



Informe de 2FAuth



Materia: Laboratorio de Redes y Sistemas Operativos

Profesor: Jose Luis Di Biase

Alumnos:

- Narizzano Ignacio
- Provvidenza Ivana
- Sciutto Malena

Consignas:

- Instalación y configuración de 2FAuth
- No usar instalación con docker
- Usar como servidor HTTP: Apache2
- Usar como motor de DB: Postgres
- Configuración para envío de emails
- Probar y configurar extensión para navegador

Índice

[Descripción del proyecto](#)

[Requisitos para instalar 2FAuth en un servidor propio:](#)

[Arquitectura](#)

[Instalación de pre-requisitos:](#)

[- Instalación de servidor Apache](#)

[- Instalación de base de datos PostgreSQL](#)

[- Instalación de PHP >= 8.4](#)

[- Instalación de php Composer global \(<https://getcomposer.org/download/>\)](#)

[Instalación de 2FAuth](#)

[Instalación de dependencias de Composer:](#)

[Configuración de Web server](#)

[Configuración de variables de entorno](#)

[Configuración de Email](#)

[Últimas configuraciones](#)

[Extensión de Navegador](#)

[Uso de 2FAuth](#)

[Configuración de extensión](#)

[Problemas y observaciones encontrados](#)

[Conclusión](#)

Descripción del proyecto

Este proyecto consiste en la instalación y puesta a punto de 2FAuth, un gestor web de autenticación de dos factores (2FA) auto hospedado. Su propósito principal es brindar una alternativa independiente, privada y accesible a las aplicaciones comerciales tradicionales, permitiendo a los usuarios almacenar, organizar y generar sus códigos de seguridad desde cualquier dispositivo con un navegador web. Al desplegar este servicio en un servidor propio utilizando Apache2 y PostgreSQL, se garantiza el control total sobre la privacidad de los datos y se habilita la capacidad de gestionar cuentas 2FA de manera segura y colaborativa en un entorno multiusuario.

Utilizamos una máquina virtual en VirtualBox con las siguientes características:

Sistema operativo: Ubuntu Desktop 26.04 LTS

Núcleos del procesador: 4.

Memoria RAM: 4GB.

Espacio en disco: 25 GB.

Requisitos para instalar 2FAuth en un servidor propio:

Sistema operativo

La documentación oficial no especifica un sistema operativo concreto. **Linux** es claramente la plataforma principal. Todos los ejemplos y comandos de la documentación usan rutas estilo Unix (`/var/www/2fauth`), comandos como `sudo`. Es el entorno donde mejor está documentado y probado.

Servidor web

Se necesita un servidor HTTP, siendo Apache y NGINX las opciones soportadas.

PHP

Se requiere PHP 8.4 o superior, junto con las siguientes extensiones: BCMath, CType, DOM, Fileinfo, GD, JSON, Mbstring, OpenSSL, PDO, Tokenizer y XML. También hay que instalar la extensión PHP correspondiente a la base de datos elegida (por ejemplo: `php8.1-sqlite3` o `php8.1-mysql`).

Base de datos

Se necesita una de las siguientes bases de datos: MariaDB 10.3+, MySQL 5.7+, PostgreSQL 10.0+, SQLite 3.26.0+ o SQL Server 2017+.

Dado que 2FAuth es una aplicación muy liviana con pocas conexiones concurrentes SQLite es la mejor opción para uso personal o con pocos usuarios. Para más usuarios se recomienda una base de datos SQL en servidor.

Composer

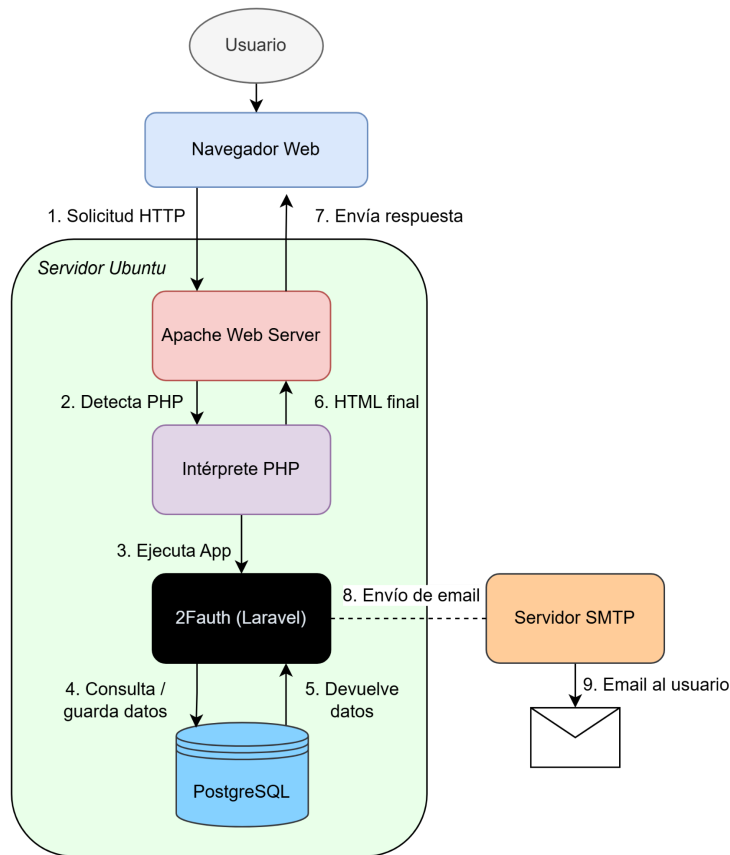
Descarga e instala todas las librerías PHP que necesita Laravel para funcionar. Luego no es necesario para el uso de la aplicación.

Navegador Web

Para poder utilizar la extensión es necesario tener Chrome o Mozilla para instalarla.

A continuación se detalla la configuración con servidor Apache, base de datos PostgreSQL, envío de mails y extensión de navegador. Sin instalación por Docker.

Arquitectura



Usuario → Apache

El usuario accede desde su navegador o dispositivo móvil al servidor Apache mediante HTTP/HTTPS.

Apache → PHP

Apache recibe las peticiones HTTP y las delega a PHP para que las procese (es el motor que ejecuta la lógica de la aplicación)

PHP → 2FAuth (Laravel)

PHP ejecuta el código de la aplicación 2FAuth, que está construida sobre el framework Laravel. Aquí se generan los códigos OTP y se gestiona todo el negocio.

2FAuth → PostgreSQL

La aplicación lee y escribe en la base de datos: guarda las cuentas 2FA, los usuarios, configuraciones y seeds de los tokens.

2FAuth → Servidor SMTP

Cuando un usuario solicita resetear su contraseña, la aplicación le envía un email a través del servidor de correo configurado.

Instalación de pre-requisitos:

- Instalación de servidor Apache

install apache2

```
ubuntu@ubuntu:~/Desktop$ sudo apt install apache2
Installing:
  apache2

Installing dependencies:
  apache2-bin  apache2-utils  libaprutil1-dbd-sqlite3  libaprutil1t64
  apache2-data  libapr1t64  libaprutil1-ldap

Suggested packages:
  apache2-doc  apache2-suexec-pristine | apache2-suexec-custom

Summary:
  Upgrading: 0, Installing: 8, Removing: 0, Not Upgrading: 288
  Download size: 1938 kB
  Space needed: 7568 kB / 1614 MB available

Continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu resolute/main amd64 libapr1t64 amd64 1.7.6-3 [109 kB]
Get:2 http://archive.ubuntu.com/ubuntu resolute/main amd64 libaprutil1t64 amd64 1.6.3-3ubuntu3 [93.8 kB]
Get:3 http://archive.ubuntu.com/ubuntu resolute/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.3-3ubuntu3 [11.4 kB]
Get:4 http://archive.ubuntu.com/ubuntu resolute/main amd64 libaprutil1-ldap amd64 1.6.3-3ubuntu3 [9226 B]
```

Para comprobar que Apache está funcionando:

systemctl status apache2

```
user@labo:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Thu 2026-07-02 18:53:45 -03; 13min ago
     Docs: https://httpd.apache.org/docs/2.4/
  Process: 1473 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 1533 (apache2)
    Tasks: 6 (limit: 4596)
  Memory: 18.5M (peak: 19.0M)
     CPU: 209ms
  CGroup: /system.slice/apache2.service
          └─1533 /usr/sbin/apache2 -k start
            └─1536 /usr/sbin/apache2 -k start
              └─1537 /usr/sbin/apache2 -k start
                └─1538 /usr/sbin/apache2 -k start
                  └─1539 /usr/sbin/apache2 -k start
                    └─1540 /usr/sbin/apache2 -k start
```

- Instalación de base de datos PostgreSQL

sudo apt install postgresql postgresql-contrib -y

```
user@labo:~$ sudo apt install postgresql postgresql-contrib -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Para comprobar que funciona:

sudo systemctl start postgresql

sudo systemctl status postgresql

```
user@labo:~$ sudo systemctl start postgresql
user@labo:~$ sudo systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/usr/lib/systemd/system/postgresql.service; enabled; preset: enabled)
   Active: active (exited) since Thu 2026-07-02 19:42:34 -03; 1min 58s ago
     Main PID: 25564 (code=exited, status=0/SUCCESS)
        CPU: 1ms
```

- Instalación de PHP >= 8.4

Permitir paquetes de ondrej:

sudo add-apt-repository ppa:ondrej/php -y

sudo apt update

```
user@labo:~$ sudo add-apt-repository ppa:ondrej/php -y
[sudo] password for user:
PPA publishes dbgsym, you may need to include 'main/debug' component
Repository: 'Types: deb
URIs: https://ppa.launchpadcontent.net/ondrej/php/ubuntu/
Suites: noble
Components: main
'
Description:
Co-installable PHP versions: PHP 5.6, PHP 7.x, PHP 8.x and most requested extensions are included. Packages are provided for *Current* Ubuntu *LTS* releases (https://wiki.ubuntu.com/Releases). Expanded Security Maintenance releases ARE NOT supported.
```

```
user@labo:~$ sudo apt update
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:2 http://ar.archive.ubuntu.com/ubuntu noble InRelease
Hit:3 http://ar.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://ar.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:5 https://ppa.launchpadcontent.net/ondrej/php/ubuntu noble InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
190 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

sudo apt install php8.4 php8.4-cli -y

```
user@labo:~$ sudo apt install php8.4 php8.4-cli -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-hwe-6.17-tools-6.17.0-19 linux-tools-6.17.0-19-generic
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libapache2-mod-php8.4 php-common php8.4-common php8.4-opcache php8.4-readline
```

Como vamos a utilizar una base de datos PostgreSQL hay que instalar la extensión de PHP correspondiente:

sudo apt install php8.4-pgsql

```
user@labo:~$ sudo apt install php8.4-pgsql
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-hwe-6.17-tools-6.17.0-19 linux-tools-6.17.0-19-generic
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libpq5
```

También debemos instalar las siguientes extensiones requeridas:

- BCMath PHP Extension
- Ctype PHP Extension
- DOM PHP Extension
- Fileinfo PHP Extension
- GD PHP Extension
- JSON PHP Extension
- Mbstring PHP Extension
- OpenSSL PHP Extension
- PDO PHP Extension
- Tokenizer PHP Extension
- XML PHP Extension

Algunas extensiones o módulos ya se instalan al instalar php, podemos verificar cuales están instaladas con el comando

php -m

Luego instalamos las extensiones que faltan:

```
sudo apt install php8.5-bcmath php8.5-dom php8.5-gd php8.5-mbstring php8.5-xml
```

```
ubuntu@ubuntu:~$ sudo apt install php8.5-bcmath php8.5-dom php8.5-gd php8.5-mbstring php8.5-xml
Note, selecting 'php8.5-xml' instead of 'php8.5-dom'
Installing:
  php8.5-bcmath  php8.5-gd  php8.5-mbstring  php8.5-xml

Summary:
  Upgrading: 0, Installing: 4, Removing: 0, Not Upgrading: 288
  Download size: 808 kB
  Space needed: 2342 kB / 1437 MB available

Get:1 http://archive.ubuntu.com/ubuntu resolute-updates/universe amd64 php8.5-bcmath amd64 8.5.4-0ubuntu1.1 [29.6 kB]
Get:2 http://archive.ubuntu.com/ubuntu resolute-updates/main amd64 php8.5-gd amd64 8.5.4-0ubuntu1.1 [40.8 kB]
Get:3 http://archive.ubuntu.com/ubuntu resolute-updates/main amd64 php8.5-mbstring amd64 8.5.4-0ubuntu1.1 [525 kB]
Get:4 http://archive.ubuntu.com/ubuntu resolute-updates/main amd64 php8.5-xml amd64 8.5.4-0ubuntu1.1 [212 kB]
Fetched 808 kB in 2s (467 kB/s)
Selecting previously unselected package php8.5-bcmath.
(Reading database ... 212859 files and directories currently installed.)
Preparing to unpack .../php8.5-bcmath_8.5.4-0ubuntu1.1_amd64.deb ...
Unpacking php8.5-bcmath (8.5.4-0ubuntu1.1) ...
Selecting previously unselected package php8.5-gd.
Preparing to unpack .../php8.5-gd_8.5.4-0ubuntu1.1_amd64.deb ...
Unpacking php8.5-gd (8.5.4-0ubuntu1.1) ...
```

- Instalación de php Composer global (<https://getcomposer.org/download/>)

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') ===
'c8b085408188070d5f52bcfe4ecfbee5f727afa458b2573b8eaaf77b3419b0bf2768dc67c
86944da1544f06fa544fd47') { echo 'Installer verified'.PHP_EOL; } else {
echo 'Installer corrupt'.PHP_EOL; unlink('composer-setup.php'); exit(1);
}"
php composer-setup.php
php -r "unlink('composer-setup.php');"
sudo mv composer.phar /usr/local/bin/composer
```

```
user@labo:~$ php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
user@labo:~$ php -r "if (hash_file('sha384', 'composer-setup.php') === 'c8b085408188070d5f52bcfe4ecfbee5f727afa458b2573b8eaaf77b3419b0bf2768dc67c86944da1544f06fa544fd47') { echo 'Installer verified'.PHP_EOL; } else { echo 'Installer corrupt'.PHP_EOL; unlink('composer-setup.php'); exit(1); }"
Installer verified
user@labo:~$ php composer-setup.php
All settings correct for using Composer
Downloading...

Composer (version 2.10.2) successfully installed to: /home/user/composer.phar
Use it: php composer.phar

user@labo:~$ php -r "unlink('composer-setup.php');"
user@labo:~$ sudo mv composer.phar /usr/local/bin/composer
user@labo:~$ composer

 _____
/ _   _/ _   _ \ _   _ \ _   _ \ _   _ \ _   _ \ _   _ \
| | | | | | | | | | | | | | | | | | | | | | | | | | | |
\_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|

Composer version 2.10.2 2026-07-01 11:24:45
```

Instalación de 2Fauth

Para instalar 2Fauth (en este caso la última versión):

Crear la carpeta: **/var/www/ mkdir 2fauth**

Clonar el repo: **git clone https://github.com/bubka/2fauth.git**

git config --global --ad.safe.directory /var/www/2fauth

curl https://api.github.com/repos/Bubka/2FAuth/releases/latest | grep "\"name\"\" | grep -Eo 'v[^\"]*' | xargs git checkout

```
user@labo:/var/www/2fauth$ sudo git clone https://github.com/bubka/2fauth.git .
Cloning into '.'...
remote: Enumerating objects: 27384, done.
remote: Counting objects: 100% (2304/2304), done.
remote: Compressing objects: 100% (1061/1061), done.
remote: Total 27384 (delta 1594), reused 1518 (delta 1230), pack-reused 25080 (from 4)
Receiving objects: 100% (27384/27384), 22.68 MiB | 9.94 MiB/s, done.
Resolving deltas: 100% (18382/18382), done.
```

Instalación de dependencias de Composer:

Desde /var/www/2fauth

sudo apt install php-curl

composer update

Siendo user el usuario del sistema: **sudo chown -R user:user /var/www/2fauth**

composer install --prefer-dist --no-scripts --no-dev

```
user@labo:/var/www/2fauth$ composer install --prefer-dist --no-scripts --no-dev
Installing dependencies from lock file
Verifying lock file contents can be installed on current platform.
Package operations: 111 installs, 0 updates, 0 removals
- Downloading doctrine/lexer (3.0.1)
- Downloading enshrined/svg-sanitize (0.22.0)
- Downloading symfony/http-foundation (v8.1.0)
- Downloading google/protobuf (v4.33.6)
```

Configuración de Web server

```
cd /etc/apache2/sites-available
sudo nano 2fauth.conf
```

y agregar lo siguiente al archivo:

```
<VirtualHost *:80>
    ServerName example.com
    ServerAdmin webmaster@example.com
    DocumentRoot /var/www/2fauth/public

    <Directory /var/www/2fauth/public>
        Options Indexes MultiViews FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Guardar y cerrar el archivo.

Permitir un nuevo VHost y reiniciar Apache2:

Deshabilitamos el que viene por defecto

```
sudo a2dissite 000-default.conf
```

Habilitamos el que creamos

```
sudo a2ensite 2fauth
```

Reiniciamos Apache

```
sudo systemctl restart apache2
```

```
user@labo:/etc/apache2/sites-available$ sudo a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
    systemctl reload apache2
user@labo:/etc/apache2/sites-available$ sudo a2ensite 2fauth
Enabling site 2fauth.
To activate the new configuration, you need to run:
    systemctl reload apache2
user@labo:/etc/apache2/sites-available$ sudo systemctl restart apache2
user@labo:/etc/apache2/sites-available$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Thu 2026-07-02 20:41:08 -03; 14s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 45742 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 45748 (apache2)
```

Agregar un Alias a la base url en la configuración del server:

```
GNU nano 7.2                                2fauth.conf *
<VirtualHost *:80>
    ServerName example.com
    ServerAdmin webmaster@example.com
    DocumentRoot /var/www/2fauth/public

    <Directory /var/www/2fauth/public>
        Options Indexes MultiViews FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
<IfModule alias_module>
    Alias /2fauth "/var/www/2fauth/public/"
</IfModule>
```

Crear la database

Abrir postgresql con

sudo -u postgres psql

Crear la db con **CREATE DATABASE "2fauth";**

```
user@labo:/etc/apache2/sites-available$ sudo -u postgres psql
psql (16.14 (Ubuntu 16.14-0ubuntu0.24.04.1))
Type "help" for help.

postgres=# CREATE DATABASE "2fauth";
CREATE DATABASE
postgres=# \l
postgres=# SELECT 1 FROM pg_database WHERE datname = '2fauth';
?column?
-----
      1
(1 row)

postgres=#
```

Crear el usuario de la db con el nombre y contraseña de tu elección (para este ejemplo user2fauth y 2fauthpswd) con **CREATE USER {username} WITH PASSWORD '{password}'**

```
postgres=# CREATE USER user2fauth WITH PASSWORD '2fauthpswd';
CREATE ROLE
```

Dar permisos al usuario creado para la db que vamos a utilizar.

GRANT ALL PRIVILEGES ON DATABASE "2fauth" TO user2fauth;

```
postgres=# GRANT ALL PRIVILEGES ON DATABASE "2fauth" TO user2fauth;
GRANT
postgres=#
```

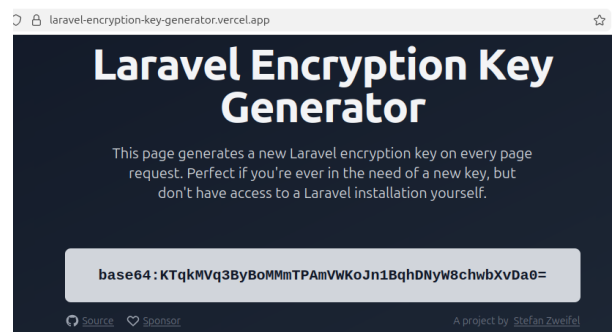
Configuración de variables de entorno

Desde /var/www/2fauth, crear un archivo .env usando la plantilla de .env.example corriendo el comando: **cp .env.example .env**

Crear una app key usando <https://laravel-encryption-key-generator.vercel.app/>

Abrir el .env con un editor de texto o usando **nano .env** y rellenar el **APP_KEY** con lo obtenido en el paso anterior.

Rellenar las variables de conexión con la db con los datos de la base creada.



Asegurarse de que las siguientes envs se vean así:

APP_URL=http://localhost/2fauth

ASSET_URL=http://localhost/2fauth

APP_SUBDIRECTORY=/2fauth

```
# DB_CONNECTION=sqlite
# DB_DATABASE="C:\\path\\to\\your\\database.sqlite"

DB_CONNECTION=pgsql
DB_DATABASE=2fauth
DB_HOST=localhost
DB_PORT=5432
DB_USERNAME=user2fauth
DB_PASSWORD=2fauthpswd
```

```
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify  ^_ Go To Line
```

Configuración de Email

Los valores utilizados en esta sección dependen del proveedor de servicio de correo que utilicemos. En este caso configuramos una de nuestras cuentas personales de Gmail para que figure como remitente de los correos que envía la aplicación. Para esto tenemos que activar la verificación en dos pasos de la cuenta de google utilizada, ya que es condición necesaria para generar una contraseña de aplicación ([ver más](#)) que es la que vamos a ingresar en el campo MAIL_PASSWORD.

Editamos el archivo `.env` en `/var/www/2fauth` con:

```
sudo nano .env
```

Completamos los siguientes campos reemplazando con nuestro correo y contraseña de aplicación.

```
MAIL_MAILER=smtp
MAIL_HOST=smtp.gmail.com
MAIL_PORT=587
MAIL_USERNAME=correo@gmail.com
MAIL_PASSWORD=contraseñaDeAplicacion
MAIL_ENCRYPTION=tls
MAIL_FROM_NAME="Seguridad 2FAuth"
MAIL_FROM_ADDRESS=correo@gmail.com
```

```
#### Mail settings ####

# Refer your email provider documentation to configure your mail settings
# Set a value for every available setting to avoid issue

MAIL_MAILER=smtp
MAIL_HOST=smtp.gmail.com
MAIL_PORT=587
MAIL_USERNAME=correo@gmail.com
MAIL_PASSWORD=contraseniaDeAplicacion
MAIL_ENCRYPTION=tls
MAIL_FROM_NAME="Seguridad 2Fauth"
MAIL_FROM_ADDRESS=correo@gmail.com
```

Una vez guardados los cambios en el archivo `.env`, debemos limpiar la caché de configuración de Laravel ejecutando el comando `sudo -u www-data php artisan config:clear` desde la carpeta del proyecto para que el sistema reconozca las nuevas credenciales de Gmail.

```
user@labo:/var/www/2fauth$ sudo -u www-data php artisan config:clear

INFO Configuration cache cleared successfully.
```

Últimas configuraciones

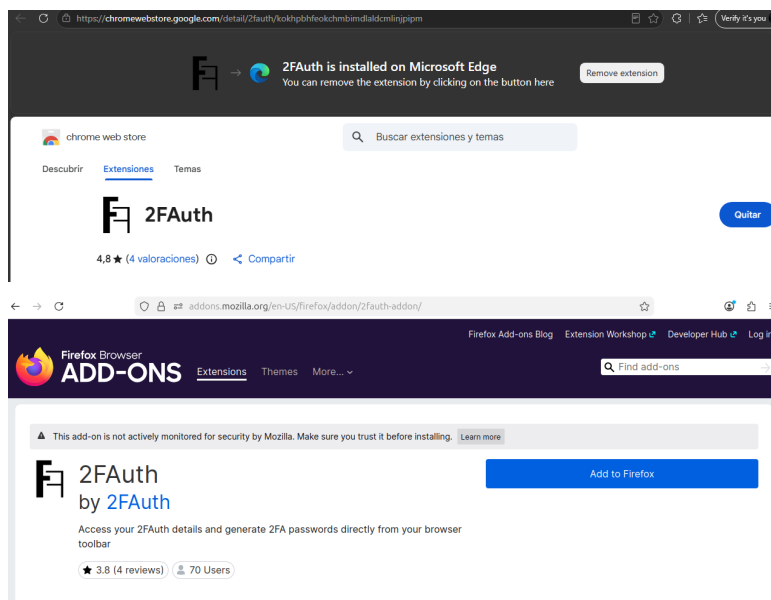
Para completar la configuración de la URL base personalizada, abrir el archivo **/var/www/2fauth/public/.htaccess**, descomentar la directiva RewriteBase y editar el valor de subdir para que coincida con el valor de APP_SUBDIRECTORY. En este caso es **/2fauth/**

Por último, correr los comandos:

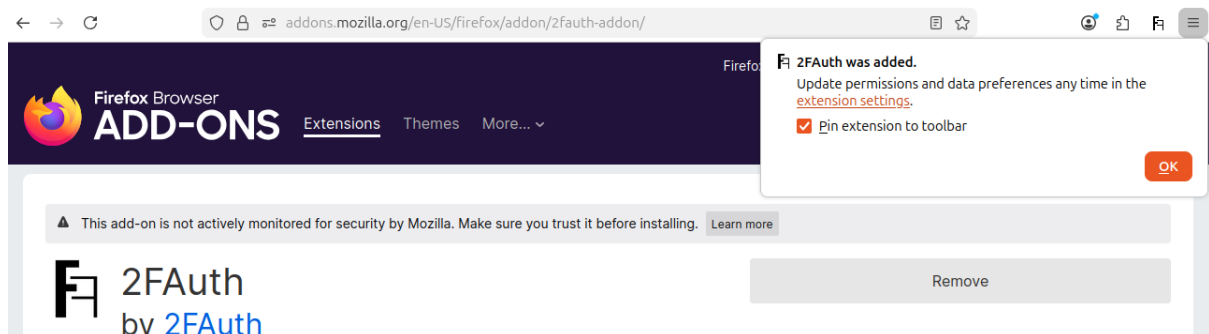
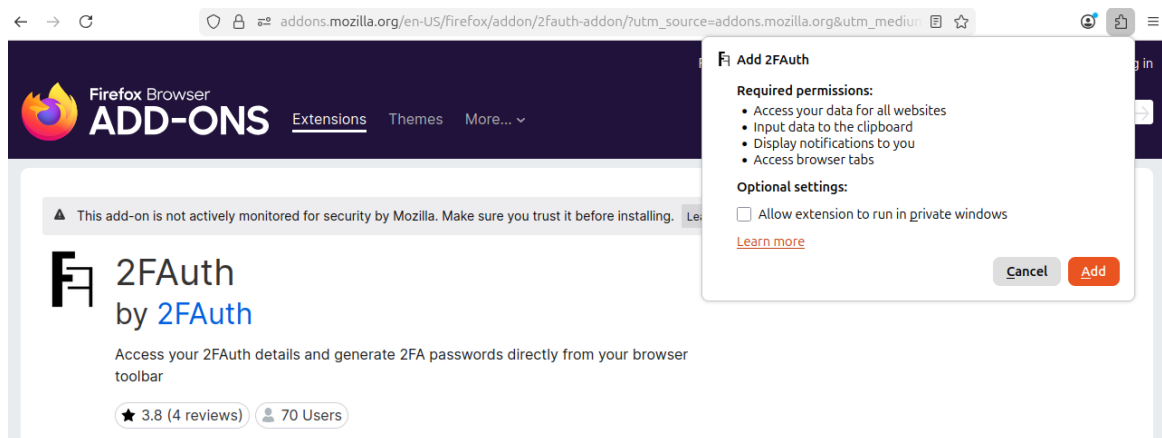
```
sudo -u www-data php /var/www/2fauth/artisan migrate:refresh
sudo -u www-data php /var/www/2fauth/artisan passport:install
sudo -u www-data php /var/www/2fauth/artisan storage:link
sudo -u www-data php /var/www/2fauth/artisan config:cache
sudo -u www-data php /var/www/2fauth/artisan 2fauth:fix-passport-key-permissions
```

Extensión de Navegador

La extensión se comunica de forma bidireccional con Apache: consulta los códigos OTP y recibe la respuesta para mostrarlos en el navegador. Está disponible para Firefox, Chrome y Edge.

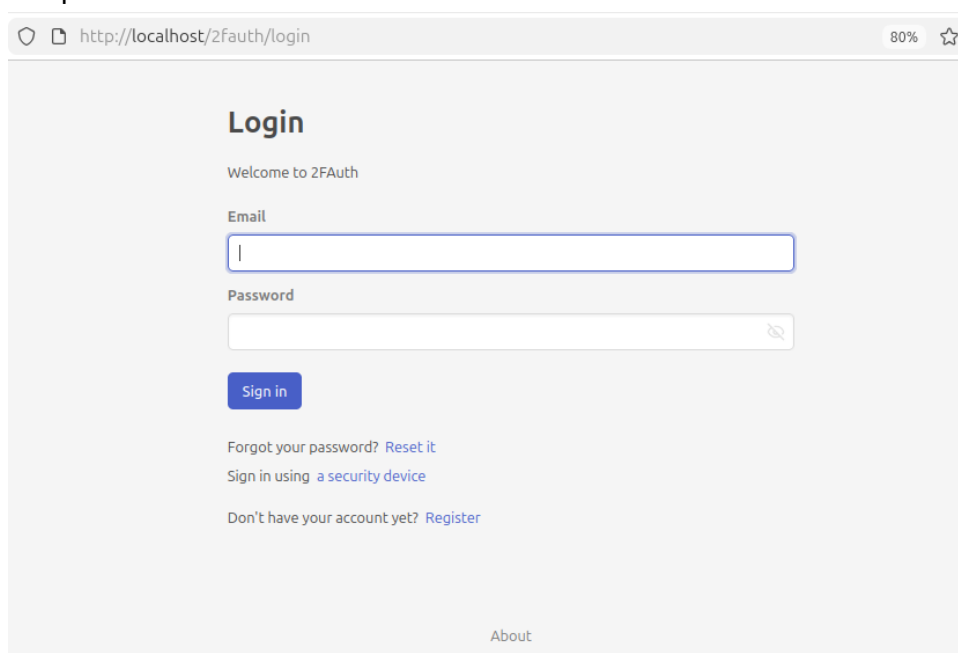


Trabajaremos con Firefox. Al agregar la extensión detalla los permisos y pregunta si deseamos agregarlo a nuestra barra de herramientas.



Uso de 2FAuth

Con todo instalado y funcionando accedemos a <http://localhost/2fauth/>
Podemos loguearnos con email o con dispositivo, registrarnos, resetear la contraseña, recuperar cuenta.



http://localhost/2fauth/register

Register

Welcome to 2FAuth

You need an account to go further, please register yourself.

Name

Email

Password

Mandatory

- 8 characters min.

Recommended (highly)

- Has lower case
- Has upper case
- Has special char
- Has number

Register

Already registered? [Sign in](#)

About

http://localhost/2fauth/password/request80%

☆

Reset password

2FAuth will send you a password reset link to this address. Click the link in the received email to set a new password.

Email

Send password reset link

[Cancel](#)

http://localhost/2fauth/webauthn/lost80%

☆

Account recovery

2FAuth will send you a recovery link to this email address. Click the link in the received email and follow the instructions.

Ensure you open the email on a device you fully own.

Email

Send recovery link

[Cancel](#)

http://localhost/2Fauth/login 80% ☆

WebAuthn login

Welcome to 2FAuth

Get ready to authenticate using (one of) your security devices. Plug your key in, remove face mask or gloves, etc.

Email

[Continue](#)

Lost your device? [Recover your account](#)

Sign in using [login & password](#)

Don't have your account yet? [Register](#)

Registro un usuario que será Admin. La documentación indica que la primera cuenta creada será el Administrador del resto de usuarios.

Register

Welcome to 2FAuth

You need an account to go further, please register yourself.

Name

Email

Password

Mandatory

- 8 characters min.

Recommended (highly)

- Has lower case
- Has upper case
- Has special char
- Has number

[Register](#)

Already registered? [Sign in](#)

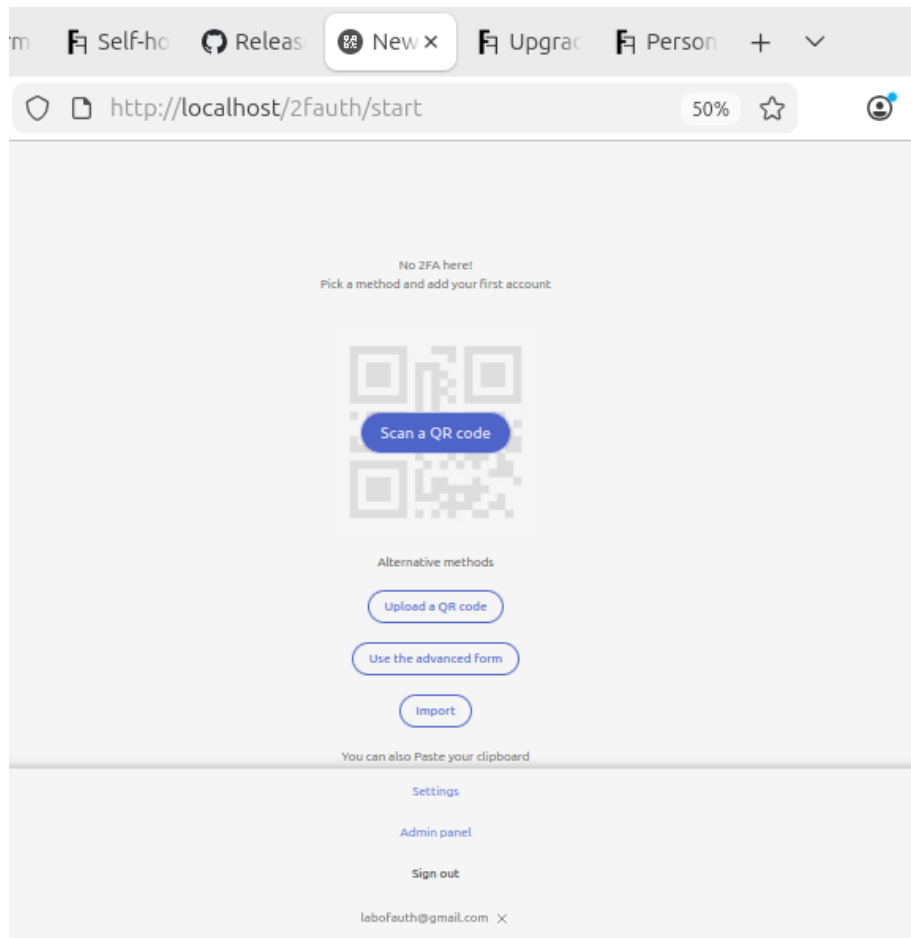
Authentication

You can enhance the security of your 2FAuth account by enabling WebAuthn authentication.

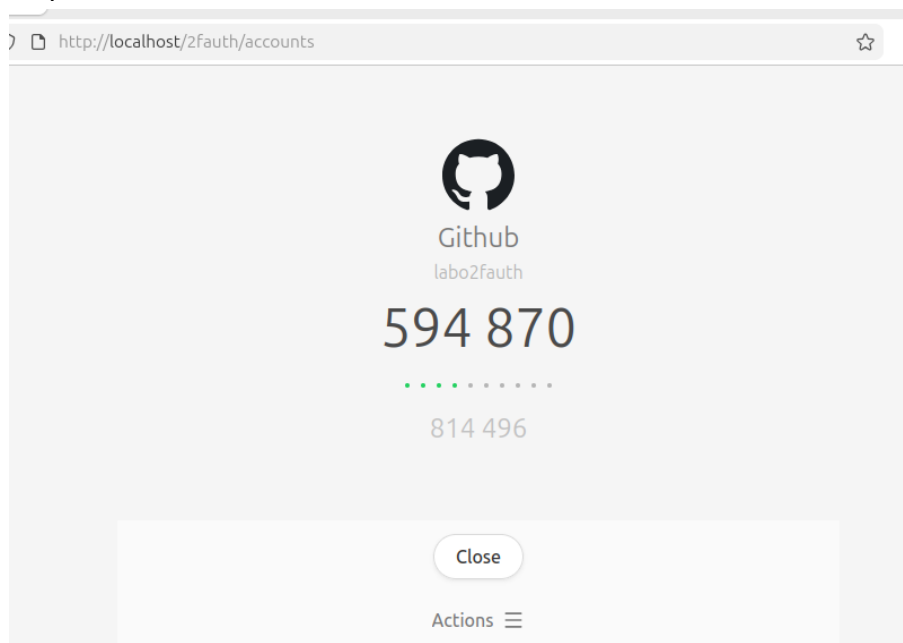
WebAuthn allows you to use trusted devices (like Yubikeys or smartphones with biometric capabilities) to sign in quickly and more securely.

[Register a device](#) [Maybe later](#)

Y podemos ver que es usuario Admin.

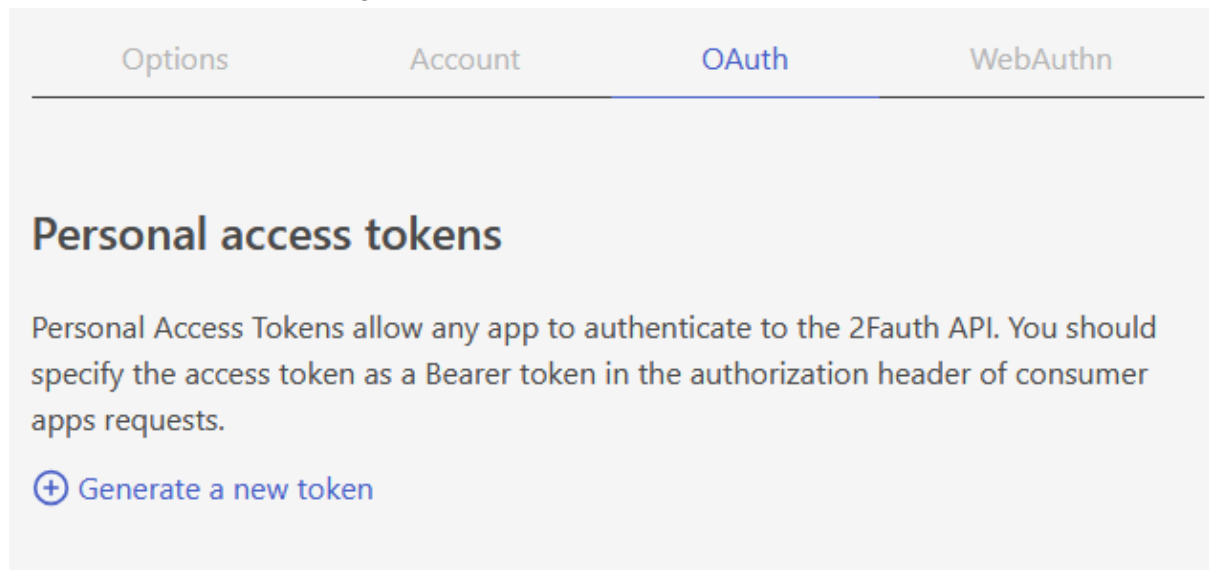


Se pudo vincular a la cuenta de GitHub

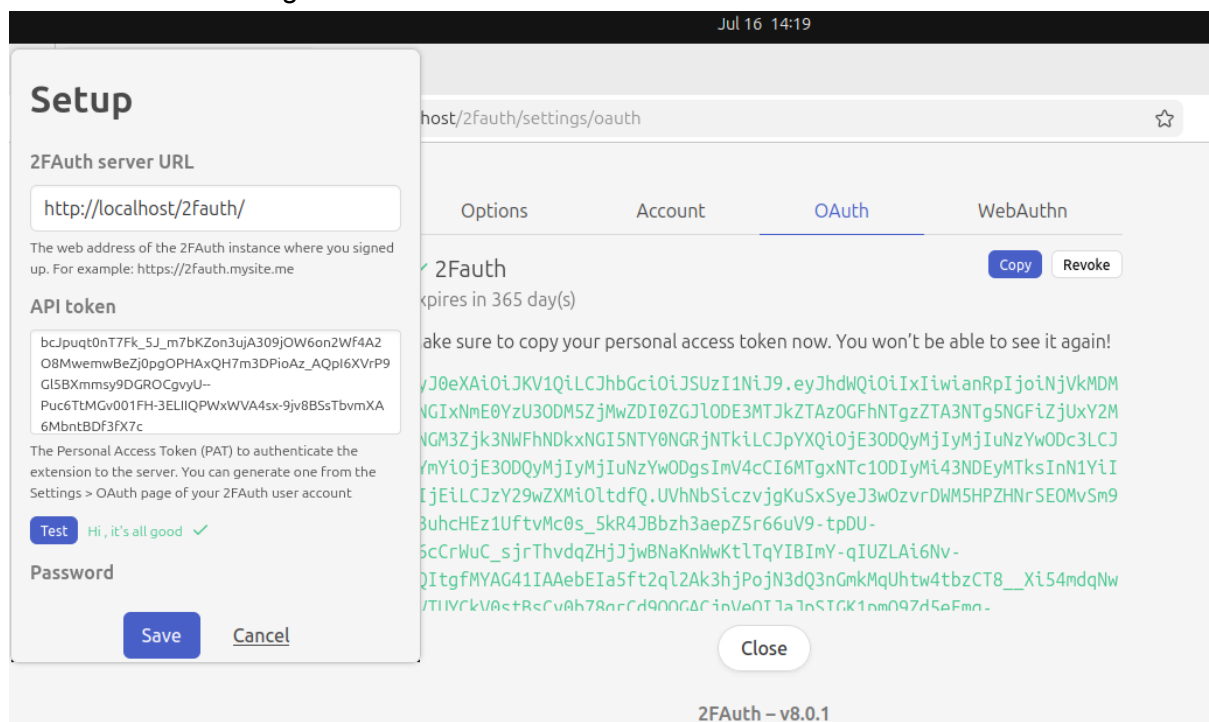


Configuración de extensión

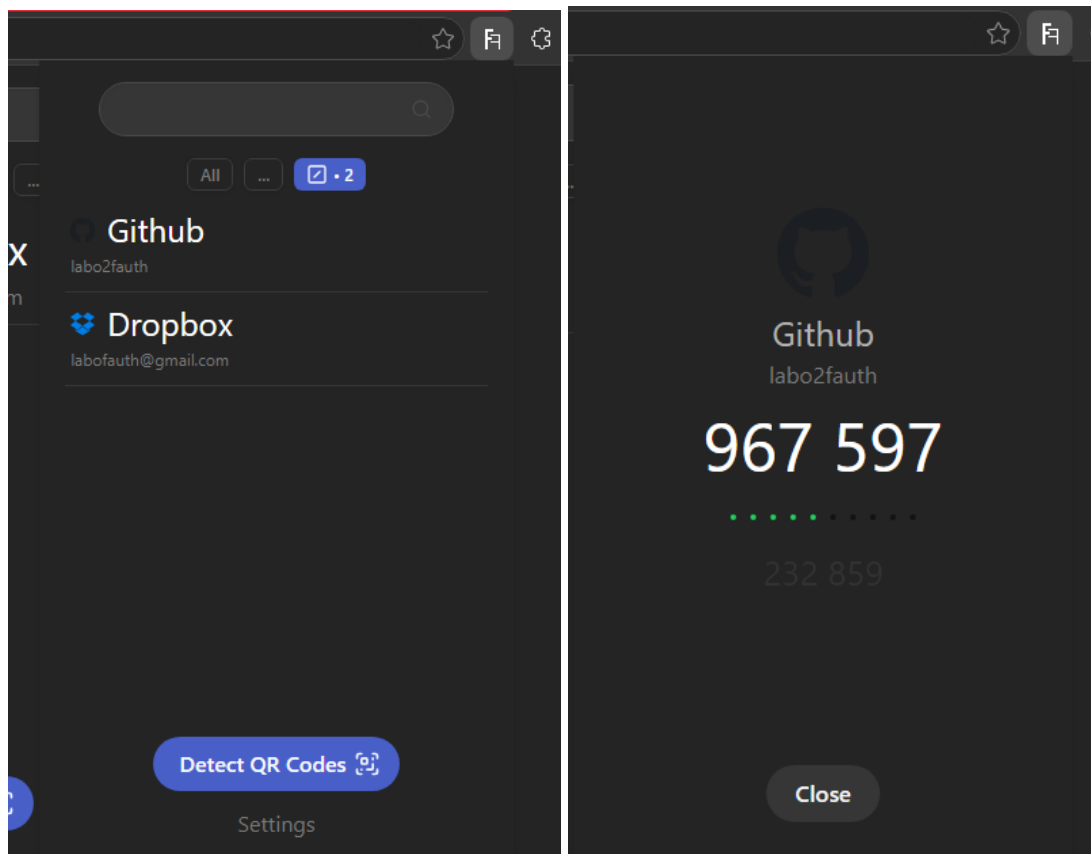
Generar un token en Settings → OAuth → Generate a new token



Una vez generado pegarlo en el recuadro de API token en la configuración de la extensión, como se ve en la imagen.



La extensión nos permite acceder a los códigos de las cuentas vinculadas y detectar un QR en caso de querer configurar una nueva.

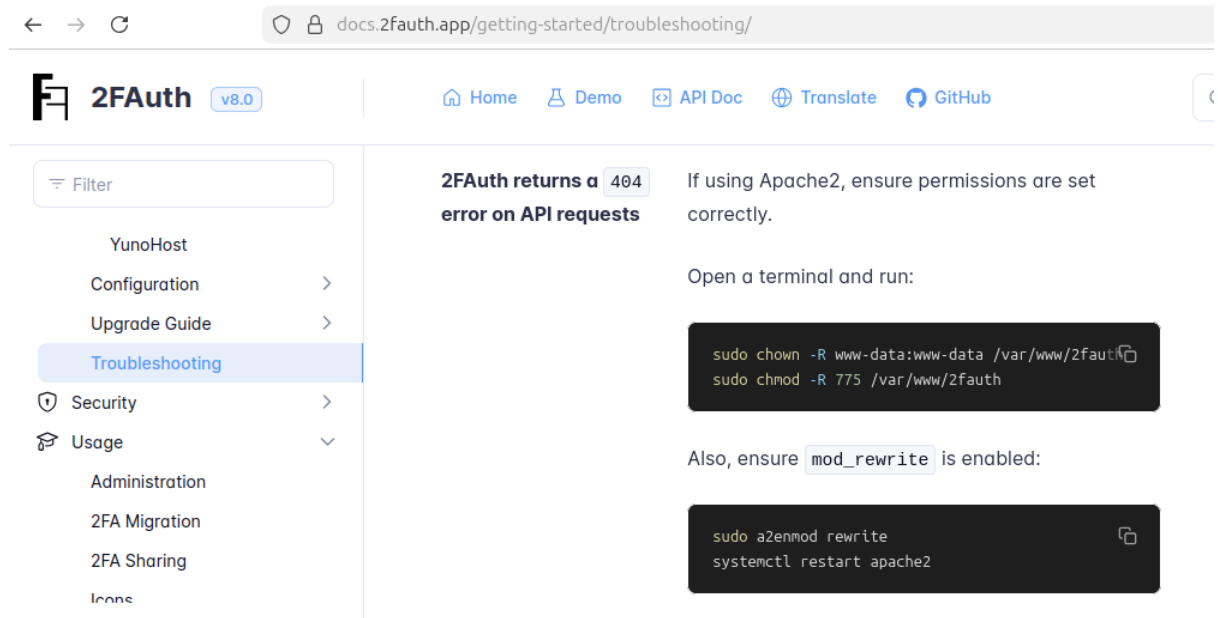


Se puede acceder a la demo para comprobar cómo funciona desde <https://demo.2fauth.app/>

Problemas y observaciones encontrados

- Permitir paquetes de cierto repositorio para instalar PHP >= 8.4 con apt install.
Para instalar PHP 8.4 fue necesario agregar el repositorio externo de Ondrej (ppa:ondrej/php) ya que los repositorios oficiales de Ubuntu no incluyen esta versión.
- Al intentar instalar las dependencias de Composer con **composer install --prefer-dist --no-scripts --no-dev** el proceso fallaba por falta de permisos que fueron otorgados para poder hacer la instalación correcta con **sudo chown -R user:user /var/www/2fauth**
- La instalación del instalador de composer requiere usar el hash publicado en <https://getcomposer.org/download/> al momento de descargar, ya que el hash cambia con cada nueva versión. Si se copia el hash de una guía desactualizada la verificación falla y el proceso se interrumpe.

- La instalación mediante el asistente CLI generó errores así que se optó por clonar el repositorio con git clone y realizar la configuración manualmente.
- Al levantar la aplicación si bien las páginas cargaban cuando se hacían llamadas a la API daba error 404. Siguiendo troubleshooting de la guía se solucionó:



Conclusión

El despliegue de 2FAuth en un servidor propio demostró ser una alternativa viable y funcional a las aplicaciones de autenticación de doble factor. Con este proyecto se logró instalar y configurar exitosamente todos los componentes del stack detallados por el profesor: Apache2 como servidor web, PHP con las extensiones requeridas, PostgreSQL como base de datos, envío de correos y extensión de navegador.

El proceso presentó algunas dificultades durante la instalación que requirieron un poco de investigación y algunos ajustes, sin embargo pudieron solucionarse.

Como resultado se obtuvo un sistema funcional, autoalojado y multiusuario que garantiza control total sobre los datos de autenticación, sin depender de servicios externos ni aplicaciones. La experiencia permitió aplicar conceptos de redes, administración de servidores Linux y configuración de servicios web en un caso de uso real.