Servidores Web de Altas Prestaciones (2015-2016)
GRADO EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA

Asegurando IIS, Apache y Nginx.

Luis Gil Guijarro

11 de mayo de 2016

Índice

Índice de figuras

Índice de tablas

Conforme el rápido crecimiento de la web y su importancia capital hoy en día a adquirido especial relevancia para las empresas poder contar con servidores web seguros.
En este documento se pretende mostrar de manera **breve** mecanismos **básicos** para asegurar los servidores web Apache, Nginx e IIS junto con las justificaciones del porqué estos mecanismos.

1. Asegurando IIS.

IIS es el servidor web de Microsoft que incluye lo que ellos llaman “Internet Information Services” (IIS) que integra una serie de tecnologias (IIS, ASP.NET...) que permiten la comunicación via web. [1] [1]

1.1. IP Address and Domain Restrictions

Una buena manera de empezar a asegurar nuestro servidor IIS consiste en filtrar el tráfico IP que llega hasta él para que solo pase aquel deseado. A pesar de que nuestro servidor web tenga un firewall en otra máquina que le sirve de primera linea de defensa y realize un primer control de que tráfico deja pasar y cual no sigue siendo útil tener en nuestro servidor web otro filtrador de tráfico constituyendo asi una nueva linea de defensa por lo que en caso de que por alguna clase de error en ese firewall externo dejara pasar un tráfico que no debe este filtrador en nuestro servidor purgaria este tráfico no deseado.

El servidor web posiblemente tenga también un firewall interno que permita controlar el tráfico que entra y que sale pero en esta explicación voy a hablar más concretamente de una característica de IIS “IP Address and Domain Restrictions” que permite que a una determinada web, fichero, carpeta o al propio servidor web puedan acceder una serie de IPs o no, es decir si tengo mi página web llamada `www.soloetssit.com` puedo hacer una regla que solo permita el acceso a esta web de las direcciones IP del rango 172.16.0.0-172.16.31.255.255 que son las direcciones que pertenecen a la intranet de la etsiit.

Para habilitar esta característica[3][2][4] en nuestro servidor IIS recien instalado nos vamos a:
Administrador del servidor (En la barra de herramientas al lado del botón inicio windows) Roles $\rightarrow$ Servidor web $\rightarrow$ Servicios de rol $\rightarrow$ Añadir servicio de rol. Vease 1.1.

[1] Para cualquier persona que quiera probar los ejemplos aquí mostrados puede encontrar una breve explicación de como instalar IIS7 en el Anexo I. 4.

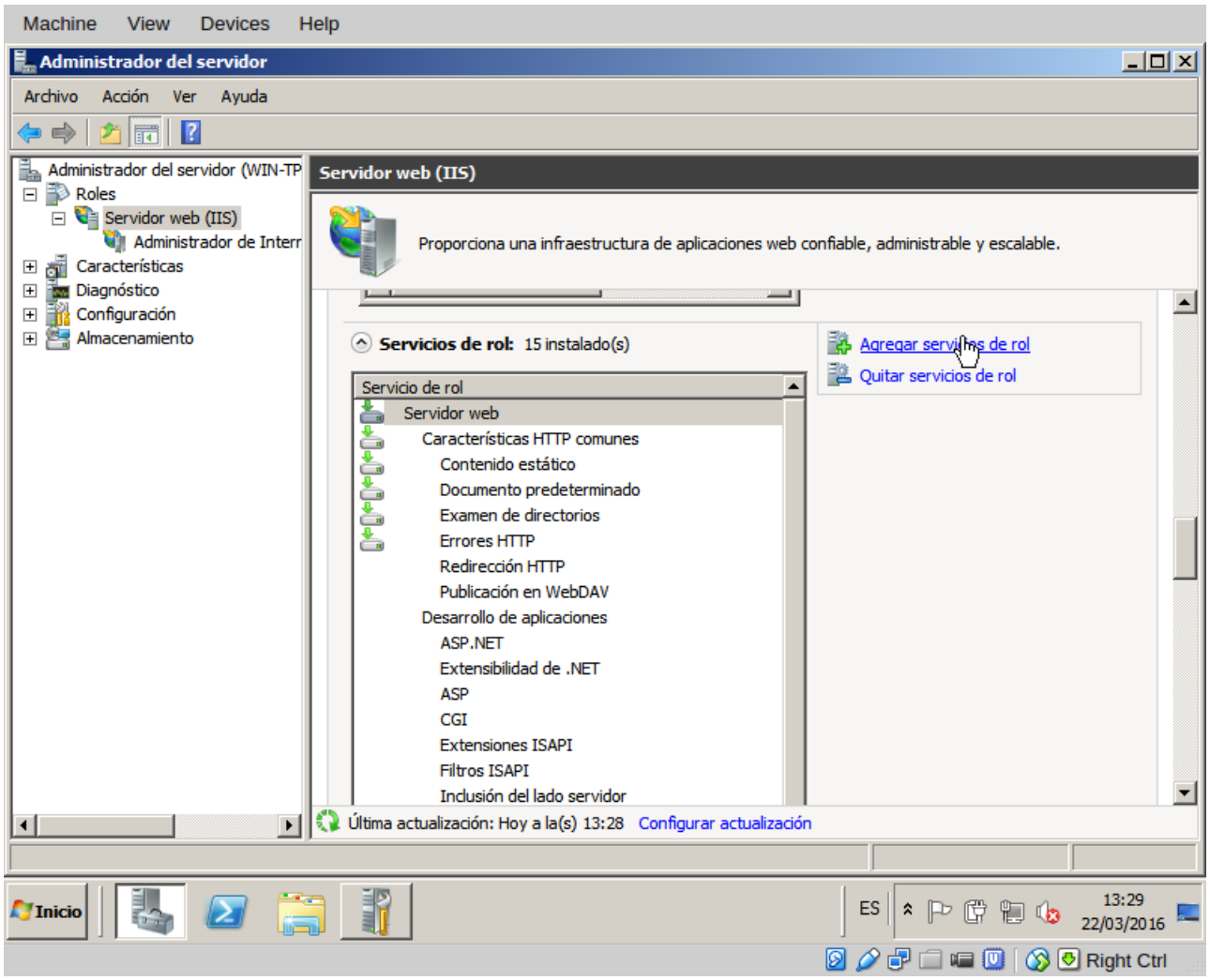

Figura 1.1: Habilitando restricciones IP: seleccionamos añadir rol dentro de Administrador del servidor.

Nos vamos a `Seguridad` y hacemos click en la casilla llamada `Restricciones de IP y de dominio` → Siguiente → Instalar 1.2.

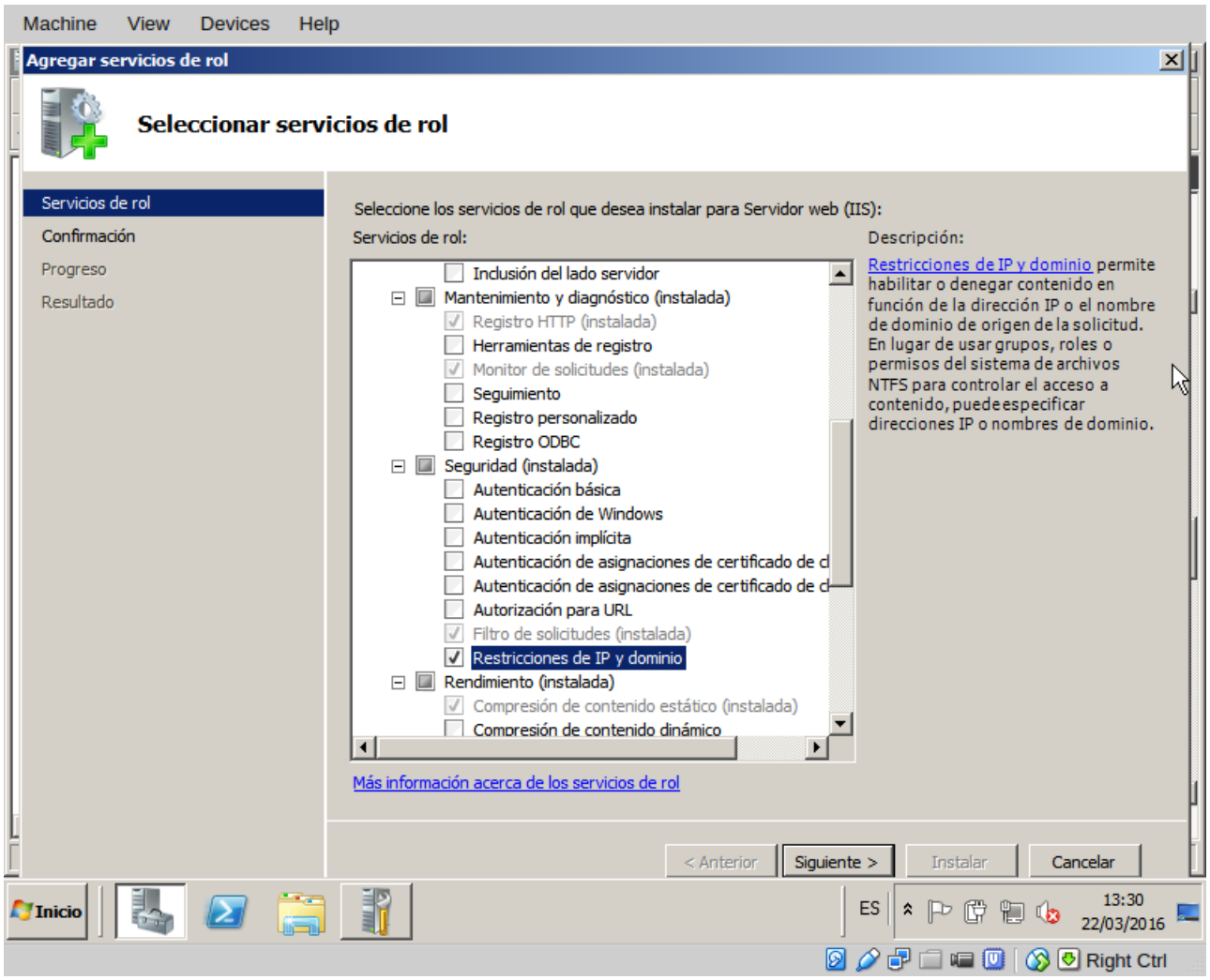

Figura 1.2: Habilitando restricciones IP: pulsado sobre `Restricciones de IP y de dominio`

Ahora nos vamos al Administrador del IIS:

Inicio → Herramientas administrativas → Administrador de Internet Infortation Service (IIS)
y hacemos click sobre el simbolo “+” al lado del nombre de nuestro servidor y pulsamos sobre sitios seleccionando después el sitio web sobre el que deseamos permitir o denegar acceso a una serie de IPs. Vease 1.3.

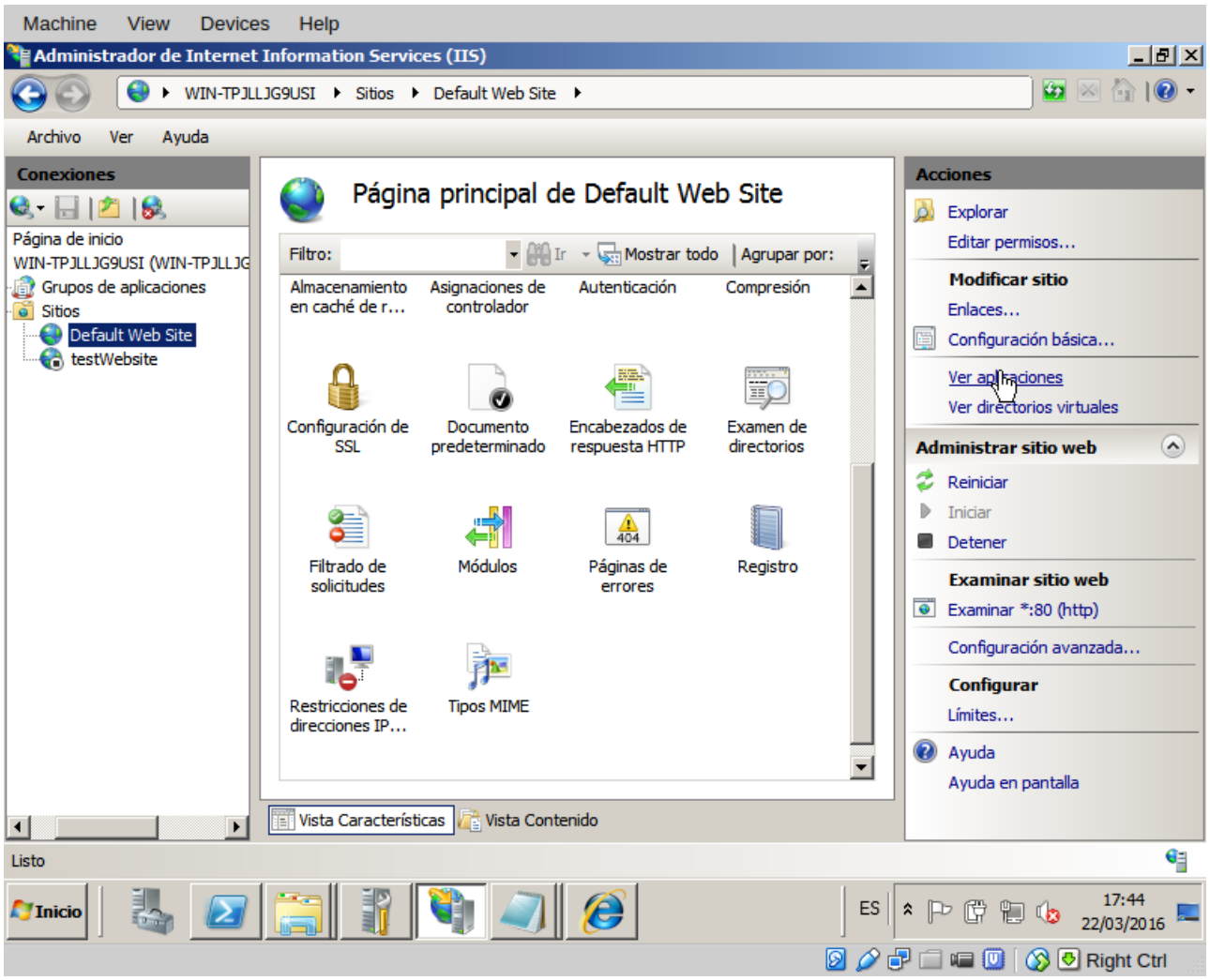

Figura 1.3: Página principal configuración sito web por defecto del IIS.

Seleccionamos `Restricciones de direcciones IP` y le damos a `Agregar entrada de permiso o Agregar entrada de denegación` en función de si deseamos prohibir o permitir a una serie de IPs acceso a nuestra web. Por ejemplo para denegar el acceso a la web que se crea por defecto al instalar el IIS a los ordenadores de la red local le damos a `Agregar entrada de denegación...` y en la venta que nos sale seleccionamos `Intervalo de direcciones IP` ponemos 192.168.1.0 y en máscara 255.255.255.0. Vease 1.4.

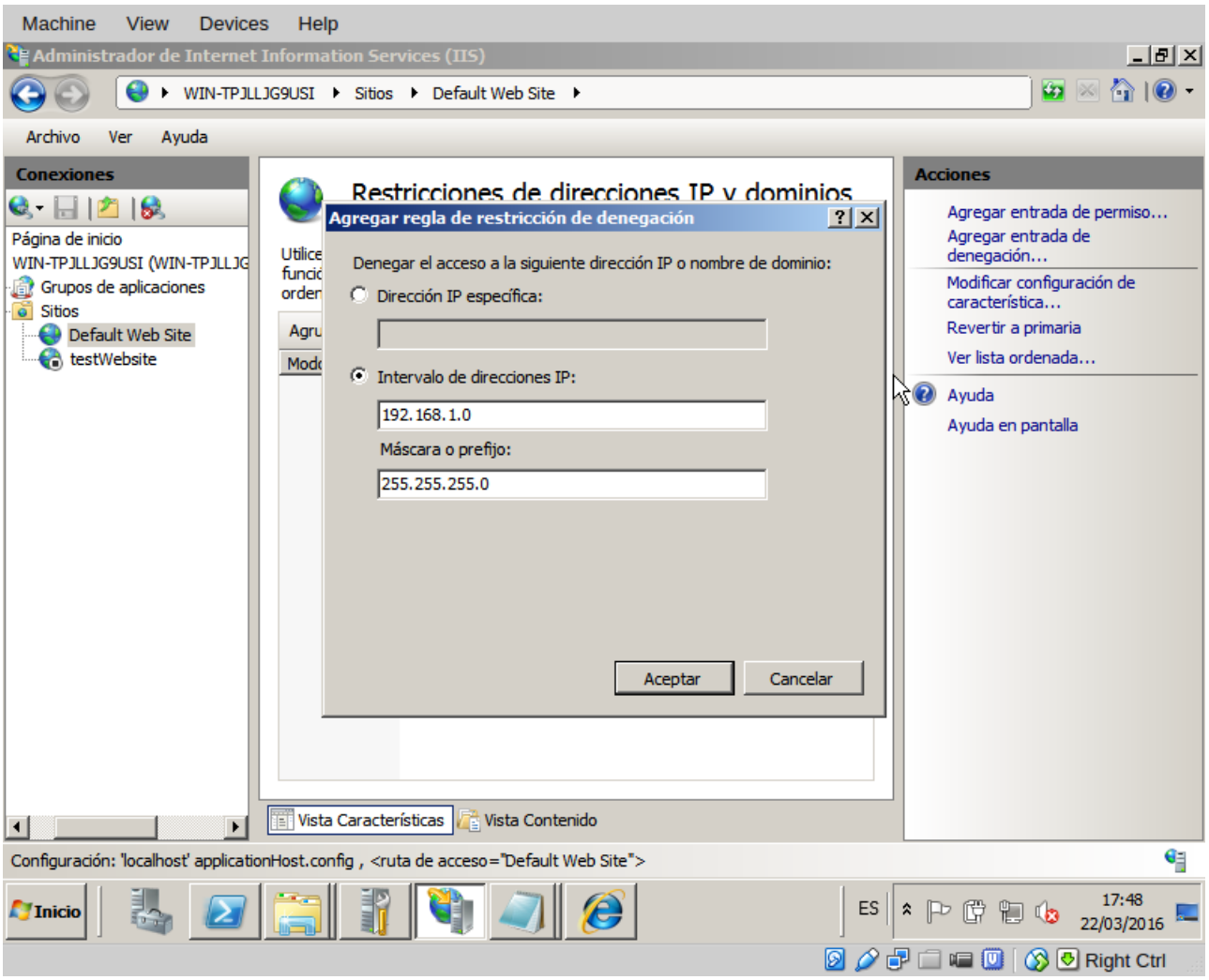

Figura 1.4: Agregamos regla de denegación para las ips del rango 192.168.1.0-192.168.1.255

Con este conseguiremos prohibir el acceso a esta web para los ordenadores de dirección 192.168.1.0-192.168.1.255 vease 1.5.

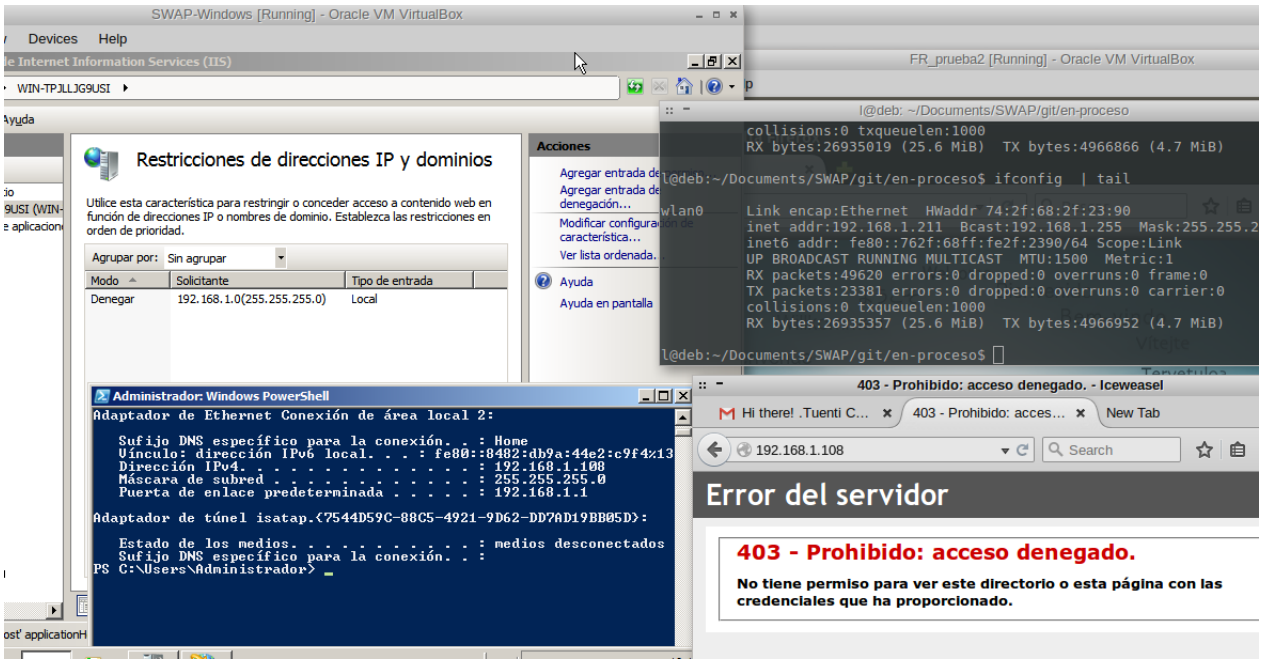

Figura 1.5: Tras denegar acceso a IPs 192.168.1.0/24 comprobamos que efectivamente deniega el acceso a la web a una ip de este rango.

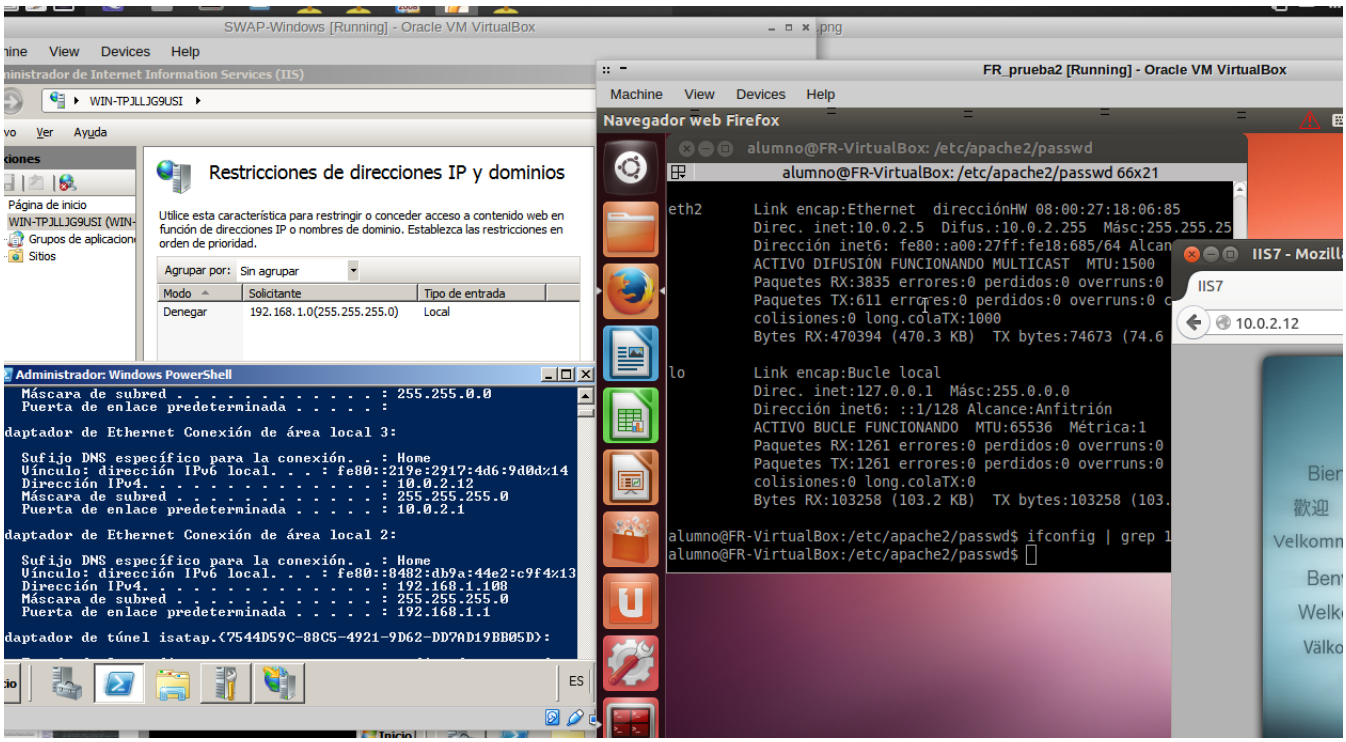

Figura 1.6: Tras denegar acceso a IPs 192.168.1.0/24 comprobamos que permite el acceso a direcciones ip de otro rango.

1.1.1. Conclusión

Al empezar a leer sobre esta característica del IIS tenía en mente permitir o denegar el acceso en función del tipo de tráfico (TCP/UDP) o bien permitir que a ciertos

elementos de la web pudieran acceder una serie de IPs y otras no. Por ejemplo a www.trabajoswap.es pudiera acceder todo el mundo pero a www.trabajoswap.es/soloetssit.html solo pudieran acceder los miembros de la intranet de la etsiit. A la hora de trabajar con `IP Address and Domain Restrictions` no te permite hilar tan fino solo aceptar o denegar el acceso a la página web en global.
Tiene utilidad ya que es una nueva linea de defensa tras (si se tiene) firewall externo, interno y esta especie de firewall más interno del IIS.

1.2. URL Authorization

Este característica del IIS sirve para que unas determinadas url solo puedan ser accedidas por determinados usuarios o grupos de usuarios. Continuando con nuestra web `www.soloetssit.com` ahora queremos que a `www.soloetssit.com/swap/` solo puedan acceder los alumnos de la clase de SWAP y no toda la Etsiit y a `www.soloetssit.com/swap/soloprofesor.` solo pueda acceder el usario profesor.

1.2.1. Habilitar URL Authorization

Para poder hacer esto en IIS al igual como hicimos para habilitar `IP Address and Domain restrictions` realizamos los siguientse pasos:

1. Nos vamos a `Administración del servidor` (Al lado del boton de inicio de windows).
2. Desplegamos roles, clickamos sobre `Servidor web (IIS)` y nos vamos a `Servicios de rol`. Vease 1.7.

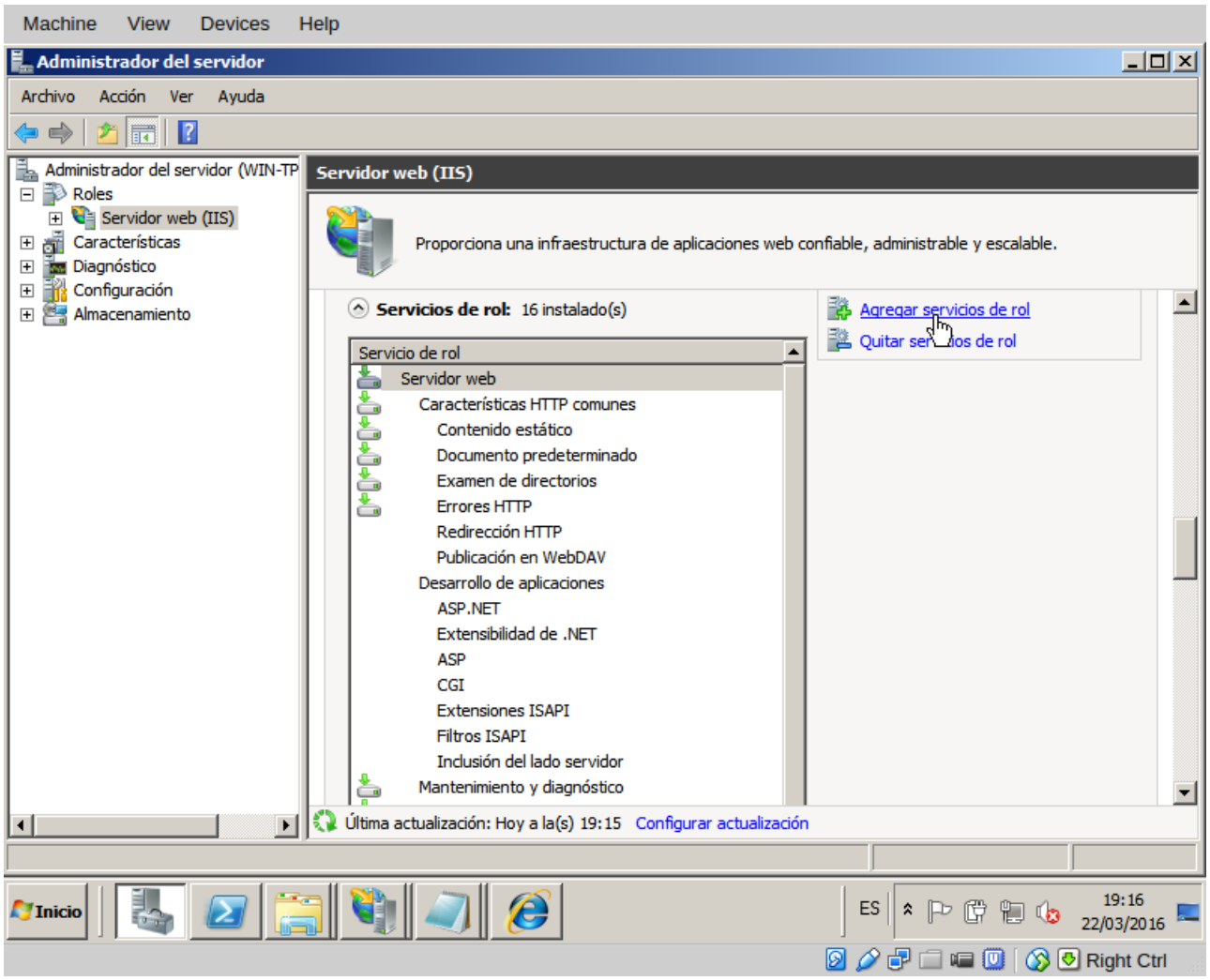

Figura 1.7: Habilitar URL Authorization: roles Servidor web nos vamos a `Servicios de rol`

3. Pulsamos sobre `Agregar rol` y nos vamos a `Seguridad` y añadimos `Autentificación básica` y `Autorización para URL`. Ahora nos vamos a `Desarrollo de aplicaciones` y añadimos `ASP.NET`.

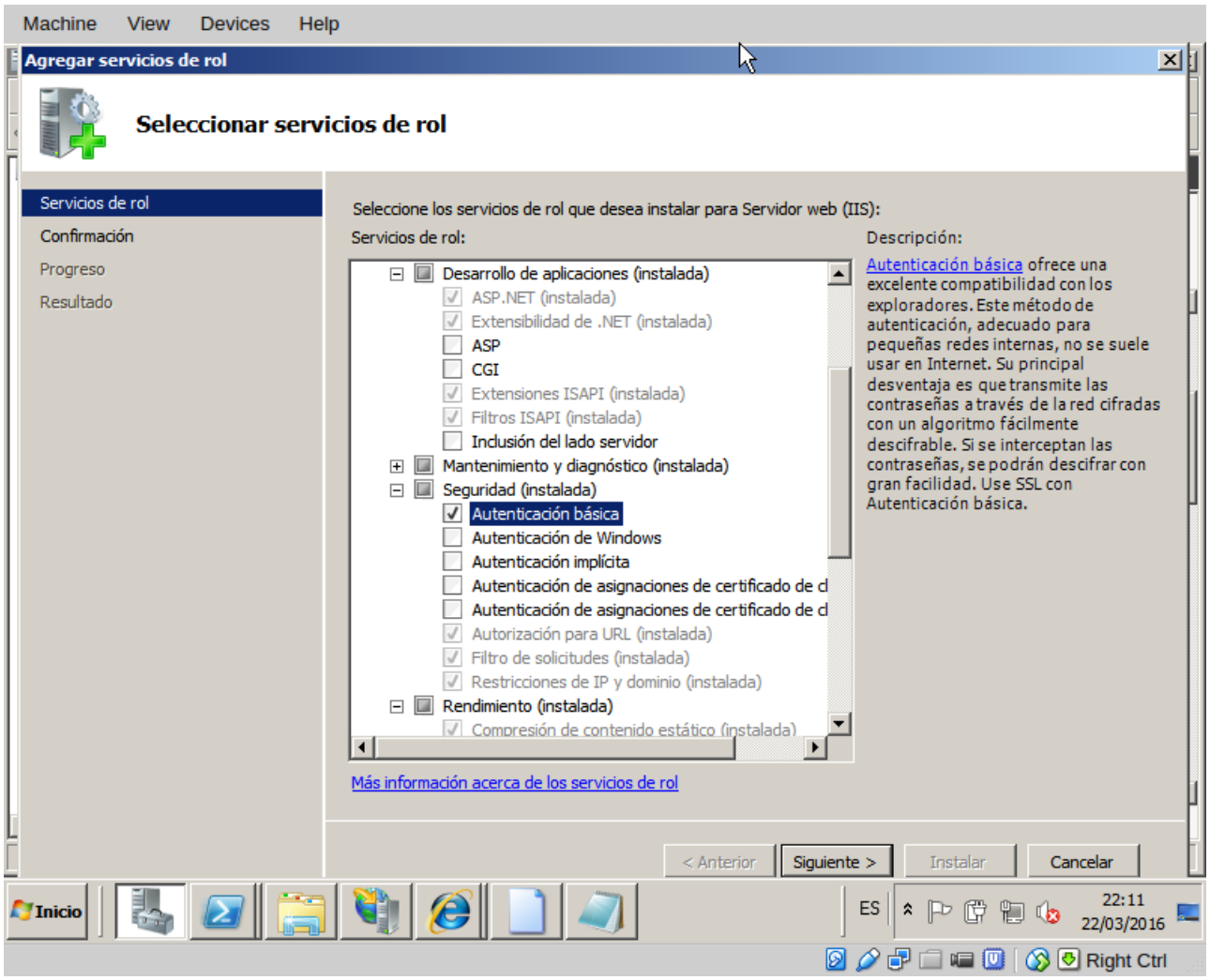

Figura 1.8: Habilitar URL Authorization: añadimos roles ASP.NET y `Autorización para URL`

4. Nos saldra una ventana (¿Desea .. requeridos para ASP.NET?) le damos a agregar servicios de rol requeridos.

5. Siguiente e Instalar. Nos debe decir algo parecido a `instalación correcta`.

1.2.2. Preparación para probar URL Authentication

Para comprobar el funcionamiento de `URL Authorization` vamos a simular un escenario en el que tenemos nuestra web `www.soloetssit.com` y queremos que a `www.soloetssit.com/swap/` solo puedan acceder los alumnos de la clase de SWAP y a `www.soloetssit.com/swap/soloprofesor.aspx` solo pueda acceder el usario profesor.
Para ello en primer lugar tenemos que crear tres cuentas y meterlos en el grupo SWAP. Seguimos el siguiente proceso:

1. Abrimos consola powershell como administrador e insertamos los siguientes comandos uno a uno:

```
net user alumno1 <contrasenia_muy_dificil> /add
net user alumno2 <contrasenia_muy_dificil> /add
```

```
net user profesorPEDRO <contrasenia_aun_mas_dificil> /add
net localgroup SWAP /add
net localgroup SWAP alumno1 /add
net localgroup SWAP alumno2 /add
```

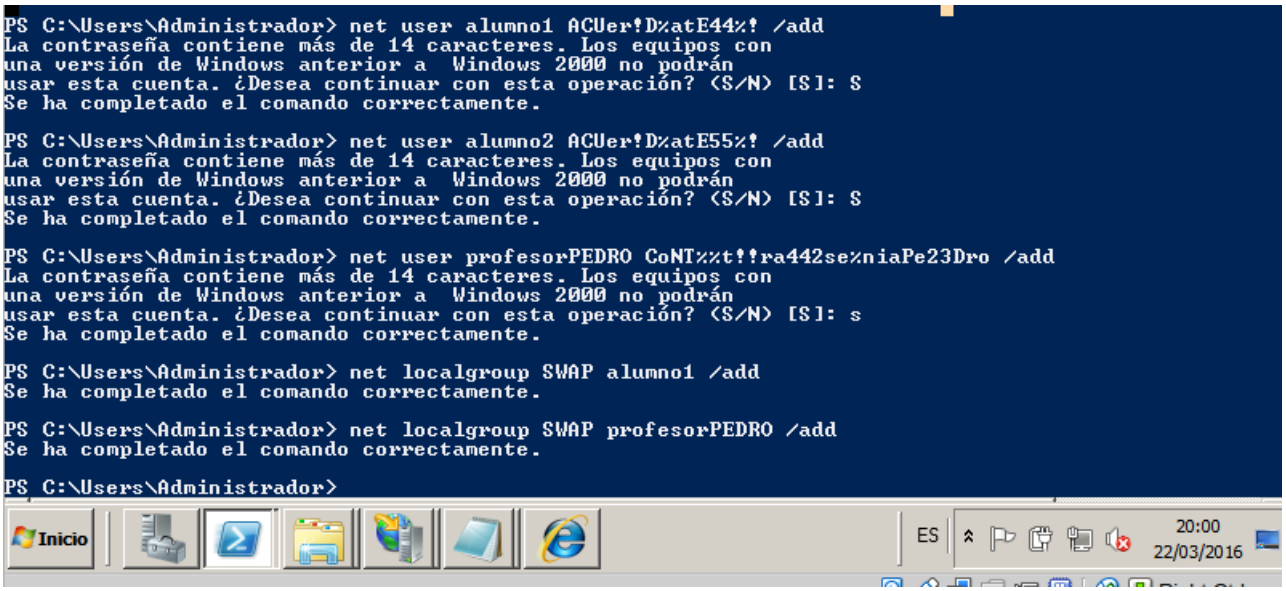

Figura 1.9: Probando URL Authorization : creamos usuarios y grupo utilizando cmd como administrador.

2

2. Ahora abrimos el `Explorador de Windows` (al lado del icono de powershell) y nos vamos a $C:\inetpub\wwwroot$.

3. Nos vamos a `Organizar` (esquina superior izquierda) $\rightarrow$ `Ver` $\rightarrow$ Escroleamos la barra hacia abajo hasta encontrar `Ocultar las extensiones de archivos para tipos conocidos` y la desenmarcamos.

[2] El comando `net localgroup SWAP` $/add$ me lo he comido, ya que en un primer intento ejecute ese comando correctamente y los demas no y en este segundo intento todos los demas comandos se han ejecutado correctamente pero el grupo SWAP ya lo tenia creado. Hay que ejecutar los comandos en el orden especificado en el punto 1.

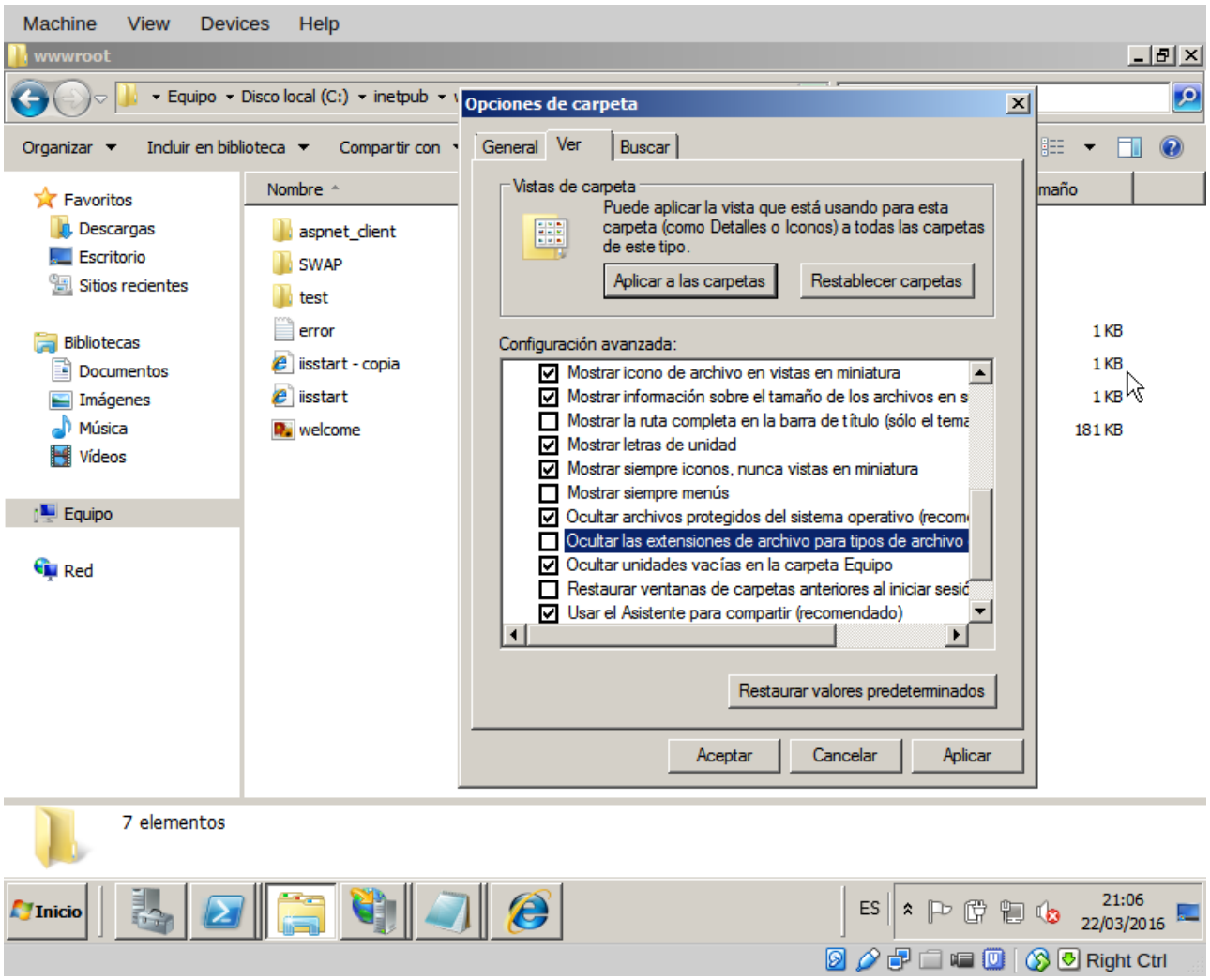

Figura 1.10: Habilitar ver extensiones de archivos: quitamos la opción de ocultar extensiones.

4. Creamos el directorio SWAP y dentro de este directorio creamos un archivo llamado “default.aspx” y pegamos el siguiente código

```
<%@Language="C#" %>
<%
    string currentUser = Request.ServerVariables["LOGON_USER"];
    if (currentUser == "")
        currentUser = "anonymous";
    Response.Write("<b>Current User:</b> " + currentUser);
%>
```

5. Dentro del mismo directorio SWAP creamos otro documento llamado documentoSecreoProfesor.aspx con el siguiente contenido:

```
<%@Language="C#" %>
<%
   string currentUser = Request.ServerVariables["LOGON_USER"];
    if (currentUser == "")
        currentUser = "anonymous";
    Response.Write("<b>Current User:</b> " + currentUser);
```

```
Response.Write("<b>My secret:</b> I used Apache before I discovered
    IIS7.</b> ");
%>
```

6. Ahora abrimos el internet explorer y ponemos en la barra de direcciones `localhost/SWAP/` y nos tiene que salir un mensaje en el que nos dice que usuario somos. A continuación probamos con `localhost/SWAP/documentoSecretoProfesor.aspx` y vemos si nos sale lo mismo que en

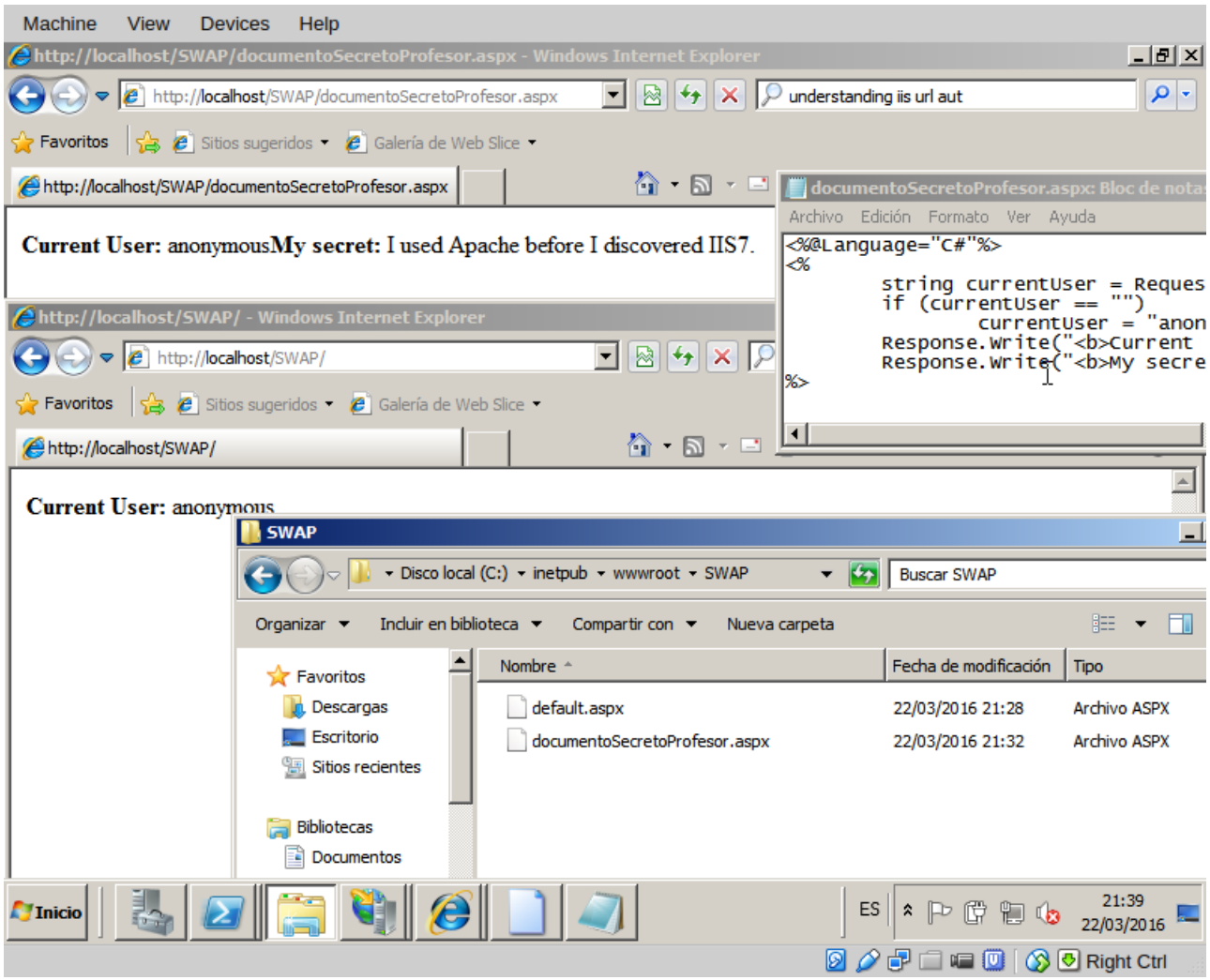

Figura 1.11: Tras escribir default.aspx y documentoSecretoProfesor.aspx abrimos el navegador web y si sale lo mismo que en esta imagen está todo correcto

1.2.3. Configurando autentificación y URL Authentication

En este punto es recomendable reiniciar el servidor por si acaso.
Volvemos otra vez a nuestra ventana 1.3 y en ella nos vamos a `Autentificación` .[3]
Desabilitamos `Autentificación anónima` y habilitamos `Autentificación básica`. 1.12

[3] Es importante estar en la pestaña de `Página principal de Default Web Site` y no en en la de otro sitio web o en la del servidor (WIN-TPJLLJG9...) porque sino la autenticación no la estaremos haciendo en el sitio web por defecto.

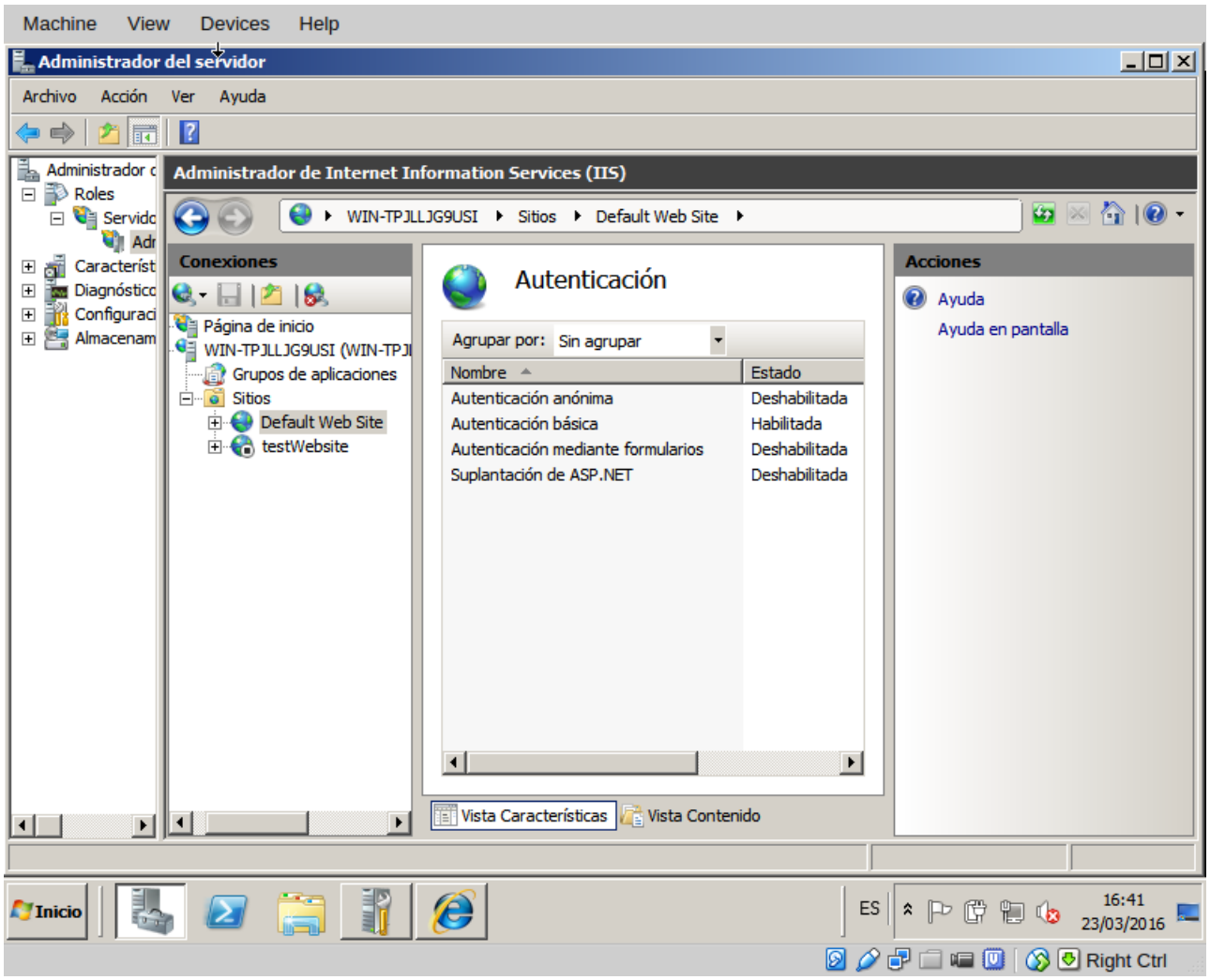

Figura 1.12: Desabilitamos autentificación anónima y habilitamos autentificación básica

Entramos en la dirección `localhost/SWAP/` y ahora nos pedirá el usuario y la contraseña 1.13.

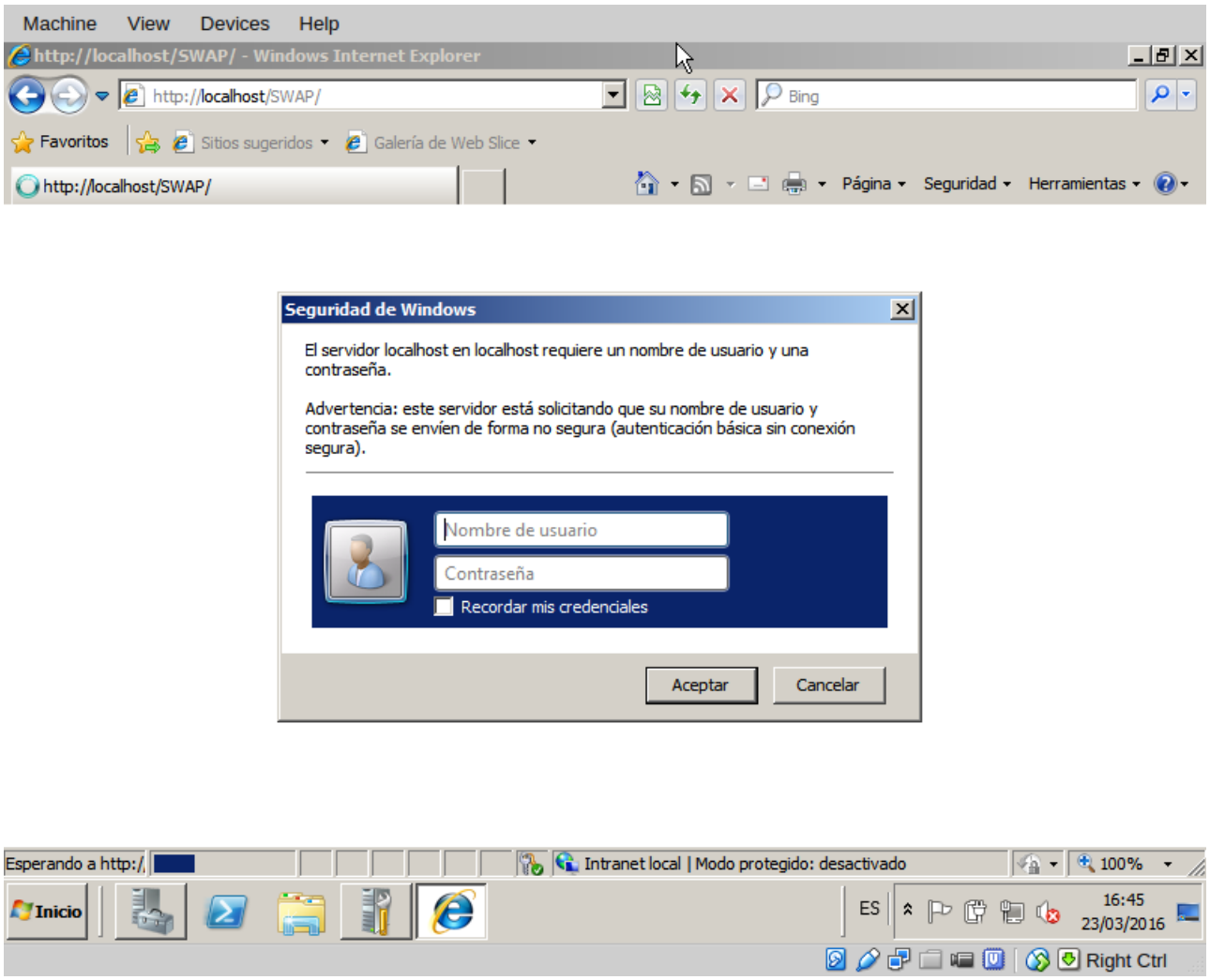

Figura 1.13: Al volver a entrar en la página `localhost/SWAP/` 1.11 ahora nos pide autentificación.

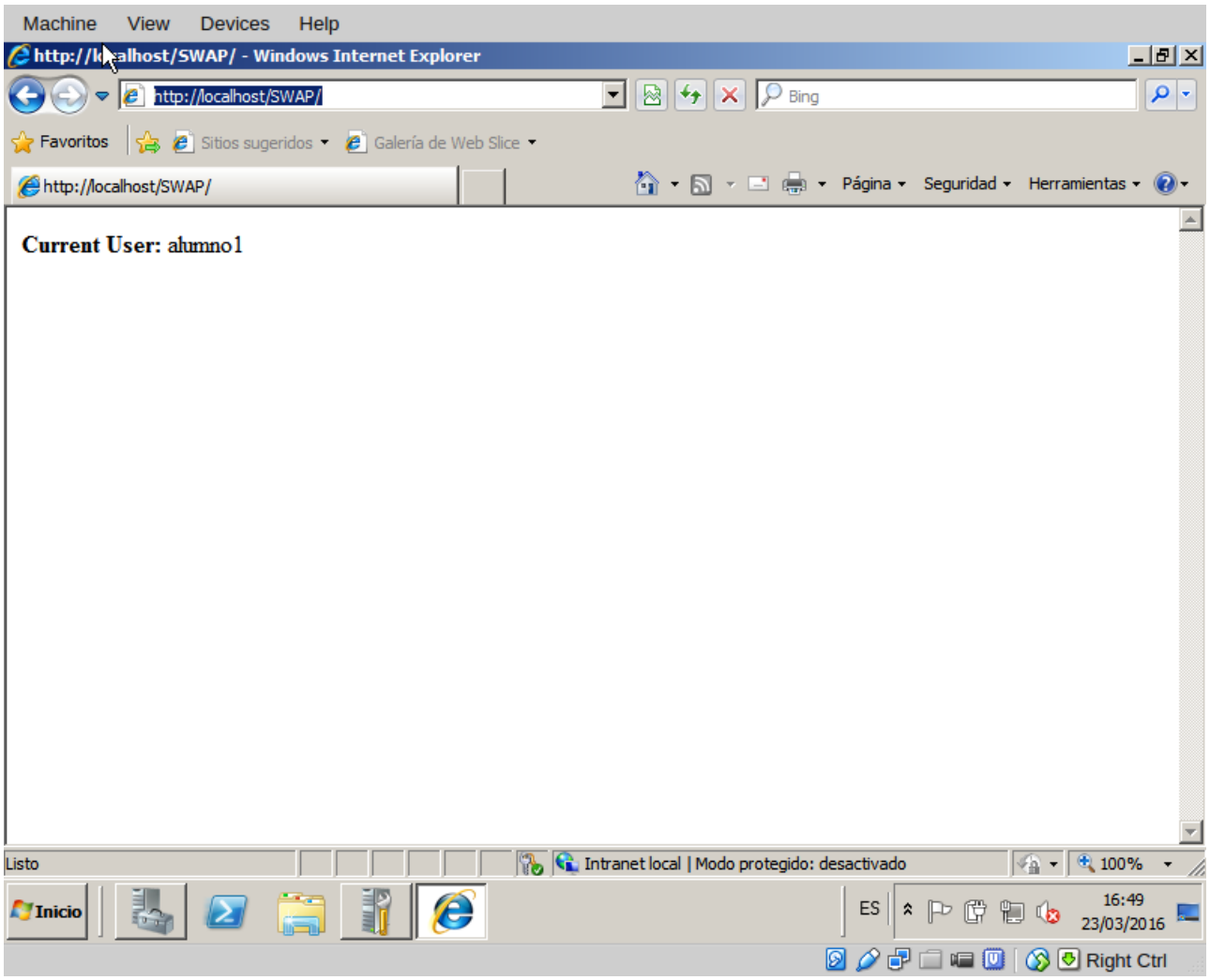

Figura 1.14: Tras insertar un usario válido al acceder a `localhost/SWAP/`

Ahora mismo cada vez que alguien intente entrar en nuestro servidor web le pedirá usuario y contraseña solamente dejando pasar a aquellos usuarios que han sido registrados en el sistema en nuestro casso alumno1, alumno2 y profesorPedro.

A continuación pasamos a restringir solamente a la página SWAP a los pertenecientes al grupo SWAP para ello nos vamos página de configuración IIS para esa página 1.3 y pulsamos sobre la carpeta SWAP y le damos a `Reglas de autorización.`

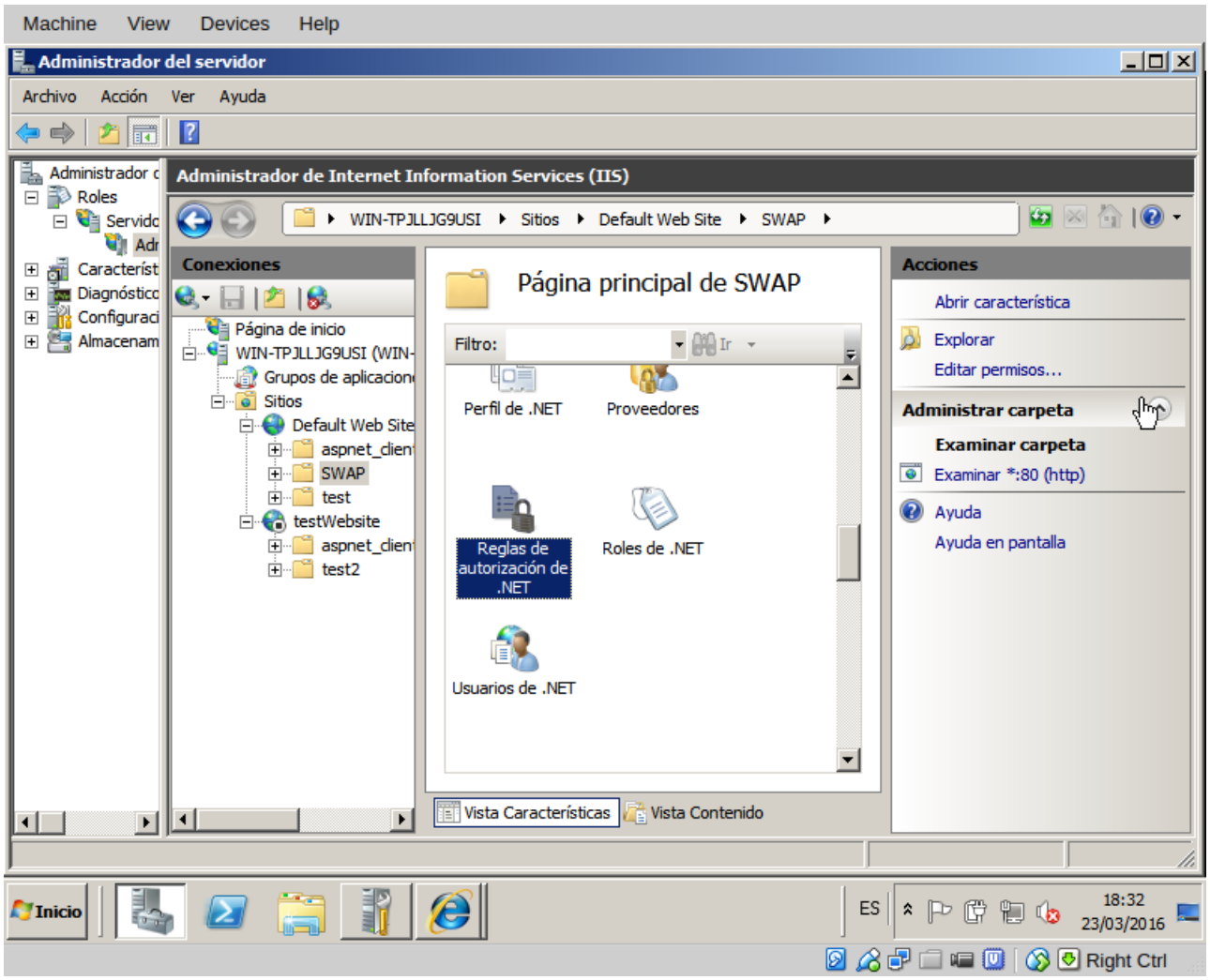

Figura 1.15: Tras pulsar sobre SWAP en la página de configuración del IIS nos vamos a `Reglas de autorización.`

En la ventana que se nos habre pulsamos sobre `Agregar reglas de permiso...` y a continuación dentro del apartado `Roles o grupos de usuarios especificados` introducimos SWAP2.2.

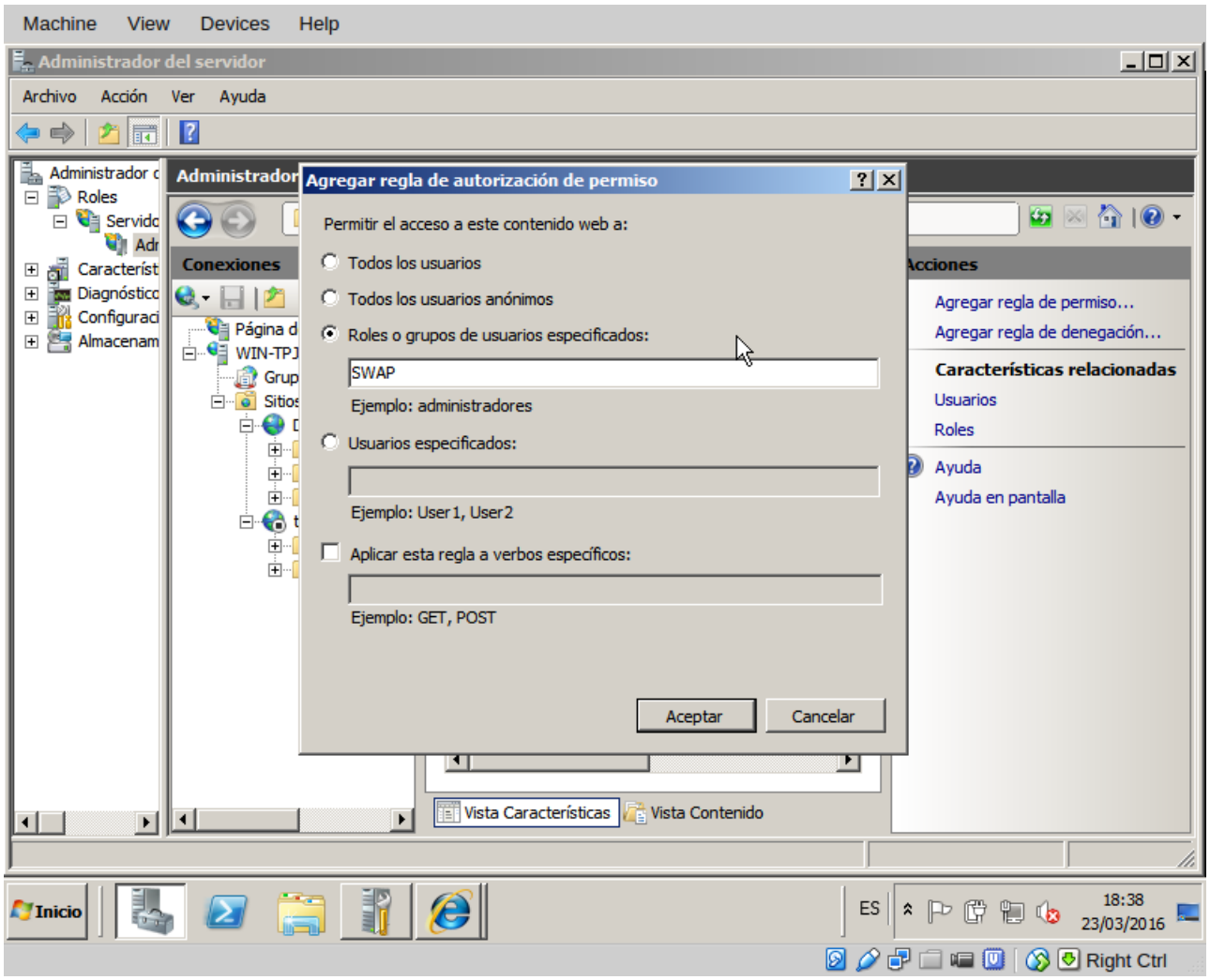

Figura 1.16: Tras pulsar sobre `Agregar reglas de permiso...` añadimos en el apartado `Roles o grupos de usuarios especificados` SWAP

Ahora al intentar acceder con un usario que no pertenece al grupo SWAP (ej: alumno2) se le deniega el acceso. 2.3

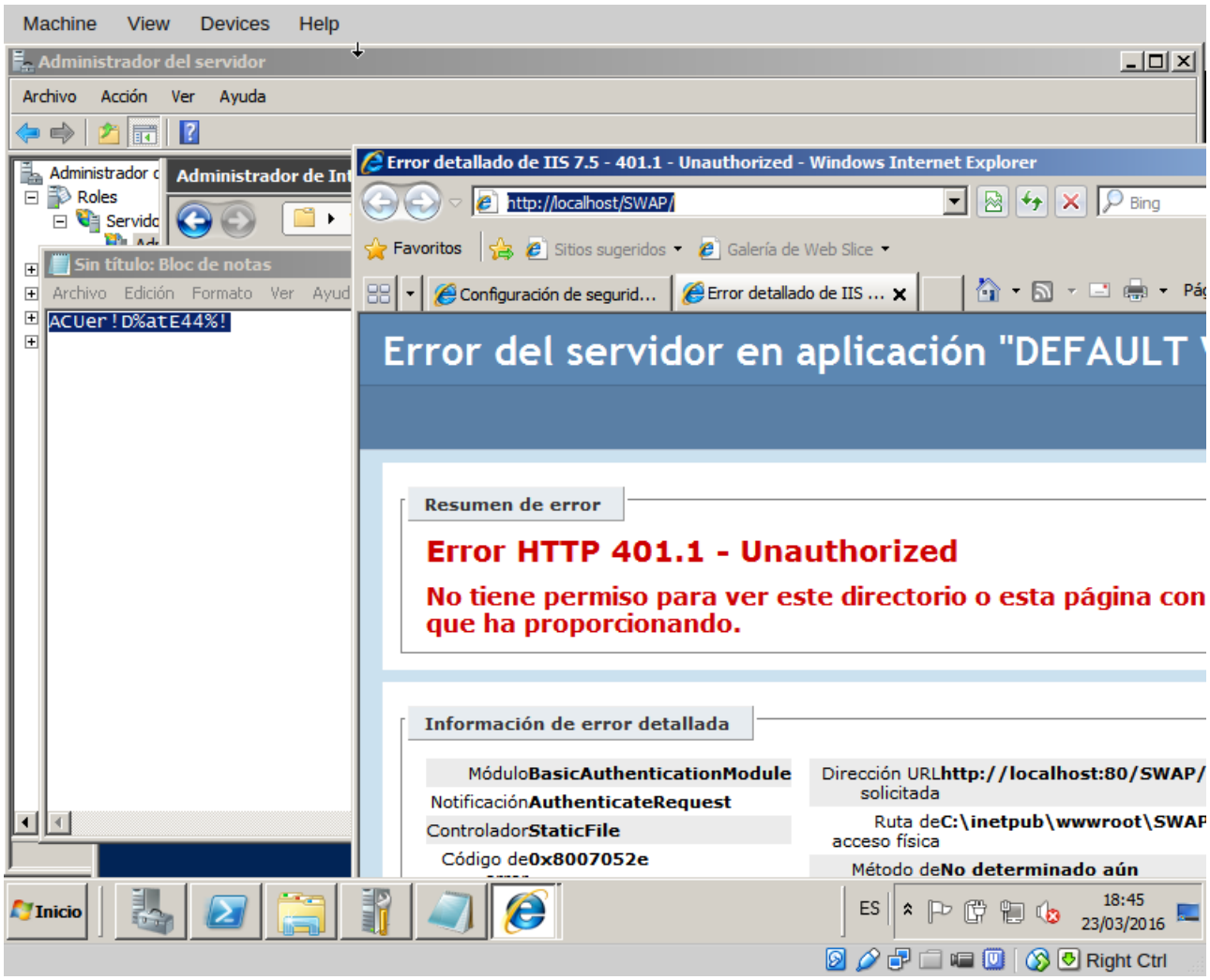

Figura 1.17: Tras intentar acceder a `localhost/SWAP` utilizando las credenciales del alumno2 comprobamos que este no tiene permitido el acceso.

Para rematar el tema vamos a denegar el acceso en lugar de a un directorio en global que puede albergar más de una página web vamos a restringir el acceso de una página web en concreto: documentoSecreto.aspx dentro del directorio SWAP de tal manera que solamente puede acceder el usuario profesorPEDRO.
En primer lugar nos vamos de nuevo a página de configuración IIS para esa página 1.3 y pulsamos sobre la carpeta SWAP. En la parte inferior pulsamos sobre una pestaña que pone vista de contenido y tras pulsar en esta pestaña pulsamos con el derecho sobre documentoSecreto.aspx. A continuación le damos a `Cambiar a vista de características` y ya es lo mismo que hizimos dos parrafos más arriba con la carpeta SWAP. Nos vamos a `Reglas de autorización`, agregar regla y dentro de usuarios especificos ponemos profesorPEDRO.

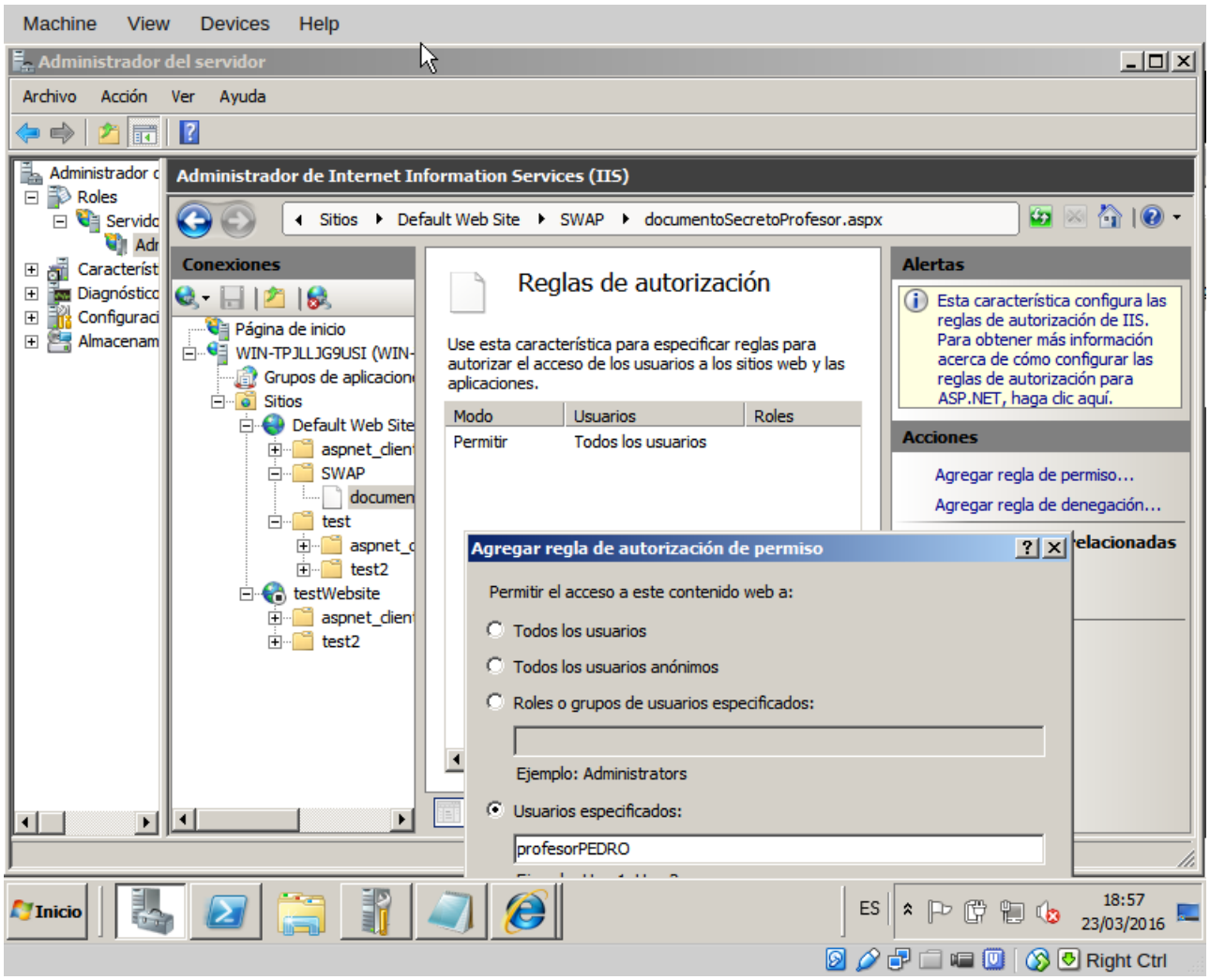

Figura 1.18: Agregamos regla de autorización para el usuario profesorPEDRO

Ahora solamente el usuario profesorPEDRO podrá acceder a esta página.

2. Nginx

Nginx es un servidor que surgió alla por 2002 para intentar hacer frente al problema a los que se enfrentaban los servidores de ser capaces de soportar decenas de miles de peticiones en poco tiempo a sus sitios web. De Nginx podemos decir que tiene fama de hacer un uso eficiente de los recursos y de ser capaz de funcionar bastante bien con un hardware básico. Suele ser una buena opción cuando se busca servir contenido estático y presenta alguna que otra complejidad cuando se quiere usar para servir contenido dinámico.
De como aumentar la seguridad de un servidor Nginx se puede encontrar muchísima información en Internet al contrario de lo que pasa con IIS de Microsoft en el que toda la información suele encontrarse en páginas oficiales.

2.1. Instalación y configuración inicial

Para instalar Nginx en cualquier distribución de Debian o derivados (Ubuntu Server 14.04 en mi caso) solamente hace falta hacer `sudo apt-get install nginx`.
Si todo va bien al poner en nuestro navegador o al hacer `curl ip_servidor_nginx` nos tiene que salir la página por defecto de Nginx.

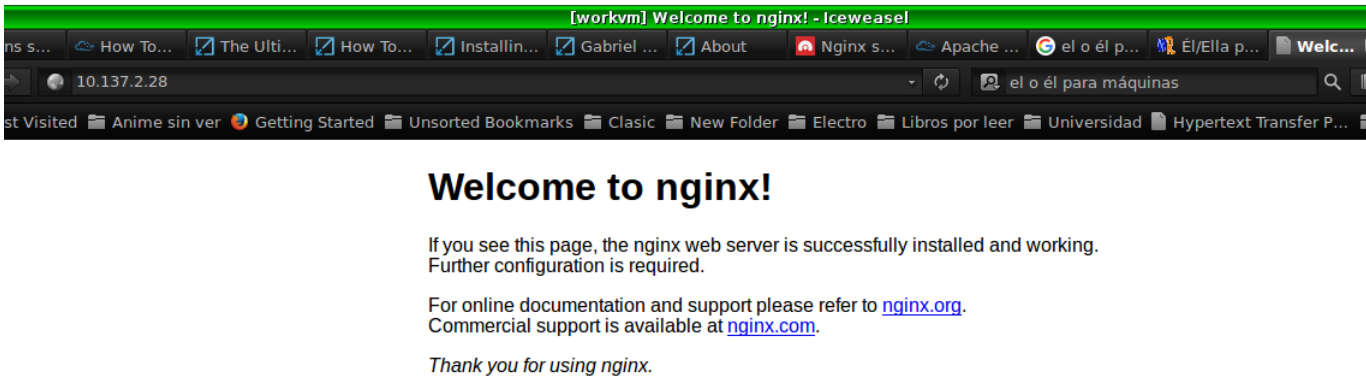

Figura 2.1: Tras instalar Nginx ponemos en el navegador la IP del servidor y nos sale página por defecto de Nginx

2.2. Asegurando Nginx

Para trastear con Nginx vamos a estar tocando el archivo de configuración `/etc/nginx/nginx.conf` y el archivo de configuración de nuestro sitio web `/etc/nginx/sites-available/default` o en mi caso `/etc/nginx/sites-available/default` ya creo un sitio distinto al que por defecto se crea en Nginx. Vease 5 para crear un sitio distinto al de por defecto que crea Nginx al intalarse. No va habrá mucha diferencia para los ejemplos que desarrollamos en este documento salvo que si se utiliza el sitio por defecto salvo que donde ponga editar `/etc/nginx/sites-available/swaptest` habrá que editarse `/etc/nginx/sites-available/default`.

2.2.1. Desabilitando server_tokens

Vamos a empezar por algo facil. Cada vez que Nginx muestra una página de error en ella nos dice la versión de Nginx que se está utilizando[6]. Esta información puede ser utilizada para buscar vulnerabilidades para esa versión de Nginx en especifico. Para eliminar esto nos vamos `/etc/nginx/nginx.conf` y en ella en la linea en la que aparece `server_tokens` la ponemos asi:
`server_tokens off`.
A continuación reinicamos Nginx (sudo nginx -s reload)

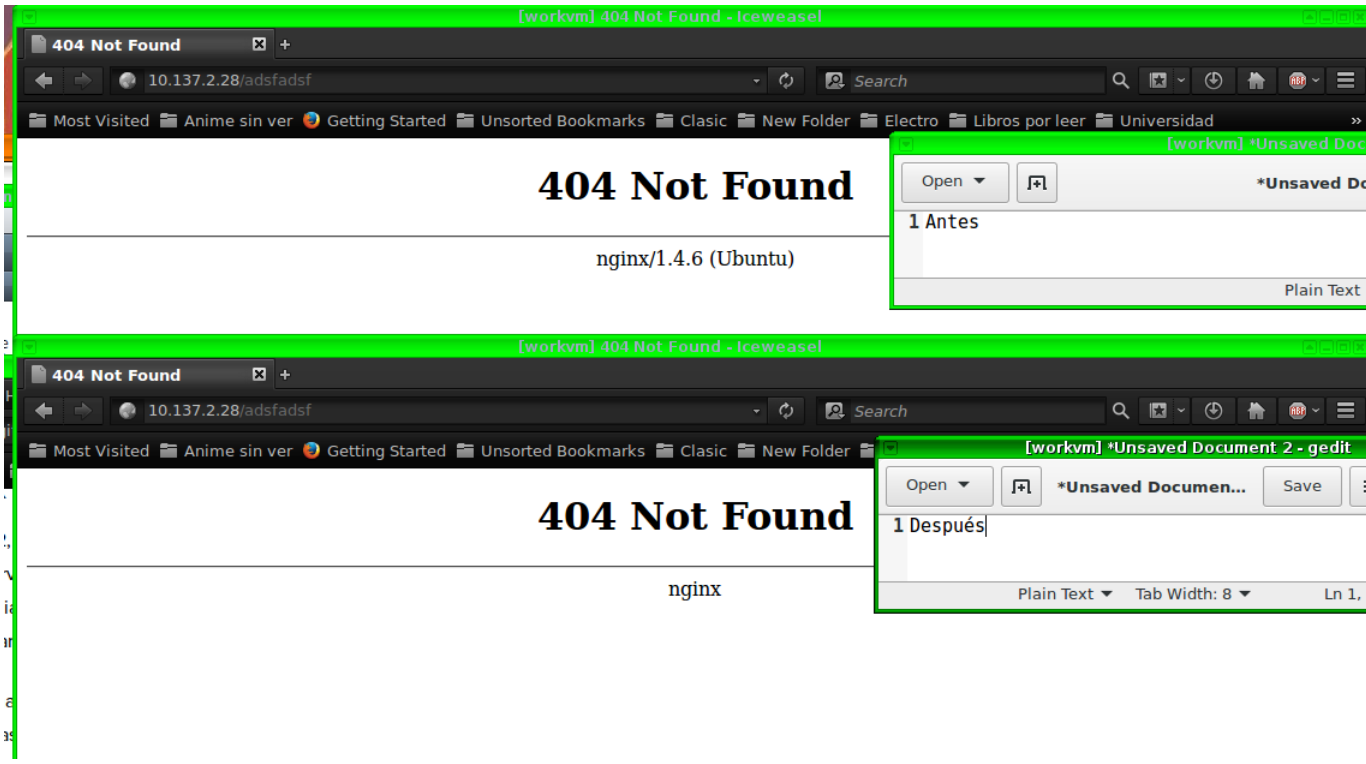

Figura 2.2: Arriba Nginx antes de desabilitar `server_tokens`, abajo tras desabilitar `server_tokens`

2.2.2. Desabilitando métodos no utilizados de HTTP

Básicamente los métodos de HTTP le dicen a nuestro servidor Nginx las acciones a llevar a cabo sobre un recurso de nuestro servidor. Los métodos más usuales son `GET` (obtener recursos del servidor), `POST` (mandar recursos al servidor), `HEAD` (debug)[5] los demás los desabilitamos. La justificación de porque desabilitar los métodos que no utilizamos básicamente es por eso de "mejor prevenir que curar" si no vamos a utilizar esos métodos mejor evitamos que se utilizen desabilitandolos. Algun posible ataque utilizando el método `TRACE` lo podemso ver en `http://publib.boulder.ibm.com/httpserv/ihsdiag/http_trace.html`.
Para desabilitar los métodos de HTTP que no deseamos utilizar en nuestro sitio web abrimos `/etc/nginx/sites-available/swaptest` y ponemos lo siguiente dentro de las llaves de `server`:

```
if ($request_method !~ ^(GET|HEAD|POST)$ )
{
    return 444;
}
```

La respuesta de código 444 significa que el servidor directamente a cortado la conexión sin devolver las cabeceras ni nada. Este tipo de respuesta puede utilizarse para confundir algún que otro ataque malware [7].

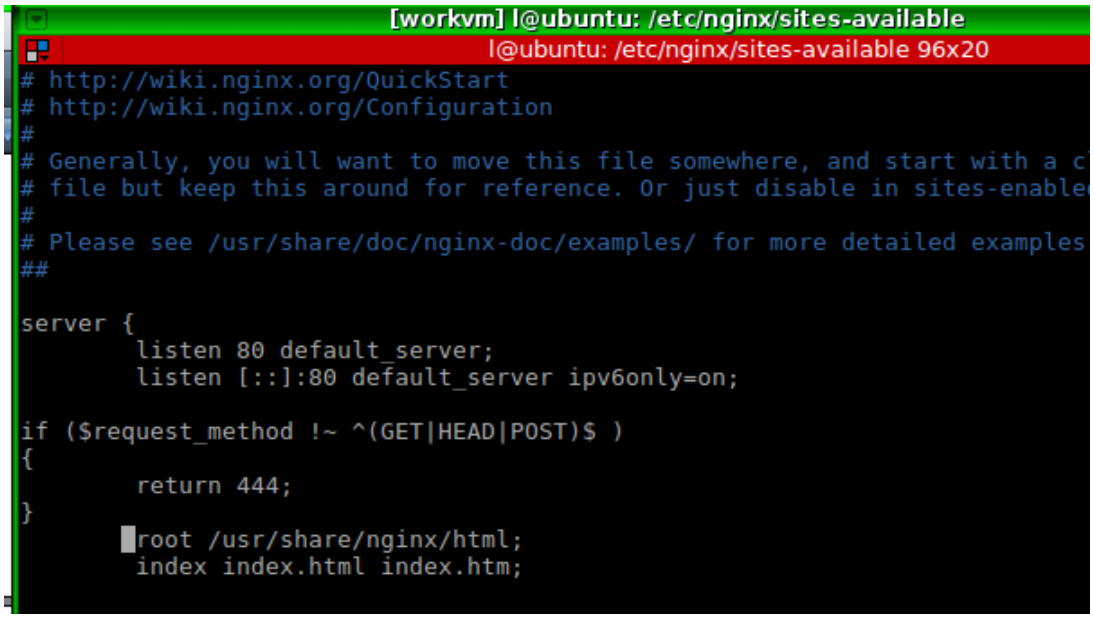

Figura 2.3: Archivo de congifuración `/etc/nginx/sites-available/swaptest` desabilitamos métodos usando if dentro del bloque server

En la siguiente figura 2.4 podemos ver a la izquierda el resultado de **no** desabilitar ningún método para nuestro sitio web a la derecha tras solo dejar habilitados `GET`, `POST` y `HEAD`.

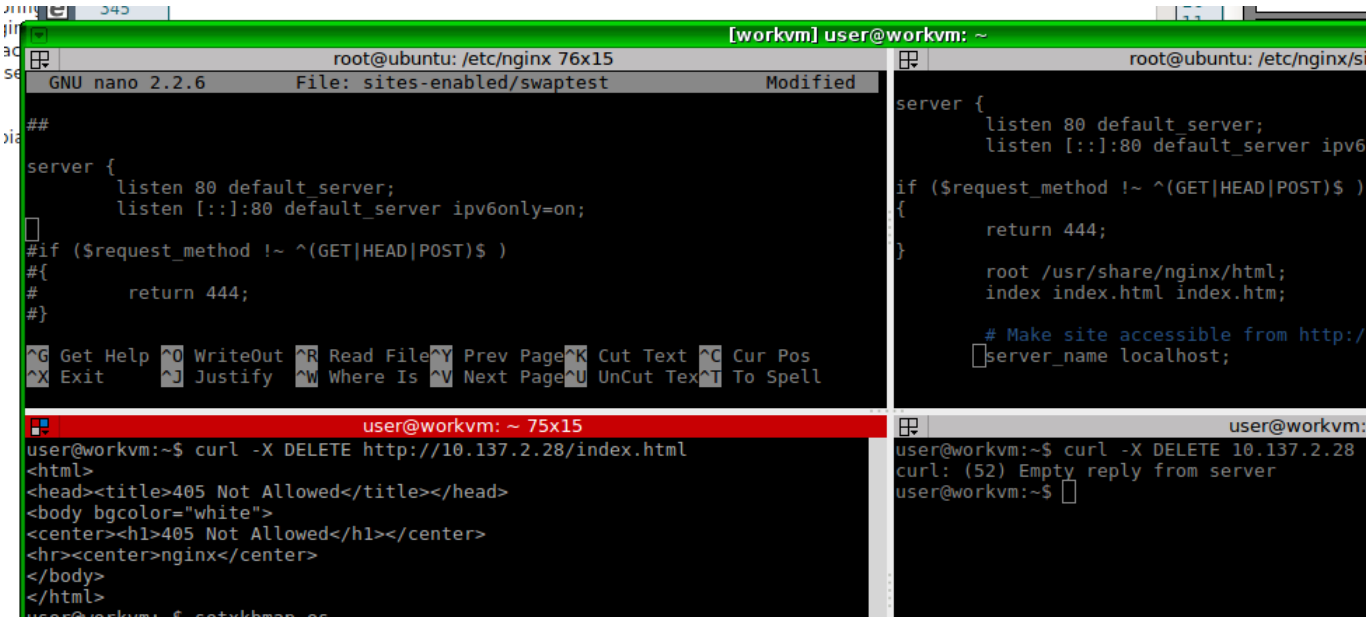

Figura 2.4: Izquierda haciendo curl a nuestro servidor con método `DELETE` habilitado, a la derecha desabilitado

3. Previniendo ataques de diccionario sobre ssh. Fail2ban

Si tenemos un servidor conectado a internet y con el cual es posible loguearse utilizando ssh entonces es susceptible de ser atacado utilizando una clase de ataques llamados de diccionario. Estos ataques consisten en teniendo una lista de strings almacenadas en un archivo de texto utilizamos cada una de estas cadenas para probar si pueden ser el

nombre de login o el password de un servicio como ssh.[8][4]

Vamos a intentar realizar este ataque. En primer lugar necesitamos un programa que nos permita de manera automática y efectiva lanzar todas las cadenas contenidas en un archivo para intentar loguearnos via ssh en nuestro servidor. Yo voy a utilizar Hydra[8]. Podemos instalarlo utilizando el gestor de paquetes de nuestra distribución.
Para hacerlo un poco más realista nos bajamos un diccionario por ejemplo el diccionario"twitter-banned.txt" y el de "facebook-firstnames.txt" de `https://wiki.skullsecurity.org/Passwords`3.1.

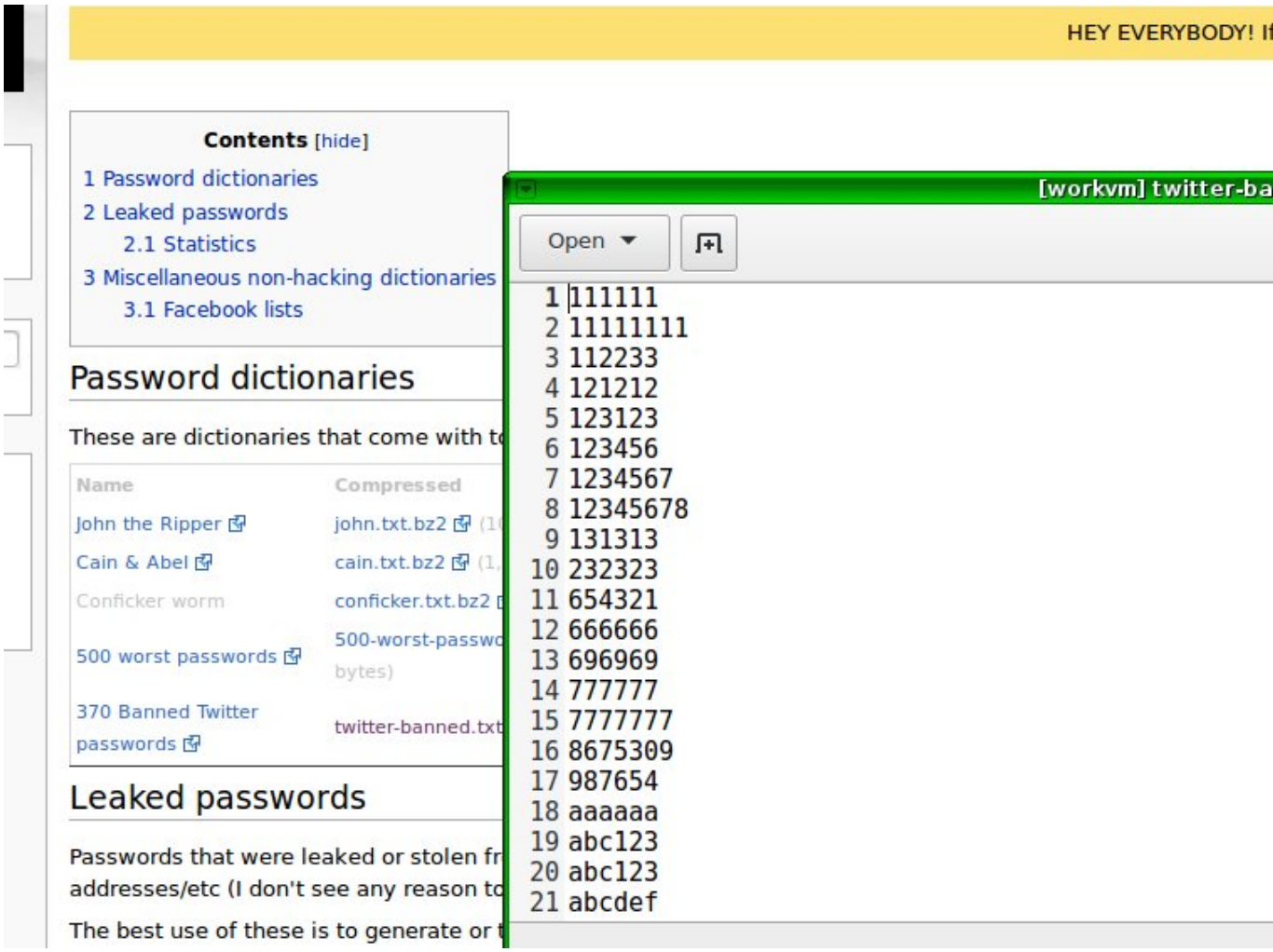

Figura 3.1: Contenido del diccionario "banned-twitter.txt"

En general no se suele conocer ni el usuario ni la contraseña asi que se suele utilizar dos diccionarios uno para intentar adivinar el usuario y otro para intentar adivinar la contraseña. Nosotros utilizaremos el de "facebook-firstnames.txt" para intentar adivinar el usuario y el de "twitter-banned.txt" para intentar adivinar la contraseña.
Con hydra utilizando la `-L` indicamos el fichero que contiene el diccionario de nombres de usuario y con `-P` indicamos el fichero que contiene el diccionario con las contraseñas y si ponemos en la orden `-t` indicamos el número de conexiones en paralelo que vamos a realizar contra el servidor. La orden completa sería:

[4]Fuente: www.fail2ban.org, man hydra

```
$ hydra -v -L 'facebook-firstnames.txt' -P 'twitter-banned.txt' -t 8 10.137.2.28
    ssh
```

Siendo 10.137.2.28 la ip a atacar y ssh indica el servicio al cual vamos a atacar en el servidor de esa ip.

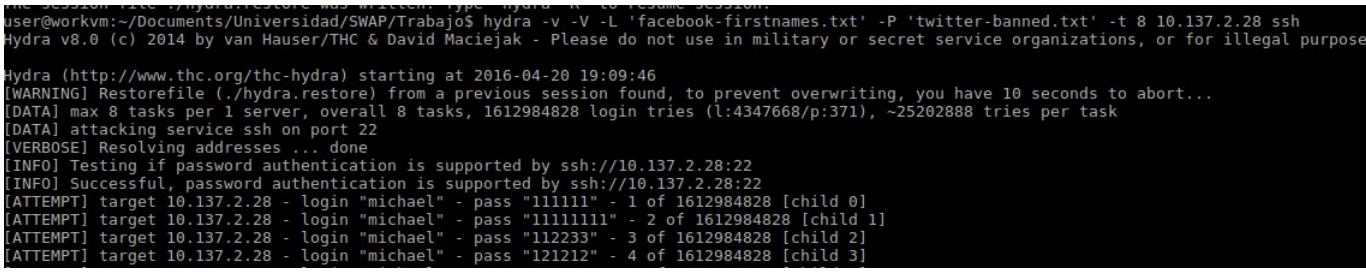

Figura 3.2: Ejecutando ataque de diccionario utilizando hydra sobre 10.137.2.28

Podemos ver en 3.2 como nos dice hydra que realiza 80 intentos de autenticación por minuto también si ponemos `-v -V` en la orden nos va diciendo que combinaciones de usuario-contraseña ha probando.

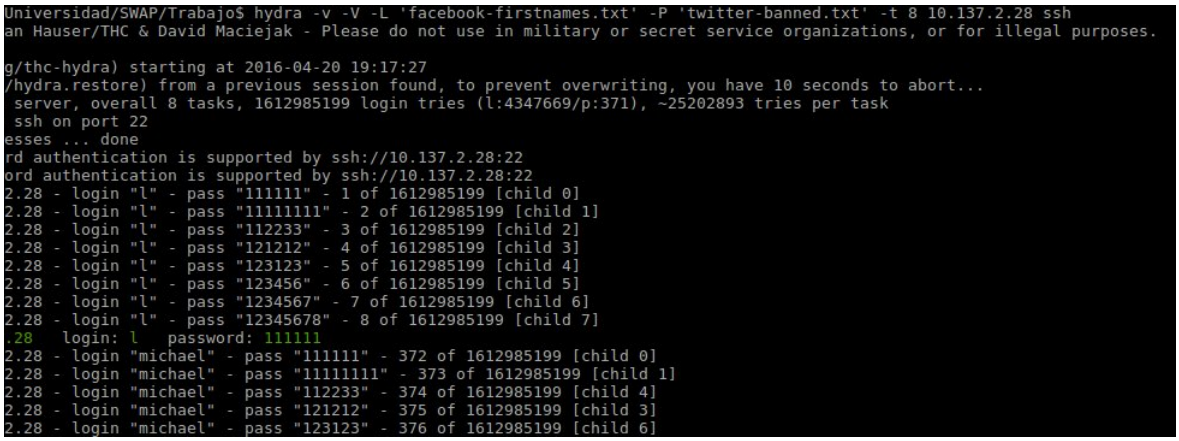

Figura 3.3: Ejecutando ataque de diccionario utilizando hydra sobre 10.137.2.28: combinación usuario: l contraseña:111111 nos permite loguearnos en el servidor utilizando ssh.

En 3.3 podemos ver como Hydra a encontra una combinación usuario-contraseña con la cual es posible loguearse en el servidor 10.137.2.28 utilizando ssh. Probamos si esto es correcto3.4.

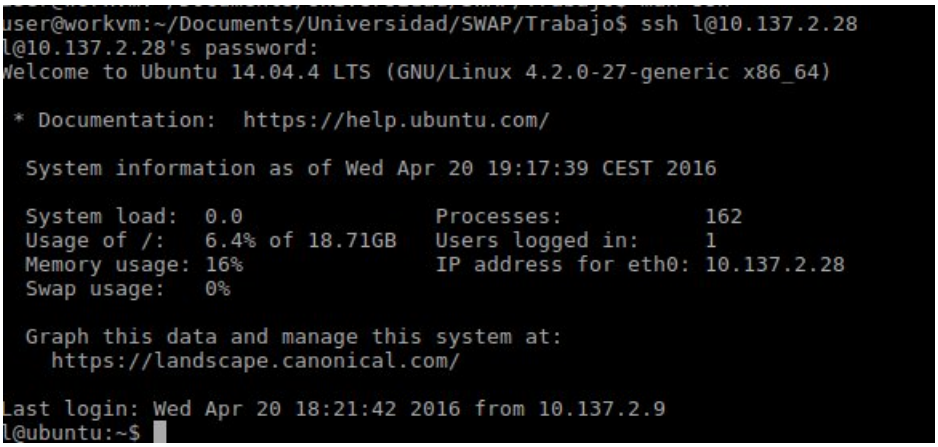

Figura 3.4: Utilizamos la combinación l-111111 para loguearnos satisfactoriamente en el servidor.

Para evitar este tipo de ataques podemos utilizar herramientas como fail2ban. Este programa mete en una lista negra aquellas IPs que intentan iniciar sesión de manera errónea. Su instalación es muy facil sencillamente utilizar apt o yum dependiendo de si utilizas Debian o Fedora para instalarlo.

Tras instalar fail2ban utilizando el gestor de paquetes procedo a modificar su archivo de configuración `/etc/fail2ban/jail.conf` [5] cambiando algunos parámetros como `bantime` a 30 para poder probarlo comodamente y no estar baneado toda la noche, también modifico dentro del apartado `[ssh]` el valor de `maxretry` a 2. La variable `maxretry` estable el máximo número de intentos erroneos que se permiten antes de ser baneado en un periodo de tiempo en mi caso 2 intentos por 30 segundos. Una vez realizado estos cambios reinicio fail2ban utilizando la orden `sudo service fail2ban restart`.

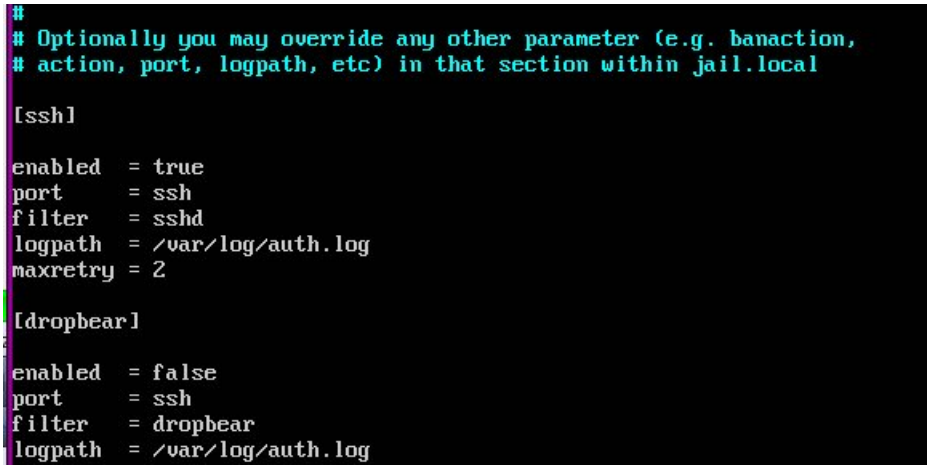

Figura 3.5: Fichero `/etc/fail2ban/jail.conf` modifico maxretry a 2 dentro del apartado [ssh].

Y realizo unos cuantos intento de login erroneos haber si me banea. Para comprobar si estas baneado ponemos en el terminal del servidor donde está instalado ssh:

[5] En otras versiones de fail2ban el archivo de configuración es jail.local.

```
$ sudo fail2ban-client status ssh
```

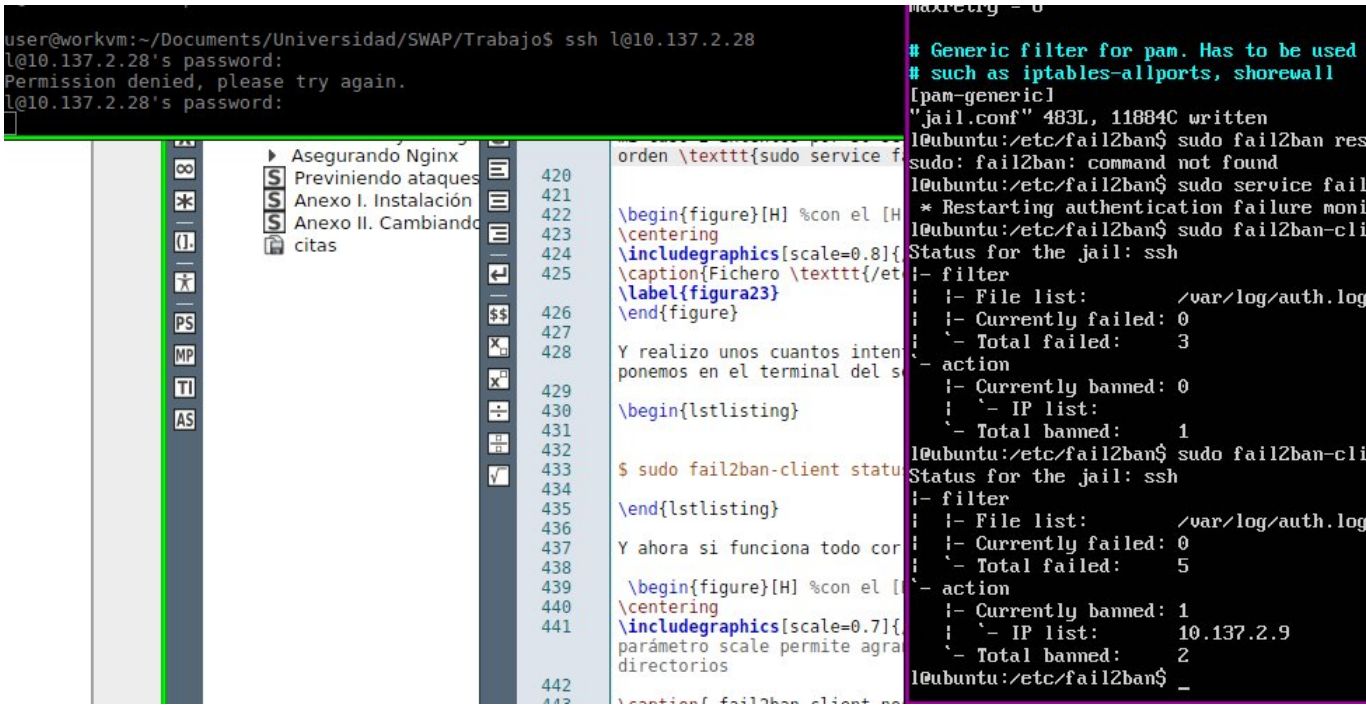

Figura 3.6: Tras el segundo fallo al iniciar sesión fail2ban nos banea quedandose en suspensión el intento de volver a insertar una nueva contraseña.

Como podemos ver en 3.6 tras 2 fallos erroneos al intentar iniciar sesión fail2ban nos banea quedandose colgado el intento de insertar una nueva contraseña.
Fail2ban bloquearía efectivamente los intento de intentar adivinar las combinaciones de usuario-contraseña para entrar en el servidor ya que a los x intentos erroneos la conexión se quedaría en suspenso añadiendose la IP infractora a una lista de IP malignas.

4. Anexo I. Instalación Windows Server 2008 y IIS

.
Para instalar Windows Server 2008 nos vamos a `https://www.microsoft.com/en-us/download/details.aspx?id=11093` y nos descargamos el .iso. Después ejecutamos Virtualbox o VMWare para instalar Windows Server 2008 utilizando este .iso.
En el proceso de instalación llegará un punto en el que se nos preguntará que versión de WS2008 queremos instalar, nosotros seleccionames la primera:

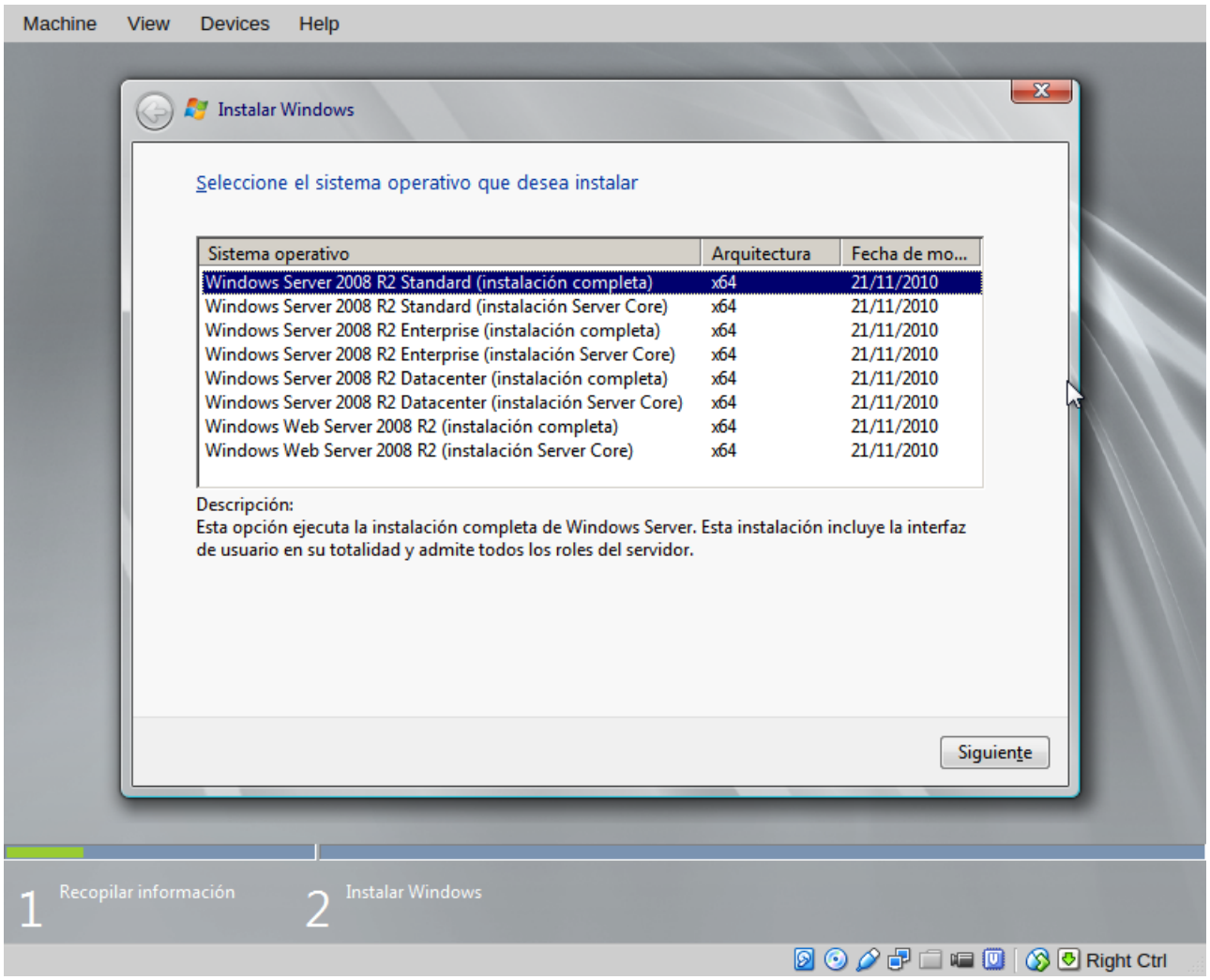

Figura 4.1: Instalación Windows: seleccionamos la primera opción.

Tras finalizar el proceso de instalación nos aparecerá una ventana llamada “Tareas de configuración inicial.” en ella nos vamos a la sección tres y seleccionamos “Añadir roles”. En la nueva pantalla que nos aparece seleccionamos “Servidor Web(IIS)”, siguiente e instalar.

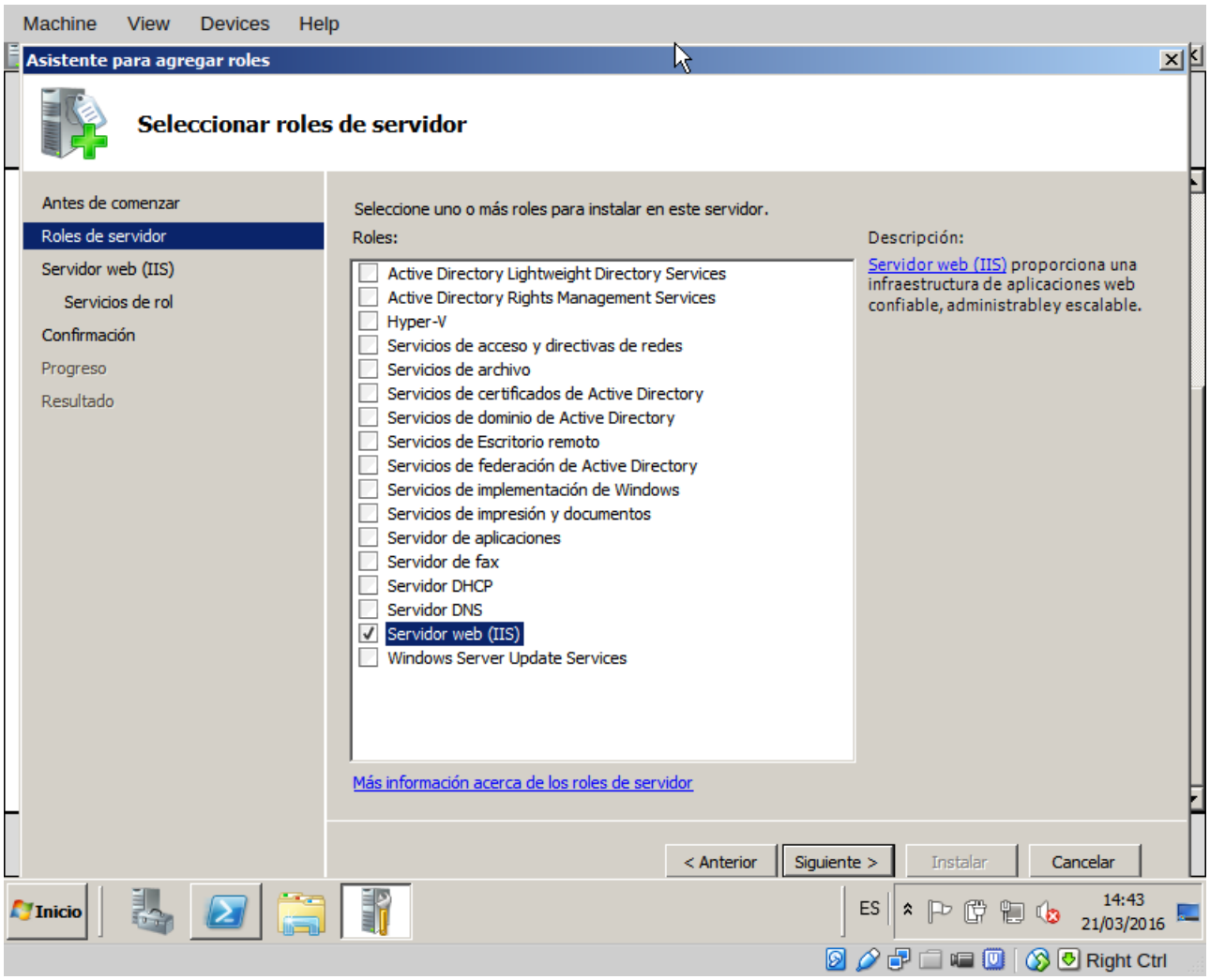

Figura 4.2: Instalación Windows: “Servidor Web(IIS)”

Le damos a cerrar abrimos powershell ponemos ipconfig y con el navegador de nuestro equipo host ponemos esa ip.

Figura 4.3: Comprobamos IIS funciona correctamente. IP navegador es la IP de Windows Server 2008.

5. Anexo II. Cambiando el sitio por defecto en Nginx

Vamos a crear un sitio por defecto al que se instala con Nginx para ello seguimos los siguientes pasos[7]:

1. Como queremos crear un sitio web del diferente al que hay por defecto en primer lugar nos desplazamos al directorio de configuración de Nginx que en Ubuntu Server es `/etc/nginx`.

2. Una vez alli eliminamos el archivo `sites-enabled/default`. Nginx permite alojar más de un sitio web, los sitios web que están activos (es decir que están habilitados que se puede acceder a ellos) tienen un enlace simbólico en este directorio que apunta a su archivo de configuración que se encuentra en el directorio `/etc/nginx/sites-available`.

3. A continuación ejecutamos `cp sites-available/default sites-available/swaptest` y le añadimos algo a la página por defecto que tiene Nginx haciendo:

```
# echo "Trabajo SWAP" > /usr/share/nginx/html/index.html
```

6

4. Realizando un enlace simbolico en `sites-enabled` nuestro sitio web pasa a ser accesible en el servidor Nginx:

[6] Puede que la ruta donde se encuentran los archivos html en otros equipos sea `/var/www/html/index.html`. En caso de de duda abrir `/etc/nginx/sites-available/default` y buscar por algo parecido a `root /usr/share/nginx/html`

```
# ln -s /etc/nginx/sites-available/swaptest sites-enabled/swaptest

# service nginx reload
```

5. Nos tiene que salir al poner en el navegador la dirección IP del servidor donde se encuntra Nginx instalado la cadena que hemos introducido en el paso 35.1.

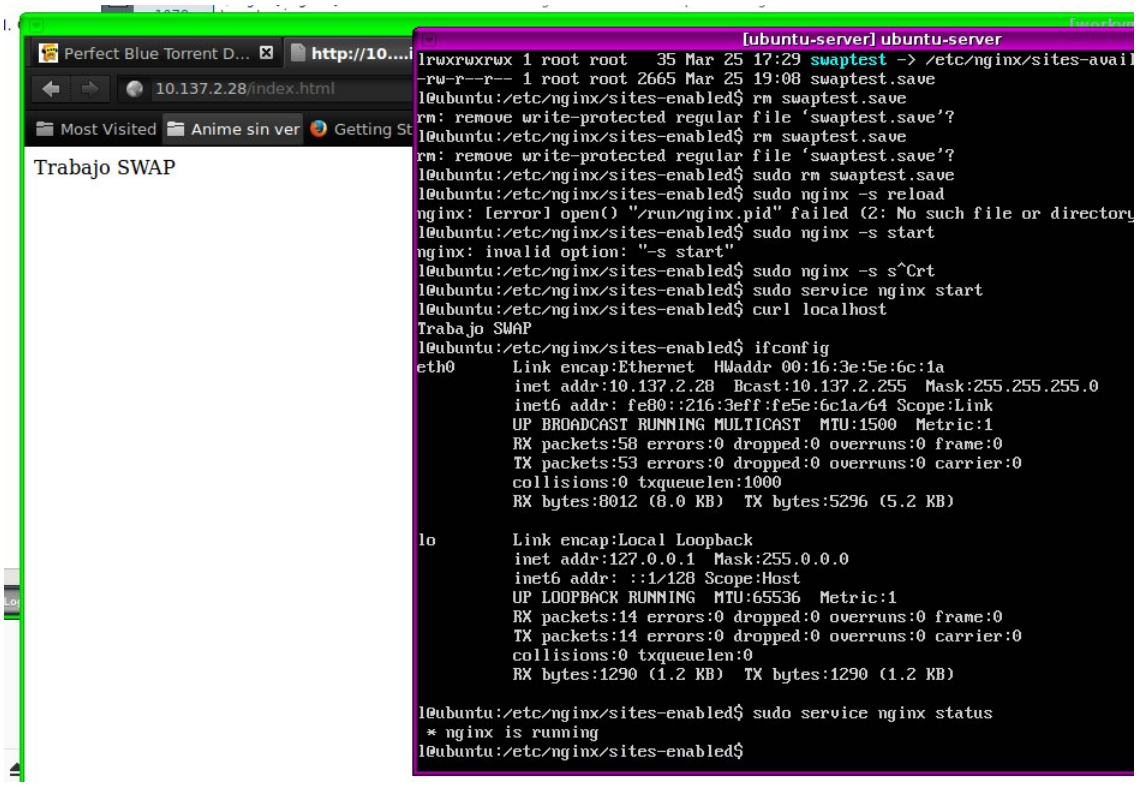

Figura 5.1: Comprobamos que al introducir la dirección IP del servidor Nginx obtenemos la cadena introducia en el paso 3

Referencias

[1] https://technet.microsoft.com/en-us/library/cc753433%28v=ws.10%29.aspx, consultada el 22 de Marzo de 2016.

[2] https://www.iis.net/configreference/system.webserver/security/authorization, consultada el 22 de Marzo de 2016.

[3] https://www.iis.net/configreference/system.webserver/security/ipsecurity, consultada el 22 de Marzo de 2016.

[4] http://www.iis.net/learn/manage/configuring-security/understanding-iis-url-authorization#Prerequisites, consultada el 22 de Marzo de 2016.

[5] https://tools.ietf.org/pdf/rfc2616.pdf, consultada el 23 de Marzo de 2016.

[6] http://www.tecmint.com/installing-lamp-and-lemp-on-debian-8-jessie/3/, consultada el 23 de Marzo de 2016.

[7] http://www.tecmint.com/installing-lamp-and-lemp-on-debian-8-jessie/3/, consultada el 23 de Marzo de 2016.

[8] http://sectools.org/tool/hydra/, consultada el 30 de Abril de 2016.