

# Cómo crear un certificado de servidor seguro

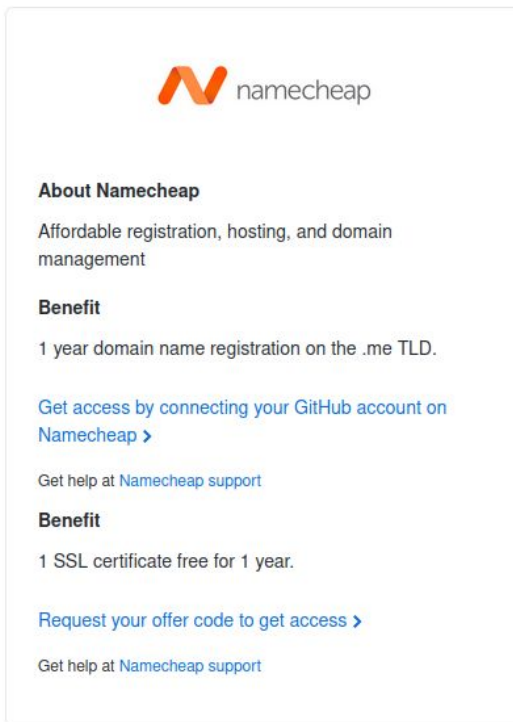
---

**Adrián Núñez Marcos**

Departamento de Lenguajes y Sistemas Inteligentes

# Namecheap

- Vamos a: <https://education.github.com/>
- Y accedemos a “Your benefits”. Una de las ofertas que nos ofrecen es **Namecheap**.
- Gracias al Github Student Developer Pack podemos obtener gratis un dominio .me durante un año.



The screenshot shows the Namecheap logo at the top. Below it, the text "About Namecheap" is followed by "Affordable registration, hosting, and domain management". Under "Benefit", it states "1 year domain name registration on the .me TLD." and provides a link "Get access by connecting your GitHub account on Namecheap >". At the bottom, it says "Get help at Namecheap support".

**About Namecheap**  
Affordable registration, hosting, and domain management

**Benefit**  
1 year domain name registration on the .me TLD.

[Get access by connecting your GitHub account on Namecheap >](#)

Get help at [Namecheap support](#)

**Benefit**  
1 SSL certificate free for 1 year.

[Request your offer code to get access >](#)

Get help at [Namecheap support](#)

# Namecheap

- Tenemos que buscar un nombre de dominio que esté libre:

**Claim your free domain**

Namecheap is giving university students a free "bundle" to kickstart their online presence.

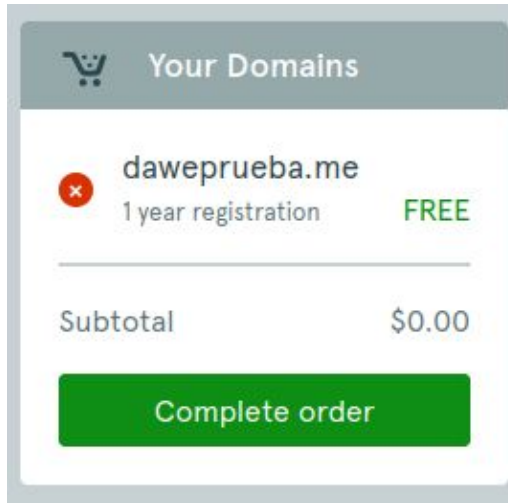
dawe\_prueba .me find

For a limited time, .me domains are **free** for students

daweprueba.me **FREE** **ADD**

# Namecheap

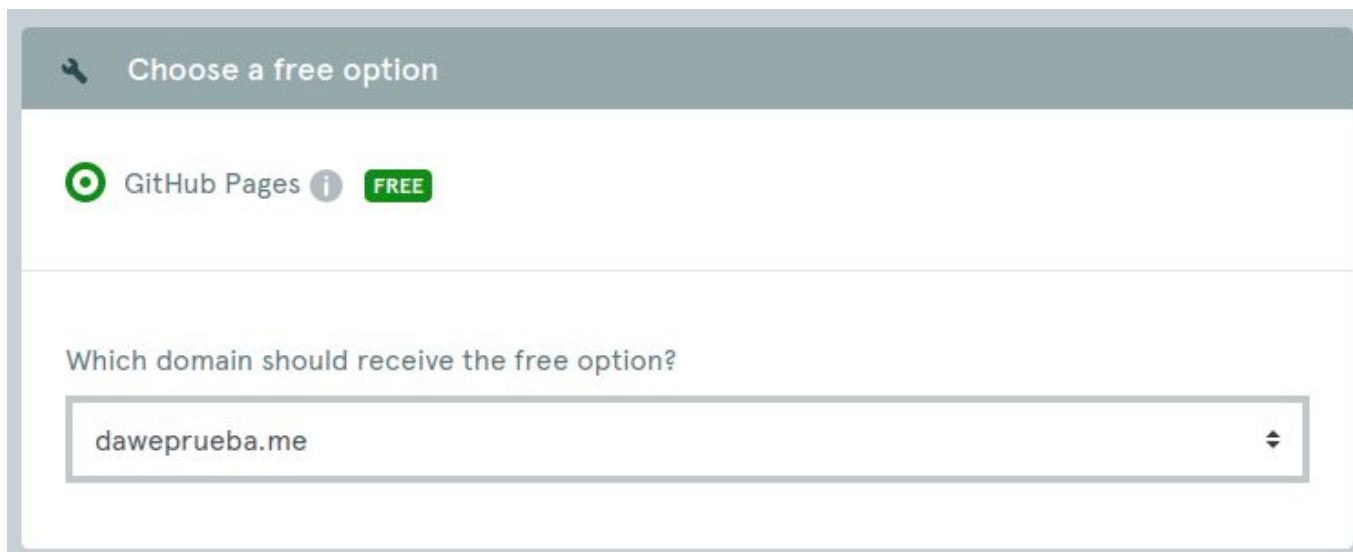
- El resto de dominios NO son gratuitos, por tanto escogeremos el .me y lo añadiremos al carrito.



- Completamos nuestro pedido si vemos que no tenemos que pagar nada.

# Namecheap


- En el siguiente paso, dejamos marcada esta opción:



The screenshot shows a modal window titled "Choose a free option" with a wrench icon. It features the GitHub Pages logo and a green "FREE" badge. Below this, it asks "Which domain should receive the free option?" and shows a dropdown menu with "daweprueba.me" selected.

# Namecheap

- Y justo debajo de la anterior opción, incluid el **email principal** de la cuenta con la que tenéis el Github Student Developer Pack.

 Student email

Currently we are only offering free .me domains and a discounted rate on other TLD's to select universities in the US, UK, Canada and Australia. Enter your student email address here. We'll check your eligibility and email you additional instructions.

If you are a part of the GitHub student program please use your primary GitHub email on the checkout form, instead of your student email address. You may learn more about the GitHub Student Developer Pack [here](#).

# Namecheap

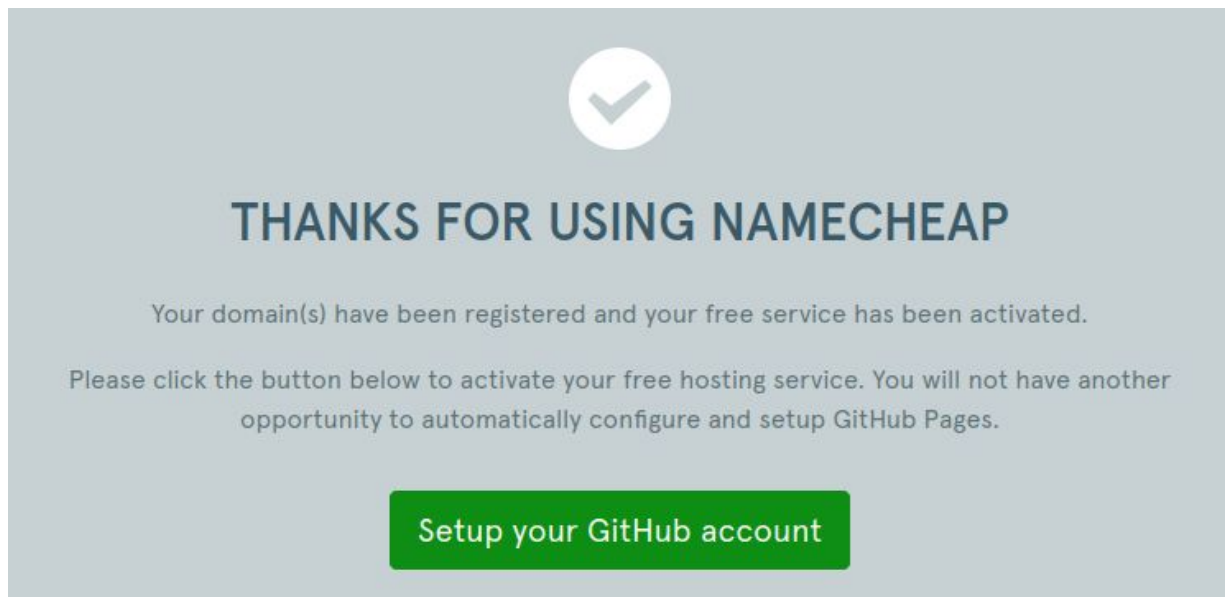
- Finalmente, necesitaremos registrarnos (pulsad el botón “Register”).



A screenshot of the Namecheap login/register interface. The form is enclosed in a light gray border. At the top, there is a dark gray header bar. On the left side of this bar is the word "Login" in white, and on the right side is a dark gray button with the word "REGISTER" in white. Below the header bar, the text "All fields are required" is displayed in a small, gray font. Underneath this text are two input fields. The first field is labeled "Username" and the second field is labeled "Password". Both fields are empty and have a light gray border.

# Namecheap

- Si el registro sale bien, veremos un mensaje como el de abajo. Pulsad en “Setup your Github account” para terminar.








# Namecheap

- Vamos a la página principal de Namecheap y nos autenticamos en nuestra cuenta. Veremos:

Recently Active in Your Account

All	Products	Expiration	
 dawepueba.me  ADD CATEGORY		Mar 20, 2024 Domain	<a href="#">MANAGE</a>

- Pulsamos en “Manage”.
- En la siguiente sección, pinchamos en la pestaña “Advanced DNS”.

# Namecheap

- Aquí veremos esta tabla:

<input type="checkbox"/> Type	Host	Value	TTL	
<input type="checkbox"/> A Record	@	139.59.186.187	Automatic	
<input type="checkbox"/> CNAME Record	www	daweprueba.me.	30 min	

- En lugar de la IP que nos proporciona Namecheap, tendríamos que poner la que nos da Digital Ocean para el droplet que vayamos a usar.
  - Con esto conseguimos que “daweprueba.me” apunte a la IP del droplet.
- Podríamos añadir subdominios como “daweprueba.me/adrian” añadiendo más elementos a la tabla

# Namecheap

- Nos conectamos mediante SSH a la IP que hemos puesto en Namecheap (de un droplet) y vamos a crear una carpeta para el proyecto de prueba:

```
> mkdir daweb-servidor-seguro
```

- Necesitaremos tener instalado Node. Crearemos un proyecto vacío mediante:

```
> cd daweb-servidor-seguro
```

```
> npm init -y
```

```
> npm install express
```

```
> mkdir public
```

# Namecheap

- También crearemos un servidor app.js con el siguiente código:

```
const express = require('express')
const app = express()
const http = require('http')
```

```
http.createServer(app).listen(80, () => { // fijaos que escuchamos en el puerto 80
  console.log('Listening...')
})
```

```
// permitimos crear en public carpetas que empiecen por '.' con la opcion dotfiles
app.use(express.static(__dirname + '/public', { dotfiles: 'allow' } ))
```

- Crearemos también dentro de public/ un fichero index.html con algún mensaje, p.e. “Kaixo”.
  - Si no lo tuviésemos, nos saldría un error “Cannot GET /”.

# Namecheap

- Para poner en marcha el servidor, al usar un puerto de valor bajo (80), necesitamos usar sudo:

> sudo node app.js

- Si no lo lanzamos con screen o similares, en cuanto cerremos la terminal dejará de funcionar.

# Letsencrypt

- Finalmente, podemos configurar este servidor para que sea seguro. Para ello, usaremos la documentación de <https://letsencrypt.org/es/>

- En el servidor instalaremos:

```
> sudo apt-get install software-properties-common  
> snap install certbot --classic
```

- Este script certbot automatiza la petición de un certificado de servidor seguro. Llamaremos para ello a:

```
> sudo certbot certonly --manual
```

# Letsencrypt

- Este último comando nos pedirá unas cuantas cosas:
  - Un email.
  - Que estemos de acuerdo con los términos del servicio -> Y
  - Suscripción a un boletín -> N
  - El dominio que queremos certificar -> dawepueba.me
- Se nos mostrará por pantalla algo similar a:

Create a file containing just this data:

EwwVWJ8RyLg3eIL9iJ3kB4i3MkV14BPdyLaLweTp8tA.yR7leUjaKutJNrjmBYRVjnAL260svhYhqEloJeninWg

And make it available on your web server at this URL:

<http://dawepueba.me/well-known/acme-challenge/EwwVWJ8RyLg3eIL9iJ3kB4i3MkV14BPdyLaLweTp8tA>

# Letsencrypt

- Por tanto, vamos a nuestro droplet, a la carpeta public/ y creamos las carpetas y el fichero que nos piden:

```
> mkdir -p .well-known/acme-challenge  
> cd .well-known/acme-challenge  
> touch EwwVWJ8RyLg3eIL9iJ3kB4i3MkV14BPdyLaLweTp8tA
```

- Y con el editor que más os guste, copiáis el string

EwwVWJ8RyLg3eIL9iJ3kB4i3MkV14BPdyLaLweTp8tA.yR7leUjaKutJNrjmBYRVjnAL260svhYhqEloJeninWg

- que os dan en la terminal dentro del fichero.



# Letsencrypt

- Antes de continuar, nos aseguramos que si entramos en la URL antes mencionada, es decir

`http://daweprueba.me/well-known/acme-challenge/EwwVWJ8RyLg3eIL9iJ3kB4i3MkV14BPdyLaLweTp8tA`

- se nos descarga el fichero que hemos creado (o se mostrará el string antes incluido). Si todo está correcto, volvemos a la terminal donde hemos llamado a certbot y pulsamos Enter.
- Nos debería salir un mensaje diciéndonos que el certificado se ha creado satisfactoriamente.

Successfully received certificate.

Certificate is saved at: `/etc/letsencrypt/live/daweprueba.me/fullchain.pem`

Key is saved at: `/etc/letsencrypt/live/daweprueba.me/privkey.pem`

This certificate expires on 2023-06-19.

These files will be updated when the certificate renews.

# Letsencrypt

- Finalmente, tenemos que modificar app.js para añadir el certificado. Nuestro nuevo código:

```
const express = require('express')
const app = express()
const fs = require('fs')
const https = require('https')
const http = require('http');

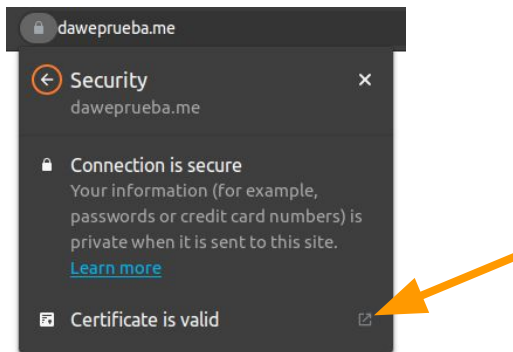
app.use(express.static(__dirname + '/public', { dotfiles: 'allow' } ))
app.get('/', (req, res) => {
  res.send('Hello HTTPS!')
})
http.createServer(app).listen(80, () => {
  console.log('Listening...')
})
https.createServer({ // otro servidor para HTTPS a la escucha en el puerto 443
  key: fs.readFileSync('/etc/letsencrypt/live/daweprueba.me/privkey.pem'), // estas dos líneas hay que personalizarlas
  cert: fs.readFileSync('/etc/letsencrypt/live/daweprueba.me/fullchain.pem')
}, app).listen(443, () => {
  console.log('Listening...')
})
```

# Letsencrypt

- Si vamos a dawepueba.me veremos que sigue apareciendo el aviso de que la conexión con este sitio no es segura.
- Si accedemos mediante HTTPS, es decir, <https://dawepueba.me>, veremos que el aviso desaparece.
- Si borramos public/index.html y recargamos, la página mostrará el mensaje “Hello HTTPS!” que hemos indicado en app.js.

# Letsencrypt

- Si pinchamos donde se señala:



# Letsencrypt

- Desplegaremos una ventana con la información general del certificado:
- Caducará en 3 meses desde el día en que la creamos.

Certificate Viewer: dawepueba.me

General

Details

Issued To

Common Name (CN)	dawepueba.me
Organization (O)	<Not Part Of Certificate>
Organizational Unit (OU)	<Not Part Of Certificate>

Issued By

Common Name (CN)	R3
Organization (O)	Let's Encrypt
Organizational Unit (OU)	<Not Part Of Certificate>

Validity Period

Issued On	Tuesday, March 21, 2023 at 8:37:50 AM
Expires On	Monday, June 19, 2023 at 9:37:49 AM

Fingerprints

SHA-256 Fingerprint	03 CD 68 18 2E CC 99 B4 0B 12 B0 36 0E 13 26 2F C2 EB 68 87 F6 F1 96 3A 78 9A C5 E2 73 56 13 0F 32 AF A4 CE 54 D3 D3 12 12 E6 54 53 26 D3 BA 4F D9 83 2F F3
SHA-1 Fingerprint	

# Letsencrypt

- Si quisiéramos renovar el certificado, tendríamos que llamar a

> certbot renew

- Aunque para no andar pendientes, tendríamos que crear un script que lo renueve cada x tiempo automáticamente.