

DIAWEB 2.0

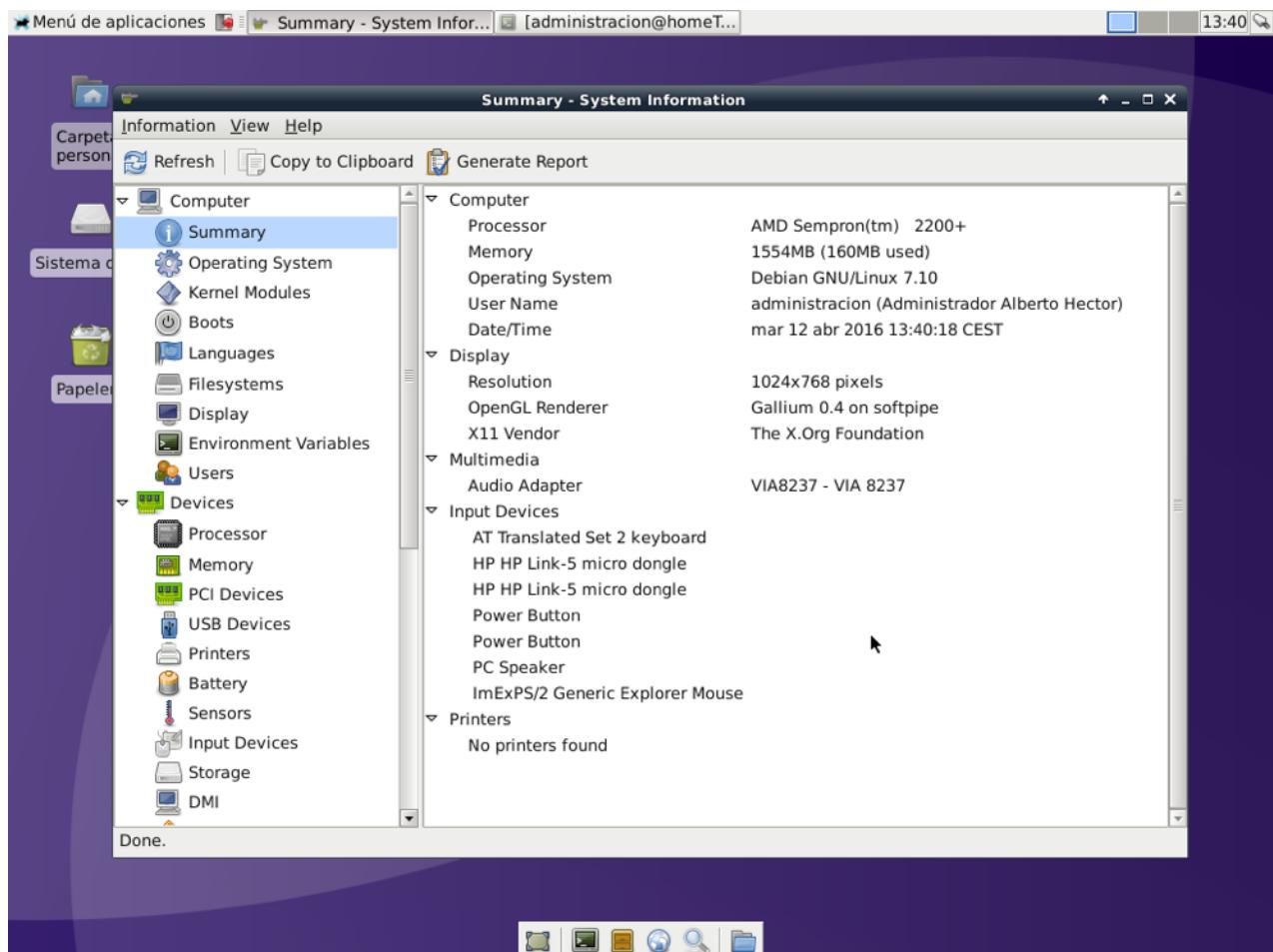
A&S Software Solutions

Índice

Introducción	3
Adecuación del entorno de desarrollo	5
Cambio de Hostname	9
Instalación Servidor SSH + cuenta de administración	10
Componentes utilizados	13
Instalación de MySQL	16
Instalación de Apache2	17
Configuración de Apache2 - Sirviendo en https	18
Configuración de Apache 2 - Permitiendo ejecución de .cgi (o .pl)	20
Configuración de Apache2 - Activando las páginas de usuario	22
Instalación de la funcionalidad mail requerida	23
Instalación del servidor ftp	26
Instalación web ftp client (nube personal + apuntes)	27
Instalación de la plataforma de estudios Moodle	29
Creación del proyecto Django y conexión con MySQL	31
Configuración de Apache2 - Servir la aplicación Django	36
Instalación de paquetes necesarios	37
Configuración del sistema de logs	38
Gestión de usuarios de Moodle con moosh	39
Diaweb 2.0 - El sistema	40
Despliegue en el entorno de producción	47
Últimas notas	50
Tour	52
Conclusión	65

Introducción

La práctica se realizará íntegramente, con motivo de realización de la memoria con capturas, en la siguiente máquina.



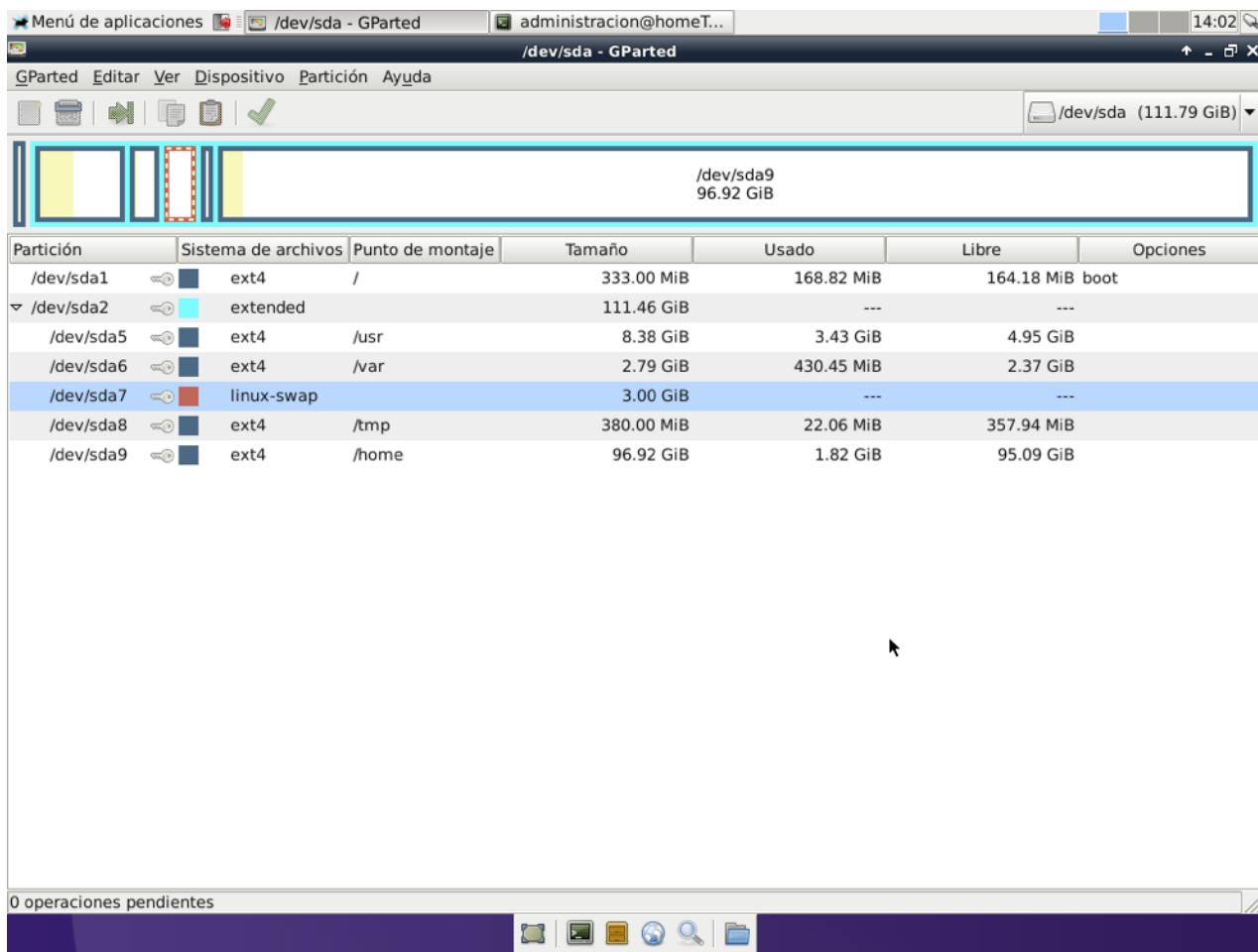
La solución se puede aplicar a cualquier sistema basado en Debian, aún así destacamos:

Se trata de una vieja torre ejecutando Debian 7 con entorno de escritorio XFCE.

Cuenta con un procesador mono-núcleo AMD Sempron a 2.2 Ghz, 1.5 Gb de RAM y un HDD de 120 Gb.

El sistema de ficheros es el creado por el instalador de Debian al seleccionar que se utilice todo el disco, sin especificar las particiones a nuestro gusto.

El estado del disco duro se muestra en la siguiente captura.



Anotaciones:

La práctica en este equipo se realiza teniendo en cuenta el espacio de este sistema 120gb y su configuración de internet (acceso total a la configuración del router que gobierna la subred). Este equipo toma el rol de **entorno test** en nuestro proceso de desarrollo.

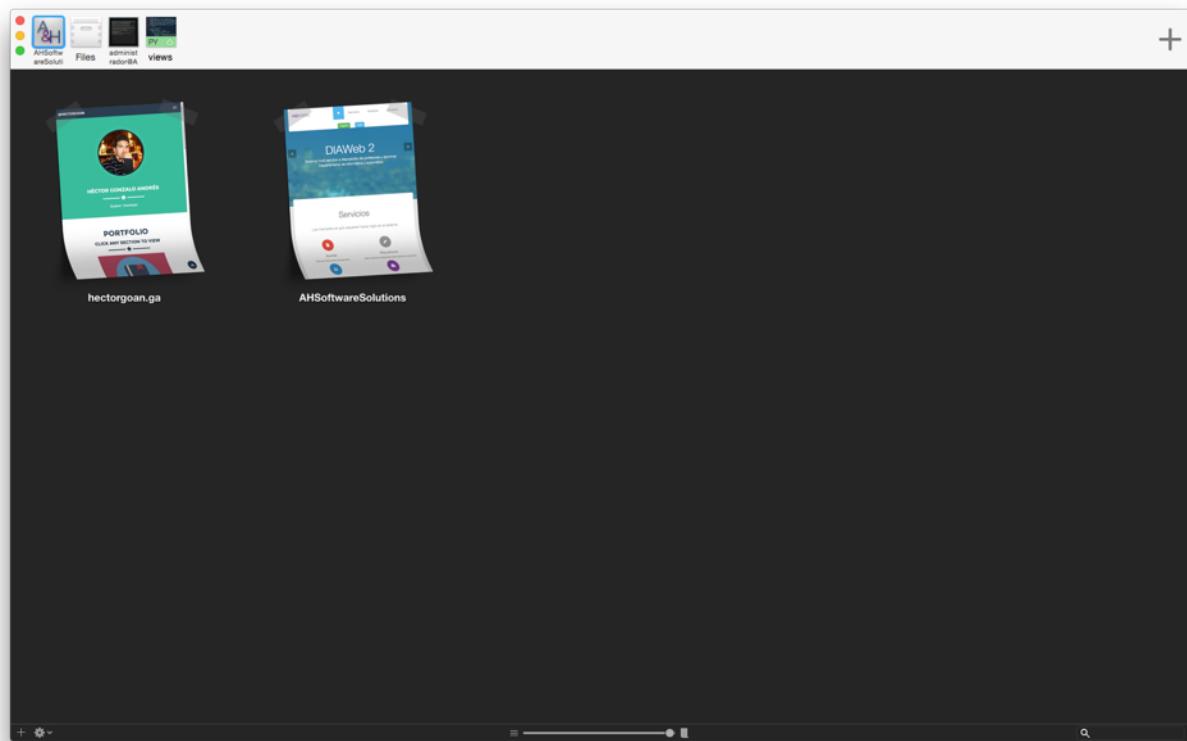
Al momento de montar el sistema en los equipos del laboratorio tuvimos que hacer ciertos cambios causados principalmente por esos dos aspectos (espacio y configuración de red). Estos cambios serán detallados, así como la solución a otros problemas surgidos durante el despliegue en el laboratorio (tomando este el rol de **entorno de producción**), más adelante en este documento.

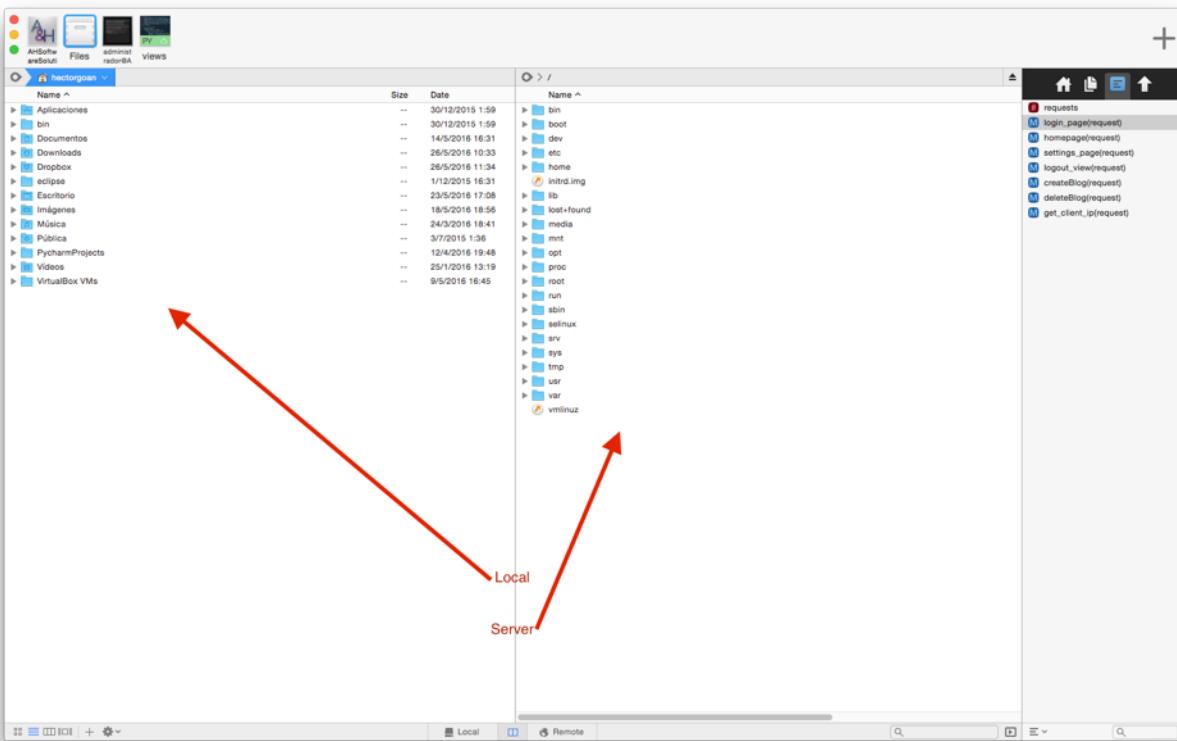
Adecuación del entorno de desarrollo

Para la realización de este trabajo necesitábamos un entorno sencillo de edición de archivos en el servidor, así como de carácter IDE ya que al realizar el código de la parte web con Django (spoiler), nos venía genial un editor con resaltado de síntesis Python.

Se aborda el problema desde dos perspectivas distintas dependiendo del S.O. del entorno de trabajo.

Mac OS X: En este SO se ha utilizado principalmente Coda 2 (<https://panic.com/coda/>), por ofrecer este una perfecta solución sftp para la gestión de archivos en el escritorio, junto con un editor con resaltado de texto y autocompletado para muchos lenguajes, así como con acceso por terminal al servidor integrado perfectamente con el resto del programa.





```

Warning: the RSA host key for '[hectorgoan.noip.me]:130' differs from the key for the IP address '[85.54.48.239]:130'
Offending key for IP in /Users/hectorgoan/.ssh/known_hosts:1
Matching host key in /Users/hectorgoan/.ssh/known_hosts:8
Are you sure you want to continue connecting (yes/no)? yes
Authenticated to hectorgoan.noip.me ([85.54.48.239]:130).
Linux #50SoftareSolutions 3.2.0-4-686-pae #1 SMP Debian 3.2.78-1 i686
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
Last login: Mon May 29 12:24:13 2016 from hectorgoan.noip.me
administrator@#50SoftareSolutions:~$ uname -a
Linux #50SoftareSolutions 3.2.0-4-686-pae #1 SMP Debian 3.2.78-1 i686 GNU/Linux
administrator@#50SoftareSolutions:~$ 
```

A dashed square icon is shown in the bottom right corner of the terminal window, with the text 'Open a document to display list of its elements.' below it.

```

# coding=utf-8
from django.template import RequestContext
from django.shortcuts import render_to_response, redirect
from myapp.forms import LoginForm, SettingsForm
from django.contrib.auth.decorators import login_required
from django.contrib.auth.models import User
import requests

def login_page(request):
    message = ''
    if request.method == "POST":
        form = LoginForm(request.POST)
        if form.is_valid():
            username = form.cleaned_data['username'].encode("utf-8")
            password = str(request.POST['password']).encode("utf-8")
            user = authenticate(username=username, password=password)
            if user is not None:
                if user.is_active:
                    login(request, user)
                    mode = get_client_ip(request)
                    mode = "debug"
                    if username == "administrador":
                        mode = "passwordOff"
                    #Call cgi script to notify that "administrador" have logged in
                    cgiURL = "https://hectorgoan.noip.me/cgi-bin/notification.pl?xxxxxx"
                    url = cgiURL
                    payload = {'ip': ip, 'hash': alhash, 'mode': mode}
                    r = requests.post(url, data=payload, verify=False)

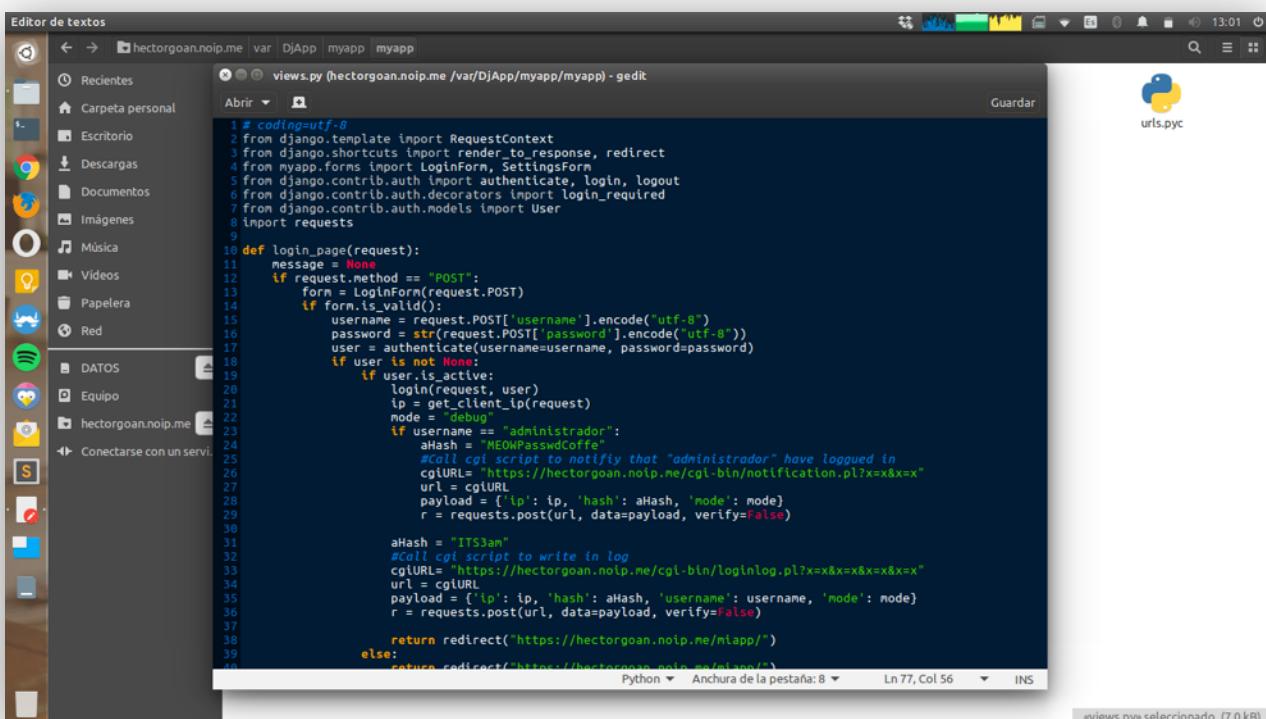
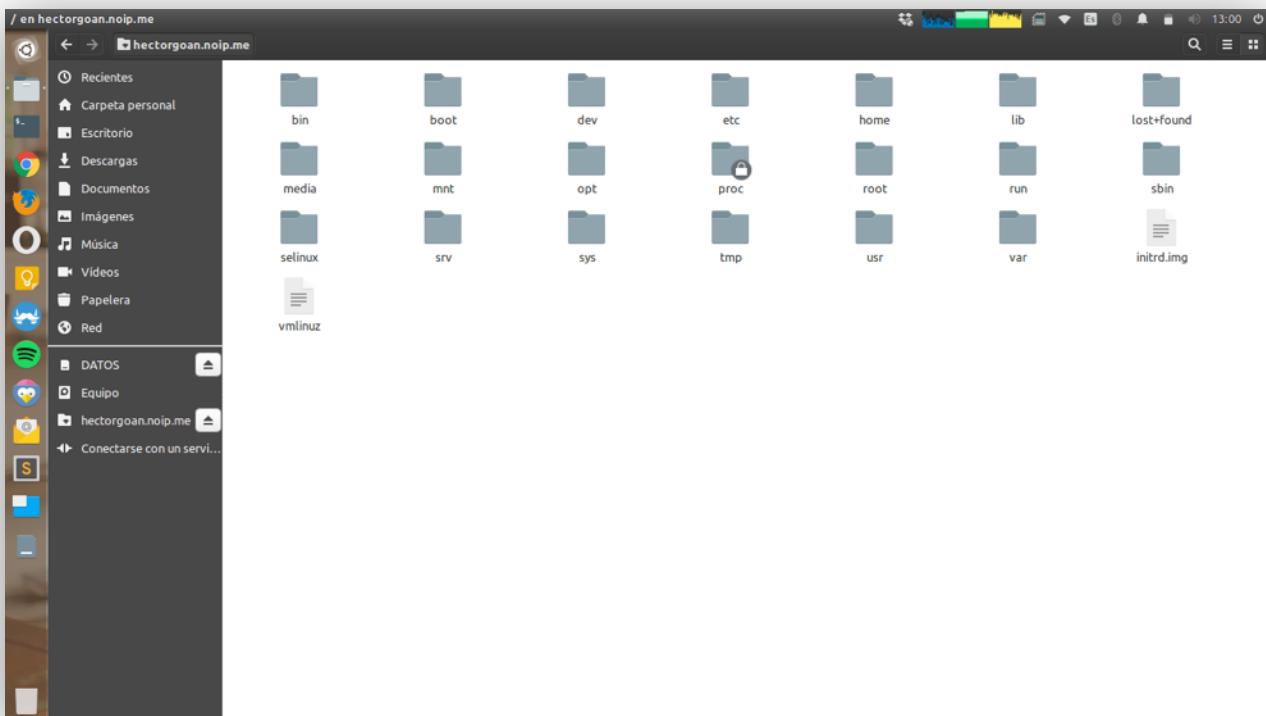
                    alhash = "IT53am"
                    #Call cgi script to write in log
                    cgiURL= "https://hectorgoan.noip.me/cgi-bin/loginlog.pl?xxxxxx"
                    url = cgiURL
                    payload = {'ip': ip, 'hash': alhash, 'username': username, 'mode': mode}
                    r = requests.post(url, data=payload, verify=False)
                    return redirect("https://hectorgoan.noip.me/miapp/")
                else:
                    return redirect("https://hectorgoan.noip.me/miapp/")
            else:
                ip = get_client_ip(request)
                mode = "warning"
                if username == "administrador":
                    alhash = "W0mPassw0r7"
                    #Call cgi script to notify that "administrador" have logged in
                    cgiURL = "https://hectorgoan.noip.me/cgi-bin/notification.pl?xxxxxx"
                    url = cgiURL
                    payload = {'ip': ip, 'hash': alhash, 'mode': mode}
                    r = requests.post(url, data=payload, verify=False)
                    message = "Nombre de usuario y/o password incorrecto"
                else:
                    form = LoginForm()
        return render_to_response('login.html', {'message': message, 'form': form}, context_instance=RequestContext(request))
    @login_required
    def homepage(request):

```

Linux (Ubuntu 16.04): En este SO se ha utilizado una combinación de Nautilus (gestor de archivos de Gnome) junto con Gedit (por mostrar este mejor capacidad para subida de archivos que Sublime Text 3).

Nautilus: <https://wiki.gnome.org/Apps/Nautilus>

Gedit: <https://wiki.gnome.org/Apps/Gedit>

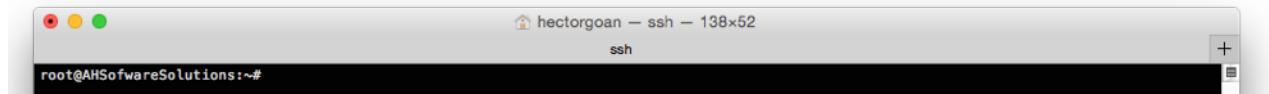
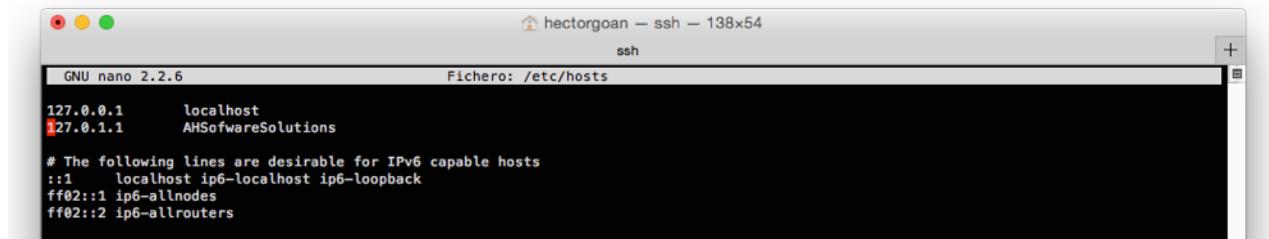
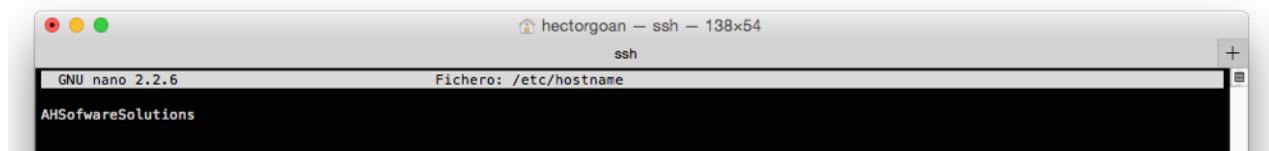
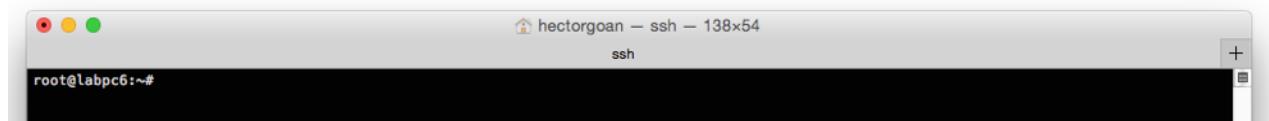


Cambio de Hostname

Tal como se indicaba en el enunciado de la práctica, tenemos que cambiar el hostname de nuestro servidor para coincidir con el nombre de nuestra empresa. En el entorno test, como la instalación del sistema operativo la hicimos nosotros, ya tenemos el nombre de host como queríamos desde un principio, pero en el entorno de producción nos encontramos que el nombre era genérico.

Por tanto, para realizar el cambio de nombre:

```
1      #loggued as root
2      #aquí escribiremos el hostname que queramos
3      nano /etc/hostname
4      #escribimos el nuevo nombre para la ip 127.0.1.1
5      nano /etc/hosts
6      reboot
```



Instalación Servidor SSH + cuenta de administración

Para trabajar mas cómodamente desde nuestro equipo de escritorio, tiene sentido instalar el servidor ssh.

Antes de instalar ningún paquete, actualizamos la información de los repositorios e instalamos actualizaciones

```
1 #Logged as root
2 apt-get update
3 apt-get upgrade
4 apt-get dist-upgrade
```

Una vez que tengamos todas las actualizaciones instaladas, podemos instalar ssh

```
1 #Logged as root
2 apt-get install ssh
```

A continuación pasaremos a configurar ssh.

```
1 #Logged as root
2 nano /etc/ssh/sshd_config
```

En el archivo que se abre:

```
#Cambiamos puerto a 130
Port 130

#15 segundos de tiempo de gracia
LoginGraceTime 15

#No permitiremos el acceso ssh al ususario root
PermitRootLogin no

#Solo permitimos acceso ssh al usuario administrador
AllowUsers administrador

#Limitamos el numero de intentos de poner contraseña a 3
MaxAuthTries 3
```

Guardamos el documento y reiniciamos el servidor ssh

```
1 service ssh restart
```

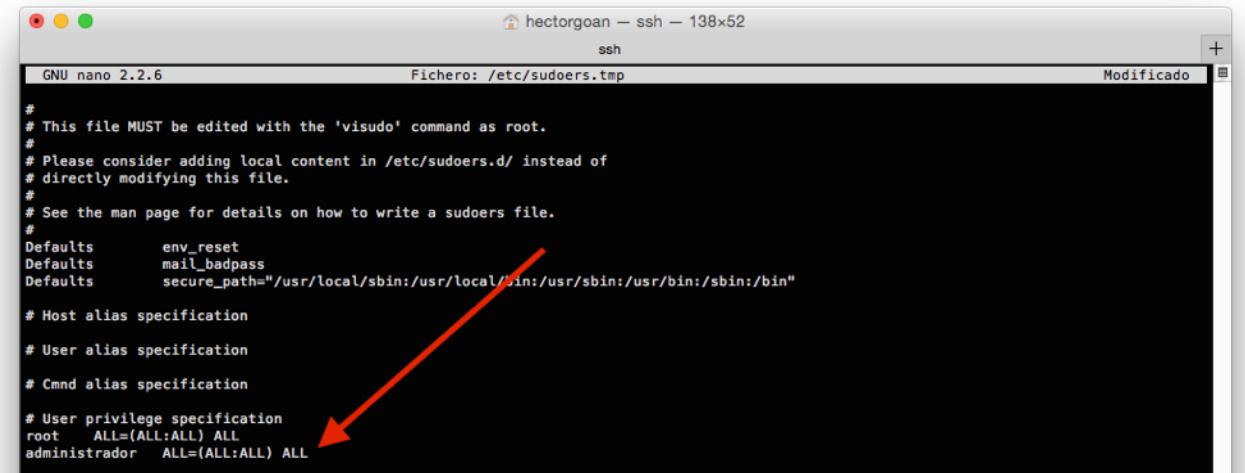
Como se ha podido observar, hemos permitido el acceso ssh únicamente al usuario "administrador", pero ¿qué pasa si queremos realizar tareas que requieran de permisos de superusuario cuando no tengamos acceso físico al servidor?

Tendremos que crear este usuario y darle permisos de superusuario mediante el comando sudo.

```
1 #Logged as root
2 #Creación del usuario administrador
3 adduser administrador
4
5 #Instalación de la herramienta sudo
6 apt-get install sudo
7
8 #Dotación de permisos a administrador
9 visudo
```

En el documento que se abre, añadiremos lo siguiente:

```
administrador    ALL=(ALL:ALL) ALL
```



```
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults      env_reset
Defaults      mail_badpass
Defaults      secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
#
# Host alias specification
#
# User alias specification
#
# Cmnd alias specification
#
# User privilege specification
root        ALL=(ALL:ALL) ALL
administrador    ALL=(ALL:ALL) ALL
```

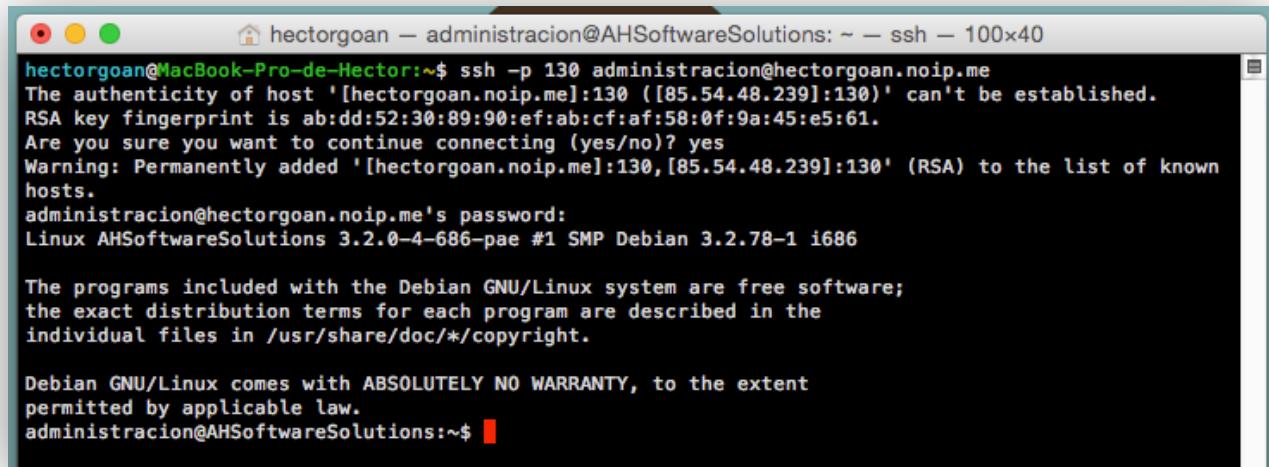
Listo, ya tenemos la cuenta de administrador lista para ser utilizada por ssh, y para realizar tareas de administración.

Nota:

Al instalar ssh en el entorno test, se hace necesario ir a la configuración del router, y abrir el puerto 130 para la ip local de nuestro equipo servidor.

En nuestro router también se ofrece la opción de utilizar un DDNS para que, aunque la ip global de nuestro router cambie podamos acceder al servidor desde cualquier lugar con conexión a internet.

Para este fin utilizamos el servicio No-IP (<http://www.noip.com/>) y por tanto ahora la dirección a nuestro servidor de test es **hectorgoan.noip.me**



```
hectorgoan — administracion@AHSoftwareSolutions: ~ — ssh — 100x40
hectorgoan@MacBook-Pro-de-Hector:~$ ssh -p 130 administracion@hectorgoan.noip.me
The authenticity of host '[hectorgoan.noip.me]:130 ([85.54.48.239]:130)' can't be established.
RSA key fingerprint is ab:dd:52:30:89:90:ef:ab:cf:af:58:0f:9a:45:e5:61.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[hectorgoan.noip.me]:130,[85.54.48.239]:130' (RSA) to the list of known hosts.
administracion@hectorgoan.noip.me's password:
Linux AHSoftwareSolutions 3.2.0-4-686-pae #1 SMP Debian 3.2.78-1 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
administracion@AHSoftwareSolutions:~$
```

Componentes utilizados

A continuación se describen las tecnologías utilizadas en el desarrollo del sistema, todas ellas cumpliendo con los requisitos especificados en el enunciado. Se da por conocido que las tareas de administración se realizan en el lenguaje Perl, aquí explicamos las herramientas de terceros.

Aplicación web

Back-end: Hemos utilizado Django (<https://www.djangoproject.com/>), framework para desarrollo de aplicaciones web que utiliza para la lógica de las mismas el lenguaje de programación Python (<https://www.python.org/>)

Front-end: Django manda renderizar su respuesta a archivos .html enriquecidos llamados templates. Para hacer más atractivos visualmente estos templates se utiliza Bootstrap (<http://getbootstrap.com/>), un framework para desarrollar interfaces web de carácter responsive de manera rápida. A parte de utilizar Bootstrap en los templates de Django también se utiliza esta tecnología en los .cgi que muestran alguna respuesta.

Servidor web

Al principio, teníamos en mente utilizar NGINX (<https://www.nginx.com/>), pero debido a problemas de incompatibilidad (sobretodo con cgi), decidimos utilizar el más veterano apache2 (<https://httpd.apache.org/>)

Gestor de bases de datos

Como servidor de bases de datos hemos decidido utilizar una solución altamente probada, MySQL (<https://www.mysql.com/>), que además cumple con los requisitos del enunciado. MySQL nos aporta estabilidad al ser tan probada y, tras realizar varias búsquedas en internet se mostró totalmente compatible con las tecnologías utilizadas en nuestro servidor.

Mail

Back-end: Postfix (<http://www.postfix.org/>) + Dovecot (<http://www.dovecot.org/>)

Utilizadas estas tecnologías al ser las recomendadas por el profesor Gabriel en la parte práctica de la asignatura.

Front-end: Queríamos dotar con un sitio web de correo a nuestro sistema. Entre las diversas alternativas se decide utilizar Roundcube (<https://roundcube.net/>) por el mismo motivo que Postfix y Dovecot.

Blog

Entre la gran diversidad de servicios CMS disponibles para instalar en servidores propios, nos decantamos por SimpleCMS (<http://get-simple.info/>) por la sencillez de instalación, la ligereza y la ausencia de dependencia de base de datos.

Ftp

Back-end: Como servidor ftp hemos utilizado el que las búsquedas por internet nos devolvían como más estable y seguro: vsftp (<https://security.appspot.com/vsftpd.html>)

Front-end: Además de proporcionar un servidor ftp para al que los usuarios pueden conectarse con clientes externos, veíamos interesante añadir un front-end web para que los alumnos puedan hacer la administración de sus archivos personales desde cualquier lugar, y sin necesidad de tener los conocimientos técnicos necesarios, ya que nuestra solución se muestra como similar a dropbox o google drive (salvando las distancias, claro). Se trata de MonstaFTP (<http://www.monstaftp.com/>), mostrado como un administrador de archivos escrito en PHP.

Portal educativo

Se utiliza el portal educativo por excelencia: Moodle (<https://moodle.org/?lang=es>).

Para la gestión de usuarios en moodle, se utiliza una herramienta externa ante la ausencia de comandos de administración en el paquete de instalación de Moodle.

Esta herramienta es moosh (<http://moosh-online.com/>), descrita como una utilidad de línea de comandos que te permite realizar las tareas más comunes con Moodle.

Muestra de estado de archivos + Portal para los administradores

Para este fin se utiliza eZ Server Monitor (<http://ezservermonitor.com/>). Nos ha llamado la atención la facilidad de customización de este servicio escrito en PHP, su ligereza y la ausencia total de dependencias del mismo.

Como dentro de Django no se pueden tener aplicaciones PHP por incompatibilidades de seguridad, por tanto, como medida de seguridad se ha encerrado el monitor con un inicio de sesión único muy sencillo realizado en php. No se utiliza ni sha5 ni crypt() para securizar este sitio. Queda claro que para un entorno real de producción habría que mejorar la seguridad del sitio.

Instalación de MySQL

Es un proceso sencillo:

```
1 #Logged as administrador
2 sudo apt-get install mysql-server mysql-client
3 #Se nos pedirá passwd para el usuario root de MySQL
```

Podemos compromar la instalación:

```
1 mysql -u root -p #insert passwd
```

Dentro de MySQL

```
1 create database prueba;
2 show databases;
```

```
hectorgoan — administracion@AHSoftwareSolutions: ~ — ssh — 105x42
administracion@AHSoftwareSolutions: ~
administracion@AHSoftwareSolutions: ~ +
```

```
administracion@AHSoftwareSolutions:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 43
Server version: 5.5.47-0+deb7u1 (Debian)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database prueba
      -> ;
Query OK, 1 row affected (0.01 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| prueba |
+-----+
4 rows in set (0.00 sec)

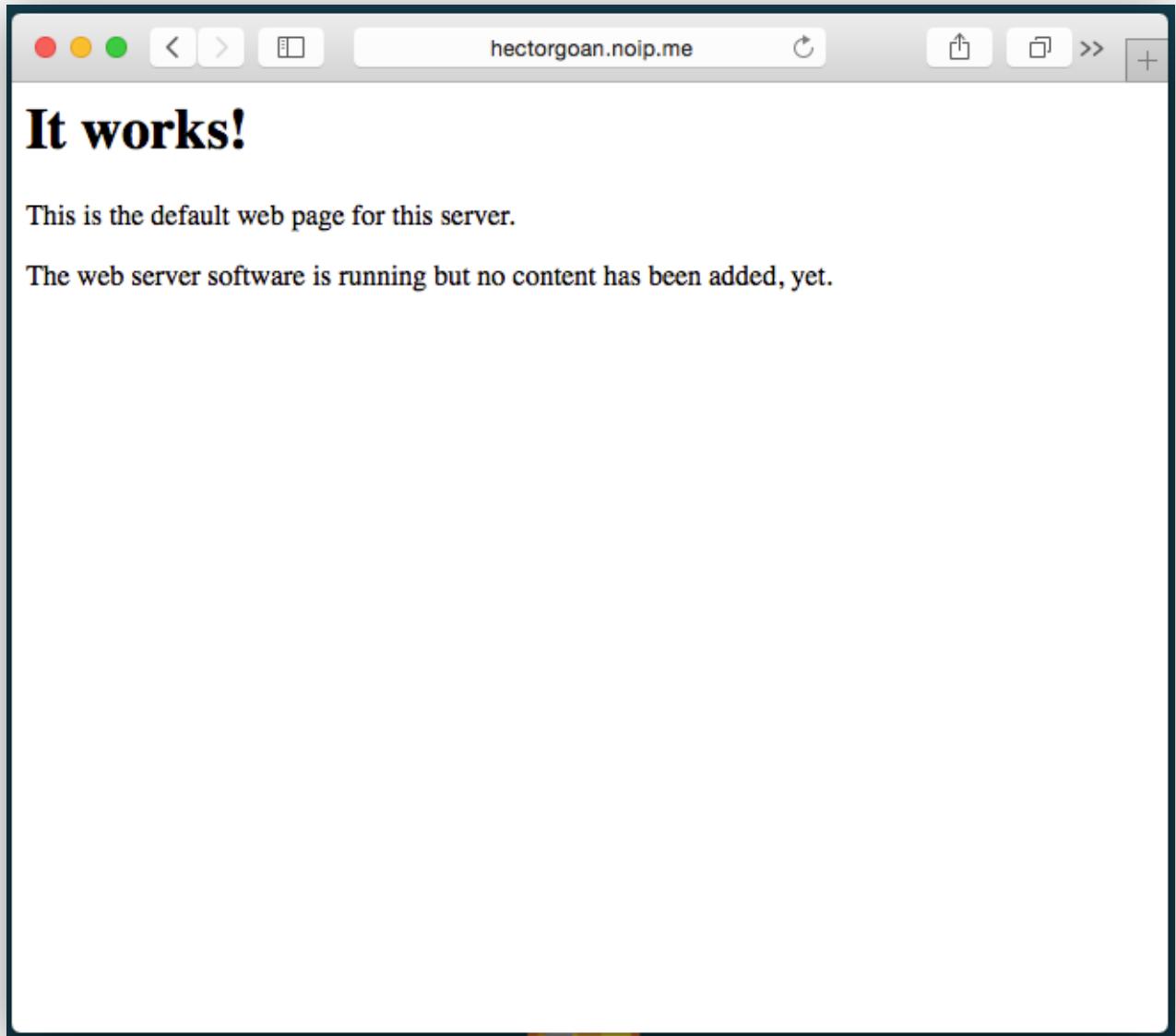
mysql>
```

Instalación de Apache2

Al igual que la instalación de MySQL, también es muy sencillo

```
1 #Logged as administrador  
2 sudo apt-get install apache2
```

Una vez que la instalación ha terminado, visitamos <http://127.0.0.1> (en caso de no tener dominio configurado) y podremos ver el mensaje de bienvenida de Apache.



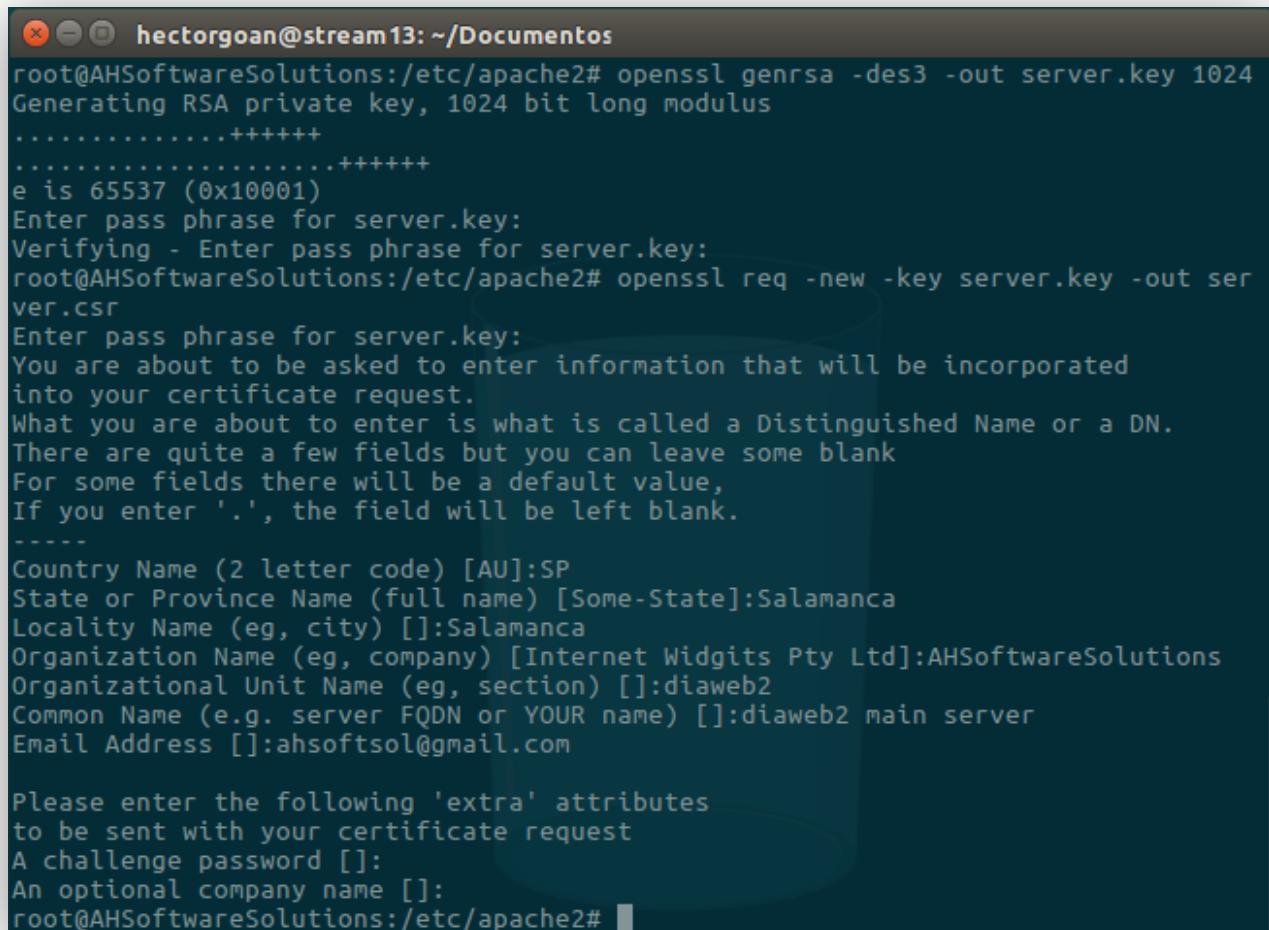
Configuración de Apache2 - Sirviendo en https

En caso de contar con un dominio, podríamos generar un certificado confiable por los navegadores utilizando el servicio letsencrypt (<https://letsencrypt.org/>).

Al ser nuestro entorno de producción un equipo el laboratorio de la facultad, se nos impide enlazar la máquina a un dominio, por ello vamos a tener que utilizar certificados autofirmados para la activación de https.

Para ello:

```
1 #Logged as root
2 #Activación del módulo por defecto
3 a2enmod ssl
4 service apache2 restart
5 #Creación de las llaves autofirmadas
6 cd /etc/apache2
7 openssl genrsa -des3 -out server.key 1024
8 #Creación del certificado con las llaves
9 openssl req -new -key server.key -out server.csr #end
```



The screenshot shows a terminal window with the following session:

```
hectorgoan@stream13: ~/Documentos
root@AHSOFTWARESOLUTIONS:/etc/apache2# openssl genrsa -des3 -out server.key 1024
Generating RSA private key, 1024 bit long modulus
.....+++++
.....+++++
e is 65537 (0x10001)
Enter pass phrase for server.key:
Verifying - Enter pass phrase for server.key:
root@AHSOFTWARESOLUTIONS:/etc/apache2# openssl req -new -key server.key -out server.csr
Enter pass phrase for server.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:SP
State or Province Name (full name) [Some-State]:Salamanca
Locality Name (eg, city) []:Salamanca
Organization Name (eg, company) [Internet Widgits Pty Ltd]:AHSOFTWARESOLUTIONS
Organizational Unit Name (eg, section) []:diaweb2
Common Name (e.g. server FQDN or YOUR name) []:diaweb2 main server
Email Address []:ahsoftsol@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []
root@AHSOFTWARESOLUTIONS:/etc/apache2#
```

```
10 openssl x509 -req -days 365 -in server.csr -signkey  
server.key -out server.crt  
11 cp server.crt /etc/ssl/certs/  
12 cp server.key /etc/ssl/private/
```

Ahora tenemos que aplicar la configuración en apache

```
nano /etc/apache2/sites-available/default-ssl
```

Buscamos la sección del documento en la que se configura ssl (actualmente contendrá las líneas "SSLCertificateFile ..." y "SSLCertificateKeyFile ...". Borramos esas dos líneas y en su lugar añadimos las siguientes:

```
SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire  
SSLCertificateFile /etc/ssl/certs/server.crt  
SSLCertificateKeyFile /etc/ssl/private/server.key #end
```

Una vez hecho esto, guardamos el archivo, salimos del editor y ejecutamos lo siguiente:

```
1 a2ensite default-ssl  
2 service apache2 restart
```

Ya podremos acceder a <https://127.0.0.1> (o la ip local o dominio del servidor) y veremos que funciona. El único problema, es que <http://127.0.0.1> también, y queremos que todo vaya por https. Por tanto, tenemos que decirle a apache que fuerze el uso de https.

```
nano /etc/apache2/sites-available/default
```

Y en este documento añadimos arriba del todo:

```
Redirect permanent / https://hectorgoan.noip.me/
```

De esta manera, todas las peticiones http a apache serán redireccionadas a https.

Configuración de Apache 2 - Permitiendo ejecución de .cgi (o .pl)

Los scripts de administración del sistema están escritos en perl, por tanto, para poder ser llamados desde la aplicación web en Django sin lanzar un thread (conllevando esto llamadas al sistema por parte de un visitante web), tenemos que buscar una manera alternativa de ejecutar nuestros scripts .pl

La primera opción que surge (y al hablar el enunciado de altas no inmediatas de alumnos) era la de servirnos de una tarea CRON que se dedicara a revisar nuevas peticiones de ejecución de algún script (las maneras son secundarias)

La segunda opción (y considerada más fina) tras hablar con el profesor Gabriel y tras decirnos éste que el alta inmediata de usuarios si que era permitida, es la de lanzar peticiones de ejecución de los .pl a apache mediante el método **post** del módulo **requests** de Python.

La ventaja de esto es que, al tratarse del método **post** y por **https**, estamos ante una manera bastante segura de hacer nuestras peticiones.

De esta manera, nuestro código Python se ocupará de toda la parte de la lógica (comprobaciones de seguridad del contenido incluidas) y del lanzamiento de la petición post (totalmente opaca al usuario) que realizará la tarea de administración en el sistema, mientras que Django se encarga de la seguridad en cuanto a sus métodos internos (protecciones contra inyecciones de código, ofuscación total del código back-end, gestión de las sesiones, etc) y del renderizado de la respuesta mediante alguna template.

Mas adelante comentaremos como planteamos soluciones de seguridad ante el lanzamiento de peticiones cgi por agentes externos, ya que como se ha comentado, nuestros .pl son la manera mediante la que se realizará la administración del sistema.

Dicho esto, vamos a ver como configurar la ejecución de scripts perl en apache:

Para poder llevar a cabo esto, hay que tener activado el mod_alias (a2enmod alias), cosa que ya viene de por si configurada con apache.

Lo siguiente a realizar es la configuración del directorio donde se alojarán los .cgi o .pl

```
nano /etc/apache2/sites-available/default
```

En este archivo, escribiremos lo siguiente dentro del host (aunque suele venir ya configurado)

```
1      ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
2      <Directory "/usr/lib/cgi-bin">
3          AllowOverride None
4          Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
5          Order allow,deny
6          Allow from all
7      </Directory>
```

Ahora, si situamos un .pl o .cgi en /usr/lib/cgi-bin/ y hacemos una consulta, veremos que funciona correctamente.

Para que los .pl o .cgi que se sitúen en /usr/lib/cgi-bin/ sean ejecutados, deben de contar con permisos de ejecución. En nuestro caso, los dotamos de permisos 755

En los .pl de administración utilizamos módulos Perl obtenidos del repositorio recomendado en clase: CPAN (<http://search.cpan.org/>)

Para facilitar la gestión de este tipo de módulos instalamos cpanminus (<http://search.cpan.org/~miyagawa/App-cpanminus-1.7042/lib/App/cpanminus.pm>)

Instalamos también un par de paquetes de utilidad en el desarrollo de scripts perl que van a ser ejecutados bajo ssl.

Instalaremos los paquetes así:

```
1      sudo apt-get install cpanminus build-essential
2      sudo apt-get install libnet-ssleay-perl
3      sudo apt-get install libcrypt-ssleay-perl
```

Nota: Estuvimos a punto de cambiar la configuración del directorio cgi-bin a otro nombre o “punto de montaje”, para ofuscar así su localización, pero al ser esta una práctica educativa, nos resultaba quizá excesivo. No obstante nos parecía digno de mención.

Configuración de Apache2 - Activando las páginas de usuario

Se trata de un proceso extremadamente sencillo

Ejecutaremos los siguientes comandos para permitir que los usuarios de nuestro sistema Linux tengan su propia página al añadir los contenidos de la misma en el directorio `public_html` en su `home`

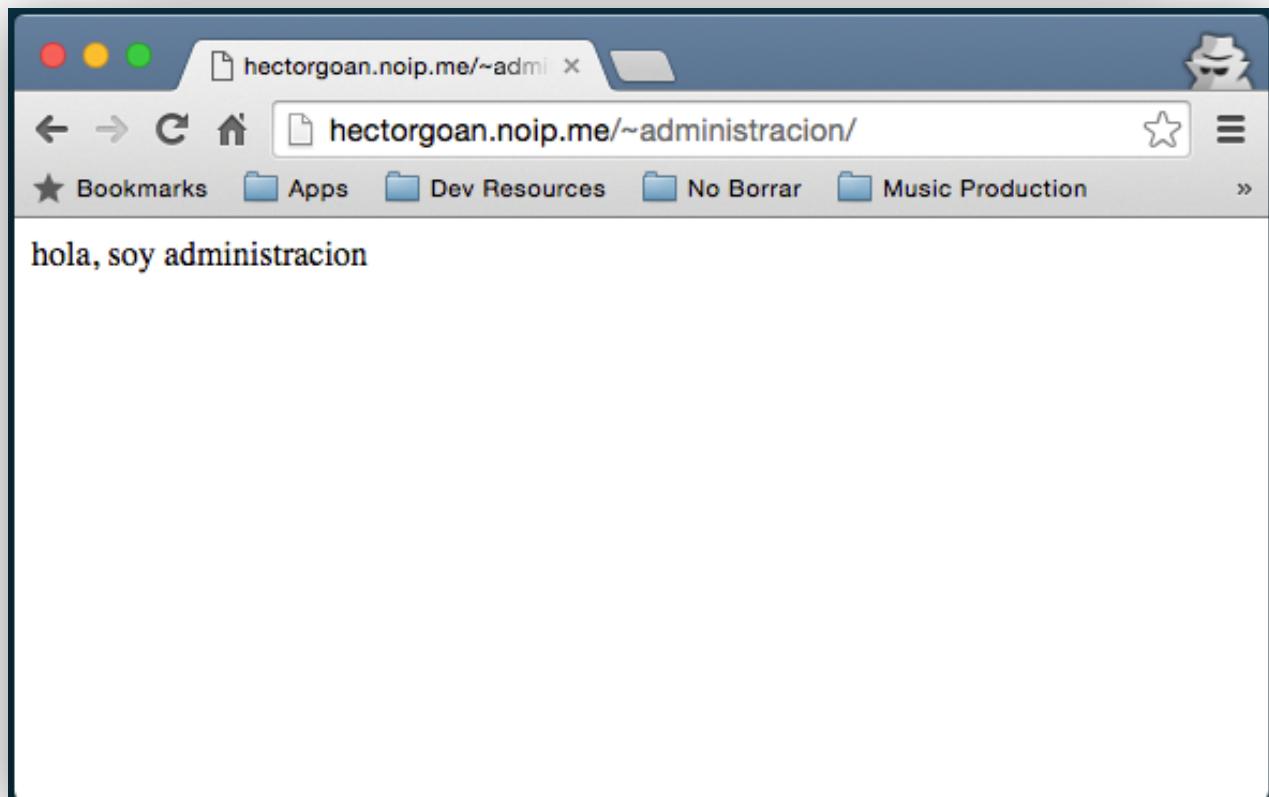
```
1 #Logged as administrador
2 su -l
3 #Now we are root
4 a2enmod userdir
5 service apache2 restart
```

Ya tenemos instalada la funcionalidad, si ahora creamos un `/public_html/index.html` en el directorio `home` de un usuario, cuando visitemos `urlserver.com/~nombreusuario` se nos mostrará la web recién creada



A terminal window titled "hectorgoan" showing the following commands:

```
hectorgoan@...:~$ su -l
root@...:~$ a2enmod userdir
root@...:~$ service apache2 restart
```



Instalación de la funcionalidad mail requerida

Como se comenta en el enunciado, la funcionalidad mail interna del sistema ha de ser 100% funcional. Además, como hemos indicado en la sección de componentes, instalaremos el webmail Roundcube.

Hemos seguido la siguiente guía para llegar a una configuración final que funciona:

<http://nolabnopathy.com/en/setup-mail-server-postfix-dovecot-roundcube-centos/>

Instalación de paquetes necesarios:

```
1 #Logged as administrador
2 su -l
3 #Now we are root
4 apt-get install postfix
5 apt-get install mailutils
```

Lo primero que tenemos que hacer es **configurar postfix**. Abrimos el archivo de configuración:

```
nano /etc/postfix/main.cf
```

En este archivo, lo principal a cambiar es el directorio de mailbox a
home_mailbox=Maildir/

Nota: En el enunciado se pedía la limitación del buzón a 3Mib, pero al introducir este límite, el sistema de correo dejaba de funcionar, así que en vez de tener:

```
mailbox_size_limit = 3145728
```

Hemos tenido que mantener

```
mailbox_size_limit = 0
```

A continuación se muestra como queda la configuración de postfix.

```

hectorgoan — administrador@AHSoftwareSolutions: ~ — ssh — 118x49
administrador@AHSoftwareSolutions: ~
GNU nano 2.2.6 Fichero: /etc/postfix/main.cf

# See /usr/share/postfix/main.cf.dist for a commented, more complete version

# Debian specific: Specifying a file name will cause the first
# line of that file to be used as the name. The Debian default
# is /etc/mailname.
#myorigin = /etc/mailname

smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
biff = no

# appending .domain is the MUA's job.
append_dot_mydomain = no

# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h

readme_directory = no

# TLS parameters
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_use_tls=yes
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

# See /usr/share/doc/postfix/TLS_README.gz in the postfix-doc package for
# information on enabling SSL in the smtp client.

myhostname = AHSoftwareSolutions
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
mydestination = AHSoftwareSolutions, AHSoftwareSolutions, localhost.localdomain, localhost
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
#mailbox_command = procmail -a "$EXTENSION"
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
home_mailbox=Maildir/
inet_protocols = ipv4


```

^G Ver ayuda ^O Guardar ^R Leer Fich ^V Pág Ant ^K CortarTxt ^C Pos actual
 ^X Salir ^J Justificar ^W Buscar ^M Pág Sig ^U PegarTxt ^T Ortografía

```
service postfix restart
```

En este punto podemos instalar dovecot

```
6 apt-get install dovecot-imapd dovecot-pop3d
```

Y procedemos a su configuración (tendremos que editar varios archivos):

```
nano /etc/dovecot/dovecot.conf
```

Incluimos esta línea al final del archivo:

```
protocols = imap pop3 lmtp
```

El siguiente archivo a editar es:

```
nano /etc/dovecot/conf.d/10-mail.conf
```

```
mail_location = maildir:~/Maildir
```

También tenemos que configurar el modo de autenticación:

```
nano /etc/dovecot/conf.d/10-auth.conf
```

```
disable_plaintext_auth = no
```

```
auth_mechanisms = plain login
```

Y por último:

```
nano /etc/dovecot/conf.d/10-master.conf
```

```
unix_listener auth-userdb {
    #mode = 0666
    user = postfix
    group = postfix
}
```

```
service dovecot restart
```

Ya podemos instalar y configurar roundcube

```
7     apt-get install roundcube
8     nano /etc/roundcube/main.inc.php
```

```
$rcmail_config['default_host'] = 'localhost';
```

Para finalizar, hacemos un enlace simbólico para tener la web en el directorio de apache y...
listo!

```
9     ln -s /usr/share/roundcube/ /var/www/mail
```

Instalación del servidor ftp

Instalaremos primero vsftpd y después monstafpt (web ftp client)

```
1 sudo apt-get install vsftpd  
2 sudo nano /etc/vsftpd.conf
```

Y en este archivo, tenemos que añadir/descomentar las siguientes líneas:

```
1 listen=YES  
2 anonymous_enable=NO  
3 local_enable=YES  
4 write_enable=YES  
5 local_umask=022  
6 dirmessage_enable=YES  
7 use_localtime=YES  
8 connect_from_port_20=YES  
9 secure_chroot_dir=/var/run/vsftpd/empty  
10 pam_service_name=vsftpd  
11 rsa_cert_file=/etc/ssl/private/vsftpd.pem  
12 syslog_enable=NO #Logs desde aqui al final  
13 log_ftp_protocol=NO  
14 xferlog_enable=YES  
15 vsftpd_log_file=/var/log/ftpLogs.log  
16 xferlog_std_format=YES  
17 xferlog_file=/var/log/ftpTransfers.log  
18 dual_log_enable=YES
```

Añadimos soporte https

```
1 mkdir /etc/vsftpd  
2 cd /etc/vsftpd  
3 openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout /  
etc/vsftpd/vsftpd.pem -out /etc/vsftpd/vsftpd.pem  
4 nano /etc/vsftpd.conf
```

Finalmente, reiniciamos el servicio

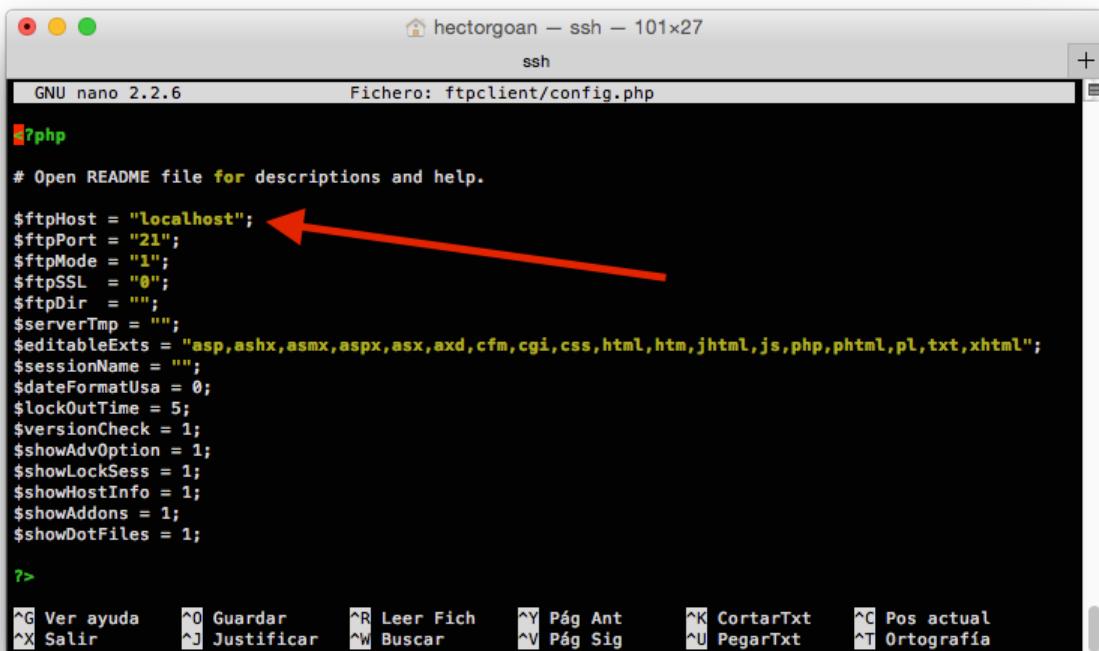
```
5 service vsftpd restart
```

Instalación web ftp client (nube personal + apuntes)

Se trata de descargar un .zip y descomprimirlo en la carpeta de nuestra web. En nuestro caso tendremos 2 configuraciones del cliente ftp. La primera será en la que cada usuario, al introducir su login, accederá a su carpeta home y permanecerá ahí enjaulado (funcionalidad nube personal). La segunda accederá al directorio apuntes, creado para que los profesores puedan añadir y eliminar contenido y los alumnos puedan descargarlo (funcionalidad apuntes).

```
1 #Logged as administrador
2 su -l
3 #Now we are root
4 cd /var/www
5 wget https://www.monstacd.com/downloads/
monsta_ftp_v1.8.5_install.zip
6 unzip monsta_ftp_v1.8.5_install.zip
7 mkdir ftpclient
8 mv mftp/* ftpclient/
9 mkdir apuntes
10 mv mftp/* apuntes/
11 rm -r mftp/
12 service apache2 restart
```

Funcionalidad nube personal (nano /var/www/ftpclient/config.php)



```
hectorgoan - ssh - 101x27
ssh
GNU nano 2.2.6      Fichero: ftpclient/config.php
?php
# Open README file for descriptions and help.

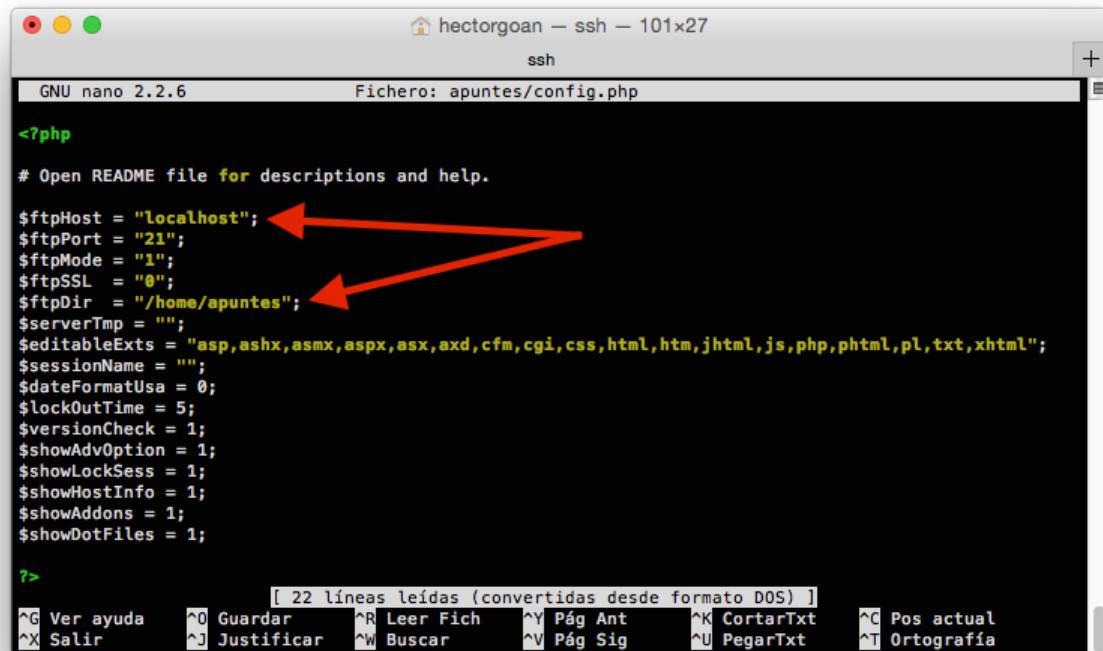
$ftpHost = "localhost"; ←
$ftpPort = "21";
$ftpMode = "1";
$ftpSSL = "0";
$ftpDir = "";
$serverTmp = "";
$editableExt = "asp,ashx,asmx,aspx,asx,axd,cfm,cgi,css,html,htm,jhtml,js,php,phtml,pl,txt,xhtml";
$sessionName = "";
$dateFormatUsa = 0;
$lockOutTime = 5;
$versionCheck = 1;
$showAdvOption = 1;
$showLockSess = 1;
$showHostInfo = 1;
$showAddons = 1;
$showDotFiles = 1;

?>
```

^G Ver ayuda ^O Guardar ^R Leer Fich ^Y Pág Ant ^K CortarTxt ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^V Pág Sig ^U PegarTxt ^T Ortografía

Funcionalidad apuntes (nano /var/www/apuntes/config.php)

```
1  sudo groupadd -g 1001 profesores
2  sudo groupadd -g 1002 alumnos
3
4  sudo mkdir /home/apuntes
5  sudo chgrp -R profesores /home/apuntes
6  sudo chmod 2775 /home/apuntes
```



The screenshot shows a terminal window titled "hectorgoan — ssh — 101x27" running "GNU nano 2.2.6". The file "apuntes/config.php" is open. The code contains configuration settings for an FTP server. Two specific lines are highlighted with red arrows pointing to them:

```
<?php
# Open README file for descriptions and help.

$ftpHost = "localhost";
$ftpPort = "21";
$ftpMode = "1";
$ftpSSL = "0";
$ftpDir = "/home/apuntes";
$serverTmp = "";
$editableExts = "asp,ashx,asmx,aspx,asx,axd,cfm,cgi,css,html,htm,jhtml,js,php,phtml,pl,txt,xhtml";
$sessionName = "";
$dateFormatUsa = 0;
$lockOutTime = 5;
$versionCheck = 1;
$showAdvOption = 1;
$showLockSess = 1;
$showHostInfo = 1;
$showAddons = 1;
$showDotFiles = 1;

?>
```

[22 líneas leídas (convertidas desde formato DOS)]

Terminal status bar:

- ^G Ver ayuda
- ^O Guardar
- ^R Leer Fich
- ^Y Pág Ant
- ^K CortarTxt
- ^C Pos actual
- ^X Salir
- ^J Justificar
- ^W Buscar
- ^V Pág Sig
- ^U PegarTxt
- ^I Ortografía

Instalación de la plataforma de estudios Moodle

Instalación de paquetes necesarios

```
1 #Logged as root
2 apt-get install graphviz aspell php5-pspell php5-curl php5-gd
  php5-intl php5-mysql php5-xmlrpc php5-ldap clamav
3 service apache2 restart
4 apt-get install git-core
```

Instalación de Moodle

```
1 cd /opt
2 sudo git clone git://git.moodle.org/moodle.git
3 cd moodle
4 git branch --track MOODLE_30_STABLE origin/MOODLE_30_STABLE
5 git checkout MOODLE_30_STABLE
6
7 mkdir /var/www/moodle
8 cp -R /opt/moodle/* /var/www/moodle/
9 mkdir /var/moodledata
10 chown -R www-data /var/moodledata
11 chmod -R 777 /var/moodledata
12 chmod -R 0755 /var/www/moodle
```

Preparación de MySQL

```
13 nano /etc/mysql/my.cnf
```

En este archivo:

```
#En basic settings
default_storage_engine = innodb
innodb_file_per_table = 1
innodb_file_format = Barracuda
```

Y después:

```
14 service mysql restart
```

A continuación crearemos la base de datos para Moodle

```
15 mysql -u root -p
```

Introducimos nuestra contraseña de MySQL y a continuación:

```
1 CREATE DATABASE moodle DEFAULT CHARACTER SET utf8 COLLATE
  utf8_unicode_ci;
2 CREATE USER 'administrador'@'localhost' IDENTIFIED BY
  'USALdebian';
3 GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, CREATE TEMPORARY
  TABLES, DROP, INDEX, ALTER ON moodle.* TO administrador@localhost
  IDENTIFIED BY 'USALdebian';
4 quit
```

Preparamos el entorno para realizar la instalación de Moodle

```
1 chmod -R 777 /var/www/moodle
2 service apache2 restart
```

Ahora visitamos la web para realizar el proceso de instalación siguiendo estas instrucciones

```
1 Cambie la ruta para moodledata
2 /var/moodledata
3
4 Tipo de BasedeDatos
5 Elija: mysqli
6
7 Configuraciones de Basededatos
8 Host server: localhost
9
10 BasedeDatos (Database): moodle
11
12 Usuario (User): moodledude (el usuario que Usted creó al
  configurar la BasedeDatos)
13
14 Contraseña (Password): passwordformoodledude (la contraseña para
  el usuario que Usted creó)
15
16 Tables Prefix: mdl_
```

Para terminar, restauramos permisos

```
1 chmod -R 0755 /var/www/moodle
2 service apache2 restart
```

Ya tenemos Moodle instalado!

Creación del proyecto Django y conexión con MySQL

Aunque Django viene con un servidor integrado, este es con propósito de desarrollo y test, para algo en producción se requiere de un servidor más potente. Como hemos utilizado apache2 para el resto de utilidades del sistema, tiene sentido utilizar este para servir nuestra aplicación Django mediante la interfaz wsgi.

Primero instalaremos los paquetes necesarios

```
1 sudo apt-get install python-pip libapache2-mod-wsgi  
#Instalación de paquetes necesarios  
2 sudo pip install virtualenv
```

Ahora es necesario crear un entorno virtual Python que alojará nuestro proyecto Django. Esto se hace normalmente con las aplicaciones Python ya que aporta aislamiento de la app que interactúa con el exterior respecto al resto del sistema, reduciendo así las posibilidades de que la integridad del servidor se vea comprometida.

Nuestro primer impulso fue el de crear el entorno virtual en el directorio /var/www/ ya que el resto de nuestros servicios web se localizan allí, pero bastó un rato de búsquedas de información en internet para darnos cuenta de que esto no era buena idea ya que, si alguien es capaz de descargar nuestro directorio www podría comprometer nuestra información, ya que en el código hay información sensible (ya no hablar si utilizamos sqlite en Django, que la base de datos se guarda junto al resto de archivos que componen la aplicación)

Por tanto, nuestra app se situará en /var/DjApp

```
1 cd /var  
2 mkdir DjApp  
3 cd DjApp  
4 mkdir myapp  
5 cd myapp
```

A continuación creamos el entorno virtual

```
6 virtualenv env
```

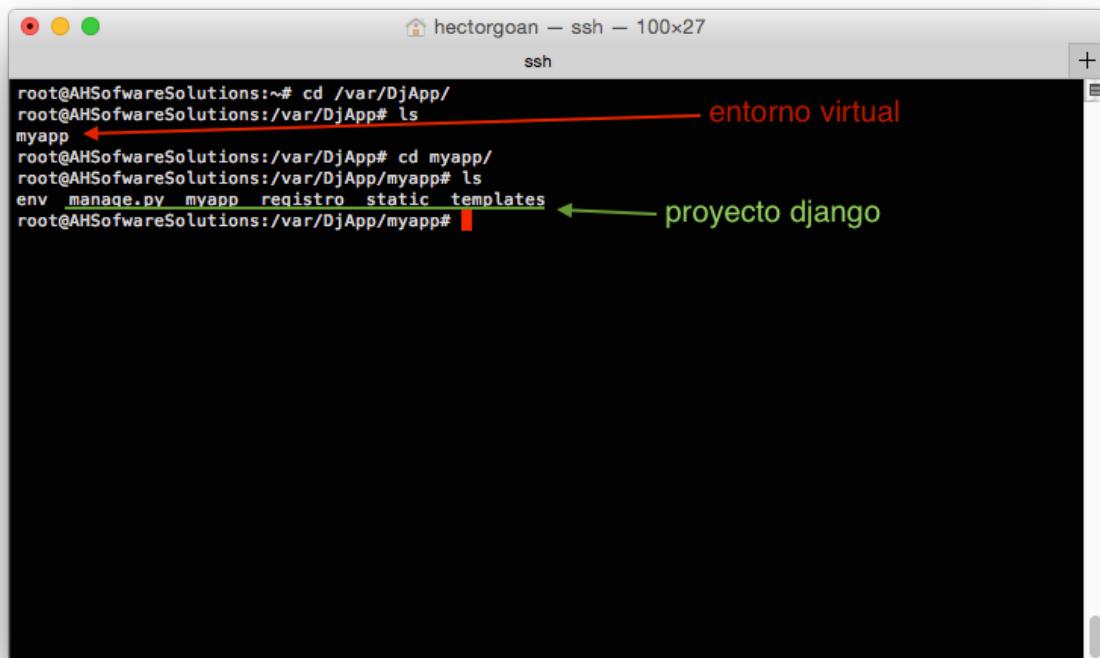
Instalamos Django en nuestro entorno virtual

```
7 env/bin/pip install django
```

Ahora creamos el proyecto Django

```
8 env/bin/django-admin startproject myapp .
```

Ahora mismo, nuestro entorno virtual "myapp" tiene la siguiente apariencia



```
root@AHSofwareSolutions:~# cd /var/DjApp/
root@AHSofwareSolutions:/var/DjApp# ls
myapp
root@AHSofwareSolutions:/var/DjApp# cd myapp/
root@AHSofwareSolutions:/var/DjApp/myapp# ls
env manage.py myapp registro static templates
root@AHSofwareSolutions:/var/DjApp/myapp#
```

Procedemos con la configuración de Django para trabajar con MySQL.

Instalamos un par de dependencias junto con el conector de MySQL de Python.

```
9 sudo apt-get install python-dev
10 sudo apt-get install libmysqlclient-dev
11
12 #conector:
13 env/bin/pip install MySQL-python
```

Ahora tenemos que crear la base de datos para el proyecto

```
14 mysql -u root -p  
1 CREATE DATABASE django;  
2 exit
```

En esta captura se pueden visualizar las operaciones realizadas para crear la base de datos.

```
hectorgoan — administracion@AHSSoftwareSolutions: ~/myproject/mydjangoproject/mydjangoproject ...  
administracion@.../mydjangoproject      administracion@...wareSolutions: ~      administracion@...areSolutions: /tmp +  
(myprojectenv) administracion@AHSSoftwareSolutions:~/myproject/mydjangoproject/mydjangoproject$ mysql -u root -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 37  
Server version: 5.5.47-0+deb7u1 (Debian)  
  
Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> show databases  
    -> ;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
+-----+  
3 rows in set (0.02 sec)  
  
mysql> create database django  
    -> ;  
Query OK, 1 row affected (0.00 sec)  
  
mysql> show databases  
    -> ;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| django |  
| mysql |  
| performance_schema |  
+-----+  
4 rows in set (0.00 sec)  
  
mysql>
```

bbdd para proyecto django

Opcionalmente podemos realizar la siguiente operación de seguridad sobre la base de datos

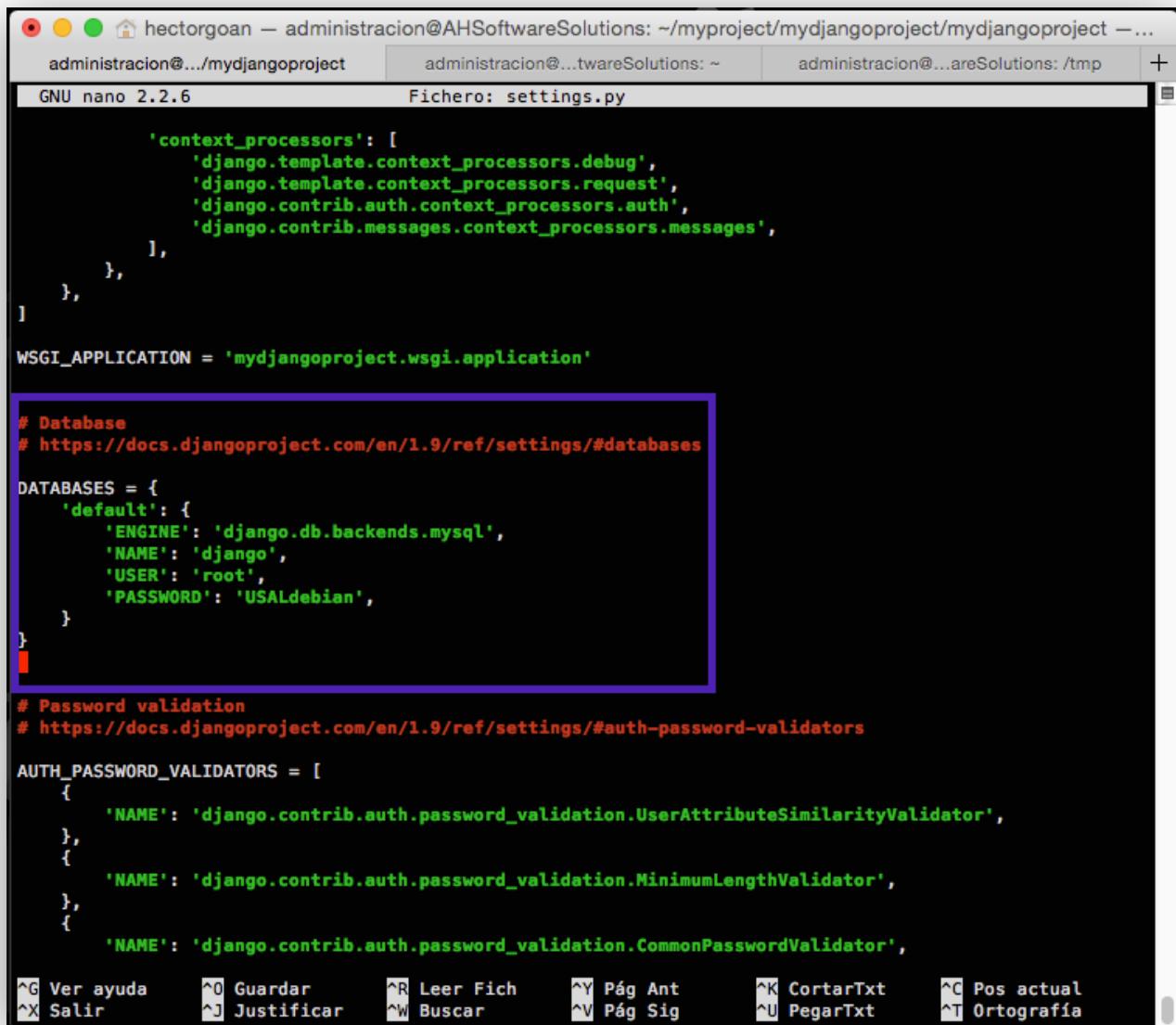
```
15 sudo mysql_secure_installation
```

Procedemos a realizar la conexión en el proyecto de django

```
16 nano /var/DjApp/myapp/settings.py
```

Modificamos la información acerca de las bases de datos, dejándola como a continuación

```
1 DATABASES = {
2     'default': {
3         'ENGINE': 'django.db.backends.mysql',
4         'NAME': 'django',
5         'USER': 'root',
6         'PASSWORD': 'USALdebian',
7     }
8 }
```



```
hectorgoan ~ administracion@AHSwiftwareSolutions: ~/myproject/mydjangoproject/mydjangoproject - ...
administracion@.../mydjangoproject administracion@...wareSolutions: ~ administracion@...areSolutions: /tmp + 
GNU nano 2.2.6 Fichero: settings.py

    'context_processors': [
        'django.template.context_processors.debug',
        'django.template.context_processors.request',
        'django.contrib.auth.context_processors.auth',
        'django.contrib.messages.context_processors.messages',
    ],
},
],
]

WSGI_APPLICATION = 'mydjangoproject.wsgi.application'

# Database
# https://docs.djangoproject.com/en/1.9/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'django',
        'USER': 'root',
        'PASSWORD': 'USALdebian',
    }
}

# Password validation
# https://docs.djangoproject.com/en/1.9/ref/settings/#auth-password validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    }
]
```

Para continuar, tenemos que mudar la base de datos

```
17 sudo env/bin/python manage.py makemigrations  
18 sudo env/bin/python manage.py migrate
```

Si consultamos la base de datos veremos que la información de Django ya está presente en la base que creamos antes para el proyecto

```
hectorgoan — administracion@AHSoftwareSolutions: ~/myproject/mydjangoproject/mydjangoproject — ssh — 114x48  
administracion@AH...ct/mydjangoproject      administracion@AHSoftwareSolutions: ~      administracion@AH...areSolutions: /tmp +  
(myprojectenv) administracion@AHSoftwareSolutions:~/myproject/mydjangoproject/mydjangoproject$ mysql -u root -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 53  
Server version: 5.5.47-0+deb7u1 (Debian)  
  
Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| django |----->                                base de datos para el proyecto django  
| mysql |  
| performance_schema |  
+-----+  
4 rows in set (0.00 sec)  
  
mysql> use django;  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
  
Database changed  
mysql> show tables;  
+-----+  
| Tables_in_django |----->          tablas creadas por django al migrar la base  
+-----+  
| auth_group  
| auth_group_permissions  
| auth_permission  
| auth_user  
| auth_user_groups  
| auth_user_user_permissions  
| django_admin_log  
| django_content_type  
| django_migrations  
| django_session |  
+-----+  
10 rows in set (0.01 sec)  
  
mysql>
```

Configuración de Apache2 - Servir la aplicación Django

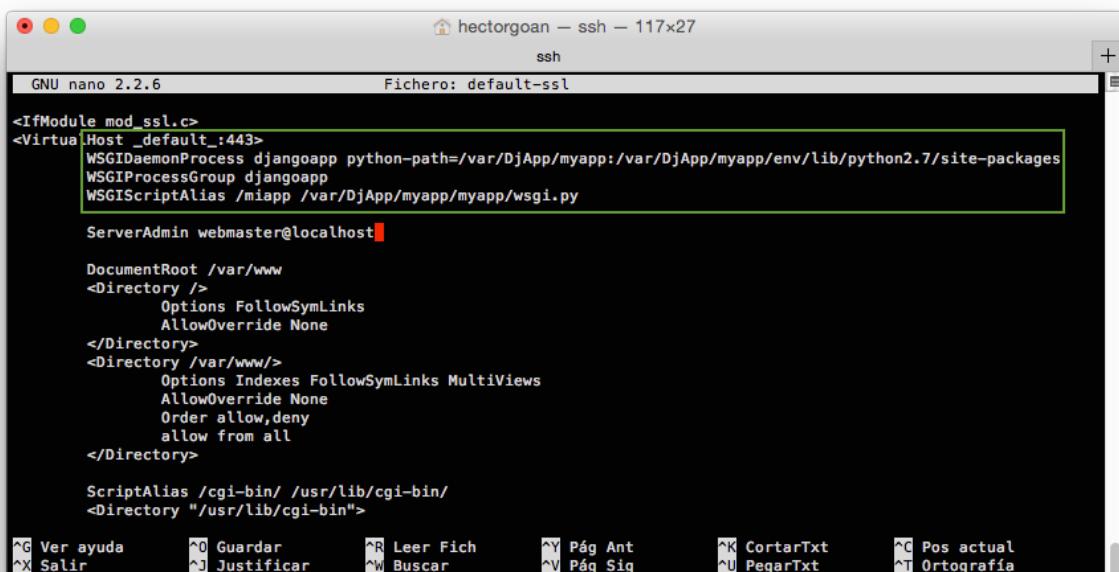
Tenemos que conectar la aplicación Django a Apache mediante la interfaz wsgi.

```
1      sudo nano /etc/apache2/sites-available/default-ssl
```

Añadiremos 3 líneas al archivo que se abre.

```
1      WSGIDaemonProcess djangoapp python-path=/var/DjApp/myapp:/var/DjApp/myapp/
env/lib/python2.7/site-packages
2      WSGIProcessGroup djangoapp
3      WSGIScriptAlias /miapp /var/DjApp/myapp/myapp/wsgi.py
```

En la captura a continuación se puede ver mejor



```
hectorgoan - ssh - 117x27
ssh
GNU nano 2.2.6          Fichero: default-ssl

<IfModule mod_ssl.c>
<VirtualHost _default_:443>
    WSGIDaemonProcess djangoapp python-path=/var/DjApp/myapp:/var/DjApp/myapp/env/lib/python2.7/site-packages
    WSGIProcessGroup djangoapp
    WSGIScriptAlias /miapp /var/DjApp/myapp/myapp/wsgi.py

    ServerAdmin webmaster@localhost

    DocumentRoot /var/www
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/www/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>

    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
    <Directory "/usr/lib/cgi-bin">
```

La explicación de como funciona la interfaz wsgi y de cómo hace Apache para procesar las peticiones para enviárselas a la aplicación Django, se encuentra en el siguiente enlace:

https://www.digitalocean.com/community/tutorials/how-to-run-django-with-mod_wsgi-and-apache-with-a-virtualenv-python-environment-on-a-debian-vps

Nota: Montamos WSGIScriptAlias en /miapp porque si la montamos en / la aplicación Django "pisa" al resto de funcionalidades de nuestra web y las deja inaccesibles. Según está configurado, todo lo que sea **serverdomain.com/...** lo tiene que procesar apache directamente, y todo lo que sea **serverdomain.com/miapp/...** tiene que pasarlo a Django por wsgi.

Instalación de paquetes necesarios

Por el camino hemos ido instalando los paquetes necesarios para todo lo logrado hasta el momento, pero, a partir de aquí empieza el desarrollo del sistema. Para este fin necesitamos instalar varios paquetes, unos necesarios para nuestro desarrollo en Python y otros para nuestro desarrollo en Perl.

Instalación de paquetes de CPAN para desarrollo en Perl

```
1 #Required in /usr/lib/cgi-bin scripts
2 sudo cpanm install Email::Send::SMTP::Gmail
3 sudo cpanm install IPC::System::Simple
4 sudo cpanm install Log::Handler
5 #Required in /usr/lib/cgi scripts
6 sudo cpanm install File::Copy
7 sudo cpanm install DynGig::Util::Setuid
8 sudo cpanm install Linux::usermod
9 sudo cpanm install File::Copy::Recursive;
10 sudo cpanm install File::chmod::Recursive;
11 sudo cpanm install File::Path
```

Instalación de paquetes para desarrollo Python (utilizados en la app Django)

```
1 sudo env/bin/pip install requests
2 sudo env/bin/pip install django-secure
```

Configuración de django-secure (habilitar https en django)

```
1 nano /var/DjApp/myapp/settings.py

INSTALLED_APPS = (
    ...
    'djangosecure',
    ...
)

MIDDLEWARE_CLASSES = (
    'djangosecure.middleware.SecurityMiddleware',
    ...
)
SECURE_SSL_REDIRECT = True
```

Configuración del sistema de logs

Ftp

El sistema de logs se configuró en el archivo /etc/vsftpd.conf a destacar las líneas:

```
vsftpd_log_file=/var/log/ftpLogs.log
xferlog_file=/var/log/ftpTransfers.log
```

Los accesos por ftp se guardarán en ftpLogs.log y las transferencias en ftpTransfers.log

SSH

El sistema lo hace por defecto en /var/log/auth

Podremos ver los loggin attempts en ssh con:

```
1 cat /var/log/auth.log | grep ssh
```

Django App

Se guardarán los logs en /var/log/djangoLog.log mediante la llamada a un .pl que hace los logs.

A mayores, cuando se haga un acceso a la cuenta administrador (o un intento fallido) se enviará un mail al sysadmin para notificar el incidente.

Gestión de usuarios de Moodle con moosh

Se comentó en clase que era de gran importancia “en el mundo real” el tema de la gestión de usuarios en Moodle, ya que es un quebradero de cabeza en los departamentos IT de las Universidades.

Como se comentó en el apartado de componentes, se hace uso de moosh.

Encontramos la información acerca de esta utilidad en internet, en la siguiente url:

<http://www.somerandomthoughts.com/blog/2012/12/05/command-line-moodle-moosh/>

Nos resultó llamativo que las utilidades que se instalan con Moodle no dispusieran de la funcionalidad necesaria para hacer la administración de los usuarios del portal.

Aquí se pueden ver las funciones del administrador por defecto de Moodle:

https://docs.moodle.org/22/en/Administration_via_command_line

Por tanto, lo que hacemos a la hora de administrar un usuario es lo siguiente desde uno de los .pl de administración (a continuación el ejemplo de añadir un usuario)

```
chdir "/var/www/moodle/";
system("moosh -n user-create --city Salamanca --country ES --
password ".$password." --email ".$email." ".$nick);
```

Nos vemos obligados a comentar esto, ya que en el enunciado se indica que el uso de system quedaría prohibido excepto para algún caso excepcional y bien justificado.

Creemos que es el caso.

La instalación de esta utilidad es muy sencilla.

```
1 git clone https://github.com/tmuras/moosh
2 cd moosh/
3 sudo ln -s $PWD/moosh.php /usr/local/bin/moosh
```

Ahora, si accedemos al directorio de Moodle /var/www/moodle podremos ejecutar cualquiera de los comandos que se ofrecen para la administración del portal. Información acerca de los comandos disponibles:

<http://moosh-online.com/commands/>

Diaweb 2.0 - El sistema

Procede la explicación del funcionamiento y estructura del sistema en el entorno test.

Posteriormente se verán las adaptaciones al entorno de producción.

Arquitectura

Se podría considerar el sistema como un conjunto de 4 componentes principales:

Servidores

Conteniendo el servidor web, el servidor de mail y el servidor FTP. El ssh se omite del diagrama al no ser este utilizado por los clientes

Directorio /var/www

Contiene la página web principal del dominio, la de moodle, los dos clientes ftp, el cliente de correo, la página de ayuda, la de estado de los servicios y la de información al administrador.

Django App

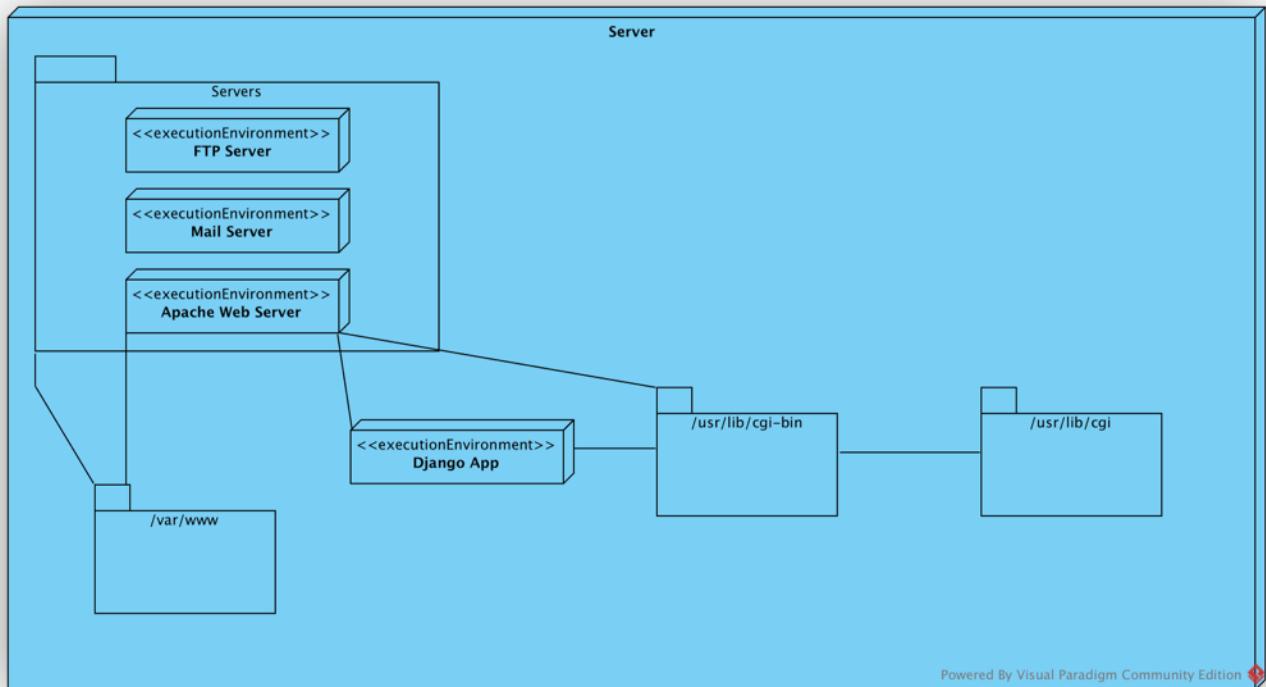
Se encarga de toda la lógica. Dirige la interacción del usuario con la web una vez que inicia sesión y se encarga de llamar a los scripts de administración.

Directorio /usr/lib/cgi-bin

Contiene los scripts de administración que son llamados desde la app de django (una petición al servidor web apache)

Directorio /usr/lib/cgi

Contiene los scripts que realmente realizan cambios en el sistema. La única manera de ejecutarlos desde fuera del sistema es si son llamados por los scripts de /usr/lib/cgi-bin por lo que la petición ha de recibir el "visto bueno" de estos.



De esta manera, un usuario que entre en la página principal y realice el proceso de registro, está haciendo que, al clickar el enlace “Registro” empiece a ejecutarse el código de la app de Django la cual nos mostrará el formulario de registro en una de sus templates. Cuando se introducen los datos del registro, estos son analizados por el código por defecto de Django (prevención de inyecciones en formularios, formato de datos, campos requeridos, etc) para después pasar a nuestro código Python. Aquí, analizamos contenido y, si es válido registramos el usuario en Django, lanzamos la petición al .pl correspondiente en /usr/lib/cgi-bin que hará las gestiones en Linux y mostramos la respuesta al usuario mediante otra template. En el script /usr/lib/cgi-bin se realizan comprobaciones extra, para evitar que alguien que haga la petición a apache intente registrar usuarios (mecanismo que explicaremos a continuación). Si todo es correcto realiza las tareas “non-root-required” como puede ser enviar un correo al usuario y llama al /usr/lib/cgi que es quien finalmente da de alta el usuario en Linux y en Moodle.

¿Por qué esto?

Por partes:

- hash
- non-root en cgi-bin

- Mecanismo bautizado "hash"

Es un mecanismo MUY MUY MUY sencillo de "encriptación punto a punto". Obviamente mejorable, pero básicamente la idea es: cuando Django llame a un .cgi manda una información asociada que solo ese .cgi conoce. Actualmente es un string que se envía en la petición y que obviamente es una medida poco fuerte, pero esto podría mejorarse mediante algún algoritmo hash o algún parámetro guardado en la db.

Funciona partiendo de la idea de que tanto el código .py de django como el .pl del cgi-bin es "imposible" de leer por los malos.

```
aHash = "SupaSecretPasswd"

#Registering user in server + notifying user by e-mail
cgiURL= "https://hectorgoan.noip.me/cgi-bin/registerUser.pl?
x=x&x=x&x=x&x=x&x=x&x=x&x=x&x=x&x=x&x=x"
url = cgiURL
payload = {'nick': aNick, 'email': anEmail, 'password': aPassword, 'nombre': aNombre, 'direccion': aDireccion, 'rol': aRol, 'hash': aHash}

r = requests.post(url, data=payload, verify=False)
```

Aquí podemos observar como junto con la información del usuario que ha solicitado registrarse se envía al cgi un "aHash". Hay que destacar que esta petición se hace de manera opaca al usuario (en ningún momento ve la url que se envía) y por https (encriptado).

Cuando la petición llega al .pl esto es lo que hacemos:

```
my $_hash;
$_hash = param('hash');

if ($_hash ne "SupaSecretPasswd")
{
    die();
}
```

Por tanto, aunque alguien por algún suponer descubriera el número de argumentos del .pl, el nombre del .pl, el directorio del .pl (se comentó la posibilidad de cambiar el estándar cgi-bin a otro directorio) y el nombre de los parámetros, aún tendría que averiguar el contenido de hash.

Queda claro y somos conscientes de que se debería de utilizar otro algoritmo para la generación/verificación del hash.

- Política non-root en cgi-bin

Nunca es buena idea tener código que pueda realizar cambios (o destrozos) en el sistema disponibles al mundo.

Hay ataques que permiten descargar todo el contenido de un directorio abierto a internet sin ningún tipo de restricción (por eso que la app django no está en /var/www). Con las mismas se pueden aprovechar vulnerabilidades para subir contenido al servidor (utilizando algún tipo de formulario de subida de archivos no securizado o que no compruebe el tipo de archivo)

Por tanto:

Dos puntos de vista:

1.

Si alguien consigue subir código a un directorio sobre el que tiene permisos de ejecución root... para que queremos más

2.

Tu cuando abres la puerta del portal al repartidor de correos ... ¿le das abierta la del piso para que te deje las cartas en la mesa?

"Vale, me has demostrado que eres tu con el hash, pero... déjame a mí que ya me encargo yo de añadirete al sistema"

Por tanto, el .pl en cgi-bin se muestra como una puerta del edificio y la petición entrante al .pl como el repartidor de correos. Cuando un extraño llega, le pedimos una identificación ("hash") y, si entra, le podemos permitir dejar el correo en los buzones ("tareas no administrativas") entre ellas notificaciones mail o muestra de mensajes html, pero de aquí en adelante ya me encargo yo de llamar al .pl que realiza los cambios en el sistema.

¿Cómo lo conseguimos?

Tanto los scripts de cgi-bin como los de cgi tienen permisos 755 así que, siendo www-data (el usuario de internet) el que hace la solicitud de cambio en el sistema, ¿cómo puede realizar las tareas?

```

root@AHSofwareSolutions:/usr/lib/cgi# ls -l
total 44
-rwxr-xr-x 1 root root 369 may 22 19:26 changeEmailUserInMoodle.pl
-rwxr-xr-x 1 root root 473 may 22 19:14 changePassUserSYS.pl
-rwxr-xr-x 1 root root 853 may 21 17:41 createBlogInSYS.pl
-rwxr-xr-x 1 root root 573 may 21 18:27 deleteBlogFromSYS.pl
-rwxr-xr-x 1 root root 526 may 22 18:53 deleteUserFromSYS.pl
-rwxr-xr-x 1 root root 369 may 25 18:04 EXTERNchangeEmailUserInMoodle.pl
-rwxr-xr-x 1 root root 399 may 25 19:23 EXTERNchangePassUserInMoodle.pl
-rwxr-xr-x 1 root root 355 may 25 18:09 EXTERNdeleteUserFromMoodle.pl
-rwxr-xr-x 1 root root 520 may 26 23:38 EXTERNregisterUserInMoodle.pl
-rwxr-xr-x 1 root root 633 may 22 01:36 loginlog.pl
-rwxr-xr-x 1 root root 1629 may 23 02:07 registerUserInSYS.pl
root@AHSofwareSolutions:/usr/lib/cgi# cd /usr/lib/cgi-bin/
root@AHSofwareSolutions:/usr/lib/cgi-bin# ls -l
total 80
-rwxr-xr-x 1 root root 726 may 22 19:29 changeEmailUserMoodle.pl
-rwxr-xr-x 1 root root 1398 may 20 01:22 changePassUser.pl
-rwxr-xr-x 1 root root 672 may 21 15:35 createBlog.pl
drwxr-xr-x 2 root root 4096 may 4 10:18 css
-rwxr-xr-x 1 root root 674 may 21 17:26 deleteUser.pl
-rwxr-xr-x 1 root root 1361 may 20 01:28 deleteUser.pl
-rwxr-xr-x 1 root root 732 may 25 18:28 EXTERNchangeEmailUserMoodle.pl
-rwxr-xr-x 1 root root 747 may 25 19:23 EXTERNchangePassUserMoodle.pl
-rwxr-xr-x 1 root root 673 may 25 19:32 EXTERNdeleteUserFromMoodle.pl
-rwxr-xr-x 1 root root 800 may 25 18:38 EXTERNregisterUserInMoodle.pl
drwxr-xr-x 2 root root 4096 may 4 10:18 fonts
drwxr-xr-x 5 root root 4096 may 4 10:18 images
drwxr-xr-x 2 root root 4096 may 4 10:18 js
-rwxr-xr-x 1 root root 745 may 22 01:39 loginlog.pl
-rwxr-xr-x 1 root root 840 may 25 17:45 mailservByeBye.pl
-rwxr-xr-x 1 root root 944 may 25 17:29 mailserv.pl
-rwxr-xr-x 1 root root 853 may 25 17:31 mailservRecover.pl
-rwxr-xr-x 1 root root 1094 may 25 17:55 notification.pl
-rwxr-xr-x 1 root root 1597 may 20 01:15 registerUser.pl
-rwxr-xr-x 1 root root 3656 may 4 10:18 sendemail.pl
root@AHSofwareSolutions:/usr/lib/cgi-bin#

```

Se realizan las tareas administrativas gracias a dos elementos

DynGig::Util::Setuid

visudo

DynGig::Util::Setuid

En los scripts administrativos se utiliza este módulo, y se añade la siguiente línea de código.

```
DynGig::Util::Setuid->sudo( 'root' );
```

¿Y no hay que meter la clave de root?

visudo

Aprovechando sudo, podemos decir que un directorio (o un archivo) pueda ser ejecutado con permisos de administrador por un usuario indicado, sin introducir ni siquiera la contraseña.

Daremos permisos a www-data sobre los archivos **específicos** que queramos.

```

# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
administrador  ALL=(ALL:ALL) ALL
www-data      ALL=(ALL) NOPASSWD:/usr/lib/cgi/registerUserInSYS.pl
administrador  ALL=(ALL) NOPASSWD:/usr/lib/cgi/registerUserInSYS.pl
www-data      ALL=(ALL) NOPASSWD:/usr/lib/cgi/changePassUserSYS.pl
administrador  ALL=(ALL) NOPASSWD:/usr/lib/cgi/changePassUserSYS.pl
www-data      ALL=(ALL) NOPASSWD:/usr/lib/cgi/deleteUserFromSYS.pl
administrador  ALL=(ALL) NOPASSWD:/usr/lib/cgi/deleteUserFromSYS.pl
www-data      ALL=(ALL) NOPASSWD:/usr/lib/cgi/createBlogInSYS.pl
administrador  ALL=(ALL) NOPASSWD:/usr/lib/cgi/createBlogInSYS.pl
www-data      ALL=(ALL) NOPASSWD:/usr/lib/cgi/deleteBlogFromSYS.pl
administrador  ALL=(ALL) NOPASSWD:/usr/lib/cgi/deleteBlogFromSYS.pl
www-data      ALL=(ALL) NOPASSWD:/usr/lib/cgi/loginlog.pl
administrador  ALL=(ALL) NOPASSWD:/usr/lib/cgi/loginlog.pl
www-data      ALL=(ALL) NOPASSWD:/usr/lib/cgi/changeEmailUserInMoodle.pl
administrador  ALL=(ALL) NOPASSWD:/usr/lib/cgi/changeEmailUserInMoodle.pl

www-data      ALL=(ALL) NOPASSWD:/usr/lib/cgi/EXTERNchangeEmailUserInMoodle.pl
administrador  ALL=(ALL) NOPASSWD:/usr/lib/cgi/EXTERNchangeEmailUserInMoodle.pl
www-data      ALL=(ALL) NOPASSWD:/usr/lib/cgi/EXTERNchangePassUserInMoodle.pl
administrador  ALL=(ALL) NOPASSWD:/usr/lib/cgi/EXTERNchangePassUserInMoodle.pl
www-data      ALL=(ALL) NOPASSWD:/usr/lib/cgi/EXTERNdeleteUserFromMoodle.pl
administrador  ALL=(ALL) NOPASSWD:/usr/lib/cgi/EXTERNdeleteUserFromMoodle.pl
www-data      ALL=(ALL) NOPASSWD:/usr/lib/cgi/EXTERNregisterUserInMoodle.pl
administrador  ALL=(ALL) NOPASSWD:/usr/lib/cgi/EXTERNregisterUserInMoodle.pl
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:
#include /etc/sudoers.d

```

^G Ver ayuda ^O Guardar ^R Leer Fich ^Y Pág Ant ^K CortarTxt ^C Pos actual
 ^X Salir ^J Justificar ^W Buscar ^V Pág Sig ^U PegarTxt ^T Ortografía

De esta manera queda cerrada la acción de registrar usuario.

Esto se repite para todas las acciones sobre el sistema que conllevan cambios sobre los usuarios Linux o Moodle.

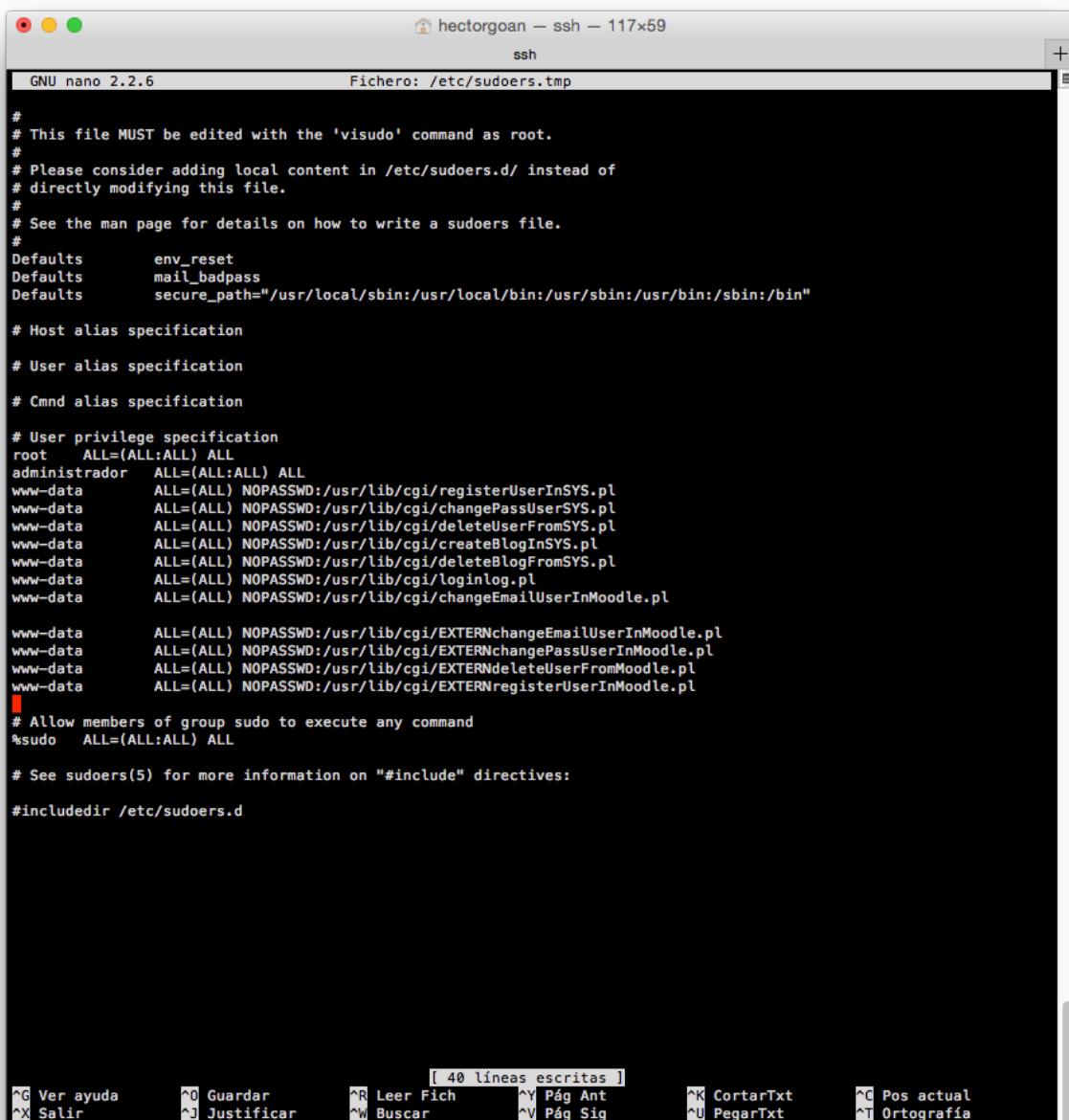
El último paso es eliminar las líneas que son iguales que las de www-data pero con el usuario administrador para terminar de dejar el sistema “cerrado”, al menos desde un punto de teoría.

La única manera de acceder por ssh es con el usuario administrador. Si alguien consiguiera entrar con el y quisiera registrar un usuario al sistema, tendría que primero saber que el usuario con permiso de login ssh es administrador, segundo la contraseña del mismo y tercero la contraseña sudo para poder ejecutar los scripts creados por nosotros, o bien los comandos de creación de usuarios.

Si alguien intenta crear un usuario haciendo la petición al .pl tendrá que primero hacerlo con el usuario www-data, segundo sacar todos los parámetros (número y nombre) y el hash.

Por tanto concluimos que es bastante improbable que si se registra un usuario en el sistema no haya sido de la manera que nosotros queremos que se haga.

:)



The screenshot shows a terminal window titled "hectorgoan — ssh — 117x59" running the "ssh" command. The window title bar says "hectorgoan — ssh — 117x59". The main area of the window displays the contents of the file "/etc/sudoers.tmp" using the "GNU nano 2.2.6" editor. The file contains a series of sudo privilege specifications for various users and groups. Most lines are preceded by a hash (#), indicating they are comments. The file includes sections for Defaults, Host alias specification, User alias specification, Cmnd alias specification, and User privilege specification. It lists numerous commands for users like "www-data" and "administrador" with their respective privileges. At the bottom of the file, there are "# See sudoers(5) for more information on "#include" directives:" and "#includedir /etc/sudoers.d". The status bar at the bottom of the terminal window shows "[40 líneas escritas]" (40 lines written). The bottom right corner of the terminal window has a small icon with a plus sign (+).

```
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults      env_reset
Defaults      mail_badpass
Defaults      secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root          ALL=(ALL:ALL) ALL
administrador  ALL=(ALL:ALL) ALL
www-data      ALL=(ALL) NOPASSWD:/usr/lib/cgi/registerUserInSYS.pl
www-data      ALL=(ALL) NOPASSWD:/usr/lib/cgi/changePassUserSYS.pl
www-data      ALL=(ALL) NOPASSWD:/usr/lib/cgi/deleteUserFromSYS.pl
www-data      ALL=(ALL) NOPASSWD:/usr/lib/cgi/createBlogInSYS.pl
www-data      ALL=(ALL) NOPASSWD:/usr/lib/cgi/deleteBlogFromSYS.pl
www-data      ALL=(ALL) NOPASSWD:/usr/lib/cgi/loginlog.pl
www-data      ALL=(ALL) NOPASSWD:/usr/lib/cgi/changeEmailUserInMoodle.pl
www-data      ALL=(ALL) NOPASSWD:/usr/lib/cgi/EXTERNchangeEmailUserInMoodle.pl
www-data      ALL=(ALL) NOPASSWD:/usr/lib/cgi/EXTERNchangePassUserInMoodle.pl
www-data      ALL=(ALL) NOPASSWD:/usr/lib/cgi/EXTERNdeleteUserFromMoodle.pl
www-data      ALL=(ALL) NOPASSWD:/usr/lib/cgi/EXTERNregisterUserInMoodle.pl

# Allow members of group sudo to execute any command
%sudo        ALL=(ALL:ALL) ALL

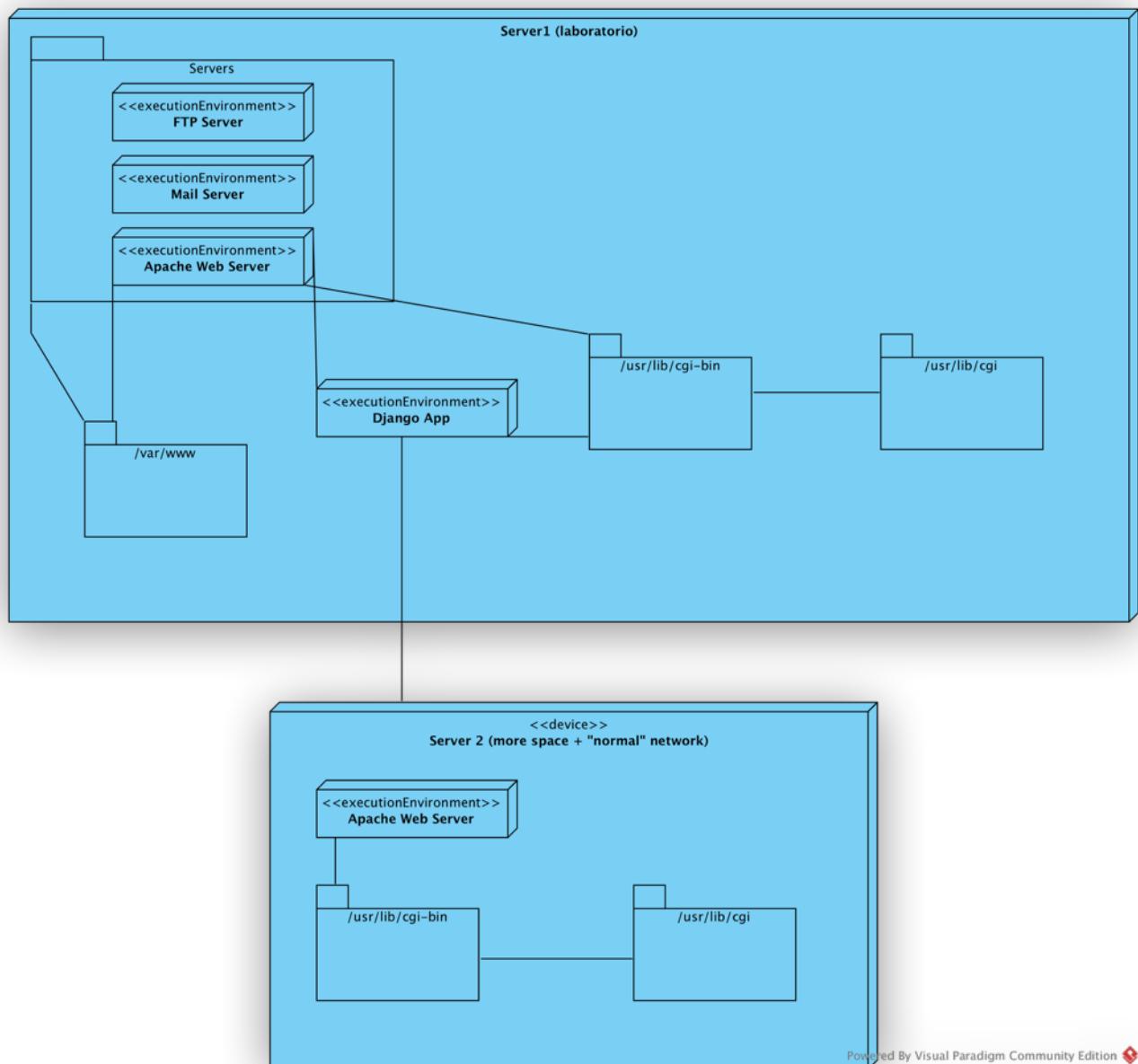
# See sudoers(5) for more information on "#include" directives:
#include /etc/sudoers.d
```

Nota: Ejemplo de todo lo comentado de seguridad:

<https://twitter.com/GammaGroupPR/status/733347252605947906>

Despliegue en el entorno de producción

La arquitectura del sistema varía cuando el entorno en el que trabajamos es en el de producción (laboratorio de la facultad)



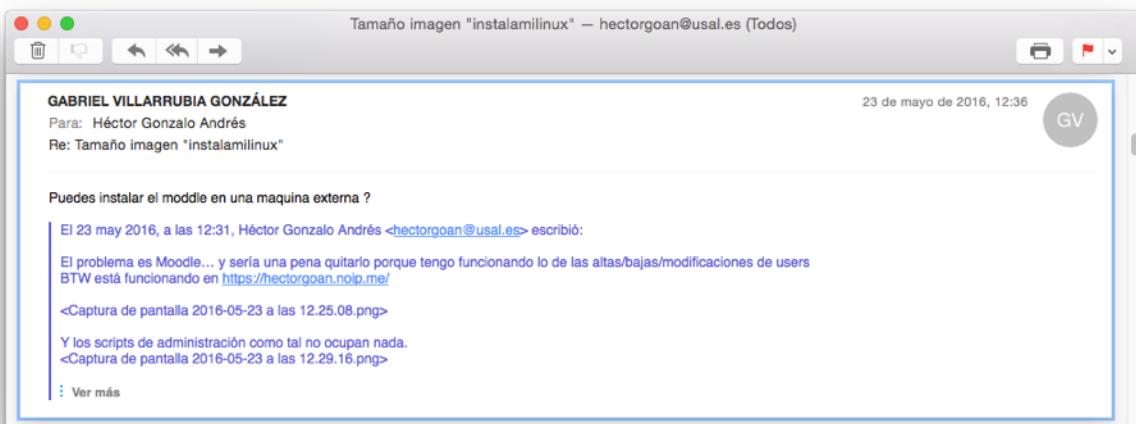
El principal reto a la hora de desplegar el sistema en producción era el del espacio.

El profesor Gabriel nos comentó que todo el tema de Moodle era lo que veía mas curioso de implementar y, tras tener todo montado en el entorno test nos pusimos en la situación de tener 2.5gb ocupados.

Esto, en el disco de 120gb del equipo test no es problema ninguno, pero la cosa se complica cuando en el de producción nos encontramos con que el espacio es mucho menor.

Como tenemos todo implementado, no queremos tirar el trabajo a la basura.

Tras intercambiar correos con Gabriel y Andrés (sysadmin) nos encontramos en la situación de que no se puede aumentar el espacio, y Gabi comenta que podemos dejar Moodle en un equipo externo.



En ese mismo instante empezamos a diseñar la arquitectura para permitir esto y nos damos cuenta de que el 90% del trabajo ya está hecho.

Basta con crear 8 scripts .pl para solucionarlo (si, son los marcados con EXTERN en las capturas)

De esta manera, introducimos también un par de cambios en la aplicación django para que, justo donde se llama a algún .pl que hace cambios en Moodle llame también a los nuevos scripts "EXTERN" localizados en otro servidor.

También tuvimos que eliminar el código de administración de Moodle de los scripts de administración del entorno de producción.

En un principio ya estaba todo... pero solo en un principio.

Cuando empezamos a hacer pruebas con el sistema nos damos cuenta de que los formularios tardan mucho en responder y los emails nunca llegan.

Esto es debido a la configuración de red del laboratorio. El módulo Email::Send::SMTP::Gmail de Perl no funciona, y nos está retrasando las operaciones con algunas partes del sistema una media de 20 segundos.

Decidimos por tanto utilizar el servidor que ya sirve Moodle como un servidor de salida de emails, por tanto tenemos que crear otros tres scripts .pl cuyo nombre sigue el patrón mailserv_.

Una vez realizados los cambios necesarios (eliminación de las notificaciones por mail en el servidor1, el del laboratorio, y desviado este trabajo al servidor2) el sistema funciona perfectamente en el entorno de producción.

:)

Se adjunta en la entrega el código del entorno test, junto con los scripts EXTERN.

Últimas notas

Creemos que no se nos olvida nada y que solo queda comentar el tema de las tarea CRON, una que hace backups y la otra que envía correos de informe diarios al administrador.

backups.sh

Los backups de la carpeta /home se hacen cada noche a las 2 AM al directorio /root/backups

```
1      #!/bin/bash
2      #Purpose = Backup of Users Data
3      #Created on 20-5-2016
4      #Author = AHSoftSol
5      #Version 1.0
6      #START
7      TIME=`date +%b-%d-%y`
8      FILENAME=backup-$TIME.tar.gz
9      SRCDIR=/home
10     DESDIR=/root/backups
11     tar -cpzf $DESDIR/$FILENAME $SRCDIR
12     #END
```

crontab -e

```
00 02 * * * /bin/bash /root/backups.sh
```

Se comentaba como tarea externa la posibilidad de hacer copias de seguridad en un servidor externo. Nos parece bastante sencillo. Como en el CRON se puede programar la ejecución de cualquier script, bastaría con que el script hiciera un sftp o ftp a un servidor externo y subiera la copia al mismo

Para solucionar el tema de tener que autenticarse en el servidor externo meteríamos la contraseña en un keychain o pondríamos como confiable nuestro equipo en el servidor almacén de backups, y entonces bastaría con añadir dos líneas al script propuesto por nosotros.

```
sftp user@backupserver
put $BACKUPFILEPATH
```

reports.sh

Necesitamos instalar el paquete acct

sudo apt-get install acct

```
1 #!/bin/bash
2 #Purpose = Daily report to the sysadmin
3 #Created on 20-5-2016
4 #Author = AHSoftSol
5 #Version 1.0
6 #START
7
8 ac -p >> usersConnectionTimes.txt
9 sa -n >> mostCalledPrograms.txt
10 sa -a >> cpuUsagePrograms.txt
11
12 mail -s "Reporte conexión usuarios" administrador <
usersConnectionTimes.txt
13 mail -s "Reporte programas mas llamados" administrador <
mostCalledPrograms.txt
14 mail -s "Reporte uso de cpu por programas" administrador <
cpuUsagePrograms.txt
15
16 #END
```

crontab -e

30 21 * * * /bin/bash /root/reports.sh

Nuestro administrador tendrá todos los días a las 21:30 de la noche el reporte del sistema

Extra:

Respecto al ejercicio extra en el que se regula el acceso a ciertas páginas a los usuarios de nuestra red (suponiendo la de la USAL), bastaría por montar un filtro de contenidos como el que nos explicó el compañero Iskren en clase. Nos decantaríamos por Squid 3 al cumplir este con la característica de ser software de dominio público y carácter open source comentada en el enunciado.

Tour

The screenshot displays the DIAWeb 2 website, which is a multi-service system for professors and students. At the top, there's a navigation bar with the A&H Software Solutions logo, a search icon, and links for Servicios, Nosotros, Contacto, Registro, and Login. Below the header, a banner reads "DIAWeb 2 Sistema multi-servicio a disposición de profesores y alumnos Departamento de informática y automática". The main content area is titled "Servicios" and contains several service icons:

- Apuntes**: Área de intercambio de apuntes.
- Blog personal**: Web individual disponible para todos los usuarios.
- Mail**: Mail interno al sistema. Puedes intercambiar mensajes con tus compañeros.
- Nube personal**: Accede o modifica tus archivos personales con este cliente ftp.
- Moodle**: Portal educativo donde aparecerán tus cursos, tareas y apuntes.
- Estado de servicios**: Comprueba cuáles de las funcionalidades del sistema están activas en este momento.
- Gestión de perfil**: Herramienta de configuración del perfil de usuario del sistema.

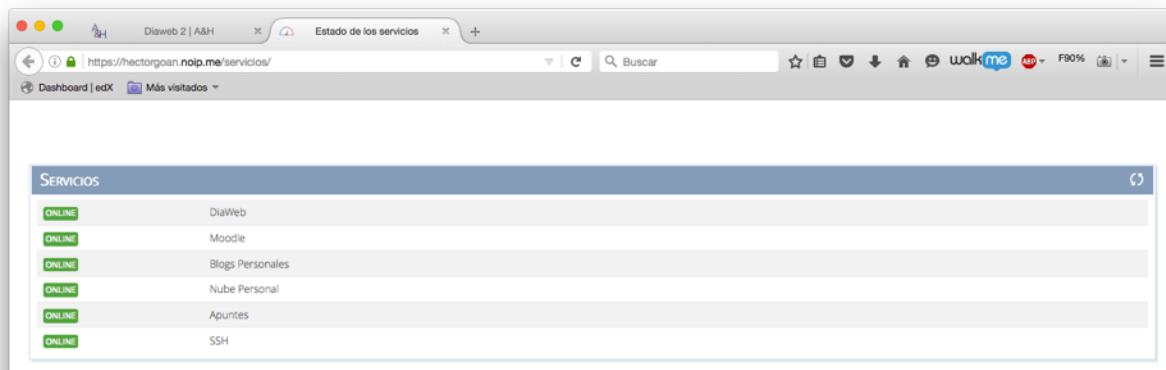
Below the services, there's a section titled "¿Perdido?" with the subtext "Te ayudamos a entender el sistema" and a blue circular icon. Further down, there's a "¡Conócenos!" section featuring two student profiles:

- Héctor**: CEO & Scrum Sysadmin. He has a profile picture and the text "Somos dos estudiantes del grado en ingeniería informática especializados en la administración de sistemas".
- Alberto**: VP & Happiness Manager. He also has a profile picture and the same descriptive text.

The bottom section is a "Contacto" form with fields for Nombre, e-mail, and Mensaje, along with an "Enviar" button. To the right, there's a "Dirección" section with address details for Dpto. Informática y Automática, Plaza de los Caídos, s/n, 37008 Salamanca (España), Tfno: (+34) 823 294653, and A&H Software Solutions, 795 Folsom Ave, Suite 600, San Francisco, CA 94107, Tfno: (+123) 456-7890. On the far right, there's a "Connect with us" section featuring social media icons for Facebook, LinkedIn, Google Plus, Twitter, Pinterest, and YouTube.

El usuario, en su primera conexión es recibido con una moderna página web en la que se le explican las funcionalidades del sistema, se le ofrece ayuda, se le pone cara al equipo detrás del desarrollo y se ofertan formas de contacto con el administrador.

En gris se muestran las funcionalidades las cuales para ser utilizadas requieren del registro/ identificación del usuario en la web. Destacamos de esta vista la capacidad de abrir las páginas de ayuda y la de **estado de los servicios**.



The screenshot shows a web browser window with the title 'Estado de los servicios'. The URL is https://hectorgaon.noip.me/servicios/. The page content is a table titled 'SERVICIOS' with the following data:

ONLINE	DiaWeb
ONLINE	Moodle
ONLINE	Blogs Personales
ONLINE	Nube Personal
ONLINE	Apuntes
ONLINE	SSH

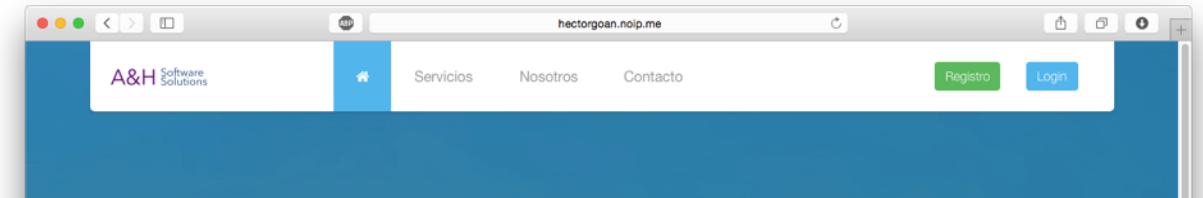
Ayuda:



The screenshot displays a series of help pages from the system:

- Registro en el sistema**: Instructions for registration, mentioning a welcome email.
- Entrar al sistema**: Instructions for logging in.
- Modificación de la cuenta**: Instructions for account modification, including personal data, blog, password change, and account cancellation.
- Recuperación de contraseña**: Instructions for password recovery.

En la barra superior de la página principal se nos ofrecen las funcionalidades de registro y de login.



Login:

A screenshot of the 'Login' page. It features a large central 'Login' heading. Below it are two input fields: 'Username:' and 'Password:', each with its own input box. Underneath these fields are three buttons: a red 'Volver' button, a green 'Entrar' button, and a blue 'Recuperar Contraseña' button. At the bottom left of the page, there is a small copyright notice: '© 2016 A&H Software Solutions. All Rights Reserved.'

Recuperar contraseña:

A screenshot of the 'Recuperar contraseña' (Forgot password) page. The page title is 'Recuperar contraseña'. Below it is a instruction: 'Introduce tu e-mail a continuación'. There is a single input field labeled 'Email:' with an associated input box. At the bottom of the form are two buttons: a red 'Cancelar' button and a green 'Enviar' button. At the very bottom left of the page is a copyright notice: '© 2016 A&H Software Solutions. All Rights Reserved.'

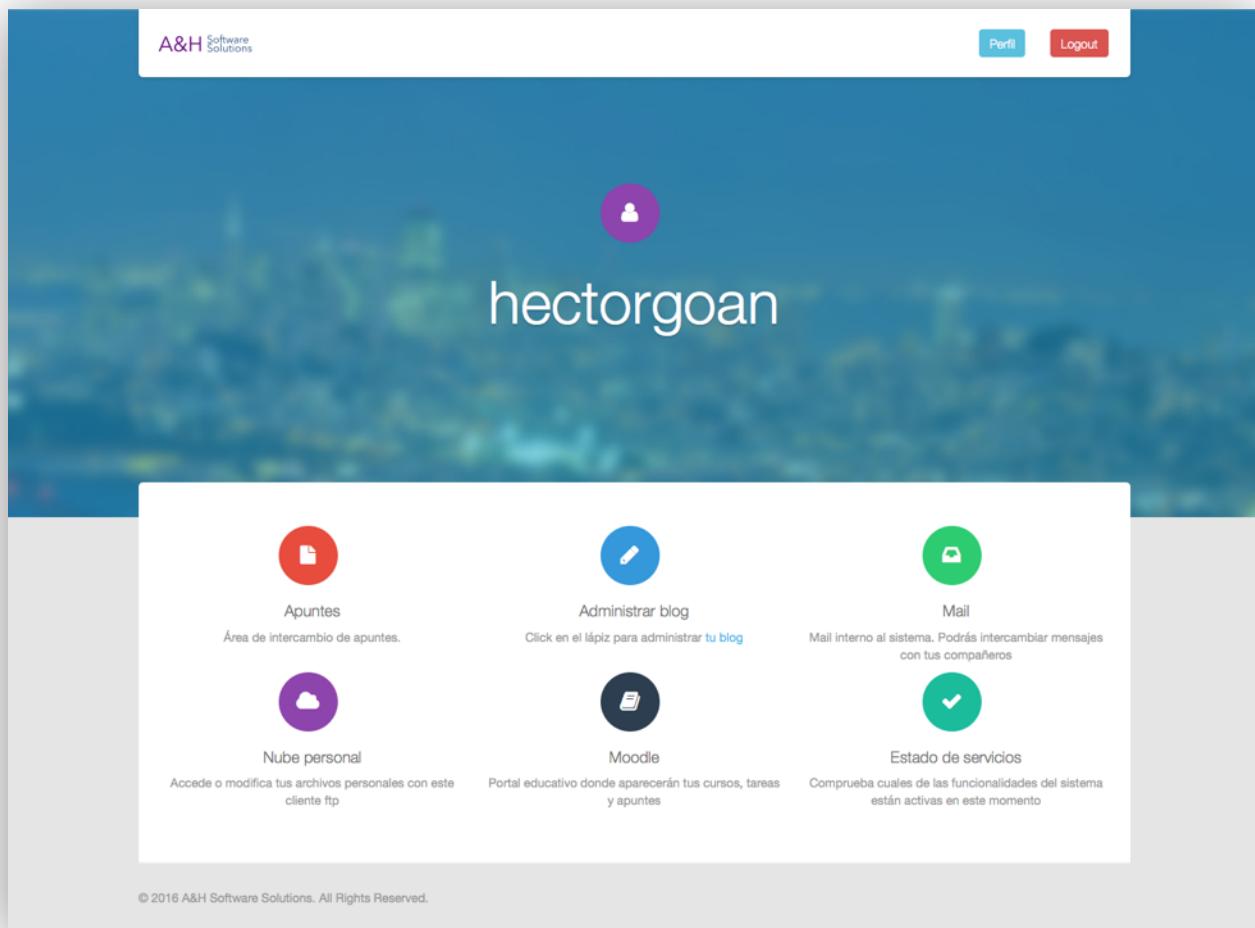
Registro:

The screenshot shows a web browser window with the URL `hectorgoan.noip.me` in the address bar. The page header features the **A&H Software Solutions** logo. The main content area has a light gray background and displays the word "Registro" in large, bold, dark gray letters. Below it, the text "Introduce tus datos a continuación" is centered. There are four input fields labeled "Nombre:", "Nick:", "Email:", and "Direccion:". To the right of these fields is a "Rol:" label followed by two radio buttons: "Alumno" and "Profesor". At the bottom of the form are two buttons: a red "Cancelar" button and a green "Enviar" button. The footer of the page contains the copyright notice "© 2016 A&H Software Solutions. All Rights Reserved."

Estamos omitiendo la captura de páginas de confirmación de operación exitosa para hacer más breve la memoria. Por mostrar alguna, la de **recuperar contraseña** sería:

The screenshot shows a web browser window with the URL `hectorgoan.noip.me` in the address bar. The page header features the **A&H Software Solutions** logo. The main content area has a light gray background and displays the word "Hecho" in large, bold, dark gray letters. Below it, the text "Mensaje enviado a hectorgoan@gmail.com" and "Encontrarás las instrucciones para volver a iniciar sesión" is centered. At the bottom of the page is a single red "Volver" button. The footer of the page contains the copyright notice "© 2016 A&H Software Solutions. All Rights Reserved."

Logged In

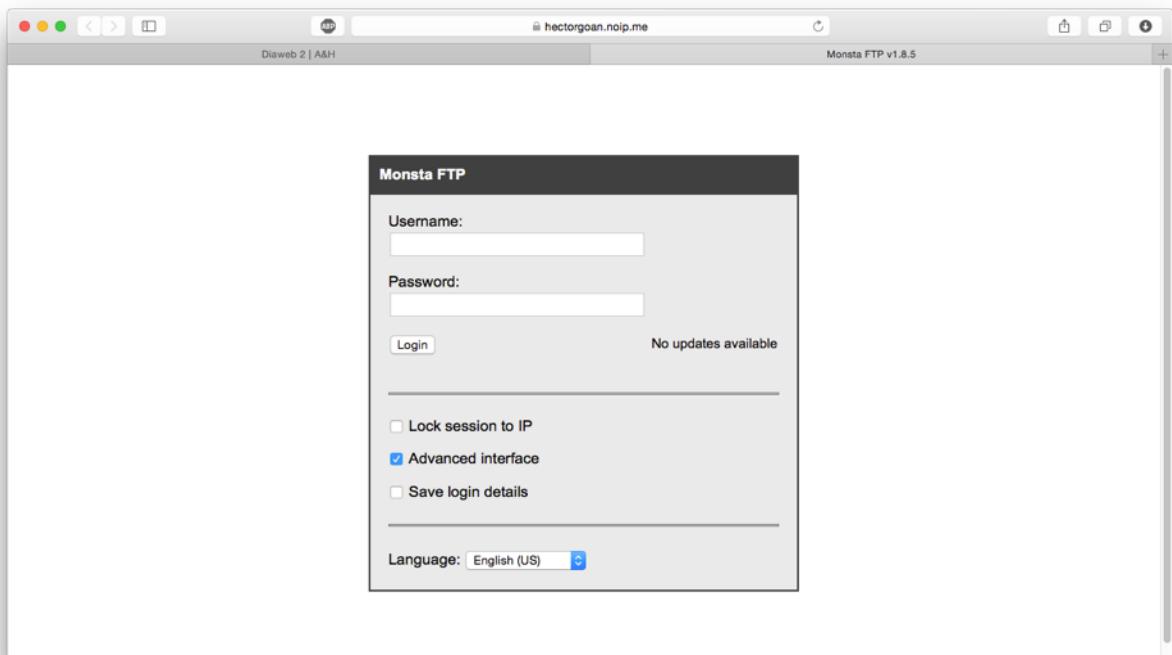


Queríamos que la página principal del sistema fuera sencilla, moderna y de alguna manera con influencias del mundo móvil, haciendo que cada una de las categorías aparezcan como una aplicación, cada una de ellas con su ícono característico, en vez de como una palabra entre un montón de palabras.

Apuntes/Nube personal

Como se indicó desde un principio, para esta funcionalidad se utiliza el gestor de archivos PHP llamado MonstaFTP.

Esta es la apariencia de la pantalla de login.

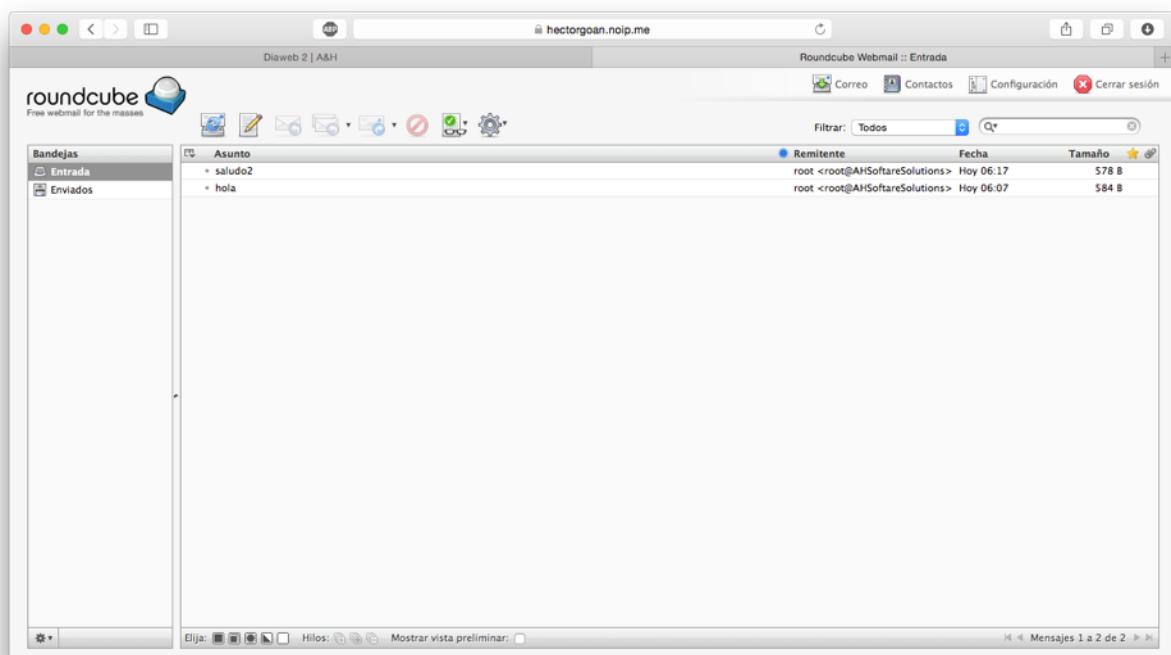
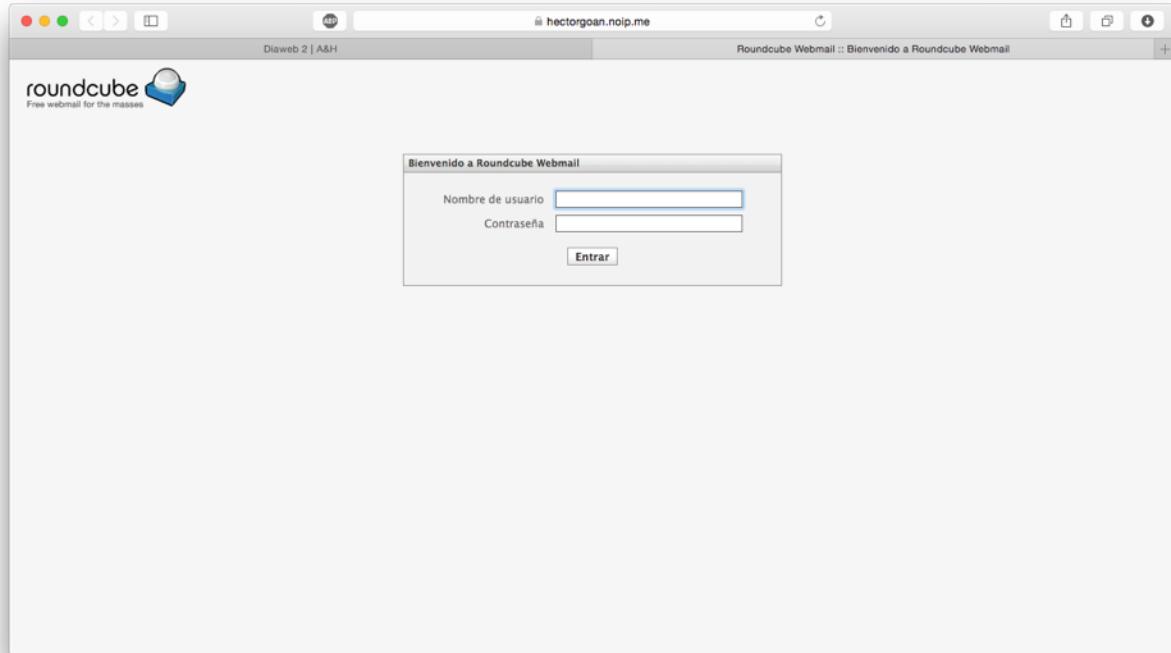


Una vez dentro de la aplicación, se continúa con la estética minimalista que caracteriza a nuestro sistema

A screenshot of the Monsta FTP application window. The top menu bar includes "Recargar", "Descargar", "Cortar", "Copiar", "Pegar", "Renombrar", "Borrar", "Permisos", and "Fin de Sesión". The main area shows a file list in a table format with columns: Nombre, Tamaño, Fecha, Hora, Usuario, Grupo, and Permisos. The table contains three entries: "mail" (23/05/16, 02:12, 1002, 1002, drwxr-xr-x), "public_html" (27/05/16, 02:09, 1002, 1002, drwxr-xr-x), and "condiciones.txt" (740, 23/05/16, 02:12, -, -, -rw-r--r--). At the bottom are buttons for "Nueva Carpeta", "Nuevo Archivo", "Fetch File", and "Subir Archivos". A status bar at the bottom right indicates "Servidor: localhost Usuario: hectorgoan Límite de Subidas: 128MB".

Mail

Una vez más, continuamos con la estética minimalista y la gama de colores gris/azul



Configuración de Perfil

Se permite la realización de varias acciones relacionadas con nuestro perfil.

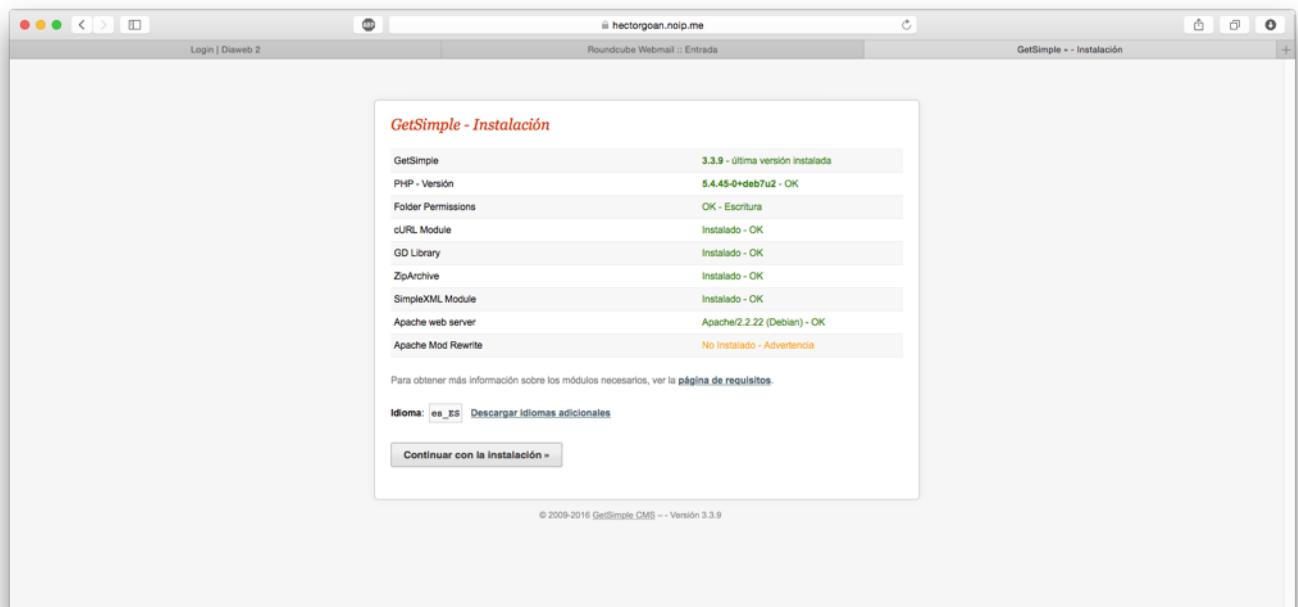
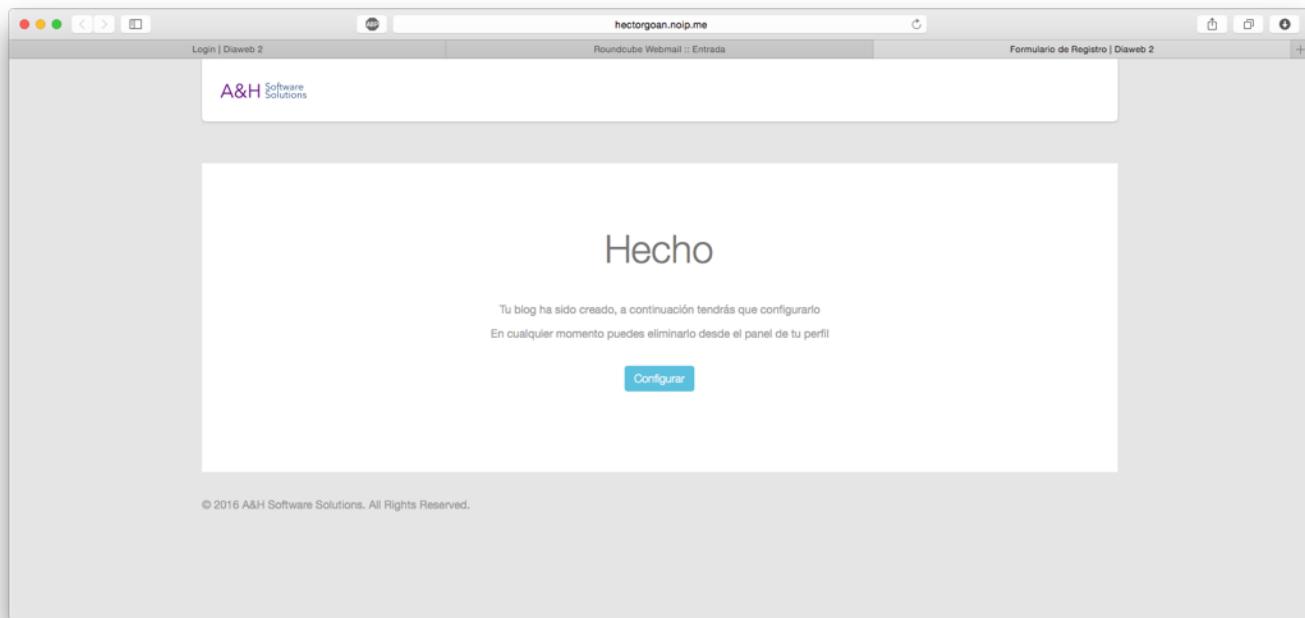
The screenshot shows a web-based profile configuration interface for a user named 'hectorgoan'. The top navigation bar includes the A&H Software Solutions logo. The main content area is divided into several sections:

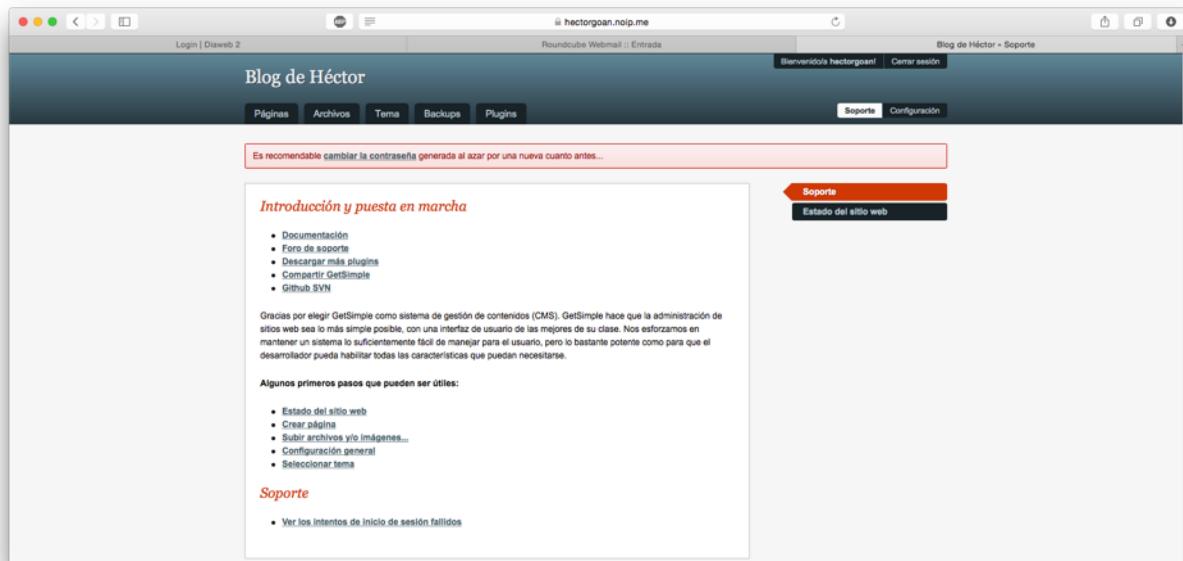
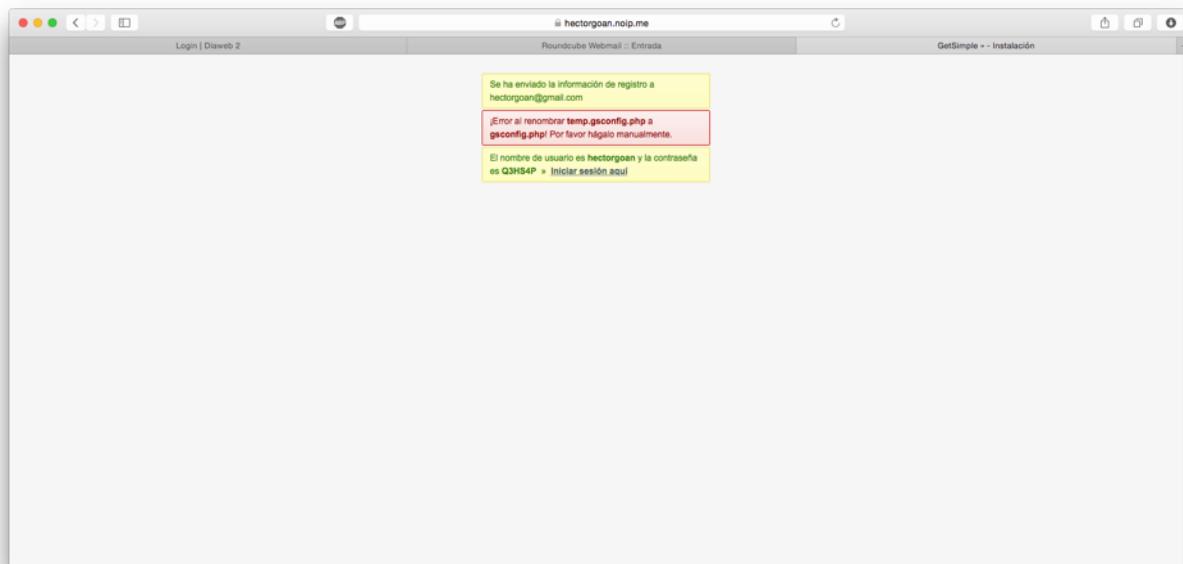
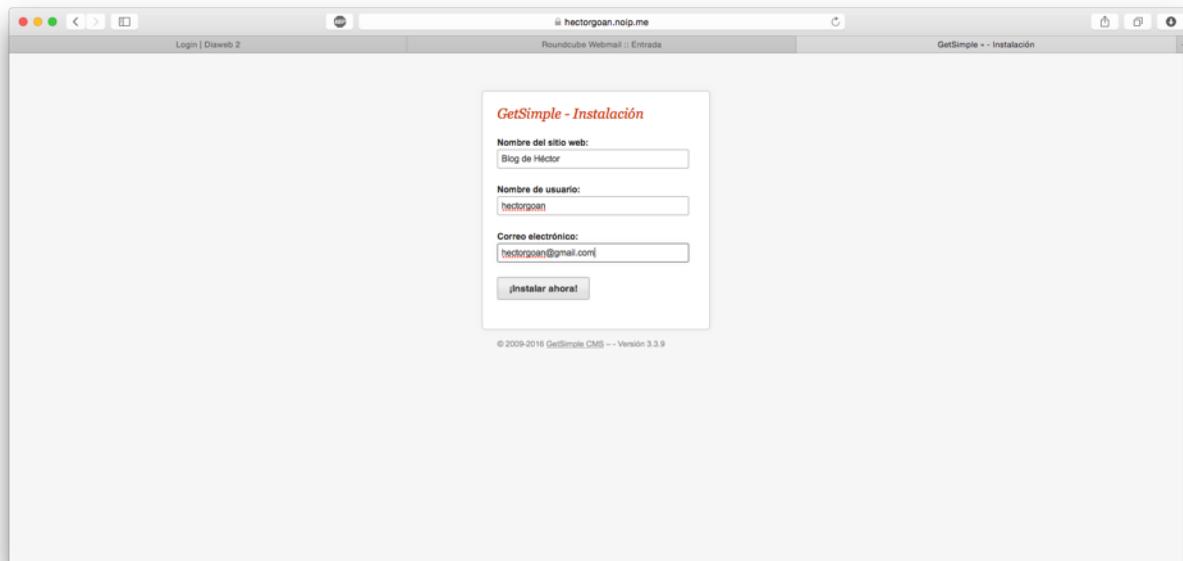
- Editar cuenta: hectorgoan**: Displays basic account information:
 - Nombre: Héctor Gonzalo Andrés
 - E-Mail: hectorgoan@gmail.com
 - Dirección: calle Granados Nº3 Bajo C
- Modificar datos personales**: A form for updating personal data with fields for Nombre, Email, and Dirección, along with Cancelar and Guardar buttons.
- Modificar web personal**: Shows the URL hectorgoan.noip.me/~hectorgoan and buttons for Crear Blog and Eliminar Blog. A note states: "Si se elimina el blog, se puede crear una web personal subiéndola a public_html mediante la nube personal".
- Cambiar contraseña**: A note stating "No se preocupe si la pierde, existe sistema de recuperación en la pantalla de login" and a Cambiar contraseña button.
- Cancelar cuenta**: A note stating "Si elimina el usuario, perderá todos sus archivos y no podrá entrar en el sistema" and an Eliminar usuario button.

At the bottom left, a copyright notice reads: "© 2016 A&H Software Solutions. All Rights Reserved."

Crear Blog

Lanza un asistente de instalación de un blog personal.





Perfil de usuario

Nombre de usuario: hectorgoan Correo electrónico: hectorgoan@gmail.com

Nombre a mostrar: Visible públicamente, distinto al nombre de usuario

Uso horario local: Ninguno Idioma: es_ES

Habilitar editor HTML

Nueva contraseña: Confirmar contraseña:

Guarda configuración o Cancelar

Blog de Héctor

Páginas Archivos Tema Backups Plugins

Administración de páginas

TÍTULO DE LA PÁGINA	FECHA
Hola!	27 May 2016

1 páginas en total

Ver todas las páginas
Crear nueva página
Administrador Menú

BLOG DE HÉCTOR

Hola!

Bienvenido a mi blog personal

PUBLISHED ON May 27th, 2016

@hectorgoan

2016 - BLOG DE HÉCTOR
CARDINAL THEME BY CAGINTRANET
POWERED BY GETSIMPLE

Moodle

The screenshot shows a web browser window with the following details:

- Title Bar:** Diaweb 2 | A&H, hectorgoan.noip.me, Roundcube Webmail :: Entrada, Studium 2 - DIA.
- Header:** DIA Moodle, Español - Internacional (es) ▾, Usado no se ha identificado. (Acceder)
- Main Content:**
 - NAVIGACIÓN:** Página Principal, Cursos
 - CALENDARIO:** CALENDARIO, mayo 2016 (Dom, Lun, Mar, Mié, Jue, Vie, Sáb). The calendar shows dates from May 1 to May 31, 2016.
 - Text: Usado no se ha identificado. (Acceder) moodle

The screenshot shows a web browser window with the following details:

- Title Bar:** Diaweb 2 | A&H, hectorgoan.noip.me, Roundcube Webmail :: Entrada, Studium 2 - DIA: Acceder al sitio.
- Header:** DIA Moodle, Español - Internacional (es) ▾, Usado no se ha identificado.
- Main Content:**
 - Studium 2 - DIA**
 - Página Principal ▶ Acceder al sitio
 - Acceder**
 - Form fields:
 - Nombre de usuario: [Input Field]
 - Contraseña: [Input Field]
 - Recordar nombre de usuario
 -
 - Links:
 - ¿Olvidó su nombre de usuario o contraseña?
 - Las 'Cookies' deben estar habilitadas en su navegador ⓘ
 - Text: Algunos cursos permiten el acceso de invitados
 - Buttons:
 -
 - Text at bottom: Usado no se ha identificado. Página Principal

Monitor de sistema:

SISTEMA	
Nombre de host	AHSoftwareSolutions
Sistema Operativo	Debian GNU/Linux 7.10 (wheezy)
Versión del kernel	3.2.0-4-686-pae
Tiempo activo	2 hours and 20 minutes
Último arranque	2016-05-27 14:16:30
Usuarios activos	1
Fecha y hora del servidor	Fri May 27 16:36:31 CEST 2016

RED			
Interfaz	IP	Recibido	Transmitido
eth0	192.168.1.110	1.68 MB	3.61 MB
lo	127.0.0.1	421.21 KB	421.21 KB

USO DE DISCO					
Sistema de archivos	Montaje	Uso	Libre	Usado	Total
rootfs	/		99.13 GB	2.52 GB	107.08 GB

MEMORIA	
Usada %	
Usada	188.67 MB
Libre	1.3 GB
Total	1.48 GB

INTERCAMBIO	
Usado %	
Usado	0 B
Libre	3 GB
Total	3 GB

ULTIMO LOGIN	
root	May Fri +0200 2017

PING	
google.com	8.093 ms
yahoo.com	154.649 ms

ESTADO DE LOS SERVICIOS		
ONLINE	Apache2	80
ONLINE	Email Server (incoming)	993
OFFLINE	Email Server (outgoing)	587
ONLINE	FTP (vsftpd)	21
ONLINE	MySQL	3306
ONLINE	SSH	130

Conclusión

Esta ha sido, sin duda una de las prácticas mas interesantes a realizar en lo que llevamos de grado.

Hemos aprendido a utilizar varias tecnologías diferentes y a combinarlas para crear algo mas grande.

Ha habido momentos de desesperación en los que algunas partes no salían adelante, pero siempre podíamos solucionarlo tirando de documentación por internet.

Lo que mas guerra ha dado ha sido el tema del e-mail, el resto ha sido bastante mas llevadero ya que los problemas surgían muy de vez en cuando y solo nos ralentizaban por un par de horas, cosa que con el mail se alargó varios días.

Aún con todo, el sabor de boca que queda después de realizarla es bueno, ya que te deja con la sensación de haber hecho algo aplicable al "mundo real"

Fin .