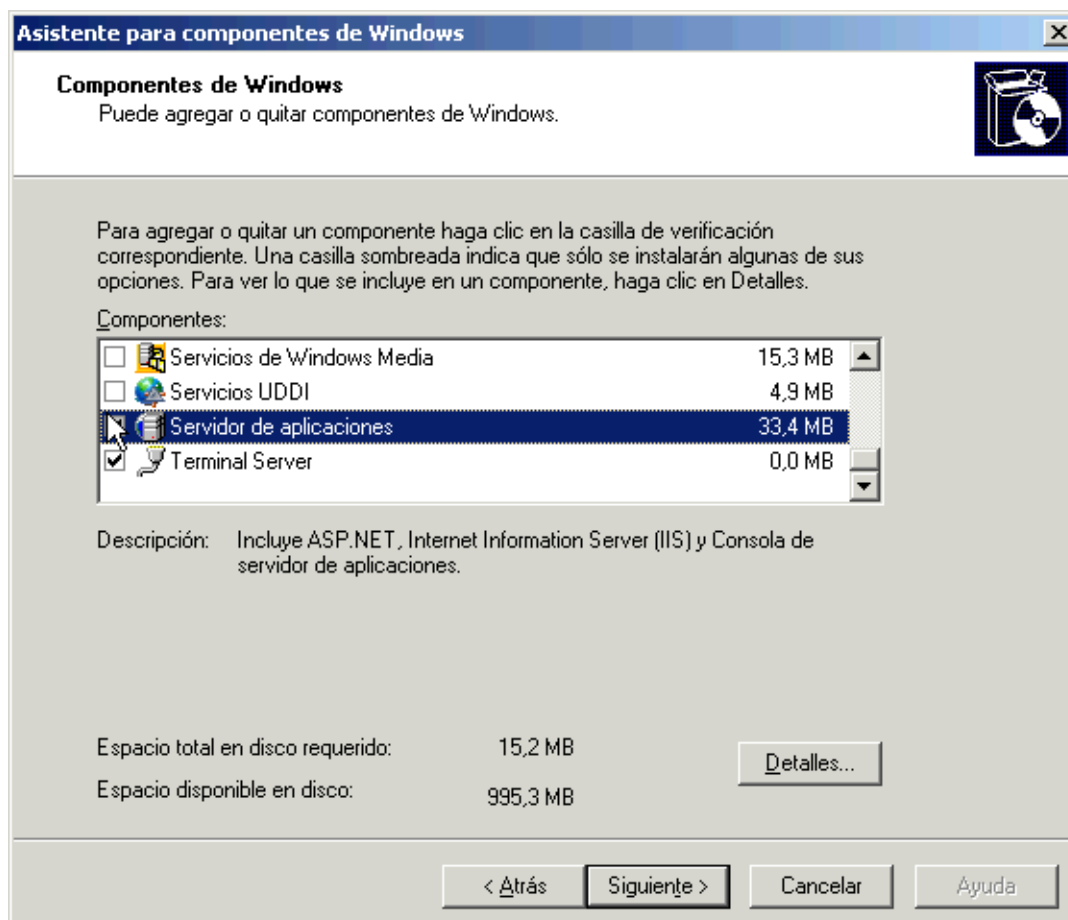
## Internet Information Server y ASP.NET

	un potencial agujero de seguridad: si alguien consigue acceder a estas páginas puede detener y romper nuestra Intranet.
<b>Sólo Server: Visual InterDev RAD</b>	Complemento para conectar los sitios web con el programa Visual Interdev para desarrollo. Este programa funciona bien con aplicaciones cliente pero no para páginas de nuestra Intranet así que no lo instalaremos.

Una vez que tenemos nuestro servidor IIS instalado o comprobado éstos se instalan en el servidor como "servicios". Un servicio es un programa especial que está continuamente en ejecución y que simplemente "espera conexiones" por ejemplo espera que le soliciten páginas web. Otros servicios pueden proporcionar vídeo bajo demanda, transferencias de ficheros (FTP) o servicios de chat. En nuestro caso nos centraremos en el servidor Web de Microsoft: Internet Information Server.

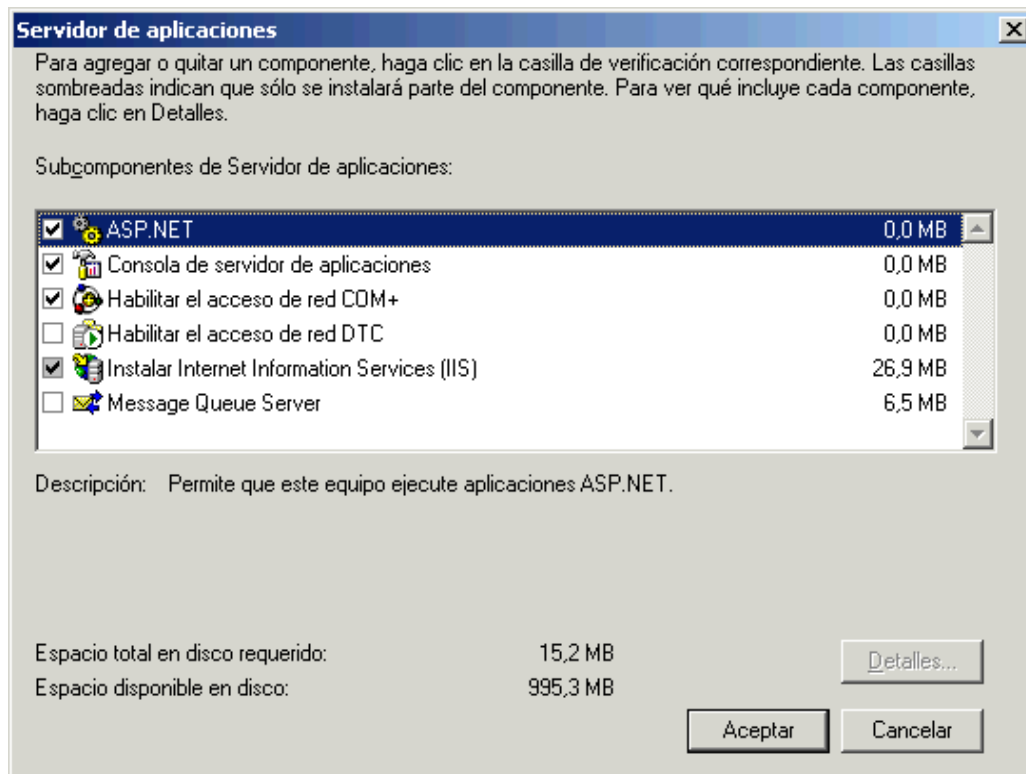
### Con Windows 2003 Server

Ahora realizaremos la instalación del servidor Web desde un Windows 2003 Server. Podemos utilizar los dos sistemas de siempre, con el asistente o con la opción de agregar/quitar programas. Utilizaremos esta última forma para buscar estos componentes:

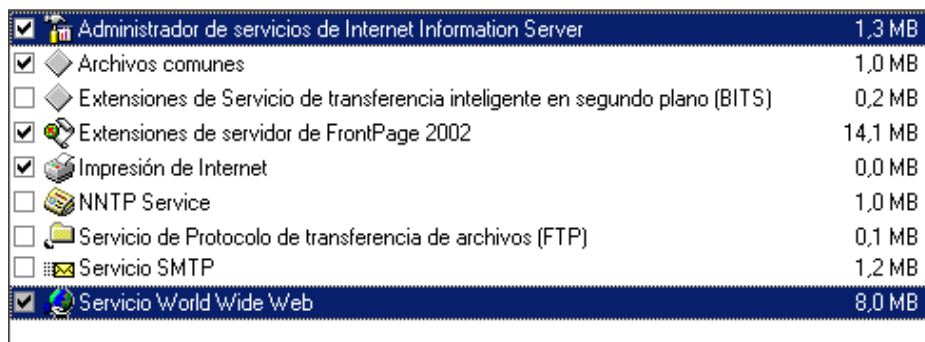


En este caso el programa se encuentra dentro del grupo "Servidor de aplicaciones":

## Internet Information Server y ASP.NET



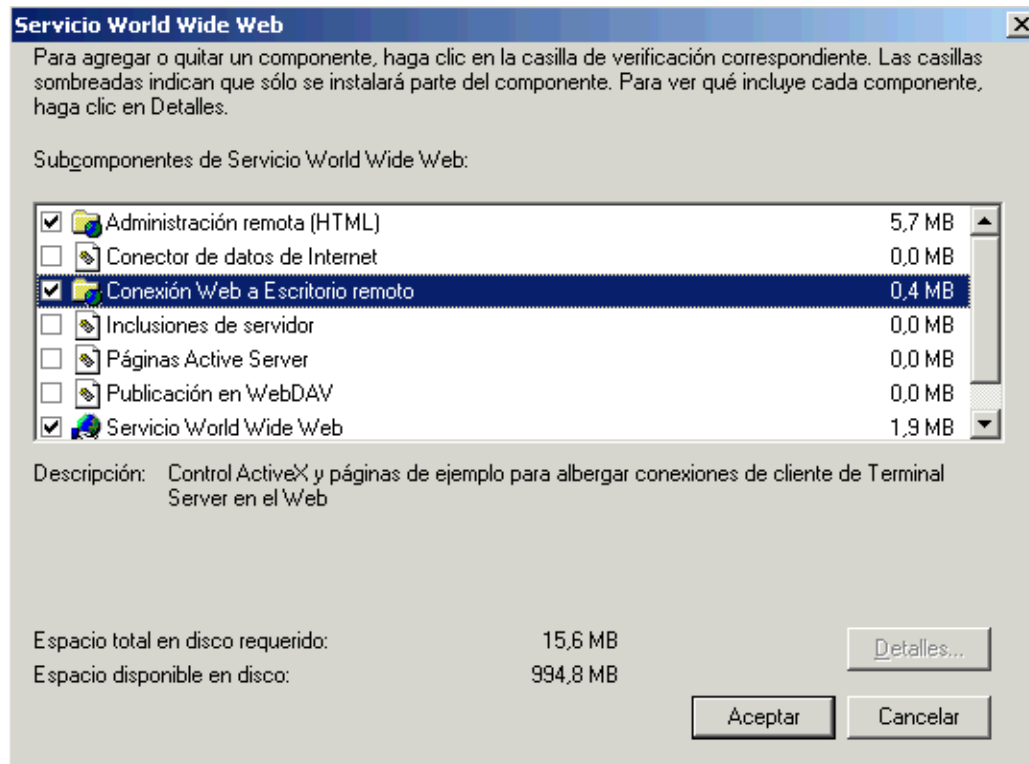
Por un lado tenemos el componente ASP.NET que es imprescindible para que nuestro IIS pueda ejecutar las páginas Web desarrollamos con programación. La marcaremos, luego marcamos la penúltima opción "Instalar Internet Information Server (IIS)" y pulsamos en "Detalles" para ver que componentes tiene:



Que corresponden con estos componentes:

- Administrador de servicios. Es el componente para nuestra consola MMC, imprescindible.
- Archivos comunes. Son los básicos para poder ejecutar IIS
- Extensiones de servicio de transferencia... son mejoras para las transferencias de archivos
- Extensiones de servidor de FrontPage 2002. Debemos instalar este componente para poder conectar FrontPage con el servidor de páginas Web.
- Impresión de Internet. Habilita la gestión de impresoras en Web. También lo vamos a marcar...
- Servicio NNT. Para activar el servicio de grupos de noticias (newsgroups) o foros de discusión
- Servicio SMTP, para transferencias de mensajes. Es el protocolo estándar de transferencia de mensajes en Internet
- Servicio WWW que es por fin, el servicio de páginas Web. Si pulsamos en detalles de éste último marcaremos estos dos componentes:

## Internet Information Server y ASP.NET

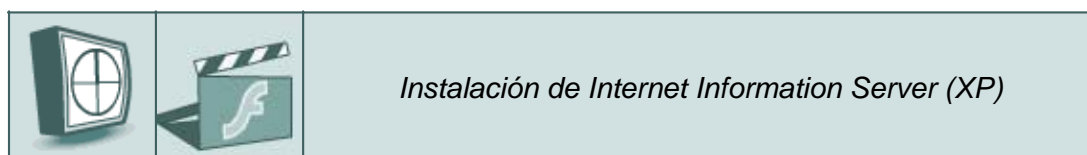


El primero es para poder administrar el IIS desde páginas Web en lugar de la consola MMC. El segundo es el cliente para Terminal Server en versión Web, donde podremos iniciar una sesión en el servidor de terminales pero desde una página Web en lugar de desde el programa cliente habitual.

Vamos a comprobar que se están ejecutando correctamente los servicios necesarios para nuestro servidor. Para esto seleccionamos la opción "servicios" dentro de la carpeta "herramientas administrativas" del panel de control. En Windows 2003 Server es algo distinto al de XP:

Nombre	Descripción	Estado	Tipo de inicio	Iniciar sesión co...
Acceso a dispositivo de interfaz humana	Habilita el acceso de entrada g...	Deshabilitado	Sistema local	
Actualizaciones automáticas	Habilita la descarga e instalaci...	Iniciado	Automático	Sistema local
Adaptador de rendimiento de WMI	Proporciona información de la ...	Manual	Sistema local	
Servicio de publicación World Wide Web	Proporciona conectividad y ad...	Iniciado	Automático	Sistema local
Servicio de administración IIS	Permite administrar los servicio...	Iniciado	Automático	Sistema local
Administrador de carga	Administra transferencias sincr...	Iniciado	Automático	Sistema local
Administrador de conexión automática d...	Crea una conexión a una red r...	Iniciado	Manual	Sistema local

Los dos servicios que deben estar funcionando son: la administración de IIS que nos va a permitir conectar la consola MMC de administración con el servidor web y el propio IIS como "Publicación en Word Wide Web". El inicio debe estar como en la imagen en "automático" de esta forma nos aseguramos que se inicia el servidor web al reiniciar el equipo o el servidor...



### 3.3. Comentarios:

Técnicamente las páginas web utilizan el protocolo HTTP de TCP/IP que corresponde al puerto 80. Por lo tanto el navegador intenta conectarse a la dirección del servidor mediante el puerto 80.

Nuestro servidor IIS no tiene ninguna desventaja del famoso Apache de Linux. De hecho si quitamos la gratuidad al servidor Apache no le queda absolutamente nada que no pueda hacer IIS, incluso al contrario: la orientación a componentes de IIS es mucho mas completa y mejor implementada que la del Apache. IIS soporta la misma carga de clientes, es fácil de instalar y configurar y su mantenimiento es trivial cosa que no se puede decir de su competidor. Uno de los temas en los que Microsoft todavía tienen que avanzar es en la seguridad, el abarcar tantos conceptos, tecnologías y componentes hace que IIS sean mas susceptible de tener agujeros de seguridad que su competidor, mas simple y limitado. En cualquier caso en las Intranets IIS de Microsoft gana por goleada. A lo largo de este curso iremos viendo porqué y que elementos podremos integrar en nuestra Intranet.

Aun así y comprendiendo la importancia del mundo Web hay una versión de Windows 2003 y 2008 Server llamada "Web Edition" que permite de una forma muy económica montar un sitio web con toda la potencia de su entorno e integrado perfectamente en nuestra red. Le han limitado algunos servicios pero dispone de todo lo necesario para desempeñar su función perfectamente. Todo a un precio muy asequible que queda perfectamente amortizado por la cantidad de horas necesarias para configurar un servidor Linux.

Como complemento a nuestra Intranet debemos utilizar bases de datos para almacenar información. Quizás el uso de las bases de datos en las Intranets sea la parte mas importante porque la función de una Intranet que es proporcionar información debe estar de alguna forma almacenada y ordenada en bases de datos.

En capítulos posteriores hablaremos en profundidad de las bases de datos pero como en esta introducción estamos explicando que nos hace falta para construir nuestra Intranet, no podíamos dejar de comentar esta parte.

IIS ofrece una administración muy sencilla que se realizará mediante el Administrador de servicios de Internet. IIS permite que el desarrollo de aplicaciones Web sea mucho más robusto y la creación de sitios Web sea más configurable y completa. Ofrece un entorno escalable basado en los componentes cliente/servidor que se pueden integrar dentro de las aplicaciones Web.

Internet Information Server es el servidor Web más rápido y recomendable para la plataforma Windows 2000/2003, ya que se encuentra integrado completamente con el Servicio de Directorios de Windows (directorío activo), esta combinación del servicio Web con los servicios del sistema operativo permite desarrollar aplicaciones basadas en la Web fiables y escalables. Que conste que esto no lo he sacado de la propaganda, Microsoft suele dejar los programas mal terminados (agujeros de seguridad, fallos) pero las ideas son buenas y casi siempre aporta facilidad de uso. Será el más criticado pero también hay que reconocer que está en el 90% de todos los ordenadores del mundo.

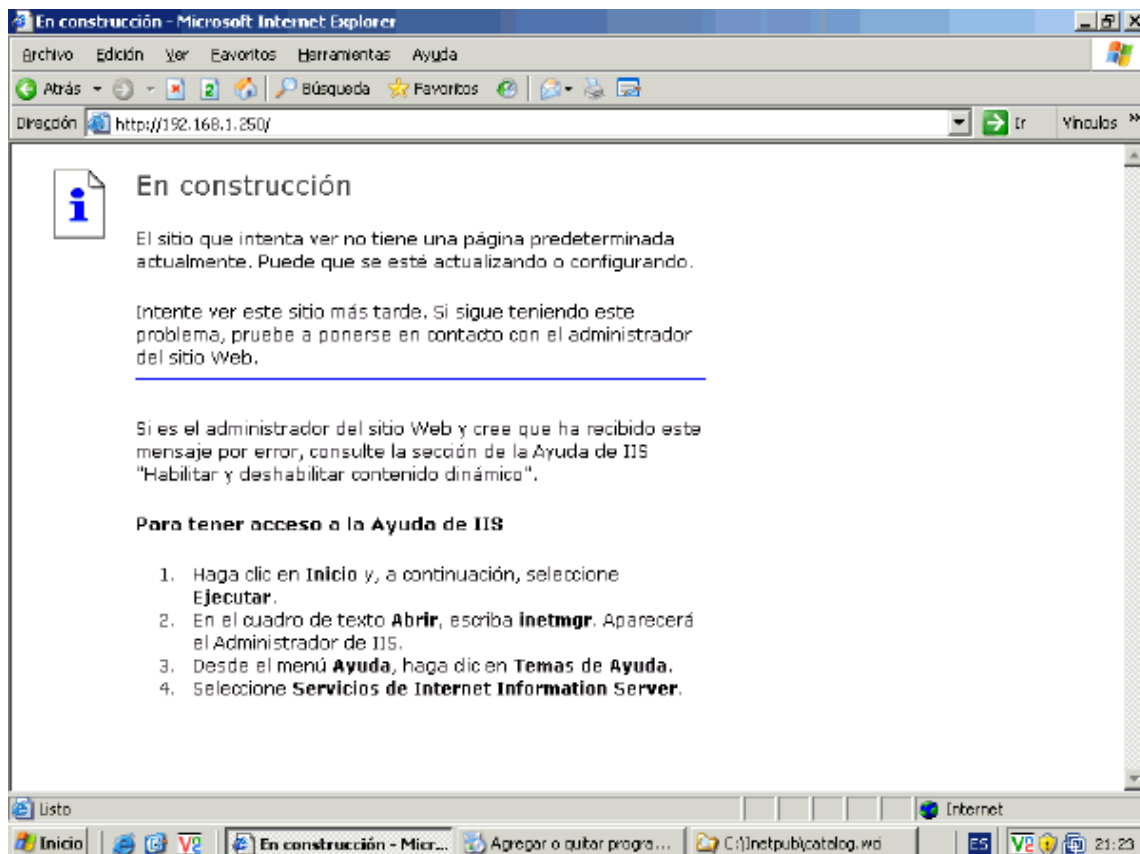
## 4. Primeros pasos con IIS

Si la instalación ha sido correcta podremos entrar ya en nuestro sitio web. Para comprobarlo basta con iniciar un explorador web y escribir en la barra de dirección el nombre de nuestro servidor web, en nuestro caso le hemos llamado "servidor". Se pueden dar dos casos, si lo estamos ejecutando desde el servidor o desde un equipo de nuestra red. No es recomendable trabajar en el propio servidor pero si fuera así recibiríamos una pantalla como esta en un servidor Windows 2000:





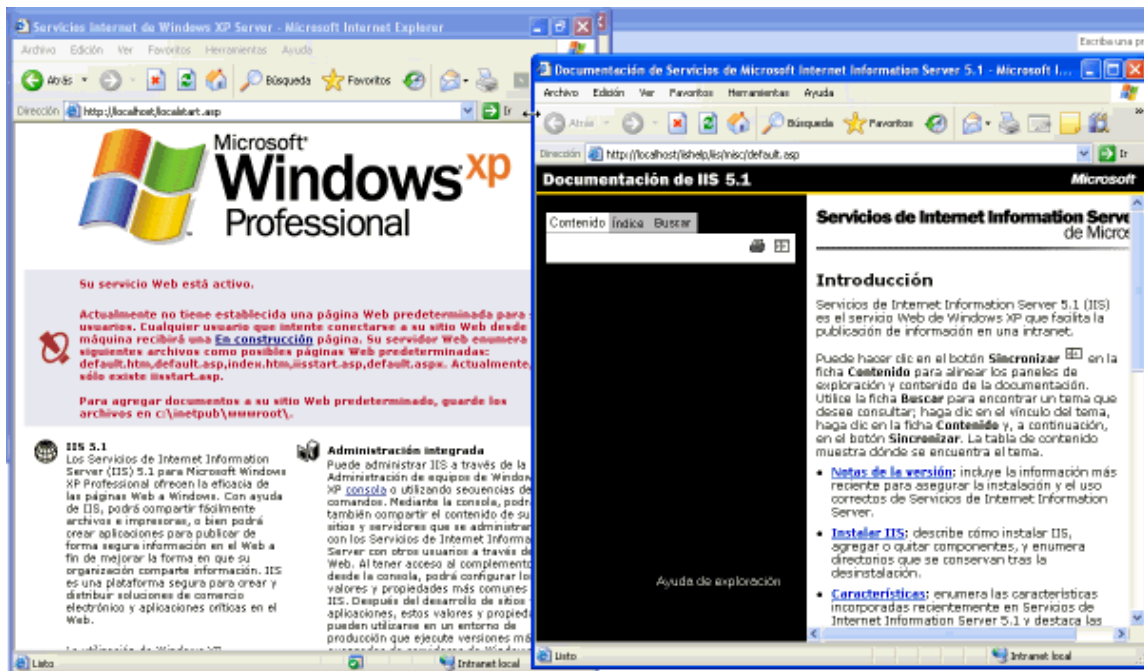
Si el servidor es Windows 2003 Server la pantalla será:





## Internet Information Server y ASP.NET

Y si estamos con XP:



Pantalla que nos da la bienvenida al servidor IIS y nos facilita alguna herramienta de administración a través de páginas web. Sin embargo lo habitual y recomendable es que trabajemos desde otro equipo. Si es este el caso en lugar de recibir una pantalla con administración e información sobre el servidor web recibimos otra por seguridad similar a la siguiente:

**Nota** Si estamos trabajando con las versiones "personales" del IIS, es decir, con W2000 Professional, Windows XP ó Vista, lógicamente estamos trabajando en local por lo que recibiremos la pantalla mencionada arriba.



### En construcción

El sitio al que intentó conectarse no tiene en este momento una página predeterminada. Es posible que esté en proceso de actualización.

Inténtelo de nuevo más tarde. Si el problema continúa, póngase en contacto con el administrador del sitio Web.

Si nos fijamos en la primera pantalla, cuando se ejecuta en el equipo que tiene el servidor ya nos avisa de que los clientes recibirán este mensaje:



**Actualmente no tiene ningún documento configurado para sus usuarios. A todos los usuarios que intenten conectar con su sitio, se les mostrará una página "En construcción".**

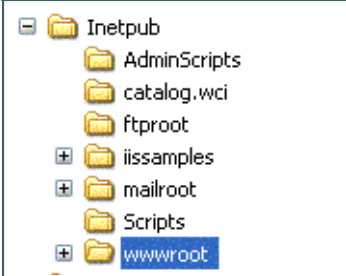
Si nos fijamos en el mensaje lo que al principio podíamos tomar como un error es simplemente un mensaje que le falta una página de inicio al servidor web. La versión 4.0 de NT Server si que activaba por defecto un sitio web de ejemplo pero ese sitio era un agujero de seguridad puesto que permitía realizar algunas

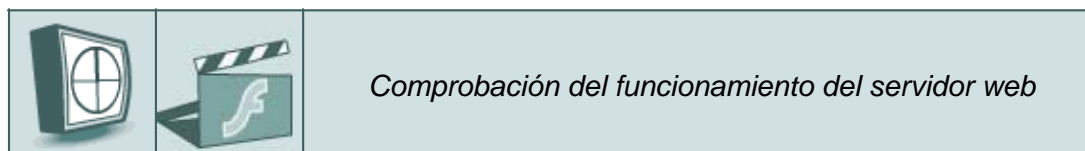
## Internet Information Server y ASP.NET

operaciones "peligrosas" por este motivo en el Server 2000/2003 Microsoft no activa ningún web sino que devuelve una página diciendo que si está funcionando pero que ahora hay que alimentarlo.

Por lo tanto, para empezar a configurar los sitios Web debe indicar los directorios que contienen los documentos que desea publicar. El servidor Web no puede publicar documentos que no están en los directorios especificados. Por lo tanto, el primer paso para desarrollar un sitio Web debe ser determinar cómo desea organizar los archivos. Después se utiliza el complemento IIS para identificar los directorios que forman parte del sitio.

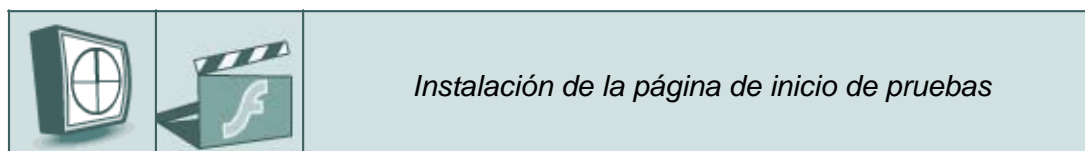
La estructura de directorios que IIS crea es la siguiente:

	<p>En la unidad C se crea el directorio "inetpub" y de ahí los servicios instalados, en la imagen están además del servicio web "wwwroot" el servicio de mensajería "mailroot" y el de transferencia de ficheros "ftproot".</p> <p>Luego... inicialmente cuando solicitemos una página irá a buscarla al directorio <b>c:\inetpub\wwwroot</b></p>
---	---



Ya es hora de saber si funciona nuestro sitio web. Para esto vamos a descargar una página sencilla de ejemplo a nuestro servidor. Para esto pulsa con el botón derecho en el siguiente enlace y a continuación "Guardar destino como..." la almacenaremos como "default.htm" es decir como página por defecto. Esta página web que no contiene imágenes la debemos copiar al equipo que tenga el servidor web y dentro del directorio "c:\inetpub\wwwroot". Si volvemos ahora a nuestro navegador y actualizamos la página o escribimos "servidor" en la barra de direcciones aparecerá nuestra página de ejemplo...

[Haz clic aquí para descargar la página de pruebas](#)



En la siguiente sección veremos como administrar básicamente nuestro Internet Information Server y las opciones básicas más importantes.


## 5. La consola Administrativa de IIS

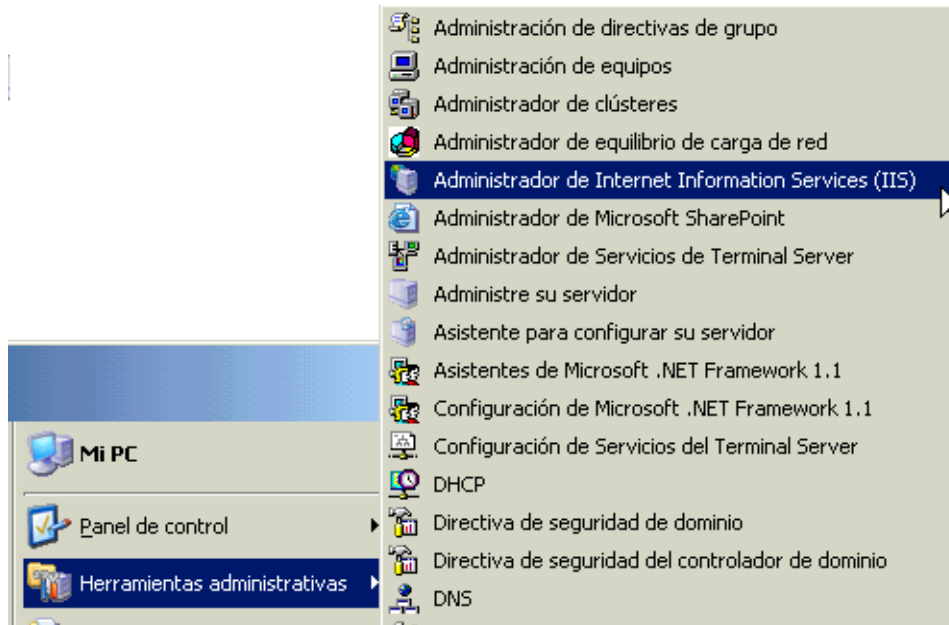
Para administrar y controlar nuestro servidor web vamos a utilizar la consola administrativa de IIS. Para utilizarla necesitamos tener instalado el componente de IIS en nuestro equipo. Si estamos trabajando directamente con el servidor o si tenemos un IIS instalado en nuestro equipo tenemos lo necesario para poder iniciar la consola.

**¡Atención!** Si no hemos instalado el servidor en nuestro equipo o no trabajamos directamente con el servidor debemos instalar los elementos necesarios para administrar el servidor web desde nuestro equipo. Para esto

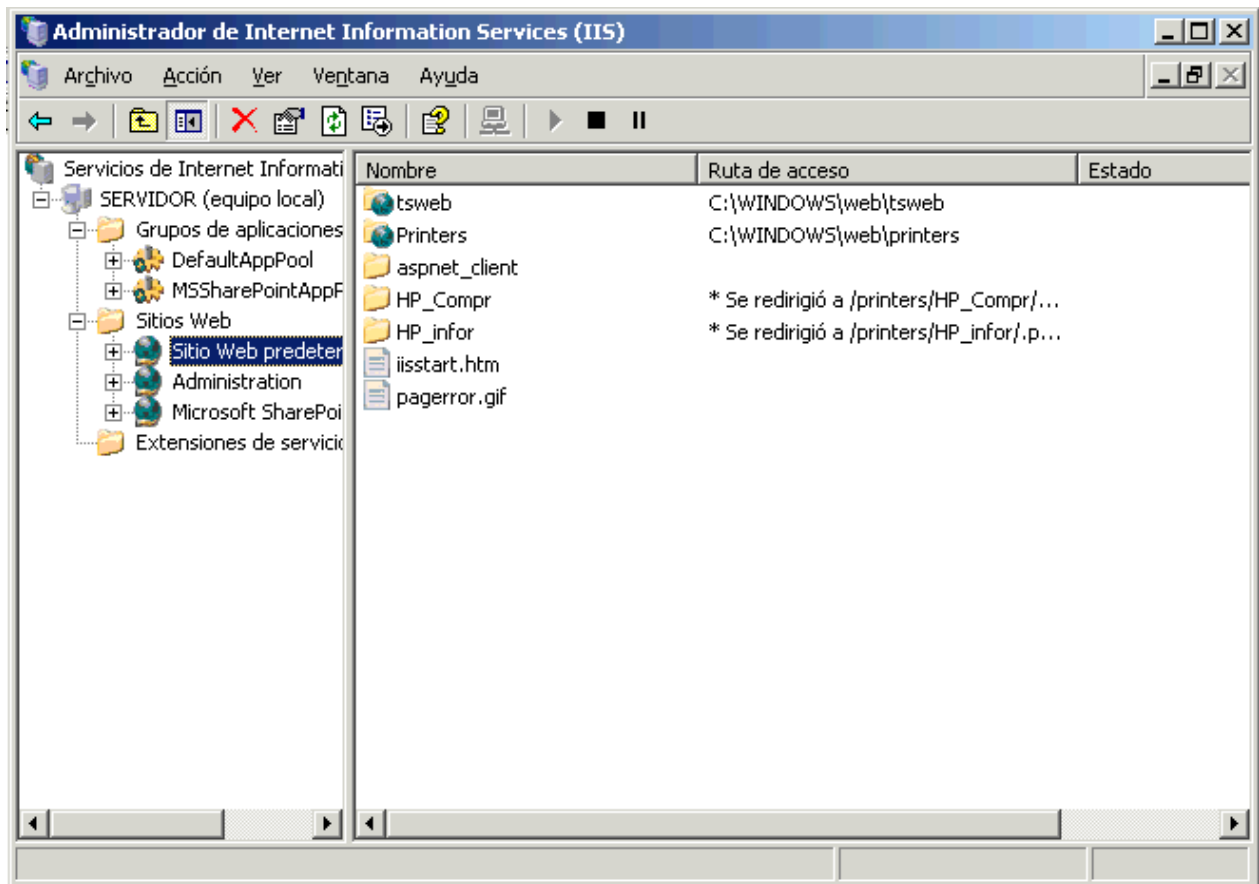
## Internet Information Server y ASP.NET

debemos instalar los elementos administrativos que incorpora Windows 2000/2003 Server. Basta con instalar el paquete que se encuentra en "Agregar programas" y luego en la sección de Internet el "Administrador de servicios Internet"

Para ponerla en marcha pulsaremos el icono  "Servicios de Internet Information Server" de las Herramientas Administrativas del Panel de Control. O si estamos en el servidor seleccionamos la opción de la siguiente figura:

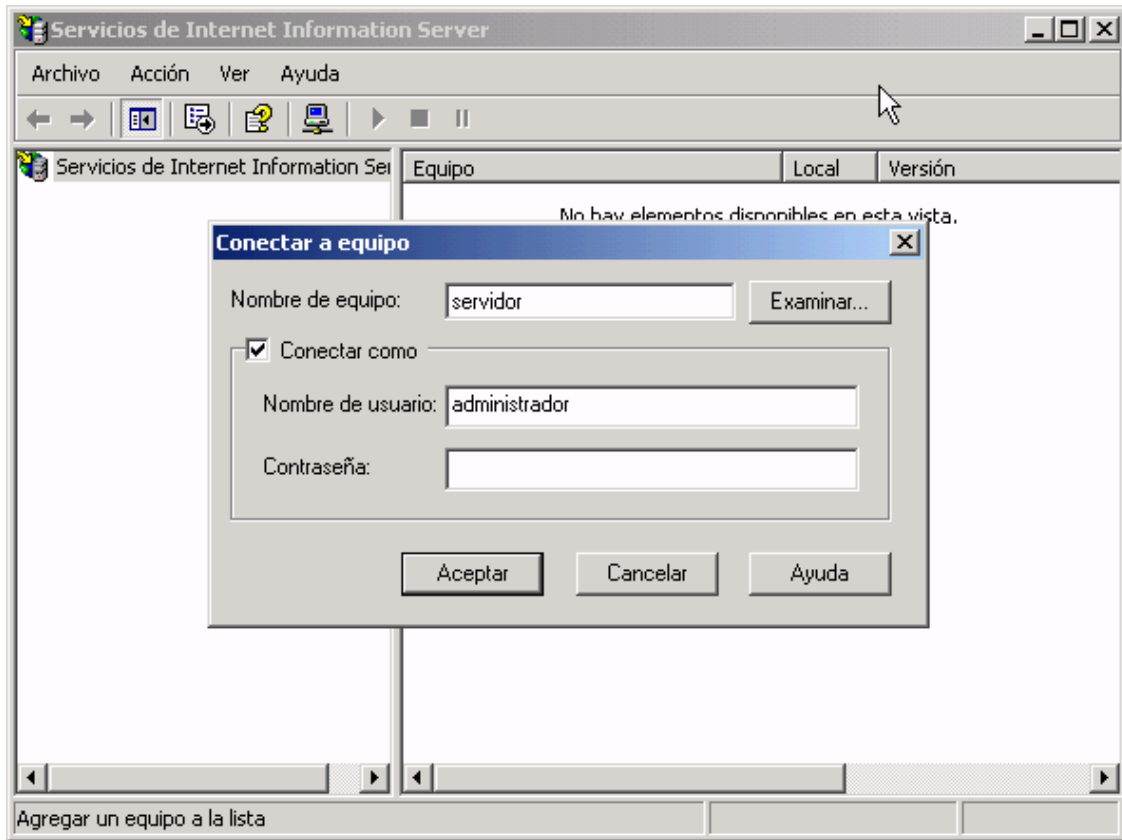


Al iniciarse, la consola tendrá un aspecto parecido a este:

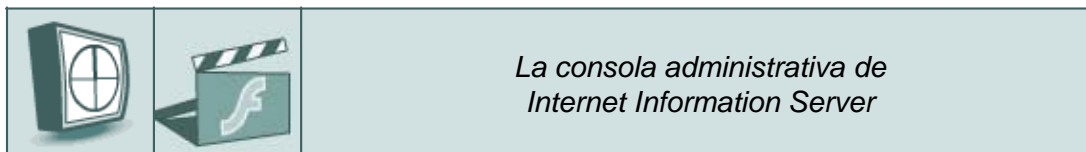


**¡Atención!** Si hemos instalado el servidor de desarrollo en nuestro equipo debemos conectarnos al servidor para poder administrarlo, así que pulsando con el botón derecho en el título debemos "conectar" con el servidor W2000/W2003/W2008. No confundirlo con el local...

Por una parte tenemos un icono con el nombre del equipo que estamos administrando. Si disponemos de mas servidores web en nuestra organización podríamos administrarlos todos desde aquí. Para conectarnos a otro bastaría con pulsar con el botón derecho en el título de la parte de la izquierda "Servicios de IIS" y seleccionar la opción "Conectar":



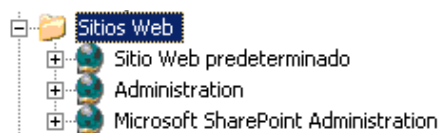
Introduciríamos en el formulario el nombre del servidor y los credenciales de inicio de sesión si se necesitan.



Estas son todas las opciones que tenemos disponibles en nuestro servidor IIS. La mayoría de ellas son conceptos sencillos y con una breve descripción podremos hacer uso de éstas. Antes de empezar a explicar sus detalles vamos a ver los elementos que nos pueden aparecer en la consola administrativa:

Dentro del Administrador de servicios de Internet se pueden distinguir varios nodos, dependiendo también de la instalación que se haya realizado de IIS. En la instalación más completa vemos lo siguiente:










- Grupos de aplicaciones. Puesto que ASP.NET funciona como un servidor de aplicaciones, IIS instala una serie de programas para administrar el funcionamiento de estos servicios.
- Sitios Web, tiene tres ramas:



- ◊ Sitio Web predeterminado. Es el sitio que crea como ejemplo y como inicio. Aquí es donde está alojada la página de "en construcción" que nos ha mostrado antes

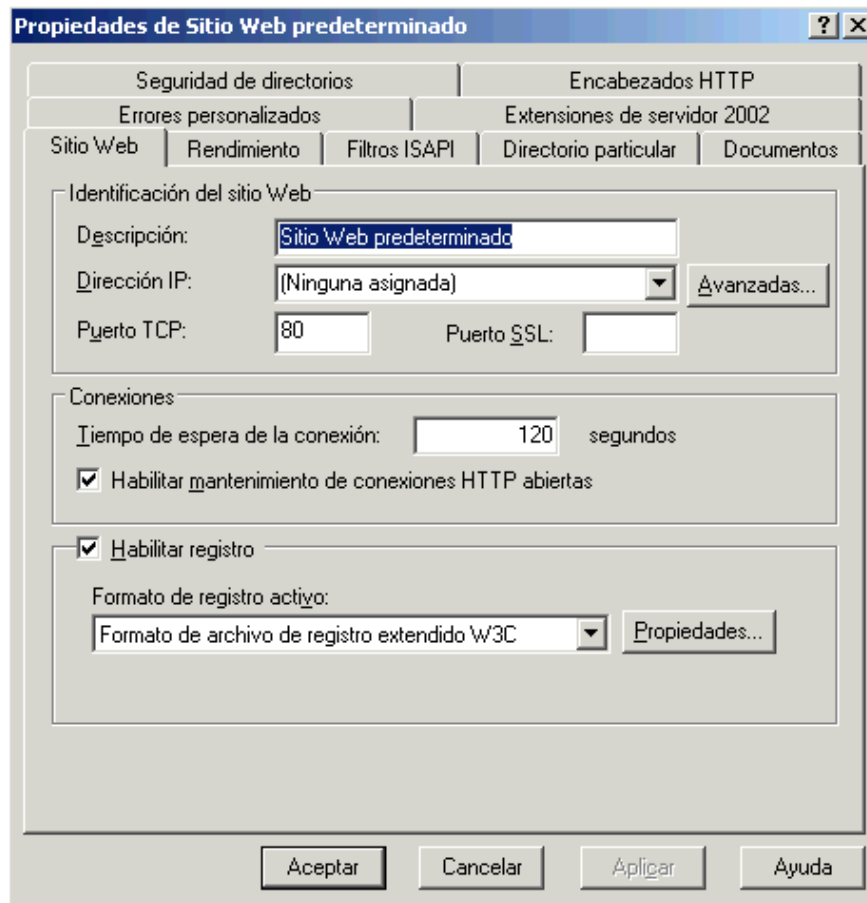
## Internet Information Server y ASP.NET

- ◇ Administración. Es un sitio Web que nos va a permitir administrar el IIS de forma remota y sin tener que utilizar la consola MMC.
- ◇ Microsoft SharePoint Administration. Son servicios especiales del software de portales de Microsoft SharePoint.
- Extensiones de servidor Web. Son programas o componentes que se pueden ir añadiendo al servidor Web para proporcionarle mas funcionalidades. Por ejemplo, uno de ellos es el componente que permite conectar IIS con FrontPage:

	Todas las extensiones CGI desc...	Prohibido
	Todas las extensiones ISAPI des...	Prohibido
	ASP.NET v1.1.4322	Permitido
	Conector de datos de Internet	Prohibido
	FrontPage Server Extensions 2002	Permitido
	Impresión de Internet	Permitido
	Inclusiones de servidor	Prohibido
	Páginas Active Server	Permitido
	WebDAV	Prohibido

En este capítulo nos vamos a centrar en el sitio Web predeterminado, que representa el sitio Web por defecto del servidor IIS y que se corresponde con <http://nombreMaquina>. Veamos ahora algunos detalles, por ejemplo... ¿Cómo es posible que tengamos dos sitios Web en un mismo equipo? Pues sí, podemos ver en la consola que tenemos dos sitios Web: el sitio Web predeterminado y el de administración.

Entonces ...¿cómo accedo a cada uno de ellos? De acuerdo, sabemos que las páginas Web (protocolo HTTP) utilizan el puerto 80 de TCP/IP. Esto es el canal de comunicaciones que va a utilizar nuestro navegador con el servidor IIS. Cuando solicitamos una página Web, nuestro navegador abre una conexión con el servidor indicado por el puerto 80. Si vemos otra vez la página de propiedades del "Sitio Web predeterminado":

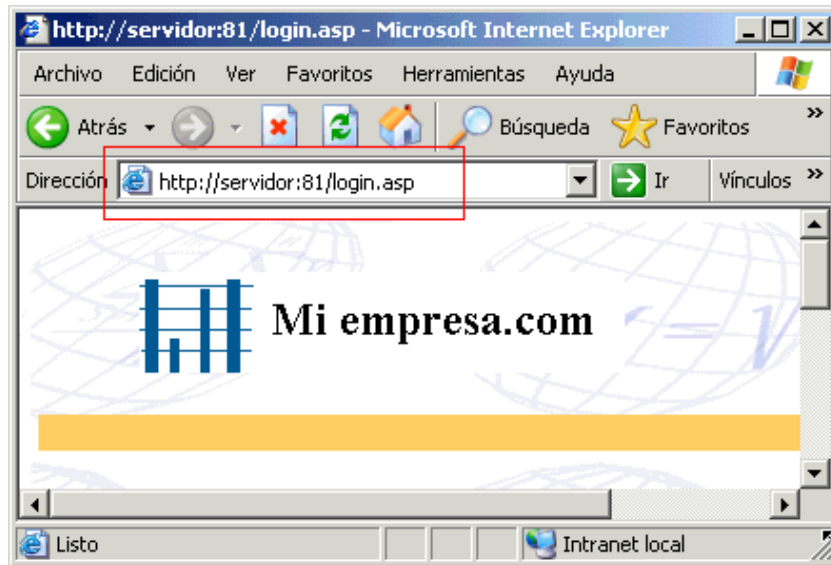


Vemos la página de las propiedades principales de nuestro servidor Web. Vemos un título en el campo "Descripción", una dirección IP, que aparece como "Ninguna asignada" y un puerto TCP. Ok, este es nuestro puerto, el 80, que es estándar para las páginas web. Si cambiamos este puerto, al 81 por ejemplo, y escribimos el nombre del servidor en el navegador vemos que no aparece ninguna página porque no puede establecer una conexión a un sitio web que exista en el puerto 81 porque el navegador por defecto busca uno en el 80:



**Nota** Todos los demás servicios de Internet también utilizan su propio puerto: 21 para FTP, 25 y 110 para el correo electrónico, 23 para el Telnet, ...

Cuando instalamos un sitio web en un puerto que no es el estándar debemos escribir en la URL a continuación del nombre del servidor el puerto TCP donde se encuentra el sitio web:

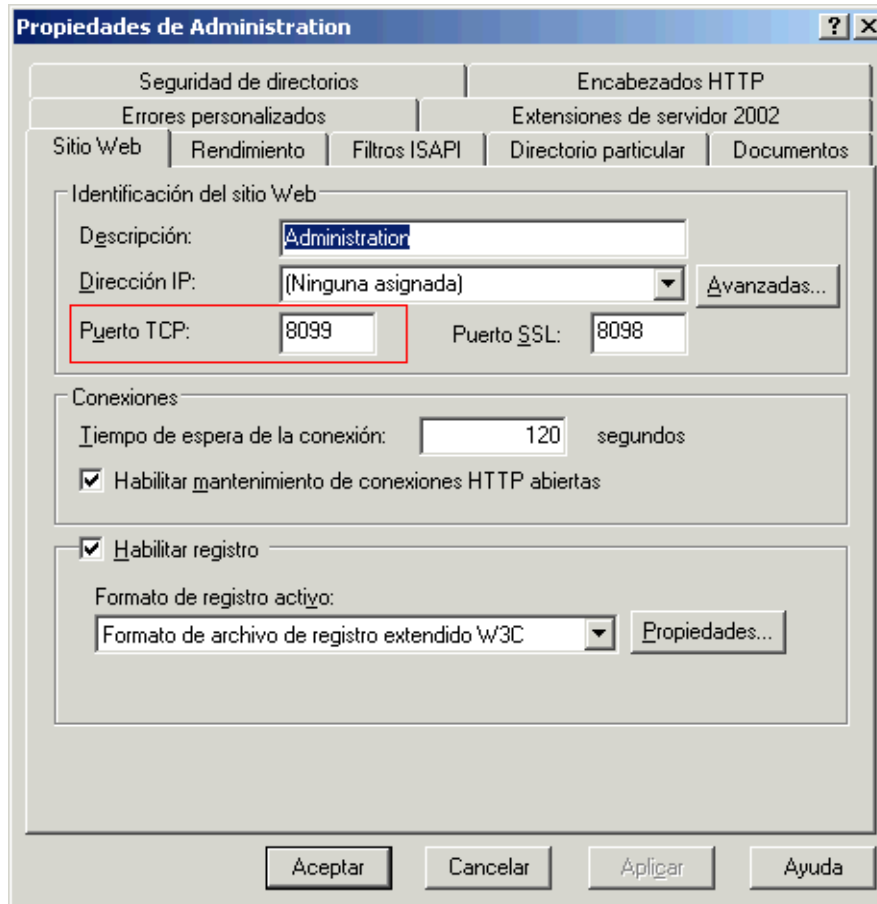


Como vemos en este caso en la URL hemos escrito "http://servidor:81" y nos ha presentado la página de inicio. Pero volvamos a dejarlo como estaba con el 80. Vemos entonces que si podemos tener varios sitios web, de hecho en la instalación hemos comprobado que tenemos nuestro sitio web predeterminado y uno de administración.

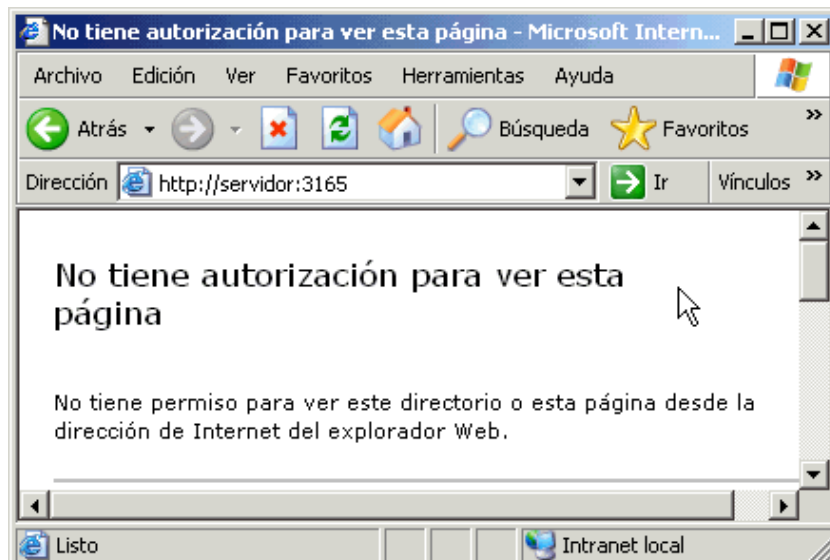
**¡Atención!** No pueden funcionar dos sitios web en el mismo puerto. Sólo puede haber uno en el puerto 80. Si creamos un sitio web y le asignamos también el 80, que está siendo utilizado por otro, éste no se iniciará. Debemos asignarle otro puerto.

Por lo tanto el sitio Web de administración no puede estar en el mismo puerto 80. Si vemos su página de propiedades vemos que en este ejemplo está en el :





Puerto 8099. Que es el puerto que debemos poner en el navegador para poder acceder a este sitio web:



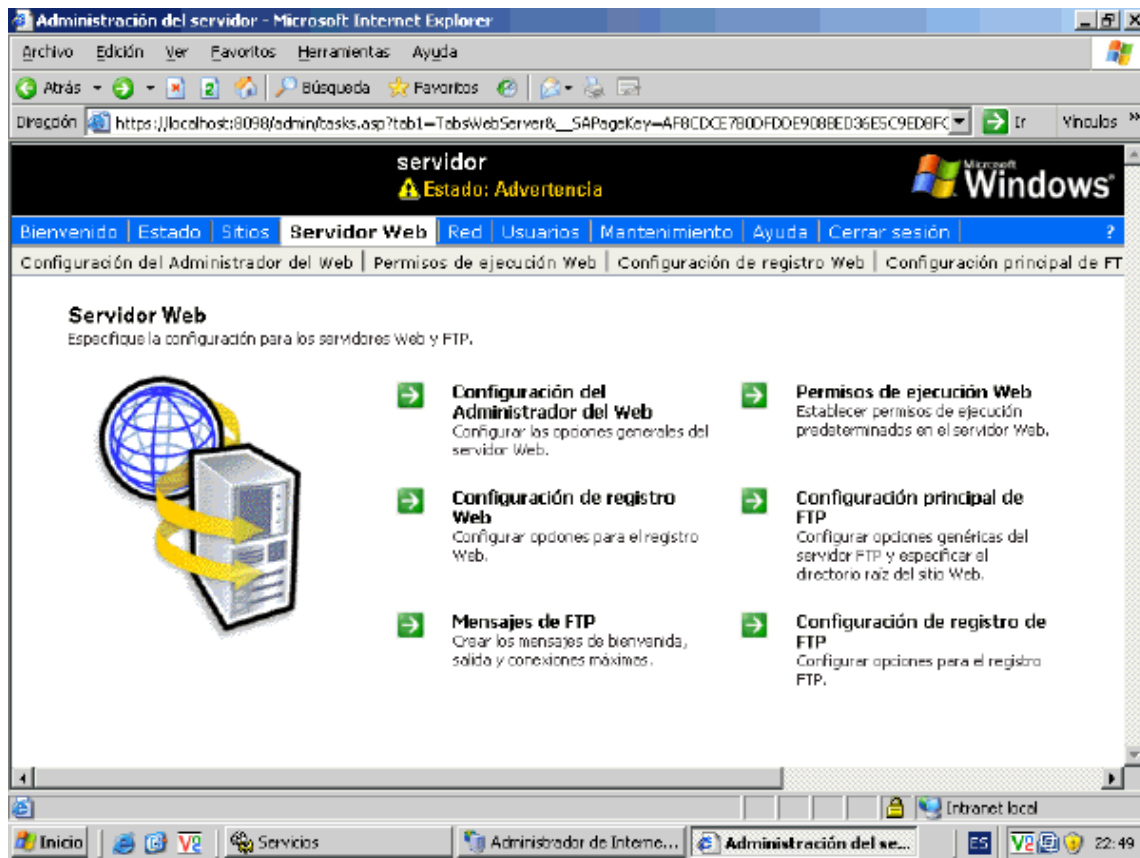
Aun así todavía nos falta algún tipo de permiso. Esto está bien, hay que darse cuenta de que el que pueda llegar a esta página puede administrar completamente el sitio Web. Podría borrarlo, desconfigurarlo o introducir una puerta trasera para posibles ataques. Entonces es normal que no sea fácil llegar hasta él. De momento, y hasta que vemos temas de seguridad sólo va a funcionar desde el propio servidor, así que para probarlo tenemos dos posibilidades:

## Internet Information Server y ASP.NET

- Movernos físicamente hasta el servidor para escribir la dirección URL:puerto
- Si estamos trabajando con el servidor en local si que nos funcionará.

Si intentamos ahora nuevamente tendremos problemas porque nos indica que debemos acceder mediante una conexión segura. En la pantalla de la configuración del puerto teníamos que para conectarnos como un sitio Web seguro, con una conexión segura SSL, debemos utilizar el puerto 8089, así que escribiremos la dirección:

<https://servidor:8089> (o el número de puerto que te indique esa pantalla). Recuerda poner el protocolo seguro HTTPS en lugar de HTTP. Si es correcto todavía nos preguntará los credenciales de conexión usuario-contraseña. Como estamos haciendo una conexión segura no nos preocupamos de poner estos datos ya que viajarán encriptados. Por fin llegaremos a la página de administración:

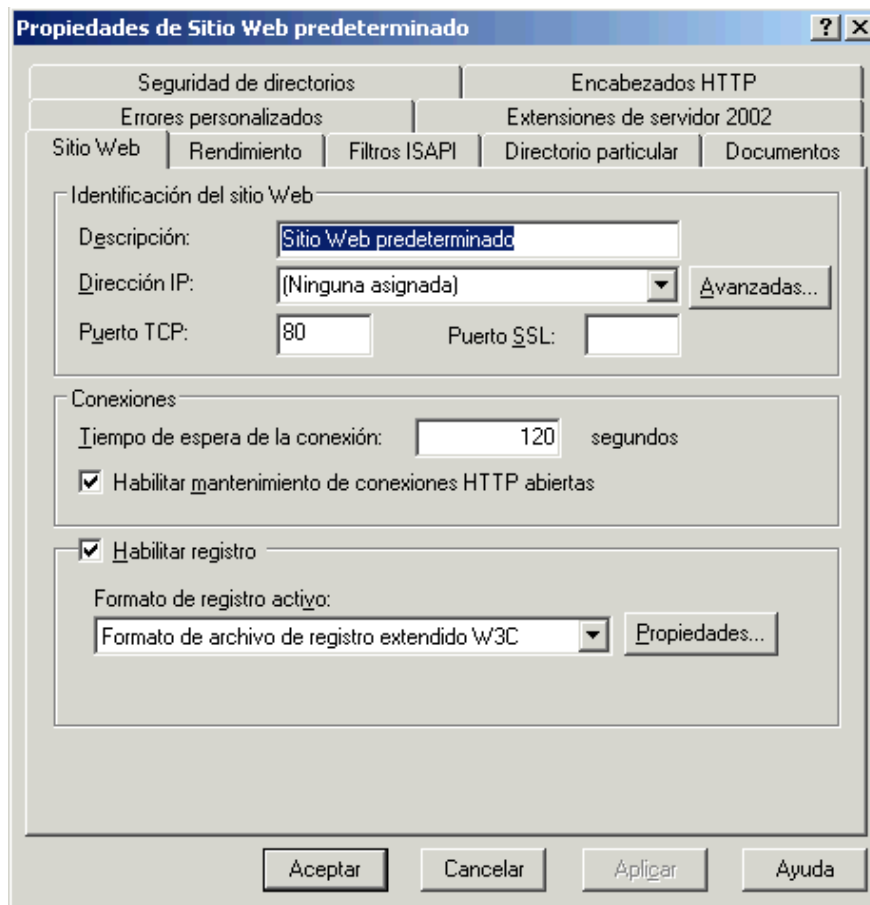


Como ves es un extenso conjunto de utilidades para administrar de forma remota el servidor. Puede que te sean de utilidad porque podrás acceder al servidor incluso desde Internet y realizar la administración completa del servidor. Si pulsas en "Servidor Web" tendrás muchas de las opciones que aparecen en la consola MMC para configurar el servidor Web

Pero nosotros seguiremos con la consola por comodidad y por rapidez. Antes de pasar a conocer qué son los directorio virtuales conozcamos mejor qué son las opciones que nos aparecían en las propiedades de nuestro sitio Web.

## 6. Propiedades del sitio web

Como vimos antes, la primera página de propiedades de IIS que nos aparecía al seleccionarlo era esta página:



En esta ficha, podremos configurar una serie de parámetros relacionados con la identificación del sitio Web, las conexiones al sitio Web y la forma de registrar los accesos a nuestro sitio Web.

En cuanto a la identificación del sitio Web tenemos cuatro parámetros: Descripción, Dirección IP, Puerto TCP y Puerto SSL.

- En Descripción indicaremos el nombre con el que identificaremos nuestro sitio dentro de la consola, no se debe confundir con el nombre de dominio o URL que utilizamos para acceder al sitio Web a través de Internet. De esta forma podremos cambiar la denominación de sitio Web Predeterminado por la que deseemos
- El siguiente parámetro es la dirección IP que tiene asociada el sitio Web, podemos elegir una de las que tenga asignadas el servidor o bien elegir la opción Todos no asignados, si elegimos esta última opción estaremos asignando al sitio Web todas las direcciones IP del servidor que se encuentren libres. De esta forma pasa a ser el sitio Web predeterminado.
- El número de puerto normalmente no será necesario modificarlo, por defecto el servicio Web se encuentra en el puerto 80.
- El puerto SSL (Secure Sockets Layer) sólo se debe utilizar para conexiones seguras, basados en certificados de cliente y servidor.
- podremos indicar el tiempo de espera máximo (time-out) utilizado para establecer una conexión, esto

aseguraré que se cerrarán todas las conexiones si el protocolo HTTP no puede establecer una conexión desde el cliente en el tiempo especificado, por defecto son 900 segundos.

- Podremos activar el registro de nuestro sitio Web seleccionando la opción correspondiente. Tenemos diferentes formatos para guardar la información relativa los accesos a nuestro sitio Web. Cada uno de estos formatos definen que datos se van a almacenar y que estructura va a tener el fichero o tabla en la que van a ser almacenados.

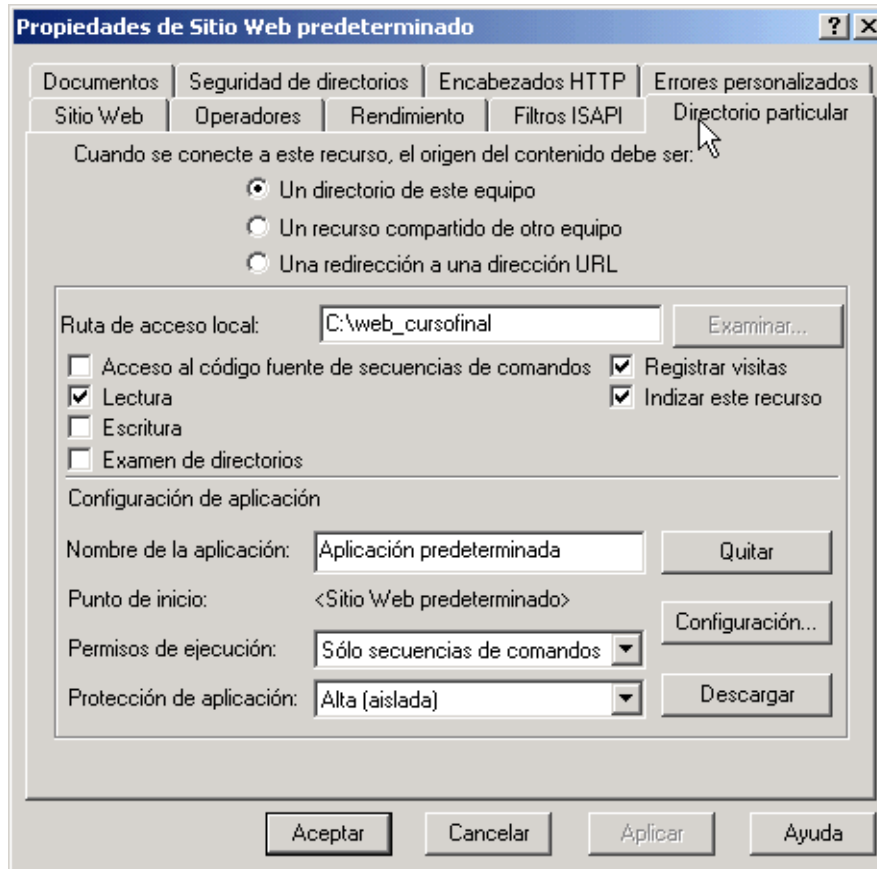
## 7. Los directorios virtuales

Para empezar, al configurar los sitios Web debe indicar los directorios que contienen los documentos que desea publicar. El servidor Web no puede publicar documentos que no están en los directorios especificados. Por lo tanto, el primer paso para desarrollar un sitio Web debe ser determinar cómo deseamos organizar los archivos.

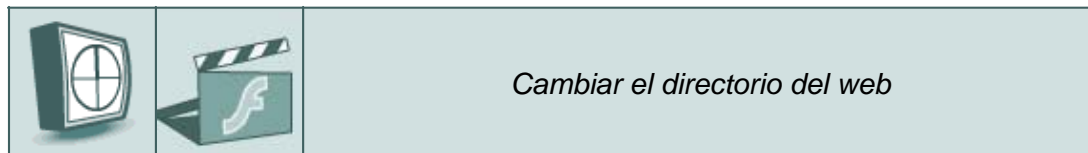
Si deseamos empezar ahora mismo sin tener que crear una estructura especial de directorios y todos los directorios se encuentran en el mismo disco duro del equipo que ejecuta los Servicios de Internet Information Server, podemos publicar los documentos inmediatamente copiando los archivos Web en el directorio principal predeterminado, C:\InetPub\Wwwroot.

Cada sitio Web o FTP debe tener un directorio particular: el directorio particular es la ubicación central de las páginas publicadas. Contiene una página principal o archivo de índice que da la bienvenida a los clientes y contiene los vínculos a otras páginas del sitio. El directorio particular se asigna al nombre de dominio del sitio o al nombre del servidor. Por ejemplo, si el nombre del dominio de Internet del sitio es [www.microsoft.com](http://www.microsoft.com) y el directorio particular es C:\Website\Microsoft, los exploradores utilizan la dirección URL <http://www.microsoft.com> para tener acceso a los archivos del directorio particular. En una intranet, si el nombre del servidor es ServidorActual, los exploradores utilizan la dirección URL <http://ServidorActual> para tener acceso a los archivos del directorio particular.

El directorio particular predeterminado se crea al instalar los Servicios de Internet Information Server y crear un sitio Web nuevo. Veamos la página de propiedades "Directorio particular"



En este caso las páginas web las está recogiendo del directorio "c:\web\_cursofinal". Podemos hacer ahora una prueba: crea una página web sencilla desde FrontPage y guárdala en un directorio del disco duro. Si elegimos este directorio en esta página propiedades veremos que por fin nos funciona nuestro sitio web. Podremos ver esta página de ejemplo... Una vez probado volvemos a seleccionar nuestro directorio c:\inetpub\wwwroot. Ya veremos el resto de las opciones en el capítulo siguiente. Terminemos con nuestros directorios virtuales...



## 7.1 ¿Qué es un directorio virtual?

Para publicar desde cualquier directorio que no esté contenido en el directorio particular, debe crear un directorio virtual. Un directorio virtual es un directorio que no está en el directorio particular pero que aparece en los exploradores de los clientes como si estuviera.

Es decir: sabemos que podemos poner páginas web en el directorio "c:\inetpub\wwwroot" y que estas se van a ver inmediatamente escribiendo en el navegador "http://servidor". Un directorio virtual es crear un directorio en nuestra página web pero que apunta físicamente a otra ruta física del disco duro diferente. Por ejemplo podemos crear un directorio virtual que llamaremos "docs" y que apunta a la ruta física "d:\intranet\documentos" para ver esto en el servidor bastaría con escribir: "http://servidor/docs"

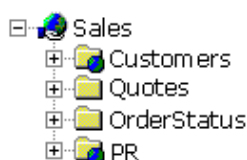
Un directorio virtual tiene un alias, un nombre que los exploradores Web utilizan para tener acceso al

directorio. Puesto que el alias suele ser más corto que el nombre de la ruta del directorio, a los usuarios les resulta más cómodo escribirlo. Un alias es más seguro; los usuarios no conocen el lugar del servidor donde están ubicados físicamente los archivos y no pueden utilizar esa información para modificar los archivos. Con los alias es más fácil mover los directorios en el sitio: En lugar de cambiar la dirección URL del directorio, puede cambiar la asignación entre el alias y la ubicación física del directorio.

Otro ejemplo, supongamos que configuramos un sitio Web para el grupo de mercadotecnia en la intranet de la compañía. La tabla siguiente muestra las asignaciones entre las ubicaciones físicas de los archivos y las direcciones URL a través de las cuales se obtiene acceso a los archivos.

Ubicación física	Alias	Ruta de la dirección URL
C:\Inetpub\wwwroot	<i>directorio particular (ninguno)</i>	http://Ventas
\\Servidor2\DatosVentas\ClientesProd	Clientes	http://Ventas/Clientes
C:\Inetpub\wwwroot\Presupuestos	<i>Ninguno</i>	http://Ventas/Presupuestos
C:\Inetpub\wwwroot\EstadoPedidos	<i>Ninguno</i>	http://Ventas/EstadoPedidos
D:\Mrcdctn\PR	PR	http://Ventas/PR

Los directorios virtuales y los físicos (directorios sin alias) aparecen en el complemento Servicios de Internet Information Server. Un directorio virtual viene indicado por un icono de carpeta con un globo en la esquina. Mira esta imagen que siguiente muestra el sitio Web del ejemplo anterior, /Clientes y /PR son directorios virtuales:



En un sitio Web simple, puede que no necesitemos añadir directorios virtuales. Basta con colocar todos los archivos en el directorio particular (directorio principal) del sitio. Si tienes un sitio complejo o quieres especificar diferentes direcciones URL para distintas partes del sitio, puedes agregar tantos directorios virtuales como necesites. En cualquier caso es un tema interesante.

A menudo nos encontraremos con que instalamos una utilidad para nuestra Intranet y la documentación la instala directamente en un directorio virtual, de esta forma está mas fácil y accesible que un fichero .htm tradicional.

## 7.2 Crear directorios virtuales

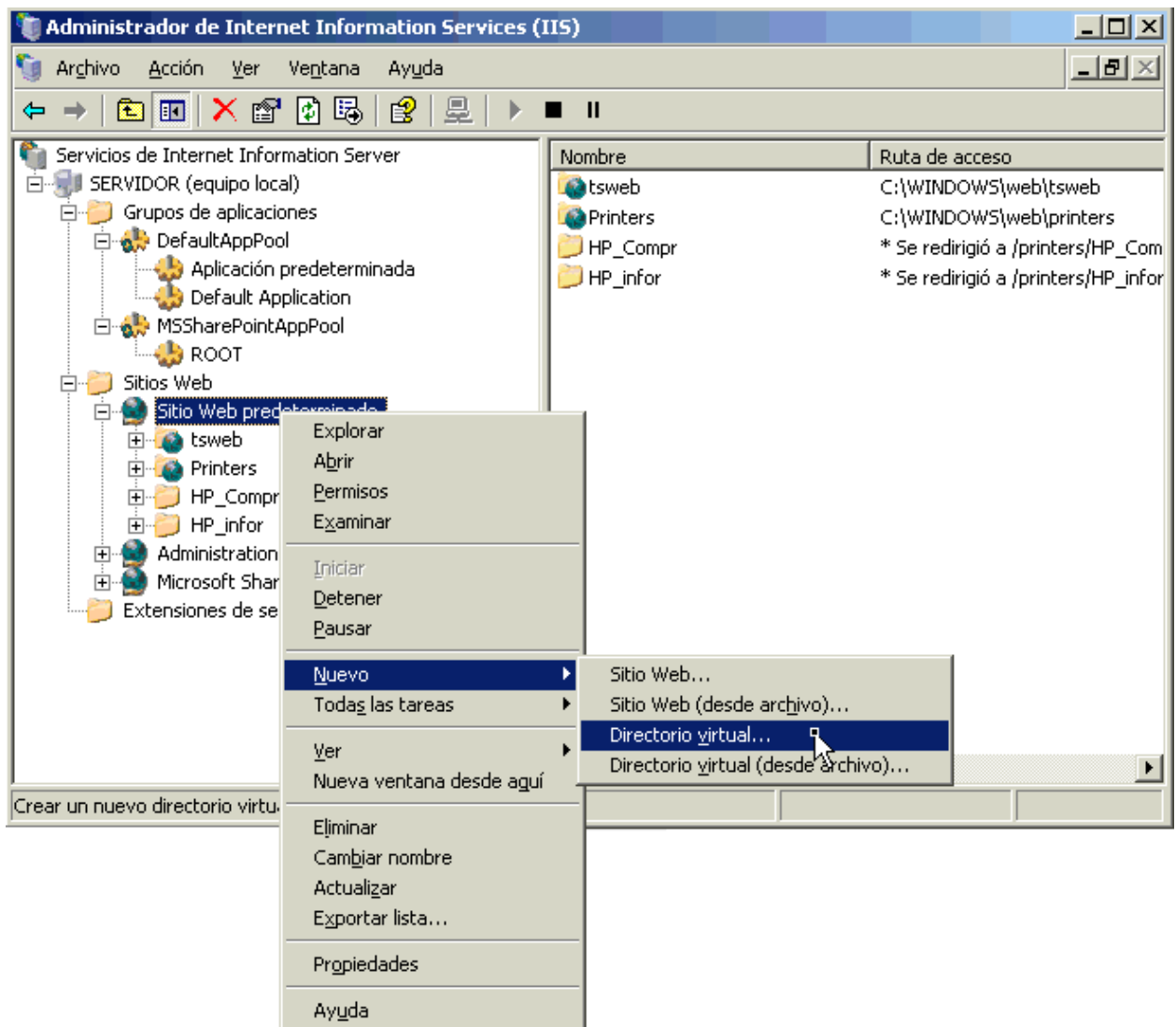
Si nuestro sitio Web contiene archivos que se encuentran en un directorio diferente del directorio particular o en otros equipos, debemos crear directorios virtuales para incluir esos archivos en el sitio Web. Para utilizar un directorio de otro equipo, debemos especificar su nombre según la nomenclatura universal (UNC) e indicar un nombre de usuario y una contraseña para los permisos de acceso.

Para crear un directorio virtual

1. En la consola administrativa, seleccionamos el sitio Web o FTP al que deseamos agregar un directorio.
2. Hacemos clic en el menú Acción, seleccionamos Nuevo y hacemos clic en Directorio virtual.

## Internet Information Server y ASP.NET

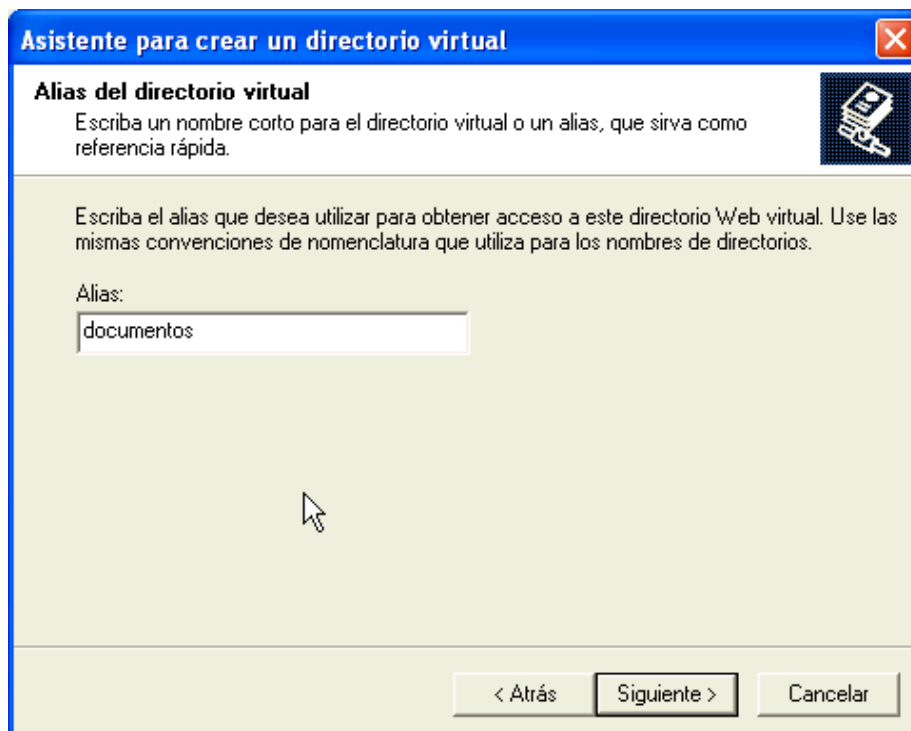
3. Utilizamos el Asistente para crear un directorio virtual para completar esta tarea.



A continuación de la pantalla de bienvenida:

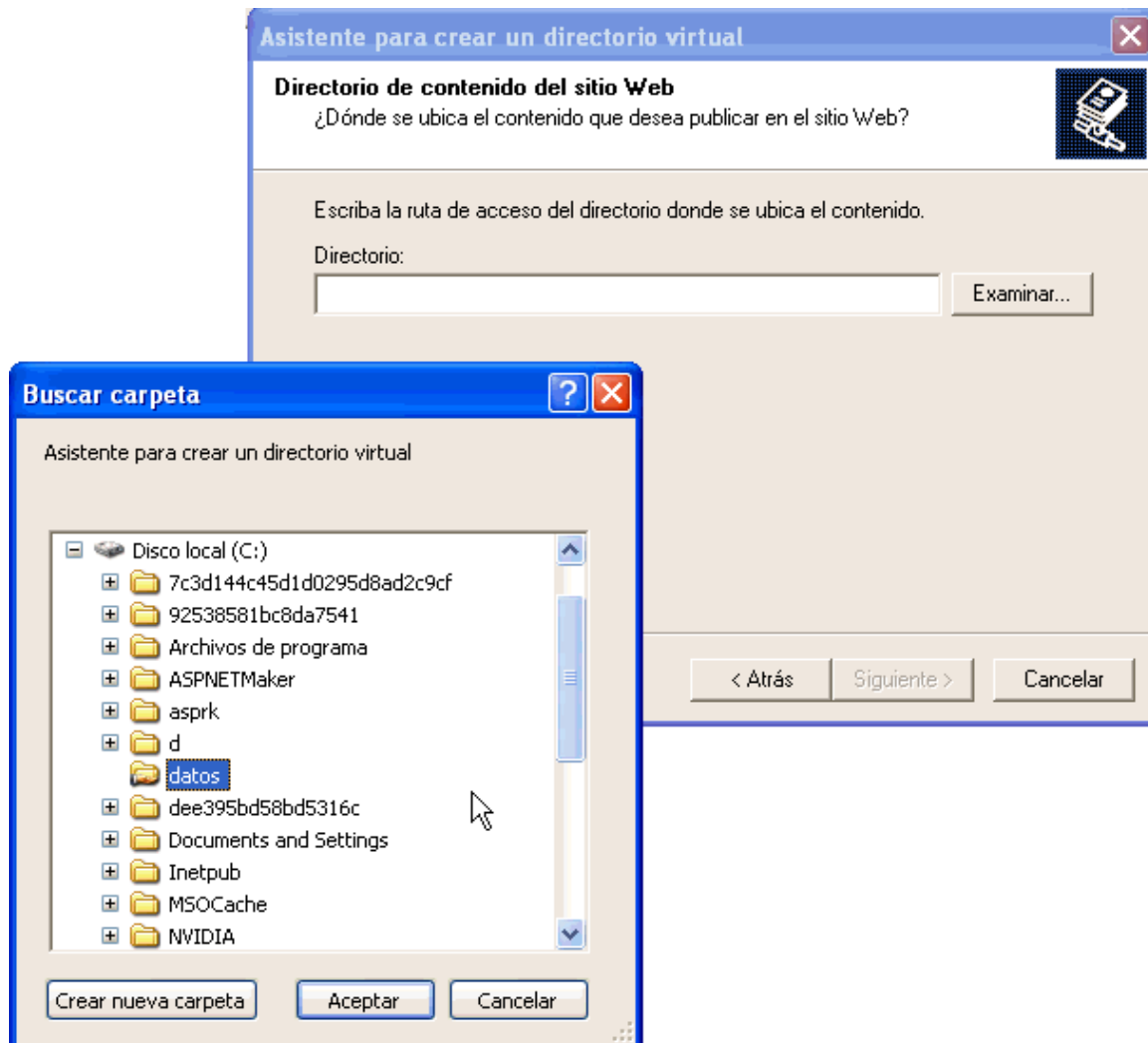


aparecerá una ventana preguntando por el alias que tendrá el sitio Web, es decir el nombre que escribiremos en el navegador: `http://servidor/alias`:

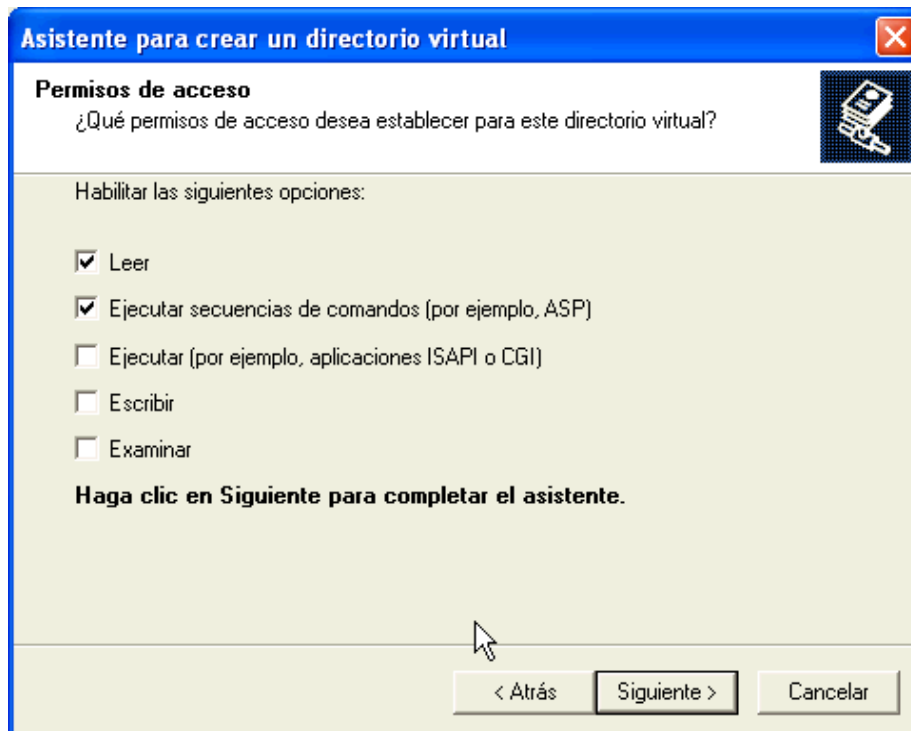


Luego tendremos que asociarlo con una ruta física del disco duro, seleccionamos un directorio o ruta:



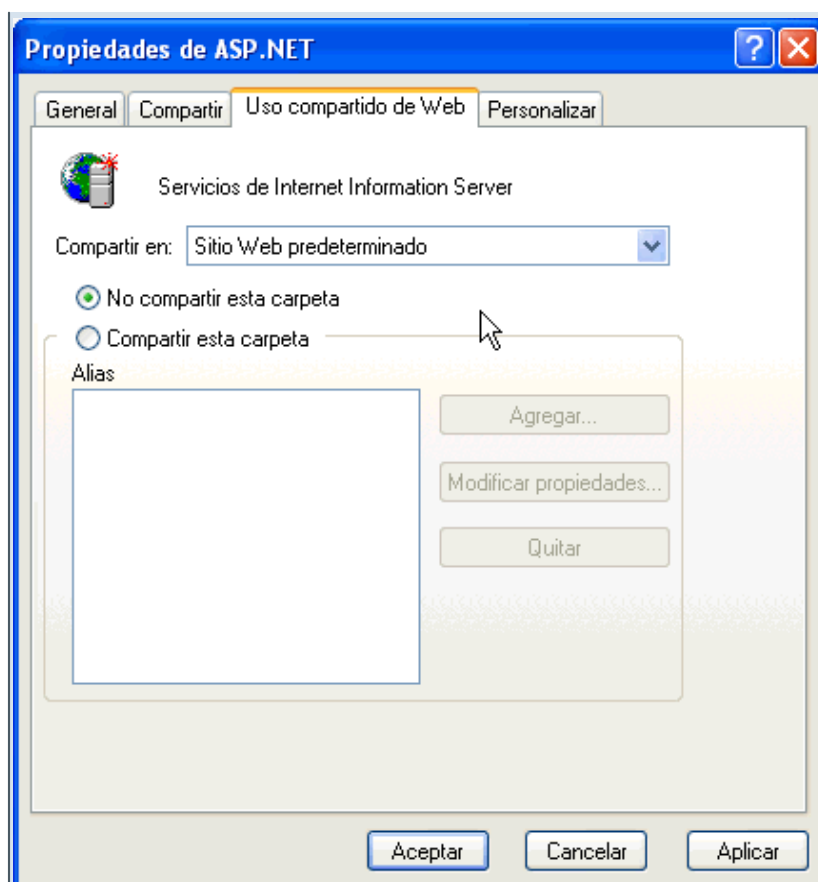


y finalmente pulsamos en "siguiente". Por último indicaremos los permisos de ejecución para este directorio:



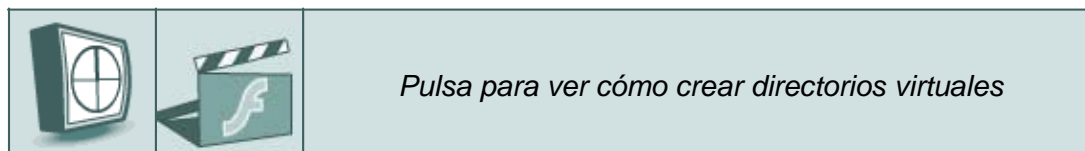
Estos permisos los veremos con detalle en el siguiente capítulo.

**Nota** Si utilizas NTFS, también puede crear un directorio virtual si haces clic con el botón derecho del ratón en un directorio en el Explorador de Windows, hace clic en Compartir y, después, selecciona la hoja de propiedades Uso compartido de Web.



Para eliminar un directorio virtual...

1. En la consola administrativa, seleccionamos el directorio virtual que queremos eliminar.
2. Hacemos clic en el menú Acción y, a continuación, en Eliminar. Al eliminar un directorio virtual no se elimina el directorio o los archivos físicos correspondientes sólo esta conexión lógica así que tranquilo



A continuación veremos con detalle en las opciones avanzadas de nuestro servidor web: Internet Information Server. Una vez configurado correctamente veremos su integración con FrontPage. En ese punto prácticamente realizaremos todo el trabajo con este programa y no será necesario la utilización de la administración de IIS para ninguna opción, exceptuando los cambios de funcionamiento de IIS.

Para ver todas sus opciones haremos como el capítulo anterior, veremos las fichas de la página de propiedades e iremos profundizando en cada una de las opciones. Finalmente dedicaremos una sección a la integración de FrontPage con Internet Information Server.

**Nota** Microsoft FrontPage es el editor de sitios Web de Microsoft, es una parte de la suite Microsoft Office 2000 Premium y un componente adicional de Microsoft Office XP y 2003. Para esta intranet utilizaremos este editor y se recomienda encarecidamente su utilización por la integración que tienen este producto con el servidor web IIS. (sobre todo la versión 2003)

## Internet Information Server y ASP.NET

A partir de la versión de Office 2.007 dejó de llamarse FrontPage para llamarse SharePoint Designer. Producto mucho mas maduro y que ha puesto tal renovación que le han puesto nombre distinto. Tienes información y la posibilidad de descargar una prueba en:

<http://office.microsoft.com/es-es/sharepointdesigner/FX100487633082.aspx>

Y como Microsoft no es nadie vendiendo ni anunciando cosas en esta página tienes las ventajas de SharePoint respecto a FrontPage:

<http://office.microsoft.com/es-es/sharepointdesigner/FX101759003082.aspx>

Personalmente he escrito este curso con SharePoint porque son todo páginas HTML normales. Cuando quiero páginas con ASP.NET utilizo el entorno de desarrollo integrado de Visual Web Developer ya que cada uno es para lo que es. El primero es una tremenda evolución de FrontPage y el segundo es una versión de Visual Studio especialmente preparada para escribir ASP.NET

# Ejercicios

Nota: Los dos primeros ejercicios no son obligatorios porque podemos pasar el curso sin instalar IIS aunque es absolutamente recomendable. Ya que podemos utilizar el servidor web que incorpora el ASP.NET Visual Developer. Este servidor web es muy sencillo y solo vale para probar las páginas pero si no tenemos posibilidad de instalar un IIS en un servidor podemos hacer el curso completo con él.

## Ejercicio 1

Comprueba que IIS está instalado y aparece la pantalla siguiente al poner en el navegador el nombre del servidor, por ejemplo <http://servidor/>



El sitio al que intentó conectarse no tiene en este momento una página predeterminada. Es posible que esté en proceso de actualización.

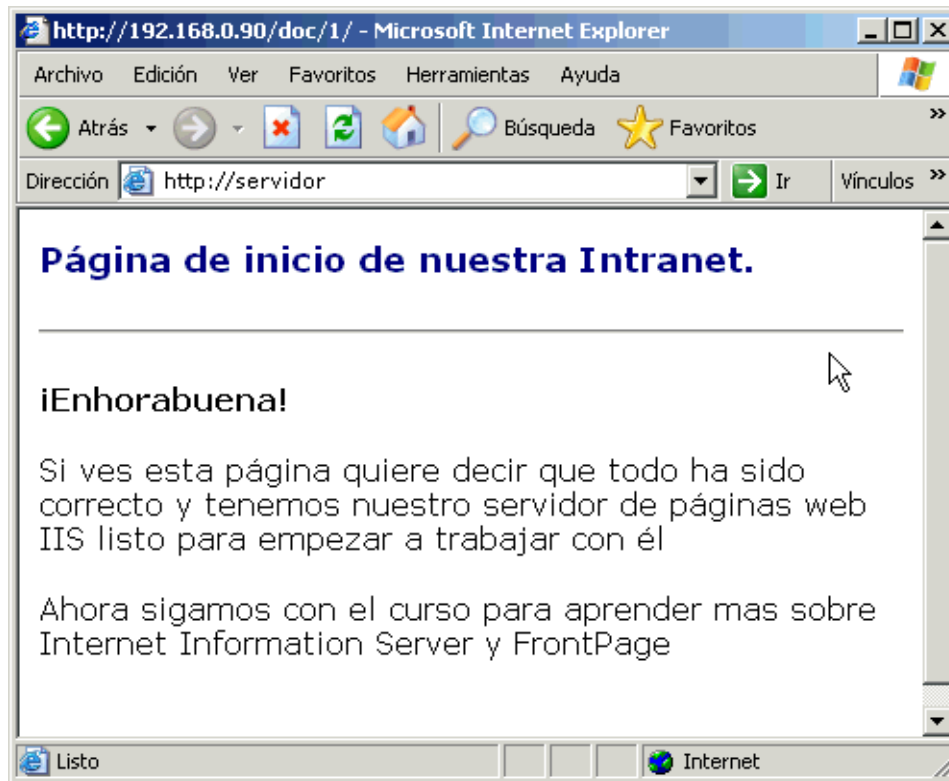
---

Inténtelo de nuevo más tarde. Si el problema continúa, póngase en contacto con el administrador del sitio Web.

## Ejercicio 2

Copiar la página de pruebas en el directorio de la web. [Haz clic aquí para descargarla](#)

Comprobar que se ve correctamente la página desde todos los equipos nota: por defecto es el <c:\inetpub\wwwroot>



### Ejercicio 3

Descarga e instala el Visual Web Developer 2.008 con todas las opciones: MSDN para la ayuda y SQL Server 2.005 Express Edition para el servidor de base de datos