

Proyecto 1

Carlos Gerardo Acosta Hernández

Andrea Itzel González Vargas

Luis Pablo Mayo Vega

Redes de Computadoras
Facultad de Ciencias, UNAM

Índice

1. Preliminares

1.1. Consideraciones sobre el proyecto

A manera de introducción, es relevante que consignemos en esta primera parte información general sobre nuestro proyecto que sirva de panorama general y permita explorar con mayor facilidad la organización de los apartados contenidos en el documento.

Comencemos por decir que para la materia de *Redes de computadoras* impartida en la *Facultad de ciencias*, UNAM durante el ciclo 2018-1, realizamos como primer proyecto del curso la implementación de un servicio *web* (una aplicación web y un servicio de correo electrónico), montado sobre una infraestructura de tres servidores que nuestro equipo de trabajo configuró. Para tal labor, pusimos en práctica diversos conceptos presentados en clase y conjuntamos múltiples servicios de red afines –puestos a disposición del público por terceros– que listamos en la sección siguiente (??); herramientas necesarias para la elaboración del proyecto.

La infraestructura del proyecto consta de dos servidores dedicados a los componentes de la aplicación web, más un tercero dedicado al servicio de correo electrónico. El entramado de red involucrado puede ser visualizado en el diagrama de red mostrado en la sección ???. Decidimos implementar como aplicación web (?? y ??) un *CRUD*¹ con un registro e inicio de sesión para usuarios; la primera instancia refleja el buen funcionamiento de la conexión entre los servidores de la aplicación, mientras que los dos casos de uso mencionados ulteriormente significarían una restricción de acceso a las demás funciones de la aplicación establecido como un nivel de seguridad. El sitio web puede ser accesado mediante la siguiente dirección de Internet:

<http://tabon.ga>

Añadido a las funcionalidades del servicio web, configuramos un servidor de correo electrónico (??), con posibilidad de permitir al usuario enviar correos a cualquier dirección utilizando una dirección de correo electrónico que incluye nuestro dominio; igualmente recibirlos. Este servicio es accesible desde la siguiente dirección:

<http://mail.tabon.ga>

La idea de configurar nuestros registros *DNS* de esa manera, a pesar de estar alojado el servicio en un tercer servidor independiente a los de la aplicación, era dar una sensación de cohesión entre los servicios finales que ofrece nuestro servicio web.

Las diferentes configuraciones de seguridad aplicados a nuestro proyecto, bajo la línea de solicitudes del documento de especificaciones provisto en el curso, se puede encontrar en la sección ??.

Ahora que hemos presentado a grandes rasgos la composición de nuestro proyecto, nos es importante mencionar las diferentes tecnologías, servicios y herramientas que utilizamos para cubrir los diferentes rubros que nos fueron demandados.

¹Create Read Update Delete

1.2. Tecnologías, servicios y herramientas utilizadas

lalalala

2. Configuración de la red

2.1. Diagrama de Red

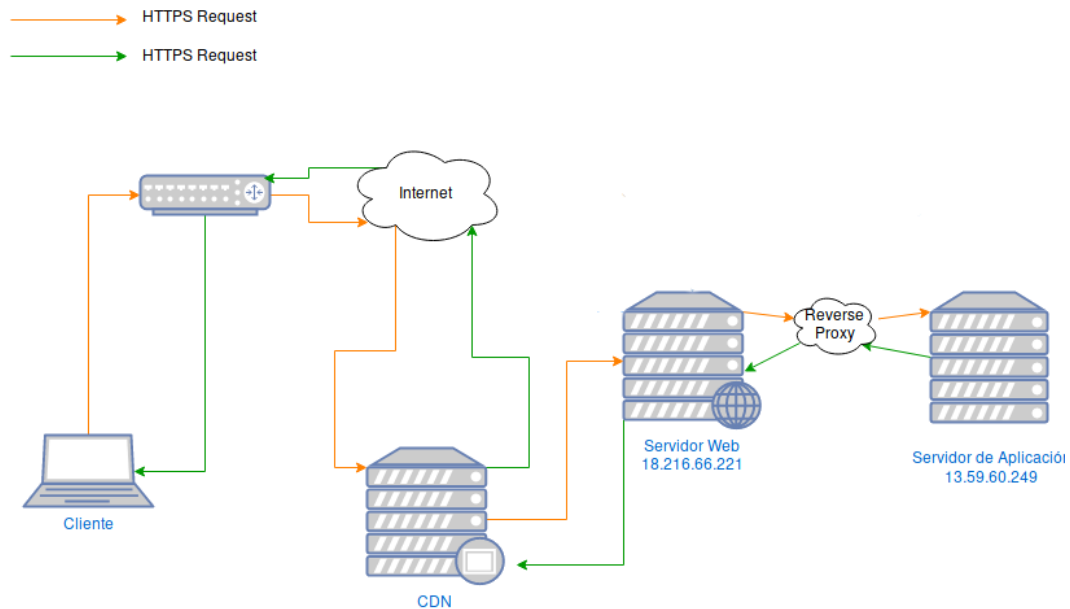


Figura 1: Diagrama de red de los servidores web, de aplicación y de correo electrónico

2.2. DNS

El servicio de DNS que utilizamos es Route 53 de Amazon, pues está integrado en la misma plataforma que los servidores y al ser diseñados para trabajar en conjunto, la integración fue mucho más fácil que con Cloudflare.

Nameservers

You can change where your domain points to here. Please be aware changes can take up to 24 hours to propagate.

- ☐ Use default nameservers
- ☒ Use custom nameservers (enter below)

Nameserver 1

NS-174.AWSDNS-21.COM

Nameserver 2

NS-1282.AWSDNS-32.ORG

Nameserver 3

NS-717.AWSDNS-25.NET

Nameserver 4

NS-1586.AWSDNS-06.CO.UK

Nameserver 5

Figura 2: Nameservers de Amazon agregados en la configuración de freenom

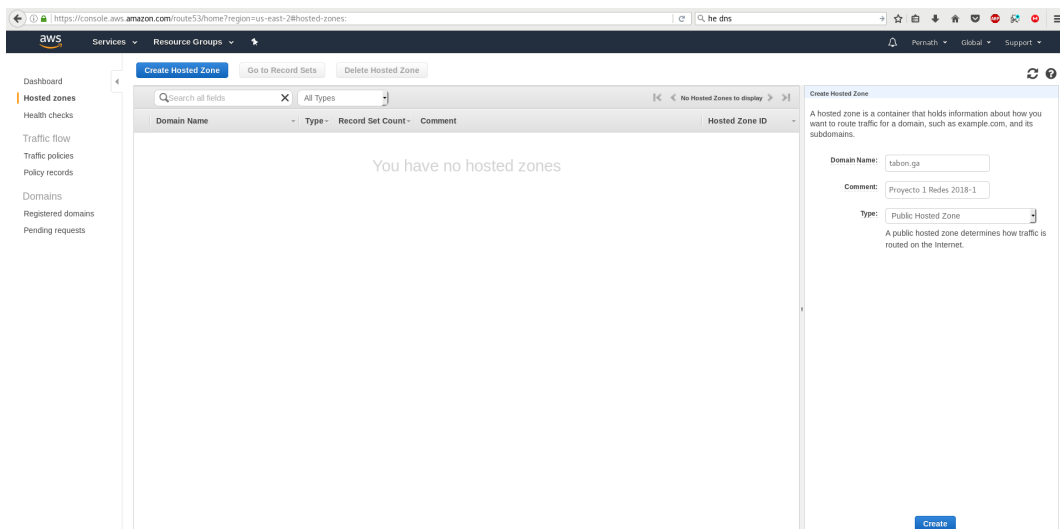


Figura 3: Panel de control de Route 53 sin zonas creadas

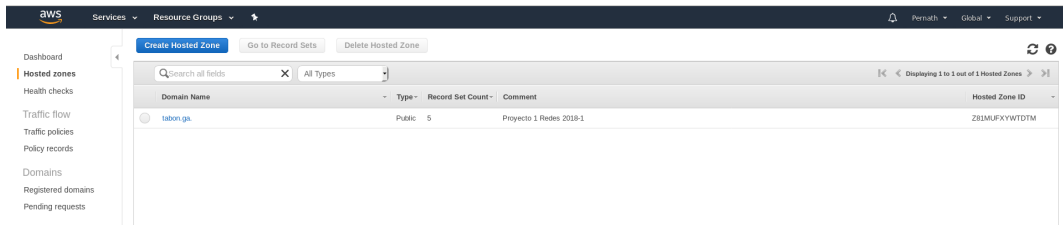


Figura 4: zonas creadas

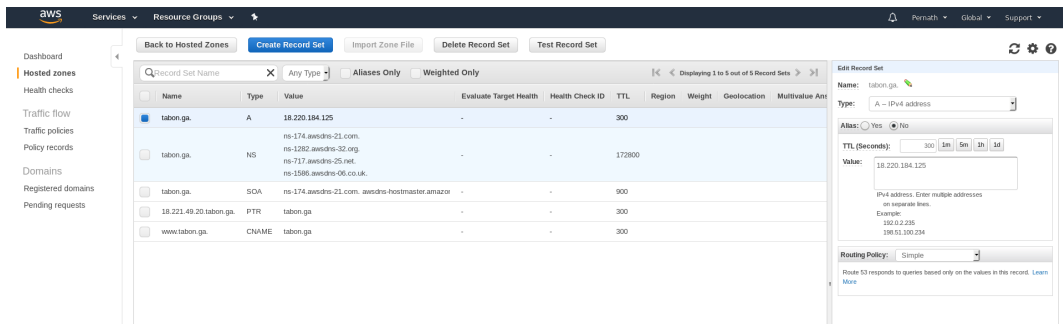


Figura 5: Detalles del registro A para IPv4

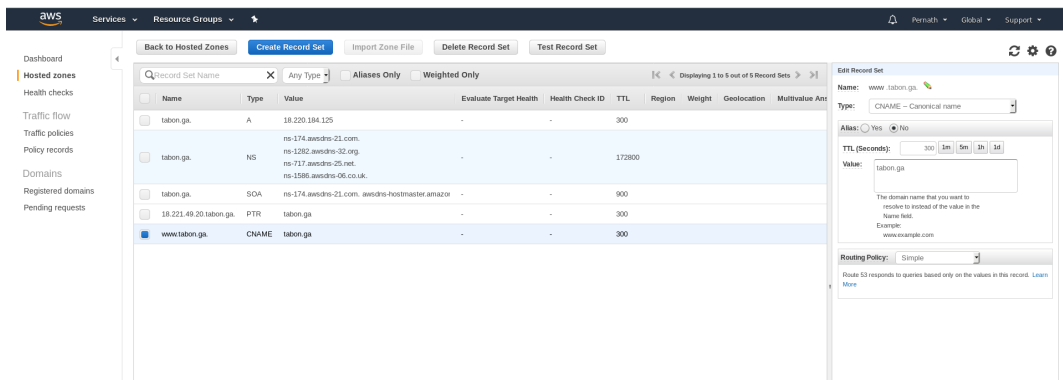


Figura 6: Detalles del registro CNAME para nombre canónico

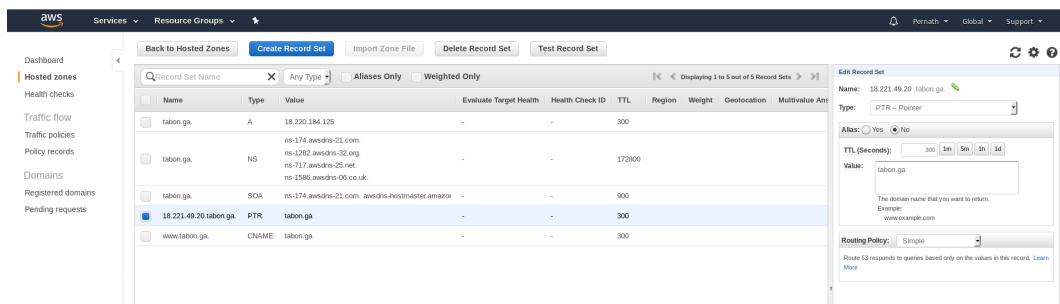


Figura 7: Detalles del registro PTR

<input type="checkbox"/>	Name	Type	Value
<input type="checkbox"/>	tabon.ga.	A	18.216.66.221
<input type="checkbox"/>	tabon.ga.	MX	10 mail.tabon.ga
<input type="checkbox"/>	tabon.ga.	NS	ns-115.awsdns-14.com. ns-689.awsdns-22.net. ns-1054.awsdns-03.org. ns-1779.awsdns-30.co.uk
<input type="checkbox"/>	tabon.ga.	SOA	ns-174.awsdns-21.com. awsdns-hostma
<input type="checkbox"/>	tabon.ga.	SPF	"v=spf1 mx ~all"
<input type="checkbox"/>	18.216.66.221.tabon.ga.	PTR	tabon.ga
<input type="checkbox"/>	ipv6.tabon.ga.	AAAA	2600:1f16:302:8901:7334:95f6:e156:2b
<input type="checkbox"/>	mail.tabon.ga.	A	13.58.30.114
<input type="checkbox"/>	mail.tabon.ga.	NS	ns-115.awsdns-14.com. ns-689.awsdns-22.net. ns-1054.awsdns-03.org. ns-1779.awsdns-30.co.uk
<input type="checkbox"/>	www.tabon.ga.	CNAME	tabon.ga

Figura 8: Vista de todos los registros

2.3. Configuración de direcciones IPv4 e IPv6

Se usaron 3 instancias de la plataforma EC2 de Amazon para los servidores de la aplicación, cada una con la distribución Ubuntu 16.04 y con manejo a través de ssh.

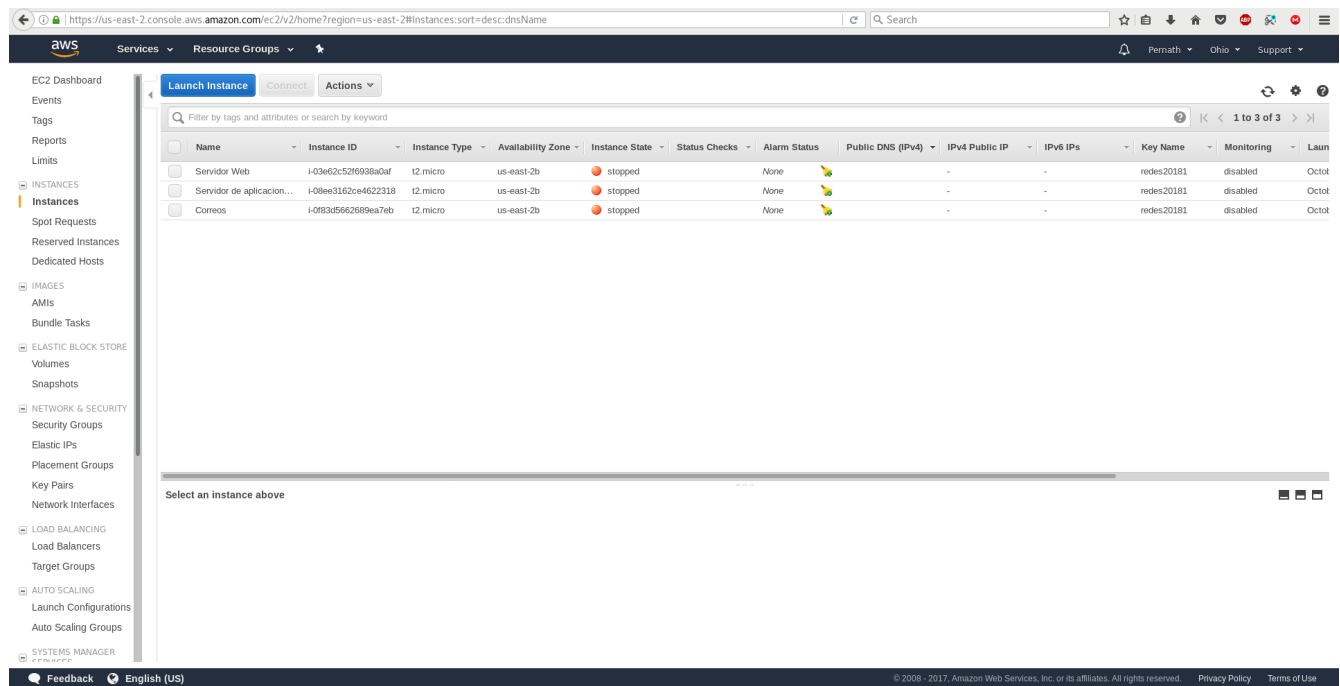


Figura 9: panel de control de las VPS

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name
Servidor Web	i-03e62c52f6938a0af	t2.micro	us-east-2b	running	2/2 checks ...	None	ec2-18-220-184-125.us-east-2.compute.amazonaws.com	18.220.184.125	-	redes2018

Figura 10: Servidor web en ejecución

2.3.1. Configuración de IPv4 públicas

Por defecto, AWS asigna direcciones IP públicas para las máquinas de manera dinámica. Es decir, cada vez que la instancia se inicializa después de haber sido detenida obtiene una IP pública diferente, lo que genera problemas al agregar servicios como DNS. Sin embargo, el mismo Amazon EC2 proporciona una herramienta que nos permite obtener IPv4 estáticas para nuestras instancias y las llaman *direcciones IP elásticas*.

Para asignar una dirección IP elástica debemos ingresar en la consola y AWS para EC2 y en el panel de navegación de la izquierda seleccionar la opción **Elastic IPs**.

Allocate new address

Actions ▾

<< < 1 to 3 of 3 > >>

<input type="checkbox"/>	Elastic IP ▴	Allocation ID ▾	Instance ▾	Private IP address ▾	Scope
<input type="checkbox"/>	13.58.30.114	eipalloc-c8b0fce6	-	-	vpc
<input type="checkbox"/>	13.59.60.249	eipalloc-68b6fa46	i-08ee3162ce4622318	172.31.19.61	vpc
<input type="checkbox"/>	18.216.66.221	eipalloc-eeb2fec0	i-03e62c52f6938a0af	172.31.21.60	vpc

Elegimos **Allocate new address** para obtener una nueva dirección elástica y luego confirmamos con **Allocate**

[Addresses](#) > **Allocate new address**

Allocate new address

Allocate a new Elastic IP address by selecting the scope in which it will be used

* Required

[Cancel](#)

Allocate

Figura 11: Asignación de IP elástica

Allocate new address



New address request succeeded

Elastic IP **18.216.55.176**

Close

Figura 12: Mensaje de éxito en la asignación de la dirección

Para asociar una dirección elástica a una instancia en ejecución nuevamente ingresamos en la opción **Elastic IPs** del panel de navegación de la consola de EC2.

Elegimos la dirección que recién se nos asignó y en su menú **Action** elegimos la opción **Associate address**.

Luego seleccionamos la instancia a la que queramos asociar la IPv4 pública y damos click en **Associate**.

2.3.2. Configuración de IPv6

Para la asignación de IPv6 necesitamos crear una *gateway* de solo salida a través de una VPC (Virtual Private Cloud). Esta gateway les permite a las máquinas que estén en la VPC establecer conexiones por IPv6 a Internet, pero impide que se hagan conexiones por IPv6 con las instancias desde algún lugar en internet.

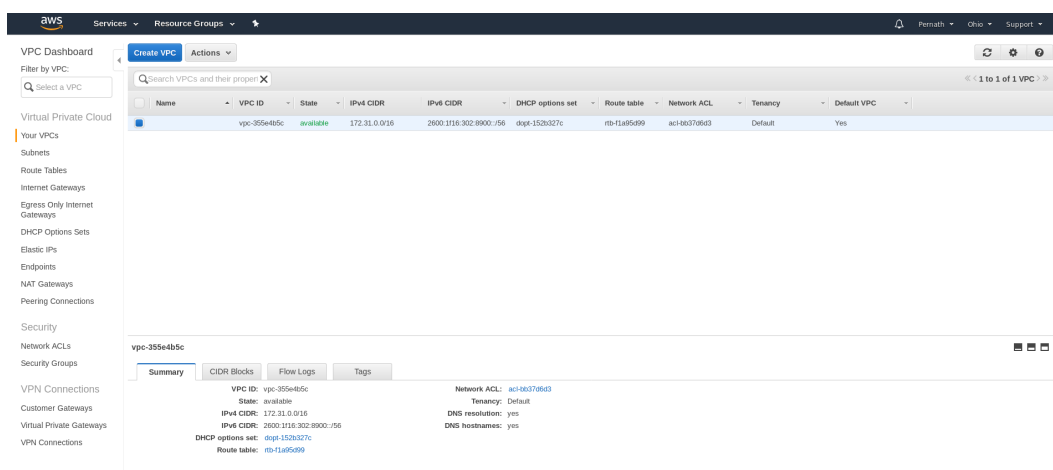


Figura 13: Creación de una VPC para las instancias de EC2

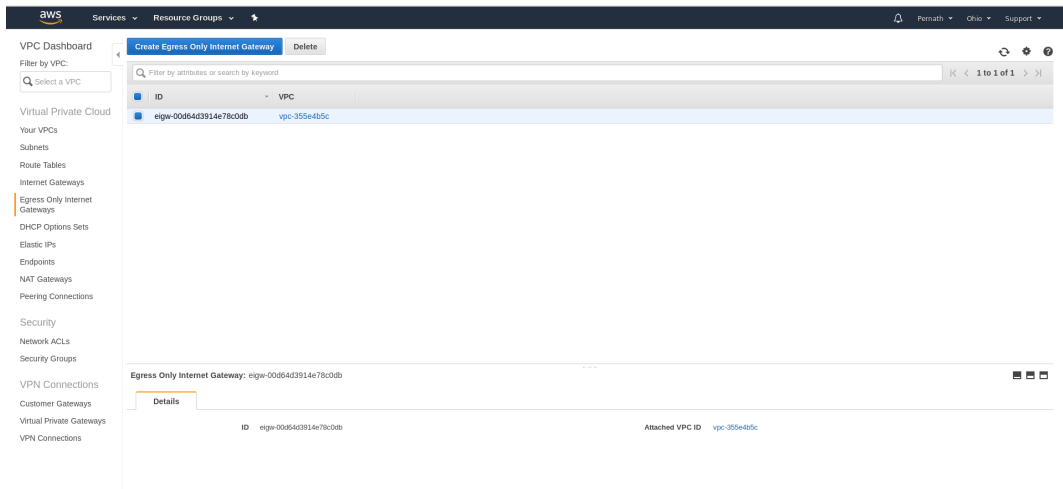


Figura 14: En el panel de VPC aparece la opción de crear una gateway de solo salida para la VPC que elijamos

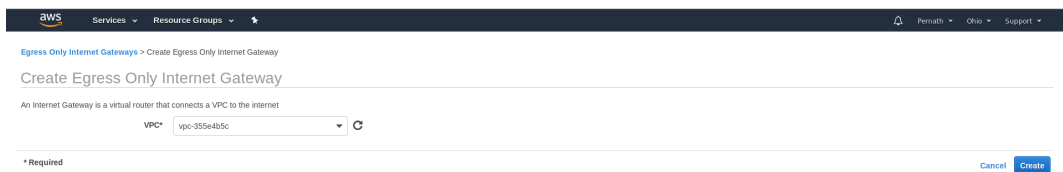


Figura 15: Creación de la gateway de solo salida para la VPC que contiene nuestras instancias de EC2

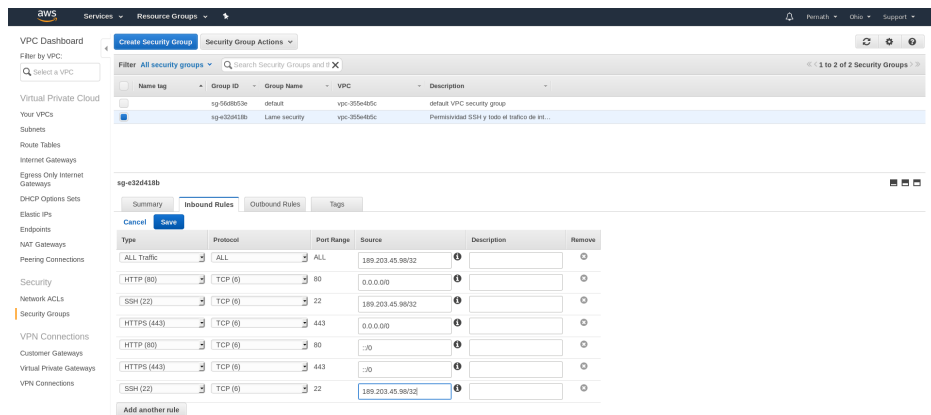


Figura 16: Editamos el grupo de seguridad (firewall) para que permita el tráfico de salida de la VPC

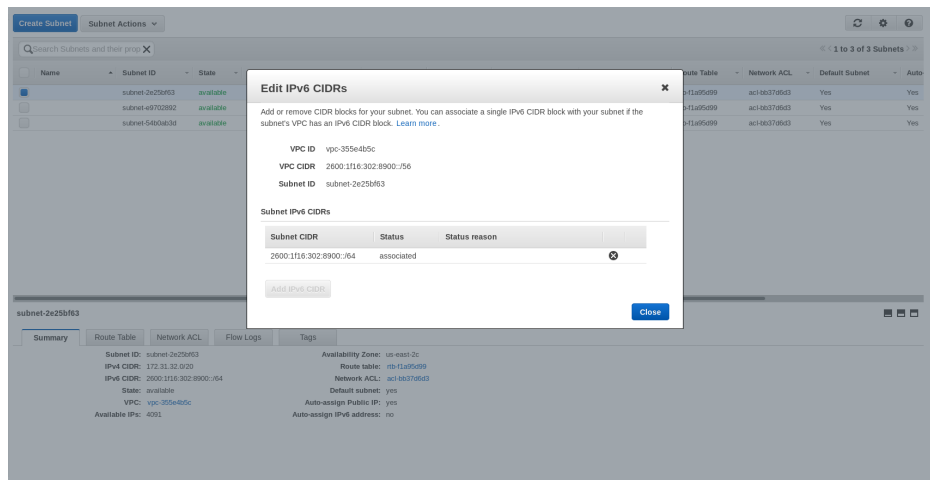


Figura 17: Creación del bloque CIDR para la generación de direcciones IPv6 públicas

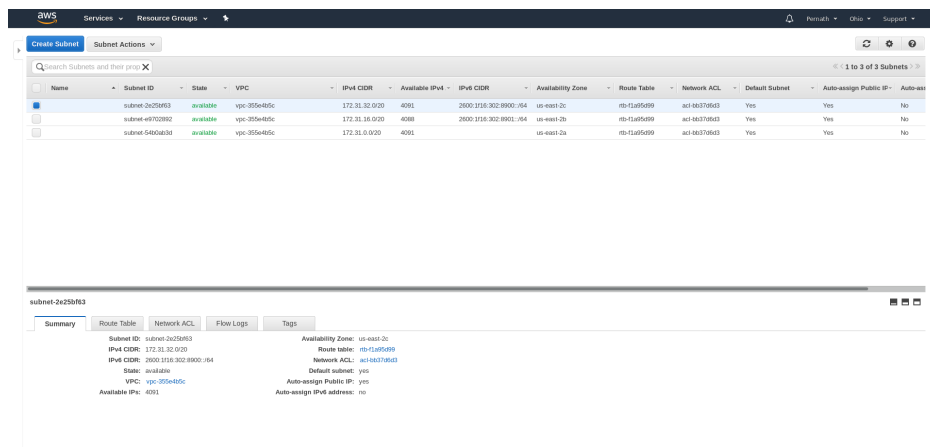


Figura 18: Subredes para las instancias

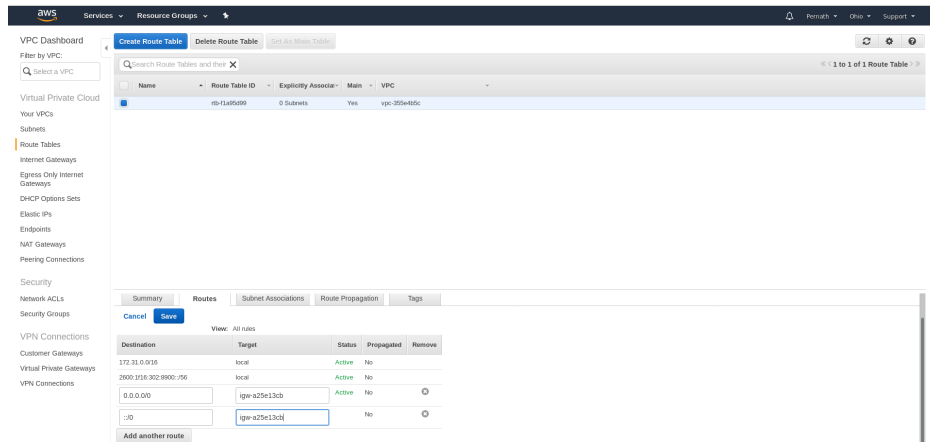


Figura 19: Subred pública en la VPC para asignarles direcciones generadas por el bloque CIDR

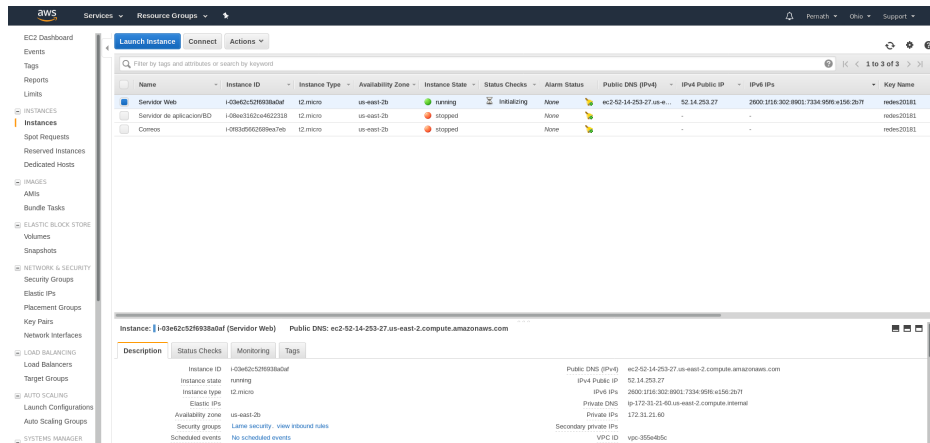


Figura 20: Asignación de direcciones IPv6 a los servidores

3. Servidor de Aplicación

3.1. Diagrama de la base de datos

Dada la simpleza de la aplicación, la base de datos solo cuenta con una tabla para los usuarios con sus respectivos atributos.

auth_user
+ id: int(11)
+ password: varchar(128)
+ last_login: datetime(6)
+ is_superuser: tinyint(1)
+ username: varchar(150)
+ first_name: varchar(30)
+ last_name: varchar(30)
+ email: varchar(254)
+ is_staff: tinyint(1)
+ is_active: tinyint(1)
+ date_joined: datetime(6)
+ crearUsuario()
+ editarUsuario()
+ verUsuario()
+ eliminarUsuario()

Figura 21: Tabla de usuario

3.2. Objetivo de la aplicación

La aplicación fue desarrollada con el fin de ilustrar las funciones básicas de una aplicación web con base de datos, es decir, *crear*, *editar*, *ver* y *eliminar* y que pueden asociarse a distintos métodos HTTP de la capa de aplicación como PUT, POST, GET, PATCH o DELETE.

A partir de esta pequeña aplicación *CRUD* puede escalarse a proyectos más grandes y de mayor complejidad gracias a la escalabilidad de **Django**, el framework usado.

3.3. Uso de la aplicación

Al ingresar al sitio, el usuario se encontrará con dos enlaces: uno para registrarse en la plataforma y otro para iniciar sesión en ésta.

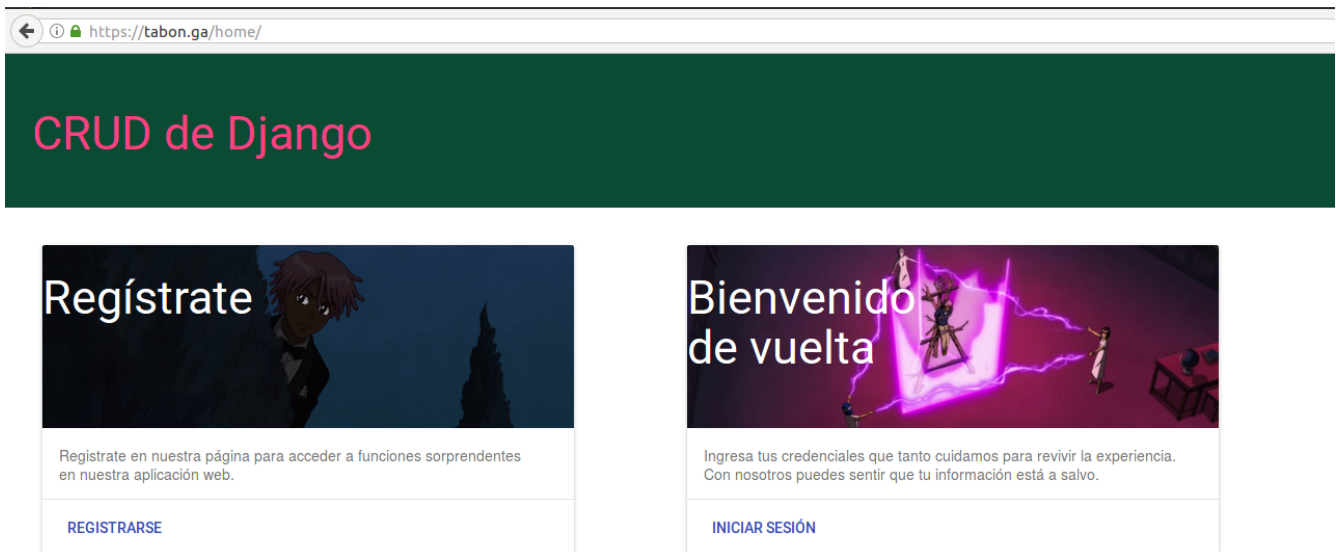


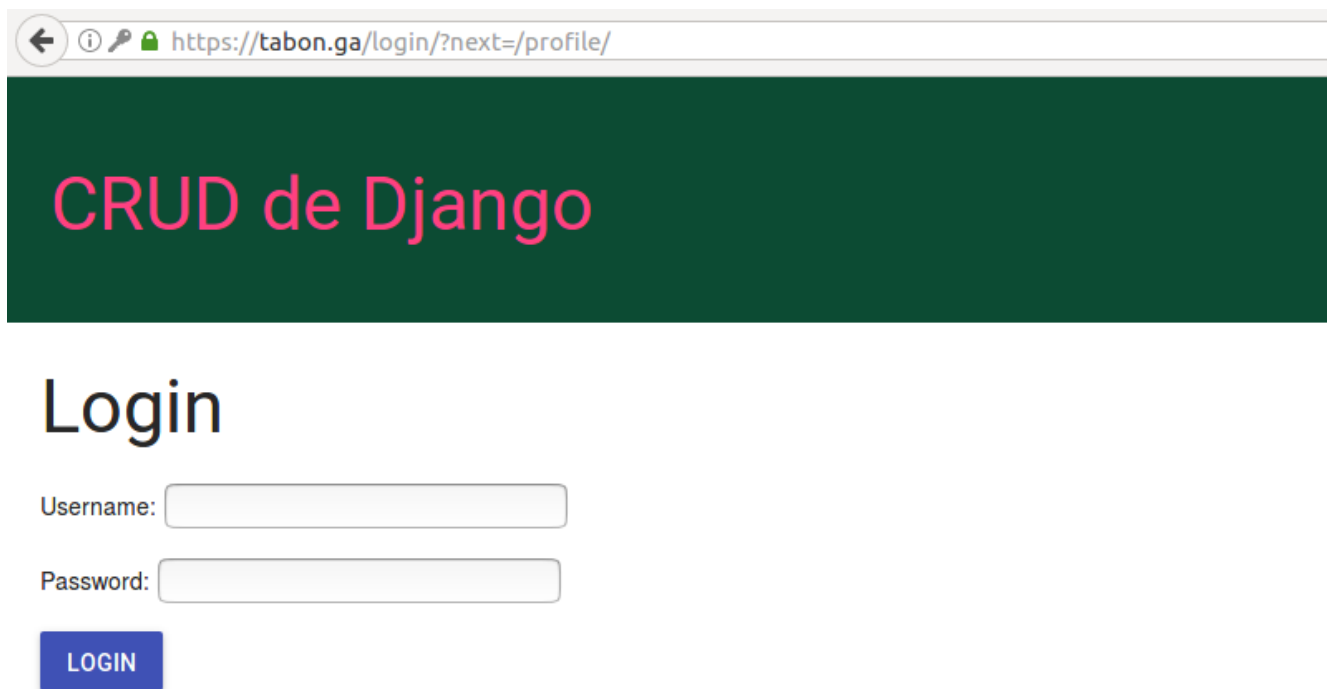
Figura 22: Página de inicio para usuarios sin sesión activa

Si no está registrado e ingresa en el enlace para registrarse, será redireccionado a un formulario para que llene sus datos y se de de alta en la aplicación.

The screenshot shows a web browser window with the address bar displaying `https://tabon.ga/home/crear`. The page has a dark green header with the text "CRUD de Django" in pink. Below the header, the main title "Crear Usuario" is displayed in a large, black, sans-serif font. The registration form consists of several input fields, each with a label above it: "Nombre de usuario" (with the value "Usuario"), "Nombre" (with the value "Nombre"), "Apellido" (with the value "App"), "Correo electrónico" (with the value "correo@mail.com"), "Password" (with four dots), and "Password confirmation" (with four dots). At the bottom of the form is a blue button labeled "ACEPTAR".

Figura 23: Formulario de registro para usuarios

Si ya está registrado, podrá ingresar a la aplicación siguiendo el enlace de *Iniciar sesión* en la página de inicio y en el formulario escribir el nombre de usuario y la contraseña con los que se registró.



The image shows a web browser window with the address bar displaying `https://tabon.ga/login/?next=/profile/`. The page has a dark green header with the text "CRUD de Django" in pink. Below the header, the word "Login" is displayed in a large, dark font. Underneath, there are two input fields: "Username:" and "Password:". Below the password field is a blue button with the text "LOGIN" in white capital letters.

Figura 24: Formulario de inicio de sesión

Una vez dentro de la aplicación, el usuario verá una lista de los usuarios registrados y tendrá las opciones de ver información más detallada sobre algún usuario o sí mismo, actualizar sus datos o eliminar su cuenta si así lo desea.



Usuarios

Usuario: Carlos

Nombre: Carlos Acosta

Correo electrónico: cgah.95@gmail.com



Usuario: palo

Nombre: Luis-Pablo Mayo

Correo electrónico: pmayov@ciencias.unam.mx



Usuario: And

Nombre: Andrea González

Correo electrónico: andreagonz@ciencias.unam.mx



Usuario: paulo

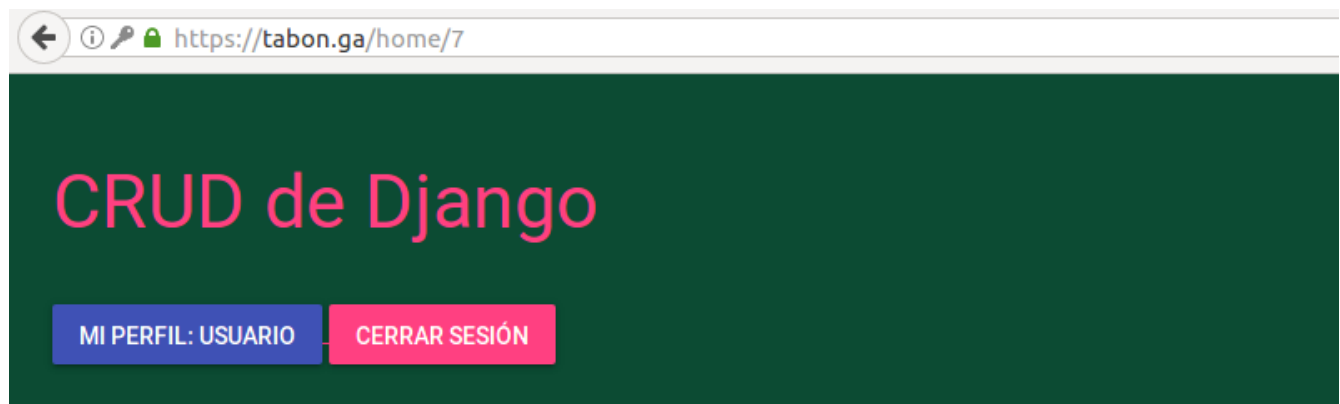
Nombre: Paulo Contreras

Correo electrónico: paulo@tabon.ga



Figura 25: Página de inicio para usuarios con sesión activa

Si elige ver información más detallada sobre un usuario, el enlace lo llevará al perfil del usuario.



Detalles de usuario

Usuario: Usuario

Nombre: Nombre App

Correo electrónico: correo@mail.com

Figura 26: Perfil de usuario

Para actualizar sus datos, será redirigido a un formulario similar al de registro.

← https://tabon.ga/home/editar/7

CRUD de Django

MI PERFIL: USUARIO CERRAR SESIÓN

Editar Usuario

Nombre de usuario
Usuario

Nombre
Nombre

Apellido
App

Correo electrónico
correo@mail.com

Contraseña nueva

Old password

ACEPTAR

Figura 27: Formulario de actualización de datos

Si quiere eliminar su cuenta se le enviará un mensaje para que confirme su acción.



Figura 28: Mensaje de confirmación de eliminación de un usuario

3.4. Gunicorn

Para poder comunicar la aplicación con el servidor web instalamos **gunicorn**, que es un servidor web ligero que utiliza la interfaz WSGI para poder comunicarse con **python**.

```
$ sudo apt-get install python3-pip
$ sudo pip3 install gunicorn
```

Creamos el archivo `/etc/systemd/system/gunicorn.service` para que **gunicorn** se ejecute con **systemd**.

```
[Unit]
Description=gunicorn daemon
After=network.target
```

```
[Service]
User=ubuntu
Group=www-data
```

```
WorkingDirectory=/home/ubuntu/djangocrud
ExecStart=/home/ubuntu/.local/bin/gunicorn --timeout 60 \
    --bind ec2-13-59-60-249.us-east-2.compute.amazonaws.com:8000 \
    djangocrud.wsgi:application
```

```
[Install]
WantedBy=multi-user.target
```

Así, para ejecutar el demonio de **gunicorn** basta utilizar el comando

```
$ systemctl start gunicorn
```

3.5. Reglas de firewall

Definimos las siguientes reglas para el **firewall** que provee **Amazon**, de manera que se restringe la comunicación únicamente con el servidor web y a través de **SSH** por el puerto 2200.

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	Description ⓘ
Custom TCP Rule	TCP	2200	0.0.0.0/0	
Custom TCP Rule	TCP	8000	18.216.66.221/32	DESDE EL SERVIDOR ...

Figura 29: Reglas de **firewall** para el servidor de aplicación

4. Servidor Web

En el servidor web instalamos **apache2** con `sudo apt-get install apache2`, de modo que funcione como servidor de proxy inverso y mande las peticiones que reciba al servidor de aplicación y a su vez reciba respuestas de éste para mandárselas al cliente.

4.1. Configuración de Apache

Para la configuración de **apache2** modificamos los siguientes archivos como se muestra:

```
/etc/apache2/sites-available/000-default.conf
```

```
<VirtualHost *:80>
    ProxyPreserveHost On

    # Redirección a HTTPS
    RewriteEngine On
```

```

RewriteCond %{HTTPS} !=on
RewriteRule ^/?(.*) https://%{SERVER_NAME}/$1 [R,L]
ServerName tabon.ga
</VirtualHost>

```

/etc/apache2/sites-available/default-ssl.conf

```

<IfModule mod_ssl.c>
<VirtualHost _default_:443>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    SSLEngine on
    ServerName tabon.ga

    # Certificados de letsencrypt
    Include /etc/letsencrypt/options-ssl-apache.conf
    Include /etc/letsencrypt/options-ssl-apache.conf
    ServerAlias www.tabon.ga
    Include /etc/letsencrypt/options-ssl-apache.conf
    SSLCertificateFile      /etc/letsencrypt/live/tabon.ga/cert.pem
    SSLCertificateKeyFile    /etc/letsencrypt/live/tabon.ga/privkey.pem
    SSLCertificateChainFile  /etc/letsencrypt/live/tabon.ga/chain.pem
    <FilesMatch "\.(cgi|shtml|phtml|php)$">
        SSLOptions +StdEnvVars
    </FilesMatch>
    <Directory /usr/lib/cgi-bin>
        SSLOptions +StdEnvVars
    </Directory>

    # HSTS
    Header always set Strict-Transport-Security: \
        "max-age=31536000; includeSubDomains"
    RequestHeader set X-Forwarded-Proto 'https' env=HTTPS
    Timeout 10000

    # Configuraciones del proxy inverso
    ProxyTimeout 10000
    ProxyBadHeader Ignore
    # Mandamos las peticiones del cliente al servidor de aplicación
    ProxyPass / http://13.59.60.249:8000/
    ProxyPassReverse / http://13.59.60.249:8000/

```

```

    </VirtualHost>
</IfModule>

/etc/apache2/conf-available/security.conf

# Se deshabilita la firma de Apache para que no se muestre
# el nombre y version del sistema operativo
ServerTokens Prod
ServerSignature Off
TraceEnable Off
Header set X-Content-Type-Options: "nosniff"

```

Habilitamos los siguientes módulos para hacer funcionar el proxy inverso y SSL.

```

$ sudo a2enmod proxy_http
$ sudo a2enmod proxy_ajp
$ sudo a2enmod rewrite
$ sudo a2enmod deflate
$ sudo a2enmod headers
$ sudo a2enmod proxy_balancer
$ sudo a2enmod proxy_connect
$ sudo a2enmod proxy_html
$ sudo a2enmod lbmethod_byrequests
$ sudo a2enmod ssl

```

Finalmente, para habilitar apache2 utilizamos

```
$ sudo systemctl start apache2
```

4.2. Reglas de firewall

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	Description ⓘ
HTTP	TCP	80	0.0.0.0/0	
HTTP	TCP	80	::/0	
Custom TCP Rule	TCP	2200	0.0.0.0/0	
Custom TCP Rule	TCP	8000	0.0.0.0/0	
Custom TCP Rule	TCP	8000	::/0	
HTTPS	TCP	443	0.0.0.0/0	
HTTPS	TCP	443	::/0	

Figura 30: Reglas de firewall para el servidor web

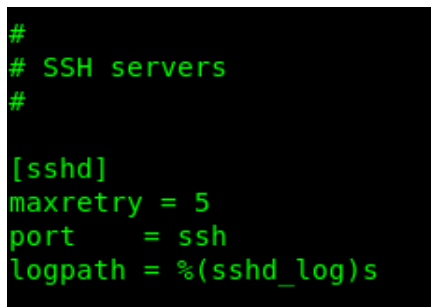
5. Aspectos de seguridad adicional

5.1. Fail2ban

Como los servidores tienen **Ubuntu** como sistema operativo, los archivos para la instalación de **fail2ban** están en los repositorios oficiales de la distribución, por lo que solo hizo falta ejecutar las instrucciones

```
# apt-get update
# apt-get install fail2ban
```

Después de haber instalado **fail2ban** en el servidor, por recomendación de los desarrolladores, creamos un archivo **jail.local** como copia del archivo **jail.conf** que está en el directorio **/etc/fail2ban/** creado durante la instalación. En este archivo, habilitamos las *jails* para restringir conexiones no autorizadas por fuerza bruta en los servicios de **ssh** y **apache** como se muestra a continuación:



```
#
# SSH servers
#

[sshd]
maxretry = 5
port     = ssh
logpath  = %(sshd_log)s
```

Figura 31: Configuración de jail para el servicio de ssh


```
#
# HTTP servers
#
[apache]
enabled = true
port = http,https
filter = apache-auth
logpath = %(apache_error_log)s
maxretry = 3
findtime = 600

[apache-auth]
enabled = true
port = http,https
maxretry = 6
findtime = 600
filter = apache-auth
logpath = %(apache_error_log)s

[apache-badbots]
# Ban hosts which agent identifies spammer robots crawling the web
# for email addresses. The mail outputs are buffered.
enabled = true
port = http,https
logpath = %(apache_access_log)s
bantime = 600
filter = apache-badbots
maxretry = 2

[apache-noscript]
enabled = true
port = http,https
logpath = %(apache_error_log)s
maxretry = 6

[apache-overflows]
enabled = true
port = http,https
filter = apache-overflows
```

Figura 32: configuración de jail para Apache

Para probar el funcionamiento de **fail2ban** se hicieron pruebas con distintas herramientas, como los *benchmarks* de Apache, y verificamos el estado de las **iptables** antes y después de tales pruebas.

```
ubuntu@ip-172-31-21-60:~$ sudo iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-N f2b-apache
-N f2b-apache-auth
-N f2b-apache-badbots
-N f2b-apache-nohome
-N f2b-apache-noscript
-N f2b-apache-overflows
-N f2b-http-get-dos
-N f2b-sshd
-A INPUT -p tcp -m multiport --dports 80,443 -j f2b-http-get-dos
-A INPUT -p tcp -m multiport --dports 80,443 -j f2b-apache
-A INPUT -p tcp -m multiport --dports 80,443 -j f2b-apache-nohome
-A INPUT -p tcp -m multiport --dports 80,443 -j f2b-apache-overflows
-A INPUT -p tcp -m multiport --dports 80,443 -j f2b-apache-noscript
-A INPUT -p tcp -m multiport --dports 80,443 -j f2b-apache-badbots
-A INPUT -p tcp -m multiport --dports 80,443 -j f2b-apache-auth
-A INPUT -p tcp -m multiport --dports 22 -j f2b-sshd
-A f2b-apache -j RETURN
-A f2b-apache-auth -j RETURN
-A f2b-apache-badbots -j RETURN
-A f2b-apache-nohome -j RETURN
-A f2b-apache-noscript -j RETURN
-A f2b-apache-overflows -j RETURN
-A f2b-http-get-dos -j RETURN
-A f2b-sshd -j RETURN
```

Figura 33: estado de iptables después de configurar fail2ban

```

Status for the jail: sshd
|- Filter
|   |- Currently failed: 1
|   |- Total failed:    6
|   `-- File list:      /var/log/auth.log
`- Actions
    |- Currently banned: 1
    |- Total banned:    1
    `-- Banned IP list: 200.68.129.145

```

Figura 34: resultado de las pruebas para ssh

```

Status for the jail: http-get-dos
|- Filter
|   |- Currently failed: 2
|   |- Total failed:    451
|   `-- File list:      /var/log/apache2/access.log
`- Actions
    |- Currently banned: 0
    |- Total banned:    1
    `-- Banned IP list:

```

Figura 35: resultado de una prueba de ataque DDoS

```

ubuntu@ip-172-31-21-60:~$ sudo iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-N f2b-apache
-N f2b-apache-auth
-N f2b-apache-badbots
-N f2b-apache-nohome
-N f2b-apache-noscript
-N f2b-apache-overflows
-N f2b-sshd
-A INPUT -p tcp -m multiport --dports 80,443 -j f2b-apache
-A INPUT -p tcp -m multiport --dports 80,443 -j f2b-apache-nohome
-A INPUT -p tcp -m multiport --dports 80,443 -j f2b-apache-overflows
-A INPUT -p tcp -m multiport --dports 80,443 -j f2b-apache-noscript
-A INPUT -p tcp -m multiport --dports 80,443 -j f2b-apache-badbots
-A INPUT -p tcp -m multiport --dports 80,443 -j f2b-apache-auth
-A INPUT -p tcp -m multiport --dports 22 -j f2b-sshd
-A f2b-apache -j RETURN
-A f2b-apache-auth -j RETURN
-A f2b-apache-badbots -j RETURN
-A f2b-apache-nohome -j RETURN
-A f2b-apache-noscript -j RETURN
-A f2b-apache-overflows -j RETURN
-A f2b-sshd -s 200.68.129.145/32 -j REJECT --reject-with icmp-port-unreachable
-A f2b-sshd -j RETURN

```

Figura 36: estado de iptables después de las pruebas

5.2. Letsencrypt

Se instaló el bot de letsencrypt en el servidor de correos y en el servidor web.

```

$ sudo apt-get install software-properties-common
$ sudo add-apt-repository ppa:certbot/certbot
$ sudo apt-get update
$ sudo apt-get install python-certbot-apache

```

Para cada servidor se corrió con `sudo certbot --apache certonly`, al terminar creó la llave y certificados.

Ejemplo en el servidor de correos:

```
ubuntu@ip-172-31-27-207:~$ sudo certbot --apache certonly
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator apache, Installer apache
Enter email address (used for urgent renewal and security notices) (Enter 'c' to
cancel): carlos-acosta@ciencias.unam.mx

-----
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.1.1-August-1-2016.pdf. You must agree
in order to register with the ACME server at
https://acme-v01.api.letsencrypt.org/directory
-----
(A)gree/(C)ancel: a

-----
Would you be willing to share your email address with the Electronic Frontier
Foundation, a founding partner of the Let's Encrypt project and the non-profit
organization that develops Certbot? We'd like to send you email about EFF and
our work to encrypt the web, protect its users and defend digital rights.
-----
(Y)es/(N)o: n

Which names would you like to activate HTTPS for?
-----
1: mail.tabon.ga
2: www.mail.tabon.ga
-----
Select the appropriate numbers separated by commas and/or spaces, or leave input
blank to select all options shown (Enter 'c' to cancel): 1 2
Obtaining a new certificate
Performing the following challenges:
tls-sni-01 challenge for mail.tabon.ga
tls-sni-01 challenge for www.mail.tabon.ga
Enabled Apache ssl module
Waiting for verification...
Cleaning up challenges

IMPORTANT NOTES:
- Congratulations! Your certificate and chain have been saved at:
  /etc/letsencrypt/live/mail.tabon.ga/fullchain.pem
  Your key file has been saved at:
  /etc/letsencrypt/live/mail.tabon.ga/privkey.pem
  Your cert will expire on 2018-02-04. To obtain a new or tweaked
  version of this certificate in the future, simply run certbot
  again. To non-interactive renew "all" of your certificates, run
  "certbot renew"
- Your account credentials have been saved in your Certbot
  configuration directory at /etc/letsencrypt. You should make a
  secure backup of this folder now. This configuration directory will
  also contain certificates and private keys obtained by Certbot so
  making regular backups of this folder is ideal.
- If you like Certbot, please consider supporting our work by:

Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
Donating to EFF: https://eff.org/donate-le
```

5.3. Reasignación de puerto de SSH

A cada servidor le cambiamos el puerto del servicio SSH que viene por defecto (el 22) por el puerto 2200. Para realizar esto sólo bastó modificar la línea

Port 22

por

Port 2200

en el archivo de configuración `/etc/ssh/sshd_config`.

5.4. Seguridad de la base de datos

Para garantizar que las conexiones entre cliente y servidor de la aplicación necesitamos que las conexiones con la base de datos también sean seguras. En MySQL están disponibles las conexiones cifradas usando el protocolo TLS (Transport Layer Security), el cual se asegura de que los datos que se reciban en una red pública sean confiables, además de contar con mecanismos para detectar

cambios en los datos, así como pérdida o repetición de la información y validación de identidades.

Como primer paso, debemos asegurarnos de instalar MySQL de manera segura, tal como lo hicimos en la práctica 3 del curso.

```
There are three levels of password validation policy:
LOW      Length >= 8
MEDIUM  Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary      file

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 2
Using existing password for root.

Estimated strength of the password: 50
Change the password for root ? ((Press y|Y for Yes, any other key for No) : No

... skipping.
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.

Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.

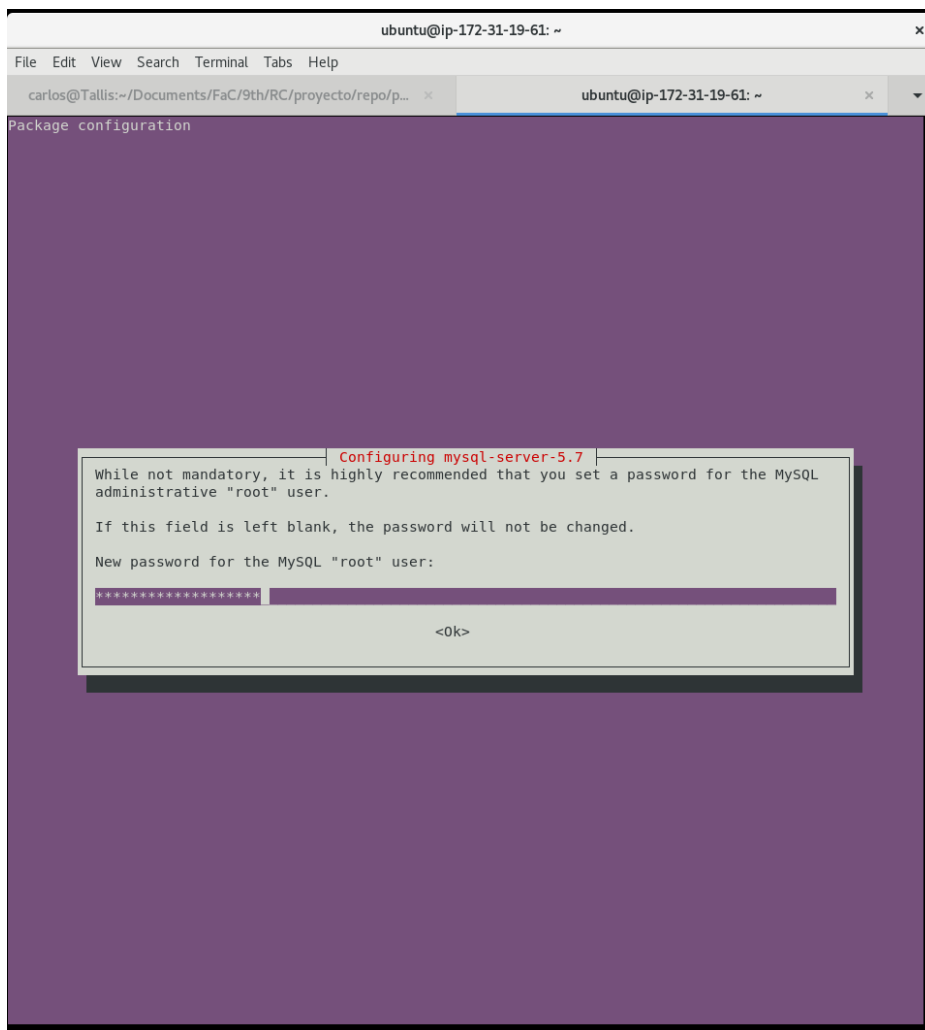
Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y
- Dropping test database...
Success.

- Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
ubuntu@ip-172-31-19-61:~$
```



Para utilizar el cifrado de TLS en las conexiones de nuestra base da datos, primero hay que crear un directorio donde alojaremos las llaves y certificados requeridos para el cifrado. En este caso, elegimos tenerlas dentro del directorio de configuraciones `/etc/mysql`, pero podrían estar en cualquier otro.

```
$ cd /etc/mysql
$ sudo mkdir ssl
$ cd ssl
```

Ahora creamos una llave para la CA (Autoridad Certificadora), de manera similar a la práctica 2 del curso, usamos `openssl` pues ya está integrado en Ubuntu por defecto.

```

ubuntu@ip-172-31-19-61:~/cert$ openssl genrsa 2048 > ca-key.pem
Generating RSA private key, 2048 bit long modulus
.....+++
.....+++
e is 65537 (0x10001)
ubuntu@ip-172-31-19-61:~/cert$ openssl req -sha1 -new -x509 -nodes -days 3650 -key ca-key.pem > ca-cert.pem
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]:MX
State or Province Name (full name) [Some-State]:CDMX
Locality Name (eg, city) []:CDMX
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Kihui-Dev
Organizational Unit Name (eg, section) []:kihui
Common Name (e.g. server FQDN or YOUR name) []:cgah.95@gmail.com
Email Address []:cgah.95@gmail.com

```

Ahora creamos una llave para el certificado del servidor y proporcionamos la información que se nos solicite.

```

ubuntu@ip-172-31-19-61:~/cert$ openssl req -sha1 -newkey rsa:2048 -days 730 -nodes -keyout server-key.pem > server-req.pem
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'server-key.pem'
-----
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]:MX
State or Province Name (full name) [Some-State]:CDMX
Locality Name (eg, city) []:CDMX
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Kihui-Dev
Organizational Unit Name (eg, section) []:kihui
Common Name (e.g. server FQDN or YOUR name) []:cgah.95@gmail.com
Email Address []:cgah.95@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:telapelastejajaja
An optional company name []:kihui

```

Ya que tenemos tanto la llave de la CA como la del certificado del servidor, podemos firmar el certificado de acuerdo al formato establecido en el estándar **X.509**.

```

ubuntu@ip-172-31-19-61:~/cert$ openssl rsa -in server-key.pem -out server-key.pem
writing RSA key
ubuntu@ip-172-31-19-61:~/cert$ openssl x509 -sha1 -req -in server-req.pem -days 730 -CA ca-cert.pem
-CAkey ca-key.pem -set_serial 01 > server-cert.pem
Signature ok
subject=C=MX/ST=CDMX/L=CDMX/O=Kihui-Dev/OU=kihui/CN=cgah.95@gmail.com/emailAddress=cgah.95@gmail.co
m
Getting CA Private Key

```

Ya que contamos con los certificados, debemos configurar MySQL para que cifre sus conexiones con ellos, pues si ingresamos ahora a la shell de MySQL y escribimos el comando `SHOW VARIABLES LIKE '%ssl%'`; nos indicará que SSL no está habilitado.

```

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.7.20-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2017, 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 GLOBAL VARIABLES LIKE 'have_%ssl';
+-----+-----+
| Variable_name | Value   |
+-----+-----+
| have_openssl  | DISABLED |
| have_ssl      | DISABLED |
+-----+-----+
2 rows in set (0.01 sec)

mysql> \q
Bye

```

Para habilitar SSL debemos agregar las rutas de las llaves y certificados que acabamos de crear en las variables correspondientes en el archivo de configuración de MySQL, que generalmente está en `/etc/mysql/my.cnf` o un archivo de nombre similar.

```

#
log_error = /var/log/mysql/error.log
#
# Here you can see queries with especially long duration
#log_slow_queries      = /var/log/mysql/mysql-slow.log
#long_query_time = 2
#log-queries-not-using-indexes
#
# The following can be used as easy to replay backup logs or for replication.
# note: if you are setting up a replication slave, see README.Debian about
#       other settings you may need to change.
#server-id             = 1
#log_bin               = /var/log/mysql/mysql-bin.log
expire_logs_days      = 10
max_binlog_size       = 100M
#binlog_do_db          = include_database_name
#binlog_ignore_db      = include_database_name
#
# * InnoDB
#
# InnoDB is enabled by default with a 10MB datafile in /var/lib/mysql/.
# Read the manual for more InnoDB related options. There are many!
#
# * Security Features
#
# Read the manual, too, if you want chroot!
# chroot = /var/lib/mysql/
#
# For generating SSL certificates I recommend the OpenSSL GUI "tinyca".
#
#ssl-ca=/etc/mysql/ssl/ca-cert.pem
#ssl-cert=/etc/mysql/ssl/server-cert.pem
[ ssl-key=/etc/mysql/ssl/server-key.pem

```

105,1

Bot

Si ahora ingresamos a la shell de MySQL y volvemos a escribir el comando `SHOW VARIABLES LIKE '%ssl%'`; nos debe afirmar que SSL está habilitado, entonces las conexiones con la base de

datos ya serán cifradas y tendremos mayor seguridad en la comunicación con ésta.

```
Welcome to the MySQL monitor.
Your MySQL connection id is 1
Server version: 5.7.20-0ubuntu0.16.04

Copyright (c) 2000, 2017, Oracle and/or its affiliates. Other names may be
owners.

Type 'help;' or '\h' for help.

mysql> SHOW GLOBAL VARIABLES;
+-----+-----+
| Variable_name | Value |
+-----+-----+
| have_openssl  | YES   |
| have_ssl      | YES   |
+-----+-----+
2 rows in set (0.00 sec)
```

6. Servidor de correos

Utilizamos una máquina virtual con Ubuntu 16.04. Registramos el nombre de dominio `mail.tabon.ga` para su dirección IP correspondiente, que en este caso es la `13.58.30.114`.

Para el funcionamiento del servidor de correos utilizamos los paquetes de `postfix`, `dovecot`, `squirrel-mail` y `apache2`. Tales paquetes se instalaron de la siguiente manera:

```
$ sudo apt install postfix
$ sudo apt install dovecot-core dovecot-imapd
$ sudo apt install apache2
```

6.1. Configuración de Postfix

En el archivo `/etc/postfix/main.cf` se configuró lo siguiente:

```
smtpd_banner = $myhostname ESMTP $mail_name
biff = no
append_dot_mydomain = no
readme_directory = no
smtpd_use_tls = yes
smtpd_tls_key_file = /etc/letsencrypt/live/mail.tabon.ga/privkey.pem
smtpd_tls_cert_file = /etc/letsencrypt/live/mail.tabon.ga/fullchain.pem
```

```

smtpd_tls_loglevel = 3
smtpd_tls_received_header = yes
smtpd_tls_session_cache_timeout = 3600s
tls_random_source = dev:/dev/urandom
smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated \
                           defer_unauth_destination
myhostname = mail.tabon.ga
mydomain = tabon.ga
myorigin = /etc/mailname
mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
relayhost =
mynetworks = 127.0.0.0/8, $mydomain, 132.247.0.0/16, 132.248.0.0/16
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all
home_mailbox = mail/
smtpd_sasl_type = dovecot

```

Se especificó donde se encuentra el certificado y llaves generados por `letsencrypt`, se modificó el banner de `smtp` y se pusieron restricciones de relay.

En el archivo `/etc/postfix/master.cf` se agregó lo siguiente:

```

smtp      inet  n       -       y       -       -       smtpd
submission inet n       -       y       -       -       smtpd
  -o smtpd_tls_security_level=encrypt
smtps     inet  n       -       y       -       -       smtpd
  -o smtpd_tls_wrappermode=yes

```

de manera que se habilitara `smtps` y se abriera el puerto correspondiente (el 465). Finalmente se reinició el servicio con

```
$ sudo systemctl restart postfix
```

6.2. Configuración de Dovecot

Se modificó el archivo `/etc/dovecot/dovecot.conf` con la siguiente línea:

```
listen = *, ::
```

En `/etc/dovecot/conf.d/10-auth.conf` se agregó:

```

disable_plaintext_auth = yes
auth_mechanisms = plain login

```

En `/etc/dovecot/conf.d/10-mail.conf`:

```
mail_location = maildir:~/mail
```

Y en /etc/dovecot/conf.d/10-ssl.conf

```
ssl = yes
ssl_cert = </etc/letsencrypt/live/mail.tabon.ga/fullchain.pem
ssl_key = </etc/letsencrypt/live/mail.tabon.ga/privkey.pem
```

6.3. Configuración de Squirrelmail

Se corrió el bot de configuración de Squirrelmail:

```
$ sudo squirrelmail-configure
```

Por lo que se mostró el menú de configuración:

```
SquirrelMail Configuration : Read: config.php (1.4.0)
```

```
-----
Main Menu --
```

1. Organization Preferences
2. Server Settings
3. Folder Defaults
4. General Options
5. Themes
6. Address Books
7. Message of the Day (MOTD)
8. Plugins
9. Database
10. Languages

D. Set pre-defined settings for specific IMAP servers

C Turn color on

S Save data

Q Quit

Command >>

Se seleccionó la opción 2. Se procedió a configurar las opciones de SMTP e IMAP, que quedaron de la siguiente manera:

```
SquirrelMail Configuration : Read: config.php (1.4.0)
```

```
-----
Server Settings
```

General

```
-----
```

1. Domain : tabon.ga
2. Invert Time : false
3. Sendmail or SMTP : SMTP

IMAP Settings

4. IMAP Server : mail.tabon.ga
5. IMAP Port : 993
6. Authentication type : login
7. Secure IMAP (TLS) : true
8. Server software : dovecot
9. Delimiter : /

SMTP Settings

4. SMTP Server : mail.tabon.ga
5. SMTP Port : 465
6. POP before SMTP : false
7. SMTP Authentication : none
8. Secure SMTP (TLS) : true
9. Header encryption key :

6.4. Configuración de Apache

Se habilitaron los modulos necesarios para que **https** funcione

```
$ sudo a2enmod ssl
$ sudo a2enmod headers
$ sudo a2enmod rewrite
```

Se copió el archivo de squirrelmail de apache a los sitios disponibles de apache, se habilitó junto con **default-ssl** y se deshabilitó el sitio por defecto de Apache.

```
$ sudo cp /etc/squirrelmail/apache.conf \
    /etc/apache2/sites-available/squirrelmail.conf
$ sudo a2ensite squirrelmail.conf
$ sudo a2ensite default-ssl
$ sudo a2dissite 000-default.conf
```

En el archivo `/etc/apache2/conf-enabled/security.conf` se modificó la firma de Apache para que no muestre su versión y el sistema operativo en uso.

```
ServerTokens Prod
ServerSignature Off
```

El archivo `/etc/apache2/sites-available/squirrelmail.conf` quedó de la siguiente manera

```
Alias /mail /usr/share/squirrelmail
<Directory /usr/share/squirrelmail>
    Options FollowSymLinks
    <IfModule mod_php.c>
        php_flag register_globals off
    </IfModule>
    <IfModule mod_dir.c>
        DirectoryIndex index.php
    </IfModule>
    <Files configtest.php>
        order deny,allow
        deny from all
        allow from 127.0.0.1
    </Files>
</Directory>

RedirectMatch ^/$ /mail/
<IfModule mod_rewrite.c>
    <IfModule mod_ssl.c>
        <Location /squirrelmail>
            RewriteEngine on
            RewriteCond %{HTTPS} !^on$ [NC]
            RewriteRule . https://%{HTTP_HOST}%{REQUEST_URI} [L]
        </Location>
    </IfModule>
</IfModule>
```

El archivo `/etc/apache2/sites-available/default-ssl.conf` quedó de la siguiente manera:

```
<IfModule mod_ssl.c>
    <VirtualHost _default_:443>
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
        SSLEngine on
        SSLCertificateFile \
            /etc/letsencrypt/live/mail.tabon.ga/fullchain.pem
        SSLCertificateKeyFile \
            /etc/letsencrypt/live/mail.tabon.ga/privkey.pem
        <FilesMatch "\.(cgi|shtml|phtml|php)$">
```

```

        SSLOptions +StdEnvVars
</FilesMatch>
<Directory /usr/lib/cgi-bin>
    SSLOptions +StdEnvVars
</Directory>
ServerName mail.tabon.ga
ServerAlias www.mail.tabon.ga
Header always set Strict-Transport-Security \
    "max-age=31536000; includeSubDomains; preload"

</VirtualHost>
</IfModule>

```

Al finalizar estas configuraciones se reiniciaron los servicios con `systemctl`.

Pudimos ver entonces la página de inicio de Squirrelmail al ingresar a `mail.tabon.ga`:



Se agregaron los registros DNS de tipo MX y SPF para el dominio `tabon.ga` que son necesarios para el correcto funcionamiento del sistema, de manera que se acepte el correo que se manda al servidor asociado a tal dominio y se mande al servidor de correos, y que se evite que los correos mandados desde éste se vayan a spam.

```

chepe@chepe:~$ dig tabon.ga mx | egrep -v ";|^ *$"
tabon.ga.      144      IN       MX       10 mail.tabon.ga.
chepe@chepe:~$ dig tabon.ga spf | egrep -v ";|^ *$"
tabon.ga.      212      IN       SPF      "v=spf1 mx ~all"
chepe@chepe:~$

```

6.5. Reglas de firewall

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ
HTTP	TCP	80	0.0.0.0/0
HTTP	TCP	80	:/0
Custom TCP Rule	TCP	2200	0.0.0.0/0
Custom TCP Rule	TCP	2200	:/0
IMAPS	TCP	993	0.0.0.0/0
IMAPS	TCP	993	:/0
SMTP	TCP	25	0.0.0.0/0
SMTP	TCP	25	:/0
SMTPS	TCP	465	0.0.0.0/0
SMTPS	TCP	465	:/0
Custom TCP Rule	TCP	587	0.0.0.0/0
Custom TCP Rule	TCP	587	:/0
HTTPS	TCP	443	0.0.0.0/0
HTTPS	TCP	443	:/0

Comentarios sobre el desarrollo del proyecto

Intentamos hacer una aplicación lo más sencilla posible, minimizando la carga de trabajo que podría tener el servidor de aplicación, para enfocarnos en los aspectos más importantes que requería el proyecto: la comunicación entre los servidores y la seguridad en esta comunicación.