

1. Creamos variables de entorno para no tener que escribirlas

```
user@serverlempjorge:~$ export DOMAIN="jorgegarciasendra.duckdns.org"
user@serverlempjorge:~$ export WEBROOT="/var/www/miweb"
```

2. Creamos el subdominio en duckDNS

The screenshot shows a web-based interface for managing domains. At the top, there's a navigation bar with 'domains' (1/6), a search bar ('http:// sub domain duckdns.org add domain'), and a 'domains' button. Below this is a table with columns: 'domain', 'current ip', 'ipv6', and 'changed'. A single row is shown for 'jorgegarciasendra' with '90.167.51.35' in the ip column and '1 day ago' in the changed column. There are buttons for 'update ip' and 'update ipv6'. A note at the bottom states: 'This site is protected by reCAPTCHA and the Google Privacy Policy and Terms of Service apply.'

3. Creamos la carpeta donde se aloja la web y el contenido de la pagina principal

```
user@serverlempjorge:~$ sudo mkdir -p $WEBROOT
user@serverlempjorge:~$ ls /var/www/
html miweb
user@serverlempjorge:~$ echo "<h1>HTTPS con Let's Encrypt (DNS-01) en NGINX Lemp</h1>" | sudo tee $WEBROOT/index.html
<h1>HTTPS con Let's Encrypt (DNS-01) en NGINX Lemp</h1>
user@serverlempjorge:~$ cat /var/www/miweb/index.html
<h1>HTTPS con Let's Encrypt (DNS-01) en NGINX Lemp</h1>
user@serverlempjorge:~$
```

4. Creamos el ServerBlock, en servername ponemos el dominio del duckdns

The screenshot shows a terminal window with the nano editor open. The file is named 'jorge.conf'. The content of the file is an NGINX configuration block:

```
server {
    listen 80;
    server_name jorgegarciasendra.duckdns.org;

    root /var/www/miweb;
    index index.html index.php;

    access_log /var/log/nginx/miweb_access.log;
    error_log /var/log/nginx/miweb_error.log;

    location / {
        try_files $uri $uri/ /index.html;
    }
}
```

5. Activamos el sitio y recargamos el Nginx

```
user@serverlempjorge:/etc/nginx/sites-available$ ls
default jorge.conf
user@serverlempjorge:/etc/nginx/sites-available$ sudo ln -s /etc/nginx/sites-available/jorge.conf /etc/nginx/sites-enabled/
user@serverlempjorge:/etc/nginx/sites-available$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
user@serverlempjorge:/etc/nginx/sites-available$ sudo systemctl reload nginx
user@serverlempjorge:/etc/nginx/sites-available$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
    Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset>
    Active: active (running) since Fri 2025-11-14 08:57:14 UTC; 33min ago
      Docs: man:nginx(8)
   Process: 6655 ExecReload=/usr/sbin/nginx -g daemon on; master_process o>
 Main PID: 805 (nginx)
    Tasks: 2 (limit: 2265)
   Memory: 3.3M (peak: 5.6M)
     CPU: 135ms
    CGroup: /system.slice/nginx.service
            └─ 805 "nginx: master process /usr/sbin/nginx -g daemon on; ma>
              ├─ 6657 "nginx: worker process"

nov 14 08:57:13 serverlempjorge systemd[1]: Starting nginx.service - A high>
nov 14 08:57:14 serverlempjorge systemd[1]: Started nginx.service - A high >
nov 14 09:30:10 serverlempjorge systemd[1]: Reloading nginx.service - A hig>
nov 14 09:30:10 serverlempjorge nginx[6655]: 2025/11/14 09:30:10 [notice] 6>
nov 14 09:30:10 serverlempjorge systemd[1]: Reloaded nginx.service - A high>
lines 1-18/18 (END)
```

6. Instalamos el pip, y el certbot con la extensión para duck-dns, actualmente es la manera mas facil de hacerlo y la mas rapida.

Instalación pip

```
user@serverlempjorge:~$ sudo apt install -y python3-pip
[sudo] password for user:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential bzip2
  cpp cpp-13 cpp-13-x86-64-linux-gnu cpp-x86-64-linux-gnu dpkg-dev
  fakeroot g++ g++-13 g++-13-x86-64-linux-gnu g++-x86-64-linux-gnu gcc
```

Instalacion certbot con extensión para duck dns.

```
user@serverlempjorge:~$ sudo pip3 install certbot-dns-duckdns --break-system-packages
Collecting certbot-dns-duckdns
  Downloading certbot_dns_duckdns-1.7.0-py3-none-any.whl.metadata (17 kB)
Collecting certbot<6.0,>=1.18.0 (from certbot-dns-duckdns)
  Downloading certbot-5.1.0-py3-none-any.whl.metadata (7.0 kB)
Requirement already satisfied: requests<3.0,>=2.20.0 in /usr/lib/python3/dist-packages (from certbot-dns-duckdns) (2.31.0)
Collecting dnspython<3.0,>=2.0.0 (from certbot-dns-duckdns)
  Downloading dnspython-2.8.0-py3-none-any.whl.metadata (5.7 kB)
Collecting certbot-dns-duckdns
  Downloading certbot-dns-duckdns-1.7.0-py3-none-any.whl (16.9 kB)
```

7. Creación de certificados para securizar la pagina web

Creamos la carpeta donde se guardará toda la información de los certificados

```
user@serverlempjorge:~$ sudo mkdir -p /etc/letsencrypt
user@serverlempjorge:~$ |
```

Creamos el archivo de configuración con el token de autenticación de DuckDNS y establecemos los permisos necesarios

```
user@serverlempjorge:~$ echo "dns_duckdns_token=736ebd7a-1f3e-409d-9b4e-7b1ae44a6c5c" | sudo tee /etc/letsencrypt/duckdns.ini
dns_duckdns_token=736ebd7a-1f3e-409d-9b4e-7b1ae44a6c5c
user@serverlempjorge:~$ sudo chmod 600 /etc/letsencrypt/duckdns.ini
user@serverlempjorge:~$ |
```

Ejecutamos certbot para obtener un certificado SSL valido

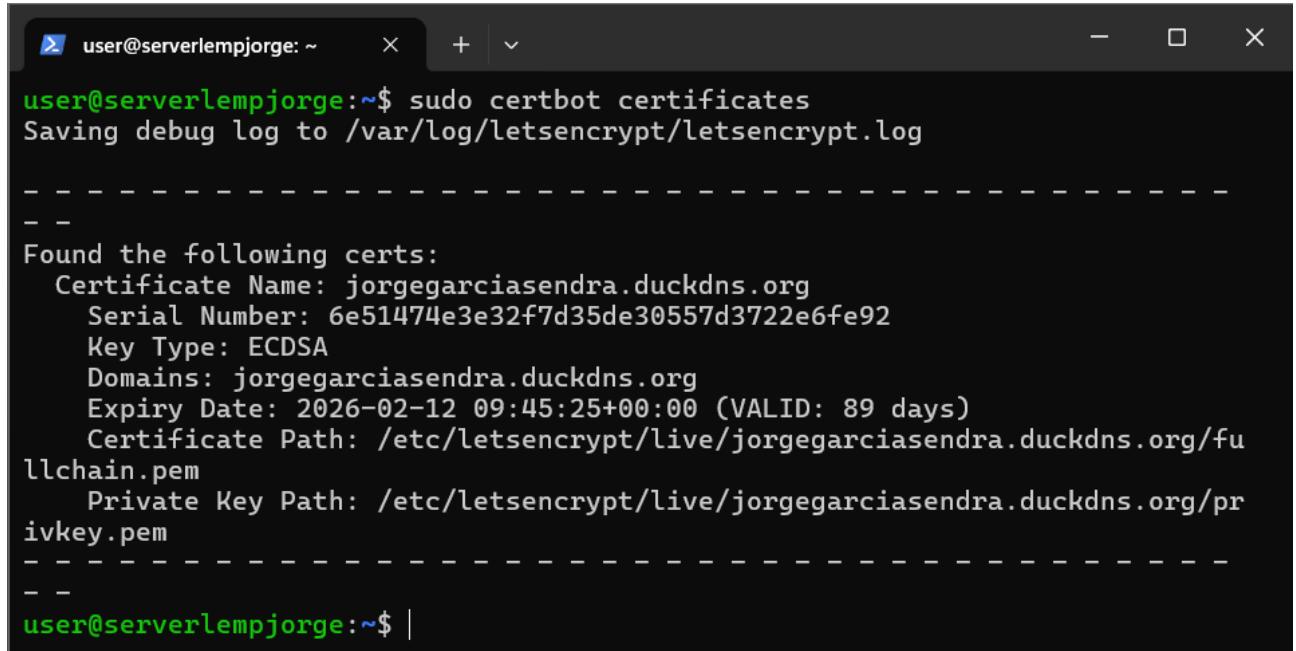
```
user@serverlempjorge:~$ sudo certbot certonly --authenticator dns-duckdns --dns-duckdns-credentials /etc/letsencrypt/duckdns.ini --dns-duckdns-propagation-seconds 60 -d jorgegarciasendra.duckdns.org
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Requesting a certificate for jorgegarciasendra.duckdns.org
Waiting 60 seconds for DNS changes to propagate

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/jorgegarciasendra.duckdns.org/fullchain.pem
Key is saved at:          /etc/letsencrypt/live/jorgegarciasendra.duckdns.org/privkey.pem
This certificate expires on 2026-02-12.
These files will be updated when the certificate renews.

NEXT STEPS:
- The certificate will need to be renewed before it expires. Certbot can automatically renew the certificate in the background, but you may need to take steps to enable that functionality. See https://certbot.org/renewal-setup for instructions.

-----
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
-----
```

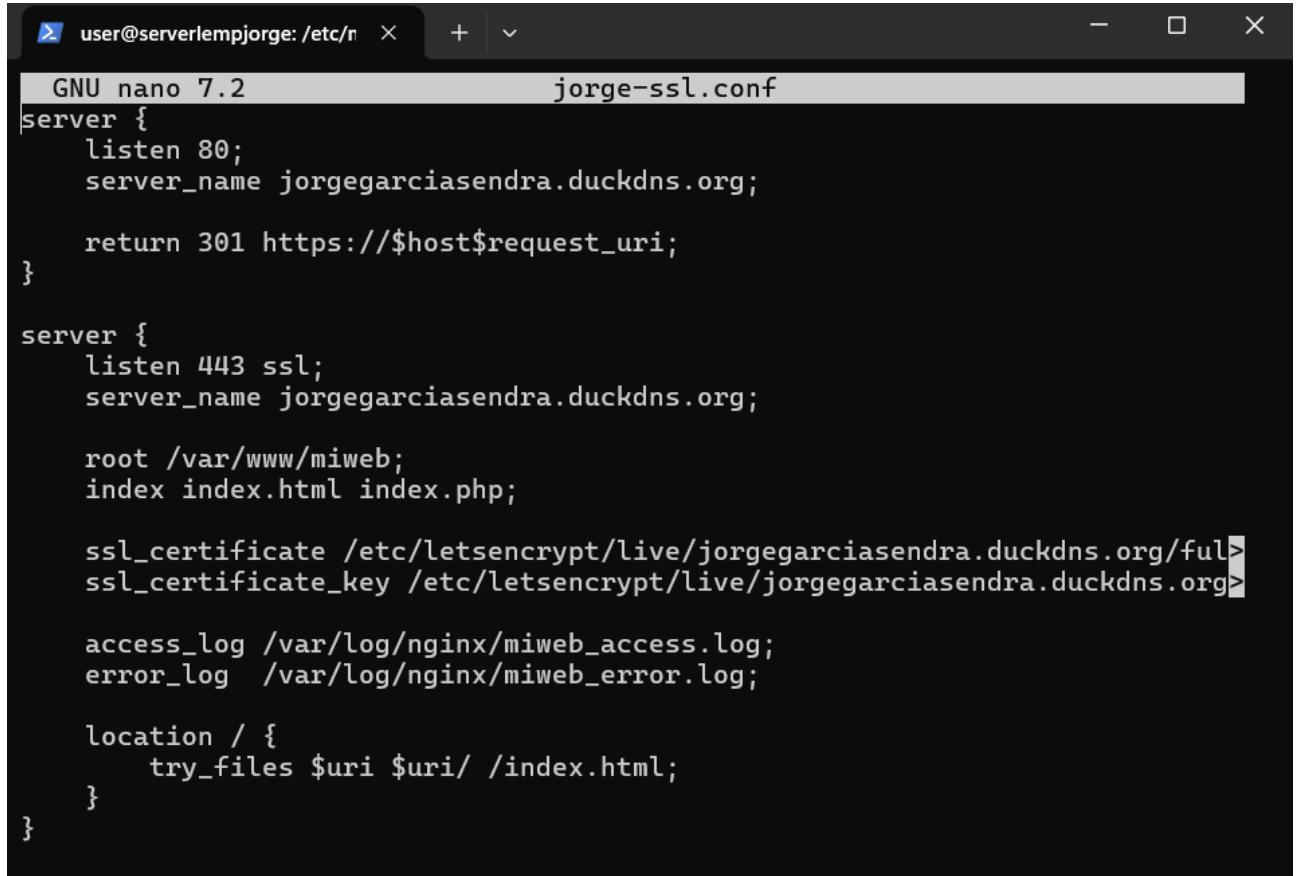
Comprobamos que el certificado se ha creado correctamente



```
user@serverlempjorge:~$ sudo certbot certificates
Saving debug log to /var/log/letsencrypt/letsencrypt.log

- - -
Found the following certs:
Certificate Name: jorgegarciasendra.duckdns.org
  Serial Number: 6e51474e3e32f7d35de30557d3722e6fe92
  Key Type: ECDSA
  Domains: jorgegarciasendra.duckdns.org
  Expiry Date: 2026-02-12 09:45:25+00:00 (VALID: 89 days)
  Certificate Path: /etc/letsencrypt/live/jorgegarciasendra.duckdns.org/fullchain.pem
  Private Key Path: /etc/letsencrypt/live/jorgegarciasendra.duckdns.org/privkey.pem
- - -
user@serverlempjorge:~$ |
```

8. Creamos el serverblock del HTTPS



```
GNU nano 7.2                               jorge-ssl.conf
server {
    listen 80;
    server_name jorgegarciasendra.duckdns.org;

    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl;
    server_name jorgegarciasendra.duckdns.org;

    root /var/www/miweb;
    index index.html index.php;

    ssl_certificate /etc/letsencrypt/live/jorgegarciasendra.duckdns.org/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/jorgegarciasendra.duckdns.org/privkey.pem;

    access_log /var/log/nginx/miweb_access.log;
    error_log  /var/log/nginx/miweb_error.log;

    location / {
        try_files $uri $uri/ /index.html;
    }
}
```

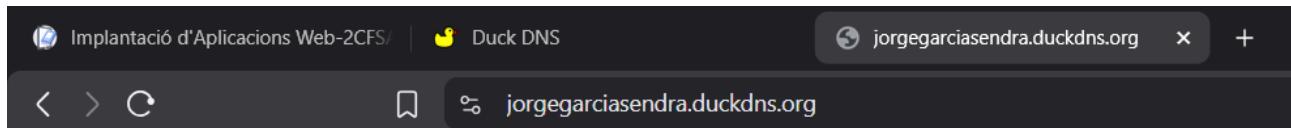
Activamos el sitio y hacemos un reload

```
user@serverlempjorge:/etc/nginx/sites-available$ sudo ln -s /etc/nginx/sites-available/jorge-ssl.conf /etc/nginx/sites-enabled/
user@serverlempjorge:/etc/nginx/sites-available$ |
```

```
user@serverlempjorge:/etc/nginx/sites-available$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
user@serverlempjorge:/etc/nginx/sites-available$ sudo systemctl reload nginx
user@serverlempjorge:/etc/nginx/sites-available$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset>
   Active: active (running) since Fri 2025-11-14 08:57:14 UTC; 1h 59min a>
     Docs: man:nginx(8)
   Process: 7982 ExecReload=/usr/sbin/nginx -g daemon on; master_process o>
   Main PID: 805 (nginx)
      Tasks: 2 (limit: 2265)
     Memory: 3.6M (peak: 6.1M)
       CPU: 202ms
      CGroup: /system.slice/nginx.service
              └─ 805 "nginx: master process /usr/sbin/nginx -g daemon on; ma>
                ├─ 7983 "nginx: worker process"

nov 14 08:57:13 serverlempjorge systemd[1]: Starting nginx.service - A high>
nov 14 08:57:14 serverlempjorge systemd[1]: Started nginx.service - A high >
nov 14 09:30:10 serverlempjorge systemd[1]: Reloading nginx.service - A hig>
nov 14 09:30:10 serverlempjorge nginx[6655]: 2025/11/14 09:30:10 [notice] 6>
nov 14 09:30:10 serverlempjorge systemd[1]: Reloaded nginx.service - A high>
nov 14 10:56:58 serverlempjorge systemd[1]: Reloading nginx.service - A hig>
nov 14 10:56:58 serverlempjorge nginx[7982]: 2025/11/14 10:56:58 [notice] 7>
nov 14 10:56:58 serverlempjorge systemd[1]: Reloaded nginx.service - A high>
lines 1-21/21 (END)
```

9. Entramos al navegador y comprobamos que funciona correctamente



HTTPS con Let's Encrypt (DNS-01) en NGINX Lemp

