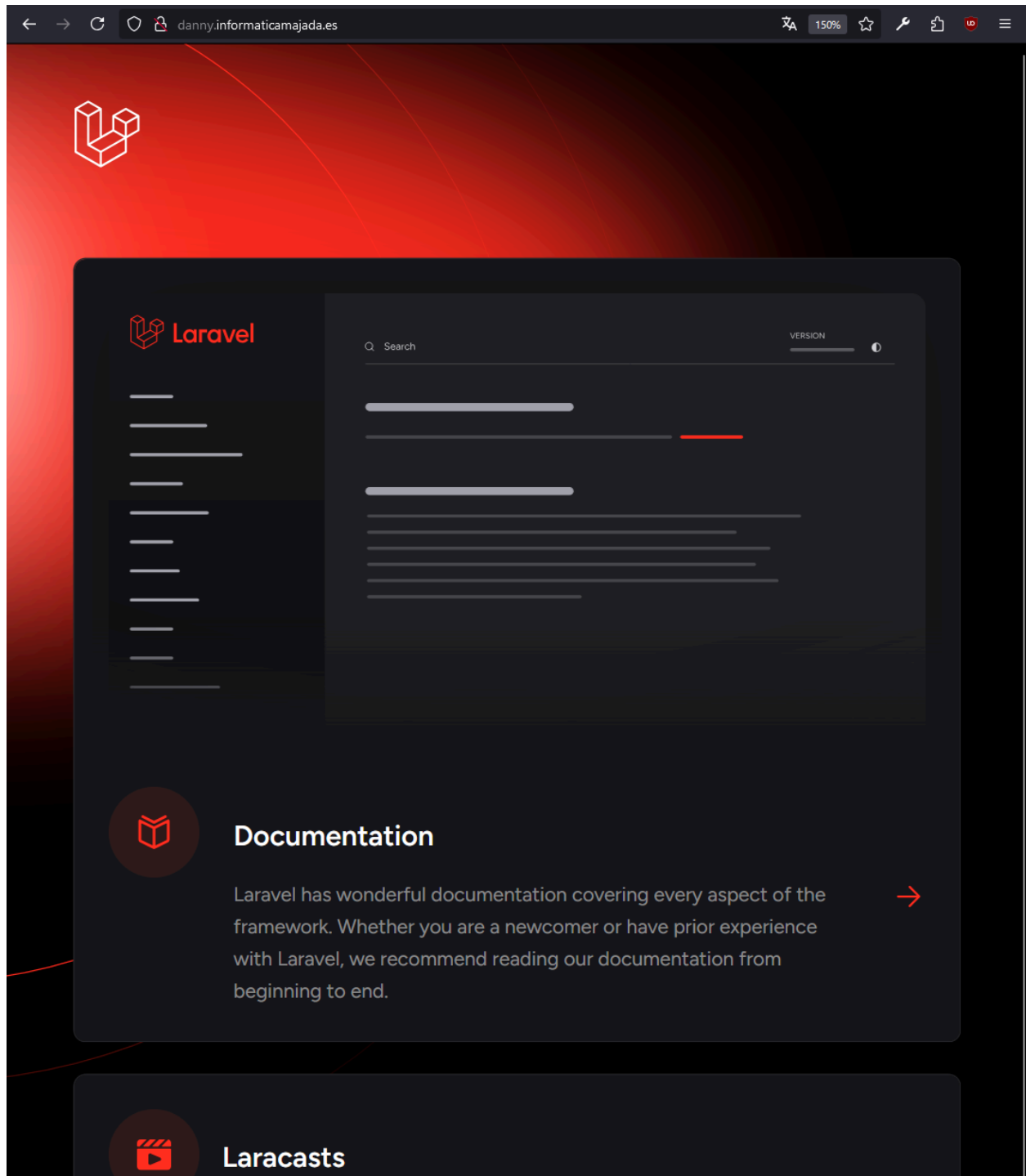


Exercise 1. Create and configure a domain name

Ping your VPS by domain name and check if it is up and running.

```
gaston@danny:~$ ping danny.informaticamajada.es
PING danny.informaticamajada.es (200.234.227.69) 56(84) bytes of data.
64 bytes from elb3a7cf-260b-48f5-8ca1-85b1fe579387.clouding.host (200.234.227.69): icmp_seq=1 ttl=64 time=0.019 ms
64 bytes from elb3a7cf-260b-48f5-8ca1-85b1fe579387.clouding.host (200.234.227.69): icmp_seq=2 ttl=64 time=0.058 ms
64 bytes from elb3a7cf-260b-48f5-8ca1-85b1fe579387.clouding.host (200.234.227.69): icmp_seq=3 ttl=64 time=0.043 ms
64 bytes from elb3a7cf-260b-48f5-8ca1-85b1fe579387.clouding.host (200.234.227.69): icmp_seq=4 ttl=64 time=0.045 ms
^C
--- danny.informaticamajada.es ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3079ms
rtt min/avg/max/mdev = 0.019/0.041/0.058/0.014 ms
gaston@danny:~$ ping informaticamajada.es
PING informaticamajada.es (217.76.128.34) 56(84) bytes of data.
64 bytes from hosting01.servidoresdns.net (217.76.128.34): icmp_seq=1 ttl=249 time=15.6 ms
64 bytes from hosting01.servidoresdns.net (217.76.128.34): icmp_seq=2 ttl=249 time=15.5 ms
64 bytes from hosting01.servidoresdns.net (217.76.128.34): icmp_seq=3 ttl=249 time=15.8 ms
64 bytes from hosting01.servidoresdns.net (217.76.128.34): icmp_seq=4 ttl=249 time=15.7 ms
^C
--- informaticamajada.es ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 15.490/15.672/15.822/0.123 ms
gaston@danny:~$ |
```

Check that you can connect to your Laravel application by the domain name (<http://amirsua.informaticamajada.es>, for example).



Check with the `nslookup` tool (or `dig` if you use Linux) what the IP of your machine is.

```
PS C:\Users\AlumGS> nslookup
Servidor predeterminado: UnKnown
Address: 172.30.1.1

> |
```

Exercise 2. Nginx configuration

In the Nginx configuration file edit the `server name` directive and replace the underscore with your domain name.

```
gaston@danny:/$ sudo nano /etc/nginx/sites-available/laravel
gaston@danny:/$ |
```

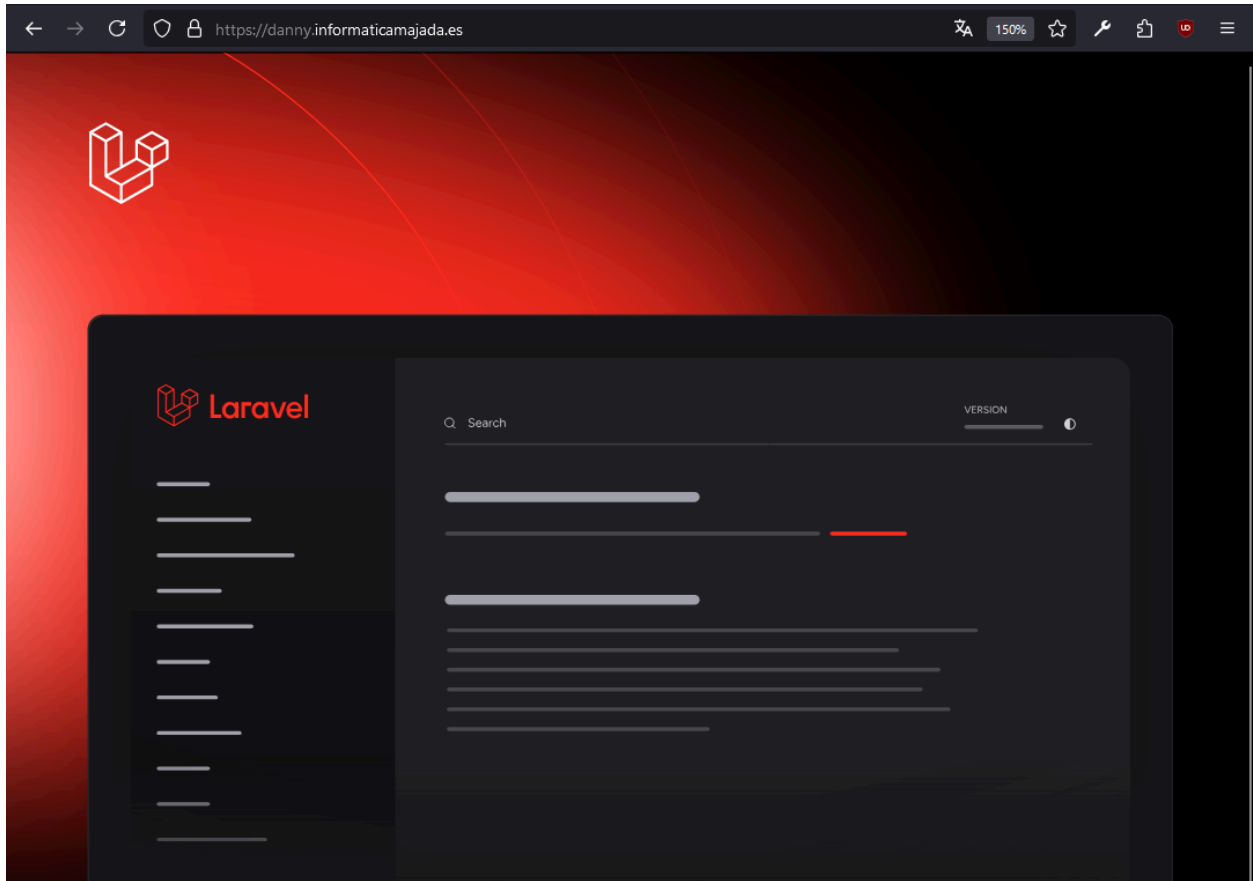
```
# Add index.php to the list if you are using PHP
index index.html index.htm index.php index.nginx-debian.html;

server_name danny.informaticamajada.es;

location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    try_files $uri $uri/ /index.php?$query_string;
}
```

```
sudo systemctl restart nginx
gaston@danny:/$ |
```

Check your Laravel app connecting to your domain name from your browser.



Detail the steps for hosting another Laravel application with other domain.

Exercise 3. Let's encrypt

Before doing anything it is a best practice to do a backup of your Nginx configuration file (`cp /etc/nginx/sites-available/reddit /etc/nginx/sites-available/reddi_backup`).

```
gaston@danny:/$ ls /etc/nginx/sites-available/  
laravel laravel_backup  
gaston@danny:/$
```

Follow this steps:

```
sudo apt install snapd
```

```
gaston@danny:/$ sudo apt install snapd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  squashfs-tools
Suggested packages:
  zenity | kdialog
The following NEW packages will be installed:
  snapd squashfs-tools
0 upgraded, 2 newly installed, 0 to remove and 5 not upgraded.
Need to get 15.1 MB of archives.
After this operation, 61.9 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://debian.repos.clouding.io/debian bookworm/main amd64 squashfs
Get:2 http://debian.repos.clouding.io/debian bookworm/main amd64 snapd a
Fetched 15.1 MB in 1s (20.9 MB/s)
Selecting previously unselected package squashfs-tools.
(Reading database ... 57295 files and directories currently installed.)
```

```
sudo snap install core; sudo snap refresh core
```

```
gaston@danny:/$ sudo snap install core; sudo snap refresh core
core 16-2.61.4-20240607 from Canonical✓ installed
snap "core" has no updates available
gaston@danny:/$ |
```

```
sudo snap install --classic certbot
```

```
gaston@danny:/$ sudo snap install --classic certbot
2024-10-23T18:53:20+02:00 INFO Waiting for automatic snapd restart...
certbot 2.11.0 from Certbot Project (certbot-eff✓) installed
gaston@danny:/$ |
```

```
sudo ln -s /snap/bin/certbot /usr/bin/certbot
```

```
sudo certbot --nginx (answer the questions)
```

```

gaston@danny:/$ sudo ln -s /snap/bin/certbot /usr/bin/certbot
gaston@danny:/$ sudo certbot --nginx
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Enter email address (used for urgent renewal and security notices)
(Enter 'c' to cancel): gastbr@gmail.com

-----
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.4-April-3-2024.pdf. You must agree in
order to register with the ACME server. Do you agree?
-----
(Y)es/(N)o: y

-----
Would you be willing, once your first certificate is successfully issued, 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 our work encrypting the web,
EFF news, campaigns, and ways to support digital freedom.
-----
(Y)es/(N)o: n
Account registered.
Please enter the domain name(s) you would like on your certificate (comma and/or
space separated) (Enter 'c' to cancel): danny.informaticamajada.es
Requesting a certificate for danny.informaticamajada.es

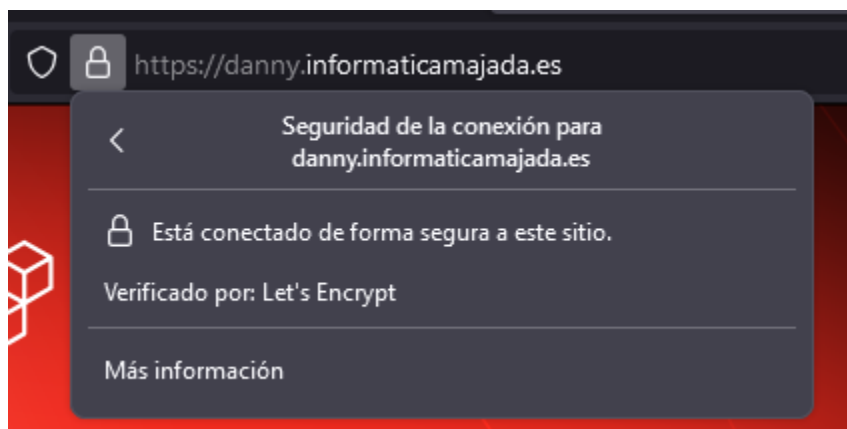
Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/danny.informaticamajada.es/fullchain.pem
Key is saved at: /etc/letsencrypt/live/danny.informaticamajada.es/privkey.pem
This certificate expires on 2025-01-21.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

Deploying certificate
Successfully deployed certificate for danny.informaticamajada.es to /etc/nginx/sites-enabled/laravel
Congratulations! You have successfully enabled HTTPS on https://danny.informaticamajada.es

-----
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
-----
gaston@danny:/$

```

Check that you can connect via https to your domain and click on the padlock to see the certificate (include a screenshot).



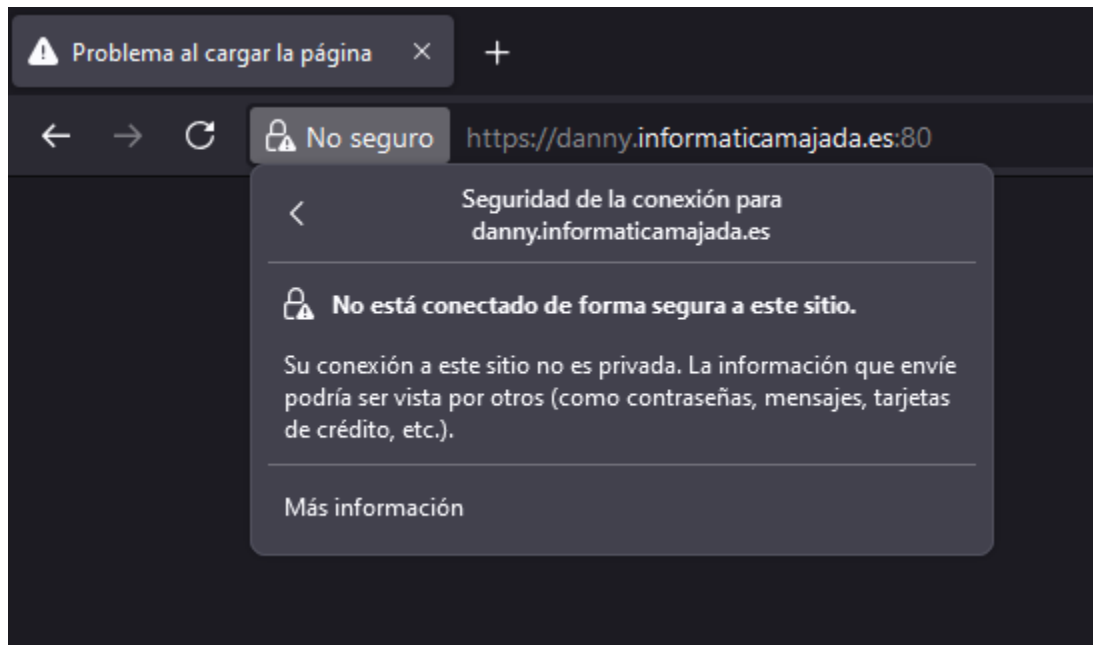
Check the configuration file and answer to these questions:

- In which folder the private key of your certificate is stored?

/etc/letsencrypt/live/[dominio]/privkey.pem

- What happens when you connect to the 80 port? Why?

Port 80 is reserved for the http protocol. In the screenshot, the browser tries to connect with a https protocol through a port not prepared for handling https (80). Connecting to port 80 with the http protocol still works, but without a certificate.



Certbot will renew your certificate automatically. Run this command to check it:

`certbot renew`

```
gaston@danny:/$ sudo certbot renew
[sudo] password for gaston:
Saving debug log to /var/log/letsencrypt/letsencrypt.log

-----
Processing /etc/letsencrypt/renewal/danny.informaticamajada.es.conf
-----
Certificate not yet due for renewal

-----
The following certificates are not due for renewal yet:
  /etc/letsencrypt/live/danny.informaticamajada.es/fullchain.pem expires on 2025-01-21 (skipped)
No renewals were attempted.
-----
```

INSTRUCTIONS

Throughout this lab we are going to set up our own domain and to use Let's encrypt to work with https.

Exercise 1. Create and configure a domain name

We want our VPS to respond to the domain name that we have provided to you. To do that you would have to access your control panel in the company in which you would have registered your domain. In our case it is already done.

Note. It is a good idea to have your own domain. It is cheap and allows us to have our portfolio public in a professional manner.

Ping your VPS by domain name and check if it is up and running.

Check that you can connect to your Laravel application by the domain name (http://amirsua.informaticamajada.es, for example).

Check with the `nslookup` tool (or `dig` if you use Linux) what the IP of your machine is.

Exercise 2. Nginx configuration

We will configure the Nginx server to serve the Laravel app only when the requests are targeting our domain. In the Nginx configuration file edit the `server_name` directive and replace the underscore with your domain name. For instance `amirsua.informaticamajada.es`. Notice that this is the default server block.

Check your Laravel app connecting to your domain name from your browser.

Detail the steps for hosting another Laravel application with other domain.

Exercise 3. Let's encrypt

Let's Encrypt is a non-profit Certificate Authority (CA) that provides an easy way to obtain and install free TLS/SSL certificates (they last 90 days and renew automatically) allowing HTTPS traffic on web servers. In this exercise we are going to:

1. Install the Let's Encrypt client called certbot
2. Download the SSL certificates for our domain.
3. Configure Nginx to use these certificates.
4. Configure automatic certificate renewal.

Before doing anything it is a best practice to do a backup of your Nginx configuration file (`cp /etc/nginx/sites-available/reddit /etc/nginx/sites-available/reddi_backup`).

Follow this steps:

```
sudo apt install snapd
```

```
sudo snap install core; sudo snap refresh core
```

```
sudo snap install --classic certbot
```

```
sudo ln -s /snap/bin/certbot /usr/bin/certbot
```

```
sudo certbot --nginx (answer the questions)
```

Check that you can connect via https to your domain and click on the padlock to see the certificate (include a screenshot).

Check the configuration file and answer to these questions:

- In which folder the private key of your certificate is stored?
- What happens when you connect to the 80 port? Why?

Certbot will renew your certificate automatically. Run this command to check it:

```
certbot renew
```