

Programación de dispositivos móviles

Unidad IV

4.- Construcción de aplicaciones de Internet inalámbricas

Computación Móvil

Se puede definir la Computación Móvil como la serie de artefactos y equipos portátiles, hardware, que hacen uso de la computación para lograr su funcionamiento, así, se tiene a las computadoras portátiles, los teléfonos celulares, los cuadernos de notas computarizados, las calculadoras de bolsillo, etc.

WML y HDML

WML ("Wireless Markup Language") y HDML ("Handheld Device Markup Language") son los dos lenguajes de marcación que son compatibles con la gran mayoría de aparatos inalámbricos en el mercado, HDML fue desarrollado por phone.com (hoy openwave.com), mientras WML fue creado por WAP Forum en 1997 por Erricson, Nokia, Motorola...y otras 250 empresas en el medio inalámbrico. Hoy en día generalmente se utiliza WML, ya que casi *todos* los navegadores en aparatos inalámbricos logran interpretar apropiadamente el lenguaje.

WML es un poco diferente que HTML pero esencialmente cumple las mismas funciones, enviar el contenido a pantalla.

WML

- Wml es el lenguaje empleado en WAP que a su vez consiste en un protocolo que permite el envío de datos por telefonía móvil.
- Da así acceso a Internet, siguiendo la arquitectura de Cliente-Servidor.
- La programación en wml hereda la mayor parte de sus construcciones sintácticas de xml. Este es un rasgo en común con html.
- El salto entre ambos lenguajes es prácticamente inmediato.

Estructura

BARAJA: El servidor de información, tras una petición del terminal WAP, envía una “baraja”, que corresponde a una unidad de información. Es el equivalente a una página Web en Internet. Generalmente esta baraja es de un tamaño pequeño, cercano al Kbyte, debido a las restricciones que impone la comunicación inalámbrica.

- **CARTA:** Un conjunto de cartas forma la baraja. Cada carta contiene información de formatos, contenidos visibles en la pantalla del terminal instrucciones a procesar. Idealmente, los contenidos visibles se ajustan a la pantalla. De no ser así nos desplazaremos con un “scroll”, mediante cursores o una rueda, en función del terminal. La idea de crear las cartas es intentar dividir la información en varias sub-páginas para presentarla de forma ordenada en la pantalla del terminal.

Cada carta de una baraja debe contener uno o más elementos.

- **ELEMENTOS:** El código de una aplicación programada en wml está formado por una serie de elementos. Se distinguen dos tipos de elementos.

El primero se identifica como un bloque:

<elemento>

.

. contenido del elemento

.

</elemento>

Ejemplo

Manejo de contenidos

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card>
<p align="center" mode="wrap">
Hola Mundo
</p>
</card>
</wml>
```

Estructura

- El primer bloque de este programa debe incluirse siempre. Es un prólogo que contiene la versión de XML que se va a utilizar y un puntero a la definición XML del lenguaje. Este prólogo lo utilizan los compiladores y servidores WAP para crear y ejecutar los programas. En cambio los terminales no ven siquiera esta parte del código.
- El segundo bloque es el de la baraja., delimitada por las etiquetas <wml>.
Se distingue también la carta dentro de sus etiquetas <card>, </card>, así como el párrafo: <p>, </p>.
Nótese que se define para este último su aspecto mediante lo que se denomina “atributos”, en este caso su posición en la línea (centrado), y el modo de presentación (envuelto), que asegura que el párrafo no desbordará por la derecha la pantalla (siguiendo en la línea siguiente). De estar definido como desenvuelto (nowrap), se deberá disponer de un mecanismo que permita ver la porción de línea del párrafo que no cabe en la pantalla.

Ejemplo2

```
<?xml version="1.0"?>  
  <!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">
```

```
<wml>  
  <card id="First_Card">  
    <do type="accept" label="Next">  
      <go href="#Second_Card"/>  
    </do>  
    <p>  
      Select <b>Next</b> to display the next card.  
    </p>  
  </card>  
  <card id="Second_Card">  
    <p>  
      This card contains the following:...  
    </p>  
  </card>  
</wml>
```

Pantalla de resultados



- Esto define un deck que contiene 2 cards, de tal forma que cuando pulsemos Next nos iremos a la segunda card, en la cual podremos elegir la opción Back para volver a la anterior.
- El objeto y propiedades de card se definirán más adelante.

El formato de los textos

Lo primero que debemos aprender es el formato de los textos, para conseguir una presentación aceptable de nuestros datos, cosa difícil debido al pequeño espacio que se nos proporciona en el navegador WML.

TODOS los textos deben ir dentro de una etiqueta **<p> Textos....</p>**

Ej.:El código **Holaa todos --> **Hola** a todos // **<i>Hola</i>a todos --> **Hola** a todos****

Algunas Aplicaciones de la Computación Móvil

A pesar de que las novelas de ciencia ficción lo anticipaban, no deja de sorprender la diversidad de aplicaciones que tiene hoy en día —y puede tener en el futuro— esta tecnología. Hay varios campos en los que no sólo es de utilidad, sino de competencia esencial para las empresas. Aquí algunos ejemplos:

Manejo de pacientes. La computación móvil permite al médico o institución mantener contacto con un paciente cuyo estado requiere continua vigilancia. Y esto no se limita al envío o recepción de mensajes (hablados o escritos), sino que incluye también el monitoreo constante de signos vitales críticos que pueden anticipar una emergencia.

Ventas directas. Posiblemente ésta sea la aplicación más evidente: poder consultar inventarios, precios y realizar pedidos en forma inmediata resulta de particular interés para cualquier empresa que se dedique a la comercialización de productos.

Servicio a clientes. La asesoría, servicio técnico y consultoría es una área en donde la computación móvil es vital. La consulta a bancos de información, bases de datos inteligentes, acopio de información actualizada y consulta de especialistas, es sólo una pequeña muestra de todo lo que puede impactar esta tecnología, sin mencionar la posibilidad de contacto permanente con el cliente.

Personal móvil en oficinas. No es raro encontrar a personal que, pese a encontrarse siempre en el mismo edificio, se mudan de lugar con frecuencia para, por ejemplo, dar soporte técnico al personal o revisar proyectos. La computación móvil no sólo les permite ser localizados con facilidad, sino que también le auxilia en la consulta de datos que por lo regular estarían en su oficina.

Profesionales viajeros. Los modernos guerreros del camino son usuarios potenciales que pueden aprovechar esta herramienta: contadores con los registros de una empresa bajo el brazo, gerentes regionales que integran metas empresariales, dirigentes corporativos que requieren información actualizada y, en fin, una gran cantidad de personas que necesitan viajar para realizar su trabajo.

Manejo de sucursales. En un mundo en el que las empresas han dejado de tener una sola oficina para expandirse, las sucursales aparecen casi sin desearlo, con una gran cantidad de datos que deben consolidarse.

Grupos de trabajo. La globalización y expansión de empresas hace que sea cada vez más común atacar proyectos con el personal adecuado, el cual no siempre trabaja bajo un mismo techo y, en ocasiones, ni siquiera en la misma ciudad o país.

Tendencias de la Computación Móvil

En fin, nunca se terminaría de enumerar el enorme número de áreas en que esta forma de computación puede cobrar una importancia vital. Pero no se crea tan a futuro, sólo en Canadá se reportaron durante 1993 ventas por \$1,000 millones de dólares y —según datos de Datacomm Research—, las ventas en Estados Unidos llegarán a \$3,700 millones de dólares en el año 2002 para la industria de computación móvil, en la que se incluye equipo, programas e infraestructura de comunicaciones (inalámbrica y servicios de redes).

Seguramente México experimentará una expansión proporcional de dicho mercado, nicho que ya empieza a ser atacado por algunas empresas. Sin embargo, en general representa una oportunidad de negocios en un mercado muy poco competido hasta ahora.

4.1 arquitectura de cliente delgado

Cliente delgado

Es un equipo de cómputo que no contiene procesador, memoria, discos duros, CD-ROM, Floppy, u otro dispositivo periférico que utiliza una PC convencional. No procesa nada de información. Contiene un chip que permite la conexión con un servidor central, mediante una tarjeta de red, conexiones a dispositivos USB (teclados, ratones, Memorias Flash USB, Discos duros externos, etc.), un monitor y conexiones de audio y video. Posee las siguientes propiedades:

Usualmente los clientes delgados no poseen dispositivos de almacenamiento. Esto es lo que hace que sean llamados clientes delgados o estaciones de trabajo sin disco. No teniendo dispositivos de almacenamiento local, se hace imperiosa la necesidad de contar con el boot de red para poder ser iniciada.

Adicionalmente, el no contar con estos dispositivos de almacenamiento local elimina también la necesidad de partes mecánicas móviles y rotatorias, lo que entre otros beneficios aporta además:

- Ausencia de ruidos, son totalmente silenciosos

- Menor consumo energético

- Muchas menos fallas

La arquitectura cliente delgado:

no trata solo de las funciones que cada quien realiza. También se tiene la funcionalidad de las distintas partes aquí es donde interviene el concepto cliente grueso/delgado.

Cuando tenemos una arquitectura cliente/servidor, también tenemos el concepto de capas lógicas dentro de la arquitectura, estas capas son áreas funcionales encapsuladas que realizan ciertas tareas.

Una capa es una área funcional distinta dentro de nuestra arquitectura pero una capa no existe por sí sola, tiene considerable interacción con otras capas, lo que por consecuencia crea la funcionalidad requerida para nuestra aplicación.

Capa de presentación: esta es la agradable GUI que se presenta al usuario. Esta capa contiene código que da formato y muestra datos de una manera fácil de leer. También captura la entrada de usuario. Esta capa contiene el código que da formato y muestra los datos de manera fácil de leer. También captura la entrada de usuario. Los datos capturados de la entrada de usuario se pasan a la capa lógica de negocios

Capa lógica de negocios: aquí se guardan las reglas de negocios para aplicación. Por ejemplo, si una persona debe ser mayor de 16 años esta capa verifica los datos y tiene éxito (y los pasa a la capa de datos) o falla y devuelve un error de capa de presentación.

Capa de administración de datos: encargado de hacer persistente toda la información, suministra y almacena información para el nivel de negocio

Beneficios de cliente delgado:

Si se requiere una modificación en el código, sólo necesita haber el cambio en un solo lugar.

Es muy fácil de implementar

Como la capa de presentación se ejecuta solo en la PC cliente, se necesitan muy pocos recursos.

Se basa lógicamente en la arquitectura cliente/servidor.

Administración de los equipos centralizada.

Menores Costos.

Bajo consumo de energía

Reducción en costos de mantenimiento.

Movilidad de sesiones de usuarios.

Actualizaciones de software y parches en un solo punto.

Desventajas:

Se incrementa el tráfico en la red, ya que cada solicitud requiere de un viaje al servidor.

Se necesitan servidores costosos para ejecutar y administrar la aplicación.

Por lo general, no solo esta disponible el acceso al sistema operativo, por lo que perdemos esta funcionalidad.

La seguridad puede llegar a convertirse en una pesadilla.

La base de código puede ser muy extensa y difícil de mantener.

4.2. - EL MODELO WAP

El modelo de programación WAP es similar al modelo de programación WWW. Este provee varios beneficios a la comunidad desarrolladora de la aplicación, incluyendo un modelo de programación familiar, una arquitectura probada, y la capacidad de mejora las herramientas ya existentes (ej. Servidores Web, XML, etc). Se han hecho optimizaciones y extensiones con el fin de encajar las características del medio ambiente inalámbrico. Donde sea posible, se han adoptado los estándares existentes o han sido usados como el punto de arranque para la tecnología WAP.

WAP define un conjunto de componentes estándares que permiten la comunicación entre los terminales móviles y los servidores de red, incluyendo

Modelo de nombre estándar - Los URLs estándar WWW son usados para identificar el contenido WAP en los servidores originales. Los URLs estándar WWW se usan para identificar fuentes locales en un dispositivo (ej. Funciones de control de llamadas)

Tipo del contenido - Todos los contenidos WAP tienen un tipo específico consistente con el tipo WWW. Esto permite que los agentes usuarios WAP procesen correctamente el contenido basado en su tipo.

Formatos de contenido estándar - Los formatos de contenido WAP están basados en tecnología WWW e incluyen marca de visualización, información de calendario, objetos de tarjetas de negocios electrónicas e imágenes.

Protocolos de comunicación estándar - Los protocolos de comunicación WAP habilitan la comunicación de la petición del browser de el terminal móvil al servidor web de la red.

Los protocolos y tipos de contenido WAP han sido optimizados para el mercado de masas, dispositivos inalámbricos de mano. WAP utiliza tecnología Proxy para hacer la conexión entre el dominio inalámbrico y el WWW. El proxy WAP está típicamente compuesto de la siguiente funcionalidad:

El Gateway del protocolo - Traduce las peticiones de la pila del protocolo WAP (WSP, WTP, WTLS, y WDP) a la pila del protocolo WWW (HTTP y TCP/IP).

Decodificadores y codificadores de contenido - Estos traducen el contenido WAP en formatos codificados compactos para reducir el tamaño de la data sobre la red.

COMPONENTES DE LA ARQUITECTURA WAP

La arquitectura WAP provee un ambiente escalable y extensible para el desarrollo de aplicaciones para dispositivos de comunicaciones móviles. Esto se logra a través de un diseño de capas de la pirámide completa del protocolo. Cada una de las capas de la arquitectura es accesible por las capas superiores, así como por otros servicios y aplicaciones.

Capa de aplicaciones (WAE)

Capa de sesión (WSP)

Capa de transición (WTP)

Capa de seguridad (WTLS)

Capa de transporte (WDP)

Ambiente de aplicación inalámbrico (WAE)

Este es un ambiente de aplicación de propósito general basado en una combinación de las tecnologías WWW y de telefonía móvil. El objetivo primario es establecer un ambiente interoperable que permitirá a los operadores y a los proveedores de servicio construir aplicaciones y servicios que puedan alcanzar una amplia variedad de plataformas inalámbricas diferentes de una manera útil y eficiente. WAE incluye un ambiente micro-browser que contiene la siguiente funcionalidad:

Wireless Markup Language (WML) – un lenguaje ligero, similar a HTML, pero optimizado para el uso de terminales móviles de mano.

WMLScript – un lenguaje de escritura ligera, similar a Java Script.

Aplicación de telefonía inalámbrica (WTA, WTAI) – interfaces de servicios de telefonía y de programación.

Formatos de contenido – un conjunto de formatos de data bien definidos, incluyendo imágenes, registros de libros de teléfonos e información de calendario. **Protocolo de sección inalámbrica (WSP).**

4.3 Desarrollo de Cliente Delgado

La creación de clientes delgados exitosos requiere al menos estas características:

- Sistema operativo abierto
- Arranque remoto
- TCO reducido
- Seguridad alta
- Componentes de hardware standard
- Configuración sencilla
- Facilidad de uso
- Administración centralizada
- Bajo consumo de ancho de banda

4.4.- tecnologías de generación de contenidos y lenguajes inalámbricos

Tras una presentación de las características más relevantes de la tecnología WAP (Protocolo para Aplicaciones Inalámbricas), y justificando la importancia crucial que en este ámbito cobra la personalización de contenidos ofrecidos, en el presente estudio se analizan los distintos modelos y tecnologías de procesamiento en servidor existentes (CGI, xSAPI, Servlets Java...etc) reflejando el "estado del arte" actual en lo referente a la generación dinámica de contenidos WAP. Además, se hace hincapié en otro importante aspecto de la personalización: la negociación del formato de presentación de los contenidos para su correcta visualización en terminales móviles concretos, presentando para ello una solución avanzada que se basa en la utilización de perfiles CC/CP y la transformación de documentos XML mediante XSL.

Técnicas de procesamiento en servidor WWW para generación de contenidos WAP

Como se puede apreciar, el modelo general de operación cliente/servidor de WAP es muy similar al usado en el dominio WWW, y, de hecho, aquél ha sido específicamente diseñado para que sea posible aprovechar la infraestructura tecnológica existente en la Web para el aporte de contenidos.

Obsérvese además que en tanto se accede en última instancia a un servidor Web estándar, los contenidos WAP devueltos pueden ser estáticos (tienen existencia previa en un sistema de archivos local o remoto accesible por el servidor) o generados dinámicamente usando tecnologías de servidor suficientemente probadas en la Web, sin que el cliente WAP pueda establecer distinción alguna sobre el mecanismo de producción de contenidos usado realmente (tal como sucedía en los clientes Web).

En general, nunca se debería confiar en la efectividad de procesos de filtrado y conversión automática de contenidos Web a formato WAP (posibilidad anteriormente mencionada), debido a que estos filtros poco pueden hacer para adaptar al 100% contenidos que no tienen en cuenta -en origen- las características físicas especiales de los terminales móviles que realmente se van a usar como clientes. Significa esto que es siempre preferible aportar contenidos directamente en formato WAP (WML, WMLScript.etc).

Practica

Configuración de los IIS

Herramientas

- Instalar los IIS
- Disco de instalación de windows xp
- Descargar ASP.net WEB MATRIX
- Editor de texto para wap
- Emulador para wap

Entorno de dispositivos móviles con asp.net

windows xp

en esta practica nosotros utilizaremos asp . Net web matrix que es una versión gratuita del visual studio.net para desarrolladores principiantes.

Parte 1

instalación de los IIS

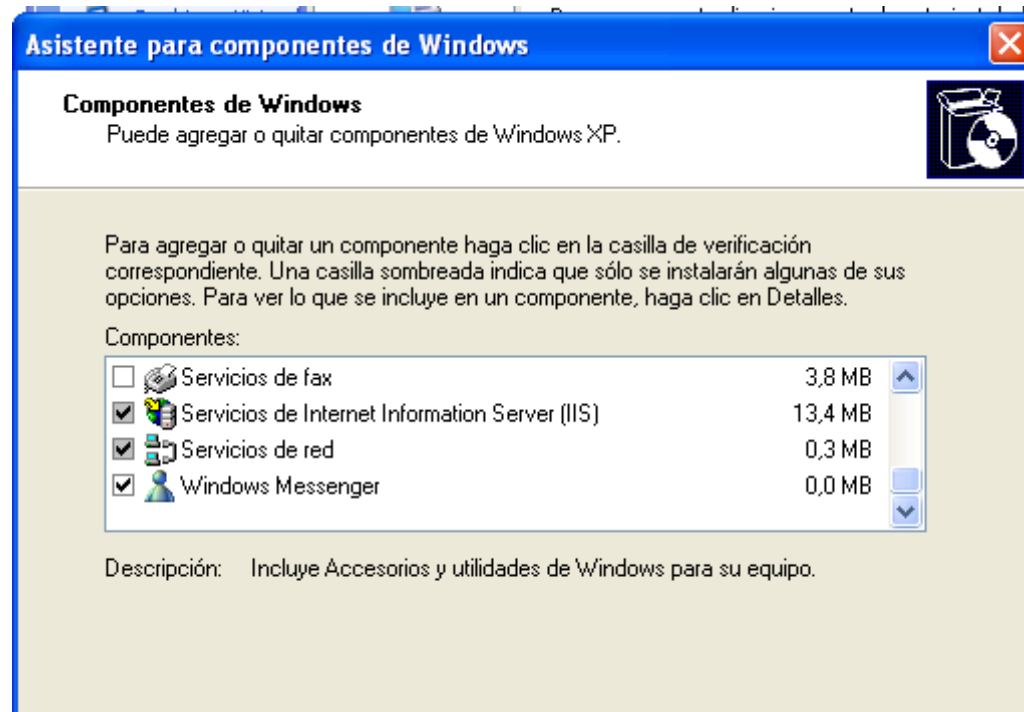
para debemos tener el disco de instalación ya sea de windows xp sp2

primero damos click en el botón de inicio seleccionamos panel de control



seleccionamos agregar o quitar programas y posteriormente damos click en componentes de windows

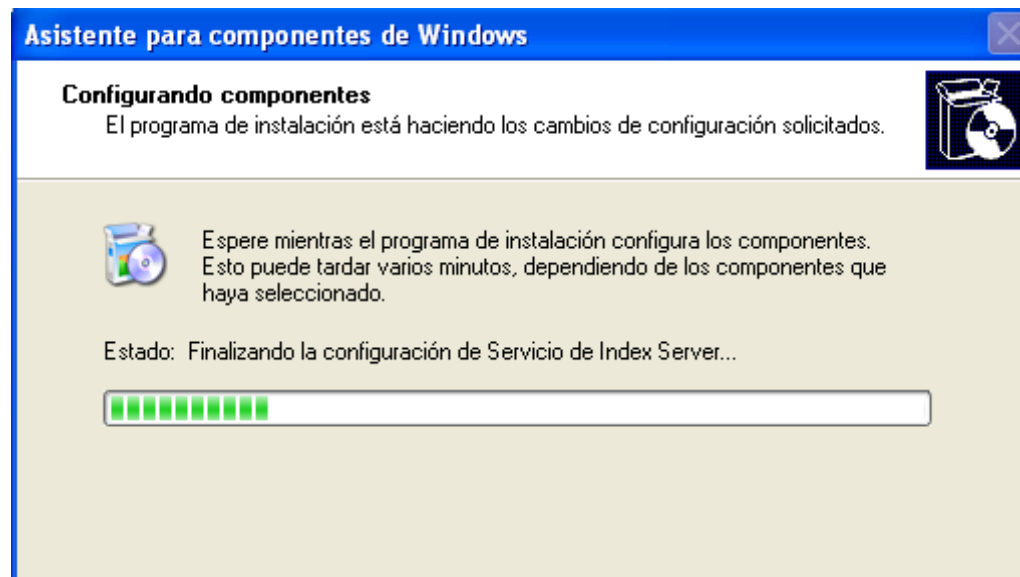
nos aparecerá una ventana como esta



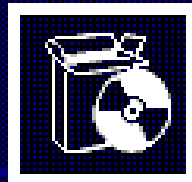
seleccionamos IIS y seguimos

es necesarios hacer hincapié en que varios archivos del sistema operativo serán copiados .

Por esa razón es necesario tener los discos de instalación, varios archivos se pueden bajar de Internet pero otros solo pueden ser copiados del disco de instalación .



Asistente para componentes de Windows



Finalización del Asistente para componentes de Windows

El Asistente para componentes de Windows se ha completado satisfactoriamente.

Para cerrar este asistente, haga clic en Finalizar.

< Atrás

Finalizar

Damos clic en el botón Finalizar, y retiramos nuestro CD de la unidad. Ahora abrimos nuestro navegador web favorito, y tecleamos en la barra de dirección: **http://localhost/** o la dirección IP **http://127.0.0.1**.

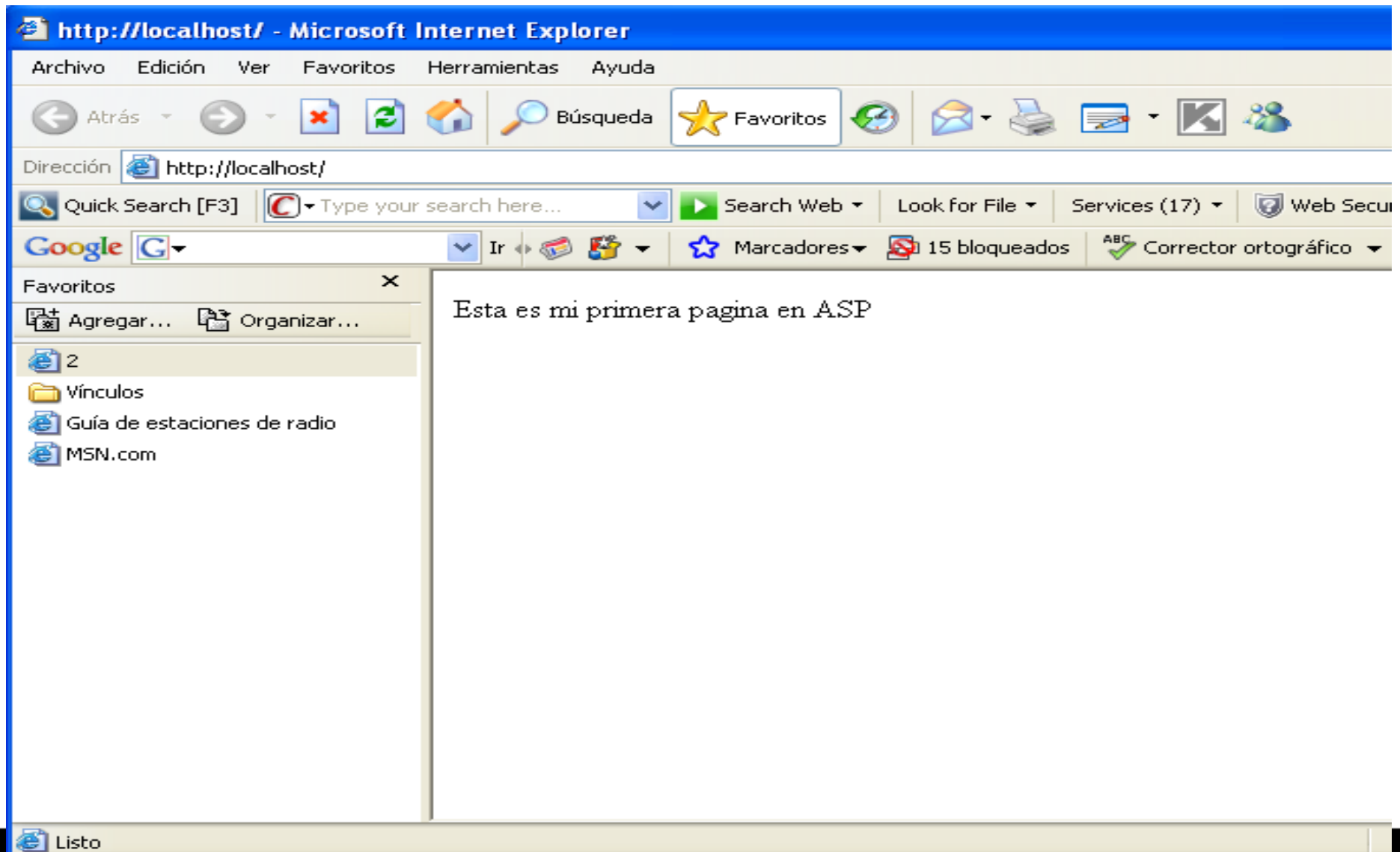
Nota: Algo muy importante, si tu navegador predeterminado es Internet Explorer, se te redireccionara a la dirección **http://localhost/localstart.asp** donde se te dará la bienvenida, y se abrirá un popup donde encontraras información del lenguaje ASP 3.0; si usas Firefox, Mozilla, Opera o Netscape , te aparecerá una ventana pidiendo un nombre de usuario y contraseña. Ignoremos esto, y nos dirigimos a donde estarán almacenados nuestros documentos, o sea, el directorio raíz del servidor, siendo la dirección referida **c:\inetpub\wwwroot** Ahí encontraremos dos archivos, **iisstart.asp** y **localstart.asp**, los cuales renombraremos a **_iisstart.asp** y **_localstart.asp**

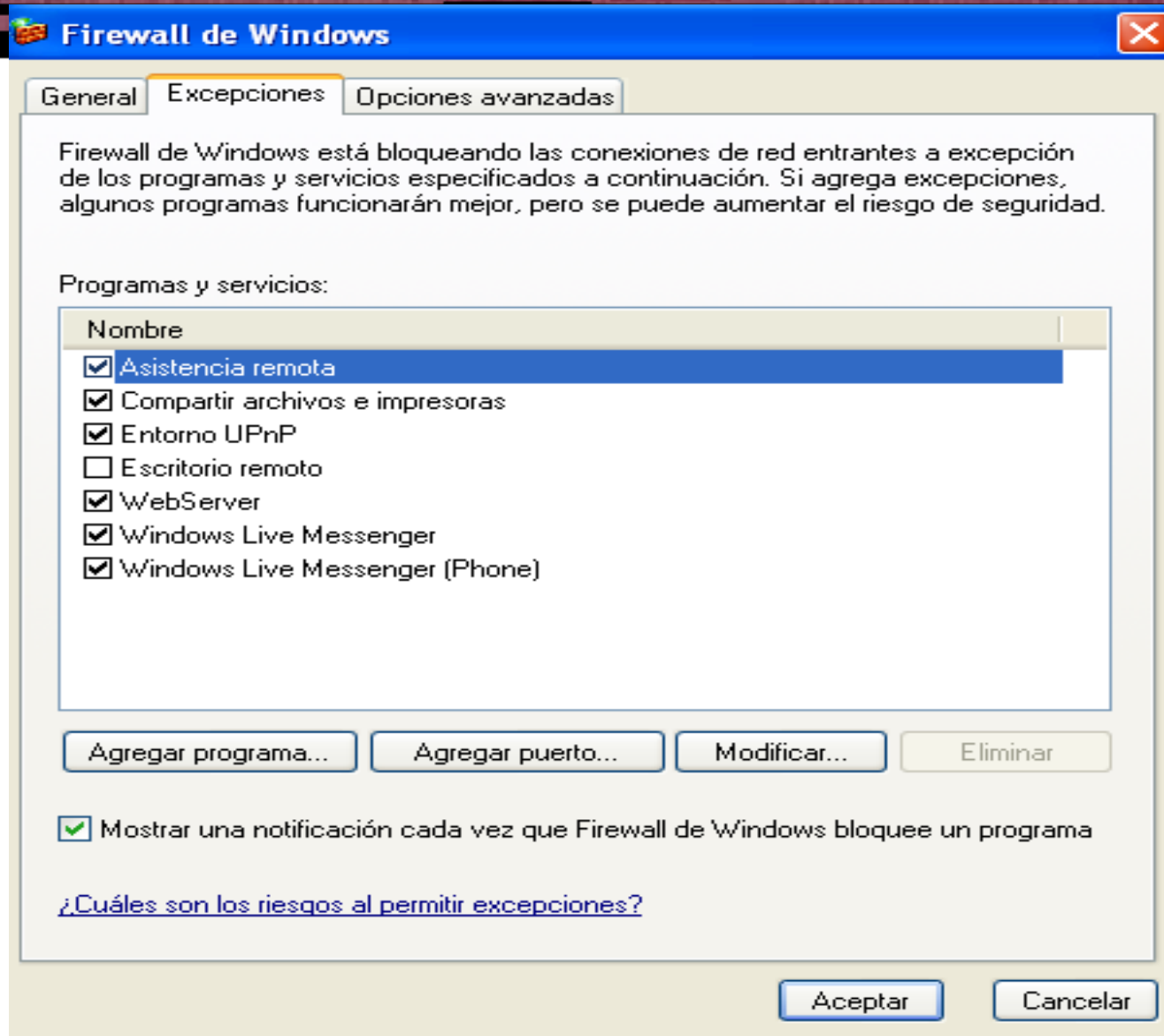
Ahora, creamos un archivo de texto plano (Por ejemplo Bloc de Notas) y escribimos lo siguiente:

```
<% Response.Write "Esta es mi primera pagina en ASP" %>
```

guardamos el archivo como **default.asp** en tipo de archivo seleccionamos todos los archivos y guardamos en esta dirección **c:\inetpub\wwwroot**

abrimos nuestro navegador y tecleamos localhost y nos tiene que aparecer que sera nuestra pagina de inicio





Así concluimos la instalación de IIS, y ya tenemos un servidor capaz de soportar las versiones, ASP 1, ASP 2 y ASP 3.

Servidor Apache + PHP + MySQL + phpMyAdmin

Servidor Apache + PHP + MySQL + phpMyAdmin

Para la instalación del Servidor Apache, he decidido utilizar el Paquete FoxServ 3.1 beta1, por ser el que menos me ha causado problemas (Sin el afán de demeritar otros paquetes que hacen esta misma función), incluyendo la instalación manual de cada paquete(Apache, PHP, MySQL y PhpMyAdmin).

Una vez descargado el **FoxServ3.1Beta1**. lo ejecutamos,y seguimos las instrucciones:

La primer ventana nos pedirá seleccionar el lenguaje de instalación, en este caso el idioma inglés es el único, así que pulsamos Next

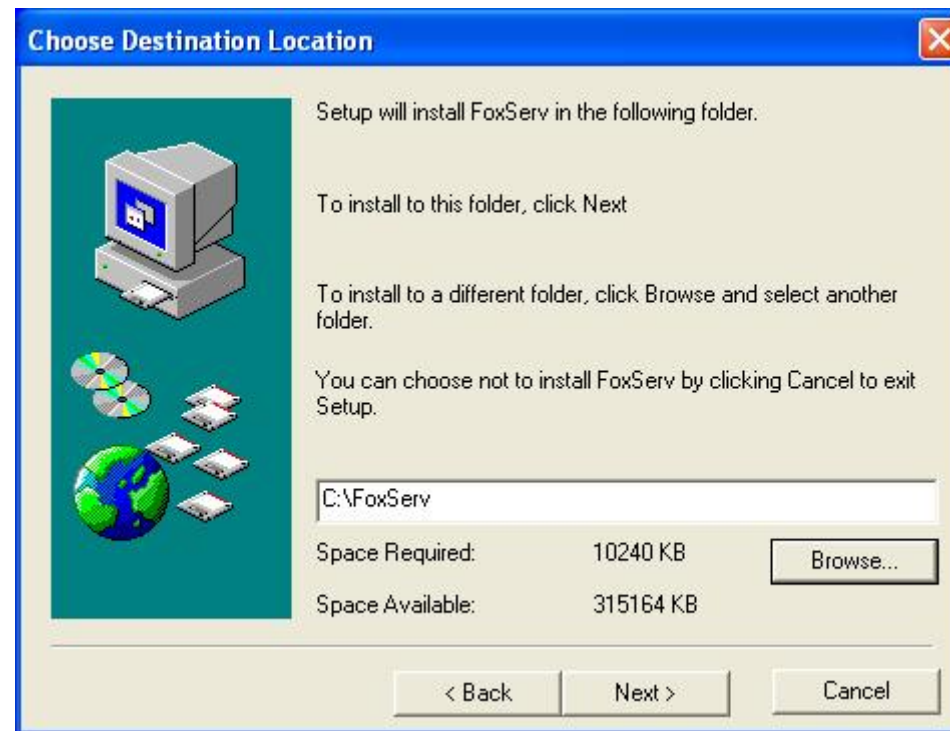
La segunda nos pregunta si queremos hacer la instalación, pulsamos Next

La tercera nos pedirá, la aceptación de licencia de uso, pulsamos Yes

La siguiente pantalla nos pide la ubicación de instalación, pulsamos Browse... y seleccionamos la unidad C: y pulsamos aceptar



aceptar



next

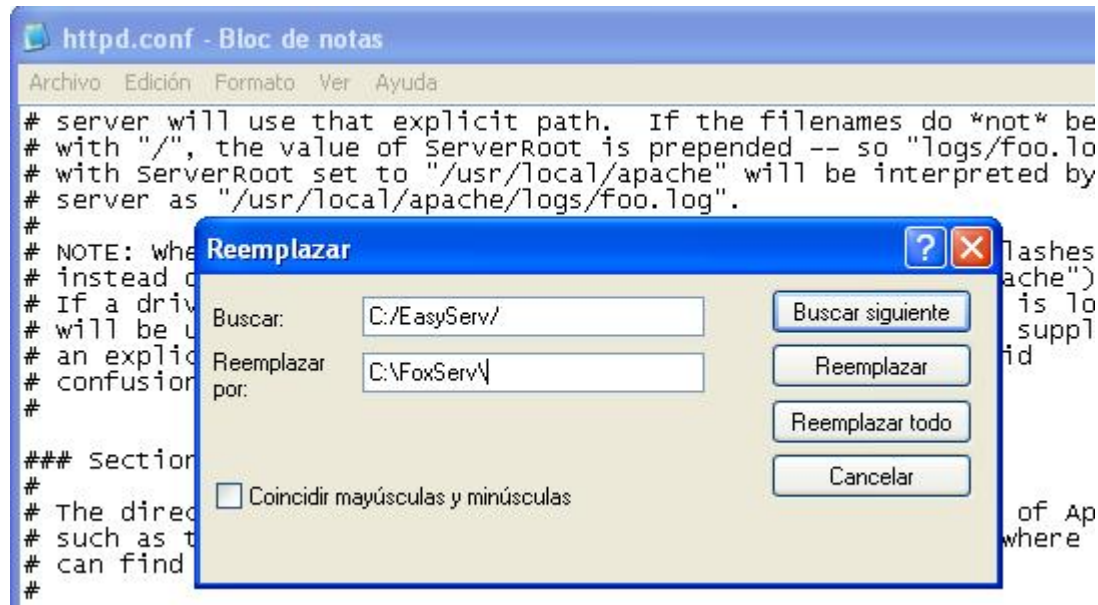
Por ultimo, nos aparece una ventana de confirmación de Destino de Instalación y Carpeta de Accesos Directos, pulsamos Next... y comienza la instalación, que durara unos minutos.

Al terminar la instalación nos aparece una ventana como esta (Ventana Finish EasyServ Install).

Pulsamos Final Step, y cerramos la ventana. Si todo salió bien, nos deberá salir la siguiente pantalla; ahí pulsamos Finish, (Inmediatamente, al pulsar Finish , nos aparecerá la ventana del instalador que se uso, simplemente pulsamos OK).

Reemplazando

no dirigimos a la siguiente dirección: C:\FoxServ\apache\conf y buscamos el archivo httpd.conf, lo abrimos con cualquier editor de texto plano (Bloc de Notas por ejemplo) y vamos a reemplazar lo siguiente:



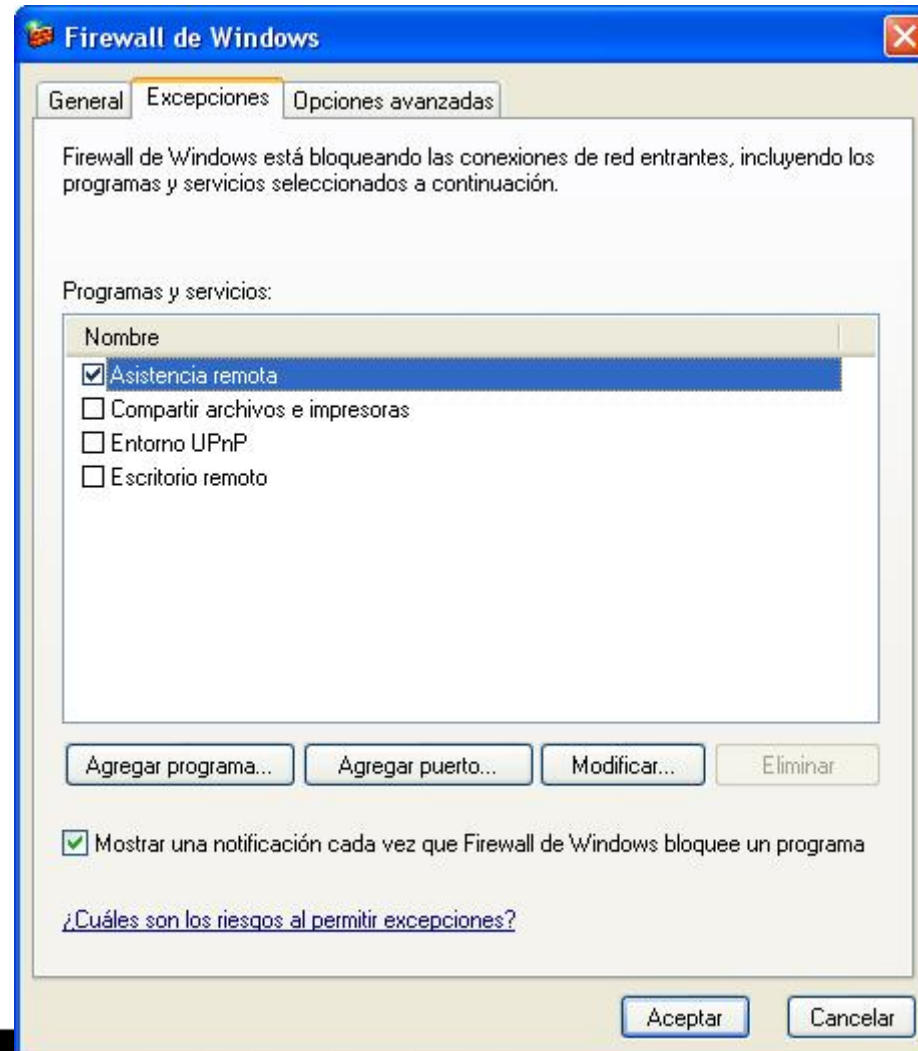
Pulsamos en Reemplazar todo. Ahora buscamos la línea Port 80 y la reemplazamos por Port 8080, guardamos los cambios y cerramos el archivo. (Por favor verifica que los cambios se hayan guardado). ahora nos dirigimos a C:\FoxServ\ y buscamos el archivo php.ini, y lo abrimos con el Bloc de Notas, o algún editor de texto plano, y verificamos que las direcciones estén expresadas como C:\FoxServ\ y no como C:\EasyServ\ ; en caso de ser así las reemplazamos, de la misma forma que lo hicimos en el archivo httpd.conf.

Nota: Puede suceder que ambos archivos httpd.conf y php.ini, te aparezcan con la dirección de instalación correcta C:\FoxServ\ , o alguno de los dos este correcto, o ninguno, (te aparecería la dirección incorrecta C:\EasyServ\) esto se debe a que en la pantalla Finish EasyServ Install, (Imagen mostrada en la parte de arriba) cuando se hace el reemplazo de cadenas, este no se haga correctamente. Me ha pasado, que a veces no se realiza bien y hay que verificar ambos archivos. Ahora nos dirigimos a C:\FoxServ\php y creamos una carpeta llamada temp, que almacenara lo referente a sesiones, esta se encuentra definida en el php.ini, pero el instalador no la crea, y así cuando queremos usar algún ejemplo de sesiones con php, nos marca error, aunque el código este correcto.

Por ultimo nos vamos nuevamente a C:\FoxServ y cambiamos el archivo php.ini a donde estén instalados los archivos de windows , que en mi caso es C:\WINDOWS ; ahí mismo, en ese mismo directorio (C:\FoxServ), encontraremos dos archivos llamados: start.bat y stop.bat que son los que nos servirán para encender y apagar el servidor Apache, así como al MySQL. Es probable, que si tienes el Service Pack 2 , el firewall te detecte cuando se usen por primera vez , tanto el Apache como el MySQL, y no te deje usarlos , o te pregunte que quieres hacer; si es así deberás escogerás la opción, permitir, y ambos se agregarían a las excepciones; pero por si acaso no te deja utilizarlos sigue estos pasos:

Desactivando el Corta Fuegos (Firewall)

Entra al panel de control y ve al Firewall de Windows, ábrelo y ve a la pestaña excepciones; debemos verificar que aparezcan los programas Apache.exe y mysqld-nt.exe.

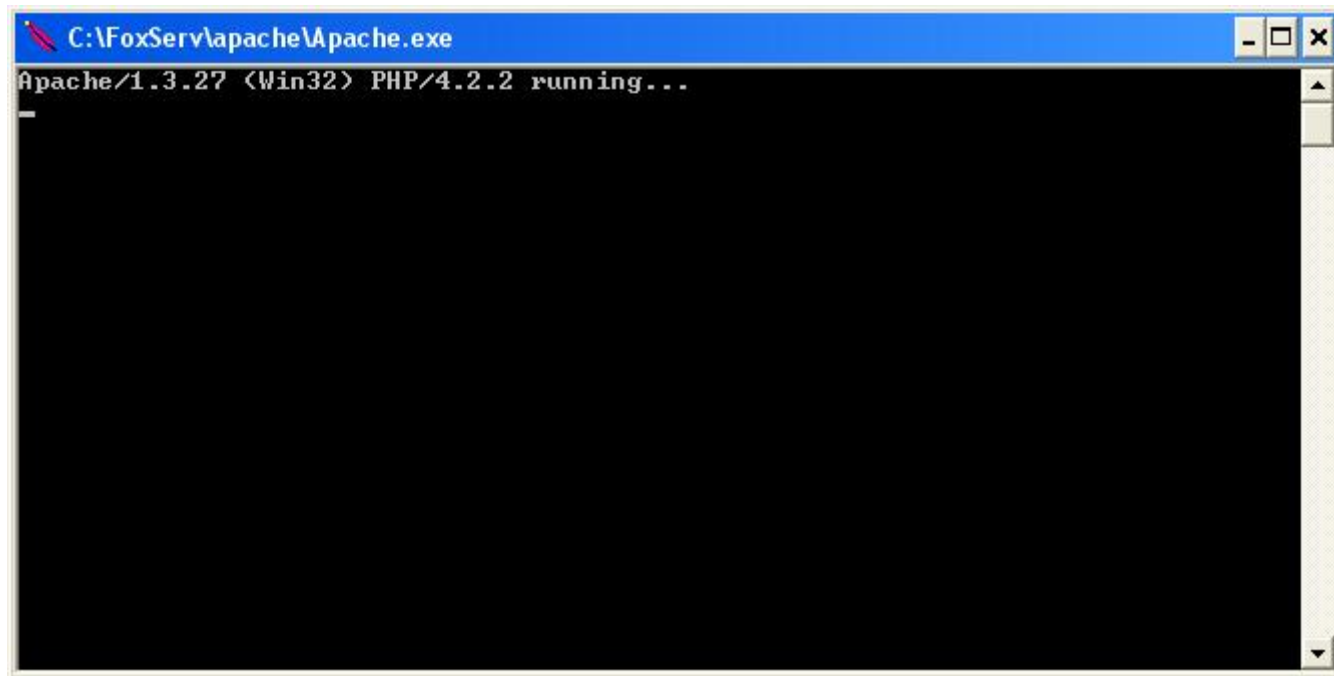


Si no te aparecen, da clic en Agregar programa y luego en Examinar; te aparecerá el cuadro de dialogo Examinar, donde dice nombre escribe **C:\FoxServ\apache\Apache.exe** y pulsa Abrir, y luego aceptar. Apache se ha agregado.

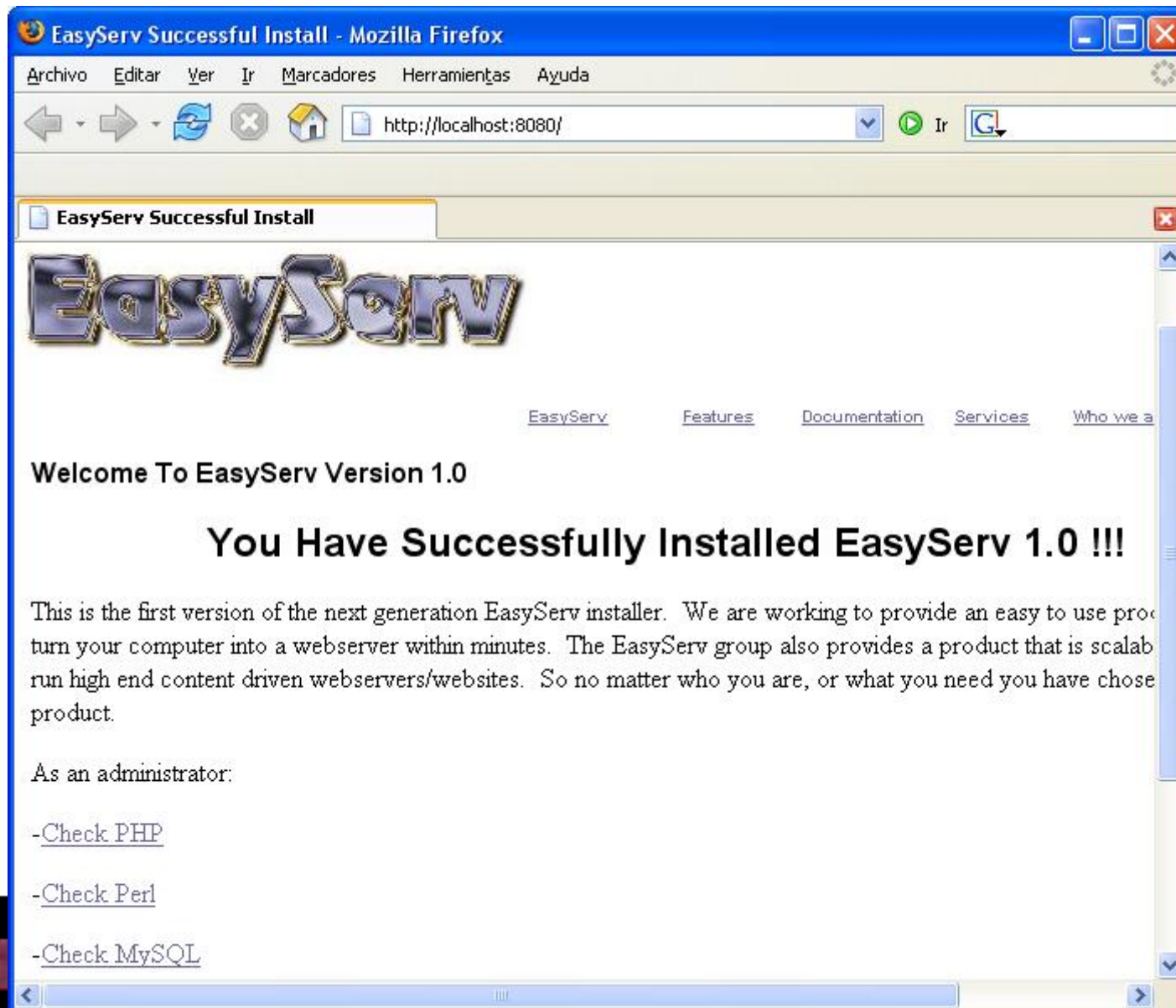
Haces la misma operación pero ahora escribes **C:\FoxServ\mysql\bin\mysqld-nt.exe** y pulsas Abrir y Aceptar. ii Ya estas listo para probar el Servidor.!!

Probando

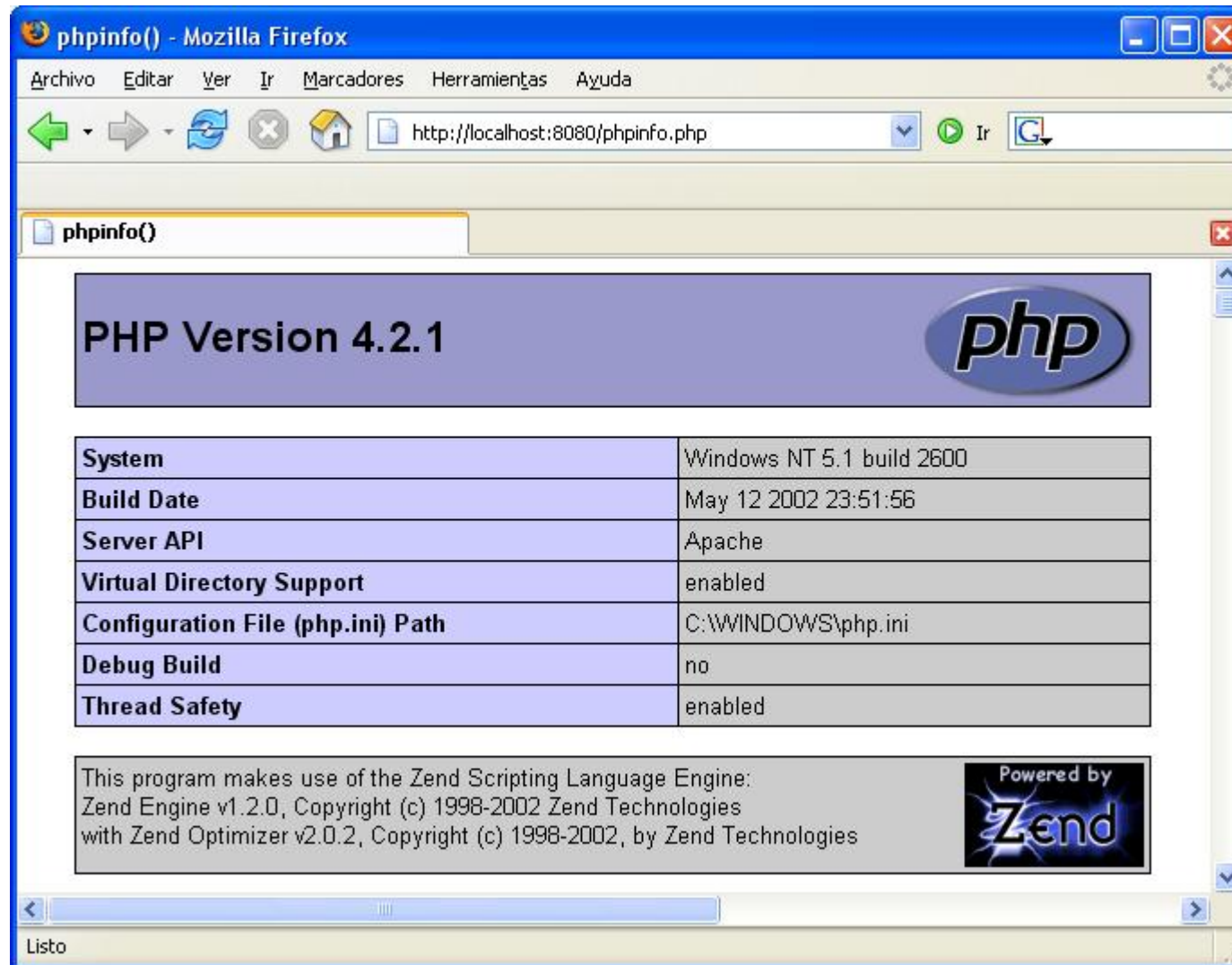
Ahora probaremos el servidor Apache, da doble-clic en el archivo start.bat, y si todo ha salido bien, deberá ver la siguiente pantalla.



Ahora abrimos nuestro navegador favorito (Internet Explorer, Firefox, Opera, Mozilla o Netscape) y tecleamos en la barra de dirección: `http://localhost:8080/` o la dirección IP `http://127.0.0.1:8080/` y nos deberá salir una pantalla como esta:



Si le damos un clic a link Chek PHP, nos saldrá la pagina con todas las directivas de php.ini

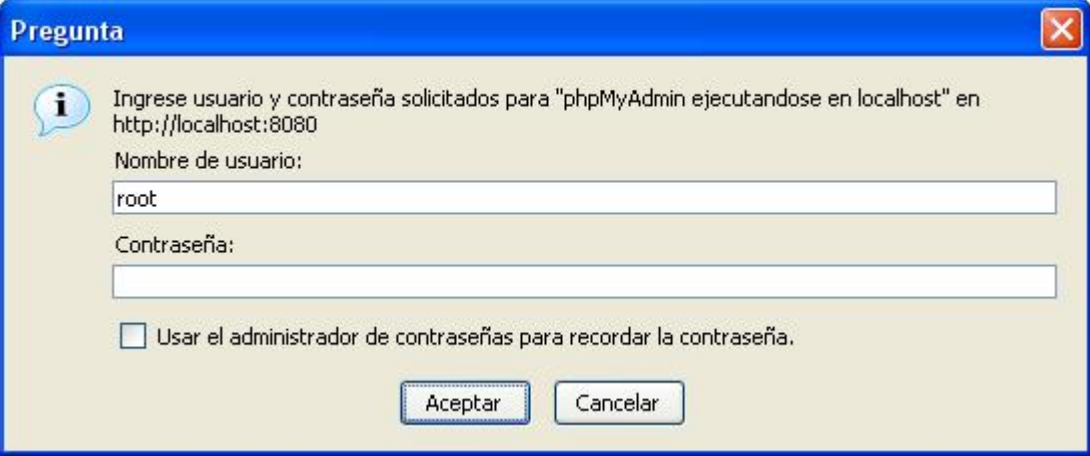


The screenshot shows a Mozilla Firefox browser window titled "phpinfo() - Mozilla Firefox". The address bar displays "http://localhost:8080/phpinfo.php". The page content includes the PHP logo and the text "PHP Version 4.2.1". Below this is a table with system information:

System	Windows NT 5.1 build 2600
Build Date	May 12 2002 23:51:56
Server API	Apache
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINDOWS\php.ini
Debug Build	no
Thread Safety	enabled

Below the table, it states: "This program makes use of the Zend Scripting Language Engine: Zend Engine v1.2.0, Copyright (c) 1998-2002 Zend Technologies with Zend Optimizer v2.0.2, Copyright (c) 1998-2002, by Zend Technologies". To the right of this text is a logo that says "Powered by Zend". The browser's status bar at the bottom shows "Listo".

Nos regresamos a la pagina principal y le damos un clic a Check MySQL y nos aparecerá una ventana pidiéndonos un nombre de usuario y contraseña



Pregunta

Ingrese usuario y contraseña solicitados para "phpMyAdmin ejecutandose en localhost" en <http://localhost:8080>

Nombre de usuario:

root

Contraseña:

☐ Usar el administrador de contraseñas para recordar la contraseña.

Aceptar Cancelar

En el usuario escribimos root, y en la contraseña , lo dejamos vacío y pulsamos aceptar. Entonces accedaremos a la la ventana principal de PhpMyAdmin



Para cerrar nuestro Servidor Apache , damos clic en el archivo **stop.bat**, y listo.

Y así culminamos la instalación, si has podido llegar con éxito hasta aquí, iii EN HORABUENA!!! en este momento tienes instalados en tu máquina, un Servidor IIS capaz de procesar paginas hasta ASP 3.0 , y un Servidor Apache capaz de procesar paginas PHP, un gestor MySQL, y un administrador PhpMyAdmin.

Como notas finales, recordemos que:
La ubicación del directorio raíz de IIS esta en **C:\Inetpub\wwwroot** y para entrar al servidor, deberás acceder con **http://localhost/** o **http://127.0.0.1/**
La ubicación del directorio raíz de Apache esta en **C:\FoxServ\www** y para entrar al servidor, deberás acceder con **http://localhost:8080/** o **http://127.0.0.1:8080/**

TERMINO DE UNIDAD IV

A CONTINUACION DESCARGAREMOS EL FRAMEWORK
de la pagina de microsoft es facil de descargar
utilizaremos framework1.1 sp2

Si hemos instalado el IIS después de haber instalado el .NET Framework, deberemos activar el Framework en el IIS para que funcionen las páginas ASP.NET, para ello abriremos una ventana de comandos y teclearemos lo siguiente:

```
C:\> C:
```

```
C:\> cd C:\WINDOWS\Microsoft.NET\Framework\v1.1.4322
```

```
C:\WINDOWS\Microsoft.NET\Framework\v1.0.3705> aspnet_regiis -i
```

Dependiendo de la de versión del .NET Framework el directorio (v1.0.3705) puede variar, en este caso es la versión 1.0 de .NET.

instalación de asp.net

Centrándonos en ASP.NET Web Matrix los lenguajes en los que se podremos trabajar son:

Visual Basic .NET

C#

C++

Veamos ahora los **requisitos de instalación** para ASP.NET Web Matrix

Requisitos mínimos:

Procesador

Pentium II, 450 MHz

Sistema Operativo y Memoria

Windows 2000 Professional - 96 MB de RAM

Windows 2000 Server - 192 MB de RAM

Windows XP Professional - 160 MB de RAM

Windows XP Home - 128MB de RAM

Visual Studio .NET vs. Web Matrix

Al fijarse detenidamente en Web Matrix se pueden descubrir varias diferencias entre los dos entornos. Visual Studio .NET es un completo entorno de desarrollo para todas las tecnologías .NET integrando con un sistema de control de código fuente como es Visual SourceSafe, mientras que Web Matrix está especialmente diseñado para la programación de ASP.NET y en general las tecnologías orientadas a Internet.

Una de las grandes diferencias entre estos entornos es que Visual Studio .NET crea proyectos ASP.NET usando separación de código, mientras que en Web Matrix se usa la técnica de código en línea a la que muchos programadores de ASP están acostumbrados y que consiste en incluir en el mismo fichero código de presentación (HTML, texto, etc.) el código ASP que crea y maneja el contenido dinámico.

En resumen, las principales diferencias entre Web Matrix y Visual Studio .NET son las siguientes:

Soluciones basadas en proyectos. Visual Studio .NET trabaja con la idea de proyecto y permite añadir varios tipos de ficheros al mismo proyecto, sin embargo, Web Matrix trata a cada fichero individualmente.

Estructura de página ASP.NET. Web Matrix crea páginas ASP.NET usando la aproximación de código en línea, mientras que Visual Studio .NET utiliza la técnica de la separación de código.

Interfaz de usuario. Web Matrix es ligero y rápido. Sin embargo, no proporciona el interfaz de usuario completo de Visual Studio .NET. Por ejemplo, Web Matrix no proporciona autocompletado de declaraciones, listas de miembros de objetos, o consejos de ayuda en la ventana de edición.

Compilación de ficheros de clase. A diferencia de Visual Studio .NET, Web Matrix no compila automáticamente los ficheros de clase dentro de los ejecutables, aunque esto puede realizarse manualmente desde la línea de comandos.

Ayuda del .NET Framework. Web Matrix no incluye documentación de referencia para el .NET Framework. En su lugar, proporciona un útil sistema de ayuda basado en el listado de las clases más comunes y sus miembros, además de un completo listado del resto de clases y componentes la librería de clases del .NET Framework. Además, Web Matrix proporciona enlaces a la .NET SDK si se encuentra instalado, y a las páginas de referencia de .NET de MSDN.

Comunidad. Web Matrix ha sido diseñado para ser una herramienta de comunidad, y proporciona varios tipos de enlaces a la comunidad on-line en el web <http://asp.net/WebMatrix>, al igual que enlaces a grupos de noticias, listas de servidores y otros sitios que proporcionan soporte a la comunidad de Web Matrix.

Precio. Web Matrix es gratuito

El grupo de desarrollo de Web Matrix de Microsoft lo describe como “un producto diseñado para acercar a la comunidad el desarrollo de ASP.NET, al mismo tiempo que proporciona un uso divertido”. Web Matrix se centra en esas tareas que satisfacen los requisitos para construir aplicaciones ASP.NET.

Instalación

A continuación se estudiará paso a paso el proceso de instalación de Web Matrix.

Descargar Web Matrix

Desde el sitio oficial de la plataforma ASP.NET, se tiene acceso directo a la página del Web Matrix Project. Desde aquí se puede acceder tanto a la instalación de Web Matrix como a la del framework ASP.NET, requerido para el funcionamiento del primero.

Evidentemente, la segunda opción es más completa (también la descarga es notablemente mayor en cuanto a tamaño) y recomendada para desarrolladores no habituados a la plataforma o noveles, ya que cuenta con documentación y ejemplos, mientras que la segunda es la más adecuada para desarrolladores experimentados, pues contiene sólo lo necesario para construir y poner en producción aplicaciones ASP.NET.

De todos modos, el primer paso del proceso de descarga de Web Matrix proporciona un test que comprueba si el framework ya está instalado en el ordenador. Si es así, la descarga del mismo no es necesaria, evidentemente.

Adicionalmente, existen una serie de requisitos de sistema que es necesario cumplir para que Web Matrix funcione. Básicamente, estos requisitos se refieren al sistema operativo, que debe ser uno de los siguientes

- Windows 2000 Professional y Server (Service Pack 2 recomendado)
- Windows XP Professional o Home Edition

Aunque las aplicaciones creadas con Web Matrix pueden ser utilizadas con cualquier navegador, la máquina de desarrollo debe tener instalado también el Internet Explorer 5.5 o superior.

Por último, se pueden descargar e instalar opcionalmente otros componentes: para desarrollar aplicaciones que requieran acceso a datos, si no se dispone de SQL Server 7.0 (SP2) o SQL Server 2000, es posible la descarga gratuita de MSDE, un motor de base de datos compatible con SQL Server; y para crear aplicaciones orientadas a dispositivos móviles como PDAs, teléfonos móviles, etc. está disponible el Mobile Internet Toolkit, cuya descarga es también gratuita.

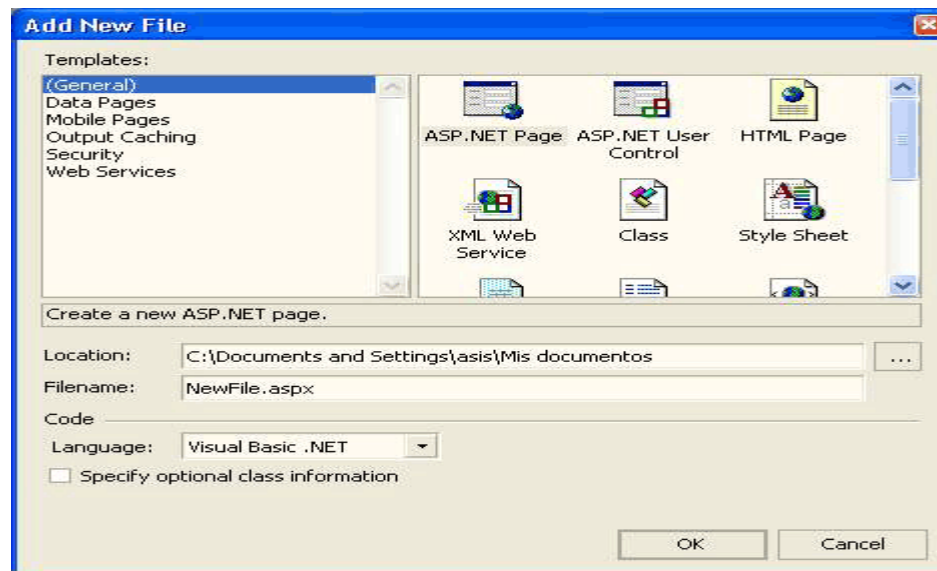
Instalación de Web Matrix

Una vez descargado el paquete de Windows Installer webmatrix.msi y comprobado que se cumplen todos los requisitos previos, se puede proceder a la instalación del producto.

El proceso de instalación es muy sencillo y está guiado por el típico asistente de Windows para la instalación de aplicaciones, por lo que no requiere de ninguna aclaración adicional.

Un paseo por la interfaz

Cuando se ejecuta Web Matrix por primera vez, aparece un cuadro de diálogo (figura 1) que permite seleccionar de entre un gran número de plantillas, el tipo de fichero que se quiere agregar. Además, se puede elegir, en función del fichero que se agregue, el lenguaje de programación que se utilizará entre Visual Basic .NET y C#.

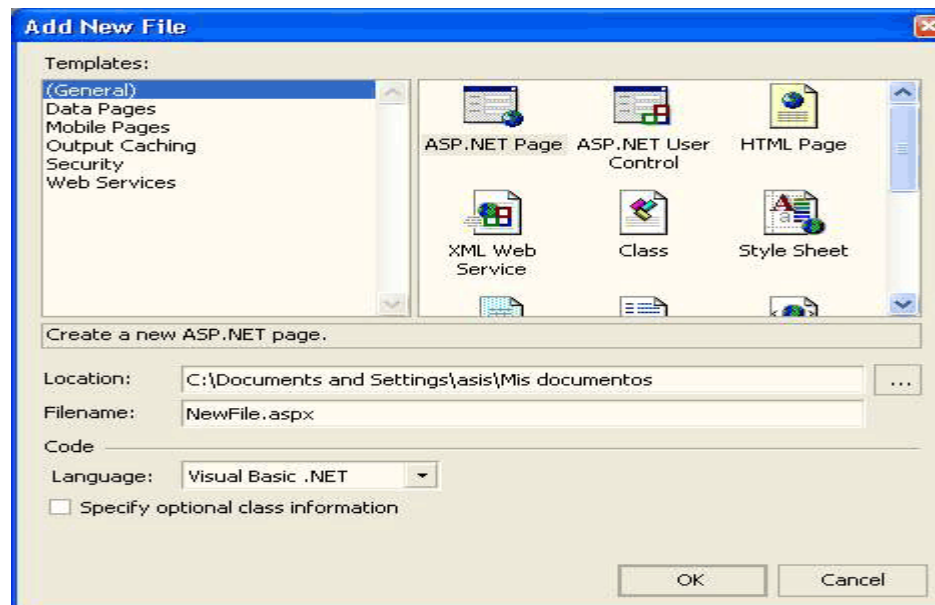


instalación de Web Matrix

Una vez descargado el paquete de Windows Installer webmatrix.msi y comprobado que se cumplen todos los requisitos previos, se puede proceder a la instalación del producto.

El proceso de instalación es muy sencillo y está guiado por el típico asistente de Windows para la instalación de aplicaciones, por lo que no requiere de ninguna aclaración adicional.

Un paseo por la interfaz Cuando se ejecuta Web Matrix por primera vez, aparece un cuadro de diálogo (figura 1) que permite seleccionar de entre un gran número de plantillas, el tipo de fichero que se quiere agregar. Además, se puede elegir, en función del fichero que se agregue, el lenguaje de programación que se utilizará entre Visual Basic .NET y C#.



Si se selecciona, por ejemplo, añadir un nuevo fichero de tipo 'página ASP.NET', el IDE se mostrará como se ve en la figura 2.

El IDE de Web Matrix resultará familiar a los desarrolladores acostumbrados a trabajar con Visual Studio .NET: consta de los elementos típicos como la barra de menú y las barras de herramientas de la parte superior al estilo Office; a la izquierda se sitúa la caja de herramientas o Toolbox, desde donde se pueden arrastrar los distintos controles disponibles a la zona central; en esa zona central se encuentra el área de trabajo, en la que se puede trabajar con varios ficheros a la vez y que dispone de varias vistas (vista de diseño, código, etc.) en función del tipo de fichero que se esté visualizando; por último, a la derecha se encuentran varias ventanas de proyecto y en la parte inferior una barra de estado

EJEMPLO

Ventana de Inicio de Web Matrix

Para poder probar los ejemplos que se describen a continuación es necesario tener instalado el .NET Framework y el ASP.NET Web Matrix Project (figura 8), siguiendo, por ejemplo, los pasos que se han dado en el apartado anterior.

Al iniciar Web Matrix se nos presenta una ventana desde la que podemos seleccionar el tipo de fichero que queremos crear, según la plantilla seleccionada tendremos diferentes tipos de ficheros, por ejemplo, si la plantilla seleccionada es (*General*), tendremos la posibilidad de crear páginas ASP.NET, Controles de Usuario ASP.NET, páginas HTML, servicios web XML, clases, hojas de estilo, ficheros de servidor y configuración del servidor, ficheros XML, entre otros.

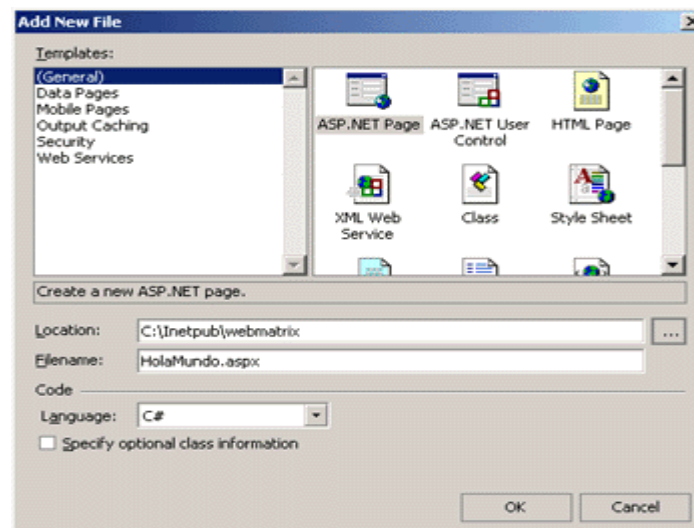
Las plantillas disponibles son: General, Data Pages, Mobile Pages, Output Caching, Security, Web Services.

Como vemos Web Matrix proporciona una amplia variedad de plantillas que facilitan el trabajo para empezar a programar.

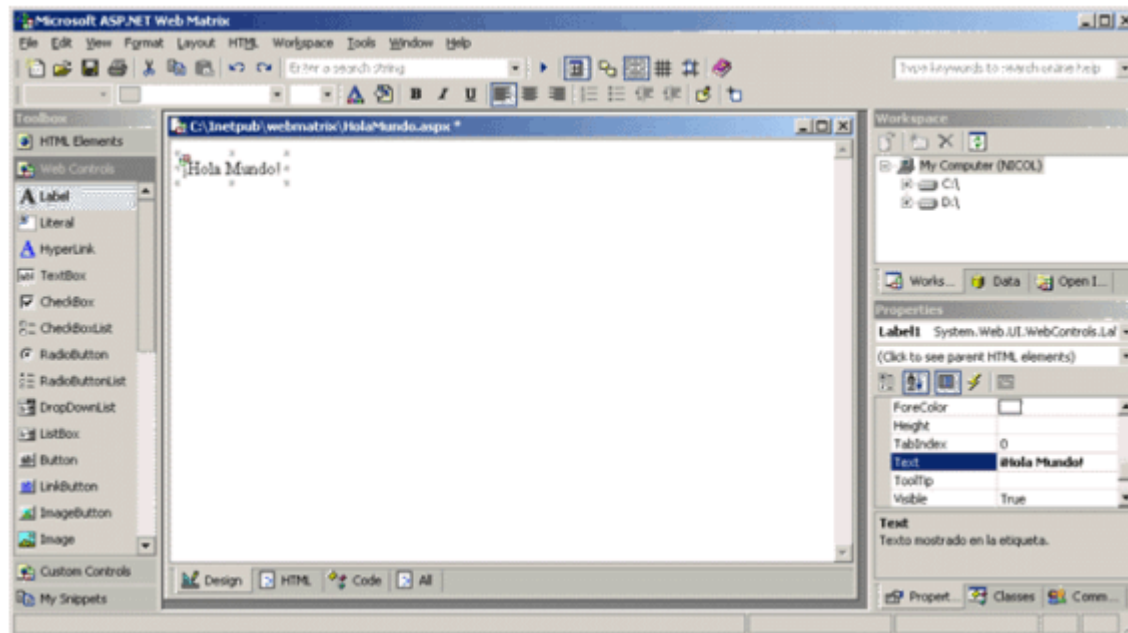
Desarrollaremos varios ejemplos. Comenzaremos por un ejemplo muy sencillo y casi inmediato que nos permitirá comprobar el funcionamiento de Web Matrix y del servidor web del mismo. Iremos aumentando la complejidad con un ejemplo formado por formularios y terminaremos con un ejemplo de acceso a base de datos.

¡Hola mundo!

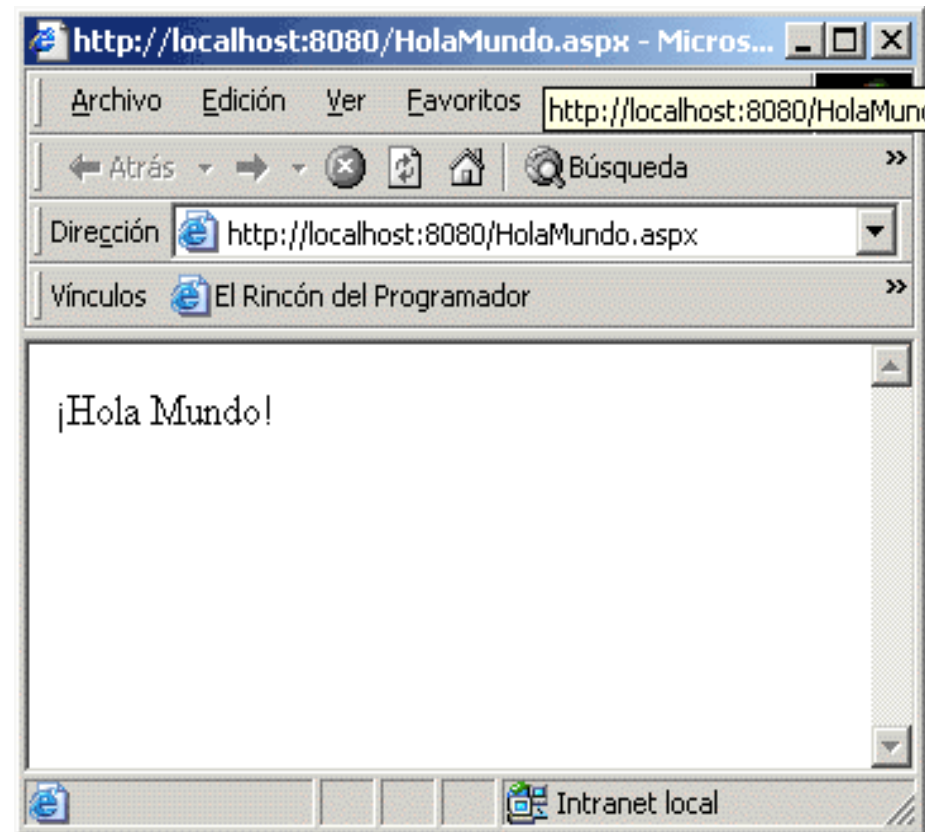
En este primer y simple ejemplo se creará una página ASP.NET (figura 9) a la que se añadirá un Web Control, exactamente el Label



Arrastramos el control web *Label* sobre nuestra página web y modificamos la propiedad *Text* con el valor `"¡Hola Mundo!"`.



Al pulsar en el botón *Start*, Web Matrix iniciará su servidor web y lanzará al navegador en la dirección de nuestra página, y podremos ver el resultado de nuestra página ASP.NET



Un simple formulario

En este segundo ejemplo desarrollaremos una página ASP.NET formada por un formulario en el que introduciremos nuestro nombre y un botón para enviar. Una vez hecho esto aplicaremos una simple validación al formulario (comprobaremos que se ha indicado un nombre) y si el formulario está cubierto se saludará al nombre indicado.

Para ello procedemos como en el ejemplo anterior, seleccionamos la plantilla (*General*) y *ASP.NET Page*. Indicamos la carpeta en donde queremos guardar la página y un nombre.

En modo diseño arrastramos los siguientes controles web a la página ASP.NET:

- 1. Label1
- 1. TextBox1
- 1. RequiredFieldValidator1
- 1. Button1
- 1. Label2

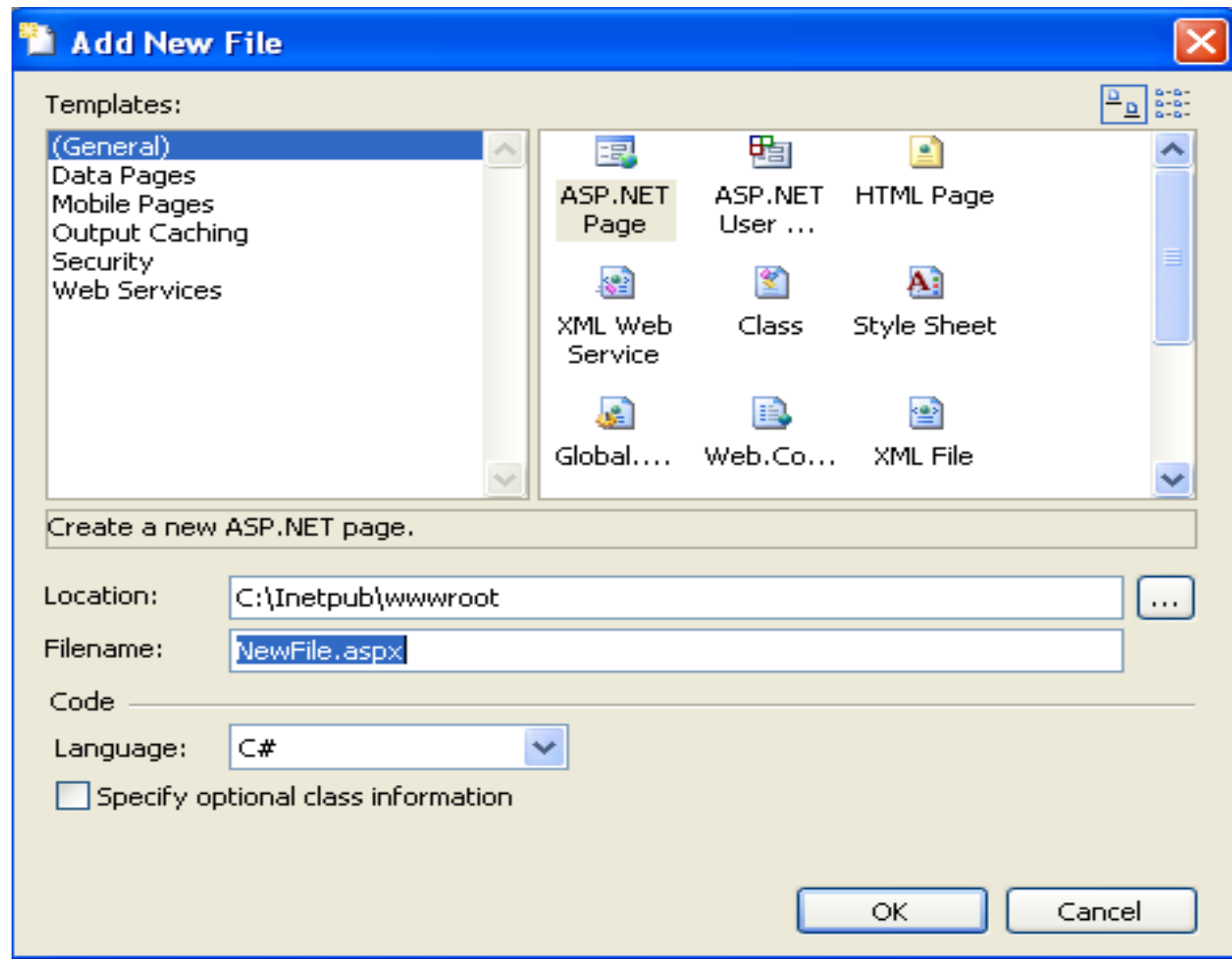
Al *Label* cambiaremos la propiedad *Text* por "Nombre: "

En la propiedad *ControlToValidate* del control *RequiredFieldValidator* seleccionamos el control *TextField*. Y cambiamos el valor de la propiedad *ErrorMessage* por el texto que deseemos que se muestre en caso de que el usuario no cubra el campo, por ejemplo

"ERROR: Debes indicar tu nombre."

En el evento Click del botón comprobaremos el formulario, y si este está cubierto escribiremos un saludo en el Label2.

Para ello haremos doble click sobre el botón así el editor generará el código del evento del ratón, y añadiremos el siguiente código:



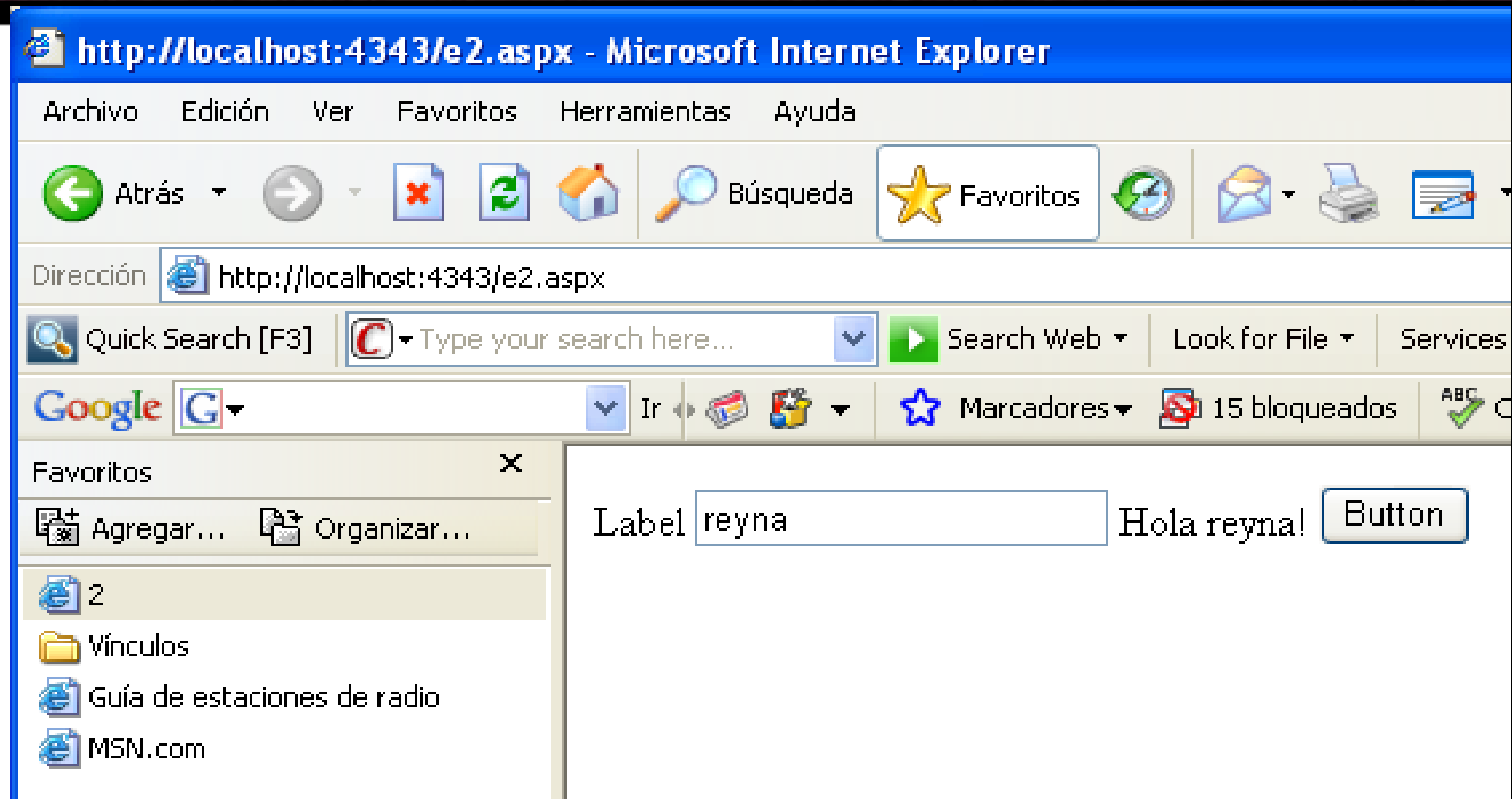
Al *Label* cambiaremos la propiedad *Text* por "Nombre: "

En la propiedad *ControlToValidate* del control *RequiredFieldValidator* seleccionamos el control *TextField*. Y cambiamos el valor de la propiedad *ErrorMessage* por el texto que deseamos que se muestre en caso de que el usuario no cubra el campo, por ejemplo "ERROR: Debes indicar tu nombre."

En el evento Click del botón comprobaremos el formulario, y si este está cubierto escribiremos un saludo en el *Label2*.

Para ello haremos doble click sobre el botón así el editor generará el código del evento del ratón, y añadiremos el siguiente código:

```
void Button1_Click(Object sender, EventArgs e) {  
    if (Page.IsValid) {  
        Label2.Text = "Hola " + TextBox1.Text + "!";  
    }  
}
```



Fin